Technical University of Denmark

Master thesis

Sparsity regularization for inverse problems using curvelets

August 8, 2013

M.Sc.-2013-88

Author:
Jacob Larsen (s083421)

DTU Compute
Department of Applied Mathematics and Computer Science

Title page

This project, corresponding to a workload of 35 ECTS points and titled *Sparsity regularization for inverse problems using curvelets*, was carried out in the period February 4, 2013 – August 9, 2013 as a thesis concluding the Mathematical Modelling and Computation Master programme on DTU.

DTU Compute project number:
M.Sc.-2013-88

Student:
Jacob Larsen (*s083421*)

Supervisors:
Kim Knudsen, DTU Compute, Scientific Computing
Jakob Lemvig, DTU Compute, Mathematics

DTU Compute
Technical University of Denmark
Matematiktorvet, building 303B
DK-2800 Kgs. Lyngby, Denmark

Summary

In inverse problems information about a physical system is achieved from observations or measurements by reversing the effect of a model that acts on the sought information. Often this approach leads to mathematical problems without existence or uniqueness of a solution, or to problems with an unstable solution in the sense that small perturbations in the observations/measurements might cause unbounded changes to the solution. This issue is known as ill-posedness.

The concept of regularization deals with ill-posedness by replacing the original problem with a nearby problem that is not ill-posed. Different regularization methods are useful in different situation. If for instance a sought solution is known in advance to be an edge-containing image, this *a priori* information can be included in the regularized model to compensate for some of the ill-posedness.

The relatively new concept of curvelets provide a way to decompose $L^2(\mathbb{R}^2)$-functions into frame-coefficients. Much like for wavelets, the curvelet coefficients with high magnitudes indicate a jump discontinuity at a certain translate of the decomposed function. This feature can be included in a sparsity regularization model that promotes a solution to have many zero-valued curvelet coefficients. The sparsity promoting feature thus promotes a solution to contain edges.

This thesis reviews theory on curvelet based regularization in comparison with the more well established edge-preserving methods total variation and wavelet-based regularization. Further, two concrete inverse problems are used to demonstrate inversions using the three different regularization methods. Namely the de-blurring of a digital image and a computed tomography problem are considered.

The curvelet based regularization method shows result with qualities close to the results of total variation regularization in a 2D tomography problem, and the use of sparse expansions in regularization appears to have a promising future due to the great attention on the subject. The computational costs and extra efforts needed to implement curvelet based regularization compared to total variation does not justify a commercialization of the method in its current version. By construction curvelets capture orientation in more directions than wavelets on 2D domains and curevelets are therefore better suited than wavelets for 2D images with singularities along curves.

Resumé (*Danish*)

I inverse problemer opnås informationer om et fysisk system fra observationer eller målinger ved at invertere effekten af en model som virker på de søgte informationer. Denne fremgangsmåde vil typisk føre til matematiske problemer uden eksistens eller entydighed af en løsning, eller til problemer med en ustabil invers således at små forstyrrelser i observationerne/målingerne kan forårsage ubegrænsede ændringer i en løsning. Dette emne er kendt som ill-posedness.

Regularisering er en metode til at håndtere ill-posedness ved at erstatte problemet med et ikke ill-posed problem som ligger tæt på det oprindelige problem. Forskellige regulariserings-metoder er brugbare i forskellige situationer. Hvis det f.eks. vides på forhånd, at en søgt løsning er et billede med kanter, kan denne *a priori* oplysning inddrages i den regulariserede model for at kompensere for at problemet er ill-posed.

Curvelets er et relativt nyt koncept som gør det muligt at nedbryde $L^2(\mathbb{R}^2)$-funktioner i frame-koefficienter. I lighed med wavelets vil curvelet-koefficienter med stor værdi indikere en diskontinuitet ved et bestemt translat. Denne egenskab kan inddrages i en sparsity regulariserings model som fremmer, at løsningen har mange curvelet koeffictienter med værdi nul. Dette fremmer derfor, at en løsning indeholder kanter.

