Intersections and sums of sets for the regularization of inverse problems

by

Bas Peters

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL
STUDIES
(Geophysics)

The University of British Columbia
(Vancouver)

May 2019

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Intersections and sums of sets for the regularization of inverse problems

submitted by **Bas Peters** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy** in **Geophysics**.

Examining Committee:

Felix J. Herrmann, Earth, Ocean and Atmospheric Sciences
Supervisor

Michael Bostock, Earth, Ocean and Atmospheric Sciences
Supervisor

Chen Greif, Computer Science
Supervisory Committee Member

Robert Rohling, Electrical and Computer Engineering
University Examiner

Purang Abolmaesumi, Electrical and Computer Engineering
University Examiner

Laurent Demanet, Department of Mathematics, Massachusetts Institute of Technology
External Examiner

Abstract

Inverse problems in the imaging sciences encompass a variety of applications. The primary problem of interest is the identification of physical parameters from observed data that come from experiments governed by partial-differential-equations. The secondary type of imaging problems attempts to reconstruct images and video that are corrupted by, for example, noise, subsampling, blur, or saturation.

The quality of the solution of an inverse problem is sensitive to issues such as noise and missing entries in the data. The non-convex seismic full-waveform inversion problem suffers from parasitic local minima that lead to wrong solutions that may look realistic even for noiseless data. To meet some of these challenges, I propose solution strategies that constrain the model parameters at every iteration to help guide the inversion.

To arrive at this goal, I present new practical workflows, algorithms, and software, that avoid manual tuning-parameters and that allow us to incorporate multiple pieces of prior knowledge. Opposed to penalty methods, I avoid balancing the influence of multiple pieces of prior knowledge by working with intersections of constraint sets. I explore and present advantages of constraints for imaging. Because the resulting problems are often non-trivial to solve, especially on large 3D grids, I introduce faster algorithms, dedicated to computing projections onto intersections of multiple sets.

To connect prior knowledge more directly to problem formulations, I also combine ideas from additive models, such as cartoon-texture decomposition and robust principal component analysis, with intersections of multiple constraint sets for the regularization of inverse problems. The result is an extension of the concept of a Minkowski set.

Examples from non-unique physical parameter estimation problems show that constraints in combination with projection methods provide control over the model properties at every iteration. This can lead to improved results when the constraints are carefully relaxed.

Lay Summary

When we send electromagnetic or seismic signals through an unknown medium (the Earth, humans) and measure the output using multiple sensors, we know input and output but not what the medium looks like inside. The goal of inverse problems is to use the input and output to compute the materials the signals passed through. While we can numerically solve such problems, there are often many answers that satisfy the measurements—i.e., non-uniqueness. The quality of the computed solutions also decreases when there is noise in the observations, or when data is missing. I propose methods to mitigate these issues by merging measured data with prior knowledge about the material. I construct new formulations that better translate expert intuition, as well as inferences from other types of observations, into a mathematical problem. The new and faster computational methods that I derive can include more pieces of prior information than existing techniques.

Preface

All presented content in this thesis is the result of research in the Seismic Laboratory for Imaging and Modeling at the University of British Columbia (Vancouver), supervised by Professor Felix J. Herrmann. All main chapters are currently published, in review, or submitted for publication. I am the primary researcher and author of all chapters. My supervisor reviewed and suggested improvements to all documents. I formulated and developed the research questions, algorithms, software, and numerical experiments.

Chapter 2 was published as Peters, Bas, and Felix J. Herrmann. "Constraints versus penalties for edge-preserving full-waveform inversion." The Leading Edge 36, no. 1 (2017): 94-100. Chapter 3 was published as Peters, Bas, Brendan R. Smithyman, and Felix J. Herrmann. "Projection methods and applications for seismic nonlinear inverse problems with multiple constraints." Geophysics (2019). Chapter 4 and 5 will be submitted for review. The software packages corresponding to chapter 4 and 5 were written by me and are available at https://github.com/slimgroup/SetIntersectionProjection.jl.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I would like to use this opportunity to thank my supervisor Professor Felix J. Herrmann for giving me the opportunity to study and take classes in various topics, always encouraging me to be critical of my own papers, presentations, and software. I am also grateful for the support for me to work on multiple topics and the freedom to explore and develop a research path. All this enabled me to acquire a broad set of skills.

I was also lucky to have many nice and helpful colleagues: students, post-docs, and other support at the Seismic Laboratory of Imaging and Modeling at the University of British Columbia. Special thanks to Henryk for always having ready some good advice for any software, hardware, or programming related questions.

Also very grateful for much love and support at home from my wife Tuğçe, and I am looking forward to post-graduation activities now that we are both done with school. Furthermore, I am also thankful for the support from far away, in the form of my parents and sister visiting me regularly.

Chapter 1

Introduction

Every day we witness all sorts of physical phenomena around us: heat dissipation, fluid flow and sound wave propagation to name a few. We know how to simulate physics numerically, given the source activation function, initial states, boundary conditions, and physical *model parameters* such as density, acoustic velocity, and electrical conductivity. This process is called *forward modeling* in the context of an *inverse problem.* The inverse problem for physical parameter estimation is using the data acquired in the real world, called the *observed data*, and use computational methods to infer the model parameters that resulted in the observed data. A prominent example in this thesis is the estimation of acoustic velocity in the subsurface of the Earth from seismic pressure signals measured near the surface. This problem is known as full-waveform inversion (FWI [Tarantola, 1986, Pratt et al., 1998, Virieux and Operto, 2009]) in the geophysical literature. More challenging parameter estimation inverse problems also estimate the source terms [Pratt, 1999, Aravkin and van Leeuwen, 2012].

Other inverse problems that feature prominently in this thesis are image and video processing tasks such as deblurring, inpainting missing pixels, noise removal, and segmentation/classification. These problems appear different from physical parameter estimation, but there are many similarities in terms of mathematical structure, algorithms, and software.

If we define the *forward modeling operator* on a grid with N grid points, acting on the vectorized grid, as $G(\cdot) : \mathbb{R}^N \rightarrow \mathbb{R}^M$, the observed data as $d \in \mathbb{R}^M$, and the model parameters as $m \in \mathbb{R}^N$, the most basic inverse problem is straightforward data-fitting, where the misfit between simulated and observed data is minimized. Mathematically, this corresponds to the goal of finding model parameters that result in the observed data when used for forward modeling, i.e.,

$$\min_{m} f(G(m) - d). \tag{1.1}$$

Unlike forward modeling, most inverse problems do not have a solution, or have a solution that is not unique and does not depend on the data continuously. These are the Hadamard conditions that define an ill-posed problem, stated informally. The *data-misfit* function $f(\cdot) : \mathbb{R}^N \rightarrow \mathbb{R}$ quantifies the difference between the predicted data, $G(m)$, and the observed data d. A canonical example is the (non-) linear least-squares misfit $1/2\| \cdot \|_2^2$. The choice of f depends on the statistical distribution of the data-fit errors that we expect, but computational arguments such as differentiability or separability are also important. Separability means that data misfit objective can be written as a sum.

When we discuss *solving* an inverse problem, we define this as the result of the procedure that we use to obtain model parameters that minimize (1.1) with respect to the model parameters. Results can be a local or global minimum of (1.1), or any other point that prevents the optimization algorithm to further decrease f significantly. In another scenario, f decreases, but there is no significant change in m. For problems where computing $G(m)$ requires time-consuming numerical simulations, a limited number of evaluations of $G(m)$ is typically the stopping criterion and the 'solution' of the inverse problem is the m we obtained when there is no more computational time left. The model parameters that provide us with the solution, as defined in this paragraph, are also named the *model estimate*.

So far, we discussed inverse problems in the context of data fitting. However, even if an inverse problem is easy to solve numerically, it is often challenging to obtain a model estimate that is close to the true parameters. Data fitting alone is usually not enough because it leaves the solution sensitive to

- forward operators that do not contain information that helps the reconstruction. For problems like image inpainting, audio declipping and image desaturation, the observed and corrupted data satisfies $G(d) = d$ for operators that map from image to image.
- data problems such as noise, a lack of well-sampled data, aliasing, and data gaps.
- widely varying model estimates based on small changes in the initial guess. Inverse problems such as inversion of seismic data to estimate rock properties have many possible solutions that may look geologically realistic, but most of them are far from the truth.
- subsampling artifacts in the model parameters that are the result of using a (randomly changing) subset of the data in iterative reconstruction algorithms to reduce the computational demand [Krebs et al., 2009, Dai et al., 2011, Herrmann and Li, van Leeuwen et al., 2011, Li et al., 2012a, Peters et al., 2016, Xue et al., 2016].

When the data and forward modeling operator do not contain sufficient information, or are corrupted, we need more input to obtain good model parameters. Additional information may come in the form of *prior* knowledge: things we know about the model parameters before we even look at the data or start solving the inverse problem. Prior information comes from many different sources. For geophysical imaging these include expert (geologist) knowledge, physical measurements in wells [Asnaashari et al., 2013], models obtained using other types of geophysical data [Lines et al., 1988, Gallardo and Meju, 2007, Hu et al., 2009, Haber and Holtzman Gazit, 2013], and models derived using the same type of data at an earlier time (time-lapse) [Asnaashari et al., Karaoulis et al., 2011, Oghenekohwo et al., 2015]. Examples of prior knowledge include minimum and maximum values of the parameters, or if the model is simple in some sense (smooth, blocky, sparse in a transform-domain, composed of a few linearly independent rows/columns). Merging data and prior knowledge ‘fixes’ the problems listed so far, and improves the model estimates, provided *a)* sufficient prior information is available and *b)* formulating the inverse problem such that all prior knowledge is actually included in the solution. Developing methods and algorithms to include as much prior information as possible is the main topic of this thesis.

Adding prior information to the inverse problem formulation in the form of penalty functions or constraints *regularizes* the problem. The regularization described so far applies to the model parameters. Some of the issues listed above, data noise and missing data, can also be tackled using data processing. For instance, noise filtering, data completion, and bandwidth extension [Li and Demanet, 2016] techniques all act on the data. However, issues related to the non-uniqueness of inverse problem solutions also occur in case of ‘perfect’ data. In this thesis, I focus on model-based regularization exclusively. Before motivating this choice, it is important to note that we can apply both model and data-based regularizations to solve an inverse problem, which may be necessary in order to obtain the best results possible. My choice is motivated by

- the intuition that inverse problem practitioners have about the model parameters. A geologist knows what the earth looks like in the subsurface, but not what characteristics gravitational, electromagnetic, or seismic data supposed to possess.

- the invariance of a model to the data. A physical model or image is independent of the type of data, sensors, and source/receiver acquisition arrays. These factors typically change for every experiment, which makes it challenging to develop general regularization techniques that work for multiple types of data and varying experimental settings.

- the invariance of various physical properties of the same model. Consider different geophysical models of the same target. For example parameters computed from gravitational data in terms of density, and a model based on seismic data in terms of acoustic velocity. While these models describe different physical parameters with varying scales, their structure is often similar. This means that these models share some properties, such as matrix rank or cardinality (number of non-zeros in a vector) of the number of discontinuities, because these properties are scale-invariant.

- the possibly much higher dimensionality of the data compared to the model. An image or video is always 2D or 3D. The observed data to obtain such a model may be higher dimensional and contain many more data points than grid points. In exploration seismology, we can work with source x-y-z locations, receiver x-y-z locations and a time/frequency coordinate. The seismic data is therefore a 7D tensor (or 5D with fixed z-coordinates) [Trad, 2008, Kreimer et al., 2013, Silva and Herrmann, 2015, Kumar et al., 2015], which makes it more difficult to work with in a computational sense than with a 3D model.

Sometimes, model and data based regularization go hand in hand. We can apply the techniques I develop in this thesis to data as well. This is not the primary application, but if we have data organized in a matrix or 3D tensor, we may use all developed algorithms directly. If data is higher dimensional, we can flatten the tensor [Kreimer et al., 2013, Kumar et al., 2015], i.e., reshaping to lower dimensional tensors (3D array or matrix).

1.1 From prior knowledge to problem formulation

So far, we discussed what prior knowledge is and why it is important for imaging. One of the most challenging parts of solving an inverse problem, is translating prior information into a mathematical formulation. There are several ways to do this. What methods are preferable depends on the prior knowledge, applications, and available algorithms. In each of the chapters, I motivate in detail why I prefer a specific formulation over the others. I will limit the following informal discussion to the basic concepts and philosophy behind the different regularization techniques.

Perhaps the most classical concept is to penalize properties that we do not want to see in the model estimate. This is known as Tikhonov/quadratic regularization. In a more general form we add p regularization terms $R_i(m) : \mathbb{R}^N \rightarrow \mathbb{R}$ that each assign large values to models that have unwanted properties. The corresponding minimization problem is

$$\min_{m} f(m) + \sum_{i=1}^{p} \alpha_i R(m). \tag{1.2}$$

The regularization functions $R_i(m)$ may be non-convex and non-differentiable. The scalars α_i balance the influence of each regularizer with respect to each other and the data misfit. Penalty methods are the most widely used regularization technique, see, e.g. [Farquharson and Oldenburg, 1998, Becker et al., 2015, Lin and Huang, 2015, Xue and Zhu, 2015, Qiu et al., 2016] for examples in geophysics.

Another formulation casts the penalty term into the objective that is minimized given a constraint on the data misfit—i.e.,

$$\min_{m} \sum_{i=1}^{p} \alpha_i R(m) \quad \text{s.t.} \quad f(m) \leq \sigma. \tag{1.3}$$

This formulation has the advantage that if we know something about the data noise level we can determine a good choice of $\sigma > 0$. If we use only a single $R_i(m)$, there are no other scalar tuning parameters, which makes it a more practical formulation. However, when multiple pieces of prior knowledge are available, this advantage no longer holds. The multiple regularization terms require multiple α_i for balancing the influence of each objective, so there is still one trade-off parameter per model property. There are examples of this approach in the geophysical literature [Constable et al., 1987, Greenhalgh et al., 2006], but it is rare for authors to work with more than one regularization function because choosing the trade-off parameters is challenging [Ellis and Oldenburg, 1994]

To avoid choosing these trade-off parameters, this thesis revolves around the following constrained formulation:

$$\min_{m} f(m) \quad \text{s.t.} \quad m \in \bigcap_{i=1}^{p} \mathcal{V}_i, \tag{1.4}$$

where the data misfit appears as the objective and the prior information as constraints. The above formulation requires the model parameters m to be an element of p sets $\mathcal{V}_i$. The model estimate m is an element of the intersection $\bigcap_{i=1}^{p} \mathcal{V}_i$. In each chapter, I explain which properties make problem (1.4) the cornerstone of this thesis. The absence of penalty parameters is an advantage of the constrained formulation for situations where we have multiple pieces of prior knowledge. The definition of each constraint is independent of all other constraint sets and requires no balancing. Moreover, any solution of problem (1.4) will satisfy all constraints.

For geophysical problems we often want to, or need to, work with many constraints or penalties. Consider seismic imaging in sedimentary geological settings. In this situation, we quickly reach the number of four pieces of prior information. Usually there is knowledge on upper and lower limits on parameter values (bound constraints), some information about variation with depth (often in the form of promoting blockiness across the sedimentary layers), as well as two different smoothness related regularization terms for the two lateral directions (along the sedimentary layers). Yet, many inversion results are ‘obviously’ not good in the eyes of the geologist/geophysicist. This implies there is more prior knowledge available that it not yet used. Several geophysical works successfully use formulation (1.4), [Zeev et al., 2006, Bello and Raydan, 2007, Lelivre and Oldenburg, 2009, Baumstein, 2013, Smithyman et al., 2015, Esser et al., 2015a, 2016b, Esser et al., 2016]. These reference are limited to a single or two constraint sets, and some of them present algorithms for specific constraints. In this thesis, I extend workflows and algorithms to more than two constraint sets, present algorithms to compute projections onto intersections that are not tied to specific sets, introduce practical software implementations with a reduced number of tuning parameters, and I work with constraints not previously used in the geophysical literature. In chapter 5, I also introduce a new problem formulation that is more general than an intersection of sets.

1.2 Visual introduction

While all inverse problems that appear in this thesis are defined on small 2D or large 3D grids, looking at some sets and intersections in $\mathbb{R}^2$ provides some visual intuition about the techniques that underpin this work. To make visualization simple by avoiding a mixture of function level-sets and constraint sets, let me first make a small modification to problem (1.4) by changing the minimization of a data-misfit to a constraint on the data-fit. The new problem formulation reads

$$\text{find } m \in \bigcap_{i=1}^{p} \mathcal{V}_i \ \bigcap \mathcal{V}_{\text{data}}. \tag{1.5}$$

This means we find a vector $m \in \mathbb{R}^N$ that is in the intersection of p constraints on the model properties, $\bigcap_{i=1}^{p} \mathcal{V}_i$, and also in a data-constraint set $\mathcal{V}_{\text{data}}$. An example of a constraint on the data fit is $\mathcal{V}_{\text{data}} = \{m \mid \sigma_1 \leq \|G(m) - d\| \leq \sigma_2\}$ with $\sigma_1 \leq \sigma_2$. Although many authors state optimization problems of the form $\min_m \|G(m) - d\|$, they sometimes intend to use the constraint $\{m \mid \sigma_1 \leq \|G(m) - d\| \leq \sigma_2\}$. This happens when researchers stop their iterative algorithm when the data-misfit drops below the noise level: $\|G(m) - d\| < \sigma_1$. The upper bound is also effectively present because there is often a rough idea about how close we should be able to match the observed data.

We consider sets that are inspired by a geophysical inverse problem in a sedimentary geological setting. Prior knowledge, in this case, is often available about the upper and lower bounds on parameter values, some smoothness in the lateral direction, and the acoustic velocity or density are generally increasing monotonically with depth in the Earth. For each element in the model vector, prior information is given by the intersection of

1. $\{m \mid l \leq m \leq u\}$: bounds on parameter values
2. $\{m \mid -\varepsilon \leq (I_z \otimes D_x)m \leq +\varepsilon\}$: parameter values change slowly in lateral direction
3. $\{m \mid 0 \leq (D_z \otimes I_x)m \leq +\infty\},$: parameter values are increasing with depth

where $\otimes$ is the Kronecker product and D_x, D_z are finite-difference matrices, and I_x and I_z are identity matrices of size that corresponds to the grid extent in x or z-direction. In Figure 1.1, we show two-parameter representations of these sets, as well as an annulus constraint on the data fit: $\mathcal{V}_{\text{data}} = \{m \,|\, \sigma_1 \leq \|G(m) - d\| \leq \sigma_2\}$.

Figure 1.2 displays the intersection of all sets that describe prior knowledge. That figure also shows the intersection of the data-fit constraint set with the sets of prior knowledge. The projection of a few random points that are outside the intersection of all sets, are examples of feasible points that satisfy all constraints. Any feasible point satisfies all pieces of prior information, and also has the desired level of data fit; these points are examples of solutions of the inverse problem 1.5. The 'full' solution of problem 1.5 is set-valued, i.e., any point in the set. This type of projection appears extensively in the following chapters.

1.3 Motives and objectives

Motivated by physical parameter estimation problems using seismic data (seismic full-waveform inversion), I highlight fundamental challenges.

- The estimation of the model parameters of a wave-equation from recorded wavefield data at a small part of the boundary of the computational domain is a notoriously non-convex problem. PDE-constrained optimization attempts to match observed oscillatory data to simulated data that is also oscillatory. Small changes in the initial guess typically lead to large changes in the final model estimate obtained using an iterative optimization algorithm. While many of the recovered models are 'obviously' incorrect, we also fine many models that are realistic,

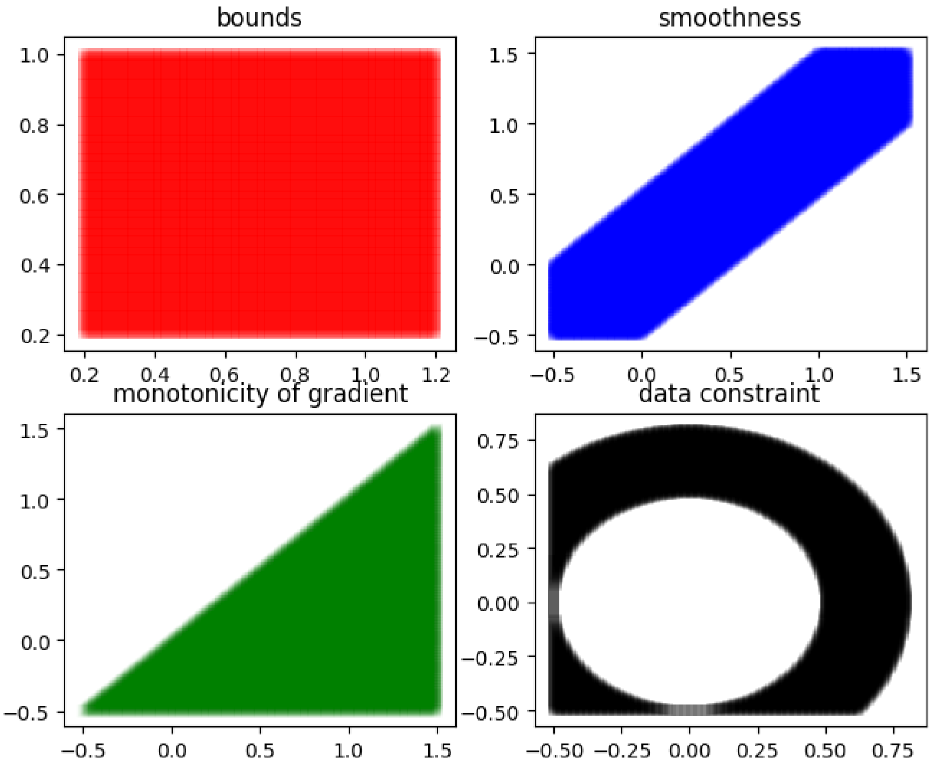


Figure 1.1: Two parameter representation of bound constraints, smoothness constraints via bounds on the gradient, monotonicity via positivity/negativity of the gradient, and an annulus constraint on the data fit.

but far from the true model parameters. These problems typically occur when there are no low-frequencies recorded (about ≤ 3 Hertz in ocean-based data acquisition), and the initial guess is far from the true model.

- Generating and recording low-frequency data is challenging for physical reasons, so assuming those low-frequency data are/will be available is not an option.

- Creating an accurate initial model from seismic data is extremely time-consuming, difficult, and requires much manual work by, e.g., first-

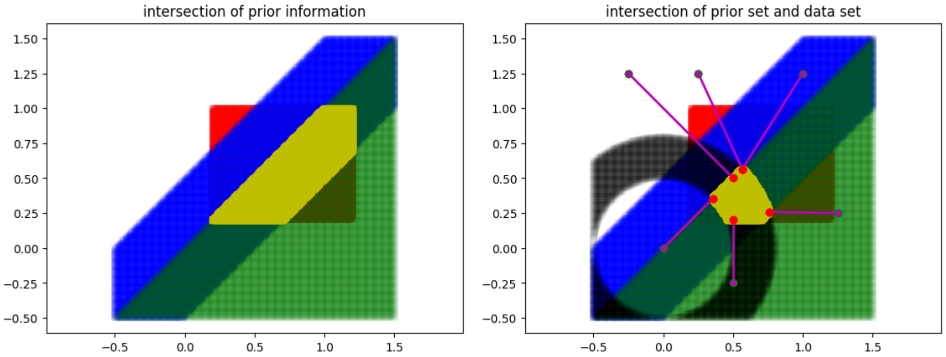


Figure 1.2: (left) The yellow highlighted patch is the intersection of the other sets that describe prior knowledge. (right) The yellow highlighted patch shows the intersection of the data constraint and all other sets. Red dots are projections of random points onto the intersection.

arrival analysis. Methods and algorithms with relatively low sensitivity to the initial model are preferable to invert seismic data.

By merging data-fitting with prior knowledge on the model parameters, we can partially mitigate the above list of challenges. Well chosen and accurate prior information has a similar effect as augmenting the missing data and can also help 'guide' iterative inversion algorithms from an inaccurate initial model to a good estimation.

The two primary objectives of this thesis are tightly linked. I want to include more prior knowledge than most research on inverse problems that only use one or two pieces of prior information, usually in the form of penalties. At the same time, I also want to make several aspects of solving inverse problems easier. More specifically, the objectives of this thesis are

- developing problem formulations, workflows, and algorithms that can include multiple pieces of prior information about model parameters and solve resulting problems on large 3D grids.

- reducing the number of parameters that need hand-tuning or algorithmic tuning at a high computational cost. These include step-length limits that need function/linear operator properties which are not readily available, stopping criteria, augmented-Lagrangian penalty, and over/under-relaxation parameters.

- applying the developed algorithms for the constrained problem formulation to non-convex seismic full-waveform inversion in various geological settings where standard formulations with quadratic penalty methods do not succeed.

1.4 Thesis outline

There are four main chapters in this thesis that follow a natural progression from relatively simple to more advanced and faster algorithms.

The intended audience for chapter 2 is a broad range of exploration geoscientists and it uses a minimal amount of mathematics to explain the concepts. I discuss some advantages of constrained formulations of inverse problems compared to penalty forms, for seismic full-waveform inversion (FWI). I show that FWI, a nonlinear and non-convex problem, with multiple penalty parameters behaves unpredictably as a function of the penalty parameter scaling. As a solution to be able to work with multiple regularizers, I present a workflow that combines three simple algorithms. This workflow is a first step that includes an arbitrary number of constraints, including ones for which we do not know the projection in closed form. The constraints in this chapter apply to geological settings that contain salt structures, i.e., large contrasts in parameter values. To verify that the regularization strategy was not just one ‘lucky’ success for a specific problem, I also apply the same constraints to a different non-convex formulation of FWI.

In chapter 3, I present an extended version of the basic framework presented in chapter 2, aimed at a general geophysical audience. Contrary to chapter 2, there are more mathematical details and faster algorithms that are only slightly more involved than the ones in chapter 2. This time, I consider a sedimentary geological setting, which means the models are mostly layered, but include challenging high-low-high acoustic velocity variation with depth, which refracts the waves such that there is little energy recorded that corresponds to waves that probed the deeper parts. To deal with this challenge, I introduce slope-constraints to geophysical problems. These constraints occur in applications like computational design and road planning in mountainous terrain. The examples show that slope-constraints can enforce smoothness or monotonicity of the parameter values. The constraints lead to better model estimates compared to penalty methods while they allow for straightforward inclusion of physical units.

While the algorithms in the framework presented in chapter 3 are faster than the ones in chapter 2, there are still opportunities to reduce computation times, which is important for 3D problems. Chapters 2 and 3 outline nested algorithms, i.e., one algorithm solves sub-problems of another algorithm, specifically the alternating direction method of multipliers (ADMM) solves sub-problems of (parallel) Dyksta's algorithm. While it may be possible to obtain limited speedups by enhancing both methods, there are two reasons why I dedicate chapter 4 to developing a single and new algorithm to compute projections onto the intersection of multiple sets. The first argument is the nuisance of having to deal with stopping criteria for both ADMM and Dykstra's algorithm. Besides additional parameters, nesting is usually inefficient. Not solving the sub-problems with sufficient accuracy will cause the framework to fail to converge, while solving sub-problems more accurate than required amounts to wasted computational time. The second reason to develop a new algorithm is the specific target problem of multiple sets. More sets mean that there is likely some similarity between the constraint. This is an opportunity that I exploit using a few simple, yet effective problem reformulation steps. The algorithmic development focusses speed and practicality. To reduce the computation times I include multilevel continuation from coarse to fine grids, hybrid coarse and fine-grained parallelism, multi-threaded matrix-vector products for banded matrices, and recently introduced automatic selection of acceleration parameters. Practical relevance of this chapter is ensured by making all algorithms available as open-source and written in Julia, stopping conditions that are more intuitive and tailored to projections, formulating the problems such that there are no manual tuning-parameters required to ensure convergence, and using various heuristics to enhance performance in case of non-convex sets. I demonstrate the capabilities on seismic full-waveform inversion and two image processing tasks where I use a simple learning method to obtain 12 pieces of prior knowledge from a few training examples.

Chapters 2, 3, and 4 all use the same problem formulation: estimated model parameters need to be an element of the intersection of multiple constraint sets. This approach captures a wide range of models and images, but there are still situations where it is difficult to describe prior knowledge using an intersection of multiple sets. A simple example is an image that is partially smooth and partially blocky, or, a smooth image with a small scale blocky pattern superimposed. In the field of image processing, such models are more conveniently described by an additive structure. Methods that add different type of image components include cartoon-texture decomposition, morphological component analysis, multi-scale analysis, and robust/sparse principal component analysis. All of these concepts use, almost exclusively, penalty methods to regularize each component. In chapter 5 I present a problem formulation, as well as algorithms, to use additive model descriptions in a constrained framework. The constrained additive formulation leads to a Minkowski set. I show that these sets are not suitable for physical parameter estimation and therefore I introduce a generalization of the Minkowski set that allows each component to be an intersection of sets, while the full model can still be an element of another intersection of sets. This concept merges and extends the problem formulation of chapters 2, 3, and 4. Using examples of seismic waveform inversion and video background-foreground segmentation, I show why a constrained version of sums of model components enables the inclusion of more pieces of prior information.

1.5 Contributions

My primary contributions to the topics introduced so far are summarized as follows:

- I provide a comprehensive investigation of how, why, and when constrained formulations for non-convex seismic parameter estimation problems are easier to use and lead to better results than penalty formulations. The presented projection-based workflow to include multiple constraints guarantees that all constraints are satisfied at each iteration, which prevents the model estimates from becoming physically unrealistic. I designed the combination of constrained problem formulation and optimization framework to avoid manual tuning parameters as much as possible, and include heuristics for defining some of the constraint sets.

- To be able to compute projections of large 3D models onto intersections of multiple convex and non-convex sets, I developed specialized algorithms and software. Different from excisting algorithms, I exploit computational similarity between the sets, specialize stopping conditions and sub-problem computations, include multilevel acceleration, while keeping the number of tuning parameters to a minimum. All presented material is available as a software package written in Julia, and this is the first package that combines all the ingredients listed above.

- I formulated a generalization of the Minkowski set. Minkowski sets combines the strenghts of constraint sets and additive model descriptions (e.g., cartoon-texture decomposition, morphological component analysis, multiscal analysis, variants of robust principal component analysis). The proposed generalization can describe more pieces of detailed prior knowledge, because each of the Minkowski set components is an intersection of sets, while the sum is also required to be an element of another intersection of sets. I also develop computational methods for computing projections onto the generalized Minkowski sets and show applications to regularizing inverse problems.

Chapter 2

Constraints versus penalties for edge-preserving full-waveform inversion

2.1 Introduction

While full-waveform inversion (FWI) is becoming increasingly part of the seismic toolchain, prior information on the subsurface model is rarely included. In that sense, FWI differs significantly from other inversion modularities such as electromagnetic and gravity inversion, which without prior information generate untenable results. Especially in situations where the inverse problem is severely ill posed, including certain regularization terms—which for instance limit the values of the inverted medium parameter to predefined ranges or that impose a certain degree of smoothness or blockiness—are known to improve inversion results significantly.

With relatively few exceptions, people have shied away from including regularization in FWI especially when this concerns edge-preserving regularization. Because of its size and sensitivity to the medium parameters, FWI differs in many respects from the above mentioned inversions, which

A version of this chapter has been published in The Leading Edge (Society of Exploration Geophysicists), 2017.

partly explains the somewhat limited success of incorporating prior information via quadratic penalty terms (Tikhonov regularization) or gradient filtering. This lack of success is further exemplified by challenges tuning these regularizations and by the fact that they do not lend themselves naturally to handle more than one type of prior information. Also, adding prior information in the form of penalties may add undesired contributions to the gradients (and Hessians).

To prevail over these challenges, we replace regularized (via additive penalty terms) inversions by inversions with 'hard' constraints. In words, instead of using regularization with penalty terms to

> find amongst all possible velocity models models that jointly fit observed data and minimize model dependent penalty terms,

we employ constrained inversions, which aim to

> find amongst all possible velocity models models that fit observed data subject to models that meet one or more constraints on the model.

While superficially these two "inversion mission statements" look rather similar, they are syntactically very different and lead to fundamentally different (mathematical) formulations, which in turn can yield significantly different inversion results and tuning-parameter sensitivities. Without going into mathematical technicalities, we define penalty approaches as methods, which add terms to a data-misfit function. Contrary to penalty formulations, constraints do not rely on local derivative information of the modified objective. Instead, constraints 'carve out' an accessible area from the data-misfit function and rely on gradient information of the data-misfit function only in combination with projections of updated models to make sure these satisfy the constraints. As a result, constrained inversions do not require differentiability of the constraints; are practically parameter free; allow for mixing and matching of multiple constraints; and most importantly, by virtue of the projections, the intermediate inversion results are guaranteed to remain within the constraint set, an extremely important feature that is more difficult if not impossible to achieve with regularizations via penalty terms.

To illustrate the difference between penalties and constraints, we consider FWI where the values and spatial variations of the inverted velocities are jointly controlled via bounds and the total-variation (TV) norm. The latter TV-norm is widely used in edge-preserved image processing [Rudin et al., 1992] and corresponds to the sum of the lengths of the gradient vectors at each spatial coordinate position.

After briefly demonstrating the effect of combining bound and TV-norm constraints on the Marmousi model, we explain in some detail the challenges of incorporating this type of prior information into FWI. We demonstrate that it is nearly impossible to properly tune the total-variation norm when included as a modified penalty term, an observation that is may very well be responsible for the unpopularity of TV-norm minimization in FWI. By imposing the TV-norm as a constraint instead, we demonstrate that these difficulties can mostly be overcome, which allows FWI to significantly improve the delineation of high-velocity high-contrast salt bodies.

2.2 Velocity blocking with total-variation norm constraints

Edge-preserving prior information derives from the premise that the Earth contains sharp edge-like unconformable strata, faults, salt or basalt inclusions. Several researchers have worked on ways to promote these edge-like features by including prior information in the form of TV-norms. If performing according to their specification, minimizing the TV-norm of the velocity model m on a regular grid with gridpoint spacing h,

$$\mathrm{TV}(m) = \frac{1}{h} \sum_{ij} \sqrt{(m_{i+1,j} - m_{i,j})^2 + (m_{i,j+1} - m_{i,j})^2}, \tag{2.1}$$

acts as applying a multidimensional “velocity blocker”. To make sure that the resulting models remain physically feasible, TV-norm minimization is combined with so-called Box constraints that make sure that each gridpoint

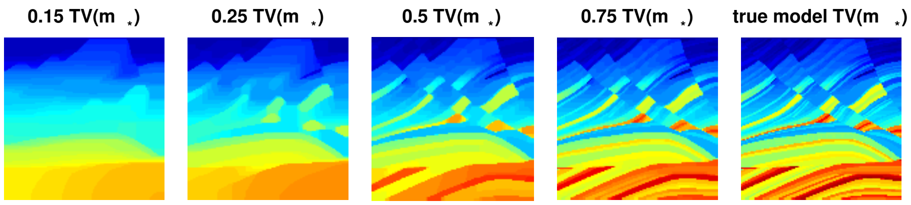


Figure 2.1: Result of projecting the true Marmousi model onto the set of bounds and limited TV-norms. Shown as a function of a fraction of the TV-norm of the true model, TV(m_*).

of the resulting velocity model remains within a user-specified interval—i.e., $m \in$ Box means that $l \leq m_{i,j} \leq u$ with l and u the lower and upper bound respectively. It can be shown, that for a given τ

$$\min_{m} \|m - m_*\|^2 \quad \text{subject to} \quad m \in \text{Box and TV}(m) \leq \tau, \qquad (2.2)$$

finds a unique blocked velocity model that is close to the original velocity model (m_*) and whose blockiness depends on the size of the TV-norm ball τ. As the size of this ball increases, the resulting blocked velocity model is less constrained, less blocky, and closer to the original model—juxtapose the TV-norm constrained velocity models in Figure 2.1 for $\tau = (0.15, 0.25, 0.5, 0.75, 1) \times \tau_{\text{true}}$ with $\tau_{\text{true}} = \text{TV}(m_*)$. The solution of Equation 2.2 is the projection of the original model onto the intersection of the box- and TV-norm constraint sets. In other words, the solution is the closest model to the input model, but which is within the bounds and has sufficiently small total-variation.

2.3 FWI with total-variation like penalties

Edge-preserved regularizations have been attempted by several researchers in crustal-scale FWI. Typically, these attempts derive from minimizing the least-squares misfit between observed (d^{obs}) and simulated data ($d^{\text{sim}}(m)$), computed from the current model iterate. Without regularization, the least-squares objective for this problem reads

$$f(m) = \|d^{\mathrm{obs}} - d^{\mathrm{sim}}(m)\|^2. \tag{2.3}$$

Now, if we follow the textbooks on geophysical inversion the most straight forward way to regularize the above nonlinear least-squares problem would be to add the following penalty term: $\alpha\|Lm\|_2^2$, where L represents the identity or a sharpening operator. The parameter α controls the trade-off between data fit and prior information, residing in the additional penalty term.

Unfortunately, this type of regularization does not fit our purpose because it smoothes the model and does not preserve edges. TV-norms (as defined in Equation 2.1), on the other hand, do preserve edges but are non-differentiable and lack curvature. Both wreak havoc because FWI relies on first- (gradient descent) and second-order (either implicitly or explicitly) derivative information.

As other researchers, including Vogel [2002b], Epanomeritakis et al. [2008], Anagaw and Sacchi [2011] and Xiang and Zhang [2016] have done before us, we can seemingly circumvent the issue of non-smoothness altogether by adding a small parameter ϵ^2 to the definition of the TV-norm in Equation 2.1. The expression for this TV-like norm now becomes

$$\mathrm{TV}_\epsilon(m) = \frac{1}{h}\sum_{ij}\sqrt{(m_{i+1,j} - m_{i,j})^2 + (m_{i,j+1} - m_{i,j})^2 + \epsilon^2} \tag{2.4}$$

and corresponds to “sand papering” the original functional form of the TV-norm at the origin so it becomes differentiable. By virtue of this mathematical property, this modified term can be added to the objective defined in Equation 2.3 — i.e., we have

$$\min_m f(m) + \alpha \mathrm{TV}_\epsilon(m). \tag{2.5}$$

For relatively simple linear inverse problems this approach has been applied with success (see e.g. Vogel [2002b]). However, as we demonstrate in the example below, this behavior unfortunately does not carry over to FWI where the inclusion of this extra tuning parameter ϵ becomes problematic.

To illustrate this problem, we revisit the subset of the Marmousi model in Figure 2.1 and invert noisy data generated from this model with a 10 Hz Ricker wavelet and with zero mean Gaussian noise added such that
$\|\text{noise}\|_2/\|\text{signal}\|_2 = 0.25$. Sources and receivers are equally spaced at 50 m and we start from an accurate smoothed model. Results of multiple warm-started inversions from 3 to 10 Hz are shown in Figure 2.2. Warm-started means we invert the data in 1 Hz batches and use the final result of a frequency batch as the initial model for the next batch. In an attempt to mimic relaxation of the constraint as in Figure 2.1, we decrease the trade-off parameter $\alpha \in (10^7, 10^6, 10^5)$ (rows of Figure 2.2) and and increase $\epsilon \in (10^{-4}, 10^{-3}, 10^{-2})$ (plotted in the columns of Figure 2.2). The latter experiments are designed to illustrate the effects of approximating the ideal TV-norm ($\text{TV}_\epsilon(m)$ for $\epsilon \to 0$).

Even though the inversion results reflect to some degree the expected behavior, namely more blocky for larger α and smaller ϵ, the reader would agree that there is no distinctive progression from "blocked" to less blocky as was clearly observed in Figure 2.1. For instance, the regularized inversion results are no longer edge preserving when the "sandpaper" parameter ϵ becomes too large. Unfortunately, this type of unpredictable behavior of regularization is common and exacerbate by more complex nonlinear inversion problems. It is difficult, if not impossible, to predict the inversion's behavior as a function of the multiple tuning parameters. While underreported, this lack of predictability of penalty-based regularization has frustrated practitioners of this type of total-variation like regularization and explains its limited use so far.

2.4 FWI with total-variation norm constraints

Following developments in modern-day optimization [Esser et al., 2016a, Esser et al., 2016], we replace the smoothed penalty term in Equation 2.5 by the intersection of box and TV-norm constraints (cf. Equation 2.1), yielding

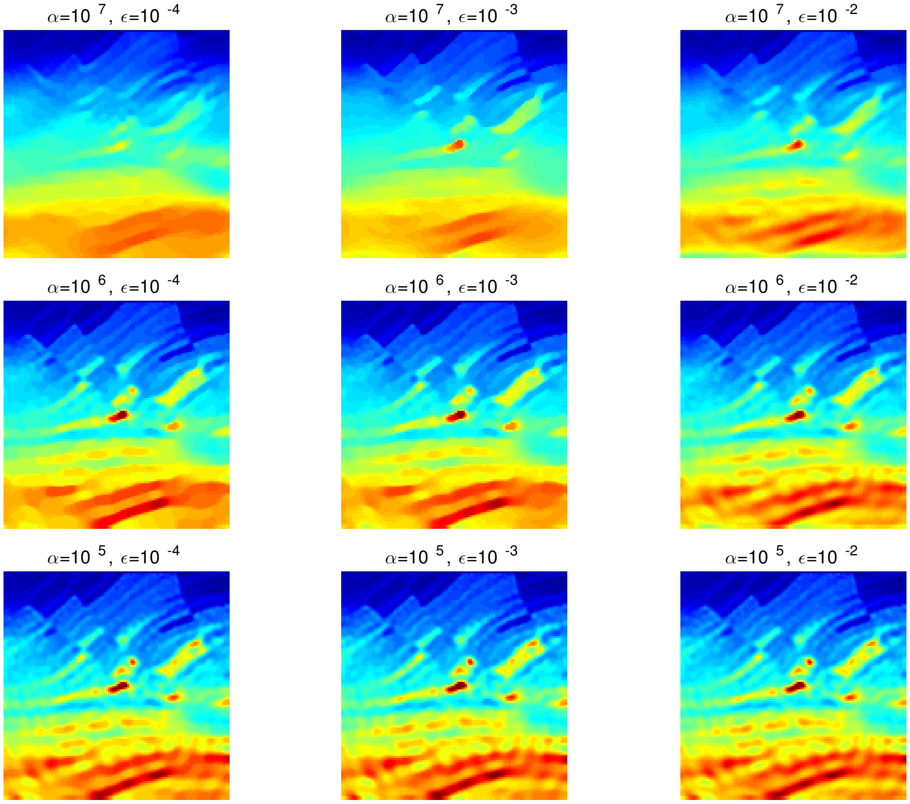


Figure 2.2: FWI results using the smoothed total-variation (TV_ϵ) as a penalty. Shows the results for various combinations of ϵ and α.

$$\min_m f(m) \quad \text{subject to} \quad m \in \text{Box and } \mathrm{TV}(m) \leq \tau, \tag{2.6}$$

which corresponds to a generalized version of Equation 2.2. Contrary to regularization with smooth penalty terms, minimization problems of the above type do not require smoothness on the constraints. Depending on the objective (data misfit function in our case), these formulations permit different solution strategies. Since the objective of FWI is highly nonlinear and computationally expensive to evaluate, we call for an algorithm design that meets the following design criteria:

- each model update depends only the current model and gradient and does not require additional expensive gradient and objective calculations;
- the updated models satisfy all constraints after each iteration;
- arbitrary number of constraints can be handled as long as their intersection is non-empty;
- manual tuning of parameters is limited to a bare minimum.

While there are several candidate algorithms that meet these criteria, we consider a projected-gradient method where at the k^{th} iteration the model is first updated by the gradient, to bring the data residual down, followed by a projection onto the constraint set $\mathcal{C}$. The projection onto the set $\mathcal{C}$ is denoted by $\mathcal{P}_{\mathcal{C}}$. The main iteration of the projected gradient algorithm is therefore given by

$$m_{k+1} = \mathcal{P}_{\mathcal{C}}(m_k - \nabla_m f(m_k)). \tag{2.7}$$

After this projection, each model is guaranteed to lie within the intersection of the Box and TV-norm constraints—i.e., $\mathcal{C} = \{m : m \in \text{Box and TV}(m) \leq \tau\}$. During the projections defined in Equation 2.2, the resulting model (m_{k+1}) is unique while it also stays as close as possible to the model after it has been updated by the gradient.

While conceptually easily stated, uniquely projecting models onto multiple constraints can be challenging especially if the individual projections do not permit closed-form solutions as is the case with the TV-norm. For our specific problem, we use Dykstra's algorithm [Boyle and Dykstra, 1986] by alternating between projections onto the Box constraint and onto the TV-norm constraint. Projecting onto an intersection of constraint sets is equivalent to running Dykstra's algorithm: $\mathcal{P}_{\mathcal{C}}(m_k - \nabla_m f(m_k)) \Leftrightarrow \text{DYKSTRA}(m_k - \nabla_m f(m_k))$.

The projection onto the box constraint is provided in closed-form, by taking the elementwise median. The projection onto the set of models with sufficiently small TV is computed via the Alternating Direction Method of

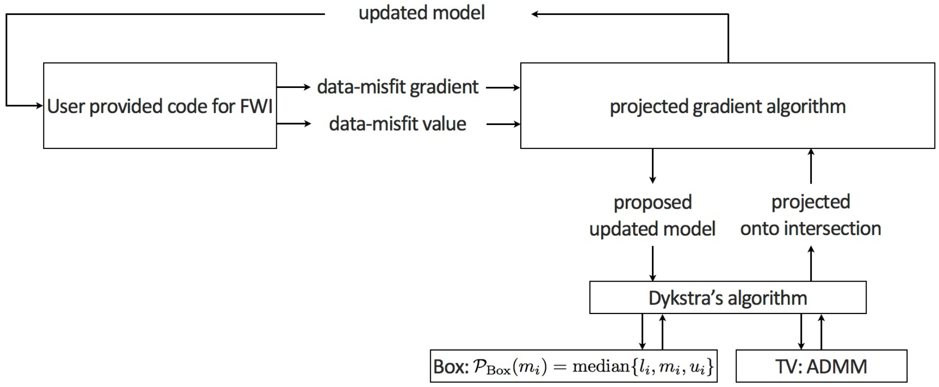

updated model
User provided code for FWI
data-misfit gradient
data-misfit value
projected gradient algorithm
proposed updated model
projected onto intersection
Dykstra's algorithm
Box: $\mathcal{P}_{\text{Box}}(m_i) = \text{median}\{l_i, m_i, u_i\}$
TV: ADMM

Figure 2.3: Constrained optimization workflow. At every FWI iteration, the user code provides data-misfit and gradient w.r.t. data-misfit only. The projected gradient algorithm uses this to propose an updated model $(m_k - \nabla_m f(m_k))$ and sends this to Dykstra's algorithm. This algorithm projects it onto the intersection of all constraints. To do this, it needs to project vectors onto each set separately once per Dykstra iteration. These individual projections are either closed-form solutions or computed by the ADMM algorithm.

Multipliers (ADMM, Boyd et al. [2011]). Dykstra's algorithm and ADMM are both free of tuning parameters in practice. The *three* steps above can be put in one nested-optimization workflow, displayed in Figure 2.3.

Dykstra's algorithm for the projection onto the intersection of constraints was first proposed by Smithyman et al. [2015] in the context of FWI and can be seen as an alternative approach to the method proposed by the late Ernie Esser and that has resulted in major breakthroughs in automatic salt flooding with TV-norm and hinge-loss constraints [Esser et al., 2016a, Esser et al., 2016].

2.5 Why constraints?

Before presenting a more elaborate example of constrained FWI on salt plays, let us first discuss why constrained optimization approaches with projections onto intersections of constraint sets are arguably simpler to use

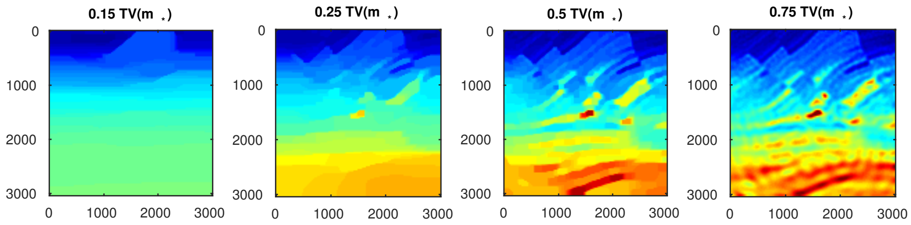


Figure 2.4: Results for constrained FWI for various total-variation budgets (τ).

than some other well known regularization techniques.

- **Constraints translate prior information and assumptions about the geology more directly than penalties.** Although the constrained formulation does not require the selection of a penalty parameter, the user still needs to specify parameters for each constraint. For Equation 2.6, this is the size of the TV ball τ. However, compared to trade-off parameter α, the τ is directly measurable from a starting or any other model that serves as a proxy.

- **Absence of user-specified weights.** Where regularization via penalty terms relies on the user to provide weights for each penalty term, unique projections onto multiple constraints can be computed with Dykstra's algorithm as long as these intersections are not empty. Moreover, the inclusion of the constraints does not alter the objective (data misfit) but rather it controls the region of $f(m)$ that our non-linear data fitting procedure is allowed to explore. This is especially important when there are many (≥ 2) constraints. For standard regularization, it would be difficult to select the weights because the different added penalties are all competing to bring down the total objective.

- **Constraints are only activated when necessary.** Before starting the inversion, it is typically unknown how 'much' regularization is required, as this depends on the noise level, type of noise, number of sources and receivers as well as the medium itself. The advantage of projection methods for constrained optimization is that they only activate the constraints when required. If a proposed model, $m_k - \nabla_m f(m_k)$, satisfies all constraints, the projection step does not do anything. The data-fitting and constraint handling are uncoupled in that sense. Penalty methods, on the other hand, modify the objective function and will for this reason always have an effect on the inversion.

- **Constraints are satisfied at each iteration.** We obtained this important property by construction of our projected-gradient algorithm. Penalty methods, on the other hand, do not necessarily satisfy the constraints at each iteration and this can make them prone to local minima.

2.6 Objective and gradients for two waveform inversion methods

To illustrate the fact that the constrained approach to waveform inversion does not depend on the specifics of a particular waveform inversion method (we only need a differentiable $f(m)$ and the corresponding gradient $\nabla_m f(m)$), we briefly describe the objective and gradient for full-waveform inversion (FWI) and Wavefield Reconstruction Inversion (WRI, van Leeuwen and Herrmann [2013]). These two methods will be used in the results section. We would like to emphasize that we do not need gradients of the constraints or anything related to the constraints. Only the projection onto the constraint set is necessary. For derivations of these gradients, see e.g., Plessix [2006] for FWI and van Leeuwen and Herrmann [2013] for WRI.

