Regularization by Denoising applied to non-linear traveltime tomography

by

Andres Alberto Ambros Vargas

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science
in
Geophysics

Department of Physics

University of Alberta

Abstract

The solution of an inverse problem such as the traveltime tomography requires a regularization function that constrains the solution and stabilizes the inversion. A traditional regularization method is the one of Tikhonov, which imposes restrictions on the solution such a small norm or smoothness. The recently published Regularization by Denoising from the signal processing field proposes to take advantage of the existing powerful denoising algorithms developed for the removal of Gaussian noise to regularize general inverse problems. In this work, we explore the application of this novelty technique for the linear and non-linear cases of the traveltime tomography problem and offer a comparison of its performance against the one from Tikhonov.

Acknowledgements

I want to thank my supervisor, Dr Mauricio Sacchi, for his time and attention and guidance on the development of this thesis.

I appreciate the time and advice of the members of my committee. Their advice and guidance helped in the development of this project.

I am also grateful to my colleagues from the Signal Analysis and Imaging Group (SAIG), for their advice, company and the learning that I had through their presentations and conversations.

I am grateful to my sponsor from México, Consejo Nacional de Ciencia y Tecnología, who provided me with the funding for my program.

I am grateful as well with the SAIG sponsors who funded part of my program.

I would like to say thank you to Landon Safron and Amsalu Anagaw who vastly helped me with the Wavefront Construction and Adaptive Weight Total Variation codes necessaries for this thesis.

Last but least, I am grateful to my friends and family, whose support allowed me to fulfil the program.

Contents

Appendices

List of Figures

CHAPTER 1

Introduction

1.1 Earth and physics

The discipline of physics encompasses the study of a vast number of subjects, ranging from the atoms to the galaxies. One of these branches combines with the field of geology yielding to the area of geophysics. This discipline has its areas of study, atmospheric sciences, geodynamics, seismology, exploration geophysics, among others (Fowler, 2004; Lowrie, 2007).

Exploration geophysics centers around measuring the physical properties of the earth, which later serve to create images of the subsurface. Such a task is possible due to the geological heterogeneities of the earth exhibiting differences in physical properties like density, electrical resistivity, or mechanical wave propagation velocities. A homogeneous Earth will be impervious to geophysical exploration methods (Claerbout, 1985). Therefore, these physical differences in the subsurface can be measured on the surface, allowing for imaging the earth's interior.

There is a wide variety of geophysical exploration methods, each one taking advantage of the contrast in different physical properties in order to create an image of the subsurface. Some of them are passive and focus on measuring potential fields like the earth's gravitational or magnetic fields (Telford et al., 1990). These measurements, in turn, allows mapping variations on density or magnetic susceptibility on the rocks in the subsurface. Other methods are active and require a stimulus to the earth, such as an electrical current, an electromagnetic field, or an explosive source (Telford et al., 1990; Claerbout, 1985). Then, it is possible to measure the earth's response and combine that information with the knowledge of the source to map the variation of physical properties underground, which are related to the local geology.

1.2 Seismic exploration

The images delivered by the geophysical exploration methods commonly serve for the location of natural resources. Seismic exploration, which uses mechanical waves to probe the subsurface, has a close bond with the exploration of oil and gas (Claerbout, 1985). This close bond is because, from the geophysical exploration methods, it is the one that provides the best resolution and most useful information for the hydrocarbon production sector (Sheriff and Geldart, 1995). Furthermore, the nature of seismic reflection allows for mapping a layered media and for identifying geological structures such as traps that can contain hydrocarbons.

The superior resolution of the seismic exploration method comes from the frequency range of the man-made seismic sources combined with the range of the seismic velocities on the rocks. The ratio of these properties leads to wavelengths ranging from a few dozens to a few hundred meters, which in turn makes possible for imaging structures about the same order of magnitude (Kallweit and Wood, 1982). The potential method, in contrast, usually resolves for structures with magnitudes of hundreds or thousands of meters due to the large wavelengths of the signals associated with these studies (Blakely, 1995).

One can make a distinction within seismic exploration for deep and shallow scenarios. On the one hand, deep-exploration is the primary tool for locating hydrocarbon reservoirs (Sheriff and Geldart, 1995). In this case, deep refers to a few kilometers, around 3 Km (U.S. Energy Information Administration, 2008), unlike global seismology, which images the earth's interior with structures of hundreds of kilometers wide. On the other hand, shallow exploration refers to the first few hundred meters below the surface and usually works in a support role for the deep-exploration techniques (Burger, 2006).

Seismic imaging of deep targets requires reflection seismology, which works with the reflections of seismic waves that originated on the surface by a seismic source (Claerbout, 1985). These reflections occur due to the different physical properties in the subsurface rocks, such as density and seismic velocity. The traveltimes of these reflected waves are recorded and used for creating a velocity model of the subsurface.

The physical heterogeneities mentioned afore give birth to seismic refractions, which, unlike its optical homonym, refer to the waves traveling in the interface of two media with different seismic velocities at the highest of these velocities (Sheriff, 2002). This phenomenon originates a whole family of waves who probe vertically downwards, then horizontally towards the receiver, and finally upwards to the surface, following Fermat's least-time principle. Shallow exploration exploits the traveltimes of the refracted waves and the direct arrivals, which are the waves coming directly from the source, in order to create a velocity map of the near-surface (Yilmaz, 2001).

1.2.1 Seismic tomography

One can organize the data from a standard seismic data acquisition campaign in different manners. Shot gathers arrange the seismic data as a set of several receivers and a single source, each receiver recording a seismogram. The direct arrivals and refractions appear on a shot gather as the first events, coining the term first-break arrivals (Sheriff, 2002).

The term seismic tomography refers to the images created employing the first-break arrivals (Claerbout, 1992; Sheriff, 2002). Etymologically, the term means to create a slice of the interior of an object using waves. The goal is for the waves to traverse the target carrying in the process information of its interior, in this case, the seismic velocity.

One can think about the physical properties of the object, varying as a function of the space. Nevertheless, computational limitations require the discretization of the target's interior in small finite cells in which the properties are constant. Thus, tomography is the reconstruction of a function from line integrals through the function (Claerbout, 1992).

In order to create an image from the cells, is it necessary to know the wave's trajectory within the target, i.e., the raypaths. In seismic exploration, there are usually two scenarios to consider. The first one is the cross-well. Here the sources and receivers are located in wells, and the target is the slice of earth between them. The second one is the first-break tomography, where sources and receivers are placed on the surface, and the area to map is below them (Claerbout, 1985).

One can describe the cross-well scenario as a linear problem and describe the raypaths as straight lines connecting the source-receivers pairs. This simplification of the problem will lead to low-contrast images but allow for a convenient and amiable first approach to the seismic tomography problem (Aster, 2013). The source-receiver geometry of the second case renders impossible a linear assumption and the consequent non-linear approach that explains the non-linear trajectory of the raypaths.

Here it is necessary to make a distinction between the refraction method and turning ray tomography. While both work with the first-breaks of the seismic records, the first one is more rigid on its assumptions. Sheriff (2002) defines a refraction survey as the mapping of geological structures through head waves. These are waves that incise on an interface at its critical angle, then travel parallel to that interface, and finally exit it at the same angle (Sheriff and Geldart, 1995; Yilmaz, 2001).

Therefore, for the refraction method to work, one requires interfaces clearly defined by sharp contrasts in velocity. These interfaces, also known as refractors, can be undulated, and one can image them the General Reciprocal Method (GRM) (Palmer, 1980, 1981). Tomography, on the other hand, do not require this assumption, since it works with continuously refracted ray paths from turning waves (Sheriff, 2002). It, however, requires a ray tracing scheme to keep track of the ray paths (White, 1989).

Tomography has the advantage of overcoming the problem of poorly defined refractors, as well as lateral variations of velocities (Bell et al., 1994; Stefani, 1995). One can think that while the refraction method solves for layers, the tomography one solves for regions. Zhu et al. (1992) explains that the refraction method assumes constant velocity layers with null vertical velocity gradient within them, which do not reflect the non-linear behavior of the first-break arrivals observed in some areas. Furthermore, while refraction tomography struggles to image velocity inversions, i.e., a low-velocity layer in the sallow region, turning ray tomography will generate an image for any region cross by a raypath (Zhu et al., 1992).

The first-break tomography is a classical geophysical problem in which new techniques can be tested and compared to a well-studied benchmark. Additionally, it has the benefit of being a method widely used in oil and gas exploration (Sheriff and Geldart, 1995), as well as in other smaller areas like geotechnics (Burger, 2006).

In hydrocarbon exploration, first arrival tomography can help to define the weathered layer, which is a low-velocity zone in the near-surface, which causes errors on the reflections traveltimes, impoverishing the signal (Zhu et al., 1992; Yilmaz, 2001). Knowing the structure and seismic velocity of this layer permits counteracting these adverse effects. Cross-well tomography is a more straightforward case of the first-break one. This reduction in complexity allows one to test new methods on a controlled scenario before jumping up to more realistic cases.

1.3 Seismic imaging or the inversion problem

Seismic tomography is an example of an inverse problem (Claerbout, 1992). Inverse theory centers on retrieving a set of physical characteristics in the form of parameters from a set of measurements of observations (Menke, 2018; Tarantola, 2005; Aster, 2013). The study of inverse theory can have a continuous approach, where the outcome in our case would be a function describing the distribution of velocities on space, or discretely, where the result would be a set of numbers describing the seismic velocity on a cell of the discretized model.

The study of inverse theory requires an understanding of the concepts of forward and inverse problem/modeling. In the seismic tomography context, forward modeling is the procedure used to compute synthetic traveltimes, while the inverse is the calculation of a velocity model from those traveltimes.

1.3.1 Regularization

Finding the solution of an inverse problem presents difficulties due to their sensitivity, stability, and the non-uniqueness or sometimes non-existence of the solution (Menke, 2018; Tarantola, 2005). These characteristics cause inverse problems to require a regularization to stabilize the inversion process, which has the inherent effect of constraining the space of solutions. The regularization directs the inversion process towards a particular parameter model.

Introducing a regularization term forces one to decide how much importance give to this term, too much, and honoring the observed data becomes irrelevant, too little, and the inversion will suffer from instability due to rank deficiency (Aster, 2013). In statistics, this is referred to as the bias-variance trade-off (Dorugade and Kashid, 2010).

On the one hand, a small trade-off parameter will lead to a biased parameter model that will not reproduce the observed traveltimes. On the other hand, a large value in the trade-off parameter will yield a parameter model with a high variance that reproduces the observed data but possess unrealistic velocity variations. The behavior described in the first case is also known as underfitting, while the second is referred to as over-fitting (Johansen, 1997; Aster, 2013).

A common regularization term is the one of Tikhonov and Arsenin (1977), which minimizes the L_2 norm of the parameter model or its first or second derivatives. This restriction allows one to find the smallest, flattest, or smoothest solution, respectively. As mentioned afore, one should note here that by picking a regularization means also deciding which kind of features one wishes to induce in the solution model.

A new approach for inverse problems regularization was the one by Venkatakrishnan et al. (2013), who proposed incorporating denoising algorithms in this process. Chan (2016) utilized this method to implicitly induce features of interest on the inverse problem of image restoration while, Chan et al. (2017) focus on proving the convergence of the method. Finally, Romano et al. (2017) extended this technique and explicitly incorporated the denoiser as part of the cost function.

Many of the breakthroughs in several areas, including tomography, are done through developments in inverse problem-solving. This area is a general field, such as mathematics, which has applications in other fields. Such is the case of the developments on regularization terms (Tikhonov and Arsenin, 1977; Rudin et al., 1992), cost functions (Zhang and Toksöz, 1998; Elad and Aharon, 2006) or inversion methods (Kaczmarz, 1937; Marquardt, 1963). We will deepen these concepts in the next chapter.

1.4 Denoisers

Researchers on signal processing like Chatterjee and Milanfar (2010), and Buades et al. (2011) explain that noise addition on digital signals, like photographies or seismic traces, occurs from the moment of acquisition. Instrument blurring, digitalization, or unexpected signal sources during the signal capture are common causes of signal noise. Therefore, denoising has been a subject of interest from the early stages of image processing.

One can think of a noisy image as one whose pixels have been deviated from its original value. The denoising algorithm has the task of reversing this process (Buades et al., 2011). The kind of deviation depends on the source of the noise and determines the denoising technique. Gaussian noise removal requires a different treatment of the erratic-noise case (Claerbout and Muir, 1973; Scales and Gersztenkorn, 1988).

Simple denoisers like the mean assume that the clean value of the pixel should be similar to the nearby ones, and thus uses them to determine the most likely value to the pixel under denoising (Buades et al., 2011). A Gaussian filter works the same way, but it assigns weights to the mean calculation based on the distance to the pixel under denoising. There is a wide variety of denoising methods. An effective denoiser must be capable of removing the noise while minimizing the distortion of the original features of the images.

1.5 The scope of this thesis

In this work, we aim to explore the capabilities of the Regularization by Denoising (RED) for the linear and non-linear traveltime tomography problem and to compare them against the conventional regularization technique of Tikhonov (Tikhonov and Arsenin, 1977). We are particularly interested in the non-linear problem because, as far as we are aware, the application of RED for non-linear cases has not been explored yet.

We are also interested in analyzing the performance of three denoising routines. Romano et al. (2017) claim that RED can exploit the capabilities of powerful denoising algorithms like Non-Local Means (Buades et al., 2011) to solve general inversion problems, however, they also claim that their method should be able to perform with arbitrary denoisers like the median filter (Tukey, 1977; Huang et al., 1979). Hence, we want to compare the results of working with a simple and sophisticated denoising engine. Additionally, we want to observe if the edge-preserving behavior of denoising algorithms like Total Variation (TV) (Rudin et al., 1992) or Adaptive Weight Total Variation (AWTV) (Liu et al., 2012) when used as regularization terms, persists on the retrieved velocity model.

Regularization by denoising is a general method for stabilizing inverse problems that aspire to provide good quality results with a straightforward implementation. We chose to test the applicability of this method for near-surface imaging because it can play a significant role in seismic imaging due to the large time shifts that might be induced by the shallow low velocities. We expect that the implementation of RED improves the results of the Tikhonov regularization and yield to a more straightforward implementation.

1.6 Thesis outline

The structure of this thesis is as follows:

- Chapter 1 tours the reader from a general background of physics to the particularities of seismic exploration. It then introduces the difficulties of solving inverse problems and finalizes establishing the contribution of this project: the application of a novelty regularization term on the inverse problem of seismic tomography.
- Chapter 2 first studies in detail the subjects from discrete inverse theory relevant to understand this project. Then, it deepens into regularization techniques, the classical one of Tikhonov, and the novelty of Regularization by Denoising. It later explores the particularities of a few numerical optimization methods and provides some brief comments on the Bayesian approach to inverse problems.

- Chapter 3 explores the application of the RED, with three different denoisers, to the cross-well tomography case, and then it compares the results to the ones obtained by conventional Tikhonov. It deepens on the details of applying these regularization techniques, like determining the trade-off parameter, the numerical implementation of RED, and a description of its denoiser functions.

- Chapter 4 describes the complexities of dealing with non-linear inverse problems. It starts by detailing the computation of the forward problem through the WaveFront Construction method, then explains the linearization process of the problem. Later, it deepens into some numerical implementation difficulties like step size and initial model determination and closes by comparing the results Regularization by Denoising and Tikhonov regularization on the first-break tomography problem.

- Chapter 5 summarizes the most significant findings of this research and provides an analysis of the performance of RED when compared to Tikhonov Regularization for both the linear and non-linear cases.

CHAPTER 2

Discrete inverse theory and regularization

In this chapter, we will give an overview of the subjects from inverse theory relevant to the understanding of this research project. This text is by no means an exhaustive guide on the subject, the interested reader can refer to Tarantola (2005); Aster (2013) and Menke (2018).

2.1 The forward problem

The study of physics addresses a broad spectrum of phenomena such as gravitational attraction, wave propagation, among others. These phenomena have causes and effects. The first ones tend to be abstract while the latter ones tangible. These causes and effects are, in turn, related by mathematical models, such as the Ray Tracing Theory and Wave Equation, which can quantify the traveltimes of the seismic waves through a medium.

The example in the previous paragraph, allows one to grasp the abstract nature of the causes of a physical phenomenon when compared with its effects. While we can perceive the effects of seismic waves, for example, during an earthquake, we are unable to ”feel” the seismic velocity distribution in the subsurface. Furthermore, we can measure the effects of the seismic waves through geophones, but determining the seismic velocities that govern the effects observed on the surface is a more complex problem because we cannot measure them directly.

Inverse theory addresses problems like the one stated in the example afore. To start, we require to define three key concepts: the observed or measured data, the forward model, and the parameter model. The importance of inverse theory on exploration geophysics resides on its capability to make indirect inferences of the subsurface geology from physical measurements taken on the surface.

The concept of data is easy to grasp because it is tangible, and we can measure it directly. It corresponds to the effects of a phenomenon. In the seismic tomography context, it corresponds to the traveltimes. Typically, one will have several data points, which one can organize in a vector fashion that, from now on, will be referred to as **d**.

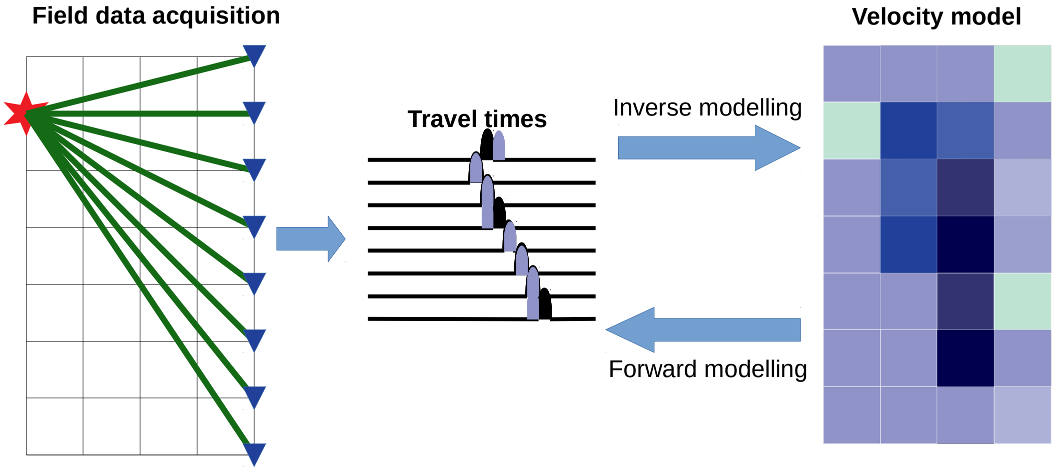

Figure 2.1: Sketch of the relationship among the traveltimes **d**, acquired on the field, the forward modelling, and the inverse modelling that leads to the velocity model **m**.

The forward model corresponds to the many physics quantitative theories developed through Human History, such as Newton's Law, Maxwell's Law, or the Wave Equation. These mathematical tools depend on a set of parameters and allow for the prediction of new data points **d**. In inverse theory, the parameters governing the outcome of the forward modelling are denominated **m**. These elements correspond to the causes of the phenomenon and are usually abstract: mass, electromagnetic properties, density, seismic velocity, among others. The sketch on Figure 2.1 describes the relationships of the three main inverse theory elements.

Figure 2.1 also introduces us to the notion that the parameter model governing the observations **d** can have more than one dimension. A 1D parameter model would consist of a single row or column of the matrix at the right side of Figure 2.1. Other examples of 1D parameter models are reflectivity series, resistivity profiles, among others.

In this project, we work with 2D parameter models in the form of discrete seismic velocity sections. It is convenient for us to apply a lexicographic rule to reshape them into column vectors. One can use a matrix **M**, with Mz rows and Mx columns, to represent a discretized section of the earth, just as described in Figure 2.1,

$$\mathbf{M} = \begin{pmatrix} m_{11} & m_{12} & \cdots & m_{1M_x} \\ m_{21} & m_{22} & \cdots & m_{2M_x} \\ \vdots & \vdots & \ddots & \vdots \\ m_{M_z1} & m_{M_z2} & \cdots & m_{M_zM_x} \end{pmatrix}. \tag{2.1}$$

Then, this matrix is rearranged into a column vector in the following way

$$\mathbf{m} = \begin{pmatrix} m_{11} & m_{21} & \cdots & m_{M_z1} & m_{12} & \cdots & m_{M_z2} & m_{1M_x} & \cdots & m_{M_zM_x} \end{pmatrix}^T. \tag{2.2}$$

This column vector **m**, has a length of $M = M_z \times M_x$.

We have established that the forward modelling relates the parameter models, the causes, and the data, the effects. This means that if we have N measurements organized in the vector **d**, each one of them is a function of the M elements of the vector **m**, i.e. $d = g(\mathbf{m})$. If the function g is linear, we will have a $N \times M$ system of equations that can we can write in a matrix fashion,

$$\mathbf{d} = \mathbf{G}\mathbf{m} + \mathbf{n}. \tag{2.3}$$

Notice the addition of the vector **n**, which represents the inherent noise of each measurement. The matrix **G** is sometimes referred to as kernel.

2.2 The inverse problem

Equation 2.3 shows that solving the forward problem is simple when this is linear. One only needs to substitute a parameter model **m** and apply the known mathematical model. The calculations might be complex and take time, but the approach is straightforward. Solving the inverse problem, however, implies retrieving **m** from the observations **d**, and as we will see in this section, it is a more interesting problem.

One can tentatively start by considering that the solution of the inverse problem will consist of finding the inverse of the matrix **G**. Nonetheless, for $\mathbf{G}^{-1}$ to exist, **G** must be square and full rank, and as we will explain in the following subsections, that is rarely the case. Additionally, the noise in the measurements may make it impossible to compute the inverse. Furthermore, an analytical expression for the function g might not exist, and even if it does, it might be unstable or lead to a non-unique solution. For these reasons, solving the inverse problem is more complicated than solving the forward one.

Authors like Tarantola (2005) and Menke (2018) provide an extensive analysis on inverse theory. We will take some of their ideas to develop our work. One of the first steps when dealing with an inverse problem is to identify the characteristics of it in terms of the number of data points N and parameters M. The relationship between these values can help us to create a rough classification of inverse problems. A classic example of inverse problems is linear regression, which consists of estimating the slope and intercept of the straight line that fits the best to a set of experimental observations. We will take this case to explain the strategies for solving an inverse problem.

2.2.1 Even-determined problems

The even-determined problem is the simplest case and occurs when there is an equal number of data-points N and parameters M. This case is ideal and has a unique solution which we can solve by

$$\mathbf{m}_{ED} = \mathbf{G}^{-1}\mathbf{d}. \tag{2.4}$$

The inverse of the matrix $\mathbf{G}$ exists in this case because it is square and the $\mathbf{d}$ is noise free.

We illustrate this case in Figure 2.3a, where we aim to find the slope and intercept of a straight line crossing two points. It is evident that there is a single solution for the slope and y-intercept and that finding the slope m_1 and the intercept m_2 is trivial.

2.2.2 Over-determined problems

A more interest case for inverse problem occurs when we have more observations N than parameters M, which leads to having more than one solution. This kind of problem is known as over-determined and is equivalent to fitting a straight line to three (or more) non-colinear points, as in the classic linear regression problem. We depict this with an example in Figure 2.3b.

Since there is no single solution, the best one can aspire to is to select the most suitable one. This quest, however, necessitates the creation of a function that qualifies the suitability of the solutions. We should mention that this is a fundamental step since the metric used to measure the properness of a solution will lead to different solutions identified as the "best one".

A common approach to determine the best possible solution is the least-squares (LS) method. LS uses the residuals, which are defined as the differences between the observations **d** and the forward model **Gm**, to quantify the quality of a solution. The idea is to find the parameters of the straight line ”closest” to all the data points, i.e., the one that has the smallest residuals. $\mathbf{d} - \mathbf{Gm}$ can have a negative or positive sign, to account for this the method works with the sum of the squares. One could envision working with absolute values instead, but this kind of function is cumbersome when differentiating.

