Regularization of Linear Systems with Sparsity Constraints with Applications to Large Scale Inverse Problems

Sergey Voronin

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Program in
Applied and Computational Mathematics

Adviser: Ingrid Daubechies

November 2012

Abstract

This thesis is about numerical methods for the regularization of large scale inverse problems with sparsity constraints. Some new methods are proposed, and applied to an inverse problem from Geotomography, the goal of which is to determine latitudinal and longitudinal corrections to a spherically symmetric wave velocity model of the Earth's interior. The problem involves a very large, badly conditioned linear system, whose solutions, expressed in an intricate coordinate system, can be sparsely represented under the action of a wavelet transformation. The methods we develop and analyze in this thesis are simple to implement, efficient and easy to parallelize on large machines. In addition, the convergence analysis for the new algorithms assumes minimal conditions on the linear systems they are applied to.

This thesis is organized as follows. After the introduction, we give in Chapter 2, an overview of existing schemes for regularization with sparsity constraints, and we introduce new material developed in the remainder of the thesis. Chapter 3 introduces a new firm thresholding based scheme that overcomes some shortcomings of soft thresholding; this scheme applies less penalty to the large coefficients of the iterates, while producing solutions of comparable sparsity. Chapter 4 introduces two novel methods based on an iteratively reweighted least squares strategy. These methods are designed to minimize a new more general sparsity promoting functional, which is especially useful for structured sparse problems, such as those encountered under the action of a wavelet transform. Detailed convergence analysis is provided for these two new algorithms. Chapter 5 discusses techniques that are useful for numerical implementation, such as a fast implementation of a randomized low rank SVD approximation and matrix column norm estimations, useful for large badly conditioned matrices. Finally, Chapter 6 presents the application, collecting ideas from the previous chapters and applying them to the inverse problem.

Acknowledgements

I would like to sincerely acknowledge the contribution of many people on my work and life while in graduate school. These include (but by no means exhaust): Ingrid Daubechies, Ignace Loris, Hugo Woerdeman, Guust Nolet, Frederik Simons, Holger Rauhut, Rick Chartrand, Jeroen Tromp, Jean Charlety, Cedric Vonesch, and Waheed Bajwa. Amongst these, I would like to extend particular thanks to my advisor, Ingrid Daubechies, for greatly enriching my academic understanding and for helping to arrange trips to visit others, which has significantly enhanced my work and graduate experience. I would like to offer particular thanks to Ignace Loris, Guust Nolet, Holger Rauhut, and Rick Chartrand for hosting me on extended research visits. I would like to thank the system administrators at Princeton and Duke: Josko Plazonic and Andrew Schretter, most notably for their help in setting up the machines that I have extensively used for my numerical work. Finally I would like to thank my family for their continuous support and encouragement.

Contents

Chapter 1

INTRODUCTION TO THE THESIS

1.1 General Remarks

In this chapter, we offer some general remarks about the thesis. We conclude the general discussion with a brief layout of the different chapters. We then describe the mathematical notation used in the rest of the thesis.

We first describe the general idea. In this thesis, we discuss techniques (such as iterative algorithms) useful for large inverse problems whose inversion step comes down to the solution of a system of linear equations $\bar{A}\bar{x} = \bar{b}$. This categorizes a large number of inverse problems. We investigate iterative methods that can be used to find solutions to such systems under the assumptions of large dimensions of $\bar{A}$, noise (in the right hand side $\bar{b}$ and possibly even in the matrix $\bar{A}$) and ill conditioning of $\bar{A}$ (characterized by the sharp dropoff of its singular values), particularly when the system is under-determined (has more columns than rows). We also pay particular attention to the case when the solution x can be sparsely represented in a properly chosen basis. The application alluded to in this thesis comes from Geotomography. In this application, we are interested in determining corrections to a spherically symmetric seismic wave velocity model (one that varies only with depth). A more complete model, which accounts for longitude and latitude as well as depth, would lead to better understanding of the Earth's interior structure. The inversion step for this applications boils down to a large under-determined linear system such as the one described above. The methods developed in this thesis, however, are applicable to a wide variety of different problems.

There are several points that we make note of and discuss here. These form the core properties of the methods we consider and analyze in the later chapters. First of all the size of the matrix in our applications is generally very large. For our Seismic Geotomography application, the dimensions of the matrix in the linear system are roughly half a million rows by three million columns. Although the matrix is sparse, the size of the matrix in a common sparse format is still over one hundred gigabytes. This effectively means that operations with such a matrix are time consuming and moreover, are only possible to perform on parallel machines with access to a large amount of RAM. We remark here that when parallel codes are at play, simplicity and transparency of the methods used is key to error-free implementation and customization. There are many techniques from convex analysis that could be applied to the problem at hand. We focus on relatively simple methods in this thesis because these are preferred by many in large scale applications, which already have such a high level of complication that many would prefer to avoid convoluted algorithms. The methods we consider involve predominantly matrix vector operations which can be readily parallelized.

Besides the large size of the systems, the data vector b almost always has noise. In general b is derived from observations, and there is always an associated error or uncertainty in the measurements. Thus, we can write: $b = \bar{b} + \text{noise}$. The true right hand side $\bar{b}$ is typically unknown; the matrix A itself is also frequently based on computations which may contain approximations. Hence, given only the noisy b and possibly noisy A (not the exact $\bar{b}$ and $\bar{A}$), we cannot compute a solution $\bar{x}$ satisfying the system exactly. Instead we look for a solution x that in some useful way approximates the true solution $\bar{x}$. Typically, we look for a solution with a particular low residual value $||Ax - b||_2$ based on some criterion, for example an a-priori estimate ν of the norm of the noise in b ($\nu \approx ||\text{noise}||_2$) or some χ^2 value (which we discuss in Chapter 6).

A third issue is that of ill-conditioning. Many matrices arising from physical inverse problems are not well conditioned. This means that their singular values fall off rapidly (usually in a non-linear fashion) and the ratio of the largest and smallest singular values, called the condition number, is very large. There are different reasons for this ill-conditioning depending on the application. In the case of our application in Geotomography, the matrix columns correspond to locations within the Earth (parametrized by a specific choice of coordinate system). Data are based on the recordings of seismic waves observed by receivers on the surface after earthquakes; the data is thus geographically limited, particularly by the location of seismic sensors. The Earth is covered mostly by water and there are few sensors in these locations; on the other hand many geological sensors are located in the United States and in the territory of the former Soviet Union. This uneven spread of data sensors is often responsible for ill-conditioning in matrices derived from such data. The result of ill conditioning is that small changes in the original system produce large changes in the solution. This effectively means that noise in the right hand side vector b can cause the solution x to be very different from $\bar{x}$ unless proper regularization methods are used.

Another aspect of physical inverse problems is that data is typically quite limited. That means that at the end we want to solve for more unknowns than there are observations and the matrix A above would have more columns than rows. In such a case, the matrix A has a non-trivial kernel, and there are infinitely many solutions that have a suitably small residual value $||Ax - b||$. Thus, to obtain a single solution, additional constraints must be set. This is again addressed by regularization; the question reduces to a judicious choice of the constraints to introduce. The classical constraint is that the two-norm of the solution be made small. This can be accomplished by minimizing $||Ax - b||_2^2 + \lambda||x||_2^2$; it is well known that this leads to an effective regularization method, which has the advantage that it can be implemented in an arbitrary choice of orthonormal basis. When feasible, the solution is computed in the orthonormal basis associated with the singular value decomposition (SVD) of A; appropriate truncation leads to a solution that is sparse with respect to this basis. In this thesis we make a particular focus on sparsity, but not with respect to the SVD basis. This can be accomplished by replacing the two-norm penalty above by a different penalty, for example, using the one-norm $||x||_1$. Sparsity is a popular and useful constraint because in many instances the data can be sparsely represented in some basis and the storage and manipulation of sparse data, especially that which is high-dimensional, is both faster and easier. Imposing sparsity often also gets rid of the small coefficients of the solution, a lot of which correspond to noise. Care must be taken to choose the right regularization parameter (such as λ) above to come up with an acceptable solution. This often takes multiple runs and is a substantial time constraint. The basis with respect to which we wish to impose sparsity has to be picked carefully: the solution is typically not sparse in its "standard" basis. However, it can often be expressed as a sparse solution under the action of some transform. In this thesis and for our Geotomography application, we make use of the wavelet transform. Wavelets are known for their compressive power and are used extensively in imaging and other areas. Functions that consist of a combination of smoothly varying large-scale features on which much more localized spiky features are superimposed, typically can be approximated well by sparse wavelet expansions. As a simple example consider the model below and its three representations with different numbers of wavelet coefficients (Figure 1.1):

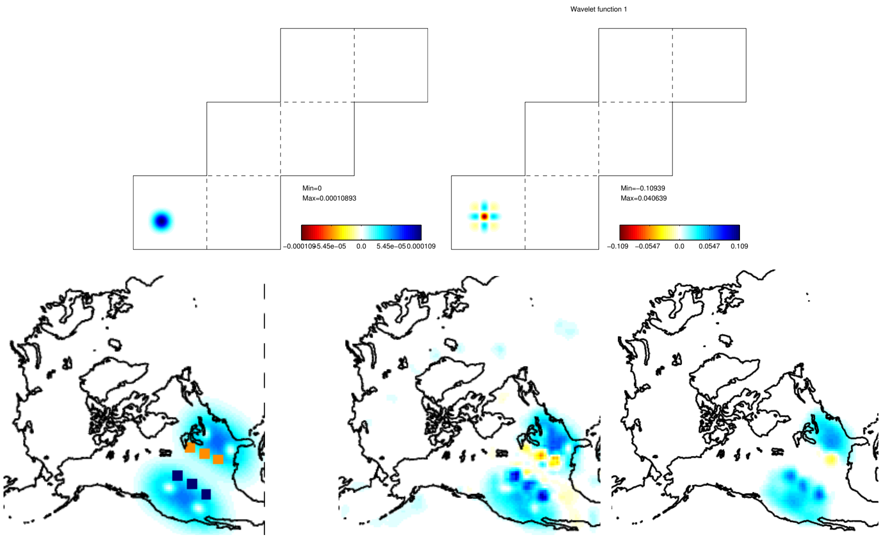


Figure 1.1: Scaling and Wavelet Function. Original model with 135000 nonzero coefficients, Reconstructions with 10000 and 160 wavelet coefficients.

These particular wavelets are able to represent the model quite accurately with just a few (of the largest) coefficients. Since we expect the solutions to the inverse problem to consist of such combinations of smooth features and spikes, it is plausible to look for sparse solutions in the wavelet domain.

Another issue is the complexity of the underlying coordinate system. The vector x expressed as a regular column vector is one dimensional. However the data it represents is often inherently multi-dimensional. Consider for example the cubed sphere coordinate system [38] which is used in our Geotomography application. It is a way of representing data on a sphere such as the Earth, by projecting at each depth on the surface of an inscribed cube:

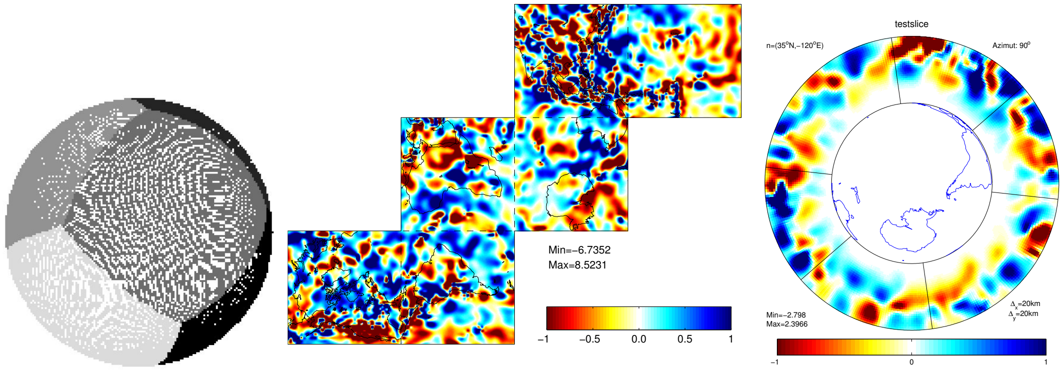


Figure 1.2: The Cubed Sphere (figure courtesy of Frederik Simons) and Sample Model on a Cubed Sphere Grid

We comment on this in more detail in a later chapter, here we simply want to point out that the solution x may indeed be a high dimensional vector in a complicated coordinate system. When the Earth is gridded via such a system as the above, each coordinate corresponds to a chunk identifier, a 2-d location on the outer surface of the chunk, and a depth. From Figure 1.1 we can see that crossing chunk boundaries correspond to big jumps in such a coordinate system. When looking for the solution in such a coordinate system we may require special treatment at the boundaries. Another aspect that calls for attention is the resolution power of the matrix. The matrix is typically such that it can only resolve x in a certain range of the coordinate system. That is, even if the expected true solution x to a physical problem contains data in all chunks, the action of the matrix on x may be sensitive to only part of x, and the information in A may just not be able to "reproduce" the other parts of x. This occurs in our Geotomography application because some columns of the matrix have very small norm and contain little information. Consider for example, the plot of column sums below for a sample matrix:

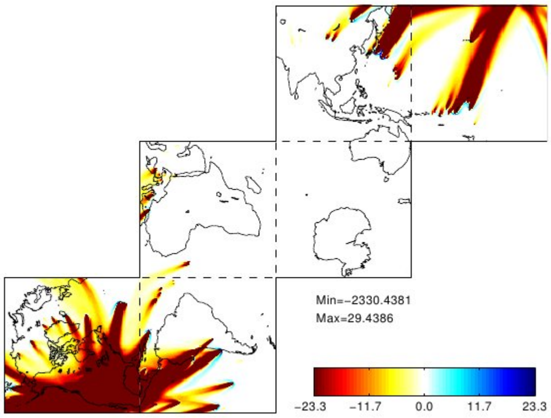


Figure 1.3: Column Sums of the Matrix

From the above figure it is clear that we would not recover anything useful beyond the above range. Hence, to obtain a meaningful solution, we would like to penalize coordinates outside the colored range with a greater sparse penalty, that is we would require those parts of the solution to be effectively zero or close to zero so that the solution is constructed over the meaningful part.

Figure 1.1 shows that a model is to a large degree compressible with the wavelet basis used there. However, there are multiple choices for wavelet transforms and perhaps some parts of a more complicated model are represented best without wavelets at all. For this reason, allowing an inversion algorithm to pick from a set of bases and assign different weights to them in some kind of a linear combination is beneficial to recovering complicated sparse solutions. We shall address this as well.

Without mentioning mathematical details, we have now made a list of issues and desirable characteristics to consider for our methods which we recall in a brief list below:

- The matrix A is very large, possibly inexact, underdetermined and ill conditioned.

- The data vector b contains noise. Since the matrix A is ill conditioned, this can lead to big differences between the obtained and noiseless solutions if regularization is not used.

- In many instances, solutions can be sparsely represented when a suitable transform is used.

- We must do multiple runs to determine the right penalty parameter. This can take a lot of time.

- Not all parts of the solution are created equal. This can be due to the coordinate system structure and the structure of the matrix. We need to be able to treat different coefficients differently.

- There are many choices of transforms (for example many different types of wavelet bases), we would like to use a combination of several and let the algorithm decide on the combination rather than pick a single family.

This thesis is about regularization algorithms designed to address these issues. The developed techniques are then applied to the inverse problem in the application. We present numerical algorithms and techniques which can be readily coded and tested as well as detailed analytical derivations. We also develop a numerical framework, including various software, for tackling the large scale computations from the inverse problem. We now briefly describe the layout of the thesis, counting this Introduction as Chapter 1. At the end of this chapter is a section on the notation used in the thesis. Chapter 2 gives a detailed mathematical introduction to regularization and sparsity, existing classes of algorithms, and new ideas, that are presented in more detail in the later chapters. Chapter 3 introduces a new algorithm called FIVTA, which is an analogue of the FISTA method introduced in Chapter 2. This method uses a new two-parameter thresholding function and is faster converging in practice, since it can produce comparable solutions to FISTA with a higher value for the regularization parameter. It is also less sensitive to the parameter value. Chapter 4 introduces two new algorithms that can be used to minimize a newly proposed generalized functional that allows us to treat different parts of the solution in different ways. The algorithms are based on a reweighted least squares idea, which approximates the nonsmooth portion of the functional with a weighted two norm. Detailed convergence analysis is exhibited for these two algorithms. Chapter 5 presents some numerical comparisons between different methods and includes some interesting ideas for the methods introduced in the previous chapters. It also presents a randomized approach to approximating matrix column norms and a fast randomized low rank SVD approximation algorithm. Finally, Chapter 6 discusses the application in Geotomography. It describes the inverse problem and shows how the methods exhibited earlier can be applied to its solution. The thesis is concluded by a few pages of summary and concluding remarks along with references. An Appendix at the end of the thesis exhibits the pseudocode for some of the algorithms mentioned in the thesis, particularly for all the newly introduced methods.

1.2 Notation

We now introduce the notation relevant to this thesis. The set of all real numbers is denoted by $\mathbb{R}$, and the set of all nonnegative numbers is denoted by $\mathbb{R}_+$. By extension of notation, we have that the set of all positive numbers is defined by $\mathbb{R}_{++}$. The set of all N-real vectors is denoted by $\mathbb{R}^N$. An element $v \in \mathbb{R}^N$ consists of N real numbers stacked in a columnwise order. A matrix $M \in \mathbb{R}^{m \times N}$ represents an object of m rows and N columns. We use the capital N instead of the lowercase n because most matrices we encounter in this thesis are underdetermined, which means that $m < N$.

For a differentiable real valued function $f : \mathbb{R}^{n_1+n_2} \to \mathbb{R}$ and vectors $x \in \mathbb{R}^{n_1}, y \in \mathbb{R}^{n_2}$, the vector $\nabla_x f(x,y) \in \mathbb{R}^{n_1}$ is defined as $\left(\frac{\partial}{\partial x_1} f(x,y), \frac{\partial}{\partial x_2} f(x,y), \ldots, \frac{\partial}{\partial x_{n_1}} f(x,y)\right)$. This notation extends to functions of a single variable or of more than two variables.

A norm on $\mathcal{V} = \mathbb{R}^N$ or $\mathbb{R}^{m \times N}$ is a real valued function $||\cdot|| : \mathcal{V} \to \mathbb{R}$ that satisfies the following criteria:

(1) $||x|| \geq 0$ for all $x \in \mathcal{V}$, and $||x|| = 0$ if and only if $x = 0$.

(2) $||\lambda x|| = |\lambda| ||x||$ for all $x \in \mathcal{V}$ and $\lambda \in \mathbb{R}$.

(3) $||x + y|| \leq ||x|| + ||y||$ for all $x, y \in \mathcal{V}$.

In $\mathbb{R}^N$, we are particularly interested in the family of so called ℓ_p-norms for $p \geq 1$ (including $p = \infty$). For any $p \geq 1$, the ℓ_p-norm of a vector $x \in \mathbb{R}^N$ is defined as

$$||x||_p = \left(\sum_{k=1}^{n} x_k^p\right)^{1/p}.$$

The ℓ_∞-norm is defined as: $||x||_\infty = \max_i |x_i|$. Other norms are explicitly defined in the thesis. We will make reference also to the so called ℓ_0 norm, which counts the number of nonzero elements of a vector. We note later in this thesis that this is in fact not a real norm and is a slight abuse of notation, yet we use this notation due to its popularity. When the type of norm is not specified, as in $||y||$, we assume the use of the ℓ_2 norm.

The space $\mathbb{R}^N$ as an inner product space, with the inner product between two elements x and y defined as $\langle x, y \rangle = x^T y = \sum_{i=1}^{N} x_i y_i$. For example, the ℓ_2-norm is induced by the Euclidean inner product, in the sense that $||x||_2 = \sqrt{\langle x, x \rangle}$ for any $x \in \mathbb{R}^N$. For any norm $||\cdot||$ on $\mathbb{R}^N$, we can define the corresponding dual norm $||\cdot||_*$ (with respect to the Euclidean inner product): for any $y \in \mathbb{R}^N$,

$$||y||_* = \max_{x \in \mathbb{R}^n, x \neq 0} \frac{\langle y, x \rangle}{||x||}.$$

$||\cdot||_*$ is a well-defined real valued function on $\mathbb{R}^N$. It is well known that for any $p \geq 0$, the dual norm of the ℓ_p-norm is the ℓ_q-norm, where q satisfies $p^{-1} + q^{-1} = 1$ (and $q = \infty$ if $p = 1$).

Next we present some definitions and basic notion about real matrices. For any vector $x = (x_1, x_2, \ldots, x_N) \in \mathbb{R}^N$, $D := \mathrm{Diag}(x) = \mathrm{Diag}(x_1, x_2, \ldots, x_N) \in \mathbb{R}^{N \times N}$ is defined as:

$$D_{ij} = \begin{cases} x_i & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

We will make use of the singular value decomposition (SVD) of a real matrix $A \in \mathbb{R}^{m \times N}$: if A is of rank r, then there exist $U \in \mathbb{R}^{m \times r}$, $V \in \mathbb{R}^{N \times r}$ and $\Sigma \in \mathbb{R}^{r \times r}$ such that

(1) $U^T U = I$, $V^T V = I$,

(2) $\Sigma = \mathrm{Diag}(\sigma_1, \sigma_2, \ldots, \sigma_r) \in \mathbb{R}^{k \times k}$ is a diagonal matrix with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$, and

(3) $A = U \Sigma V^T$.

This is known as the economic form of the SVD [25]. For $1 \leq i \leq \min\{m, N\}$, the i-th largest singular value of A is defined to be σ_i, with $\sigma_j = 0$ for $j = r+1, \ldots, \min\{m, N\}$ whenever $r < \min\{m, N\}$. For convenience the following notation is also used:

the largest singular value of A: $\sigma_{\max}(A) = \sigma_1$;

the smallest singular value of A: $\sigma_{\min}(A) = \sigma_{\min\{m,N\}}$.

The spectral norm of A is defined as $||A||_2 = \sigma_{\max}(A)$. The generalized inverse of $A \in \mathbb{R}^{m \times N}$ with SVD $A = U\Sigma V^T$, is defined as $A^+ = V\Sigma^{-1}U^T$ (and $\Sigma^{-1} = \text{Diag}(\sigma_1^{-1}, \sigma_2^{-1}, \ldots, \sigma_r^{-1}) \in \mathbb{R}^{k \times k}$).

By the matrix W we refer to a linear transform. We use W because in this thesis we deal primarily with wavelet transforms, but the concepts apply equally well to other linear transforms. By the notation AW^{-1} which appears frequently in the thesis, we mean a product of two matrices A and W^{-1}, the latter representing an inverse transform. Note that explicit knowledge of W or W^{-1} is not required and is in fact never used in this thesis; only the end result of the application of W, W^{-1}, and $(W^{-1})^T$ to vectors is necessary.

We now describe what we mean by the minimization problems posed in this thesis. Given functions $f, g : \mathbb{R}^n \rightarrow \mathbb{R}$, a minimization problem in the form $\min_x f(x)$ s.t. $g(x) = 0$ refers to finding the minimum value of $f(x)$ that can be attained among all $x \in \mathbb{R}^n$ that satisfies $g(x) = 0$. We denote the optimal value by $\min_x\{f(x) : g(x) = 0\}$, and the set of optimal solutions by $\arg\min_x\{f(x) : g(x) = 0\}$. If the minimization problem is assumed to have only one optimal solution $\bar{x}$, we write $\bar{x} = \arg\min_x\{f(x) : g(x) = 0\}$ (rather than $\bar{x} \in \arg\min_x\{f(x) : g(x) = 0\}$). Sometimes, the uniqueness of the optimal solution may depend on the properties of f. For instance, f may be a function of a matrix and the uniqueness may depend on the properties of this matrix. Additionally, we would be concerned mostly with unconstrained problems. For such problems, the constraint $g(x) = 0$ does not exist. In many places in this thesis, we would thus use the notation $\bar{x} = \arg\min_x\{f(x)\}$. By this statement, we mean that we are interested to find a vector $\bar{x}$ which minimizes the function $f(x)$. There might be another minimizer $\bar{y}$ which produces the same value. That is, it may be possible that $f(\bar{x}) = f(\bar{y})$ and $\bar{x} \neq \bar{y}$. By the statement, $\bar{x} = \arg\min_x\{f(x)\}$, we mean that we are interested in finding a single minimizing vector $\bar{x}$ for f such that $f(\bar{x}) \leq f(y)$ for all vectors y.

Chapter 2

REGULARIZATION, SPARSITY AND ALGORITHMS

2.1 Overview

This chapter is a mathematical introduction into the rest of the thesis. We give an introduction to regularization of linear systems, sparsity, and different schemes for sparse regularization that can be used for large scale problems without strong conditions on the matrix. Since there exist many available algorithms, we try to survey a few different categories to which most of the existing methods can be assigned. We also introduce the areas in which the later chapters of the thesis make new contributions.

2.2 Regularization

We now proceed to give a motivation for regularization, in the context of ill-conditioned systems and noisy data vectors. We are particularly interested in systems that are under-determined, as motivated in the introduction. When the system $Ax = b$ is under-determined, having more columns than rows, an infinite number of solutions of the system exist. In this case, we wish to impose constraints on the solution to pick a particular solution amongst many. Since A is not a square matrix, it only has a generalized inverse as defined below. We start by introducing some notation. We have $A \in \mathbb{R}^{m \times n}$ with $m < n$ and nonzero $b \in \mathbb{R}^m$ which is noisy (equal to the noiseless unknown $\bar{b}$ plus noise). We are interested in the case when A is ill-conditioned, which means that the ratio between the largest and smallest non-zero singular values $(\sigma_{\max}(A)/\sigma_{\min}(A))$ is large.

We first consider the under-determined form of A, setting aside the issue of noise in b. In this case, the system $Ax = b$ has infinitely many solutions generically and additional constraints must be imposed for uniqueness. The simplest and most classical constraint to put on the solution is the minimum of the ℓ_2-norm. Consider then, solving the problem $\min_x ||x||_2$ s.t. $Ax = b$. It turns out that the solution is given in terms of the generalized inverse A^+b. Using the theory of Lagrange multipliers, we define the Lagrangian:

$$L(x, y) = ||x||_2^2 + y^T(Ax - b),$$

where y is the vector of Lagrange Multipliers. Taking the gradient we get:

$$\nabla_x L(x, y) = 2x + A^T y = 0 \implies x = -\frac{1}{2}A^T y.$$

We plug this into the constraint $Ax = b$ which, using $A = U\Sigma V^T$ leads to:

$$A(-\frac{1}{2}A^T y) = b \implies -\frac{1}{2}U\Sigma^2 U^T y = b \implies U^T y = -2\Sigma^{-2}U^T b,$$

so the pseudo-inverse solution we get is:

$$\begin{aligned} x &= -\frac{1}{2}A^T y = -\frac{1}{2}(U\Sigma V^T)^T y = -\frac{1}{2}V\Sigma U^T y \\ &= -\frac{1}{2}V\Sigma(-2\Sigma^{-2}U^T b) = V\Sigma^{-1}U^T b = A^+ b. \end{aligned}$$

This simple and naive solution, however, can give rather meaningless results when the system is not well behaved. When A is ill-conditioned, many singular values σ_k will be very small, so that the matrix Σ^{-1} which has diagonal terms $\frac{1}{\sigma_k}$ will have very large entries. In this case the generalized inverse matrix, when computed with finite precision arithmetic, will not give an accurate solution.

Next, we consider the issue of noise. We can also see that this naive solution x will be very sensitive to errors in the right hand side vector b. This is apparent from the covariance matrix of the solution, as we now show. For this, we suppose that $b = \bar{b} + e$ where the noise vector e behaves like white noise. Its different entries are then uncorrelated, each having mean 0 and standard deviation ν. If in addition the elements of $\bar{b}$ and e are uncorrelated we have:

$$\operatorname{Cov}(e) = E[(e - E[e])(e - E(e))^T] = E[ee^T] = \nu^2 I,$$

and:

$$\operatorname{Cov}(b) = E[(b - E[b])(b - E(b))^T] = E[ee^T] = \nu^2 I.$$

A property of covariance is that for a random vector v and a matrix A, $\operatorname{Cov}(Av) = A\operatorname{Cov}(v)A^T$. We may then derive the spectral norm (i.e., the largest singular value) of the covariance matrix $||\operatorname{Cov}(x)||$:

$$\operatorname{Cov}(x) = \operatorname{Cov}(A^+ b) = A^+ \operatorname{Cov}(b)(A^+)^T = \nu^2(A^T A)^+ \implies ||\operatorname{Cov}(x)||_2 = \frac{\nu^2}{\sigma_{\min}^2},$$

where $\sigma_{\min}$ is the smallest singular value of A. We see from the above that if A is not well conditioned, the covariance matrix is likely to have very large elements, since the smallest singular value of A will be small. This indicates that $x = A^+b$ is very sensitive to data errors.

In the presence of noise when the true right hand side $\bar{b}$ is not known, we may not even be able to find a solution to $Ax = b$ because the matrix and the right hand side b (including the noise) may simply be incompatible (i.e. $b \notin Ran(A)$) so that the problem has no solution. Instead we generally try to minimize $||Ax - b||_2^2$ (possibly with additional constraints, as we will see below) and the generalized inverse can also be used for this optimization problem. The function $||Ax - b||_2^2$ is convex and differentiable, so the minimum satisfies:

$$\nabla_x ||Ax - b||_2^2 = 0 \implies 2A^T(Ax - b) = 0 \implies A^TAx = A^Tb.$$

In fact, a common choice of solution of $A^TAx = A^Tb$ would be directly through the generalized inverse:

$$x = (A^TA)^+A^Tb = (V\Sigma^{-2}V^T)V\Sigma U^Tb = A^+b, \tag{2.2.1}$$

because of all the solutions to $A^TAx = A^Tb$, A^+b has the smallest ℓ_2-norm: $A^TAx = A^Tb$ if and only if $x = A^+b + d$ for some $d \in \ker(A^TA) = \text{range}(A^TA)^\perp = \text{range}(A^+)^\perp$, and

$$\|A^+b + d\|^2 = \|A^+b\|^2 + 2d^T(A^+b) + \|d\|^2 = \|A^+b\|^2 + \|d\|^2 \geq \|A^+b\|^2.$$

The covariance matrix of the least squares solution is thus as given above:

$$\text{Cov}(x) = \nu^2(A^TA)^{-1}.$$

Indeed, any solution $x = A^+b + d$ with $d \in \ker(A^TA)$ has the same covariance matrix and thus suffers from the same problem when A is ill conditioned.

We now look at the most classical case of regularization: Tikhonov ℓ_2 regularization. We will not be interested in purely ℓ_2 regularization in this thesis because it does not lead to sparse solutions, but we will take away some ideas from Tikhonov regularization. Here we replace the constrained system $Ax = b$ by the minimization of $||Ax-b||_2$ with a constraint on the ℓ_2-norm of the solution $||x||_2$. That is we would like to minimize $||Ax - b||_2$ while keeping $||x||_2$ below some number, say α. Equivalently, we can minimize $||x||_2$ and keep $||Ax - b||_2$ below some number β. By the theory of Lagrange multipliers we can show that these problems are equivalent to solving:

$$\min_x ||Ax - b||_2^2 + \lambda ||x||_2^2$$

for some suitable regularization parameter λ. Since both terms of the above are quadratic, we can take the gradient and obtain the solution in linear form:

$$2A^T(Ax - b) + 2\lambda x_t = 0 \implies (A^TA + \lambda I)x_t = A^Tb \implies x_t = (A^TA + \lambda I)^{-1}A^Tb.$$

Above, this inverse matrix is generally used only if it has a particularly simple form. If this is not the case, we use a method like conjugate gradients to solve the linear system. The benefit of the above formulation is that it filters out the effects of small singular values. We can see this by plugging in the SVD, $A = U\Sigma V^T$ into the

Tikhonov solution x_t to obtain:

$$
\begin{aligned}
x_t &= \left((U\Sigma V^T)^T(U\Sigma V^T) + \lambda I\right)^{-1} A^T b \\
&= \left(V\Sigma^2 V^T + \lambda V I V^T\right)^{-1} (U\Sigma V^T)^T b \\
&= \left(V(\Sigma^2 + \lambda I)V^T\right)^{-1} V\Sigma U^T b \\
&= \left(V(\Sigma^2 + \lambda I)^{-1} V^T\right) V\Sigma U^T b \\
&= V(\Sigma^2 + \lambda I)^{-1}\Sigma U^T b \\
&= V \operatorname{Diag}\left(\frac{\sigma_i}{\sigma_i^2 + \lambda}\right) U^T b.
\end{aligned}
$$

We see that the effect of the regularization is to filter the small singular values σ_i, by replacing each σ_i by $\frac{\sigma_i}{\sigma_i^2+\lambda}$, which prevents the singular values smaller than λ from dominating the solution. Next, we can also see that the covariance matrix for the solution compares favorably to that of the naive solution (assuming the same white noise conditions). Letting $D = \operatorname{Diag}(\frac{\sigma_i}{\sigma_i^2+\lambda})$,

$$
\operatorname{Cov}(x_t) = \operatorname{Cov}(VDU^T b) = VDU^T \operatorname{Cov}(b) UDV^T = \nu^2 VDU^T UDV^T = \nu^2 V D^2 V^T.
$$

In fact, we have that:

$$
|| \operatorname{Cov}(x_t)||_2 = \nu^2 \|D^2\|_2 \leq \frac{\nu^2}{4\lambda}.
$$

This is because the function $h(t) := \frac{t}{t^2+\lambda}$ achieves a maximum at $t = \sqrt{\lambda}$, with the value $\frac{1}{2\sqrt{\lambda}}$.

From the above description, the benefit of the simple ℓ_2 regularization is quite clear as it allows us to compute reasonable solutions when the matrix A is not well conditioned and when the right hand side data vector b has noise. From the introduction, however, we know that we are interested in cases where the solutions are sparse. We discuss the concept of sparsity in the following subsection.

2.3 Sparsity

We now discuss the concept of sparsity and sparse solutions. We have seen what sparsity is in Chapter 1, where we saw an example of a model that is sparse under the action of a wavelet transform. The most direct measure of sparsity is the number of nonzero entries of a vector, sometimes called its ℓ_0-norm, $||x||_0$. Note that this is in fact not a proper norm; consider for example that:

$$3 = ||(2,2,2)||_0 \neq |2|||(1,1,1)||_0 = 2 \times 3 = 6.$$

It is instructive to know that we can approximate the ℓ_0-norm using the ℓ_p-norm as $p \to 0$; we can define the ℓ_0-norm as [20]:

$$||x||_0 = \lim_{p\to 0} ||x||_p^p = \lim_{p\to 0} \sum_{k=1}^{N} |x_k|^p.$$

This is motivated by the picture below where we display the graph of the function $f(x_k, p) = |x_k|^p$ for different values of p between 0 and 2. From this plot, we can see that as p approaches zero $f(x_k, p)$ approaches the indicator function which is 1 for nonzero x_k, so that $\lim_{p\to 0} ||x||_p^p$ counts the nonzeros of the signal. We also plot below a picture of projections unto the ℓ_1 and ℓ_2 balls. These show the solutions to the minimization problems:

$$\min\{|x| + |y| : a_1 x + b_1 y = c_1\} \quad \text{and} \quad \min\{x^2 + y^2 : a_2 x + b_2 y = c_2\}.$$

The first problem corresponds to ℓ_1 minimization in two dimensions and the second to ℓ_2 minimization. The intersection point for the first problem has a zero x or y component (hence a sparse solution). For the ℓ_2 case, the solution would be sparse only if the slope of the line is zero or infinity. This gives some motivation for why the choice $p < 2$ in the penalty $\sum_{k=1}^{N} |x_k|^p$ leads to sparser solutions than $p = 2$.

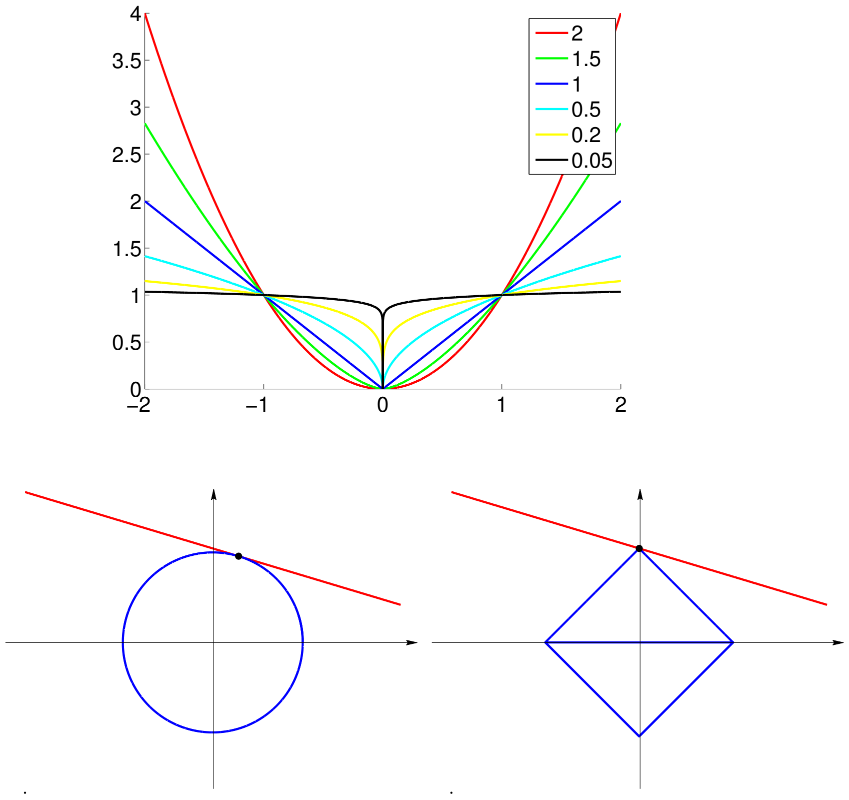


Figure 2.1: $|x|^p$ plotted for $p = 2, 1.5, 1, 0.5, 0.2, 0.05$. Illustration of ℓ_2 minimization and ℓ_1 minimization in $\mathbb{R}^2$: we observe a sparse solution in the ℓ_1 case.

For the constrained case, the two corresponding minimization problems giving sparse solution are:

$$\min_x ||x||_0 \text{ s.t. } Ax = b, \tag{P_0}$$

$$\min_x ||x||_1 \text{ s.t. } Ax = b. \tag{P_1}$$

The main result from the theory of compressed sensing [7, 8, 15, 16, 18, 24] is that under certain conditions on A, the ostensibly NP-hard problem (P_0) and the convex problem (P_1) give identical answers. The conditions depend on the so called *restricted isometry property* (RIP) of the matrix A. If there exists a constant σ_k such that for every k-sparse vector $x \in \mathbb{R}^N$ the following holds:

$$(1-\sigma_k)||x||_2^2 \leq ||Ax||_2^2 \leq (1+\sigma_k)||x||_2^2, \tag{2.3.1}$$

then the matrix A is said to satisfy the k-restricted isometry property with restricted isometry constant σ_k [8]. The equivalence result from compressed sensing is as follows: Let σ_k be the smallest number such that (2.3.1) holds for all k-sparse vector $x \in \mathbb{R}^N$. If $\sigma_k < \sqrt{2}-1$, then for all k-sparse vectors x such that $Ax = b$, the solution of (P_1) is equal to the solution of (P_0) [8]. This landmark result is particularly important because the norm $||x||_1$ is the closest (convex) norm to $||x||_0$ and, although neither $||x||_0$ nor $||x||_1$ are smooth, dealing with (P_1) is substantially easier. The reason for this is that for convex functions, local optimality conditions (i.e. $f(x) \leq f(x+tv)$ for $t \in \mathbb{R}$) are enough to determine the global minimum value. Lastly ℓ_1 minimization has proven regularization benefits [11]. We note here also that under some special conditions minimzing the ℓ_1 norm may not give sparse solutions. An example is mentioned in Chapter 3, with a matrix and right hand side vector of all ones. However, in most practical cases one considers, minimizing the ℓ_1 norm leads to a sparse solution.

The Restricted Isometry Property is a very restrictive condition on the matrix A. The RIP in the compressive sensing sense above will not be satisfied if there exists for example a sparse vector in the null space of A (in that case $||Ax|| = 0$ and it is not greater than $(1-\sigma_k)||x||_2^2$). This can occur when a small number of columns of A are linearly dependent, more precisely when one of the columns can be expressed as a linear combination of the $(k-1)$ others. In fact, a non-random matrix that is not well conditioned and coming from a physical inverse problem like the one mentioned in the Introduction is highly unlikely to satisfy the RIP. In addition, it is difficult to check if in fact a matrix does satisfy it. On the other hand, even if ℓ_1 minimization does not give the sparsest solution, it still does give reasonable sparse solutions for most cases we encounter, including for systems with not well conditioned matrices and noisy right hand sides.

The above discussion gives rise to two functionals that parallel those of Tikhonov regularization, replacing the ℓ_2 penalty by one involving the ℓ_0 or the ℓ_1-norm:

$$||Ax - b||_2^2 + 2\tau||x||_0 \quad \text{and} \quad ||Ax - b||_2^2 + 2\tau||x||_1.$$

The second is substantially easier to deal with than the first because it is convex - for any $\gamma \in (0, 1)$:

$$||\gamma x + (1-\gamma)y||_1 \leq ||\gamma x||_1 + ||(1-\gamma)y||_1 = \gamma||x||_1 + (1-\gamma)||y||_1$$

The conditions for the global minimizer (the local optimality conditions) can easily be determined, but unlike for ℓ_2 minimization they cannot be expressed in a linear form. Below, we derive the optimality conditions for the ℓ_1 functional but first we make a small comment. In the previous section we mentioned that in our application the models are sparse under the action of some (wavelet) transform. This means that we do not necessarily expect x to be sparse, but we do expect $w = Wx$ to be sparse where W denotes the wavelet transform matrix. Thus, we would instead want to minimize $||Mx - b||_2^2 + 2\tau||Wx||_1$, but with the substitution $x = W^{-1}w$ this takes the same form as above: $||MW^{-1}w - b||_2^2 + 2\tau||w||_1$ where we take $A = MW^{-1}$. Thus, even if the sparsity is induced by a transform, as is true in many applications, the above form of the functionals is applicable for the analysis.

Lemma 2.3.1. *The necessary and sufficient component-wise optimality conditions for the minimizer of the functional* $F(x) = ||Ax - b||_2^2 + 2\tau||x||_1$*, where* $\tau > 0$*, are:*

$$\begin{array}{rcll} [A^T(b - Ax)]_k & = & \tau \operatorname{sgn}(x_k) & , \forall\, k \textit{ with } x_k \neq 0, \\ |\left(A^T(b - Ax)\right)_k| & \leq & \tau & , \forall\, k \textit{ with } x_k = 0. \end{array} \tag{2.3.2}$$

Proof. The conditions stated above are for a general vector x with components x_k for $k \in (1, \ldots, N)$ some of whose components are zero and others are nonzero. We derive N conditions below, one for each index k. First note that F is convex and hence every local minimizer is a global minimizer. Suppose x is a local minimizer of F. Then for any $t \in \mathbb{R}$ and $z \in \mathbb{R}^N$, $F(x) \leq F(x + tz)$ holds (since x is assumed to be a minimizer), which implies:

$$t^2||Az||_2^2 + 2t\langle z, A^T(Ax - b)\rangle + 2\tau\left(||x + tz||_1 - ||x||_1\right) \geq 0. \tag{2.3.3}$$

Note that $||x||_1 = \sum_{k=1}^{N} |x_k| = \sum_{x_k \neq 0} x_k \operatorname{sgn}(x_k)$, and if $x_k \neq 0$, then $\operatorname{sgn}(x_k + tz_k) = \operatorname{sgn}(x_k)$ for small t. So for small enough t,

$$||x + tz||_1 = \sum_{x_k \neq 0} (x_k + tz_k) \operatorname{sgn}(x_k) + \sum_{x_k = 0} |tz_k| = ||x||_1 + t \sum_{x_k \neq 0} \operatorname{sgn}(x_k) z_k + |t| \sum_{x_k = 0} |z_k|. \tag{2.3.4}$$

Then for small $t \neq 0$, (2.3.3) becomes:

$$t^2||Az||_2^2 + 2t\langle z, A^T(Ax - b)\rangle + 2\tau t \left(\sum_{x_k \neq 0} \operatorname{sgn}(x_k) z_k + \operatorname{sgn}(t) \sum_{x_k = 0} |z_k| \right) \geq 0.$$

The first term can be made arbitrarily small compared to the other so that we require:

$$2t\left(\langle z, A^T(Ax - b)\rangle + \tau \sum_{x_k \neq 0} \operatorname{sgn}(x_k) z_k \right) + 2\tau|t| \sum_{x_k = 0} |z_k| \geq 0. \tag{2.3.5}$$

To get the k-th condition, fix a $z = z_k e_k$ with $e_k = (0, \ldots, 1, \ldots, 0)$ with a 1 in the k-th position and an arbitrary nonzero z_k. If $x_k \neq 0$, then substituting this z into (2.3.5) gives:

$$2t\left(\langle z, A^T(Ax-b)\rangle + \tau\,\mathrm{sgn}(x_k)z_k\right) \geq 0$$

Since this holds for both positive and negative t we must have:

$$z_k\langle e_k, A^T(Ax-b)\rangle + \tau\,\mathrm{sgn}(x_k)z_k = 0.$$

which reduces to:

$$\left(A^T(b-Ax)\right)_k = \tau\,\mathrm{sgn}(x_k) \quad ; \quad x_k \neq 0$$

Next, for $x_k = 0$ we have, substituting $z = z_k e_k$ into (2.3.5) that:

$$2|t|\left(\mathrm{sgn}(t)z_k\langle e_k, A^T(Ax-b)\rangle + \tau|z_k|\right) \geq 0$$

Since $t \neq 0$ we must have that:

$$\mathrm{sgn}(t)z_k\langle e_k, A^T(Ax-b)\rangle + \tau|z_k| \geq 0$$

But t can be positive or negative, so we end up with the condition $|(A^T(b-Ax))_k| \leq \tau$ for $x_k = 0$.

For the other direction, suppose x satisfies (2.3.2). We need to prove that it is then a minimizer of F. Since F is convex, any local minimizer is necessarily global and we show that x is a local minimizer. For any $z \in \mathbb{R}^N$ and $t \in \mathbb{R}$,

$$\begin{aligned}\langle tz, A^T(Ax-b)\rangle &\geq -t\sum_{x_k\neq 0}\tau z_k\,\mathrm{sgn}(x_k) - |t|\sum_{x_k=0}|z_k||A^T(Ax-b)_k| \\ &\geq -\tau t\sum_{x_k\neq 0}\mathrm{sgn}(x_k)z_k - \tau|t|\sum_{x_k\neq 0}|z_k|.\end{aligned}$$

For small enough nonzero t, (2.3.4) holds, so

$$\begin{aligned} & ||A(x+tz)-b||_2^2 + 2\tau||x+tz||_1 \\ = \; & ||Ax-b||_2^2 + 2t\langle z, A^T(Ax-b)\rangle + t^2||Az||_2^2 + 2||x||_1 \\ & \quad +2\tau t \sum_{x_k \neq 0} \text{sgn}(x_k)z_k + 2\tau|t| \sum_{x_k=0} |z_k| \\ \geq \; & ||Ax-b||_2^2 + 2\tau||x||_1. \end{aligned}$$

Therefore x is a global minimizer of F. This completes both directions of the proof.

□

It is not straightforward to find solutions for these nonlinear equations. We may note, however, that $x = 0$ is a solution for $\tau \geq \max_k(|(A^Tb)_k|)$, which we can see by plugging in $x = 0$ into the optimality conditions. One approach to finding a solution compatible with the optimality condition is to start with $x = 0$ and at $\tau_1 = \max_k(|(A^Tb)_k|)$. Then for steps $i \geq 2$, we decrease τ and pick nonzero components of x such that the above conditions are satisfied. This principle underlies the LARS-LASSO algorithms [19]; these methods however tend to become very slow as matrices get large. A faster converging coordinate wise method, especially in cases a good estimate of the support set is available does exist. We discuss randomized coordinate descent later in this chapter.

We now discuss some methods for the minimization of the ℓ_1 functional:

$$F(x) = \min_x ||Ax-b||_2^2 + 2\tau||x||_1. \tag{2.3.6}$$

The main difficulty with the above ℓ_1 functional is the non smooth term $||x||_1 = \sum_{k=1}^N |x_k|$. Otherwise, if all the terms of the functional could be differentiated, a number of different methods such as steepest descent or conjugate gradients could then be used on the derivative set equal to zero. As this is not possible, two approaches are available: either to deal directly with the non-smooth parts by means of the so called soft thresholding operator, which we introduce below, or to approximate the non-smooth portion ($||x||_1$) by a smooth approximation. In the latter case, we are able to utilize the same approaches as discussed above, but the quality depends on the quality of the approximation. Another possibility is to deal with the dual problem of ℓ_1 minimization, which we also discuss. We mention also the coordinate descent method, which works by optimizing a single entry of the functional at a time, keeping the others fixed. We split our discussion into separate subsections. In each subsection we cover a different category of the methods. We also introduce the new approaches that are discussed in detail in later chapters of the thesis.

2.4 Algorithms for ℓ_1 minimization: Soft Thresholding

In this section we consider methods that treat the non-smooth terms directly, by means of the soft thresholding operator.

Definition 2.4.1. *The soft thresholding operator S_τ is defined by:*

$$S_\tau(x) = \begin{cases} x - \tau, & x \geq \tau; \\ 0, & -\tau \leq x \leq \tau; \\ x + \tau, & x \leq -\tau \ . \end{cases}$$

The operator can be defined for a vector component-wise as $(\mathbb{S}_\tau(x))_k = S_\tau(x_k) \quad \forall\, k = 1, \ldots, N$.

Lemma 2.4.2. *The soft thresholding operator satisfies the minimization problem:*

$$\mathbb{S}_\tau(b) = \arg\min_x ||x-b||^2 + 2\tau||x||_1.$$

The soft thresholding operator is also non-expansive:

$$||\mathbb{S}_\tau(x) - \mathbb{S}_\tau(y)||_2 \leq ||x-y||_2,$$

and for any $\alpha \in \mathbb{R}$:

$$\alpha\mathbb{S}_\tau(x) = \mathbb{S}_{\alpha\tau}(\alpha x).$$

Proof. Since:

$$||x-b||_2^2 + 2\tau||x||_1 = \sum_{k=1}^{N} \left((x_k - b_k)^2 + 2\tau|x_k|\right),$$

the proof follows by looking at the one dimensional case of the above functional, namely the function:

$$f(x) = (x-b)^2 + 2\tau|x|.$$

First note that f is a strictly convex function and has a unique minimizer. Suppose $b \geq \tau$. In this case, $S_\tau(b) = b - \tau$. Since $f(b-\tau) = \tau^2 + 2\tau(b-\tau) = 2b\tau - \tau^2$, for all $x \in \mathbb{R}$, $(|x| \geq x)$ and so:

$$\begin{aligned} f(x) &\geq x^2 - 2bx + b^2 + 2\tau x \\ &= x^2 - 2(b-\tau)x + (b-\tau)^2 - (b-\tau)^2 + b^2 \\ &= (x-(b-\tau))^2 + 2b\tau - \tau^2 \\ &\geq 2b\tau - \tau^2 = f(b-\tau) = f(S_\tau(b)), \end{aligned}$$

so that $S_\tau(b)$ is a minimizer of f in this case. Next, suppose $b \leq -\tau$. Since $f(b+\tau) = \tau^2 - 2\tau(b+\tau) = -2b\tau - \tau^2$, for all $x \in \mathbb{R}$, $(|x| \geq -x)$ and so:

$$\begin{aligned} f(x) &\geq x^2 - 2bx + b^2 - 2\tau x \\ &= x^2 - 2(b+\tau)x + (b+\tau)^2 - (b+\tau)^2 + b^2 \\ &= (x - (b+\tau))^2 - 2b\tau - \tau^2 \\ &\geq -2b\tau - \tau^2 = f(b+\tau) = f(S_\tau(b)). \end{aligned}$$

Finally, if $|b| \leq \tau$, then $2\tau|x| \geq 2|b||x| = 2|bx|$ and so:

$$f(x) \geq (x-b)^2 + 2|bx| = x^2 + 2(|bx| - bx) + b^2 \geq b^2 = f(0) = f(S_\tau(b)).$$

Hence, the minimizer of the function is given by the soft-thresholding operator applied to b. The result now follows, since soft thresholding applied to a vector is a component wise operation applied to each component.

For the proof that the soft thresholding operator is non-expansive, we must look at the different cases which depend on the values of x and y with respect to τ. Consider for example the case when x and y are both greater than τ. Then $S_\tau(x) = x - \tau$ and $S_\tau(y) = y - \tau$. Then we have that:

$$||S_\tau(x) - S_\tau(y)||_2 = ||x - \tau - (y - \tau)||_2 = ||x - y||_2.$$

Next consider the case $x > \tau$ and $y < -\tau$. Then $S_\tau(x) = x - \tau$ and $S_\tau(y) = y + \tau$ and:

$$||S_\tau(x) - S_\tau(y)||_2 = ||x - \tau - (y + \tau)||_2 = ||(x-y) - 2\tau||_2 \leq ||x - y||_2,$$

since in this case $(x - y) > 2\tau$. The proof follows by direct verification in all the different cases. The proof that

$$\alpha \mathbb{S}_\tau(x) = \mathbb{S}_{\alpha\tau}(\alpha x)$$

also follows by direct verification. □

We now discuss the ISTA and FISTA schemes [2], which rely directly on Lemma 2.4.2. These iterative algorithms involve the use of the soft thresholding function. We first introduce ISTA, the Iterative Soft Thresholding Algorithm using a surrogate functional approach. This concept also applies to other algorithms discussed in this thesis. The main idea is to add and subtract terms from the functional $F(x) = ||Ax - b||_2^2 + 2\tau||x||_1$ such that the minimization is easier to carry out. For instance, we would like to get rid of the $||Ax||_2^2$ terms from the first term in the functional. We use the method of so called majorization-minimization, to force the successive iterates to reduce the value of the functional. The value of this approach is that it allows us to deduce easily imortant properties of the algorithm such as $||x^{n+1} - x^n||_2 \to 0$ and the boundedness of x^n. We reuse this approach in later chapters to analyze more complicated algorithms which we introduce. This majorization-minimization approach works by picking a two parameter function $G(x, y)$ with the following properties:

$$G(x, y) \geq F(x) \;\; \forall\, x, y, \quad \text{and} \quad G(x, x) = F(x), \tag{2.4.1}$$

and defining the algorithm by: $x^{n+1} = \arg\min G(x, x^n)$. The first inequality above implies in particular that $F(x^{n+1}) \leq G(x^{n+1}, x^n)$. Thus, by the above definition for x^{n+1} it follows that:

$$F(x^{n+1}) \leq G(x^{n+1}, x^n) \leq G(x^n, x^n) = F(x^n),$$

and so such an algorithm leads to a decrease of the value of the cost functional. The difficult part is to construct a suitable function G.

For the ℓ_1 functional, a suitable function G can be defined by:

$$G(x,y) = ||Ax-b||_2^2 - ||A(x-y)||_2^2 + ||x-y||_2^2 + 2\tau||x||_1. \tag{2.4.2}$$

Note that if $||A||_2 < 1$, then:

$$||A(x-y)||_2^2 \leq ||A||_2^2||x-y||_2^2 \leq ||x-y||_2^2 \implies G(x,y) \geq F(x) \;\; \forall\, x,y.$$

Also, $G(x,x) = F(x)$ for all x. Thus G satisfies the criteria in (2.4.1).