2.7 Results

To evaluate the performance of our constrained waveform-inversion methodology, we present the West part of the BP 2004 velocity model [Billette and Brandsberg-Dahl, 2005], Figure 2.5. The inversion strategy uses simultaneous sources and noisy data. We present results for two different waveform inversion methods and two different noise levels. As we can clearly see from Figures 2.6 and 2.7, FWI with bound constraints ($l = 1475\,\text{m/s}$ and

	Expression
Objective FWI:	$f(m) = \frac{1}{2}\|Pu - d^{\text{obs}}\|_2^2$
Objective WRI:	$f(m) = \frac{1}{2}\|P\bar{u} - d^{\text{obs}}\|_2^2 + \frac{\lambda^2}{2}\|A(m)\bar{u} - q\|_2^2$
Field FWI:	$u = A^{-1}q$
Field WRI:	$\bar{u} = (\lambda^2 A(m)^* A(m) + P^*P)^{-1}(\lambda^2 A(m)^* q + P^* d^{\text{obs}})$
Adjoint FWI:	$v = -A^{-*}P^*(Pu - d^{\text{obs}})$
Adjoint WRI:	none
Gradient FWI:	$\nabla_m f(m) = G(m,u)^* v$
Gradient FWI:	$\nabla_m f(m) = \lambda^2 G(m,\bar{u})^* \big(A(m)\bar{u} - q\big)$
Partial derivative FWI:	$G(m,u) = \partial A(m)u/\partial m$
Partial derivative WRI:	$G(m,u) = \partial A(m)\bar{u}/\partial m$

Table 2.1: Objectives and gradients for full-waveform inversion (FWI) and wavefield reconstruction inversion (WRI). Source term: q, discrete Helmholtz system: $A(m)$, complex-conjugate transpose $(^*)$, matrix P selects the values of the predicted wavefields u and $\bar{u}$ at the receiver locations. The scalar λ balances the data-misfit versus the wavefield residual.

$u = 5000\,\text{m/s}$) only is insufficient to steer FWI in the correct direction despite the fact we used a reasonably accurate starting model (Figure 2.5b) by smoothing the true velocity model (Figure 2.5a). WRI with bound constraint only does better, but the results are still unsatisfactory. The results obtained by including TV-norm constraints, on the other hand, lead to a significant improvement and sharpening of the salt.

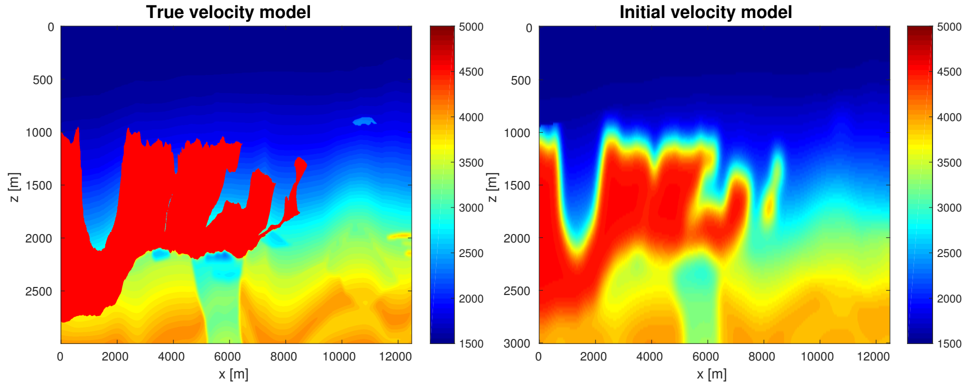
We arrived at this result via a practical workflow where we select the $\tau = \text{TV}(m_0)$, such that the initial model (m_0) satisfies the constraints. We run our inversions with the well-established multiscale frequency continuation strategy keeping the value of τ fixed. Next, we rerun the inversion with the same multiscale technique, but this time with a slightly larger τ, such that more details can enter into the solution. We select $\tau = 1.25 \times \text{TV}(m_1)$, where m_1 is the inversion result from the first inversion. This is repeated one more time, so we run the inversions three times, each run uses a different constraint. For comparison (juxtapose Figures 2.6a and 2.6b), we do the same for the inversions with the box constraints except in that case we do not impose the TV-norm constraint and keep the box constraints fixed.

As before, our inversions are carried out over multiple frequency batches with a time-harmonic solver for the Helmholtz equation and for data generated with a 15Hz Ricker wavelet. The inversions start at 3 Hz and run up to 9 Hz. The data contains noise, so that measured over all frequencies, the noise to signal ratio is $\|\text{noise}\|_2/\|\text{signal}\|_2 = 0.25$ for the first example and $\|\text{noise}\|_2/\|\text{signal}\|_2 = 0.50$ for the second example. This means that the 3 Hz data is noisier than frequencies closer to the peak frequency. Frequency domain amplitude data is shown in Figure 2.8 for the starting frequency. The starting model is kinematically correct because it is a smoothed version of the true model (cf. Figure 2.5a and 2.5b). The model size is about 3 km by 12 km, discretized on a regular grid with a gridpoint spacing of 20 meters.

The main goal of this experiment is to delineate the top and bottom of the salt body, while working with noisy data and only 8 (out of 132 sequential sources) simultaneous sources redrawn independently after each gradient calculation. The simultaneous sources activate every source at once, with a Gaussian weight. The distance between sources is 80 meters while the receivers are spaced 40 meters apart.

As we can see, limiting the total-variation norm serves two purposes. *(i)* We keep the model physically realistic by projecting out highly oscillatory and patchy components appearing in the inversion result where the TV-norm is not constrained. These artifacts are caused by noise, source crosstalk and by missing low frequencies and long offsets that lead to a non-trivial null space easily inhabited by incoherent velocity structures that hit the bounds. *(ii)* We prevent otherwise ringing artifacts just below and just above the transition into the salt. These are typical artifacts caused by the inability of regular FWI to handle large velocity contrasts. Because the artifacts increase the total-variation by a large amount, limiting the total-variation norm mitigates this well-known problem to a reasonable degree.

The noisy data, together with the use of 8 simultaneous sources effectively creates "noisy" gradients because of the source crosstalk. Therefore, our projected gradient algorithm can be interpreted as "denoising" where



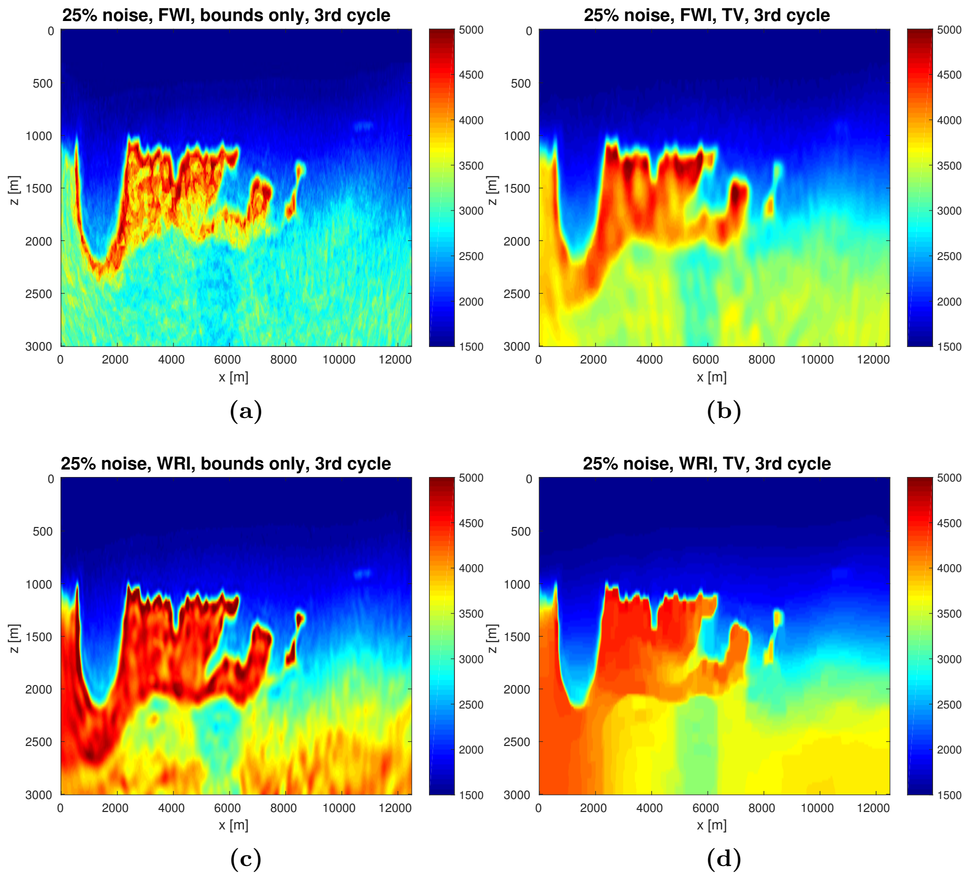
(a)

(b)

Figure 2.5: True and initial models for FWI and WRI, based on the BP 2004 model.

we map at each iteration incoherent energy onto coherent velocity structure. For this reason, the results with TV-norm constraints are drastically improved compared to the inversions carried out with bound constraints only. The inability of bounds constrained FWI to produce reasonable results for FWI with source encoding was also observed by Esser et al. [2015b] (see his Figure 19). While removing the bounds could possibly avoid some of these artifacts from building up, it would lead to physically unfeasible low and high velocities, which is something we would need to avoid at all times.

Again when the TV-norm and box constraints are applied in tandem, the results are very different. Artifacts related to velocity clipping no longer occur because they are removed by the TV-norm constraint while the inclusion of this constraint also allows us to improve the delineation of top/bottom salt and the salt flanks. The results also show that WRI, by virtue of including the wavefields as unknowns, is more resilient to noise and local minima compared to FWI and that WRI obtains a better delineation of the top and bottom of the salt structure.



(a)

(b)

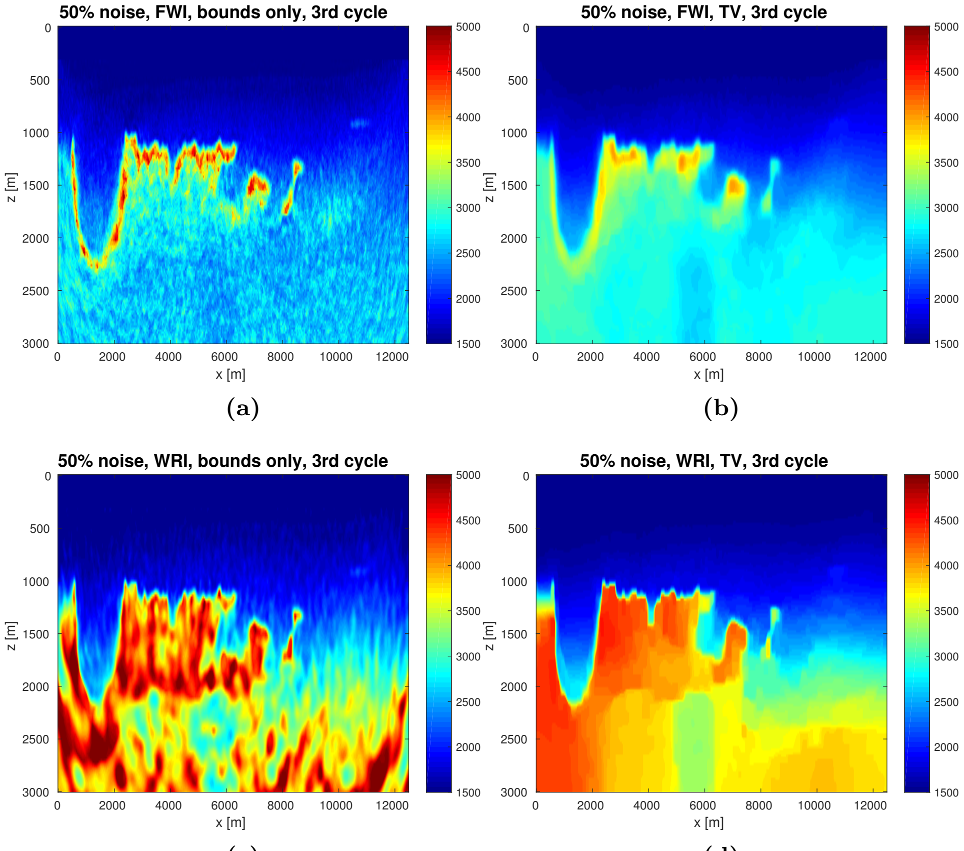
(c)

(d)

Figure 2.6: Estimated models for FWI and WRI for 25% data noise, based on the BP 2004 model. Estimated models are shown for box constraints only (a and c) and for box constraints combined with total-variation constraints (b and d).

2.8 Discussion and summary

Our purpose was to demonstrate the advantages of including (non-smooth) constraints over adding penalties in full-waveform inversion (FWI). While this text is certainly not intended to extensively discuss subtle technical details on how to incorporate non-smooth edge-preserving constraints in full-waveform inversion, we explained the somewhat limited success of including total-variation (TV) norms into FWI. By means of stylized examples, we



(a)

(b)

(c)

(d)

Figure 2.7: Estimated models for FWI and WRI for 50% data noise, based on the BP 2004 model. Estimated models are shown for box constraints only and for box constraints combined with total-variation constraints.

revealed an undesired lack of predictability of the inversion results as a function of the trade-off and smoothing parameters when we include TV-norm regularization as an added penalty term. We also made the point that many of the issues of including multiple pieces of prior information can be overcome when included as intersections of constraints rather than as the sum of several weighted penalties. In this way, we were able to incorporate the edge-preserving TV-norm and box constraints controlling the spatial

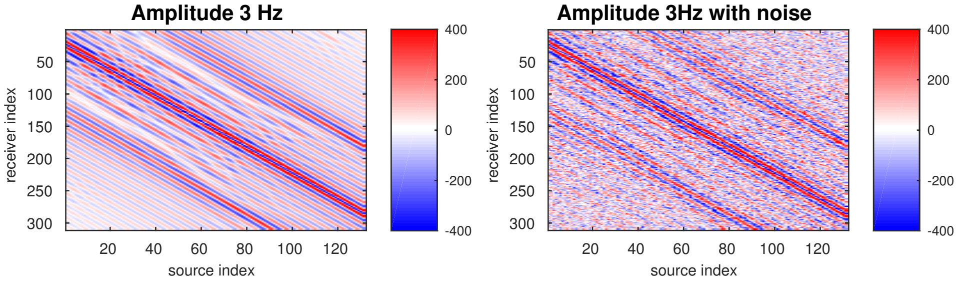


Figure 2.8: Frequency panels of the lowest frequency data for the example based on the BP 2004 model with 25% noise. All examples use noisy data, but the figure also displays data without noise for reference.

variations as well as the permissible range of inverted velocities with one parameter aside from the lower and upper bounds for the seismic velocity. As the stylized examples illustrate, this TV-norm parameter predictably controls the degree blockiness of the inverted velocity models making it suitable for FWI on complex models with sharp boundaries.

Even though the salt body example we presented is synthetic and inverted acoustically with the "inversion crime", it clearly illustrates the important role properly chosen constraints can play when combined with search extensions such as Wavefield Reconstruction Inversion (WRI). Without TV-norm constraints, artifacts stemming from source crosstalk, noise and from undesired fluctuations when moving in and out of the salt overcome FWI because the inverted velocities hit the upper and lower bounds too often. If we include the TV-norm, this effect is removed and we end up with a significantly improved inversion result with clearly delineated salt. This example also illustrates that the constrained optimization approach applies to any waveform inversion method. Results are presented for FWI and WRI, where WRI results delineate the salt structure better and exhibit more robustness to noise.

The proposed workflow and algorithms are explained in more details, and replaced with faster variants, in the following chapter.

Chapter 3

Projection methods and applications for seismic nonlinear inverse problems with multiple constraints

3.1 Introduction

We propose an optimization framework to include prior knowledge in the form of constraints into nonlinear inverse problems that are typically hampered by the presence of parasitic local minima. We favor this approach over more commonly known regularization via (quadratic) penalties because including constraints does not alter the objective, and therefore first- and second-order derivative information. Moreover, constraints do not need to be differentiable, and most importantly, they offer guarantees that the updated models meet the constraints at each iteration of the inversion. While we focus on seismic full-waveform inversion (FWI), our approach is more general and applies in principle to any linear or nonlinear geophysical in-

A version of this chapter has been published in Geophysics, Society of Exploration Geophysicists, 2018.

verse problem as long as its objective is differentiable so it can be minimized with local derivative information to calculate descent directions that reduce the objective.

In addition to the above important features, working with constraints offers several additional advantages. For instance, because models always remain within the constraint set, inversion with constraints mitigates the adverse effects of local minima which we encounter in situations where the starting model is not accurate enough or where low-frequency and long-offset data are missing or too noisy. In these situations, derivative-based methods are likely to end up in a local minimum mainly because of the oscillatory nature of the data and the non-convexity of the objective. Moreover, the costs of data acquisition and limitations on available computational resources also often force us to work with only small subsets of data. As a result, the inversions may suffer from artifacts. Finally, noise in the data and modeling errors can also give rise to artifacts. We will demonstrate that by adding constraints, which prevent these artifacts from occurring in the estimated models, our inversion results can be greatly improved and make more geophysical and geological sense.

To deal with each of the challenging situations described above, geophysicists traditionally often rely on Tikhonov regularization, which corresponds to adding differentiable quadratic penalties that are connected to Gaussian Bayesian statistics on the prior. While these penalty methods are responsible for substantial progress in working with geophysical ill-posed and ill-conditioned problems, quadratic penalties face some significant shortcomings. Chiefly amongst these is the need to select a penalty parameter, which weights the trade-off between data misfit and prior information on the model. While there exists an extensive body of literature on how to choose this parameter in the case of a single penalty term [e.g., Vogel, 2002a, Zhdanov, 2002, Sen and Roy, 2003, Farquharson and Oldenburg, 2004, Mueller and Siltanen, 2012], these approaches do not easily translate to situations where we want to add more than one type of prior information. There is also no simple prior distribution to bound pointwise values on the model without making assumptions on the underlying and often unknown statistical distribution [see Backus, 1988, Scales and Snieder, 1997, Stark, 2015]. By working with constraints, we avoid making these types of assumptions.

3.1.1 Outline

Our primary goal is to develop a comprehensive optimization framework that allows us to directly incorporate multiple pieces of prior information in the form of multiple constraints. The main task of the optimization is to ensure that the inverted models meet all constraints during each iteration. To avoid certain ambiguities, we will do this with projections so that the updated models are unique, lie in the intersection of all constraints and remain as close as possible to the model updates provided by FWI without constraints.

There is an emerging literature on working with constrained optimization, see Lelivre and Oldenburg [2009]; Zeev et al. [2006]; Bello and Raydan [2007]; Baumstein [2013]; Smithyman et al. [2015]; Esser et al. [2015a]; Esser et al. [2016b]; Esser et al. [2018], and Chapter 2 of this thesis. Because this is relatively new to the geophysical community, we first start with a discussion on related work and what the limitations are of unconstrained regularization methods. Next, we discuss how to include (multiple pieces of) prior information with constraints. This discussion includes projections onto convex sets and how to project onto intersections of convex sets. After describing these important concepts, we combine them with nonlinear optimization and describe concrete algorithmic instances based on spectral projected gradients and Dykstra's algorithm. We conclude by demonstrating our approach on an FWI problem.

3.1.2 Notation

Before we discuss the advantages of constrained optimization for FWI, let us first establish some mathematical notation. Our discretized unknown models live on regular grids with N grid points represented by the model vector $m \in \mathbb{R}^N$, which is the result of vectorizing the 2D or 3D models. In Table 3.1 we list a few other definitions we will use.

description	symbol
data-misfit	$f(m)$
gradient w.r.t. medium parameters	$\nabla_m f(m)$
set (convex or non-convex)	$\mathcal{C}$
intersection of sets	$\bigcap_{i=1}^{p} \mathcal{C}_i$
any transform-domain operator	$A \in \mathbb{C}^{M \times N}$
discrete derivative matrix in 1D	D_z or D_x
cardinality (# of nonzeros) or ℓ_0 'norm'	$\text{card}(\cdot) \Leftrightarrow \| \cdot \|_0$
ℓ_1 norm (one-norm)	$\| \cdot \|_1$

Table 3.1: Notation used in this chapter.

3.1.3 Related work

A number of authors use constraints to include prior knowledge in nonlinear geophysical inverse problems. Most of these works focus on only one or maximally two constraints. For instance; Zeev et al. [2006]; Bello and Raydan [2007] and Métivier and Brossier [2016] consider nonlinear geophysical problems with only bound constraints, which they solve with projection methods. Because projections implement these bounds exactly, these methods avoid complications that may arise if we attempt to approximate bound constraints by differentiable penalty functions. While standard differentiable optimization can minimize the resulting objective with quadratic penalties, there is no guarantee the inverted parameters remain within the specified range at every grid point during each iteration of the inversion. Moreover, there is also no consistent and easy way to add multiple constraints reflecting complementary aspects (e.g., bounds and smoothness) of the underlying geology. Bound constraints in a transformed domain are discussed by Lelivre and Oldenburg [2009].

Close in spirit to the approach we propose is recent work by Becker et al. [2015], who introduces a quasi-Newton method with projections and proximal operators [see, e.g., Parikh and Boyd, 2014] to add a single ℓ_1 norm constraint or penalty on the model in FWI. These authors include this non-differentiable norm to induce sparsity on the model by constraining the ℓ_1

norm in some transformed domain or on the gradient as in total-variation minimization. While their method uses the fact that it is relatively easy to project on the ℓ_1-ball, they have to work on the coefficients rather than on the physical model parameters themselves, and this makes it difficult to combine this transform-domain sparsity with say bound constraints that live in another transform-domain. As we will demonstrate, we overcome this problem by allowing for multiple constraints in multiple transform-domains simultaneously.

Several authors present algorithms that can incorporate multiple constraints simultaneously. The implementation of multiple constraints for inverse problems entails some subtle, but important algorithmic details. We will discuss these in this chapter. For instance, the work by Baumstein [2013] employs the well-known projection-onto-convex-sets (POCS) algorithm, which can be shown to converge to the projection of a point only in special cases, see, e.g., work by Escalante and Raydan [2011] and Bauschke and Combettes [2011]. Projecting the updated model parameters onto the intersection of multiple constraints solves this problem and offers guarantees that each model iterate (model after each iteration) remains after projection the closest in Euclidean distance to the unconstrained model and at the same time satisfies all the constraints. Different methods exist to ensure that the model estimate at every iteration remains within the non-empty intersection of multiple constraint sets. Most notably, we would like to mention the work by the late Ernie Esser [Esser et al., 2018], who developed a scaled gradient projection method for this purpose involving box constraints, total-variation, and hinge-loss constraints. Esser et al. [2018] arrived at this result by using a primal-dual hybrid gradient (PDHG) method, which derives from Lagrangians associated with total-variation and hinge-loss minimization. To allow for more flexibility in the number and type of constraints, we propose the use of Dykstra's algorithm [Dykstra, 1983, Boyle and Dykstra, 1986] instead. We refer to Smithyman et al. [2015] and Chapter 2 for examples of successful geophysical applications of multiple constraints to FWI and its distinct advantages over adding constraints as weighted penalties.

3.2 Limitations of unconstrained regularization methods

In the introduction, we stated our requirements on a regularization framework for nonlinear inverse problems. While there is a large number of successful regularization approaches such as Tikhonov regularization, change of variables, gradient filtering and modified Gauss-Newton, these methods miss one or more of our desired properties listed in the introduction. Below we will show why the above methods do not generalize to multiple constraints or do so at the cost of introducing additional manual tuning parameters.

3.2.1 Tikhonov and quadratic regularization

Perhaps the most well known and widely used regularization technique in geophysics is the addition of quadratic penalties to a data-misfit function. Let us denote the model vector with medium parameters by $m \in \mathbb{R}^N$ (for example velocity) where the number of grid points is N. The total objective with quadratic regularization $\phi(m) : \mathbb{R}^N \rightarrow \mathbb{R}$ is given by

$$\phi(m) = f(m) + \frac{\alpha_1}{2}\|R_1 m\|_2^2 + \cdots + \frac{\alpha_p}{2}\|R_p m\|_2^2. \tag{3.1}$$

In this expression, the data misfit function $f(m) : \mathbb{R}^N \rightarrow \mathbb{R}$ measures the difference between predicted and observed data. A common choice for the data-misfit is

$$f(m) = \frac{1}{2}\|d^{\text{pred}}(m) - d^{\text{obs}}\|_2^2, \tag{3.2}$$

where d^{obs} and $d^{\text{pred}}(m)$ are observed and predicted (from the current model m) data, respectively. The predicted data may depend on the model parameters in a nonlinear way.

There are p regularization terms in equation 3.1, all of which describe different pieces of prior information in the form of differentiable quadratic penalties weighted by scalar penalty parameters $\alpha_1, \alpha_2, \ldots, \alpha_p$. The operators $R_i \in \mathbb{C}^{M_i \times N}$ are selected to penalize unwanted properties in m—i.e., we select each R_i such that the penalty terms become large if the model estimate does not lie in the desired class of models. For example, we will promote smoothness of the model estimate m if we add horizontal or vertical discrete derivatives as R_1 and R_2.

Aside from promoting certain properties on the model, adding penalty terms also changes the gradient and Hessian—i.e., we have

$$\nabla_m \phi(m) = \nabla_m f(m) + \alpha_1 R_1^* R_1 m + \alpha_2 R_2^* R_2 m \tag{3.3}$$

and

$$\nabla_m^2 \phi(m) = \nabla_m^2 f(m) + \alpha_1 R_1^* R_1 + \alpha_2 R_2^* R_2. \tag{3.4}$$

Both expressions, where the symbol * refers to the complex conjugate transpose, contain contributions from the penalty terms designed to add certain features to the gradient and to improve the spectral properties of the Hessian by applying a shift to the eigenvalues of $\nabla_m^2 \phi(m)$.

While regularization of the above type has been applied successfully, it has two important disadvantages. First, it is not straightforward to encode one's confidence in a starting model other than including a reference model (m_{ref}) in the quadratic penalty term—i.e., $\alpha/2\|m_{\mathrm{ref}} - m\|_2^2$ (see, e.g., Farquharson and Oldenburg [2004] and Asnaashari et al. [2013]). Unfortunately, this type of penalty tends to spread deviations with respect to this reference model evenly so we do not have easy control over its local values (cf. box constraints) unless we provide detailed prior information on the covariance. Secondly, quadratic penalties are antagonistic to models that exhibit sparse structure—i.e., models that can be well approximated by models with a small total-variation or by transform-domain coefficient (e.g., Fourier, wavelet, or curvelet) vectors with a small ℓ_1-norm or cardinality ($\| \cdot \|_0$ "norm"). Regrettably, these sparsifying norms are non-differentiable, which often leads to problems when they are added to the objective by smoothing or reweighting the norms. In either case, this can lead to slower convergence, to unpredictable behavior in nonlinear inverse problems [Anagaw, 2014, page 110; Lin and Huang, 2015, and Chapter 2] or to a worsening of the conditioning of the Hessian [Akcelik et al., 2002]. Even without smoothing non-differential penalties, there are still penalty parameters to select [Farquharson and Oldenburg, 1998, Becker et al., 2015, Lin and Huang, 2015, Xue and Zhu, 2015, Qiu et al., 2016]. Finally, these issues with quadratic penalties are not purely theoretical. For instance, when working with a land dataset, Smithyman et al. [2015] found that the above limitations of penalty terms hold in practice and found that constraint optimization overcomes these limitations, an observation motivating this work.

3.2.2 Gradient filtering

Aside from adding penalties to the data-misfit, we can also remove undesired model artifacts by filtering the gradients of $f(m)$. When we minimize the data objective (cf. equation 3.1) with standard gradient descent, this amounts to applying a filter to the gradient when we update the model—i.e., we have

$$m^{k+1} = m^k - \gamma s(\nabla_m f(m)), \tag{3.5}$$

where γ is the step-length and $s(\cdot)$ a nonlinear or linear filter. For instance; Brenders and Pratt [2007] apply a 2D spatial low-pass filter to prevent unwanted high-wavenumber updates to the model when inverting low-frequency seismic data. The idea behind this approach is that noise-free low-frequency data should give rise to smooth model updates. While these filters can remove unwanted high-frequency components of the gradient, this method has some serious drawbacks.

First, the gradient is no longer necessarily a gradient of the objective function (equation 3.1) after applying the filter. Although the filtered gradient may under certain technical conditions remain a descent direction, optimization algorithms, such as spectral projected gradient (SPG) [Birgin et al., 1999] or quasi-Newton methods [Nocedal and Wright, 2000], expect true gradients when minimizing (constrained) objectives. Therefore gradient filtering can generally not be used in combination with these optimization algorithms, without giving up their expected behavior. Second, it is not straightforward to enforce more than one property on the model in this way. Consider, for instance, a two-filter case where $s_1(\cdot)$ is a smoother and $s_2(\cdot)$ enforces upper and lower bounds on the model. In this case, we face the unfortunate ambiguity $s_2(s_1(\nabla_m f(m))) \neq s_1(s_2(\nabla_m f(m)))$. Moreover, this gradient will have non-smooth clipping artifacts if we smooth first and then apply the bounds. Anagaw and Sacchi [2017] present a method that filters the updated model instead of a gradient, but it is also not clear how to extend this filtering technique to more than one model property.

3.2.3 Change of variables / subspaces

Another commonly employed method to regularize nonlinear inverse problems involves certain (possibly orthogonal) transformations of the original model vector. While somewhat reminiscent of gradient filtering, this approach entails a change of variables, see, e.g., Jervis et al. [1996]; Shen et al. [2005]; Shen and Symes [2008] for examples in migration velocity analysis and Kleinman and den Berg [1992]; Guitton et al. [2012]; Guitton and Daz [2012]; Li et al. [2013] for examples in the context of waveform inversion. This approach is also known as a subspace method [Kennett and Williamson, 1988, Oldenburg et al., 1993]. We can invoke this change of variables by transforming the model into $p = Tm$, where T is a (not necessarily invertible) linear operator. This changes the unconstrained optimization problem $\min_m f(m)$ into another unconstrained problem $\min_p f(p)$. To see why this might be helpful, we observe that the gradient becomes $\nabla_p f(p) = T^* \nabla_m f(m)$, which shows how T can be designed to 'filter' the gradient. The matrix T can also represent a subspace (limited number of basis vectors such as splines, wavelets). Just as with gradient filtering, a change of variables does not easily lend itself to multiple transforms aimed at incorporating complementary pieces of prior information. However, subspace information fits directly into the constrained optimization approach if we constrain our models to be elements of the subspace. The constrained approach has the advantage that we can straightforwardly combine it with other constraints in multiple transform-domains; all constraints in the proposed framework act on the variables m in the physical space since we do not minimize subspace/transform-domain coefficients p.

3.2.4 Modified Gauss-Newton

A more recent successful attempt to improve model estimation for certain nonlinear inverse problems concerns imposing curvelet domain ℓ_1-norm based sparsity constraints on the model updates [Herrmann et al., 2011, Li et al., 2012b, 2016]. This approach converges to local minimizers of $f(m)$ (and hopefully a global one) because sparsity constrained updates provably remain descent directions (Burke [1990], chapter 2; Herrmann et al. [2011]). However, there are no guarantees that the curvelet coefficients of the model itself will remain sparse unless the support (= locations of the non-zero coefficients) is more or less the same for each Gauss-Newton update [Li et al., 2016]. Zhu et al. [2017] use a similar approach, but they update the transform (also known as a dictionary when learning or updating the transform) at every FWI iteration.

In summary, while regularizing the gradients or model updates leads to encouraging results for some applications, the constrained optimization approach proposed in this work enforces constraints on the model estimate itself, without modifying the gradient. More importantly, while imposing constraints via projections may superficially look similar to the above methods, our proposed approach differs fundamentally in two main respects. Firstly, it projects uniquely on the intersection of arbitrarily many constraint sets — effectively removing the ambiguity of order in which constraints are applied. Secondly, it does not alter the gradients because it imposes the projections on the proposed model updates, i.e., we will project $m^{k+1} = m^k - \nabla_m f(m)$ onto the constraint set.

3.3 Including prior information via constraints

Before we introduce constrained formulations of nonlinear inverse problems with multiple convex and non-convex constraint sets, we first discuss some important core properties of convex sets, of projections onto convex sets, and of projections onto intersections of convex sets. These properties provide guarantees that our approach generalizes to arbitrarily many constraint sets, i.e., one constraint set is mathematically the same as many constraint sets.

The presented convex set properties also show that there is no need to worry about the order in which we use the sets to avoid ambiguity, as was the case for gradient filtering and for naive implementations of constrained optimization. The constrained formulation also stays away from penalty parameters, yet still offers guarantees all constraints are satisfied at every iteration of the inversion.

3.3.1 Constrained formulation

To circumvent problems related to incorporating multiple sources of possibly non-differentiable prior information, we propose techniques from constrained optimization [Boyd and Vandenberghe, 2004, Boyd et al., 2011, Parikh and Boyd, 2014, Beck, 2015, Bertsekas, 2015]. The key idea of this approach is to minimize the data-misfit objective while at the same time making sure that the estimated model parameters satisfy constraints. These constraints are mathematical descriptors of prior information on certain physical (e.g., maximal and minimal values for the wavespeed) and geological properties (e.g., velocity models with unconformities that lead to discontinuities in the wavespeed) on the model. We build our formulation on earlier work on constrained optimization with up to three constraint sets as presented by Lelivre and Oldenburg [2009]; Smithyman et al. [2015]; Esser et al. [2015a]; Esser et al. [2016b]; Esser et al. [2018], and Chapter 2.

Given an arbitrary but finite number of constraint sets (p), we formulate our constrained optimization problem as follows:

$$\min_{m} f(m) \text{ subject to } m \in \bigcap_{i=1}^{p} \mathcal{C}_i. \tag{3.6}$$

As before, $f(m) : \mathbb{R}^N \rightarrow \mathbb{R}$ is the data-misfit objective, which we minimize over the discretized medium parameters represented by the vector $m \in \mathbb{R}^N$. Prior knowledge on this model vector resides in the indexed constraint sets $\mathcal{C}_i$, for $i = 1 \cdots p$, each of which captures a known aspect of the Earth's subsurface. These constraints may include bounds on permissible parameter values, desired smoothness or complexity, or limits on the number of layers in sedimentary environments and many others.

In cases where more than one piece of prior information is available, we want the model vector to satisfy these constraints simultaneously, such that we keep control over the model properties as is required for strategies that relax constraints gradually, see Esser et al. [2016b] and Chapter 2. Because it is difficult to think of a nontrivial example where the intersection of these sets is empty, it is safe to assume that there is at least one model that satisfies all constraints simultaneously. For instance, a homogeneous medium will satisfy many constraints, because its total-variation is zero, it has a rank of 1 and has parameter values between minimum and maximum values. We denote the mathematical requirement that the estimated model vector satisfies p constraints simultaneously by $m \in \bigcap_{i=1}^{p} \mathcal{C}_i$. The symbol $\bigcap_{i=1}^{p}$ indicates the intersection of p items. Before we discuss how to solve constrained nonlinear geophysical inverse problems, let us first discuss projections and examples of projections onto convex and non-convex sets.

3.3.2 Convex sets

A projection of m onto a set $\mathcal{C}$ corresponds to solving

$$\mathcal{P}_{\mathcal{C}}(m) = \operatorname*{arg\,min}_{x} \frac{1}{2} \|x - m\|_2^2 \text{ subject to } x \in \mathcal{C}. \tag{3.7}$$

Amongst all possible model vectors x, the above optimization problem finds the vector x that is closest in Euclidean distance to the input vector m while it lies in the constraint set. For a given model vector x, the solution of this optimization problem depends on the constraint set $\mathcal{C}$ and its properties. For instance, the above projection is unique for a convex $\mathcal{C}$.

To better understand how to incorporate prior information in the form of one or more constraint sets, let us first list some important properties of constraint sets and their intersection. These properties allow us to use relatively simple algorithms to solve Problem 3.6 by using projections of the above type. First of all, most optimization algorithms require the constraint sets to be convex. Intuitively, a set is convex if any point on the line segment connecting any couple of points in a set is also in the set—i.e., for all $x \in \mathcal{C}$

and $y \in \mathcal{C}$. In that case, the following relation holds:

$$cx + (1 - c)y \in \mathcal{C} \quad \text{for } 0 \leq c \leq 1. \tag{3.8}$$

There are a number of advantages when working with convex sets, namely

i. The intersection of convex sets is also a convex set. This property implies that the properties of a convex set also hold for the intersection of arbitrarily many convex sets. Practically, if an optimization algorithm is defined for a single convex set, the algorithm also works in case of arbitrarily many convex sets, as the intersection is still a single convex set.

ii. The Euclidean projection onto a convex set (equation 3.7) is unique (Boyd and Vandenberghe [2004], section 8.1). When combined with property *(i)*, this implies that the projection onto the intersection of multiple convex sets is also unique. In this context, a unique projection means that given any point outside a convex set, there exists one point in the set which is closest (in a Euclidean sense) to the given point than any other point in the set.

iii. Projections onto convex sets are non-expansive (Bauschke and Combettes [2011], section 4.1-4.2, or Dattorro [2010], E.9.3). If we define the projection operator as $\mathcal{P}_{\mathcal{C}}(x)$ and take any couple of points x and y, the non-expansive property is stated as: $\|\mathcal{P}_{\mathcal{C}}(x) - \mathcal{P}_{\mathcal{C}}(y)\| \leq \|x - y\|$. This property guarantees that projections of estimated models onto a convex set are 'stable'. In this context, stability implies that any pair of models moves closer or remain equally distant to each other after projection. This prevents increased separation after projection of pairs of models.

While these properties make convex sets favorites amongst practitioners of (convex) optimization, restricting ourselves to convexity is sometimes too limiting for our application. In the following sections, we may use non-convex sets in the same way as a convex set, but in that case, the above properties generally do not hold. Performance of the algorithms then needs empirical verification.

Actual projections onto a single set themselves are either available in closed-form (e.g., for bounds and certain norms) or are computed iteratively (with the alternating direction method of multipliers, ADMM, see e.g., Boyd et al. [2011] and Appendix A) when closed form-expressions for the projections are not available.

3.4 Computing projections onto intersections of convex sets

Our problem formulation, equation 3.6, concerns multiple constraints, so we need to be able to work with multiple constraint sets simultaneously to make sure the model iterates satisfy all prior knowledge. To avoid intermediate model iterates to become physically and geologically unfeasible, we want our model iterates to satisfy a predetermined set of constraints at every iteration of the inversion process. Because of property *(i)* (listed above), we can treat the projection onto the intersection of multiple constraints as the projection onto a single set. This implies that we can use relatively standard (convex) optimization algorithm to solve Problem 3.6 as long as the intersection of the different convex sets is not empty. We define the projection on the intersection of multiple sets as

$$\mathcal{P}_{\mathcal{C}}(m) = \arg\min_{x} \|x - m\|_2^2 \quad \text{s.t.} \quad x \in \bigcap_{i=1}^{p} \mathcal{C}_i. \tag{3.9}$$

The projection of m onto the intersection of the sets, $\bigcap_{i=1}^{p} \mathcal{C}_i$, means that we find the unique vector x, in the intersection of all sets, that is closest to m in the Euclidean sense. To find this vector, we compute the projection onto this intersection via Dykstra's alternating projection algorithm [Dykstra, 1983, Boyle and Dykstra, 1986, Bauschke and Koch, 2015]. We made this choice because this algorithm is relatively simple to implement (we only need projections on each set individually) and contains no manual tuning parameters. By virtue of property *(ii)*, projecting onto each set separately and cyclically, Dykstra's algorithm finds the unique projection on the intersection as long as all sets are convex [Boyle and Dykstra, 1986, Theorem 2].

To illustrate how Dykstra's algorithm works, let us consider the following toy example where we project the point (2.5, 3.0) onto the intersection of two constraint sets, namely a halfspace ($y \leq 2$, this corresponds to bound constraints in two dimensions) and a disk ($x^2 + y^2 \leq 3$, this corresponds to a $\| \cdot \|_2$-norm ball), see Figure 3.1. If we are just interested in finding a feasible point in the set that is not necessarily the closest, we can use the projection onto convex sets (POCS) algorithm (also known as von Neumann's alternating projection algorithm) whose steps are depicted by the solid black line in Figure 3.1. The POCS algorithm iterates $\mathcal{P}_{\mathcal{C}_2}(\ldots(\mathcal{P}_{\mathcal{C}_1}(\mathcal{P}_{\mathcal{C}_2}(\mathcal{P}_{\mathcal{C}_1}(m)))))$, so depending on whether we first project onto the rectangle or disk, POCS finds two different feasible points. Like POCS, Dykstra's algorithm projects onto each set in an alternating fashion, but unlike POCS, the solution path that is denoted by the red dashed line provably ends up at a single unique feasible point that is closest to the starting point. The solution found by Dykstra's algorithm is independent of the order in which the constraints are imposed. POCS does not project onto the intersection of the two convex sets; it just solves the convex feasibility problem

$$\textbf{find } x \in \bigcap_{i=1}^{p} \mathcal{C}_i \tag{3.10}$$

instead. POCS finds a model that satisfies all constraints but which is non-unique (solution is either (1.92, 2.0) or (2.34, 1.87) situated at Euclidean distances 1.16 and 1.14) and not the projection at $(2.0, \sqrt{2^2 + 3^2} \approx 2.24)$ at a minimum distance of 1.03. This lack of uniqueness and vicinity to the true solution of the projection problem leads to solutions that satisfy the constraints, but that may be too far away from the initial point and this may adversely affect the inversion. See also [Escalante and Raydan, 2011, Example 5.1; Dattorro, 2010, Figure 177 & Figure 182, and Bauschke and Combettes [2011], Figure 29.1] for further details on this important point.

The geophysical implication of this difference between Dykstra's algorithm and POCS is that the latter may end up solving a problem with unnecessarily tight constraints, moving the model too far away from the descent direction informed by the data misfit objective. We observe this phenomenon of being too constrained in Figure 3.1 where the two solutions from POCS are not on the boundary of both sets, but instead relatively 'deep' inside one of them. Aside from potential "over constraining", the results from POCS may also differ depending which of the individual constraints is activated first leading to undesirable side effects. The issue of "over constraining" does not just occur in geometrical two-dimensional examples and it is not specific to the constraints from the previous example. Figure 3.2 shows what happens if we project a velocity model (with Dykstra's algorithm) or find two feasible models with POCS, just as we show in Figure 3.1. The constraint is the intersection of bounds ($\{m \,|\, l_i \leq m_i \leq u_i\}$) and total-variation ($\{m \,|\, \|Am\|_1 \leq \sigma\}$ with scalar $\sigma > 0$ and $A = (D_x^T \; D_z^T)^T$). While one of the POCS results is similar to the projection, the other POCS result has much smaller total-variation than the constraint enforces, i.e., the result of POCS is not the projection but a feasible point in the interior of the intersection. To avoid these issues, Dykstra's algorithm is our method of choice to include two or more constraints into nonlinear inverse problems. Algorithm 1 summarizes the main steps of Dykstra's approach, which aside from stopping conditions, is parameter free. In Figure 3.3 we show what happens if we replace the projection (with Dykstra's algorithm) in projected gradient descent with POCS. Projected gradient descent solves an FWI problem with bounds and total-variation constraints while using a small number of sources and receivers and an incorrectly estimated source function. The results of Dykstra's algorithm and POCS are different, while the results using POCS depend on the ordering of the sets. Dykstra's algorithm always finds the Euclidean projection onto the intersection of convex sets, which is a unique point. Therefore, it does not matter in what order we project onto each set as part of Dykstra's algorithm.

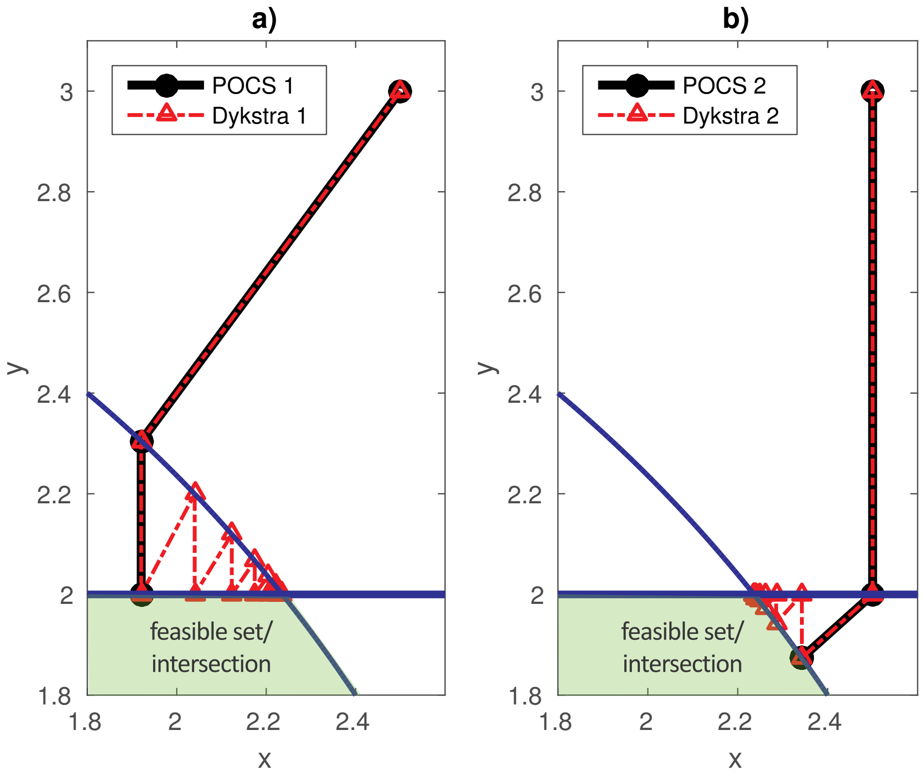


Figure 3.1: The trajectory of Dykstra's algorithm for a toy example with two constraints: a maximum 2-norm constraint (disk) and bound constraints. The feasible set is the intersection of a halfspace and a disk. The circle and horizontal lines are the boundaries of the sets. The difference between the two figures is the ordering of the two sets. The algorithms in (a) start with the projection onto the disk, in (b) they start with the projection onto the halfspace. The projection onto convex sets (POCS) algorithm converges to different points, depending onto which set we project first. In both cases, the points found by POCS are not the projection onto the intersection. Dykstra's algorithm converges to the projection of the initial point onto the intersection in both cases, as expected.

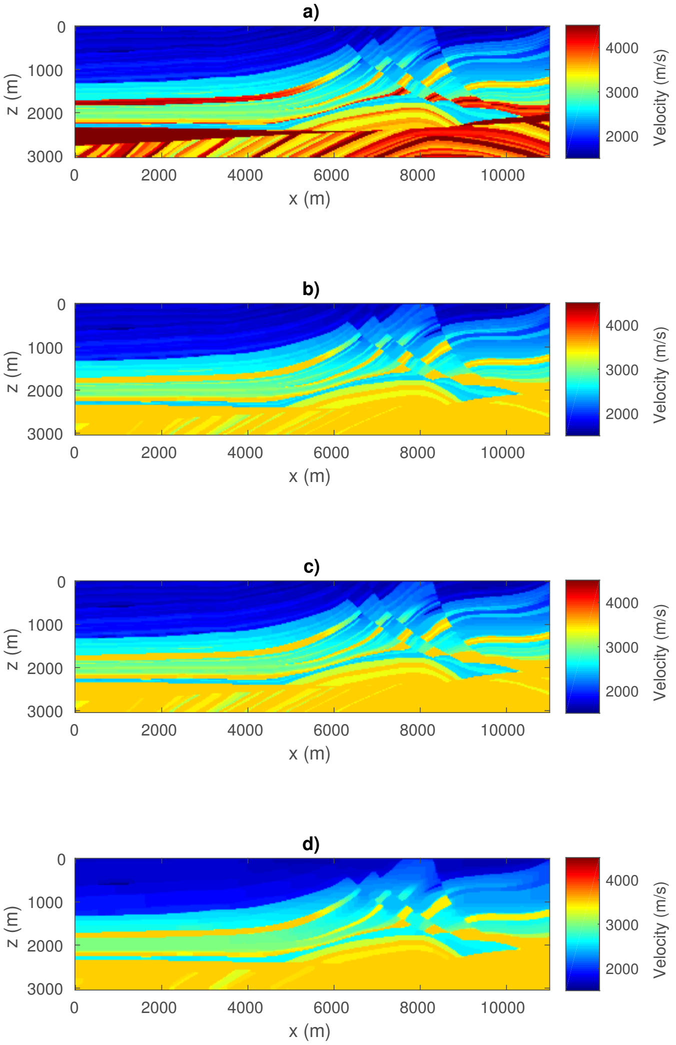


Figure 3.2: The Marmousi model (a), the projection onto an intersection of bound constraints and total-variation constraints found with Dykstra's algorithm (b) and two feasible models found by the POCS algorithm (c) and (d). We observe that one of the POCS results (c) is very similar to the projection (b), but the other result (d) is very different. The different model (d) has a total-variation much smaller than requested. This situation is analogous to Figure 3.1.

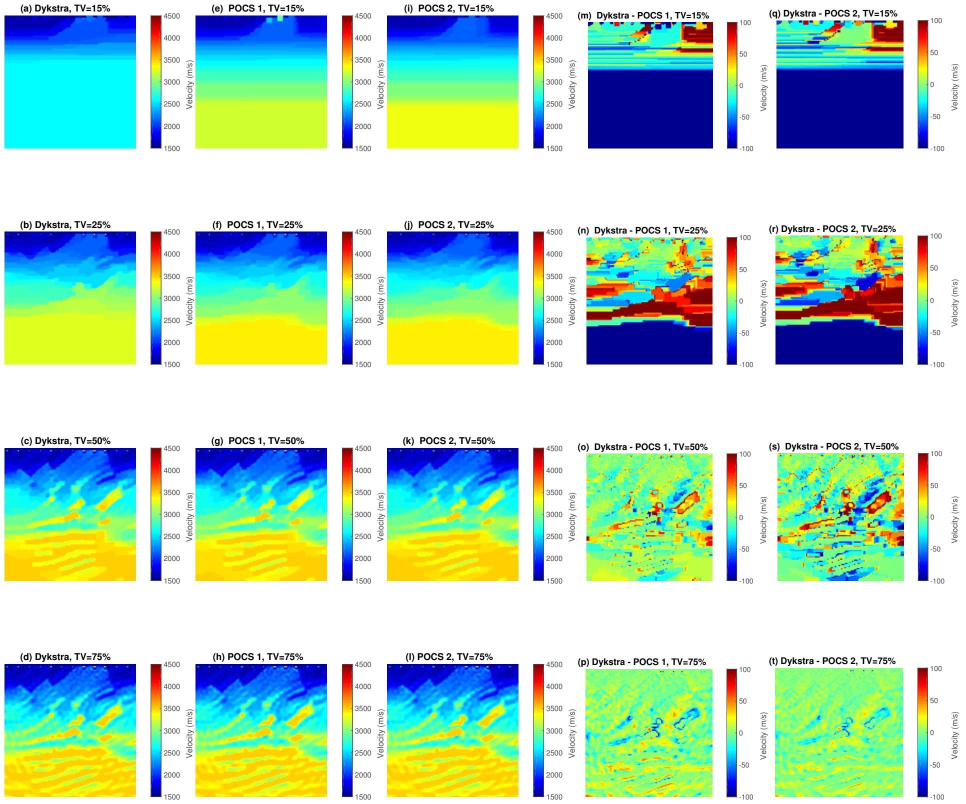


Figure 3.3: FWI with an incorrect source function with projections (with Dykstra's algorithm) and FWI with two feasible points (with POCS) for various TV-balls (as a percentage of the TV of the true model) and bound constraints. Also shows differences (rightmost two columns) between results. The results show that using POCS inside a projected gradient algorithm instead of the projection leads to different results that also depend on the order in which we provide the sets to POCS. This example illustrates the differences between the methods and it is not the intention to obtain excellent FWI results.