Now that we have a way to asses the quality of the possible solutions, we can express it as a function

$$\phi_{LS}(\mathbf{m}) = ||\mathbf{d} - \mathbf{Gm}||_2^2. \tag{2.5}$$

This is called a cost or objective function, and it is an important concept in inverse theory because finding the minimum of the objective function is equivalent to find the solution of the inverse problem. Notice that the metric used corresponds to the L_2 norm.

The L_n norm

In the context of linear algebra, a norm is a function that assigns a value greater or equal to zero to any vector, intending to quantify its length (Grossman, 1980). A common norm in the signal processing and inverse problem context is the L_n norm which establishes that norm of any vector $\mathbf{u} = (u_1, u_2, \ldots, u_N)$ can be calculated as

$$\|\mathbf{u}\|_n = \left[\sum_i^N |u_i|^n\right]^{\frac{1}{n}}. \tag{2.6}$$

A particular case of the L_n norm occurs when $n = 2$. This consideration leads to the Euclidian norm, which in in simple terms is the way we commonly compute distances, i.e. the length one can measure with a rule.

Effect of a L_n norm in the cost function

Selecting the L_2 norm to quantify the quality of the residuals, also known as data misfit, has more profound implications than it might seem. One of them is that we want our objective function to be smooth and convex, which is a property that we will exploit ahead. Another one is that it assumes that the noise $\mathbf{n}$ from equation 2.3 has a Gaussian distribution. We dive into this second subject in the Bayesian approach section at the end of this chapter.

It is possible, however, to use other norms from the L_n family to define the cost function. As we mentioned before, such selection will change what is considered the best solution. One can roughly say that higher-order norms imply a shorter tailed distribution on the noise and thus more accurate data, while a lower order norm suggests more spread one and hence less certainty about the data (Menke, 2018).

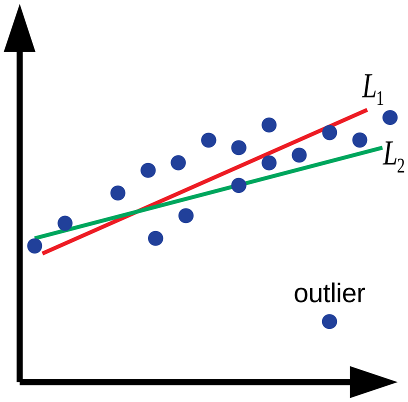

Figure 2.2: Two lines are fitting a set of data points. The red one uses the L_1 norm assuming less certainty on the data distribution and thus largely ignoring the outlier. The green one, on the other hand, expects the data to be more accurate and gives the same importance to all data points. Adapted from (Menke, 2018).

The example of the effect of the L_n norm on the retrieved parameters, which appears in Figure 2.2, shows that those higher-order norms are more affected by outliers. This effect makes sense when we think that they work under the assumption of high-accuracy data. Lower norms, on the other hand, expect the data to be less accurate, making the influence of outliers to diminish.

Claerbout and Muir (1973) noted the resistance to outliers in parameter estimation achieved by the use of low-order norms. Thus, they advocated for the usage of L_1 to deal with non-Gaussian noise but warns about the increase in computing time. The subject is then retaken by Scales and Gersztenkorn (1988), who deepens into the topic once the computation power of home computers has improved. Later applications are described by Bube and Langan (1997), who uses the L_1 norm to deal with erratic-noise in the seismic tomography problem.

The least-squares solution

Minimization of equation 2.5 will lead us to the ”best” solution, assuming Gaussian noise, from the available ones in an over-determined problem. To find the minimum of the cost function, one can exploit the fact that the objective function is convex and smooth and apply the first derivative criterion. To differentiate equation 2.5 with respect to $\mathbf{m}$[1]one should first consider that $||\mathbf{d}-\mathbf{Gm}||_2^2 = \left[(\mathbf{d}-\mathbf{Gm})^T(\mathbf{d}-\mathbf{Gm})\right]$, so we have

$$\frac{\partial\phi_{LS}(\mathbf{m})}{\partial\mathbf{m}} = \frac{\partial}{\partial\mathbf{m}}\left[(\mathbf{d}-\mathbf{Gm})^T(\mathbf{d}-\mathbf{Gm})\right]. \tag{2.7}$$

When computing the derivative of the right-hand side of equation: 2.7 we obtain

$$\frac{\partial\phi_{LS}(\mathbf{m})}{\partial\mathbf{m}} = -\mathbf{G}^T(\mathbf{d}-\mathbf{Gm}) + (\mathbf{d}-\mathbf{Gm})^T(-\mathbf{G}). \tag{2.8}$$

Reordering the right-hand side gives us:

$$\frac{\partial\phi_{LS}(\mathbf{m})}{\partial\mathbf{m}} = -\mathbf{G}^T\mathbf{d} + \mathbf{G}^T\mathbf{Gm} - \mathbf{G}^T(\mathbf{d}-\mathbf{Gm}), \tag{2.9}$$

[1] We use vector derivatives. Petersen and Pedersen (2012) provide an extensive repository for matrices relationships, derivatives, and identities.

$$\frac{\partial \phi_{LS}(\mathbf{m})}{\partial \mathbf{m}} = -\mathbf{G}^T\mathbf{d} + \mathbf{G}^T\mathbf{G}\mathbf{m} - \mathbf{G}^T(\mathbf{d} - \mathbf{G}\mathbf{m}), \tag{2.10}$$

and finally

$$\frac{\partial \phi_{LS}(\mathbf{m})}{\partial \mathbf{m}} = 2\mathbf{G}^T\mathbf{G}\mathbf{m} - 2\mathbf{G}^T\mathbf{d}. \tag{2.11}$$

The minimum of the cost function occurs when $\nabla_{\mathbf{m}}\phi_{LS} = 0$, applying this condition to equation 2.11 leaves us with the best solution for the over-determined problem,

$$\mathbf{m}_{LS} = [\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\mathbf{d}. \tag{2.12}$$

Equation 2.12 is commonly known as the LS solution and will be a recurrent one in this work.

2.2.3 Under-determined problems

When there are more parameters than observations, $N < M$, the problem is called under-determined and cannot be solved by the cost function in equation 2.12 alone. Following our linear regression example, this is equivalent to trying to fit a line to a single data point, as depicted in Figure 2.3c. There is no way to achieve it. However, similarly to the over-determined case, we can do the second best thing and find the "least-bad" solution.

A strategy to solve this kind of problem is to constrain the space of solutions. We can do this by considering some a priori information, i.e., attempt to simplify the problem by applying beforehand knowledge or assumptions. For example, perhaps we know that the straight line in 2.3c crosses the origin, or that its slope is equal to zero. In both of these cases, the assumption simplifies the problem and allows us to find a solution.

The formal way for introducing these assumptions and a priori information into the cost function is a regularization term. This term is an additional algebraic term in the right-hand side of equation 2.12 that constraints the parameter model. A well-known regularization term is the one that induces the smallest possible magnitude for the solution, which leads to the minimum norm solution. In our linear regression example, this means that we are looking for the solution whose sum of squares of slope and intercept is as small as possible. The cost function for the minimum norm solution would be,

$$\phi_{MN}(\mathbf{m}) = ||\mathbf{d} - \mathbf{G}\mathbf{m}||_2^2 + \mu||\mathbf{m}||_2^2. \tag{2.13}$$

The μ variable in equation 2.17 is called the trade-off parameter, and its value represents the relative importance given to each part of the cost function. Setting the cost function to zero gives us the LS solution while increasing it too much makes the observed data irrelevant and will result in a model with a small norm, but that does not fit the data. We should then select a μ value that minimizes both terms as best as possible.

The regularization term in equation 2.13 is smooth and convex by design. Once more, this facilitates the differentiation and hence the minimization of the cost function. Notice that for minimizing equation 2.13, we only have to differentiate the regularization term and add it to the right-hand side of equation 2.11 which describes the derivative of the data misfit term.

The derivative of the minimum norm regularization term is

$$\frac{\partial||\mathbf{m}||_2^2}{\partial\mathbf{m}} = \frac{\partial}{\partial\mathbf{m}}\left(\mathbf{m}^T\mathbf{m}\right) = 2\mathbf{m}. \tag{2.14}$$

We can write the gradient of ϕ_{MN} as

$$\frac{\partial\phi_{MN}(\mathbf{m})}{\partial\mathbf{m}} = \frac{\partial\phi_{LS}(\mathbf{m})}{\partial\mathbf{m}} + \mu\frac{\partial||\mathbf{m}||_2^2}{\partial\mathbf{m}}. \tag{2.15}$$

If we substitute equations 2.11 and 2.14 in 2.15 we obtain the gradient of the minimum norm objective function:

$$\frac{\partial\phi_{MN}(\mathbf{m})}{\partial\mathbf{m}} = 2\mathbf{G}^T\mathbf{G}\mathbf{m} - 2\mathbf{G}^T\mathbf{d} + 2\mu\mathbf{m}. \tag{2.16}$$

Setting the gradient to zero and solving for $\mathbf{m}$ finally leave us with the minimum-norm solution for the undetermined problem,

$$\mathbf{m}_{MN} = [\mathbf{G}^T\mathbf{G} + \mu\mathbf{I}]^{-1}\mathbf{G}^T\mathbf{d}, \tag{2.17}$$

where $\mathbf{I}$ is the identity matrix. Equation 2.17 is also known as the Damped LS solution (Aster, 2013).

Aster (2013) and Menke (2018) explain that the values added in the diagonal of $\mathbf{G}^T\mathbf{G}$ by $\mu\mathbf{I}$ in equation 2.17 help to avoid singularity, and thus guarantee the existence of the inverse matrix. From linear algebra, we know that a singular matrix has eigenvalues equal to zero. Thus, the regularization term stabilizes the problem by adding a small quantity in the diagonal elements and stabilizing the Characteristic Equation (Grossman, 1980).

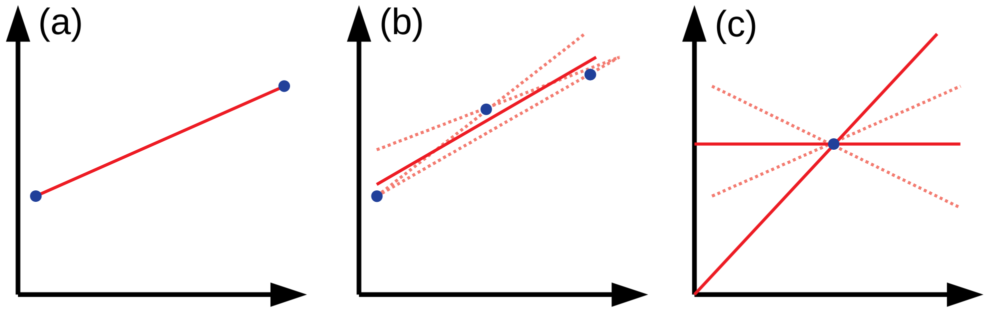

Figure 2.3: (a) Even-determined example, $N = M$, a single solution. (b) Over-determined case. The solid line represents the LS solution, while the dashed ones correspond to other possible solutions. (c) Under-determined case. The dashed pale lines represent the infinite number of lines that can be fit. The solid ones represent the solutions constrained to be either horizontal or to pass by the origin.

The classification provided above is the one published by Menke (2018). We chose it because it is intuitive and fits the scope of this thesis. Nonetheless, Aster (2013) offers a slightly different view by analyzing inverse problems in terms of the Singular Value Decomposition (SVD). In the end, he still relies on using the M and N relationship to classify the inverse problems. However, he goes into more detail on the behaviour of the eigenvectors in each case.

Ill-posed problems

Hansen (1998) explains how the so-called Hamarad's conditions describe a well-posed problem. The first one is that a solution exists. The second is that there is only one, and the third one that such a solution is continuous. Most geophysics inverse problems do not fulfil these properties and are hence called ill-posed problems. The solution of ill-posed problems requires regularization for stabilizing them. Therefore, the regularization term has two simultaneous roles. On the one hand, it induces previously known features on the solution, and on the other hand, it stabilizes the problem to deal with the ill-posedness.

One can also describe ill-posedness in terms of singular values. Aster (2013) sustains that there is a connection between the condition number of the matrix $\mathbf{G}$ and the decay rate of its singular values. In rough terms, the steeper the drop, the more ill-conditioned the problem is. Moreover, the discretization of $\mathbf{m}$ also affects the singular values decay rate. The finer the discretization, the more poorly conditioned the problem becomes.

2.3 Regularization of inverse problems

As we have seen, solving an inverse problem requires a regularization. This term in the cost function will both constrain the space of solutions and stabilize the problem. In the upcoming subsections, we will talk about two regularization techniques: the Tikhonov regularization and the Regularization by Denoising.

2.3.1 Tikhonov regularization

We introduced the minimum norm solution when discussing under-determined problems in the previous section. This constraint appeared as a regularization term in the cost function, which minimized the norm of $\mathbf{m}$. This regularization is a particular case of what it is known as the Tikhonov Regularization

(Tikhonov and Arsenin, 1977), whose general cost function is

$$\phi_{TK}(\mathbf{m}) = ||\mathbf{d} - \mathbf{Gm}||_2^2 + \mu||\mathbf{Rm}||_2^2. \tag{2.18}$$

Where $\mathbf{R}$ is a matrix whose value can change to induce different behaviors on $\mathbf{m}$. Notice that if $\mathbf{R} = \mathbf{I}$ we obtain the minimum norm cost function (equation 2.13) hence that it is also known as the Zeroth-Order Thikhonov or Quadratic Regularization.

Two of the features one might be keen to induce are flatness and smoothness. The first one refers to a parameter model in which the adjacent parameters have small differences, while the second one will retrieve a model whose derivatives small. The flat solutions require the application of a discrete differentiation operator which can be defined as

$$\mathbf{L}_1 = \begin{pmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{pmatrix}. \tag{2.19}$$

For the smooth solution one needs to use an operator that approximates the second derivative

$$\mathbf{L}_2 = \begin{pmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \end{pmatrix}. \tag{2.20}$$

Notice that the operators $\mathbf{L}_1$ and $\mathbf{L}_2$ are designed to work on the matrix form of vector $\mathbf{m}$, i.e. matrix $\mathbf{M}$ from equation 2.1. For this operator to work with the equations presented in this document, it is necessary to transform them into $\mathbf{D}_1$ and $\mathbf{D}_2$. The process is simple and similar to the lexicographic rule at the beginning of the chapter, is long, so we defer it to Appendix A.

Another consideration is that for a 2D parameter model, the application of the flatness and smoothness operator requires a more detailed explanation. We decided to include that explanation in Appendix A.

Substituting $\mathbf{R} = \mathbf{D}_1$ will result in the flat or First-Order Tikhonov Regularization solution while making $\mathbf{R} = \mathbf{D}_2$ will retrieve the smooth, or Second-Order Tikhonov Regularization one. Regardless of the value of $\mathbf{R}$, the derivative with respect to $\mathbf{m}$ would be

$$\frac{\partial \phi_{TK}}{\partial \mathbf{m}} = 2\mathbf{G}^T\mathbf{G}\mathbf{m} - 2\mathbf{G}^T\mathbf{d} + 2\mu\mathbf{R}^T\mathbf{R}\mathbf{m}. \tag{2.21}$$

The minimum of the objective function occurs when the gradient is null, hence, the estimated parameter model is

$$\mathbf{m}_{TK} = [\mathbf{G}^T\mathbf{G} + \mu\mathbf{R}^T\mathbf{R}]^{-1}\mathbf{G}^T\mathbf{d}. \tag{2.22}$$

The Tikhonov regularization is a well-known and widely used one. In this sense, it can be considered conventional, and thus we will use it as a benchmark for a newly developed Regularization by Denoising (RED).

2.3.2 Additional regularizations

There exist more regularization terms, which allow inducing different behaviours in the solution model. For example, one may want $\mathbf{m}$ to stay as close as possible to a given reference model $\mathbf{m}_0$ and then change the objective function from 2.18 to

$$\phi(\mathbf{m}) = ||\mathbf{d} - \mathbf{G}\mathbf{m}||_2^2 + \mu||\left(\mathbf{m} - \mathbf{m}_0\right)||_2^2. \tag{2.23}$$

$\mathbf{Hm} = \mathbf{h}$ regularization

Menke (2018) suggests that a family of constrains of the form $\mathbf{Hm} - \mathbf{h}$ are easy to implement and lead to the cost function

$$\phi(\mathbf{m}) = ||\mathbf{d} - \mathbf{Gm}||_2^2 + \mu \left(\mathbf{Hm} - \mathbf{h}\right). \tag{2.24}$$

Examples of this type of constrain include, but are not limited to, keeping the mean of the parameters constant and fixing the value of a parameter. The first one is achieved by making $\mathbf{H} = \frac{1}{M}[1, 1, \ldots, 1]$ and $\mathbf{h} = [h]$, while the second one requires $\mathbf{H} = [0, \ldots, 0, 1, 0 \ldots, 0]$ and $\mathbf{h} = [h]$. Notice that in these examples $\mathbf{H}$ is a row vector that when multiplied by $\mathbf{m}$ results in a scalar.

It is also possible to regularize the inverse problem by imposing inequality constraints on the solution. This is handy when one knows beforehand that the elements of $\mathbf{m}$ are positive or bounded to a range. These problems can be stated as

$$||\mathbf{d} - \mathbf{Gm}||_2^2 \text{ subject to } \mathbf{m} \leq \mathbf{0}, \tag{2.25}$$

and

$$||\mathbf{d} - \mathbf{Gm}||_2^2 \text{ subject to } \mathbf{l} \leq \mathbf{m}. \leq \mathbf{u} \tag{2.26}$$

Equation 2.25 is known as Non-Negative Least-Squares (NNLS) while equation 2.26 is named Bounded-Variable Least-Squares (BVLS) (Lawson and Hanson, 1995).

L_1 regularization

An alternative of the Tikhonov Regularization is the L_1 regularization, stated as

$$\phi(\mathbf{m}) = ||\mathbf{d} - \mathbf{Gm}||_2^2 + \mu||\mathbf{m}||_1. \tag{2.27}$$

This constraint on the parameter model induces sparsity, which means that we want as many elements of $\mathbf{m}$ as possible to be equal to 0. Applications of this method are, for example the recovery of reflectivity series through deconvolution from seismic records and signal reconstruction (Candès et al., 2006).

Finding the minimum of the objective function in equation 2.27 requires complex methods like the Iteratively Reweighted Least-Squares (IRLS). This is a long-time method attributed to Beaton and Tukey (1974), with applications ranging from traveltime tomography (Scales et al., 1988), to Compressed Sensing (Daubechies et al., 2008). Aster (2013) offers an amiable explanation of the method. Alternatives to the IRLS are the Iterative Soft Thresholding Algorithm (ISTA) (Daubechies et al., 2003) and the Fast Iterative Soft Thresholding Algorithm (FISTA) (Beck and Teboulle, 2009).

Total Variation

Another interesting regularization option is the Total Variation (TV) (Rudin et al., 1992). Unlike First and Second-Order Tikhonov, which promotes smoothness, the goal of this regularization term is to induce discontinuous jumps in the parameter model. Aster (2013) presents the 1D TV regularization term as

$$TV_{1D}(\mathbf{m}) = \sum_{i=1}^{M-1} |m_{i+1} - m_i|, \tag{2.28}$$

which is equivalent to the L_1 norm of the derivative of the parameter model

$$TV_{1D}(\mathbf{m}) = \|\mathbf{D}_1\mathbf{m}\|_1. \tag{2.29}$$

For a 2D parameter model, as in the Tikhonov Regularization case, one is interested in the gradient rather than a discrete approximation of the first derivative (Aster, 2013). Therefore, one obtains the equation

$$TV_{2D}(\mathbf{m}) = \sum_{i=1}^{M_z-1} \sum_{j=1}^{M_x-1} |M_{i+1,j} - M_{i,j}| + \sum_{i=1}^{M_z-1} \sum_{j=1}^{M_x-1} |M_{i,j+1} - M_{i,j}|, \tag{2.30}$$

where $M_{i,j}$ refers to the elements of the matrix form of the parameter model described in equation 2.1. As in the 1D case, we can rewrite equation 2.30 in an operator form

$$TV_{2D}(\mathbf{m}) = \|\nabla \mathbf{m}\|_1. \tag{2.31}$$

While the L_1 regularization minimizes the non-zero values in $\mathbf{m}$, the TV regularization looks for the solution with the minimum number of discontinuities. In this way, one induces a step-wise parameter model. Aster (2013) suggest the use of IRLS, ISTA, or FISTA for finding the minimum of an objective function regularized through equation 2.29 or 2.31.

Aster (2013) also introduces a TV regularization term that considers the L_2 norm of the gradient of the parameter model,

$$TV_{2D-L_2}(\mathbf{m}) = \sum_{i=1}^{M_z-1} \sum_{j=1}^{M_x-1} \sqrt{|M_{i+1,j} - M_{i,j}|^2 + |M_{i,j+1} - M_{i,j}|^2}, \tag{2.32}$$

this can be written as

$$TV_{2D-L_2}(\mathbf{m}) = \|\nabla \mathbf{m}\|_2. \tag{2.33}$$

Presumably, the reason to propose equation 2.33 is to avoid the complications of working with a non-differentiable absolute value.

Several publications apply the TV regularization with the variant of equation 2.33. Such is the case of Chambolle (2004) for image denoising and zooming, Anagaw (2010)for seismic imaging, Liu et al. (2012) for X-ray tomography, and Anagaw and Sacchi (2020) for the Full Waveform Inversion problem combining TV with Regularization by Denoising.

Regardless of the particular variant of the TV regularization term, we can write the objective function

$$\phi_{TV}(\mathbf{m}) = ||\mathbf{d} - \mathbf{G}\mathbf{m}||_2^2 + \mu TV(\mathbf{m}). \tag{2.34}$$

The term $TV(\mathbf{m})$ will be substituded by equation 2.29, 2.31, or 2.33 accordingly to the characteristics of the problem.

2.3.3 Regularization by Denoising

In the next subsection, we will deepen in what is the main subject of this thesis, a new type of regularization technique named Regularization by Denoising (RED). As we have seen, the selection of the regularization determines has a substantial impact on the retrieved model and thus is a critical part of the inversion process. Such is its importance that the proponents of RED refer to this process as the ”holy grail” of inverse problems.

Romano et al. (2017) claimed that the problem of removing zero-mean Gaussian noise from an image is solved and that one should look for strategies to exploit the power of these denoising algorithms. As evidence of this bold statement, we refer to the publications of Chatterjee and Milanfar (2010) and Levin and Nadler (2011), who question the theoretical limit for denoising giving the similar performance of the different state of the art approaches to the denoising problem and wonder if there is still room for improvement. In that spirit, Romano et al. (2017) proposed the RED technique. Through this scheme, they incorporate a denoising engine in the regularization term of the cost function.

Venkatakrishnan et al. (2013), Chan (2016), and Chan et al. (2017) pioneered the integration of denoising algorithms in the inversion process. They utilized the Alternating Directions Method of Multipliers (ADMM) (Gabay and Mercier, 1976; Boyd et al., 2010) to split the cost function and incorporate a denoising step, naming the procedure Plug-and-Play Priors (P^3) after the plug-in structure facilitated by the ADMM algorithm.

Although it is possible to find an analytical expression that relates RED and P^3 (Romano et al., 2017), the conditions it considers will rarely be met. Furthermore, RED has advantages over P^3. Firstly, it has a clearly defined penalty term in the cost function. Secondly, that regularization term is a convex function that guarantees convergence, and finally, it has a more straightforward implementation (Romano et al., 2017).

The RED term is

$$\rho(\mathbf{m}) = \mathbf{m}^T \left[\mathbf{m} - \mathbf{f}(\mathbf{m})\right], \tag{2.35}$$

where $\mathbf{f}$ is the denoiser. The cost function for RED will then be

$$\phi_{RED}(\mathbf{m}) = ||\mathbf{d} - \mathbf{G}\mathbf{m}||_2^2 + \mu\mathbf{m}^T \left[\mathbf{m} - \mathbf{f}(\mathbf{m})\right], \tag{2.36}$$

or if we consider equation 2.5 we can use a more compact notation:

$$\phi_{RED} = \phi_{LS} + \mu\rho. \tag{2.37}$$

This is an interesting proposal because the purpose of denoisers is to improve the quality of the data $\mathbf{d}$, and here they are applied to the model parameter.