Dette speciale gennemgår teori om curvelet baseret regularisering i sammenligning med de mere veletablerede kant-bevarende metoder total variation og wavelet baseret regularisering. Derudover bliver to konkrete inverse problemer anvendt til at demonstrere inverteringer ved hjælp af de forskellige regulariserings metoder. Navnligt anvendes de-blurring af et digitalt billede og et tomografi problem.

Curvelet baseret regularisering viser resultater med kvaliteter tæt på resultater fra total variation regularisering i et 2D tomografi problem, og anvendelsen af sparse expansions i regularisering synes at have en lovende fremtid pga. den omfattende opmærksomhed på området. Curvelet baseret regularisering kræver mere computerkraft og er mere kompliceret at implementere i forhold til total variation regularisering og der er derfor ikke grundlag for at benytte metoden kommercielt i dens nuværende form. Curvelets opfanger orientering i flere retninger end wavelets på 2D domæner og curvelets er derfor bedre end wavelets til 2D billeder med kanter langs med kurver.

Preface

The content of this thesis is addressed to readers with some familiarity in especially mathematics, algebra and functional analysis. Experience with MATLAB programming is advantageous for readers wanting to reproduce numerical results of this work. All content is free to use.

Against the background of a M.Sc. Eng. on the DTU programme *Mathematical Modelling and Computation* following a B.Sc. Eng. on the DTU programme *Medicine and Technology* in the period September 2008 to August 2013, areas of particular interest to the undersigned are expressed in this work.

A special thanks to the two supervisors Kim and Jakob for their time and advises during the project. Also thanks to Ph.D. student Jakob Sauer Jørgensen and Post Doctoral student Martin Skovgaard Andersen for useful discussions. In general thanks to DTU and many highly skilled and very committed researchers/educators for providing a very interesting education on a high level.

Kgs. Lyngby, August 8, 2013

Jacob Larsen

Contents

Chapter 1

Introduction

1.1 Inverse problems, regularization and curvelets

Inverse problems and regularization are widely used in different industries to extract information of physical systems from observed measurements. Hospitals are examples where technology uses mathematics of inverse problems on daily basis to reconstructs images of interior regions of human bodies in different scanner techniques.

The topics of inverse problems and regularization are also currently fields subject to intense research due to their potential and the demand of methods to obtain solutions to inverse problems of e.g. better qualities. In particular the approach of adding constraints in a model by knowing or assuming certain properties of a sought solution is a concept showing interesting potentials to improve the qualities of reconstructed solutions. When working with images such an a priori information is for instance that the sought image contains edges to a considerable extend. By using appropriate mathematical theories a function associated with the sought image can be required to belong to a certain function space and thereby promoting that a solution contains edges.

The relatively new topic of curvelets, which has emerged from the more established topic of wavelets, is another field subject to intense research. Curvelets and wavelets (and also e.g. shearlets, ridgelets and contourlets) share the property that a mathematical function can be represented in a reorganized matter that is associated with certain properties of the function. As the name suggests curvelets are constructed to capture singularities in functions along curves. This feature is therefore expected to be well suited to capture edges in e.g. medical images.

This project reviews fundamental issues regarding inverse problems and regularization in general, and uses a couple of specific (simulated) inverse problems to examine the utility of a method to reconstruct solutions by combining the concept of regularization with the concept of curvelets. The more well-established total variation and wavelet based regularization methods which have similar applicability as the curvelet based method, are also carried out for comparison of the different methods.

1.2 Organization of the report

Linear inverse problems and ill-posedness of inverse problems is presented in Chapter 2, along with a few general theories and tools regarding certain operators that makes inverse problems ill-posed.

Before turning to the concept of regularization in Chapter 4, where also the two methods total variation and wavelet based regularization are presented, Chapter 3 comments on assumptions when problems are discretized.

The two specific edge preserving regularization methods total variation and wavelet based regularization are used to demonstrate reconstructions of a 1D deconvolution problem in Chapter 5 and a 2D computed tomography problem (CT) in Chapter 6. These two chapters also demonstrates how deconvolution and CT can be modelled both continuously and discretely.

In Chapter 7 the second generation curvelets are reviewed, leading to the examination of curvelet based regularization in Chapter 8.