Algorithm 1 Dykstra's algorithm, following the notation of Birgin and Raydan [2005], to compute the projection of m onto the intersection of p convex sets: $\mathcal{P}_{\mathcal{C}}(m) = \arg\min_x \|x - m\|_2^2$ s.t. $x \in \bigcap_{i=1}^p \mathcal{C}_i$. y_i are auxiliary vectors.

```
Algorithm DYKSTRA($m, \mathcal{P}_{\mathcal{C}_1}, \mathcal{P}_{\mathcal{C}_2}, \dots, \mathcal{P}_{\mathcal{C}_p}$)
0a. $x_p^0 = m$, $k = 1$ //initialize
0b. $y_i^0 = 0$ for $i = 1, 2, \dots, p$ //initialize
 WHILE stopping conditions not satisfied DO
1.     $x_0^k = x_p^{k-1}$
     FOR $i = 1, 2, \dots, p$
2.         $x_i^k = \mathcal{P}_{\mathcal{C}_i}(x_{i-1}^k - y_i^{k-1})$
     END
     FOR $i = 1, 2, \dots, p$
3.         $y_i^k = x_i^k - (x_{i-1}^k - y_i^{k-1})$
     END
4.     $k = k + 1$
 END
output: $x_p^k$
```

3.5 Nonlinear optimization with projections

So far, we discussed a method to project models onto the intersection of multiple constraint sets. Now we propose and discuss a method to combine projections onto an intersection with nonlinear data-fitting. Aside from our design criteria (multiple constraints instead of competing penalties; guarantees that model iterations remain in constraint set), we need to include a clean separation of misfit/gradient calculations and projections so that we avoid additional computationally costly PDE solves at all times. This separation also allows us to use different codes bases for each task (objective/gradient calculations versus projections). We first describe the basic projected gradient descent method, which serves as an introduction to our method of choice: the spectral projected gradient method.

3.5.1 Projected gradient descent

The simplest first-order algorithm that minimizes a differentiable objective function subject to constraints is the projected gradient method (e.g., Beck

[2014], section 9.4). This algorithm is a straightforward extension of the well-known gradient-descent method [e.g., Bertsekas, 2015, section 2.1] involving the following updates on the model:

$$m^{k+1} = \mathcal{P}_{\mathcal{C}}\big(m^k - \gamma \nabla_m f(m^k)\big). \tag{3.11}$$

A line search determines the scalar step length $\gamma > 0$. This algorithm first takes a gradient-descent step, involving a gradient calculation, followed by the projection of the updated model back onto the intersection of the constraint sets. By construction, the computationally expensive gradient computations (and data-misfit for the line search) are separate from the often much cheaper projections onto constraints. The projection step itself guarantees that the model estimate m^k satisfies all constraints at every k^{th} iteration.

Figure 3.4 illustrates the difference between gradient descent to minimize a two variable non-convex objective $\min_m f(m)$, and projected gradient descent to minimize $\min_m f(m)$ s.t. $m \in \mathcal{C}$. If we compare the solution paths for gradient and projected gradient descent, we see that the latter explores the boundary as well as the interior of the constraint set $\mathcal{C} = \{m \,|\, \|m\|_2 \leq \sigma\}$ to find a minimizer. This toy example highlights how constraints pose upper limits (the set boundary) on certain model properties but do not force solutions to stay on the constraint set boundary. Because one of the local minima lies outside the constraint set, this example also shows that adding constraints may guide the solution to a different (correct) local minimizer. This is exactly what we want to accomplish with constraints for FWI: prevent the model estimate m^k to converge to local minimizers that represent unrealistic models.

3.5.2 Spectral projected gradient

Standard projected gradient has two important drawbacks. First, we need to project onto the constraint set after each line search step. To be more specific, we need to calculate the step-length parameter $\gamma \in (0, 1]$ if the objective of the projected model iterate is larger than the current model

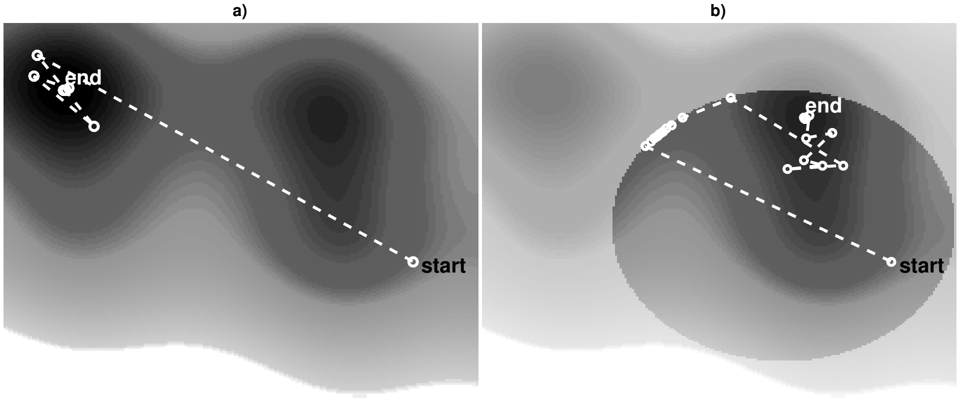


Figure 3.4: Example of the iteration trajectory when (a) using gradient descent to minimize a non-convex function and (b) projected gradient descent to minimize a non-convex function subject to a constraint. The constraint requires the model estimate to inside the elliptical area in (b). The semi-transparent area outside the ellipse is not accessible by projected gradient descent. There are two important observations: 1) The constrained minimization converges to a different (local) minimizer. 2) The intermediate projected gradient parameter estimates can be in the interior of the set or on the boundary. Black represents low values of the function.

iterate—i.e., $f\left(\mathcal{P}_{\mathcal{C}}(m^k - \gamma\nabla_m f(m^k))\right) > f(m^k)$. In that case, we need to reduce γ and test again whether the data-misfit is reduced. For every reduction of γ, we need to recompute the projection and evaluate the objective, which is too expensive. Second, first-order methods do not use curvature information, which involves the Hessian of $f(m)$ or access to previous gradient and model iterates. Projected gradient algorithms are therefore often slower than Newton, Gauss-Newton, or quasi-Newton algorithms for FWI without constraints.

To avoid these two drawbacks and possible complications arising from the interplay of imposing constraints and correcting for Hessians, we use the spectral projected gradient method (SPG; Birgin et al. [1999]; Birgin et al. [2003]); an extension of the standard projected gradient algorithm (equation 3.11), which corresponds to a simple scalar scaling (related to the eigenvalues of the Hessian, see Birgin et al. [1999] and Dai and Liao [2002]). At model iterate k, the SPG iterations involve the step

$$m^{k+1} = m^k + \gamma p^k, \tag{3.12}$$

with update direction

$$p^k = \mathcal{P}_{\mathcal{C}}\big(m^k - \alpha \nabla_m f(m^k)\big) - m^k. \tag{3.13}$$

These two equations define the core of SPG, which differs from standard projected gradient descent in three different ways:

i. The spectral stepsize α [Barzilai and Borwein, 1988, Raydan, 1993, Dai and Liao, 2002] is calculated from the secant equation [Nocedal and Wright, 2000, section 6.1] to approximate the Hessian, leading to an accelerated convergence. An interpretation of the secant equation is to mimic the action of the Hessian by the scalar α and use finite-difference approximations for the second derivative of $f(m)$. This approach is closely related to the idea behind quasi-Newton methods. We compute α as the solution of

$$D^k = \underset{D=\alpha I}{\arg\min} \|Ds^k - y^k\|_2, \tag{3.14}$$

where $y^k = \nabla_m f(m^{k+1}) - \nabla_m f(m^k)$ and $s^k = m^{k+1} - m^k$, and I the identity matrix. This results in scaling by $\alpha = s_k^* s_k / s_k^* y_k$ derived from gradient and model iterates from the current and previous SPG iterations. Clearly, this is computationally cheap because α is not computed by a separate line search. We may also need a safeguard against excessively large values of α, defined as $\alpha = \text{minimum}(\alpha, \alpha_{\max})$. Because we work with geophysical inverse problems, we can require a value of α, such that α times the gradient has a 'reasonable' physical scaling, i.e., we do not want $\alpha_{\max}$ times the gradient to have a norm larger than the current model parameters. Very large values of α could lead to unphysical models (before projection) and to an unnecessarily large number of line-search steps to determine γ. We thus require $\alpha_{\max} \|\nabla_m f(m^k)\|_2 \leq \|m\|_2$.

ii. Spectral projected gradient employs non-monotone [Grippo and Sciandrone, 2002] inexact line searches to calculate the γ in equation 3.12. In Algorithm 2, step 4c enforces a non-monotone Armijo line-search condition. As for all FWI problems, $f(m)$ is not convex so we cannot use an exact line-search. Non-monotone means that the objective function value is allowed to increase temporarily, which often results in faster convergence and fewer line search steps, see, e.g., Birgin et al. [1999] for numerical experiments. Our intuition behind this is as follows: gradient descent iterations often exhibit a 'zig-zag' pattern when the objective function behaves like a 'long valley' in a certain direction. When the line searches are non-monotone, the objective does not always have to go down so we can take relatively larger steps along the valley in the direction of the minimizer that are slightly 'uphill', increasing the objective temporarily.

iii. Each SPG iteration requires only one projection onto the intersection of constraint sets to compute the update direction (equation 3.13) and does not need additional projections for line search steps. This is a significant computational advantage over standard projected gradient descent, which computes one projection per line search step, see equation 3.11. From equations 3.12 and 3.13, we observe that p^k lies on the line between the previous model estimate (m^k) and the proposed update, projected back onto the feasible set—i.e., $\mathcal{P}_{\mathcal{C}}(m^k - \alpha \nabla_m f(m^k))$. Therefore, m^{k+1} is on the line segment between these two points in a convex set and the new model will satisfy all constraints simultaneously at every iteration (see equation 3.8). For this reason, any line search step that reduces γ will also be an element of the convex set. Works by Zeev et al. [2006] and Bello and Raydan [2007] confirm that SPG with non-monotone line searches can lead to significant acceleration on FWI and seismic reflection tomography problems with bound constraints compared to projected gradient descent.

In summary, each SPG iteration in Algorithm 2 requires at the k^{th} iteration a single evaluation of the objective $f(m^k)$ and gradient $\nabla_m f(m^k)$. In fact, SPG combines data-misfit minimization (our objective) with imposing constraints, while keeping the data-misfit/gradient and projection computations separate. When we impose the constraints, the objective and gradient do not change. Aside from computational advantages, this separation allows us to use different code bases for the objective $f(m)$ and its gradient $\nabla_m f(m)$ and the imposition of the constraints. The above separation of responsibilities also leads to a modular software design, which applies to different inverse problems that require (costly) objective and gradient calculations.

3.5.3 Spectral projected gradient with multiple constraints

We now arrive at our main contribution where we combine projections onto multiple constraints with nonlinear optimization with costly objective and gradient calculations using a spectral projected gradient (SPG) method. Recall from the previous section that the projection onto the intersection of convex sets in SPG is equivalent to running Dykstra's algorithm (Algorithm 1) —i.e., we have

$$\begin{aligned}&\mathcal{P}_{\mathcal{C}}(m^k - \alpha\nabla_m f(m^k))\\&= \mathcal{P}_{\mathcal{C}_1 \bigcap \mathcal{C}_2 \bigcap \cdots \bigcap \mathcal{C}_p}(m^k - \alpha\nabla_m f(m^k))\\&\Leftrightarrow \text{DYKSTRA}(m^k - \alpha\nabla_m f(m^k), \mathcal{P}_{\mathcal{C}_1}, \ldots, \mathcal{P}_{\mathcal{C}_p}).\end{aligned} \tag{3.15}$$

With this equivalence established, we arrive at our version of SPG presented in Algorithm 2, which has appeared in some form in the non-geophysical literature in Birgin et al. [2003] and Schmidt and Murphy [2010].

The proposed optimization algorithm for nonlinear inverse problems with multiple constraints (equation 3.6) has the following three-level nested structure:

Algorithm 2 $\min_m f(m)$ s.t. $m \in \bigcap_{i=1}^{p} \mathcal{C}_i$ with spectral projected gradient, non-monotone line searches and combined with Dykstra's algorithm.

```
input:
     // one projector per constraint set
     $\mathcal{P}_{\mathcal{C}_1}, \mathcal{P}_{\mathcal{C}_2}, \ldots$
     $m^0$ //starting model
Initialization
0.   $M =$ integer //history length for $f(m^k)$
0.   select $\eta \in (0,1)$, select initial $\alpha$
0.   $k = 1$, select sufficient descent parameter $\epsilon$
  WHILE stopping conditions not satisfied DO
1.   $f(m^k)$, $\nabla_m f(m^k)$ //objective & gradient
     // project onto intersection of sets:
2.   $r^k = \text{DYKSTRA}\big(m^k - \alpha \nabla_m f(m^k), \mathcal{P}_{\mathcal{C}_1}, \mathcal{P}_{\mathcal{C}_2}, \ldots\big)$
3.   $p^k = r^k - m^k$ // update direction
     //save previous M objective:
4a.  $f_{\text{ref}} = \{f_k, f_{k-1}, \ldots, f_{k-M}\}$
4b.  $\gamma = 1$
4c.  IF $f(m^k + \gamma p^k) < \max(f_{\text{ref}}) + \epsilon\gamma\nabla_m f(m^k)^* p^k$
        $m^{k+1} = m^k + \gamma p$ // update model iterate
        $y_k = \nabla_m f(m^{k+1}) - \nabla_m f(m^k)$
        $s_k = m^{k+1} - m^k$
        $\alpha = \frac{s_k^* s_k}{s_k^* y_k}$ // spectral steplength
        $k = k + 1$
     ELSE
        $\gamma = \eta\gamma$ //step size reduction,
        go back to 4c
 END
output: $m^k$
```

1. At the top level, we have a possibly non-convex optimization problem with a differentiable objective and multiple constraints:

$$\min_m f(m) \text{ subject to } m \in \bigcap_{i=1}^{p} \mathcal{C}_i,$$

   which we solve with the spectral projected gradient method;

2. At the next level, we project onto the intersection of multiple (convex)

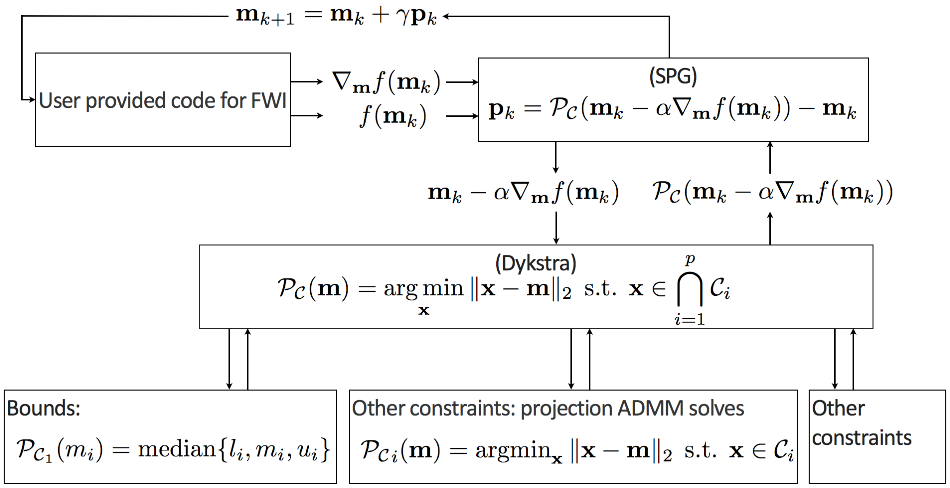

$\mathbf{m}_{k+1} = \mathbf{m}_k + \gamma \mathbf{p}_k$
User provided code for FWI
$\nabla_{\mathbf{m}} f(\mathbf{m}_k)$
$f(\mathbf{m}_k)$
(SPG)
$\mathbf{p}_k = \mathcal{P}_{\mathcal{C}}(\mathbf{m}_k - \alpha \nabla_{\mathbf{m}} f(\mathbf{m}_k)) - \mathbf{m}_k$
$\mathbf{m}_k - \alpha \nabla_{\mathbf{m}} f(\mathbf{m}_k)$
$\mathcal{P}_{\mathcal{C}}(\mathbf{m}_k - \alpha \nabla_{\mathbf{m}} f(\mathbf{m}_k))$
(Dykstra)
$\mathcal{P}_{\mathcal{C}}(\mathbf{m}) = \arg\min_{\mathbf{x}} \|\mathbf{x} - \mathbf{m}\|_2 \text{ s.t. } \mathbf{x} \in \bigcap_{i=1}^{p} \mathcal{C}_i$
Bounds:
$\mathcal{P}_{\mathcal{C}_1}(m_i) = \text{median}\{l_i, m_i, u_i\}$
Other constraints: projection ADMM solves
$\mathcal{P}_{\mathcal{C}i}(\mathbf{m}) = \text{argmin}_{\mathbf{x}} \|\mathbf{x} - \mathbf{m}\|_2 \text{ s.t. } \mathbf{x} \in \mathcal{C}_i$
Other constraints

Figure 3.5: The 3-level nested constrained optimization workflow.

sets:

$$\mathcal{P}_{\mathcal{C}}(m) = \arg\min_{x} \|x - m\|_2 \text{ subject to } x \in \bigcap_{i=1}^{p} \mathcal{C}_i$$

implemented via Algorithm 1 (Dykstra's algorithm);

3. At the lowest level, we project onto individual sets:

$$\mathcal{P}_{\mathcal{C}_i}(m) = \arg\min_{x} \|x - m\|_2 \text{ subject to } x \in \mathcal{C}_i$$

for which we use ADMM (see Appendix B) if there is no closed-form solution available.

While there are many choices for the algorithms at each level, we base our selection of any particular algorithm on their ability to solve each level without relying on additional manual tuning parameters. We summarized our choices in Figure 3.5, which illustrates the *three*-level nested optimization structure.

3.6 Numerical example

As we mentioned earlier, full-waveform inversion (FWI) faces problems with parasitic local minima when the starting model is not sufficiently accurate, and the data are cycle skipped. FWI also suffers when no reliable data are available at the low end of the spectrum (typically less than 3 Hz) or at offsets larger than about two times the depth of the model. Amongst the myriad of recent, sometimes somewhat ad hoc proposals to reduce the adverse effects of these local minima, we show how the proposed constrained optimization framework allows us to include prior knowledge on the unknown model parameters with guarantees that our inverted models indeed meet these constraints for each updated model.

Let us consider the situation where we may not have precise prior knowledge on the actual model parameters itself, but where we may still be in a position to mathematically describe some characteristics of a good starting model. With a good starting model, we mean a model that leads to significant progress towards the true model during nonlinear inversion. So our strategy is to first improve our starting model — by constraining the inversion such that the model satisfies our expectation of what a starting model looks like — followed by a second cycle of regular FWI. We relax constraints for the second cycle to allow for model updates that further improve the data fit. We present two different inversion strategies with up to three different types of constraints. Figure 3.6 shows the actual and initial starting models for this 2D FWI experiment. For this purpose, we take a 2D slice from the BG Compass velocity and density model. We choose this model because it contains realistic velocity “kick back”, which is known to challenge FWI. The original model is sampled at 6 m, and we generate “observed data” by running a time-domain [Louboutin et al., 2017] simulation code with the velocity and density models given in Figure 3.6. The sources and receivers (56 each) are located near the surface, with 100 m spacing. A coarse source and receiver spacing of a 100 m amounts to about one spatial wavelength at the highest frequency in the water; well below the spatial Nyquist sampling rate.

To mimic realistic situations where the forward modeling for the inversion misses important aspects of the wave physics, we invert for velocity only while fixing the density to be equal to one everywhere. While there are better approximations to the density model than the one we use, we intentionally use a rough approximation of the physics to show that constraints are also beneficial in that situation. To add another layer of complexity, we solve the inverse problem in the frequency domain [Da Silva and Herrmann, 2017] following the well-known multiscale frequency continuation strategy of Bunks [1995]. To deal with the situation where ocean bottom marine data are often severely contaminated with noise at the low-end of the spectrum, we start the inversion on the frequency interval $3-4$ Hz. We define this interval as a frequency batch. We subsequently use the result of the inversion with this first frequency batch as the starting model for the next frequency batch, inverting data from the $4-5$ Hz interval. We repeat this process up to frequencies on the interval $14-15$ Hz. As stopping conditions for SPG, we use a maximum of 30 data-misfit evaluations for the first frequency batch and ten for every subsequent frequency batch. SPG also terminates, and we proceed to the next frequency batch if the data-misfit change, gradient or update direction are numerically insignificant. We also estimate the unknown frequency spectrum of the source on the fly during each iteration, using the variable projection method by Pratt [1999]; Aravkin and van Leeuwen [2012]. To avoid additional complications, we assume the sources and receivers to be omnidirectional with a flat spatial frequency spectrum.

While frequency continuation and on-the-fly source estimation are both well-established techniques by now, the combination of velocity-only inversion and a poor starting model remains challenging because we *(i)* ignore density variations in the inversion, which means we can never hope to fit the observed data fully; *(ii)* we miss the velocity kick back at roughly $300-500$ m in the starting model; and *(iii)* we invert on an up to roughly $10\times$ coarser grid compared to the fine $6\,m$ grid on which the "observed" time-domain data were generated. Because of these challenges, battle-tested multiscale workflows for FWI, where we start at the low frequencies and gradually work our way up to higher frequencies, fail even if we impose bound constraints

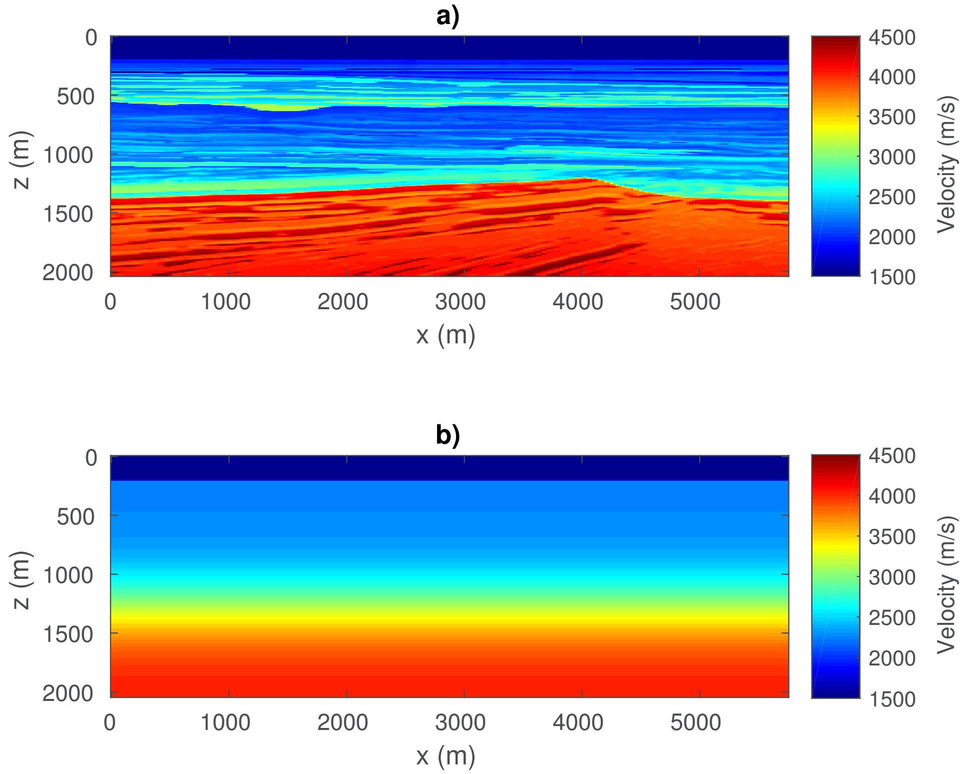


Figure 3.6: True (a) and initial (b) velocity models for the example.

(minimum of 1425 (m/s) and maximum 5000 (m/s)) values for the estimated velocities) on the model. See Figure 3.7. Only the top 700 m of the velocity model is inverted reasonably well. The bottom part, on the other hand, is far from the true model almost everywhere. The main discontinuity into the ≥ 4000 (m/s) rock is not at the correct depth and does not have the right shape.

To illustrate the potential of adding more constraints on the velocity model, we follow a heuristic that combines multiple warm-started multi-scale FWI cycles with a relaxation of the constraints. This approach was successfully employed in earlier work by Esser et al. [2016b]; Esser et al. [2018], and Chapter 2. We present two different strategies with different constraints that both lead to improved results, which shows that there is

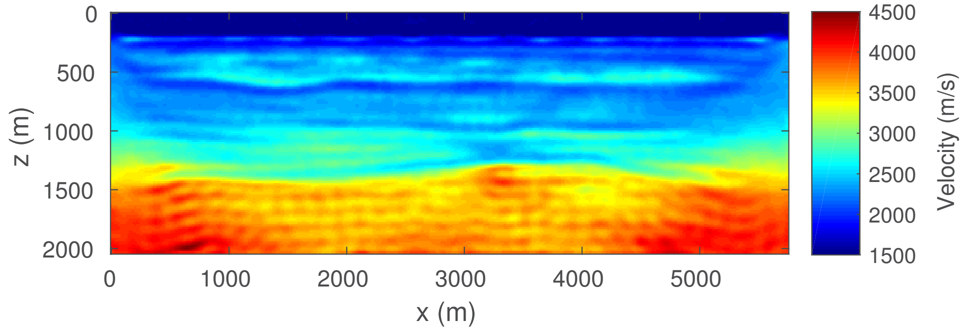


Figure 3.7: Model estimate obtained by FWI with bound constraints only.

more than one way to use multiple constraints to arrive at the desired results. Since we are dealing with a relatively undistorted sedimentary basin (see Figure 3.6), we impose constraints that limit lateral variations and force the inverted velocities to increase monotonically with depth during the first inversion cycle. In the second cycle, we relax this condition. We accomplish this by combining box constraints with slope constraints in the vertical direction (described in detail in Appendix B). To enforce continuity in the lateral direction, we work with tighter slope constraints in that direction. Specifically, we limit the variation of the velocity per meter in the depth direction (z-coordinate) of the discretized model $m[i, j] = m(i\Delta z, j\Delta x)$. Mathematically, we enforce $0 \leq (m[i+1, j] - m[i, j])/\Delta z \leq +\infty$ for $i = 1 \cdots n_z$ and $j = 1 \cdots n_x$, where n_z, n_x are the number of grid points in the vertical and lateral direction, and Δz the grid size in depth. With this slope constraint, the inverted velocities are only allowed to increase monotonically with depth, but there is no limit on how fast the velocity can increase in that direction. We impose lateral continuity by selecting the lateral slope constraint as $-\varepsilon \leq (m[i, j+1] - m[i, j])/\Delta x \leq +\varepsilon \quad$ for all $i = 1 \cdots n_z, j = 1 \cdots n_x$. The scalar ε is a small number set in the physical units of velocity (meter/second) change per meter and Δx is the grid size in the lateral direction. We select $\varepsilon = 1.0$ for this example.

Compared to other methods that enforce continuity, e.g., via a sharpening operator in a quadratic penalty term, these slope constraints have several advantages. First, they have a natural interpretable physical parameter ε with the units of velocity (meter/second) change per meter. Second, they are met at each point in the model—i.e., they are applied and enforced pointwise; and most importantly these slope constraints do not impose more structure than needed. For instance, the vertical slope constraint only enforces monotonic increases and nothing else. We do not claim that other methods, such as Tikhonov regularization, cannot accomplish these features. We claim that we do this without nebulous parameter tuning and with guarantees that our constraints are satisfied at each iteration of FWI.

The FWI results with slope constraints for $3-4$ Hz data are shown in Figure 3.8a. This result from the first FWI cycle improves the starting model significantly without introducing geologically unrealistic artifacts. This partially inverted model can now serve as input for the second FWI cycle where we invert data over a broader frequency range between $3-15$ Hz (cf. Figure 3.8b) using box constraints only. Apparently, adding slope constraints during the first cycle is enough to prevent the velocity model from moving in the wrong direction while allowing for enough freedom to get closer to the true model underlying the success of the second cycle without slope constraints. This example demonstrates that keeping the recovered velocity model after the first FWI cycle in check — via not too constrained constraints — can be a successful strategy even though final velocity model does not lie in the constraint set imposed during the first FWI cycle where velocity kick back was not allowed. We kept the computational overhead of this multi-cycle FWI method to a minimum by working with low-frequency data only during the first cycle, which reduces the size of the computational grid by a factor of about fourteen.

The second strategy is similar to the total-variation constraint continuation strategies proposed by Esser et al. [2016b], Esser et al. [2018], and in Chapter 2 to deal with salt structures. We will show that this strategy can also be beneficial for sedimentary geology. The experimental setting is the same as before. This time we use two different constraints instead of three: bounds and TV constraints as in Chapter 2. The (anisotropic) TV constraint is defined as $\{m \mid \|Am\|_1 \leq \sigma\}$, where the matrix A con-

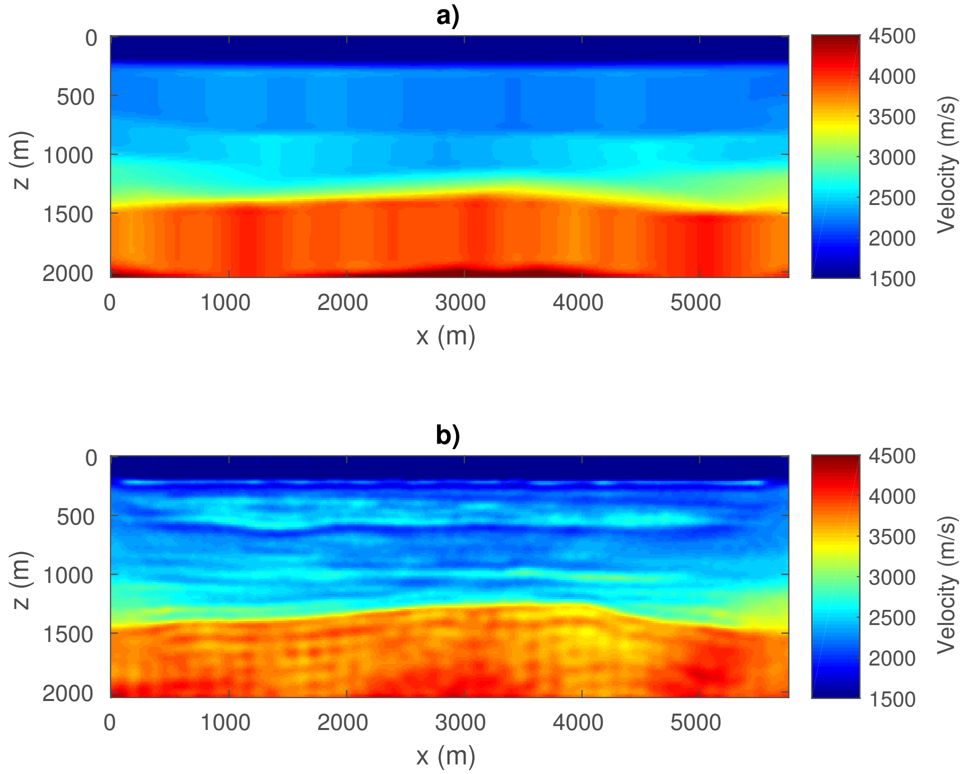


Figure 3.8: (a) Model estimate obtained by FWI from $3-4$ Hz data with bound constraints, a vertical slope constraint and a constraint on the velocity variation per meter in the horizontal direction. (b) Model estimate by FWI from $3-15$ Hz data with bound constraints and using the result from (a) as the starting model.

tains the discretized horizontal and vertical derivative matrices. We select $\sigma = 1.0\|Am_0\|_1$ for the first cycle that uses $3-4$ Hz data only, i.e., the TV-constraint is set to the TV of the initial model, m_0, see Figure 3.6b. The second cycle works with $3-15$ Hz data, as before. This time we use bound constraint only. The results in Figure 3.9 show that the first cycle with a tight TV constraint improves on the laterally invariant starting model (Figure 3.6b), but also displays an incorrect low-velocity zone in the

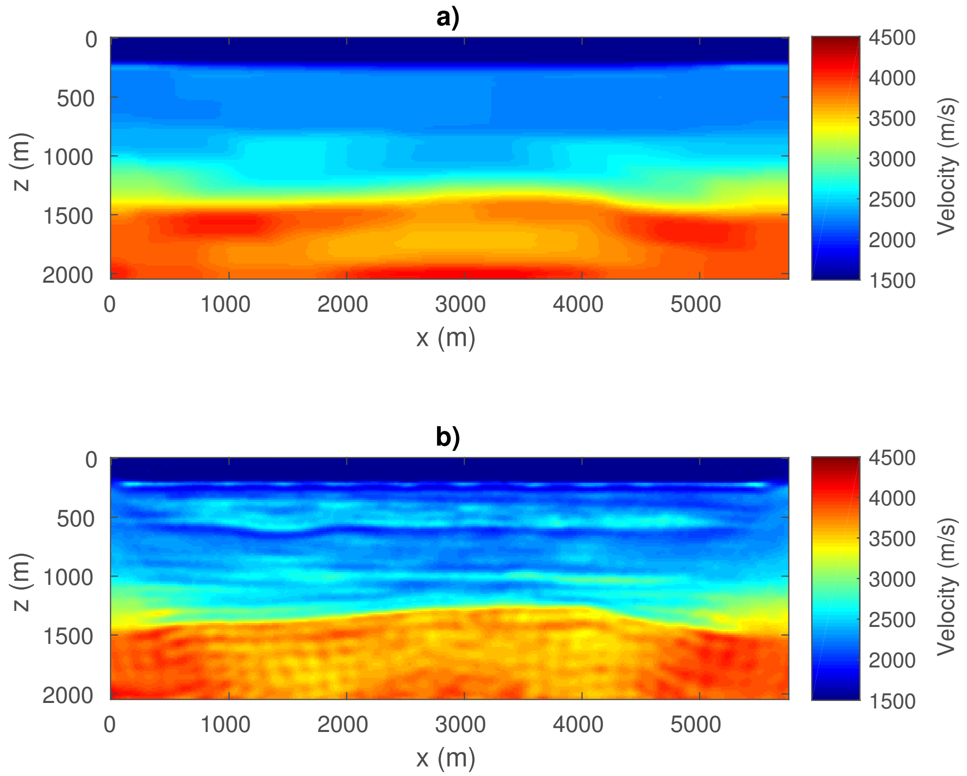


Figure 3.9: (a) Model estimate obtained by FWI from $3 - 4$ Hz data with bound constraints and total-variation constraints. (b) Model estimate by FWI from $3 - 15$ Hz data with bound constraints and using the result from (a) as the starting model.

high-velocity rock near the bottom of the model. The result of the second constrained FWI cycle, Figure 3.9b shows that the first cycle improved the starting model sufficiently, such that the second cycle using all frequency data can estimate a model similar to the true model.

Both FWI results with multiple constraints appear to be much closer to the true model than the FWI result that uses bound constraints only. We gain more insight into the quality of the models by looking at reverse-time migrations (RTM) for each of the three FWI results. We show the RTM results and true reflectivity of the velocity model in Figure 3.10. The results based on FWI with bound constraints show the least similarity with the true reflectivity (Figures 3.10a and 3.10d) because a number of strong reflectors are missing. The RTM results based on FWI with bound constraints only also do not show coherent layers below a depth of 1500m depth. The other RTM images based on FWI with multiple constraints (Figures 3.10b, 3.10c, 3.10e, and 3.10f) are similar to each other and closer to the true reflectivity.

This example was designed to illustrate how our framework for constrained FWI can be of great practical use for FWI problems where good starting models are missing or where low-frequencies and long offsets are absent. Our proposed method is not tied to a specific constraint. For different geological settings, we can use the same approach, but with different constraints. We presented two different strategies. The preferable strategy depends on the available prior knowledge. Computationally, both strategies work with constraints for which we can compute the projections as in Appendix A.

3.6.1 Comparison with a quadratic penalty method

We repeat the FWI experiment from the previous section, but this time we regularize using one of the most widely used regularization techniques in the geophysical literature: the quadratic penalty method. This comparison illustrates the benefits of the constrained formulation as we described in earlier sections.

To apply a quadratic penalty method as in equation 3.1, we need to come up with penalty functions that represent our prior information, and we also need to find one scalar penalty parameter per penalty function, such that the final model satisfies all prior information. The first piece of prior information is that a starting model is smooth in the lateral direction. The penalty function $R_1(m) = \alpha_1/2\|D_x m\|_2^2$ promotes smoothness in the lateral direction, using the lateral finite-difference matrix D_x. The second piece of prior information is that a starting model has an almost monotonically increasing velocity with depth. We use $R_2(m) = \alpha_2/2\|D_z m\|_2^2$ to promote vertical smoothness. We see two disadvantages of quadratic penalties com-

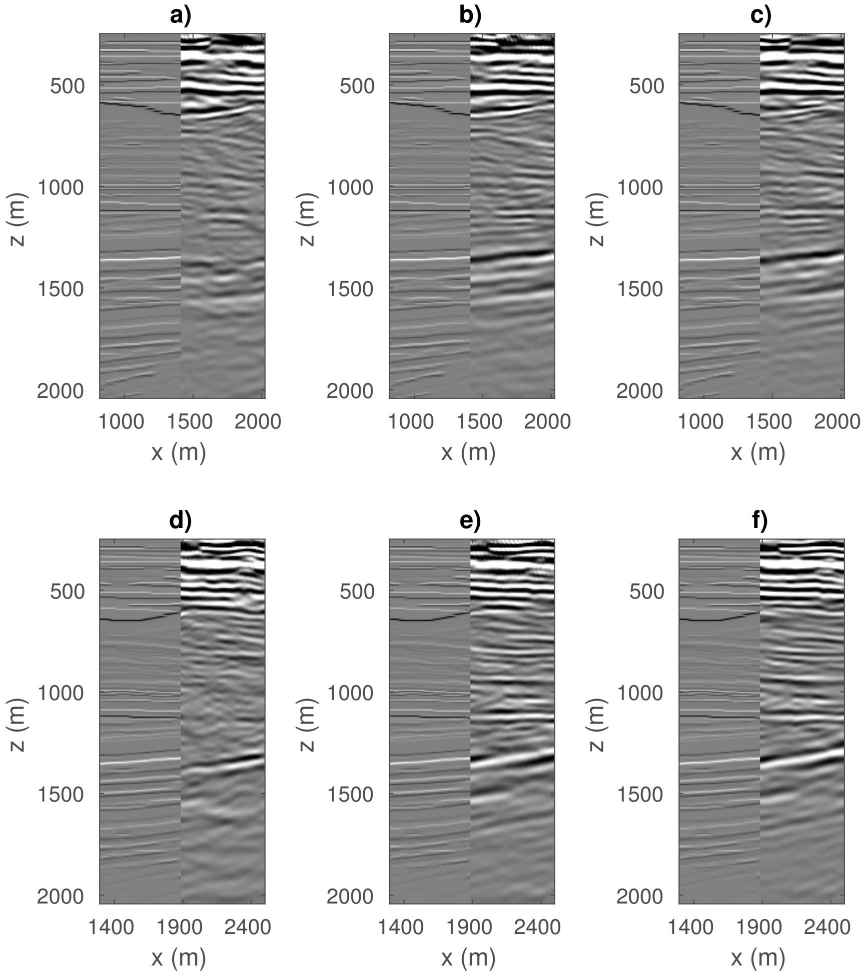


Figure 3.10: Comparison of reverse time migration (RTM) results based on the FWI velocity models (right halves) and the true reflectivity (left halves). Figures (a) and (d) show RTM based on the velocity model from FWI with bounds only (Figure 3.7). Figures (b) and (e) show RTM results based on the velocity model from FWI with bounds, horizontal and vertical slope constraints (Figure 3.8b). Figures (c) and (f) show RTM results based on the velocity model from FWI with bounds and total-variation constraints (Figure 3.9b). RTM results based on FWI with bound constraints, (a) and (d), miss a number of reflectors that are clearly present in the other RTM results.

pared to the constrained formulation. First, the quadratic penalty function R_2 does not generally lead to monotonicity. In order to promote monotonicity with a penalty function, we would need to work with non-differentiable functions. Alternatively, we could smooth the function, but this introduces another smoothing parameter and leads to unpredictable behavior of FWI as a function of parameter choices, as discussed in Lin and Huang [2015] and Chapter 2. The second disadvantage of the penalty approach is the selection of penalty parameters α_1 and α_2. Whereas the constrained formulation allows us to select the maximum variation of the velocity per meter, the penalty approach requires two parameters without clear physical meaning. These two parameters have no direct relation to the prior information. The effect of a penalty parameter depends on the data-misfit, as well on all other penalty parameters. We simplify the regularization task for the quadratic penalty method by ignoring a penalty function to enforce bounds on the velocities. We use projection onto the bounds so we can focus on the effect to two penalties.

We show FWI results in Figures 3.11 and 3.12, based on various combinations of penalty parameters to illustrate the well-known effect that it is easy to over/under estimate a parameter, leading to a result that does not have the desired properties. We selected the penalty parameters by manual fine-tuning. Some of the results in Figures 3.11 and 3.12 look similar to the true model, but contain some critical artifacts. Most noticeable is the peak of the high-velocity (+4000m/s) rock at the bottom part of the model, which should be located close to $x = 4000$m. The results from quadratic penalty regularization put the peak at the wrong location and often show a flat top rather than a peak. The estimated velocities of the high-velocity rock at the bottom of the model are also lower than in the model obtained with slope constraints, Figure 3.8.

Another observation about the penalty method FWI results in Figures 3.11 and 3.12, is that larger penalty parameters lead to more smoothness, but it is not clear and intuitive how much the penalty parameters should be increased to obtain the desired level of smoothness. In contrast, constraints provide a way to set the limits on smoothness that will be sat-

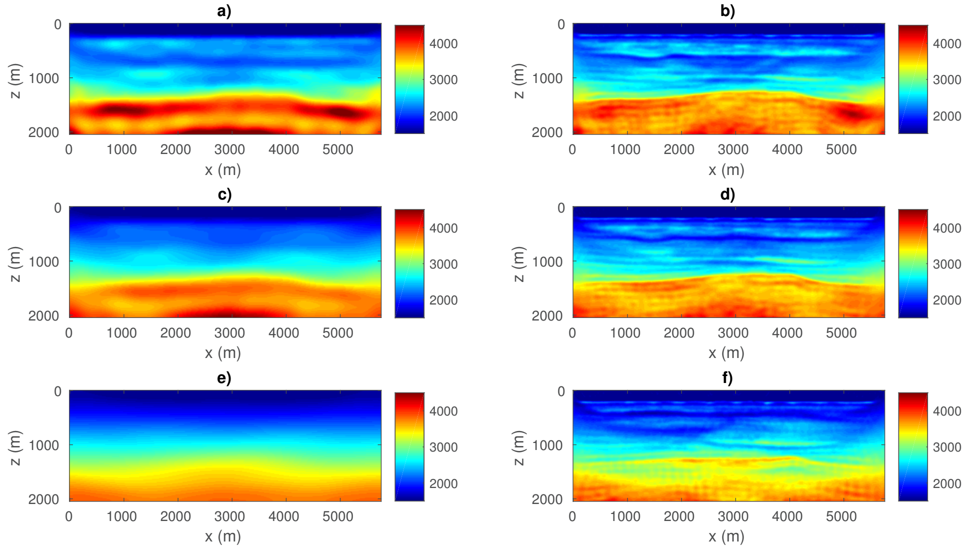


Figure 3.11: Results from FWI with regularization by a quadratic penalty method to promote horizontal and vertical smoothness. As for the constrained FWI example, the first FWI cycle uses $3 - 4$ Hz data and is with regularization (left column), the second cycle uses $3 - 15$ Hz data and does not use regularization (right column). Figure (a) uses regularization parameter $\alpha_1 = \alpha_2 = 1e5$, (c) uses $\alpha_1 = \alpha_2 = 1e6$, and (e) uses $\alpha_1 = \alpha_2 = 1e7$.

isfied at every FWI iteration by construction of the projection method. For example, if we want to increase the smoothness by a factor of two, we need to constrain the velocity variation per meter to half the previous limit, see Chapter 2 for FWI examples that illustrate this point.

3.7 Discussion

Our main contribution in solving optimization problems with multiple constraints is that we employ a hierarchical divide and conquer approach to handle problems where objectives and gradient evaluations require PDE solves. We arrive at this result by splitting each problem into simpler and therefore computationally more manageable subproblems. We start from the top with

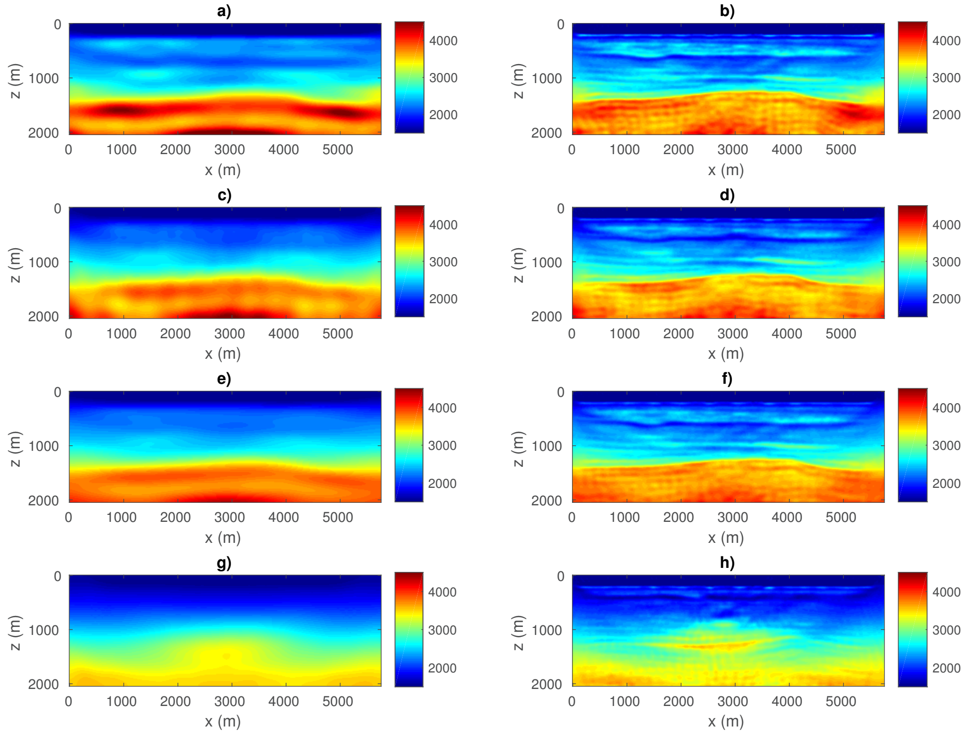


Figure 3.12: Results from FWI with regularization by a quadratic penalty method to promote horizontal and vertical smoothness. As for the constrained FWI example, the first FWI cycle uses $3-4$ Hz data and is with regularization (left column), the second cycle uses $3-15$ Hz data and does not use regularization (right column). Figure (a) uses regularization parameter $\alpha_1 = 1e6$, $\alpha_2 = 1e5$, (c) uses $\alpha_1 = 1e5$, $\alpha_2 = 1e6$, (e) uses $\alpha_1 = 1e7$, $\alpha_2 = 1e6$, and (g) uses $\alpha_1 = 1e6$, $\alpha_2 = 1e7$.

spectral projected gradient (SPG), which splits the constrained optimization problem into an optimality (decreasing the objective) and feasibility (satisfying all constraints) problem, and continue downwards by satisfying the individual constraints using Dykstra's algorithm. Even at the lowest level, we employ this strategy when there is no closed form projection available for the constraints. We use the alternating direction method of multipliers (ADMM) for the examples. As a result, we end up with an algorithm that remains computationally feasible for large-scale problems where evaluation of objectives and gradients is computationally costly.

So far, the minimization of our optimality problem relied on first-order derivative information only and what is essentially a scalar approximation of Hessian via SPG. Theoretically, we can also incorporate Dykstra's algorithm into projected quasi-Newton [Schmidt et al., 2009] or (Gauss-) Newton methods [Schmidt et al., 2012, Lee et al., 2014]. However, unlike SPG, these approaches usually require more than one projection computation per FWI iteration to solve quadratic sub-problems with constraints. We would require a more careful evaluation to see if second-order methods in this case indeed provide advantages compared to projected first-order methods such as SPG.

We also would like to note that there exist parallel versions of Dykstra's algorithm and similar algorithms [Censor, 2006, Combettes and Pesquet, 2011, Bauschke and Koch, 2015]. These algorithms compute all projections in parallel, so each Dykstra iteration takes as much time as the slowest projection computation. As a result, the time per Dykstra iteration does not necessarily increase if there are more constraint sets.

While the primary application and motivation for our work is full-waveform inversion, the developed framework also applies to other geophysical inverse problems; specifically, problems where the data-misfit and gradient evaluation require the solution of many partial-differential equations.

3.8 Conclusions

Because of its computational complexity and notorious local minima, full-waveform inversion easily ranks amongst one of the most challenging non-linear inverse problems. To meet this challenge, we introduced a versatile optimization framework for (non)linear inverse problems with the following key features: *(i)* it invokes prior information via projections onto the intersection of multiple (convex) constraint sets and thereby avoids reliance on cumbersome trade-off parameters; *(ii)* it allows for imposing arbitrarily many constraints simultaneously as long as their intersection is non-empty; *(iii)* it projects the updated models uniquely on the intersection at every iteration and as such stays away from ambiguities related to the order in which the constraints are invoked; *(iv)* it guarantees that model updates satisfy all constraints simultaneously at each iteration and *(v)* it is built on top of existing code bases that only need to compute data-misfit objective values and gradients. These features in combination with our ability to relax and add other constraints that have appeared in the geophysical literature offer a powerful optimization framework to mitigate some of the adverse effects of local minima.

Aside from promoting certain to-be-expected model properties, our examples also confirmed that invoking multiple constraints as part of a multi-cycle inversion heuristic can lead to better results. We observe improvements during the first full-waveform inversion cycle(s) if the constraint sets are tight enough to prevent unrealistic geological features to enter into the model estimate. Provided the inversions make some progress to the solution, later inversion cycles will benefit if the tight constraints are subsequently relaxed either by dropping them or by increasing the size of the constraint set. This strategy follows the heuristic of first estimating a better starting model, or otherwise simple model, followed by introducing more details. Constraints provide us with precise control of the maximum model complexity at each FWI iteration. Our examples confirm this important aspect and clearly demonstrate the advantages of working with constraints that are satisfied at each iteration of the inversion.

Compared to many other regularization methods, our approach is easily extendable to other convex or non-convex constraints. However, for non-convex constraints, we can no longer offer certain guarantees, except that all sub-problems in the alternating direction method of multipliers remain solvable without the need to tune trade-off parameters manually. We can do this because we work with projections onto the intersection of multiple sets and we split the computations into multiple pieces that have closed-form solutions.

Chapter 4

Algorithms and software for projections onto intersections of convex and non-convex sets with applications to inverse problems.

4.1 Introduction

We consider problems of the form

$$\mathcal{P}_{\mathcal{V}}(m) \in \arg\min_{x} \frac{1}{2}\|x-m\|_2^2 \quad \text{subject to} \quad x \in \bigcap_{i=1}^{p} \mathcal{V}_i, \tag{4.1}$$

which is the projection of a vector $m \in \mathbb{R}^N$ onto the intersection of p convex and possibly non-convex sets $\mathcal{V}_i$. The projection in equation (4.1) is unique if all sets are closed and convex. The projection operation is a common tool used for solving constrained optimization problems of the form

$$\min_m f(m) \quad \text{subject to} \quad m \in \bigcap_{i=1}^{p} \mathcal{V}_i. \tag{4.2}$$

Examples of algorithms that use projections include spectral projected gradient descent [SPG, Birgin et al., 1999], projected quasi-Newton [Schmidt et al., 2009], and projected Newton-type methods [Bertsekas, 1982, Schmidt et al., 2012]. In the above optimization problem, the function $f(m) : \mathbb{R}^N \rightarrow \mathbb{R}$ is at least twice differentiable and may also be non-convex. Alternatively, proximal algorithms solve

$$\min_m f(m) + \iota_{\mathcal{V}}(m), \tag{4.3}$$

which is equivalent to (4.2) and where $\iota_{\mathcal{V}}(m)$ is the indicator function of the set $\mathcal{V} \equiv \bigcap_{i=1}^{p} \mathcal{V}_i$, which returns zero when we are in the set and infinity otherwise. Because applications may benefit from using non-convex sets $\mathcal{V}_i$, we also consider those sets in the numerical examples. While we do not provide convergence guarantees for this case, we will work with some useful/practical heuristics.