Expanding (2.4.2) we have:

$$\begin{aligned} G(x,y) &= ||Ax||_2^2 - 2\langle Ax, b\rangle + ||b||_2^2 \\ &- ||Ax||_2^2 + 2\langle Ax, Ay\rangle - ||Ay||_2^2 + ||x||_2^2 - 2\langle x, y\rangle + ||y||_2^2 + 2\tau||x||_1 \\ &= ||x||_2^2 - 2\langle x, y + A^T(b-Ay)\rangle + 2\tau||x||_1 - ||Ay||_2^2 + ||y||_2^2 + ||b||_2^2 \\ &= ||x - (A^Tb - A^TAy + y)||_2^2 + 2\tau||x||_1 \\ &\qquad + \left(||y||_2^2 + ||b||_2^2 - ||Ay||_2^2 - ||y + A^Tb + A^TAy||_2^2\right). \end{aligned}$$

Hence given $y = x^n$,

$$\begin{aligned} \arg\min_x G(x,x^n) &= \arg\min_x ||x - \left(A^Tb - A^TAx^n + x^n\right)||^2 + 2\tau||x||_1 \\ &= \mathbb{S}_\tau\left(A^Tb - A^TAx^n + x^n\right), \end{aligned}$$

which leads to the scheme:

$$x^{n+1} = \mathbb{S}_\tau\left(x^n + A^Tb - A^TAx^n\right). \tag{ISTA}$$

We now want to show that the iterates of (ISTA) converge to a minimizer of the ℓ_1 functional. We will prove that this is true for all converging subsequences of the iterates and we will show that a converging subsequence exists. Global convergence algorithms for this algorithm can be found in [11]. First, we need a lemma that characterizes the fixed point for an ℓ_1 minimizer. We need the notion of subdifferential, an analogue of derivatives but for non-differentiable convex functions.

Definition 2.4.3. *The subdifferential of a convex function $f : \mathbb{R}^N \to \mathbb{R}$ at $x \in \mathbb{R}^N$ is the set*

$$\partial f(x) = \left\{ d \in \mathbb{R}^N : f(y) \geq f(x) + \langle d, y - x \rangle, \ \forall \, y \right\}.$$

Lemma 2.4.4. *If f is differentiable at x, then $\partial f(x) = \{\nabla f(x)\}$. $\bar{x}$ is a minimizer of f if and only if $0 \in \partial f(\bar{x})$.*

More properties of subdifferentials can be found in [36]. Now we state the lemma which characterizes the minimizer. We would then show that the limit point of converging subsequences satisfies this fixed point form.

Lemma 2.4.5. *The minimizer of the ℓ_1 functional $F(x) = ||Ax - b||_2^2 + 2\tau ||x||_1$ is characterized by the relation:*

$$x = \mathbb{S}_\tau (x + A^T b - A^T A x).$$

Proof. We recall first the previous result:

$$\mathbb{S}_\tau(c) = \arg\min_x \left\{ ||x - c||^2 + 2\tau ||x||_1 \right\} = \arg\min_x \left\{ \sum_{k=1}^{N} (x_k - c_k)^2 + 2\tau \sum_{k=1}^{N} |x_k| \right\}.$$

Now using subdifferentials we have that since $\partial_x ||x - c||_2^2 = \{2(x - c)\}$ and by the above lemma:

$$\bar{x} \in \arg\min_x ||x - c||_2^2 + 2\tau ||x||_1 \iff 0 \in 2(\bar{x} - c) + 2\tau \partial \|\bar{x}\|_1. \tag{2.4.3}$$

Since $\mathbb{S}_\tau(c) = \arg\min_x ||x-c||_2^2 + 2\tau||x||_1$ we have that:

$$0 \in (\bar{x} - c) + \tau\partial_x \|\bar{x}\|_1 \iff \bar{x} = \mathbb{S}_\tau(c).$$

Now we have that $\bar{x}$ is a minimizer of the convex functional $||Ax-b||_2^2 + 2\tau||x||_1$ if and only if:

$$0 \in A^T(A\bar{x} - b) + \tau\partial\|\bar{x}\|_1 = \big(\bar{x} - (\bar{x} + A^Tb - A^TA\bar{x})\big) + \tau\partial\|\bar{x}\|_1.$$

Setting $c = \bar{x} + A^Tb - A^TA\bar{x}$ and using the above argument the result:

$$\bar{x} = \arg\min_x \|x - (\bar{x} + A^Tb - A^TA\bar{x})\|^2 + 2\tau\|x\|_1 = \mathbb{S}_\tau(\bar{x} + A^Tb - A^TA\bar{x})$$

follows. □

Now we prove a lemma about the iterates of ISTA.

Lemma 2.4.6. *For the ISTA scheme defined through the majorization-minimization procedure:*

$$x^{n+1} = \arg\min_x G(x, x^n) = \mathbb{S}_\tau\left(x^n + A^Tb - A^TAx^n\right)$$

with $||A||_2 < 1$ *and*

$$G(x,y) = ||Ax-b||_2^2 + ||x-y||_2^2 - ||A(x-y)||_2^2 + 2\tau||x||_1,$$

we have that $||x^n - x^{n+1}||_2 \to 0$ *and that the iterates* (x^n) *are bounded.*

Proof. Notice that for $||A||_2 < 1$ we have:

$$||A(x-y)||_2^2 \le ||A||_2^2||x-y||_2^2 \le ||x-y||_2^2 \implies ||x-y||_2^2 - ||A(x-y)||_2^2 \ge 0. \quad (2.4.4)$$

Now we can write:

$$\begin{aligned} G(x^{n+1}, x^n) &= ||Ax^{n+1} - b||_2^2 + ||x^{n+1} - x^n||_2^2 - ||A(x^n - x^{n+1})||_2^2 + 2\tau||x^{n+1}||_1 \\ G(x^{n+1}, x^{n+1}) &= ||Ax^{n+1} - b||_2^2 + 2\tau||x^{n+1}||_1. \end{aligned}$$

Then

$$G(x^{n+1}, x^n) - G(x^{n+1}, x^{n+1}) = ||x^{n+1} - x^n||_2^2 - ||A(x^n - x^{n+1}||_2^2,$$

and hence, we have that:

$$G(x^{n+1}, x^{n+1}) \leq G(x^{n+1}, x^n) \leq G(x^n, x^n)$$

where the first inequality follows from (2.4.4) and the second follows from $x^{n+1} = \arg\min_x G(x, x^n)$. Now we can write the following telescoping sum:

$$\begin{aligned} \sum_{n=1}^{P} \big(G(x^{n+1}, x^n) - G(x^{n+1}, x^{n+1})\big) &\leq \sum_{k=1}^{P} \big(G(x^n, x^n) - G(x^{n+1}, x^{n+1})\big) \\ &= G(x^1, x^1) - G(x^{P+1}, x^{P+1}) \leq C, \end{aligned}$$

which implies that:

$$\sum_{n=1}^{P} \big(G(x^{n+1}, x^n) - G(x^{n+1}, x^{n+1})\big) = \sum_{n=1}^{P} \big(||x^n - x^{n+1}||_2^2 - ||A(x^n - x^{n+1})||_2^2\big) \leq C.$$

Since $||A(x^n - x^{n+1})||_2^2 \leq ||A||_2^2||x^n - x^{n+1}||_2^2$ and $||A||_2 < 1$:

$$\begin{aligned} ||x^n - x^{n+1}||_2^2 - ||A(x^n - x^{n+1})||_2^2 &\geq ||x^n - x^{n+1}||_2^2 - ||A||_2^2||x^n - x^{n+1}||^2 \\ &= (1 - ||A||_2^2)||x^n - x^{n+1}||^2. \end{aligned}$$

Consequently, we have:

$$\begin{aligned}
&\sum_{n=1}^{P}(1-||A||_2^2)||x^n-x^{n+1}||^2 \leq \sum_{n=1}^{P}\left(||x^n-x^{n+1}||_2^2-||A(x^n-x^{n+1})||_2^2\right) \leq C\\
\implies\quad &\sum_{n=1}^{P}||x^n-x^{n+1}||_2^2 \leq \frac{C}{(1-||A||_2^2)} = C_2\\
\implies\quad &\sum_{n=1}^{\infty}||x^n-x^{n+1}||_2^2 < \infty\\
\implies\quad &||x^n-x^{n+1}||_2 \to 0.
\end{aligned}$$

For the proof that (x^n) is bounded, consider:

$$||x^n||_1 \leq G(x^n,x^n) = ||Ax^n-b||_2^2+2\tau||x^n||_1 \leq G(x^{n-1},x^{n-1}) \leq G(x^0,x^0) = C_3.$$

The boundedness of the ℓ_1-norm implies $||x^n||_2 \leq C_3$. □

The boundedness of the iterates implies the existence of a converging subsequence (x^{n_k}) with:

$$x^{n_k} \to \bar{x} \quad \text{and} \quad x^{n_k+1} \to \bar{x},$$

where the second follows by $||x^n - x^{n+1}||_2 \to 0$. Thus it follows that:

$$\lim_{k\to\infty} x^{n_k+1} = \lim_{k\to\infty} \mathbb{S}_\tau(x^{n_k}+A^Tb-A^TAx^{n_k}) \implies \bar{x} = \mathbb{S}_\tau(\bar{x}+A^Tb-A^TA\bar{x}).$$

By the previous lemma it follows that $\bar{x}$ minimizes the ℓ_1 penalty functional and at least one subsequence converging to $\bar{x}$ exists. In fact, we can show that the limit point of every converging subsequence is an ℓ_1 minimizer:

Lemma 2.4.7. *Every convergent subsequence of (x^n) converges to a local minimizer of $F(x)$.*

Proof. Take any convergent subsequence x^{m_j} which converges to $\hat{x}$. Let $n_{l_r} \to \bar{x}$. $\exists j_r$ such that $m_{j_r} > n_{l_r}$ by definition of a subsequence. Since $G(x^{n+1},x^{n+1}) \leq G(x^n,x^n)$

it follows that:

$$F(x^{m_{j_r}}) = G(x^{m_{j_r}}, x^{m_{j_r}}) \leq G(x^{n_{l_r}}, x^{n_{l_r}}) = F(x^{n_{l_r}}) \to F(\bar{x})$$

Taking the limit of the left side as $r \to \infty$ we have:

$$F(\hat{x}) \leq F(\bar{x})$$

so $\hat{x}$ is a minimizer. □

Having discussed the ISTA scheme using the majorization minimization approach, we not look more at majorization minimization to see how it can be used to derive schemes for related algorithms. First, let us see how it applies to ℓ_2 regularization. Going back to the functional $F_2(x) = ||Ax - b||_2^2 + 2\lambda||x||_2^2$ we have the majorizer given by:

$$\begin{aligned} G_2(x, x^n) &= ||Ax - b||_2^2 + 2\lambda||x||_2^2 + ||x - x^n||_2^2 - ||A(x - x^n)||_2^2 \\ &= ||x||_2^2 - 2\langle x, x^n + A^T b - A^T A x^n \rangle + 2\lambda||x||_2^2 + K_2. \end{aligned}$$

Setting $x^{n+1} = \arg\min_x G_2(x, x^n)$ we arrive at:

$$\nabla_x G_2(x, x^n) = 2x - 2x^n - 2A^T b + 2A^T A x^n + 4\lambda x = 0,$$

and the scheme:

$$x^{n+1} = \frac{1}{1 + 2\lambda}(x^n + A^T b - A^T A x^n).$$

We see that the thresholding has been replaced by the damping term $\frac{1}{1+2\lambda}$.

Now we look at what happens when both the ℓ_1 and ℓ_2 penalties are included:

$$\begin{aligned} G(x, x^n) &= ||Ax - b||_2^2 + ||x - x^n||_2^2 - ||A(x - x^n)||_2^2 + 2\tau||x||_1 + 2\lambda||x||_2^2 \\ &= ||x - (x^n + A^T b - A^T A x^n)||_2^2 + 2\tau||x||_1 + 2\lambda||x||_2^2 + K_3. \end{aligned}$$

Now we look at the one dimensional version of this functional:

$$f(x) = (x - b)^2 + 2\tau|x| + 2\lambda x^2,$$

and we want to minimize this with respect to x. Consider first the case $b > 0$. Then since the last two terms, $|x|$ and x^2 are both positive for any sign of x, we should also have $x > 0$ so that $(x - b)^2$ is small. We then have:

$$\begin{aligned} \bar{x} = \arg\min_x \left\{2\tau x + 2\lambda x^2 + (x - b)^2\right\} &\implies 2\tau + 4\lambda\bar{x} + 2(\bar{x} - b) = 0 \\ &\implies (1 + 2\lambda)\bar{x} = b - \tau \\ &\implies \bar{x} = \frac{b - \tau}{1 + 2\lambda} \text{ for } b > \tau. \end{aligned}$$

Similarly for $b < 0$ we get that $x < 0$ and we get:

$$\begin{aligned} \bar{x} = \arg\min_x \left\{-2\tau\bar{x} + 2\lambda x^2 + (x - b)^2\right\} &\implies -2\tau + 4\lambda\bar{x} + 2(\bar{x} - b) = 0 \\ &\implies \bar{x} = \frac{b + \tau}{1 + 2\lambda} \text{ for } b < -\tau. \end{aligned}$$

Finally if, $|b| \leq \tau$ we have that $2\tau|x| \geq 2|b||x| = 2|bx|$ and so:

$$f(x) \geq (x - b)^2 + 2|bx| + 2\lambda x^2 = x^2 + 2(|bx| - bx) + 2\lambda x^2 + b^2 \geq b^2 = f(0).$$

so that zero is a minimizer for this case. Hence, we have the following solution:

$$\arg\min_x (x-b)^2 + 2\tau|x| + 2\lambda x^2 = \begin{cases} \dfrac{b-\tau}{1+2\lambda} & \text{if } b > \tau, \\ 0 & \text{if } |b| \leq \tau, \\ \dfrac{b+\tau}{1+2\lambda} & \text{if } b < -\tau. \end{cases}$$

We thus have that by the properties of soft thresholding:

$$\arg\min_x (x-b)^2 + 2\tau|x| + \lambda x^2 = \frac{1}{1+\lambda}\mathbb{S}_\tau(b),$$

which motivates the following iteration for the minimization of $||Ax-b||_2^2 + 2\tau||x||_1 + \lambda||x||_2^2$:

$$x^{n+1} = \frac{1}{1+\lambda}\mathbb{S}_\tau\left(x^n + A^T b - A^T A x^n\right).$$

Going back now to ℓ_1 minimization, we now discuss some alternatives to the ISTA scheme. The main problem with (ISTA) is that it is known to have a slow rate of convergence. The paper [2] gives the following result:

Theorem 2.4.8 ([2], Theorem 3.1)**.** *For any $x^0 \in \mathbb{R}^N$, let x^n be the iterates generated by* (ISTA)*. If $||A||_2 < 1$, then for all n we have the following estimate:*

$$F(x^n) - F(\bar{x}) \leq \frac{||x^0 - \bar{x}||^2}{n},$$

where $\bar{x}$ is any minimizer of (2.3.6)*.*

The proof follows from the paper, where we have used in their notation $f(x) = ||Ax-b||^2$ so that we can take for instance the Lipschitz constant to be $L(f) = 2$ when $||A||_2 < 1$ since:

$$||\nabla f(x) - \nabla f(y)|| = ||2A^T Ax - 2A^T Ay|| \leq 2||A^T A||_2||x-y||_2 < 2||x-y||_2.$$

One improvement on this scheme is to use instead of the soft-thresholding operator a projection onto the ℓ_1 ball [13]. Replacing the above iteration by:

$$x^{n+1} = P_R\left(x^n + A^T b - A^T A x^n\right),$$

where the projection onto the ℓ_1 ball P_R can be expressed in terms of the soft thresholding function. The algorithm has better numerical properties but still a slow rate of convergence.

In practice, when using soft thresholding for ℓ_1 minimization one uses the FISTA algorithm, which has a much faster rate of convergence. The algorithm is designed to minimize the function $f(x) + g(x)$, where f is a continuously differentiable convex function with Lipschitz continuous gradient (i.e., $||\nabla f(x) - \nabla f(y)||_2 \leq L||x - y||_2$ for some constant L), and g is a continuous convex function, but possibly non-smooth, as in the case of $2\tau||x||_1$. The FISTA algorithm uses the function (proximal mapping):

$$p_L(y) = \arg\min_x \left\{ g(x) + \frac{L}{2}\left\| x - (y - \frac{1}{L}\nabla f(y)) \right\|^2 \right\}$$

to define the following algorithm:

$$\begin{aligned}
y^1 &= x^0 \in \mathbb{R}^N \quad , \quad t_1 = 1 \quad \text{and iterate:} \\
x^n &= p_L(y^n) = \arg\min_x \left\{ g(x) + \frac{L}{2}\left\| x - (y - \frac{1}{L}\nabla f(y)) \right\|^2 \right\} \\
t_{n+1} &= \frac{1 + \sqrt{1 + 4t_n^2}}{2} \\
y^{n+1} &= x^n + \frac{t_n - 1}{t_{n+1}}(x^n - x^{n-1}).
\end{aligned} \tag{FISTA}$$

Let us now work out the above for $F(x) = ||Ax - b||_2^2$ and $G(x) = 2\tau||x||_1$. We have:

$$||\nabla f(x) - \nabla f(y)||_2 = ||2A^T Ax - 2A^T Ay||_2 = ||2A^T A(x - y)||_2 \leq 2||A^T A||_2||x - y||_2,$$

so that the Lipschitz constant is $L = 2||A^TA||_2$. If we scale A by $\frac{1}{\alpha}$ where $\alpha = \sigma_{\max}(A)$ then $L = 2(1)^2 = 2$. Then we have that:

$$\begin{aligned}
& \arg\min_x \left\{ g(x) + \frac{L}{2}\left\| x - (y - \frac{1}{L}\nabla f(y)) \right\|^2 \right\} \\
= \; & \arg\min_x \left\{ 2\tau||x||_1 + \frac{2}{2}\left\| x - (y - \frac{1}{2}2A^T(Ay-b)) \right\|^2 \right\} \\
= \; & \arg\min_x \left\{ 2\tau||x||_1 + \left\| x - (y - A^T(Ay-b)) \right\|^2 \right\} = \mathbb{S}_\tau(y - A^T(Ay-b)),
\end{aligned}$$

so the each step of FISTA above is just the same soft thresholding as before but now applied to the vector y which is a linear combination of the previous two iterates. The main result concerning this algorithm is that the convergence rate is now much faster:

Theorem 2.4.9 ([2], Theorem 4.4)**.** *For any $x^0 \in \mathbb{R}^N$, let x^n be the iterates generated by* (FISTA)*. If $||A||_2 < 1$, then for all n we have the following estimate:*

$$F(x^n) - F(\bar{x}) \leq \frac{2L||x^0 - \bar{x}||^2}{(n+1)^2},$$

where $\bar{x}$ is any minimizer of (2.3.6)*.*

The ISTA and FISTA methods are amongst the most popular for ℓ_1 minimization and are widely used. In my experience, FISTA is one of the fastest general purpose methods across different applications. The choice of soft thresholding, however, also has an evident shortcoming in that it shrinks the large components of the input vector. This motivates the search for an alternative algorithm using a different thresholding function, which towards the end of the iteration, keeps the large coefficients at constant size. We describe more on this later in this chapter and a method based on a modified thresholding function (IVTA) is the basis of the following chapter.

2.5 Algorithms for ℓ_1 minimization: Dual Methods

We proceed now to describe methods that do not use thresholding. In particular, we would like to mention an approach based on the dual of the ℓ_1-norm, as mentioned in [47]. We first prove that the dual of the ℓ_1-norm is given by the ℓ_∞-norm:

Lemma 2.5.1. *Consider the ℓ_1-norm defined by:*

$$\|x\|_1 := \sum_{i=1}^{N} |x_i| \quad \forall\, x \in \mathbb{R}^N.$$

Then the dual of $\|\cdot\|_1$ is given by:

$$\|y\|_\infty := \max_i \{|y_i|\}.$$

Proof. Fix any $y \in \mathbb{R}^N$, and consider

$$\alpha := \max_{x \in \mathbb{R}^N} \frac{\langle x, y\rangle}{\|x\|_1}.$$

For any $x \in \mathbb{R}^N$,

$$\langle x, y\rangle \leq \sum_{i=1}^{N} |x_i|\,|y_i| \leq \max_i \{|y_i|\}\, \|x\|_1,$$

so $\alpha \leq \max_i \{|y_i|\}$. To show equality, consider an example where k is such that $|y_k| = \max_i \{|y_i|\}$. Define

$$\bar{x}_i = \begin{cases} 0 & \text{if } i \neq k, \\ \operatorname{sgn}(y_k) & \text{if } i = k. \end{cases}$$

Then $\|\bar{x}\|_1 = 1$ and

$$\langle \bar{x}, y\rangle = \operatorname{sgn}(y_k) y_k = |y_k| = \max_i \{|y_i|\}.$$

Thus, we have:

$$\alpha = \frac{\langle \bar{x}, y \rangle}{||\bar{x}||_1} = \max_i \{|y_i|\}\,.$$

Therefore the dual norm of $\| \cdot \|_1$ is given by

$$||y||_\infty := \max_i \{|y_i|\}\,.$$

□

The idea of dual methods is to solve the dual problem and use that solution to obtain the solution to the original problem. We now briefly describe the dual approach as summarized in [6]: Given a constrained problem:

$$\min_x f(x) \quad \text{s.t.} \quad Ax = b,$$

the Lagrangian is defined as:

$$L(x, y) = f(x) + y^T(b - Ax).$$

The dual function is:

$$g(y) = \min_x L(x, y)$$

and the dual problem becomes $\max_y g(y)$. Once we solve the dual problem for $\bar{y} = \arg\max_y g(y)$ we can use as the solution to the original problem, the solution of:

$$\bar{x} = \arg\min_x L(x, \bar{y})$$

We now describe the dual augmented Lagrangian approach for the constrained problem for ℓ_1 minimization:

$$\min_x \ \|x\|_1 \quad \text{s.t.} \quad Ax = b, \tag{2.5.1}$$

where $A \in \mathbb{R}^{m\times N}$ and $b \in \mathbb{R}^m$. We want to compute the dual of this problem. The Lagrangian of (5.4.1) is given by

$$L(x,y) = \|x\|_1 + y^T(b - Ax).$$

We need to compute $\min_x L(x,y)$ for each fixed y. Since the function $L(\cdot, y)$ is separable, we have

$$\begin{aligned}
\min_x L(x,y) &= b^T y + \min_x \sum_{i=1}^N |x_i| - (A^T y)_i x_i \\
&= b^T y + \sum_{i=1}^N \min_{x_i} \left\{ |x_i| - (A^T y)_i x_i \right\} \\
&= \begin{cases} b^T y & \text{if } \left|(A^T y)_i\right| \leq 1 \;\forall\, i = 1, \ldots, N \\ -\infty & \text{otherwise,} \end{cases}
\end{aligned}$$

where we have used the fact that for any $\beta \in \mathbb{R}$,

$$\min_{t\in\mathbb{R}} \; |t| - \beta t = \begin{cases} 0 & \text{if } |\beta| \leq 1 \\ -\infty & \text{otherwise.} \end{cases} \tag{2.5.2}$$

To see this, consider for example $t > 0$. Then $|t| - \beta t = t(1 - \beta)$. For $\beta \leq 1$, the minimum value is 0 at $\beta = 1$. For $\beta > 1$, $t(1 - \beta) \to -\infty$. The dual of (5.4.1) is obtained by taking the maximum of $\min_x L(x,y)$ with respect to y:

$$\max_y \; b^T y \quad \text{s.t.} \quad \left|(A^T y)_i\right| \leq 1,$$

or equivalently,

$$\max_y \; b^T y \quad \text{s.t.} \quad \|A^T y\|_\infty \leq 1. \tag{D}$$

We can introduce a new variable z and change the sign of b to obtain the equivalent problem:

$$\min_{y} \ -b^T y \quad \text{s.t.} \quad z = A^T y, \ \|z\|_\infty \le 1. \tag{2.5.3}$$

Then we apply the idea of the augmented Lagrangian by removing the constraint $z = A^T y$ and making it a penalty term, so we would be looking at the optimization problem

$$\min_{x,y,z} \ L_\mu(y,z,x) := -b^T y - x^T(z - A^T y) + \frac{\mu}{2}\|z - A^T y\|_2^2 \quad \text{s.t.} \quad \|z\|_\infty \le 1. \tag{2.5.4}$$

Differentiating L with respect to different variables we get:

$$\begin{aligned}
\nabla_x L_\mu(y,z,x) &= A^T y - z \\
\nabla_y L_\mu(y,z,x) &= -b + Ax + \mu A(A^T y - z) \\
\nabla_z L_\mu(y,z,x) &= -x + \mu(z - A^T y).
\end{aligned}$$

Now, as in [46] we perform the minimization $\min_{x,y,z} \ L_\mu(y,z,x)$ by keeping two of the three variables constant and updating one variable of the x, y, z at a time. We then have: Now observe that when $x = x^n$ and $y = y^n$ are fixed,

$$\begin{aligned}
& \min_z \{L_\mu(y^n, z, x^n) : \|z\|_\infty \le 1\} \\
= \ & -b^T y^n + (x^n)^T A^T y^n + \min_z \left\{ \frac{\mu}{2}\|z - A^T y^n\|_2^2 - (x^n)^T z : \|z\|_\infty \le 1 \right\} \\
= \ & -b^T y^n + (x^n)^T A^T y^n + \sum_{k=1}^{N} \min_{z_k} \left\{ \frac{\mu}{2}(z_k - (A^T y^n)_k)^2 - x_k^n z_k : |z_k| \le 1 \right\}.
\end{aligned}$$

Hence, we have that:

$$\begin{aligned}
&\arg\min_z \{L_\mu(y^n, z, x^n) : \|z\|_\infty \leq 1\} \\
= \;& \left\{\bar{u} : \forall\, k,\; \bar{u}_k = \mathbb{P}_{[-1,1]}\left(\arg\min_{z_k}\left\{\frac{\mu}{2}(z_k - (A^T y^n)_k)^2 - x_k^n z_k\right\}\right)\right\} \\
= \;& \left\{\bar{u} : \forall\, k,\; \bar{u}_k = \mathbb{P}_{[-1,1]}\left(\frac{x_k^n}{\mu} + (A^T y^n)_k\right)\right\},
\end{aligned}$$

where $\mathbb{P}_S$ is the component-wise projection onto set S. Thus, the update for z becomes:

$$z^{n+1} = \mathbb{P}_{[-1,1]}\left(\frac{z^n}{\mu} + (A^T y^n)\right).$$

Once this update has been made, we can proceed to update y. Differentiating $-b^T y - x^T(z - A^T y) + \frac{\mu}{2}\|z - A^T y\|_2^2$ with respect to y and setting to zero we have:

$$-b + Ax + \mu A(A^T y - z) = 0.$$

Solving for y and plugging in z^{n+1} and x^n we get that:

$$\mu A A^T y^{n+1} = \mu A z^{n+1} - (Ax^n - b) \implies (AA^T) y^{n+1} = Az^{n+1} - \frac{1}{\mu}(Ax^n - b).$$

The above update can be approximated simply with one iteration of the conjugate gradient algorithm. For the update in the x direction we use the gradient $\nabla_x L_\mu(x, y, z) = A^T y - z$ to write the iteration:

$$x^{n+1} = x^n - \mu(z^{n+1} - A^T y^{n+1}).$$

Thus, a dual augmented Lagrangian method may be implemented as:

$$\begin{aligned} z^{n+1} &= \mathbb{P}_{[-1,1]}\left(\frac{z^n}{\mu} + (A^T y^n)\right) \\ (AA^T)y^{n+1} &= \left(Az^{n+1} - \frac{1}{\mu}(Ax^n - b)\right) \\ x^{n+1} &= x^n - \mu(z^{n+1} - A^T y^{n+1}). \end{aligned}$$

The major computation at each iteration consists of updating the variable y. We find that the single conjugate gradient iteration approach as mentioned in [46] gives good numerical performance. The background to this method (called DALM for Dual Augmented Lagrangian Method) and FISTA are stated here for completeness and because of their good numerical performance, we compare all the newly developed algorithms to these methods.

2.6 Algorithms for ℓ_1 minimization: Multiplier Methods

Having discussed in the above subsection, the dual augmented Lagrangian method, we present here a short discussion on the method of the Alternating Direction Method of Multipliers, based on the reference in [6]. We will present the details here, since we will revisit this method later in the application section.

The idea of ADMM is to solve the following problem:

$$\min_{x,y} f(x) + g(z) \quad \text{s.t.} \quad Ax + Bz = c.$$

The method summarized below, is useful in practice, largely because it converges with small assumptions on f and g and a lot of problems can be posed in the above form. The most important assumptions for convergence are that f and g should be convex. The augmented Lagrangian for the corresponding system is:

$$L_\mu(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + \frac{\mu}{2}||Ax + Bz - c||_2^2.$$

The ADMM scheme proceeds by carrying out the minimization of $L_\mu(x, z, y)$ with respect to x and z and then the dual update step:

$$\begin{aligned} x^{n+1} &= \arg\min_x L_\mu(x, z^n, y^n) \\ z^{n+1} &= \arg\min_z L_\mu(x^{n+1}, z, y^n) \\ y^{n+1} &= y^n + \mu(Ax^{n+1} + Bz^{n+1} - c). \end{aligned}$$

Let us now apply the ADMM algorithm to the minimization of the unconstrained ℓ_1 functional:

$$\min\left\{\frac{1}{2}||Mx - b||_2^2 + \tau||x||_1\right\} = \min\left\{\frac{1}{2}||Mx - b||_2^2 + \tau||z||_1 : z - x = 0\right\}.$$

In reference to the above, we thus have:

$$f(x) = \frac{1}{2}||Mx - b||_2^2 \ , \ g(z) = \tau||z||_1 \ , \ A = -I \ , \ B = I \ , \ c = 0.$$

Applying the above, we have:

$$L_\mu(x, z, y) = \frac{1}{2}||Mx - b||_2^2 + \tau||z||_1 + y^T(z - x) + \frac{\mu}{2}||z - x||_2^2$$

Let us find the update scheme for x^n by carry out the minimization with respect to x:

$$\begin{aligned} & \frac{\partial}{\partial x}\left(\frac{1}{2}||Mx-b||_2^2 - y^T x + \frac{\mu}{2}||z-x||_2^2\right) = M^T(Mx-b) - y - \mu(z-x) = 0 \\ \implies \quad & M^T M x + \mu x - M^T b - y - \mu z = 0 \\ \implies \quad & (M^T M + \mu)x = M^T b + y + \mu z \\ \implies \quad & x = (M^T M + \mu I)^{-1}\left(M^T b + y + \mu z\right). \end{aligned}$$

If we assume that y, the vector of Lagrange multipliers, is negative, we have:

$$x^{n+1} = (M^T M + \mu I)^{-1}\left(M^T b - y^n + \mu z^n\right).$$

We now derive the update scheme for z^n.

$$\begin{aligned} \arg\min_z \left\{\tau||z||_1 + y^T z + \frac{\mu}{2}||z-x||_2^2\right\} &= \arg\min_z \left\{2\frac{\tau}{\mu}||z||_1 + \frac{2}{\mu}y^T z + ||z-x||_2^2\right\} \\ &= \arg\min_z \left\{2\frac{\tau}{\mu}||z||_1 + ||z-(x-\frac{1}{\mu}y)||_2^2\right\} \\ &= \mathbb{S}_{\frac{\tau}{\mu}}\left(x - \frac{1}{\mu}y\right). \end{aligned}$$

Assuming y is negative, we get the scheme:

$$z^{n+1} = \mathbb{S}_{\frac{\tau}{\mu}}\left(x^{n+1} + \frac{1}{\mu}y^n\right).$$

Finally, substituting in $A = -I$, $B = I$, and $c = 0$ we get the following update scheme for y:

$$y^{n+1} = y^n + \mu(x^{n+1} - z^{n+1}).$$

The parameter μ can be set initially to a fixed value and increased by a constant factor at each iteration. The disadvantage of the above scheme for the ℓ_1 problem is that it calls for a linear solve at each iteration, although that can be approximated, for example, by a single step of the CG algorithm at each iteration.

2.7 Algorithms for ℓ_1 minimization: Coordinate Descent Method

We now present a third approach called coordinate descent (see for example [45]) which optimizes over individual entries of the functional, one at a time, while keeping the other entries fixed. That is, given some initial guess x^0, at each iteration, the algorithm randomly selects an entry in the current solution vector and keeping all the other entries fixed changes this entry so as to decrease the value of the ℓ_1 functional. This randomness of the sweep pattern is essential to the speed of convergence, although of course, if the dimension of x is large, this method will tend to be slow. In practice, however, we find that the method works when the choice of entries is restricted to the identified (or estimated) support of the sparse signal. We discuss some ideas for support identification in the numerical experiments chapter, but in general we find that this is not possible to do with ill conditioned matrices.

As mentioned above idea of the coordinate descent method is that given $F(x) = F(x_1, \dots, x_N)$ we want to update one coordinate at a time while fixing all the others at each step. Choose a coordinate x_j. We would like to derive a formula to update this coordinate so that the ℓ_1 function evaluated at the new vector with this coordinate changed has a smaller than or equal value than before. We have:

$$\bar{x}_j = \arg\min_u F(x_1, \dots, x_{j-1}, u, x_{j+1}, \dots, x_N),$$

and the solution at the new iterate $n+1$ becomes: $(x_1^n, \dots, \bar{x}_j, \dots, x_N^n)$. To derive the formula for updating each component, we must expand out the terms in $F(x)$ and separate the terms involving x_j from the rest. Consider then:

$$||Ax - b||_2^2 = ||Ax||_2^2 - 2\langle Ax, b\rangle + ||b||_2^2 = \langle A^T Ax, x\rangle - 2\langle A^T b, x\rangle + ||b||_2^2.$$

Only the first two terms have dependence on x_j. Consider the first term.

$$\begin{aligned}
& \langle A^T Ax, x\rangle \\
= \; & \sum_{i=1}^{N} \left(\sum_{k=1}^{N} (A^T A)_{i,k} x_k \right) x_i = \sum_{i=1}^{N} \sum_{k\neq j} (A^T A)_{i,k} x_k x_i + \sum_{i=1}^{N} (A^T A)_{i,j} x_j x_i \\
= \; & \sum_{i\neq j} \sum_{k\neq j} (A^T A)_{i,k} x_k x_i + \sum_{k\neq j} (A^T A)_{j,k} x_k x_j + \sum_{i\neq j} (A^T A)_{i,j} x_j x_i + (A^T A)_{j,j} (x_j)^2 \\
= \; & \sum_{i\neq j} \sum_{k\neq j} (A^T A)_{i,k} x_k x_i + 2 \sum_{k\neq j} (A^T A)_{j,k} x_k x_j + (A^T A)_{j,j} (x_j)^2.
\end{aligned}$$

Next:

$$2\langle A^T b, x\rangle = 2 \sum_{k\neq j} (A^T b)_k x_k + 2(A^T b)_j x_j \quad \text{and} \quad ||x||_1 = |x_j| + \sum_{k\neq j} |x_k|.$$

Now taking only terms that have x_j into account the minimization problem reduces to:

$$\begin{aligned}
& \arg\min_{x_j} \left\{ ||Ax - b||_2^2 + 2\tau ||x||_1 \right\} \\
= \; & \arg\min_{x_j} \left\{ (A^T A)_{j,j} (x_j)^2 - 2(A^T b)_j x_j + 2 \sum_{k\neq j} (A^T A)_{j,k} x_k x_j + 2\tau |x_j| \right\} \\
= \; & \arg\min_{x_j} \left\{ ||A_j||_2^2 (x_j)^2 - 2(A^T b)_j x_j + 2 \sum_{k\neq j} (A^T A)_{j,k} x_k x_j + 2\tau |x_j| \right\},
\end{aligned}$$

where $A_j = Ae_j$, the j-th column of the matrix A. Now we have that $(A^TA)_{j,k} = \sum_{l=1}^M A^T_{j,l}A_{l,k}$ and:

$$\sum_{k\neq j}(A^TA)_{j,k}x_kx_j = \sum_{k\neq j}\sum_{l=1}^M A^T_{j,l}A_{l,k}x_kx_j = \sum_{l=1}^M\sum_{k\neq j} A^T_{j,l}A_{l,k}x_kx_j = \sum_{l=1}^M\sum_{k\neq j} A_{l,j}A_{l,k}x_kx_j,$$

so that the terms in the minimization above add to:

$$\begin{aligned} & 2\sum_{k\neq j}(A^TA)_{j,k}x_kx_j - 2(A^Tb)_jx_j = 2\sum_{l=1}^M\sum_{k\neq j} A_{l,k}A_{l,j}x_kx_j - 2\sum_{k=1}^M A_{k,j}b_kx_j \\ = & \quad 2\sum_{l=1}^M\sum_{k\neq j} A_{l,k}A_{l,j}x_kx_j - 2\sum_{l=1}^M A_{l,j}b_lx_j \\ = & \quad 2\sum_{l=1}^M A_{l,j}\left(\sum_{k\neq j}A_{l,k}x_k - b_l\right)x_j. \end{aligned}$$

Thus, we have that:

$$\arg\min_{x_j}\left\{||Ax-b||_2^2 + 2\tau||x||_1\right\} = \arg\min_{x_j}\left\{||A_j||_2^2(x_j)^2 - 2\beta_jx_j + 2\tau|x_j|\right\},$$

with $\beta_j = \sum_{l=1}^M A_{l,j}\left(b_l - \sum_{k\neq j}A_{l,k}x_k\right)$ and $A_j = Ae_j$. Set $x_j = u$, we then minimize:

$$\begin{aligned} \arg\min_u\left\{||A_j||^2u^2 - 2\beta_ju + 2\tau|u|\right\} &= \arg\min_u\left\{||A_j||^2(u - \frac{\beta_j}{||A_j||^2})^2 + 2\tau|u|\right\} \\ &= \arg\min_u\left\{(u - \frac{\beta_j}{||A_j||^2})^2 + 2\frac{\tau}{||A_j||^2}|u|\right\}. \end{aligned}$$

Since $S_\tau(b) = \arg\min_u\left\{(u-b)^2 + 2\tau|u|\right\}$ we have:

$$\arg\min_u\left\{(u - \frac{\beta_j}{||A_j||^2})^2 + 2\frac{\tau}{||A_j||^2}|u|\right\} = S_{\frac{\tau}{||A_j||^2}}\left(\frac{\beta_j}{||A_j||^2}\right) = \frac{1}{||A_j||^2}S_\tau(\beta_j),$$

using the previously defined soft thresholding operator. Thus, at each iteration, we update one coordinate x_j using the formula:

$$\bar{x}_j = \frac{1}{||A_j||^2} S_\tau(\beta_j) \quad \text{with} \quad \beta_j = \sum_{l=1}^{m} A_{l,j} \left(b_l - \sum_{k \neq j} A_{l,k} x_k \right).$$

We see from above that the implementation is given in simple form in terms of the soft thresholding operator and we discuss the computational details further in the numerical section of the thesis.

2.8 New Approaches for ℓ_1 minimization: Modified Thresholding and Reweighted Least Squares

We now describe some ideas for new sparse regularization algorithms, which are then analyzed in more detail in subsequent chapters. First, we go back to look at the Soft-Thresholding operator. We observe from the figure below that a property of this operation is that it shrinks the large coefficients of the vector, something we may not want to do at later stages of the iteration, when the larger components have already been determined. This property of soft thresholding can be clearly seen in the plot below:

In Chapter 3, we propose a new set of algorithms (IVTA/FIVTA) that use a different thresholding function, which does not have this property. The algorithms are shown to have some numerical advantages over the soft thresholding schemes, most notably in terms of the speed of convergence.

We now describe another approach to the problem, which consists of replacing the non-smooth part of the ℓ_1 functional by a smooth approximation, i.e. replacing

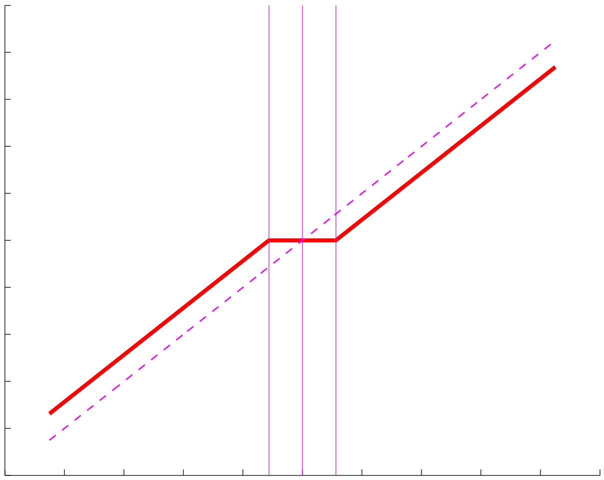


Figure 2.2: Soft thresholding function. Lines (solid) at ±τ and (dashed) identity function. We see that for larger values the soft thresholding function stays away from the identity, thus penalizing the large components.

$||Ax-b||_2^2+2\tau||x||_1$ by $||Ax-b||_2^2+2\tau\theta(x)$ where $\theta(x)$ is some smooth approximation to the ℓ_1-norm $||x||_1$. The advantage of this approach is that the new functional, which is an approximation to the original, is now entirely smooth and can be differentiated, so a method like steepest descent can be used to work with the gradient of the new functional. In addition, since the new functional is convex, all that remains is to set its derivative to zero and solve the corresponding linear equations, which may be done by a number of different methods. In particular the popular conjugate gradient method may be used.

The non-smooth part of the ℓ_1 functional is the ℓ_1-norm $||x||_1 = \sum\limits_{k=1}^{N} |x_k|$. We now describe a simple method to construct a smooth approximation to $||x||_1$. This can be accomplished by approximating the absolute value function $|x|$ by convolving it with a bump function parametrized by its "width". As this width tends to zero, the smoothed result will approach the original function. As an example, consider the Gaussian Function defined as $f(x) = \frac{1}{2\pi\sigma^2}e^{\frac{-x^2}{2\sigma^2}}$ with $g(x) = |x|$. The convolution will be a smooth function, but as $\sigma \to 0$, the result will approximate the non-smooth $|x|$. The routine, but tedious calculation is shown below:

Lemma 2.8.1. *Let:*

$$f(t) = \frac{1}{2\pi\sigma^2} e^{\frac{-t^2}{2\sigma^2}} \quad , \quad g(t) = |t| \quad , \quad H(T,t) = \int_{-T}^{T} f(s)g(s-t)ds.$$

We have that:

$$(f * g)(t) = \lim_{T \to \infty} H(T,t) = t\,\mathrm{erf}\left(\frac{t}{\sqrt{2}\sigma}\right) + \sqrt{\frac{2}{\pi}}\sigma e^{\frac{-t^2}{2\sigma^2}},$$

where the error function is defined as: $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-u^2} du.$

Proof. Fix $\sigma > 0$. For $t \in \mathbb{R}$, let us define:

$$f(t) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{t^2}{2\sigma^2}\right) \quad \text{and} \quad g(t) = |t|;$$

then the convolution is given by $(f * g)(t) = \lim_{T \to \infty} H(T,t)$ where:

$$\sqrt{2\pi\sigma^2} H(T,t) \quad := \quad \sqrt{2\pi\sigma^2} \int_{-T}^{T} f(s)g(t-s)ds = \int_{-T}^{T} \exp\left(-\frac{s^2}{2\sigma^2}\right)|t-s|ds.$$

Expanding the above we have that:

$$\begin{aligned}
& \int_{-T}^{T} \exp\left(-\frac{s^2}{2\sigma^2}\right)|t-s|ds \\
= \quad & \int_{-T}^{t} \exp\left(-\frac{s^2}{2\sigma^2}\right)(t-s)ds + \int_{t}^{T} \exp\left(-\frac{s^2}{2\sigma^2}\right)(s-t)ds \\
= \quad & t\left(\int_{-T}^{t} \exp\left(-\frac{s^2}{2\sigma^2}\right)ds - \int_{t}^{T} \exp\left(-\frac{s^2}{2\sigma^2}\right)ds\right) \\
& \qquad + \int_{-T}^{t} \exp\left(-\frac{s^2}{2\sigma^2}\right)(-s)ds + \int_{t}^{T} \exp\left(-\frac{s^2}{2\sigma^2}\right)sds \\
= \quad & \sqrt{2}\sigma t\left(\int_{-T/\sqrt{2}\sigma}^{t/\sqrt{2}\sigma} \exp\left(-u^2\right)du - \int_{t/\sqrt{2}\sigma}^{T/\sqrt{2}\sigma} \exp\left(-u^2\right)du\right) \\
& \qquad + \sigma^2\left(\int_{-T}^{t} \exp\left(-\frac{s^2}{2\sigma^2}\right)\left(-\frac{s}{\sigma^2}\right)ds - \int_{t}^{T} \exp\left(-\frac{s^2}{2\sigma^2}\right)\left(-\frac{s}{\sigma^2}\right)ds\right).
\end{aligned}$$

Next, making use of the definition of the error function and of the fundamental theorem of calculus we have:

$$\begin{aligned}\sqrt{2\pi\sigma^2}H(T,t) &= \sqrt{\frac{\pi}{2}}\sigma t\left(\operatorname{erf}\left(\frac{t}{\sqrt{2}\sigma}\right)-\operatorname{erf}\left(\frac{-T}{\sqrt{2}\sigma}\right)-\operatorname{erf}\left(\frac{T}{\sqrt{2}\sigma}\right)+\operatorname{erf}\left(\frac{t}{\sqrt{2}\sigma}\right)\right)\\ &\quad +\sigma^2\left(\int_{-T}^{t}\frac{d}{ds}\left[\exp\left(-\frac{s^2}{2\sigma^2}\right)\right]ds-\int_{t}^{T}\frac{d}{d}\left[\exp\left(-\frac{s^2}{2\sigma^2}\right)\right]ds\right)\\ &= \sqrt{2\pi}\sigma t\operatorname{erf}\left(\frac{t}{\sqrt{2}\sigma}\right)+2\sigma^2\left(\exp\left(-\frac{t^2}{2\sigma^2}\right)-\exp\left(-\frac{T^2}{2\sigma^2}\right)\right),\end{aligned}$$

so that, since $\exp\left(-\frac{T^2}{2\sigma^2}\right)\to 0$ as $T\to\infty$, we have:

$$(f*g)(t)=\lim_{T\to\infty}H(T,t)=t\operatorname{erf}\left(\frac{t}{\sqrt{2}\sigma}\right)+\sqrt{\frac{2}{\pi}}\sigma\exp\left(-\frac{t^2}{2\sigma^2}\right).$$

□

The result of the above lemma is that for small σ the following approximation holds:

$$|x_k|\approx x_k\operatorname{erf}\left(\frac{x_k}{\sqrt{2}\sigma}\right)+\sqrt{\frac{2}{\pi}}\sigma e^{\frac{-x_k^2}{2\sigma^2}}$$

We illustrate the result of the above computation in the figure below, where we can clearly see that the convolution smooths out the sharp corner of the absolute value, at the expense of being off the true value around zero:

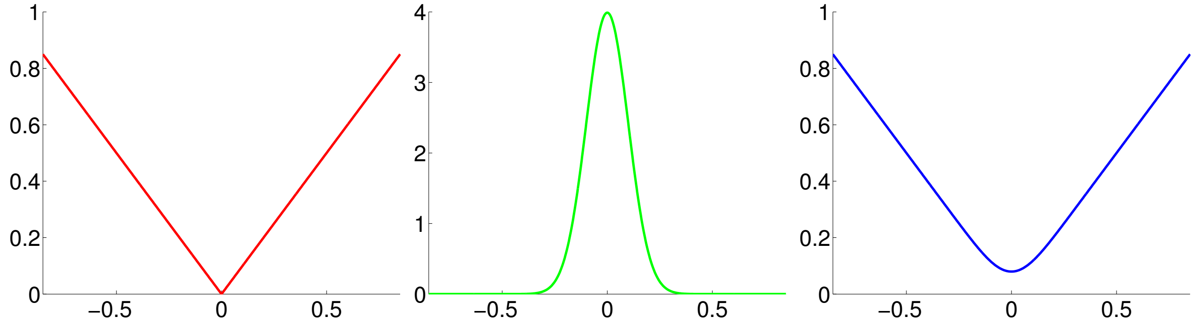


Figure 2.3: Absolute value, Gaussian with $\sigma = 0.1$ and the resulting smooth approximation from the convolution.

It follows that we can approximate the ℓ_1-norm as:

$$||x||_1 = \sum_{k=1}^{N} |x_k| \approx \sum_{k=1}^{N} \left(x_k \operatorname{erf}\left(\frac{x_k}{\sqrt{2}\sigma}\right) + \sqrt{\frac{2}{\pi}}\sigma e^{\frac{-x_k^2}{2\sigma^2}} \right).$$

This approximation is now differentiable at all values of x_k. We now use the approximation:

$$||Ax-b||_2^2 + 2\tau||x||_1 \approx ||Ax-b||_2^2 + 2\tau \sum_{k=1}^{N} \left(x_k \operatorname{erf}\left(\frac{x_k}{\sqrt{2}\sigma}\right) + \sqrt{\frac{2}{\pi}}\sigma e^{\frac{-x_k^2}{2\sigma^2}} \right),$$

and compute the gradient to the right hand side:

$$2A^T(Ax-b) + 2\tau \left(\operatorname{erf}(\frac{x_1}{\sqrt{2}\sigma}) + x_1 \frac{2}{\sqrt{\pi}} \frac{1}{\sqrt{2}\sigma} e^{-\frac{x_1^2}{2\sigma^2}} - \sqrt{\frac{2}{\pi}}\sigma \frac{2x_1}{2\sigma^2} e^{-\frac{x_1^2}{2\sigma^2}}, \ldots, \right.$$
$$\left. \operatorname{erf}(\frac{x_N}{\sqrt{2}\sigma}) + x_N \frac{2}{\sqrt{\pi}} \frac{1}{\sqrt{2}\sigma} e^{-\frac{x_N^2}{2\sigma^2}} - \sqrt{\frac{2}{\pi}}\sigma \frac{2x_N}{2\sigma^2} e^{-\frac{x_N^2}{2\sigma^2}} \right).$$

Based on this, a simple steepest descent scheme may be implemented.

Another possible approximation to the non-smooth portion is possible and is the basis of two new algorithms discussed later in the thesis. One comment on the above approximation is that the approximation is of the same quality, regardless of the number of iterations of the algorithm. That is, the smoothed approximation is always some distance away from the ℓ_1-norm $||x||_1$, no matter how many iterations of steepest descent or a similar scheme we implement. The algorithms in chapter 3 of the thesis present a different, iteratively reweighted approach, which produces an approximation that is closer to the original as the iterations progress. We consider the relation:

$$||x||_1 = \sum_{k=1}^{N} |x_k| = \sum_{k=1}^{N} \frac{x_k^2}{|x_k|} = \sum_{k=1}^{N} \frac{x_k^2}{\sqrt{x_k^2}} \quad \text{if } x_k \neq 0 \quad \forall k.$$

To adjust for the case when some $x_k = 0$ (which in case of sparse solutions will always occur), we can instead use the approximation:

$$||x||_1 = \sum_{k=1}^{N} |x_k| = \sum_{k=1}^{N} w_k^n x_k^2 = \sum_{k=1}^{N} \frac{x_k^2}{\sqrt{x_k^2 + \epsilon_n^2}}$$

where the $w_k^n = \frac{1}{\sqrt{x_k^2+\epsilon_n^2}}$ represents a weight and where the parameter $\epsilon_n \to 0$ as $n \to \infty$. As we will see in a later chapter, algorithms based on this approach are indeed possible and work well, provided the sequence of (ϵ_n)'s is chosen in a right way. These algorithms, along with their convergence analysis, are amongst the main theoretical contributions of this thesis.

Next, having discussed a few different approaches to ℓ_1 minimization, we see that similar methods are possible for the non-convex ℓ_0 functional.

2.9 Algorithms for ℓ_0 minimization

In the same way, as for the ℓ_1 functional, similar algorithms for ℓ_0 can also be used, but the corresponding numerical implementation becomes harder due to the strong nonconvexity of the functional. Special care must be taken to avoid local minima and in general algorithms for the minimization of the ℓ_0 function do not produce good results in case of ill-conditioning and noise. The first amongst the algorithms we mention are the so called Greedy algorithms. These algorithms typlically solve a very simple optimization problem in each iteration, and obtain the "best immediate or local solution" which at the same time is feasible for the original problem. This class includes the classical matching pursuits [23, 30, 35], othogonal matching pursuit [42], CoSaMP [31], and iterative support detection introduced in [44]. These algorithms tend to perform well for RIP abiding matrices, but brake down outside this regime. For this reason, we do not discuss the above methods in detail as they are not directly applicable to our problem. We now mention a few other methods for ℓ_0 minimization, since in particular, they can be used in conjunction with the ℓ_1 methods once an accurate estimate of the support has been obtained (taking care to then threshold outside this support set). We state now, two algorithms involving the hard thresholding operator, known as IHT [4, 5] and its accelerated variant AIHT [3]:

$$x^{n+1} = H_K(x^n + A^T b - A^T A x^n), \tag{2.9.1}$$

where $H_K(x)$ is a nonlinear operator that sets all but the largest K elements of x to zero. The convergence of the method to a local minimizer of

$$\min_x ||Ax - b||_2^2 \text{ s.t. } ||x||_0 \leq s \tag{2.9.2}$$

has been proved in [4]. There are some techniques for accelerating convergence. In particular, we discuss what is known as the AIHT variant. Given the iterate x^n we compute $\overline{x^{n+1}}$ in the usual fashion and compute along side it a different candidate. We then either set $x^{n+1} = \overline{x^{n+1}}$ or to the update.

$$a_1 = \frac{\langle b - A\overline{x^{n+1}}, A(\overline{x^{n+1}} - x^n)\rangle}{||A(\overline{x^{n+1}} - x^n)||_2^2}$$

and $p^{n+1} = x^{n+1} + a_1(\overline{x^{n+1}} - x^n)$.

$$a_2 = \frac{\langle b - Ap^{n+1}, A(p^{n+1} - x^{n-1})\rangle}{||A(p^{n+1} - x^{n-1})||_2^2}$$

and $q^{n+1} = p^{n+1} + a_2(p^{n+1} - x^{n-1})$. Then we can update x^{n+1} based on the condition: if $||b - AH_K(q^{n+1})||_2^2 > ||b - A\overline{x^{n+1}}||_2^2$ then we set: $x^{n+1} = \overline{x^{n+1}}$ and otherwise, we use: $x^{n+1} = H_K(q^{n+1})$. The above formulas come from the conditions:

$$a_1 = \arg\min_a ||b - Ap^{n+1}||^2 \quad \text{and} \quad a_2 = \arg\min_a ||b - Aq^{n+1}||^2.$$

Expanding the expression for a_1 we have

$$\begin{aligned} & ||b - Ap^{n+1}||^2 \\ = \quad & ||b - A\left(\overline{x^{n+1}} + a_1(\overline{x^{n+1}} - x^n)\right)||^2 \\ = \quad & ||b - A\overline{x^{n+1}}||^2 - 2a_1\langle b - A\overline{x^{n+1}}, A(\overline{x^{n+1}} - x^n)\rangle + a_1^2||A(\overline{x^{n+1}} - x^n)||^2. \end{aligned}$$

Differentiating the above with respect to a_1 setting to zero and solving for a_1 we recover the condition for a_1.