Denoising engine properties

Local homogeneity

The denoisers used in RED must fulfil two properties. The first one is called local homogeneity, and follows the definition

$$\mathbf{f}(c\mathbf{m}) = c\mathbf{f}(\mathbf{m}). \tag{2.38}$$

where c is an arbitrary scalar. The authors relaxed this condition and demanded it to be valid solely when c is very close to 1.

An important remark for the local homogeneity property appears when one considers a directional derivative such as

$$\nabla_{\mathbf{m}}\mathbf{f}(\mathbf{m})\mathbf{m} = \frac{\mathbf{f}(\mathbf{m} + \epsilon\mathbf{m}) - \mathbf{f}(\mathbf{m})}{\epsilon}. \tag{2.39}$$

When we substitute equation 2.38 into 2.39 we obtain

$$\mathbf{f}(\mathbf{m}) = \left[\nabla_{\mathbf{m}}\mathbf{f}(\mathbf{m})\right]\mathbf{m}, \tag{2.40}$$

this becomes relevant when calculating the gradient of the objective function.

Strong passivity

The denoising engine must also satisfy the strong passivity property, which refers to the requirement that the denoiser's Jacobian is stable. In other words, its spectral radius must be equal or lesser than one

$$\eta\left(\nabla_{\mathbf{m}}\mathbf{f}(\mathbf{m})\right) \leq 1. \tag{2.41}$$

This property can also be interpreted as the requirement for the denoiser to not amplify the norm of $\mathbf{m}$, so we can write it as

$$\|\mathbf{f}(\mathbf{m})\| \leq \|\mathbf{m}\|. \tag{2.42}$$

Minimizing the cost function

Now that we established the requisites of the denoisers, we can turn back our attention to the cost function. Once again, we will compute its derivative and set it equal to zero. One can notice that as in the Tikhonov Regularization, ϕ_{RED} has two terms. We can refer to the first one as the LS or data misfit term and to ρ as the prior term. This name comes from the fact that this algebraic expression carries our prior knowledge of the parameter model. It is not apparent, but the regularization term ρ is also convex, which guarantees the convergence of the method (Romano et al., 2017).

We already have the gradient of ϕ_{LS} in equation 2.11, so we will focus on differentiating ρ, which we can write as

$$\frac{\partial\rho(\mathbf{m})}{\partial\mathbf{m}} = \frac{\partial}{\partial\mathbf{m}}\left[\mathbf{m}^T\mathbf{m} - \mathbf{m}^T f(\mathbf{m})\right]. \tag{2.43}$$

Computing the derivative gives us

$$\frac{\partial\rho(\mathbf{m})}{\partial\mathbf{m}} = 2\mathbf{m} - \left[f(\mathbf{m}) + \nabla_{\mathbf{m}} f(\mathbf{m})\mathbf{m}\right], \tag{2.44}$$

Here, we can use the local homogeneity property to deal with the derivative of the denoiser. For that, we substitute equation 2.40 in 2.44 and obtain

$$\frac{\partial \rho(\mathbf{m})}{\partial \mathbf{m}} = 2\left[\mathbf{m} - f(\mathbf{m})\right]. \tag{2.45}$$

The gradient of RED's cost function, depicted in equation 2.37, can be expressed as

$$\frac{\partial \phi_{RED}}{\partial \mathbf{m}} = \frac{\partial \phi_{LS}}{\partial \mathbf{m}} + \mu \frac{\partial \rho}{\partial \mathbf{m}} \tag{2.46}$$

Substituting equations 2.11 and 2.45 in 2.46 leads to

$$\frac{\partial \phi_{RED}}{\partial \mathbf{m}} = 2\mathbf{G}^T\mathbf{G}\mathbf{m} - 2\mathbf{G}^T\mathbf{d} + 2\mu\left[\mathbf{m} - f(\mathbf{m})\right]. \tag{2.47}$$

When we set this gradient to zero we finally obtain

$$\mathbf{G}^T\mathbf{G}\mathbf{m} - \mathbf{G}^T\mathbf{d} + \mu\mathbf{m} - \mu\mathbf{f}(\mathbf{m}) = 0. \tag{2.48}$$

The authors of RED suggest three ways to solve equation 2.48: Steepest Descent, Fixed-Point, and ADMM. In any case, it involves an iterative method in which the denoiser is applied at least once per iteration. We will deepen more into the numerical methods for solving these problems in the following section.

On the denoisers

Romano et al. (2017) enlist several denoisers which comply with the two properties mentioned afore. They sustain that RED should benefit from the power of state of the art denoising methods, such as K-SVD filters (Elad and Aharon, 2006), Non-Local Means (NLM) (Buades et al., 2011), Block-Matching and 3D filtering (BM3D) (Dabov et al., 2007), and Trainable Nonlinear Reaction-Diffusion (TNRD) (Chen and Pock, 2017), among others. Nevertheless, as they deepen into their results, they suggest that any denoising engine may be able to regularize the inversion. What this means is that even a simple denoiser, such as the median filter, may meet the properties and stabilize the inverse problem.

In this work, we will focus our attention on three denoisers: the median filter, NLM, and Adaptive Weight Total Variation denoising (AWTV) (Liu et al., 2012), which is a modification of the Total Variation (TV) denoising technique proposed by Rudin et al. (1992). We selected the first two because we are interested in comparing the performance of a state of the art denoiser with a simple one when used for inverse problems regularization. The AWTV filter comes from its edge-preserving capabilities, and we want to explore if it can enforce this feature on tomography images.

It is worth to mention that we were initially interested in working with the TV as our edge-preserving denoiser. However, Anagaw and Sacchi (2020) describe that this denoising does not satisfy the local homogeneity property, and suggest instead utilizing the AWTV, the authors provide a mathematical proof of the satisfaction of this property. Furthermore, they deployed this denoiser with RED for the Full-Waveform Inversion problem with exceptional results.

2.4 Numerical optimization methods

We have stated the inverse problem as the solution of a system of equations, which take the form of large matrices. Solving the problem requires computing the inverse of such matrices, which can be expensive in terms of time and computer memory. Therefore, we will describe some techniques and algorithms which can be useful when dealing with large data sets.

2.4.1 Direct solution

Employing the direct solution method for solving equations 2.4, 2.12 and 2.17 requires the storage of the large matrix $\mathbf{G}^T\mathbf{G}$ and the computation of its inverse. If the problem is small enough and the computer sufficiently powerful, this method is the simplest for estimating the parameter. One only needs to employ one of the abundant numerical methods for solving equations systems such as Gauss-Jordan elimination, LU, or Cholesky decomposition, among others (Press et al., 1992).

Another limitation of the direct solution approach is that some matrix factorization methods like SVD, Cholesky, or QR experience trouble when dealing with sparse matrices, like the ones obtained when working with tomography problems (Aster, 2013). A sparse matrix is one whose elements are mostly zero. From a numerical perspective, it is more convenient to store an array of this nature in a different way than a dense one. One should only track the non-zero values and their indexes. Furthermore, it is possible to take advantage of this by using specially designed subroutines for storage and multiplication of sparse matrices and vectors (Scales, 1987). Some environments, like Julia and MATLAB, already include functions for this in their standard libraries.

2.4.2 Iterative Methods

Some alternatives to the direct solution method are the Kaczmarz's algorithm (Kaczmarz, 1937), the Algebraic Reconstruction Technique (ART) (Gordon et al., 1970), and the Simultaneous Iterative Reconstruction Technique (SIRT) (Gilbert, 1972). We will briefly describe these techniques because they were designed for tomographic problems, and early developments of seismic travel-time tomography employed these methods.

Kaczmarz's algorithm has an external cycle of iterations that starts with an initial model $\mathbf{m}_0$ and an internal one in which the current parameter model is projected in each row of $\mathbf{G}$. The external cycle repeats until convergence, which can be faster or slower, depending on the order of the rows. We can observe this in Figure 2.4. If there is more than one minimum, the algorithm will converge to the closest one.

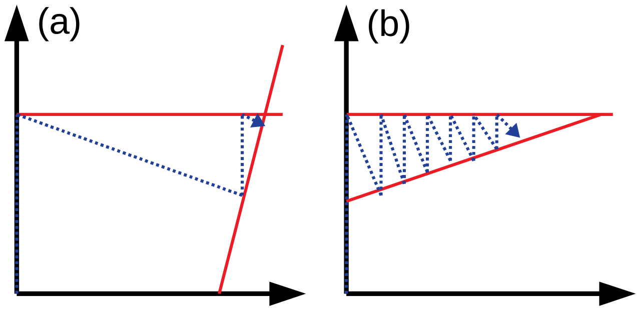

Figure 2.4: Kaczmarz algorithm applied to find the solution of a 2x2 equation system. (a) Shows the effect of the rows of **G** being nearly orthogonal, wile (b) displays what happens when they are nearly parallel. Adapted from Aster (2013)

Some variations of the Kaczmarz's method address the problem of the convergence rate by randomizing the rows of the matrix **G** (Strohmer and Vershynin, 2009) while others attempt to guide the order of the row selection to further speed the method (Lanteri et al., 2019). The ART is a modified version of Kaczmarz's algorithm that accelerates the convergence rate by reducing stored elements and the number of multiplications. However, it has the disadvantage of producing noisier results. SIRT improves this problem, but at the cost of having slightly slower computation times.

2.4.3 Gradient based methods

The iterative methods of Kaczmarz, ART, and SIRT are numerical techniques that allow solving the system of equations posed by the tomography problem without directly computing the inverse of a large matrix (Aster, 2013). The gradient-based methods are general optimization methods that offer an alternative approach (Strang, 1987). These methods focus on finding the minimum of convex functions. Thus the importance of emphasizing that the data misfit term $\|\mathbf{d} - \mathbf{Gm}\|_2^2$ and all the enlisted regularization terms fulfil this property. Examples of these methods are the Steepest Descent and the Conjugate Gradient.

Steepest Descent

Steepest Descent is the simplest of the gradient descent methods. According to Press et al. (1992), the idea is to start at an initial parameter model $\mathbf{m}_0$, compute the gradient at that point, advance a length α in that direction, and repeat until reaching the minimum. One can express this process through the iterative rule

$$\mathbf{m}_{i+1} = \mathbf{m}_i - \alpha \nabla_{\mathbf{m}} \phi(\mathbf{m}_i). \tag{2.49}$$

The problem of this method is determining the value of the step size α. On the one hand, if it is too large, we may overshoot the minimum. This situation can lead to jumping back and forth over the minimum, especially in a narrow valley of the cost function. On the other hand, a small step size will make little progress towards the minimum and require several iterations to reach the minimum.

One can envision a dynamic step length to overcome this problem. Tools like the Armijo Rule can help to achieve such a feature. However, this requires even more computations to determine the step length. All these drawbacks, which are more evident in Figure 2.5, cause this method to be slow and to promote the search for alternatives (Press et al., 1992).

Conjugate Gradient

An alternative to the Steepest Descent (SD) is the Conjugate Gradient (CG) method. This was originally developed by Hestenes and Stiefel (1952) for solving a positive definite system of equations of the form $\mathbf{Ax} = \mathbf{b}$. Aster (2013) provides a simple explanation of the method. First, consider that the solution of the equation system is equivalent to the minimum of the cost function

$$\phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x}. \tag{2.50}$$

The equivalency becomes evident after deriving the equation to find its minimum: $\nabla_{\mathbf{x}}\phi(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}$. Additionally, the positive definite assumption guarantees that this cost function will be convex and with a single minimum.

The CG method premise is that there is a vector basis $\mathbf{p}_i$ for the solution $\mathbf{x}$, i.e. $\mathbf{x}$ is a linear combination of $\mathbf{p}_i$ elements:

$$\mathbf{x} = \sum_i^n = \alpha_i\mathbf{p}_i \tag{2.51}$$

If we make this vector basis to be mutually conjugate with respect to $\mathbf{A}$ we will have that

$$\mathbf{p}_i^T\mathbf{A}\mathbf{p}_j = 0 \text{ for } i \neq j \tag{2.52}$$

Finally, if we substitute equation 2.51 in 2.50 and then apply the mutually conjugate property we obtain

$$\phi(\boldsymbol{\alpha}) = \sum_i^n \frac{1}{2}(\alpha_i^2\mathbf{p}_i^T\mathbf{A}\mathbf{p}_i - 2\alpha_i\mathbf{b}^T\mathbf{p}_i). \tag{2.53}$$

Now, the cost function depends on $\boldsymbol{\alpha}$, so we derive and equal to zero to have

$$\alpha_i = \frac{\mathbf{b}^T\mathbf{p}_i}{\mathbf{p}_i^T\mathbf{A}\mathbf{p}_i}. \tag{2.54}$$

Fulfilling the mutually conjugate property makes the problem easier. From this point, an iterative algorithm will update an initial solution $\mathbf{x}_0$ by adding elements of the basis $\mathbf{p}$, which depend on the residuals $\mathbf{r}$ scaled by the scalar α. The residuals are the difference between the vector $\mathbf{b}$ and the current solution. The detailed algorithm is below.

Algorithm 1 Conjugate Gradient

```
Inputs:
    Matrix $\mathbf{A}$
    vector $\mathbf{b}$
    residual tolerance $\epsilon$
Initialize:
    $\mathbf{x} = \mathbf{x}_0$
    $\beta = 0$
    $\mathbf{p} = \mathbf{0}$
    $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$
repeat
    $\mathbf{p} = -\mathbf{r}_1 + \beta\mathbf{p}$
    $\alpha = \frac{\|\mathbf{r}_1\|_2^2}{\mathbf{p}^T\mathbf{A}\mathbf{p}}$
    $\mathbf{x} = \mathbf{x} + \alpha\mathbf{p}$
    $\mathbf{r}_2 = \mathbf{r} + \alpha\mathbf{A}\mathbf{p}$
    $\beta = \frac{\|\mathbf{r}_2\|_2^2}{\|\mathbf{r}_1\|_2^2}$
    $\mathbf{r}_1 = \mathbf{r}_2$
until $\beta < \epsilon$
```

Theoretically, the algorithm is supposed to reduce the residual at each iteration and to converge at n iterations, making it a faster alternative to Steepest Descent. In practice, however, an exact solution might never be achieved due to round-off errors. Therefore, one can implement a criterion such as the relative decrease of the residual to stop the algorithm. The contrast on the convergence rate between CG and SD is portraited in Figure 2.5.

This method can also take advantage of sparse matrices. CG employs $\mathbf{A}$ and its transpose while avoiding the computation of their product. This last one might not be sparse even if the kernel matrix is. Nevertheless, CG only requires the products of $\mathbf{A}$ with a vector, which can be computed efficiently at a low memory cost for a sparse $\mathbf{A}$.

Additionally, it might not even be necessary to build the kernel matrix. Many forward problems can be expressed as a function (operator), which is compatible with CG since this last one only needs products of **A** and its transpose, i.e., the adjoint operator. The decision of which of these methods employ depends on the problem to solve.

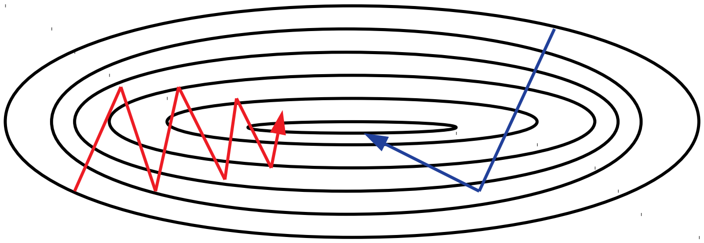

Figure 2.5: SD and CG applied to find the solution of a two variable quadratic equation. The red line represents the SD steps towards the minimum, the blue one represents the CG ones. Adapted from Press et al. (1992)

Conjugate Gradient Least-Squares

The CG method requires a positive-definite system of equations, i.e., a symmetric matrix. For the case of an asymmetric system, like an over-determined problem, we need to rearrange the LS solution in equation 2.12 to what is known as normal equations

$$[\mathbf{G}^T\mathbf{G}]\mathbf{m}_{LS} = \mathbf{G}^T\mathbf{d}. \tag{2.55}$$

The ”normal” name comes from the fact that one is effectively projecting the vector **d** into the vector subspace defined by the range of the matrix **G**.

Since $\mathbf{G}^T\mathbf{G}$, is a positive definite matrix, we can now apply the CG method to equation 2.55, Scales (1987) did for the traveltime tomography problem. This process leads to the method known as Conjugate Gradient Least-Squares (CGLS). There are a couple of things that one should consider for the numerical implementation and which are noticeable in algorithm 2. First, it produces a minor round-off error to factor $\mathbf{G}^T$ when computing the residuals, i.e. $\mathbf{r} = \mathbf{G}^T(\mathbf{Gm} - \mathbf{d})$ (Aster, 2013). Second, as in CG, we do not need to store $\mathbf{G}^T\mathbf{G}$ but only the products of $\mathbf{G}$ and $\mathbf{G}^T$ with vectors.

Algorithm 2 Conjugate Gradient Least-Squares

```
Inputs:
    Asymmetric matrix $\mathbf{G}$
    Observed data $\mathbf{d}$
    residual tolerance $\epsilon$
Initialize:
    $\mathbf{m} = \mathbf{m}_0$
    $\beta = 0$
    $\mathbf{p} = \mathbf{0}$
    $\mathbf{s} = -\mathbf{d}$
    $\mathbf{r} = \mathbf{G}^T\mathbf{s}$
repeat
    $\mathbf{p} = -\mathbf{r}_1 + \beta\mathbf{p}$
    $\alpha = \frac{\|\mathbf{r}_1\|_2^2}{(\mathbf{p}^T\mathbf{G}^T)(\mathbf{Gp})}$
    $\mathbf{m} = \mathbf{m} + \alpha\mathbf{p}$
    $\mathbf{s} = \mathbf{s} + \alpha\mathbf{Gp}$
    $\mathbf{r}_2 = \mathbf{G}^T\mathbf{s}$
    $\beta = \frac{\|\mathbf{r}_2\|_2^2}{\|\mathbf{r}_1\|_2^2}$
    $\mathbf{r}_1 = \mathbf{r}_2$
until $\beta < \epsilon$
```

We can also apply the CGLS method for the Tikhonov Regularization. For this, we only need to consider the following system with augmented matrices

$$\mathbf{G}_A\mathbf{m} = \mathbf{d}_A. \tag{2.56}$$

Where

$$\mathbf{G}_A = \begin{pmatrix} \mathbf{G} \\ \mu\mathbf{R} \end{pmatrix} \tag{2.57}$$

and

$$\mathbf{d}_A = \begin{pmatrix} \mathbf{d} \\ \mathbf{0} \end{pmatrix}. \tag{2.58}$$

Once again, the system 2.56 needs to be written in terms of its normal equations

$$\left[\mathbf{G}_A^T\mathbf{G}_A\right]\mathbf{m} = \mathbf{G}_A^T\mathbf{d}_A, \tag{2.59}$$

to make it positive definite and be able to apply the CG method.

2.5 The Bayesian approach

The approach discussed so far for solving inverse problems has a mathematical inspiration because it is about stability and smoothness (Menke, 2018). The Bayes method is not deterministic, so it is senseless to talk about estimating specific values for $\mathbf{m}$ as before. Instead, the Bayesian method considers the parameter model a random variable with a certain probability distribution $p(\mathbf{m})$ (Aster, 2013).

2.5.1 The Bayes Theorem

The Bayes theorem from probability theory states that

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}. \tag{2.60}$$

Where $P(\cdot)$ is the probability of occurrence of and event, A and B are independent random events, and $P(A|B)$, is the probability of A given B, i.e conditional probability. Equation 2.60 is also valid for probability density functions (pdf's). Thus, if we consider that the observed data $\mathbf{d}$ and the parameter model $\mathbf{m}$ are now random variables such as A and B we can write

$$p(\mathbf{m}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{m})p(\mathbf{m})}{p(\mathbf{d})}. \tag{2.61}$$

Each one of the terms in equation 2.61 has a profound meaning. To start, $p(\mathbf{m})$ is the "prior" or "a priori" distribution, and it represents the beforehand knowledge that we possess or assume of the parameter model. How well one incorporates the prior knowledge is a hot debate, leading to being the Bayes approach to be considered subjective, "unscientific" (Aster, 2013) or at the very least complicated (Ulrych et al., 2001). Simple priors as a Gaussian or uniform distribution are relatively easy to include, but a constraint such as the one imposed by TV is hard to conceive as a pdf.

The second term on equation 2.61 we should discuss is the $p(\mathbf{d}|\mathbf{m})$, sometimes referred as the "likelihood" term, and it represents the probability of having our observed data set $\mathbf{d}$ given the parameter model. Notice that the selection we make for the prior will affect this term. Then we have the $p(\mathbf{d})$ or the probability that the data is observed, which is a constant value given by

$$p(\mathbf{d}) = \int p(\mathbf{d}|\mathbf{m})p(\mathbf{m})d\mathbf{m}. \tag{2.62}$$

Notice that the integral in equation 2.62 spans over all the possible parameter models. This fulfills the role of a a normalizing constant so the integral of $p(\mathbf{m}|\mathbf{d})$,is equal to one (Aster, 2013). Lastly, $p(\mathbf{m}|\mathbf{d})$, also known as the a posteriori probability, represents the likelihood of having a parameter model $\mathbf{m}$ given the observed data. Obtaining this pdf is the objective of the Bayesian inversion, a probability distribution that describes the parameter model based on the data observations.

2.5.2 A thought experiment

Bayesian inversion is a vast field of study, with several publications, with its benefits and downfalls. Addressing specific examples is out of the scope of this project since it would require the explanation of several theoretical issues. Duijndam (1988) covers in detail this subject, providing seismic applications. Nonetheless, we believe that it is worth developing a thought experiment that enriches the understanding of the subject.

Inspired by an example published by Menke (2018), let us imagine a simple experiment. We are trying to determine the density of an object, and this will require us to measure the mass and volume of said object. Each measurement will likely lead to a different density value due to the inherent measurement errors.

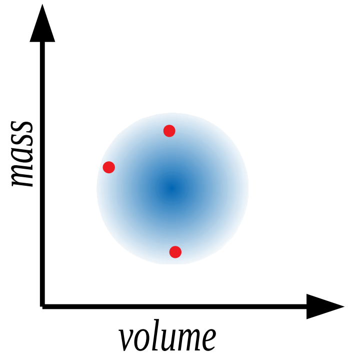

Figure 2.6: Distribution of the data points collected in our experiment. The red dots represent our available data pairs while the cloud is the result of the very large number of experiments. Adapted from Menke (2018)

An infinite number of measurements will produce a point cloud plot of mass versus volume like the one in Figure 2.6. This cloud resembles a probability distribution, and will correspond to $p(\mathbf{d})$. If we consider that it has a normal distribution, we can characterize it by a mean and variance. The first one is matching the center of the cloud while the second relating to its radius. Notice that this consideration is similar to the one made in the previous approach when introducing the L_2 norm on the data misfit.

For estimating the density, we will count with a finite number of measurements, represented by the red points in Figure 2.6. These red points are our data set $\mathbf{d}$ and, along with our prior distribution assumption, are the key for obtaining the a posteriori distribution.

A reasonable assumption for the prior would be to consider that it also has a Gaussian distribution. Notice that this consideration is effectively governing the likelihood term, in a similar fashion that the forward problem depends on $\mathbf{m}$. Reciprocally, the addition or subtraction of data points changes $p(\mathbf{m}|\mathbf{d})$. Clustering red points in Figure 2.6 will lead to a longer tail a priori pdf and vice-versa.

2.6 Summary

We open this section by establishing the concepts forward and inverse problems. Then, we disclose some fundamental concepts of discrete inverse theory, such as how to classify and solve different inverse problems. Later, we make a brief description of ill-posedness and the complications that arise with the solution of inverse problems. We paid particular attention to explaining with simple graphic examples.