The main applications of interest in this work are inverse problems for the estimation of physical (model) parameters ($m \in \mathbb{R}^n$) from observed data ($d_{\text{obs}} \in \mathbb{C}^s$). Notable examples are geophysical imaging problems with seismic waves [full-waveform inversion, see, e.g., Tarantola, 1986, Pratt et al., 1998, Virieux and Operto, 2009] for acoustic velocity estimation and direct-current resistivity problems [DC-resistivity, see, e.g., Haber, 2014] to obtain electrical conductivity information. These problems all have 'expensive' forward operators, i.e., evaluating the objective $f(m)$ requires solutions of many partial-differential-equations (PDEs) if the PDE constraints are implicit in $f(m)$, which corresponds to a reduced data-misfit [Haber et al., 2000]. In our context, each set $\mathcal{V}_i$ describes a different type of prior information on the model m. Examples of prior knowledge as convex sets are bounds on parameter values, smoothness, matrix properties such as the nuclear norm, and whether or not the model is blocky with sharp edges (total-variation like constraints via the ℓ_1 norm). Non-convex sets that we use in the numerical examples include the annulus (minimum and maximum ℓ_2 norm), limited matrix rank, and vector cardinality.

Aside from the constrained minimization as in problem (4.2), we consider feasibility (also known as set-theoretic estimation) problem formulations [e.g., Youla and Webb, 1982, Trussell and Civanlar, 1984, Combettes, 1993, 1996]. Feasibility only formulations accept any point in the intersection of sets $\mathcal{V}_i$ that describe constraints on model parameter properties, and a data-fit constraint $\mathcal{V}_p^{\text{data}}$ that ties the unknown model vector x to the observed data $d_{\text{obs}} \in \mathbb{R}^M$ via a forward operator $F \in \mathbb{R}^{M \times N}$. Examples of data-constraint sets are $\mathcal{V}^{\text{data}} = \{x \mid l \leq (Fx - d_{\text{obs}}) \leq u\}$ and $\mathcal{V}^{\text{data}} = \{x \mid \|Fx - d_{\text{obs}}\|_2 \leq \sigma\}$. The upper and lower bounds are vectors l and u and $\sigma > 0$ is a scalar that depends on the noise level. The forward operators are linear and often computationally 'cheap' to apply. Examples include masks and blurring kernels. In case there is a good initial guess available, we can choose to solve a projection rather than feasibility problem by adding the squared ℓ_2 distance term as follows:

$$\min_x \frac{1}{2}\|x - m\|_2^2 \quad \text{s.t.} \quad \begin{cases} x \in \mathcal{V}_p^{\text{data}} \\ x \in \bigcap_{i=1}^{p-1} \mathcal{V}_i \end{cases} . \tag{4.4}$$

To demonstrate the benefits of this constrained formulation, we recast joint denoising-deblurring-inpainting and image desaturation problems as projections onto the intersection of sets. Especially when we have a few training examples from which we can learn constraint set parameters, the feasibility and projection approaches conveniently add many pieces of prior knowledge in the form of multiple constraint sets, but without any penalty or trade-off parameters. For instance, [Combettes and Pesquet, 2004] show that we can observe 'good' choices of parameters that define the constraint sets, such as the average of the total variation of a few training images. We address increasing computational demand that comes with additional constraint sets with a reformulation of problem (4.4), such that we take into account similarity between sets, and split the problem up into simple parallel computations where possible.

Projected gradient and similar algorithms naturally split problem (4.2) into a projection and data-fitting part. In this setting, software for computing projections onto the intersection of sets can then work together with codes for physical simulations that compute $f(m)$ and $\nabla_m f(m)$, as we show in one of the numerical examples. See `dolfin-adjoint` [Farrell et al., 2013], `Devito` [Kukreja et al., 2016, Louboutin et al., 2018] in `Python` and `WAVEFORM` [Da Silva and Herrmann, 2017], `jInv` [Ruthotto et al., 2017], and `JUDI` [Witte et al., 2018] in `Julia` for examples of recent packages for physical simulations that also compute $\nabla_m f(m)$.

Compared to regularization via penalty functions (that are not an indicator function), constrained problem formulations (4.2 and 4.4) have several advantages when solving physical parameter estimation problems. Penalty methods

$$\min_m f(m) + \sum_{i}^{p} \alpha_i R(m) \tag{4.5}$$

add prior knowledge through $p \geq 1$ penalty functions $R_i(m) : \mathbb{R}^N \rightarrow \mathbb{R}$ with scalar weights $\alpha_i > 0$ to the data-misfit term $f(m)$. Alternatively, we can add penalties to the objective and work with a data constraint instead—i.e., we have

$$\min_m \sum_{i=1}^{p} \alpha_i R_i(m) \quad \text{s.t.} \quad f(m) \leq \sigma, \tag{4.6}$$

generally referred to as Basis Pursuit Denoise [Mallat and Zhang, 1992, Chen et al., 2001, van den Berg and Friedlander, 2009, Aravkin et al., 2014], Morozov/residual regularization [Ivanov et al., 2013], or Occam's inversion [Constable et al., 1987]. The scalar σ relates to the noise level in the data. For convex constraints/objectives/penalties, constrained, penalty and data-constrained problems are equivalent under certain conditions and for specific α - σ pairs [Vasin, 1970, Gander, 1980, Golub and von Matt, 1991, van den Berg and Friedlander, 2009, Aravkin et al., 2016, Tibshirani, 2017], but differ in algorithmic implementation and in their ability to handle multiple pieces of prior information ($p > 1$). In that case, the simplicity of adding penalties is negated by the challenge of selecting multiple trade-off parameters (α_i). For this, and for reasons we list below, we prefer constrained formulations that involve projections onto the intersection of constraint sets (problem 4.1). Constrained formulations

- **satisfy prior information at every iteration** PDE-based inverse problems require the solutions of PDEs to evaluate the objective function $f(m)$ and its gradient. The model parameters need to be in an interval for which the mesh (PDE discretization) is suitable, i.e., we have to use bound constraints. Optimization algorithms that satisfy all constraints at every iteration also give the user precise control of the model properties when solving problem (4.2) using a projection-based algorithm. This allows us to start solving a non-convex inverse problem with certain constraints, followed by a solution stage with 'looser' constraints. [Smithyman et al., 2015, Esser et al., 2016b, Esser et al., 2016], as well as examples in Chapter 2 apply this strategy to seismic full-waveform inversion to avoid local minimizers that correspond to geologically unrealistic models.

- **require a minimum number of manual tuning parameters for multiple constraints** We want to avoid the time-consuming and possibly computationally costly procedure of manually tuning numerous nuisance parameters. While we need to define the constraint sets, we avoid the scalar weights that penalty functions use. Constraint sets have the advantage that their definitions are independent of all other constraint definitions. For penalty functions, the effect of the weights α_i associated with each R_i on the solutions of an inverse problem depends on all other α_i and R_i. For this reason, selecting multiple scalar weights to balance multiple penalty functions becomes increasingly difficult as we increase the number of penalties.

- **make direct use of prior knowledge** We can observe model properties from training examples and use this information directly as constraints [Combettes and Pesquet, 2004, see also numerical examples

> in this work]. Penalty and basis-pursuit type methods first need to translate this information into penalty functions and scalar weights.

Most classical and recently proposed methods to project onto an intersection of multiple (convex) sets, such as Dykstra's algorithm and variants [Dykstra, 1983, Boyle and Dykstra, 1986, Censor, 2006, Bauschke and Koch, 2015, López and Raydan, 2016, Aragón Artacho and Campoy, 2018], (see also Appendix C), use projections onto each set separately, $\mathcal{P}_{\mathcal{V}_i}(\cdot)$, as the main computational component. The projection is a black box, and this may create difficulties if the projection onto one or more sets has no known closed-form solution. We then need another iterative algorithm to solve the sub-problems. This nesting of algorithms may lead to problems with the selection of appropriate stopping criteria for the algorithm that solves the sub-problems. In that case, we need two sets of stopping criteria: one for Dykstra's algorithm itself and one for the iterative algorithm that computes the individual projections. For this reason, it may become challenging to select stopping criteria for the algorithm that computes a single projection. For example, projections need to be sufficiently accurate such that Dykstra's algorithm converges. At the same time, we do not want to waste computational resources by solving sub-problems more accurately than necessary. A second characteristic of the black-box projection algorithms is that they treat every set individually and do not attempt to exploit similarities between the sets. If we work with multiple constraint sets, some of the set definitions may include the same or similar linear operators in terms of sparsity (non zero) patterns.

Besides algorithms that are designed to solve a specific projection problem onto the intersection of multiple sets, there exist software packages capable of solving a range of generic optimization problems. However, many of the current software packages are not designed to compute projections onto intersections of multiple constraint sets where we usually do not know the projection onto each set in closed form. This happens, for instance, when the set definitions include linear operators A that satisfy the relation $AA^\top \neq \alpha I$ for $\alpha > 0$. A package such as `Convex` for Julia [Udell et al., 2014], an example of disciplined convex programming (DCP), does not handle non-convex sets and requires lots of memory even for 2D problems. The high memory demands are a result of the packages that `Convex` can call as the back-end, for example, `SCS` [O'Donoghue et al., 2016] or `ECOS` [Domahidi et al., 2013]. These solvers work with matrices that possess a structure similar to

$$\begin{pmatrix} \star & A^\top \\ A & \star \end{pmatrix}, \tag{4.7}$$

where the matrix A vertically stacks all linear operators that are part of equality constraints. Both the block-structured system (4.7) and A become prohibitively large in case we work with multiple constraint sets that include a linear operator in their definitions. The software that comes closer to our implementation is `Epsilon` [Wytock et al., 2015], which is written in Python. Like our proposed algorithms, `Epsilon` also employs the alternating direction method of multipliers (ADMM), but reformulates optimization problems by emphasizing generalized proximal mappings as in equation (4.12, see below). Linear equality constraints then appear as indicator functions, which leads to different linear operators ending up in different sub-problems. In contrast, we work with a single ADMM sub-problem that includes all linear operators. The `ProxImaL` software [Heide et al., 2016] for Python is designed for linear inverse problems in imaging using ADMM with a similar problem reformulation. However, `ProxImaL` differs fundamentally since it applies regularization with a relatively small number of penalty functions. While in principle it should be possible to adapt that package to constrained problem formulations by replacing penalties with indicator functions, `ProxImaL` is in its current form not set up for that purpose. Finally there is `StructuredOptimization` [Antonello et al., 2018] in `Julia`. This package also targets inverse problems by smooth+non-smooth function formulations. Different from the goal of this work, `StructuredOptimization` focusses on problems with easy to compute generalized proximal mappings (4.12), i.e., penalty functions or constraints that are composed with linear operators that satisfy $AA^\top = \alpha I$. In contrast, we focus on the situation where we have many constraints with operators ($AA^\top \neq \alpha I$) that make generalized proximal mappings (4.12) difficult to compute. Below, we list additional benefits of our approach compared to existing packages that can solve intersection projection problems.

4.1.1 Contributions

Our aim is to design and implement parallel computational optimization algorithms for solving projection problems onto intersections of multiple constraint sets in the context of inverse problems. To arrive at this optimization framework, `SetIntersectionProjection` , we propose

- an implementation that avoids nesting of algorithms and exploits similarities between constraint sets, unlike black-box alternating projection methods such as Dykstra's algorithm. Taking similarities between sets into account allows us to work with many sets at a relatively small increase in computational cost.

- algorithms that are based on a relaxed variant of the simultaneous direction method of multipliers [SDMM, Afonso et al., 2011, Combettes and Pesquet, 2011, Kitic et al., 2016]. By merging SDMM with recently developed schemes for automatically adapting the augmented-Lagrangian penalty and relaxation parameters [Xu et al., 2017b,a], we achieve speedups when solving problem (4.1) compared to the straightforward application of operator splitting such as the alternating direction method of multipliers (ADMM) that use fixed parameters or older updating schemes.

- a software design specifically for set intersection projection problems. Our specializations enhance computational performance and include *(i)* a relatively simple multilevel strategy for ADMM-based algorithms that does part of the computations on significantly coarser grids; *(ii)* solutions of banded linear systems in compressed diagonal format (CDS) with multi-threaded matrix-vector products (MVP). These MVPs are faster than general purpose storage formats like compressed sparse column storage (CSC). Unlike linear system solves by Fourier diagonalization they support linear operators with spatially varying (blurring) kernels and various boundary conditions. See discussion by, e.g., [Almeida and Figueiredo, 2013, O'Connor and Vandenberghe, 2017] *(iii)* more intuitive stopping criteria based on set feasibility.

- to make our work available as a software package in `Julia` [Bezanson et al., 2017]. Besides the algorithms, we also provide scripts for setting up the constraints, projectors and linear operators, as well as various examples. All presented timings, comparisons, and examples are reproducible.

- an implementation that is suitable for small matrices (2D) up to larger tensors (3D models, at least $m \in \mathbb{R}^{300\times300\times300}$). Because we solve simple-to-compute sub-problems in closed form and independently in parallel, the proposed algorithms work with large models and many constraints. We achieve this because there is only a single inexact linear-system solve that does not become much more computationally expensive as we add more constraint sets. To improve the performance even further, we also provide a multilevel accelerated version.

To demonstrate the capabilities of our optimization framework and implementation, we provide examples of how projections onto an intersection of multiple constraint sets can be used to solve linear image processing tasks such as denoising and deconvolution and more complicated inverse problems including nonlinear parameters estimation problems with PDEs.

4.2 Notation, assumptions, and definitions

Our goal is to estimate the model vector (e.g., discretized medium parameters such as the acoustic wave speed) $m \in \mathbb{R}^N$, which in 2D corresponds to a vectorized (lexicographically ordered) matrix of size $n_z \times n_x$ with z the vertical coordinate and x the horizontal direction. There are $N = n_x \times n_z$ elements in a 2D model. Our work applies to 2D and 3D models but to keep the derivations simpler we limit ourselves to 2D models discretized on a regular grid. We use the following discretization for the vertical derivative in our constraints

$$D_z = \frac{1}{h_z} \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix}, \tag{4.8}$$

where h_z is the vertical grid size. We define the discretized vertical derivative for the 2D model as the Kronecker product of D_z and the identity matrix corresponding to the x-dimension: $D_z \otimes I_x$.

The indicator function of a convex or non-convex set $\mathcal{C}$ is defined as

$$\iota_{\mathcal{C}}(m) = \begin{cases} 0 & \text{if } m \in \mathcal{C}, \\ +\infty & \text{if } m \notin \mathcal{C}. \end{cases} \tag{4.9}$$

We define the Euclidean projection onto a convex or non-convex set $\mathcal{C}$ as

$$\mathcal{P}_{\mathcal{C}}(m) = \arg\min_x \|x - m\|_2^2 \quad \text{s.t.} \quad m \in \mathcal{C}. \tag{4.10}$$

This projection is unique if $\mathcal{C}$ is a closed and convex set. If $\mathcal{C}$ is a non-convex set, the projection may not be unique so the result is any vector in the set of minimizers of the projection problem. The proximal map of a function $g(m) : \mathbb{R}^N \rightarrow \mathbb{R} \cup \{+\infty\}$ is defined as

$$\text{prox}_{\gamma,g}(m) = \arg\min_x g(x) + \frac{\gamma}{2}\|x - m\|_2^2, \tag{4.11}$$

so $\text{prox}_{\gamma,g}(m) : \mathbb{R}^N \rightarrow \mathbb{R}^N$, where $\gamma > 0$ is a scalar. The case when $g(x)$ includes a linear operator $A \in \mathbb{R}^{M \times N}$ is of particular interest to us and we make it explicit with the definition

$$\text{prox}_{\gamma,g\circ A}(m) = \arg\min_x g(Ax) + \frac{\gamma}{2}\|x - m\|_2^2. \tag{4.12}$$

Even though $\text{prox}_{\gamma,g}(m)$ is often available in closed-form solution, or cheap to compute [Combettes and Pesquet, 2011, Parikh and Boyd, 2014, Beck, 2017, Chapter 6 & 7], $\text{prox}_{\gamma,g\circ A}(m)$ is usually not available in closed form if $AA^\top \neq \alpha I, \alpha > 0$ and more expensive to compute. Here, the symbol $^\top$ refers to (Hermitian) transpose. The proximal map for the indicator function is the projection:

$$\text{prox}_{\gamma,\iota_\mathcal{C}}(m) = \mathcal{P}_{\iota_\mathcal{C}}(m)$$

with $\mathcal{P}_{\iota_\mathcal{C}}(m)$ defined as in (4.10). The intersection of an arbitrary number of convex sets, $m \in \bigcap_{i=1}^p \mathcal{C}_i$, is also convex. We assume that all constraints are chosen consistently, such that the intersection of all selected constraint sets is nonempty:

$$\bigcap_{i=1}^{p} \mathcal{C}_i \neq \emptyset. \tag{4.13}$$

This means we define constraints such that there is at least one element in the intersection. This assumption is not restrictive in practice because apparently contradicting constraint sets often have a non-empty intersection. For example, ℓ_1-norm based total-variation constraints and smoothness promoting constraints have at least one model in their intersection: a homogeneous model has a total-variation equal to 0 and maximal smoothness.

We use $m[i]$ to indicate entries of the vector m. Subscripts like y_i refer to one of the sub-vectors that are part of $\tilde{y} = (y_1^\top \, y_2^\top, \dots, y_p^\top)^\top$.

The Euclidean inner product of two vectors is denoted as $a^\top b$, and $\|a\|_2^2 = a^\top a$.

4.3 PARSDMM: Exploiting similarity between constraint sets

As we briefly mentioned in the introduction, currently available algorithms for computing projections onto the intersection of closed and convex sets do not take similarity between sets into account. They also treat projections onto each set as a black box, which means they require another iterative algorithm (and stopping conditions) to compute projections that have no closed-form solution. In our Projection Adaptive Relaxed Simultaneous Direction Method of Multipliers (PARSDMM), we avoid nesting multiple algorithms and explicitly exploit similarities between the $i = 1, 2, \ldots, p$ linear operators $A_i \in \mathbb{R}^{M_i \times N}$. We accomplish this by writing each constraint set $\mathcal{V}_i$ in problem (4.1)) as the indicator function of a 'simple' set ($\iota_{\mathcal{C}_i}$) and a possibly non-orthogonal linear operator: $x \in \mathcal{V}_i \Leftrightarrow A_i x \in \mathcal{C}_i$. We formulate projection of $m \in \mathbb{R}^N$ onto the intersection of p sets as

$$\min_x \frac{1}{2}\|x - m\|_2^2 + \sum_{i=1}^{p} \iota_{\mathcal{C}_i}(A_i x). \tag{4.14}$$

PARSDMM is designed to solve inverse problems that call for multiple pieces of prior knowledge in the form of constraints. Each piece of prior knowledge corresponds to a single set, and we focus on intersections of two up to about 16 sets, which we found adequate to regularize inverse problems. To avoid technical issues with non-convexity, we, for now, assume all sets to be closed and convex.

We use ADMM as a starting point. ADMM is known to solve intersection projection (and feasibility) problems [Boyd et al., 2011, Pakazad et al., 2015, Bauschke and Koch, 2015, Jia et al., 2017, Tibshirani, 2017, Kundu et al., 2017]. However, it remains a black-box algorithm and struggles with projections that do not have closed-form solutions. For completeness and to highlight the differences with the algorithm we propose below, we present in Appendix C a black box algorithm for the projection onto the intersection of sets based on ADMM.

The augmented Lagrangian

To start the derivation of PARSDMM, we introduce separate vectors $y_i \in \mathbb{R}^{M_i}$ for each of the $i = 1, \ldots, p$ constraint sets of problem (4.14) and we add linear equality constraints as follows:

$$\min_{x, \{y_i\}} \frac{1}{2}\|x - m\|_2^2 + \sum_{i=1}^{p} \iota_{\mathcal{C}_i}(y_i) \quad \text{s.t.} \quad A_i x = y_i. \tag{4.15}$$

The augmented Lagrangian [e.g., Nocedal and Wright, 2000, Chapter 17] of problem (4.15) is a basis for ADMM (see (4.19) below). To ensure that the x-minimization remains quadratic (see derivation below), we make this minimization problem independent of the distance term $\frac{1}{2}\|x - m\|_2^2$. This choice has the additional benefit of allowing for other functions that measure distance from m. We remove the direct coupling of the distance term by introducing additional variables and constraints $y_{p+1} = A_{p+1}x = I_N x$. For this purpose, we define $\frac{1}{2}\|x - m\|_2^2 = f(y_{p+1})$ and create the function

$$\tilde{f}(\tilde{y}) = f(y_{p+1}) + \sum_{i=1}^{p} \iota_{\mathcal{C}_i}(y_i), \tag{4.16}$$

where we use the $\tilde{\cdot}$ symbol to indicate concatenated matrices and vectors, as well as functions that are the sum of multiple functions to simplify notation. The concatenated matrices and vectors read

$$\tilde{A} = \begin{pmatrix} A_1 \\ \vdots \\ A_{p+1} = I_N \end{pmatrix}, \quad \tilde{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_{p+1} \end{pmatrix}, \quad \tilde{v} = \begin{pmatrix} v_1 \\ \vdots \\ v_{p+1} \end{pmatrix}. \tag{4.17}$$

The vectors $v_i \in \mathbb{R}^{M_i}$ are the Lagrangian multipliers that occur in the augmented Lagrangian for the projection problem, after one more reformulation step. We always have $A_{p+1}x = I_N x = y_{p+1}$ for the Euclidean projection that uses the squared ℓ_2-distance $\frac{1}{2}\|x - m\|_2^2$. With these new definitions, problem (4.15) becomes

$$\min_{x,\tilde{y}} \tilde{f}(\tilde{y}) \quad \text{s.t.} \quad \tilde{A}x = \tilde{y}. \tag{4.18}$$

This formulation has the same form as problems that regular ADMM solves—i.e., $\min_{x,y} f(x) + g(y)$ s.t. $Ax + By = c$. It follows that we can guarantee convergence under the same conditions as for ADMM. According to [Boyd et al., 2011, Eckstein and Yao, 2015], ADMM converges when $f(x) : \mathbb{R}^{N_1} \to \mathbb{R} \cup \{+\infty\}$ and $g(y) : \mathbb{R}^{N_2} \to \mathbb{R} \cup \{+\infty\}$ are proper and convex. The linear equality constraints involve matrices $A \in \mathbb{R}^{M \times N_1}$ and

$B \in \mathbb{R}^{M \times N_2}$ and vectors $x \in \mathbb{R}^{N_1}$, $y \in \mathbb{R}^{N_2}$ and $c \in \mathbb{R}^M$.

To arrive at the main iterations of PARSDMM, we now derive an algorithm for the projection problem stated in (4.18), based on the augmented Lagrangian

$$
\begin{aligned}
&L_{\rho_1,\ldots,\rho_{p+1}}(x, y_1, \ldots, y_{p+1}, v_1, \ldots, v_{p+1}) = \\
&\sum_{i=1}^{p+1} \Big[f_i(y_i) + v_i^\top (y_i - A_i x) + \frac{\rho_i}{2} \| y_i - A_i x \|_2^2 \Big].
\end{aligned} \tag{4.19}
$$

As we can see, this expression has a separable structure with respect to the Lagrangian multipliers v_i, and the auxiliary vectors y_i. Following the ADMM variants for multiple functions, as formulated by [Song et al., 2016, Kitic et al., 2016, Xu et al., 2017c], we use a different penalty parameter $\rho_i > 0$ for each index i. In this way, we make sure all linear equality constraints $A_i x = y_i$ are satisfied sufficiently by running a limited number of iterations. Because the different matrices A_i may have widely varying scalings and sizes, a fixed penalty for all i could cause slow convergence of x towards one of the constraint sets. Additionally, to further accelerate the algorithm we also introduce a different relaxation parameter (γ_i) for each index i. After we derive the main steps of our proposed algorithm, we describe the automatic selection of the scalar parameters.

The iterations

With the above definitions, iteration counter k, and inclusion of relaxation parameters, which we assume to be limited to the interval $\gamma_i \in [1, 2)$ [see Xu et al., 2017b], the iterations can be written as

$$
\begin{aligned}
x^{k+1} &= \arg\min_x \sum_{i=1}^{p+1} \Big(\frac{\rho_i^k}{2} \| y_i^k - A_i x + \frac{v_i^k}{\rho_i^k} \|_2^2 \Big) \\
\bar{x}_i^{k+1} &= \gamma_i^k A_i x_i^{k+1} + (1 - \gamma_i^k) y_i^k \\
y_i^{k+1} &= \arg\min_{y_i} \Big[f_i(y_i) + \frac{\rho_i^k}{2} \| y_i^k - \bar{x}_i^{k+1} + \frac{v_i^k}{\rho_i^k} \|_2^2 \Big] \\
v_i^{k+1} &= v_i^k + \rho_i^k (y_i^{k+1} - \bar{x}_i^{k+1}).
\end{aligned}
$$

To arrive at our final algorithm, we rewrite these iterations in a more explicit form as

$$
\begin{aligned}
x^{k+1} &= \Big[\sum_{i=1}^{p} (\rho_i^k A_i^\top A_i) + \rho_{p+1}^k I_N \Big]^{-1} \sum_{i=1}^{p+1} \Big[A_i^\top (\rho_i^k y_i^k + v_i^k) \Big] \\
\bar{x}_i^{k+1} &= \gamma_i^k A_i x_i^{k+1} + (1 - \gamma_i^k) y_i^k \\
y_i^{k+1} &= \operatorname{prox}_{f_i, \rho_i^k} (\bar{x}_i^{k+1} - \frac{v_i^k}{\rho_i^k}) \\
v_i^{k+1} &= v_i^k + \rho_i^k (y_i^{k+1} - \bar{x}_i^{k+1}).
\end{aligned}
$$

In this expression, we used the fact that A_{p+1} is always the identity matrix of size N for projection problems. Without over/under relaxation [$\bar{x}_i^{k+1}$ computation, Eckstein and Bertsekas, 1992, Iutzeler and Hendrickx, 2017, Xu et al., 2017b], these iterations are known as SALSA [Afonso et al., 2011] or the simultaneous direction method of multipliers [SDMM, Combettes and Pesquet, 2011, Kitic et al., 2016]. The derivation in this section shows that ADMM/SDMM solve the projection onto an intersection of multiple closed and convex sets. However, the basic iterations from (4.20) are not yet a practical and fast algorithm, because there are scalar parameters that need to be selected, no stopping conditions, and no specializations to constraints typically found in the imaging sciences. Therefore, we add automatic scalar parameter selection to the iterations (4.20), as well as linear system solves, stopping conditions, and multilevel acceleration specialized to computing projections onto intersections of many sets.

Computing the proximal maps

The proximal maps in the iterations (4.20) become projections onto simple sets (e.g., bounds/ℓ_1 and ℓ_2 norm-ball/cardinality/rank), which permit closed-form solutions that do not depend on the ρ_i. When $f_{p+1}(w) = 1/2\|w - m\|_2^2$, (squared ℓ_2 distance of w to the reference vector m) the proximal map is also available in closed form:

$$\begin{aligned}\operatorname{prox}_{f_{p+1},\rho_{p+1}}(w) &= \arg\min_z 1/2\|z-m\|_2^2 + \rho_{p+1}/2\|z-w\|_2^2 \\ &= (m+\rho_{p+1}w)/(1+\rho_{p+1}).\end{aligned} \tag{4.20}$$

We thus avoided sub-problems for projections that require other convex optimization algorithms for their solutions.

Solving the linear system and automatic parameter selection

We can also see from (4.20) that the computation of x^{k+1} involves the solution of a single problem where all linear operators are summed into one system of normal equations. The system matrix equals

$$C \equiv \sum_{i=1}^{p+1}(\rho_i A_i^\top A_i) = \sum_{i=1}^{p}(\rho_i A_i^\top A_i) + \rho_{p+1} I_N \tag{4.21}$$

and is by construction always positive-definite because $\rho_i > 0$ for all i. The minimization over x is therefore uniquely defined. As suggested by Xu et al. [2017a], we adapt the ρ_i's every two iterations using the scheme we discuss below.

While we could use direct matrix factorizations of C, we would need to refactorize every time we update any of the ρ_i's. This would make computing x^{k+1} too costly. Instead, we rely on warm-started iterative solvers with x^k used as the initial guess for x^{k+1}. There exist several alternatives including LSQR [Paige and Saunders, 1982] to solve the above linear system (x^{k+1} computation in 4.20) iteratively. We choose the conjugate-gradient (CG) method on the normal equations for the following reasons:

1. Contrary to LSQR, transforms that satisfy $A_i^\top A_i = \alpha I_N$ are free for CG because we explicitly form the sparse system matrix C, which already includes the identity matrix.

2. By limiting the relative difference between the ρ_i and ρ_{p+1}, where the latter corresponds to the identity matrix in (4.21), we ensure C

is sufficiently well conditioned so squaring the condition number does not become a problem.

3. For many transforms, the matrices $A_i^\top A_i$ are sparse and have at least partially overlapping sparsity patterns (discrete derivative matrices for one or more directions, orthogonal transforms). Multiplication with $\sum_{i=1}^{p+1}(\rho_i A_i^\top A_i)$ is therefore not much more expensive than multiplication with a single $A_i^\top A_i$. However, LSQR requires matrix-vector products with all A_i and $A_i^\top$ at every iteration.

4. Full reassembly of C at iteration k is not required. Every time we update any of the ρ_i's, we update C by subtracting and adding the block corresponding to the updated ρ_i. If the index that changed is indicated by $i = u$, the system matrix for the next x^{k+1} computation becomes

$$\begin{aligned} C^{k+1} = \sum_{i=1}^{p+1}(\rho_i^{k+1} A_i^\top A_i) &= \sum_{i=1}^{p+1}(\rho_i^k A_i^\top A_i) - (\rho_u^k A_u^\top A_u) + (\rho_u^{k+1} A_u^\top A_u) \\ &= C^k + A_u^\top A_u(\rho_u^{k+1} - \rho_u^k). \end{aligned} \tag{4.22}$$

For each ρ_i update, forming the new system matrix involves a single addition of two sparse matrices (assuming all $A_i^\top A_i$'s are pre-computed).

To further save computation time, we solve the minimization with respect to x inexactly. We select the stopping criterion for CG adaptively in terms of the relative residual of the normal equations—i.e., we stop CG if the relative residual drops below

$$0.1\| \Big[\sum_{i=1}^{p}(\rho_i^k A_i^\top A_i) + \rho_{p+1}^k I_N \Big] x - \sum_{i=1}^{p+1} \Big[A_i^\top(\rho_i^k y_i^k + v_i^k) \Big] \|_2 / \| \sum_{i=1}^{p+1} \Big[A_i^\top (\rho_i^k y_i^k + v_i^k) \Big] \|_2. \tag{4.23}$$

Empirically, we found that a reduction of the relative residual by a factor of ten represents a robust choice that also results in time savings for solving problem (4.18) compared to a fixed and accurate stopping criterion for the x-minimization step. The stopping criterion for CG is relatively inexact during the first few iterations from (4.20) and requests more accurate solutions later on, such that the conditions on inexact sub-problem solutions from [Eckstein and Bertsekas, 1992] will be satisfied eventually.

Just like standard ADMM, we may also require a large number of iterations (4.20) for a fixed penalty parameter ρ_i for all i [e.g., Nishihara et al., 2015, Xu et al., 2017a]. It is better to update ρ_i^k and γ_i^k every couple of iterations to ensure we reach a good solution in a relatively small number of iterations. For this purpose, we use Xu et al. [2017a]'s automatic selection of ρ_i^k and γ_i^k for ADMM. Numerical experiments by Xu et al. [2016] show that these updates also perform well on various non-convex problems. The updates themselves are based on a Barzilai-Borwein spectral step size [Barzilai and Borwein, 1988] for Douglas-Rachford (DR) splitting applied to the dual of $\min_{x,y} f(x) + g(y)$ s.t. $Ax + By = c$ and derive from equivalence between ADMM and DR on the dual [Eckstein and Bertsekas, 1992, Esser, 2009].

Exploiting parallelism

Given the grid size of 3D PDE-based parameter estimation problems, performance is essential. For this reason, we seek a parallel implementation that exploits multi-threading offered by modern programming languages such as Julia [Bezanson et al., 2017]. Since the computational time for the x-minimization using the conjugate-gradient algorithm is dominated by the matrix-vector products (MVP) with C, we concentrate our efforts there by using compressed diagonal storage (CDS), see, e.g., [Saad, 1989, Sern et al., 1990, Kotakemori et al., 2008]. This format stores the non-zero bands of the matrix as a dense matrix, and we compute MVPs directly in this storage sytem. These MVPs are faster than the more general Compressed Sparse Column (CSC) format. CDS has the additional benifit that it can efficiently handle matrices generated by spatially varying (blurring, derivative) kernels. We can use CDS if all matrices $A_i^\top A_i$ have a banded sparsity-pattern. Using Julia's multi-threading, we compute the MVPs with C in parallel. In cases where the $A_i^\top A_i$'s do not have a banded structure we revert to computations in the standard Compressed Sparse Column (CSC) format.

Aside from matrix-vector products during the inner iterations, most calculation time in (4.20) is used for $\bar{x}_i^{k+1}$, y_i^{k+1}, v_i^{k+1}, ρ_i^{k+1}, and γ_i^{k+1}. To reduce these costs, we compute these quantities in parallel. This is relatively straightforward to do because each problem is independent so that the operations for the p constraints can be carried out by different Julia workers where each worker either uses Julia threads, multi-threaded BLAS [OpenBLAS, Wang et al., 2013], or multi-threaded Fourier-transforms [FFTW library, Frigo and Johnson, 2005].

Stopping conditions

So far, we focussed on reducing the time for each iteration of (4.20). However, the total computational time depends on the total number of iterations and therefore on the stopping conditions. For our problems, a good stopping criterion guarantees solutions that are close to all constraint sets, and at a minimal distance from the point we want to project. When working with a single constraint set, stopping criteria based on a combination of the primal $r^{\text{pri}} = \|\tilde{y} - \tilde{A}x^k\|$ and dual residual $r^{\text{dual}} = \|\tilde{\rho}\tilde{A}^\top(\tilde{y}^k - \tilde{y}^{k-1})\|$ are adequate as long as both become sufficiently small [e.g., Boyd et al., 2011, Kitic et al., 2016, Xu et al., 2017a]. However, the situation is more complicated in situations where we work with multiple constraint sets. In that case, the $\tilde{y}$ and $\tilde{A}$ contain a variety of vectors and linear operators that correspond to the different constraint sets. Since these operators are scaled differently and have different dimensions, it becomes more difficult to determine the relationship between the size of the residuals and the accuracy of the solution. In other words, it becomes challenging to decide at what primal and dual residual to stop such that we are close to all constraint sets.

Instead of considering residuals, it may be more intuitive to look at feasibilities by dropping the quadratic part of the projection problem (4.15). This means that we only insist that the final solution needs to be an element of every set $\mathcal{V}_i$ when considering our stopping criterion. This holds if x is in the intersection of the constraint sets but requires projections onto each $\mathcal{V}_i$

to verify, a situation we want to avoid in PARSDMM. Instead, we rely on the transform-domain set feasibility error

$$r_i^{\text{feas}} = \frac{\|A_i x - \mathcal{P}_{\mathcal{C}_i}(A_i x)\|}{\|A_i x\|}, \; i = 1 \cdots p, \tag{4.24}$$

to which we have access at a relatively low cost since we already computed $A_i x$ in the iterations from (4.20). Our first stopping criterion thus corresponds to a normalized version of the objective when solving convex multiple set split-feasibility problems [Censor et al., 2005]. We added this normalization in (4.24) to account for different scalings and sizes of the linear operators A_i.

The projections onto the constraint sets $\mathcal{P}_{\mathcal{C}_i}(\cdot)$ themselves, are relatively cheap to compute since they only include projections onto sets such as norm-balls, bounds, cardinality sets. By testing for transform-domain feasibility every few iterations only (5 or 10 typically), we further reduce the computational costs for our stopping condition.

Satisfying constraints alone for $i = 1 \cdots p$ does not indicate whether or not x^k is close to the projection onto the intersection of the p different constraint sets or whether it is just a feasible point, possibly 'deep' inside the intersection. If x^k is indeed the result of the projection of m, then $\|x^k - x^{k-1}\|$ approaches a stationary point, assuming that x^k converges to the projection. We make this property explicit by considering the maximum relative change x^k over the s previous iterations: $j \in S \equiv \{1, 2, \dots, s\}$. The relative evolution of x at the kth iteration thus becomes

$$r^{\text{evol}} = \frac{\max_{j \in S}\{\|x^k - x^{k-j}\|\}}{\|x^k\|}. \tag{4.25}$$

By considering the history (we use $s = 5$ in our numerical examples), our stopping criterion becomes more robust to oscillations in $\|x^k - x^{k-1}\|$ as a function of k. So we propose to stop PARSDMM if

$$r^{\text{evol}} < \varepsilon^{\text{evol}} \quad \text{and} \quad r_i^{\text{feas}} < \varepsilon_i^{\text{feas}} \quad \forall\, i. \tag{4.26}$$

During our numerical experiments, we select $\varepsilon^{\text{evol}} = 10^{-2}$ and $\varepsilon_i^{\text{feas}} = 10^{-3}$, which balance sufficiently accurate solutions and short solution times. These are still two constants to be chosen by the user, but we argue that r_i^{feas} may relate better to our intuition on feasibility because it behaves like a distance to each set separately. The evolution term $\|x^k - x^{k-1}\|$ is found in many optimization algorithms and is especially informative for physical parameter estimation problems where practitioners often have a good intuition to which $\|x^k - x^{k-1}\|$ the physical forward model $f(x)$ is sensitive.

The PARSDMM algorithm

We summarize our discussions from the previous sections in the following Algorithms.

4.3.1 Multilevel PARSDMM

Inverse problems with data-misfit objectives that include PDE forward models typically need a fine grid for stable physical simulations. At the same time, we often use constraints to estimate ‘simple’ models—i.e. models that are smooth, have a low-rank, are sparse in some transform-domain, and that may not need many grid points for accurate representations of the image/model. This suggests we can reduce the total computational time of PARSDMM (Algorithm 3) by using a multilevel continuation strategy. The multilevel idea presented in this section applies to the projection onto the intersection of constraint sets only and not to the grids for solving PDEs. Our approach proceeds as follows: we start at a coarse grid and continue towards finer grids. While inspired by multigrid methods for solving linear systems, the proposed multilevel algorithm does not cycle between coarse and fine grids. By using the solution at the coarse grid as the initial guess for the solution on the finer grid, the convergence guarantees are the same as for the single level version of our algorithm. As long as the computationally cheap coarse grid solutions are ‘good’ initial guesses for the finer grids, this multilevel approach, which is similar to multilevel ADMM by Macdonald and Ruthotto [2018], can lead to substantial reductions in computational

Algorithm 3 Projection Adaptive Relaxed Simultaneous Direction Method of Multipliers (PARSDMM) to compute the projection onto an intersection, including automatic selection of the penalty parameters and relaxation.

```
Algorithm PARSDMM
inputs:
  $m$ //point to project
  $A_1, A_2, \ldots, A_p, A_{p+1} = I_N$ //linear operators
  //norm/bound/cardinality/... projectors:
  $\text{prox}_{f_i,\rho_i}(w) = \mathcal{P}_{\mathcal{C}_i}(w)$ for $i = 1, 2, \ldots, p$
  //prox for the squared distance from $m$ :
  $\text{prox}_{f_i,\rho_{p+1}}(w) = (m + \rho_{p+1} w)/(1 + \rho_i)$
  select $\rho_i^0$, $\gamma_i^0$, update-freqency
  optional: initial guess for $x$, $y_i$ and $v_i$
initialize:
  $B_i = A_i^\top A_i$ //pre-compute for all $i$
  $C = \sum_{i=1}^{p+1} (\rho_i B_i)$ //pre-compute
  $k = 1$
WHILE not converged
    $x^{k+1} = C^{-1} \sum_{i=1}^{p+1} \Big[ A_i^\top (\rho_i^k y_i^k + v_i^k) \Big]$ //CG, stop when (4.23) holds
    FOR $i = 1, 2, \ldots, p+1$ //compute in parallel
        $s_i^{k+1} = A_i x^{k+1}$
        $\bar{x}_i^{k+1} = \gamma_i^k s_i^{k+1} + (1 - \gamma_i^k) y_i^k$
        $y_i^{k+1} = \text{prox}_{f_i,\rho_i}(\bar{x}_i^{k+1} - \frac{v_i^k}{\rho_i^k})$
        $v_i^{k+1} = v_i^k + \rho_i^k (y_i^{k+1} - \bar{x}_i^{k+1})$
        stop if conditions (4.26) hold
        If mod$(k,$ update-freqency$) = 1$
           $\{\rho_i^{k+1}, \gamma_i^{k+1}\}$ = adapt-rho-gamma$(v_i^k, v_i^{k+1}, y_i^{k+1}, s_i^{k+1}, \rho_i^k)$
       End if
    END
    FOR $i = 1, 2, \ldots, p+1$ //update $C$ if necessary
       If $\rho_i^{k+1} \neq \rho_i^k$
          $C \leftarrow C + B_i(\rho_i^{k+1} - \rho_i^k)$
       End if
    END
    $k \leftarrow k + 1$
END
output: $x$
```

Algorithm 4 Adapt ρ and γ according to [Xu et al., 2017b] with some modifications to save computational work. The constant $\varepsilon^{\text{corr}}$ is in the range $[0.1 - 0.4]$ as suggested by [Xu et al., 2017b]. Quantities from the previous call to adapt-rho-gamma have the indication k_0. Actual implementation computes and re-uses some of the inner products and norms.

```
Algorithm adapt-rho-gamma
input: $v_i^k, v_i^{k+1}, y_i^{k+1}, s_i^{k+1}, \rho_i^k$
$\varepsilon^{\text{corr}} = 0.3$
$\hat{v}^{k+1} = v_i^k + \rho_i^k(y_i^k - s_i^{k+1})$
$\Delta\hat{v} = \hat{v}_i^{k+1} - \hat{v}^{k_0}$
$\Delta v = v_i^{k+1} - v^{k_0}$
$\Delta\hat{h} = s_i^{k+1} - s^{k_0})$
$\Delta\hat{g} = -(y_i^{k+1} - y^{k_0})$
$\alpha^{\text{corr}} = \frac{\Delta\hat{h}^\top \Delta\hat{v}}{\|\Delta\hat{h}\|\|\Delta\hat{v}\|}$
$\beta^{\text{corr}} = \frac{\Delta\hat{g}^\top \Delta v}{\|\Delta\hat{g}\|\|\Delta v\|}$
  If $\alpha^{\text{corr}} > \varepsilon^{\text{corr}}$
      $\hat{\alpha}^{\text{MG}} = \frac{\Delta\hat{h}^\top \Delta\hat{v}}{\Delta\hat{h}^\top \Delta\hat{h}}, \hat{\alpha}^{\text{SD}} = \frac{\Delta\hat{v}^\top \Delta\hat{v}}{\Delta\hat{h}^\top \Delta\hat{v}}, \hat{\alpha} = \begin{cases} \hat{\alpha}^{\text{MG}} & \text{if } 2\hat{\alpha}^{\text{MG}} > \hat{\alpha}^{\text{SD}} \\ \hat{\alpha}^{\text{SD}} - 0.5\hat{\alpha}^{\text{MG}} & \text{if else} \end{cases}$
  End
  If $\beta^{\text{corr}} > \varepsilon^{\text{corr}}$
      $\hat{\beta}^{\text{MG}} = \frac{\Delta\hat{g}^\top \Delta v}{\Delta\hat{g}^\top \Delta\hat{g}}, \hat{\beta}^{\text{SD}} = \frac{\Delta v^\top \Delta v}{\Delta\hat{g}^\top \Delta v}, \hat{\beta} = \begin{cases} \hat{\beta}^{\text{MG}} & \text{if } 2\hat{\beta}^{\text{MG}} > \hat{\beta}^{\text{SD}} \\ \hat{\beta}^{\text{SD}} - 0.5\hat{\beta}^{\text{MG}} & \text{if else} \end{cases}$
  End
$\{\rho^{k+1}, \gamma^{k+1}\} = \begin{cases} \{\sqrt{\hat{\alpha}\hat{\beta}}, 1 + \frac{2\sqrt{\hat{\alpha}\hat{\beta}}}{\hat{\alpha}+\hat{\beta}}\} & \text{if } \alpha^{\text{corr}} > \varepsilon^{\text{corr}} \,\&\, \beta^{\text{corr}} > \varepsilon^{\text{corr}} \\ \{\hat{\alpha}, 1.9\} & \text{if } \alpha^{\text{corr}} > \varepsilon^{\text{corr}} \,\&\, \beta^{\text{corr}} \leq \varepsilon^{\text{corr}} \\ \{\hat{\beta}, 1.1\} & \text{if } \alpha^{\text{corr}} \leq \varepsilon^{\text{corr}} \,\&\, \beta^{\text{corr}} > \varepsilon^{\text{corr}} \\ \{\rho^k, 1.5\} & \text{if } \alpha^{\text{corr}} \leq \varepsilon^{\text{corr}} \,\&\, \beta^{\text{corr}} \leq \varepsilon^{\text{corr}} \end{cases}$
set and save for next call to adapt-rho-gamma:
$\hat{v}^{k_0} \leftarrow \hat{v}_i^{k+1}$, $v^{k_0} \leftarrow v_i^{k+1}$,
$s^{k_0} \leftarrow s_i^{k+1}$, $y^{k_0} \leftarrow y_i^{k+1}$
save $v_i^{k+1}, y_i^{k+1}$ for next call to adapt-rho-gamma
output: $\rho_i^{k+1}$, $\gamma_i^{k+1}$
```

cost as we will demonstrate in the numerical example of the next section.

To arrive at a workable multilevel implementation for Algorithm 3, we need to concern ourselves with the initialization of ADMM-type iterations and initial guesses for x and y_i, v_i for all $i \in \{1, \dots, p, p+1\}$. After initialization of the coarsest grid with all zero vectors, we move to a finer grid by interpolating x and all y_i, v_i. Since the solution estimate $x \in \mathbb{R}^N$ always refers to an image or a tensor, we are free to reshape and interpolate it to a finer grid. The situation for vectors v_i and y_i is a bit more complicated, as their dimensions depend on the corresponding A_i. To handle these, we do a relatively simple interpolation.

Example. When A_i is a discrete derivative matrix, then the vectors v_i and y_i live on a grid that we know at every level of the multilevel scheme. If we have $A_i = D_z \otimes I_x$, where D_z is the first-order finite-difference matrix as in (4.8), we know that $A_i \in \mathbb{R}^{((n_z-1)n_x)\times(n_z \times n_x)}$ and therefore $v_i \in \mathbb{R}^{(n_z-1)n_x}$ and $y_i \in \mathbb{R}^{(n_z-1)n_x}$. We can thus reshape the associated vectors v_i and y_i as an image (in 2D) of size $(n_z - 1 \times n_x)$ and interpolate it to the finer grid for the next level, working from coarse to fine. In 3D, we follow the same approach. We also need a coarse version of m at each level: m_l for $l = n_{\text{levels}}, n_{\text{levels}} - 1, \dots, 1$. We simply obtain the coarse models by applying an anti-alias filter and subsampling the original m. In principle, any subsampling and interpolation technique may be used in this multilevel framework. Our numerical experiments interpolate to finer grids using the simple nearest-neighbor method. Numerical experiments with other types of interpolations did not show a reduction of the number of PARSDMM iterations at the finest grid.

We decide the number of levels (n_{levels}) and the coarsening factor ahead of time. Together with the original grid, these determine the grid at all levels so we can set up the linear operators and proximal mappings at each level. This set-up phase is a one time cost since its result is reused every time we project a model m onto the intersection of constraint sets. The additional computational costs of the multilevel scheme are the interpolation of x and all y_i, v_i to a finer grid, but this happens only once per level and not every ML-PARSDMM (Algorithm 5) iteration. So the computational overhead we incur from the interpolations is small compared to the speedup of Algorithm 5.

Algorithm 5 Multilevel PARSDMM to compute the projection onto an intersection using a multilevel strategy.

```
inputs:
  $n_{\text{levels}}$ //number of levels
  $l = \{n_{\text{levels}}, n_{\text{levels}} - 1, 1\}$
  $\text{grid}_l$  //grid info at each level $l$
  $m_l$      //model to project at every level $l$
  $A_{1,l}, A_{2,l}, \ldots, A_{p+1,l}$ //linear operators at every level
  // norm/bound/cardinality/... projectors at each level:
  $\text{prox}_{f_{i,l},\rho_i}(w) = \mathcal{P}_{\mathcal{C}_i}(w)$ for $i = 1, 2, \ldots, p$
  // proximal map for the squared distance from $m$ at each level:
  $\text{prox}_{f_{p+1,l},\rho_{p+1}}(w) = (m_l + \rho_{p+1} w)/(1 + \rho_{p+1})$
//start at coarsest grid
FOR $l = n_{\text{levels}}, n_{\text{levels}} - 1, \ldots, 1$
    //solve on current grid:
    $(x_l, \{y_{i,l}\}, \{v_{i,l}\}) = \text{PARSDMM}(m_l, \{A_{i,l}\}, \{\text{prox}_{f_{i,l},\rho_i}\}, x_l, \{y_{i,l}\}, \{v_{i,l}\})$
    $x_l \rightarrow x_{l-1}$ //interpolate to finer grid
    FOR $i = 1, 2, \ldots, p + 1$
        $y_{i,l} \rightarrow y_{i,l-1}$ //interpolate to finer grid
        $v_{i,l} \rightarrow v_{i,l-1}$ //interpolate to finer grid
    END
END
output: $x$ at original grid (level 1)
```

4.4 Software and numerical examples

The software corresponding to this paper is available at https://github.com/slimgroup. The main design principles of our code implementing the PARSDMM algorithm include *(i)* performance, it needs to scale to imposing multiple constraints on 3D models up to at least 300^3 grid points; *(ii)* specialization to the specific and fixed problem structure (4.14); and *(iii)* flexibility to work with multiple linear operators and projectors. Because of these design choices, the user only needs to provide the model to project, m, and pairs of linear operators and projectors onto simple sets:

$\{(A_1, \mathcal{P}_{\mathcal{C}_1}), (A_2, \mathcal{P}_{\mathcal{C}_2}), \ldots, (A_p, \mathcal{P}_{\mathcal{C}_p})\}$. The software adds the identity matrix and the proximal map for the distance squared from m. These are all computational components required to solve intersection projection problems as formulated in (4.16).

To reap benefits from modern programming language design, including just-in-time compilation, multiple dispatch, and mixing distributed and multi-threaded computations, we wrote our software package in Julia 0.6. Our code uses parametric typing, which means that the same scripts can run in `Float32` (single) and `Float64` (double) precision. As expected, most components of our software run faster with `Float32` with reduced memory consumption. The timings in the following examples use `Float32`.