Having described the two thresholding methods above, we now mention that methods based on the approximation of the ℓ_0-norm also exist. Let us mention one such approximation from [21]:

$$||x||_0 = N - \lim_{n\to\infty} G_\sigma(x) \quad \text{where} \quad G_\sigma(x) = \sum_{k=1}^{N} e^{\left(\frac{-x_k^2}{2\sigma^2}\right)},$$

where N is the dimension of the vector x. Then we can replace $||Ax-b||_2^2+2\tau||x||_0$ by the functional $||Ax-b||_2^2 + 2\tau\left(N - G_\sigma(x)\right)$ and for small σ this would approximate the above functional. As before in the ℓ_1 approximation case, both terms of the new functional are smooth. In the above reference a steepest descent algorithm was suggested. Here we instead show a scheme based on Newton's method. The gradient is:

$$\nabla F(x) = 2A^T(Ax-b) + 2\tau\frac{1}{\sigma^2}(x_1 e^{-\frac{x_1^2}{2\sigma^2}}, \ldots, x_N e^{-\frac{x_N^2}{2\sigma^2}}).$$

For Newton's method we can compute the Hessian matrix:

$$\nabla^2 F(x) = 2A^TA + 2\tau\frac{1}{\sigma^2}\,\mathrm{Diag}\left((x_k^2 - \sigma^2)e^{\frac{-x_k^2}{2\sigma^2}}\right).$$

and then iterate:

$$(\nabla^2 F(x^n))\delta_{x^n} = -\nabla F(x^n) \quad ; \quad x^{n+1} = x^n + \delta_{x^n}.$$

However, care must be taken to avoid local minimum that can be present because of the highly non-convex $||x||_0$ term. The general strategy is to prescribe a decreasing sequence of σ's and solve the above for each σ in a successively, reusing the previous solution as the initial guess for the new one.

The ℓ_0 methods mentioned above are introduced to show the similarity between the classes of methods for ℓ_0 and ℓ_1 problems. The performance of the above schemes however, depends rather strongly on tight RIP bounds of A. Thus, they will not work well for the matrix in our application which does not satisfy such bounds. One approach is to perform first ℓ_1 minimization and then switch to ℓ_0. The IRLS algorithms discussed in a later section provide a way to easily and gradually switch from convex to nonconvex optimization and are likely to be more successful for not well conditioned systems; avoiding the sharp switch.

2.10 Chapter Remarks and Conclusions

This chapter serves as a mathematical introduction into the rest of the thesis and places its contents into place inside a larger theme: the regularization of large scale inverse problems with sparsity constraints for not well-conditioned and noisy systems arising from physical inverse problems. The chapter described why regularization is necessary, gave an introduction to sparsity, the ℓ_0 and ℓ_1 functionals that are minimized to get sparse solutions and then gave several categories of methods for treating the convex ℓ_1 functional followed by a short discussion on ℓ_0. We saw several different ways to deal with the ℓ_1 functional: directly treating the non-smooth term via soft thresholding, the dual space approach, splitting methods, approximating the smooth term by a differentiable function, and minimization by a coordinate wise method. The next two chapters describe new material and are extensions of ideas presented here, presented with detailed convergence arguments. First, we discuss an alternative to the simple ISTA scheme, that while having the same form, uses a different thresholding function and exhibits more promising numerical performance. Next, we introduce two new methods based on the iteratively reweighted least squares approach, where we replace the non-smooth portion of the sparsity promoting functional by a smooth approximation. We also generalize the functional we are minimizing, introducing a more general sparsity promoting penalty.

Chapter 3

AN ALTERNATIVE TO SOFT THRESHOLDING: A NEW SPARSITY-TARGETING VARIABLE THRESHOLDING ALGORITHM

3.1 Overview

In the previous chapter, we saw that direct minimization of the ℓ_1 functional $||Ax - b||_2^2 + 2\tau||x||_1$ leads to the use of the soft thresholding operator $\mathbb{S}_\tau(x)$. We saw also that the soft thresholding operator has the property that it penalizes even the large entries of the vector x. In this chapter we introduce an algorithm (IVTA) and its correspoding fast version (FIVTA) which uses a different thresholding function that does not have this property. The motivation for the new algorithms comes from considering the performance of the soft-thresholding algorithms at different regularization parameters τ. The figure below shows the decrease versus iteration of $||x^n - x^{n-1}||_2$ at two different regularization parameters τ with the FISTA scheme, for a sparse input:

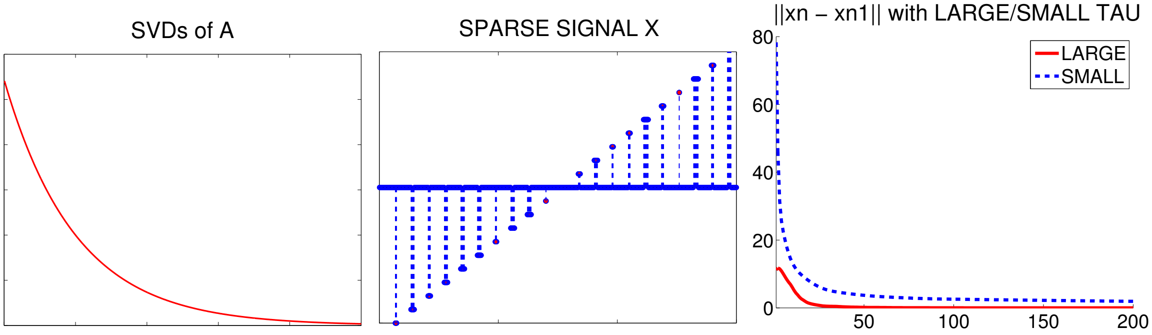


Figure 3.1: Left: distribution of matrix singular values. Center: sparse input vector. Right: $||x^n - x^{n-1}||_2$ versus iteration at $\tau = \frac{||A^T b||_\infty}{3}$ and $\tau = \frac{||A^T b||_\infty}{30000}$.

We observe that $||x^n - x^{n-1}||_2$ is decreased substantially faster at the higher τ. Hence, we expect (and observe in practice) that the convergence is faster at a higher parameter τ. On the other hand, the regularization parameter is typically chosen in order to obtain a given end residual value $||Ax_\tau - b||_2$ where x_τ is the solution of the algorithm obtained at the specified τ. Thus, of numerical interest, would be an algorithm which produces the same residual $||Ax - b||_2$ value at a higher τ as compared to soft thresholding. At this higher τ the numerical convergence would then be faster.

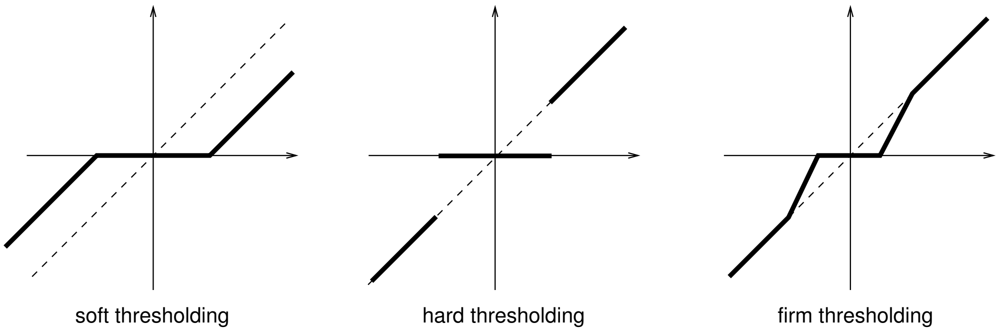
As motivation, we take the results from the work in [29]. There, a two-step procedure was considered to lower the residual value, in which one first finds the minimizer $\widetilde{x}$ of $||Ax - b||_2^2 + \tau||x||_1$, and then repeats the procedure after substituting $\widetilde{b} = 2b - A\widetilde{x}$ for b, thus "adding on to" b an extra piece to protect against the shrinkage in the soft thresholding. In the case where A is the identity operator, $\widetilde{x} = \mathbb{S}_{\frac{\tau}{2}}(b)$ and one easily checks that the end result $\widetilde{\mathbb{S}}_{\frac{\tau}{2}}(b)$ of the two-step procedure (i.e. $\widetilde{\mathbb{S}}_{\frac{\tau}{2}}(b) = \arg\min_x \left(||x - (b + (b - \mathbb{S}_{\frac{\tau}{2}}(b)))|| + \tau||x||_1\right) = \mathbb{S}_{\frac{\tau}{2}}(2b - \mathbb{S}_{\frac{\tau}{2}}(b)))$ is given by

$$\widetilde{\mathbb{S}}_\tau(b)_k = \begin{cases} b_k & |b_k| \geq \tau, \\ \operatorname{sign}(b_k)\,(2|b_k| - \tau) & \tau \geq |b_k| \geq \tau/2, \\ 0 & |b_k| \leq \tau/2. \end{cases} \tag{3.1.1}$$

We see that the above operator $\widetilde{\mathbb{S}}_\tau$ does not penalize the large entries (those bigger in absolute magnitude than τ). For this reason (and because in [29] the two-step approach was shown to have numerical advantages), it seems plausible to construct an algorithm similar to ISTA, that is based around a different thresholding function with the above property, in the hope that it would lead to a decrease of the final residual value $||Ax - b||_2$ at the same τ compared to soft thresholding. This is the subject of this chapter. Just like in the case of FISTA for ISTA, a corresponding fast version of the algorithm is also posed. What follows is largely taken from the published paper [43] on this work.

3.2 Firm Thresholding

This operation $\widetilde{\mathbb{S}}_\tau$, intermediate between soft and hard thresholding (see Fig. 1), is called "firm" thresholding (see e.g. [22] and [14]).



The firm thresholding operator $\widetilde{\mathbb{S}}_\tau$ is continuous in parameter τ and does not shrink large entries, as desired. It may be of interest, in comparing soft, firm and hard thresholding, to note the results in [14]. There in a rigorous study of phase transitions in the behavior of limits for different iterative thresholding algorithms, as a function of the parameters in the problem, it is shown that firm thresholding has a phase transition threshold that is strictly better than that of soft thresholding. No such result is available for iterative hard thresholding.

In [29] it was observed that, for a toy problem inspired by seismography, the solution of the two-step procedure described above, now for an operator A representing the effect of scattering through a multi-layered Earth model on seismic waves, led to a better, "more physical" solution of the original problem. For the large scale problems towards which the study in [29] provided a warm up, it would be (computationally) costly to systematically carry out the iterative solution procedure twice. In this paper we investigate a direct approach to achieve a similar goal, without this costly second iteration. This direct approach makes use of firm thresholding, and leads to a non-convex weight function (which will be equal to $h_{\frac{\tau}{2},\tau}$, in the notation we introduce below), which has the advantage that minimization using this weight function leaves sufficiently large entries untouched, as opposed to soft-thresholding.

The non-convexity of the functional corresponding to firm thresholding, means that minimization of the functional via an iterative procedure, starting from some initial x_0, gives no guarantees that the limit provides a global (as opposed to a local) minimizer. Worse, the very feature that the minimization of the new functional does not affect, ultimately, sufficiently large entries in its minimizer, means that, unless the iteration is set up carefully, the whole procedure can fail to be regularizing, and may, for certain initial data b, lead to non-converging sequences $(x^n)_{n\in\mathbb{N}}$. In addition, it is easy to set up situations where simply minimizing $h_{\frac{\tau}{2},\tau}$, leaving the large entries unchanged, leads to highly non-sparse solutions, thus defeating our purpose.

For these reasons we propose a hybrid approach, in which we switch from soft thresholding to firm thresholding when the iterate becomes increasingly sparse. In fact we have a more gradual approach. We start with soft thresholding (which corresponds to weight function $h_{\tau,\tau}$ in our notation) and when the sparsity increases we use a weight function $h_{\rho,\tau}$, $\frac{\tau}{2} \le \rho \le \tau$, where ρ is determined by the sparsity of the current iterate; the weight function becomes $h_{\frac{\tau}{2},\tau}$ when the signal has a close to expected level of sparsity. To guard against non-termination we have to introduce an auxiliary term to make sure that when we switch weight functions there is still an overall quantity that we are reducing in each step. Next, to obtain boundedness of the sequence of iterates we have to restrict the use of the weight function $h_{\frac{\tau}{2},\tau}$ to when the iterate x^n is assured to satisfy $\|Ax^n\| \ge s\|x^n\|$, for some $s > 0$. This is done by assuming that for sufficiently sparse signals x, the matrix A satisfies $\|Ax\| \ge s\|x\|$. In other words, we only allow the use of the weight function $h_{\frac{\tau}{2},\tau}$ when the number of nonzero entries in the iterate x is below the spark of the matrix A. Recall from [17] that the *spark* of a matrix A is the smallest number of linearly dependent columns in the matrix.

This paper is organized as follows. In Section 2 we discuss the different weight functions. In Section 3 we introduce the new algorithm and we show its termination. In Section 4 we present numerical results.

3.3 Weight functions

Proposition 3.3.1. *Let* $0 < \frac{\tau}{2} \leq \rho \leq \tau$*, and define the function* $h_{\rho,\tau} : \mathbb{R} \to [0, \infty)$ *via*

$$h_{\rho,\tau}(x) = \begin{cases} (4\rho - 2\tau)|x| + 2(\tau - \rho)^2, & |x| \geq 2(\tau - \rho); \\ -\frac{1}{2}x^2 + 2\rho|x|, & |x| < 2(\tau - \rho). \end{cases} \tag{3.3.1}$$

Put

$$\mathbb{S}_{\rho,\tau}(a) = \text{argmin}_{x \in \mathbb{R}} (x - a)^2 + h_{\rho,\tau}(x). \tag{3.3.2}$$

Then

$$\mathbb{S}_{\rho,\tau}(a) = \begin{cases} a - (2\rho - \tau), & a \geq \tau; \\ 2(a - \rho), & \rho < a < \tau; \\ 0, & -\rho \leq a \leq \rho; \\ 2(a + \rho), & -\tau < a < -\rho \ ; \\ a + (2\rho - \tau), & a \leq -\tau \ . \end{cases} \tag{3.3.3}$$

Notice that

$$|\mathbb{S}_{\rho,\tau}(a)| \leq |a| \tag{3.3.4}$$

for all $a \in \mathbb{R}$, and all $0 < \frac{\tau}{2} \leq \rho \leq \tau$. In addition, note that when $\rho = \frac{\tau}{2}$ then $\mathbb{S}_{\rho,\tau}(a) = a$ for $|a| \geq \tau$, while for any other ρ, $\mathbb{S}_{\rho,\tau}(a) \neq a$ when $a \neq 0$. Thus when the larger entries are likely to be the correct value it is advantageous to take $\rho = \frac{\tau}{2}$.

Proof. Let $f(x) = (x - a)^2 + (4\rho - 2\tau)x + 2(\tau - \rho)^2$ and $g(x) = (x - a)^2 - \frac{1}{2}x^2 + 2\rho x$. Then $f'(x) = 0$ if and only if $x = a - (2\rho - \tau)$ and $g'(x) = 0$ if and only if $x = 2(a - \rho)$. Notice that $a - (2\rho - \tau) \geq 2(\rho - \tau)$ if and only if $a \geq \tau$, and $0 < 2(a - \rho) < 2(\tau - \rho)$ if and only if $\rho < a < \tau$. Next, one needs to compare the values at the different critical points. E.g., when $a \geq \tau$, one computes that

$$\begin{aligned} f(a-(2\rho-\tau)) &= (2\rho-\tau)^2 + 2(a-(2\rho-\tau))(2\rho-\tau) + 2(\rho-\tau)^2 \\ &= a^2 - (a-(2\rho-\tau)^2) + 2(\tau-\rho)^2 \\ &\leq a^2 - (\tau-(2\rho-\tau))^2 + 4(\tau-\rho)^2 = a^2 \end{aligned}$$

where the latter is the value at 0 of $(x-a)^2 + h_{\rho,\tau}(x)$. □

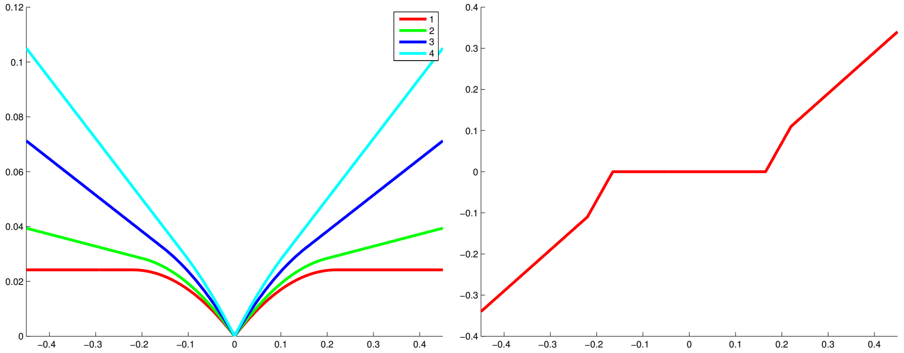


Figure 3.2: Left: $h_{\rho,\tau}(x)$ for $\rho = 0.5, 0.55, 0.65, 0.75$ of τ. Right: $\mathbb{S}_{\rho,\tau}(x)$ for $\rho = 0.75\tau$.

In the following we shall allow vectors x as the argument of $h_{\rho,\tau}$ and $\mathbb{S}_{\rho,\tau}$. With an abuse of notation, we define for $x \in \mathbb{R}^N$,

$$h_{\rho,\tau}(x) = \sum_{i=1}^{N} h_{\rho,\tau}(x_i) \in \mathbb{R}, \ \mathbb{S}_{\rho,\tau}(x) = (\mathbb{S}_{\rho,\tau}(x_i))_{i=1}^N \in \mathbb{R}^N.$$

Next, let us describe the fixed points of the mapping $x \mapsto \mathbb{S}_{\rho,\tau}(x + A^*(b - Ax))$. This mapping will be used in the algorithm in the next section, and the convergence result leads to a fixed point of this mapping. As usual we denote $||x||_\infty = \max_i |x_i|$.

Proposition 3.3.2. *Let $A \in \mathbb{R}^{m\times N}$ and $b \in \mathbb{R}^m$. Suppose $x \in \mathbb{R}^N$ satisfies $x = \mathbb{S}_{\rho,\tau}(x + A^*(b - Ax))$. Then*

- $x_i = 0 \Rightarrow |(A^*(b - Ax))_i| < \rho$,

- $0 < x_i \leq 2(\tau - \rho) \Rightarrow x_i = 2\rho - 2(A^*(b - Ax))_i$,

- $x_i \geq 2(\tau - \rho) \Rightarrow (A^*(b - Ax))_i = 2\rho - \tau$,

- $-2(\tau - \rho) \leq x_i < 0 \Rightarrow x_i = 2\rho - 2(A^*(b - Ax))_i$,

- $x_i \leq -2(\tau - \rho) \Rightarrow (A^*(b - Ax))_i = -(2\rho - \tau)$.

In addition, if $\rho \geq ||A^*b||_\infty$ *then* $x = 0$ *satisfies* $x = \mathbb{S}_{\rho,\tau}(x + A^*(b - Ax))$.

Proof. Proof is by direct verification. □

3.4 Iterative Varied Thresholding Algorithm

Let A be an $m \times N$ real matrix, and b a vector in $\mathbb{R}^m$. We seek a sparse vector x so that

$$\|Ax - b\|$$

is small. In our algorithm we will use the weight function $h_{\rho,\tau}$ with different ρ's and τ's. As the proposed algorithm is a variation on the iterative soft-thresholding algorithm (ISTA) where now the thresholding function changes based on the sparsity level of the iterate. We will call our algorithm the Iterative Varied Thresholding Algorithm (IVTA). We start with explaining how to proceed when a value for τ has been chosen. The algorithm is given below:

Algorithm 1: IVTA for Sparse Signal Recovery

```
Input : An $m \times N$ matrix $A$ and a vector $b \in \mathbb{R}^m$, a leverage $L$, tolerance $\epsilon$,
        an estimated sparsity level $K$, and an initial guess $x^0$.
Output: An estimate $\hat{x} \in \mathbb{R}^N$ of the signal $x$
$i \leftarrow 0$;
$\rho_0 \leftarrow \tau$;
$K_0 \leftarrow \frac{N}{5}$;
$L_0 \leftarrow L$;
$a \leftarrow 0$;
begin
    $x^{i+1} \leftarrow \mathbb{S}_{\rho_i,\tau}(x^i + A^*b - A^*Ax^i)$;
    if $\|x^i - x^{i+1}\| \leq \epsilon$ then
        break
    end
    $K_{i+1} \leftarrow nnz(x^{i+1})$;
    $\sigma \leftarrow \frac{\tau}{2}(1 + \frac{\max(K_{i+1}-K_0,a)}{N-K_0})$;
    if $\sigma \leq \rho_i$ then
        $L_{i+1} \leftarrow L_i$, $\rho_{i+1} \leftarrow \sigma$
    else
        $L_{i+1} \leftarrow L_i + h_{\rho_i,\tau}(x^{i+1}) - h_{\sigma,\tau}(x^{i+1})$
        if $L_{i+1} \geq 0$ then
            $\rho_{i+1} \leftarrow \sigma$
        else
            $\rho_{i+1} \leftarrow \max(\rho_i, \frac{\tau}{2}(1 + \frac{1}{N-K_0})), a \leftarrow 1$
        end
    end
    $i \leftarrow i + 1$;
end
$\hat{x} \leftarrow x^{(i)}$
```

The algorithm uses the following constants:

ϵ: the iteration at a given τ stops when $\|x^i - x^{i+1}\| \leq \epsilon$;

K: Estimate for the number of nonzeros of the signal. If the iterate becomes K-sparse, the weight function $h_{\frac{\tau}{2},\tau}$ is used. This choice does not fix or bound the number of nonzero entries in the final solution.

L: a 'leverage' or safety margin. Indeed, it is conceivable that, by a fluke, a very sparse iterate occurs that is not close to the actual solution x. If that happens, we expect a later iterate to be less sparse, to which the algorithm will respond by increasing ρ. This increase in ρ causes a corresponding increase in $h_{\rho,\tau}(x^i)$. If this can happen unchecked, then it can spoil convergence. The leverage quantity L is defined to keep track of this: if the sum of all the increases of $h_{\rho,\tau}(x^i)$, due to increases in ρ, exceeds a certain ceiling (the initial value of L), then we will not allow ρ to increase in any of the further iterations steps, so that from then on, $h_{\rho,\tau}(x^i)$ can only decrease. With this restriction in place, the number of nonzero entries of later iterates may exceed K; on the other hand, when $\rho = \frac{\tau}{2}$ convergence is guaranteed only if the number of nonzeros does not exceed K. To ensure convergence we therefore also put in place (via the parameter a) a lower bound on ρ that is strictly larger than $\frac{\tau}{2}$, as soon as $L < 0$ has occurred. (In practice, this safety valve in the algorithm turns out not to be activated, for the applications we considered.)

We now prove through a series of propositions that the algorithm terminates. Let us start by defining:

$$\begin{aligned} F_{L,\rho}(x) &= \|Ax-b\|^2 + h_{\rho,\tau}(x) + L, && (3.4.1) \\ G_{L,\rho}(x;\tilde{x}) &= F_{L,\rho}(x) + \langle (I - A^*A)(x-\tilde{x}), x-\tilde{x}\rangle. && (3.4.2) \end{aligned}$$

Lemma 3.4.1. $G_{L,\rho_n}(x^{n+1}, x^n) \leq G_{L,\rho_n}(x^n, x^n)$.

Proof. Consider that $\mathbb{S}_{\rho,\tau}(a) = \arg\min_x (x-a)^2 + h_{\rho,\tau}(x)$ and that $x^{n+1} = \mathbb{S}_{\rho_n,\tau}(x^n + (A^*b) - (A^*Ax^n)) = \mathbb{S}_{\rho_n,\tau}(y^n)$, where $y^n = x^n + (A^*b) - (A^*Ax^n)$. This implies that $x^{n+1} = \arg\min_x ||x - y^n||^2 + h_{\rho_n,\tau}(x)$. Consequently:

$$||x^{n+1} - y^n||^2 + h_{\rho_n,\tau}(x^{n+1}) \leq ||x^n - y^n||^2 + h_{\rho_n,\tau}(x^n).$$

Plugging in $y^n = x^n + (A^*b) - (A^*Ax^n)$ we get:

$$||x^{n+1} - x^n||^2 - 2\langle x^{n+1} - x^n, A^*(b - Ax^n)\rangle + h_{\rho_n,\tau}(x^{n+1}) \leq h_{\rho_n,\tau}(x^n).$$

Next,

$$\begin{aligned}
G_{L,\rho_n}(x^n,x^n) &= ||Ax^n-b||_2^2+h_{\rho_n,\tau}(x^n)+L,\\
G_{L,\rho_n}(x^{n+1},x^n) &= ||Ax^{n+1}-b||_2^2+||x^{n+1}-x^n||_2^2-||A(x^{n+1}-x^n)||_2^2+h_{\rho_n,\tau}(x^{n+1})+L\\
&= ||Ax^n-b||_2^2-2\langle x^{n+1}-x^n, A^*(b-Ax^n)\rangle+h_{\rho_n,\tau}(x^{n+1})+L\\
&\leq ||Ax^n-b||_2^2+h_{\rho_n,\tau}(x^n)-||x^{n+1}-x^n||_2^2+L\\
&\leq G_{L,\rho_n}(x^n,x^n),
\end{aligned}$$

and the result follows. □

Lemma 3.4.2. *For $\rho_1 \geq \rho_2 \in [\frac{\tau}{2},\tau]$ we have $h_{\rho_1,\tau}(x) \geq h_{\rho_2,\tau}(x)$.*

Proof. Fix any $x \in \mathbb{R}$. Depending on the value of x, we have three cases to consider for the functions h. Consider first $|x| \geq 2(\tau-\rho_2) > 2(\tau-\rho_1)$:

$$h_{\rho_1,\tau}(x) = (4\rho_1-2\tau)|x|+2(\tau-\rho_1)^2 \quad \text{and} \quad h_{\rho_2,\tau}(x) = (4\rho_2-2\tau)|x|+2(\tau-\rho_2)^2.$$

Then

$$\begin{aligned}
h_{\rho_1,\tau}(x)-h_{\rho_2,\tau}(x) &= 4(\rho_1-\rho_2)|x|+2(\rho_1^2-2\tau\rho_1-\rho_2^2+2\tau\rho_2)\\
&= 4(\rho_1-\rho_2)|x|+2(\rho_1-\rho_2)(\rho_1+\rho_2-2\tau)\\
&= 2(\rho_1-\rho_2)(2|x|+\rho_1+\rho_2-2\tau).
\end{aligned}$$

Now, $|x| \geq 2(\tau-\rho_2)$ yields that $\frac{1}{2}|x| \geq \tau-\rho_2$ and $|x| \geq 2(\tau-\rho_1)$ yields that $\frac{1}{2}|x| \geq \tau-\rho_1$ so that $|x| \geq 2\tau-\rho_1-\rho_2$. So this means $\rho_1+\rho_2-\tau \geq -|x|$ and so $2(\rho_1-\rho_2)(2|x|+\rho_1+\rho_2-2\tau) \geq 0$, which implies that $h_{\rho_1,\tau}(x) \geq h_{\rho_2,\tau}(x)$.

Next consider the case $2(\tau-\rho_1) \leq |x| < 2(\tau-\rho_2)$. In this case:

$$h_{\rho_1,\tau}(x) = (4\rho_1 - 2\tau)|x| + 2(\tau-\rho_1)^2 \quad \text{and} \quad h_{\rho_2,\tau}(x) = -\frac{1}{2}x^2 + 2\rho_2|x|.$$

Then

$$\begin{aligned} h_{\rho_1,\tau}(x) - h_{\rho_2,\tau}(x) &= (4\rho_1 - 2\tau)|x| + 2(\tau-\rho_1)^2 + \frac{1}{2}x^2 - 2\rho_2|x| \\ &= (4\rho_1 - 2\rho_2 - 2\tau)|x| + \frac{1}{2}\left(4(\tau-\rho_1)^2 + x^2\right). \end{aligned}$$

Now, by the arithmetic/geometric mean inequality: $\frac{p+q}{2} \geq \sqrt{pq}$ for $p, q \geq 0$, we obtain

$$\frac{1}{2}\left(4(\tau-\rho_1)^2 + x^2\right) \geq \sqrt{4(\tau-\rho_1)^2x^2}.$$

So

$$\begin{aligned} &h_{\rho_1,\tau}(x) - h_{\rho_2,\tau}(x) = (4\rho_1 - 2\rho_2 - 2\tau)|x| + \frac{1}{2}\left(4(\tau-\rho_1)^2 + x^2\right) \\ &\geq (4\rho_1 - 2\rho_2 - 2\tau)|x| + \sqrt{4(\tau-\rho_1)^2x^2} \\ &= (4\rho_1 - 2\rho_2 - 2\tau)|x| + 2(\tau-\rho_1)|x| = (4\rho_1 - 2\rho_2 - 2\tau + 2\tau - 2\rho_1)|x| = 2(\rho_1-\rho_2)|x| \geq 0. \end{aligned}$$

So that in this case, $h_{\rho_1,\tau}(x) \geq h_{\rho_2,\tau}(x)$ as well.

Finally, consider the case $|x| < 2(\tau-\rho_1) < 2(\tau-\rho_2)$. Then

$$h_{\rho_1,\tau}(x) = -\frac{1}{2}x^2 + 2\rho_1|x| \quad \text{and} \quad h_{\rho_2,\tau}(x) = -\frac{1}{2}x^2 + 2\rho_2|x|,$$

and thus

$$h_{\rho_1,\tau}(x) - h_{\rho_2,\tau}(x) = 2(\rho_1-\rho_2)|x| \geq 0.$$

So in all the possible cases, $h_{\rho_1,\tau}(x) \geq h_{\rho_2,\tau}(x)$. □

Using the above two lemmas we are ready now to prove convergence.

Proposition 3.4.3. *Let $\|A\| < 1$. The above iteration yields that $\|x^{n+1} - x^n\| \to 0$ as $n \to \infty$.*

Proof. Notice that in the algorithm it can happen that the number a will remain 0 all the time, or that it gets set to 1. In the latter case we will from then on have that ρ will no longer increase and the L will no longer play a role. This case when $\rho_n \geq \rho_{n+1}$ will be covered under $a = 0$ case, so for the remainder we will assume that a remains 0, which means that all L_n are nonnegative.

We will show that for all n we have that

$$(1 - \|A\|^2)\|x^n - x^{n+1}\|^2 \leq F_{L_n,\rho_n}(x^n) - F_{L_{n+1},\rho_{n+1}}(x^{n+1}). \tag{3.4.3}$$

There are two cases, one where L remains unchanged from one iteration to the next, and one where L changes value. In the first case, $L_n = L_{n+1}$, which implies that $\rho_n \geq \rho_{n+1}$. Then we get that:

$$\begin{aligned}F_{L_n,\rho_n}(x^n) &= G_{L_n,\rho_n}(x^n; x^n) \geq G_{L_n,\rho_n}(x^{n+1}; x^n)\\ &= F_{L_n,\rho_n}(x^{n+1}) + \langle (I - A^*A)(x^n - x^{n+1}), x^n - x^{n+1}\rangle\\ &\geq F_{L_{n+1},\rho_{n+1}}(x^{n+1}) + (1 - \|A\|^2)\|x^n - x^{n+1}\|^2\\ &\geq F_{L_{n+1},\rho_{n+1}}(x^{n+1}),\end{aligned}$$

where in the first inequality we used Lemma 3.4.2 and that $L_n = L_{n+1}$. Next, let us consider the case when L changes value. In that case we have that

$$L_{n+1} = L_n + h_{\rho_n,\tau}(x^{n+1}) - h_{\rho_{n+1},\tau}(x^{n+1}). \tag{3.4.4}$$

But then we get that:

$$\begin{aligned}
F_{L_n,\rho_n}(x^n) &= G_{L_n,\rho_n}(x^n;x^n) \geq G_{L_n,\rho_n}(x^{n+1};x^n) \\
&= F_{L_n,\rho_n}(x^{n+1}) + \langle (I - A^*A)(x^n - x^{n+1}), x^n - x^{n+1} \rangle \\
&= F_{L_{n+1},\rho_n}(x^{n+1}) + \langle (I - A^*A)(x^n - x^{n+1}), x^n - x^{n+1} \rangle \\
&\geq F_{L_{n+1},\rho_{n+1}}(x^{n+1}) + (1 - \|A\|^2)\|x^n - x^{n+1}\|^2 \\
&\geq F_{L_{n+1},\rho_{n+1}}(x^{n+1}).
\end{aligned}$$

This establishes the proof of (3.4.3).

From (3.4.3) we now get that

$$(1 - \|A\|^2) \sum_{n=0}^{p} \|x^{n+1} - x^n\|^2 \leq F_{L_0,\rho_0}(x_0) - F_{L_{n+1},\rho_{n+1}}(x_{n+1}) \leq F_{L_0,\rho_0}(x_0).$$

As $1 - \|A\|^2 > 0$, we get that $\sum_{n=0}^{\infty} \|x^{n+1} - x^n\|^2$ converges, and thus we must have that $\|x^{n+1} - x^n\|$ converges to 0 as $n \to \infty$. □

It should be noticed that the function $h_{\rho,\tau}$ for $\rho = \frac{\tau}{2}$ flattens out. Therefore, boundedness of $h_{\rho,\tau}(x^n)$ does not imply that $(x^n)_n$ is bounded. Still, using this function has the advantage that optimizing over it leaves certain entries of x fixed. This is useful when one starts approaching the true solution. Our algorithm is set up so that ρ reaches the value $\frac{\tau}{2}$ when the number of nonzero entries in x^n is below a given constant K. We can show convergence when K is chosen in such a way so that no K columns in A are linearly dependent; that is, we choose $K < \text{spark}(A)$. First consider the following example.

Example 3.4.4. Let A be so that its kernel contains a vector z with all nonzero entries. Let v and b be so that $Av = b$. Choose $R \in \mathbb{R}$ so that all entries in

$x_0 = v + Rz$ have absolute value larger than τ. Then one easily checks that

$$x^1 := \mathbb{S}_{\frac{\tau}{2},\tau}(x^0 + A^*(b - Ax^0)) = S_{\frac{\tau}{2},\tau}(x_0) = x_0,$$

so that an iteration scheme

$$x^{n+1} = \mathbb{S}_{\frac{\tau}{2},\tau}(x^n + A^*(b - Ax^n))$$

would not lead to a desired sparse solution.

Introduce for $A \in \mathbb{R}^{m\times N}$ and $1 \le p \le N$, the number

$$s(A; p) = \min_{|X|=p} s_p(A|X),$$

where X ranges over all subsets of $\{1, \ldots, N\}$ with cardinality p, $A|X$ stands for the $n \times p$ matrix obtained from A by omitting all columns except the ones indexed by X, and s_p denotes the pth singular value (where s_1 is the largest singular value).

Lemma 3.4.5. *Let $A \in \mathbb{R}^{m\times N}$ be a contraction, and $1 \le p \le N$. Put $s = s(A; p)$. Let x be a vector with at most p nonzero entries. Then*

$$\|(I - A^*A)x\| \le \sqrt{1 - s^2}\|x\|. \tag{3.4.5}$$

Proof. Let X have cardinality p so that the support of the vector x lies in X. Let $\tilde{x} \in \mathbb{R}^p$ consist of the components of x indexed by X. Then

$$\begin{aligned}\|(I - A^*A)^{1/2}x\|^2 &= \langle (I_N - A^*A)x, x\rangle \\ &= \langle (I_p - (A|X)^*(A|X))\tilde{x}, \tilde{x}\rangle \\ &\le (1 - s^2)\|\tilde{x}\|^2 \\ &= (1 - s^2)\|x\|^2.\end{aligned}$$

And thus

$$\|(I - A^*A)x\| \leq \|(I - A^*A)^{1/2}\|\|(I - A^*A)^{1/2}x\| \leq \sqrt{1 - s^2}\|x\|,$$

as $\|(I - A^*A)^{1/2}\| \leq 1$. □

We now have the following key result.

Theorem 3.4.6. *Let $A \in \mathbb{R}^{m \times N}$ be a strict contraction, and let $0 \leq K \leq N$ be chosen so that*

$$s := s(A, K) > 0. \tag{3.4.6}$$

Then Algorithm 1 produces a bounded sequence $(x^n)_{n \in \mathbb{N}}$.

Proof. For the same reason as in the proof of Proposition 3.4.3 we will go by the assumption that the value for a remains 0. Indeed, if a is set to equal 1 at some point, we will have that from then on $\rho_n \geq r$, where $r = \frac{\tau}{2}(1 + \frac{1}{N})$. This case will be covered below. So let us assume that $a = 0$ throughout.

First observe that there exists an M so that

$$h_{r,\tau}(x) \leq F_{L_0,\rho_0}(x^0)$$

implies that $\|x\| \leq M$. Indeed, since $r > \frac{\tau}{2}$, we have that $h_{r,\tau}(x) \to \infty$ as $\|x\| \to \infty$. As $h_{\rho,\tau}(x) \geq h_{r,\tau}$ when $\rho \geq r$, we also get for such ρ that $h_{\rho,\tau}(x) \leq F_{L_0,\rho_0}(x^0)$ implies that $\|x\| \leq M$.

We have two cases in the algorithm to consider: $\rho_n \geq r$ and $\rho_n = \frac{\tau}{2}$.

The latter only occurs when x^n has at most K nonzero entries. We will show that in either case

$$\|x^n\| \leq R + \sqrt{1 - s^2}\|x^{n-1}\|, \tag{3.4.7}$$

where $R = \max\{M, \|A^*b\|\}$.

First suppose that $\rho_n \geq r$. From the proof of Proposition 3.4.3 we have that

$$h_{\rho_n}(x^n) \leq F_{L_n,\rho_n}(x^n) \leq F_{L_0,\rho_0}(x^0),$$

and thus, by the definition of M, we have that

$$\|x^n\| \leq M \leq R \leq R + \sqrt{1-s^2}\|x^{n-1}\|.$$

Next, when $\rho_n < r$ we must have that $\rho_n = \frac{\tau}{2}$ and that x^n has at most K nonzero entries. Using the definition of x^{n+1} and inequality (3.3.4) we have that

$$|(x^{n+1})_p| \leq |(x^n + A^*(b - Ax^n))_p|, \; p = 1, \ldots, N.$$

Thus, by using (3.4.5), we get

$$\begin{aligned}
\|x^{n+1}\| &\leq \|x^n + A^*(b - Ax^n)\| \\
&\leq \|A^*b\| + \|(I - A^*A)x^n\| \\
&\leq R + \sqrt{1-s^2}\|x^n\|,
\end{aligned}$$

proving (3.4.7) in this case as well.

Now, using (3.4.7) repeatedly, we get

$$\begin{aligned}
\|x^n\| &\leq R + \sqrt{1-s^2}\|x^{n-1}\| \\
&\leq R + \sqrt{1-s^2}(R + \sqrt{1-s^2}\|x^{n-2}\|) \\
&\leq \cdots \leq R\sum_{p=0}^{n}(\sqrt{1-s^2})^p \\
&\leq \frac{R}{1-\sqrt{1-s^2}},
\end{aligned}$$

yielding the desired boundedness.

□

In general we cannot guarantee a unique solution, but even in the more classical case with ℓ_1 minimization this can not be expected when A has a kernel (the typical case). E.g., when

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, b = \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

any $x = \begin{pmatrix} a & b \end{pmatrix}^T$ with a and b nonnegative and $a + b = 1$ yields an optimal solution under ℓ_1 minimization. We can, however, guarantee the following.

Proposition 3.4.7. *The sequence (x^n) constructed above has limit points x satisfying $x = \mathbb{S}_{\rho,\tau}(x + A^*(b - Ax))$ for some $\frac{\tau}{2} \leq \rho \leq \tau$.*

Proof. The sequence is bounded and therefore it has a convergent subsequence $(x^{n_k})_k$, say, with limit x, say. As $\lim_{n\to\infty} \|x^n - x^{n+1}\| = 0$, we get that $\lim_{k\to\infty} x^{n_k+1} = x$. Next, we have that

$$x^{n_k+1} = \mathbb{S}_{\rho_{n_k+1},\tau}(x^{n_k} + A^*(b - Ax^{n_k})).$$

As the values for ρ_n lie in a finite set, infinitely many of $\{\rho_{n_k+1}\}_k$ are the same, equaling ρ, say. But then there is a subsequence x^{m_k} so that for all k

$$x^{m_k+1} = \mathbb{S}_{\rho,\tau}(x^{m_k} + A^*(b - Ax^{m_k})).$$

Taking the limit on both sides yields $x = \mathbb{S}_{\rho,\tau}(x + A^*(b - Ax))$. □

Note that the above proposition implies that the algorithm is not gauranteed to minimize any specific function, since ρ is not fixed throughout the iteration but varies with the sparsity of the sucessive iterates, decreasing from $\rho = \tau$ (soft thresholding) to the minimum value just above $\frac{\tau}{2}$. Since $h_{\rho,\tau}(x)$ is not convex for $\rho < \tau$, this behavior helps the algorithm avoid local minima in the nonconvex functional $||Ax-b||_2^2+h_{\rho,\tau}(x)$. Assuming an infinite number of iterations, the IVTA algorithm will stop at a local minimum of this functional for $\rho = \lim_{n\to\infty} \rho_n$ (which will vary depending on the input, but will always be in the range $(\frac{\tau}{2}, \tau)$).

3.5 Numerical Experiments

In this section, we present some simple numerical examples to test the performance of our algorithm. As in the case of the iterative soft-thresholding scheme $x^{n+1} = \mathbb{S}_\tau(x^n + A^*b - A^*Ax^n)$, the IVTA step $x^{n+1} = \mathbb{S}_{\rho^n,\tau}(x^n + A^*b - A^*Ax^n)$ has a corresponding accelerated version, which we term FIVTA, based on the idea of the FISTA speedup (introduced in [2] based on the work of Nesterov [32]):

$$y^0 = x^0,\ x^n = \mathbb{S}_{\rho^n,\tau}(y^n + A^*(b - Ay^n)),$$

$$y^{n+1} = x^n + \frac{t_n - 1}{t_{n+1}}(x^n - x^{n-1}),$$

where t_n is a sequence of numbers, generated by: $t_{n+1} = \frac{1+\sqrt{1+4t_n^2}}{2}$ and $t_1 = 1$. We have tested both the IVTA and FIVTA scheme and found FIVTA to have similar advantages over IVTA as FISTA has over ISTA (see [2]). Consequently we have chosen to use FIVTA in all of our comparisons. We note here that other choices of numerical schemes are possible. For instance, we found that we could apply the monotone version of FISTA to IVTA ([1]). This results in a slightly updated scheme:

$$y^0 = x^0,\ z^n = \mathbb{S}_{\rho^n,\tau}(y^n + A^*(b - Ay^n)),\ x^n = \arg\min_x \left(F(x)\colon x = z^n, x = x^{n-1}\right),$$

$$y^{n+1} = x^n + \frac{t_n}{t_{n+1}}(z^n - x^n) + \frac{t_n - 1}{t_{n+1}}(x^n - x^{n-1}),$$

with $F(x) = ||Ax - b||_2^2 + h_{\frac{\tau}{2},\tau}(x)$. We find that this scheme sometimes converges faster. Finally the two-step overrelaxation acceleration followed by hard thresholding as proposed in [3] also looks promising. Below, we only use the FIVTA scheme for comparisons, but different variants are possible.

We will test our algorithm in the setting of a sparse recovery problem where noise is present. In such a problem one has an estimate of the noise level in the observations b and one looks for a solution x for which $||Ax - b||$ is at this noise level. This leads to a criterion for choosing a proper parameter τ since every choice of τ leads to a particular value of $||Ax_\tau - b||$ where x_τ represents the solution of the algorithm obtained at that τ. Thus, the approach consists of two steps: preliminary runs to identify the right τ and a longer run at the proper τ to find the desired solution up to some convergence criterion such as $\frac{||x^{n+1}-x^n||}{||x^n||} < \epsilon$.

For identifying the preferred parameter τ, the most efficient choice turns out to be a simple scheme where we proceed down from the choice of $\tau = ||A^*b||_\infty$ running the algorithm for a few iterations at each τ starting from the previous solution as the initial guess. Given A and b define

$$\tau_{\max} = ||A^*b||_\infty.$$

The update of τ is based on two constants C and N_τ which in our case we chose to be 5000 and 20. After this choice we set

$$\tau_{\min} = \frac{||A^*b||_\infty}{C}$$

$$\tau\text{step} = \frac{\log(\tau_{\max}) - \log(\tau_{\min})}{N_\tau - 1}.$$

Starting at $\tau = \tau_{\max}$ we iterate according to one of the above schemes (IVTA or FIVTA). After convergence, we update τ to equal $\exp(\tau - \tau\text{step})$ and we use the solution at previous τ as the starting guess for the new iteration. The choice of $\tau_{\max}$ is based on the last observation in Proposition 3.3.2 and the optimality condition for ℓ_1 optimization. In the later, we have $x = 0$ for $\tau \geq ||A^*b||_\infty$ and for FIVTA we expect $x = 0$ for $\tau \geq 2||A^*b||_\infty$, although we find that starting at $\tau = ||A^*b||_\infty$ is sufficient if a low enough $||Ax - b||$ value is ultimately desired. The number N_τ represents the total number of different parameters τ. The benefit for the use of such a scheme versus a more complicated binary search approach is that the solutions at adjacent τ are similar and once one has been obtained, starting with that as the initial guess quickly gives the solution at the next τ. Hence, the residual level at each τ can be identified relatively quickly using this scheme, and a hundred iterations or so at each τ is generally sufficient.

Once the right τ has been identified, a longer run is performed at this given τ. This is where the advantage of FIVTA manifests itself, since the convergence at the right τ is often significantly faster than for soft thresholding methods. This is because we are able to get a more appropriate solution with a lower $||Ax_\tau - b||$ value at a higher parameter τ with FIVTA than with FISTA. The general property of the various available thresholding schemes is that convergence becomes slow for small parameter τ. Consequently, it is desirable to have a lower value for $||Ax_\tau - b||$ at a higher τ so that the noise level matching solution can be obtained more accurately in fewer iterations.

We describe how we construct the test cases for our numerical experiments below. First, the non-zero entries of the vector x are picked, with some degree of randomness, according to the specifications of the example. Next, the matrix A is constructed by generating a random unitary $m \times m$ matrix U, a random unitary $N \times N$ matrix of which the first m rows are retained, giving the $m \times N$ matrix V, and setting $A := U \cdot D \cdot V$, where D is a diagonal $m \times m$ matrix, the diagonal elements of which are given by the singular value distribution for A pre-assigned in the example. Finally, random i.i.d. white noise of the prescribed variance or norm is added to the vector Ax, to define the "data vector" $b := Ax + noise$.

Now we describe the experiments. Experiment set 1 illustrates that FIVTA converges faster than FISTA at the noise matching τ. In fact this τ is also faster to find since it's closer to the maximum value of $||A^*b||_\infty$. This is illustrated for differently conditioned examples. Experiment set 2 illustrates what happens when different properties of the linear system are varied: the number of nonzeros, the noise in b, the uncertainty in A. We see similar or in some cases even better performance to that of FISTA. Experiment set 3 shows that in cases where the right τ cannot be accurately determined, FIVTA often does better, since we have reconstructions with lower $||Ax_\tau - b||$ value, by taking a randomly chosen τ below $||A^*b||_\infty$. If we simply want to do a single run to estimate the solution, then it is advisable to use FIVTA at a reasonably high τ. In experiment set 4, we show that the knowledge of the true number of nonzeros in the unknown signal x is not essential to FIVTA. A rough estimate is enough, as long as the number of nonzeros is not too large. Finally we illustrate a simple example of image denoising, useful for images that are sparsely represented in the wavelet domain.

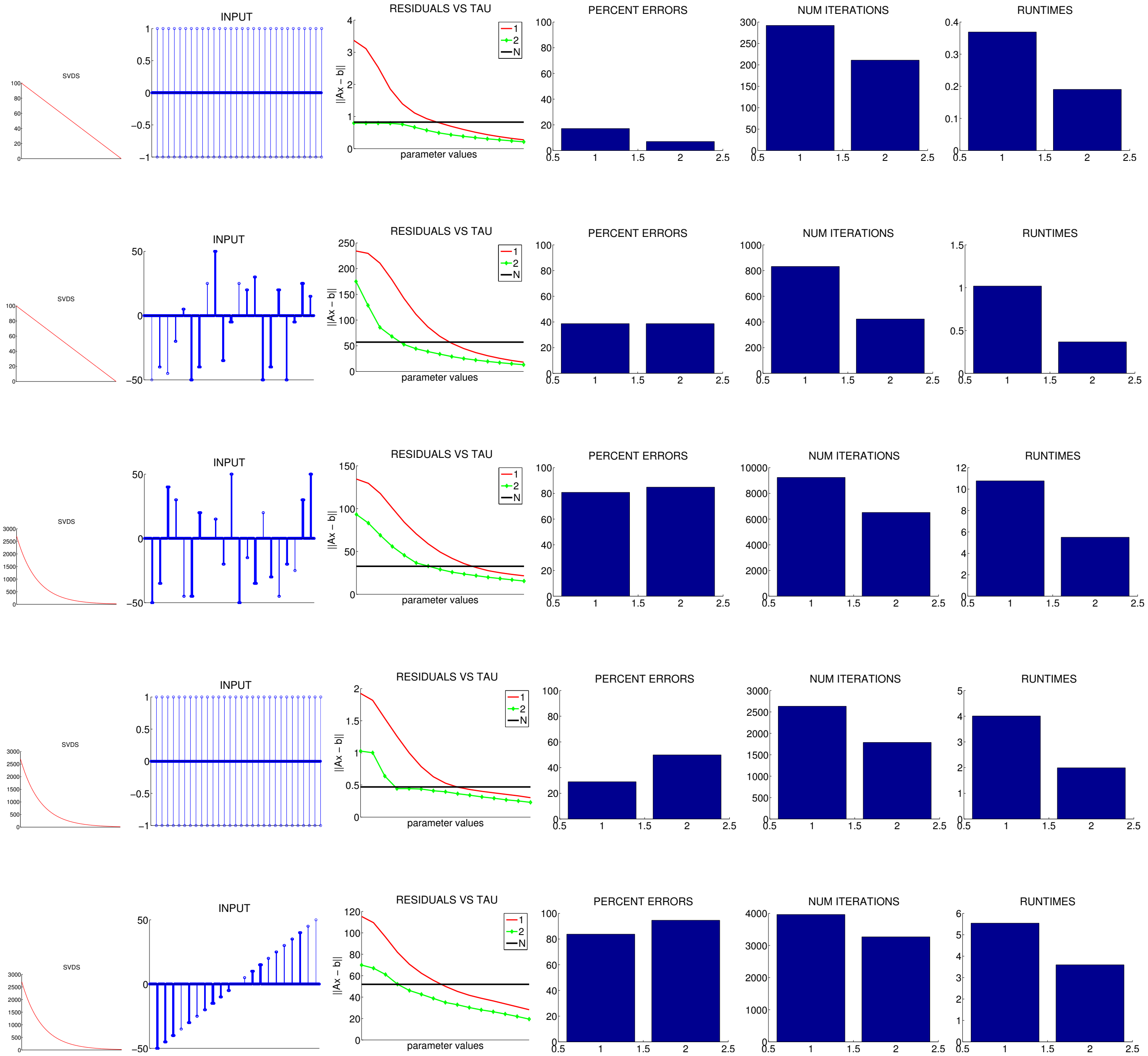


Figure 3.3: Experiments 1: svds, input, $||Ax_\tau - b||$ vs parameter, percent errors, number of iterations and runtimes at the noise matching parameter for FISTA (1) and FIVTA (2) for several examples. We observe that in all the examples, FIVTA matches the noise level residual $||Ax_\tau - b||$ at a higher τ than FISTA which leads to faster convergence at that parameter.

Algorithm 2: FIVTA for Sparse Signal Recovery

```
**Input** : An $m \times N$ matrix $A$ and a vector $b \in \mathbb{R}^m$, a leverage $L$, tolerance $\epsilon$,
            an estimated sparsity level $K$, and an initial guess $x^0$.
**Output**: An estimate $\hat{x} \in \mathbb{R}^N$ of the signal $x$
$i \leftarrow 0$;
$\rho_0 \leftarrow \tau$;
$K_0 \leftarrow \frac{N}{5}$;
$L_0 \leftarrow K_0||A^*b||_1$;
$a \leftarrow 0$;
begin
    if $i = 1$ then
        $x^{i+1} \leftarrow \mathbb{S}_{\rho_i,\tau}(x^i + A^*b - A^*Ax^i)$;
    end
    else
        $y^{i+1} = x^i + \frac{t_i - 1}{t_{i+1}}(x^i - x^{i-1})$;
        $x^{i+1} \leftarrow \mathbb{S}_{\rho_i,\tau}(y^i + A^*b - A^*Ay^i)$
    end
    if $\|x^i - x^{i+1}\| \leq \epsilon$ then
        break
    end
    $K_{i+1} \leftarrow nnz(x^{i+1})$;
    $\sigma \leftarrow \frac{\tau}{2}(1 + \frac{\max(K_{i+1} - K_0, a)}{N - K_0})$;
    if $\sigma \leq \rho_i$ then
        $L_{i+1} \leftarrow L_i$, $\rho_{i+1} \leftarrow \sigma$
    end
    else
        $L_{i+1} \leftarrow L_i + h_{\rho_i,\tau}(x^{i+1}) - h_{\sigma,\tau}(x^{i+1})$
        if $L_{i+1} \geq 0$ then
            $\rho_{i+1} \leftarrow \sigma$
        end
        else
            $\rho_{i+1} \leftarrow \max(\rho_i, \frac{\tau}{2}(1 + \frac{1}{N - K_0})), a \leftarrow 1$
        end
    end
    $i \leftarrow i + 1$;
end
$\hat{x} \leftarrow x^{(i)}$
```

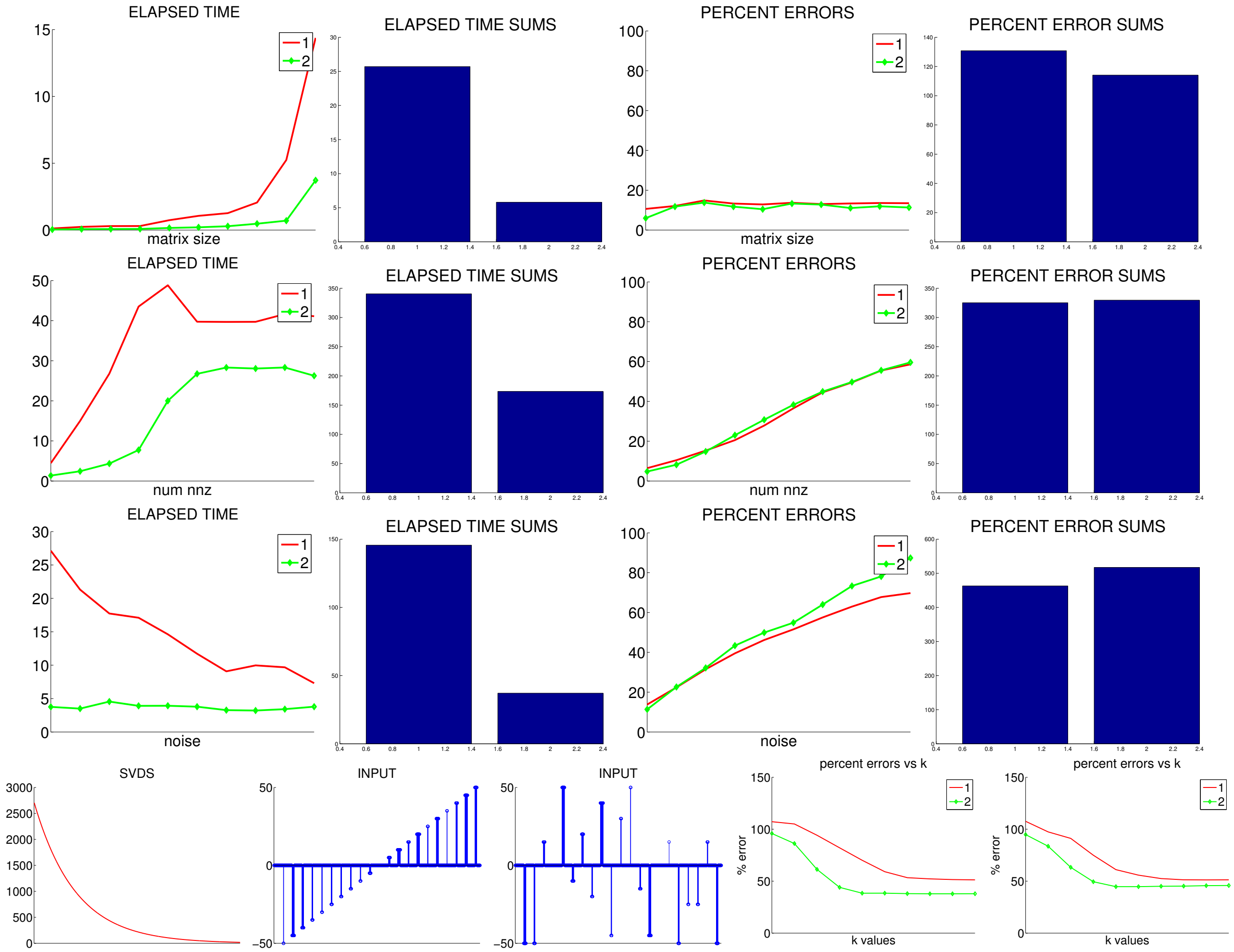


Figure 3.4: Experiments 2: runtimes and percent errors for FISTA (1) and FIVTA (2) measured at the noise matching τ, plotted versus increasing matrix sizes, noise levels in b and number of nonzeros in signal x. When number of nonzeros is fixed: we use 15 percent nonzeros, when noise is fixed we use 10 percent noise in b. All the quantities are medians over 10 different runs. Last row shows also an experiment where we introduce errors into the matrix A by using a low rank SVD approximation $U_k \Sigma_k V_k^T$ with different values of k. We see similar levels of performance between FISTA and FIVTA at significantly less runtime for FIVTA.