The next section deepens into the subject of regularization of inverse problems, mentioning both the classical technique of Tikhonov and the new method of Regularization by Denoising. After that, we explain the specific ways to implement these solutions on a computer. We describe a few algorithms useful for solving the equation system associated with a linear inverse problem, such as the direct method, Steepest Descent, and Conjugate Gradient.

At the end of the chapter, we make some brief comments on the Bayesian inversion, with the sole goal of offering a different approach and make the connection to the probability point of view of the inverse theory.

CHAPTER 3

Linear traveltime tomography via Regularization by Denoising

The traveltime tomography problem consists of computing a velocity model from traveltime records. The particular case of a cross-well tomography consists of positioning several sources and receivers inside wells and creating a velocity map of the section between them. One can use the velocity images to correlate information among wells, to support other methods, like seismic or resistivity imaging, or even to monitor productive hydrocarbon fields. Cross-well tomography is a well-known problem in the oil industry, and in this chapter, we are interested in addressing this inverse problem and asses the potential of RED to improve the results achieved with conventional methods like the Tikhonov Regularization.

3.1 The cross-well model

We will start by setting up the problem geometry, as in Figure 3.1. We placed 41 sources and 41 receivers on parallel lines on the opposite sides of a square section. Intervals of 0.25 [m] separate both sources and receivers. The square section has sides of 1 [Km], discretized on $M = 51 \times 51 = 2601$ cells. The number of raypaths is equal to the sources-receiver pairs, leading to $N = 1681$, and since $N > M$, we have an under-determined problem. Notice that M will depend on the size we choose for the cells. On the one hand, large cells will be easy to compute but will lead to poor resolution and a useless physical model. On the other hand, small cells will lead to larger computation times. Furthermore, if the cell is too small, it will lead to poorer conditioning of the tomographic matrix (Aster, 2013).

One can describe the forward problem with equation 2.3. For this example, $\mathbf{d}$ corresponds to the traveltimes, while the vector parameter $\mathbf{m}$ corresponds to the constant slowness in each cell. The matrix $\mathbf{G}$ connects the domains of the data and the parameters. Its rows correspond to a source-receiver pair raypath, while its columns match the model cells. The magnitudes in the rows are equal to the distances the raypath travels in each cell. Notice that since each raypath crosses only a portion of the velocity model, $\mathbf{G}$ will be sparse, as shown in Figure 3.2b.

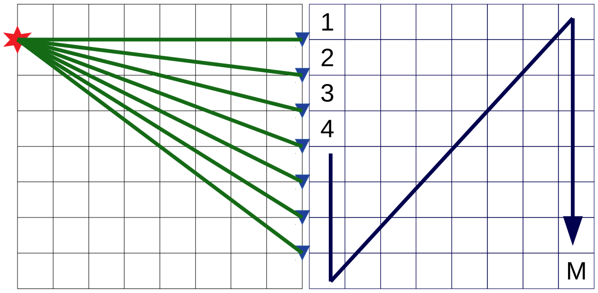

Figure 3.1: Example of a cross-well tomography. The red star is the source, the green lines the raypaths, and the receivers appear in blue. The right-hand side of the figure shows the parametrization of the between-wells section.

3.2 The Tikhonov approach

As we know from the previous chapter, the solution of the inverse problem requires a regularization. A conventional approach in tomography is to use the Tikhonov regularization, i.e., equation 2.22. Due to the short scale of these examples, we decided to use the direct solution to solve this equation.

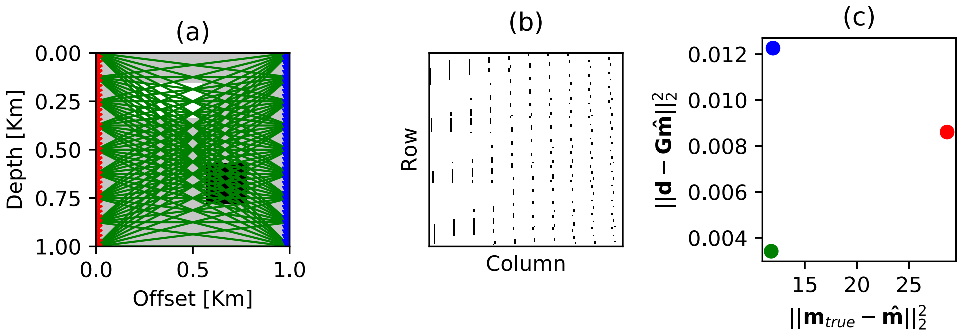

Figure 3.2: (a), ray coverage of the problem. Only a fifth of the raypaths are shown for convenience. (b), portion of the kernel matrix G. (c), data misfit vs model misfit for three values of **R** in the Tikhonov Regularization. Red, green and blue correspond to the DLS, flat and smooth solutions respectively.

3.2.1 Minimum norm, flat and smooth solutions

We propose a true velocity model which we display on Figure 3.3a and an acquisition geometry shown in Figure 3.2a. We estimated the parameter model through equation 2.22 ($\mathbf{m}_{TK} = [\mathbf{G}^T\mathbf{G}+\mu\mathbf{R}]^{-1}\mathbf{G}^T\mathbf{d}$) via the CGLS method. As one can recall, the Tikhonov Regularization allows for three classical solutions: the DLS or minimum norm, flat, or smooth solutions. These $\mathbf{m}$ estimations are obtained by setting $\mathbf{R}$ equal to $\mathbf{I}$, $\mathbf{D}_1$ and $\mathbf{D}_2$ respectively. The velocity models retrieved by these regularizations are are shown in figures 3.3b to 3.3d.

One can compute the traveltimes for the true, minimum norm, flat, and smooth velocity models in Figure 3.3 by solving the forward problem ($\mathbf{d} = \mathbf{Gm}$). These traveltimes appear in Figure 3.4, where we can see that the estimated traveltimes for the three solutions are reasonably similar to the observed ones. As mentioned in the previous chapter, non-uniqueness is a common phenomenon when solving inverse problems.

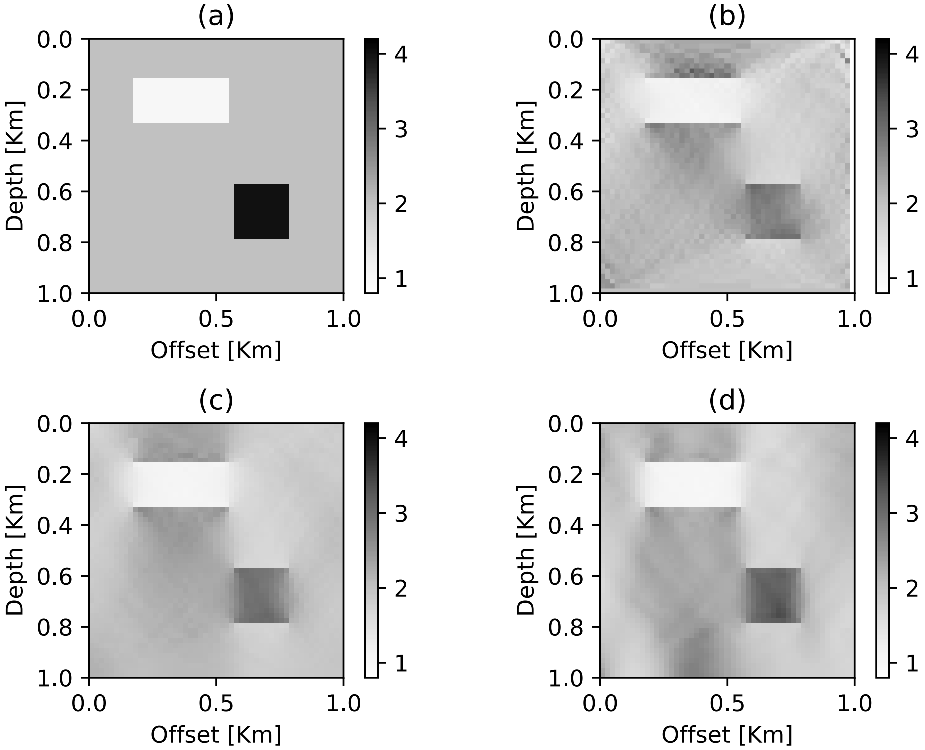

Figure 3.3: (a) true, (b) minimum norm ($\mathbf{R} = \mathbf{I}$), (c) flat ($\mathbf{R} = \mathbf{D}_1$), and (d) smooth ($\mathbf{R} = \mathbf{D}_2$) veloicity models computed by solving equation, units are [Km/s]

Nevertheless, the comparison with the true velocity model in Figure 3.3 allows us to see that some solutions are better than others. Although the minimum norm solution resembles the true model, it displays heterogeneities in areas that should be homogeneous. The smooth solution shows changes of the same magnitude but smoothed, keep in mind that this regularization aims to minimize the spatial derivatives of the model. The flat regularization seems to provide the model that resembles the most to the true one. In order to quantify the quality of the retrieved models, we created Figure 3.2c, which shows the fit to the data and the true model for each solution, leaving the flat solution as the best one for this example.

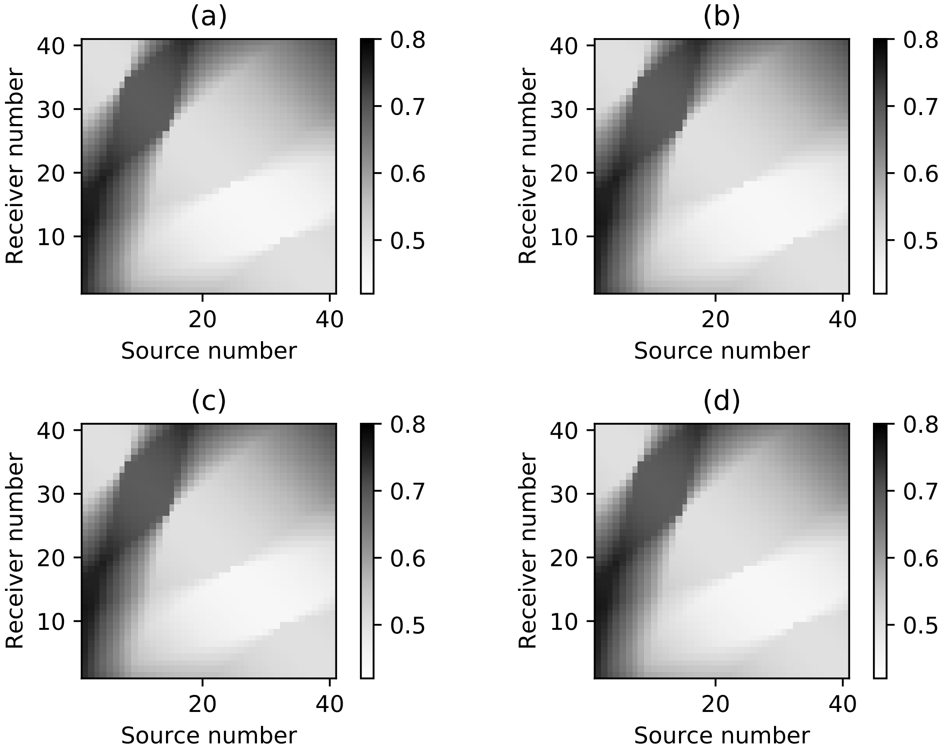

Figure 3.4: (a) true and estimated traveltimes for the, (b) minimum norm, (c) flat, and (d) smooth solutions. Units are [s]

3.2.2 The trade-off parameter

In this first step, we solved the inverse problem with different values for $\mathbf{R}$ but using the same value for the trade-off parameter, which was selected arbitrarily. Nonetheless, a proper methodology exists for determining the proper μ. For this, we need to plot an L-curve, which compares the two terms of the cost function: the data misfit norm $||\mathbf{d}-\mathbf{G}\mathbf{m}||_2^2$, and the seminorm of the parameter model $||\mathbf{R}\mathbf{m}||_2^2$ (Aster, 2013).

We altered the traveltimes for the true model by adding Gaussian noise with zero mean and a standard deviation equal to 0.01 times the maximum traveltime, which led to a noise variance of $\sigma^2 \approx 6[ms]$. We can observe these noisy traveltimes in Figure 3.5c. Then, we ran the inversion algorithm with $\mathbf{R} = \mathbf{D}_1$ for 9 different values of μ ranging from 10^{-2} to 10^{2}. The L-curve showing the relationship of the data misfit norm and the retrieved parameter models semi-norm for these nine values of μ appears in Figure 3.5a.

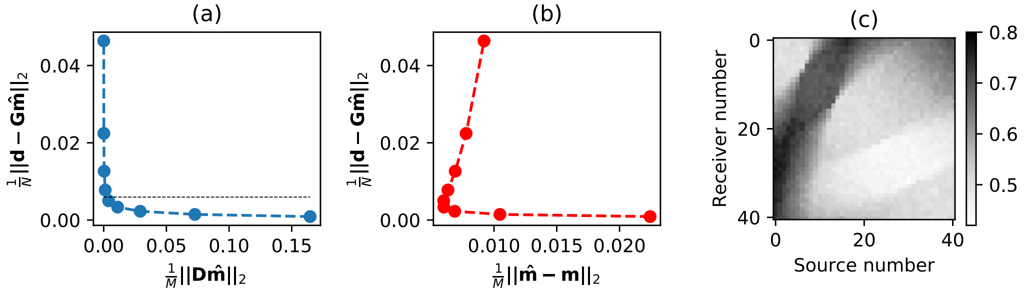

Figure 3.5: (a) L-curve showing the trade-off between fitting the data and honoring the constraint. The horizontal line is the variance of the added noise. (b) data misfit, versus model misfit, as in the L-curve, each point represents a solution for a given μ value. The higher the data misfit, the higher the μ value. (c) observed traveltimes with added noise.

The upper panels of Figure 3.6 shows three velocity models retrieved with the $\mathbf{R} = \mathbf{D}_1$ and different trade-off values, while the lower ones display its corresponding traveltime residuals, i.e., the subtraction of the estimated traveltimes $\mathbf{Gm}$ from the observed ones $\mathbf{d}$. By observing the recovered velocity models, one can see that as the trade-off parameter value increases, the model becomes flatter and the residuals larger. This phenomenon occurs because we are no longer honoring the observed data and are instead only obtaining a flat model and is known as under-fitting. On the other hand, when the μ value is small, the residuals decrease in magnitude, but the velocity model is not correctly constrained, this is known as over-fitting.

The proper trade-off value should be that one that balances the data misfit and the regularization term. In Figure 3.5a that is the solution with $\mu = 1.0$. One way to estimate the trade-off parameter value is to consider the normalized data misfit as a function of μ. Then we will look for the μ, which makes this value equal to the variance of $\mathbf{d}$. We can write this in equation form as

$$\frac{\|\mathbf{d} - \mathbf{G}\hat{\mathbf{m}}_\mu\|_2^2}{N} = \sigma_{\mathbf{d}}^2. \tag{3.1}$$

This method requires to know or estimate the value of $\sigma_{\mathbf{d}}^2$.

We decided to plot the data misfit norm versus the model misfit norm in Figure 3.5b to corroborate the methodology for selecting the trade-off parameter. We can notice that the selected optimal model corresponds to the best data and model fits. One should keep in mind that this test is only possible with synthetic data.

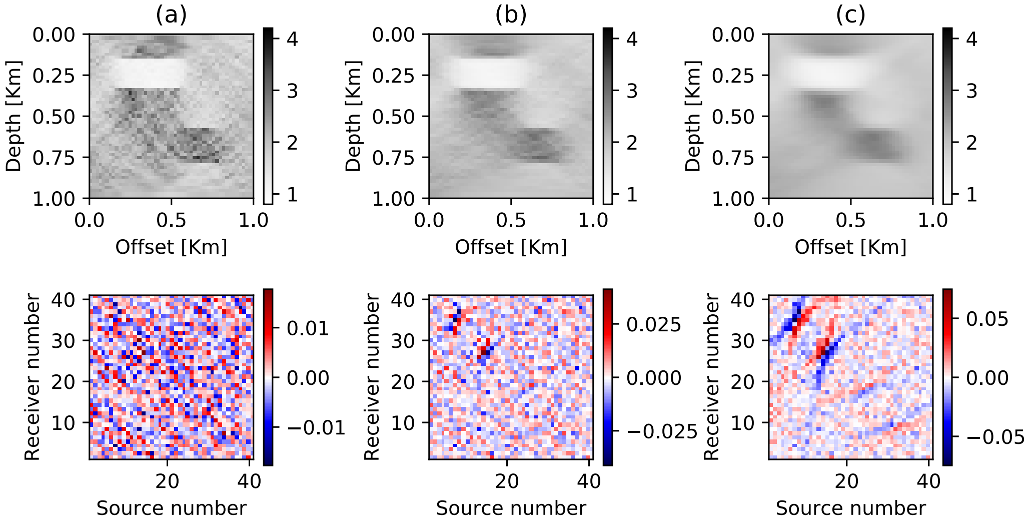

Figure 3.6: Effect of the trade-off parameter in the inverted models. Each column shows a different value for μ: (a), $\mu = 10^{-1}$. (b), $\mu = 10^{0.0}$. (c), $\mu = 10^{1}$. The upper row shows the inverted velocity model while the lower one shows the traveltime residuals $\mathbf{d} - \mathbf{Gm}$.

3.3 The RED approach

In the past chapter, we introduced the ideas of Romano et al. (2017) about using an image denoiser algorithm as a regularization term. In there, we stopped the discussion at the cost function (equation 2.36) because we were yet to discuss the subsection about the numerical solutions methods for inverse problems. In this section, we will retake the RED method, explain the numeric techniques for its execution, deepen into the denoising engines, and finally, apply it for the cross-well example used with Tikhonov.

Romano et al. (2017) offer three means for finding the minimum of the RED cost function (equation 2.36). The first one is the simplest: Steepest Descent. For this, we consider gradient-nulling equation on 2.48: $\mathbf{G}^T\mathbf{G}\mathbf{m} - \mathbf{G}^T\mathbf{d} + \mu\mathbf{m} - \mu\mathbf{f}(\mathbf{m}) = 0$. To deal with the subtraction $\mu(\mathbf{m} - \mathbf{f}(\mathbf{m}))$, we simply consider $\mu(\mathbf{m}_{i-1} - \mathbf{f}(\mathbf{m}_i))$ as shown in algorithm 3.

Algorithm 3 RED via SD

```
Inputs:
    Kernel $\mathbf{G}$
    Observed data $\mathbf{d}$
    Step length $\alpha$
Initialize:
    $\mathbf{m} = \mathbf{m}_0$
repeat
    $\mathbf{m}_f = f(\mathbf{m})$
    $\mathbf{m} = \mathbf{m} - \alpha\left[\mathbf{G}^T(\mathbf{G}\mathbf{m} - \mathbf{d}) + \mu(\mathbf{m} - \mathbf{m}_f)\right]$
until convergence
```

The authors suggest the improvement of algorithm 3 through the application of the CG strategy instead. Nevertheless, they also refer that since these two are gradient-based methods, they are subject to the same shortcoming: a single denoising activation per iteration. This method is not as effective because the effect of the denoising engine will be minor, which leads to a large number of iterations.

An alternative to the gradient-based methods exits in the form of the Alternating Direction Method of Multipliers (ADMM) and the Fixed-Point (FP) strategy. We will skip the first one and focus on the second since it was the one that provided the best results for the authors of the RED. Moreover, it is simple to understand and to implement, as shown in algorithm 4.

Algorithm 4 RED via FP

```
Inputs:
    Kernel $\mathbf{G}$
    Observed data $\mathbf{d}$
Initialize:
    $\mathbf{m} = \mathbf{m}_0$
repeat
    $\mathbf{m}_f = f(\mathbf{m})$
    $\mathbf{A} = \mathbf{G}^T\mathbf{G} + \lambda\mathbf{I}$
    $\mathbf{b} = \mathbf{G}^T\mathbf{d} + \mathbf{m}_f$
    solve for $\mathbf{Az} = \mathbf{b}$ for $\mathbf{z}$ (e.g. CG)
    $\mathbf{m} = \mathbf{z}$
until maximum iterations
```

The FP method for finding the roots of a general function $f(x)$ requires to rewrite f as $x = g(x)$ and then apply the recursive iteration $x_{k+1} = g(x_k)$ to find the fixed-points of g. For this method to converge $|\frac{d}{dx}g(x)| \leq 1$ for a given interval. We want to find the minimum of the gradient of the cost function 2.36. For that, we need to find the roots of the gradient described by equation 2.48. Romano et al. (2017) explain that one can rewrite equation 2.48 as the iterative expression

$$\mathbf{m}_{k+1} = \left[\mathbf{G}^T\mathbf{G} + \mu\mathbf{I}\right]^{-1}\left[\mathbf{G}^T\mathbf{d} + \mu\mathbf{f}(\mathbf{m}_k)\right], \tag{3.2}$$

and that the FP method will converge because

$$\| \left[\mathbf{G}^T\mathbf{G} + \mu\mathbf{I}\right]^{-1} \mu\nabla\mathbf{f}(\mathbf{m}_k) \| \leq 1, \tag{3.3}$$

We preferred the FP method and applied it to the velocity model presented at the beginning of the chapter. Since the problem is small, we can solve the $\mathbf{Az} = \mathbf{b}$ system directly. We want to test the performance of three denoising engines as well as the behavior of the trade-off parameter on each denoiser.

3.3.1 The denoising engines

The power of the RED method resides in the capabilities of the denoising algorithms to remove Gaussian noise while preserving the signal (Romano et al., 2017). Therefore, it is relevant to expand into these routines. In that spirit, we will now explain in detail the denoising engines utilized in this project. The first one is a median filter, a straightforward yet robust denoising method (Claerbout and Muir, 1973; Huang et al., 1979). The second one, the Non-Local Means (NLM), is a state of the art denoiser (Buades et al., 2011; Bonar and Sacchi, 2012). The third and final one is a modification of the Total Variation (TV) denoising approach (Rudin et al., 1992).

Median filter

The median filter is a robust measure of central tendency. Robustness, in this context, refers to the capability of dealing with the effect of outliers. The importance of this property becomes evident in the following example. Let us consider a group of three numbers: 10.1, 9.9, and 10. The mean and median for this group are the same: 10.0.

Nevertheless, if we change the value of the last element to 100.0, i.e., if we introduce an outlier, the mean will change and will no longer be representative of the group. In the meantime, the median will be impervious to the outlier and preserve its value. This simple case shows how robustness can help an estimator to avoid the bias of erroneous data.

One can apply the median filter as a moving window on an image to remove the noise (outliers) while preserving the edges of the features on the image.

Claerbout and Muir (1973) claimed that this estimator could serve as a comprehensive tool for geophysical data fitting. One can also use the mean instead of the median for denoising purposes. The idea here is that the averaging process will reduce the variance of the noise depending on the number of elements in the window.

Non-Local Means

The reduction in the noise variance mentioned afore only happens if the pixels in the window are ”similar”, i.e., the same color. Let us think about the photography of a person. On the one hand, removing the noise on features like the cheeks will be easy because it is a vast area with similar pixels. On the other hand, elongated features like the eyelashes will be hard to denoise because the pixel neighborhood contains different colors.

The previous example teaches us that similar pixels have no reason to be nearby (Buades et al., 2011). The NLM method faces this problem by setting two moving windows. The first one is a patch around the pixel being denoised, while the second one scans the whole image for similar pixels. Later, it assigns a value to the denoised pixel depending on the mean and similarity of the two windows. In this way, the method aims to reduce the noise variance using pixels across the whole image while avoiding blurring as much as possible.