We provide scripts that the set up the linear operators and projectors for regular grids in 2D and 3D. It is not necessary to use these scripts as the solver is agnostic to the specific construction of the projectors or linear operators. Table (4.1) displays the constraints we currently support. For example, when the user requests the script to set up minimum and maximum bounds on the discrete gradient in the z-direction of the model, the script returns the discrete derivative matrix $A = I_x \otimes D_z$ and a function $\mathcal{P}_{\text{bounds}}(\cdot)$ that projects the input onto the bounds. The software currently supports the identity matrix, matrices representing the discrete gradient and the operators that we apply matrix-free: the discrete cosine/Fourier/wavelet/curvelet [Ying et al., 2005] transforms.

For the special case of orthogonal linear operators, we leave the linear operator inside the set definition because we know the projection onto $\mathcal{V}$ in closed form. For example, if $\mathcal{V} = \{x \mid \|Ax\|_1 \leq \sigma\}$ with discrete Fourier transform (DFT) matrix $A \in \mathbb{C}^{N \times N}$, the projection is known in closed form as $\mathcal{P}_{\mathcal{V}}(x) = A^* \mathcal{P}_{\|\cdot\| \leq \sigma}(Ax)$, where * denotes the complex-conjugate transpose and $\mathcal{P}_{\|\cdot\| \leq \sigma}$ is the projection onto the ℓ_1-ball. We do this to keep all other computations in PARSDMM (Algorithm 3) real, because complex-valued vectors require more storage and will slow down most computations.

As an example of our code, we show how to project a 2D model m onto the intersection of bound constraints and the set of models that have monotonically increasing parameter values in the z-direction.

descriptions	set
bounds	$\{m \mid l[i] \leq m[i] \leq u[i]\}$
transform-domain bounds	$\{m \mid l[i] \leq (Am)[i] \leq b[i]\}$
transform-domain ℓ_1	$\{m \mid \|Am\|_1 \leq \sigma\}$
transform-domain ℓ_2	$\{m \mid \|Am\|_2 \leq \sigma\}$
transform-domain annulus	$\{m \mid \sigma_l \leq \|Am\|_2 \leq \sigma_u\}$
transform-domain nuclear norm	$\{m \mid \sum_{j=1}^{k} \lambda[j] \leq \sigma\}$, $Am = \text{vec}(\sum_{j=1}^{k} \lambda[j] u_j v_j^\top)$ is the SVD.
transform-domain cardinality	$\{m \mid \text{card}(Am) \leq k\}$, k is a positive integer
transform-domain rank	$\{m \mid Am = \text{vec}(\sum_{j=1}^{r} \lambda[j] u_j v_j^\top)\}$, $r < \min(n_z, n_x)$
subspace constraints	$\{m \mid m = Ac, \;\; c \in \mathbb{C}^M\}$

Table 4.1: Overview of constraint sets that the software currently supports. A new constraint requires the projector onto the set (without linear operator) and a linear operator or equivalent matrix-vector product together with its adjoint. Vector entries are indexed as $m[i]$.

```
using SetIntersectionProjection

#the following optional lines of
#code set up linear operators and projectors

#grid information ( (dz,dx),(nz,nx) )
comp_grid = compgrid( (25.0, 6.0), (341, 400) )

#initialize constraint information
constraint = Vector{SetIntersectionProjection.set_definitions}()

#set up bound constraints
m_min        = 1500.0          #minimum velocity
m_max        = 4500.0          #maximum velocity
set_type     = "bounds"        #bound constraint set
TD_OP        = "identity"      #identity in the set definition
```

```
app_mode     = ("matrix","")    #bounds applied to the model as a matrix
custom_TD_OP = ([],false)       #no custom linear operators

push!(constraint, set_definitions(set_type,TD_OP,m_min,m_max,app_mode,custom_TD_OP))

# #bounds on parameters in a transform-domain (vertical slope constraint)
m_min        = 0.0
m_max        = 1e6
set_type     = "bounds"
TD_OP        = "D_z"            #discrete derivative in z-direction
app_mode     = ("matrix","")
custom_TD_OP = ([],false)

push!(constraint, set_definitions(set_type,TD_OP,m_min,m_max,app_mode,custom_TD_OP))

options = PARSDMM_options() #get default options

#get projectors onto simple sets, linear operators, set information
(P_sub,TD_OP,set_Prop) = setup_constraints(constraint,comp_grid,Float32)

#precompute and distribute quantities once, reuse later
(TD_OP,B) = PARSDMM_precompute_distribute(TD_OP,set_Prop,comp_grid,options)

#project onto intersection
(x,log_PARSDMM) = PARSDMM(m,B,TD_OP,set_Prop,P_sub,comp_grid,options)
```

Our software also allows for simultaneous use of constraints that apply to the 2D/3D model and constraints that apply to each column or row separately, except for sets based on the singular value decomposition. The linear operator remains the same if we define constraints for all rows, columns, or both. The difference is that the projection onto a simple set is now applied to each row/column independently in parallel via a multi-threaded loop.

4.4.1 Parallel Dykstra versus PARSDMM

One of our main goals was to create an algorithm that computes projections onto an intersection of sets that contains fewer manual tuning parameters, stopping conditions, and that is also faster than black-box type projection algorithms, such as parallel Dykstra's algorithm (see Appendix C). To see how the proposed PARSDMM algorithm compares to parallel Dykstra's algorithm, we need to set up a fair experimental setting that includes the subproblem solver in parallel Dykstra's algorithm. Fortunately, if we use Adaptive Relaxed ADMM (ARADMM) [Xu et al., 2017b] for the projection subproblems of parallel Dykstra's algorithm, both PARSDMM (Algorithm 3) and Parallel Dykstra-ARADMM have the same computational components. ARADMM also uses the same update scheme for the augmented Lagrangian penalty and relaxation parameters as we use in PARSDMM. This similarity allows for a comparison of the convergence as a function of the basic computational components. We manually tuned ARADMM stopping conditions to achieve the best performance for parallel Dykstra's algorithm overall.

The numerical experiment is the projection of a 2D geological model (341×400 pixels) onto the intersection of three constraint sets that are of interest to the seismic imaging examples by [Esser et al., 2016, Yong et al., 2018], and in Chapter 3:

1. $\{m \mid \sigma_1 \leq m[i] \leq \sigma_2\}$: bound constraints
2. $\{m \mid \|Am\|_1 \leq \sigma\}$ with $A = [(I_x \otimes D_z)^\top \;\; (D_x \otimes I_z)^\top]^\top$: anisotropic total-variation constraints
3. $\{m \mid 0 \leq ((I_x \otimes D_z)m)[i] \leq \infty\}$: vertical monotonicity constraints

For these sets, the primary computational components are *(i)* matrix-vector products in the conjugate-gradient algorithm. The system matrix has the same sparsity pattern as $A^\top A$, because the sparsity patterns of the linear operators in set number one and three overlap with the pattern of $A^\top A$. Parallel Dykstra uses matrix-vector products with $A^\top A$, $(D_x \otimes I_z)^\top (D_x \otimes I_z)$, and I in parallel. *(ii)* projections onto the box constraint set and the ℓ_1-ball. Both parallel Dykstra's algorithm and PARSDMM compute these in parallel. *(iii)* parallel communication that sends a vector from one to all parallel processes (x^{k+1} in Algorithm 3), and one map-reduce parallel sum that gathers the sum of vectors on all workers (the right-hand side for the x^{k+1} computation in Algorithm 3). The communication is the same for PARSDMM and parallel Dykstra's algorithm so we ignore it in the experiments below.

Before we discuss the numerical results, we discuss some details on how we count the computational operations mentioned above:

- Matrix-vector products in CG: At each PARSDMM iteration, we solve a single linear system with the conjugate-gradient method. Parallel Dykstra's algorithm simultaneously computes three projections by running three instances of ARADMM in parallel. The projections onto sets two and three solve a linear system at every ARADMM iteration. For each parallel Dykstra iteration, we count the total number of sequential CG iterations, which is determined by the maximum number of CG iterations for either set number two or three.

- ℓ_1-ball projections: PARSDMM projects onto the ℓ_1 ball once per iteration. Parallel Dykstra projects (number of parallel Dykstra iterations) × (number of ARADMM iterations for set number two) times onto the ℓ_1 ball. Because ℓ_1-ball projections are computationally more intensive (we use the algorithm from Duchi et al. [2008]) compared to projections onto the box (element-wise comparison) and also less suitable for multi-threaded parallelization, we focus on the ℓ_1-ball projections.

The results in Figure 4.1 show that PARSDMM requires much fewer CG iterations and ℓ_1-ball projections to achieve the same relative set feasibility error in the transform-domain as defined in equation (4.24). In contrast to the curves corresponding to parallel Dykstra's algorithm, we see that PARSDMM converges in an oscillatory fashion, which is caused by changing the relaxation and augmented-Lagrangian penalty parameters.

Because non-convex sets are an important application for us, we compare the performance for a non-convex intersection as well:

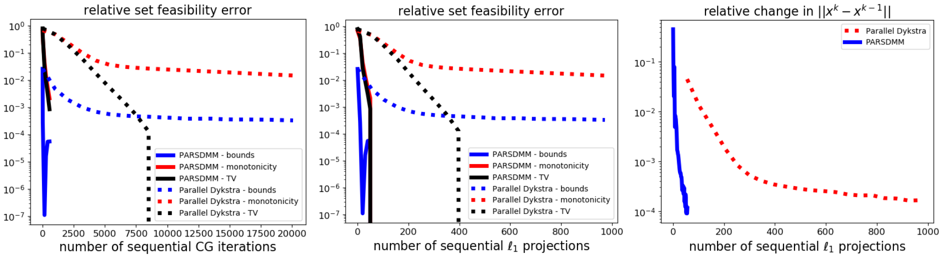


Figure 4.1: Relative transform-domain set feasibility (equation 4.24) as a function of the number of conjugate-gradient iterations and projections onto the ℓ_1 ball. This figure also shows relative change per iteration in the solution x.

1. $\{m \mid \sigma_1 \leq m[i] \leq \sigma_2\}$: bound constraints
2. $\{m | (I_x \otimes D_z)m = \text{vec}(\sum_{j=1}^{r} \lambda[j] u_j v_j^*)\}$, where $r < \min(n_z, n_x)$, $\lambda[j]$ are the singular values, and u_j, v_j are singular vectors: rank constraints on the vertical gradient of the image

We count the computational operations in the same way as in the previous example, but this time the computationally most costly projection is the projection onto the set of matrices with limited rank via the singular value decomposition. The results in Figure 4.2 show that the convergence of parallel Dykstra's algorithm almost stalls: the solution estimate gets closer to satisfying the bound constraints, but there is hardly any progress towards the rank constraint set. PARSDMM does not seem to suffer from non-convexity in this particular example.

We used the single-level version of PARSDMM such that we can compare the computational cost with Parallel Dykstra. The PARSDMM results in this section are therefore pessimistic in general, as the multilevel version can offer additional speedups, which we show next.

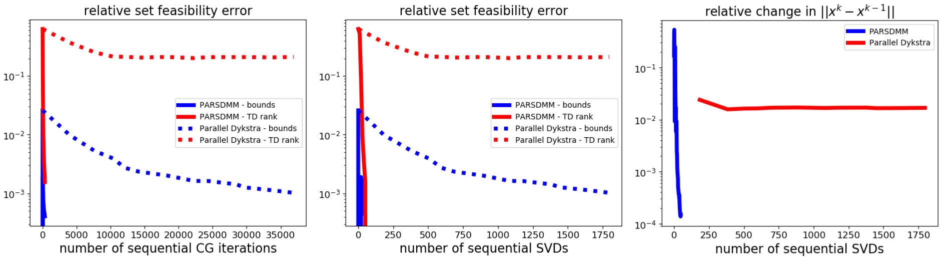


Figure 4.2: Relative transform-domain set feasibility (equation 4.24) as a function of the number of conjugate-gradient iterations and projections onto the set of matrices with limited rank via the SVD. This figure also shows relative change per iteration in the solution x.

4.4.2 Timings for 2D and 3D projections

The proposed PARSDMM algorithm (algorithm 3) is suitable for small 2D models ($\approx 50^2$ pixels) all the way up to large 3D models (at least 300^3). To get an idea about solution times versus model size, as well as how beneficial the parallelism and multilevel continuation are, we show timings for projections of geological models onto two different intersections for the four modes of operation: PARSDMM, parallel PARSDMM, multilevel PARSDMM, and multilevel parallel PARSDMM. As we mentioned, the multilevel version has a small additional overhead compared to single-level PARSDMM because of one interpolation procedure per level. Parallel PARSDMM has communication overhead compared to serial PARSDMM. However, serial PARSDMM still uses multi-threading for projections, the matrix-vector product in the conjugate-gradient method, and BLAS operations, but the y_i and v_i computations in Algorithm 3 remain sequential for every $i = 1, 2, \cdots, p, p+1$, contrary to parallel PARSDMM. We carry our computations out on a dedicated cluster node with 2 CPUs per node with 10 cores per CPU (Intel Ivy Bridge 2.8 GHz E5-2680v2) and 128 GB of memory per node.

The following sets are used in Chapter 3 to regularize a geophysical inverse problem and form the intersection for our first test case:

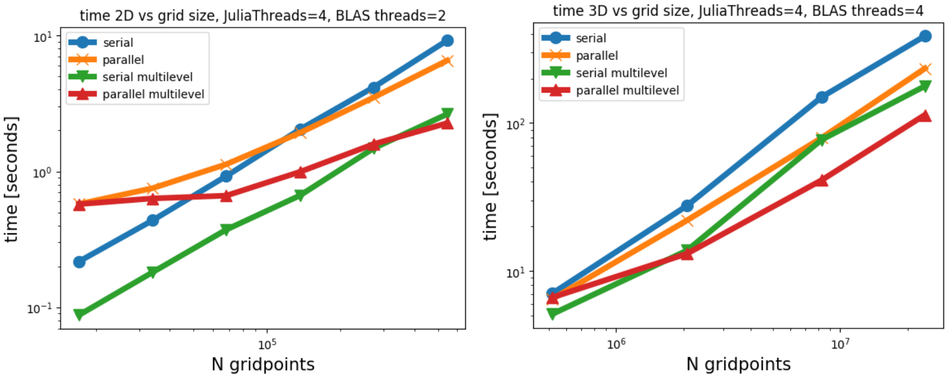


Figure 4.3: Timings for a 2D and 3D example where we project a geological model onto the intersection of bounds, lateral smoothness, and vertical monotonicity constraints.

1. $\{m \,|\, \sigma_1 \leq m[i] \leq \sigma_2\}$: bound constraints
2. $\{m \,|\, -\sigma_3 \leq ((D_x \otimes I_z)m)[i] \leq \sigma_3\}$: lateral smoothness constraints. There are two of these constraints in the 3D case: for the x and y direction separately.
3. $\{m \,|\, 0 \leq ((I_x \otimes D_z)m)[i] \leq \infty\}$: vertical monotonicity constraints

The results in Figure 4.3 show that the multilevel strategy is much faster than the single-level version of PARSDMM. The multilevel overhead costs are thus small compared to the speedup. It also shows that, as expected, the parallel versions require some communication time, so the problems need to be large enough for the parallel version of PARSDMM to offer speedups compared to its serial counterpart.

The previous example uses four constraint sets that each use a different linear operator, but all of them are a type of bound constraint. The y_i computation (projection onto a simple set in closed form) in PARSDMM (Algorithm 3) is therefore fast for all sets. As a result, parallel PARSDMM should lead to a speedup compared to serial computations of all y_i, as we verify in Figure 4.3. We now show an example where one of the sets uses a much more time-consuming y_i computation than the other set, which

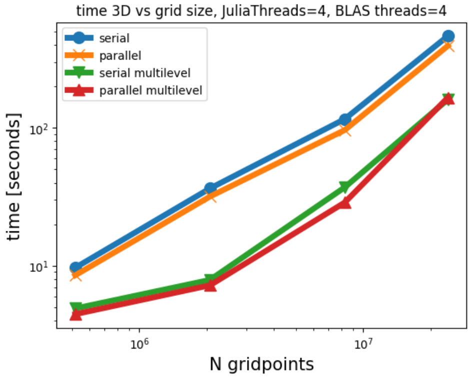


Figure 4.4: Timings for a 3D example where we project a geological model onto the intersection of bound constraints and an ℓ_1-norm constraint on the vertical derivative of the image. Parallel computation of all y_i and v_i does not help in this case, because the ℓ_1-norm projection is much more time consuming than the projection onto the bound constraints. The time savings for other computations in parallel are then canceled out by the additional communication time.

leads to the expectation that parallel PARSDMM only offers minor speedups compared to serial PARSDMM. The second constraint set onto which we project is the intersection of:

1. $\{m \mid \sigma_1 \leq m[i] \leq \sigma_2\}$: bound constraints
2. $\{m \mid \|(I_x \otimes I_y \otimes D_z)m\|_1 \leq \sigma_3\}$, with a constraint that is 50% of the true model: $\sigma_3 = 0.5\|(I_x \otimes I_y \otimes D_z)m_*\|_1$: directional anisotropic total-variation

Figures 4.3 and 4.4 show that parallel computations of the y_i and v_i vectors in PARSDMM is not always beneficial, depending on the number of constraint sets, model size, and time it takes to project onto each set.

4.4.3 Geophysical parameter estimation with constraints

Seismic full-waveform inversion (FWI) estimates rock properties (acoustic velocity in this example) from seismic signals (pressure) measured by hydrophones. FWI is a partial-differential-equation (PDE) constrained optimization problem where after eliminating the PDE constraint, the simulated data, $d_{\text{predicted}}(m) \in \mathbb{C}^M$, are connected nonlinearly to the unknown model parameters, $m \in \mathbb{R}^N$. We assume that we know the source and receiver locations, as well as the source function. A classic example of an objective for FWI is the nonlinear least-squares misfit $f(m) = 1/2\|d_{\text{obs}} - d_{\text{predicted}}(m)\|_2^2$, which we use for this numerical experiment.

FWI is a problem hampered by local minima. Empirical evidence in [Esser et al., 2016, Yong et al., 2018] and Chapters 2 and 3 suggests that we can mitigate issues with parasitic local minima by insisting that all model iterates be elements of the intersection of multiple constraint sets. This means that we add regularization to the objective $f(m) : \mathbb{R}^N \to \mathbb{R}$ in the form of multiple constraints—i.e., we have

$$\min_m f(m) \quad \text{s.t.} \quad m \in \mathcal{V} = \bigcap_{i=1}^{p} \mathcal{V}_i. \tag{4.27}$$

While many choices exist to solve this constrained optimization problem, we use the spectral projected gradient (SPG) algorithm with a non-monotone line search [Birgin et al., 1999] to solve the above problem. SPG uses information from the current and previous gradient of $f(m)$ to approximate the action of the Hessian of $f(m^k)$ with the scalar α: the Barzilai-Borwein step length. At iteration k, SPG updates the model iterate as follows:

$$m^{k+1} = (1-\gamma)m^k - \gamma \mathcal{P}_{\mathcal{V}}(m^k - \alpha \nabla_m f(m^k)), \tag{4.28}$$

where the non-monotone line search determines $\gamma \in (0, 1]$. This line-search requires a lower function value than the maximum function value of the previous five iterations for our numerical experiment. We see that the model iterate m^k is, because of the projection onto $\mathcal{V}$, feasible at every iteration. Moreover, m^k remains feasible for line-search steps to estimate γ if we assume the initial point to be feasible and use convex sets only. In this case, the model iterates m^k and trial points form a line segment. Because both endpoints are in a convex set, the m^{k+1} remain feasible. As a result, we only need a single projection onto the intersection of the different constraints ($\mathcal{P}_{\mathcal{V}}$) for each SPG iteration. We use PARSDMM (Algorithm 3) and multilevel PARSDMM (Algorithm 5) to compute this projection. The total number of SPG iterations plus line-search steps is limited to the relatively small number of ten, because these require the solution of multiple PDEs, which is computationally intensive, especially in 3D.

The experimental setting is as follows: The Helmholtz equation models the wave propagation in an acoustic model. The data acquisition system is a vertical-seismic-profiling experiment with sources at the surface and receivers in a well, see Figure 4.5. All boundaries are perfectly-matched-layers (PML) that absorb outgoing waves as if the model is spatially unbounded. The challenges that we address by constraining the model parameters are: one-sided 'source illumination' that often leads to spurious artifacts in the source-receiver direction, a limited frequency range ($3 - 10$ Hertz), and the non-convexity of the data-misfit $f(m)$. We use the software by Da Silva and Herrmann [2017] to simulate seismic data and compute $f(m)$ and $\nabla_m f(m)$.

This example illustrates that *(a)* adding multiple constraints results in better parameter estimation compared to one or two constraint sets for this example; *(b)* non-convex constraints connect more directly to certain types of prior knowledge about the model than convex sets do; *(c)* we can solve problems with non-convex sets reliably enough such that the results almost satisfy all constraints; *(d)* multilevel PARSDMM for computing projections onto non-convex intersections performs better empirically than the single-level scheme.

The prior knowledge consists of: *(a)* minimum and maximum velocities ($2350 - 2650$ m/s); *(b)* The anomaly is rectangular , but we do not know the size, aspect ratio, or location.

Before we add multiple non-convex constraints, let us look at what happens with simple bound and total-variation constraints. Figure 4.5 shows the true model, initial guess, and the estimated models using various combinations of constraints. The data acquisition geometry causes the model estimate with bound constraints to be an elongated diagonal anomaly that is incorrect in terms of size, shape, orientation, and parameter values.

Anisotropic total-variation (TV) seems like a good candidate to promote 'blocky' model structures, but it may be difficult to select a total-variation constraint, i.e., the size of the TV-ball. The result in Figure 4.5(d) shows that even in the unusual case that we know and use a TV constraint equal to the TV of the true model, we obtain a model estimate that shows minor improvements compared to the estimation with bounds only. While many of the oscillations outside of the rectangular anomaly are damped, the shape of the anomaly itself is still far from the truth.

As we will demonstrate, the inclusion of multiple non-convex cardinality and rank constraints help the parameter estimation in this example. From the prior information that the anomaly is rectangular and aligned with the domain boundaries, we deduce that the rank of the model is equal to two. We also know that the cardinality of the discrete gradient of each row and each column is less than or equal to two as well. If we assume that the anomaly is not larger than half the total domain extent in each direction, we know that the cardinality of the discrete derivative of the model (in matrix format) is not larger than the number of grid points in each direction. To summarize, the following constraint sets follow from the prior information:

1. $\{x \mid \operatorname{card}((D_z \otimes I_x)x) \leq n_x\}$
2. $\{x \mid \operatorname{card}((I_z \otimes D_x)x) \leq n_z\}$
3. $\{x \mid \operatorname{rank}(x) \leq 3\}$
4. $\{x \mid 2350 \leq x[i] \leq 2650 \, \forall i\}$
5. $\{x \mid \operatorname{card}(D_x X[i,:]) \leq 2 \text{ for } i \in \{1, 2, \ldots, n_z\}\}$, $X[i,:]$ is a row of the 2D model
6. $\{x \mid \operatorname{card}(D_z X[:,j]) \leq 2 \text{ for } j \in \{1, 2, \ldots, n_x\}\}$, $X[:,j]$ is a column of the 2D model

We use slightly overestimated rank and matrix cardinality constraints compared to the true model to mimic the more realistic situation that not all prior knowledge was correct. The results in Figure 4.5 use single-level PARSDMM to compute projections onto the intersection of constraints, and show that an intersection of non-convex constraints and bounds can lead to improved model estimates. Figure 4.5(e) is the result of working with constraints $[1, 2, 4]$, Figure 4.5(f) uses constraints $[1, 2, 4, 5, 6]$, and Figure 4.5(g) uses all constraints $[1, 2, 3, 4, 5, 6]$. The result with rank constraints and both matrix and row/column-based cardinality constraints on the discrete gradient of the model is the most accurate in terms of the recovered anomaly shape. All results in Figure 4.5 that work with non-convex sets are at least as accurate as the result obtained with the true TV in terms of anomaly shape. Another important observation is that all non-convex results estimate a lower-than-background velocity anomaly, although not as low as the true anomaly. Contrary, the models obtained using convex sets show incorrect higher-than-background velocity artifacts in the vicinity of the true anomaly location.

Figure 4.6 is the same as Figure 4.5, except that we use multilevel PARSDMM (Algorithm 5) with three levels and a coarsening of a factor two per level. Comparing single level with multilevel computations of the projection, we see that the multilevel version of PARSDMM performs better in general. In Figures 4.5(e) and 4.5(f), we see that the result of single-level PARSDMM inside SPG does not exactly satisfy constraint set numbers 5 and 6, because the cardinality of the derivative of the model in x and z directions is not always less than or equal to two for each row and column. The results from multilevel PARSDMM inside SPG, Figure 4.6(a) and 4.6(b), satisfy the constraints on the cardinality of the derivative of the image per row and column. As a result, the models are closer to the rectangular shape of the true model. This is only one example with a few different constraint combinations so we cannot draw general conclusions about the performance of single versus multilevel schemes, but the empirical findings are encouraging and in line with observations by Macdonald and Ruthotto [2018].

4.4.4 Learning a parametrized intersection from a few training examples

In the introduction, we discussed how to formulate inverse problems as a projection or feasibility problem (4.4). With the following two examples we show that our algorithm (4.15) is a good candidate to solve inverse problems

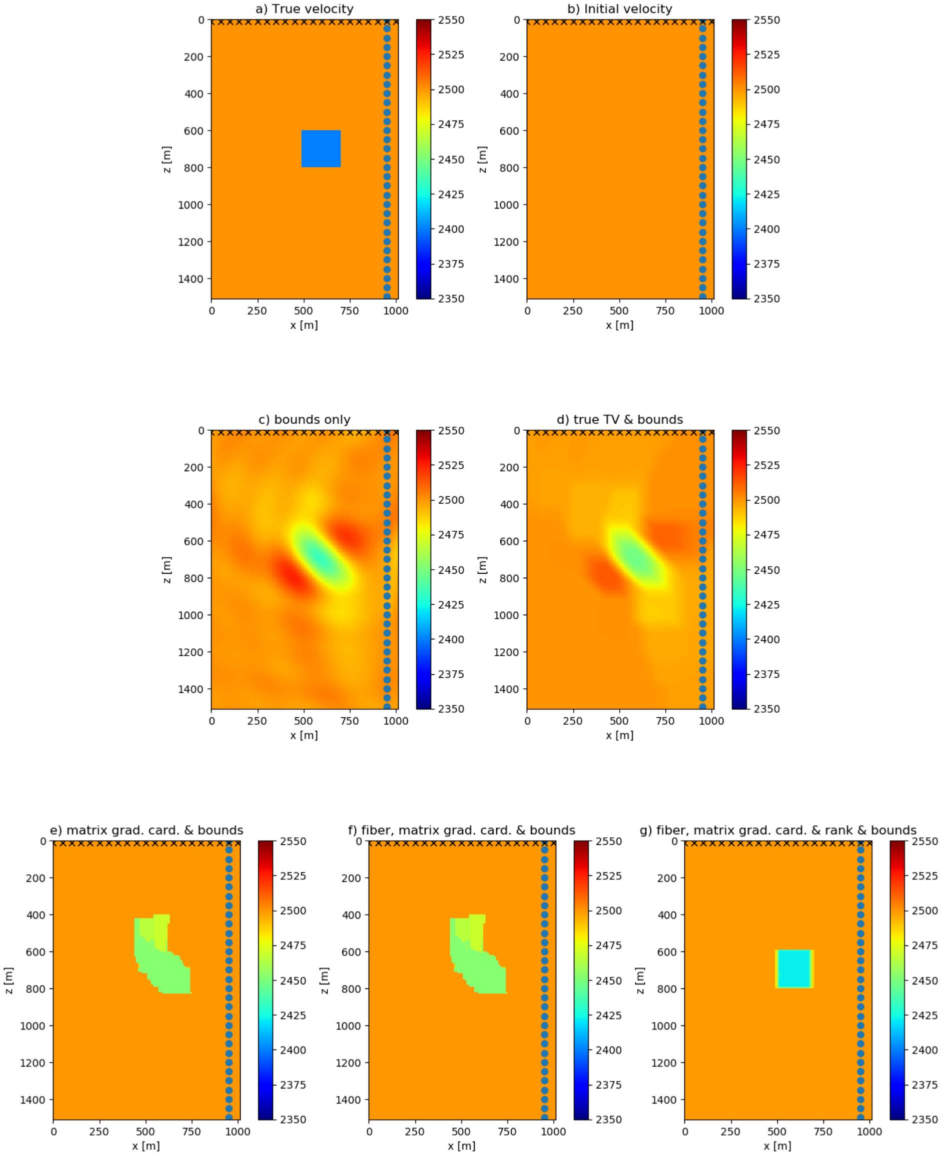


Figure 4.5: True, initial, and estimated models with various constraint combinations for the full-waveform inversion example. Crosses and circles represent sources and receivers, respectively. All projections inside the spectral projected gradient algorithm are computed using single-level PARSDMM.

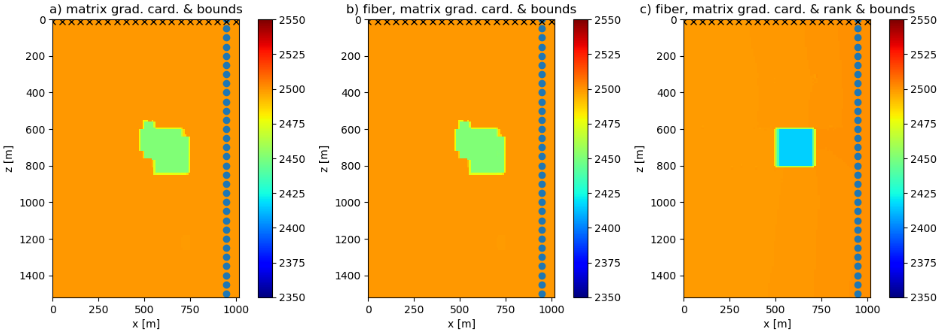


Figure 4.6: Estimated models with various constraint combinations for the full-waveform inversion example. Crosses and circles represent sources and receivers, respectively. All projections inside the spectral projected gradient algorithm are computed using coarse-to-fine multilevel PARSDMM with three levels and a coarsening of a factor two per level.

as a projection or feasibility problem, because we mitigate rapidly increasing computation times for problems with many sets, by taking the similarity between linear operators in set definitions into account. Of course, we can only use multiple constraint sets if we have multiple pieces of prior information. Combettes and Pesquet [2004] present a simple solution and note that for 15 out of 20 investigated data-sets, 99% of the images have a total-variation within 20% of the average total variation of the data-set. The average total-variation serves as a robust constraint that typically leads to good results. Here we follow the same reasoning, but we will work with many constraint sets that we learn from a few example images. To summarize, our learning and solution strategy is as follows:

1. Observe the constraint parameters of various constraints in various transform-domains for all training examples (independently in parallel for each example and each constraint).
2. Add a data-fit constraint to the intersection.
3. The solution of the inverse problem is the projection of an initial guess

m onto the learned intersection of sets

$$\min_{x,\{y_i\}} \frac{1}{2}\|x-m\|_2^2+\sum_{i=1}^{p-1} \iota_{\mathcal{C}_i}(y_i)+\iota_{\mathcal{C}_p^{\text{data}}}(y_p) \quad \text{s.t.} \quad \begin{cases} A_i x = y_i \\ F x = y_p \end{cases}, \tag{4.29}$$

where F is a linear forward modeling operator and we solve this problem with Algorithm 3.

Before we proceed to the examples, it is worth mentioning the main advantages and limitations of this strategy. Because all set definitions are independent of all other sets, there are no penalty/weight parameters, and we avoid hand-tuning the constraint definitions. Unlike neural networks for imaging inverse problems that often need large numbers of training examples, we can observe 'good' constraints from just one or a few example images. Methods that do not require training, such as basis-pursuit type formulations [e.g., Lustig et al., 2007, Candès and Recht, 2009, van den Berg and Friedlander, 2009, Becker et al., 2011, Aravkin et al., 2014], often minimize the ℓ_1 norm or nuclear norm of transform-domain coefficients (total-variation, wavelet) of an image subject to a data-fit constraint. However, without learning, these methods require hand picking a suitable transform for each class of images. We will work with many transform-domain operators simultaneously, so that at least some of the constraint/linear operator combinations will describe uncorrupted images with small norms/bounds/-cardinality, but not noisy/blurred/masked images. Note that we are not learning any dictionaries, but work with pre-defined transforms such as the Fourier basis, wavelets, and linear operators based on discrete gradients. A limitation of the constraint learning strategy that we use here is that it does not generalize very well to other classes of images and dataset.

For both of the examples we observe the following constraint parameters from exemplar images:

1. $\{m \mid \sigma_1 \leq m[i] \leq \sigma_2\}$ (upper and lower bounds)
2. $\{m \mid \sum_{j=1}^{k} \lambda[j] \leq \sigma_3\}$ with $m = \text{vec}(\sum_{j=1}^{k} \lambda[j] u_j v_j^*)$ is the SVD of the image (nuclear norm)

3. $\{m \mid \sum_{j=1}^{k} \lambda[j] \leq \sigma_4\}$, with $(I_x \otimes D_z)m = \text{vec}(\sum_{j=1}^{k} \lambda[j] u_j v_j^*)$ is the SVD of the vertical derivative of the image (nuclear norm of discrete gradients of the image, total-nuclear-variation). Use the same for the x-direction.
4. $\{m \mid \|Am\|_1 \leq \sigma_5\}$ with $A = ((I_x \otimes D_z)^\top \; (D_x \otimes I_z)^\top)^\top$ (anisotropic total-variation)
5. $\{m \mid \sigma_6 \leq \|m\|_2 \leq \sigma_7\}$ (annulus)
6. $\{m \mid \sigma_8 \leq \|Am\|_2 \leq \sigma_9\}$ with $A = ((I_x \otimes D_z)^\top \; (D_x \otimes I_z)^\top)^\top$ (annulus of the discrete gradients of the training images)
7. $\{m \mid \|Am\|_1 \leq \sigma_{10}\}$ with $A =$ discrete Fourier transform (ℓ_1-norm of DFT coefficients)
8. $\{m \mid -\sigma_{11} \leq ((D_x \otimes I_z)m)[i] \leq \sigma_{12}\}$ (slope-constraints in x and z direction, bounds on the discrete gradients of the image)
9. $\{m \mid l[i] \leq (Am)[i] \leq u[i]\}$, with $A =$ discrete cosine transform (pointwise bound-constraints on DCT coefficients)

These are nine types of convex and non-convex constraints on the model properties (11 sets passed to PARSDMM because sets three and eight are applied to the two dimensions separately). For data-fitting, we add a pointwise constraint, $\{x \mid l \leq (Fx - d_{\text{obs}}) \leq u\}$ with a linear forward model $F \in \mathbb{R}^{M \times N}$.

Joint deblurring-denoising-inpainting

The goal of the first example is to recover a $[0 - 255]$ grayscale image from 20% observed pixels of a blurred image (25 pixels known motion blur), where each observed data point also contains zero-mean random noise in the interval $[-10 - 10]$. The forward operator F is thus a subsampled banded matrix (restriction of an averaging matrix). As an additional challenge, we do not assume exact knowledge of the noise level and work with the overestimation $[-15 - 15]$. The data set contains a series of images from 'Planet Labs PlanetScope Ecuador' with a resolution of three meters, available at openaerialmap.org. There are 35 patches of 1100×1100 pixels for training, some of which are displayed in Figure 4.7.

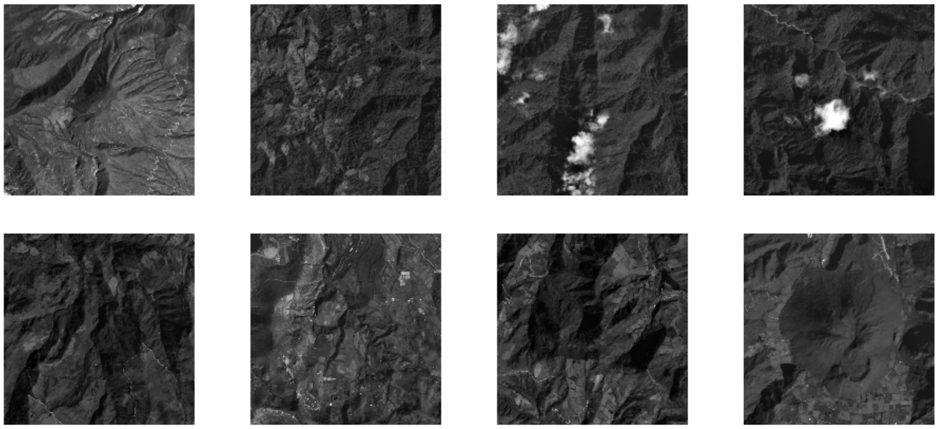


Figure 4.7: A sample of 8 out of 35 training images.

We compare the results of the proposed PARSDMM algorithm with the 11 learned constraints, with a basis pursuit denoise (BPDN) formulation. Basis-pursuit denoise recovers a vector of wavelet coefficients, c, by solving $\min_c \|c\|_1$ s.t. $\|FW^*c - d_{\text{obs}}\|_2 \leq \sigma$ (BPDN-wavelet) with the SPGL1 toolbox [van den Berg and Friedlander, 2009]. The matrix W represents the wavelet transform: Daubechies Wavelets as implemented by the SPOT linear operator toolbox (http://www.cs.ubc.ca/labs/scl/spot/index.html) and computed with the Rice Wavelet Toolbox (RWT, github.com/ricedsp/rwt).

In Figure 4.8 we see that an overestimation of σ in the BPDN formulation results in oversimplified images, because the ℓ_2-ball constraint is too large which leads to a coefficient vector c that has an ℓ_1-norm that is smaller than the ℓ_1-norm of the true image. The values for l and u in the data-fit constraint $\{x \mid l \leq (Fx - d_{\text{obs}}) \leq u\}$, are also too large. However, the results from the projection onto the intersection of multiple constraints suffer much less from overestimated noise levels, because there are many other constraints that control the model properties. The results in Figure 4.8 show that the learned set-intersection approach achieves a higher PSNR for all evaluation images compared to the BPDN formulation.

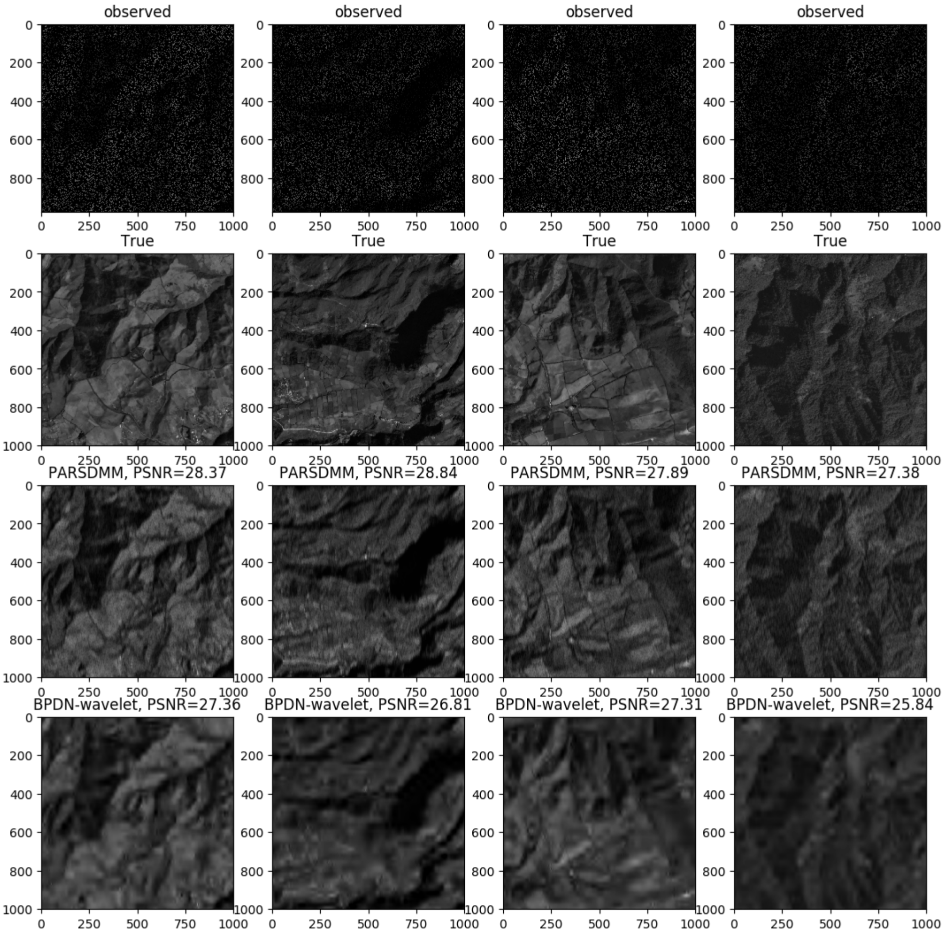


Figure 4.8: Reconstruction results from 80% missing pixels of an image with motion blur (25 pixels) and zero-mean random noise in the interval $[-10, 10]$. Results that are the projection onto an intersection of 12 learned constraints sets with PARSDMM are visually better than BPDN-wavelet results.

Image desaturation

To illustrate the versatility of the strategy, algorithm and constraint sets from the previous example, we now solve an image desaturation problem for a different data set. The only two things we need to change are the constraint set parameters, which we observe from new training images (Figure 4.9), as well as a different linear forward operator F. The data set contains image patches (1500 × 1250 pixels) from the 'Desa Sangaji Kota Ternate' image with a resolution of 11 centimeters, available at openaerialmap.org. The corrupted observed images are saturated grayscale and generated by clipping the pixel values from $0-60$ to 60 and from $125-255$ to 125, so there is saturation on both the dark and bright pixels. If we have no other information about the pixels at the clipped value, the desaturation problem implies the point-wise bound constraints [e.g., Mansour et al., 2010]

$$\begin{cases} 0 \leq x[i] \leq 60 & \text{if } d^{\text{obs}}[i] = 60 \\ x[i] = d^{\text{obs}}[i] & \text{if } 60 \leq d^{\text{obs}}[i] \leq 125 \\ 125 \leq x[i] \leq 255 & \text{if } d^{\text{obs}}[i] = 125 \end{cases}. \tag{4.30}$$

The forward operator is thus the identity matrix. We solve problem (4.29) with these point-wise data-fit constraints and the model property constraints listed in the previous example.

Figure 4.10 shows the results, true and observed data for four evaluation images. Large saturated patches are not desaturated accurately everywhere, because they contain no non-saturated observed pixels that serve as 'anchor' points.

Both the desaturation and the joint deblurring-denoising-inpainting example show that PARSDMM with multiple convex and non-convex sets converges to good results, while only a few training examples were sufficient to estimate the constraint set parameters. Because of the problem formulation, algorithms, and simple learning strategy, there were no parameters to hand-pick.

Figure 4.9: A sample of 8 out of 16 training images.

4.5 Discussion and future research directions

We developed algorithms to compute projections onto intersections of multiple sets that help us setting up and solving constrained inverse problems. Our design choices, together with the constrained formulation, minimize the number of parameters that we need to hand-pick for the problem formulation, algorithms, and regularization. Our software package `SetIntersectionProjection` should help inverse problem practitioners to test various combinations of constraints for faster evaluation of their strategies to solve inverse problems. Besides practicality, we want our work to apply to not just toy problems, but also to models on larger 3D grids. We achieved this via automatic adjustment of scalar algorithm parameters, parallel implementation, and multilevel acceleration. There are some limitations, but also opportunities to increase computational performance that we will now discuss.

Regarding the scope of the `SetIntersectionProjection` software package, it is important to emphasize that satisfying a constraint for our applications in imaging inverse problems is different from solving general (non-convex) optimization problems. When we refer to ‘reliably’ solving a non-

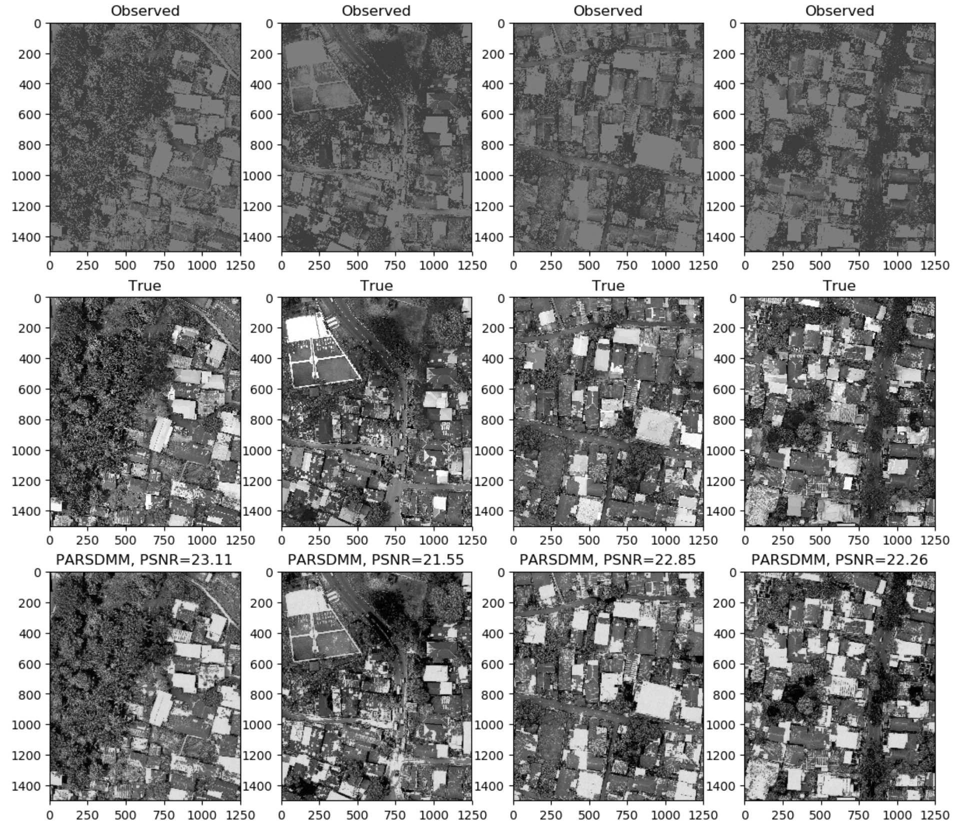


Figure 4.10: Reconstruction results from recovery from saturated images as the projection onto the intersection of 12 constraint sets.

convex problem, we are satisfied with an algorithm that usually approximates the solution well. For example, if we seek to image a model m that has k discontinuities, we add the constraint $\{m \,|\, \operatorname{card}(Dm) \leq k\}$ where D is a derivative operator. A satisfying solution for our applications has k large vector entries, whereas all others are small. We do not need to find a vector that has a cardinality of exactly k, because the estimated model is the same for practical purposes if the results are assessed qualitatively/visually, or where the expected resolution is much lower than the fine details we could potentially improve. Moreover, the forward operator for the inverse problem often is not sensitive to small changes in the model, and we do not benefit from spending much more computational time trying to find a more accurate solution to the non-convex problem. Besides the multilevel projection and automatic adjustment of augmented-Lagrangian parameters that we already use, [Diamond et al., 2018] present several other heuristics that can improve the solution of non-convex problems in the context of ADMM-based algorithms. Future work could test if these heuristics are computationally feasible for our often large-scale problems and if they cooperate with our other heuristics.

The proposed algorithms are currently set up for general sparse, sparse and banded, and orthogonal matrices such as the discrete Fourier transform. A general and non-orthogonal dense matrix, A_i, will slow down the solution of the linear system with $\sum_{i=1}^{p+1} A_i^\top A_i$, and is therefore not supported. However, if the dense matrix is flat, $A_i \in \mathbb{R}^{M \times N}$ with $M \ll N$, such as a learned transform (dictionary), we can use this as a subspace constraint. This means that the model parameters $m \in \mathbb{R}^N$ are an element of the set $\{m \,|\, m = A_i c, \, c \in \mathbb{R}^N\}$ with coefficient vector c. The projection onto this set is known in closed form, and we do not move the dense linear operator out of the set and into the normal equations in the x-minimization step of (4.20) because $A_i^\top A_i$ would become a large and dense matrix.

Besides the limitations and scope of this work, we highlight two ways how we can reduce computation times for Algorithm 3 and its multilevel version. First, we recognize that our algorithms use ADMM as its foundation, which is a synchronous algorithm. This means that the computations of the projections (y-update) in parallel are as slow as the most time-consuming projection. Without fundamentally changing the algorithms to asynchronous or randomized projection methods, we can take a purely software-based approach. Because we compute projections in parallel, where each projection uses several threads, we are free to reallocate threads from the fastest projection to the slowest and reduce the total computational time.

A second computational component that may be improved is the inexact linear systems solve with the conjugate-gradient (CG) method. We do not use a preconditioner at the moment. Preliminary tests with a simple diagonal (Jacobi) preconditioner or multigrid V-cycle did reduce the number of CG iterations, but not the running time for CG in general. There are a few challenges we face when we design a preconditioner: *(i)* users may select a variety of linear operators *(ii)* the system matrix is the weighted sum of multiple linear systems in normal equation form, where the weights may change every two PARSDMM iterations *(iii)* the number of CG iterations varies per PARSDMM iteration and is often less than ten, which makes it hard for preconditioners to reduce the time consumption if they require some computational overhead or setup cost.

Finally we mention how our software package can work cooperatively with recent developments in plug-and-play regularizers [Venkatakrishnan et al., 2013]. The general idea is to use image processing techniques such as non-local means and BM3D, or pre-trained neural networks [Zhang et al., 2017, Bigdeli and Zwicker, 2017, Fan et al., 2017, Chang et al., 2017, Aggarwal et al., 2018, Buzzard et al., 2018], as a map $g(x) : \mathbb{R}^N \rightarrow \mathbb{R}^N$ that behaves like a proximal operator or projector. Despite the fact that these plug-and-play algorithms do not generally share non-expansiveness properties with projectors [Chan et al., 2017], they are successfully employed in optimization algorithms based on operator-splitting. In our case, we use a neural network as the projection operator with the identity matrix as associated linear operator. In this way we can combine data-constraints and other prior information with a network. A potential challenge with the plug-and-play concept for constrained optimization is the difficulty to verify that the intersection of constraints is effectively non-empty, i.e., can $g(x)$ map to points in the intersection of the other constraint sets? Some preliminary tests showed encouraging results and we will explore this line of research further.

4.6 Conclusions

We developed novel algorithms and the corresponding 'SetIntersectionProjection' software for the computation of projections onto the intersection of multiple constraint sets. These intersection projections are an important tool for the regularization of inverse problems. They may be used as the projection part of projected gradient/(quasi-)Newton algorithms. Projections onto an intersection also solve set-based formulations for linear image processing problems, possibly combined with simple learning techniques to extract set definitions from example images. Currently available algorithms for the projection onto intersections of sets are efficient if we know the projections onto each set in closed form. However, many sets of interest include linear operators that require other algorithms to solve sub-problems. The presented methods and software are designed to work with multiple constraint sets based on small 2D, as well as larger 3D models. We enhance computational performance by specializing the software for projection problems, exploiting different levels of parallelism on multi-core computing platforms, automatic selection of scalar (acceleration) parameters, and a coarse-to-fine grid multilevel implementation. The software is practical, also for non-expert users, because we do not need manual step-size selection or related operator norm computations and the algorithm inputs are pairs of linear operators and projectors which the software also generates. Another practical feature is the support for simultaneous set definitions based on the entire image/tensor and each slice/row/column. Because we focus on multiple constraints, there is less of a need to choose the 'best' constraint with the 'best' linear operator/transform for a given inverse problem. More constraints are not much more difficult to deal with than a one or two constraints, also in terms of computational cost per iteration. We demonstrated the versatility of the presented algorithms and software using examples from partial-differential-equation based parameter estimation and image processing. These examples also show that the algorithms perform well on problems that include non-convex sets.