Now we test the dependence of FIVTA on τ and K. What if τ is not known in advance (for instance, what if the noise level is unknown and we cannot predict what τ to use). What if the number of nonzeros in the signal cannot be estimated?

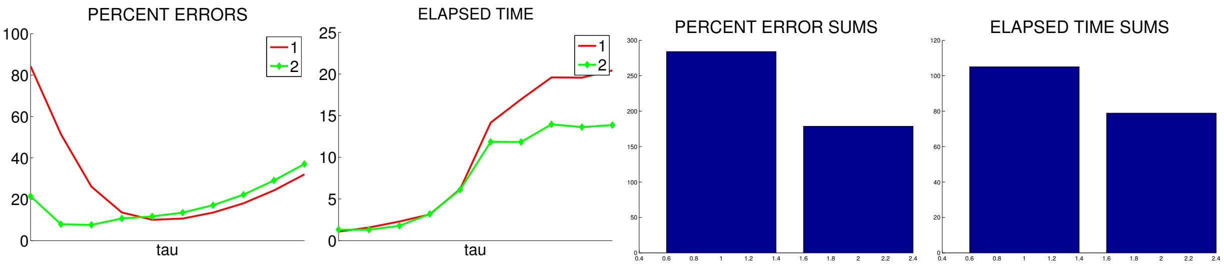


Figure 3.5: Experiments 3 - Dependence on τ: Percent errors and elapsed times as a function of τ. For each τ, we run FISTA and FIVTA from zero initial guess until convergence or maximum number of iterations (3000) is reached. We plot the medians for 10 different runs, a well conditioned random matrix with 15 percent nonzeros is used in each case. We observe that for τ closer to $\max(|A^*b|)$, FIVTA performs better. That is, FIVTA offers an advantage if the right τ to use cannot be precomputed in advance.

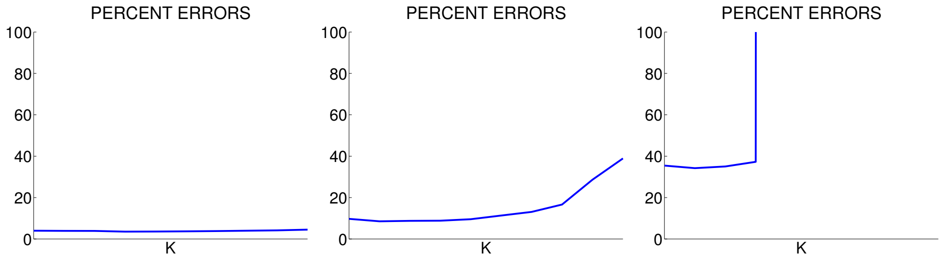


Figure 3.6: Experiments 4 - Dependence on K: We plot the percent error of the reconstruction at the noise matching τ for different starting values of K from $\frac{1}{3}$ to 3 times the actual number of nonzeros in the signal. The three plots correspond to three different signals with increasing percent of nonzeros: 3 percent, 10 percent, and 30 percent. Notice that at 30 percent nonzeros (of a total of 1500 entries), we cannot consider the signal to be truly sparse. Indeed, if K is taken to be larger than the spark, the algorithm will blow up as the third graph illustrates.

Finally we present a simple example of image denoising. Assume we are given a noisy image $\tilde{x}$. We would like to denoise the image, or recover something closer to the true noiseless image x. We assume the image can be sparsely represented in the wavelet domain. Below we take two hand made images with just five percent nonzeros in the wavelet representation. We then add 40 percent random Gaussian noise to each image to produce $\tilde{x}$. Given y in the wavelet domain, we look to optimize $||Fy-\tilde{x}||_2^2+2\tau||y||_1$ with FISTA and with FIVTA, which optimizes $||Fy-\tilde{x}||_2^2+h_{\bar{\rho},\tau}(y)$ provided that $\bar{\rho} = \lim_{n\to\infty}\rho_n$ exists. The map F is the inverse map of a redundant wavelet basis using the 'db3' family. Having obtained y we plot Fy as a denoised image. Since in this test we know the noiseless image x we pick a τ that results in the minimum norm error between Fy and x for each algorithm. We show that after 1000 iterations at this best τ, FIVTA denoises the image better than FISTA, a possible application of the algorithm.

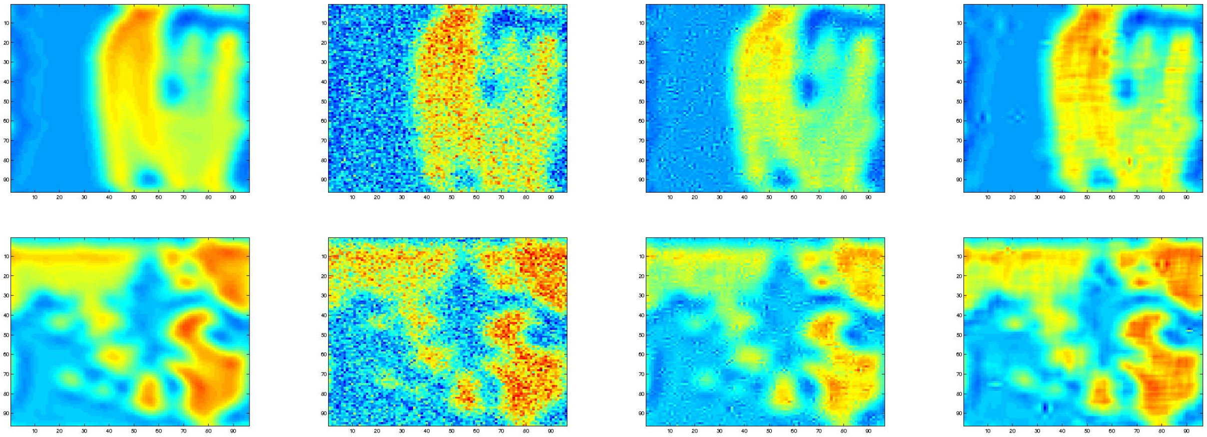


Figure 3.7: Experiments 5. On each row: original image (96x96), noisy image, denoised image with FISTA, denoised image with FIVTA, each using 1000 iterations. We observe that the image produced by FIVTA has less noise and is closer to the original.

3.6 Chapter Remarks and Conclusions

We have presented an iterative algorithm, IVTA, which uses a different type of thresholding and its corresponding fast version called FIVTA. The algorithm was designed with the idea of lowering the residual term $||Ax - b||_2$ faster than with soft thresholding. For a similar value of τ as compared to soft thresholding, the algorithm produces a solution of comparable sparsity but with a lower residual value. For this reason, in the case that the noise variance is known, the convergence of the algorithm is substantially faster than that of FISTA since the desired solution would be computed at a higher value of τ and convergence of soft thresholding methods, including FISTA, is known to be slow for small values of τ. We have exhibited some numerical experiments to show this is the case. In particular FIVTA seems better to use when the right thresholding parameter is not known. We show more numerical experiments in Chapter 5.

Chapter 4

TWO NEW ITERATIVELY REWEIGHTED LEAST SQUARES ALGORITHMS FOR THE MINIMIZATION OF A GENERALIZED SPARSITY PROMOTING FUNCTIONAL

4.1 Overview

In Chapter 2, we briefly discussed how we can come up with sparse regularization algorithms by replacing the non-smooth portion of the sparsity promoting functional by a smooth approximation. The technique was to convolve the non-smooth function with a narrow Gaussian, which works, but does not produce a particularly accurate algorithm. In this chapter, we discuss two algorithms based on a different smoothing approach, which approximates the one-norm in terms of a reweighted two norm $||x||_{2,w} = \sum_{k=1}^{N} w_k x_k^2$, which is smooth. The approximation can be made by noting that:

$$|x_k| = \frac{x_k^2}{|x_k|} = \frac{x_k^2}{\sqrt{x_k^2}} \approx \frac{x_k^2}{\sqrt{x_k^2 + \epsilon^2}}$$

where in the rightmost term, a small $\epsilon \neq 0$ is used, to insure the denominator is finite, regardless of the value of x_k. Thus, given an estimate of the signal x at the n-th iteration, x^n, an approximation to the ℓ_1 takes the form:

$$||x||_1 \approx \sum_{k=1}^{N} \frac{x_k^2}{\sqrt{(x_k^n)^2 + \epsilon_n^2}}$$

where the right hand side is a reweighted two-norm with weights $w_k^n = \frac{1}{\sqrt{(x_k^n)^2+\epsilon_n^2}}$ where $\epsilon_n \to 0$ as $n \to \infty$. The approximation is thus expected to get more accurate as the iterations progress. The key to the methods described in this chapter is to choose the parameters in a way that leads to good numerical performance and allows for convergence analysis without placing strict conditions on the properties of the matrix, other than its spectral norm. In this chapter, we discuss two versions of an Iteratively Reweighted Least Squares (IRLS) method, designed for the minimization of the more general functional:

$$F(x) = ||Ax - b||_2^2 + 2\sum_{k=1}^{N} \lambda_k |x_k|^{q_k},$$

where $A \in \mathbb{R}^{m \times N}$, $b \in \mathbb{R}^m$, and for $k = 1, \dots, N$, $\lambda_k \geq 0$ and $1 \leq q_k < 2$. We argue that this more general functional, which is capable of penalizing different parts of the solution vector in different ways, is well suited for many applications, especially those involving structured sparsity via a transform. In our numerical experiments, the new methods offer similar performance to the popular FISTA algorithm for $q_k = 1$, but are more general and better applicable to problems requiring mixed norm regularization.

The first algorithm, IRLS, can be implemented as a simple iterative scheme of the form:

$$x_k^{n+1} = \frac{1}{1 + q_k \lambda_k w_k^n} (x_k^n + (A^T b)_k - (A^T A x^n)_k),$$

with certain weights w_k^n. The second algorithm, called IRLS SYS, where the "SYS" refers to the fact that it involves a linear system each step, can be implemented as:

$$(A^T A + \Phi^n) x^{n+1} = A^T b, \tag{4.1.1}$$

where Φ^n is a diagonal matrix containing the product of iteratively adapting weights w_k^n and the constant $q_k \lambda_k$ on its diagonal. Because the only difference in the linear system between steps $n+1$ and n consists in slightly changing the diagonal terms in Φ^n to obtain Φ^{n+1}, at different iterations we are to solve linear systems that differ only slightly from one another. The linear system (4.1.1) can be solved, for example, using a conjugate gradient scheme. As the initial guess we can reuse the previous solution, to speed up convergence. In later stages of the algorithm, where the solution changes at a slower rate than at the beginning, typically fewer inner conjugate gradient iterations are needed. Hence, the scheme is found to be efficient, even though a few iterations of a solver, such as a conjugate gradient method, are typically needed at every step of the algorithm.

The algorithms presented here are shown to converge for general matrices A with spectral norm $||A||_2$ less than one. In fact, when in the generalized functional all $q_k > 1$, we show that the iterates themselves converge to the unique minimizer.

4.2 Generalized Sparsity Promoting Functional

We first discuss the motivation for using the functional:

$$F(x) := ||Ax - b||_2^2 + 2\sum_{k=1}^{N} \lambda_k |x_k|^{q_k}, \tag{4.2.1}$$

where $A \in \mathbb{R}^{m \times N}$, $b \in \mathbb{R}^m$, and for $k = 1, \ldots, N$, $\lambda_k \geq 0$ and $1 \leq q_k < 2$. Then we derive the optimality conditions and state a uniqueness theorem. As is easily seen the popular ℓ_1-functional: $||Ax - b||_2^2 + 2\lambda||x||_1$ is a special case of the above. However, we can get sparse solutions for any q_k below 2, except that for larger q_k the solutions tend to be less sparse. In general, the more general functional is advantageous for a few different reasons. First of all, the sparseness of a solution tends to be the result of a transform through which it is expressed. That is to say, a sparse solution may not necessarily exist in the default, Euclidean basis, but may exist under a suitable transformation. Such is the case for example for images and wavelets. Many images can be sparsely represented under a wavelet transform. A wavelet transform, however, gives structure to the data. The transformed vector consists of several different parts: in addition to wavelet functions at different scales, it also contains a superposition of coarse scaling functions. There is no need to treat all of these parts in the same way. Often, for example, we may want to keep most of the scaling coefficients instead of setting them to zero. On the other hand, there may be many fine wavelet coefficients, many of which are very small and can be safely set to zero, since we typically expect the wavelet part to be sparse. This kind of operation can be accomplished with the above functional., e.g. by setting q_k close to 2 for the scaling coefficients, but $q_k = 1$ for wavelets.

There are other applications of the above functional, where penalizing components differently may be useful. First is the case when a complicated coordinate system is used. The vector x may inherently represent a multi-dimensional structure, such as the cubed sphere geometry, discussed in a later chapter. Within this structure different parts (such as different chunks in the cubed sphere) may need to be treated differently. These different parts correspond to different coefficients sets of x and this different treatment is most easily accomplished with the above functional. Finally, in applications we may choose to look for a solution that is represented sparsely by using a few different bases. That is, instead of finding some solution w such that $x = W^{-1}w$ somehow satisfies the data, we may instead look for a solution that is represented as: $x = \alpha_1 W_1^{-1} w_1 + \cdots + \alpha_N W_N^{-1} w_N$. In this application, we typically use a combined matrix $\left(AW_1^{-1}, \ldots, AW_N^{-1}\right)$ and a combined vector of the form $(w_1, \ldots, w_N)$. It is clear that different parts of this vector would demand separate treatment depending on how much we would like to weigh contributions of different bases to have.

Below we present some results on the above functional. We start by deriving the optimality conditions for the functional.

Lemma 4.2.1. *The conditions for the minimizer of the functional* $F(x) = ||Ax - b||_2^2 + 2\sum_{k=1}^{N} \lambda_k |x_k|^{q_k}$ *are:*

$$\begin{aligned} (A^T(b - Ax))_k &= \lambda_k \operatorname{sgn}(x_k) q_k |x_k|^{q_k - 1}\,, & x_k &\neq 0 \\ (A^T(b - Ax))_k &= 0\,, & x_k &= 0 \quad (q_k > 1) \\ \left|(A^T(b - Ax))_k\right| &\leq \lambda_k\,, & x_k &= 0 \quad (q_k = 1) \end{aligned} \tag{4.2.2}$$

Proof. For the case $1 \leq q_k \leq 2$, $F(x)$ is convex, so any local minimizer is necessarily global and to characterize the minimizer it is necessary only to work out the conditions corresponding to $F(x) \leq F(x + tz)$ for all $t \in \mathbb{R}$ and all $z \in \mathbb{R}^N$. $F(x) \leq F(x + tz)$

implies that

$$t^2||Az||^2 + 2t\langle z, A^T(Ax-b)\rangle + 2\sum_{k=1}^{N} \lambda_k \left(|x_k + tz_k|^{q_k} - |x_k|^{q_k}\right) \geq 0. \tag{4.2.3}$$

We shall derive N conditions, one for each index $k \in \{1, \ldots, N\}$. To get the k-th condition, we consider z of the special form $z = z_k e_k$ (i.e. all entries of z are 0, except for the k-th entry). At this point, we have to consider two different cases: $x_k \neq 0$ and $x_k = 0$. When $x_k \neq 0$, we expand $f(t) = |x_k + tz_k|^{q_k}$ in Taylor series around 0 to get $f(t) = f(0) + tf'(0) + O(t^2)$. When t is small, $\operatorname{sgn}(x_k + tz_k) = \operatorname{sgn}(x_k)$ if $x_k \neq 0$. For $x_k > 0$, we have $|x_k + tz_k| = x_k + tz_k$, so $f(t) = (x_k + tz_k)^{q_k}$ and:

$$f'(t) = q_k z_k (x_k + tz_k)^{q_k - 1} = \operatorname{sgn}(x_k) q_k z_k |x_k + tz_k|^{q_k - 1}.$$

For $x_k < 0$, $|x_k + tz_k| = -(x_k + tz_k)$, so $f(t) = |x_k + tz_k|^{q_k} = (-x_k - tz_k)^{q_k}$ and:

$$f'(t) = q_k(-z_k)(-x_k - tz_k)^{q_k - 1} = \operatorname{sgn}(x_k) q_k z_k |x_k + tz_k|^{q_k - 1}.$$

Thus, for both signs of x_k: $f'(t) = \operatorname{sgn}(x_k) q_k z_k |x_k + tz_k|^{q_k - 1}$, implying that $f'(0) = \operatorname{sgn}(x_k) q_k z_k |x_k|^{q_k - 1}$. So for some constant $C > 0$:

$$\begin{aligned} f(t) &= |x_k + tz_k|^{q_k} = |x_k|^{q_k} + t \operatorname{sgn}(x_k) q_k z_k |x_k|^{q_k - 1} + O(t^2) \\ &\leq |x_k|^q + t \operatorname{sgn}(x_k) q_k z_k |x_k|^{q_k - 1} + Ct^2, \end{aligned}$$

and, recalling that $z = z_k e_k$, (4.2.3) implies that

$$\begin{aligned} & t^2||A(z_k e_k)||^2 + 2t\langle z_k e_k, A^T(Ax-b)\rangle + 2\lambda_k \left(t \operatorname{sgn}(x_k) q_k z_k |x_k|^{q_k - 1} + Ct^2\right) \geq 0 \\ \Longrightarrow \quad & t^2 \left(||A(z_k e_k)||^2 + C\lambda_k\right) + 2t \left(z_k (A^T(Ax-b))_k + \lambda_k \operatorname{sgn}(x_k) q_k z_k |x_k|^{q_k - 1}\right) \geq 0. \end{aligned}$$

The first term can be made arbitrary small with respect to the second term, so as this holds for all t this implies:

$$z_k(A^T(Ax-b))_k + \lambda_k \operatorname{sgn}(x_k) q_k z_k |x_k|^{q_k-1} = 0,$$

which leads to:

$$(A^T(b-Ax))_k = \lambda_k \operatorname{sgn}(x_k) q_k |x_k|^{q_k-1}\,,\ x_k \neq 0.$$

Note that when $q_k = 1$ we recover the familiar condition for ℓ_1 minimization:

$$(A^T(b-Ax))_k = \lambda_k \operatorname{sgn}(x_k)\,,\ x_k \neq 0.$$

When $x_k = 0$, recalling that $z = z_k e_k$, (4.2.3) gives:

$$t^2||A(z_k e_k)||^2 + 2t\langle z_k e_k, A^T(Ax-b)\rangle + 2\lambda_k |t|^{q_k}|z_k|^{q_k} \geq 0. \tag{4.2.4}$$

Making the substitutions $t^2 = |t|^2$, $t = |t| \operatorname{sgn}(t)$, we obtain

$$|t|^2||A(z_k e_k)||^2 + |t|\left(2\operatorname{sgn}(t) z_k (A^T(Ax-b))_k + 2\lambda_k |t|^{q_k-1}|z_k|^{q_k}\right) \geq 0. \tag{4.2.5}$$

In this case, we have to consider the case $q_k = 1$ and $q_k > 1$ separately. When $q_k > 1$ we have that:

$$|t|^2||Az||^2 + 2\lambda_k |t|^{q_k}|z_k|^{q_k} + 2|t| \operatorname{sgn}(t) z_k (A^T(Ax-b))_k \geq 0.$$

Since $q_k > 1$, the first two terms on the left have greater powers of $|t|$ than the last term and can be made arbitrary small by picking t small enough. This means we must have:

$$2\,\mathrm{sgn}(t) z_k (A^T(Ax-b))_k \geq 0$$

for all t, which can be true only if $(A^T(Ax-b))_k = 0$. Thus, we conclude that the condition is :

$$(A^T(b-Ax))_k = 0\,,\ x_k = 0 \quad (q_k > 1).$$

For $q_k = 1$ we have from (3) that:

$$\begin{aligned} & |t|^2||Az||^2 + |t|\left(2\,\mathrm{sgn}(t) z_k (A^T(Ax-b))_k + 2\lambda_k |z_k|\right) && \geq 0 \\ \Longrightarrow \qquad & \mathrm{sgn}(t) z_k (A^T(Ax-b))_k + \lambda_k |z_k| && \geq 0. \end{aligned}$$

Now consider the two cases: where t and z_k have the same sign: $\mathrm{sgn}(t) = \mathrm{sgn}(z_k)$ or opposite signs: $\mathrm{sgn}(t) = -\,\mathrm{sgn}(z_k)$. Then we have, respectively:

$$(A^T(Ax-b))_k + \lambda_k \geq 0 \quad \text{and} \quad -\,(A^T(Ax-b))_k + \lambda_k \geq 0,$$

so we obtain the condition :

$$\left|(A^T(b-Ax))_k\right| \leq \lambda_k\,,\ x_k = 0 \quad (q_k = 1).$$

Thus, we can summarize the component-wise conditions for the minimizer of $F(x)$ as in (4.2.2). □

The functional $F(x)$ defined in (4.2.1) may not have a unique minimizer. However, we can prove the lemma given below, which effectively says that any two minimizers would have the same degree of sparseness and fit:

Lemma 4.2.2. *Let u and v both be minimizers of the functional $F(x)$ defined in (4.2.1). Then we have: $||Au-b||_2 = ||Av-b||_2$ and $\sum_{k=1}^{N} \lambda_k |u_k|^{q_k} = \sum_{k=1}^{N} \lambda_k |v_k|^{q_k}$.*

Additionally, if we have that all $q_k > 1$ for all k, then we have uniqueness (i.e. $u = v$).

Proof. Let $p(x) = ||Ax - b||_2^2$ and $q(x) = 2\sum_{k=1}^{N} \lambda_k |x_k|^{q_k}$, so $F(x) = p(x) + q(x)$. Both $p(x)$ and $q(x)$ are convex functions. This means that:

$$p(tu + (1-t)v) \leq tp(u) + (1-t)p(v) \quad \text{and} \quad q(tu + (1-t)v) \leq tq(u) + (1-t)q(v)$$

for $t \in [0,1]$. We have:

$$F(u) = ||Au - b||_2^2 + 2\sum_{k=1}^{N} \lambda_k |u_k|^{q_k} = ||Av - b||_2^2 + 2\sum_{k=1}^{N} \lambda_k |v_k|^{q_k} = F(v),$$

and

$$\begin{aligned} F\left(\frac{u+v}{2}\right) &= p\left(\frac{u+v}{2}\right) + q\left(\frac{u+v}{2}\right) \leq \frac{1}{2}\left(p(u) + p(v)\right) + \frac{1}{2}\left(q(u) + q(v)\right) \\ &= \frac{1}{2}\left(F(u) + F(v)\right) \leq \frac{1}{2}\left(F\left(\frac{u+v}{2}\right) + F\left(\frac{u+v}{2}\right)\right) = F\left(\frac{u+v}{2}\right), \end{aligned}$$

where the first inequality follows by convexity and the second inequality follows because u and v are both minimizers of F. It follows that we can replace all the inequalities above by equalities. Hence, in particular, we have that:

$$p(u) + p(v) = 2p\left(\frac{u+v}{2}\right) \implies ||Au - b||_2^2 + ||Av - b||_2^2 = 2\left\|A\left(\frac{u+v}{2}\right) - b\right\|_2^2.$$

Expanding both sides we have:

$$\begin{aligned}
&||Au||^2 + ||Av||^2 - 2\langle Au, b\rangle + ||b||^2 - 2\langle Av, b\rangle + ||b||^2 \\
&= 2\left(\frac{1}{4}||A(u+v)||^2 - 2\left\langle A\left(\frac{u+v}{2}\right), b\right\rangle + ||b||^2\right) \\
&\implies ||Au||^2 + ||Av||^2 = \frac{1}{2}||A(u+v)||^2 \\
&\implies \frac{1}{2}||Au||^2 + \frac{1}{2}||Av||^2 - \langle Au, Av\rangle = 0 \\
&\implies \frac{1}{2}||A(u-v)||_2^2 = 0.
\end{aligned}$$

Thus, $||A(u-v)|| = 0 \implies Au = Av \implies ||Au-b||^2 = ||Av-b||^2 \implies p(u) = p(v)$. Since we have that $F(u) = F(v)$, we must also have $q(u) = q(v)$.

In the special case where $q_k > 1$ for all k, the functional $F(x)$ is strictly convex, and has thus a unique global minimizer. □

4.3 Connection to Previous Work

In this section, we state the connection between the work in this paper and previous work in [12]. The algorithm in [12] dealt with the constrained case (without noise), i.e. it sought to minimize, using an IRLS approach, $||x||_1$ among all x satisfying $Ax = b$,and can be motivated by the two lemmas below. The algorithm considered the minimization problem in terms of the reweighted ℓ_2 norm $||x||_{2,w} = \sum_{k=1}^{N} w_k x_k^2$.

$$[\min ||x||_1 \quad \text{s.t.} \quad Ax = b] \to \left[\min_x ||Dx||_2^2 \quad \text{s.t.} \quad Ax = b\right]$$

where $||Dx||_2^2 = \sum_{k=1}^{N} D_{k,k}^2 x_k^2$ and we have that $D^T D = D^2 = W^{-1}$ where W is the weight matrix with the weights on the diagonals. We now derive (informally) the form of the solution. Rewriting the problem above in terms of Lagrange multipliers we have:

$$\begin{aligned}
& \hat{x} = \arg\min_x \left(||Dx||_2^2 + y^T(Ax - b) \right) \\
\implies \quad & 2D^T D\hat{x} + A^T y = 0 \\
\implies \quad & \hat{x} = -\frac{1}{2}(D^T D)^{-1} A^T y \\
\implies \quad & A\hat{x} = -\frac{1}{2} A(D^T D)^{-1} A^T y = b \\
\implies \quad & y = -2 \left(A(D^T D)^{-1} A^T \right)^{-1} b.
\end{aligned}$$

Plugging in the expression for y in the expression for x we obtain:

$$\hat{x} = (D^T D)^{-1} A^T \left(A(D^T D)^{-1} A^T \right)^{-1} b = W A^T \left(A W A^T \right)^{-1} b,$$

which corresponds to ([12], equation 1.9). We note that the matrix $D^T D$ above is invertible, since the diagonal weight matrix W has no zero weights. The above linear system can then be solved by a method such as the conjugate gradient algorithm.

The algorithm in [12] can be motivated by the two lemmas below, which show some degree of equivalence between minimizing the one-norm and solving a weighted ℓ_2 problem, at least in the case when none of the components vanish. We begin with a lemma that characterizes an ℓ_1 minimizer:

Lemma 4.3.1. *Consider all x such that $Ax = b$; we denote this set $\theta(b)$. An element x of $\theta(b)$ has minimal ℓ_1-norm among all elements $z \in \Phi(b)$ if and only if for all d in null space of A, we have:*

$$\left|\sum_{x_i \neq 0} \operatorname{sgn}(x_i) d_i\right| \leq \sum_{x_i = 0} |d_i|$$

Proof. Consider first the case when $x \in \theta(b)$ has minimal ℓ_1-norm within $\theta(b)$. This means that for any d in the null space of A and $t \in \mathbb{R}$, we have: $||x||_1 \leq ||x + td||_1$. (Note that $x + td$ is in $\theta(b)$ since $A(x + td) = b + 0 = b$ because $Ad = 0$.) Thus:

$$\sum_{i=1}^{N} |x_i + td_i| \geq \sum_{i=1}^{N} |x_i| = \sum_{x_i \neq 0} \operatorname{sgn}(x_i) x_i.$$

Note now that, for t sufficiently small, we have $\sum_{i=1}^{N} |x_i + td_i| = \sum_{x_i \neq 0} (x_i + td_i) \operatorname{sgn}(x_i + td_i) + \sum_{x_i = 0} |td_i|$. For small t, we also have $\operatorname{sgn}(x_i + td_i) = \operatorname{sgn}(x_i)$ whenever $x_i \neq 0$. Thus, we obtain:

$$\sum_{x_i \neq 0} \operatorname{sgn}(x_i) x_i + \sum_{x_i \neq 0} \operatorname{sgn}(x_i) td_i + \sum_{x_i = 0} |td_i| \geq \sum_{x_i \neq 0} \operatorname{sgn}(x_i) x_i.$$

Thus:

$$\sum_{x_i \neq 0} \operatorname{sgn}(x_i) td_i + \sum_{x_i = 0} |td_i| \geq 0.$$

For $t \neq 0$, dividing by $|t|$ we get:

$$\sum_{x_i \neq 0} \operatorname{sgn}(x_i) \operatorname{sgn}(t) d_i + \sum_{x_i = 0} |d_i| \geq 0.$$

Taking $t > 0$ and then $t < 0$ we have:

$$\begin{array}{lc} & -\sum_{x_i \neq 0} \operatorname{sgn}(x_i) d_i + \sum_{x_i = 0} |d_i| \geq 0 \quad \text{and} \quad \sum_{x_i \neq 0} \operatorname{sgn}(x_i) d_i + \sum_{x_i = 0} |d_i| \geq 0 \\ \Longrightarrow & \sum_{x_i = 0} |d_i| \geq \left| \sum_{x_i \neq 0} \operatorname{sgn}(x_i) d_i \right|. \end{array}$$

Now we show that $\sum_{x_i = 0} |d_i| \geq |\sum_{x_i \neq 0} \operatorname{sgn}(x_i) d_i|$ implies that $||x + td||_1 \geq ||x||_1$ for arbitrary $d \in \ker(A)$ and t:

$$||x||_1 = \sum_{i=1}^{N} |x_i| = \sum_{x_i \neq 0} \operatorname{sgn}(x_i) x_i = \sum_{x_i \neq 0} \operatorname{sgn}(x_i)(x_i + td_i) - \sum_{x_i \neq 0} \operatorname{sgn}(x_i) td_i.$$

Next:

$$\begin{aligned} \sum_{x_i \neq 0} \operatorname{sgn}(x_i)(x_i + td_i) - t \sum_{x_i \neq 0} \operatorname{sgn}(x_i) d_i &\leq \sum_{x_i \neq 0} \operatorname{sgn}(x_i)(x_i + td_i) + |t| \left| \sum_{x_i \neq 0} \operatorname{sgn}(x_i) d_i \right| \\ &\leq \sum_{x_i \neq 0} \operatorname{sgn}(x_i)(x_i + td_i) + |t| \sum_{x_i \neq 0} |d_i|, \end{aligned}$$

where the last inequality follows by the assumption. Since $\operatorname{sgn}(x_i)(x_i + td_i) \leq |x_i + td_i|$, regardless of the relative sizes of x_i, t, d_i, we have thus:

$$||x||_1 \leq \sum_{x_i \neq 0} |x_i + td_i| + \sum_{x_i = 0} |td_i| = ||x + td||_1.$$

□

Next, we show the connection between ℓ_1-minimization and weighted-ℓ_2 minimization through the following lemma:

Lemma 4.3.2. *Take $A \in \mathbb{R}^{m\times N}$, $b \in \mathbb{R}^m$, and suppose $x \in \mathbb{R}^N$ satisfies $Ax = b$ and $x_i \neq 0$ for all i. Then the following are equivalent:*

(1) $\quad$ *x solves*

$$\min ||z||_1 \text{ s.t. } Az = b. \tag{4.3.1}$$

(2) $\quad$ *For any $d \in \ker(A)$, $\displaystyle\sum_{i=1}^{N} d_i \operatorname{sgn}(x_i) = 0$.*

(3) $\quad$ *x solves $\min ||z||_{2,w}$ among all z such that $Az = b$ where $w_i = \frac{1}{|x_i|}$.*

Proof. First, we prove (1) $\implies$ (2). Pick $d \in \ker(A)$. Then for any $t \in \mathbb{R}$, $td \in \ker(A)$. By Lemma 4.3.1, (1) $\implies$ $\left|\displaystyle\sum_{i=1}^{N} td_i \operatorname{sgn}(x_i)\right| \leq 0$, since $x_i \neq 0$ for all i. This implies $\displaystyle\sum_{i=1}^{N} d_i \operatorname{sgn}(x_i) = 0$ since the above holds for both positive and negative t.

Next, we prove (2) $\implies$ (1). Suppose $\displaystyle\sum_{i=1}^{N} d_i \operatorname{sgn}(x_i) = 0$ for $d \in \ker(A)$. We want to show that $||x + td||_1 \geq ||x||_1$ for all t:

$$\begin{aligned} ||x + td||_1 &= \sum_{i=1}^{N} |x_i + td_i| \geq \sum_{i=1}^{N} (x_i + td_i) \operatorname{sgn}(x_i) \\ &= \sum_{i=1}^{N} x_i \operatorname{sgn}(x_i) + t \sum_{i=1}^{N} d_i \operatorname{sgn}(x_i) = ||x||_1 + 0 = ||x||_1 \end{aligned}$$

by the assumption. Hence, it follows that $||x + td||_1 \geq ||x||_1$.

Next, we prove (2) $\implies$ (3). Suppose $\sum\limits_{x_i \neq 0} d_i \operatorname{sgn}(x_i) = 0$ for $d \in \ker(A)$. We want to show that $||x + td||_{2,w}^2 \geq ||x||_{2,w}^2$. We have:

$$\begin{aligned} ||x + td||_{2,w}^2 &= ||x||_{2,w}^2 + t^2||d||_{2,w}^2 + 2t\sum_{i=1}^{N} \frac{x_i}{|x_i|} d_i \\ &= ||x||_{2,w}^2 + t^2||d||_{2,w}^2 + 2\sum_{i=1}^{N} \operatorname{sgn}(x_i) d_i \\ &= ||x||_{2,w}^2 + t^2||d||_{2,w}^2 \ \geq \ ||x||_{2,w}^2. \end{aligned}$$

Finally, we prove (3) $\implies$ (2). Suppose for $d \in \ker(A)$ and for $t \in \mathbb{R}$, we have: $||x + td||_{2,w}^2 \geq ||x||_{2,w}^2$. Expanding we have:

$$\begin{aligned} & ||x||_{2,w}^2 \leq ||x + td||_{2,w}^2 = ||x||_{2,w}^2 + t^2||d||_{2,w}^2 + 2t\sum_{i=1}^{N} \operatorname{sgn}(x_i) d_i \\ \implies \quad & 2t\sum_{i=1}^{N} \operatorname{sgn}(x_i) d_i + t^2||d||_{2,w}^2 \geq 0 \implies 2t\sum_{i=1}^{N} \operatorname{sgn}(x_i) d_i \geq 0 \text{ for all } t \\ \implies \quad & \sum_{i=1}^{N} \operatorname{sgn}(x_i) d_i = 0. \end{aligned}$$

□

Note that the equivalence between item (1) and item (3) in Lemma 4.3.2 cannot be used directly to design an algorithm to find the solution to (4.3.1). For, since the solution to (4.3.1) is unknown, we cannot directly make use of (3), which involves the solution of (1) to determine the weights. However, we can iterate on the weights, updating them as an approximate solution to (3) is improved, starting from some initial guess. In this way an iterative algorithm can be posed as is done in [12]. In the same spirit as above we can state the following lemma, which extends the above result to our generalized functional, which is unconstrained but contains a penalization term:

Lemma 4.3.3. *Take $A \in \mathbb{R}^{m\times N}$, $b \in \mathbb{R}^m$, $\lambda_k \geq 0$ for $k = 1, \dots, N$, $1 \leq q_k < 2$ for $k = 1, \dots, N$, and consider $x \in \mathbb{R}^N$ such that $x_k \neq 0$ for all k. Then the following are equivalent:*

(1) $\quad x$ *solves* $\min\limits_z ||Az - b||_2^2 + 2\sum\limits_{k=1}^{N} \lambda_k |z_k|^{q_k}.$

(2) $\quad \langle Ax - b, Ad\rangle + \sum\limits_{k=1}^{N} \lambda_k q_k d_k \operatorname{sgn}(x_k)|x_k|^{q_k-1} = 0$ *for every* $d \in \mathbb{R}^N$.

(3) $\quad x$ *solves* $\min_z ||Az - b||_2^2 + \sum\limits_{k=1}^{N} \lambda_k w_k z_k^2$ *with weights* $w_k = q_k |x_k|^{q_k-2}$.

Proof. We first show that (1) $\implies$ (2) and vice-versa. Since for $1 < q_k \leq 2$, the functional: $F(z) = ||Az - b||_2^2 + 2\sum\limits_{k=1}^{N} \lambda_k |z_k|^{q_k}$ is convex, each minimizer x at which F is differentiable, satisfies $(\nabla F(x))_k = 0$. Differentiating $F(z)$ with respect to z, for z with $z_k \neq 0$ for all k:

$$(\nabla F(z))_k = (A^T(Az - b))_k + 2\lambda_k q_k \operatorname{sgn}(z_k)|z_k|^{q_k-1}.$$

It follows that F is differentiable at x, and $(\nabla F(x))_k = 0$ implies

$$(A^T(Ax - b))_k + 2\lambda_k q_k \operatorname{sgn}(x_k)|x_k|^{q_k-1} = 0. \tag{4.3.2}$$

Since x minimizes F, (4.3.2) holds for all k, and in particular for any $d \in \mathbb{R}^N$ we have:

$$\langle Ax - b, Ad\rangle + 2\sum_{k=1}^{N} \lambda_k q_k d_k \operatorname{sgn}(x_k)|x_k|^{q_k-1} = 0. \tag{4.3.3}$$

Conversely, if (4.3.3) holds for all $d \in \mathbb{R}^N$, then for $k = 1, \dots, N$, by taking $d = e_k$ (4.3.2) holds. Also, since $x_k \neq 0$ for all k, F is differentiable at x and $\nabla F(x) = 0$. Then x minimizes the convex function F.

To show the equivalence between (1) and (3) consider the functions:

$$F(z) = ||Az - b||_2^2 + 2\sum_{k=1}^{N} \lambda_k |z_k|^{q_k} \quad \text{and} \quad G(z) = ||Az - b||_2^2 + 2\sum_{k=1}^{N} \lambda_k w_k z_k^2,$$

with $w_k = q_k |x_k|^{q_k - 2}$. Both F and G are convex. Since we assume that $x_k \neq 0$ for all k, both are differentiable at x. In fact, we have that, for z such that $z_k \neq 0$ for all k,

$$\begin{aligned} \frac{1}{2}\frac{\partial}{\partial z_k} F(z) &= A^T(Az - b) + \lambda_k q_k \operatorname{sgn}(z_k)|z_k|^{q_k - 1} \text{ and} \\ \frac{1}{2}\frac{\partial}{\partial z_k} \nabla G(z) &= A^T(Az - b) + \lambda_k w_k z_k. \end{aligned}$$

Plugging in $z = x$ and $w_k = q_k |x_k|^{q_k - 2}$ we see that both expressions above evaluate to $A^T(Ax - b) + \lambda_k q_k \operatorname{sgn}(x_k)|x_k|^{q_k - 1}$ since $w_k x_k = q_k |x_k|^{q_k - 2} |x_k| \operatorname{sgn}(x_k) = q_k \operatorname{sgn}(x_k)|x_k|^{q_k - 1}$. If (1) holds then we must have that $\nabla F(x) = 0$. But this means that $\nabla G(x) = 0$ as well, since $\nabla F(x) = \nabla G(x)$. Therefore, x solves (1) if and only if it solves (3) and the lemma is established. □

Lemma 4.3.3 shows that, at least in principle, a reweighted two-norm algorithm for the minimization of our functional is possible. The methods and verification of convergence of the algorithms are the topic of the following two subsections.

4.4 IRLS Algorithm

We now present analysis of the first IRLS algorithm. Given an initial guess x^0, an initial parameter $\epsilon^0 = 1$, the matrix A with $||A||_2 < 1$, the (possibly noisy) right hand side b, $1 \leq q_k < 2$ and $\lambda_k \geq 0$ for $k = 1, \ldots, N$, and fixed parameters $0 < \gamma < 1$ and $0 < \alpha < 1$, the iterative scheme is given by:

$$\begin{aligned} w_k^n &= \frac{1}{\left((x_k^n)^2 + (\epsilon^n)^2\right)^{\frac{2-q_k}{2}}} \\ x_k^{n+1} &= \frac{1}{1 + q_k \lambda_k w_k^n} \left(x_k^n + (A^T b)_k - (A^T A x^n)_k\right) \qquad \text{(IRLS)} \\ \epsilon^{n+1} &= \min(\epsilon^n, ||x^{n+1} - x^n||^\gamma + \alpha^{n+1}) \quad ; \quad 0 < \gamma < 1 \; , \; 0 < \alpha < 1. \end{aligned}$$

The main convergence result is that, if the functional

$$F(x) = ||Ax - b||_2^2 + 2 \sum_{k=1}^{N} \lambda_k |x_k|^{q_k} \tag{4.4.1}$$

has a unique minimizer (this is the case if $q_k > 1$ for all k, or if $\ker(A) = \{0\}$, but it can also be true, generically, even when these conditions are not met), then the iterative algorithm converges to this minimizer. More generally, all accumulation points of the sequence $(x^n)_n$ are minimizers for the functional. Note that the traditional ℓ_1 functional is a special case of the functional $F(x)$.

We show that the iterates x^n are bounded and that there exist converging subsequences, the limit of which satisfies the optimality conditions for the minimization of $F(x)$.

Lemma 4.4.1. *Define the surrogate functional:*

$$\begin{aligned} G(x, a, w, \epsilon) &= ||Ax - b||_2^2 - ||A(x - a)||_2^2 + ||x - a||_2^2 \\ &+ \sum_{k=1}^{N} \lambda_k \left(q_k w_k ((x_k)^2 + \epsilon^2) + (2 - q_k)(w_k)^{\frac{q_k}{q_k - 2}}\right). \end{aligned}$$

Then the minimization procedure:

$$w^{n+1} = \arg\min_{w} G(x^{n+1}, a, w, \epsilon^{n+1})$$

leads to

$$w_k^{n+1} = \frac{1}{((x_k^{n+1})^2 + (\epsilon^{n+1})^2)^{\frac{2-q_k}{2}}}. \tag{4.4.2}$$

In addition, the minimization procedure:

$$x^{n+1} = \arg\min_{x} G(x, x^n, w^n, \epsilon^n)$$

produces the iterative scheme:

$$(x^{n+1})_k = \frac{1}{1 + q_k \lambda_k (w^n)_k} \left((x^n)_k - (A^T A x^n)_k + (A^T b)_k \right). \tag{4.4.3}$$

Proof. First, note the derivation of the weights via: $w^{n+1} = \arg\min_w G(x^{n+1}, a, w, \epsilon^{n+1})$, where:

$$\begin{aligned} G(x^{n+1}, a, w, \epsilon^{n+1}) &= ||Ax^{n+1} - b||_2^2 - ||A(x^{n+1} - a)||_2^2 + ||x^{n+1} - a||_2^2 \\ &\quad + \sum_{k=1}^{N} \lambda_k \left(q_k w_k ((x_k^{n+1})^2 + (\epsilon^{n+1})^2) + (2 - q_k)(w_k)^{\frac{q_k}{q_k - 2}} \right). \end{aligned}$$

Only the last term of G has a dependence on w. We have:

$$\begin{aligned} & \frac{\partial}{\partial w_k} \left(q_k w_k ((x_k^{n+1})^2 + (\epsilon^{n+1})^2) + (2 - q_k)(w_k)^{\frac{q_k}{q_k - 2}} \right) = 0 \\ \Longrightarrow \quad & q_k \left((x_k^{n+1})^2 + (\epsilon^{n+1})^2 \right) + (2 - q_k) \frac{q_k}{q_k - 2} (w_k)^{\frac{q_k}{q_k - 2} - 1} = 0 \\ \Longrightarrow \quad & w_k^{n+1} = \frac{1}{\left((x_k^{n+1})^2 + (\epsilon^{n+1})^2 \right)^{\frac{2-q_k}{2}}}. \end{aligned}$$

Next, let us check that the statement:

$$x_k^{n+1} = (\arg\min_x G(x, x^n, w^n, \epsilon^n))_k$$

recovers the iterative scheme (4.4.3). Consider:

$$\begin{aligned} G(x, x^n, w^n, \epsilon^n) = \ & ||Ax - b||_2^2 - ||A(x - x^n)||_2^2 + ||x - x^n||_2^2 \\ & + \sum_{k=1}^{N} \lambda_k \left(q_k w_k^n ((x_k)^2 + (\epsilon^n)^2) + (2 - q_k)(w_k^n)^{\frac{q_k}{q_k - 2}} \right), \end{aligned} \tag{4.4.4}$$

and differentiate with respect to x, then take the k-th component and set to zero. Removing terms which do not depend on x, we can then write the above as:

$$\frac{\partial}{\partial x_k} \left(||Ax - b||_2^2 - ||A(x - x^n)||_2^2 + ||x - x^n||_2^2 + \sum_{l=1}^{N} q_l \lambda_l w_l^n x_l^2 \right) = 0$$

and the result is:

$$-2(A^T b)_k + 2(A^T A x^n)_k + 2x_k - 2x_k^n + 2q_k \lambda_k w_k^n x_k = 0.$$

Then we solve for x_k and define x_k^{n+1} to be the result:

$$\begin{aligned} x^{n+1} &= \arg\min_x G(x, x^n) \\ \Longrightarrow \quad x_k(1 + q_k \lambda_k w_k) &= x_k^n + (A^T b)_k - (A^T A x^n)_k \\ \Longrightarrow \quad x_k^{n+1} &= \frac{1}{1 + q_k \lambda_k w_k} (x^n + A^T b - A^T A x^n)_k. \end{aligned}$$

□

Remark 4.4.2. *Assume that as $n \to \infty$, $x^n \to x$ and $\epsilon^n \to 0$. Notice that with the weights in* (4.4.2)*, we have that:*

$$w_k^n(x_k^n)^2 = \frac{(x_k^n)^2}{((x_k^n)^2 + (\epsilon^n)^2)^{\frac{2-q_k}{2}}} \to (x_k)^{q_k} \quad as \quad k \to \infty.$$

From (4.4.4)*, we have*

$$G(x^n, x^n, w^n, \epsilon^n) \quad = \quad ||Ax^n - b||_2^2 + 2\sum_{k=1}^{N} \lambda_k((x_k^n)^2 + (\epsilon^n)^2)^{\frac{q_k}{2}},$$

since:

$$\begin{aligned} & q_k w_k^n \left((x_k)^2 + (\epsilon^n)^2\right) + (2 - q_k)(w_k^n)^{\frac{q_k}{q_k-2}} \\ = \quad & q_k \left((x_k^n)^2 + (\epsilon^n)^2\right)^{\left(\frac{q_k-2}{2}+\frac{2}{2}\right)} + (2 - q_k) \left((x_k^n)^2 + (\epsilon^n)^2\right)^{\left(\frac{q_k-2}{2}\frac{q_k}{q_k-2}\right)} \\ = \quad & 2\left((x_k^n)^2 + (\epsilon^n)^2\right)^{\frac{q_k}{2}}. \end{aligned}$$

As $n \to \infty$, and assuming $x^n \to x$ and $\epsilon^n \to 0$, we have that:

$$\lim_{n\to\infty} G(x^n, x^n, w^n, \epsilon^n) = ||Ax - b||_2^2 + 2\sum_{k=1}^{N} \lambda_k(x_k)^{q_k},$$

so we recover the functional we would like to minimize.

The constructions provided in Lemma 4.4.1 are used below to show the relevant properties of the IRLS algorithm.

Lemma 4.4.3. *Assume that the spectral norm $||A||_2 \leq 1$. (This can be accomplished by a simple rescaling.) Then the iterates generated by* (IRLS) *satisfy $||x^n - x^{n-1}|| \to 0$ and are bounded in norm.*

Proof. From Lemma 4.4.1:

$$\begin{aligned} x^{n+1} &= \arg\min_x G(x, x^n, w^n, \epsilon^n) \\ w^{n+1} &= \arg\min_w G(x^{n+1}, a, w, \epsilon^{n+1}). \end{aligned}$$

Using these constructions we write down a sequence of inequalities:

$$\begin{aligned} G(x^{n+1}, x^{n+1}, w^{n+1}, \epsilon^{n+1}) &\leq G(x^{n+1}, x^{n+1}, w^n, \epsilon^{n+1}) \quad [A] \\ &\leq G(x^{n+1}, x^n, w^n, \epsilon^{n+1}) \quad [B] \\ &\leq G(x^{n+1}, x^n, w^n, \epsilon^n) \quad [C] \\ &\leq G(x^n, x^n, w^n, \epsilon^n). \quad [D] \end{aligned}$$

We now offer explanations for $[A\text{–}D]$. First, $[A]$ follows from $w^{n+1} = \arg\min_w G(x^{n+1}, a, w, \epsilon^{n+1})$. Next, $[B]$ follows from $||A(x - x^n)||_2 \leq ||A||_2||x - x^n||_2 \leq ||x - x^n||_2$ for $||A||_2 \leq 1$ so that $||x - x^n||_2^2 - ||A(x - x^n)||_2^2 \geq 0$. Next, $[C]$ follows from $\epsilon^{n+1} \leq \epsilon^n$. Finally, $[D]$ follows from $x^{n+1} = \arg\min_x G(x, x^n, w^n, \epsilon^n)$. We now set up a telescoping sum of non-negative terms:

$$\begin{aligned} &\sum_{n=1}^{P} \left(G(x^{n+1}, x^n, w^n, \epsilon^{n+1}) - G(x^{n+1}, x^{n+1}, w^n, \epsilon^{n+1}) \right) \\ \leq\ &\sum_{n=1}^{P} \left(G(x^n, x^n, w^n, \epsilon^n) - G(x^{n+1}, x^{n+1}, w^{n+1}, \epsilon^{n+1}) \right) \\ =\ &G(x^1, x^1, w^1, \epsilon^1) - G(x^{N+1}, x^{N+1}, w^{N+1}, \epsilon^{N+1}) \leq C \end{aligned}$$

for some fixed constant C. All this implies:

$$\begin{aligned} &\sum_{n=1}^{P} \left(||x^n - x^{n+1}||_2^2 - ||A(x^n - x^{n+1})||_2^2 \right) \\ =\ &\sum_{n=1}^{P} \left(G(x^{n+1}, x^n, w^n, \epsilon^{n+1}) - G(x^{n+1}, x^{n+1}, w^n, \epsilon^{n+1}) \right) \leq C. \end{aligned}$$

Since $||A(x^n - x^{n+1})||_2^2 \leq ||A||_2^2 ||x^n - x^{n+1}||_2^2$ and $||A||_2 < 1$:

$$\begin{aligned} ||x^n - x^{n+1}||_2^2 - ||A(x^n - x^{n+1})||_2^2 &\geq ||x^n - x^{n+1}||_2^2 - ||A||_2^2 ||x^n - x^{n+1}||_2^2 \\ &= (1 - ||A||_2^2)||x^n - x^{n+1}||_2^2. \end{aligned}$$

Consequently, we have:

$$\begin{aligned} & \sum_{n=1}^{P} (1 - ||A||_2^2)||x^n - x^{n+1}||^2 &\leq& \sum_{n=1}^{P} \left(||x^n - x^{n+1}||_2^2 - ||A(x^n - x^{n+1})||_2^2\right) \leq C \\ \Longrightarrow \quad & \sum_{n=1}^{\infty} ||x^n - x^{n+1}||^2 &<& \infty \\ \Longrightarrow \quad & ||x^n - x^{n+1}|| &\to& 0. \end{aligned}$$

To prove that the (x^n)'s are bounded, consider:

$$G(x^n, x^n, w^n, \epsilon^n) = ||Ax^n - b||_2^2 + 2\sum_{k=1}^{N} \lambda_k \left((x_k^n)^2 + (\epsilon^n)^2\right)^{\frac{q_k}{2}},$$

We have that $\lambda_k |x_k^n|^{q_k} \leq G(x^n, x^n, w^n, \epsilon^n)$. This implies that:

$$|x_k^n| \leq \left(\frac{1}{\lambda_k} G(x^n, x^n, w^n, \epsilon^n)\right)^{\frac{1}{q_k}} \leq \left(\frac{1}{\lambda_k} G(x^0, x^0, w^0, \epsilon^0)\right)^{\frac{1}{q_k}}$$

This implies that: $||x^n||_1 = \sum_{k=1}^{N} |x_k^n| \leq N \left(\frac{1}{\lambda_k} G(x^0, x^0, w^0, \epsilon^0)\right)^{\frac{1}{q_k}} =: C'$. □

Lemma 4.4.4. *There exists a special subsequence (n_l) such that for every member of the subsequence we have:*

$$\epsilon^{n_l} = ||x^{n_l} - x^{n_l - 1}||^{\gamma} + \alpha^{n_l} < \epsilon^{n_l - 1}.$$

Additionally, there is a subsequence of this subsequence (n_{l_r}) such that $(x^{n_{l_r}})_r$ is convergent.