Adaptive Weight Total Variation

In the previous chapter, we introduced the TV regularization and how it induces parameter models with discontinuities. One can set up the denoising problem as an inverse one by considering **G** from equation 2.34 equal to **I**. Then, by applying TV regularization, one obtains

$$\phi_{TV}(\mathbf{u}) = \|\mathbf{v} - \mathbf{u}\|_2^2 + \mu TV(\mathbf{u}), \tag{3.4}$$

where $\mathbf{v}$ is the noisy image, $\mathbf{u}$ is the estimated clean one, and

$$TV(\mathbf{u}) = \sum_i \sqrt{(\mathbf{D}_x\mathbf{u})_i^2 + (\mathbf{D}_x\mathbf{u})_i^2}. \tag{3.5}$$

The matrices $\mathbf{D}_x$ and $\mathbf{D}_z$ in equation 3.5 refer to the horizontal and vertical derivatives of the image $\mathbf{v}$. Therefore, one can rewrite equation 3.4 as

$$\phi_{TV}(\mathbf{u}) = \|\mathbf{v} - \mathbf{u}\|_2^2 + \mu\|\nabla\mathbf{u}\|_2. \tag{3.6}$$

The TV denoising method was first proposed by Rudin et al. (1992), and it produces a signal with a piece-wise behaviour, which makes it well suited for edge-preserving (Chambolle, 2004; Zhu and Chan, 2008; Selesnick and Chen, 2013). Anagaw and Sacchi (2020) sustain that the TV denoiser does not fulfill the local homogeneity property described in equation 2.40, and propose a way to mimic such property through what they describe as Adaptive Weight Total Variation (AWTV) (Liu et al., 2012).

AWTV modifies the TV regularization term in equation 3.5 to

$$AW(\mathbf{u}) = \sum_i \sqrt{\omega(\mathbf{x}_i)\mathbf{x}_i^2 + \omega(\mathbf{z}_i)\mathbf{z}_i^2}, \tag{3.7}$$

where $\mathbf{x} = \mathbf{D}_x\mathbf{v}$ and $\mathbf{z} = \mathbf{D}_z\mathbf{v}$. The term $\omega(\mathbf{u})$ in equation 3.7 is the parameter model gradient-dependent adaptive weight determined by

$$\omega(a) = e^{-\left(\frac{a}{\gamma}\right)^2}, \tag{3.8}$$

where γ is a scale factor which controls the smoothing (diffussion) at the edges of the image and a is a single element of the horizontal or vertical differences of the image. Notice that if $\gamma \to \infty$ one returns to the classic TV problem, while if it is too small, the method will be inefficient to remove noise. With these definitions settled, one can compactly write the AWTV denoising problem as

$$\phi_{AWTV}(\mathbf{u}) = \|\mathbf{v} - \mathbf{u}\|_2^2 + \mu AW(\mathbf{u}). \tag{3.9}$$

We selected the median filter because we want to prove that even a simple denoiser can perform the regularization task delivering decent results, as it did for Romano et al. (2017) for the inverse problems of image deblurring and super-resolution. In contrast, the NLM is a state of the art denoiser, whose fulfilling of the local homogeneity and strong passivity properties has been already proved by the proponents of the RED method, and that should provide the best outcomes. Lastly, the AWTV is an edge-preserving denoiser, which produced outstanding results for Full Waveform Inversion regularization (Anagaw and Sacchi, 2020), and we are interested in exploring the possibility of edge preservation in tomography.

3.3.2 Linear RED results

We present now the results of the cross-well tomography inversion via RED using the same noisy traveltimes from Figure 3.5c. To start, we decided to plot an L-curve to determine the proper μ value. We found that we required different ranges for the trade-off parameter for each denoiser. Thus, we tested eight values for each case, ranging from $10^{-0.4}$ to $10^{2.4}$, $10^{-0.5}$ to $10^{1.8}$, and $10^{-1.5}$ to 10^{2}, for the median, AWTV, and NLM denoisers respectively.

The application of a denoising function on the regularization term may lead to the incorrect perception that we are denoising the velocity model when what we are actually doing is promoting the recovery of a solution with desirable features. The over-fitting velocity model for the Tikhonov solution in Figure 3.6c is equivalent to a not regularized solution that displays high-frequency variations. The First Order Tikhonov regularization extracts these high-frequency features through a derivative, so their presence is minimized when we find the minimum argument of the cost function.

The removal of noise in an image is the removal of high-frequency features. Thus, the RED technique aims to promote the recovery of a velocity model that lacks these high-frequency features by including the parameter residual $\mathbf{m} - f(\mathbf{m})$ in the cost function. Therefore, RED extracts the high-frequency components of the velocity model through denoising functions specifically tailored for this purpose.

Each denoising engine $f(\mathbf{m})$ depends on a set of parameters that change according to factors like the size of the image, the features one wants to recover, and the amplitude of the high-frequency variations one is interested in removing. Hence, we require to determine these parameters before running the inversion algorithm. In order to find the most suitable parameters for the denoising engines, we designed a test in which we simulated the presence of undesirable high-frequency features on the velocity model and its removal by the denoising engines.

We simulated the high-frequency features on the velocity model by adding Gaussian noise. RED is designed to work with denoising engines that excel at removing this kind of noise. The combination of parameters for each denoiser that provided the best removal of the noise was selected as the proper one for the inversion. For the median case, we tested two window sizes, 3×3 and 5×5, and determined that the earlier was superior. The publication from Buades et al. (2011) offered guidance for defining the NLM parameters and helped us to determine a 21×21 search window, a 3×3 similarity window, a 0.05 degree of filtering, and a simple averaging kernel.

The application of the AWTV denoiser requires solving the inverse problem posed by equation 3.9. For this task, we follow the fast gradient-based approach of Li-yan and Zhi-hui (2011). The parameter configuration which provided the best results was a trade-off of 10^{-1} and $\gamma = 10^{-0.7}$. During our experiments, we tested the capabilities of the TV denoiser and found that it did not produce a proper L-curve, having even negative values for $\mathbf{m}^T\left(\mathbf{m} - f(\mathbf{m})\right)$, which further confirm that the TV denoiser.

We exhibit the computed L-curves in the upper panels of Figure 3.7, while in the lower panels, we show the normalized data misfit versus the model misfit. The left column (red curves) of the figure correspond to the median filter, the central one (green) to the AWTV, and the right one (cyan) to NLM. The lower panels allow us to observe that, as in the Tikhonov case in Figure 3.5, the selected model reasonably matches the one with a low model misfit.

The panels displaying the L-curves also show the noise level as a black dashed line. We can use this as a criterion to select the optimal solution since it does not make sense to fit the data below this misfit level. The selected optimal model appears in Figure 3.7 with a black star, and corresponds to a value for the trade-off parameter of 0.1 for the median filter case, 0.631 for the AWTV, and 0.1 for the NLM.

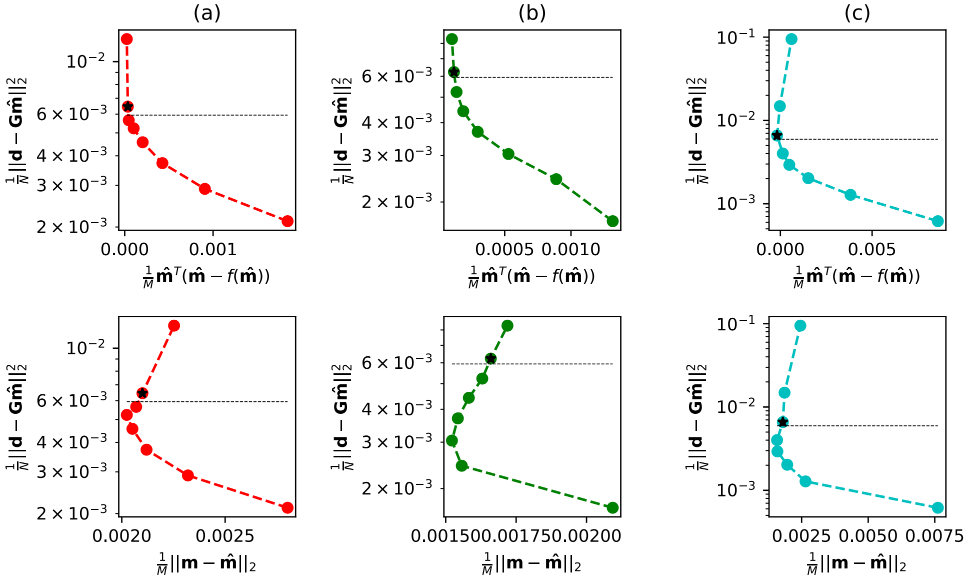

Figure 3.7: The upper row shows the L-curves, while the lower one displays the normalized data misfit vs model misfit plots for the (a) median (red), (b) AWTV (green), and (c) NLM (cyan) cases. Each point represents a solution with a different μ value, the dashed line represents the variance of the traveltime noise, and the black star the optimal model.

The lower row of Figure 3.7 demonstrates that the selected optimal solutions are not the ones with the lowest model misfit. This situation occurs due to noise contamination. Nevertheless, one should keep in mind that we can only compute the model misfit because we are working with synthetic data and that the L-curve is fulfilling its purpose of providing a solution that honors the observed data while avoiding over-fitting.

Figure 3.8 displays three inverted models for each denoiser: the left, central, and right columns correspond to the median, AWTV, and NLM denoisers, respectively. The central row of the figure shows the optimal models, the upper one exhibits an example of a model that over-fits the data, and the bottom one displays an under-fitting case.

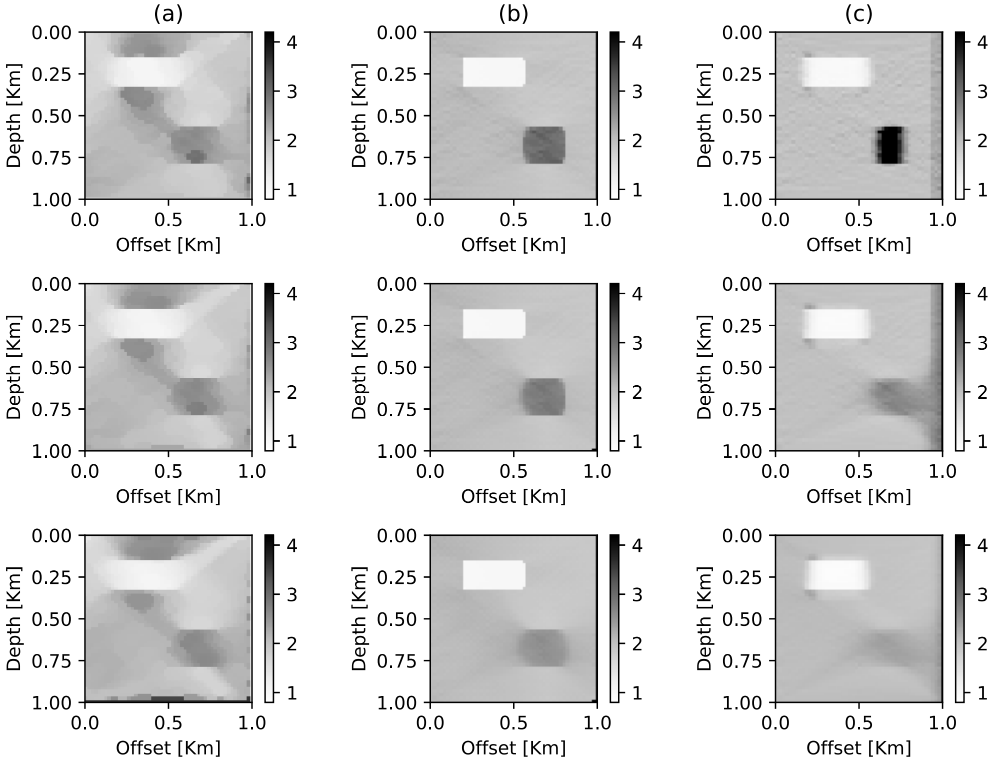

Figure 3.8: Effect of the trade-off parameter in the inverted models using RED. The columns (a), (b), and (c), correspond to the median, TV, and NLM denoisers, respectively. Each row corresponds to a different μ value, the central one represents the optimal, while the upper and lower represents a lower and higher than the optimal μ values respectively.

We mentioned afore that the suggested optimal models from the L-curves were not the ones with the lowest model misfit. This situation becomes evident when looking at the over-fitting examples of Figure 3.8. In these panels, one can notice how these velocity models resemble more the true model, particularly for the AWTV and NLM cases.

We have mentioned that our numerical examples suggest that a solution classified by the L-curve criterion as over-fitting is the one with the lowest velocity model misfit. Nonetheless, one should keep in mind that this evaluation is only possible when working with synthetic data and that most of the time, the sole information available for inversion is the observed data and our assumptions on the parameter model. Fitting the data below the noise level implies not honoring the observations but the noise and possibly introducing unrealistic information on the velocity model.

The velocity models in Figure 3.8 demonstrate that a denoising engine can regularize the linear tomographic inversion problem. Furthermore, a comparison of the Tikhonov and RED results shows that the results delivered by the later one are superior. We can quantify this statement by noticing that the values for the normalized model norm for RED are smaller than the ones for flat Tikhonov.

The AWTV results manage to preserve the edges of the velocity model, unlike the conventional regularization and the other two denoisers, The NLM selected optimal solution seems blurry, as the Tikhonov one, however, the one classified as over-fitting by the L-curve criterion manages to recover the actual magnitudes of the velocity model. Even the simplest denoiser can better preserve the edges than the Tikhonov case.

The results for the cross-well tomography in this chapter encourage the application of RED on the more complex scenario of turning-ray tomography, which is a non-linear problem. We have proved that the denoisers can stabilize the inversion process, but we still want to explore if it can deliver superior results to the conventional technique.

3.4 Summary

In this chapter, we explored the application of RED on a linear tomography problem and compared its performance with a conventional regularization technique, such as the one of Tikhonov.

We started performing inversion on a synthetic model using the traditional regularization. First, we selected the most convenient operator: flat over the smooth and minimum norm. Then, we explained the process for finding the proper trade-off parameter, and the effects of this last one on the retrieved models.

We then continued to the application of RED, starting by introducing the numerical methods to implement it, and continuing with the characteristics of the selected denoising engines. We then repeated the trade-off parameter selection process for each denoising engine.

We closed the chapter by showing the best models for each denoising engine. We found that all the denoisers retrieved better results than the conventional approach. The AWTV delivered the best results, preserving the edges while the NLM seemed capable of retrieving the original amplitudes.

CHAPTER 4

Non-linear traveltime tomography via Regularization by Denoising

We have seen that the RED can improve the results of conventional regularization for a linear cross-well traveltime tomography scenario. Now we will explore its performance for another application in geophysics, the so-called first-break tomography, which is a non-linear problem. The goal of the first-break tomography is to create a velocity model of the near subsurface.

This kind of study is used for some geotechnical applications but is also of interest to the oil industry because near-surface affects deeper reflectors (Zhu et al., 1992; Stefani, 1995). A traditional 2D acquisition places the receivers and sources over a line on the earth's surface (Claerbout, 1985), and due to the increase of seismic velocity with depth, the waves bend their raypath and return to the surface (Telford et al., 1990).

4.1 The forward problem

In order to properly state the forward non-linear problem, we need to modify equation 2.3 to

$$\mathbf{d} = \boldsymbol{\mathcal{G}}(\mathbf{m}) + \mathbf{n}. \tag{4.1}$$

Notice that $\boldsymbol{\mathcal{G}}(\mathbf{m})$ is no longer a matrix-vector product, but a function whose input is a velocity model and its output is the traveltimes, for a given source-receiver geometry.

4.1.1 Ray tracing

In the linear case, we considered that the raypaths were a straight line connecting the source and the receivers. In reality, the trajectory of the waves has a non-linear dependence on the velocity. The raypaths are "attracted" by higher velocity regions while "repelled" by the low-velocity zones, following Fermat's least traveltime principle. Hence, we need to trace the raypaths for each source-receiver pair.

The wave equation

The mathematical model that describes the behavior of body waves is the elastic wave equation. This one originates from Newton's equations of motion, and Hooke's Law, Aki and Richards (2002) provides exhaustive details of its derivation. The elastic wave equation states

$$\rho\frac{\partial^2\mathbf{u}}{\partial t^2} = (\lambda + \mu)\nabla(\nabla \bullet \mathbf{u}) + \mu\nabla^2\mathbf{u}. \tag{4.2}$$

Where ρ is the medium density, t refers to the time, $\mathbf{u}$ represents the displacements field, $\nabla \bullet \mathbf{u}$ its divergence, and the pair λ, μ refer to the first and second Lamé parameters, which are elastic constants that represent the mechanical properties of the medium.

Notice that the mechanical properties of the medium will change through space, different rock layers will have different properties. Nevertheless, in this project, we work under a high-frequency assumption, which means that the wave-lengths we consider are small relative to the rate of change of the mechanical properties. Therefore, we are not considering diffraction nor scattering (Chapman, 1976).

The high-frequency assumption allows the application of the Helmholtz Theorem to equation 4.2. In this way, we divide the wave equation into a scalar and vector potentials. The first one is associated with the P or compressional waves while the second one with the S or shear waves (Bleistein et al., 2002). At the moment we are only interested in the first ones, described by the scalar potential

$$\nabla^2 P = \frac{1}{v^2}\frac{\partial^2 P}{\partial t^2}, \tag{4.3}$$

were the scalar potential P also represents the pressure field. This expression is also known as Kirchhoff's integral solution to the scalar wave equation (Yilmaz, 2001).

Yilmaz (2001) suggests to consider a plane wave of the form

$$P = P_0 \exp\left[-i2\pi f(t - T)\right], \tag{4.4}$$

where $i = \sqrt{-1}$, f is the frequency of the wave and T is associated with the traveltime at any point in space. Equation 4.4 is in fact an harmonic solution to equation 4.3 (Rawlinson and Sambridge, 2005).

Substitution of equation 4.4 in 4.3 leads to an expression in the complex numbers domain. Its imaginary part is referred to as the transport equation while the real one is called the Eikonal equation and states

$$\left(\frac{\partial T(x,z)}{\partial x}\right)^2 + \left(\frac{\partial T(x,z)}{\partial z}\right)^2 = \frac{1}{v^2(x,z)}. \tag{4.5}$$

Since $T(x,z)$ provides the traveltime to any location in the model, setting $T(x,z) = constant$ defines a wavefront. Given that the rays are perpendicular to the wavefronts, $\nabla T(x,z)$ defines the raypath at any point in space (Chapman, 1976; Cervený, 1987).

In order to solve equation 4.5 one can decouple the problem in a system of ordinary differential equations (Bleistein et al., 2002), which leads to

$$\begin{array}{rcl} \frac{d}{dt}\mathbf{r} & = & v^2(x,z)\nabla T(x,z) \\ \frac{d}{dt}\nabla T(x,z) & = & -\frac{\nabla v(x,z)}{v(x,z)} \end{array} . \tag{4.6}$$

Where $\mathbf{r}$ is a position vector indicating the (x, z) location in the model, and $\nabla T(x, z)$ is a vector whose magnitude is given by the slowness $\frac{1}{v(x,z)}$, and its direction by the angle θ, measured with respect to the vertical. We can simplify the system in 4.6 and write it as as Vinje et al. (1993)

$$\begin{array}{rcl} \frac{dx}{dt} & = & v\sin(\theta) \\ \frac{dz}{dt} & = & v\cos(\theta) \\ \frac{d\theta}{dt} & = & -cos(\theta)\frac{dv}{dx} + \sin(\theta)\frac{dv}{dz}. \end{array} \tag{4.7}$$

This ordinary differential equation system allows us to trace the rays by solving an initial value problem, we utilized the Runge-Kutta method (Press et al., 1992) for this task.

Ray tracing methods

According to Vinje et al. (1993), Zhang and Toksöz (1998), and Rawlinson and Sambridge (2005) there exist a variety of approaches for the ray tracing problem, which we can classify into shooting method, bending method, grid-based methods, and the Wavefront Construction (WFC) method. In the next paragraphs, we will briefly describe each one of them.

Shooting Method

The shooting method starts by setting a source, initial point, and a receiver, final point. Then, it will iteratively vary the initial take-off angle θ_0, each time solving the system 4.7 until the traced ray reaches the receiver. Alternatively, it will compute a fan of rays and then attempt to interpolate the value of the take-off angle from the surrounding rays. The shooting method is not an effective technique because of the dependence of the raypath on the slowness is highly non-linear.

Bending method

The bending method will reformulate the ray tracing problem as (Scales, 1987)

$$t(ray) = \int_{ray} \frac{1}{v(x,z)} dl \tag{4.8}$$

and solve for the take-off angle as in an LS problem. The main drawback of this formulation is the calculation of the sensitivity matrix **G**, once more because of the non-linearity of the problem. Moreover, this method struggles with the multipathing associated to complex velocity models (Vinje et al., 1993; Rawlinson and Sambridge, 2005).

Grid based methods

Grid-based methods solve the Eikonal equation through finite differences. This approach is attractive because they can simulate wave propagation in all the models, find diffractions, refractions, and illuminate shadow zones (Zhang and Toksöz, 1998; Vinje et al., 1993).

Zhang and Toksöz (1998) sustains that grid methods have had significant developments in three areas, the first one is solving an Eikonal equation by Finite Differences (Vidale, 1988) and more recently the Fast Marching Method (Rawlinson and Sambridge, 2005). The second is applying an analytical solution to expand the wavefront (Vinje et al., 1993). The last one is utilizing graph theory to find the shortest path (Moser, 1989)

4.1.2 The Wavefront construction method

As we have seen, efficient ray tracing in a complex velocity model is not easy (Yilmaz, 2001), and even the grid-based method cannot deal with multipathing. Nonetheless, inspired by the work of Moser (1989) on the shortest path method, Vinje et al. (1993) proposed the Wavefront Construction (WFC) method, which takes elements from the shooting method and graph theory. It has the power of grid methods for simulating the wavefront in all the models, covering shadow zones, and accounting for refractions and discontinuous interfaces. Additionally, unlike all the previous methods, it can account for multipathing and multiarrivals.

WFC starts by solving the equation system 4.7 as an initial point problem to compute the expanding wave from a single source in all directions. The points composing the wavefront become the new initial condition of the problem, and the process is repeated. Eventually, these points will diverge, and, as shown in Figure 4.1 (left), we will require to interpolate a new ray. A third-degree polynomial approximates the wavefront between the points, and a new initial point is established across it (Vinje et al., 1993).

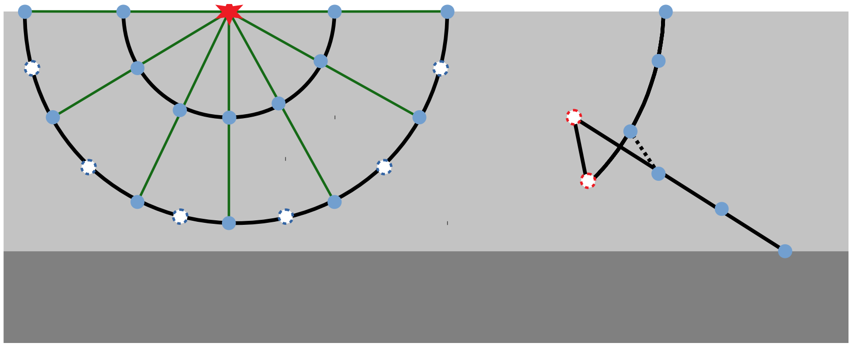

Figure 4.1: At left, interpolation of new rays (in red). At right, elimination of of rays (dotted) due to the wavefront crossing itself.

The wavefront will also eventually leave the boundaries or the model or cross itself when this occurs, the rays that track the wavefront must be terminated. An example of this situation is when the head-wave overcomes the direct arrivals. In this situation, we need to stop tracing the section of the wavefront involved in the crossing, as sketched in Figure 4.1 (right).

The WFC method can deal with complex velocity models in a reasonable amount of time. This capability becomes essential during the inversion because, according to Zhang and Toksöz (1998), the raytracing step is the most time-consuming in the non-linear seismic traveltime tomography. Finally, the

WFC method is simple. It only requires a small calibration of the parameters before running. The first one is the maximum distance between points in the wavefront, which triggers the interpolation of new rays. The second one is the initial number of rays. Too many or too few may cause the algorithm to have problems detecting wavefront self-crossings. An example of the application of the WFC method for a complex velocity model is presented in Figure 4.2.