Chapter 5

Minkowski sets for the regularization of inverse problems.

5.1 Introduction

The inspiration for this work is twofold. First, we want to build on the success of regularization with intersections of constraint sets and projection methods, see [Gragnaniello et al., 2012, Pham et al., 2014a,b, Smithyman et al., 2015, Esser et al., 2016, Yong et al., 2018], and the examples presented in Chapters 2, 3, and 4. These works regularize a parameter estimation problem $\min_m f(m)$ for $f : \mathbb{R}^N \to \mathbb{R}$, by constraining the model m to an intersection of p convex and possibly non-convex sets, $\bigcap_{i=1}^p \mathcal{C}_i$. The corresponding optimization problem reads $\min_m f(m)$ s.t. $m \in \bigcap_{i=1}^p \mathcal{C}_i$ with $m \in \mathbb{R}^N$ the vector of model parameters. In the references mentioned above, this type of problem is successfully solved with projection-based algorithms. However, prior knowledge represented as a single set or as an intersection of different sets may not capture all we know. For instance, if the model contain oscillatory as well as blocky features. Because these are fundamentally different properties, working with one or multiple constraint sets alone may not able to express the simplicity of the entire model.

Motivated by ideas from morphological component analysis (MCA, Osher et al. [2003]; Schaeffer and Osher [2013]; Starck et al. [2005]; Ono et al. [2014]) and robust or sparse principal component analysis (RPCA, Candès et al. [2011]; Gao et al. [2011a]; Gao et al. [2011b]), we consider an additive model structure. CTD/MCA exploit this structure by decomposing m into two or more components—e.g., into a blocky cartoon-like component $u \in \mathbb{R}^N$ and an oscillatory component $v \in \mathbb{R}^N$. For this purpose, a penalty method is often used stating the decomposition problem as

$$\min_{u,v} \|m - u - v\| + \frac{\alpha}{2}\|Au\| + \frac{\beta}{2}\|Bv\|. \tag{5.1}$$

While this method has been successful, it requires careful choices for the penalty parameters $\alpha > 0$ and $\beta > 0$. These parameters determine how 'much' of m ends up in each component, but also depend on the noise level. In addition, the value of these parameters relates to the choices for the linear operators $A \in \mathbb{C}^{M_1 \times N}$ and $B \in \mathbb{C}^{M_2 \times N}$. When working with multiple constraints, we have seen that avoiding penalty parameters is more practical. Decomposition problems suggest the same, as the number of regularizers involved is likely to be even larger.

To handle situations where the model contains two or more morphological components, we explore the use of Minkowski sets for regularizing inverse problems. For this purpose, we require that the vector of model parameters, m, is an element of the Minkowski set $\mathcal{V}$, or vector sum of two sets $\mathcal{C}_1$ and $\mathcal{C}_1$, which is defined as

$$\mathcal{V} \equiv \mathcal{C}_1 + \mathcal{C}_1 = \{m = u + v \mid u \in \mathcal{C}_1, v \in \mathcal{C}_2\}. \tag{5.2}$$

A vector m is an element of $\mathcal{V}$ if it is the sum of vectors $u \in \mathcal{C}_1$ and $v \in \mathcal{C}_2$. Each set describes particular model properties for each component. These include total-variation, sparsity in a transform domain (Fourier, wavelets, curvelets, shearlets) or matrix rank. For practical reasons, we assume that all sets, sums of sets, and intersections of sets are non-empty, which implies that our optimization problems have at least one solution. Moreover, we will use the property that the sum of p sets $\mathcal{C}_i$ is convex if all $\mathcal{C}_i$ are convex [Hiriart-Urruty and Lemaréchal, 2012, Page 24]. We apply our set-based regularization with the Euclidean projection operator:

$$\mathcal{P}_{\mathcal{V}}(m) \in \arg\min_{x} \frac{1}{2}\|x - m\|_2^2 \quad \text{s.t} \quad x \in \mathcal{V}. \tag{5.3}$$

This projection allows us to use Minkowski constraint sets in algorithms such as (spectral) projected gradient (SPG, Birgin et al. [1999]), projected quasi-Newton [PQN, Schmidt et al., 2009], projected Newton algorithms [Bertsekas, 1982, Schmidt et al., 2012], and proximal-map based algorithms if we include the Minkowski constraint as an indicator function. We define this indicator function for a set $\mathcal{V}$ as

$$\iota_{\mathcal{V}}(m) = \begin{cases} 0 & \text{if } m \in \mathcal{V}, \\ +\infty & \text{if } m \notin \mathcal{V}, \end{cases} \tag{5.4}$$

and the proximal map for a function $g(m) : \mathbb{R}^N \rightarrow \mathbb{R} \cup \{+\infty\}$ as $\text{prox}_{\gamma,g}(m) = \arg\min_x g(x) + \frac{\gamma}{2}\|x - m\|_2^2$, with $\gamma > 0$. The proximal map for the indicator function of a set is the projection: $\text{prox}_{\gamma,\iota_{\mathcal{V}}}(m) = \mathcal{P}_{\iota_{\mathcal{V}}}(m)$.

While the above framework is powerful, it lacks certain critical features needed for solving problems that involve physical parameters. For instance, there is, in general, no guarantee that the sum of two or more components lies within lower and upper bounds or satisfies other crucial constraints. It is also not straightforward to include multiple pieces of prior information for each component.

5.1.1 Related work

The above introduced decomposition strategies of morphological component analysis or cartoon-texture (MCA, Osher et al. [2003]; Schaeffer and Osher [2013]; Starck et al. [2005]; Ono et al. [2014]) and robust or sparse principal component analysis (RPCA, Candès et al. [2011]; Gao et al. [2011a]; Gao et al. [2011b]) share the additive model construction with multiscale decompositions in image processing [e.g., Meyer, 2001, Tadmor et al., 2004]. While each of the sets that appear in a Minkowski sum can describe a particular scale, this is not our primary aim or motivation. We use the summation structure to build more complex models out of simpler ones, more aligned with cartoon-texture decomposition and robust principal component analysis.

Projections onto Minkowski sets also appear in computational geometry, collision detection, and computer-aided design [e.g., Dobkin et al., 1993, Varadhan and Manocha, 2006, Lee et al., 2016], but the problems and applications are different. In our case, sets describe model properties and prior knowledge in $\mathbb{R}^N$. In computational geometry, sets are often the vertices of polyhedral objects in $\mathbb{R}^2$ or $\mathbb{R}^3$ and do not come with closed-form expressions for projectors or the Minkowski sum. We do not need to form the Minkowski set explicitly, and we show that projections onto the set are sufficient to regularize inverse problems.

5.1.2 Contributions and outline

We propose a constrained regularization approach suitable for inverse problems with an emphasis on physical parameter estimation. For our applications, this implies that we need to work with multiple constraints for each component while offering assurances that the sum of the components also adheres to certain constraints. For this purpose, we introduce generalized Minkowski sets and a formulation void of penalty parameters. As [Gragnaniello et al., 2012, Pham et al., 2014b,a, Smithyman et al., 2015, Esser et al., 2016, Yong et al., 2018] and earlier chapters in this thesis use projection-based optimization methods, we introduce projections on these generalized sets, followed by a discussion on important algorithmic details and the formulation of inverse problems based on these sets.

Because we are working with constraints, we do not have to worry about selecting trade-off parameters. With the projections, we can also ensure that the model parameters for each iteration of the inversion are within a generalized Minkowski set. As before, we are in a position to relax the constraints gradually. This idea proved to be a successful tactic to solve nonconvex geophysical inverse problems. (See [Smithyman et al., 2015, Esser et al., 2016, Yong et al., 2018] and previous chapters.)

For the software implementation, we extend the open-source `Julia` software package 'SetIntersectionProjection' presented in Chapter 4. The software is suitable for small 2D models, as well as for larger 3D geological models or videos, as we will show in the numerical examples section using seismic parameter estimation and video processing examples. These examples also demonstrate that the proposed problem formulation, algorithm, and software allow us to define constraints based on the entire 2D/3D model, but also simultaneously on slices/rows/columns/fibers of that model. This feature enables us to include certain prior knowledge more directly into the inverse problem.

5.2 Generalized Minkowski set

It is challenging to select a single constraint set or intersection of multiple sets to describe models and images that contain distinct morphological components u and v. While the Minkowski set allows us to define different sets for the different components, problems may arise when working with physical parameter estimation applications.

For instance, there is usually prior knowledge about the physically realistic values in $m \in \mathbb{R}^N$. Moreover, in the previous chapters, we showed successful applications of multiple constraints on the model parameters, and we want to combine that concept with constraints on the components.

The second extension of the basic concept of a Minkowski set is that we allow the constraint set on each component to be an intersection of multiple sets. In this way, we can include multiple pieces of prior information about each component.

We denote the proposed generalized Minkowski constraint set for the regularization of inverse problems as

$$\mathcal{M} \equiv \{m = u + v \mid u \in \bigcap_{i=1}^{p} \mathcal{D}_i, \; v \in \bigcap_{j=1}^{q} \mathcal{E}_j, \; m \in \bigcap_{k=1}^{r} \mathcal{F}_k\}, \tag{5.5}$$

where the model estimate $m \in \mathbb{R}^N$ is an element of the intersection of r sets $\mathcal{F}_k$ and also the sum of two components $u \in \mathbb{R}^N$ and $v \in \mathbb{R}^N$. The vector u is an element of the intersection of p sets $\mathcal{D}_i$, v is an element of the intersection of q sets $\mathcal{E}_j$. It is conceptually straightforward to extend set definition 5.5 to a sum of three or more components, but we work with two components for the remainder of this paper for notational convenience. In the discussion section, we highlight some potential computational challenges that come with a generalized Minkowski sets of more than two components.

The convexity of $\mathcal{M}$ follows from the properties of the sets $\mathcal{D}_i$, $\mathcal{E}_j$ and $\mathcal{F}_k$. From the definition 5.5, we see that $\bigcap_{i=1}^{p} \mathcal{D}_i$, $\bigcap_{j=1}^{q} \mathcal{E}_j$, and $\bigcap_{k=1}^{r} \mathcal{F}_k$ are closed and convex if $\mathcal{D}_i$, $\mathcal{E}_j$ and $\mathcal{F}_k$ are closed and convex for all i, j and k. It follows that $\mathcal{M}$ is a convex set, because it is the intersection of a convex intersection with the Minkowski sum $\bigcap_{i=1}^{p} \mathcal{D}_i + \bigcap_{j=1}^{q} \mathcal{E}_j$, which is also convex. To summarize in words, m is an element of the intersection of two convex sets, one is the convex Minkowski sum, the other is a convex intersection. The set $\mathcal{M}$ is therefore also convex. Note that convexity and closedness of $\bigcap_{i=1}^{p} \mathcal{D}_i$ and $\bigcap_{j=1}^{q} \mathcal{E}_j$ does not imply their sum is closed.

In the following section, we propose an algorithm to compute projection onto the generalized Minkowski set.

5.3 Projection onto the generalized Minkowski set

In the following section, we show how to use the generalized Minkowski set (Equation (5.5)) to regularize inverse problems with computationally cheap or expensive forward operators. First, we need to develop an algorithm to compute the projection onto $\mathcal{M}$, which we denote by $\mathcal{P}_{\mathcal{M}}(m)$. Using $\mathcal{P}_{\mathcal{M}}(m)$, we can formulate inverse problems as a projection, or use the projection operator inside projected gradient/Newton-type algorithms. Each constraint set definition may include a linear operator (the transform-domain operator) in its definition. We make the linear operators explicit, because the projection operator corresponding to, for example, $\{x \mid \|x\|_2 \leq \sigma\}$, is available in closed form and easy to compute, but the projection onto $\{x \mid \|Ax\|_2 \leq \sigma\}$ is not when $AA^T \neq \alpha I$ for $\alpha > 0$ [Combettes and Pesquet, 2011; Parikh and Boyd, 2014; Beck, 2017, Chapter 6 & 7; Diamond et al., 2018]. Let us introduce the linear operators $A_i \in \mathbb{R}^{M_i \times N}$, $B_j \in \mathbb{R}^{M_j \times N}$, and $C_k \in \mathbb{R}^{M_k \times N}$. With indicator functions and exposed linear operators, we formulate the projection of $m \in \mathbb{R}^N$ onto set (5.5) as

$$\begin{aligned}\mathcal{P}_{\mathcal{M}}(m) = \operatorname*{arg\,min}_{u,v,w} \frac{1}{2}\|w - m\|_2^2 + \sum_{i=1}^{p} \iota_{\mathcal{D}_i}(A_i u) \\ + \sum_{j=1}^{q} \iota_{\mathcal{E}_j}(B_j v) + \sum_{k=1}^{r} \iota_{\mathcal{F}_k}(C_k w) + \iota_{w=u+v}(w, u, v),\end{aligned} \tag{5.6}$$

where $\iota_{w=u+v}(w, u, v)$ is the indicator function for the equality constraint $w = u+v$ that occurs in the definition of $\mathcal{M}$. The sets $\mathcal{D}_i$, $\mathcal{E}_j$ and $\mathcal{F}_k$ have the same role as in the previous section. The above problem is the minimization of a sum of functions acting on different as well as shared variables. We recast it in a standard form such that we can solve it using algorithms based on the alternating direction method of multipliers (ADMM, e.g., Boyd et al. [2011]; Eckstein and Yao [2015]). Rewriting in a standard form allows us to benefit from recently proposed schemes for selecting algorithm parameters that decrease the number of iterations and lead to more robust algorithms in case we use non-convex sets [Xu et al., 2017b, ; Xu et al., 2016]. As a first step, we introduce the vector $x \in \mathbb{R}^{2N}$ that stacks two out of the three optimization variables in 5.6 as

$$x \equiv \begin{pmatrix} u \\ v \end{pmatrix}. \tag{5.7}$$

We substitute this new definition in Problem (5.6) and eliminate the equality constraints $w = u + v$ to arrive at

$$\begin{aligned}\mathcal{P}_{\mathcal{M}}(m) = \arg\min_{x} \frac{1}{2}\| \begin{pmatrix} I_N & I_N \end{pmatrix} x - m\|_2^2 + \sum_{i=1}^{p} \iota_{\mathcal{D}_i}(\begin{pmatrix} A_i & 0 \end{pmatrix} x) \\ + \sum_{j=1}^{q} \iota_{\mathcal{E}_j}(\begin{pmatrix} 0 & B_j \end{pmatrix} x) + \sum_{k=1}^{r} \iota_{\mathcal{F}_k}(\begin{pmatrix} C_k & C_k \end{pmatrix} x),\end{aligned} \tag{5.8}$$

where $0_{M_i \times N} \in \mathbb{R}^{M_i \times N}$ indicate all zeros matrices of appropriate dimensions. Next, we take the linear operators out of the indicator function such that we end up with sub-problems that are projections with closed-form solutions. Thereby we avoid the need for nesting iterative algorithms to solve sub-problems related to the indicator functions of constraint sets.

To separate indicator functions and linear operators, we introduce additional vectors y_i for $i \in \{1, 2, \ldots, p + q + r + 1\}$ of appropriate dimensions. From now on we use $s = p + q + r + 1$ to shorten notation. With the new variables, we rewrite problem formulation (5.8) and add linear equality constraints to obtain

$$\begin{aligned}\mathcal{P}_{\mathcal{M}}(m) = \arg\min_{\{y_i\},x} \frac{1}{2}\|y_s - m\|_2^2 + \sum_{i=1}^{p} \iota_{\mathcal{D}_i}(y_i) + \sum_{j=1}^{q} \iota_{\mathcal{E}_j}(y_j) \\ + \sum_{k=1}^{r} \iota_{\mathcal{F}_k}(y_k) \text{ s.t. } \tilde{A}x = \tilde{y},\end{aligned} \tag{5.9}$$

where

$$\tilde{A}x = \tilde{y} \Leftrightarrow \begin{pmatrix} A_1 & 0 \\ \vdots & 0 \\ A_p & 0 \\ 0 & B_1 \\ 0 & \vdots \\ 0 & B_q \\ C_1 & C_1 \\ \vdots & \vdots \\ C_r & C_r \\ I_N & I_N \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_p \\ y_{p+1} \\ \vdots \\ y_{p+q} \\ y_{p+q+1} \\ \vdots \\ y_{p+q+r} \\ y_{p+q+r+1} \end{pmatrix}.$$

Now define the new function

$$\tilde{f}(\tilde{y},m) \equiv \sum_{i=1}^{s} f_i(y_i,m) \equiv \frac{1}{2}\|y_s - m\|_2^2 + \sum_{i=1}^{p} \iota_{\mathcal{D}_i}(y_i) + \sum_{j=1}^{q} \iota_{\mathcal{E}_j}(y_j) + \sum_{k=1}^{r} \iota_{\mathcal{F}_k}(y_k), \tag{5.10}$$

such that we obtain the projection problem in the standard form

$$\mathcal{P}_M(m) = \arg\min_{x,\tilde{y}} \tilde{f}(\tilde{y},m) \text{ s.t. } \tilde{A}x = \tilde{y}. \tag{5.11}$$

If x and $\tilde{y}$ are a solution to this problem, the equality constraints enforce $u + v = y_{p+q+r+1}$ and we recover the projection of m as $y_{p+q+r+1}$ or as $\begin{pmatrix} I_N & I_N \end{pmatrix} x$. Now that Problem (5.11) is in a form that we can solve with the ADMM algorithm, we proceed by writing down the augmented Lagrangian for Problem (5.11) [Nocedal and Wright, 2000, Chapter 17] as

$$L_{\rho_1,\dots,\rho_s}(x, y_1, \dots, y_s, v_1, \dots, v_s) = \sum_{i=1}^{s} \left[f_i(y_i) + v_i^T (y_i - \tilde{A}_i x) + \frac{\rho_i}{2} \|y_i - \tilde{A}_i x\|_2^2 \right],$$

where $\rho_i > 0$ are the augmented Lagrangian penalty parameters and $v_i \in \mathbb{R}^{M_i}$ are the vectors of Lagrangian multipliers. We denote a block-row of the matrix $\tilde{A}$ as $\tilde{A}_i$. The relaxed ADMM iterations with relaxation parameters

$\gamma_i \in (0,2]$ and iteration counter l are given by

$$\begin{aligned}
x^{l+1} &= \arg\min_x \sum_{i=1}^s \frac{\rho_i^l}{2} \| y_i^l - \tilde{A}_i x + \frac{v_i^l}{\rho_i^l} \|_2^2 = \Big[\sum_{i=1}^s (\rho_i^l \tilde{A}_i^T \tilde{A}_i) \Big]^{-1} \sum_{i=1}^s \Big(\tilde{A}_i^T (\rho_i^l y_i^l + v_i^l) \Big) \\
\bar{x}_i^{l+1} &= \gamma_i^l \tilde{A}_i x_i^{l+1} + (1 - \gamma_i^l) y_i^l \\
y_i^{l+1} &= \arg\min_{y_i} \Big[f_i(y_i) + \frac{\rho_i^l}{2} \| y_i^l - \bar{x}_i^{l+1} + \frac{v_i^l}{\rho_i^l} \|_2^2 \Big] = \operatorname{prox}_{f_i, \rho_i} (\bar{x}_i^{l+1} - \frac{v_i^l}{\rho_i^l}) \\
v_i^{l+1} &= v_i^l + \rho_i^l (y_i^{l+1} - \bar{x}_i^{l+1}).
\end{aligned}$$

These iterations are equivalent to the Simultaneous Direction Method of Multipliers (SDMM, Combettes and Pesquet [2011]; Kitic et al. [2016]) and the SALSA algorithm [Afonso et al., 2011], except that we have an additional relaxation step. In fact, the iterations are identical to the algorithm presented in the previous chapter to compute the projection onto an intersection of sets, but here we solve a different problem and have different matrix structures. We briefly mention the main properties of each sub-problem.

x^{l+1} **computation**. This step is the solution of a large, sparse, square, symmetric, and positive-definite linear system. The system matrix has the following block structure:

$$Q \equiv \sum_{i=1}^s (\rho_i \tilde{A}_i^T \tilde{A}_i) = \\ \begin{pmatrix} \sum_{i=1}^p \rho_i A_i^T A_i + \sum_{k=1}^r \rho_k C_k^T C_k + \rho_s I_N & \sum_{k=1}^r \rho_k C_k^T C_k + \rho_s I_N \\ \sum_{k=1}^r \rho_k C_k^T C_k + \rho_s I_N & \sum_{j=1}^q \rho_j B_j^T B_j + \sum_{k=1}^r \rho_k C_k^T C_k + \rho_s I_N \end{pmatrix}. \tag{5.12}$$

This matrix is symmetric and positive-definite if $\tilde{A}$ has full column rank. We assume this is true in the remainder because many $\tilde{A}_i$ have full column rank, such as discrete-derivative based matrices and transform matrices including the DFT and various wavelets. We compute x^{l+1} with the conjugate gradient (CG) method, warm started by x^l as initial guess. We choose CG instead of an iterative method for least-squares problems such as LSQR [Paige and Saunders, 1982], because solvers for least-squares work with $\tilde{A}$ and $\tilde{A}^T$ separately and need to compute a matrix-vector product (MVP) with each $\tilde{A}_i$

and $\tilde{A}_i^T$ at every iteration. This becomes computationally expensive if there are many linear operators, as is the case for our problem. CG uses a single MVP with Q per iteration. The cost of this MVP does not increase if we add orthogonal matrices to $\tilde{A}$. If the matrices in $\tilde{A}$ have (partially) overlapping sparsity patterns, the cost also does not increase (much). We pre-compute all $\tilde{A}_i^T \tilde{A}_i$ for fast updating of Q when one or more of the ρ_i change (see below).

y_i^{l+1} **computation**. For every index i, we can compute $\operatorname{prox}_{f_i,\rho_i}(\bar{x}_i^{l+1} - \frac{v_i^l}{\rho_i^l})$ independently in parallel. For indices $i \in \{1, 2, \dots, s-1\}$, the proximal maps are projections onto sets $\mathcal{D}$, $\mathcal{E}$ or $\mathcal{F}$. These projections do not include linear operators and we know the solutions in closed from (e.g., ℓ_1-norm, ℓ_2-norm, rank, cardinality, bounds).

$\rho_i^{l+1}, \gamma_i^{l+1}$ **updates**. We use the updating scheme for ρ and γ from adaptive-relaxed ADMM, introduced by Xu et al. [2017b]. Numerical results show that this updating scheme accelerates the convergence of ADMM [Xu et al., 2017a,b,c], and is also robust when solving some non-convex problems [Xu et al., 2016]. We use a different relaxation and penalty parameter for each function $f_i(y_i)$, as do Song et al. [2016]; Xu et al. [2017c], which allows ρ_i and γ_i to adapt to the various linear operators of different dimensions that correspond to each constraint set.

Parallelism and communication. The only serial part of the algorithm defined in (5.12) is the x^{l+1} computation. We use multi-threaded MVPs in the compressed diagonal format if Q has a banded structure. The other parts of the iterations 5.12, y_i^{l+1}, v_i^{l+1},‘$\rho_i^{l+1}, \gamma_i^{l+1}$, are all independent so we can compute them in parallel for each index i. There are two operations in 5.12 that require communication between workers that carry out computations in parallel. We need to send x^{l+1} to every worker that computes a y_i^{l+1}, v_i^{l+1}, ρ_i^{l+1}, and γ_i^{l+1}. The second and last piece of communication is the map-reduce parallel sum to form the right-hand side for the next iteration when we compute $x^{l+1} = \sum_{i=1}^s \Big(\tilde{A}_i^T (\rho_i^l y_i^l + v_i^l) \Big)$.

In practice, we will use the proposed algorithm to solve problems that often involve non-convex sets. Therefore, we do not provide guarantees that algorithms like ADMM behave as expected, because their convergence proofs typically require closed, convex and proper functions, see, e.g., Boyd et al. [2011]; Eckstein and Yao [2015]. This is not a point of great concern to us, because the main motivation to base our algorithms on ADMM is rapid empirical convergence, ability to deal with many constraint sets efficiently, and strong empirical performance in case of non-convex sets that violate the standard assumptions for the convergence of ADMM.

5.4 Formulation of inverse problems with generalized Minkowski constraints

So far, we proposed a generalization of the Minkowski set ($\mathcal{M}$, equation 5.5), and developed an algorithm to compute projections onto this set. The next step to solve inverse problems where the generalized Minkowski set describes the prior knowledge is to combine the set $\mathcal{M}$ with a data-fitting procedure. We discuss two formulations of such an inverse problem. One is primarily suitable when the data-misfit function is computationally expensive to evaluate, which means we assume that evaluation of $f(m)$ and $\nabla_m f(m)$ is more time-consuming than projections onto the generalized Minkowski set $\mathcal{M}$. The second formulation is for inverse problems where the forward operator is both linear and computationally inexpensive to evaluate. We discuss the two approaches in more detail below.

5.4.1 Inverse problems with computationally expensive data-misfit evaluations

We consider a non-linear and possibly non-convex data-misfit function $f(m) : \mathbb{R}^N \rightarrow \mathbb{R}$ that depends on model parameters $m \in \mathbb{R}^N$. Our assumptions for this inverse problem formulation is that the computational budget allows for much fewer data-misfit evaluations than the required number of iterations to project onto the generalized Minkowski set, as defined in 5.12. We can deal with this imbalance by attempting to make as much progress towards minimizing $f(m)$, while always satisfying the constraints. The minimization of the data-misfit, subject to satisfying the generalized Minkowski constraint is then formulated as

$$\min_m f(m) \text{ s.t. } m \in \mathcal{M}. \tag{5.13}$$

If we solve this problem with algorithms that use a projection onto $\mathcal{M}$ at every iteration, the model parameters m satisfy the constraints at every iteration; a property desired by several works in non-convex geophysical parameter estimation, see [Smithyman et al., 2015, Esser et al., 2016, Yong et al., 2018], and the geophysical examples presented in the previous chapters. These works obtain better model reconstructions from non-convex problems by carefully changing the constraints during the data-fitting procedure. The first two numerical experiments in this work use the spectral projected gradient algorithm (SPG, Birgin et al. [1999]; Birgin et al. [2003]). SPG iterates

$$m^{l+1} = (1-\gamma)m^l - \gamma \mathcal{P}_{\mathcal{M}}(m^l - \alpha \nabla_m f(m^l)), \tag{5.14}$$

where $\mathcal{P}_{\mathcal{M}}$ is the Euclidean projection onto $\mathcal{M}$. The Barzilai-Borwein [Barzilai and Borwein, 1988] step-length $\alpha > 0$ is a scalar approximation of the Hessian that is informed by previous model estimates and gradients of $f(m)$. A non-monotone line-search estimates the scalar $\gamma \in (0, 1]$ and prevents $f(m)$ from increasing too many iterations in sequence. The line-search backtracks between two points in a convex set if $\mathcal{M}$ is convex and the initial m_0 is feasible, so every line-search iterate is feasible by construction. SPG thus requires a single projection onto $\mathcal{M}$ if all constraint sets are convex.

5.4.2 Linear inverse problems with computationally cheap forward operators

Contrary to the previous section, we now assume a linear relation between the model parameters $m \in \mathbb{R}^N$ and the observed data, $d_{\text{obs}} \in \mathbb{R}^M$. The second assumption, for the problem formulation in this section, is that the evaluation of the linear forward operator is not much more time consuming than other computational components in the iterations from 5.12. Examples of such operators $G \in \mathbb{R}^{M \times N}$ are masks, identity matrices, and blurring kernels. We may then put data-fitting and regularization on the same footing and formulate an inverse problem with constraints as a feasibility or projection problem. Both these formulations add a data-fit constraint to the constraints that describe model properties [Youla and Webb, 1982, Trussell and Civanlar, 1984, Combettes, 1993, 1996]. The numerical examples in this work use the point-wise data-fit constraint: $\mathcal{G}^{\text{data}} \equiv \{m \mid l[i] \leq (Gm - d_{\text{obs}})[i] \leq u[i]\}$ with lower and upper bounds on the misfit. We use the notation $l[i]$ for entry i of the lower-bound vector l. The data-fit constraint can be any set onto which we know how to project. An example of a global data-misfit constraint is the norm-based set $\mathcal{G}^{\text{data}} \equiv \{m \mid \sigma_l \leq \|Gm - d_{\text{obs}}\| \leq \sigma_u\}$ with scalar bounds $\sigma_l < \sigma_u$. This set is non-convex if $\sigma_l > 0$, e.g., the annulus constraint in case of the ℓ_2 norm. This set has a 'hole' in the interior of the set that explicitly avoids fitting the data noise in ℓ_2 norm sense.

We denote our formulation of a linear inverse problem with a data-fit constraint, and a generalized Minkowski set constraint (Equation 5.5) on the model estimate as

$$\min_{x,u,v} \frac{1}{2}\|x - m\|_2^2 \quad \text{s.t.} \quad \begin{cases} x = u + v \\ u \in \bigcap_{i=1}^{p} \mathcal{D}_i, v \in \bigcap_{i=1}^{q} \mathcal{E}_j, x \in \bigcap_{i=1}^{r} \mathcal{F}_k \\ x \in \mathcal{G}^{\text{data}} \end{cases} . \tag{5.15}$$

The solution is the projection of an initial guess, m, onto the intersection of a data-fit constraint and a generalized Minkowski constraint on the model parameters. As before, there are constraints on the model x, as well as the components u and v. Problem 5.15 is the same as before in Equation (5.5) and we can solve it with the algorithm from the previous section. In the current case, we have one additional constraint on the sum of the components.

5.5 Numerical examples

5.5.1 Seismic full-waveform inversion 1

We start with a numerical example originally presented in Chapter 4. We repeat the experiment and show how a Minkowski set describes the provided prior knowledge naturally and results in a better model estimate compared to a single constraint set or intersection of multiple sets. The problem is to estimate the acoustic velocity $m \in \mathbb{R}^N$ of the model in Figure 5.1, from observed seismic data modeled by the Helmholtz equation. This problem, known as full-waveform inversion (FWI, Tarantola [1986]; Pratt et al. [1998]; Virieux and Operto [2009]), is often formulated as the minimization of a differentiable, but non-convex data-fit

$$f(m) = \frac{1}{2} \| d_{\text{predicted}}(m) - d_{\text{observed}} \|_2^2, \tag{5.16}$$

where the partial-differential-equation constraints are already eliminated and are part of $d_{\text{predicted}}(m)$, see, e.g., Haber et al. [2000]. The observed data, d_{observed} are discrete frequencies of $\{3.0, 6.0, 9.0\}$ Hertz.

Figure 5.1 shows the true model, initial guess for m, and the source and receiver geometry. We assume prior information about the bounds on the parameter values, and that the anomaly has a rectangular shape with a lower velocity than the background.

The results in figure 5.1 using bounds or bounds and the true anisotropic total-variation (TV) as a constraint, do not lead to a satisfying model estimate. The result with TV is marginally better compared to bound constraints only. The diagonally shaped model estimates are mostly due to the source and receiver positioning, known as vertical seismic profiling (VSP) in geophysics. To obtain a better model, we used a variety of intersections including non-convex sets in Chapter 4.

Here we will show that the generalized Minkowski set $\mathcal{M}$ (Equation 5.5) can also provide an improved model estimate, but using convex sets only. If we have the prior knowledge that the anomaly we need to find has a lower velocity than the background medium, we can easily and naturally include this information as a Minkowski set. The following four sets summarize our prior knowledge:

1. $\mathcal{F}_1 = \{x \mid 2350 \leq x[i] \leq 2550\}$: bounds on sum
2. $\mathcal{F}_2 = \{x \mid \|((D_z \otimes I_x)^T \;\; (I_z \otimes D_x)^T)^T x\|_1 \leq \sigma\}$: anisotropic total-variation on sum
3. $\mathcal{D}_1 = \{u \mid -150 \leq u[i] \leq 0\}$: bounds on anomaly
4. $\mathcal{E}_1 = \{v \mid v[i] = 2500\}$: bounds on background

The generalized Minkowski set combines the four above sets as $(\mathcal{F}_1 \bigcap \mathcal{F}_2) \bigcap (\mathcal{D}_1 + \mathcal{E}_1)$. In words, we fix the background velocity, require any anomaly to be negative, and the total model estimate has to satisfy bound constraints and have a low anisotropic total-variation. To minimize the data-misfit subject to the generalized Minkowski constraint,

$$\min_{m} \frac{1}{2}\|d_{\text{predicted}}(m) - d_{\text{observed}}\|_2^2 \text{ s.t. } m \in (\mathcal{F}_1 \bigcap \mathcal{F}_2) \bigcap (\mathcal{D}_1 + \mathcal{E}_1), \quad (5.17)$$

we use the same algorithm as the original example in Chapter 4, which is the spectral projected gradient (SPG, Birgin et al. [1999]) algorithm with 15 iterations and a non-monotone line search with a memory of five function values. The result that uses the generalized Minkowski constraint (Figure 5.1) is much better compared to bounds and the correct total-variation because the constraints on the sign of the anomaly prevent incorrect high-velocity artifacts.

While there are other ways to fix a background model and invert for an anomaly, this example illustrates that our proposed regularization approach incorporates information on the sign of an anomaly conveniently and the constraints remain convex. It is straightforward to change and add constraints on each component, also in the more realistic situation that the background is not known and should not be fixed, as we show in the following example.

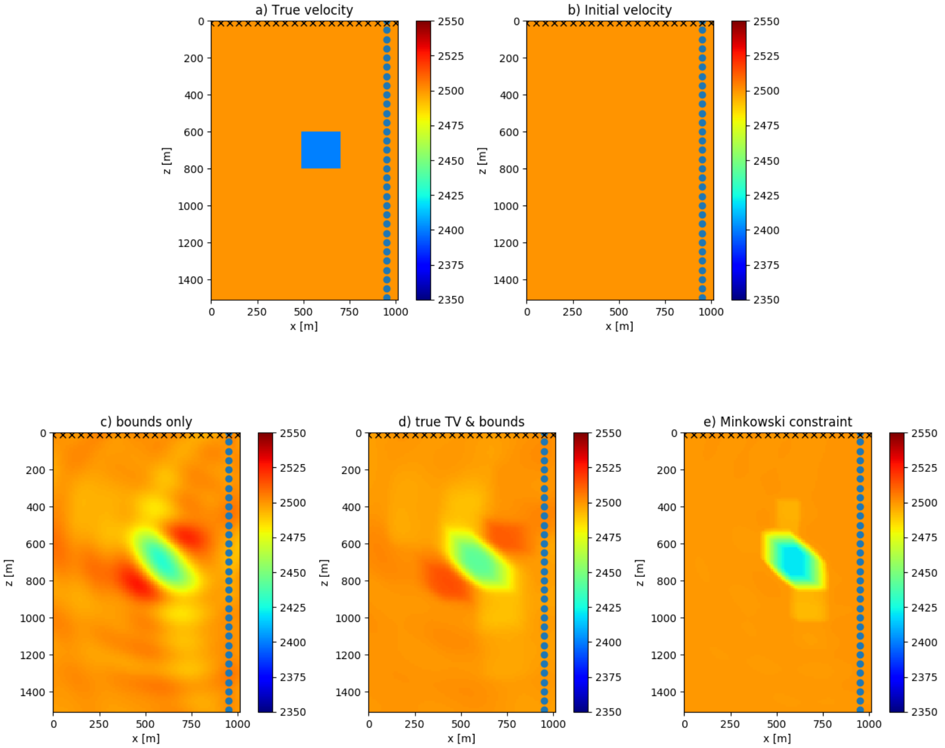


Figure 5.1: The true model for the data generation for the full-waveform inversion 1 example, the initial guess for parameter estimation, and the model estimates with various constraints. Crosses and circles indicate receivers and sources, respectively.

5.5.2 Seismic full-waveform inversion 2

This time, the challenge is to estimate a model (Figure 5.2 a) that has both a background and an anomaly component that are very different from the initial guess (Figure 5.2 b). This means we can no longer fix one of the two components of the generalized Minkowski sum.

The experimental setting is a bit different from the previous example. The sources are in one borehole, the receivers in another borehole at the other side of the model (cross-well full-waveform inversion). Except for a single high-contrast anomaly, the velocity is increasing monotonically, both gradually and discontinuously. The prior knowledge we assume consists of *i)* upper and lower bounds on the velocity and also on the anomaly *ii)* the model is relatively simple in the sense that we assume it has a rank of at most five *iii)* the background parameters are increasing monotonically with depth *iv)* the background is varying smoothly in the lateral direction *v)* the size of the anomaly is not larger than one fifth of the height of the model and not larger than one third of the width of the model. We do not assume prior information on the total-variation of the model, but for comparison, we show the result when we use the true total-variation as a constraint. The following sets formalize the aforementioned prior knowledge:

1. $\mathcal{F}_1 = \{x \mid 2350 \leq x[i] \leq 2850\}$
2. $\mathcal{F}_2 = \{x \mid \|((I_x \otimes D_z)^T \; (D_x \otimes I_z)^T)^T x\|_1 \leq \sigma\}$
3. $\mathcal{F}_3 = \{x \mid \operatorname{rank}(x) \leq 5\}$
4. $\mathcal{D}_1 = \{x \mid 2350 \leq x[i] \leq 2850\}$
5. $\mathcal{D}_2 = \{u \mid 0 \leq (D_z \otimes I_x)u \leq \infty\}$
6. $\mathcal{D}_3 = \{u \mid -0.1[m/s]/m \leq (I_z \otimes D_x)u \leq 0.1[m/s]/m\}$
7. $\mathcal{E}_1 = \{v \mid 300 \leq v[i] \leq 350\}$
8. $\mathcal{E}_2 = \{v \mid \operatorname{card}(v) \leq (n_z/5 \times n_x/3)\}$

As before, the sets $\mathcal{F}_k$ act on the sum of components, $\mathcal{D}_i$ describe component one (background), and $\mathcal{E}_j$ constrain the other component (anomaly). Figure 5.2 c show the model m found by SPG applied to the problem $\min_m f(m)$ s.t. $m \in \mathcal{F}_1$. We see oscillatory features in the result with bound constraints only, but the main issue is the appearance of a low-velocity artifact, located just below the true anomaly. Figure 5.2 d shows that even if we know the correct total-variation, the result is less oscillatory than using just bound constraints, but still shows an erroneous low-velocity anomaly. When we also include the rank constraint, i.e., we use the set $\mathcal{F}_1 \bigcap \mathcal{F}_2 \bigcap \mathcal{F}_3$, the result does not improve (Figure 5.2 e). The generalized Minkowski set $\big(\bigcap_{k=1}^3 \mathcal{F}_k\big) \bigcap \big(\bigcap_{i=1}^3 \mathcal{D}_i + \bigcap_{j=1}^2 \mathcal{E}_j\big)$ does not yield a result with the large incorrect low-velocity artifact just below the correct high-velocity anomaly (5.2 g), even though we did not include information on the sign of the anomaly

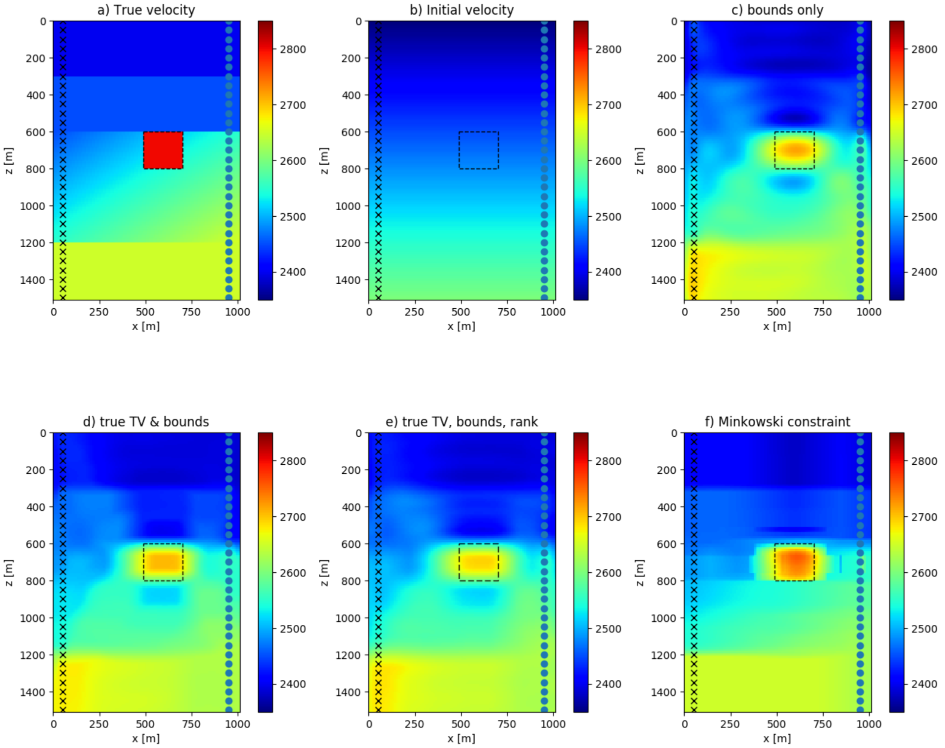


Figure 5.2: The true and initial models corresponding to the full-waveform inversion 2 example. Figure shows parameter estimation results with various intersections of sets, as well as the result using a generalized Minkowski constraint set. Only the result obtained with the generalized Minkowski set does not show an incorrect low-velocity anomaly.

as we did in the previous example. There are still two smaller horizontal and vertical artifacts. Overall, the Minkowski set based constraint results in the best background and anomaly estimation.

This example shows that the generalized Minkowski set allows for inclusion of prior knowledge on the two (or more) different components, as well as their sum. The results show that this leads to improved model estimates if prior knowledge is available on both the components and the sum. Information that we may have about a background or anomaly is often difficult or impossible to include in an inverse problem as a single constraint set or intersection of multiple sets, but easy to include in the summation structure of the generalized Minkowski set. In many practical problems, we do have some information about an anomaly. When looking for air or water filled voids and tunnels in engineering or archeological geophysics, we know that the acoustic wave propagation velocity is usually lower than the background and we also have at least a rough idea about the size of the anomaly. In seismic hydrocarbon exploration, there are high-contrast salt structures in the subsurface, almost always with higher acoustic velocity than the surrounding geology.

5.5.3 Video processing

Background-anomaly separation is a common problem in video processing. A particular example is security camera video, $T \in \mathbb{R}^{n_x \times n_y \times n_t}$, where x and y are the two spatial coordinates and t is the time. The separation problem is often used to illustrate robust principal component analysis (RPCA), and related convex and non-convex formulations of sparse + low-rank decomposition algorithms [e.g., Candès et al., 2011, Netrapalli et al., 2014, Kang et al., 2015, Driggs et al., 2017].

In this example, we show that the generalized Minkowski set for an inverse problem, proposed in Equation (5.15), is also suitable for background-anomaly separation in image and video processing, and illustrate the advantages of working with a constrained formulation, as opposed to the more common penalty formulation. To include multiple pieces of prior knowledge, we choose to work with the video in tensor format and use the flexibility of our regularization framework to impose constraints on the tensor, as well as on individual slices and fibers. This is different from RPCA approaches that matricize or flatten the video tensor to a matrix of size $n_x n_y \times n_t$, such that each column of the matrix is a vectorized time-slice [Candès et al., 2011, Netrapalli et al., 2014, Kang et al., 2015], and also differs from tensor-based RPCA methods that work with a tensor only [Zhang et al., 2014, Wang and

Navasca, 2015]. Contrary to many sparse + low-rank decomposition algorithms, our set-based framework is not tied to any specific constraint, and we can mix various constraints for the two components and obtain multiple background-anomaly separation algorithms.

Beyond the basic decomposition problem, the escalator video comes with some additional challenges. There is a dynamic background component (the escalators steps) and there are reflections of people in the glass that are weak anomalies and duplicates of persons. The video contains noise and part of the background pixel intensity changes significantly (55 on a $0 - 255$ grayscale) over time. We subtract the mean of each frame as a pre-processing step to mitigate the change in intensity. Below we describe simple methods to derive prior knowledge for the video, as well as for the background and anomaly component.

constraint sets for background We use the last 20 time frames to derive constraints for the background because these frames do not contain people. From these frames, we use the minimum and maximum value for each pixel over time as the bounds for the background component, denoted as set $\mathcal{D}_1$. The second constraint is the subspace spanned by the last 20 frames. We require that each time frame of the background be a linear combination of the training frames organized as a matrix $S \in \mathbb{R}^{n_x n_y \times 20}$, where each column is a vectorized video frame of T. We denote this constraint as $\mathcal{D}_2 = \{u \mid u = Sc, c \in \mathbb{R}^{20}\}$, with coefficient vector c, which we obtain during the projection operation: $\mathcal{P}_{\mathcal{D}_2}(u) = S(S^T S)^{-1} S^T u$. After computing the singular value decomposition $S = U \Sigma V^T$, the projection simplifies to $\mathcal{P}_{\mathcal{D}_2}(u) = U U^T u$

constraint sets for sum of components We constraint the sum of the background and anomaly components to the interval of grayscale values $[0 - 255]$ minus the mean of each time-frame, denoted as set $\mathcal{F}_1$.

constraint sets for anomaly We also have bound constraints, set $\mathcal{E}_1$, on the anomaly component that we define as the bounds on the sum minus the bounds on the background. To enhance the quality of the anomaly component, we add various types of sparsity constraints. If we would have some example video available like we have for the background component, we could observe properties of the anomaly, i.e., how many pixels are typically anomalies (people). As the escalator video is only 200 time-frames long, we instead use some rough estimates of the anomaly properties to define three non-convex constraint sets. We choose to apply constraints to each time-slice separately because this makes it easier to convert basic observations or intuition into a set. The first type of sparsity constraint is the set $\mathcal{E}_2 = \{T \mid \operatorname{card}(T_{\Omega_i}) \leq (n_x/4 \times n_y/4) \; \forall i \in \{1, 2, \dots, n_t\}\}$ where T_{Ω_i} is a time slice of the video tensor. This constraint limits the number of anomaly pixels in each frame to 1/16 of the total number of pixels in each time slice. The second and third constraint sets are limits on the vertical and horizontal derivative of each time-frame image separately. If we assume the prior knowledge that there are no more than ten persons in the video at each time, we can use $\mathcal{E}_3 = \{T \mid \operatorname{card}((I_x \otimes D_y) \operatorname{vec}(T_{\Omega_i})) \leq 480, \; i \in \{1, 2, \dots, n_t\}\}$, based on the rough estimate of 10 persons × 12 pixels wide × 4 boundaries (the four vertical boundaries are background - head - upper body - legs - background). Similarly for the horizontal direction, we define $\mathcal{E}_4 = \{T \mid \operatorname{card}((D_x \otimes I_y) \operatorname{vec}(T_{\Omega_i})) \leq 440, \; i \in \{1, 2, \dots, n_t\}\}$, based on the estimate of 10 persons × 22 pixels tall × 2 boundaries (the horizontal boundaries are background - person - background).

Putting it all together, we project the video onto the generalized Minkowski set defined in (5.6), i.e., we solve

$$\min_x \frac{1}{2} \|x - \operatorname{vec}(T)\|_2^2 \text{ s.t. } x \in \mathcal{F}_1 \bigcap \Big(\bigcap_{i=1}^{2} \mathcal{D}_i + \bigcap_{j=1}^{4} \mathcal{E}_j \Big) \tag{5.18}$$

using the iterations derived in Equation 5.12. Our formulation implies that the projection of a vector is always the sum of the two components, but this does not mean that x is equal to T at the solution, because we did not include a constraint on x that says we need to fit the data accurately. We did not use a data-fit constraint because it is not evident how tight we want to fit the data or how much noise there is. By computing the projection of the original video, we still include a sense of proximity to the observed data.

The result of the generalized Minkowski decomposition of the video

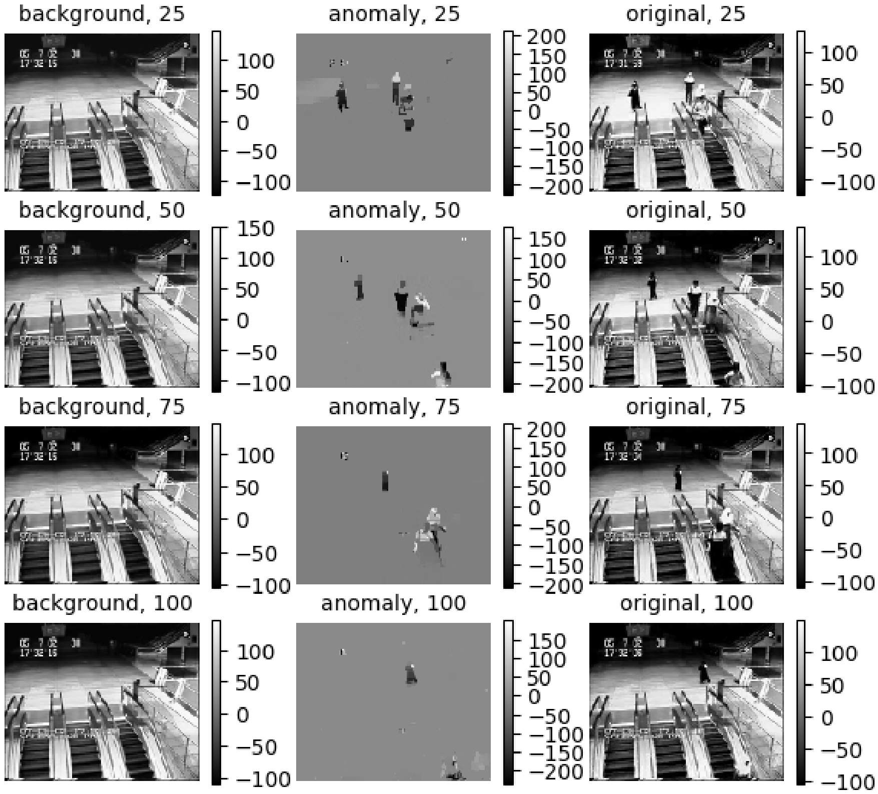


Figure 5.3: Results of the generalized Minkowski decomposition applied to the escalator video. The figure shows four frames. The most pronounced artifacts are in the time stamp. This example illustrates that the constrained approach is suitable to observe and apply constraint properties obtained from a few frames of background only video.

shown in Figure (5.3), is visually better than the six methods compared by Driggs et al. [2017]. The compared results often show blurring of the escalator steps in the estimated background, especially when a person is on the escalator. Several results also show visible escalator structure in the anomaly estimation. Our simple approach does not suffer from these two problems. We do not need to estimate any penalty or trade-off parameters, but rely on constraint sets whose parameters we can observe directly or estimate from a few training frames. We were able to conveniently mix constraints on slices and fibers of the tensor by working with the constrained formulation.

5.6 Discussion

So far, we described the concepts and algorithms for the case of a Minkowski sum with only two components. Our approach can handle more than two components, but the linear systems in Equation (5.12) will become larger. A better solver than plain conjugate-gradients can mitigate increased solutions times due to larger linear systems, possibly by taking the block structure into account.

A Minkowski sum of more than two components can also make it less intuitive what type of solutions are in the Minkowski sum of sets. We can regain some intuition about the generalized Minkowski set by looking at sampled elements from the set. Samples are simply obtained by projecting vectors (possible solutions, reference models, random vectors, . . .) onto the target set.