Proof. By the definition of the ϵ^n's:

$$\epsilon^n = \min(\epsilon^{n-1}, ||x^n - x^{n-1}||^\gamma + \alpha^n),$$

we know that $\epsilon^n \to 0$, since $||x^n - x^{n-1}|| \to 0$ and $\alpha^n \to 0$. It follows that a subsequence (n_l) must exist such that $\epsilon^{n_l} < \epsilon^{n_l - 1}$, for otherwise, there would be some N_0 such that for $n \geq N_0$, $\epsilon^{n+1} = \epsilon^n$ and the sequence of ϵ^n's would not converge to zero. The fact that n_{l_r} exists is a consequence of the boundedness of the iterates (x^n), which implies the existence of a weakly converging subsequence of the x^{n_l} (strongly in a finite dimensional space). □

Lemma 4.4.5. *The limit $\overline{x} = \lim_{r\to\infty} x^{n_{l_r}}$ of the converging subsequence $(x^{n_{l_r}})_{r\in\mathbb{N}}$ of* (IRLS) *satisfies the optimality conditions for the functional:*

$$||Ax - b||_2^2 + 2\sum_{k=1}^{N} \lambda_k |x_k|^{q_k}$$

for $1 \leq q_k < 2$, namely:

$$\begin{aligned} (A^T(b - Ax))_k &= \lambda_k \operatorname{sgn}(x_k) q_k |x_k|^{q_k - 1}\,, & x_k &\neq 0 \\ (A^T(b - Ax))_k &= 0\,, & x_k &= 0 \quad (q_k > 1) \\ \left|(A^T(b - Ax))_k\right| &\leq \lambda_k\,, & x_k &= 0 \quad (q_k = 1). \end{aligned} \tag{4.4.5}$$

Proof. Consider the subsequence (n_{l_r}) of the previous lemma. For this subsequence we have from the above lemma that:

$$\epsilon^{n_{l_r}} = ||x^{n_{l_r}} - x^{n_{l_r}-1}||^\gamma + \alpha^{n_{l_r}} < \epsilon^{n_{l_r}-1}.$$

This subsequence is also convergent:

$$\lim_{r\to\infty} x_k^{n_{l_r}} = \overline{x}_k \quad \text{for} \quad k = 1, \ldots, N.$$

For each k, we consider three separate cases, depending on the limit $\overline{x}_k$ and on the value of q_k:

(1) $\overline{x}_k \neq 0$ and $1 \leq q_k < 2$,

(2) $\overline{x}_k = 0$ and $q_k = 1$,

(3) $\overline{x}_k = 0$ and $q_k > 1$.

Consider now the first case: $\lim_{r\to\infty} x^{n_{l_r}} = \overline{x_k} \neq 0$. Since $x^{n_{l_r}} \to \overline{x}$ and $||x^n - x^{n+1}|| \to 0$, we have that: $x_k^{n_{l_r}+1} \to \overline{x}_k$. Plugging the subsequence into the iteration, we have:

$$x_k^{n_{l_r}+1} + q_k\lambda_k w_k^{n_{l_r}} x_k^{n_{l_r}+1} = x_k^{n_{l_r}} + (A^T(b - Ax^{n_{l_r}}))_k.$$

Taking the limit as $r \to \infty$ we then recover:

$$\overline{x_k} + q_k\lambda_k \lim_{r\to\infty} w_k^{n_{l_r}} x_k^{n_{l_r}+1} = \overline{x_k} + (A^T(b - A\overline{x}))_k,$$

and we thus recover:

$$\lim_{r\to\infty} w_k^{n_{l_r}} x_k^{n_{l_r}+1} = \frac{1}{q_k\lambda_k}(A^T(b - A\overline{x}))_k. \tag{4.4.6}$$

Since we want to compute $[A^T(b - A\bar{x})]_k$, we are interested in the value of $\lim_{r\to\infty} w_k^{n_{l_r}} x_k^{n_{l_r}+1}$. In the case $\lim_{r\to\infty} x^{n_{l_r}} = \overline{x_k} \neq 0$, we obtain

$$\lim_{r\to\infty} w_k^{n_{l_r}} x_k^{n_{l_r}+1} = \lim_{l\to\infty} w_k^{n_{l_r}} x_k^{n_{l_r}} \frac{x_k^{n_{l_r}+1}}{x_k^{n_{l_r}}} = \lim_{l\to\infty} x_k^{n_{l_r}} w_k^{n_{l_r}},$$

since $||x^{n+1} - x^n|| \to 0$ implies $\lim_{r\to\infty} \frac{x_k^{n_{l_r}+1}}{x_k^{n_{l_r}}} = 1$. This implies:

$$\begin{aligned}
\lim_{r\to\infty} w_k^{n_{l_r}} x_k^{n_{l_r}+1} &= \lim_{r\to\infty} \frac{x_k^{n_{l_r}}}{\left((x_k^{n_{l_r}})^2 + (\epsilon^{n_{l_r}})^2\right)^{\frac{2-q_k}{2}}} = \frac{\overline{x_k}}{((\overline{x_k})^2 + 0)^{\frac{2-q_k}{2}}} \\
&= \frac{\operatorname{sgn}(\overline{x_k})\,|\overline{x_k}|}{|\overline{x_k}|^{2-q_k}} = \operatorname{sgn}(\overline{x_k})\,|\overline{x_k}|^{q_k-1}.
\end{aligned}$$

Thus, we obtain that: $(A^T(b - A\overline{x}))_k = \lambda_k q_k \operatorname{sgn}(\overline{x_k})\,|\overline{x_k}|^{q_k-1}$, as required.

Consider now the situation in which $\lim_{r\to\infty} x^{n_{l_r}} = \overline{x_k} = 0$. We have

$$x_k^{n_{l_r}} + q_k\lambda_k w_k^{n_{l_r}-1} x_k^{n_{l_r}} = x_k^{n_{l_r}-1} + (A^T(b - Ax^{n_{l_r}-1}))_k,$$

from which we recover:

$$\lim_{r\to\infty} w_k^{n_{l_r}-1} x_k^{n_{l_r}} = \frac{1}{q_k\lambda_k}(A^T(b - A\overline{x}))_k.$$

We set:

$$\Gamma_k = \lim_{r\to\infty} \left(w_k^{n_{l_r}-1} x_k^{n_{l_r}}\right)^2.$$

For case 2, we would like to show that $\Gamma_k \leq 1$; for case 3, that $\Gamma_k = 0$. We start by supposing that $\Gamma_k > 0$, and derive consequences from this. In case 3, they will lead to a contradiction to $\lim_{r\to\infty} x^{n_{l_r}} = \overline{x_k} = 0$; in case 2 we shall see that $\Gamma_k \leq 1$ must follow. So suppose $\Gamma_k > 0$. Then, for any σ with $0 < \sigma < 1$, there exists some r_0 large enough such that for all $r > r_0$, we have that:

$$(w_k^{n_{l_r}-1} x_k^{n_{l_r}})^2 \geq \Gamma_k(1 - \sigma).$$

Now let $s_k^n = ((x_k^n)^2 + (\epsilon^n)^2)^{\frac{1}{2}}$. Since $w_k^n = ((x_k^n)^2 + (\epsilon^n)^2)^{\frac{q_k-2}{2}}$, we have that $s_k^n = (w_k^n)^{\frac{1}{q_k-2}}$. It follows that:

$$\begin{aligned}
|x_k^{n_{l_r}}|^2 &\geq \Gamma_k(1-\sigma)(s^{n_{l_r}-1})^{2(2-q_k)} \\
&= \Gamma_k(1-\sigma)\left((x_k^{n_{l_r}-1})^2 + (\sigma^{n_{l_r}-1})^2\right)^{2-q_k} \\
&> \Gamma_k(1-\sigma)\left((x_k^{n_{l_r}-1})^2 + (\sigma^{n_{l_r}})^2\right)^{2-q_k} \\
&= \Gamma_k(1-\sigma)\left((x_k^{n_{l_r}-1})^2 + \left(||x^{n_{l_r}} - x^{n_{l_r}-1}||^\gamma + \alpha^{n_{l_r}}\right)^2\right)^{2-q_k} \\
&> \Gamma_k(1-\sigma)\left((x_k^{n_{l_r}-1})^2 + |x_k^{n_{l_r}-1} - x_k^{n_{l_r}}|^{2\gamma}\right)^{2-q_k}.
\end{aligned}$$

Let us use the substitutions:

$$u = x_k^{n_{l_r}-1} \quad \text{and} \quad v = x_k^{n_{l_r}} - x_k^{n_{l_r}-1} \qquad \Longrightarrow \; u + v = x_k^{n_{l_r}}.$$

Then we can rewrite $|x_k^{n_{l_r}}|^2 > \Gamma_k(1-\sigma)\left((x_k^{n_{l_r}-1})^2 + |x_k^{n_{l_r}-1} - x_k^{n_{l_r}}|^{2\gamma}\right)^{2-q_k}$ as:

$$(u+v)^2 > \Gamma_k(1-\sigma)\left(u^2 + |v|^{2\gamma}\right)^{2-q_k},$$

where:

$$(u+v)^2 = u^2 + 2uv + v^2 \leq u^2 + Ku^2 + \frac{1}{K}v^2 + v^2 = (1+K)u^2 + \left(1 + \frac{1}{K}\right)v^2$$

for all $K > 0$. Thus, we have that:

$$\Gamma_k(1-\sigma)\left(u^2 + |v|^{2\gamma}\right)^{2-q_k} < (1+K)u^2 + \left(1 + \frac{1}{K}\right)v^2. \tag{4.4.7}$$

Consider now case 2 where $q_k = 1$. We then have that:

$$\left(\Gamma_k(1-\sigma) - (1+K)\right)u^2 < \left(1 + \frac{1}{K}\right)v^2 - \Gamma_k(1-\sigma)v^{2\gamma}. \tag{4.4.8}$$

Since $\gamma > 0$, the right hand side $(1+\frac{1}{K})v^2 - \Gamma_k(1-\sigma)v^{2\gamma}$ will be strictly less than zero for sufficiently large r. That is because, since $||x^n - x^{n+1}|| \to 0$, we know that $|v| = \left|x_k^{n_{l_r}} - x_k^{n_{l_r}-1}\right| \to 0$ as $r \to \infty$ and since $2\gamma < 2$, $v^{2\gamma}$ goes to zero more slowly than v^2, which makes the whole term negative for r sufficiently large. But from (4.4.8),

$$\left(1+\frac{1}{K}\right)v^2 - \Gamma_k(1-\sigma)v^{2\gamma} < 0 \implies (\Gamma_k(1-\sigma) - (1+K)) < 0;$$

we have thus that:

$$\Gamma_k(1-\sigma) < (1+K) \text{ for all } K > 0 \qquad \implies \Gamma_k(1-\sigma) \leq 1.$$

Since σ is arbitrary small we conclude that $\Gamma_k \leq 1$. By (4.4.6) and $q_k = 1$, This means:

$$\Gamma_k = \lim_{r\to\infty}\left(w_k^{n_{l_r}-1}x_k^{n_{l_r}}\right)^2 \leq 1 \implies \lim_{r\to\infty}\left|w_k^{n_{l_r}-1}x_k^{n_{l_r}}\right| \leq 1 \implies \left|\frac{1}{\lambda_k}(A^T(b-A\overline{x}))_k\right| \leq 1.$$

This is the desired optimality condition for case 2.

Consider now case 3, where $1 < q_k < 2$. We have from (4.4.7) that:

$$\Gamma_k(1-\sigma)\left(u^2 + |v|^{2\gamma}\right)^{2-q_k} < (1+K)u^2 + \left(1+\frac{1}{K}\right)v^2 \quad \text{for all } K > 0.$$

This means in particular that:

$$\begin{aligned}\Gamma_k(1-\sigma)u^{2(2-q_k)} &< (1+K)u^2 + \left(1+\frac{1}{K}\right)v^2 \quad \text{and}\\ \Gamma_k(1-\sigma)v^{2\gamma(2-q_k)} &< (1+K)u^2 + \left(1+\frac{1}{K}\right)v^2.\end{aligned}$$

Then the average of the terms is also smaller than this quantity:

$$\frac{1}{2}\Gamma_k(1-\sigma)\left(u^{2(2-q_k)}+v^{2\gamma(2-q_k)}\right) < (1+K)u^2+\left(1+\frac{1}{K}\right)v^2.$$

Moving terms we have:

$$u^{2(2-q_k)}\left(\frac{1}{2}\Gamma_k(1-\sigma)-(1+K)u^{2(q_k-1)}\right) < \left(1+\frac{1}{K}\right)v^2-\frac{1}{2}\Gamma_k(1-\sigma)v^{2\gamma(2-q_k)}.$$

Since $1 < q_k$ and thus $2-q_k < 1$, we have that for v sufficiently small, the right hand side is negative, by the same logic as in the previous case (since $\Gamma_k > 0$ by assumption and for large r, $v \to 0$ and the first term will go to zero faster than the second). Thus, by the above inequality, for r sufficiently large, both the right and the left hand side are negative. Since $u^{2(2-q_k)}$ is non-negative, that is only possible when:

$$\frac{1}{2}\Gamma_k(1-\sigma)-(1+K)u^{2(q_k-1)} < 0$$

for r sufficiently large. However, $\lim_{r\to\infty} u^{2(q_k-1)} = \lim_{r\to\infty}(x_k^{n_{l_r}-1})^{2(q_k-1)} = 0$, so that this condition is impossible. This contradicts our original assumption that $\Gamma_k > 0$. Hence, we conclude that $\Gamma_k = 0 \implies (A^T(b-A\overline{x}))_k = 0$, which is the right optimality condition. □

4.5 IRLS SYS Algorithm

We now present some analysis for the second IRLS method. Let us restate the method. Given any initial guess x^0, the matrix A with $||A||_2 < 1$, the iterative scheme is:

$$x^{n+1} = \arg\min_x ||Ax - b||_2^2 + \sum_{k=1}^{N} \lambda_k q_k w_k^n (x_k)^2$$

with the weights and sequence of ϵ^n defined as:

$$\begin{aligned} w_k^n &= \frac{1}{((x_k^n)^2 + (\epsilon^n)^2)^{\frac{2-q_k}{2}}} \\ \epsilon^n &= \min(\epsilon^{n-1}, |G(x^{n-2}, w^{n-2}, \epsilon^{n-2}) - G(x^{n-1}, w^{n-1}, \epsilon^{n-1})|^{\frac{\gamma}{2}} + \alpha^n), \end{aligned}$$

where:

$$G(x, w, \epsilon) = ||Ax - b||_2^2 + \sum_{k=1}^{N} \lambda_k \left(q_k w_k ((x_k)^2 + \epsilon^2) + (2 - q_k)(w_k)^{\frac{q_k}{q_k - 2}} \right)$$

and $1 < q_k \leq 2$, $0 < \alpha < 1$ and $0 < \gamma < \min \frac{2}{4 - q_k^2}$. We prove that this algorithm converges to the minimizer of the functional $||Ax - b||_2^2 + 2 \sum_{k=1}^{N} \lambda_k |x_k|^{q_k}$ with $1 \leq q_k < 2$ if the functional has a unique minimizer, and that any accumulation point of the x^n is a minimizer if there are several. (Note that the traditional ℓ_1 functional is a special case of the above functional.) We show that the iterates x^n are bounded and that all converging subsequences satisfy the optimality conditions of the above functional.

We now describe how the algorithm is implemented in practice. Introducing the diagonal matrix D^n, note that we can rewrite the last term of the iterative method as:

$$\sum_{k=1}^{N} \lambda_k q_k w_k^n (x_k)^2 = \sum_{k=1}^{N} (D^n)_{kk}^2 (x_k)^2 = ||D^n x||_2^2,$$

where D^n is a diagonal matrix containing on its diagonal the elements $(D^n)_{kk} = \sqrt{\lambda_k q_k w_k^n}$. We may then rewrite the algorithm as:

$$x^{n+1} = \arg\min_x ||Ax - b||_2^2 + ||D^n x||_2^2,$$

which can be solved via the system of equations:

$$(A^T A + (D^n)^T (D^n)) x^{n+1} = A^T b$$

. Since D^n is diagonal this is equivalent to setting $\Phi^n = (D^n)^2$ where $\Phi_{k,k}^n = \lambda_k q_k w_k^n$ and solving $(A^T A + \Phi^n) x^{n+1} = A^T b$ at each iteration. This can then be solved using for example the conjugate gradient scheme, reusing the previous iterate as the starting iterate for the CG scheme at each iteration.

Lemma 4.5.1. *Using the surrogate functional:*

$$G(x, w, \epsilon) = ||Ax - b||_2^2 + \sum_{k=1}^{N} \lambda_k \left(q_k w_k ((x_k)^2 + \epsilon^2) + (2 - q_k)(w_k)^{\frac{q_k}{q_k - 2}} \right)$$

we can recover the weights:

$$(w^{n+1})_k = \frac{1}{((x_k^{n+1})^2 + (\epsilon^{n+1})^2)^{\frac{2-q_k}{2}}}$$

using the minimization procedure:

$$w^{n+1} = \arg\min_w G(x^{n+1}, w, \epsilon^{n+1})$$

and the iterative scheme:

$$x^{n+1} = \arg\min_x \left\{ ||Ax - b||_2^2 + \sum_{k=1}^{N} \lambda_k q_k w_k^n (x_k)^2 \right\}$$

using the minimization procedure:

$$x^{n+1} = \arg\min_x G(x, w^n, \epsilon^n).$$

Proof. First, for the weights, we have $w^{n+1} = \arg\min_w G(x^{n+1}, w, \epsilon^{n+1})$ and only the last term of G has a dependence on w. We have:

$$\begin{aligned}
& \frac{\partial}{\partial w_k}\left(q_k w_k((x_k)^2 + \epsilon^2) + (2-q_k)(w_k)^{\frac{q_k}{q_k-2}}\right) = 0 \\
\implies \quad & q_k((x_k)^2 + \epsilon^2) + (2-q_k)\frac{q_k}{q_k-2}(w_k)^{\frac{q_k}{q_k-2}-1} = 0 \\
\implies \quad & w_k = \frac{1}{((x_k)^2 + \epsilon^2)^{\frac{2-q_k}{2}}}
\end{aligned}$$

since $\dfrac{q_k}{q_k-2} - \dfrac{q_k-2}{q_k-2} = \dfrac{2}{q_k-2}$.

Now consider the functional:

$$G(x, w^n, \epsilon^n) = ||Ax-b||_2^2 + \sum_{k=1}^{N} \lambda_k \left(q_k w_k^n((x_k)^2 + (\epsilon^n)^2) + (2-q_k)(w_k^n)^{\frac{q_k}{q_k-2}}\right);$$

evaluating this at $x = x^n$ gives

$$G(x^n, w^n, \epsilon^n) = ||Ax-b||_2^2 + 2\sum_{k=1}^{N} \lambda_k((x_k^n)^2 + (\epsilon^n)^2)^{\frac{q_k}{2}},$$

since:

$$\begin{aligned}
& q_k w_k^n((x_k)^2 + (\epsilon^n)^2) + (2-q_k)(w_k^n)^{\frac{q_k}{q_k-2}} \\
= \quad & q_k((x_k^n)^2 + (\epsilon^n)^2)^{\left(\frac{q_k-2}{2}+\frac{2}{2}\right)} + (2-q_k)((x_k^n)^2 + (\epsilon^n)^2)^{\left(\frac{q_k-2}{2}\frac{q_k}{q_k-2}\right)} \\
= \quad & 2((x_k^n)^2 + (\epsilon^n)^2)^{\frac{q_k}{2}}.
\end{aligned}$$

We note that as $n \to \infty$, and assuming $x^n \to x$ and $\epsilon^n \to 0$, we have that:

$$\lim_{n\to\infty} G(x^n, w^n, \epsilon^n) = ||Ax - b||_2^2 + 2\sum_{k=1}^{N} \lambda_k (x_k)^{q_k},$$

so we recover the functional we would like to minimize.

Next, let us check that the statement:

$$x^{n+1} = \arg\min_{x} G(x, w^n, \epsilon^n)$$

recovers the iterative scheme. Consider:

$$G(x, w^n, \epsilon^n) = ||Ax - b||_2^2 + \sum_{k=1}^{N} \lambda_k \left(q_k w_k^n ((x_k)^2 + (\epsilon^n)^2) + (2 - q_k)(w_k^n)^{\frac{q_k}{q_k - 2}} \right).$$

Keeping from above only the terms that depend on x, we see that we recover:

$$x^{n+1} = \arg\min_{x} \left\{ ||Ax - b||_2^2 + \sum_{k=1}^{N} \lambda_k q_k w_k^n (x_k)^2 \right\}.$$

□

Lemma 4.5.2. *The G functions satisfy $G(x^{n+1}, w^{n+1}, \epsilon^{n+1}) \leq G(x^n, w^n, \epsilon^n)$.*

Proof. First note the relations:

$$\begin{aligned} x^{n+1} &= \arg\min_{x} G(x, w^n, \epsilon^n) \\ w^{n+1} &= \arg\min_{w} G(x^{n+1}, w, \epsilon^{n+1}) \\ \epsilon_{n+1} &\leq \epsilon_n. \end{aligned}$$

Applying the above, we have:

$$G(x^{n+1}, w^{n+1}, \epsilon_{n+1}) \leq G(x^{n+1}, w^n, \epsilon^{n+1}) \leq G(x^{n+1}, w^n, \epsilon^n) \leq G(x^n, w^n, \epsilon^n).$$

□

Lemma 4.5.3. *The iterates (x^n) are bounded in norm.*

Proof. To prove that the (x^n)'s are bounded, consider:

$$G(x^n, w^n, \epsilon^n) = ||Ax^n - b||_2^2 + 2\sum_{k=1}^{N} \lambda_k \left((x_k^n)^2 + (\epsilon^n)^2\right)^{\frac{q_k}{2}}.$$

We have that $\lambda_k |x_k^n|^{q_k} \leq G(x^n, w^n, \epsilon^n)$. By Lemma 4.5.2 it follows that:

$$|x_k^n| \leq \left(\frac{1}{\lambda_k} G(x^n, w^n, \epsilon^n)\right)^{\frac{1}{q_k}} \leq \left(\frac{1}{\lambda_k} G(x^0, w^0, \epsilon^0)\right)^{\frac{1}{q_k}}.$$

This implies that: $||x^n||_1 = \sum_{k=1}^{N} |x_k^n| \leq N \left(\frac{1}{\lambda_k} G(x^0, w^0, \epsilon^0)\right)^{\frac{1}{q_k}} = C$. Thus, the ℓ_2-norm of x^n is bounded as well. □

Lemma 4.5.4. *There exists a strictly increasing sequence (n_l) such that for every member of the subsequence we have:*

$$\epsilon^{n_l+1} = |G(x^{n_l-1}, w^{n_l-1}, \epsilon^{n_l-1}) - G(x^{n_l}, w^{n_l}, \epsilon^{n_l})|^{\frac{\gamma}{2}} + \alpha^{n_l+1}.$$

Additionally, there is a subsequence of this sequence (n_{l_r}) such that:

$$\lim_{r \to \infty} x_k^{n_{l_r}} = \overline{x_k}.$$

Proof. By the definition of the ϵ^n's:

$$\epsilon^n = \min(\epsilon^{n-1}, |G(x^{n-2}, w^{n-2}, \epsilon^{n-2}) - G(x^{n-1}, w^{n-1}, \epsilon^{n-1})|^{\frac{\gamma}{2}} + \alpha^n).$$

We have that $\epsilon^n \to 0$ since by Lemma 4.5.2 we know that $|G(x^{n-2}, w^{n-2}, \epsilon^{n-2}) - G(x^{n-1}, w^{n-1}, \epsilon^{n-1})| \to 0$ and $\alpha^n \to 0$ since $0 < \alpha < 1$. It follows that a sequence (n_l) with the desired properties must exist, for otherwise, there would be some N_0 such that for $n \geq N_0$, $\epsilon^{n+1} = \epsilon^n$ and the sequence of ϵ's would not converge to zero. The fact that n_{l_r} exists is a consequence of the boundedness of the sequence (x^{n_l}), which implies the existence of a weakly converging subsequence (strongly in a finite dimensional space). □

Lemma 4.5.5. *Let $s^n_k = ((x^n_k)^2 + (\epsilon^n)^2)^{\frac{1}{2}}$. Then there exists $\eta \in \mathbb{R} < \infty$ such that $s^n_k \leq \eta$ for all n, k.*

Proof. Since $s^n_k = ((x^n_k)^2 + (\epsilon^n)^2)^{\frac{1}{2}}$ and (x^n) are bounded, we have:

$$s^n_k \leq |x^n_k| + \epsilon^n \leq ||x^n|| + \epsilon^0 \leq \eta.$$

□

Lemma 4.5.6. *Consider the sequence (n_l) introduced in Lemma 4.5.4. For each element n_l of the sequence, ϵ^{n_l+1} is bounded below as follows:*

$$\epsilon^{n_l+1} \geq C(s^{n_l}_k)^{q_k\gamma}(s^{n_l-1}_k - s^{n_l}_k)^{2\gamma}, \quad \textit{for each } k \in \{1, \ldots, N\}.$$

Proof. Recall that we have:

$$\epsilon^{n_l+1} = |G(x^{n_l-1}, w^{n_l-1}, \epsilon^{n_l-1}) - G(x^{n_l}, w^{n_l}, \epsilon^{n_l})|^{\frac{\gamma}{2}} + \alpha^{n_l+1}.$$

Based on the inequalities derived earlier:

$$|G(x^n, w^n, \epsilon^n) - G(x^{n+1}, w^{n+1}, \epsilon^{n+1})| \geq |G(x^{n+1}, w^n, \epsilon^{n+1}) - G(x^{n+1}, w^{n+1}, \epsilon^{n+1})|.$$

Decomposing the (non-negative) difference $G(x^{n+1}, w^n, \epsilon^{n+1}) - G(x^{n+1}, w^{n+1}, \epsilon^{n+1})$ into a sum:

$$G(x^{n+1}, w^n, \epsilon^{n+1}) - G(x^{n+1}, w^{n+1}, \epsilon^{n+1}) = \sum_{k=1}^{N} \lambda_k T_k^n,$$

and then we compute a lower bound on each of the T_k^n. Recall that:

$$G(x, w, \epsilon) = ||Ax - b||_2^2 + \sum_{k=1}^{N} \lambda_k \left(q_k w_k ((x_k)^2 + \epsilon^2) + (2 - q_k)(w_k)^{\frac{q_k}{q_k - 2}} \right).$$

We have:

$$\begin{aligned} G(x^{n+1}, w^{n+1}, \epsilon^{n+1}) &= ||Ax^{n+1} - b||_2^2 + 2 \sum_{k=1}^{N} \lambda_k ((x_k^{n+1})^2 + (\epsilon^{n+1})^2)^{\frac{q_k}{2}} \\ G(x^{n+1}, w^n, \epsilon^{n+1}) &= ||Ax^{n+1} - b||_2^2 \\ &\quad + \sum_{k=1}^{N} \lambda_k \left(q_k w_k^n ((x_k^{n+1})^2 + (\epsilon^{n+1})^2) + (2 - q_k)(w_k^n)^{\frac{q_k}{q_k - 2}} \right). \end{aligned}$$

Next, recall that:

$$s_k^n = ((x_k^n)^2 + (\epsilon^n)^2)^{\frac{1}{2}} \quad \Longrightarrow \quad w_k^n = (s_k^n)^{q_k - 2} \quad \Longrightarrow \quad (w_k^n)^{\frac{q_k}{q_k - 2}} = (s_k^n)^{q_k}.$$

and we can rewrite:

$$\begin{aligned} G(x^{n+1}, w^{n+1}, \epsilon^{n+1}) &= ||Ax^{n+1} - b||_2^2 + 2 \sum_{k=1}^{N} \lambda_k (s_k^{n+1})^{q_k} \\ G(x^{n+1}, w^n, \epsilon^{n+1}) &= ||Ax^{n+1} - b||_2^2 + \sum_{k=1}^{N} \lambda_k \left(q_k (s_k^n)^{q_k - 2} (s_k^{n+1})^2 + (2 - q_k)(s_k^n)^{q_k} \right). \end{aligned}$$

Thus, $G(x^{n+1}, w^n, \epsilon^{n+1}) - G(x^{n+1}, w^{n+1}, \epsilon^{n+1}) = \sum_{k=1}^{N} \lambda_k T_k^n$ with

$$T_k^n = \left(q_k (s_k^n)^{q_k - 2} (s_k^{n+1})^2 + (2 - q_k)(s_k^n)^{q_k} - 2(s_k^{n+1})^2\right).$$

We now find a lower bound on T_k^n. Introduce the substitutions:

$$x = s_k^{n+1} \quad , \quad t = s_k^n - s_k^{n+1} \qquad \Longrightarrow \qquad s_k^n = x + t.$$

Note that by Lemma 4.5.5 $x \leq \eta$ and $(x + t) \leq \eta$ so in particular $x + ct \leq \eta$ for any $c \in (0, 1)$. Using the substitutions above and dropping the subscripts, we have that:

$$\begin{aligned} T &= q(x+t)^{q-2} x^2 + (2-q)(x+t)^q - 2x^q \\ &= (x+t)^{q-2} \left(q x^2 + (2-q)(x+t)^2 - 2x^q (x+t)^{2-q}\right) \\ &= (x+t)^{q-2} x^2 \left(q + (2-q)(1 + \frac{t}{x})^2 - 2(1 + \frac{t}{x})^{2-q}\right). \end{aligned}$$

We would like to bound the inside of the () and consequently all of T. We do this by introducing a function Φ defined by:

$$\Phi(u) = q + (2-q)\left(1 + u\frac{t}{x}\right)^2 - 2\left(1 + u\frac{t}{x}\right)^{2-q}.$$

Then, by construction $\Phi(1) = \big(q + (2-q)(1+\frac{t}{x})^2 - 2(1+\frac{t}{x})^{2-q}\big)$ and $\Phi(0) = 0$. Also:

$$\begin{aligned}
\Phi'(u) &= \frac{t}{x}\left(2(2-q)(1+u\frac{t}{x}) - 2(2-q)(1+u\frac{t}{x})^{1-q}\right) \qquad \Longrightarrow \qquad \Phi'(0) = 0 \\
\Phi''(u) &= \left(\frac{t}{x}\right)^2\left(2(2-q) - 2(2-q)(1-q)(1+u\frac{t}{x})^{1-q}\right) \\
&= 2(2-q)\left(\frac{t}{x}\right)^2\left(1 + (q-1)(1+u\frac{t}{x})^{-q}\right) \\
&= 2(2-q)t^2x^{q-2}\left(x^{-q} + (q-1)(x+ut)^{-q}\right) \\
&\geq 2(2-q)t^2x^{q-2}q\eta^{-q}
\end{aligned}$$

for $u \in (0,1)$ where since $x \leq \eta$ and $x + t \leq \eta$, we have $x + ut \leq \eta$. It follows from Taylor's theorem that:

$$\Phi(1) = \Phi(0) + \Phi'(0) + \frac{1}{2}\Phi''(u) \geq 2(2-q)qt^2x^{q-2}\eta^{-q} \quad , \quad \text{for some } u \in (0,1).$$

From the above, we can write the following lower bound for all of T:

$$T \geq 2(2-q)qt^2\eta^{-q}x^{q-2}x^2(x+t)^{q-2} \geq (2-q)qx^qt^2\eta^{-2}.$$

Going back to the original notation we have the following estimate:

$$T_k^n \geq (2-q_k)q_k\eta^{-2}(s_k^n - s_k^{n+1})^2(s_k^{n+1})^{q_k}.$$

Notice that $C = (2 - q_k)q_k\eta^{-2}$ does not depend on n. Summarizing the above derivations, we have shown that:

$$\begin{aligned}
& |G(x^n, w^n, \epsilon^n) - G(x^{n+1}, w^{n+1}, \epsilon^{n+1})| \\
\geq\ & |G(x^{n+1}, w^n, \epsilon^{n+1}) - G(x^{n+1}, w^{n+1}, \epsilon^{n+1})| \\
=\ & \sum_{k=1}^{N} \lambda_k \left(q_k (s_k^n)^{q_k - 2} (s_k^{n+1})^2 + (2 - q_k)(s_k^n)^{q_k} - 2(s_k^{n+1})^2 \right) \\
=\ & \sum_{k=1}^{N} \lambda_k T_k^n \\
\geq\ & C \sum_{k=1}^{N} (s_k^n - s_k^{n+1})^2 (s_k^{n+1})^{q_k}.
\end{aligned}$$

It then follows from the definition of ϵ^n:

$$\epsilon^n = \min(\epsilon^{n-1}, |G(x^{n-2}, w^{n-2}, \epsilon^{n-2}) - G(x^{n-1}, w^{n-1}, \epsilon^{n-1})|^{\frac{\gamma}{2}} + \alpha^n),$$

and the definition of the subsequence n_l that we have the estimate:

$$\begin{aligned}
& (\epsilon^{n_l+1})^2 = \left[|G(x^{n_l-1}, w^{n_l-1}, \epsilon^{n_l-1}) - G(x^{n_l}, w^{n_l}, \epsilon^{n_l})|^{\frac{\gamma}{2}} + \alpha^{n_l} \right]^2 \\
& \geq |G(x^{n_l-1}, w^{n_l-1}, \epsilon^{n_l-1}) - G(x^{n_l}, w^{n_l}, \epsilon^{n_l})|^{\gamma} \\
& \geq C \left(\sum_{k=1}^{N} (s_k^{n_l})^{q_k} (s_k^{n_l-1} - s_k^{n_l})^2 \right)^{\gamma} \geq C (s_k^{n_l})^{q_k \gamma} (s_k^{n_l-1} - s_k^{n_l})^{2\gamma}.
\end{aligned}$$

□

We will also use the following standard lemma, proved here for the sake of completeness.

Lemma 4.5.7. *Let $F : \mathbb{R}_{++} \to \mathbb{R}$ be twice differentiable with $F''(y) > 0$. Let $\bar{y} > 0$ be the (unique) minimizer of F. If y^n is a convergent sequence such that the limit z is positive and $|F(y^n) - F(\bar{y})| \to 0$, then $z = \bar{y}$.*

Proof. Note that $F'' > 0$ implies that F is strictly convex, so F has at most one minimizer. Hence it suffices to show that $z = \bar{y}$. In fact,

$$0 \leq |F(\bar{y}) - F(z)| \leq |F(\bar{y} - F(y^n)| + |F(y^n) - F(z)|.$$

Since $|F(\bar{y}) - F(y^n)| \to 0$ by assumption and $|F(y^n) - F(z)| \to 0$ by the continuity of F and assumption of $y^n \to z$, we have thus $|F(\bar{y}) - F(z)| = 0$. Since F has only one minimizer, we have that $z = \bar{y}$. □

Lemma 4.5.8. *In the IRLS SYS iteration, we have that for the subsequence n_{l_r} and for the case when $\lim_{r\to\infty} x_k^{n_{l_r}} = \overline{x_k} \neq 0$, the weights $w^{n_{l_r}}$ satisfy:*

$$\lim_{r\to\infty} \left(w_k^{n_{l_r}-1} - w_k^{n_{l_r}} \right) = 0.$$

Proof. We start by looking at the k-th term of the difference between $G(x^n, w^n, \epsilon^n)$ and $G(x^n, w^{n-1}, \epsilon^n)$ and we use telescoping sums. We have:

$$G(x, w, \epsilon) = ||Ax - b||^2 + \sum_{k=1}^{N} \lambda_k q_k \left(w_k((x_k)^2 + (\epsilon_k)^2) + \frac{2 - q_k}{q_k} w_k^{\frac{q_k}{q_k - 2}} \right).$$

Now if we make the substitution $\beta_k^n = (x_k^n)^2 + (\epsilon_k^n)^2$ and plug in the weights $w_k^n = (\beta_k^n)^{\frac{q_k - 2}{2}}$ then we recover:

$$\begin{aligned} G(x^n, w^n, \epsilon^n) &= ||Ax^n - b||_2^2 + 2\sum_{k=1}^{N} \lambda_k (\beta_k^n)^{\frac{q_k}{2}} \\ G(x^n, w^{n-1}, \epsilon^n) &= ||Ax^n - b||_2^2 + \sum_{k=1}^{N} \lambda_k q_k \left(w_k^{n-1} \beta_k^n + \frac{2 - q_k}{q_k} (w_k^{n-1})^{\frac{q_k}{q_k - 2}} \right). \end{aligned}$$

If we now take the difference of the two, the first terms drop out; taking the k-th term in the remainder gives:

$$\begin{aligned}
\Gamma_k &= \frac{1}{\lambda_k}\left(G(x^n, w^{n-1}, \epsilon^n) - G(x^n, w^n, \epsilon^n)\right)_k \\
&= \frac{1}{\lambda_k}\left(q_k \beta_k^n w_k^{n-1} + (2-q_k)(w_k^{n-1})^{\frac{q_k}{q_k-2}} - 2(\beta_k^n)^{\frac{q_k}{2}}\right) \\
&= \frac{1}{\lambda_k}(\beta_k^n)^{\frac{q_k}{2}}\left(q_k w_k^{n-1}(\beta_k^n)^{\frac{2-q_k}{2}} + (2-q_k)(w_k^{n-1}(\beta_k^n)^{\frac{2-q_k}{2}})^{\frac{q_k}{q_k-2}} - 2\right).
\end{aligned}$$

Now note that since we have $G(x^n, w^n, \epsilon^n) \leq G(x^{n+1}, w^{n+1}, \epsilon^{n+1})$ (and based on the inequalities in Lemma 4.5.1) we then have:

$$\sum_{n=1}^{\infty} \Gamma_k \leq \sum_{n=1}^{\infty} \frac{1}{\lambda_k}\left(G(x^{n-1}, w^{n-1}, \epsilon^{n-1}) - G(x^n, w^n, \epsilon^n)\right) < \infty.$$

Therefore we have that:

$$w_k^{n-1}(\beta_k^n)^{\frac{2-q_k}{2}} + \frac{2-q_k}{q_k}\left((w_k^{n-1})(\beta_k^n)^{\frac{2-q_k}{2}}\right)^{\frac{q_k}{q_k-2}} - \frac{2}{q_k} \to 0 \text{ as } n \to \infty. \tag{4.5.1}$$

Now replace n by the subsequence n_{l_r} and let $\theta_k^{n_{l_r}} = w_k^{n_{l_r}-1}(\beta_k^{n_{l_r}})^{\frac{2-q_k}{2}} = \frac{w_k^{n_{l_r}-1}}{w_k^{n_{l_r}}}$. The denominator stays bounded below strictly above zero since we assumed $\overline{x_k} \neq 0$. Then we can rewrite (4.5.1) as:

$$\theta_k^{n_{l_r}} + \frac{2-q_k}{q_k}(\theta_k^{n_{l_r}})^{\frac{q_k}{q_k-2}} - \frac{2}{q_k} \to 0.$$

Let the strictly convex function $F : \mathbb{R}_{++} \to \mathbb{R}$ be defined by:

$$F(\theta) = \theta + \frac{2-q_k}{q_k}\theta^{\frac{q_k}{q_k-2}}.$$

Since $F(1) = 1 + \frac{2-q_k}{q_k} = \frac{2}{q_k}$ we have that $\lim_{l\to\infty} |F(\theta_k^{n_{l_r}}) - F(1)| \to 0$. Since 1 is the unique minimizer of F, by Lemma 4.5.7 we conclude that:

$$\lim_{l\to\infty} |\theta_k^{n_{l_r}} - 1| \to 0 \qquad \Longrightarrow \qquad \lim_{l\to\infty} \left(w_k^{n_{l_r}-1} - w_k^{n_{l_r}} \right) \to 0.$$

□

Lemma 4.5.9. *The limit of the converging subsequence* $\overline{x} = \lim_{l\to\infty} x^{n_l}$ *of IRLS satisfies the optimality conditions for the functional* $||Ax-b||_2^2 + 2\sum_{k=1}^{N} \lambda_k |x_k|^{q_k} \quad , 1 \leq q_k < 2$*:*

$$\begin{aligned}
(A^T(b-Ax))_k &= \lambda_k \operatorname{sgn}(x_k) q_k |x_k|^{q_k-1}\,, && x_k \neq 0 \\
(A^T(b-Ax))_k &= 0\,, && x_k = 0 \quad (q_k > 1) \\
\big|(A^T(b-Ax))_k\big| &\leq \lambda_k\,, && x_k = 0 \quad (q_k = 1)
\end{aligned} \tag{4.5.2}$$

Proof. Consider the subsequence (n_{l_r}). For this subsequence we have from the above lemma that:

$$\epsilon^{n_{l_r}+1} = |G(x^{n_{l_r}-1}, w^{n_{l_r}-1}, \epsilon^{n_{l_r}-1}) - G(x^{n_{l_r}}, w^{n_{l_r}}, \epsilon^{n_{l_r}})|^{\frac{\gamma}{2}} + \alpha^{n_{l_r}+1}$$

and $\lim_{r\to\infty} x^{n_{l_r}} = \overline{x}$. For each $k \in \{1, \ldots, N\}$, we consider three separate cases, depending on the limit $x_k^{n_{l_r}}$ as $r \to \infty$ and on the value of q_k. The first case is when $\lim_{r\to\infty} x_k^{n_{l_r}} \neq 0$. The second case is when $\lim_{r\to\infty} x_k^{n_{l_r}} = 0$ and $q_k > 1$. The final case is when $\lim_{l\to\infty} x_k^{n_{l_r}} = 0$ and $q_k = 1$. Recall the iteration procedure:

$$x^n = \arg\min_x ||Ax-b||_2^2 + \sum_k q_k \lambda_k w_k^{n-1} (x_k)^2$$

Upon differentiation with respect to x_k we have: $(A^T(Ax^n - b))_k + q_k\lambda_k w_k^{n-1}x_k^n = 0$ so that plugging in (n_{l_r}) and taking the limit as $r \to \infty$ we have:

$$\frac{1}{q_k\lambda_k}(A^T(b - A\overline{x}))_k = \lim_{r\to\infty} w_k^{n_{l_r}-1}x_k^{n_{l_r}}.$$

By Lemma 4.5.6 we have the lower bound:

$$(\epsilon^{n_l+1})^2 \geq C(s_k^{n_l})^{q_k\gamma}(s_k^{n_l-1} - s_k^{n_l})^{2\gamma}.$$

First, consider the case $\lim_{r\to\infty} x_k^{n_{l_r}} = \overline{x_k} \neq 0$. In this case, we have, by Lemma 4.5.8, that $\lim_{r\to\infty}\left(w_k^{n_{l_r}-1} - w_k^{n_{l_r}}\right) = 0$ together with $w_k^{n_{l_r}} \geq C > 0$ for all r, from which it follows that:

$$\lim_{r\to\infty} x_k^{n_{l_r}}w_k^{n_{l_r}-1} = \lim_{r\to\infty} x_k^{n_{l_r}}w_k^{n_{l_r}}\frac{w_k^{n_{l_r}-1}}{w_k^{n_{l_r}}} = \lim_{r\to\infty} x_k^{n_{l_r}}w_k^{n_{l_r}}\lim_{r\to\infty}\frac{w_k^{n_{l_r}-1}}{w_k^{n_{l_r}}} = \lim_{r\to\infty} x_k^{n_{l_r}}w_k^{n_{l_r}}.$$

The last limit can now be evaluated directly:

$$\begin{aligned}\lim_{l\to\infty} x_k^{n_l}w_k^{n_l} &= \lim_{l\to\infty}\frac{x_k^{n_l}}{((x_k^{n_l})^2 + (\epsilon_k^{n_l})^2)^{\frac{2-q_k}{2}}} = \frac{\overline{x}}{((\overline{x_k})^2 + 0)^{\frac{2-q_k}{2}}} \\ &= \frac{\overline{x_k}}{|\overline{x_k}|^{2-q_k}} = \frac{|\overline{x_k}|\operatorname{sgn}(\overline{x_k})|}{|\overline{x_k}|^{2-q_k}} = \operatorname{sgn}(\overline{x_k})|\overline{x_k}|^{q_k-1}.\end{aligned}$$

We thus recover the correct condition when $\overline{x_k} \neq 0$, namely that: $(A^T(b - A\overline{x}))_k = \lambda_k q_k \operatorname{sgn}(\overline{x_k})|\overline{x_k}|^{q_k-1}$.

Consider now cases 2 and 3 for which $\lim_{r\to\infty} x^{n_{l_r}} = \overline{x_k} = 0$. For case 2, $q_k > 1$ and for case 3, $q_k = 1$. We would like to show that $(A^T(b - A\overline{x}))_k = 0$ if $q_k > 1$ and $\frac{1}{\lambda_k}|(A^T(b - A\overline{x}))_k| \leq 1$ if $q_k = 1$. We set:

$$\Gamma_k = \lim_{r\to\infty}(w_k^{n_{l_r}-1}x_k^{n_{l_r}})^2$$

For case 2, we would like to show that $\Gamma_k = 0$. To do this, we shall suppose that $\Gamma_k > 0$ and derive a contradiction to $\lim_{r\to\infty} x^{n_{l_r}} = \overline{x_k} = 0$. So let's assume $\Gamma_k > 0$. For some $0 < \epsilon < 1$, there exists some r_0 large enough such that for all $r > r_0$, we have then that:

$$(w_k^{n_{l_r}-1} x_k^{n_{l_r}})^2 \geq \Gamma_k(1-\epsilon).$$

From this it follows that using $s_k^n = (w_k^n)^{\frac{-1}{2-q_k}}$:

$$\begin{aligned} |x_k^{n_{l_r}}|^2 &\geq \Gamma_k(1-\epsilon)(s^{n_{l_r}-1})^{2(2-q_k)} = \Gamma_k(1-\epsilon)\left((x_k^{n_{l_r}-1})^2 + (\epsilon^{n_{l_r}-1})^2\right)^{2-q_k} \\ &> \Gamma_k(1-\epsilon)\left((x_k^{n_{l_r}-1})^2 + (\epsilon^{n_{l_r}+1})^2\right)^{2-q_k}, \end{aligned}$$

where we have used $\epsilon^{n_{l_r}-1} \geq \epsilon^{n_{l_r}} \geq \epsilon^{n_{l_r}+1}$. From the upper bound that we have previously derived on ϵ^{n_l+1} we have:

$$(\epsilon^{n_{l_r}+1})^2 \geq C(s^{n_{l_r}})^{q_k\gamma}(s_k^{n_{l_r}-1} - s_k^{n_{l_r}})^{2\gamma}.$$

It follows that:

$$\begin{aligned} |x_k^{n_{l_r}}|^{\frac{2}{2-q_k}} &> (\Gamma_k(1-\sigma))^{\frac{1}{2-q_k}}\left((x_k^{n_{l_r}-1})^2 + C(s^{n_{l_r}})^{q_k\gamma}(s_k^{n_{l_r}-1} - s_k^{n_{l_r}})^{2\gamma}\right) \\ &> (\Gamma_k(1-\sigma))^{\frac{1}{2-q_k}}\left((x_k^{n_{l_r}-1})^2 + C(|x_k^{n_{l_r}}|)^{q_k\gamma}(s_k^{n_{l_r}-1} - s_k^{n_{l_r}})^{2\gamma}\right) \\ &> \widetilde{\Gamma_k}C|x_k^{n_{l_r}}|^{q_k\gamma}(s_k^{n_{l_r}-1} - s_k^{n_{l_r}})^{2\gamma} \quad \text{where} \quad \widetilde{\Gamma_k} = (\Gamma_k(1-\sigma))^{\frac{1}{2-q_k}}. \end{aligned}$$

Let $\beta_k = \frac{2}{2-q_k} - q_k\gamma$ and assume that $\beta_k > 0$ and $\frac{\beta_k}{2\gamma} > 1$. (We will show later that this holds with the choice of γ that we have prescribed.) Then we have:

$$\begin{aligned} |x_k^{n_{l_r}}|^{\beta_k} &> \widetilde{\Gamma_k}C|s_k^{n_{l_r}-1} - s_k^{n_{l_r}}|^{2\gamma} \implies |s_k^{n_{l_r}-1} - s_k^{n_{l_r}}| < (\widetilde{\Gamma_k}C)^{\frac{-1}{2\gamma}}|x_k^{n_{l_r}}|^{\frac{\beta_k}{2\gamma}} \\ s_k^{n_{l_r}-1} &\geq s_k^{n_{l_r}} - |s_k^{n_{l_r}-1} - s_k^{n_{l_r}}| > |x_k^{n_{l_r}}| - (\widetilde{\Gamma_k}C)^{\frac{-1}{2\gamma}}|x_k^{n_{l_r}}|^{\frac{\beta_k}{2\gamma}}, \end{aligned}$$

since $s_k^{n_{l_r}} > |x^{n_{l_r}}|$. Because $\frac{\beta_k}{2\gamma} > 1$ and $x_k^{n_{l_r}} \to 0$, we have that $|x_k^{n_{l_r}}|^{\frac{\beta_k}{2\gamma}}$ goes to zero faster than $|x_k^{n_{l_r}}|$. Thus, we can pick a sufficiently large r such that $|x_k^{n_{l_r}}|^{\frac{\beta_k}{2\gamma}} < \sigma$. We then have:

$$s_k^{n_{l_r}-1} > (1-\sigma)|x_k^{n_{l_r}}|.$$

We already saw that for sufficiently large r, and assuming $\Gamma_k > 0$, we had: $|x_k^{n_{l_r}}|^2 \geq \Gamma_k(1-\sigma)(s_k^{n_{l_r}-1})^{2(2-q_k)}$, so that $|x_k^{n_{l_r}}|^{\frac{1}{2-q_k}} \geq \widetilde{\Gamma_k} s_k^{n_{l_r}-1}$. Combined with the previous argument this implies that:

$$|x_k^{n_{l_r}}|^{\frac{1}{2-q_k}} > \widetilde{\Gamma_k}(1-\sigma)|x_k^{n_{l_r}}|.$$

Now, for the case $1 < q_k < 2$: $\frac{1}{2-q_k} > 1$. This means we can divide both sides of the above by $|x_k^{n_{l_r}}|$. This would then imply that $|x_k^{n_{l_r}}| \geq C > 0$ for all r, in contradiction to the condition that $\lim_{r\to\infty} x_k^{n_{l_r}} = 0$. This means that we must have $\Gamma_k = 0$ when $1 < q_k < 2$ and $\lim_{r\to\infty} x_k^{n_{l_r}} = 0$. Recalling that $\frac{1}{q_k\lambda_k}(A^T(b - A\overline{x}))_k = \lim_{r\to\infty} w_k^{n_{l_r}-1} x_k^{n_{l_r}}$, this then implies that:

$$\Gamma_k = \lim_{r\to\infty} (w_k^{n_{l_r}-1} x_k^{n_{l_r}})^2 = 0 \implies \lim_{r\to\infty} w_k^{n_{l_r}-1} x_k^{n_{l_r}} = 0 \implies (A^T(b - A\overline{x}))_k = 0$$

which is the proper optimality condition for this case.

Finally, consider the case $\lim_{r\to\infty} x_k^{n_{l_r}} = 0$ and $q_k = 1$. We then have:

$$\begin{array}{rc} & |x_k^{n_{l_r}}|^{\frac{1}{2-q_k}} > \widetilde{\Gamma_k}(1-\sigma)|x_k^{n_{l_r}}| \\ \implies & |x_k^{n_{l_r}}| > \widetilde{\Gamma_k}(1-\sigma)|x_k^{n_{l_r}}| = \Gamma_k(1-\sigma)^2|x_k^{n_{l_r}}| \\ \implies & \Gamma_k(1-\sigma)^2 < 1. \end{array}$$

This implies that:

$$\Gamma_k = \lim_{r\to\infty} (w_k^{n_{l_r}-1} x_k^{n_{l_r}})^2 \leq 1 \implies \lim_{r\to\infty} |w_k^{n_{l_r}-1} x_k^{n_{l_r}}| \leq 1 \implies \frac{1}{\lambda_k}|(A^T(b - A\overline{x}))| \leq 1$$

which is the condition we need for $q_k = 1$.

It remains to check that $\beta_k = \frac{2}{2-q_k} - q_k\gamma$ satisfies the conditions: $\beta_k > 0$ and $\frac{\beta_k}{2\gamma} > 1$ for our choice of γ:

$$0 < \gamma < \frac{2}{4 - q_k^2}$$

The condition $\frac{\beta_k}{2\gamma} > 1$ is equivalent to $\frac{1}{\gamma(2-q_k)} - \frac{q_k}{2} > 1$ and this implies $\beta_k = \frac{2}{2-q_k} - q_k\gamma > 0$. The two conditions on β_k are thus equivalent, and we need γ to satisfy only:

$$\frac{1}{\gamma(2-q_k)} - \frac{q_k}{2} > 1 \implies \frac{1}{\gamma(2-q_k)} > \frac{2+q_k}{2} \implies \gamma < \frac{2}{4-q_k^2}.$$

Hence, the choice $0 < \gamma < \frac{2}{4-q_k^2}$ satisfies the conditions used in the above proof. □

4.6 More on Convergence

In the previous sections we have shown that for both of the schemes, there exists a converging subsequence and that the limit point of this subsequence is a minimizer of the functional the algorithm is designed to minimize. If the functional is convex (as is the case if all $q_k \geq 1$) then this minimizer is indeed a global minimizer. The subsequence we pick was indeed a special subsequence. It was chosen in a way that put special conditions on the ϵ^n's. We picked a subsequence n_l of the iterates n such that the members of this subsequence satisfy: $\epsilon^{n_l+1} < \epsilon^{n_l}$. However, since the limit of this special subsequence is a minimizer: i.e. $F(\bar{x}) \leq F(x)$, we can show that for any converging subsequence (x^{m_j}), the limit point is also a minimizer of the cost functional.

Lemma 4.6.1. *Suppose that (m_j) is a sequence of strictly increasing positive integers such that the subsequence x^{m_j} from either of the IRLS schemes are convergent, i.e., $x^{m_j} \to \hat{x}$. Then $F(\bar{x}) = F(\hat{x})$, where $\bar{x} = \lim_{l\to\infty} x^{n_{l_r}}$ and (n_{l_r}) is given in Lemma 4.5.4.*

Proof. First note that F is continuous, so we have $F(x^{m_j}) \to F(\hat{x})$. Since $m_j \to \infty$, for every r, there exists j_r such that $m_{j_r} > n_{l_r}$, which implies

$$\begin{aligned} F(x^{m_{j_r}}) &\leq G(x^{m_{j_r}}, x^{m_{j_r}}, w^{m_{j_r}}, \epsilon^{m_{j_r}}) \leq G(x^{n_{l_r}}, x^{n_{l_r}}, w^{n_{l_r}}, \epsilon^{n_{l_r}}) \\ &= ||Ax^{n_{l_r}} - b||_2^2 + 2\sum_{k=1}^{N} \lambda_k((x_k^{n_{l_r}})^2 + (\epsilon^n)^2)^{\frac{q_k}{2}}. \end{aligned}$$

Taking the limit of both sides, we have that:

$$F(\hat{x}) \leq F(\bar{x}) = \min_x F(x),$$

since $\epsilon^n \to 0$. Hence $F(\hat{x}) = F(\bar{x})$. □

Note that in the above lemma, while $F(\tilde{x}) = F(\bar{x})$, it is possible that that $\tilde{x} \neq \bar{x}$. However, in the case when $F(x)$ has a unique minimizer (such as the case when all $q_k > 1$ in the generalized functional) the above result leads to a powerful conclusion. In that case, lemma 4.6.1 says that all converging subsequences go to the minimizer. In fact, this together with boundedness then implies that the iterates themselves converge to the minimizer. The proof relies on the lemma below:

Lemma 4.6.2. *Suppose (x^n) is a bounded sequence in $\mathbb{R}^N$, and all its convergent subsequences converge to the same limit $\bar{x}$. Then (x^n) is a convergent sequence with limit $\bar{x}$.*

Proof. Suppose $x^n \not\to \bar{x}$. Then $\exists \epsilon > 0$ and subsequence x^{n_k} s.t. $|x^{n_k} - \bar{x}| > \epsilon$ for all n_k but x^{n_k} is bounded also, so there exists a convergent subsequence $x^{n_{k_j}}$ which goes to $\bar{x}$. Hence the above is impossible and $x^n \to \bar{x}$. □

We now state a convergence result about the sequence of iterates for the first IRLS algorithm, one that relies only on identifying a single convergent subsequence of the iterates (x^{n_j}) with "finite jumps". By "finite jump", we mean that there exists a constant K such that $|n_{j+1} - n_j| \leq K$ for all j; in other words, the magnitude of jump from one iterate to the next in the subsequence does not go to infinity. Such a subsequence can often be observed in practice.