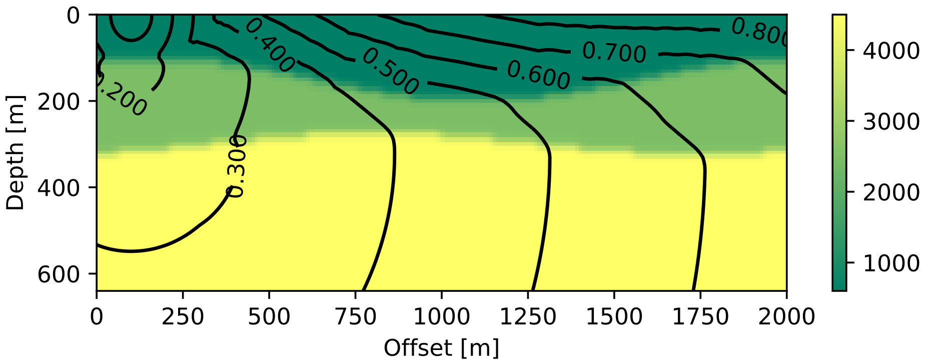

Figure 4.2: Example of the WFC results for a complex velocity model.

4.2 The inverse problem

Once the problem of ray tracing has been solved, and we have a way to compute $\boldsymbol{\mathcal{G}}(\mathbf{m})$ for any velocity model, we can focus on the inversion problem.

4.2.1 Linearization of the forward problem

Once more, the inverse problem consists of minimizing a cost function by calculating its derivative and setting it equal to zero. We want to avoid dealing with the derivatives of the non-linear operator. Therefore, we use the Taylor Series expansion to linearize the problem. Taylor Series allows to express any function $f(x)$ as

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}(x-x_0) + \frac{f''(x_0)}{2!}(x-x_0)^2 + \frac{f'''(x_0)}{3!}(x-x_0)^3 \cdots . \quad (4.9)$$

Hence, we can linearize the function $\boldsymbol{\mathcal{G}}(\mathbf{m})$ around an initial model $\mathbf{m}_0$ and we obtain

$$\boldsymbol{\mathcal{G}}(\mathbf{m}) \approx \boldsymbol{\mathcal{G}}(\mathbf{m}_0) + \mathbf{J_0}(\mathbf{m} - \mathbf{m}_0). \tag{4.10}$$

Where $\mathbf{J}_0 = \frac{\partial \boldsymbol{\mathcal{G}}(\mathbf{m})}{\partial \mathbf{m}}|_{\mathbf{m}=\mathbf{m}_0}$ is the Jacobian or sensitivity matrix.

4.2.2 The cost functions

We need to adapt the cost function on equation 2.36, however, the derivatives will change now due to the extra terms from equation 4.10. For the data misfit term, we now have

$$\phi_{\mathbf{d}}(\mathbf{m}) = ||\mathbf{d} - \boldsymbol{\mathcal{G}}(\mathbf{m}_0) - \mathbf{J}_0\mathbf{m} + \mathbf{J}_0\mathbf{m}_0||_2^2. \tag{4.11}$$

Deriving 4.11 with respect to $\mathbf{m}$ leads to

$$\frac{\partial \phi_{\mathbf{d}}(\mathbf{m})}{\partial \mathbf{m}} = 2[\mathbf{J_0}^T\mathbf{J_0}(\mathbf{m} - \mathbf{m}_0) - \mathbf{J}_0^T(\mathbf{d} - \boldsymbol{\mathcal{G}}(\mathbf{m}_0))]. \tag{4.12}$$

If we consider $\mathbf{m} = \mathbf{m}_0 + \Delta\mathbf{m}$, i.e. that the future model is equal to the initial model used for the linearization, plus a perturbation, we finally obtain

$$\nabla \phi_{\mathbf{d}}(\mathbf{m}) = 2[\mathbf{J_0}^T\mathbf{J_0}\Delta\mathbf{m} - \mathbf{J}_0^T(\mathbf{d} - \boldsymbol{\mathcal{G}}\,(\mathbf{m}_0))]. \tag{4.13}$$

The importance of the linearization is that the Jacobian or sensitivity matrix is equalt to G when the problem is linear (Constable et al., 1987).

Non-Linear Tikhonov case

In the same way, as we did before, we can define two equations for the non-linear tomography inversion. One for the Tikhonov Regularization and another one for RED. For the first one we require now to use the gradient of the Tikhonov Regularization term computed in the rightmost side of equation

2.21

$$\frac{\partial}{\partial \mathbf{m}}\|\mathbf{Rm}\|_2^2 = 2\mathbf{R}^T\mathbf{Rm}. \tag{4.14}$$

When substituting $\mathbf{m} = \mathbf{m}_0 + \Delta\mathbf{m}$ we obtain

$$\nabla\phi_{\mathbf{m}}(\mathbf{m}) = 2[\mathbf{R}^T\mathbf{R}\mathbf{m}_0 + \mathbf{R}^T\mathbf{R}\Delta\mathbf{m}], \tag{4.15}$$

as the gradient of the Tikhonov regularization term for the non-linear case.

Now, if we merge equations 4.13 and 4.15, and make them equal to zero, we obtain,

$$[\mathbf{J_0}^T\mathbf{J_0} + \mu\mathbf{R}^T\mathbf{R}]\boldsymbol{\Delta}\mathbf{m} = \mathbf{J_0}^T[\mathbf{d} - \mathcal{G}(\mathbf{m_0})] - \mu\mathbf{R}^T\mathbf{R}\mathbf{m}_0. \tag{4.16}$$

Once more, the details on the usage of $\mathbf{R}$ for a 2D model are included in Appendix A.

Notice that this equation 4.16 solves for $\Delta\mathbf{m}$ instead of $\mathbf{m}$. Thus, we have to solve this problem iteratively. We start with the initial model $\mathbf{m}_0$ and update it in each iteration by adding the computed perturbation. We solved for the perturbation using the LSCG method. Additionally we utilized the Levenberg-Marquardt criterion (Marquardt, 1963), which adds an $\alpha\mathbf{I}$ term to the left-hand side of equation 4.16 to steer the model updates towards the SD direction (Constable et al., 1987; Tarantola, 2005).

Non-Linear RED case

We already computed in equation 2.45 the gradient of the regularization term of RED as $\frac{\partial\rho(\mathbf{m})}{\partial\mathbf{m}} = 2\left[\mathbf{m} - f(\mathbf{m})\right]$. When considerin $\mathbf{m} = \mathbf{m}_0 + \Delta\mathbf{m}$, this equation becomes

$$\frac{\partial\rho(\mathbf{m})}{\partial\mathbf{m}} = 2\left[(\mathbf{m}_0 + \Delta\mathbf{m}) - \mathbf{f}(\mathbf{m})\right]. \tag{4.17}$$

Combining this equation with the gradient of the data misfit for the non-linear case obtained in equation 4.13, one can finally write

$$[\mathbf{J}_0^T\mathbf{J}_0 + \mu\mathbf{I}]\Delta\mathbf{m} = \mathbf{J}_0^T(\mathbf{d} - \boldsymbol{\mathcal{G}}\left(\mathbf{m}_0\right)) - \mu(\mathbf{m}_0 - \mathbf{f}(\mathbf{m})). \tag{4.18}$$

Notice that this equation requires to apply the denoising engine in the updated model. Romano et al. (2017) suggest to face this situation by using the fixed-point strategy which allows us to consider $\mathbf{f}(\mathbf{m}) \approx \mathbf{f}(\mathbf{m}_0)$, finally leading to

$$[\mathbf{J}_0^T\mathbf{J}_0 + \mu\mathbf{I}]\Delta\mathbf{m} = \mathbf{J}_0^T(\mathbf{d} - \boldsymbol{\mathcal{G}}\left(\mathbf{m}_0\right)) - \mu(\mathbf{m}_0 - \mathbf{f}(\mathbf{m}_0)). \tag{4.19}$$

The approach used in equations 4.16, 4.19, of solving a non-linear problem by iterative perturbing an initial guess is called ”creeping”, because it slowly converges by making small steps towards the final solution. It is possible to use an alternative approach called ”jumping”, which solves directly for the updated model, supposedly in a faster fashion.

Jumping offers a more natural approach by imposing the constraints on the actual model instead of in $\Delta\mathbf{m}$. However, if the initial model is nearby the optimal one, creeping will offer better control on the step size and avoid jumping over the minimum of the cost function (Scales and Gersztenkorn, 1990).

We would like to mention at this point that the application of RED on non-linear problems is a novelty, the only bibliography on this subject at this point is the one of Anagaw and Sacchi (2020), which is still on progress and it is on a significantly different type of geophysical problem. Romano et al. (2017) only provided examples for linear cases. Therefore, the regularization of the inverse problem posed by equations 4.16, 4.19 through RED, as well as its numerical implementation via FP is, as far as we are aware, a new topic.

4.2.3 The Levenberg-Marquardt method

A well-known numerical optimization method for minimizing a vector-valued function, like $h(\mathbf{m})$, is the Newton method, which can be described by the equation

$$\mathbf{H}_0 \Delta\mathbf{m} = -\nabla h(\mathbf{m}_0), \tag{4.20}$$

where $\mathbf{H}_0$ is the Hessian matrix containing the second derivatives of $h(\mathbf{m})$, valuated at the model $\mathbf{m}_0$. This initial model is iteratively updated with the computed perturbations until a point where $\nabla h(\mathbf{m}_0) = 0$ is reached. This method has quadratic convergence, which means that the number of accurate digits double each iteration (Aster, 2013).

A variation of the Newton method for non-linear LS is the Gauss-Newton (GN) method. This one has the advantage of avoiding the Hessian calculation through the approximation $\mathbf{H} \approx 2\mathbf{J}^T\mathbf{J}$, substituting this expression in equation 4.20 leads to

$$\mathbf{J}_0^T \mathbf{J}_0 \Delta\mathbf{m} = \mathbf{J}_0^T(\mathbf{d} - \boldsymbol{\mathcal{G}}(\mathbf{m}_0)). \tag{4.21}$$

Applying this expression to a cost function with the Tikhonov regularization term yields another way of obtaining equation 4.16 (Constable et al., 1987; Zhang and Toksöz, 1998; Sun et al., 2018).

The Levenberg-Marquardt (LM) method is a variation of the GN method introduced by Marquardt (1963) as a way to improve convergence rate. Equations 4.16, 4.19, and 4.21 can be expressed in a general form as $\mathbf{A}\Delta\mathbf{m} = \mathbf{b}$. The idea of LM is to introduce an extra $\alpha\mathbf{I}$ term in the left-hand side of this equation so it becomes

$$(\mathbf{A} + \alpha\mathbf{I})\Delta\mathbf{m} = \mathbf{b}. \tag{4.22}$$

The goal of LM is to accelerate the convergence rate of the problem stated in equation 4.22 by escalating the perturbation magnitude. Additionally, the added term further stabilizes the system by avoiding singularity and steer iterative updates toward the direction of steepest descent (Aster, 2013). The parameter α changes through the iterations altering the convergence rate to ensure that the updated model advances towards the minimum. On the one hand, when it is large, the method follows an SD update, performing small steps to avoid overshooting the minimum. On the other hand, when α is small, the method goes back to a GN nature and advances aggressively towards the minimum. In this way, one reduces the possibility of overshooting the minimum and computing further iterations.

4.3 Numerical examples

In this section, we introduce some numerical examples of tomographic inversion. We start with a small scale model and present some of the difficulties we stumble upon when solving the non-linear problem, such as the step size for the perturbations and the initial model. Later, we propose a more complex synthetic real model and perform the tomographic inversion on it.

4.3.1 Step size

As mentioned afore the tomographic inversion addressed in this project requires for the iterative solution of equations 4.16 and 4.19 by computing the parameter model perturbation $\Delta\mathbf{m}$. During our first tests performing the tomographic inversion, we struggle to determine the magnitude of this perturbation, i.e., the step size towards the minimum of the cost function.

The value of $\Delta\mathbf{m}$ was too large, and the system of equation seemed unstable. Large steps towards the minimum can cause overshooting, recalibration of the step size, and more forward problem computations. This situation is undesirable because it slows the inversion process, Zhang and Toksöz (1998) sustain that this is the most time-consuming step during the tomographic inversion.

The LM method provided a solution for step size and instability problems. As suggested by Aster (2013), we introduced a variable $\alpha\mathbf{I}$ term in the matrix term of the left-hand side of equations 4.16 and 4.19. According to Zhang and Toksöz (1998), who refers to $\alpha\mathbf{I}$ as a damping term to control the size of the model updates, this additional term should not affect the final retrieved model.

In order to show the effect of the additional term introduced by the LM method, we designed an experiment. We proposed a true and initial model composed of two layers. The first one has the interface at a 100 [m] depth, while the later one has it at 50 [m] depth. The models appear in Figure 4.3a and 4.3c respectively. They are composed of $100 \times 30 = 3000$ square cells of 10 [m] per size and illuminated by 13 sources and 100 receivers separated by intervals of 80 [m] and 10 [m] respectively.

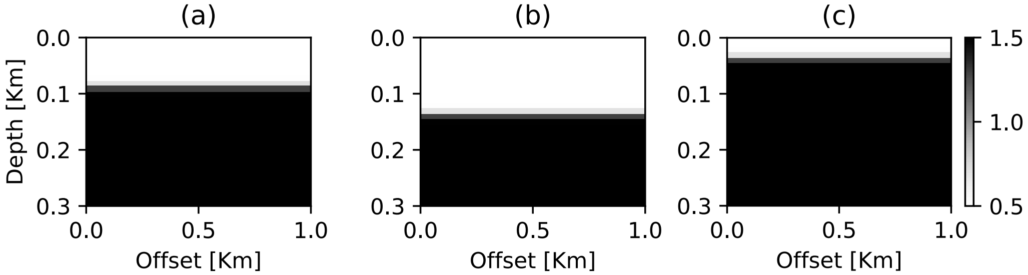

Figure 4.3: (a) simple true velocity model. Simple initial models, with interface above, (b) and below (c) the true one. The velocities are in Km/s

We computed one perturbation with the Tikhonov regularization, applying the LM extra term, and using 6 different values for α: $0, 10^{-3}$, 10^{-2}, 10^{-1}, 10^{0}, and 10^{1}. We present the $\mathbf{\Delta m}$ for each one of these cases in Figure 4.4. One can notice that the computed $\mathbf{\Delta m}$ are not scaled versions of each other.

An explanation to the results in Figure 4.4 is that the value of α alternates the updating direction between the SD and the one of GN. The first one is simply the gradient direction, while the second one uses the information of the curvature to converge faster. Another interpretation might be that at this stage, the problem is not sufficiently stabilized by the regularization term. This makes sense if we consider that $\mathbf{Rm}_0 \approx \mathbf{0}$ and $\mathbf{m}_0 - f(\mathbf{m}_0 \approx 0$ for the initial model in Figure 4.3c. As pointed out by Aster (2013), LM can stabilize the problem when the regularization term is not enough.

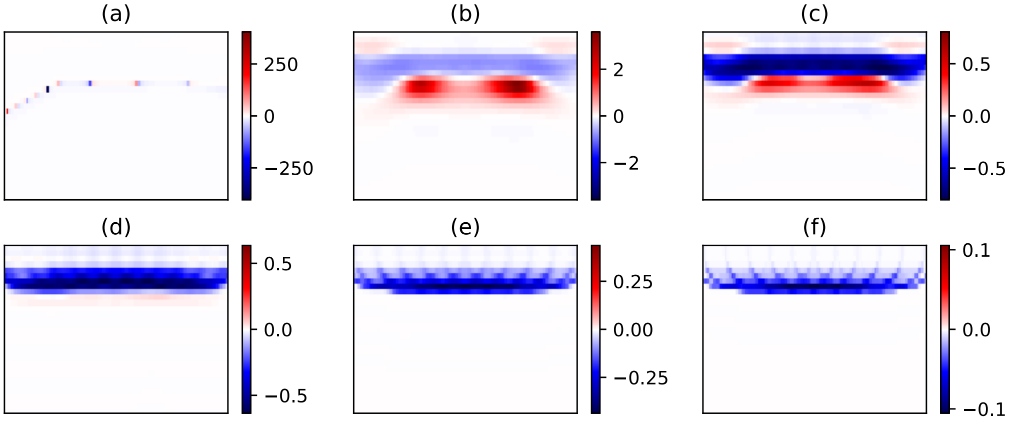

Figure 4.4: $\Delta\mathbf{m}$ for the first iteration using Tikhonov regularization and different α values: (a) 0, (b) 10^{-3}, (c) 10^{-2}, (d) 10^{-1}, (e) 10^{0}, and (f) 10^{1}. The velocities perturbations are in Km/s

We had to rely on a large value of α to stabilize the problem during the first iterations. Nevertheless, this parameter is dynamic and adjusts its value through iterations, decreasing its magnitude each time the new model reduces the cost function value and increasing it when it does not. Using the initial model in Figure 4.3c, we ran 15 iterations using three different α values: 10^{-2}, 10^{-1}, and 10^{0} to display how the convergence rate is affected by this parameter and how it updates its value during the inversion process. In this test, the decreasing factor for α was 0.9 and the increasing one 5.

Figure 4.5 shows that the first part of the inversion takes successive accepted models leading to smaller a α and larger strides towards the minimum. Later, when the step becomes too large, and the cost function closer to its minimum, the algorithm rejects the perturbation, and α increases its value to perform smaller steps. Eventually, either the iterations end or the perturbation becomes neglectable, so the algorithm stops. This test tells us that the initial value of α should be large enough to stabilize the inversion but small enough to maintain a fast convergence rate.

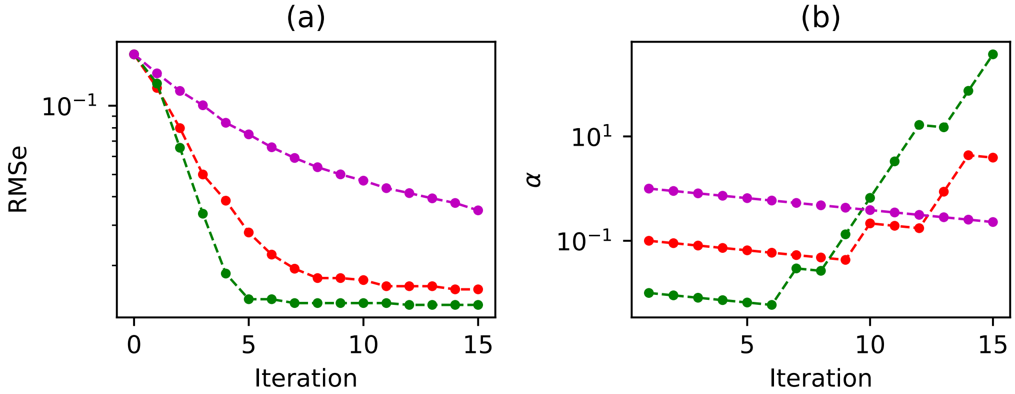

Figure 4.5: Convergence rate using three different initial α values: magenta $\alpha = 10^0$, red $\alpha = 10^{-1}$, and green $\alpha = 10^{-2}$. (b) changes in the α value through iterations.

4.3.2 Initial model

A common problem in non-linear optimization is the influence of the initial model on the final solution. The wrong initial model can lead to a local minimum. For this reason, we designed another test and proposed a second initial model, which appears in Figure 4.3b, where the interface is 50 [m] below the original one in Figure 4.3a. Once more, we executed our inversion algorithm, which produced the final models shown in Figure 4.6. Here, one can observe that while the first initial model converges, the second one does not. This problem becomes clearer in Figure 4.7, which shows the effect of the sum of all the perturbations.

We explain the effects exhibited in Figure 4.7 with Fermat's least time principle and the velocity gradient. We noticed that when the velocity perturbations are negative, the raypaths of the updated model avoid the new low-velocity region and take a more profound new trajectory. This change allows for a gradual coverage of more cells, which permits a more homogeneous perturbation of the velocity model.

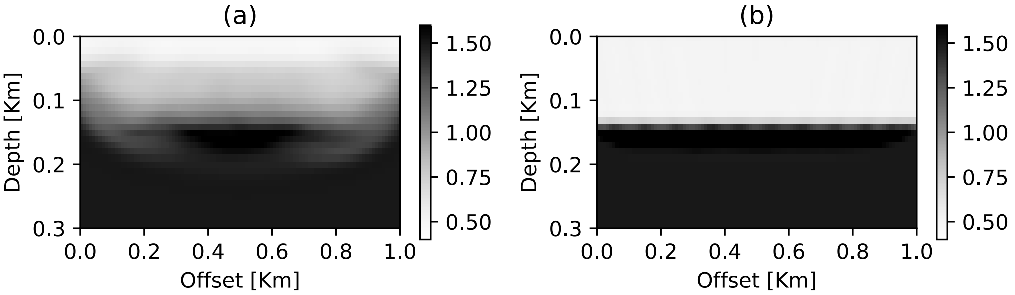

Figure 4.6: Effect of the initial model. (a) and (b) are both final models, the first one uses 4.3c as initial model, while the second uses 4.3b as its initial guess.

In the opposite case, when a positive velocity perturbation occurs, the rays crossing those cells do not want to change its trajectory. Even more, nearby rays will bend its path towards this new high-velocity region to reduce their traveltimes. If only positive velocity perturbations occur, the effect will build up during iterations, making the inversion process to sample only a handful of the model's cells. This effect forces the algorithm to explain the data misfit with a small portion of cells, causing it to fall into a local minimum.

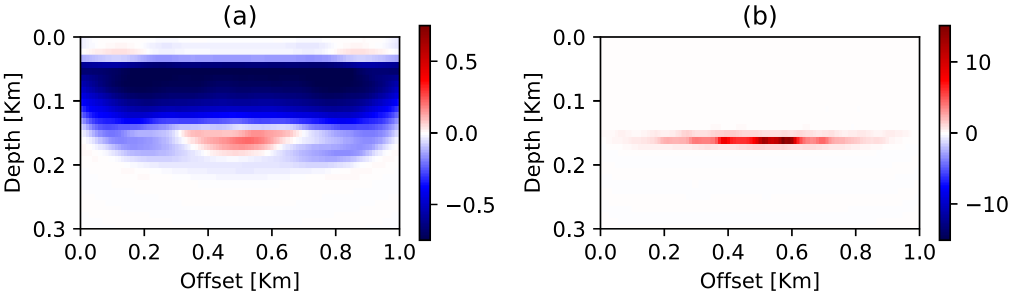

Figure 4.7: Total change produced by the model perturbations when the initial model is (a) 3.6a, and (b) 3.6b. One can appreciate the convergece/divergence caused by the initial model selection.

The tests above allowed us to encounter and solve numerical difficulties before facing a more complex problem. One should design the initial model to avoid only positive velocity perturbations. Of course, this might not be possible since the true model is unknown. Nonetheless, if the model perturbations start to construct a pattern like the one in Figure 4.7b, one could safely assume that the initial model needs to be changed.

4.4 Tomographic inversion with synthetic data

Now that we have solved these first issues of the numerical implementation, we propose a new true model closer to a real geological scenario. Panel (a) of Figure 4.8 shows this new true velocity model, while panel (b) displays the initial one. Both models are composed by three layers of 0.5 Km/s, 1.5 Km/s and 3.0 Km/s, and discretized in 200 × 50 = 10000 square cells of 10 [m] per size, and illuminated by 21 sources and 201 receivers (4221 sx-rx pairs). The sources are separated by 100 [m] intervals while the receivers by 10 [m] ones.