All numerical examples were set up to illustrate how we use the generalized Minkowski set for the regularization of inverse problems, given multiple pieces of prior knowledge on the two components of a model, as well as prior information on the sum of the components. Application evaluation of the proposed regularization approach to more realistic examples, as in chapter 2 and 3, is left for future work.

5.7 Conclusions

Inverse problems for physical parameter estimation and image and video processing often encounter model parameters with complex structure, so that it is difficult to describe the expected model parameters with a single set or intersection of multiple sets. In these situations, it may be easier to work with an additive model description where the model is the sum of morphologically distinct components.

We presented a regularization framework for inverse problems with the Minkowski set at its core. The additive structure of the Minkowski set allows us to enforce prior knowledge in the form of separate constraints on each of the model components. In that sense, our work differs from current approaches that rely on additive penalties for each component. As a result, we no longer need to introduce problematic trade-off parameters.

Unfortunately, the Minkowski set by itself is not versatile enough for physical parameter estimation because we also need to enforce bound constraints and other prior knowledge on the sum of the two components to ensure physical feasibility. Moreover, we would like to use more than one constraint per component to incorporate all prior knowledge that we may have available.

To deal with this situation, we proposed a generalization of the Minkowski set by defining it as the intersection of a Minkowski set with another constraint set on the sum of the components. With this construction, we can enforce multiple constraints on the model parameters, as well as multiple constraints on each component.

To solve inverse problems with these constraints, we discuss how to project onto generalized Minkowski sets based on the alternating direction method of multipliers. The projection enables projected-gradient based method to minimize nonlinear functions subject to constraints. We also showed that for linear inverse problems, the linear forward operator fits in the projection computation directly as a data-fitting constraint. This makes the inversion faster if the application of the forward operator does not take much time.

Numerical examples show how the generalized Minkowski set helps to solve non-convex seismic parameter estimation problems and a background-anomaly separation task in video processing, given prior knowledge on the model parameters, as well as on the components.

Chapter 6

Discussion

In each of the chapters, I discussed and developed formulations and computational methods for inverse problems from a constrained optimization point of view. So far, I did not discuss the statistical and Bayesian [e.g, Tarantola, 2005] interpretation much.

The aim of this thesis is not to take a side in the frequentist versus Bayesian debate [Scales and Snieder, 1997, Stark, 2015]. Nor did I study the relation and connections between the two approaches to inverse problems. Constraints became the core of this work because we found them easy to use beneficial when working with seismic field data [Smithyman et al., 2015]. Rather than focus on the differences between constraints and prior distributions [see Backus, 1988, Scales and Snieder, 1997, Stark, 2015], I would like to relate this thesis to the Bayesian approach by discussing and developing similarities that may help bridge the conceptual gap between the two approaches.

First, constraints can also incorporate statistical prior knowledge. For instance, we can constrain the model parameters to have the same histogram or correlations as an example image, possibly in a transform-domain [e.g., Portilla and Simoncelli, 2000, Peyré, 2009, Fadili and Peyre, 2011, Mei et al., 2015]. This is common in the field of texture synthesis, and I used similar ideas in chapter 4 to obtain and include prior information from examples to image processing inverse problems.

The second conceptual similarity between Bayesian and constrained approaches is the notion of prior and posterior distributions where both the prior and data observations influence the latter. The constrained formulation gives rise to a similar structure. Based on a problem formulation where we have a data-fit constraint and multiple constraints that describe prior knowledge ($\mathcal{V}_i$), we may consider the *prior information set*, $\bigcap_{i=1}^p \mathcal{V}_i$, somewhat analogous to a prior probability distribution. Samples from $\bigcap_{i=1}^p \mathcal{V}_i$ are easy to obtain by solving feasibility or projection problems, starting at random points or examples of solutions of similar inverse problems. Each sample, s_j^{prior} is an element of the intersection of constraint sets that describe model properties: $s_j^{\text{prior}} \in \bigcap_{i=1}^p \mathcal{V}_i$. If the intersection is convex, we can construct more samples as convex combinations: $\gamma_1 s_1^{\text{prior}} + \gamma_2 s_2^{\text{prior}} + \cdots$ with $\gamma_1 + \gamma_2 + \cdots = 1$ and $\gamma_1 \geq 0, \gamma_2 \geq 0, \cdots$.

Analogous to the posterior distribution, we can look at the intersection of the constraint sets that describe model properties with the data-fit constraint set,

$$\bigcap_{i=1}^{p} \mathcal{V}_i \; \bigcap \mathcal{V}_{\text{data}}. \tag{6.1}$$

The resulting set contains all models that satisfy the prior knowledge as well as the observed data. This collection of models can provide us with a sense of uncertainty/spread in the model estimate.

To provide some intuition about the statements in this section, we visualize using classical image inpainting. This is essentially a higher dimensional version of the geometrical example in the introduction. Note that the solutions of the image and video processing examples in chapters 4 and 5 were also points in a set defined as 6.1.

The true image and observed data are shown in Figure (6.1). The true image is a simple texture, the observed data are vertical bands, which spread out more and more from left to right, so the number of missing pixels increases from left to right.

We will use the three constraint sets to describe prior knowledge that were already used in chapter 3 to describe typical acoustic velocity models

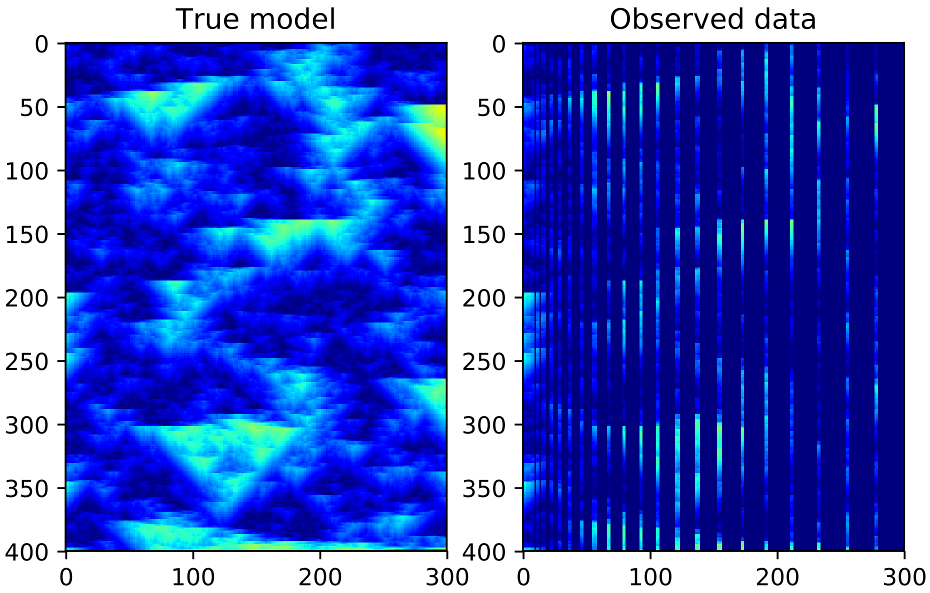


Figure 6.1: The true image (left), and the observed data (right) that consists of vertical bands of the true image, increasingly sparsely sampled from left to right.

in sedimentary geological settings. This means we enforce *1)* bound constraints; *2)* lateral smoothness; *3)* with depth (going from top to bottom) parameter values can increase arbitrarily fast, but can only decrease slowly.

To generate samples from the intersection of prior knowledge, we project models filled with random numbers onto the intersection. Figure (6.2) shows three samples.

Figure (6.3) displays three samples from the intersection of sets that contain data information and prior knowledge. The data constraints are bounds to match the observed data. The samples from this intersection are the projections of the samples from the prior information sets as shown in Figure (6.1). Also in Figure(6.2), we show the difference between the true image and the samples from the intersection of prior and data information. Because of the bound constraints at the observations, there is almost no error at the data locations, which leads to the striped pattern in both the

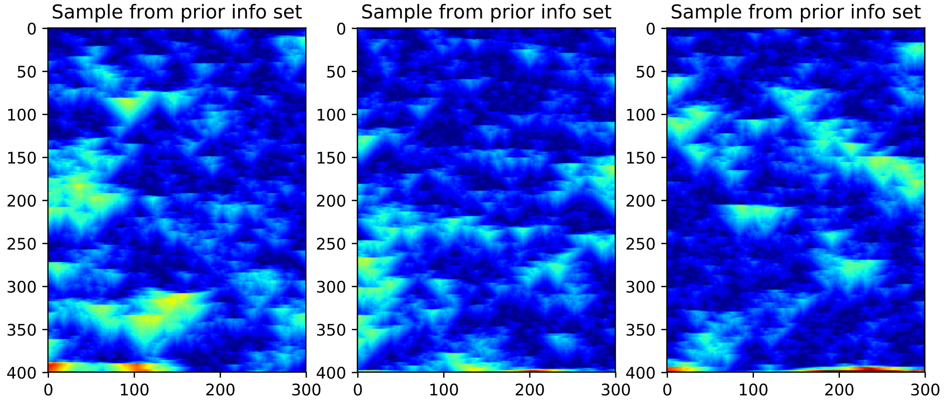


Figure 6.2: Three samples from the prior information set, which is the intersection of bounds, lateral smoothness, and parameter values that are limited to decrease slowly in the downward direction. Samples are the result of projecting random images onto the intersection.

samples and the error.

As all three models satisfy all our prior knowledge and data information, there is no way to tell which one is the ‘best’ result. A simple way to analyze the results, not free of caveats, is to look at the point-wise maximum, minimum, and difference of the three model estimates. In Figure (6.4) we display these derived quantities. Comparing Figures (6.3) and (6.4) shows that the areas in the model with large variation between maximum and minimum also correspond to areas with large error. Furthermore, we observe, as expected, that the error and spread generally increase with decreasing observation density. The multiple models provide some quantitative insight.

This example provides some intuition about sets that describe prior knowledge and data constraints. Even if it is not obvious what happens if we take the intersection of multiple sets, random samples visualize what type of models are in the intersection. Besides random samples, we can also manually construct prior samples, project models expected to be similar to the true model, and take convex combinations of prior samples to generate additional insight quickly.

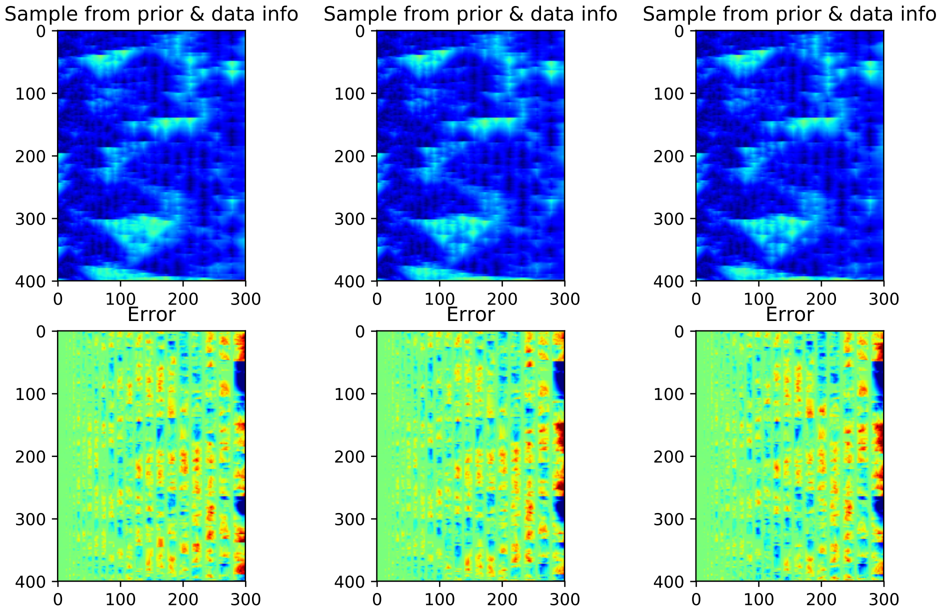


Figure 6.3: Samples from the intersection of sets that describe prior knowledge and data observations. The bottom row shows the difference between the sample from the top row and the true model from Figure (6.1).

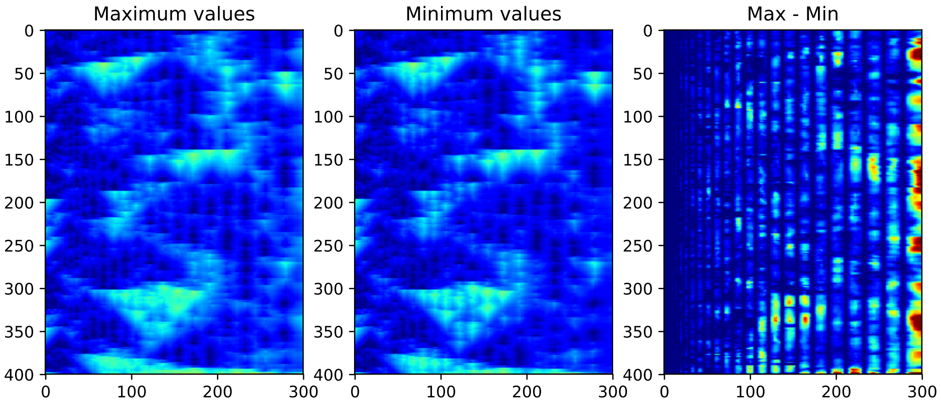


Figure 6.4: Pointwise maximum and minimum values, as well as the difference of the three samples from Figure (6.3).

Chapter 7

Conclusions

Inverse problems in the imaging sciences range from linear inverse problems such as cleaning and reconstructing images, to partial-differential-equation based geophysical parameter estimation where the data relates nonlinearly to the model parameters. Solving a problem in either of these two categories requires prior information (regularization) on the model parameters to achieve state-of-art results. Regularization combats issues introduced by data deficiencies and inherent nonuniqueness of the solutions of an inverse problem. While we apply regularization based on what we expect from the final estimate, in challenging non-convex inverse problems like full-waveform inversion we also greatly benefit from applying (possibly different) regularization to the intermediate results at every iteration. Such a procedure can prevent the model estimates from becoming physically/geologically unrealistic, which halts progress towards the correct model parameters in later iterations.

In this thesis, I proposed contributions to various aspects of solving inverse problems. This includes problem formulations, how to work with multiple pieces of prior knowledge, algorithms, software design, practical high-performance implementation, applications in image and video processing, and specific solutions strategies for seismic full-waveform inversion.

In the following, I summarize the conclusions per topic:

Inverse problem formulations. In chapter 2 and 3, I motivate why

I prefer to work with multiple constraints instead of multiple penalty functions for the regularization of non-convex seismic full-waveform inversion. In the first four chapters, I regularize using an intersection of convex and non-convex sets. I present and discuss a few main arguments in favor of constraints: *i)* no need to select multiple scalar penalty parameters because each constraint is imposed independently of the others; *ii)* some constraints are set directly in terms of physical quantities; *iii)* I show that the solution of seismic full-waveform inversion behaves predictably as a function of constraint 'size', but less predictable when we vary trade-off parameters in penalty formulations; *(iv)* constraints in combination with projections offer guarantees that the model parameters remain in the constraint set at every iteration. For non-convex problems, this can help avoid local minima when the constraints are relaxed gradually. I demonstrate various successful applications with different combinations of constraints using this strategy.

My primary contribution to advocating intersections of sets is the specific application to full-waveform inversion in combination with controlling the properties of intermediate model estimates. In chapter 4, I also show that simple machine learning can provide us with many (≥ 10) pieces of prior information, that serve as constraints using an intersection of sets, something that would be more complicated if not impossible in case of multiple penalty functions where we need to balance the influence of ten or more penalties.

In chapter 5, I introduced a new problem formulation that merges and extends previous work on intersections of constraints sets and additive model structures such as cartoon-texture decomposition, morphological component analysis, robust principal component analysis, and multi-scale image descriptions. Additive descriptions of model parameters add two or more components to generate an image. This separation makes it easier to include prior information when it is difficult to describe all model parameters using a single property, i.e., when the model contains morphologically distinct components. The constrained version of an additive model that I propose is based on the Minkowski set, or vector sum of sets.

I showed that this set by itself is of limited use for the regularization of inverse problems, because we want, and need, constraints on the sum of the components as well. Moreover, motivated by the examples in chapter 2 and 3, I also want to include multiple pieces of prior knowledge on each component. I proposed to generalize the Minkowski set by allowing each of the two components to be an intersection themselves, and also enforce an intersection of constraints on the sum of the components.

In summary, the model is an element of an intersection of a sum of intersections and another intersection. The extensions to a Minkowski set that I introduced, make it easier to include more pieces of prior knowledge. Numerical examples in video segmentation and seismic full-waveform inversion illustrate this benefit.

Algorithms. My contributions to the algorithmic side of regularization via intersections of sets split up between chapters 2 & 3 on the one hand and chapters 4 & 5 on the other hand. In chapter 2 and 3, I combine three existing algorithms to create an easy to use and versatile workflow that adds multiple constraint sets to an inverse problem. The target problems for the algorithms in chapters 2 and 3 are partial-differential-equation based parameter estimation, particularly seismic waveform inversion. The philosophy of this framework is to split the complicated problem, minimization of a non-convex data-fit function subject to multiple constraints, into simpler and simpler computational pieces until we can solve each part easily and in closed form. Starting from the top, I use the spectral projected gradient algorithm (SPG) to create separate data-fitting and feasibility problems. We ensure feasibility at every SPG iteration by projecting onto the intersection of constraint sets using Dykstra's algorithm. Whenever one of Dykstra's sub-problems, a projection onto a single set is not known in closed form, I use the alternating direction method of multipliers (ADMM) to compute it. This framework is the first effort to add an arbitrary intersection constraint to seismic full-waveform inversion.

In chapter 4, I merge the functionality of Dykstra's algorithm and ADMM to compute projections onto an intersection of sets. Nesting ADMM inside Dykstra's algorithm does not exploit possible similarity between sets and requires stopping conditions such that both algorithms operate together efficiently. Therefore, I developed a new algorithm for computing projections onto the intersection of multiple constraint sets specifically. Whereas Dykstra's algorithm treats every projection onto a single set as a black box, I focus on efficient treatment of sets that include non-orthogonal linear operators in their definitions. Numerical examples show that this approach is much faster because I formulated the problem such that it takes similarity between the linear operators into account. The proposed algorithm achieves good empirical performance on problems with non-convex sets by using a multilevel coarse-to-fine grid continuation and an automatic selection scheme for the augmented-Lagrangian penalty parameters that occur in ADMM-based solvers. The algorithms apply to problems defined on small 2D and large 3D grids ($\approx 400^3$), by virtue of solving all sub-problems in parallel, automatic selection methods for augmented-Lagrangian and relaxation parameters, multilevel acceleration, solving sparse and banded linear systems using multi-threaded matrix-vector products in the compressed diagonal storage format, and careful implementation of the proposed algorithms in Julia. Examples show that the proposed algorithm enabled developing regularization strategies using many different constraint sets for both small and large-scale inverse problems.

In chapter 5 there are no new algorithms, but I show that the algorithms from chapter 4 apply to more than just intersections of sets. I reformulate the projection onto the extended Minkowski set such that it can use the algorithms in chapter 4. The primary computational difference between projections onto intersections of sets and the extended Minkowski set is that certain linear operators become larger block-structured linear operators in the sums of sets scenario.

Software and implementation. All algorithms presented in chapter 4 for the computation of projections onto intersections of sets are part of a software package that I developed: `SetIntersectionProjection`. This functionality of the package serves as the projection step inside a projected gradient-based algorithm, or it can directly solve an inverse problem stated as a projection or feasibility problem. There are a few reasons why I implemented everything in Julia. First, all code uses parametric typing such that everything works in single and double precision, without any code modifications or copying parts of the code with minor modifications. A second argument in favor of Julia is the convenient implementation of coarse and fine-grained parallelism. I used coarse-scale parallelism to compute subproblems of the algorithms from chapter 4 in parallel. Each of these subproblems is then also solved in parallel, using either Julia threads or standard multithreaded libraries for linear algebra and Fourier-transform based operations. Another simple trick that speeds up the computations, at the cost of some increased peak memory usage, is keeping all vector-valued quantities in memory and overwrite them in-place, thereby avoiding time-consuming memory re-allocation. The numerical examples showed that the combination of problem formulation, algorithms, and implementation make the software package suitable to quickly test various combinations of constraints for a range of small and large-scale inverse problems.

Besides the algorithms, I also included scripts that set up linear operators and projectors onto simple sets. These two building blocks are the input for the software to compute the projection onto the intersection of sets. The modular software design still allows users to work with their custom linear operators and projectors, as the algorithms themselves do not depend on a specific projector or operator construction.

Applications. The most prominently featured application in this thesis is seismic full-waveform inversion (FWI), where we estimate acoustic velocity from observed seismic waves. Most exploration experiments have sources and receivers on only one side of the computational domain. While data noise and missing observations have a moderate impact on the recovered velocity models, the main challenge for FWI is the combination of an inaccurate initial guess and unavailable low-frequency data. These factors often cause the estimated model parameters to be geologically unrealistic. In chapter 2 and 3, I developed strategies to mitigate this problem by using constraints on the model parameters. The core of the approach is to start solving the inverse problem using low-frequency data and ‘tight’ constraints, continuing to higher frequency data and ‘looser’ constraints. While some incarnations of this concept have been around for a long time, such as working from smooth to less-smooth models by reducing the penalty parameters for Tikhonov regularization, I extended these ideas in the following ways: *i)* I use a constrained formulation with three different types of constraints. *ii)* Constraint sets do not depend on penalty parameters and after projection onto the intersection, the model parameters satisfy all constraints exactly. This approach provides accurate control of the model properties. *iii)* I show that tight to loose constraints for FWI works with total-variation constraints, as well as with slope constraints that induce monotonicity or smoothness. *iv)* Using a number of numerical examples, I show that the described strategy is a useful tool for FWI in general. Specifically, I demonstrate that a relaxation of multiple constraints works for two different formulations of FWI, for both sedimentary geology and model with high-contrast salt structures, and also when we do not know the source function and the observed synthetic data is modeled using more complex physics on finer grids.

Limitations, ongoing developments, and future research directions. Chapters 2, 3, 4, and 5 generally follow the proposed future research directions from the previous chapter. Chapter 3 introduces new algorithms that are faster than the ones in chapter 2, and illustrates the presented workflows on a more realistic example. In chapter 4, the main limitations of chapter 3 were tackled: avoiding nested algorithms and computing projections onto intersections of sets on large 3D grids, which requires a much faster implementation compared to the one in Chapter 3. In Chapter 4, I also extend the applications to the image processing tasks of denoising, deblurring, inpainting, and desaturation. The discussion and conclusions sections in chapter 4 describe ways to increase computational performance. In chapter 5, I do not continue the research on computational performance but address the more important limitation of the intersections of sets concept that underpins chapters 2, 3, and 4. This limitation arises when the geophysical models or images have a complex structure that is not easily described by standard sets (e.g., total-variation, low-rank, smooth) or intersections. Therefore, the Minkowksi set and the proposed generalization offer additional freedom to describe complex models and use more detailed prior knowledge.

There is a main limitation remaining that has not been discussed so far: what to do when there is no good prior information available to define constraint sets? In chapters 2 and 3, I introduced heuristics to select the maximum total-variation and smoothness, but they are still heuristics. The image and video processing examples in chapters 4 and 5 rely on examples to derive useful constraints. However, these training examples need to be relatively similar to the evaluation images. The challenge is to construct more quantitative ways to select constraint parameters for full-waveform inversion and relax the similarity requirements for training data in image and video processing. In what follows, I outline a proposal to combine the strengths of the methods and algorithms in this thesis, with recent developments in neural network research. The goal is to find additional ways to obtain information about 'good' constraints for PDE-based inverse problems and image/video processing.

In the past few years, many regularization techniques based on neural networks have been proposed. These include networks that *a)* map a corrupted image to a large scalar and a good image to a small scalar, thereby acting as a non-linear penalty function; *b)* directly map a corrupted image to the reconstruction, or map observed data to the model parameters. This type of end-to-end training is less flexible, in the sense that the network effectively includes the forward map and regularization. New forward maps or different regularization requires additional training of the network; *c)* act as the proximal map or projection operator as part of algorithms like proximal gradient. This approach combines neural networks with custom forward modeling operators, and is also known as plug-and-play regularization; *d)* map low-accuracy solutions of inverse problems into higher quality ones by removing artifacts in the image or estimated model parameters.

The different ways of using neural networks to solve inverse problems show state-of-art results. Each of the approaches comes with some limitations. First and foremost, networks typically require a large number of training data and labels to train. Below, I propose an alternative way to incorporate a neural network, which hopefully requires a relatively small amount of data, and that is easy and fast to train. At the same time, there is still the flexibility to change the (nonlinear) forward modeling operator.

I propose to use a neural network that maps a corrupted image into a scalar that describes a property of the clean image. These properties include ℓ_1, ℓ_2 or nuclear norms of the image, possibly in a transform domain. I can then use the scalar properties to define constraint sets for the regularization of an inverse problem. The two-step approach requires networks that map an image to a scalar rather than to an image, so perhaps the network can be shallower and narrower network and need fewer training examples compared to the four types of neural network regularization mentioned above.

Learning image properties and the constrained formulation for an inverse problem is a good combination because each constraint set is defined independently of all other constraint sets. Therefore, we can train one network per image property, independent of all other networks. Another advantage is that the constraints do not depend on the inverse problem or data-misfit function. Trained networks can, therefore, define constraints for most inverse problems.

Besides image processing, I also aim for the more ambitious goal of estimating model parameter properties from data obtained by physical experiments. For example, learning a direct map from observed seismic data to a scalar property of the velocity model in which the waves propagated. This type of problem has a nonlinear forward modeling operator that maps the model parameters to data. Our goal of training networks on a relatively small number of examples is especially important for geophysical inverse problems, where examples are scarce and selecting regularization is difficult. The two-step approach may alleviate some of the difficulties.

Initial training and testing of networks that predict image and data properties from corrupted inputs or data showed promising results. However, the added value of this idea still needs to be proven. The next questions are: 1) what is the number of training examples versus reconstruction error tradeoff compared to other approaches to using networks for inverse problems? 2) what type of network designs are suitable for learning to predict model properties? 3) are the currently available synthetic geophysical models realistic and diverse enough?

Bibliography

M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo. An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems. *IEEE Transactions on Image Processing*, 20(3):681–695, March 2011. ISSN 1057-7149. doi:10.1109/TIP.2010.2076294. → pages 82, 89, 134

H. K. Aggarwal, M. P. Mani, and M. Jacob. Model based image reconstruction using deep learned priors (modl). In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 671–674, April 2018. doi:10.1109/ISBI.2018.8363663. → page 123

V. Akcelik, G. Biros, and O. Ghattas. Parallel multiscale gauss-newton-krylov methods for inverse wave propagation. In *Supercomputing, ACM/IEEE 2002 Conference*, pages 41–41, Nov 2002. doi:10.1109/SC.2002.10002. → page 40

M. S. C. Almeida and M. Figueiredo. Deconvolving images with unknown boundaries using the alternating direction method of multipliers. *IEEE Transactions on Image Processing*, 22(8):3074–3086, Aug 2013. ISSN 1057-7149. doi:10.1109/TIP.2013.2258354. → page 83

A. Y. Anagaw. *Full waveform inversion using simultaneous encoded sources based on first-and second-order optimization methods*. PhD thesis, University of Alberta, 2014. → page 40

A. Y. Anagaw and M. D. Sacchi. Full waveform inversion with total variation regularization. In *Recovery-CSPG CSEG CWLS Convention*, 2011. → page 21

A. Y. Anagaw and M. D. Sacchi. Edge-preserving smoothing for simultaneous-source fwi model updates in high-contrast velocity models. *GEOPHYSICS*, 0(ja):1–18, 2017. doi:10.1190/geo2017-0563.1. URL https://doi.org/10.1190/geo2017-0563.1. → page 42

N. Antonello, L. Stella, P. Patrinos, and T. van Waterschoot. Proximal Gradient Algorithms: Applications in Signal Processing. *ArXiv e-prints*, Mar. 2018. → page 81

F. J. Aragón Artacho and R. Campoy. A new projection method for finding the closest point in the intersection of convex sets. *Computational Optimization and Applications*, 69(1):99–132, Jan 2018. ISSN 1573-2894. doi:10.1007/s10589-017-9942-5. URL https://doi.org/10.1007/s10589-017-9942-5. → page 80

A. Aravkin, R. Kumar, H. Mansour, B. Recht, and F. J. Herrmann. Fast methods for denoising matrix completion formulations, with applications to robust seismic data interpolation. *SIAM Journal on Scientific Computing*, 36(5):S237–S266, 2014. doi:10.1137/130919210. URL https://doi.org/10.1137/130919210. → pages 78, 115

A. Y. Aravkin and T. van Leeuwen. Estimating nuisance parameters in inverse problems. *Inverse Problems*, 28(11):115016, 2012. ISSN 0266-5611. → pages 1, 62

A. Y. Aravkin, J. V. Burke, D. Drusvyatskiy, M. P. Friedlander, and S. Roy. Level-set methods for convex optimization. *arXiv preprint arXiv:1602.01506*, 2016. → page 78

A. Asnaashari, R. Brossier, S. Garambois, F. Audebert, P. Thore, and J. Virieux. Time-lapse seismic imaging using regularized full-waveform inversion with a prior model: which strategy? *Geophysical Prospecting*, 63(1):78–98. doi:10.1111/1365-2478.12176. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/1365-2478.12176. → page 3

A. Asnaashari, R. Brossier, S. Garambois, F. Audebert, P. Thore, and J. Virieux. Regularized seismic full waveform inversion with prior model information. *GEOPHYSICS*, 78(2):R25–R36, 2013. doi:10.1190/geo2012-0104.1. URL http://dx.doi.org/10.1190/geo2012-0104.1. → pages 3, 40

G. E. Backus. Comparing hard and soft prior bounds in geophysical inverse problems. *Geophysical Journal International*, 94(2):249, 1988. doi:10.1111/j.1365-246X.1988.tb05899.x. URL +http://dx.doi.org/10.1111/j.1365-246X.1988.tb05899.x. → pages 36, 151

J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.

doi:10.1093/imanum/8.1.141. URL http://imajna.oxfordjournals.org/content/8/1/141.abstract. → pages 56, 92, 137

A. Baumstein. Pocs-based geophysical constraints in multi-parameter full wavefield inversion. EAGE, 06 2013. → pages 7, 36, 38

H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer Publishing Company, Incorporated, 1st edition, 2011. ISBN 1441994661, 9781441994660. → pages 38, 46, 48

H. H. Bauschke and V. R. Koch. Projection methods: Swiss army knives for solving feasibility and best approximation problems with halfspaces. *Contemporary Mathematics*, 636:1–40, 2015. → pages 47, 73, 80, 86, 196, 199

A. Beck. *Introduction to Nonlinear Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2014. doi:10.1137/1.9781611973655. URL http://epubs.siam.org/doi/abs/10.1137/1.9781611973655. → page 53

A. Beck. On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes. *SIAM Journal on Optimization*, 25(1): 185–209, 2015. doi:10.1137/13094829X. URL http://dx.doi.org/10.1137/13094829X. → page 44

A. Beck. *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017. doi:10.1137/1.9781611974997. URL http://epubs.siam.org/doi/abs/10.1137/1.9781611974997. → pages 85, 131

S. Becker, L. Horesh, A. Aravkin, E. van den Berg, and S. Zhuk. General optimization framework for robust and regularized 3d fwi. In *77th EAGE Conference and Exhibition 2015*, 2015. → pages 6, 37, 40

S. R. Becker, E. J. Candès, and M. C. Grant. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, 3(3):165, Jul 2011. ISSN 1867-2957. doi:10.1007/s12532-011-0029-5. URL https://doi.org/10.1007/s12532-011-0029-5. → page 115

L. Bello and M. Raydan. Convex constrained optimization for the seismic reflection tomography problem. *Journal of Applied Geophysics*, 62(2): 158 – 166, 2007. ISSN 0926-9851. doi:http://dx.doi.org/10.1016/j.jappgeo.2006.10.004. URL http://www.sciencedirect.com/science/article/pii/S0926985106001467. → pages 7, 36, 37, 57

D. P. Bertsekas. Projected newton methods for optimization problems with simple constraints. *SIAM Journal on Control and Optimization*, 20 (2):221–246, 1982. doi:10.1137/0320018. URL https://doi.org/10.1137/0320018. → pages 76, 127

D. P. Bertsekas. *Convex Optimization Algorithms*. Athena Scientific, 2015. → pages 44, 54

J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017. doi:10.1137/141000671. URL https://doi.org/10.1137/141000671. → pages 83, 92

S. A. Bigdeli and M. Zwicker. Image restoration using autoencoding priors. *arXiv preprint arXiv:1703.09964*, 2017. → page 123

F. Billette and S. Brandsberg-Dahl. The 2004 BP velocity benchmark. *67th EAGE Conference & Exhibition*, (June):13–16, 2005. URL http://www.earthdoc.org/publication/publicationdetails/?publication=1404. → page 27

E. G. Birgin and M. Raydan. Robust stopping criteria for dykstra's algorithm. *SIAM Journal on Scientific Computing*, 26(4):1405–1414, 2005. doi:10.1137/03060062X. URL http://dx.doi.org/10.1137/03060062X. → page 53

E. G. Birgin, J. M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. on Optimization*, 10(4):1196–1211, Aug. 1999. ISSN 1052-6234. doi:10.1137/S1052623497330963. URL http://dx.doi.org/10.1137/S1052623497330963. → pages 41, 55, 56, 57, 76, 109, 127, 137, 140

E. G. Birgin, J. M. Martnez, and M. Raydan. Inexact spectral projected gradient methods on convex sets. *IMA Journal of Numerical Analysis*, 23(4):539, 2003. doi:10.1093/imanum/23.4.539. URL +http://dx.doi.org/10.1093/imanum/23.4.539. → pages 55, 58, 137

S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787. → pages 44, 46

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, Jan. 2011. ISSN 1935-8237. doi:10.1561/2200000016. URL http://dx.doi.org/10.1561/2200000016. → pages 25, 44, 47, 86, 87, 93, 131, 136, 192, 194, 202

J. P. Boyle and R. L. Dykstra. *A Method for Finding Projections onto the Intersection of Convex Sets in Hilbert Spaces*, pages 28–47. Springer New York, New York, NY, 1986. ISBN 978-1-4613-9940-7. doi:10.1007/978-1-4613-9940-7_3. URL http://dx.doi.org/10.1007/978-1-4613-9940-7_3. → pages 24, 38, 47, 48, 80, 199

A. J. Brenders and R. G. Pratt. Full waveform tomography for lithospheric imaging: results from a blind test in a realistic crustal model. *Geophysical Journal International*, 168(1):133–151, 2007. doi:10.1111/j.1365-246X.2006.03156.x. URL http://gji.oxfordjournals.org/content/168/1/133.abstract. → page 41

C. Bunks. Multiscale seismic waveform inversion. *Geophysics*, 60(5):1457, Sept. 1995. ISSN 1070485X. doi:10.1190/1.1443880. URL http://link.aip.org/link/?GPY/60/1457/1&Agg=doi. → page 62

J. Burke. Basic convergence theory. Technical report, University of Washington, 1990. → page 43

G. Buzzard, S. Chan, S. Sreehari, and C. Bouman. Plug-and-play unplugged: Optimization-free reconstruction using consensus equilibrium. *SIAM Journal on Imaging Sciences*, 11(3):2001–2020, 2018. doi:10.1137/17M1122451. URL https://doi.org/10.1137/17M1122451. → page 123

E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717, Apr

2009. ISSN 1615-3383. doi:10.1007/s10208-009-9045-5. URL https://doi.org/10.1007/s10208-009-9045-5. → page 115

E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, June 2011. ISSN 0004-5411. doi:10.1145/1970392.1970395. URL http://doi.acm.org/10.1145/1970392.1970395. → pages 126, 127, 144

Y. Censor. Computational acceleration of projection algorithms for the linear best approximation problem. *Linear Algebra and its Applications*, 416(1):111 – 123, 2006. ISSN 0024-3795. doi:http://dx.doi.org/10.1016/j.laa.2005.10.006. URL http://www.sciencedirect.com/science/article/pii/S0024379505004891. → pages 73, 80

Y. Censor, T. Elfving, N. Kopf, and T. Bortfeld. The multiple-sets split feasibility problem and its applications for inverse problems. *Inverse Problems*, 21(6):2071, 2005. → page 94

S. H. Chan, X. Wang, and O. A. Elgendy. Plug-and-play admm for image restoration: Fixed-point convergence and applications. *IEEE Transactions on Computational Imaging*, 3(1):84–98, March 2017. ISSN 2333-9403. doi:10.1109/TCI.2016.2629286. → page 123

J. H. R. Chang, C. Li, B. Pczos, and B. V. K. V. Kumar. One network to solve them all solving linear inverse problems using deep projection models. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5889–5898, Oct 2017. doi:10.1109/ICCV.2017.627. → page 123

S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001. doi:10.1137/S003614450037906X. URL https://doi.org/10.1137/S003614450037906X. → page 78

P. Combettes. The convex feasibility problem in image recovery. volume 95 of *Advances in Imaging and Electron Physics*, pages 155 – 270. Elsevier, 1996. doi:https://doi.org/10.1016/S1076-5670(08)70157-5. URL http://www.sciencedirect.com/science/article/pii/S1076567008701575. → pages 77, 138

P. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In H. H. Bauschke, R. S. Burachik, P. L. Combettes,

V. Elser, D. R. Luke, and H. Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, volume 49 of *Springer Optimization and Its Applications*, pages 185–212. Springer New York, 2011. ISBN 978-1-4419-9568-1. doi:10.1007/978-1-4419-9569-8_10. URL http://dx.doi.org/10.1007/978-1-4419-9569-8_10. → pages 73, 82, 85, 89, 131, 134

P. L. Combettes. The foundations of set theoretic estimation. *Proceedings of the IEEE*, 81(2):182–208, Feb 1993. ISSN 0018-9219. doi:10.1109/5.214546. → pages 77, 138

P. L. Combettes and J. C. Pesquet. Image restoration subject to a total variation constraint. *IEEE Transactions on Image Processing*, 13(9): 1213–1222, Sept 2004. ISSN 1057-7149. doi:10.1109/TIP.2004.832922. → pages 77, 79, 114

S. C. Constable, R. L. Parker, and C. G. Constable. Occams inversion: A practical algorithm for generating smooth models from electromagnetic sounding data. *GEOPHYSICS*, 52(3):289–300, 1987. doi:10.1190/1.1442303. URL http://dx.doi.org/10.1190/1.1442303. → pages 6, 78

C. Da Silva and F. J. Herrmann. A Unified 2D/3D Large Scale Software Environment for Nonlinear Inverse Problems. *ArXiv e-prints*, Mar. 2017. → pages 62, 78, 110

W. Dai, X. Wang, and G. T. Schuster. Least-squares migration of multisource data with a deblurring filter. *GEOPHYSICS*, 76(5): R135–R146, 2011. doi:10.1190/geo2010-0159.1. URL https://doi.org/10.1190/geo2010-0159.1. → page 3

Y. Dai and L. Liao. Rlinear convergence of the barzilai and borwein gradient method. *IMA Journal of Numerical Analysis*, 22(1):1, 2002. doi:10.1093/imanum/22.1.1. URL +http://dx.doi.org/10.1093/imanum/22.1.1. → page 56

J. Dattorro. *Convex Optimization & Euclidean Distance Geometry*. Meboo Publishing USA, 2010. → pages 46, 48

S. Diamond, R. Takapoui, and S. Boyd. A general system for heuristic minimization of convex functions over non-convex sets. *Optimization Methods and Software*, 33(1):165–193, 2018.

doi:10.1080/10556788.2017.1304548. URL https://doi.org/10.1080/10556788.2017.1304548. → pages 122, 131

D. P. Dobkin, J. Hershberger, D. G. Kirkpatrick, and S. Suri. Computing the intersection-depth of polyhedra. *Algorithmica*, 9:518–533, 1993. → page 128

A. Domahidi, E. Chu, and S. Boyd. Ecos: An socp solver for embedded systems. In *Control Conference (ECC), 2013 European*, pages 3071–3076. IEEE, 2013. → page 81

D. Driggs, S. Becker, and A. Aravkin. Adapting Regularized Low Rank Models for Parallel Architectures. *ArXiv e-prints*, Feb. 2017. → pages 144, 148

J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l1-ball for learning in high dimensions. In A. McCallum and S. Roweis, editors, *Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008)*, pages 272–279. Omnipress, 2008. → page 104

R. L. Dykstra. An algorithm for restricted least squares regression. *Journal of the American Statistical Association*, 78(384):837–842, 1983. doi:10.1080/01621459.1983.10477029. URL http://www.tandfonline.com/doi/abs/10.1080/01621459.1983.10477029. → pages 38, 47, 80, 199

J. Eckstein and D. P. Bertsekas. On the douglas—rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1):293–318, Apr 1992. ISSN 1436-4646. doi:10.1007/BF01581204. URL https://doi.org/10.1007/BF01581204. → pages 89, 92

J. Eckstein and W. Yao. Understanding the convergence of the alternating direction method of multipliers: Theoretical and computational perspectives. *Pac. J. Optim. To appear*, 2015. → pages 87, 131, 136

R. G. Ellis and D. W. Oldenburg. Applied geophysical inversion. *Geophysical Journal International*, 116(1):5, 1994. doi:10.1111/j.1365-246X.1994.tb02122.x. URL +http://dx.doi.org/10.1111/j.1365-246X.1994.tb02122.x. → page 6

I. Epanomeritakis, V. Akelik, O. Ghattas, and J. Bielak. A newton-cg method for large-scale three-dimensional elastic full-waveform seismic inversion. *Inverse Problems*, 24(3):034015, 2008. URL http://stacks.iop.org/0266-5611/24/i=3/a=034015. → page 21

R. Escalante and M. Raydan. *Alternating Projection Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2011. ISBN 1611971934, 9781611971934. → pages 38, 48

E. Esser. Applications of lagrangian-based alternating direction methods and connections to split bregman. 2009. → page 92

E. Esser, L. Guasch, T. van Leeuwen, A. Y. Aravkin, and F. J. Herrmann. *Automatic salt delineation Wavefield Reconstruction Inversion with convex constraints*, chapter 257, pages 1337–1343. 2015a. doi:10.1190/segam2015-5877995.1. URL http://library.seg.org/doi/abs/10.1190/segam2015-5877995.1. → pages 7, 36, 44

E. Esser, L. Guasch, T. van Leeuwen, A. Y. Aravkin, and F. J. Herrmann. Total variation regularization strategies in full waveform inversion for improving robustness to noise, limited data and poor initializations. Technical Report TR-EOAS-2015-5, 06 2015b. URL https://www.slim.eos.ubc.ca/Publications/Public/TechReport/2015/esser2015tvwri/esser2015tvwri.html. → page 30

E. Esser, L. Guasch, F. J. Herrmann, and M. Warner. Constrained waveform inversion for automatic salt flooding. *The Leading Edge*, 35 (3):235–239, mar 2016a. ISSN 1070-485X. doi:10.1190/tle35030235.1. URL http://library.seg.org/doi/10.1190/tle35030235.1. → pages 22, 25

E. Esser, L. Guasch, F. J. Herrmann, and M. Warner. Constrained waveform inversion for automatic salt flooding. *The Leading Edge*, 35 (3):235–239, 2016b. doi:10.1190/tle35030235.1. URL http://dx.doi.org/10.1190/tle35030235.1. → pages 7, 36, 44, 45, 63, 65, 79, 197

E. Esser, L. Guasch, T. van Leeuwen, A. Y. Aravkin, and F. J. Herrmann. Total-variation regularization strategies in full-waveform inversion. *ArXiv e-prints*, Aug. 2016. → pages 7, 22, 25, 79, 103, 109, 125, 128, 129, 137

E. Esser, L. Guasch, T. van Leeuwen, A. Y. Aravkin, and F. J. Herrmann. Total variation regularization strategies in full-waveform inversion. *SIAM Journal on Imaging Sciences*, 11(1):376–406, 2018. doi:10.1137/17M111328X. URL https://doi.org/10.1137/17M111328X. → pages 36, 38, 44, 63, 65

J. M. Fadili and G. Peyre. Total variation projection with first order schemes. *IEEE Transactions on Image Processing*, 20(3):657–669, March 2011. ISSN 1057-7149. doi:10.1109/TIP.2010.2072512. → page 151

K. Fan, Q. Wei, L. Carin, and K. A. Heller. An inner-loop free solution to inverse problems using deep neural networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2370–2380. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/6831-an-inner-loop-free-solution-to-inverse-problems-using-deep-neural-networks.pdf. → page 123

C. G. Farquharson and D. W. Oldenburg. Non-linear inversion using general measures of data misfit and model structure. *Geophysical Journal International*, 134(1):213, 1998. doi:10.1046/j.1365-246x.1998.00555.x. URL +http://dx.doi.org/10.1046/j.1365-246x.1998.00555.x. → pages 6, 40

C. G. Farquharson and D. W. Oldenburg. A comparison of automatic techniques for estimating the regularization parameter in non-linear inverse problems. *Geophysical Journal International*, 156(3):411–425, 2004. doi:10.1111/j.1365-246X.2004.02190.x. URL http://gji.oxfordjournals.org/content/156/3/411.abstract. → pages 35, 40

P. Farrell, D. Ham, S. Funke, and M. Rognes. Automated derivation of the adjoint of high-level transient finite element programs. *SIAM Journal on Scientific Computing*, 35(4):C369–C393, 2013. doi:10.1137/120873558. URL https://doi.org/10.1137/120873558. → page 78

M. Frigo and S. G. Johnson. The design and implementation of fftw3. *Proceedings of the IEEE*, 93(2):216–231, Feb 2005. ISSN 0018-9219. doi:10.1109/JPROC.2004.840301. → page 93

L. A. Gallardo and M. A. Meju. Joint two-dimensional cross-gradient imaging of magnetotelluric and seismic traveltime data for structural and lithological classification. *Geophysical Journal International*, 169(3): 1261–1272, 2007. doi:10.1111/j.1365-246X.2007.03366.x. URL http://dx.doi.org/10.1111/j.1365-246X.2007.03366.x. → page 3

W. Gander. Least squares with a quadratic constraint. *Numerische Mathematik*, 36(3):291–307, Sep 1980. ISSN 0945-3245. doi:10.1007/BF01396656. URL https://doi.org/10.1007/BF01396656. → page 78

H. Gao, J.-F. Cai, Z. Shen, and H. Zhao. Robust principal component analysis-based four-dimensional computed tomography. *Physics in Medicine and Biology*, 56(11):3181, 2011a. URL http://stacks.iop.org/0031-9155/56/i=11/a=002. → pages 126, 127

H. Gao, H. Yu, S. Osher, and G. Wang. Multi-energy ct based on a prior rank, intensity and sparsity model (prism). *Inverse Problems*, 27(11): 115012, 2011b. URL http://stacks.iop.org/0266-5611/27/i=11/a=115012. → pages 126, 127

G. H. Golub and U. von Matt. Quadratically constrained least squares and quadratic problems. *Numerische Mathematik*, 59(1):561–580, Dec 1991. ISSN 0945-3245. doi:10.1007/BF01385796. URL https://doi.org/10.1007/BF01385796. → page 78

D. Gragnaniello, C. Chaux, J. C. Pesquet, and L. Duval. A convex variational approach for multiple removal in seismic data. In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 215–219, Aug 2012. → pages 125, 128

S. A. Greenhalgh, Z. Bing, and A. Green. Solutions, algorithms and inter-relations for local minimization search geophysical inversion. *Journal of Geophysics and Engineering*, 3(2):101, 2006. URL http://stacks.iop.org/1742-2140/3/i=2/a=001. → page 6

L. Grippo and M. Sciandrone. Nonmonotone globalization techniques for the barzilai-borwein gradient method. *Computational Optimization and Applications*, 23(2):143–169, Nov 2002. ISSN 1573-2894. doi:10.1023/A:1020587701058. URL http://dx.doi.org/10.1023/A:1020587701058. → page 57

A. Guitton and E. Daz. Attenuating crosstalk noise with simultaneous source full waveform inversion. *Geophysical Prospecting*, 60(4):759–768,

2012. ISSN 1365-2478. doi:10.1111/j.1365-2478.2011.01023.x. URL http://dx.doi.org/10.1111/j.1365-2478.2011.01023.x. → page 42

A. Guitton, G. Ayeni, and E. Daz. Constrained full-waveform inversion by model reparameterization. *GEOPHYSICS*, 77(2):R117–R127, 2012. doi:10.1190/geo2011-0196.1. URL http://dx.doi.org/10.1190/geo2011-0196.1. → page 42

E. Haber. *Computational methods in geophysical electromagnetics*. SIAM, 2014. → page 76

E. Haber and M. Holtzman Gazit. Model fusion and joint inversion. *Surveys in Geophysics*, 34(5):675–695, Sep 2013. ISSN 1573-0956. doi:10.1007/s10712-013-9232-4. URL https://doi.org/10.1007/s10712-013-9232-4. → page 3

E. Haber, U. M. Ascher, and D. Oldenburg. On optimization techniques for solving nonlinear inverse problems. *Inverse Problems*, 16(5): 1263–1280, Oct. 2000. ISSN 0266-5611. doi:10.1088/0266-5611/16/5/309. URL http://stacks.iop.org/0266-5611/16/i=5/a=309?key=crossref.98f435f9ee66231b63da02b10f82a60b. → pages 76, 139

B. S. He, H. Yang, and S. L. Wang. Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. *Journal of Optimization Theory and Applications*, 106(2):337–356, 2000. ISSN 1573-2878. doi:10.1023/A:1004603514434. URL http://dx.doi.org/10.1023/A:1004603514434. → page 193

F. Heide, S. Diamond, M. Nießner, J. Ragan-Kelley, W. Heidrich, and G. Wetzstein. Proximal: Efficient image optimization using proximal algorithms. *ACM Trans. Graph.*, 35(4):84:1–84:15, July 2016. ISSN 0730-0301. doi:10.1145/2897824.2925875. URL http://doi.acm.org/10.1145/2897824.2925875. → page 81

F. Herrmann, X. Li, A. Y. Aravkin, and T. Van Leeuwen. A modified, sparsity-promoting, gauss-newton algorithm for seismic waveform inversion. In *SPIE Optical Engineering+ Applications*, pages 81380V–81380V. International Society for Optics and Photonics, 2011. → page 43

F. J. Herrmann and X. Li. Efficient least-squares imaging with sparsity promotion and compressive sensing. *Geophysical Prospecting*, 60(4): 696–712. doi:10.1111/j.1365-2478.2011.01041.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2478.2011.01041.x. → page 3

J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of convex analysis*. Springer Science & Business Media, 2012. → page 127

W. Hu, A. Abubakar, and T. M. Habashy. Joint electromagnetic and seismic inversion using structural constraints. *GEOPHYSICS*, 74(6): R99–R109, 2009. doi:10.1190/1.3246586. URL https://doi.org/10.1190/1.3246586. → page 3

F. Iutzeler and J. M. Hendrickx. A generic online acceleration scheme for optimization algorithms via relaxation and inertia. *Optimization Methods and Software*, 0(0):1–23, 2017. doi:10.1080/10556788.2017.1396601. URL https://doi.org/10.1080/10556788.2017.1396601. → page 89

V. K. Ivanov, V. V. Vasin, and V. P. Tanana. *Theory of linear ill-posed problems and its applications*, volume 36. Walter de Gruyter, 2013. → page 78

M. Jervis, M. K. Sen, and P. L. Stoffa. Prestack migration velocity estimation using nonlinear methods. *GEOPHYSICS*, 61(1):138–150, 1996. doi:10.1190/1.1443934. URL http://dx.doi.org/10.1190/1.1443934. → page 42