Lemma 4.6.3. *Let (x^n) be a sequence that satisfies $||x^n - x^{n+1}|| \to 0$ as $n \to \infty$. Suppose that (x^n) contains a subsequence (x^{n_j}) such that $x^{n_j} \to \bar{x}$ as $j \to \infty$ and $|n_{j+1} - n_j| \leq K$ for all j. where $K < \infty$. Then (x^n) is convergent sequence with limit $\bar{x}$.*

Proof. Fix any $\epsilon > 0$. We show that there exists an N_ϵ so that for all $n \geq N_\epsilon$ we have that $||x^n - \bar{x}|| < \epsilon$.

By the assumption that $x^{n_j} \to \bar{x}$ as $j \to \infty$, there exists a constant M_1 such that for $j \geq M_1$, we have that $||x^{n_j} - \bar{x}|| < \frac{\epsilon}{2}$. Also there exists a constant N_2 such that for $n > N_2$, we have $||x^n - x^{n+1}|| < \frac{\epsilon}{2K}$. Take any $N_3 \geq \max(N_2, n_{M_1+1})$; then for any $n > N_3$, pick integer j such that $n_j > n \geq n_{j-1}$, $|n_j - n| \leq n_j - n_{j-1} \leq K$ holds because of the assumption that $|n_{j+1} - n_j| \leq K$ for all j. Hence:

$$||x^{n_j} - x^n||_2 = \sum_{l=1}^{n_j - n} ||x^{n+l} - x^{n+l-1}||_2 \leq \sum_{l=1}^{n_j - n} \frac{\epsilon}{2K} = (n_j - n)\frac{\epsilon}{2K} \leq \frac{\epsilon}{2}.$$

Finally this means that:

$$||x^n - \bar{x}||_2 = ||x^n - x^{n_j} + x^{n_j} - \bar{x}|| \leq ||x^n - x^{n_j}|| + ||x^{n_j} - \bar{x}|| \leq \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon.$$

□

Now we can state a convergence result for the first IRLS algorithm under the "finite jump" assumption.

Proposition 4.6.4. *Let* (x^n) *be a sequence generated by* (IRLS) *with a given* $A \in \mathbb{R}^{m\times N}$ *satisfying* $||A||_2 \leq 1$ *and* $b \in \mathbb{R}^m$. *If there exists a convergent sequence* (x^{m_l}) *of* (x^n) *such that* $|m_{l+1} - m_l| < K$ *for all* l, *where* $K < \infty$, *then* (x^n) *converges to a minimizer of* $F(x)$ *defined in* (4.4.1).

Proof. By Lemma 4.4.3, we have that $||x^n - x^{n-1}|| \to 0$. By Lemma 4.6.1, (x^{m_l}) converges to a minimizer $\hat{x}$ of F. It follows from Lemma 4.6.3 that (x^n) also converges to $\hat{x}$. □

4.7 Chapter Remarks and Conclusions

In this chapter, two new Iteratively Reweighted Least Squares Algorithms were introduced and their convergence properties were thoroughly analyzed. This chapter has several new contributions. First, the algorithms are applied to a more general sparsity promoting functional:

$$F(x) = ||Ax - b||_2^2 + 2\sum_{k=1}^{N} \lambda_k |x_k|^{q_k},$$

for $1 \leq q_k < 2$. The ℓ_1 functional that was previously discussed is a special case of the above functional. Additionally, while convergence has been shown only for $1 \leq q_k < 2$, it is possible to use the algorithms with $q_k < 1$, perhaps starting with convex minimization (i.e. $q_k \geq 1$) and then moving to non-convex minimization later in the iteration. Recently, several papers have confirmed that there are numerical advantages to non-convex sparse minimization, for example [9]. Perhaps the most important contribution of this chapter is the detailed convergence analysis, which does not assume restrictive assumptions on the matrix A, only that its spectral norm be less than one, which is easily accomplished by rescaling. Finally, while the first IRLS scheme is somewhat similar in form to the basic ISTA algorithm, the second scheme is more powerful since at each iteration it involves a linear system solve, so the fact that convergence is established without restrictive assumptions is an important example that such a class of algorithms is possible for the type of problems we consider in this thesis. That the convergence analysis for the second algorithm holds without showing that $||x^{n+1} - x^n||_2 \to 0$ is also of analytical interest. For both algorithms, we showed that when $q_k > 1$ for all k, the sequence of iterates x^n converges to the unique minimizer. We show some numerical expamples with the algorithms in the next chapter.

Chapter 5

NUMERICS AND IDEAS FOR LARGE SCALE PROBLEMS

5.1 Overview

In the past chapters we have listed several different algorithms. These included existing algorithms (such as FISTA, DALM, and Coordinate Descent) and several new algorithms (FIVTA and the two IRLS schemes) which we have introduced and for which we have provided detailed convergence analysis. In this chapter we discuss numerics and some useful ideas for large scale problems, where the matrix A may have many thousands or several millions of columns. We discuss an inverse matrix replacement strategy for the second IRLS scheme where the matrix $(A^TA + \Phi_n)^{-1}$ is replaced by the inverse of a smaller matrix when A is under-determined. We also discuss how the idea of using different weights for different coefficients, introduced in the previous chapter for IRLS, can be extended to other algorithms. In particular we discuss a weighted dual approach (a reweighted norm approach for the DALM algorithm). We make some comments on using coordinate descent for large systems (in particular, we mention support identification and column norm estimation). We also discuss a fast randomized approach for a rank-k SVD approximation.

5.2 Inverse Matrix Replacement for IRLS SYS

The second IRLS scheme involves at each iteration a linear solve:

$$x^{n+1} = (A^T A + \Phi_n)^{-1} A^T b$$

with $(\Phi_n)_{k,k} = q_k \lambda_k w_k^n$, as discussed in the last chapter. We note that between different iterations, the only thing that changes in the above formula is the diagonal portion Φ_n. Some possibilities exist to make use of this fact and we mention one here, that may be of use when A is a large underdetermined matrix. When A is of size $m \times N$ (with $N > m$), the matrix $A^T A$ is of size $N \times N$ and hence very large. This does not present a major challenge unless we would like to compute the explicit formula for this inverse, which in practice we would probably not do. However, we mention here that it is possible to express the inverse of this matrix in terms of a smaller inverse, using the Woodbury matrix inverse identity which we state below:

Lemma 5.2.1. *Take $D \in \mathbb{R}^{n\times n}$, $U \in \mathbb{R}^{n\times k}$, $C \in \mathbb{R}^{k\times k}$, and $V \in \mathbb{R}^{k\times n}$. Assume that D and C are invertible. Then $D + UCV$ is invertible if and only if $C^{-1} + VD^{-1}U$ is, and the following identity holds:*

$$(D + UCV)^{-1} = D^{-1} - D^{-1}U\left(C^{-1} + VD^{-1}U\right)^{-1} VD^{-1}.$$

Proof. The easiest proof of the identity is given using block matrix inversion as suggested in [26]. We give it here for completeness. Consider the block matrix system:

$$\begin{pmatrix} D & U \\ V & -C^{-1} \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} I \\ 0 \end{pmatrix},$$

which reduces to:

$$DX + UY = I \big(\implies X = D^{-1}(I - UY)\big) \text{ and } VX - C^{-1}Y = 0 \, (\implies Y = CVX) \, .$$

Plugging $Y = CVX$ into the first equation we get: $(D + UCV)X = I$ and plugging $X = D^{-1}(I - UY)$ into the second equation we get: $VD^{-1}(I - UY) = C^{-1}Y$. This can be expanded as:

$$\begin{aligned} VD^{-1} - VD^{-1}UY = C^{-1}Y \quad &\implies \quad VD^{-1} = (VD^{-1}U + C^{-1})Y \\ &\implies \quad (VD^{-1}U + C^{-1})^{-1}VD^{-1} = Y. \end{aligned}$$

We now substitute this Y into $DX + UY = I$ to get:

$$DX + U(VD^{-1}U + C^{-1})^{-1}VD^{-1} = I \implies X = D^{-1} - D^{-1}U(VD^{-1}U + C^{-1})^{-1}VD^{-1}.$$

But from $(D + UCV)X = I$ we have that $X = (D + UCV)^{-1}$ so we have the identity:

$$(D + UCV)^{-1} = D^{-1} - D^{-1}U(VD^{-1}U + C^{-1})^{-1}VD^{-1}.$$

□

Now we use the above identity to express the form $(A^TA + \Phi_n)^{-1}A^T$ in terms of a different form which involves the inverse of a much smaller matrix when the matrix A has more columns than rows.

Lemma 5.2.2. *Using the Woodbury matrix inverse formula, we have that:*

$$(A^TA + \Phi_n)^{-1}A^T = (\Phi_n)^{-1}A^T \left(I_m + A(\Phi_n)^{-1}A^T\right)^{-1}.$$

Proof. If D, C and $D + UCV^T$ are invertible, then

$$(D + UCV^T)^{-1} = D^{-1} - D^{-1}U(C^{-1} + V^TD^{-1}U)^{-1}V^TD^{-1}. \tag{5.2.1}$$

In particular, if $C = I_k$, then

$$(D + UV^T)^{-1} = D^{-1} - D^{-1}U(I + V^TD^{-1}U)^{-1}V^TD^{-1}. \tag{5.2.2}$$

Then letting $U = A^T$ and $V^T = A$ we have that:

$$\begin{aligned} & (D + A^TA)^{-1} && = D^{-1} - D^{-1}A^T(I_m + AD^{-1}A^T)^{-1}AD^{-1} \\ \implies & (D + A^TA)^{-1}A^T && = D^{-1}A^T - D^{-1}A^T(I_m + AD^{-1}A^T)^{-1}AD^{-1}A^T \\ & && = D^{-1}A^T(I_m - (I_m + AD^{-1}A^T)^{-1}AD^{-1}A^T). \end{aligned}$$

Notice that:

$$\begin{aligned} & I_m = (I_m + AD^{-1}A^T)^{-1}(I_m + AD^{-1}A^T) \\ & \quad = (I_m + AD^{-1}A^T)^{-1} + (I_m + AD^{-1}A^T)^{-1}AD^{-1}A^T \\ \implies \quad & I_m - (I_m + AD^{-1}A^T)^{-1}AD^{-1}A^T = (I_m + AD^{-1}A^T)^{-1}, \end{aligned}$$

Hence, we have that:

$$(D + A^T A)^{-1} A^T = D^{-1} A^T (I_m + A D^{-1} A^T)^{-1}.$$

Setting $D = \Phi_n$, the result in the lemma follows. $\square$

Thus, when A has more columns than rows and $A \in \mathbb{R}^{m \times N}$ then the expression on the left, which involves an inverse of an $N \times N$ matrix, can be expressed in terms of an inverse matrix of size $m \times m$ which can be significantly smaller when $m < N$.

5.3 Coordinate Descent Method, Support Identification, and Column Norm Estimation

We now discuss some simple ideas for the coordinate descent algorithm, which are important for its application to large scale problems. As previously mentioned in Chapter 1, the idea of the coordinate descent method is that given the objective functional, say $F(x) = F(x_1, \dots, x_N) = ||Ax - b||_2^2 + 2\tau ||x||_1$, we update one coordinate at a time, while fixing all the other coordinates fixed, updating the single chosen coordinate in a way that the objective value of F is decreased. The formula for updating the coordinate x_j so that the ℓ_1 function evaluated at the new vector, with this coordinate changed, has a value less than or equal to the value at the previous iteration is given by:

$$\bar{x}_j = \frac{1}{||A_j||^2} \mathbb{S}_\tau(\beta_j) \quad \text{with} \quad \beta_j = \sum_{l=1}^{m} A_{l,j} \left(b_l - \sum_{k \neq j} A_{l,k} x_k \right).$$

The method should, however, not be programmed as above, since the above formula would be too slow for large problems even if direct access to the matrix elements is available. Instead the expression for β_j can be rewritten as:

$$\beta_j = \sum_{l=1}^{m} (Ae_j)_l \left(b_l - (Ax)_l + (Ae_j)_l x_j\right)$$

which is a form that can be easily programmed. Above, e_j corresponds to the standard unit vector with a 1 at the j-th location and zeros elsewhere. The scheme however, would still be slow if the support cannot be estimated. That's because if the dimension of the solution is large, than the search space of randomized coordinate descent is also just as large. On the other hand, if an accurate estimate of the support is available, then coordinate descent can be used over this presumably much smaller set. The convergence of the coordinate descent algorithm relies on a random sweep pattern to choose the next coordinate to be updated; we have found in experiments however, that when this sweep pattern is restricted to the identified support space, the algorithm has no convergence issues. We now discuss using coordinate descent with the Iterative Support Detection (ISD) idea from [44]. The idea of ISD is to run some algorithm like FISTA for a chosen number of iterations and than take all entries with absolute magnitude above a certain tolerance as the support set. Once this set has been identified we no longer wish to penalize (or threshold) the coefficient in this set and we run the algorithm again penalizing entries only outside this support set.

Algorithm 3: ISD Algorithm

```
$i \leftarrow 0$;
$\mathcal{J} \leftarrow \{1, 2, \ldots, n\}$;
repeat
    $x^{(i)} \leftarrow \arg\min_x \{\|Ax - b\|_2^2 + 2\tau \|x_{\mathcal{J}}\|_1\}$;
    Update $\epsilon$;
    $\mathcal{J} \leftarrow \{j : |x_j^{(i)}| < \epsilon\}$;
until converged;
```

The main step of the ISD algorithm $x^{(i)} \leftarrow \arg\min_x \{\|Ax - b\|_2^2 + 2\tau \|x_{\mathcal{J}}\|_1\}$ can be implemented in a simple way using the IRLS scheme. We use the generalized functional version:

$$||Ax - b||_2^2 + 2\sum_{k=1}^{N} \lambda_k |x_k|^{q_k}$$

with $q_k = 1$ and $\lambda_k = \tau$ only for $k \in \mathcal{J}$, otherwise set $\lambda_k = 0$. We refer to such an algorithm as IRLS ISD. The ISD scheme critically depends on the update rule for ϵ. Two simple formulations are proposed in the above paper. The first is simply:

$$\epsilon \leftarrow \frac{\|x^{(i)}\|_\infty}{3^i}$$

which often leads to acceptable results. On the other hand, it is rather very arbitrary. Another rule is to look for the smallest (component number) k such that $|(x^{(s)})_{k+1}| - |(x^{(s)})_k| > \tau$ where $x^{(s)}$ is the sorted sequence of the solution at some iterate, sorted by absolute magnitude. We then set $\epsilon = |(x^{(s)})_k|$. This scheme is advertised to perform well mostly for signals with a fast decaying distribution of nonzero values, although we have seen that this scheme can be used on various different signals. The connection with coordinate descent is that at the end of this procedure, it can be used on the set J^C, the complement of J to improve on the ISD solution. The simple algorithm is stated below.

Algorithm 4: ISD Algorithm with M steps of Coordinate Descent

```
$i \leftarrow 0$;
$\mathcal{J} \leftarrow \{1, 2, \ldots, n\}$;
**repeat**
    $x^{(i)} \leftarrow \arg\min_x \{\|Ax - b\|_2^2 + 2\tau \|x_{\mathcal{J}}\|_1\}$;
    $\epsilon \leftarrow \frac{\|x^{(i)}\|_\infty}{3^i}$;
    $\mathcal{J} \leftarrow \{j : |x_j^{(i)}| < \epsilon\}$;
**until** *converged or termination condition reached*;
$x \leftarrow x^{(i)}$;
$u \leftarrow Ax$;
**for** $n = 1, 2, \ldots, M$ **do**
    Pick random index $j$ from $\mathcal{J}^C$;
    $v = Ae_j$;
    $\beta_j = \sum_{l=1}^{M} v_l \left(b_l - u_l + v_l x_j\right)$;
    $x_j = \frac{1}{||A_j||^2} S_\tau(\beta_j)$;
**end**
$\bar{x} \leftarrow x$
```

We now discuss another issue related to applying the coordinate descent scheme to large problems: the computation of the column norms $||Ae_j||_2$. When the matrix has several millions of columns, such as the case in our application, computing explicitly that many column norms becomes time consuming and difficult. If the matrix is not explicitly available, but is given in the form $M = AW^{-1}$, as in the application in the next chapter, W being a chosen wavelet transform, then computing the column norms is even more expensive since in computing $||Me_j||_2 = ||AW^{-1}e_j||_2$ we must apply for every j the Wavelet transform. If the matrix has a lot of columns this approach is not practical. In practice, however, we do not need to know the column norms exactly, but only approximately. We now describe a randomized procedure for approximating the column norms. Let $v_j = AW^{-1}e_j$ with $v_j \in \mathbb{R}^m$ if $A \in \mathbb{R}^{m \times N}$. Now take m basis vectors u_i which span $\mathbb{R}^m$. Then we can write:

$$v_j = c_1 u_1 + \cdots + c_m u_m \quad \text{with} \quad c_i = \langle v_j, u_i \rangle.$$

Then we have that:

$$||v_j||^2 = \sum_{i=1}^{m} |\langle v_j, u_i\rangle|^2.$$

Now we use the approximation:

$$||v_j||^2 \approx \frac{m}{K}\sum_{i=1}^{K} |\langle v_j, u_i\rangle|^2$$

where we only use K basis vectors instead of m and scale appropriately. Substituting for v_j, we have:

$$||AW^{-1}e_j||^2 \approx \frac{m}{K}\sum_{i=1}^{K} |\langle AW^{-1}e_j, u_i\rangle|^2.$$

Now we can make use of the inner product relation:

$$\langle AW^{-1}e_j, u_i\rangle = \langle W^{-1,t}A^t u_i, e_j\rangle = \left[W^{-1,t}A^t u_i\right]_j$$

where $[\dots]_j$ refers to the j-th column of the vector. This means that to approximate any column norm, we must compute K vectors: $p_i = W^{-1,t}A^t u_i$ for $i = 1, \dots, K$ and then the $j - th$ norm is given approximately by:

$$||AW^{-1}e_j||^2 \approx \frac{m}{K}\sum_{i=1}^{K} [p_i]_j^2$$

For the basis vectors u_i we can simply take random, normalized vectors in $\mathbb{R}^N$ since such vectors (if N is large enough) are likely to be nearly orthogonal. The number of K matrix vector multiplications required for the estimation varies with the matrix, but in our experience, can be some multiple of $\log(m)$.

5.4 Variable Penalty Idea Applied to a Dual Space Algorithm

In the previous chapter, we showed detailed analysis for two IRLS schemes which minimize the more general sparsity promoting functional:

$$||Ax - b||_2^2 + 2\sum_{k=1}^{N} \lambda_k |x_k|^{q_k} \text{ with } 1 \leq q_k < 2.$$

The advantage of this functional is that it allows us to penalize different coefficients of the vector in different ways and assuming we know something about the underlying structure of the sparse solution (for example when it's expressed with wavelets), we can use this capability to our advantage. This idea, however, can be applied also to other algorithms. The difficult part is to show that the convergence results still hold, which we do not attempt to do here, but we do show that it is possible to apply the reweighted idea to a dual space method and since DALM is known in practice to be fast, we expect the presented method to work well numerically. We start first with the ℓ_1-norm and derive the dual of the weighted ℓ_1-norm. Below we show that it is given by a weighted ℓ_∞-norm, a result which gives us an idea of how to apply the DALM scheme for weighted norm minimization.

Lemma 5.4.1. *Let $w \in \mathbb{R}^N$ be positive, and consider the weighted ℓ_1 norm defined by:*

$$\|x\|_{1,w} := \sum_{i=1}^{N} w_i |x_i| \quad \forall\, x \in \mathbb{R}^N.$$

Then the dual of $\| \cdot \|_{1,w}$ is given by:

$$\|y\|_{\infty,w^{-1}} := \max_i \left\{ \left| \frac{1}{w_i} y_i \right| \right\}.$$

Proof. Fix any $y \in \mathbb{R}^N$, and consider

$$\alpha := \max_{x \in \mathbb{R}^N} \frac{\langle x, y \rangle}{\|x\|_{1,w}}.$$

For any $x \in \mathbb{R}^N$,

$$\langle x, y \rangle \leq \sum_{i=1}^{N} |w_i x_i| \left| \frac{1}{w_i} y_i \right| \leq \max_i \left\{ \left| \frac{1}{w_i} y_i \right| \right\} \|x\|_{1,w},$$

so $\alpha \leq \max_i \left\{ \left| \frac{1}{w_i} y_i \right| \right\}$. To show equality, consider an example where k is such that $\left| \frac{1}{w_k} y_k \right| = \max_i \left\{ \left| \frac{1}{w_i} y_i \right| \right\}$. Define

$$x_i^* = \begin{cases} 0 & \text{if } i \neq k, \\ \dfrac{\operatorname{sgn}(y_k)}{w_k} & \text{if } i = k. \end{cases}$$

Then $\|x^*\|_{1,w} = 1$ and

$$\langle x^*, y \rangle = \frac{\operatorname{sgn}(y_k)}{w_k} y_k = \left| \frac{1}{w_k} y_k \right| = \max_i \left\{ \left| \frac{1}{w_i} y_i \right| \right\}.$$

Thus, we have:

$$\alpha = \frac{\langle x^*, y \rangle}{\|x^*\|_{1,w}} = \max_i \left\{ \left| \frac{1}{w_i} y_i \right| \right\}.$$

Therefore the dual norm of $\| \cdot \|_{1,w}$ is given by

$$\|y\|_{\infty, w^{-1}} := \max_i \left\{ \left| \frac{1}{w_i} y_i \right| \right\}.$$

□

We recall from Chapter 1, that the dual space approach involves an ℓ_∞ norm. Thus, based on the above result, that norm should in turn be replaced by a weighted norm.

Now, instead of the ℓ_1 constrained minimization problem, consider the weighted problem:

$$\min_x \ \|x\|_{1,w} \quad \text{s.t.} \quad Ax = b. \tag{5.4.1}$$

We will now perform similar analysis to Chapter 1 and derive the iteration for the weighted problem. We already have an idea of what it should be based on the above result. Commenting now on the weights, we can express some power of $|x|$ as:

$$|x|^q = w|x| \implies w = |x|^{q-1} = \left(\sqrt{x^2}\right)^{q-1}.$$

Thus, to allow for zero entries, we can take the weights to be:

$$w_k^n = \left(\sqrt{(x_k^n)^2 + (\epsilon_n)^2}\right)^{q_k - 1}$$

with some choice of $\epsilon_n \to 0$. Proceeding with the analysis as before, we will have the Lagrangian:

$$L(x, y) = \|x\|_{1,w} + y^T(b - Ax).$$

For each fixed y, we need to compute $\min_x L(x, y)$. Since the function $L(\cdot, y)$ is separable, we have

$$\begin{aligned}
\min_x L(x,y) &= b^T y + \min_x \sum_{i=1}^N w_i|x_i| - (A^T y)_i x_i \\
&= b^T y + w_i \sum_{i=1}^N \min_{x_i} \left\{ |x_i| - \frac{(A^T y)_i}{w_i} x_i \right\} \\
&= \begin{cases} b^T y & \text{if } \left|\frac{(A^T y)_i}{w_i}\right| \leq 1 \ \forall\, i = 1, \ldots, N \\ -\infty & \text{otherwise.} \end{cases}
\end{aligned}$$

where the last equality follows using (2.5.2).Hence the dual of (5.4.1) is

$$\max_y \; b^T y \quad \text{s.t.} \quad \left| \frac{(A^T y)_i}{w_i} \right| \leq 1,$$

or equivalently,

$$\max_y \; b^T y \quad \text{s.t.} \quad \|A^T y\|_{\infty, w^{-1}} \leq 1. \tag{5.4.2}$$

Using the same analysis as in Chapter 2, we arrive at:

$$\min_{x,y,z} \; L_\mu(y, z; x) := -b^T y - x^T(z - A^T y) + \frac{\mu}{2}\|z - A^T y\|_2^2 \quad \text{s.t.} \quad \|z\|_{\infty, w^{-1}} \leq 1. \tag{5.4.3}$$

The minimization with respect to x and y and the corresponding update schemes remain the same as before. However, the minimization with respect to z will now involve a projection over a different interval. When x^n and y^n are fixed we have:

$$\begin{aligned} & \min_z \{L_\mu(y^n, z; y^n) : \|z\|_{\infty, w^{-1}} \leq 1\} \\ = \;& -b^T y^n + (y^n)^T A^T y^n + \min_z \left\{ \frac{\mu}{2}\|z - A^T y^n\|_2^2 - (y^n)^T z : \|z\|_{\infty, w^{-1}} \leq 1 \right\} \\ = \;& -b^T y^n + (y^n)^T A^T y^n + \sum_{i=1}^{N} \min_{z_i} \left\{ \frac{\mu}{2}(z_i - (A^T y^n)_i)^2 - y_i^n z_i : |z_i| \leq w_i \right\}, \end{aligned}$$

so that:

$$\begin{aligned} & \arg\min_z \{L_\mu(y^n, z; y^n) : \|z\|_{\infty, w^{-1}} \leq 1\} \\ = \;& \left\{ \bar{u} : \forall\, i,\; \bar{u}_i = \mathbb{P}_{[-w_i, w_i]} \left(\arg\min_{z_i} \left\{ \frac{\mu}{2}(z_i - (A^T y^n)_i)^2 - y_i^n z_i \right\} \right) \right\} \\ = \;& \left\{ \bar{u} : \forall\, i,\; \bar{u}_i = \mathbb{P}_{[-w_i, w_i]} \left(\frac{y_i^n}{\mu} + (A^T y^n)_i \right) \right\}. \end{aligned}$$

Numerically, the following algorithm may then be used:

Algorithm 5: ALM Algorithm for the Dual of the Reweighted ℓ_1 norm.

```
Input  : An $m \times N$ matrix $A$ with $||A|| < 1$, an initial guess $N \times 1$ vector
         $x^0$, a parameter $\tau < \big(\max_i(|(A^Tb)_i|)\big)^{2-p}$, tolerance $\gamma$, the maximum
         number of iterations $M$, a set of weights $q_k \in \mathbb{R}$ for $k = [1, \ldots, N]$.
Output: A sparse vector $\bar{x}$ with small $||A\bar{x} - b||_2$.

$\epsilon \leftarrow 1$;
$w^0 \leftarrow (1, \ldots, 1)$;
for $n = 0, 1, \ldots, M$ do
    for $k = 1, \ldots, N$ do
        $z_k^{n+1} = \mathcal{P}_{[-w_k^n, w_k^n]}\left(\frac{1}{\beta}x_k^n + (A^Ty^n)_k\right)$;
    end
    $y^{n+1} = (AA^T)^{-1}\left(Az^{n+1} - \frac{(Ax^n - b)}{\beta}\right)$;
    $x^{n+1} = x^n - \beta(z^{n+1} - A^Ty^{n+1})$;
    for $k = 1, \ldots, N$ do
        $w_k^{n+1} = \sqrt{(x_k^{n+1})^2 + (\epsilon^2)}^{q_k - 1}$;
    end
    $\epsilon = \sqrt{||x^{n+1} - x^n||_2}$;
    if $\|x^n - x^{n+1}\| \leq \gamma$ then
        break
    end
end
$\bar{x} = x^{n+1}$;
```

The linear solve step above $y^{n+1} = (AA^T)^{-1}\left(Az^{n+1} - \frac{(Ax^n - b)}{\beta}\right)$ may again be approximated by one or more iterations of the conjugate gradient scheme.

5.5 Regularization Parameter Estimation

We now discuss the choice of regularization parameter (such as τ in the case of minimizing the ℓ_1 functional $||Ax - b||_2^2 + 2\tau||x||_1$). In general, the regularization algorithms discussed in this thesis are useful for noisy right hand sides b, i.e. $b = b_t + \text{noise}$. While the true, uncorrupted, b_t may not be known, an estimate of the noise norm $\nu = ||\text{noise}||_2$ is often available. In this case, τ is chosen such that the corresponding solution x_τ satisfies $||Ax_\tau - b|| \approx \nu$. This is accomplished by means of a continuation strategy outlined below. This strategy is fast since we reuse the initial guess at each step as we vary the τ starting from the known minimizer $x = 0$ at $\tau = ||A^T b||_\infty$:

Algorithm 6: Continuation strategy for picking τ

```
Input  : An $m \times N$ matrix $A$, an $m$-vector $b$, the maximum number of outer
         iterations $M$ and an estimate of the noise norm $\nu \approx ||\text{noise}||_2$.
Output: An estimate of $\bar{\tau}$ such that $||Ax_{\bar{\tau}} - b||_2 \approx ||\nu||_2$
$\tau_{\max} = ||A^T b||_\infty$;
$\tau_{\min} = \frac{||A^T b||_\infty}{10000}$;
$S = \frac{\log \tau_{\max} - \log \tau_{\min}}{M - 1}$;
$x^0 = 0$;
for $i = 1, \ldots, M$ do
    $\tau_i = e^{(\log \tau_{\max} - S(i-1))}$;
    $x^{(i)} \leftarrow \arg\min_x \{\|Ax - b\|_2^2 + 2\tau_i \|x\|_1\}$;   % use $x^{(i-1)}$ as initial point when
    solving for $x^{(i)}$
end
$j \leftarrow \arg\min_i \left\{ \left| \|Ax^{(i)} - b\|_2^2 - \nu^2 \right| : 1 \leq i \leq M \right\}$;
$\bar{\tau} \leftarrow \tau_j$;
```

We now describe a technique that may be useful for the estimation of the regularization parameter when an estimate of the norm of the noise is not available. In this case, one idea is to use the L-curve method, i.e., finding the point of maximum curvature of the $(\log ||Ax_\tau - b||_2, \log ||x_\tau||_1)$ plot to estimate the right τ [28]. Defining:

$$\bar{\epsilon} = \log ||x_\tau||_1 \quad \text{and} \quad \bar{\rho} = \log ||Ax_\tau - b||_2.$$

We can then compute the curvature by the formula:

$$\bar{c_\tau} = 2\frac{\bar{\rho}'\bar{\epsilon}'' - \bar{\rho}''\bar{\epsilon}'}{((\bar{\rho}')^2 + (\bar{\epsilon}')^2)^{\frac{3}{2}}}$$

which can be approximated via finite differences. What we can observe is that the point of maximum curvature of the log-log plot corresponds roughly to the region where the solution (with that particular τ) has lowest percent error. Consider for example the curvature plotted for a simple example using the FISTA algorithm:

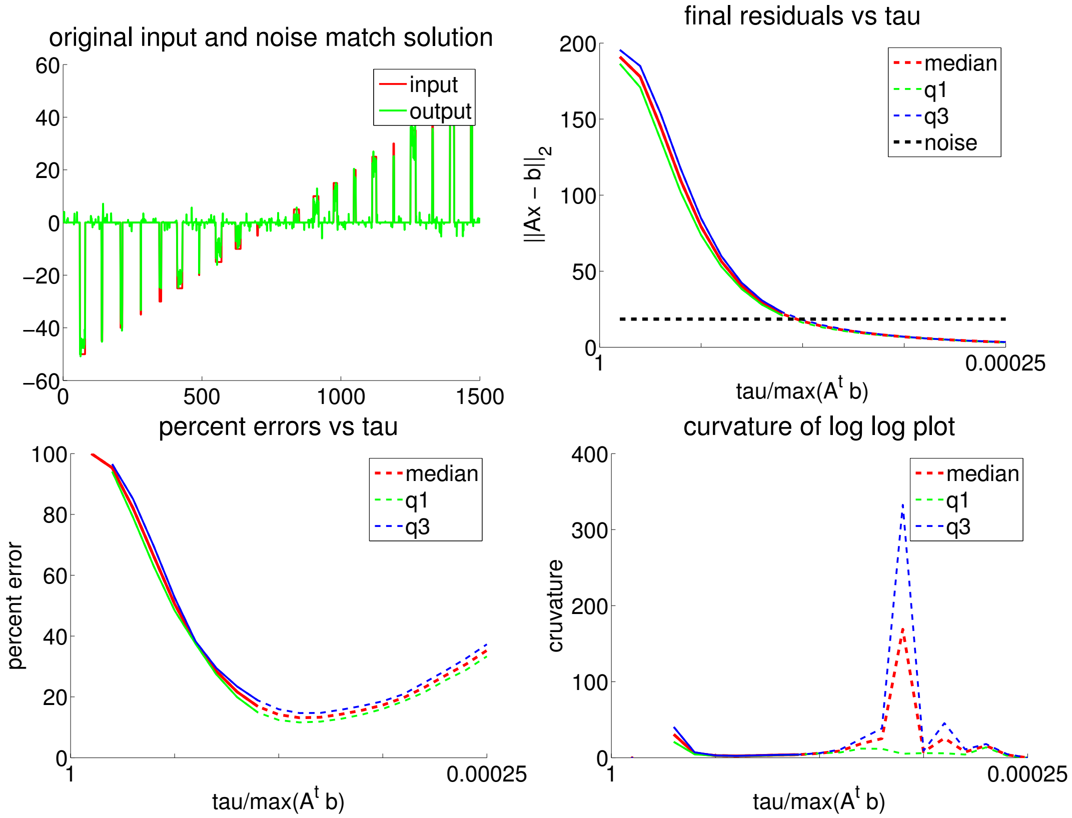


Figure 5.1: Well conditioned staircase input and output at noise matching τ, residuals versus noise, percent errors versus τ, and curvature versus τ (observe that the lowest point of percent error curve is roughly where the curvature is highest).

5.6 Randomized Low Rank Approximation

In this section, we discuss an efficient implementation of a randomized low rank approximation algorithm. Such an algorithm is useful when we have to repeatedly apply a very large not well conditioned matrix to vectors. The idea is to approximate such a large matrix by the rank-k SVD approximation: $A \approx U\Sigma_k V^T$ where Σ_k contains the largest k singular values of A and U and V are orthogonal matrices. For a large $m \times N$ matrix A, the sizes of U, Σ_k, and V would be $m \times k$, $k \times k$, and $N \times k$. Once these three matrices are obtained, matrix vector products with A can be approximated as:

$$Av \approx U(\Sigma_k(V^T v)).$$

When A is a very large under-determined matrix (i.e. $m << N$) significant cost savings can be obtained by making this approximation since k can be significantly less than $\min(m, N)$. In general, the computation of the SVD is a very expensive procedure and cannot be applied to very large matrices. The goal of this section is to state a randomized algorithm that can be applied to very large matrices and produce a rank-k SVD approximation in reasonable time. This new algorithm will be an adaptation of an existing method proposed by Martinsson et al. in [27]. We first state the algorithm proposed in the above paper:

Algorithm 7: Computing a rank-k SVD approximation $U\Sigma V^T$ of matrix A

```
Input  : $m \times N$ matrix $A$,
         $k$ : the desired rank for approximating $A$.
Output: $m \times k$ matrix $U$ satisfying $U^TU = I_k$,
         $N \times k$ matrix $V$ satisfying $V^TV = I_k$,
         $k \times k$ diagonal matrix $\Sigma$ with nonnegative diagonal entries.

  1. Draw an $N \times k$ Gaussian random matrix $\Omega$.
     Omega = randn(n,k)

  2. Form the $m \times k$ sample matrix $Y = A\Omega$.
     Y = A * Omega

  3. Form an $m \times k$ orthonormal matrix $Q$ such that $Y = QR$.
     [Q, R] = qr(Y)

  4. Form the $k \times N$ matrix $Q^TA$.
     B = Q' * A

  5. Compute the SVD of the small matrix $B$: $B = \hat{U}\Sigma V^T$.
     [Uhat, Sigma, V] = svd(B)

  6. Form the matrix $U = Q\hat{U}$.
     U = Q * Uhat
```

We make some simple modifications to the above algorithm so that it is easier to apply to large under-determined matrices. The first three steps can be replaced by a Gram-Schmidt Orthogonalization procedure, so that forming Ω and Y becomes unnecessary, as is carrying out the QR factorization. Instead, we are able to build up Q directly. Finally, we would like to perform the singular value decomposition on the matrix BB^T which will be of a small size $k \times k$. Computing the eigenvectors V is then possible through the following relations:

$$B = U\Sigma V^T = \sum_{i=1}^{k} \sigma_i u_i v_i^T \quad \text{and} \quad Bv_i = \sigma_i u_i.$$

The vectors v_i can be computed from u_i using the relation:

$$v_i = \frac{1}{\sqrt{\beta_i}} B^T u_i$$

where the β_i are eigenvalues of BB^T and u_i the corresponding eigenvectors. The faster version of the above algorithm thus involves computations with the small $k \times k$ matrix BB^T. The above relations are based on the standard lemmas presented below, of which we include proofs for the sake of completeness.

Lemma 5.6.1. *Let $A \in \mathbb{R}^{m \times p}$ and $B \in \mathbb{R}^{m \times n}$. Then* $\text{range}(A) \subseteq \text{range}(B)$ *if and only if $A = BR$ for some $R \in \mathbb{R}^{n \times p}$.*

Proof. If $\text{range}(A) \subseteq \text{range}(B)$, then for each $i = 1, \ldots, p$, the i-th column of A given by $a_i = Ae_i$ lies in the range of B. Hence $a_i = Br_i$ for some $r_i \in \mathbb{R}^n$. We can write:

$$\begin{aligned} A &= \left(\begin{bmatrix} a_1 & a_2 & \cdots & a_p \end{bmatrix}\right) = \left(\begin{bmatrix} Br_1 & Br_2 & \cdots & Br_p \end{bmatrix}\right) = B\left(\begin{bmatrix} r_1 & r_2 & \cdots & r_p \end{bmatrix}\right) \\ &= BR. \end{aligned}$$

For the converse, suppose that $A = BR$ for some $R \in \mathbb{R}^{n \times p}$. If $y \in \text{range}(A)$, then there exists $x \in \mathbb{R}^p$ such that $y = Ax = B(Rx)$, so $y \in \text{range}(B)$. Hence $\text{range}(A) \subseteq \text{range}(B)$. □

Lemma 5.6.2. *Let $B \in \mathbb{R}^{m \times n}$. Then* $\text{range}(BB^T) = \text{range}(B)$.

Proof. Suppose $y \in \text{range}(BB^T)$. That means there exists x such that $y = BB^Tx = B(B^Tx)$, so that $y \in range(B)$ also. Hence, $\text{range}(BB^T) \subseteq \text{range}(B)$. It remains to show that $\text{range}(B) \subseteq \text{range}(BB^T)$. Suppose that $x \in \ker(BB^T)$. This means that $BB^Tx = 0$. Next:

$$||B^Tx||_2^2 = (B^Tx)^T(B^Tx) = x^TBB^Tx = x^T(BB^Tx) = x^T0 = 0$$

So we have that: $||B^T x||_2 = 0$ which implies that $B^T x = 0$, which means that $x \in \ker(B^T)$. Hence we have:

$$\ker(BB^T) \subseteq \ker(B^T) \implies (\ker(B^T))^\perp \subseteq (\ker(BB^T))^\perp$$

which implies:

$$\text{range}(B) = (\ker(B^T))^\perp \subseteq (\ker(BB^T))^\perp = \text{range}(BB^T).$$

□

Proposition 5.6.3. *Let $B \in \mathbb{R}^{k \times n}$ and assume that B has full rank.*

Let $UDU^T \in \mathbb{R}^{m \times m}$ be the spectral decomposition of BB^T, that is, $U^T U = I_k$ and $D = \text{Diag}(\beta_1, \beta_2, \dots, \beta_k)$ is diagonal, with eigenvalues of BB^T given by $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_k > 0$. Then defining

$$\begin{aligned} \Sigma &= \text{Diag}(\sqrt{\beta_1}, \sqrt{\beta_2}, \dots, \sqrt{\beta_k}), \\ v_i &= \frac{1}{\sqrt{\beta_i}} B^T u_i \in \mathbb{R}^n \quad \textit{for } i = 1, \dots, k, \quad \textit{where } U = \left(\begin{bmatrix} u_1 & u_2 & \cdots & u_k \end{bmatrix} \right), \\ V &= \left(\begin{bmatrix} v_1 & v_2 & \cdots & v_k \end{bmatrix} \right) \in \mathbb{R}^{n \times k}, \end{aligned}$$

we have that:

1. *V is orthogonal, that is, $V^T V = I_k$, and*

2. *$B = U \Sigma V^T$.*

Proof. For any $1 \leq i \leq k$, $U^T u_i = e_i \in \mathbb{R}^k$ since $U^T U = I_k$. Hence for any $1 \leq i, j \leq k$,

$$\begin{aligned}(V^TV)_{ij} &= v_i^T v_j = \frac{1}{\sqrt{\beta_i \beta_j}} u_i^T BB^T u_j = \frac{1}{\sqrt{\beta_i \beta_j}} u_i^T UDU^T u_j = \frac{1}{\sqrt{\beta_i \beta_j}} e_i^T D e_j \\ &= \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}\end{aligned}$$

Hence $V^T V = I_k$.

Next, we have that $\text{range}(BB^T) \subseteq \text{range}(U)$. This follows since by assumption $BB^T = UDU^T$. This means that $y \in \text{range}(BB^T) \implies y = BB^T x = U(DU^T)x \implies y \in \text{range}(U)$. By lemma 5.6.2 and lemma 5.6.1 we have that:

$$\text{range}(B) = \text{range}(BB^T) \subseteq \text{range}(U) \implies \text{range}(B) \subseteq \text{range}(U) \implies B = UR$$

for some matrix $R \in \mathbb{R}^{k \times n}$. Next, since $\frac{1}{\sqrt{\beta_i}} U e_i = \frac{1}{\sqrt{\beta_i}} u_i$ this implies that:

$$\begin{aligned} U\Sigma^{-1} &= U \left(\begin{bmatrix} \frac{1}{\sqrt{\beta_1}} e_1 & \frac{1}{\sqrt{\beta_2}} e_2 & \cdots & \frac{1}{\sqrt{\beta_k}} e_k \end{bmatrix} \right) \\ &= \left(\begin{bmatrix} \frac{1}{\sqrt{\beta_1}} u_1 & \frac{1}{\sqrt{\beta_2}} u_2 & \cdots & \frac{1}{\sqrt{\beta_k}} u_k \end{bmatrix} \right) \\ \implies \quad B^T U \Sigma^{-1} &= \left(\begin{bmatrix} \frac{1}{\sqrt{\beta_1}} B^T u_1 & \frac{1}{\sqrt{\beta_2}} B^T u_2 & \cdots & \frac{1}{\sqrt{\beta_k}} B^T u_k \end{bmatrix} \right) = V \\ \implies \quad & R^T = (R^T U^T) U = B^T U = V\Sigma \\ \implies \quad & R = (V\Sigma)^T = \Sigma V^T \\ \implies \quad & B = UR = U\Sigma V^T. \end{aligned}$$

□

The modified algorithm then becomes:

Algorithm 8: FAST RSVD: Computing a rank-k SVD approximation $U\Sigma V^T$ of matrix A

```
Input  : $m \times N$ matrix $A$,
         $k$ : the desired rank for approximating $A$.
Output : $m \times k$ matrix $U$ satisfying $U^T U = I_k$,
         $N \times k$ matrix $V$ satisfying $V^T V = I_k$,
         $k \times k$ diagonal matrix $\Sigma$ with nonnegative diagonal entries.
Form the matrix $Q$ by computing columns $q_r$ for $r \in (1, \dots, k)$:
for $r = 1, 2, \dots, k$ do
    $p_r = \texttt{rand}(n, 1)$;
    $y_r = Ap_r$;
end
$q_1 = \frac{y_1}{||y_1||}$;
for $r = 2, \dots, k$ do
    $q_r = \frac{y_r - \sum_{j=1}^{r-1} \langle y_r, q_j \rangle q_j}{\left\| y_r - \sum_{j=1}^{r-1} \langle y_r, q_j \rangle q_j \right\|_2}$;
    $Q(:, r) = q_r$;
end
Form the $k \times k$ matrix $BB^T$ and take its SVD:
$B = Q^T A$;
Compute eigenvectors $\bar{U}$ and matrix of eigenvalues $D$ of $BB^T$:
$[\bar{U}, D] = \texttt{eigs}(BB^T)$;
Compute eigenvalues matrix $\Sigma$:
for $r = 1, 2, \dots, k$ do
    $\Sigma_{r,r} = \sqrt{D_{r,r}}$;
end
Compute the $n \times k$ matrix $V$:
for $r = 1, 2, \dots, k$ do
    $v_r = \frac{1}{\Sigma_{r,r}} A^T Q \bar{u}_r$;
    $V(:, r) = v_r$;
end
Compute the $m \times k$ matrix $U$:
$U = Q\bar{U}$;
```

Below, we present some plots that show the quality of a rank-k SVD approximation for differently conditioned matrices of size 100 × 500. That is, for three differently conditioned matrices A_1, A_2, and A_3, we plot the percent errors $100\frac{||A-U_k\Sigma_k V_k^T||_2}{||A||_2}$ as a function of k. What we observe is what we expect: that for matrices for which the singular values fall of rapidly a small k is sufficient to get a good approximation. On the other hand, for well conditioned matrices, where the singular values fall off slowly, a large k is necessary.

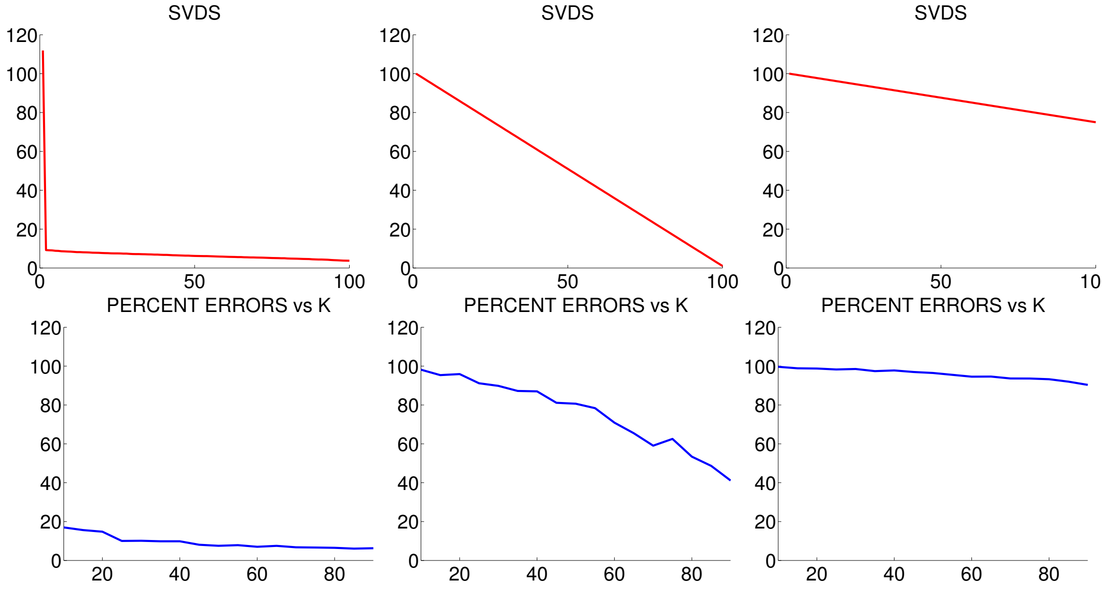


Figure 5.2: Singular value distributions and percent errors as a function of k. We may observe that for matrices with fast decaying singular value distributions, only a small k is necessary to obtain a good low rank approximation.

5.7 Numerical Comparisons

In this section we perform numerical tests with some of the algorithms which we have considered. We pick the fastest of the existing schemes: FISTA and DALM and compare their performance against FIVTA and the two IRLS schemes on simple examples with well and not well conditioned matrices. In each case, we start with a sparse signal x, and for each run, we make a matrix A with a certain singular value distribution, explicitly controlled by using the reverse SVD procedure to construct $A = U\Sigma V^T$, where the orthogonal matrices U and V are generated via a QR decomposition of a random matrix. We take an input x and then run several runs with the different algorithms. For each run within an example, Σ is kept fixed, but the U and V are different so the different systems are slightly different. Then we compute the median and first and third quartiles of different quantities from different runs. We do this for two types of conditioning of A: one where the singular values fall off linearly and another in which they fall off in a strongly non-linear fashion. In each case, some noise is added to the right hand side b. The solution we plot is at the so called noise level: we look for a solution such that its $||Ax - b||_2$ value is close to the norm of the noise. For this we utilize the approach for choosing the regularization parameter described earlier in this chapter. We now show two different singular value distributions and a sparse input on which we run reconstructions with the different algorithms.

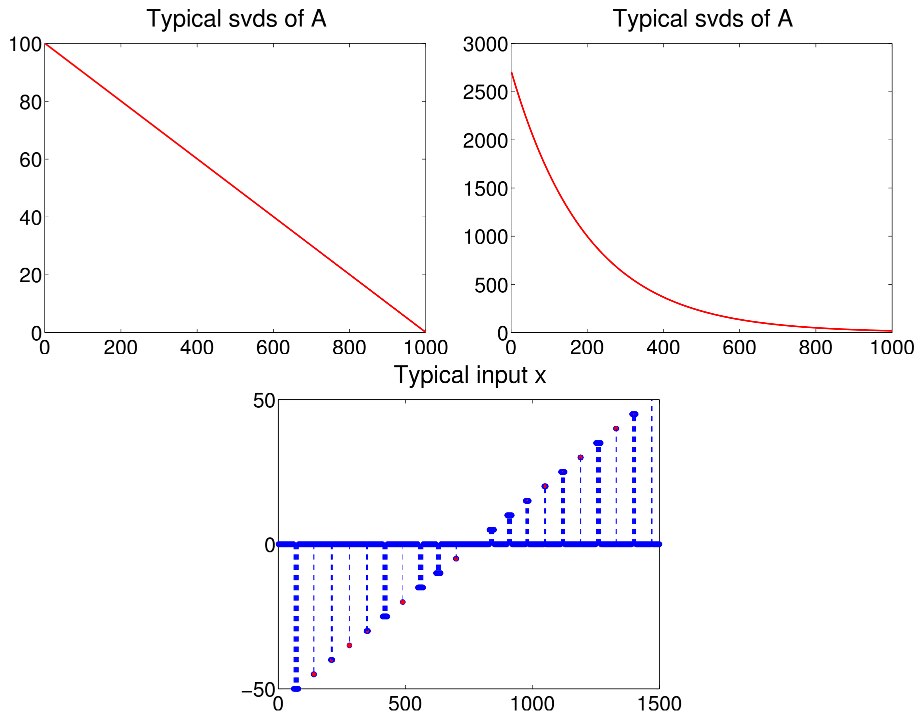


Figure 5.3: Two distributions of singular values (well conditioned and worser conditioned) and a staircase arranged sparse input.

In the two sets of figures below, we plot several quantities for the above examples (a well conditioned and a worder conditioned staircase input). We plot the residuals $||Ax - b||_2$ versus the regularization parameter τ, percent errors between the reconstruction and the original versus τ and the solution at the τ for which $||Ax - b||_2$ is close to the norm of the noise. We plot the quantities one row at a time, each corresponding to a different algorithm: FISTA, DALM, FIVTA, IRLS, and IRLS SYS from the top down. The IRLS methods are used with $q_k = 1$ for all k. We observe below that the behavior of these quantities are quite similar for the different algorithms.

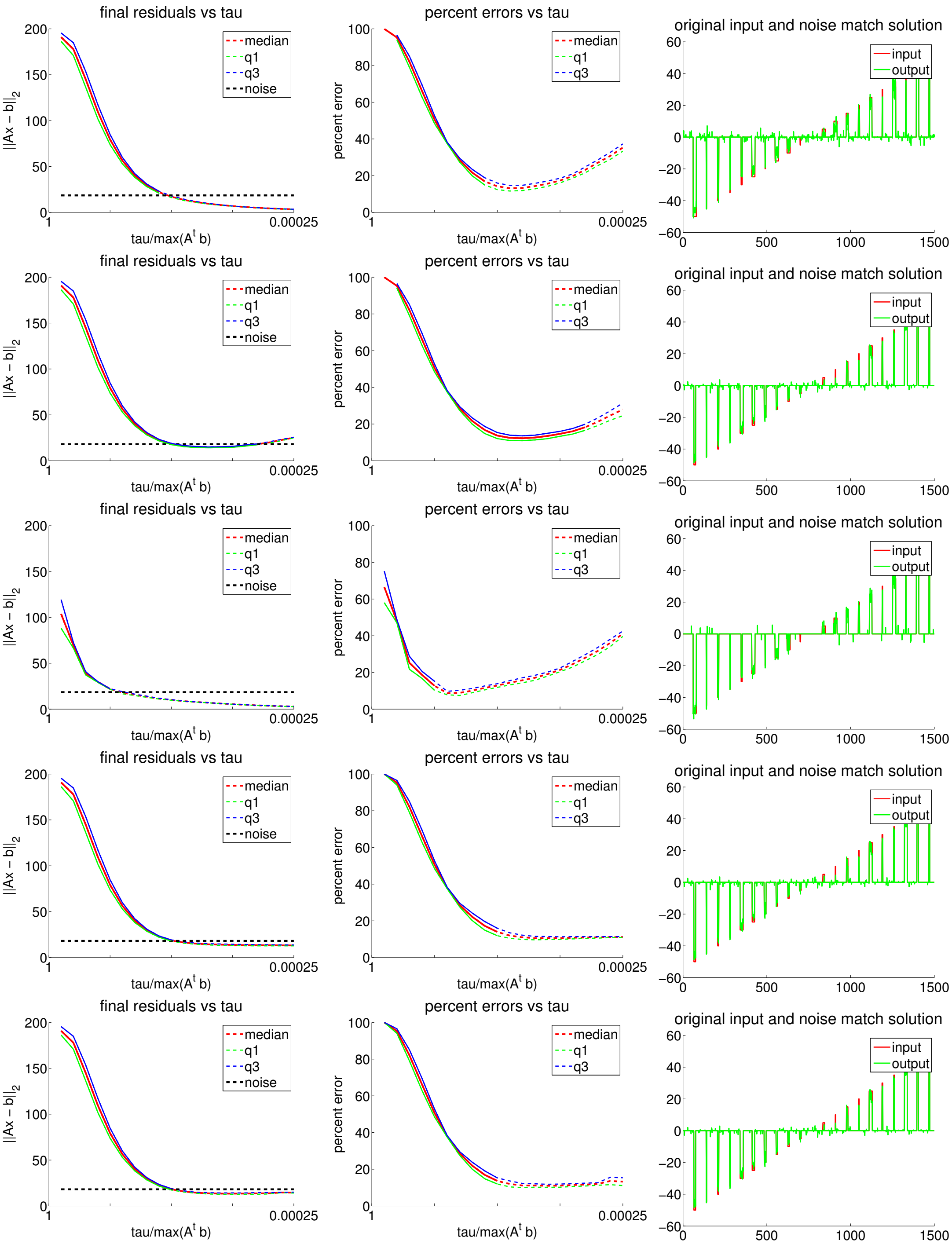


Figure 5.4: Well conditioned staircase. FISTA, DALM, FIVTA, IRLS, IRLS SYS.