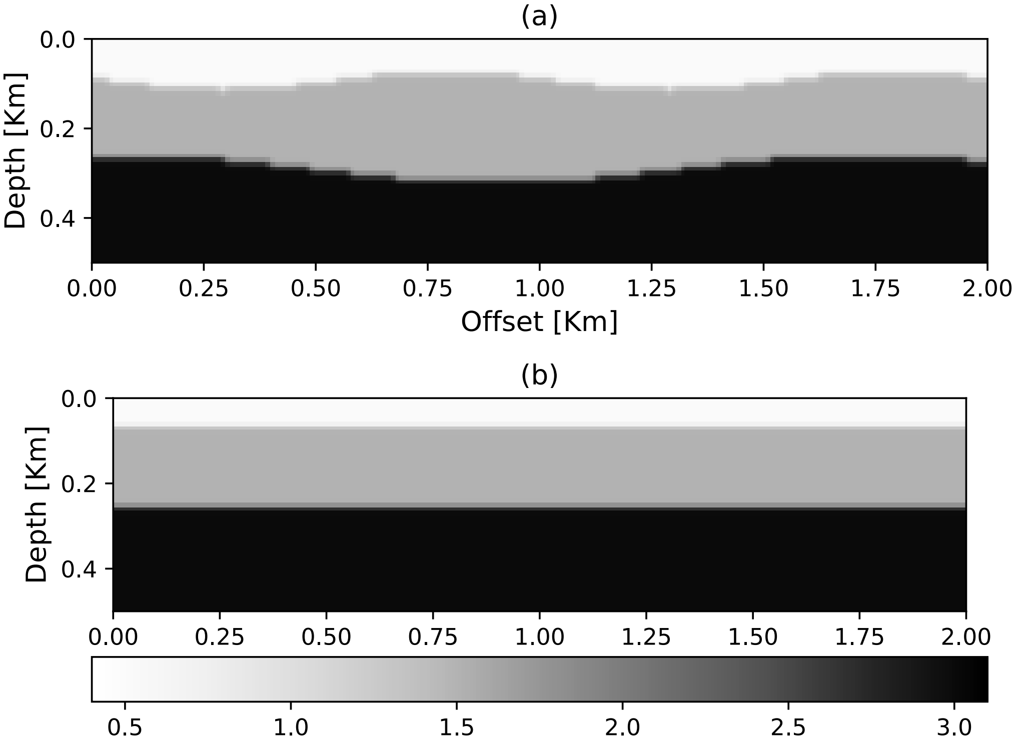

Figure 4.8: (a) true and (b) initial complex velocity models. Velocities in Km/s

As mentioned before, forward modeling is expensive in time. Thus, we took a couple of considerations to reduce it. First, we compute a single case of the Tikhonov Regularization, and we use an initial model that closely resembles the true one. Solving a non-linear inverse problem is a significant task by itself. We instead focus our attention on the feasibility of using denoising algorithms as regularization terms. Furthermore, when solving a problem of this nature with real data, one could dispose of suitable initial models from previous seismic data processing steps.

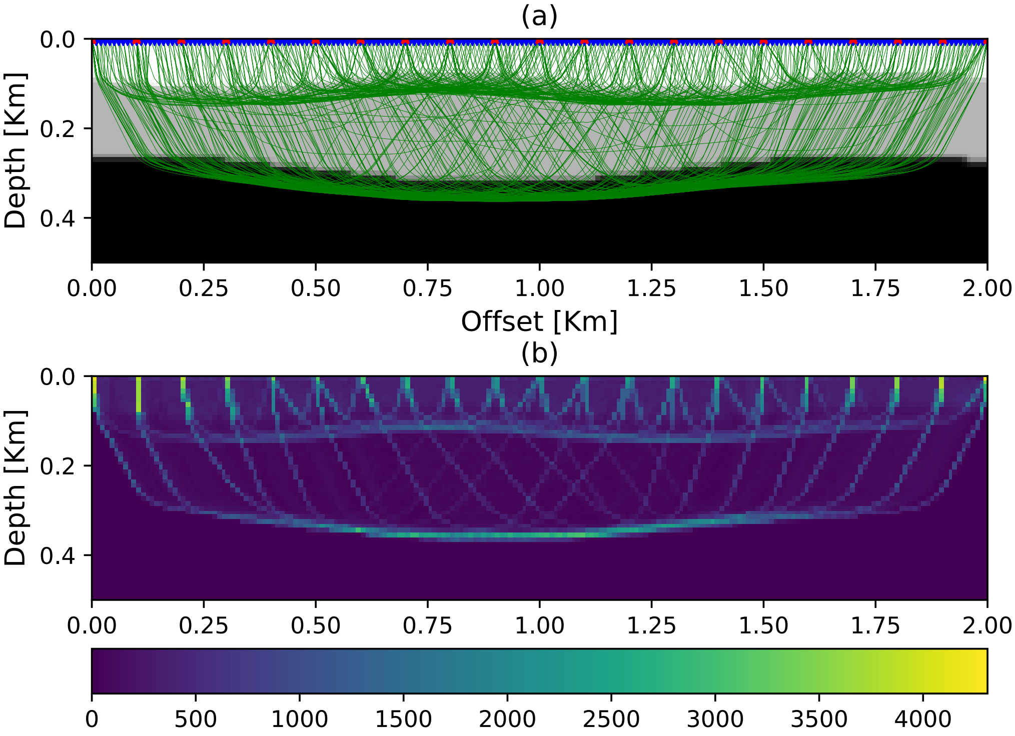

Figure 4.9: (a) raypath coverage plotted over the true velocity model and (b) ray fold for the same model. Only a fifth of the total raypaths is displayed.

Execution of the forward problem on the true model allows us to observe the raypath coverage, shown in Figure 4.9a. From this image, one can notice that there are regions of the model which are not covered and thus will not affect the traveltimes. Panel (b) of the same figure shows the ray fold, which is the technical term for the number of rays crossing each cell. This analysis is relevant because it shows which areas of the velocity model will have the most influence on the computed velocity perturbations.

4.4.1 Inverted velocity models

In order to determine the value of the trade-off parameter, we computed an L-curve for each of the four cases (Tikhonov, Median, NLM, and AWTV). We tested five values for each case. For the Tikhonov one, we split the trade of parameter in two, a horizontal and a vertical one. We gave more five times more importance to the horizontal component since we were expecting a layer-like structure.

The μ values tested for the L-curves in Figure 4.10a were 10^{0}, $10^{-0.5}$, 10^{-1}, $10^{-1.5}$, and $10^{-2.0}$, for the Tikhonov case. For RED-median in panel b of the same Figure we had 0.04, 0.02, 0.01, 0.005, and 0.0025. For RED-NLM in the same figure we tested 0.6, 0.30, 0.15, 0.075, and 0.0375. Finally for RED-AWTV in Figure 4.10d we tried 10^{0}, $10^{-0.5}$, 10^{-1}, $10^{-1.5}$, and 10^{-2}. The traveltimes had an added Gaussian noise with variance of 10 [mS], and this level appears as a dashed line in Figure 4.10a. The three RED results however did not manage to reach the same level of fit with a well-looking model, so for them we settled for a value of 35 [mS].

We computed the tomographic inversion for four cases whose results we display in Figure 4.11. Panel (a) shows the outcome of the flat Tikhonov regularization, (b) the one of RED using the median filter, (c) RED using NLM, and (d) RED using TV. We configured a stop criterion of a maximum of 50 iterations or velocity perturbations smaller than 1 m/s in absolute value.

The models retrieved by the tomographic inversion largely resemble the true one. For the Tikhonov case, we decided to give a more significant magnitude to the horizontal derivatives since we expected the inverted model to have a somewhat horizontal-layer structure. The split trade of parameter gave

$\mu_x = 10^{-1}$ for the horizontal component and $\mu_z = 10^{-2}$ for the vertical one. We set the parameter α of the LM method to 10^0.

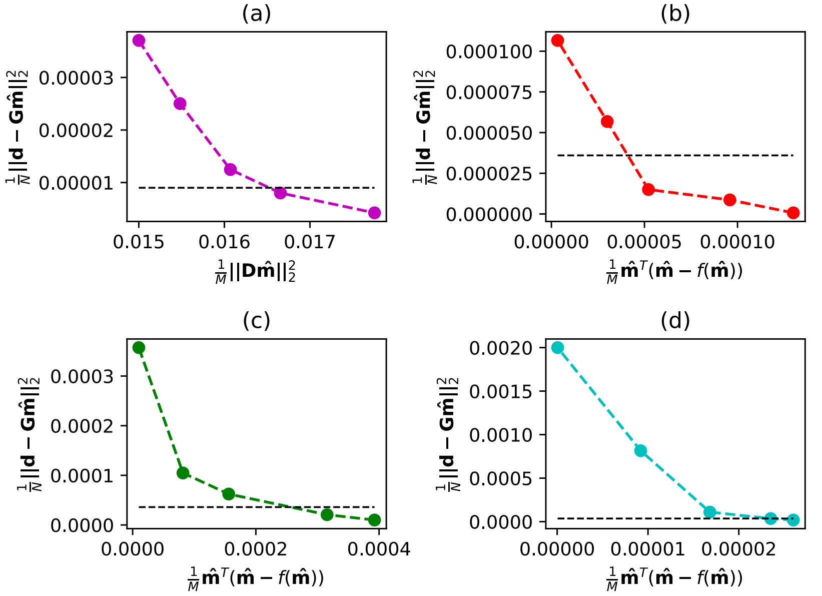

Figure 4.10: L-curves for the (a) Tikhonov, (b) median, (c) NLM, and (d) AWTV cases. Each point represents a μ value and the dashed line the noise level.

The layers composing the velocity model were correctly retrieved, having only a small artifact in the lower interface around the 1.75 Km offset. Similar artifacts appear in the publications from Zhang and Toksöz (1998), and Sun et al. (2018) and presumably occur due to the lack of ray coverage at the edges of the model.

Our tests provided the best results for the RED-median filter method when the trade-off parameter was equal to 0.02, $\alpha = 10^0$, and the window size of the denoiser set to 3×3. This simple denoising algorithm was able to successfully stabilize the inversion and produce a velocity model that honors the traveltimes, proving right the statement of Romano et al. (2017) of using RED with an arbitrary denoiser as a general method for solving inverse problems. One can appreciate the effects of the median filter enforcing some edge-preserving like features on the interfaces between the 1.0 Km and 1.5 Km offsets. Alas, there is an inadequate definition at the central section of the lower interface, and the inverted model presents the same artifact as the Tikhonov case.

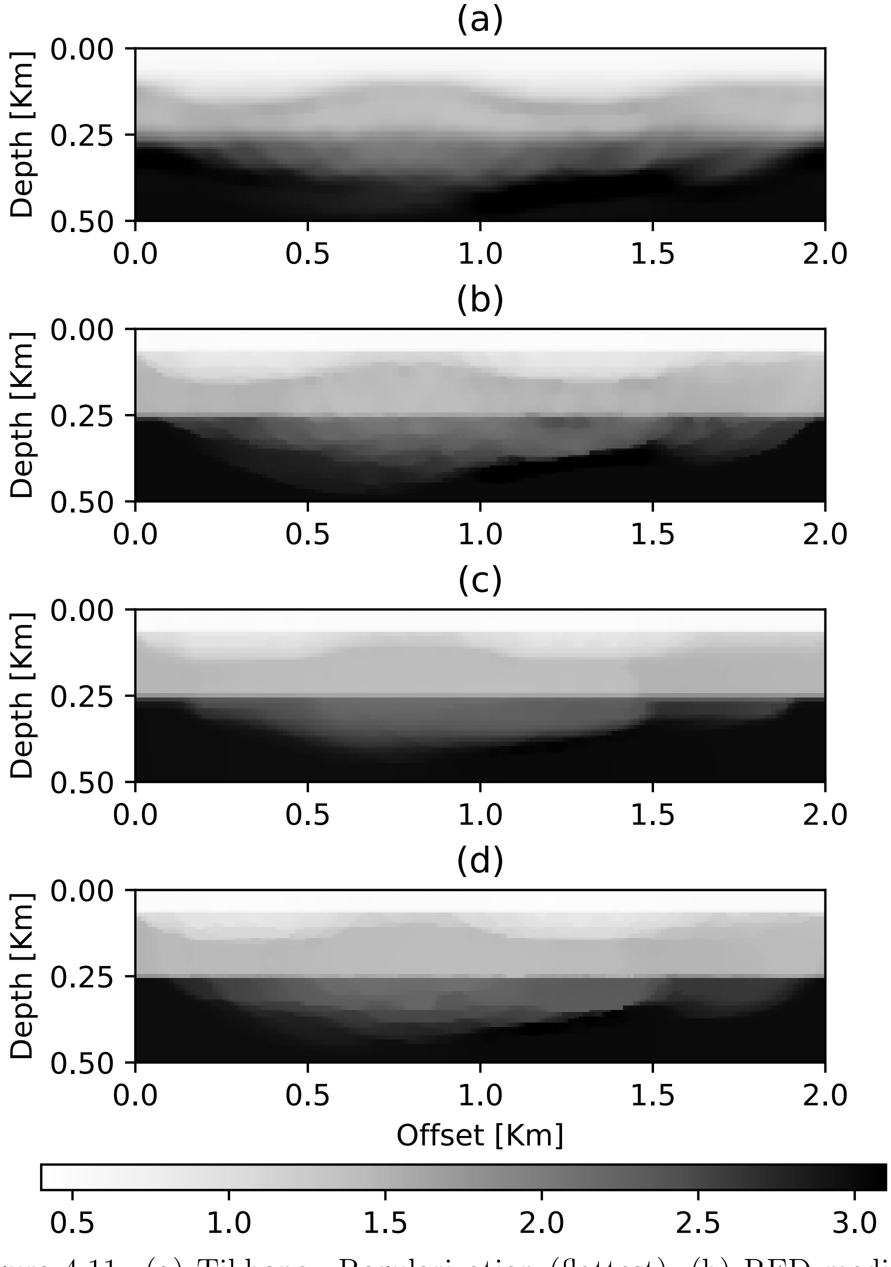

Figure 4.11: (a) Tikhonov Regularization (flattest), (b) RED median filter, (c) RED-NLM, and (d) RED-AWTV inverted models. The velocities are in Km/s.

The inverted model obtained trough RED-NLM was expected to be the best one. The denoiser seems to promote homogeneity within the layers, making this model lack the raypaths tracks. Nevertheless, the interfaces look blurrier than expected, and the depressions on the upper interface are hardly defined. On a positive note, it avoids the artifact of the previous two inversions. The tests with this method gave the best results when $\mu = 0.15$, $\alpha = 10^0$, search window set to 21×21, similarity window to 5×5, degree of filtering to 0.015, and a Gaussian kernel for the NLM.

Finally, for the RED-AWTV model, we set the parameters to $\mu = 10^{-1}$, $\alpha = 10^0$, the trade-off the denoiser to 10^{-2}, and the scale factor of the weights to $\gamma = 0.06$. The inverted model resembles the true one, yet its interfaces are blurry, particularly the deeper one. Similarly to the NLM case, it shows some homogeneity within the layers, and diminishment of the artifact at the 1.75 Km offset mark. We can see the well-known piece-wise promoting behavior of the TV denoiser in the lower interface around the 1 Km offset.

4.4.2 Convergence rates and RMSe

Figure 4.12 displays the Root Mean Square error (RMSe) decrease with the iterations for each one of the inverted models. One can promptly see that the four decay rates are similar, and the differences among them are probably only related to the differences in their trade-off parameter value. The Tikhonov regularization scenario reaches its minimum faster than any RED case. We explain this by the lesser number of denoiser activations for the non-linear case when compared with the linear one. The FP approach of the linear problem allowed for several denoising activations using the same kernel. Unfortunately, the non-linear approach of equations 4.16 and 4.19 permits only one denoising operation.

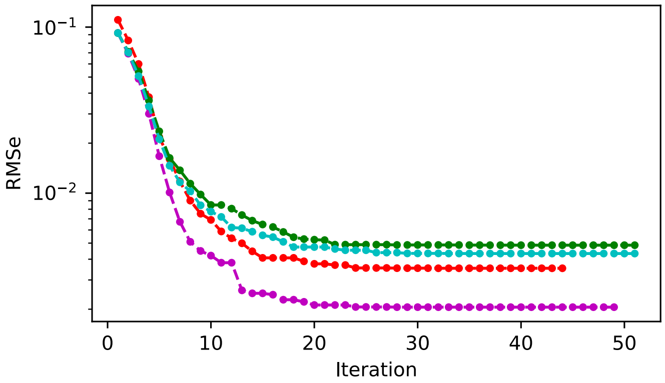

Figure 4.12: RMS decay with respect to iterations. Magenta corresponds to the Tikhonov case, red to the RED-median, green to RED-NLM, and cyan to RED-AWTV.

4.4.3 Residual traveltimes

A comparison between the observed traveltimes and the computed ones for the final velocity models of each of the four inversions appears in Figure 4.13. In general terms, all of the inverted models produce traveltime curves that reasonably fit the observed ones, as one can recall from chapter two, non-uniqueness in the solution is a common occurrence when solving inverse problems.

Although the variations on the final RMSe for each regularization are proof of the differences in the traveltimes for each case, the four panels of Figure 4.13 exhibit the same problems. The inverted dromochrones struggle to fit the sharp jumps in the true model observed travel tines at (0.4 Km, 0.8 s), (0.8 Km, 0.9 s), (1.5 Km, 0.9 s), and (1.6 Km, 0.9 s). This problem is no surprise since these jumps are probably related to the more delicate features of the velocity model.

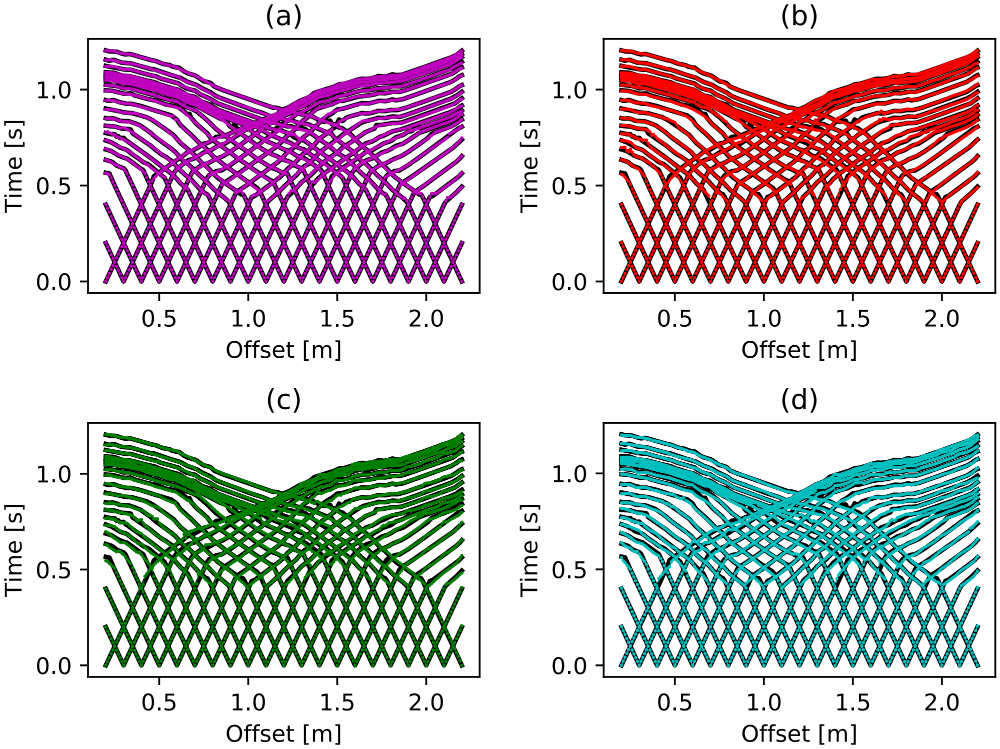

Figure 4.13: Observed traveltimes and computed traveltimes for the final models. (a) Tikhonov, (b) RED-median, (c) RED-NLM, and (d) RED-TV cases

Finally, in the four panels of Figure 4.14, we can appreciate the distribution of the residuals for each of the computed solutions. All of the histograms present a satisfactory zero-mean normal distribution and a reasonable standard deviation, around 3 [ms] for Tikhonov and RED, and around 6 [ms] for NLM and AWTV. Once more, this difference is most likely related to the different magnitude of the trade-off parameter.

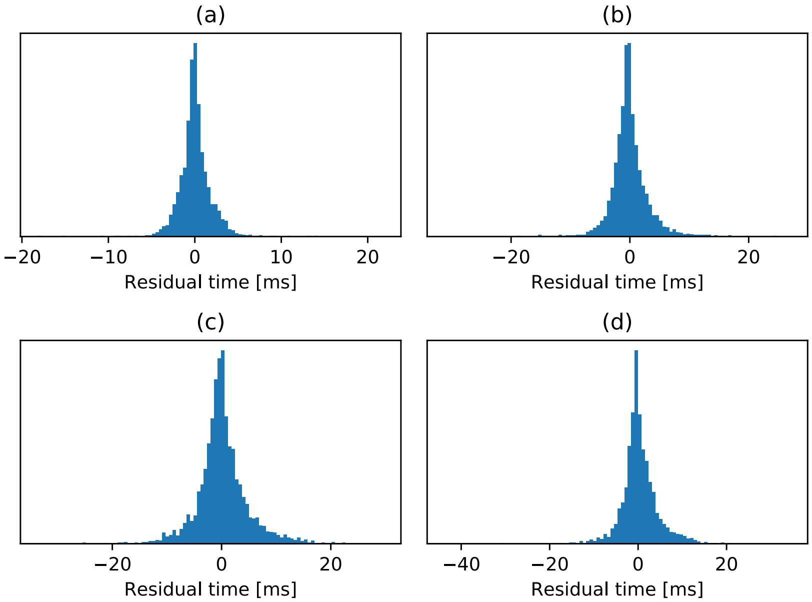

Figure 4.14: Residual distribution for the (a) Tikhonov, (b) RED-median, (c) RED-NLM, and (d) RED-AWTV cases.

4.5 Summary

In this chapter, we explored the application of RED, initially proposed for linear inverse problems, to the non-linear case of first-break tomography. We started by describing how to solve the non-linear forward problem and explaining the ray tracing theory and methods. From the available methodology, we selected the WaveFront Construction.

We explained how to linearize the forward problem in order to solve for a small model perturbation. After that, we introduce the Levenberg-Marquardt method as a tool to further stabilize the inversion and speed up convergence. Later, we consider some issues on the numerical implementation like the step size and initial model selection.

Finally, we applied the non-linear tomographic inversion on a complex synthetic model. We obtained four results, one for the flat Tikhonov regularization, and three using RED and the median filter, NLM, and TV denoisers. We measured the solution's quality by comparing their RMSe values, their model misfit, and the distributions of their residuals.

CHAPTER 5

Discussion and Conclusions

5.1 Discussion

The implementation of RED for a linear problem is straightforward and appealing to apply to inverse problems in other fields. The non-linear case, however, presents difficulties, presumably, due to the inability to use the full power of the denoiser. Nevertheless, the application of denoising engines in the velocity models may facilitate promoting features of interest.

The two flat layers numerical example showed in chapter four showed that the Tikhonov Regularization excelled in the interface shifting case since it could easily constrain the solution to have no lateral velocity changes. RED, however, performed poorly in this scenario. It tended to develop artifacts ad the edges of the model, where the ray coverage ended.

Another experiment, not included in this work, was another model with two layers but resembling a cavity. For this case, however, Tikhonov needed careful calibration of the trade-off parameters (horizontal and vertical) to retrieve a decent model. RED, on the other hand, allowed for a more robust treatment in the trade-off parameter and was able to image the cavity and to remove artifacts in the inverted model through the denoiser. The median filter performed particularly well in this test. The cavity model, or one composed of thickness-varying layers, is more realistic and exciting, scenario than a set of horizontal layers. In this sense, we consider that RED has the upper hand in performance.

We would like to make some comments on the importance of selecting an adequate initial model. The initial guess is fundamental to find the solution of a non-linear problem. Gradient descent methods such as the one implemented in this work require the initial guess to be near the global optimum to avoid local minima. Nevertheless, in a multidimensional problem like this one, it might be unclear to know if one is close to the global optimum. We discover that positioning the initial model's interfaces above the ones of the true model provided significantly better results, i.e., the initial must only lower its velocities in each iteration.