Z. Jia, X. Cai, and D. Han. Comparison of several fast algorithms for projection onto an ellipsoid. *Journal of Computational and Applied Mathematics*, 319:320 – 337, 2017. ISSN 0377-0427. doi:https://doi.org/10.1016/j.cam.2017.01.008. URL http://www.sciencedirect.com/science/article/pii/S0377042717300122. → page 86

Z. Kang, C. Peng, and Q. Cheng. Robust pca via nonconvex rank approximation. In *2015 IEEE International Conference on Data Mining*, pages 211–220, Nov 2015. doi:10.1109/ICDM.2015.15. → page 144

M. Karaoulis, A. Revil, D. D. Werkema, B. J. Minsley, W. F. Woodruff, and A. Kemna. Time-lapse three-dimensional inversion of complex conductivity data using an active time constrained (atc) approach. *Geophysical Journal International*, 187(1):237–251, 2011. doi:10.1111/j.1365-246X.2011.05156.x. URL http://dx.doi.org/10.1111/j.1365-246X.2011.05156.x. → page 3

B. L. N. Kennett and P. R. Williamson. *Subspace methods for large-scale nonlinear inversion*, pages 139–154. Springer Netherlands, Dordrecht, 1988. ISBN 978-94-009-2857-2. doi:10.1007/978-94-009-2857-2_7. URL http://dx.doi.org/10.1007/978-94-009-2857-2_7. → page 42

S. Kitic, L. Albera, N. Bertin, and R. Gribonval. Physics-driven inverse problems made tractable with cosparse regularization. *IEEE Transactions on Signal Processing*, 64(2):335–348, Jan 2016. ISSN 1053-587X. doi:10.1109/TSP.2015.2480045. → pages 82, 88, 89, 93, 134

R. Kleinman and P. den Berg. A modified gradient method for two-dimensional problems in tomography. *Journal of Computational and Applied Mathematics*, 42(1):17 – 35, 1992. ISSN 0377-0427. doi:http://dx.doi.org/10.1016/0377-0427(92)90160-Y. URL http://www.sciencedirect.com/science/article/pii/037704279290160Y. → page 42

H. Kotakemori, H. Hasegawa, T. Kajiyama, A. Nukada, R. Suda, and A. Nishida. Performance evaluation of parallel sparse matrix-vector products on sgi altix3700. *Lecture Notes in Computer Science*, 4315: 153–166, 2008. → page 92

J. R. Krebs, J. E. Anderson, D. Hinkley, R. Neelamani, S. Lee, A. Baumstein, and M.-D. Lacasse. Fast full-wavefield seismic inversion using encoded sources. *GEOPHYSICS*, 74(6):WCC177–WCC188, 2009. doi:10.1190/1.3230502. URL http://dx.doi.org/10.1190/1.3230502. → page 3

N. Kreimer, A. Stanton, and M. D. Sacchi. Tensor completion based on nuclear norm minimization for 5d seismic data reconstruction. *GEOPHYSICS*, 78(6):V273–V284, 2013. doi:10.1190/geo2013-0022.1. URL https://doi.org/10.1190/geo2013-0022.1. → page 5

N. Kukreja, M. Louboutin, F. Vieira, F. Luporini, M. Lange, and G. Gorman. Devito: Automated fast finite difference computation. In *2016 Sixth International Workshop on Domain-Specific Languages and High-Level Frameworks for High Performance Computing (WOLFHPC)*, pages 11–19, Nov 2016. doi:10.1109/WOLFHPC.2016.06. → page 78

R. Kumar, C. D. Silva, O. Akalin, A. Y. Aravkin, H. Mansour, B. Recht, and F. J. Herrmann. Efficient matrix completion for seismic data reconstruction. *GEOPHYSICS*, 80(5):V97–V114, 2015. doi:10.1190/geo2014-0369.1. URL https://doi.org/10.1190/geo2014-0369.1. → page 5

A. Kundu, F. Bach, and C. Bhattacharyya. Convex optimization over intersection of simple sets: improved convergence rate guarantees via an exact penalty approach. *ArXiv e-prints*, Oct. 2017. → page 86

J. D. Lee, Y. Sun, and M. A. Saunders. Proximal newton-type methods for minimizing composite functions. *SIAM Journal on Optimization*, 24(3): 1420–1443, 2014. doi:10.1137/130921428. URL http://dx.doi.org/10.1137/130921428. → page 73

Y. Lee, E. Behar, J.-M. Lien, and Y. J. Kim. Continuous penetration depth computation for rigid models using dynamic minkowski sums. *Computer-Aided Design*, 78:14 – 25, 2016. ISSN 0010-4485. doi:https://doi.org/10.1016/j.cad.2016.05.012. URL http://www.sciencedirect.com/science/article/pii/S001044851630032X. SPM 2016. → page 128

P. G. Lelivre and D. W. Oldenburg. A comprehensive study of including structural orientation information in geophysical inversions. *Geophysical Journal International*, 178(2):623, 2009. doi:10.1111/j.1365-246X.2009.04188.x. URL +http://dx.doi.org/10.1111/j.1365-246X.2009.04188.x. → pages 7, 36, 37, 44, 196

M. Li, O. Semerci, and A. Abubakar. A contrast source inversion method in the wavelet domain. *Inverse Problems*, 29(2):025015, 2013. URL http://stacks.iop.org/0266-5611/29/i=2/a=025015. → page 42

X. Li, A. Aravkin, T. van Leeuwen, and F. Herrmann. Fast randomized full-waveform inversion with compressive sensing. *Geophysics*, 77(3): A13, 2012a. ISSN 00168033. doi:10.1190/geo2011-0410.1. → page 3

X. Li, A. Y. Aravkin, T. van Leeuwen, and F. J. Herrmann. Fast randomized full-waveform inversion with compressive sensing. *GEOPHYSICS*, 77(3):A13–A17, 2012b. doi:10.1190/geo2011-0410.1. URL http://dx.doi.org/10.1190/geo2011-0410.1. → page 43

X. Li, E. Esser, and F. J. Herrmann. Modified Gauss-Newton full-waveform inversion explained–-why sparsity-promoting updates do matter. *Geophysics*, 81(3):R125–R138, 05 2016. doi:10.1190/geo2015-0266.1. URL https://www.slim.eos.ubc.ca/Publications/Public/Journals/Geophysics/2016/li2015GEOPmgn/li2015GEOPmgn.pdf. → page 43

Y. E. Li and L. Demanet. Full-waveform inversion with extrapolated low-frequency data. *GEOPHYSICS*, 81(6):R339–R348, 2016. doi:10.1190/geo2016-0038.1. URL https://doi.org/10.1190/geo2016-0038.1. → page 4

Y. Lin and L. Huang. Acoustic- and elastic-waveform inversion using a modified total-variation regularization scheme. *Geophysical Journal International*, 200(1):489–502, 2015. doi:10.1093/gji/ggu393. URL http://gji.oxfordjournals.org/content/200/1/489.abstract. → pages 6, 40, 41, 70

L. R. Lines, A. K. Schultz, and S. Treitel. Cooperative inversion of geophysical data. *GEOPHYSICS*, 53(1):8–20, 1988. doi:10.1190/1.1442403. URL https://doi.org/10.1190/1.1442403. → page 3

W. López and M. Raydan. An acceleration scheme for dykstra's algorithm. *Computational Optimization and Applications*, 63(1):29–44, Jan 2016. ISSN 1573-2894. doi:10.1007/s10589-015-9768-y. URL https://doi.org/10.1007/s10589-015-9768-y. → page 80

M. Louboutin, P. Witte, M. Lange, N. Kukreja, F. Luporini, G. Gorman, and F. J. Herrmann. Full-waveform inversion, part 1: Forward modeling. *The Leading Edge*, 36(12):1033–1036, 2017. doi:10.1190/tle36121033.1. URL https://doi.org/10.1190/tle36121033.1. → page 61

M. Louboutin, P. Witte, M. Lange, N. Kukreja, F. Luporini, G. Gorman, and F. J. Herrmann. Full-waveform inversion, part 2: Adjoint modeling. *The Leading Edge*, 37(1):69–72, 2018. doi:10.1190/tle37010069.1. URL https://doi.org/10.1190/tle37010069.1. → page 78

M. Lustig, D. Donoho, and J. M. Pauly. Sparse mri: The application of compressed sensing for rapid mr imaging. *Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007. ISSN 1522-2594. doi:10.1002/mrm.21391. URL http://dx.doi.org/10.1002/mrm.21391. → page 115

J. Macdonald and L. Ruthotto. Improved susceptibility artifact correction of echo-planar mri using the alternating direction method of multipliers. *Journal of Mathematical Imaging and Vision*, 60(2):268–282, Feb 2018. ISSN 1573-7683. doi:10.1007/s10851-017-0757-x. URL https://doi.org/10.1007/s10851-017-0757-x. → pages 95, 112

S. Mallat and Z. Zhang. Adaptive time-frequency decomposition with matching pursuits. In *[1992] Proceedings of the IEEE-SP International*

Symposium on Time-Frequency and Time-Scale Analysis, pages 7–10, Oct 1992. doi:10.1109/TFTSA.1992.274245. → page 78

H. Mansour, R. Saab, P. Nasiopoulos, and R. Ward. Color image desaturation using sparse reconstruction. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 778–781, March 2010. doi:10.1109/ICASSP.2010.5494984. → page 119

X. Mei, W. Dong, B.-G. Hu, and S. Lyu. Unihist: A unified framework for image restoration with marginal histogram constraints. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. → page 151

L. Métivier and R. Brossier. The seiscope optimization toolbox: A large-scale nonlinear optimization library based on reverse communication. *GEOPHYSICS*, 81(2):F1–F15, 2016. doi:10.1190/geo2015-0031.1. URL https://doi.org/10.1190/geo2015-0031.1. → page 37

Y. Meyer. *Oscillating patterns in image processing and nonlinear evolution equations: the fifteenth Dean Jacqueline B. Lewis memorial lectures*, volume 22. American Mathematical Soc., 2001. → page 128

J. Mueller and S. Siltanen. *Linear and Nonlinear Inverse Problems with Practical Applications*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2012. doi:10.1137/1.9781611972344. URL http://epubs.siam.org/doi/abs/10.1137/1.9781611972344. → page 35

P. Netrapalli, N. U N, S. Sanghavi, A. Anandkumar, and P. Jain. Non-convex robust pca. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1107–1115. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5430-non-convex-robust-pca.pdf. → page 144

R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. I. Jordan. A general analysis of the convergence of admm. In *Int. Conf. Mach. Learn.*, volume 37, pages 343–352, 2015. → pages 92, 193

J. Nocedal and S. J. Wright. *Numerical optimization*. Springer, 2000. → pages 41, 56, 87, 133, 192

D. O'Connor and L. Vandenberghe. Total variation image deblurring with space-varying kernel. *Computational Optimization and Applications*, 67 (3):521–541, Jul 2017. ISSN 1573-2894. doi:10.1007/s10589-017-9901-1. URL https://doi.org/10.1007/s10589-017-9901-1. → page 83

B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068, Jun 2016. ISSN 1573-2878. doi:10.1007/s10957-016-0892-3. URL https://doi.org/10.1007/s10957-016-0892-3. → page 81

F. Oghenekohwo, R. Kumar, E. Esser, and F. J. Herrmann. Using common information in compressive time-lapse full-waveform inversion. In *77th EAGE Conference and Exhibition 2015*, 2015. → page 3

D. W. Oldenburg, P. R. McGillivray, and R. G. Ellis. Generalized subspace methods for large-scale inverse problems. *Geophysical Journal International*, 114(1):12, 1993. doi:10.1111/j.1365-246X.1993.tb01462.x. URL +http://dx.doi.org/10.1111/j.1365-246X.1993.tb01462.x. → page 42

S. Ono, T. Miyata, and I. Yamada. Cartoon-texture image decomposition using blockwise low-rank texture characterization. *IEEE Transactions on Image Processing*, 23(3):1128–1142, March 2014. ISSN 1057-7149. doi:10.1109/TIP.2014.2299067. → pages 126, 127

S. Osher, A. Sol, and L. Vese. Image decomposition and restoration using total variation minimization and the h1. *Multiscale Modeling and Simulation*, 1(3):349–370, 2003. doi:10.1137/S1540345902416247. URL http://dx.doi.org/10.1137/S1540345902416247. → pages 126, 127

C. C. Paige and M. A. Saunders. Lsqr: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Softw.*, 8(1): 43–71, Mar. 1982. ISSN 0098-3500. doi:10.1145/355984.355989. URL http://doi.acm.org/10.1145/355984.355989. → pages 90, 134, 194

S. K. Pakazad, M. S. Andersen, and A. Hansson. Distributed solutions for loosely coupled feasibility problems using proximal splitting methods. *Optimization Methods and Software*, 30(1):128–161, 2015. doi:10.1080/10556788.2014.902056. URL https://doi.org/10.1080/10556788.2014.902056. → page 86

N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, 2014. ISSN 2167-3888.

doi:10.1561/2400000003. URL http://dx.doi.org/10.1561/2400000003. → pages 37, 44, 85, 131

B. Peters and F. J. Herrmann. Constraints versus penalties for edge-preserving full-waveform inversion. *The Leading Edge*, 36(1): 94–100, 2017. doi:10.1190/tle36010094.1. URL http://dx.doi.org/10.1190/tle36010094.1.

B. Peters, F. Herrmann, and T. V. Leeuwen. *Parallel reformulation of the sequential adjoint-state method*, pages 1411–1415. 2016. doi:10.1190/segam2016-13966771.1. URL https://library.seg.org/doi/abs/10.1190/segam2016-13966771.1. → page 3

B. Peters, B. R. Smithyman, and F. J. Herrmann. Projection methods and applications for seismic nonlinear inverse problems with multiple constraints. *GEOPHYSICS*, 0(ja):1–100, 2018. doi:10.1190/geo2018-0192.1. URL https://doi.org/10.1190/geo2018-0192.1.

J. Petersson and O. Sigmund. Slope constrained topology optimization. *International Journal for Numerical Methods in Engineering*, 41(8): 1417–1434, 1998. ISSN 1097-0207. doi: 10.1002/(SICI)1097-0207(19980430)41:8⟨1417::AID-NME344⟩3.0.CO;2-N. URL http://dx.doi.org/10.1002/(SICI)1097-0207(19980430)41:8⟨1417::AID-NME344⟩3.0.CO;2-N. → page 196

G. Peyré. Sparse modeling of textures. *Journal of Mathematical Imaging and Vision*, 34(1):17–31, May 2009. ISSN 1573-7683. doi:10.1007/s10851-008-0120-3. URL https://doi.org/10.1007/s10851-008-0120-3. → page 151

M. Q. Pham, C. Chaux, L. Duval, and J. C. Pesquet. A constrained-based optimization approach for seismic data recovery problems. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2377–2381, May 2014a. doi:10.1109/ICASSP.2014.6854025. → pages 125, 128

M. Q. Pham, L. Duval, C. Chaux, and J. C. Pesquet. A primal-dual proximal algorithm for sparse template-based adaptive filtering: Application to seismic multiple removal. *IEEE Transactions on Signal Processing*, 62(16):4256–4269, Aug 2014b. ISSN 1053-587X. doi:10.1109/TSP.2014.2331614. → pages 125, 128

R.-E. Plessix. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International*, 167(2):495–503, 2006. ISSN 1365-246X. doi:10.1111/j.1365-246X.2006.02978.x. URL http://dx.doi.org/10.1111/j.1365-246X.2006.02978.x. → page 27

J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–70, Oct 2000. ISSN 1573-1405. doi:10.1023/A:1026553619983. URL https://doi.org/10.1023/A:1026553619983. → page 151

G. Pratt, C. Shin, and G. Hicks. Gauss-Newton and full Newton methods in frequency-space seismic waveform inversion. *Geophysical Journal International*, 133(2):341–362, May 1998. ISSN 0956540X. doi:10.1046/j.1365-246X.1998.00498.x. URL http://doi.wiley.com/10.1046/j.1365-246X.1998.00498.x. → pages 1, 76, 139

R. G. Pratt. Seismic waveform inversion in the frequency domain, part 1: Theory and verification in a physical scale model. *GEOPHYSICS*, 64(3): 888–901, 1999. doi:10.1190/1.1444597. URL https://doi.org/10.1190/1.1444597. → pages 1, 62

L. Qiu, N. Chemingui, Z. Zou, and A. Valenciano. Full-waveform inversion with steerable variation regularization. *SEG Technical Program Expanded Abstracts 2016*, pages 1174–1178, 2016. doi:10.1190/segam2016-13872436.1. URL http://library.seg.org/doi/abs/10.1190/segam2016-13872436.1. → pages 6, 41

M. Raydan. On the barzilai and borwein choice of steplength for the gradient method. *IMA Journal of Numerical Analysis*, 13(3):321–326, 1993. doi:10.1093/imanum/13.3.321. URL http://imajna.oxfordjournals.org/content/13/3/321.abstract. → page 56

L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60:259–268, 1992. URL http://www.sciencedirect.com/science/article/pii/016727899290242F. → page 19

L. Ruthotto, E. Treister, and E. Haber. jinv–a flexible julia package for pde parameter estimation. *SIAM Journal on Scientific Computing*, 39 (5):S702–S722, 2017. doi:10.1137/16M1081063. URL https://doi.org/10.1137/16M1081063. → page 78

E. K. Ryu and S. Boyd. Primer on monotone operator methods. *Appl. Comput. Math*, 15(1):3–43, 2016. → page 193

Y. Saad. Krylov subspace methods on supercomputers. *SIAM Journal on Scientific and Statistical Computing*, 10(6):1200–1232, 1989. doi:10.1137/0910073. URL https://doi.org/10.1137/0910073. → page 92

J. A. Scales and R. Snieder. To bayes or not to bayes? *GEOPHYSICS*, 62 (4):1045–1046, 1997. doi:10.1190/1.6241045.1. URL http://dx.doi.org/10.1190/1.6241045.1. → pages 36, 151

H. Schaeffer and S. Osher. A low patch-rank interpretation of texture. *SIAM Journal on Imaging Sciences*, 6(1):226–262, 2013. doi:10.1137/110854989. URL http://dx.doi.org/10.1137/110854989. → pages 126, 127

M. Schmidt and K. Murphy. Convex structure learning in log-linear models: Beyond pairwise potentials. In Y. W. Teh and M. Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 709–716, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL http://proceedings.mlr.press/v9/schmidt10a.html. → page 58

M. Schmidt, E. Van Den Berg, M. P. Friedlander, and K. Murphy. Optimizing costly functions with simple constraints: A limited-memory projected quasi-newton algorithm. In *Proc. of Conf. on Artificial Intelligence and Statistics*, 2009. → pages 73, 76, 127

M. Schmidt, D. Kim, and S. Sra. *Projected Newton-type Methods in Machine Learning*, volume 35, chapter 11, pages 305–327. MIT Press, 04 2012. → pages 73, 76, 127

M. Sen and I. Roy. Computation of differential seismograms and iteration adaptive regularization in prestack waveform inversion. *GEOPHYSICS*, 68(6):2026–2039, 2003. doi:10.1190/1.1635056. URL http://dx.doi.org/10.1190/1.1635056. → page 35

F. J. Sern, F. J. Sanz, M. Kindeln, and J. I. Badal. Finite-element method for elastic wave propagation. *Communications in Applied Numerical Methods*, 6(5):359–368, 1990. ISSN 1555-2047. doi:10.1002/cnm.1630060505. URL http://dx.doi.org/10.1002/cnm.1630060505. → page 92

P. Shen and W. W. Symes. Automatic velocity analysis via shot profile migration. *GEOPHYSICS*, 73(5):VE49–VE59, 2008. doi:10.1190/1.2972021. URL http://dx.doi.org/10.1190/1.2972021. → page 42

P. Shen, W. W. Symes, and C. C. Stolk. Differential semblance velocity analysis by waveequation migration. *SEG Technical Program Expanded Abstracts 2003*, pages 2132–2135, 2005. doi:10.1190/1.1817759. URL http://library.seg.org/doi/abs/10.1190/1.1817759. → page 42

C. D. Silva and F. J. Herrmann. Optimization on the hierarchical tucker manifold applications to tensor completion. *Linear Algebra and its Applications*, 481:131 – 173, 2015. ISSN 0024-3795. doi:https://doi.org/10.1016/j.laa.2015.04.015. URL http://www.sciencedirect.com/science/article/pii/S0024379515002530. → page 5

B. Smithyman, B. Peters, and F. Herrmann. Constrained waveform inversion of colocated vsp and surface seismic data. In *77th EAGE Conference and Exhibition 2015*, 2015. → pages 7, 25, 36, 38, 41, 44, 79, 125, 128, 129, 137, 151

C. Song, S. Yoon, and V. Pavlovic. Fast admm algorithm for distributed optimization with adaptive penalty. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 753–759. AAAI Press, 2016. URL http://dl.acm.org/citation.cfm?id=3015812.3015924. → pages 88, 135

J. L. Starck, M. Elad, and D. L. Donoho. Image decomposition via the combination of sparse representations and a variational approach. *IEEE Transactions on Image Processing*, 14(10):1570–1582, Oct 2005. ISSN 1057-7149. doi:10.1109/TIP.2005.852206. → pages 126, 127

P. B. Stark. Constraints versus priors. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):586–598, 2015. doi:10.1137/130920721. URL http://dx.doi.org/10.1137/130920721. → pages 36, 151

E. Tadmor, S. Nezzar, and L. Vese. A multiscale image representation using hierarchical (bv,l2) decompositions. *Multiscale Modeling & Simulation*, 2(4):554–579, 2004. doi:10.1137/030600448. URL https://doi.org/10.1137/030600448. → page 128

A. Tarantola. A strategy for nonlinear elastic inversion of seismic reflection data. *GEOPHYSICS*, 51(10):1893–1903, 1986. doi:10.1190/1.1442046. URL http://library.seg.org/doi/abs/10.1190/1.1442046. → pages 1, 76, 139

A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, 2005. doi:10.1137/1.9780898717921. URL https://epubs.siam.org/doi/abs/10.1137/1.9780898717921. → page 151

R. J. Tibshirani. Dykstra's algorithm, admm, and coordinate descent: Connections, insights, and extensions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 517–528. Curran Associates, Inc., 2017. → pages 78, 86, 199

D. Trad. *Five dimensional seismic data interpolation*, pages 978–982. 2008. doi:10.1190/1.3063801. URL https://library.seg.org/doi/abs/10.1190/1.3063801. → page 5

H. Trussell and M. Civanlar. The feasible solution in signal restoration. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2): 201–212, April 1984. ISSN 0096-3518. doi:10.1109/TASSP.1984.1164297. → pages 77, 138

M. Udell, K. Mohan, D. Zeng, J. Hong, S. Diamond, and S. Boyd. Convex optimization in julia. In *2014 First Workshop for High Performance Technical Computing in Dynamic Languages*, pages 18–28, Nov 2014. doi:10.1109/HPTCDL.2014.5. → page 80

E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2): 890–912, 2009. doi:10.1137/080714488. URL https://doi.org/10.1137/080714488. → pages 78, 115, 117

T. van Leeuwen and F. J. Herrmann. Mitigating local minima in full-waveform inversion by expanding the search space. *Geophysical Journal International*, 195:661–667, 10 2013. doi:10.1093/gji/ggt258. → page 27

T. van Leeuwen, A. Y. Aravkin, and F. J. Herrmann. Seismic waveform inversion by stochastic optimization. *International Journal of Geophysics*, 2011, 2011. → page 3

G. Varadhan and D. Manocha. Accurate minkowski sum approximation of polyhedral models. *Graphical Models*, 68(4):343 – 355, 2006. ISSN 1524-0703. doi:https://doi.org/10.1016/j.gmod.2005.11.003. URL http://www.sciencedirect.com/science/article/pii/S1524070306000191. PG2004. → page 128

V. V. Vasin. Relationship of several variational methods for the approximate solution of ill-posed problems. *Mathematical notes of the Academy of Sciences of the USSR*, 7(3):161–165, Mar 1970. ISSN 1573-8876. doi:10.1007/BF01093105. URL https://doi.org/10.1007/BF01093105. → page 78

S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 945–948, Dec 2013. doi:10.1109/GlobalSIP.2013.6737048. → page 123

J. Virieux and S. Operto. An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6):WCC1–WCC26, 2009. URL http://dx.doi.org/10.1190/1.3238367. → pages 1, 76, 139

C. Vogel. *Computational Methods for Inverse Problems*. Society for Industrial and Applied Mathematics, 2002a. doi:10.1137/1.9780898717570. URL http://epubs.siam.org/doi/abs/10.1137/1.9780898717570. → page 35

C. Vogel. *Computational Methods for Inverse Problems*. SIAM, 2002b. → page 21

Q. Wang, X. Zhang, Y. Zhang, and Q. Yi. Augem: Automatically generate high performance dense linear algebra kernels on x86 cpus. In *2013 SC - International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pages 1–12, Nov 2013. doi:10.1145/2503210.2503219. → page 93

X. Wang and C. Navasca. Adaptive low rank approximation for tensors. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015. → page 144

P. Witte, M. Louboutin, K. Lensink, M. Lange, N. Kukreja, F. Luporini, G. Gorman, and F. J. Herrmann. Full-waveform inversion, part 3: Optimization. *The Leading Edge*, 37(2):142–145, 2018.

doi:10.1190/tle37020142.1. URL https://doi.org/10.1190/tle37020142.1. → page 78

M. Wytock, P.-W. Wang, and J. Zico Kolter. Convex programming with fast proximal and linear operators. *ArXiv e-prints*, Nov. 2015. → page 81

S. Xiang and H. Zhang. Efficient edge-guided full waveform inversion by canny edge detection and bilateral filtering algorithms. *Geophysical Journal International*, 2016. doi:10.1093/gji/ggw314. URL http://gji.oxfordjournals.org/content/early/2016/08/22/gji.ggw314.abstract. → page 21

Z. Xu, S. De, M. Figueiredo, C. Studer, and T. Goldstein. An empirical study of admm for nonconvex problems. In *NIPS workshop on nonconvex optimization*, 2016. → pages 92, 131, 135, 193

Z. Xu, M. Figueiredo, and T. Goldstein. Adaptive ADMM with Spectral Penalty Parameter Selection. In A. Singh and J. Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 718–727, Fort Lauderdale, FL, USA, 20–22 Apr 2017a. PMLR. URL http://proceedings.mlr.press/v54/xu17a.html. → pages 82, 90, 92, 93, 135, 193

Z. Xu, M. A. T. Figueiredo, X. Yuan, C. Studer, and T. Goldstein. Adaptive relaxed admm: Convergence theory and practical implementation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017b. → pages 82, 88, 89, 97, 103, 131, 135

Z. Xu, G. Taylor, H. Li, M. A. T. Figueiredo, X. Yuan, and T. Goldstein. Adaptive consensus ADMM for distributed optimization. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3841–3850, International Convention Centre, Sydney, Australia, 06–11 Aug 2017c. PMLR. URL http://proceedings.mlr.press/v70/xu17c.html. → pages 88, 135

Z. Xue and H. Zhu. Full waveform inversion with sparsity constraint in seislet domain. *SEG Technical Program Expanded Abstracts 2015*, pages 1382–1387, 2015. doi:10.1190/segam2015-5932019.1. URL http://library.seg.org/doi/abs/10.1190/segam2015-5932019.1. → pages 6, 41

Z. Xue, Y. Chen, S. Fomel, and J. Sun. Seismic imaging of incomplete data and simultaneous-source data using least-squares reverse time migration with shaping regularization. *GEOPHYSICS*, 81(1):S11–S20, 2016. doi:10.1190/geo2014-0524.1. URL https://doi.org/10.1190/geo2014-0524.1. → page 3

L. Ying, L. Demanet, and E. Candes. 3d discrete curvelet transform. In *Wavelets XI*, volume 5914, page 591413. International Society for Optics and Photonics, 2005. → page 100

P. Yong, W. Liao, J. Huang, and Z. Li. Total variation regularization for seismic waveform inversion using an adaptive primal dual hybrid gradient method. *Inverse Problems*, 34(4):045006, 2018. URL http://stacks.iop.org/0266-5611/34/i=4/a=045006. → pages 103, 109, 125, 128, 129, 137

D. C. Youla and H. Webb. Image restoration by the method of convex projections: Part 1-theory. *IEEE Transactions on Medical Imaging*, 1 (2):81–94, Oct 1982. ISSN 0278-0062. doi:10.1109/TMI.1982.4307555. → pages 77, 138

N. Zeev, O. Savasta, and D. Cores. Non-monotone spectral projected gradient method applied to full waveform inversion. *Geophysical Prospecting*, 54(5):525–534, 2006. ISSN 1365-2478. doi:10.1111/j.1365-2478.2006.00554.x. URL http://dx.doi.org/10.1111/j.1365-2478.2006.00554.x. → pages 7, 36, 37, 57

K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep cnn denoiser prior for image restoration. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2808–2817, July 2017. doi:10.1109/CVPR.2017.300. → page 123

Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer. Novel methods for multilinear data completion and de-noising based on tensor-svd. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. → page 144

M. S. Zhdanov. *Geophysical inverse theory and regularization problems*, volume 36. Elsevier, 2002. → page 35

L. Zhu, E. Liu, and J. H. McClellan. Sparse-promoting full-waveform inversion based on online orthonormal dictionary learning.

GEOPHYSICS, 82(2):R87–R107, 2017. doi:10.1190/geo2015-0632.1. URL http://dx.doi.org/10.1190/geo2015-0632.1. → page 43

Appendix A

Alternating Direction Method of Multipliers (ADMM) for the projection problem.

We show how to use Alternating Direction Method of Multipliers (ADMM) to solve projection problems. Iterative optimization algorithms are necessary in case there is no closed-form solution available. The basic idea is to split a 'complicated' problem into several 'simple' pieces. Consider a function that is the sum of two terms and where one of the terms contains a transform-domain operator: $\min_x h(x) + g(Ax)$. We proceed by renaming one of the variables, $Ax \rightarrow z$ and we also add the constraint $Ax = z$. This new problem is $\min_{x,z} h(x) + g(z)$ s.t. $Ax = z$. The solution of both problems is the same, but algorithms to solve the new formulation are typically simpler. This formulation leads to an algorithm that can solve all projection problems discussed in this thesis. Different projections only need different inputs but require no algorithmic changes.

As an example, consider the projection problem for ℓ_1 constraints in a transform-domain (e.g., total-variation, sparsity in the curvelet domain).

The corresponding set is $\mathcal{C} \equiv \{m \,|\, \|Am\|_1 \leq \sigma\}$ and the associated projection problem is

$$\mathcal{P}_{\mathcal{C}}(m) = \arg\min_x \frac{1}{2}\|x - m\|_2^2 \quad \text{s.t} \quad \|Ax\|_1 \leq \sigma. \tag{A.1}$$

ADMM solves problems with the structure: $\min_{m,z} h(m) + g(z) \text{s.t.} Ax + Bz = c$. The projection problem is of the same form as the ADMM problem. To see this, we use the indicator function on a set $\mathcal{C}$ as

$$\iota_{\mathcal{C}}(m) = \begin{cases} 0 & \text{if } m \in \mathcal{C}, \\ +\infty & \text{if } m \notin \mathcal{C}. \end{cases} \tag{A.2}$$

The indicator function $\iota_{\ell_1}(Am)$ corresponds to the set $\mathcal{C}$ that we introduced above. We use the indicator function and variable splitting to rewrite the projection problem as

$$\begin{aligned} \mathcal{P}_{\mathcal{C}}(m) &= \arg\min_x \frac{1}{2}\|x - m\|_2^2 \quad \text{s.t} \quad \|Ax\|_1 \leq \sigma \\ &= \arg\min_x \frac{1}{2}\|x - m\|_2^2 + \iota_{\ell_1}(Am) \\ &= \arg\min_{x,z} \frac{1}{2}\|x - m\|_2^2 + \iota_{\ell_1}(z) \quad \text{s.t} \quad Ax = z. \end{aligned} \tag{A.3}$$

We have $c = 0$ and $B = -I$ for all projection problems in this thesis. The problem stated in the last line is the sum of two functions acting on different variables with additional equality constraints. This is exactly what ADMM solves. The following derivation is mainly based on Boyd et al. [2011]. Identify $h(x) = \frac{1}{2}\|x - m\|_2^2$ and $g(z) = \iota_{\mathcal{C}}(z)$. ADMM uses the augmented-Lagrangian [Nocedal and Wright, 2000, chapter 17] to include the equality constraints $Ax - z = 0$ as

$$L_\rho(x, z, v) = h(x) + g(z) + v^*(Ax - z) + \frac{\rho}{2}\|Ax - z\|_2^2. \tag{A.4}$$

The scalar ρ is a positive penalty parameter and v is the vector of Lagrangian multipliers. The derivation of the ADMM algorithm is non-trivial, see e.g., Ryu and Boyd [2016] for a derivation. Each ADMM iteration (k) has three main steps:

$$\begin{aligned} x^{k+1} &= \arg\min_x L_\rho(x, z^k, v^k) \\ z^{k+1} &= \arg\min_z L_\rho(x^{k+1}, z, v^k) \\ v^{k+1} &= v^k + \rho(Ax^{k+1} - z^{k+1}). \end{aligned}$$

ADMM will converge to the solution as long as ρ is positive and reaches a stable value eventually. The choice of ρ does influence the number of iterations that are required [Nishihara et al., 2015, Xu et al., 2017a, 2016] and the performance on non-convex problems [Xu et al., 2016]. We use an adaptive strategy to adjust ρ at every iteration, see He et al. [2000]. The derivation proceeds in the scaled form with $u = v/\rho$. Reorganizing the equations leads to

$$\begin{aligned} x^{k+1} &= \arg\min_x \Big(h(x) + \frac{\rho}{2} \|Ax - z^k + u^k\|_2^2 \Big) \\ z^{k+1} &= \arg\min_z \Big(g(z) + \frac{\rho}{2} \|Ax^{k+1} - z + u^k\|_2^2 \Big) \\ u^{k+1} &= u^k + Ax^{k+1} - z^{k+1}. \end{aligned}$$

Now insert the expressions for $h(x)$ and $g(z)$ to obtain the more explicitly defined iterations

$$\begin{aligned} x^{k+1} &= \arg\min_x \Bigg(\frac{1}{2} \|x - m\|_2^2 + \frac{\rho}{2} \|Ax - z^k + u^k\|_2^2 \Bigg) \\ z^{k+1} &= \arg\min_z \Big(\iota_{\mathcal{C}}(z) + \frac{\rho}{2} \|Ax^{k+1} - z + u^k\|_2^2 \Big) \\ u^{k+1} &= u^k + Ax^{k+1} - z^{k+1}. \end{aligned}$$

If we replace the minimization steps with their respective closed-form solutions, we have the following pseudo-algorithm:

$$\begin{aligned} x^{k+1} &= (\rho A^* A + I)^{-1} \big(\rho A^* (z^k - u^k) + m \big) \\ z^{k+1} &= \mathcal{P}_{\mathcal{C}}(Ax^{k+1} + u^k) \\ u^{k+1} &= u^k + Ax^{k+1} - z^{k+1}. \end{aligned}$$

This shows that the second minimization step in the ADMM algorithm to compute a projection is a different projection. The projection part of ADMM for the transform-domain ℓ_1 constraint ($z^{k+1} = \mathcal{P}_{\mathcal{C}}(Ax^{k+1} + u^k) = \arg\min_z 1/2\|z - v\|_2^2$ s.t $\|z\|_1 \leq \sigma$, with $v = Ax^{k+1} + u^k$) is a much simpler problem than the original projection problem (equation A.1) because we do not have the transform-domain operator multiplied with the optimization variable. The x-minimization step is equivalent to the least-squares problem

$$x^{k+1} = \arg\min_x \left\| \begin{pmatrix} \sqrt{\rho} A \\ I \end{pmatrix} x - \begin{pmatrix} \sqrt{\rho}(z^k - u^k) \\ m \end{pmatrix} \right\|_2 \tag{A.5}$$

We can solve the x-minimization problem using direct (QR-factorization) or iterative methods (LSQR [Paige and Saunders, 1982] on the least-squares problem or conjugate-gradient on the normal equations). We adjust the penalty parameter ρ every ADMM cycle. We recommend iterative algorithms for this situation, to avoid recomputing the QR factorization every ADMM iteration. Iterative methods allow for the current estimate of x as the initial guess. Moreover, z and u change less as the ADMM iterations progress, meaning that the previous x is a better and better initial guess. Therefore, the number of LSQR iterations typically decreases as the number of ADMM iterations increases. Algorithm 6 shows the ADMM algorithm to compute projections, including automatic adaptive penalty parameter adjustment. For numerical experiments in this thesis, we use $\mu = 10$, $Au = 2$ as suggested by Boyd et al. [2011].

If we have a different constraint set, but same transform-domain operator, we only change the projector that we pass to ADMM. If the constraint set is the same, but the transform-domain operator is different, we provide a different A to ADMM. Therefore, the various types of transform-domain ℓ_1, cardinality or bound constraints all use ADMM to compute the projection, but with (partially) different inputs.

Algorithm 6 ADMM to compute the projection, including automatic (heuristic) penalty parameter adjustment.

```
input: $m$, transform-domain operator$A$,
norm/bound/cardinality projector $\mathcal{P}_{\mathcal{C}}$
$x_0 = m$, $z_0 = 0$, $u_0 = 0$, $k = 1$,
select $Au > 1$, $\mu > 1$, $\rho > 0$
  WHILE not converged
     $x^{k+1} = (\rho A^* A + I)^{-1}\big(\rho A^*(z^k - u^k) + m\big)$
     $z^{k+1} = \mathcal{P}_{\mathcal{C}}(Ax^{k+1} + u^k)$
     $u^{k+1} = u^k + Ax^{k+1} - z^{k+1}$
      $r = Ax^{k+1} - z^{k+1}$
      $s = \rho A^*(z^{k+1} - z^k)$
     IF $\|r\| > \mu\|s\|$  //increase penalty
        $\rho = \rho Au$
        $u = u/Au$
     IF $\|s\| > \mu\|r\|$  //decrease penalty
        $\rho = \rho/Au$
        $u = uAu$
     ELSE
        $\rho$ //do nothing
     END
  END
output: $x$
```

Appendix B

Transform-domain bounds / slope constraints

Our main interest in transform-domain bound constraints originates from the special case of slope constraints, see, e.g., Petersson and Sigmund [1998] and Bauschke and Koch [2015] for examples from computational design. Lelivre and Oldenburg [2009] propose a transform-domain bound constraint in a geophysical context, but use interior point algorithms for implementation. In our context, slope means the model parameter variation per distance unit, over a predefined path in the model. For example, the slope of the 2D model parameters in the vertical direction (z-direction) form the constraint set

$$\mathcal{C} \equiv \{m \mid b^l_j \leq ((D_z \otimes I_x)m)_j \leq b^u_j\}, \tag{B.1}$$

with Kronecker product $\otimes$, identity matrix with dimension equal to the x-direction I_x, and D_z is a 1D finite-difference matrix corresponding to the z-direction. b^l_j is element j of the lower bound vector. An appealing property of this constraint is the physical meaning in a pointwise sense. If the model parameters are acoustic velocity in meters per second and the grid is also in units of meters, the constraint then defines the maximum velocity increment/decrement per meter in a direction. This type of direct physical meaning of a constraint is not available for ℓ_1, rank or Nuclear norm constraints; those constraints assign a single scalar value to a property of the entire model.

There are different modes of operation of the slope constraint:

Approximate monotonicity. The acoustic velocity generally increases with depth inside the Earth. This means the parameter values increase (approximately) monotonically with dept (positivity of the vertical discrete gradient). The set $\mathcal{C} \equiv \{m \mid -\varepsilon \leq ((D_z \otimes I_x)m)_j \leq +\infty\}$ describes this situation, where $\varepsilon > 0$ is a small number. Exact monotonicity corresponds to $\varepsilon = 0$, which means we allow the model parameter values to increase arbitrarily fast with increasing depth, but enforce a slow decrease of parameter values when looking into the depth direction.

Smoothness. We obtain a type of smoothness by setting both bounds to small numbers: $\mathcal{C} \equiv \{m \mid -\varepsilon_1 \leq ((D_z \otimes I_x)m)_j \leq +\varepsilon_2\}$, where $\varepsilon_1 > 0$, $\varepsilon_2 > 0$ are small numbers. This type of smoothness results in a different projection problem than if smoothness is obtained using constraints based on norms or subspaces. Another difference is that the slope constraint is inherently locally defined.

The slope constraint may be defined along any path using any discrete derivative matrix. Higher order derivatives lead to bounds on different properties. Approximate monotonicity of parameter values can also be obtained using other constraints. Esser et al. [2016b] use the norm based hinge-loss constraint. However, we prefer to work with linear inequalities because norm based constraints are not defined pointwise and do not have the direct physical interpretation as described above. Figure B.1 shows what happens when we project a velocity model onto the different slope constraint sets.

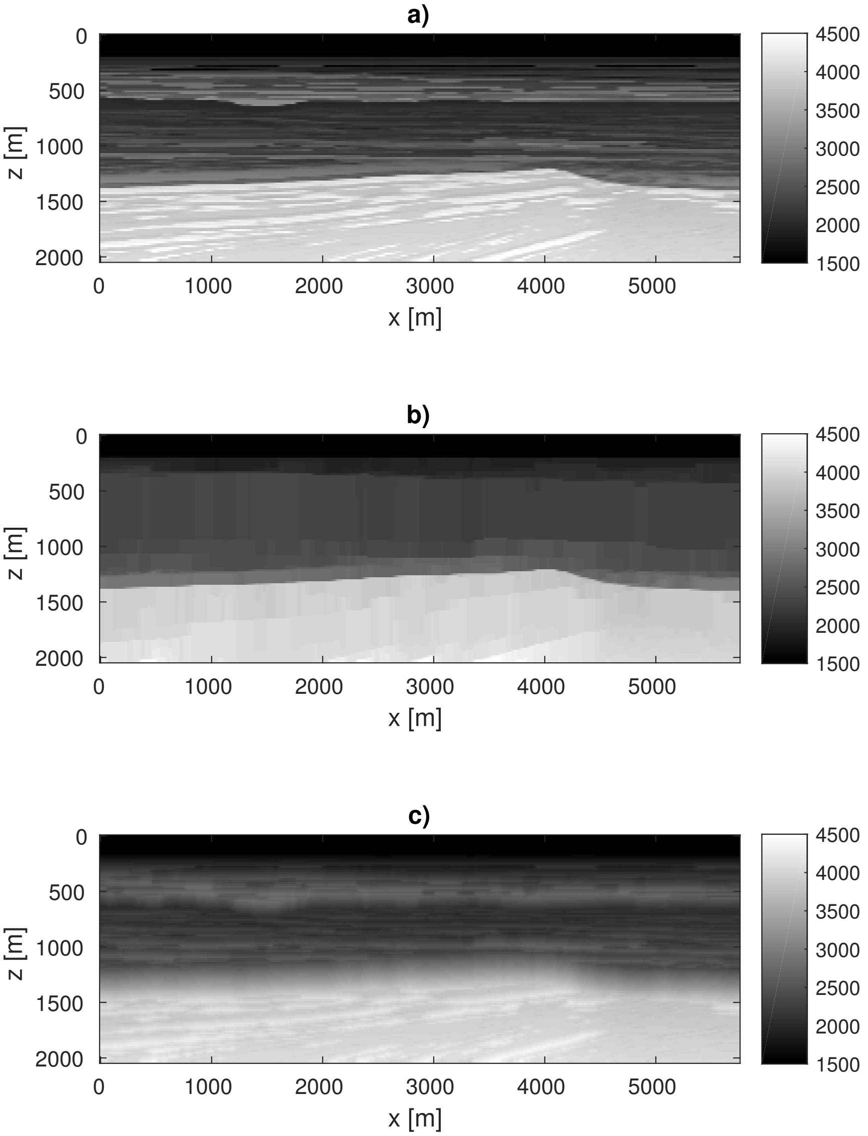


Figure B.1: The figure shows the effect of different slope constraints when we project a velocity model (a). Figure (b) shows the effect of allowing arbitrary velocity increase with depth, but only slow velocity decrease with depth. Lateral smoothness (c) is obtained by bounding the upper and lower limit on the velocity change per distance interval in the lateral direction.

Appendix C

Black-box alternating projection methods

We briefly show that the proposed PARSDMM algorithm (Algorithm 3) is different, but closely related to black-box alternating projection algorithms for the projection onto an intersection of sets. We base this Appendix on the alternating direction method of multipliers (ADMM). The ADMM algorithm is closely related to Dykstra's algorithm [Dykstra, 1983, Boyle and Dykstra, 1986] for projection problems, as described by [Bauschke and Koch, 2015, Tibshirani, 2017], including the conditions that lead to equivalency.

The parallel Dykstra algorithm (Algorithm 7) projects the vector $m \in \mathbb{R}^N$ onto an intersection of p sets using projections onto each set separately with projectors $\mathcal{P}_{\mathcal{V}_1}, \mathcal{P}_{\mathcal{V}_2}, \ldots, \mathcal{P}_{\mathcal{V}_p}$. If the definitions of the sets $\mathcal{V}_i$ include non-orthogonal linear operators, these projections are often non-trivial and their computation requires another iterative algorithm.

To show the similarity and difference with PARSDMM and parallel Dykstra, we proceed with a derivation similar to Algorithm 3, but different in such a way that the final algorithm is black-box, i.e., it uses projections onto the sets $\mathcal{V}_i$ and the linear operators are 'hidden'.

First we rewrite the projection problem of m onto the intersection of sets

Algorithm 7 Parallel Dykstra's algorithm to compute $\arg\min_x \frac{1}{2}\|x - m\|_2^2$ s.t. $x \in \bigcap_{i=1}^{p} \mathcal{V}_i$.

```
Algorithm Parallel-DYKSTRA($m, \mathcal{P}_{\mathcal{V}_1}, \mathcal{P}_{\mathcal{V}_2}, \dots, \mathcal{P}_{\mathcal{V}_p}$)
input:
  model to project: $m$
  projectors onto sets $\mathcal{P}_{\mathcal{V}_1}, \mathcal{P}_{\mathcal{V}_2}, \dots, \mathcal{P}_{\mathcal{V}_p}$
//initialize
0a.   $x^0 = m$, $k = 1$
0b.   $v_i^0 = x^0$ for $i = 1, 2, \dots, p$
0c.   select weights $\rho_i$ such that $\sum_{i=1}^{p} \rho_i = 1$
  while stopping conditions not satisfied do
      FOR $i = 1, 2, \dots, p$
1.          $y_i^{k+1} = \mathcal{P}_{\mathcal{V}_i}(v_i^k)$
      END
2.    $x^{k+1} = \sum_{i=1}^{p} \rho_i y_i^{k+1}$
      FOR $i = 1, 2, \dots, p$
3.          $v_i^{k+1} = x^{k+1} + v_i^k - y_i^{k+1}$
      END
4.    $k \leftarrow k + 1$
  END
output: $x$
```

$\mathcal{V}_i$,

$$\min_x \frac{1}{2}\|x - m\|_2^2 + \sum_{i=1}^{p-1} \iota_{\mathcal{V}_i}(x) \tag{C.1}$$

as

$$\min_x \frac{1}{2}\|x - m\|_2^2 + \sum_{i=1}^{p-1} \iota_{\mathcal{C}_i}(A_i x). \tag{C.2}$$

Where we exposed linear operators A_i by rewriting the indicator functions $\iota_{\mathcal{V}_i}(x) \rightarrow \iota_{\mathcal{C}_i}(A_i x)$. Now we introduce additional variables and equality constraints to set up a parallel algorithm as

$$\min_{x,\{y_i\}} \frac{1}{2}\|y_p - m\|_2^2 + \sum_{i=1}^{p-1} \iota_{\mathcal{C}_i}(A_i y_i) \quad \text{s.t.} \quad x = y_i \; \forall i. \tag{C.3}$$

This problem is suitable for solving with ADMM if we recast it as

$$\min_{x,\tilde{y}} \tilde{f}(\tilde{A}\tilde{y}) \quad \text{s.t.} \quad \tilde{D}x = \tilde{y}, \tag{C.4}$$

with

$$\tilde{f}(\tilde{y}) \equiv \frac{1}{2}\|y_p - m\|_2^2 + \sum_{i=1}^{p-1} \iota_{\mathcal{C}_i}(A_i y_i) \tag{C.5}$$

and

$$\tilde{D} \equiv \begin{pmatrix} I_1 \\ \vdots \\ I_p \end{pmatrix}, \quad \tilde{y} \equiv \begin{pmatrix} y_1 \\ \vdots \\ y_p \end{pmatrix}, \quad \tilde{A} \equiv \begin{pmatrix} A_1 \\ \vdots \\ A_p \end{pmatrix}. \tag{C.6}$$

The linear equality constraints enforce that all y_i are copies of x at the solution of problem (C.3). The difference with PARSDMM is that we leave the A_i inside the indicator functions instead of moving them to the linear equality constraints. The corresponding augmented Lagrangian with penalty parameters $\rho_i > 0$ is

$$L_{\rho_1,\dots,\rho_p}(x, y_1, \dots, y_p, v_1, \dots, v_p) = \sum_{i=1}^{p} \Bigg[\tilde{f}_i(A_i y_i) + v_i^\top (y_i - x) + \frac{\rho_i}{2}\|y_i - x\|_2^2 \Bigg]. \tag{C.7}$$

The ADMM iterations with a relaxation parameters γ_i are then given by

$$\begin{aligned}
x^{k+1} &= \arg\min_x \sum_{i=1}^p \Big[\frac{\rho_i^k}{2}\|y_i^k - x + \frac{v_i^k}{\rho_i^k}\|_2^2\Big] \\
&= \frac{\sum_{i=1}^p \Big[\rho_i^k y_i^k + v_i^k\Big]}{\sum_{i=1}^p \rho_i^k} \\
\bar{x}_i^{k+1} &= \gamma_i^k x_i^{k+1} + (1-\gamma_i^k) y_i^k \\
y_i^{k+1} &= \arg\min_{y_i} \Big[f_i(A_i y_i) + \frac{\rho_i}{2}\|y_i - \bar{x}_i^{k+1} + \frac{v_i^k}{\rho_i^k}\|_2^2\Big] \\
&= \operatorname{prox}_{f_i \circ A_i, \rho_i^k}(\bar{x}_i^{k+1} - \frac{v_i^k}{\rho_i^k}) \\
v_i^{k+1} &= v_i^k + \rho_i^k (y_i^{k+1} - \bar{x}_i^{k+1}).
\end{aligned}$$

The difference with Algorithm 3 is that the linear operators A_i move from the x^{k+1} computation to the y_i^{k+1} computation. This means the x^{k+1} computation is now a simple averaging step instead of a linear system solution. The y_i^{k+1} changed from evaluating proximal maps (almost always in closed-form), into evaluations of proximal maps involving linear operators (usually not known in closed-form). The proximal maps $\operatorname{prox}_{f_i \circ A_i, \rho_i^k}$ for $i = 1, \dots, p-1$ are projections onto $\mathcal{V}_i$, except for $i = p$, which is the proximal map for $\frac{1}{2}\|y_p - m\|_2^2$. We need another iterative algorithm to compute the y_i^{k+1} at relatively high computation cost. The algorithm as a whole becomes more complicated, because we need additional stopping criteria for the algorithm that computes the y_i updates.

This iterations from (C.8) are similar to parallel Dykstra (Algorithm 7) and are, in essence, ADMM applied to a standard consensus form optimization problem [Boyd et al., 2011, problem 7.1].