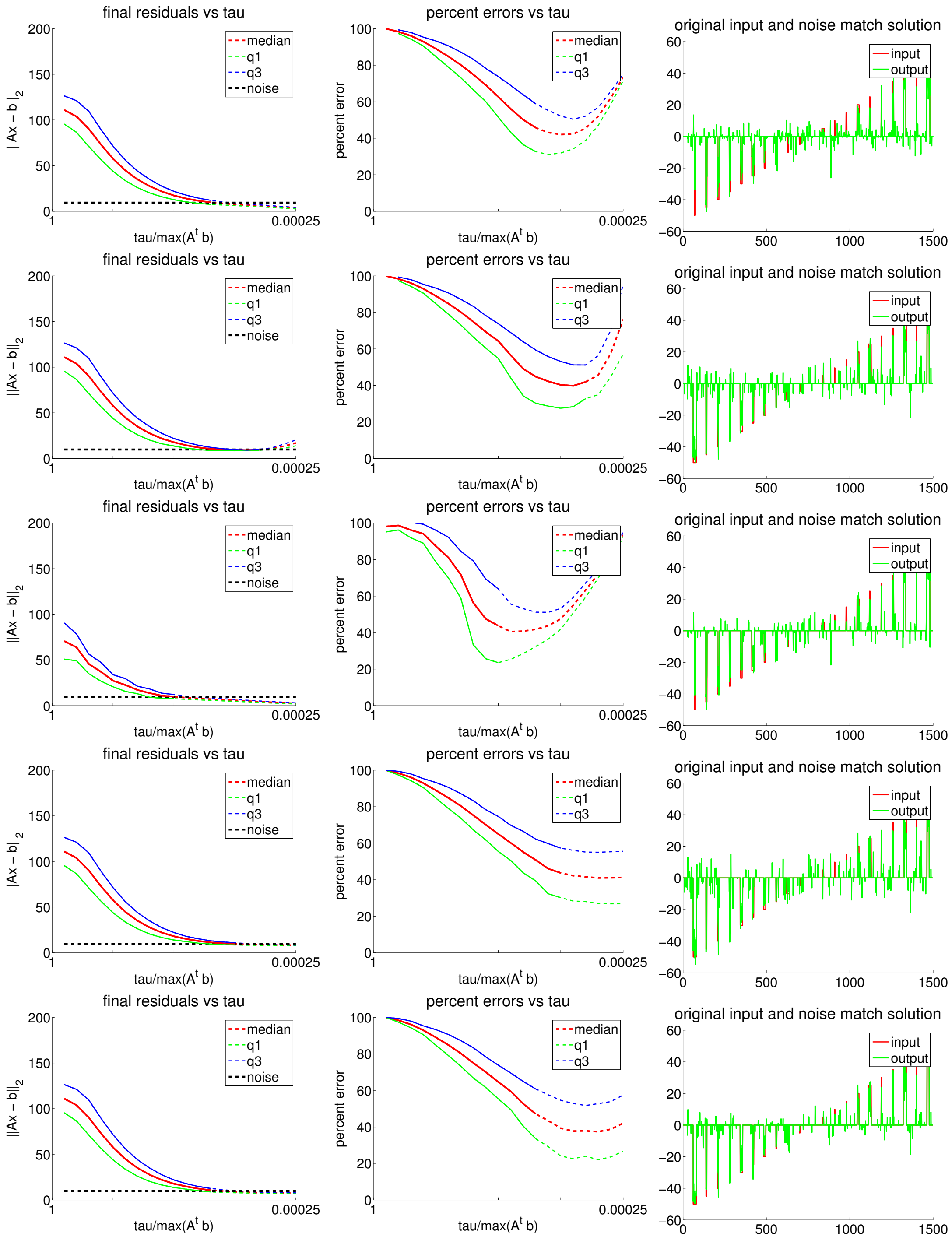


Figure 5.5: Ill conditioned staircase. FISTA, DALM, FIVTA, IRLS, IRLS SYS.

In the above plots we see that for the FIVTA scheme, the intersection of the residuals with the noise curve occurs at a higher τ compared to the other algorithms; otherwise the numerical performance of the five methods is quite similar. Next, we make a comment on the number of nonzeros produced in the solution of the different algorithms.

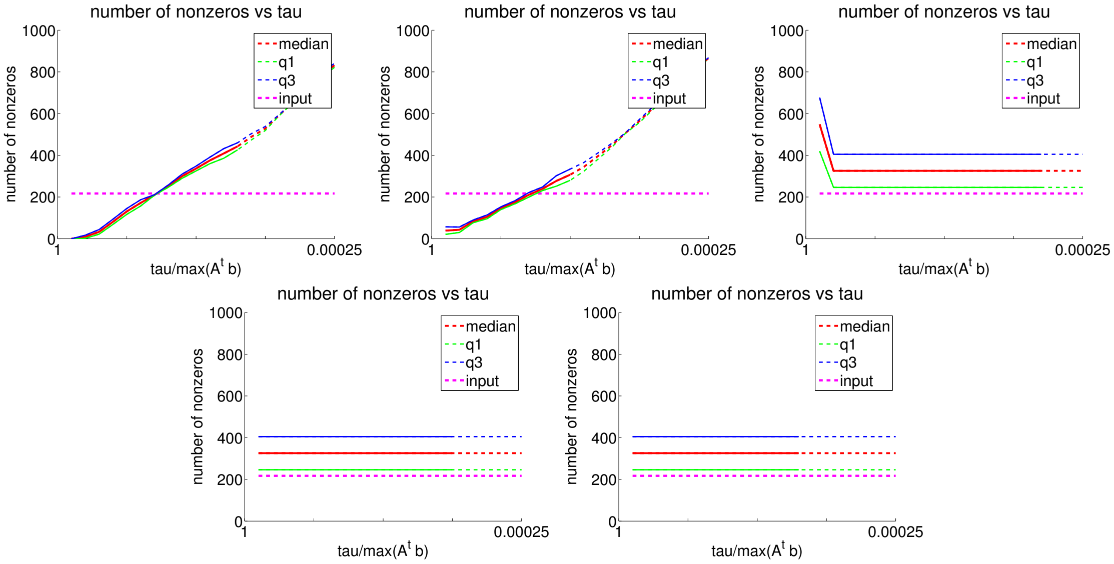


Figure 5.6: Number of nonzeros in solution versus parameter τ for FISTA, FIVTA, DALM, IRLS, and IRLS SYS

From the above figure we see that the number of nonzeros in the solution differs for the thresholding based schemes from the other algorithms. Only in the the two thresholding schemes, FISTA and FIVTA, do we explicitly set coefficients to zero at each iteration. For the other algorithms, this behavior is not present by default. We may however, explicitly control the number of nonzeros at each iteration, by keeping only a portion of the largest nonzero elements. This is done for the above tests so the the nonzero curves for DALM, IRLS, and IRLS SYS above remain constant. Note that this is simply a choice we made, we could have not zeroed out any elements at the end of each iteration, or zeroed out those whose absolute magnitude is smaller than τ, but then we would have small noise like elements pop up in the solution. Another alternative is to apply a filter at the very end to remove these small components.

We now present the results of some other numerical experiments. First, we present some results of sparse image reconstruction algorithms. We take a sparse image as a vector x (by stacking up the matrix rows), multiply it against a sensing matrix A, add some noise, and then try to reconstruct x from A and $b = Ax +$ noise using several different algorithms. We are interested in the quality of the reconstruction as the number of rows of the sensing matrix is varied (as the number of rows, or measurements is increased, the quality of reconstruction improves). This corresponds in some sense to the number of measurements of the noisy image collected with the sensing matrix. The sensing matrix is designed such that the RIP conditions for it are satisfied so this becomes an application of compressive sensing. We use a random Gaussian sensing matrix.

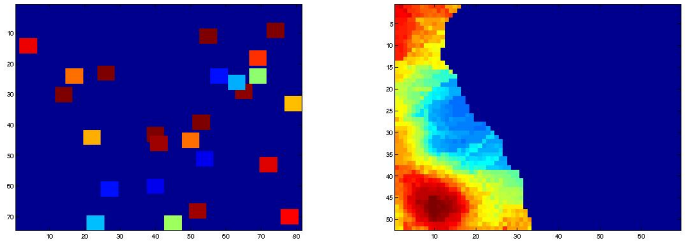


Figure 5.7: Two sparse images (A and B) used for reconstructions.

In the next page, we present the image reconstructions with the different algorithms. We again observe that for a fixed number of observations, the reconstructed images, as obtained with the different algorithms, look very much alike.

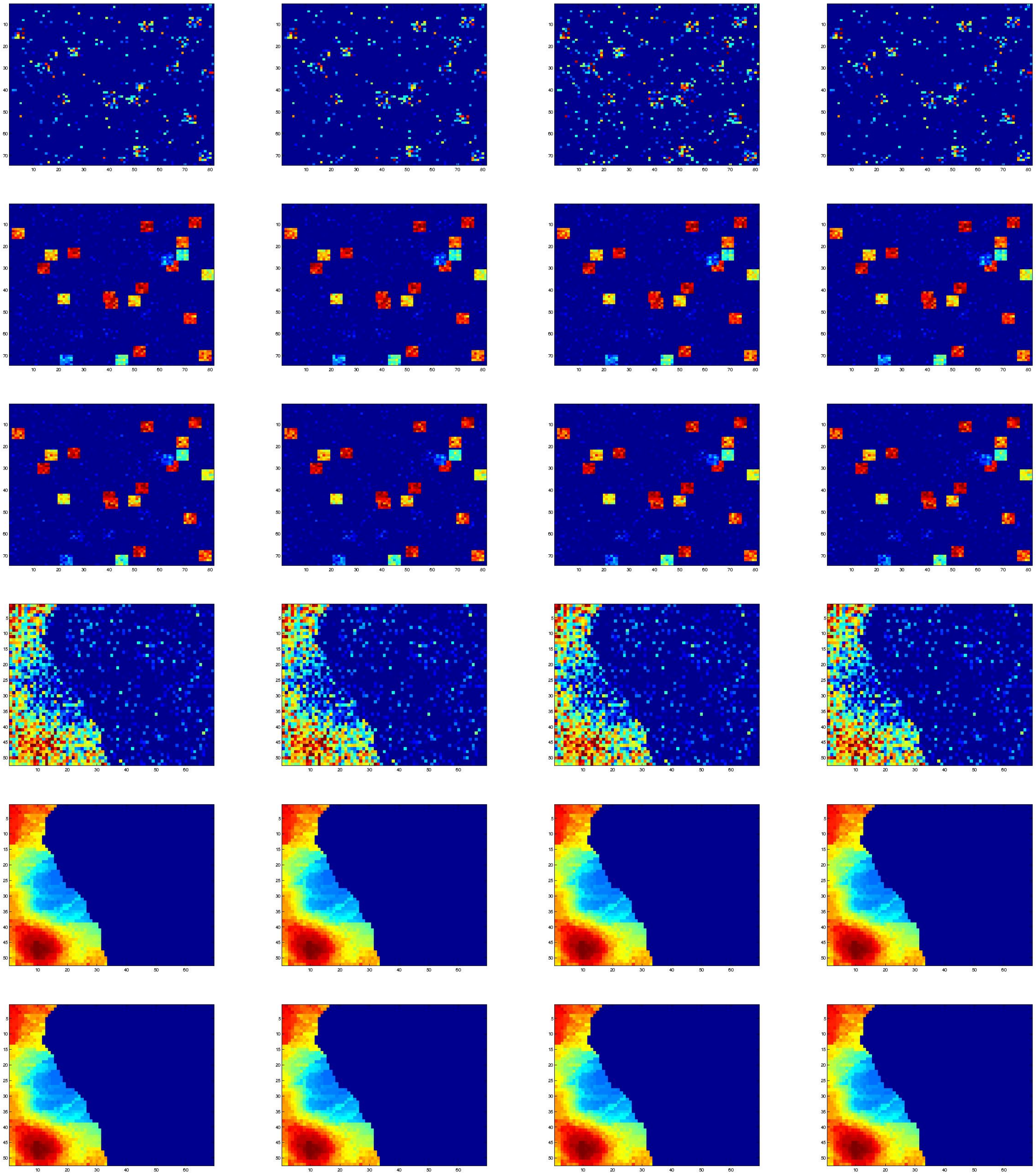


Figure 5.8: Reconstructions with FISTA (column 1), DALM (column 2), FIVTA (column 3), and IRLS (column 4) with sensing matrices with increasing numbers of nonzeros for two inputs (image A: rows 1-3; image B: rows 4-6).

Next, we run tests where we vary the noise level in the right hand side b and the matrix A and the number of nonzeros in the sparse input as we run the reconstructions with the different methods, to see how well the different algorithms are resistant to small perturbations in these quantities. We expect worser reconstructions as the errors or the number of nonzeros to reconstruct is increased. This is indeed what we observe but the performance is similar across different methods below:

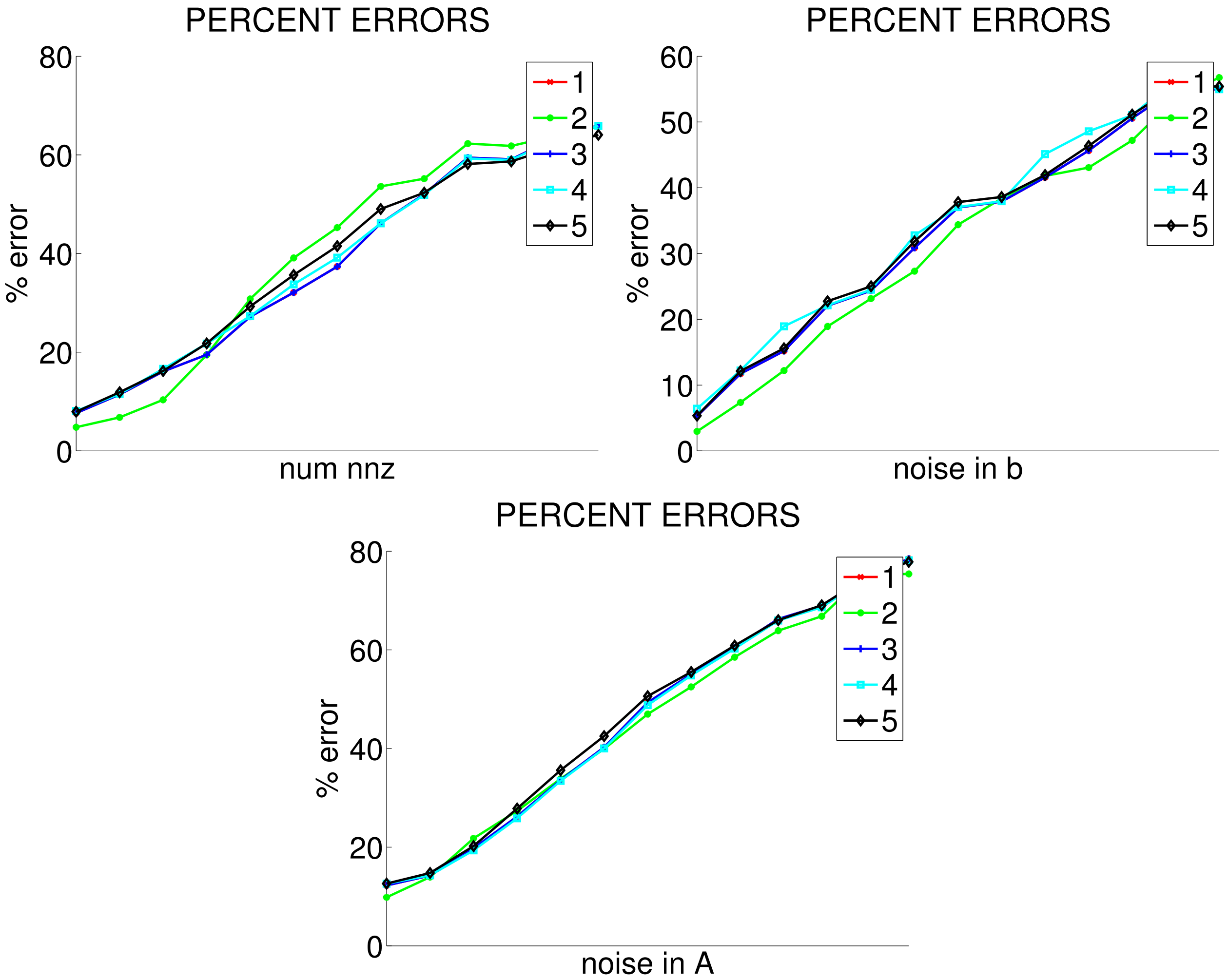


Figure 5.9: Percent errors in reconstruction versus increasing number of nonzeros in input, increasing noise in right hand side b and increasing noise in matrix A for different algorithms (FISTA, DALM, FIVTA, IRLS, and IRLS SYS); medians plotted over 10 runs.

We now observe that from all the examples given above, the algorithms seem to perform similarly. Indeed, since they optimize similar problems (the functions that DALM and FIVTA minimize are slightly different, but are still of a similar form), we do not expect the converged results to be very different. We now investigate the speed of convergence. First, we compare how FISTA, IRLS, and IRLS SYS decrease the value of the ℓ_1 functional $||Ax - b||_2^2 + 2\tau||x||_1$ for different systems. We see below that for the examples we consider, IRLS SYS decreases the functional more rapidly than the other algorithms.

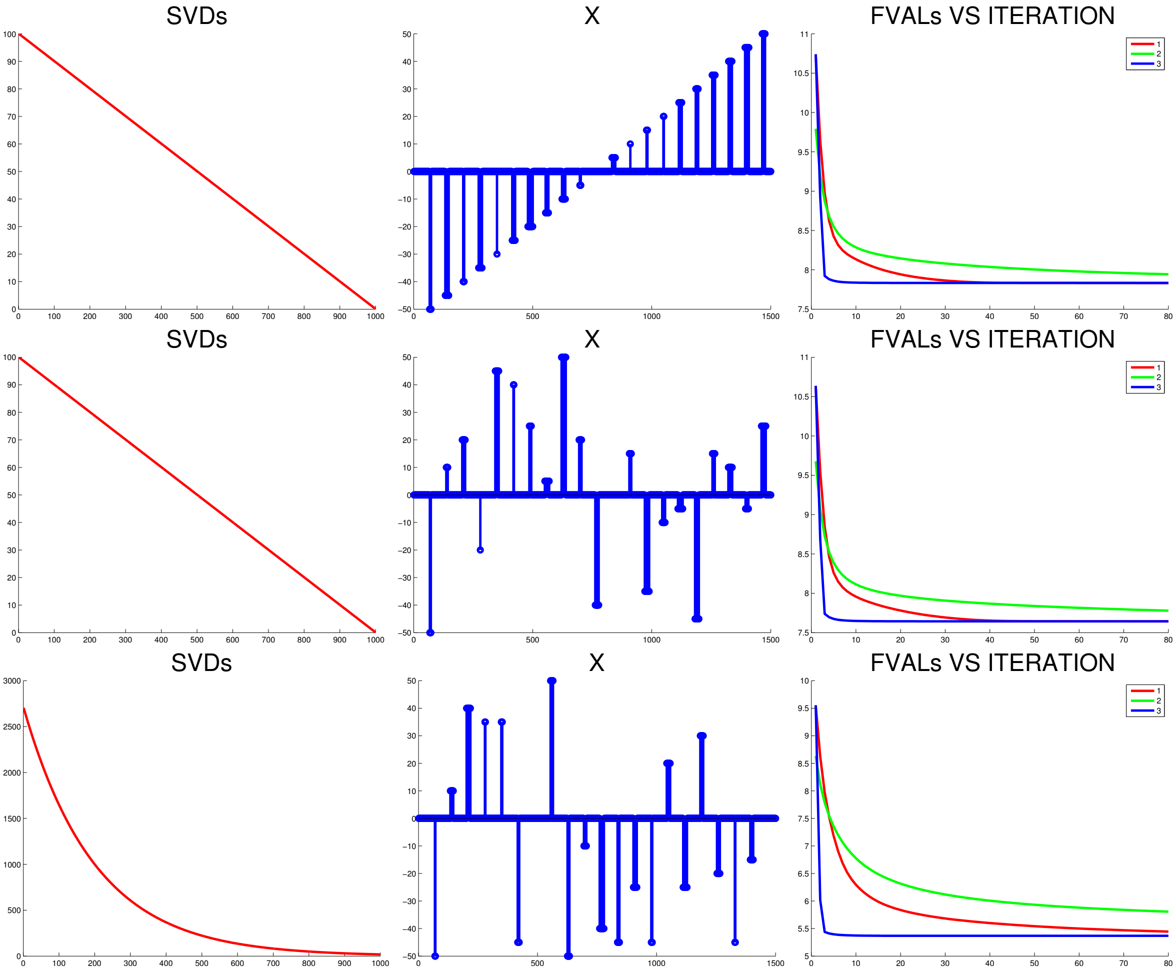


Figure 5.10: Decrease of ℓ_1 functional versus iteration at noise-matching τ versus iteration for FISTA (1-red), IRLS (2-green), and IRLS SYS (3-blue). Left column: singular value distribution, center: sparse input, right: functional values versus iteration.

5.8 Chapter Remarks and Conclusions

In this chapter we discussed a number of ideas and techniques useful in numerical implementation, especially in the case of large scale problems. We described an inverse replacement strategy for the second IRLS scheme. For the coordinate descent method, we presented a discussion on its implementation for larger problems: in particular we presented a useful way of obtaining estimated matrix column norms for a matrix and discussed its integration with an ISD scheme. We also commented on the application of the iteratively reweighted norms idea from Chapter 4 to other algorithms: we presented a reweighted norm approach for the dual space method from Chapter 2. Finally, we presented a fast version of a randomized SVD algorithm that can be used to obtain much smaller three matrix approximations to large ill-conditioned matrices. This is especially useful when the matrix we use consists of a transform (such as the product matrix AW^{-1}). Each application of such a matrix involves the application of the transform matrix W^{-1}. However, via the approximation $AW^{-1} \approx U\Sigma_k V^T$, the transform can be removed. We concluded the chapter by presenting the results of some simple numerical tests which show similar final reconstructions amongst the different algorithms and similar stability to noise in A and b. In terms of runtime, we observed that the second IRLS scheme is able to decrease the functional faster than the other methods.

Chapter 6

APPLICATION FROM GEOTOMOGRAPHY

6.1 Overview

The past few chapters introduced multiple algorithms for the regularization of solutions to large linear systems using sparsity constraints (although with the IRLS methods developed in Chapter 4, the desired degree of sparsity can be easily varied). We mentioned in passing that the motivation for these methods was an application in Geotomography and that the benefit of the simple (in terms of operations) algorithms was to allow them to be applied to very large problems. In this chapter, we present the application in more detail: we describe the mathematical formulation of the forward and inverse problems and show how we can apply the techniques that we have developed in the previous chapters. We also discuss the advantage of the mixed norm minimization accomplished with the IRLS schemes, by using a dictionary of different bases to represent the solution. Additional details regarding the application may be found in our group paper [41].

6.2 Travel-Time Tomography and Model Parameters

The problem we consider comes from global seismic tomography and we are interested in determining the three-dimensional elastic wave speed structure of the Earth, usually in the form of deviations from a spherically symmetric reference model (which is "radial" in the sense that it only varies with depth in the Earth.) To first order, the Earth's structure is that of a layered set of shells, dependent only on the radial distance downward from the surface. What concerns us in this application are the three-dimensional perturbations to this "background" or "reference" state. We now present a brief explanation of the above. A seismic event, such as an earthquake, occurring at some three-dimensional location within the Earth volume, sends out seismic waves which travel outward to the surface where they can be measured by recorders stationed at various parts of the world. Earthquakes only occur within a certain depth interval (typically confined to the first 700 km from the surface, whereas the Earth's radius is about 6731 km) and in narrow zones (primarily on plate boundaries), but the seismic waves they generate directly sample most of the Earth's volume [33]. Different regions of the Earth's mantle (the solid volume, below several tens of kilometers of crust and down to its liquid core, about halfway down the radius) are distinct compositionally and also thermally. The seismic wave velocities vary accordingly, and thus, by mapping seismic wave speeds, geophysicists attempt to construct the detailed structure of the Earth's interior temperature and composition.

There are two types of seismic waves: P (primary) and S (secondary) waves. Primary waves travel faster through the Earth and are the first waves to be picked up by seismometers when an earthquake event is measured. They are compressional waves that are longitudinal in nature. They move particles (such as rock) in the travel direction of the wave. On the other hand, the slower S-waves move the rock perpendicularly to the wave travel direction. As the waves move through boundaries between materials of different wave speeds, they experience refractions. In the simplest sense, we can make use of relations from geometric optics such as Snell's law to determine the rough paths of these waves, if we know the velocities of the materials through which they travel. Thus, given a first model of wave velocities, we can determine (at least approximately) how the waves move and then consider perturbations to the velocity model to image more detailed three-dimensional structure. As such, we can infer the nature of the Earth's interior from measurements on the Earth's surface by receivers (seismometers) of the waves' arrival times.[1]

The knowledge of seismic wave velocities at different locations inside the Earth can thus be used to deduce its interior structure. This is the reason why geophysicists are interested in accurate models of seismic wave velocities. In first approximation, the Earth is spherically symmetric, and there is a small handful of radial reference models with respect to which all current three-dimensional models are being formulated. In the reference models, the velocity changes only across depths, but not across latitudes and longitudes at a fixed depth. The Earth is, however, not truly spherically symmetric; for instance, it is known that there are hot plumes at various locations, which affect the wave velocities at these points. The goal of the application mentioned here (the solution to the inverse problem) consists of finding deviations to the spherically symmetric model of wave velocities at different latitudes and longitudes. These deviations are computed from a big under-determined linear system, obtained by a first order approximation of the difference between the computed and observed travel times, where the computed travel times come from the spherically symmetric model. The matrix encodes the wave propagation for various earthquake-receiver

[1] For example, S-waves are unable to travel through liquid. Thus they are unable to travel through the Earth's outer core. Hence, from some seismic events at some locations we do not observe any S-waves at all and this allows us to conclude that some portion of the Earth (the outer core) is indeed liquid, while the inner core is solid.

pairs: seismic waves from earthquakes are picked up at different receivers throughout the world. However, with the current data set [39], we are mostly constrained to receivers located in the western part of the USA. This means that the resolving power of the matrix is also limited to this location, and we are not able to solve for deviations from spherical symmetry on a global scale. However, we formulate the mathematical set up to be able to deal with global data and models, which will become available to us in the future.

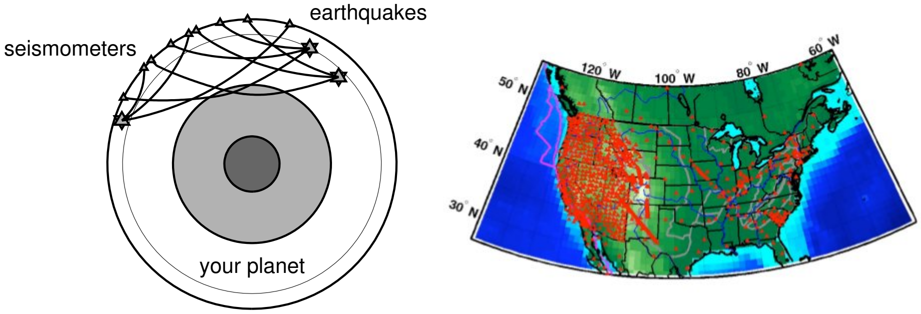


Figure 6.1: Ray paths of the compressional (P) waves emanating from various earthquakes in the *ak135* reference model. Figure courtesy of Frederik J. Simons. Distribution of Seismic Stations corresponding to our data set. Figure courtesy of Guust Nolet.

We now describe the forward and inverse problems from the application that we have described.

Forward Problem: The forward problem consists of predicting the arrival times of compressional (P-) waves at some location on the Earth's surface, given a velocity field $v(\mathbf{r}) = v(r, \theta, \phi)$, where $r = ||\mathbf{r}||_2$ is the radius (or measure of depth) and θ and ϕ the colatitude and longitude, respectively. For spherically symmetric Earth models, the reference velocity $v_0(\mathbf{r})$ varies with radius only, and to first order, $v(\mathbf{r}) = v_0(\mathbf{r}) + \delta v(\mathbf{r})$. The arrival time of the wave, as predicted with this velocity model $v(\mathbf{r})$ is given by:

$$t(\Delta) = \int_{s(0)}^{s(\Delta)} v^{-1}\left(\mathbf{r}(s)\right)\, ds,$$

where Δ is the epicentral distance, the arc length between the earthquake epicenter and the station location. The integration above, is along the ray path from epicenter to the station receiver. That is, given an assumed velocity field (an existing model of wave velocities inside the Earth, say a spherically symmetric one) and the location and time of the event (an earthquake), the forward problem consists of predicting the arrival times of the compressional waves at a given location on the Earth's surface.

Inverse Problem: The inverse problem consists of finding the deviations $\delta v(\mathbf{r})$ to a model velocity field $v_0(\mathbf{r})$ from many different observations recorded on seismograms from different earthquakes. The observations are the travel-time anomalies, $\delta t(\Delta)$, i.e. the difference between the arrival times of the seismic waves in the actual Earth, and those predicted via the forward model in the reference Earth, at different epicentral distances Δ. Sticking with the geometric optics approximation (aka Ray theory, at infinite frequency), and to first order, the travel-time anomaly is given by an integral over the ray in the background/reference model $v_0(\mathbf{r})$:

$$\delta t = \int_{s(0)}^{s(\Delta)} \delta v^{-1}\left(\mathbf{r}(s)\right)\, ds = -\int_{s(0)}^{s(\Delta)} \frac{\delta v(\mathbf{r})}{v_0^2(\mathbf{r})} ds.$$

This equation can be discretized to yield a linear inverse problem, in which the objective is to determine the distribution $\delta v^{-1}\left(\mathbf{r}\right)$ from the observations δt. Discretizing in voxels gives the problem in the familiar linear system form $Ax = b$, where b is the vector of delay times δt and x is the vector of the unknown slowness perturbations $\delta v^{-1}(\mathbf{r})$ and A is the sensitivity matrix that is filled with the path lengths of the rays in the discretized voxels. That is, the rows of the matrix correspond to different earthquake-receiver pairs and the columns to the discrete voxel elements j (a location inside the Earth). For each earthquake-receiver pair, the seismic waves trace out a ray that travels from the epicenter of the earthquake through different voxels to the location of the receiver. In this setup, $A_{i,j}$ is the length of the segment ds_j, the portion of the ray inside discrete element j for ray i:

$$[\delta t_i] = [- \quad A_{i,j} \quad -][\delta^{-1} v],$$

where $A_{i,j} = [ds(j)]_i$ and $\delta t_i = \int_{\text{ray}_i} \delta^{-1} v\left(\mathbf{r}(s_i)\right) ds_i$ so the system of equations reads as:

$$\delta t_i = \sum_j [ds(j)]_i \delta^{-1} v(j) \implies b_i = \sum_j A_{i,j} x_j.$$

Taking into account the finite-frequency effects of the wave field (as opposed to ray theory) leads to more complex relations between the observations δt and the model. Instead of

$$\delta t = \int_{\text{ray}} [-v_0^{-1}(\mathbf{r})] \left[\frac{\delta v(\mathbf{r})}{v_0(\mathbf{r})}\right] ds$$

when more physics of the wave problem is taken into account, the relation becomes:

$$\delta t = \int\int\int_{\text{earth}} K(\mathbf{r}) \left[\frac{\delta v(\mathbf{r})}{v_0(\mathbf{r})}\right] d\mathbf{r} \qquad (6.2.1)$$

Here, the $K(\mathbf{r})$ is a sensitivity kernel that, depending on the approximations used [33], describes more or less completely the interactions between scattered portions of the wavefield in a medium that is only slightly perturbed away from the reference, as before. Various ways exist to compute the kernels $K(\mathbf{r})$ for the various wave types considered. One of the most widely known travel-time kernels is of the "banana-donut" type [10], named after its characteristic topology. We recognize here that no matter how the observations δt to the Earth model $\frac{\delta v(\mathbf{r})}{v_0(\mathbf{r})}$ are modeled, both types lead to linear inverse problems after discretization. Letting the unknowns be $m(\mathbf{r}) = \frac{\delta v(\mathbf{r})}{v_0(\mathbf{r})}$, we can write the discrete form of the above as:

$$(\delta t)_i = \sum_{j=1}^{N} m_j \int [K(r)]_{i,j} dr \implies b_i = \sum_{j=1}^{N} K_{i,j} m_j \implies b = Ax \qquad (6.2.2)$$

where $K_{i,j} = \int [K(r)]_{i,j} dr$ the integration of kernel i (corresponding to the i-th earthquake-receiver pair) over grid segment j. The linear system $Ax = b$ has $A_{i,j} = K_{i,j}$, $x = [m_j]$ and $b = [(\delta t)_i]$.

Rather then solving the system in its original form, we assume that the unknowns $m(\mathbf{r}) = \frac{\delta v(\mathbf{r})}{v_0(\mathbf{r})}$ are sparse in the wavelet domain. That is, with a suitable set of wavelet functions $W_j(\mathbf{r})$, they may be expanded as:

$$m(\mathbf{r}) = \sum_{j=0}^{J} w_j W_j(\mathbf{r})$$

with few $w_j \neq 0$. Plugging into (6.2.1), this becomes:

$$\delta t = \sum_{j=0}^{J} w_j \int\int\int_{\text{earth}} K(\mathbf{r}) W_j(\mathbf{r}) d\mathbf{r} \qquad (6.2.3)$$

Discretizing, we again come up with a linear system:

$$(\delta t)_i = \sum_{j=0}^{N} w_j \int [K(r)]_{i,j} W_j(r) dr = \sum_{j=0}^{N} w_j \bar{K}_{i,j}$$

where the kernels are now integrated with the wavelet functions: $\bar{K}_{i,j} = \int [K(r)]_{i,j} W_j(r) dr$. The above then becomes a linear system with the matrix composed of elements $\bar{K}_{i,j}$. The disadvantage of this formulation is that because the matrix elements are the integrals of the kernels with the wavelet functions, for any new set of wavelet functions (W_j), we must rebuild the matrix. Thus, we instead go back to (6.2.2) and use instead of the vector $x = (m_j)$ its wavelet representation $x = W^{-1} w$:

$$b_i = \sum_{j=1}^{N} K_{i,j} \left(\sum_{k=1}^{N} W^{-1}_{j,k} w_j \right) = \sum_{j=1}^{N} \sum_{k=1}^{N} K_{i,j} W^{-1}_{j,k} w_j$$

where W^{-1} represents the inverse transform matrix. Now due to the presence of noise in the b vector and the assumption of sparsity on the coefficients w_j we can formulate (in the simplest case) the following minimization problem:

$$\min_{w_j} ||b_i - \sum_{j=1}^{N} \sum_{k=1}^{N} K_{i,j} W^{-1}_{j,k} w_j||_2^2 + 2\tau \sum_{k=1}^{N} |w_j| \implies \min_{w} ||AW^{-1}w - b||_2^2 + 2\tau ||w||_1$$

with the matrix A being composed of elements $K_{i,j}$.

The data set we use consists of the delay times in [40], which have a distributed set of earthquakes on a global scale but fix the station set in North America, as evident from Figure 6.2. In this data set, A is up to 500,000 by 3,000,000 and not well conditioned; b is noisy.

6.3 Cubed Sphere Grid and Wavelets

We now discuss what can be referred to as tools for the experiment: we discuss the coordinate grid, wavelets, and the characteristics of the matrix. We first comment on the coordinate system that we use. The choice is somewhat influenced by the use of the wavelet transform. The coordinate system must cover various points inside the Earth, over which our sensitivity kernels are defined. While it is easy to adopt a spherical coordinate system, this demands some special treatment, especially at the poles where there are singularities. Spherical wavelets are also more challenging to construct and program than those on a flat two dimensional grid. In particular the singularities call for special numerical measures. For this reason, we use a so called cubed-sphere grid [37], a projection of the sphere onto a cube. More explicitly stated, we project the surface content of a sphere of a certain radius onto the six faces of a cube. There are a number of different depth layers (37 in our system) and each depth layer corresponds to a sphere of that radius and its surface contents are mapped onto the six chunks (faces) of the corresponding cube. Thus, a column vector expressed in the cubed sphere coordinate system is inherently 4 dimensional: there is a depth layer coordinate $(1 - 37)$, the chunk number $(1 - 6)$ and the x and y indices within the chunk, each of which go from $1 - 128$. Thus, the total number of grid points (and variables) in our system is $37 \times 6 \times 128 \times 128$ which is over three million. The cubed sphere projection for a sample model is illustrated below:

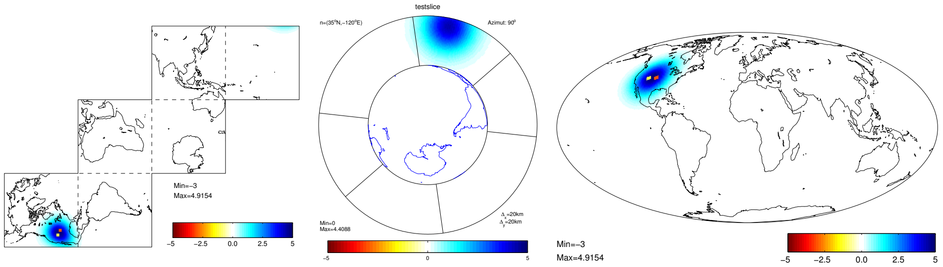


Figure 6.2: Model on the cube, depth slice, projection onto spherical surface

The advantage of this formulation is that the wavelet transform, when applied to a vector that is encoded in this coordinate system, can be done largely by applying the transform over the individual chunks, each of which has an underlying 2D coordinate system. At the beginning and end of the routine, extra work must be done to handle chunk boundaries, but the main portion of the work is easily parallelizable as it involves independent operations over the six chunks.

There is a number of different wavelet transforms we can use; we will see shortly that it is not necessary to make a precise choice a priori; the algorithms we have developed (in particular, the IRLS schemes), can easily be used for many different transforms. The transforms vary mostly with respect to smoothness of the underlying wavelet basis. For example, the classical Haar transform has sharp drop-offs and can represent sharper cut-offs and features, while a smoother transform like CDF4-2 is better at representing smoother features. Each transform consists of scaling and wavelet functions; the number of wavelet functions depends on the transform. Consider a single level of the CDF4-2 transform and the corresponding wavelet and scaling functions below:

Figure 6.3: Scaling and Wavelet Functions of a CDF4-2 transform

Previously we discussed minimization problems such as:

$$||Ax - b||_2^2 + 2\tau||x||_1$$

Below we will refer to vectors x and w. In this notation, the vector x will be in the voxel domain (corresponding to the cubed sphere grid we introduced above). The vector w is the representation of x in the wavelet domain. They are related through the following relations involving the forward transform W and the inverse transform W^{-1}:

$$w = Wx \quad \text{and} \quad x = W^{-1}w$$

When we look for a sparse solution in the wavelet domain, we mean that we expect the vector w to be sparse; the vector x may in fact be not very sparse. However, for the solutions we obtain we would still like for $||Ax - b||_2$ to be small. Instead of the above ℓ_1 functional we therefore instead use the functional:

$$||Ax - b||_2^2 + 2\tau||w||_1 = ||AW^{-1}w - b||_2^2 + 2\tau||w||_1$$

so that the sparsity promoting penalty is on w not on x. With this formulation, all the previous algorithms can now be used with the matrix AW^{-1} instead of A. This is possible to do even if AW^{-1} is not available explicitly, since this matrix and its transpose only need to be applied to vectors. Notice that: $(AW^{-1})^T = W^{-1,T}A^T$, implying that the so-called "inverse transpose transform" is required for computation. This transform can be obtained by applying the forward transform with the wavelet filters used from the inverse transform.

A vector in the wavelet domain consists of scaling coefficients and wavelet coefficients of different levels (depending on the number of levels of the applied transform). For a transform with P levels:

$$w = (s, w_1, \ldots, w_P)$$

The different portions of the wavelet vector w can reconstruct different parts of the original model. That is, given $x = W^{-1}w$, we claim that $W^{-1}(s, 0, \ldots, 0)$, $W^{-1}(0, w_1, \ldots, 0)$, $W^{-1}(0, 0, w_2, \ldots, 0)$, and so on, all reconstruct x within some tolerance. Of course, the number of coefficients at each scale is different, so this means some coefficients are more crucial to the reconstruction than others. In Figure 6.3 we illustrate this using an example model decomposed with two levels of wavelets (using the CDF4-2 transform): From Figure 6.3 we see that the scaling coefficients

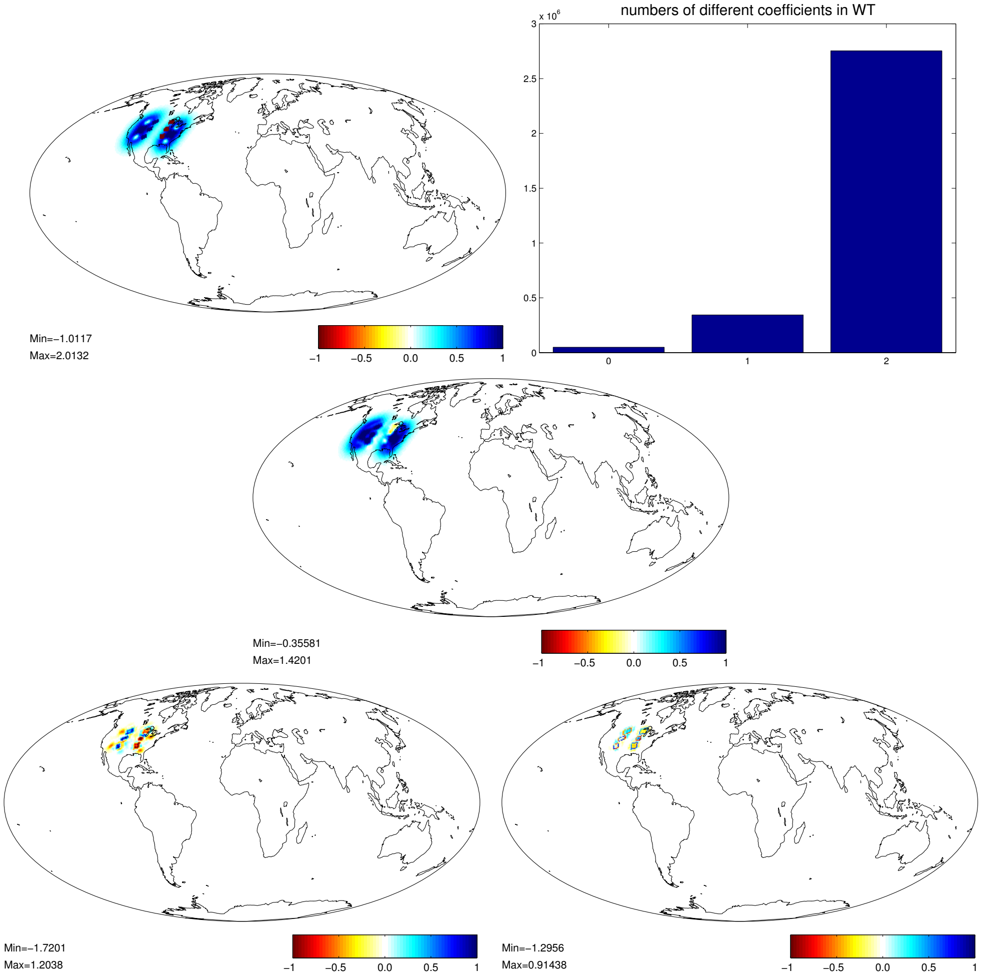


Figure 6.4: Top row: original model, numbers of different coefficients: scaling and two wavelet scales. Middle row: reconstruction using only the scaling coefficients: $W^{-1}(s, 0, 0)$. Bottom row: reconstructions using only the fine and coarse wavelet coefficients: $W^{-1}(0, w_1, 0)$, and $W^{-1}(0, 0, w_2)$.

almost entirely represent the Gaussian ball, and the wavelet coefficients are needed to represent the “holes” and the small cubes in the plots. Still the biggest amount of information is contained in the small number of scaling coefficients. It is thus clear from this example that in certain situations different coefficients demand different treatment. For instance, if we expect to recover a model of this type it may make sense to impose an ℓ_2-penalty on the scaling coefficients so that none of them are forced to zero, while on the wavelet coefficients we can safely impose an ℓ_1-penalty since only few of the large number of wavelet coefficients are nonzero, meaning that we are interested in a sparse solution. The IRLS schemes, which are designed to minimize the general functional:

$$||AW^{-1}w - b||_2^2 + 2\sum_{k=1}^{N} \lambda_k |w_k|^{q_k} \quad \text{for} \quad 1 \leq q_k < 2$$

can be used to do this. Suppose that w corresponds to wavelet coefficients. For the above examples we have $P+1$ index sets corresponding to scaling coefficients and the P wavelet scales. For the k in the scaling wavelet scale we can set q_k close to two and perhaps a lower λ_k. For the other k we can set $q_k = 1$ and perhaps a slightly higher λ_k.

6.4 Matrix Properties and Sample Reconstructions

We now discuss some properties of our matrix and the types of reconstructions that we can expect to obtain from it. This is a prelude to the next section, where we discuss how we carry out the actual inversions with real data. As we mentioned before, the matrix consists of data from the integration of sensitivity kernels at different points inside the Earth (on a cubed sphere grid). The number of variables in our models and hence the number of columns of our matrix is this same number: $37 \times 6 \times 128 \times 128 = 3,637,248$. The rows correspond to the available source-receiver pair data contained in the right hand side b. In the most complete data set we have about half a million source-receiver pairs, hence this many rows. The first thing to note about the matrix is that because of the locations of the receiving stations (most of which are located over the Western USA as shown above) the reconstructions we obtain are mostly limited to this location. In Figure 6.4 we show a plot of the column sums of the matrix:

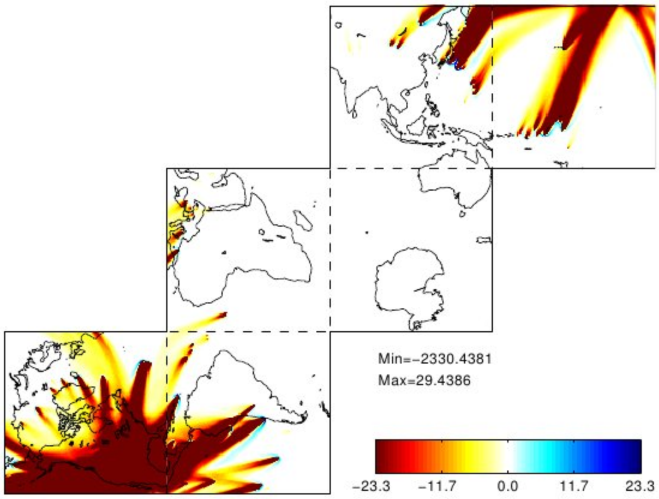


Figure 6.5: Matrix Column Sums.

Since, we cannot recover much from empty data, we do not expect to be able to recover much beyond the colored regions above. However, as we will see below, we can still recover reasonable models over the USA with this data set. The region over the USA is also somewhat limited, as we show in the reconstruction of the synthetic models below. Before showing the output we discuss here what we mean by a reconstruction. We generate a model x and use our matrix to obtain Ax. We then add some Gaussian random noise (5 percent) to the right hand side to obtain $b = Ax + \text{noise}$. We then reconstruct x by running some algorithm (FISTA, IRLS, DALM, etc) at a τ such that $||Ax - b|| \approx ||\text{noise}||$. This τ is chosen by hand, estimated using a few different short runs. We now show two reconstructions of a USA model obtained by minimizing:

$$||AW^{-1}w - b||_2^2 + 2\tau||w||_1$$

for two different transform W, a Haar and a CDF transform. We show these reconstructions below in Figure 6.4.

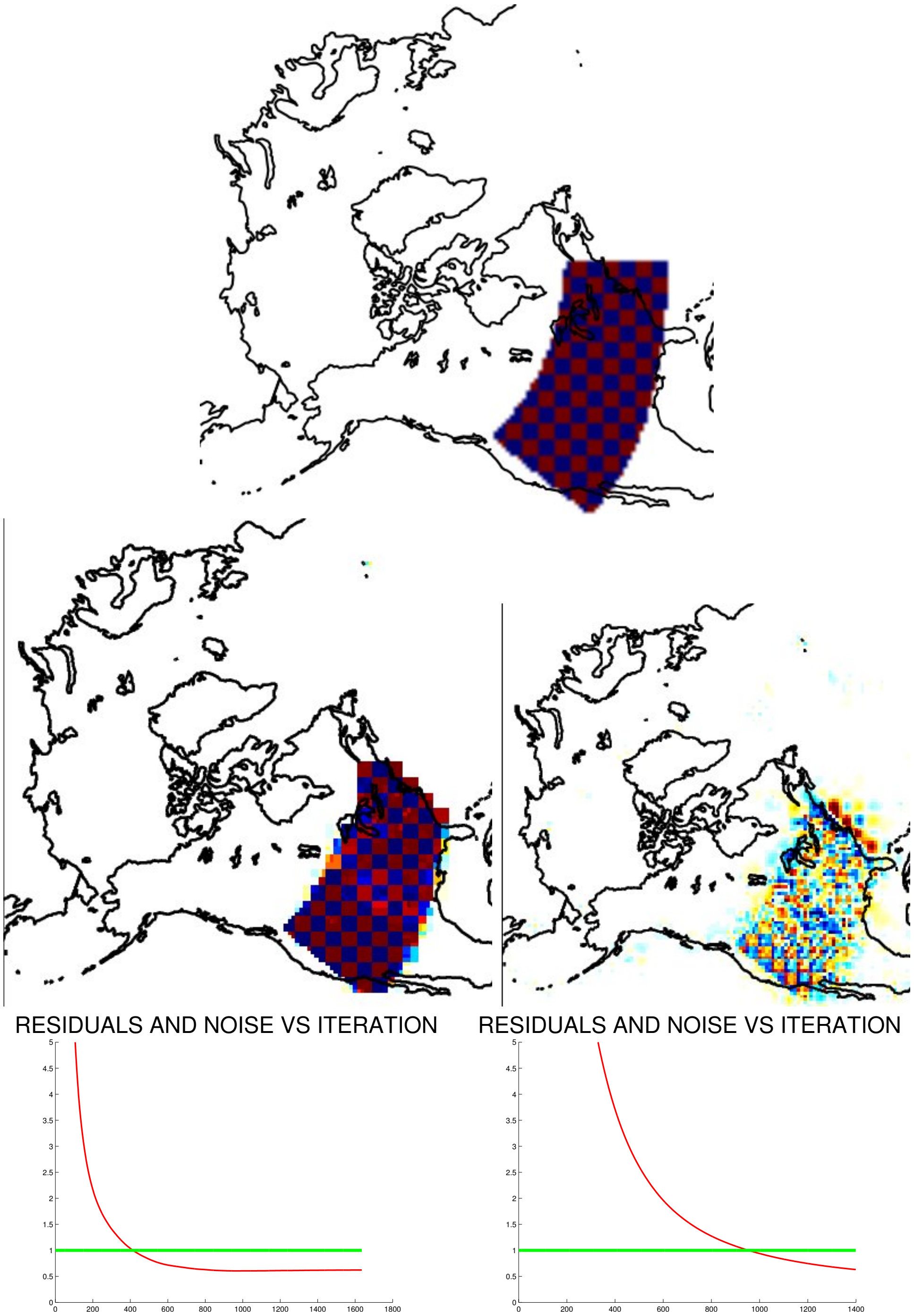


Figure 6.6: USA checkerboard and reconstructions with FISTA using the Haar and CDF Transforms followed by plots of the residuals versus noise.

Figure 6.4 illustrates in which regions we can expect to obtain reasonable reconstructions from our inverse problem. In particular, we do not expect to resolve as much in the mid USA region, but we do obtain sensible reconstructions in the Western and

Eastern portions, with the Western USA being the area of best reconstructions. This is not suprising, given the plot of the receiver location stations we saw earlier.

In addition, we see that the output is quite different for the models. In the case illustrated in Figure 6.4, we would almost certainly prefer the Haar reconstruction because of the sharp drop-offs in the input. In many other cases, however, the Haar reconstruction is substantially worse than that with CDF wavelets.

In general, we would like to let the algorithm pick from different bases automatically. This can be done by minimizing:

$$||A\left(W_1^{-1}w_1 + \cdots + W_B^{-1}w_B\right) - b||_2^2 + 2\tau_1||w_1||_1 + \cdots + 2\tau_B||w_B||_1$$

for B different bases. For one of the bases we can even choose the identity (meaning the voxel basis) since certain features may be best represented without wavelets. In the reconstructions below (Figure 6.4), we use Haar, CDF4-2, and the Identity basis. For Haar and CDF4-2 we use 2 levels of the wavelet transform (out of the 4 that is possible with our routines). Of course, this is just a choice we make for simplicity and one we find that works reasonably well; it would be even better to use more different basis: Haar with 1 to 4 levels, CDF4-2 with 1 to 4 levels and so on. For simplicity we pick one universal τ and use that for all the different bases.

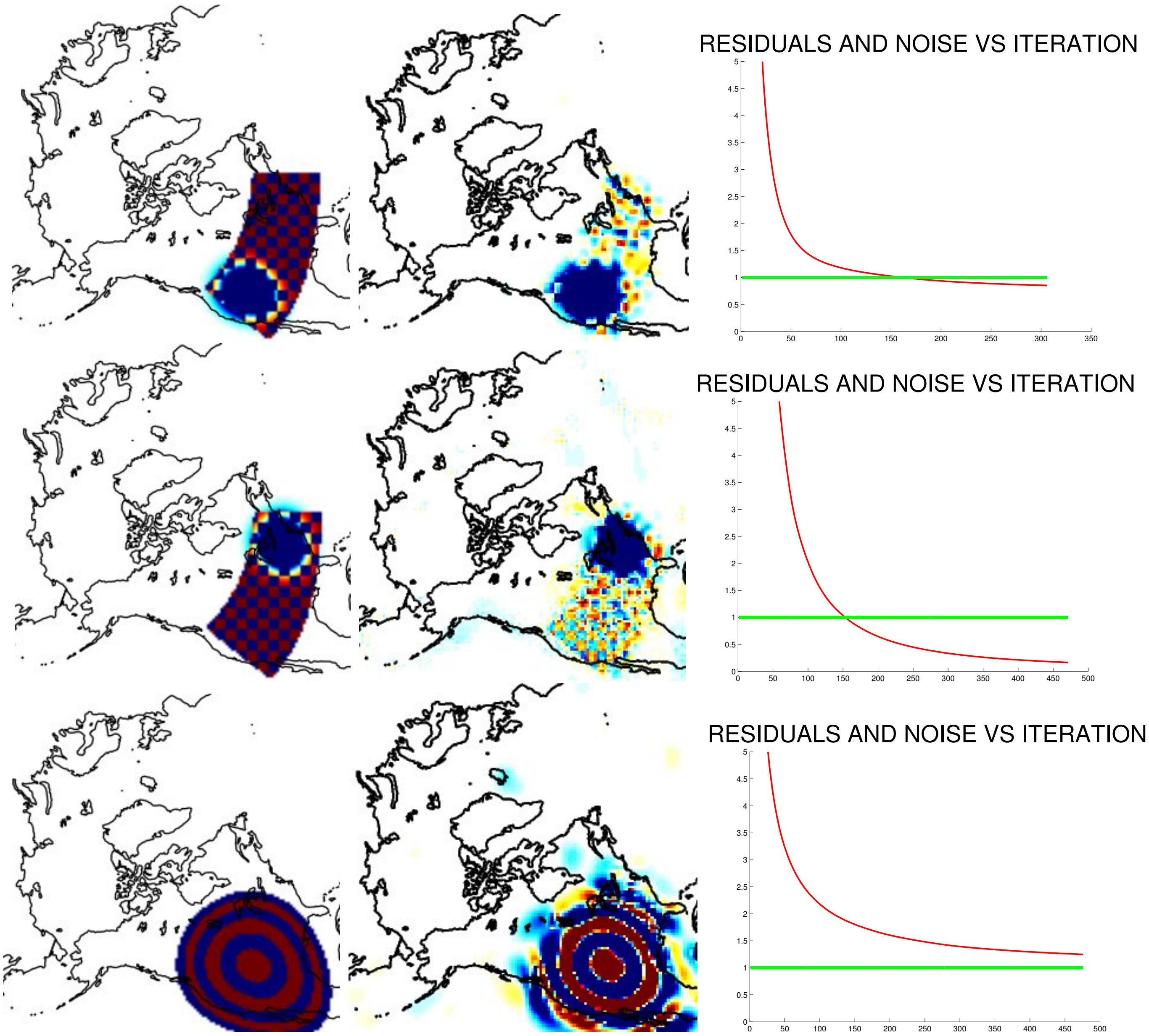


Figure 6.7: Three input models and their reconstructions (at a fixed depth) with the FISTA scheme with a 3 wavelet combination basis at 400 iterations. Left: input model, center: reconstruction, right: plot of residual norm $(||Ax - b||_2^2)$ and noise vector norm versus iteration. The second model is not computed at the right residual level but the reconstruction still exhibits good behavior.