We attribute the observation afore to the dependence of the raypaths on the velocity. We noticed that when the initial guess requires to increase its velocity in a given region, it causes the raypaths in the next iteration to "stuck" in there. Consecutive iterations tend to increase the velocity even more, which further reduces the rays' capability to trace other model cells, particularly the deeper ones. In contrast, when we reduce the velocities in the initial model, the perturbed region "repeals" the rays, which look for higher velocity zones. This change spreads the raypaths and allows for better coverage. In summary, increasing the velocities of the initial model has the risk of falling in local minima. We also found that forcing the inversion to fit first the low-velocity regions and then the high-velocity ones provided reasonably good results.

The implementation of a denoising function on the velocity model can lead to the misunderstanding that there is noise in it. A useful way to avoid this confusion is to think that the denoising engines are removing the high-frequency or "bad" features of the parameter model. The Tikhonov regularization does a similar task. The terms $\mathbf{Dm}$ and $\mathbf{m} - f(\mathbf{m})$ from the Tikhonov flat case and RED are high-pass functions acting on the parameter model whose value we minimize along with the data misfit term.

A regularization should constraint the space of solutions, so it does not have high-frequency meaningless features. Tikhonov achieves this goal by keeping the norm of the parameter model small or by ensuring a smooth behavior. RED, on the other hand, reduces the high-frequency components of the parameter model with a denoiser. Another interesting approach is one of Fomel (2007), which promotes features of interest in the retrieved parameter model by sculpting it through the application of shaping functions.

5.2 Conclusions

We proved that it is possible to use image denoising routines to regularize the inverse problem of traveltime tomography. For that, we applied RED to a linear case, such as cross-well problem, and non-linear one: first-break tomography. We were interested in testing the effectiveness of this method. Hence, we compared it with a commonly used regularization: Tikhonov Regularization. We decided to test the performance of three denoising algorithms: median filter, NLM, and TV. The first one is a simple denoiser, the second one is a state of the art one, and the last one, an algorithm designed for edge preservation.

We found that RED is superior to Tikhonov for the linear case, mainly when working with noisy data. The median filter had the least surprising results from the RED implementations, having the largest misfits in terms of **d** and **m**. Nevertheless, it did not exhibit the issues of TV when plotting an L-curve, where this last one showed negative values for the regularization term. We attribute this problem to TV not fulfilling accurately the local homogeneity condition required by RED. Further implementations of TV variations like the AWTV may solve this issue (Anagaw and Sacchi, 2020). Despite this issue, RED-TV was able to successfully retrieve a velocity model while preserving its edges, unlike Tikhonov or RED-median.RED-TV presents the opportunity of developing an alternative for edge-preserving inversion to the L_1 regularization. This last one is a proved method, yet RED may protrude it for its straightforward implementation, The linear inversion trough RED-NLM proved the assertion that sophisticated denoising engines can improve the quality of the solution of inverse problems (Romano et al., 2017). The resulting velocity models have defined edges, no as nitid as the ones obtained through TV, but with better preservation on the amplitudes and without the issues aforementioned. Furthermore, given that the regularization induces constraints, one can select a denoiser that promotes features of interest. This ability extends to the non-linear case.

We determined that it is also possible to regularize a complex non-linear problem, like first-break tomography, via RED. Even though one loses part of the power of RED due to the lesser number of denoising activations, when compared with the linear case, the retrieved subsurface images were at least as good as the ones obtained through Tikhonov. Nonetheless, RED presents a couple of advantages over the conventional regularization. Firstly, it is easier to control the trade-off parameter. This effect is more evident with the median filter, where μ is simply a measure of how much one desires to filter the model. Secondly, the denoising functions allow us to induce certain features like sharp edges or perhaps smoothness more easily.

A significant contribution of this project is the application of RED to a non-linear problem. Romano et al. (2017) set up RED for linear inverse problems, and as far as we are aware, the only work on non-linear RED is the one from Anagaw and Sacchi (2020). This one is still under development and its focus on full-waveform inversion. Therefore, the development of equations RED regularized equations for the non-linear problem, as well as its numerical resolution through FP is, as far as we are aware, a new topic.

Romano et al. (2017) proposed their method intending to take advantage of state of the art denoising methods. Nonetheless, they proved in their paper that a simple filter like the median one was able to regularize inverse problems (image denoising and super-resolution) with decent results. In this work, we stumble with the same situation. Applying RED with a 3x3 median filter was easy. The effects of the regularization were simple to understand, and the results had a similar quality to the ones from Tikhonov.

The main drawback attributed to RED is on the computation time due to the multiple activations of sophisticated denoising techniques. This disadvantage has even led to some developments to deal with this problem, like the ones of Hong et al. (2019). We observed this time consumption problem when working on the linear case, which required up to 50 denoise activations per inversion. In the non-linear case, however, we could only use the denoiser once per model update. Thus, reducing the computation time and eliminating the need for RED acceleration.

Bibliography

Aki, K., and P. G. Richards, 2002, Quantitative seismology, 2nd ed.: University Science Books.

Anagaw, A. Y., 2010, Edge-preserving seismic imaging using the Total Variation method: Presented at the GeoCanada 2010 Working with the Earth.

Anagaw, A. Y., and M. D. Sacchi, 2020, Edge-preserving fwi via Regularization by Denoising. Signal Analysis and Imaging Group, paper on progress.

Aster, R. C., 2013, Parameter estimation and inverse problems, 2nd ed.: Elsevier.

Beaton, A. E., and J. W. Tukey, 1974, The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data: Technometrics, **16**, 147–185.

Beck, A., and M. Teboulle, 2009, A Fast Iterative Shrinkage-Thresholding Algorithm for linear inverse problems: SIAM Journal on Imaging Sciences, **2**, 183–202.

Bell, M. L., R. Lara, and W. C. Gray, 1994, 1, *in* Application of turning-ray tomography to the offshore Mississippi Delta: Socierty of Exploration Geophysicist, 1509–1512.

Blakely, R. J., 1995, Potential theory in gravity and magnetic applications: Cambridge University Press.

Bleistein, N., J. W. S. Jr, and J. K. Cohen, 2002, Mathematics of multidimensional seismic imaging, migration, and inversion: Springer.

Bonar, D., and M. Sacchi, 2012, Denoising seismic data using the Non-Local Means algorithm: Geophysics, **77**, A5–A8.

Boyd, S., N. Parikh, E. Chu, B. Peleato, and J. Eckstein, 2010, Distributed optimization and statistical learning via the Alternating Direction Method of Multipliers: Foundations and Trends in Machine Learning, **3**.

Buades, A., B. Coll, and J.-M. Morel, 2011, Non-Local Means Denoising: Image Processing Online, **1**, 208–212.

Bube, K. P., and R. T. Langan, 1997, Hybrid $\ell1/\ell2$ minimization with applications to tomography: Geophysics, **62**, 1183–1195.

Burger, H. R., 2006, Introduction to applied geophysics : exploring the shallow subsurface, 2nd ed.: New York: W.W. Norton.

Candès, E. J., J. K. Romberg, and T. Tao, 2006, Stable signal recovery from incomplete and inaccurate measurements: Communications on Pure and Applied Mathematics, **59**, 1207–1223.

Cervený, V., 1987, Ray tracing algorithms in three-dimensional laterally varying layered structures, *in* Seismic Tomography with Applications in Global Seismology and Exploration Geophysics: Springer, 99–133.

Chambolle, A., 2004, An algorithm for Total Variation minimization and applications: Journal of Mathematical Imaging and Vision, **20**, 89–97.

Chan, S. H., 2016, Algorithm-induced prior for image restoration: arXiv e-prints, **abs/1602.00715**.

Chan, S. H., X. Wang, and O. A. Elgendy, 2017, Plug-and-play ADMM for image restoration: Fixed-point convergence and applications: IEEE Transactions on Computational Imaging, **3**, 84–98.

Chapman, C. H., 1976, Exact and Approximate Generalized Ray Theory in Vertically Inhomogeneous Media: Geophysical Journal International, **46**, 201–233.

Chatterjee, P., and P. Milanfar, 2010, Is denoising dead?: IEEE Transactions on Image Processing, **19**, 895–911.

Chen, Y., and T. Pock, 2017, Trainable Non-linear Reaction Diffusion: A flexible framework for fast and effective image restoration: IEEE Transactions on Pattern Analysis and Machine Intelligence, **39**, 1256–1272.

Claerbout, J. F., 1985, Imaging the earths interior: Blackwell Scientific Publications.

———, 1992, Earth sounding analysis: Blackwell Scientific Publications.

Claerbout, J. F., and F. Muir, 1973, Robust modeling with erratic data: Geophysics, **38**, 826–844.

Constable, S. C., R. L. Parker, and C. G. Constable, 1987, Occam's inversion: A practical algorithm for generating smooth models from electromagnetic sounding data: Geophysics, **52**, 289–300.

Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, 2007, Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering: IEEE Transactions on Image Processing, **16**, 2080–2095.

Daubechies, I., M. Defrise, and C. De Mol, 2003, An Iterative Thresholding Algorithm for linear inverse problems with a sparsity constraint: arXiv Mathematics e-prints, math/0307152.

Daubechies, I., R. DeVore, M. Fornasier, and C. Sinan Gunturk, 2008, Iteratively Re-weighted Least Squares minimization for sparse recovery: arXiv e-prints, arXiv:0807.0575.

Dorugade, and D. N. Kashid, 2010, Alternative method for choosing ridge parameter for regression: Applied Mathematical Sciences, **4**, 447–456.

Duijndam, A. J. W., 1988, Bayesian estimation in seismic inversion. part I: Principles: Geophysical Prospecting, **36**, 878–898.

Elad, M., and M. Aharon, 2006, Image denoising via sparse and redundant representations over learned dictionaries: IEEE Transactions on Image Processing, **15**, 3736–3745.

Fomel, S., 2007, Shaping regularization in geophysical-estimation problems: GEOPHYSICS, **72**, R29–R36.

Fowler, C. M. R., 2004, The solid earth: An introduction to global geophysics, 2 ed.: Cambridge University Press.

Gabay, D., and B. Mercier, 1976, A dual algorithm for the solution of nonlinear variational problems via finite element approximation: Computers and Mathematics with Applications, **2**, 17 – 40.

Gilbert, P., 1972, Iterative methods for the three-dimensional reconstruction of an object from projections: Journal of Theoretical Biology, **36**, 105 – 117.

Gordon, R., R. Bender, and G. T. Herman, 1970, Algebraic Reconstruction

Techniques (ART) for three-dimensional electron microscopy and X-ray photography: Journal of Theoretical Biology, **29**, 471 – 481.

Grossman, S. I., 1980, Elementary linear algebra: Belmont, Calif: Wadsworth Pub. Co.

Hansen, P. C., 1998, Rank-deficient and discrete ill-posed problems: Society for Industrial and Applied Mathematics.

Hestenes, M. R., and E. Stiefel, 1952, Methods of conjugate gradients for solving linear systems: Journal of Research of the National Bureau of Standards, **49**, 409–436.

Hong, T., Y. Romano, and M. Elad, 2019, Acceleration of RED via vector extrapolation: Journal of Visual Communication and Image Representation, **63**, 102575.

Huang, T., G. J. Tang, and G. Y. Tang, 1979, A fast two-dimensional median filtering algorithm: IEEE Transactions on Acoustics, Speech, and Signal Processing, **27**, 13–18.

Johansen, T. A., 1997, On Tikhonov regularization, bias and variance in nonlinear system identification: Automatica, **33**, 441 – 446.

Kaczmarz, S., 1937, Angenherte auflsung von systemen linearer gleichungen: Bulletin International de l'Acadmie Polonaise des Sciences et des Lettres. Classe des Sciences Mathmatiques et Naturelles, 355–357.

Kallweit, R. S., and L. C. Wood, 1982, The limits of resolution of zerophase wavelets: Geophysics, **47**, 1035–1046.

Lanteri, A., M. Maggioni, and S. Vigogna, 2019, A biased kaczmarz algorithm for clustered equations: New Statistical Developments in Data Science, Springer International Publishing, 447–456.

Lawson, C. L., and R. J. Hanson, 1995, Solving least squares problems, 1st ed.: Society for Industrial and Applied Mathematics.

Levin, A., and B. Nadler, 2011, Natural image denoising: Optimality and inherent bounds: CVPR 2011, 2833–2840.

Li-yan, W., and W. Zhi-hui, 2011, Fast gradient-based algorithm for Total Variation regularized tomography reconstruction: 2011 4th International Congress on Image and Signal Processing, 1572–1576.

Liu, Y., J. Ma, Y. Fan, and Z. Liang, 2012, Adaptive-weighted Total Variation minimization for sparse data toward low-dose X-ray computed tomography image reconstruction: Physics in Medicine & Biology, **57**, 7923–7956.

Lowrie, W., 2007, Fundamentals of geophysics, 2nd ed.: Cambridge University Press.

Marquardt, D. W., 1963, An algorithm for least-squares estimation of nonlinear parameters: SIAM Journal on Applied Mathematics, **11**, 431–441.

Menke, W., 2018, Geophysical data analysis: discrete inverse theory, 4th ed.: Elsevier.

Moser, T. J., 1989, Efficient seismic ray tracing using graph theory: SEG Technical Program Expanded Abstracts 1989, 1106–1108.

Palmer, D., 1980, The generalized reciprocal method of seismic refraction interpretation, 1st ed.: Society of Exploration Geophysicists.

———, 1981, An introduction to the generalized reciprocal method of seismic refraction interpretation: Geophysics, **46**, 1508–1518.

Petersen, K. B., and M. S. Pedersen, 2012, The matrix cookbook. (Version 20121115).

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992, Numerical recipes in c: The art of scientific computing, 2nd ed.: Cambridge University Press.

Rawlinson, N., and M. Sambridge, 2005, The fast marching method: An effective tool for tomographic imaging and tracking multiple phases in complex layered media: Exploration Geophysics, **36**.

Romano, Y., M. Elad, and P. Milanfar, 2017, The little engine that could: Regularization by denoising (RED): SIAM Journal on Imaging Sciences, **10**, 1804–1844.

Rudin, L. I., S. Osher, and E. Fatemi, 1992, Nonlinear total variation based noise removal algorithms: Physica D, **60**, 259–268.

Scales, J. A., 1987, Tomographic inversion via the conjugate gradient method: Geophysics, **52**, 179–185.

Scales, J. A., and A. Gersztenkorn, 1988, Robust methods in inverse theory: Inverse Problems, **4**, 1071–109.

———, 1990, Regularisation of nonlinear inverse problems: imaging the near-surface weathering layer: Inverse Problems, **6**, 115–131.

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

Selesnick, I. W., and P. Y. Chen, 2013, Total Variation Denoising with Overlapping Group Sparsity (GS-TVD): Presented at the IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP).

Sheriff, R. E., 2002, Encyclopedic dictionary of applied geophysics, 4th ed.: Society of Exploration Geophysicists. Geophysical References.

Sheriff, R. E., and L. P. Geldart, 1995, Exploration seismology, 2nd ed.: Cambridge University Press.

Stefani, J. P., 1995, Turning-ray tomography: Geophysics, **60**, 1917–1929.

Strang, G., 1987, Introduction to applied mathematics, 1st ed.: Wellesley, MA: Wellesley-Cambridge Press.

Strohmer, T., and R. Vershynin, 2009, A randomized kaczmarz algorithm with exponential convergence: Journal of Fourier Analysis and Applications, **15**, 262278.

Sun, M., M. D. Sacchi, and J. Zhang, 2018, An efficient tomographic inversion method based on the stochastic approximation: Geophysics, **83**, R283–R296.

Tarantola, A., 2005, Inverse problem theory and methods for model parameter estimation, 2nd ed.: Elsevier.

Telford, W. M., L. P. Geldart, and R. E. Sheriff, 1990, Applied geophysics, 2nd ed.: Cambridge University Press.

Tikhonov, A. N., and V. Y. Arsenin, 1977, Solutions of ill-posed problems: W.H. Winston.

Tukey, J. W., 1977, Exploratory data analysis, 1st ed.: Reading.

Ulrych, T. J., M. D. Sacchi, and A. Woodbury, 2001, A bayes tour of inversion: A tutorial: Geophysics, **66**, 55–69.

U.S. Energy Information Administration, 2008, Average depth of crude oil and natural gas wells. (Consulted 02-Feb-2020).

Venkatakrishnan, S. V., C. A. Bouman, and B. Wohlberg, 2013, Plug-and-play priors for model based reconstruction: 2013 IEEE Global Conference on Signal and Information Processing, 945–948.

Vidale, J., 1988, Finite-difference calculation of travel times: Bulletin of the Seismological Society of America, **78**, 2062–2076.

Vinje, V., E. Iversen, and Håvar Gjøystdal, 1993, Traveltime and amplitude estimation using wavefront construction: Geophysics, **58**, 1157–1166.

White, D. J., 1989, Two-dimensional seismic refraction tomography: Geophysical Journal International, **97**, 223–245.

Yilmaz, O., 2001, Seismic data analysis, 2nd ed.: Society of Exploration Geophysicists, **I**.

Zhang, J., and M. N. Toksöz, 1998, Nonlinear refraction traveltime tomography: Geophysics, **63**, 1726–1737.

Zhu, M., and T. Chan, 2008, An efficient primal-dual hybrid gradient algorithm for total variation image restoration: Technical report, IMAGERS UCLA.

Zhu, X., D. P. Sixta, and B. G. Angstman, 1992, Tomostatics: Turningray tomography + static corrections: The Leading Edge, **11**, 15–23.

APPENDIX A

Tikhonov in 2D considerations

In Chapter 2 of this thesis, we introduced the Tikhonov regularization term, which works with a matrix operator $\mathbf{R}$ that can take different values, such as an identity matrix or discrete differences operators. The use of these operators is precise for a 1D parameter model $\mathbf{m}$ but not for a 2D one.

A.1 Derivative operator in vector column form

We described in Chapter 2 that the parameter model has two forms, a $M_z \times M_x$ matrix $\mathbf{M}$, and a column vector $\mathbf{m}$ with length $M = M_z \times M_x$. We can approximate the first derivative of the parameter model in 1D with the discrete differences operator

$$\mathbf{L}_1 = \begin{pmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{pmatrix}, \tag{A.1}$$

Similarly, we can approximate the second derivative with

$$\mathbf{L}_2 = \begin{pmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \end{pmatrix}. \tag{A.2}$$

This operator works on $\mathbf{m}$, and must have dimensions of $M \times M$, so some columns might need to be removed.

For a 2D parameter model, we have two components per derivative, one in the direction across the columns, $\mathbf{L}_x$, and the other in the direction across the rows, $\mathbf{L}_z$. One should design these operators manually, so its product with the parameter model is a matrix showing the differences between the columns or rows of $\mathbf{M}$ respectively.

For the first derivative, we have the operators

$$\mathbf{L}_z = \begin{pmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{pmatrix}, \tag{A.3}$$

for the vertical derivative and

$$\mathbf{L}_x = \begin{pmatrix} 1 & & & \\ \vdots & & & \\ -1 & 1 & & \\ & \vdots & & \\ & -1 & & \\ & & \ddots & 1 \\ & & & \vdots \\ & & & -1 \end{pmatrix}, \tag{A.4}$$

for the horizontal one. The vertical ellipsis in $\mathbf{L}_x$ indicate a skip of $M_z - 1$ columns. Notice that both operators are banded matrices. Once more, the operators must be square to preserve the dimensions of the parameter model, so some rows or columns might need to be removed.

After designing the operators, $\mathbf{L}_x$ and $\mathbf{L}_z$ to work on $\mathbf{M}$, one can envision following a similar process for $\mathbf{m}$. When one attempts to replicate the effects of the derivative operators in the column vector form, one obtains the following interesting operators.

$$\mathbf{D}_z = \begin{pmatrix} \mathbf{L}_z & & & \\ & \mathbf{L}_z & & \\ & & \ddots & \\ & & & \mathbf{L}_z \end{pmatrix}, \tag{A.5}$$

and,

$$\mathbf{D}_x = \begin{pmatrix} \mathbf{L}_x & & & \\ & \mathbf{L}_x & & \\ & & \ddots & \\ & & & \mathbf{L}_x \end{pmatrix}. \tag{A.6}$$

In both cases the matrices $\mathbf{L}_x$ and $\mathbf{L}_z$ repeat M_x times.

The process of obtaining equations A.5 and A.6 is reproducible with the Kronecker products. This binary matrix operator uses the symbol $\otimes$ and yields block matrices. Therefore we could also say

$$\mathbf{D_z} = \mathbf{I}_{M_x} \otimes \mathbf{L}_z, \tag{A.7}$$

and

$$\mathbf{D_x} = \mathbf{I}_{M_x} \otimes \mathbf{L}_x. \tag{A.8}$$

We can also define operators to estimate the second derivatives in each direction, for the second vertical derivative we have,

$$\mathbf{L}_{zz} = \begin{pmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & & \\ & & & 1 & -2 & 1 \end{pmatrix}, \tag{A.9}$$

and for the second horizontal derivative we have

$$\mathbf{L}_{xx} = \begin{pmatrix} 1 & \cdots & -2 & \cdots & 1 & & & & \\ & & 1 & \cdots & -2 & \cdots & 1 & & \\ & & & & & \ddots & & & \\ & & & & 1 & \cdots & -2 & \cdots & 1 \end{pmatrix}^{T}. \tag{A.10}$$

This time the horizontal ellipsis in $\mathbf{L}_{xx}$ indicate a skip of $M_z - 1$ columns. We can now use the Kronecker products to define,

$$\mathbf{D_{zz}} = \mathbf{I}_{M_x} \otimes \mathbf{L}_{zz}, \tag{A.11}$$

and

$$\mathbf{D_{xx}} = \mathbf{I}_{M_x} \otimes \mathbf{L}_{xx}. \tag{A.12}$$

A.2 Regularization term for the 2D case

In order to apply the flat Tikhonov regularization to a 2D parameter model, we substituted $\mathbf{R} = \nabla$ and expanded the term. The gradient of the parameter model will have two parts, each one describing its variation in each dimension, we can name these components as $\mathbf{x}$ and $\mathbf{z}$, so

$$\nabla \mathbf{m} = \begin{pmatrix} \mathbf{x} \\ \mathbf{z} \end{pmatrix}. \tag{A.13}$$

where

$$\mathbf{x} = \mathbf{D}_x\mathbf{m} \tag{A.14}$$

and

$$\mathbf{z} = \mathbf{D}_z\mathbf{m}. \tag{A.15}$$

One should keep in mind that $\mathbf{x}$ and $\mathbf{z}$ are both column vectors representing matrices that are discrete approximations to the horizontal and vertical derivatives of a scalar space dependent parameter model function $m(x, z)$.

The L_2 norm of $\nabla\mathbf{m}$ will then be

$$||\nabla\mathbf{m}||_2^2 = \begin{pmatrix}\mathbf{x}\\ \mathbf{z}\end{pmatrix}^T \begin{pmatrix}\mathbf{x}\\ \mathbf{z}\end{pmatrix}. \tag{A.16}$$

Since $\mathbf{x}$ and $\mathbf{z}$ are both column vectors the equation afore is equivalent to the product of a row column vector and its transpose and, as established by the L_2 norm, results in a scalar

$$||\nabla\mathbf{m}||_2^2 = \sum x_i^2 + \sum z_i^2. \tag{A.17}$$

Given the numerical approximations of equations A.14 and A.15, the equation afore is equivalent to

$$||\nabla\mathbf{m}||_2^2 = ||\mathbf{D}_x\mathbf{m}||_2^2 + ||\mathbf{D}_z\mathbf{m}||_2^2. \tag{A.18}$$

We have found means for a 2D parameter model the First Order Tikhonov regularization term splits in two. This separation of the derivatives in each direction resulted convenient because it allowed us to give different weights to each one. Therefore, we opted to emulate this behaviour for the Second Order Tikhonov Regularization and substituted $\mathbf{D}_x$ and $\mathbf{D}_z$ for $\mathbf{D}_{xx}$ and $\mathbf{D}_{zz}$ in equation A.18.