.

To judge the quality of the reconstructed models, we also include below the plots of the histograms of the residuals, plotted back to back with a histogram of the normal distribution with the same mean and variance as the residuals as well as the QQ plots, which compare the distribution of the residuals against the normal distribution.

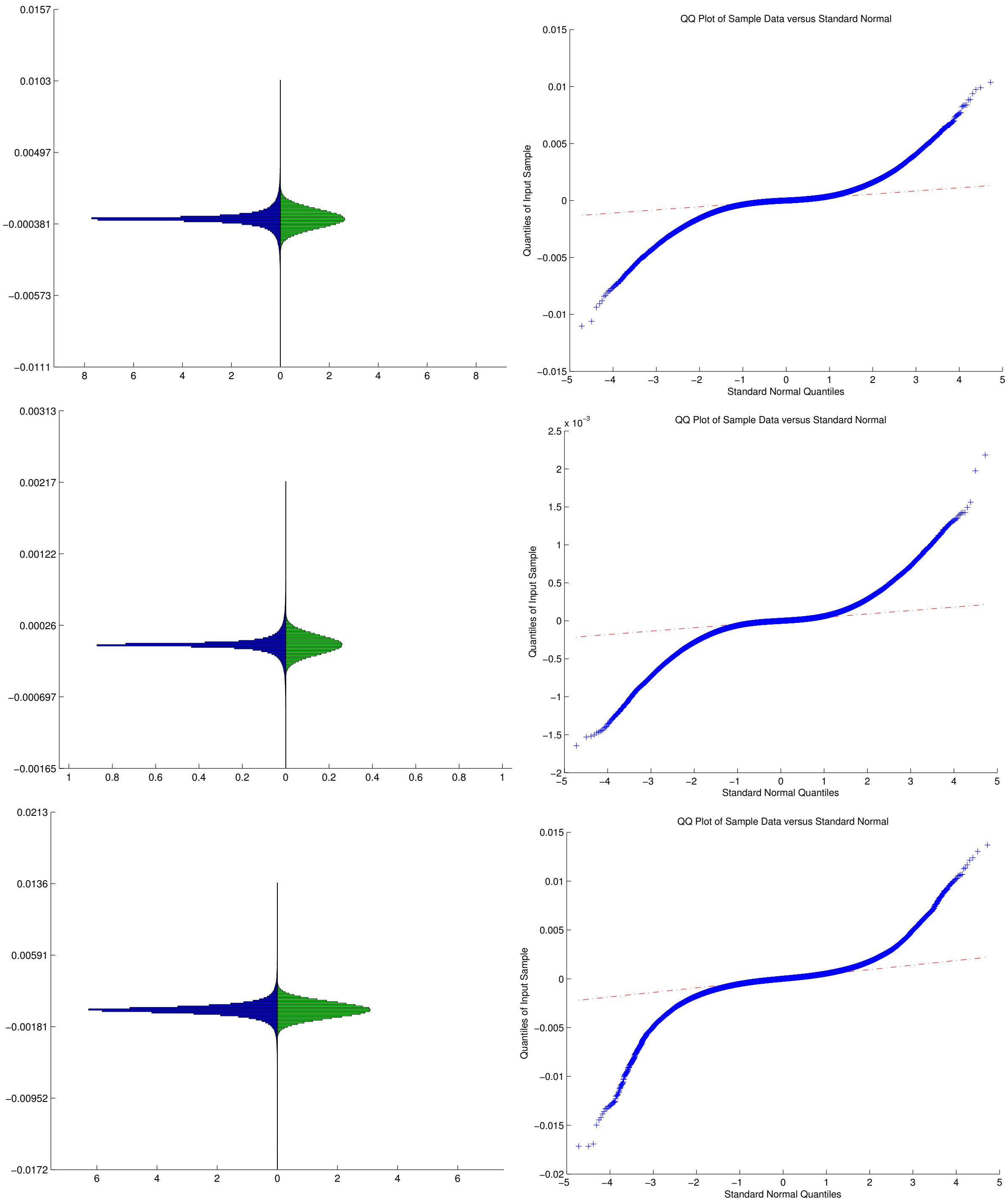


Figure 6.8: Histograms of resdiduals (left histogram) and of corresponding random normal distribution with same mean and variance (right histogram) and QQ plots which measure how close the residuals are to the normal distribution

We see from the above plots that the residuals are nicely distributed around zero with no long tails. From the QQ plots we see that they do not follow the normal distribution, but in fact they are even more centered around zero. In general, the reconstructions we obtain are close to the original models, which means that we should be able to obtain suitable reconstructions with real data.

6.5 Inversion

In the previous section, we mentioned some properties of the matrix and presented some reconstructions of synthetically constructed models with different sets of features. Here we discuss the inversion of a data set [39] that is centered over the Western US, a region where we expect (based on previous discussion in this chapter) to obtain good reconstructions. We now discuss the general mathematical set up. In the inversion of the data set, more terms come into play. Apart from the matrix A, we also have correction and damping terms. The correction terms serve to correct certain data about station location information and earthquake source locations, since earthquake locations are only known approximately, which directly affects travel times. These data are then apended as columns to the end of the matrix. This process introduces more variables into the system. As a result, it is customary to damp these extra variables or impose an ℓ_2 penalty on them. Without additional regularization terms, we would like to solve the least squares problem:

$$\min_{w_1,\ldots,w_B,v} \left\| \begin{bmatrix} AW_1^{-1} & \cdots & AW_B^{-1} & C \\ 0 & \cdots & 0 & D \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_B \\ v \end{bmatrix} - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|_2^2$$

where $W_1, \dots, W_B$ are a collection of wavelet bases, C are the correction terms and $D = \epsilon I$ are damping terms for the corrections. Assuming we would like to impose an ℓ_1 penalty on all components, we obtain the following minimization problem:

$$\min_{w_1,\dots,w_N,v} \left\| \begin{bmatrix} AW_1^{-1}, \dots, AW_B^{-1} & C \\ 0 \dots 0 & D \end{bmatrix} \begin{bmatrix} w_1 \\ \dots \\ w_B \\ v \end{bmatrix} - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|_2^2 + \tau_1 ||w_1||_1 + \dots + \tau_B ||w_B||_1$$

Expanding this yields:

$$\min_{\bar{w},v} ||AW_1^{-1}w_1 + \dots + AW_B^{-1}w_B + Cv - b||_2^2 + ||Dv||_2^2 + \tau_1||w_1||_1 + \dots + \tau_B||w_B||_1 \quad (6.5.1)$$

where we have used $\bar{w} = (w_1, \dots, w_B)$. The regularization parameters $\tau_1, \dots, \tau_B$ can be chosen based on the weight we would like to assign to each basis. For example, if we would like to penalize the Haar transform more, we would make the τ corresponding to it higher. Additionally, we can utilize different penalties, not just the ℓ_1 norm. The different parts $w_1, \dots, w_B$ each correspond to a different basis. In particular, if each is a different wavelet basis we can choose to penalize the different coefficients of each basis differently (for example an ℓ_2 penalty on the scaling coefficients and an ℓ_1 penalty on the rest).

We now proceed to describe how we can carry out the minimization above. We describe two methods: one based on so called alternate minimization and one based on a splitting approach (the ADMM multipler method introduced in Chapter 2). For the first method, we alternate between minimizing $\bar{w} = (w_1, \dots, w_N)$ and v. First, holding v constant in (6.5.1), we have:

$$\min_{\bar{w}} \left[||AW_1^{-1}w_1 + \dots + AW_B^{-1}w_B - K||_2^2 + \tau_1||w_1||_1 + \dots + \tau_B||w_B||_1 \right]$$

with $K = b - Cv$. Let $M = \left(AW_1^{-1}, \dots, AW_B^{-1}\right)$. We thus have:

$$\min_{\bar{w}} \left[||M\bar{w} - K||_2^2 + \tau_1 ||w_1||_1 + \cdots + \tau_B ||w_B||_1 \right]$$

This sparse regularization can then be performed using a number of different methods introduced in the previous chapters (IRLS, FISTA, etc).

Then, holding $\bar{w}$ constant in (6.5.1) we have:

$$\min_{v} \left[||Cv - J||_2^2 + ||Dv||_2^2 \right]$$

where:

$$J = b - \left(AW_1^{-1}w_1 + \cdots + AW_B^{-1}w_B\right) = b - M\bar{w}$$

The solution to the quadratic minimization problem is given by:

$$C^T(Cv - J) + D^T Dv = 0 \implies (C^T C + D^T D)v = C^T J = C^T(b - M\bar{w})$$

We thus arrive at the following simple algorithm, which updates the minimization problem separately for the two different variables:

Algorithm 9: Alternate Minimization Algorithm

```
Input  : The matrices $A$, $C$, $D$ and the bases $W_1^{-1}, \dots, W_B^{-1}$, a collection of
         thresholds $\tau_1, \tau_2, \tau_B$ and a maximum number of iterations $M$.
Output: An estimate of the minimizing $v$ and $\bar{w}$ and solution in voxel space
         $\bar{x}$
$\bar{w}^0 = (0, \dots, 0)^T$;
$v^0 = 0$;
for $n = 0, 1, \dots, M$ do
    $K = b - Cv^n$;
    $\bar{w}^{n+1} = \arg\min_{\bar{w}} ||M\bar{w}^n - K||_2^2 + \sum_{j=1}^{B} \tau_j ||w_j||_1$;
    Solve the linear system: $(C^TC + D^TD)v^{n+1} = C^T(b - M\bar{w})$;
    $x^{n+1} = W_1^{-1}w_1 + \dots W_B^{-1}w_B$;
end
$\bar{x} = x^{n+1}$;
```

As an alternative to this algorithm, we may consider using the ADMM method introduced in Chapter 2. We now briefly describe this second approach to handling (6.5.1). Consider now for simplicity of analysis that $\tau_1 = \tau_2 = \cdots = \tau_B = \tau$ and introduce the matrix M that we just defined. Thus, we have:

$$\min_{\bar{w},v} \left[||M\bar{w} + Cv - b||_2^2 + ||Dv||_2^2 + \tau ||\bar{w}||_1 \right] \tag{6.5.2}$$

We scale by $\frac{1}{2}$ and rewrite as:

$$\min_{\bar{w},v} \left[\frac{1}{2}||M\bar{w} + Cv - b||_2^2 + \frac{1}{2}||Dv||_2^2 + \frac{1}{2}\tau ||z||_1 \right] \quad \text{s.t.} \quad \bar{w} - z = 0 \tag{6.5.3}$$

Now we note that the smooth part $\frac{1}{2}||M\bar{w} + Cv - b||_2^2 + \frac{1}{2}||Dv||_2^2$ and the non-smooth part $\frac{1}{2}\tau||z||_1$ are separable, so the ADMM method can be applied. The augmented Lagrangian functional for (6.5.3) becomes:

$$L_\mu(\bar{w}, v, z, y) = \frac{1}{2}||M\bar{w} + Cv - b||_2^2 + \frac{1}{2}||Dv||_2^2 + \frac{1}{2}\tau||z||_1 + y^T(w - z) + \frac{\mu}{2}||\bar{w} - z||_2^2$$

The above is differential in w and v with gradients:

$$\begin{aligned}
\nabla_{\bar{w}} L &= M^T(M\bar{w} + Cv - b) + y + \mu(\bar{w} - z) \\
\nabla_v L &= C^T(M\bar{w} + Cv - b) + D^T Dv
\end{aligned}$$

Thus, minimizing the Lagrangian over $\bar{w}$, v, and z yields:

$$\begin{aligned}
\nabla_{\bar{w}} L = 0 \quad &\Longrightarrow \quad M^T(M\bar{w} + Cv - b) + y + \mu(\bar{w} - z) = 0 \\
\nabla_v L = 0 \quad &\Longrightarrow \quad C^T(M\bar{w} + Cv - b) + D^T Dv = 0 \\
\therefore \quad \arg\min_z L(\bar{w}, v, z, y) \quad &= \quad \arg\min_z \left[\frac{1}{2}\tau||z||_1 - y^T z + \frac{\mu}{2}||\bar{w} - z||_2^2\right] \\
&= \quad \arg\min_z \left[\frac{\tau}{\mu}||z||_1 - \frac{2}{\mu}y^T z + ||z - \bar{w}||_2^2\right] \\
&= \quad \arg\min_z \left[\frac{\tau}{\mu}||z||_1 + ||z - (\bar{w} + \frac{1}{\mu}y)||_2^2\right] \\
&= \quad \mathbb{S}_{\frac{\tau}{2\mu}}\left(\bar{w} + \frac{1}{\mu}y\right).
\end{aligned}$$

Rearranging $\nabla_{\bar{w}} L = 0$ and $\nabla_v L = 0$ we have:

$$\begin{aligned}
& M^T(M\bar{w} + Cv) + \mu\bar{w} = M^T b - y + \mu z \\
\Longrightarrow \quad & (M^T M + \mu I)\bar{w} + M^T Cv = M^T b - y + \mu z \\
& C^T(M\bar{w} + Cv) + D^T Dv = C^T b \\
\Longrightarrow \quad & C^T M\bar{w} + (C^T C + D^T D)v = C^T b.
\end{aligned}$$

Thus, setting $\bar{w}^{n+1} = \arg\min_{\bar{w}} L_\mu(\bar{w}, v, z^n, y^n)$ and $v^{n+1} = \arg\min_v L_\mu(\bar{w}, v, z^n, y^n)$, we have the update system:

$$\begin{bmatrix} M^T M + \mu I & M^T C \\ C^T M & C^T C + D^T D \end{bmatrix} \begin{bmatrix} \bar{w}^{n+1} \\ v^{n+1} \end{bmatrix} = \begin{bmatrix} M^T b - y^n + \mu z^n \\ C^T b \end{bmatrix}$$

followed by the updates:

$$
\begin{aligned}
z^{n+1} &= \mathbb{S}_{\frac{\tau}{2\mu}}\left(\bar{w}^{n+1} + \frac{1}{\mu}y^n\right) \\
y^{n+1} &= y^n + \mu(\bar{w}^{n+1} - z^{n+1}) \\
\mu &= \rho\mu
\end{aligned}
$$

for $\rho > 1$. In practice however, the above linear system for $\bar{w}$ and v is difficult to implement directly. Numerically, we would need to use some kind of an alternate scheme to update the two in sequence. We summarize one possible ADMM approach below:

Algorithm 10: ADMM Based Minimization Algorithm

```
Input  : The matrices $M = (AW_1^{-1}, \ldots, AW_N^{-1})$, $C$, and $D$; a set of thresholds $\tau_1, \tau_2, \ldots, \tau_N$; a parameter $\rho > 1$ and a maximum number of iterations $M_i$.
Output: An estimate of the minimizing $v$ and $\bar{w}$ and solution in voxel space $\bar{x}$
$\bar{w}^0 = (0, \ldots, 0)^T$;
$v^0 = 0$;
$\mu = 1$;
for $n = 0, 1, \ldots, M_i$ do
    $(M^T M + \mu I)\bar{w}^{n+1} = -M^T C v^n + \mu z^n - y^n + M^T b$;
    $(C^T C + D^T D)v^{n+1} = -C^T M \bar{w}^{n+1} + C^T b$;
    $z^{n+1} = \mathbb{S}_{\frac{\tau}{2\mu}}\left(w^{n+1} + \frac{1}{\mu}y^n\right)$;
    $y^{n+1} = y^n + \mu(\bar{w}^{n+1} - z^{n+1})$;
    $\mu = \rho\mu$;
    $x^{n+1} = M\bar{w}^{n+1}$;
end
$\bar{x} = x^{n+1}$;
```

We now present some solutions. For our data set, the norm of the noise in the data is not known. To compare our solutions with previous models, we must use another measure of reconstruction quality: χ^2. The reduced χ^2 value (normalized by the number of degrees of freedom):

$$\chi^2 = \sum_{k=1}^{N} \frac{|(Ax-b)_k|^2}{N}$$

is a measure of the goodness of fit. A value close to one indicates a good fit within the noise level in the supplied data b. We compare our reconstructions to those obtained in [39]. Based on the discussion in the above reference, we look at solutions with a χ^2 value of about 0.6. We now present some reconstructions with different choices of bases. Each basis is defined by the wavelet name and the number of levels in the transform. For example, 'cdf 2' refers to the CDF-42 transform with 2 levels.

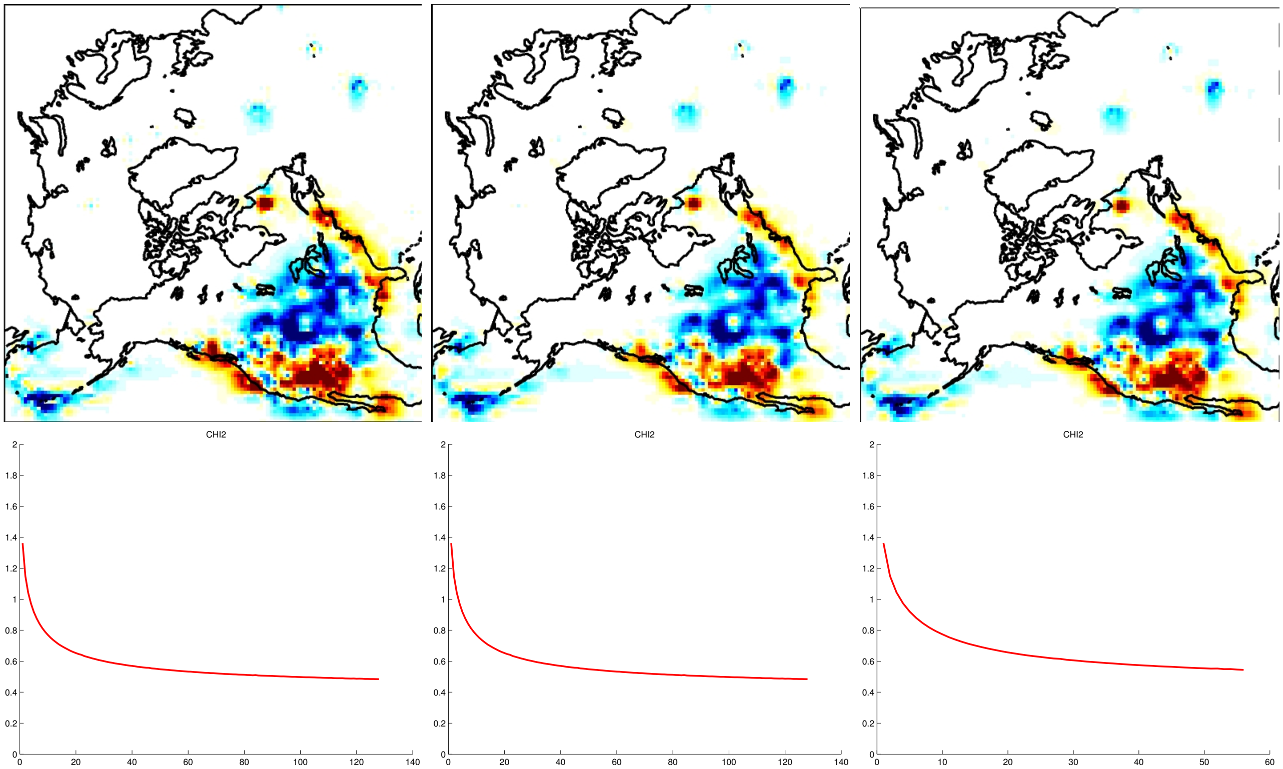


Figure 6.9: Three outputs of linear combinations of three wavelets. The plots at the top show the inversion results with different sets of wavelet basis. Left: haar 2, cdf 2, cdf 1; middle: cdf3, cdf2, cdf; left: cdf 2, cdf 1, and identity (no wavelets); The bottom plots show chi-square values versus iteration (the plots are all at about the same level).

Next we compare the dictionary wavelet solution to the solution of the data set obtained with the LSQR algorithm [34] (basically a Tikhonov regularization approach, see Chapter 2) at a similar χ^2 value; the scaling of the two is different because different units are used:

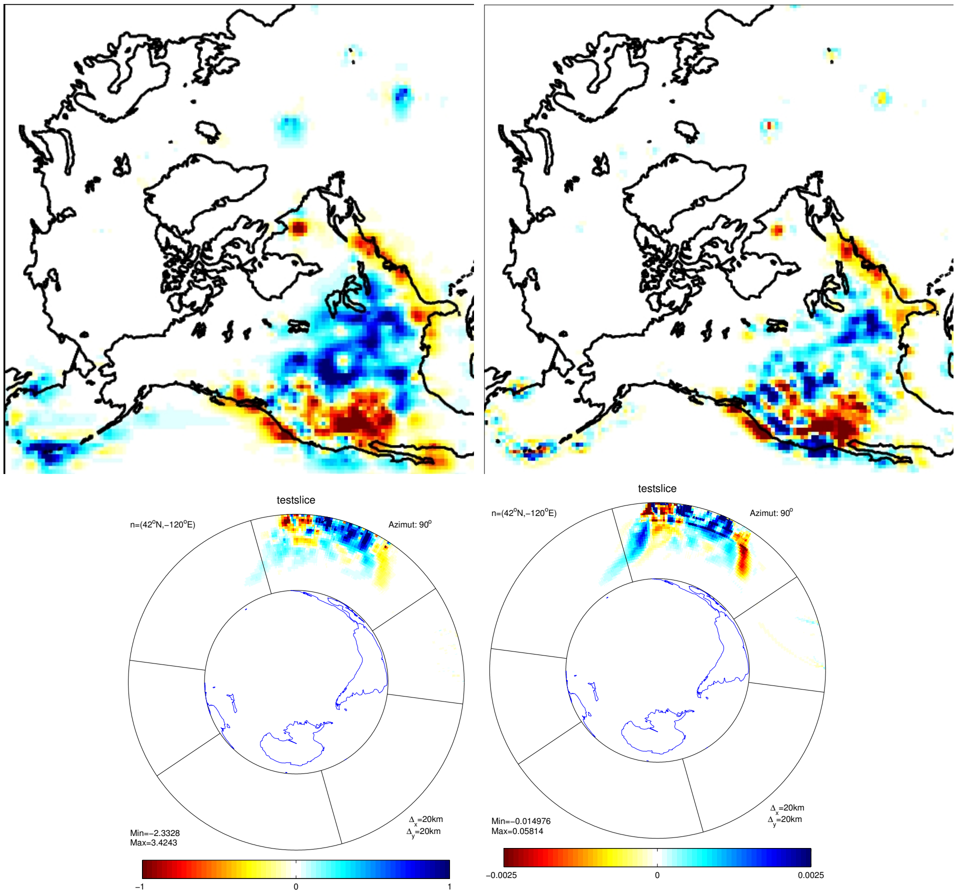


Figure 6.10: Mixed norm dictionary wavelet solution with cdf2, haar2, and identity bases (left) versus LSQR solution (right). Right figure courtesy of Jean Charlety.

In the mixed norm dictionary wavelet solution above, we imposed the ℓ_2 penalty on the scaling coefficients of the cdf and haar wavelets and the coefficient of the identity basis and the ℓ_1 penalty on all other coefficients. Of course, different choices are possible leading to slightly different results. We make two comments about the above results: the first is that the χ^2 value of both plots is about the same, while the plots look somewhat different. The second comment is that we see more detail in the middle of the US with the dictionary wavelet solution, although it is also somewhat smoother due to the use of the cdf transform. It is hard to prefer one over the other just from the above picture, but we claim that the dictionary wavelet approach is more general to using Tikhonov regularization, because when the true model makes it possible, the use of different bases and penalties (including ℓ_2 penalty on identity) will bring out more details in the final result as compared to the result obtained using just a single ℓ_2 penalty on the identity basis as in the LSQR algorithm.

6.6 Brief Description of Developed Software

In this section we briefly summarize the developed software. The software package consists of codes that allow one to do the inversion based on the above discussion. The above reconstructions, including the ones with real data presented in this chapter, have been obtained using this package. We mention in particular that a parallel matrix vector multiply package has been developed. This package allows a user to perform matrix vector and matrix transpose vector multiplications with very large matrices inside the MATLAB environment, by overloading the default MATLAB matrix vector multiplication operation and executing instead a parallel C code to do the operation. In addition, the package handles, implicitly, the use of a dictionary of different wavelet transforms.

6.7 Chapter Remarks and Conclusions

In this chapter, we presented the application in Geotomography and gave details of the corresponding inverse problem which is solved in the form of a least squares problem with constraints. We summarized the mathematical details leading up to the inversion and showed sample reconstructions obtained using schemes that were discussed in the previous chapters of the thesis. The main takeaway from this chapter is a construction that is of use to this and other applications. In particular, the idea of using a dictionary of different bases should be appealing for many models with different sets of features. This chapter gives an example of how the mathematical material developed in the previous chapters may be applied to inverse problems from the physical sciences.

Chapter 7

SUMMARY AND CONCLUSIONS

In this chapter, we summarize the key ideas presented in the first six chapters of this thesis. The methods presented in this thesis grew out of the work in an application in Geotomography, and the constraints and challenges imposed by this problem gave the motivation for the algorithms and techniques that are presented in the previous chapters. This thesis has contributions in three parts: new numerical techniques (which are easy to implement and test, and which may be useful for different applications apart from the one we describe), detailed analytical derivations for some of these techniques (such as derivations regarding optimality conditions and algorithmic properties such as boundedness that would be useful for proving properties of related algorithms), and a tested large scale computational framework, including software, for the inverse problem.

The inverse problem, described in detail in Chapter 6, has several key characteristics which impact the type of algorithms that may be used to find reconstructions. It involves a very large underdetermined linear system with a noisy data vector and a badly conditioned matrix, from which we would like to find a solution in the least squares sense, since we cannot hope to find a solution to the exact constrained problem because of the noise. Since the linear system is not well conditioned and underdetermined, additional constraints must be imposed to make the problem well posed. We use here the assumption (verified in practice) that a sparse solution may be found under the action of a transform (for example a wavelet transform) and these are the solutions we are interested in obtaining. Since the system we work with in the application is very big (over three million columns), we focus on simple, easy to implement and easy to parallelize methods which do not require significantly more than matrix vector and matrix transpose vector multiplications. The reason for this is that access to the full matrix for our problem is not easily available since that would require encoding the different rows with a specific choice of wavelet transform. Since we would like to be able to use different transforms, it is easiest for us to make use of methods which require only the result of matrix vector multiplications rather than the individual elements of the matrix. We now describe the contents of the different chapters.

Following the Introduction, Chapter 2 of the thesis gave an introduction to sparse regularization. Two typical constraints are often enforced to give sparse solutions: the minimization of the ℓ_0 and ℓ_1 norms. In the context of noisy problems, this corresponds to solving the problems: $\min ||x||_0$ s.t. $||Ax-b||_2 \leq \epsilon$ and $\min ||x||_1$ s.t. $||Ax-b||_2 \leq \epsilon$ which for some constant τ correspond to minimizing the ℓ_0 and ℓ_1 functionals:

$$G(x) = ||Ax-b||_2^2 + 2\tau||x||_0 \quad \text{and} \quad F(x) = ||Ax-b||_2^2 + 2\tau||x||_1$$

When x is expected to be sparse only under a certain transform (in wavelets in our case), we can replace $||x||_1$ above with $||Wx||_1$ where W denotes the desired transform. While directly solving the ℓ_0 problem is combinatorially hard, a wide variety of approximate methods from compressed sensing exist, that give the right solution with high probability if the right constraints on the system are satisfied. However, the ℓ_0 penalty, which counts the number of nonzero elements, is highly non-convex and difficult to deal with numerically, because of the existence of local minima. Additionally, methods which minimize the ℓ_0 functional directly, often rely on the matrix satisfying the Restricted Isometry Property, which amounts to satisfying a condition of the form:

$$(1-\sigma_s)||y||_2^2 \leq ||A_s y||_2^2 \leq (1+\sigma_s)||y||_2^2$$

for every s-columned submatrix A_s of A for some small constant σ_s and sparse vectors y. This is problematic for large matrices which are not well conditioned, since any such matrix A_s would likely have a nonzero null space that contains some sparse vectors (which would mean that $A_s y = 0$ and violate the RIP condition). Thus, in Chapter 2, we come to the conclusion that methods based on the ℓ_0 norm are difficult to use for problems with our requirements. Hence, the methods developed in this thesis do not attempt to work directly with the ℓ_0 norm.

We go on to describe several classes of methods for ℓ_1 optimization, which involves a convex functional with a single global minimum. As discussed in Chapter 2, the main difficulty here is treating the non-smooth term $||x||_1 = \displaystyle\sum_{k=1}^{N} |x_k|$. There are two possibilities: to leave the term as is or to replace the term by a smooth approximation. If the non-smooth term is left untouched, subgradient methods must be used since the term is not differentiable. Subgradient methods then lead to the soft-thresholding operation which is a component wise nonlinear operator defined via the minimization problem: $\mathbb{S}_\tau(b) = \arg\min_x ||x-b||^2 + 2\tau||x||_1$. A simple scheme called the Iterative Soft Thresholding Algorithm (ISTA) can then be used to minimize the ℓ_1 functional. The scheme can be derived from a so called majorization-minimization approach, where instead of minimizing the original function, we minimize the function which majorizes it, resulting in a simpler problem and a very straightforward algorithm:

$$
\begin{aligned}
& x^{n+1} = \arg\min_{x} ||Ax - b||_2^2 + ||x - x^n||_2^2 - ||A(x - x^n)||_2^2 + 2\tau||x||_1 \\
\implies & x^{n+1} = \mathbb{S}_\tau(x^n + A^T b - A^T A x^n)
\end{aligned}
$$

By use of the above majorization-minimization approach, we show in Chapter 2 that it makes it easy to show key properties of the above scheme: such as the boundedness of (x^n) and $||x^{n+1} - x^n||_2 \to 0$, and makes it possible to prove that all limit points from the subsequences of sequence (x^n) satisfy the correct optimality conditions for the ℓ_1 functional. This is shown to hold with minimal assumptions on the spectral norm of the matrix: $||A||_2 < 1$ and for any initial guess x^0. The analysis we present here, for this previously well known and popular scheme, is key to the rest of the thesis, where we use similar analytical techniques to analyze more complicated schemes introduced in the later chapters.

The disadvantage of ISTA is its slow speed of convergence. In particular, it was shown in [2] that:

$$
F(x^n) - F(\bar{x}) \leq \frac{C||x^0 - \bar{x}||_2^2}{2n}
$$

where $\bar{x} = \lim_{n\to\infty} x^n$. This alone motivates the search for different algorithms, which are of about the same complexity but which converge faster or minimize a more general functional. For a faster scheme, we introduce in Chapter 2 the existing FISTA algorithm which employs a simple trick from the work of Nesterov [32] and performs the same soft thresholding operation on a linear combination of the past two solutions:

$$
\begin{aligned}
z^n &= x^{n-1} + \frac{t_{n-1} - 1}{t_n}(x^{n-1} - x^{n-2}) \quad , \quad t_n \in \mathbb{R} \\
x^{n+1} &= \mathbb{S}_\tau(z^n + A^T b - A^T A z^n)
\end{aligned}
$$

FISTA, by this simple change, has a significantly faster rate of convergence:

$$F(x^n) - F(\bar{x}) \leq \frac{C_2||x^0 - \bar{x}||_2^2}{(n+1)^2}$$

We then turn to discussing other fast schemes based on approaches that do not involve the direct use of soft thresholding. Another fast algorithm, DALM, also discussed in Chapter 2, is based on the dual space approach, which we revisit later in Chapter 5 in the context of reweighted norms. The idea behind the method is to replace the constrained version of the ℓ_1 minimization problem by its dual, yielding a method of comparable speed to the FISTA scheme (but without a proved convergence bound like the above). Chapter 2 introduces other interesting schemes, including coordinate descent and the ADMM algorithm, both of which we discuss again later.

Chapter 3 introduced a method based on a different two-parameter thresholding function $\mathbb{S}_{\rho,\tau}$. While the FIVTA algorithm no longer minimizes the ℓ_1 functional, it was shown to produce solutions of comparable sparsity and with some improved numerical properties. The most obvious of these, is the ability to provide a similar end residual level $||A\bar{x} - b||_2$ using a higher τ compared to FISTA. That is, for a given τ, if $\bar{x}$ is the solution obtained with FIVTA and $\bar{y}$ the FISTA solution at the same τ, we observe that $||A\bar{x} - b||_2 < ||A\bar{y} - b||_2$. Hence, FIVTA can produce comparable solutions to FISTA at a significantly higher τ. At this higher τ the observed numerical convergence of FIVTA is significantly faster than that of FISTA at the corresponding lower τ used to obtain the same final residual value. The fact that soft thresholding is not necessarily optimal for sparse applications is not surprising. In particular, non-convex minimization (which can be done with a different thresholding function from soft thresholding) has been shown to yield better results in compressive sensing applications, by being able to recover more with fewer available data; see for example [9].

In Chapter 4, instead of treating the non-smooth term of the ℓ_1 functional directly and using the soft thresholding operator, we have presented two algorithms that arise out of replacing the non-smooth portion by a smooth approximation. One such approximation is possible, simply by convoluting the absolute value function with a smooth function having a shrinking support, such as a narrow Gaussian, which was shown in Chapter 2. This simple approach results in a smooth approximation to the ℓ_1 functional but is rather crude and does not give good numerical results. For the two algorithms in Chapter 4, we instead use a reweighted approach that gets more accurate as the iterations progress:

$$||x||_1 = \sum_{k=1}^{N} |x_k| \approx \sum_{k=1}^{N} w_k^n x_k^2$$

where to minimize the ℓ_1 functional, we use the weight $w_k^n = \frac{1}{\sqrt{(x_k^n)^2 + (\epsilon_n)^2}}$ with the parameter $\epsilon_n \to 0$. The trick to proving convergence turns out to be in picking the parameter sequence (ϵ_n) in a suitable way so that the limit points of a converging subsequence satisfy the required optimality conditions and for that we start with the same analytical tools introduced in Chapter 2 for the ISTA scheme. Using a majorization-minimization approach similar to that of ISTA, we derived the first IRLS algorithm:

$$x_k^{n+1} = \frac{1}{1 + \tau w_k^n} \left(x_k^n + A^T b - A^T A x_k^n \right)$$

being similar in form to the ISTA scheme. We showed full convergence results using the ϵ_n defined simply by $\epsilon_n = \min(\epsilon_{n-1}, \sqrt{||x^n - x^{n-1}|| + \alpha^n})$ for some small $\alpha > 0$ and some initial $\epsilon_0 = 1$. We show that with minor modifications for the weights, the scheme can be extended to minimize a new, more general functional:

$$||Ax - b||_2^2 + 2 \sum_{k=1}^{N} \lambda_k |x_k|^{q_k} \quad , \quad 1 \leq q_k < 2$$

which is useful in a structured sparse application, such as the one in our application (the structure being imposed by the wavelet transform). The analysis of this functional and of the two IRLS algorithms that minimize it is an important new result of this thesis.

By similar construction to the ISTA scheme using majorization-minimization, we prove several favorable properties, amongst them that $||x^n - x^{n-1}||_2 \to 0$ and that (x^n) is bounded. We then construct a special converging subsequence and show that its limit point satisfies the optimality conditions of the above generalized functional. In fact, for a more powerful statement, we show that all limit points from any converging subsequence satisfy the optimality conditions.

In the second part of Chapter 4, we discuss a scheme that is more complicated at each step, yet more powerful and with better numerical performance. The scheme reads simply as:

$$x^{n+1} = (A^T A + \tau \Phi_n)^{-1} A^T b$$

looking somewhat as a generalization of the classical Tikhonov algorithm introduced in Chapter 2. In the above, Φ_n is a diagonal matrix containing the elements $\lambda_k q_k w_k^n$ where w_k^n are defined as before. At each iteration, this IRLS method requires a linear solve, which can be done, for instance, using some variant of a conjugate gradient method. At this step, the solution of the previous iteration can be used as a warm start for the linear solve so that at later iterations very few inner linear solve iterations are required. With this scheme, the number of outer iterations is significantly smaller and the total runtime can be reduced. The difficulty in the proofs lies again in picking the right subsequence ϵ_n, which is more challenging in this case, since the estimate $||x^{n+1} - x^n||_2 \to 0$ does not (at least readily) come about. The fact that despite this, convergence analysis has been exhibited (to the generalized functional using the same generalized weights as above), should be of analytical interest for the analysis of similar algorithms in the future. We again show that all limit points from any converging subsequence satisfy the optimality conditions. The second IRLS scheme for the minimization of the new generalized functional is a powerful new numerical method for various different applications, including the one we consider.

Chapter 5 of the thesis introduced numerical techniques. We discuss another way to compute the iterates of the second IRLS scheme, which in the case of underdetermined systems, results in the use of a smaller inverse matrix. Since in the scheme $x^{n+1} = (A^T A + \tau \Phi_n)^{-1} A^T b$ the only thing that changes between iterations is the diagonal matrix Φ_n, ideas for approximation of the inverse update are possible. In the context of rewighted norms introduced in Chapter 4, we mention that the idea can be applied also to other algorithms. In particular, we discuss a mixed norm variant of the dual space augmented lagrangian method introduced in Chapter 2. Although not thoroughly analyzed, the method seems to have good numerical performance. We mention also more practical approaches to the coordinate descent scheme, as well as randomized algorithms for computing the low rank SVD approximation of a matrix and an approach for estimating the column norms. Both are useful for large problems since direct techniques (for example the full or even partial SVD) are far too expensive to compute for large problems without the use of randomization. The randomized SVD algorithm provides a way of obtaining approximate solutions to large problems and has been tested through our application on very large matrices. We conclude the chapter with some numerical illustrations of the different methods.

In Chapter 6, we presented the application in Geotomography, discussing in more detail the ideas behind the big underdetermined linear system that we work with. After a simplified discussion of the physics of the inverse problem for the corrections to the spherically symmetric wave velocity model, we discussed more details about the matrix from our data set and the cubed sphere coordinate system. We then posed the final form of the inverse problem which contains additional terms due to corrections to the available data. We presented two approaches to solving this final form, based on alternate minimization and the ADMM scheme first introduced in Chapter 2. We discussed the application of the previously mentioned algorithms to the inverse problem and interesting approaches such as the mixed norm formulation, where we use a collection of different basis to represent the solution. We feel that this new mixed basis approach, easily applicable using the developed IRLS algorithms, has an advantage over the classical ℓ_2 penalty technique employed for many years by Geophysicists for these types of problems. Additionally, we briefly discussed the developed software framework for solving the inverse problem, taking advantage of parallel computing.

Chapter 8

APPENDIX

8.1 Overview

We provide here the pseudocode for some of the algorithms discussed in the text. In particular, we present the pseudocode for the newly introduced methods for which full analysis has been provided. We include the pseudocode for the following algorithms from the text below: FISTA, FIVTA, IRLS (ℓ_1 version), IRLS with FISTA speedup (which allows the FISTA speedup to be used for the mixed norm case) and IRLS SYS.

8.2 Pseudocode of Algorithms

Algorithm 11: FISTA Algorithm

```
Input  : An $m \times N$ matrix $A$, an initial guess $N \times 1$ vector $x^0$, a parameter $\tau < \max_i(|(A^T b)_i|)$, tolerance $\epsilon$, and the maximum number of iterations $M$.
Output: A vector $\bar{x}$, close to the vector minimizing the $\ell_1$ functional.
$y^0 = x^0$;
$t_1 = 1$;
for $n = 0, 1, \ldots, M$ do
    $x^{n+1} = \mathbb{S}_\tau(y^n + A^T b - A^T A y^n)$;
    $t_{n+1} = \frac{1+\sqrt{1+4t_n^2}}{2}$;
    $y^{n+1} = x^n + \frac{t_n - 1}{t_{n+1}}(x^n - x^{n-1})$;
    if $\|x^n - x^{n+1}\| \leq \epsilon$ then
        break
    end
end
$\bar{x} = x^{n+1}$;
```

Algorithm 12: FIVTA for Sparse Signal Recovery

```
Input  : An $m \times N$ matrix $A$ and a vector $b \in \mathbb{R}^m$, a leverage $L$, tolerance $\epsilon$,
         an estimated sparsity level $K$, and an initial guess $x^0$.
Output: An estimate $\hat{x} \in \mathbb{R}^N$ of the signal $x$
$i \leftarrow 0$;
$\rho_0 \leftarrow \tau$;
$K_0 \leftarrow \frac{N}{5}$;
$L_0 \leftarrow K_0||A^*b||_1$;
$a \leftarrow 0$;
begin
    if $i = 1$ then
        $x^{i+1} \leftarrow \mathbb{S}_{\rho_i,\tau}(x^i + A^*b - A^*Ax^i)$;
    end
    else
        $y^{i+1} = x^i + \frac{t_i-1}{t_{i+1}}(x^i - x^{i-1})$;
        $x^{i+1} \leftarrow \mathbb{S}_{\rho_i,\tau}(y^i + A^*b - A^*Ay^i)$
    end
    if $\|x^i - x^{i+1}\| \leq \epsilon$ then
        break
    end
    $K_{i+1} \leftarrow nnz(x^{i+1})$;
    $\sigma \leftarrow \frac{\tau}{2}(1 + \frac{\max(K_{i+1}-K_0,a)}{N-K_0})$;
    if $\sigma \leq \rho_i$ then
        $L_{i+1} \leftarrow L_i$, $\rho_{i+1} \leftarrow \sigma$
    end
    else
        $L_{i+1} \leftarrow L_i + h_{\rho_i,\tau}(x^{i+1}) - h_{\sigma,\tau}(x^{i+1})$
        if $L_{i+1} \geq 0$ then
            $\rho_{i+1} \leftarrow \sigma$
        end
        else
            $\rho_{i+1} \leftarrow \max(\rho_i, \frac{\tau}{2}(1 + \frac{1}{N-K_0})), a \leftarrow 1$
        end
    end
    $i \leftarrow i + 1$;
end
$\hat{x} \leftarrow x^{(i)}$
```

Algorithm 13: IRLS ℓ_1 version

```
Input  : An $m \times N$ matrix $A$, an initial guess $N \times 1$ vector $x^0$, a parameter
         $\tau < \max_i(|(A^T b)_i|)$, a tolerance $\epsilon$, and the maximum number of
         iterations $M$.
Output: A vector $\bar{x}$, close to the vector minimizing the $\ell_1$ functional.

$\alpha = 10^{-3}$;
$\epsilon_0 = 1$;
for $k = 1, \ldots, N$ do
    $w_k^0 = \frac{1}{\sqrt{(x_k^0)^2 + \epsilon_0^2}}$;
    $x_k^1 = \frac{1}{1+\tau w_k^0}(x_k^0 + (A^T b)_k - (A^T A x^0)_k)$;
end
for $n = 1, 2, \ldots, M$ do
    for $k = 1, 2, \ldots, N$ do
        $\epsilon_n = \min(\epsilon^{n-1}, \sqrt{||x^n - x^{n-1}||} + \alpha^n)$;
        $w_k^n = \frac{1}{\sqrt{(x_k^n)^2 + \epsilon_n^2}}$;
        $x_k^{n+1} = \frac{1}{1+\tau w_k^n}(x_k^n + (A^T b)_k - (A^T A x^n)_k)$;
    end
    if $\|x^n - x^{n+1}\| \leq \epsilon$ then
        break
    end
end
$\bar{x} = x^{n+1}$;
```

Algorithm 14: IRLS with FISTA speedup ℓ_1 version

```
Input  : An $m \times N$ matrix $A$, an initial guess $N \times 1$ vector $x^0$, a parameter
         $\tau < \max_i(|(A^T b)_i|)$, a tolerance $\epsilon$, and the maximum number of
         iterations $M$.
Output: A vector $\bar{x}$, close to the vector minimizing the $\ell_1$ functional.

$\alpha = 10^{-3}$;
$\epsilon_0 = 1$;
for $k = 1, \ldots, N$ do
    $w_k^0 = \frac{1}{\sqrt{(x_k^0)^2 + \epsilon_0^2}}$;
    $x_k^1 = \frac{1}{1+\tau w_k^0}(x_k^0 + (A^T b)_k - (A^T A x^0)_k)$;
end
$y^1 = x^1$;
$t_1 = 1$;
for $n = 1, 2, \ldots, M$ do
    for $k = 1, 2, \ldots, N$ do
        $\epsilon_n = \min(\epsilon^{n-1}, \sqrt{||x^n - x^{n-1}||} + \alpha^n)$;
        $w_k^n = \frac{1}{\sqrt{(x_k^n)^2 + \epsilon_n^2}}$;
        $x_k^{n+1} = \frac{1}{1+\tau w_k^n}(y_k^n + (A^T b)_k - (A^T A y^n)_k)$;
    end
    if $\|x^n - x^{n+1}\| \leq \epsilon$ then
        break
    end
    $t_{n+1} = \frac{1+\sqrt{1+4t_n^2}}{2}$;
    $y^{n+1} = x^n + \frac{t_n - 1}{t_{n+1}}(x^n - x^{n-1})$;
end
$\bar{x} = x^{n+1}$;
```

Algorithm 15: IRLS SYS ℓ_1 version

```
Input  : An $m \times N$ matrix $A$, an initial guess $N \times 1$ vector $x^0$, a parameter
         $\tau < \max_i(|(A^T b)_i|)$, a tolerance $\epsilon$, and the maximum number of
         iterations $M$.
Output: A vector $\bar{x}$, close to the vector minimizing the $\ell_1$ functional.

$\alpha = 10^{-3}$;
$\epsilon_0 = 1$;
for $k = 1, 2, \ldots, N$ do
    $w_k^0 = \frac{1}{\sqrt{(x_k^0)^2 + (\epsilon_0)^2}}$;
    $(\Phi_0)_{k,k} = \tau w_k^0$;
end
Solve: $(A^T A + \Phi_0)x^1 = A^T b$;
$\epsilon_1 = \min(\epsilon_0, \frac{1}{N}\sqrt{||x^1 - x^0||})$;
for $k = 1, 2, \ldots, N$ do
    $w_k^1 = \frac{1}{\sqrt{(x_k^1)^2 + (\epsilon_1)^2}}$;
    $(\Phi_1)_{k,k} = \tau w_k^1$;
end
Solve: $(A^T A + \Phi_1)x^2 = A^T b$;
for $n = 2, 3, \ldots, M$ do
    $\epsilon_n = \min(\epsilon_{n-1}, |G(x^{n-1}, w^{n-1}, \epsilon^{n-1}) - G(x^{n-2}, w^{n-2}, \epsilon^{n-2})|^{\frac{1}{4}} + \alpha^n)$;
    With: $G(x, w, \epsilon) = ||Ax - b||_2^2 + \tau \sum_{k=1}^{N} \left( w_k(x_k^2 + \epsilon^2) + \frac{1}{w_k} \right)$;
    for $k = 1, 2, \ldots, N$ do
        $w_k^n = \frac{1}{\sqrt{(x_k^n)^2 + (\epsilon_n)^2}}$;
        $(\Phi_n)_{k,k} = \tau w_k^n$;
    end
    Solve: $(A^T A + \Phi_n)x^{n+1} = A^T b$;
    if $\|x^n - x^{n+1}\| \leq \epsilon$ then
        break
    end
end
$\bar{x} = x^{n+1}$;
```

Chapter 9

Bibliography

[1] Amir Beck and Marc Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Trans. Image Processing*, 18(11):2419 –2434, Nov. 2009.

[2] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

[3] Thomas Blumensath. Accelerated iterative hard thresholding. *Signal Processing*, 92(3):752 – 756, 2012.

[4] Thomas Blumensath and Mike E. Davies. Iterative thresholding for sparse approximations. *J. Fourier Anal. Appl.*, 14(5-6):629–654, 2008.

[5] Thomas Blumensath and Mike E. Davies. Iterative hard thresholding for compressed sensing. *Appl. Comput. Harmon. Anal.*, 27(3):265–274, 2009.

[6] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, January 2011.

[7] Emmanuel J. Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inform. Theory*, 52(2):489–509, 2006.

[8] Emmanuel J. Candes and Terence Tao. Decoding by linear programming. *IEEE Trans. Inform. Theory*, 51(12):4203–4215, 2005.

[9] Rick Chartrand. Fast algorithms for nonconvex compressive sensing: Mri reconstruction from very few data. In *Int. Symp. Biomedical Imaing*, 2009.

[10] F. A. Dahlen, S.-H. Hung, and Guust Nolet. Frechet kernels for finite-frequency traveltimes. *Geophysical Journal International*, 141(1):157–174, 2000.

[11] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

[12] Ingrid Daubechies, Ronald DeVore, Massimo Fornasier, and C. Sinan Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Comm. Pure Appl. Math.*, 63(1):1–38, 2010.

[13] Ingrid Daubechies, Massimo Fornasier, and Ignace Loris. Accelerated projected gradient method for linear inverse problems with sparsity constraints. *J. Fourier Anal. Appl.*, 14(5-6):764–792, 2008.

[14] David Donoho, Iain Johnstone, and Andrea Montanari. Accurate prediction of phase transitions in compressed sensing via a connection to minimax denoising. *arXiv:1111.1041*, 2011.

[15] David L. Donoho. For most large underdetermined systems of linear equations the minimal l_1-norm solution is also the sparsest solution. *Comm. Pure Appl. Math.*, 59(6):797–829, 2006.

[16] David L. Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via l^1 minimization. *Proc. Natl. Acad. Sci. USA*, 100(5):2197–2202 (electronic), 2003.

[17] David L. Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via l^1 minimization. *Proc. Natl. Acad. Sci. USA*, 100(5):2197–2202 (electronic), 2003.

[18] David L. Donoho and Xiaoming Huo. Uncertainty principles and ideal atomic decomposition. *IEEE Trans. Inform. Theory*, 47(7):2845–2862, 2001.

[19] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Ann. Statist.*, 32(2):407–499, 2004. With discussion, and a rejoinder by the authors.

[20] Michael Elad. *Sparse and redundant representations.* Springer, New York, 2010. From theory to applications in signal and image processing, With a foreword by Alfred M. Bruckstein.

[21] H. Firouzi, M. Farivar, M. Babaie-Zadeh, and C. Jutten. Approximate Sparse Decomposition Based on Smoothed L0-Norm. *ArXiv e-prints*, November 2008.

[22] Massimo Fornasier and Holger Rauhut. Iterative thresholding algorithms. *Appl. Comput. Harmon. Anal.*, 25(2):187–208, 2008.

[23] Jerome H. Friedman and Werner Stuetzle. Projection pursuit regression. *J. Amer. Statist. Assoc.*, 76(376):817–823, 1981.

[24] Jean-Jacques Fuchs. On sparse representations in arbitrary redundant bases. *IEEE Trans. Inform. Theory*, 50(6):1341–1344, 2004.

[25] Gene H. Golub and Charles F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[26] William W. Hager. Updating the inverse of a matrix. *SIAM Rev.*, 31(2):221–239, 1989.

[27] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53(2):217–288, 2011.

[28] Per Christian Hansen and Dianne Prost O'Leary. The use of the L-curve in the regularization of discrete ill-posed problems. *SIAM J. Sci. Comput.*, 14(6):1487–1503, 1993.

[29] Ignace Loris, Guust Nolet, Ingrid Daubechies, and F. A. Dahlen. Tomographic inversion using ℓ_1-norm regularization. *Geophys. J. Int.*, 170:359–370, 2007.

[30] S.G. Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12):3397 –3415, dec 1993.

[31] D. Needell and J. A. Tropp. CoSaMP: iterative signal recovery from incomplete and inaccurate samples. *Appl. Comput. Harmon. Anal.*, 26(3):301–321, 2009.

[32] Yurii Nesterov. Gradient methods for minimizing composite objective function. *www.optimization-online.org*, (2007076), 2007.

[33] G. Nolet. *A Breviary of Seismic Tomography: Imaging the Interior of the Earth and Sun*. Cambridge University Press, 2008.

[34] Christopher C. Paige and Michael A. Saunders. LSQR: an algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Software*, 8(1):43–71, 1982.

[35] Shie Qian and Dapang Chen. Signal representation using adaptive normalized gaussian functions. *Signal Processing*, 36(1):1 – 11, 1994.

[36] R. Tyrrell Rockafellar. *Convex analysis.* Princeton Landmarks in Mathematics. Princeton University Press, Princeton, NJ, 1997. Reprint of the 1970 original, Princeton Paperbacks.

[37] C. Ronchi, R. Iacono, and P.S. Paolucci. The cubed sphere: A new method for the solution of partial differential equations in spherical geometry. *Journal of Computational Physics*, 124(1):93 – 114, 1996.

[38] Robert Sadourny. *Forced Geostrophic Adjustment in Large Scale Flow.* Laboratorie de Météorologie Dynamique, Paris, France, 1972.

[39] Karin Sigloch. *Multiple-frequency body-wave tomography.* Ph.d. thesis, Princeton University, mar 2008.

[40] Karin Sigloch, Nadine McQuarrie, and Guust Nolet. Two-stage subduction history under North America inferred from multiple-frequency tomography. *Nature Geoscience*, jun 2008.

[41] Frederik J. Simons, Ignace Loris, Guust Nolet, Ingrid C. Daubechies, S. Voronin, J. S. Judd, P. A. Vetter, J. Charlty, and C. Vonesch. Solving or resolving global tomographic models with spherical wavelets, and the scale and sparsity of seismic heterogeneity. *Geophysical Journal International*, 187(2):969–988, 2011.

[42] Joel A. Tropp and Anna C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inform. Theory*, 53(12):4655–4666, 2007.

[43] S. Voronin and H. Woerdeman. A new iterative firm-thresholding algorithm for inverse problems with sparsity constraints. *Applied and Computational Harmonic Analysis*, 2012.

[44] Yilun Wang and Wotao Yin. Sparse signal reconstruction via iterative support detection. *SIAM J. Imaging Sci.*, 3(3):462–491, 2010.

[45] Tong T. Wu and Kenneth Lange. Coordinate descent algorithms for lasso penalized regression. March 2008.

[46] Allen Y. Yang, Arvind Ganesh, Zihan Zhou, Shankar Sastry, and Yi Ma. A review of fast l1-minimization algorithms for robust face recognition. *CoRR*, abs/1007.3753, 2010.

[47] Junfeng Yang and Yin Zhang. Alternating direction algorithms for ℓ_1-problems in compressive sensing. *SIAM J. Sci. Comput.*, 33(1):250–278, 2011.