A list of important mathematical symbols is found in Appendix A, and the fundamental and frequently used Fourier transform is explained in Appendix B. Appendix C shows the most important CT reconstructions carried out in Chapters 6 and 8 in full-size. Finally all relevant MATLAB source code is attached as Appendix D.

Chapter 2

Inverse problems and ill-posedness

This chapter presents a definition of linear inverse problems and a general form for mathematically modelling a physical linear problem to be inverted. Ill-posedness of inverse problems is reviewed and is the motivation for applying regularization, the subject in chapter 4. Finally, two classes of operators that makes inverse problems ill-posed are defined, and the singular value expansion is mentioned as a tool to examine ill-posedness in the class of compact operators.

2.1 Linear inverse problems and forward modelling

The task of reconstructing information about a physical system is called an inverse problem when the reconstruction is based on data that is modelled as relative to the sought information. A model describing how a physical cause is mapped into some effect as in Figure 2.1 leads to a *forward model* for the inverse problem.

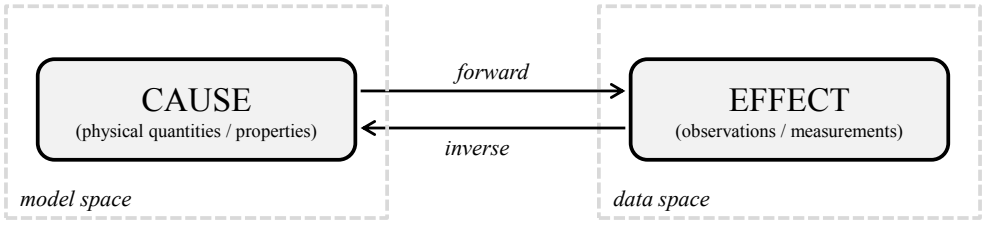


Figure 2.1: Forward and inverse modelling.

Consider two normed vector spaces V_m and V_d as the *model* space and *data* space in Figure 2.1 respectively. A bounded and linear operator $K\colon \mathcal{D}(K) \to V_d$, defined on a domain $\mathcal{D}(K) \subset V_m$, that maps a cause into the effect is called a *forward operator* [32]. In practice measured effects are corrupted by errors, and with a $\delta > 0$, a function ε

satisfying $\|\varepsilon\|_{V_d} \leq \delta$ models the errors. Then for a sought function $f \in \mathcal{D}(K)$ and an acquired function $g \in V_d$, a *forward problem* is modelled as

$$g = Kf + \varepsilon. \tag{2.1}$$

Given g and K, the recovery of some or all of f in (2.1) is an inverse problem. If the inverse K^{-1} exists a *naive* inversion f^* is defined as

$$f^* := K^{-1}g. \tag{2.2}$$

The naive inversion is only applicable on some inverse problems. In particular, the inversion (2.2) makes no sense if no or multiple functions f^* exist, or if K^{-1} is unbounded such that even a small δ makes f^* differ significantly from f.

Inverse problems are categorized as being either *well-posed* or *ill-posed*. This subject is elaborated in the following section, motivated by the fact that the deconvolution and tomography problems considered in later chapters (and many other inverse problems) are ill-posed.

2.2 Ill-posedness

A problem is said to be ill-posed if it is not well-posed. Formally, Hadamards conditions classifies a problem to be well-posed when all of the following three conditions are complied [32]:

- (H1) Existence (a solution exists).
- (H2) Uniqueness (the solution is unique).
- (H3) Stability (the solution is continuously dependent on the measurements).

Thus, if at least one of Hadamards conditions are not met the problem is ill-posed. In the general inverse problem (2.1) Hadarmards conditions are then complied when K meets the following three conditions:

- (H1) K is surjective ($\forall g \in V_d,\ \exists f \in V_m$ such that $g = Kf$).
- (H2) K is injective ($\forall f_1, f_2 \in V_m\colon\ Kf_1 = Kf_2 \Rightarrow f_1 = f_2$).
- (H3) K has a bounded/continuous inverse ($\forall g \in V_d\colon\ \|K^{-1}g\|_{V_m} \leq C\|g\|_{V_d}$ for a constant $C > 0$).

The topic of regularization deals with ill-posedness by replacing an ill-posed problem with a well-posed problem having a solution close to the true one. Known or assumed *a priori* information about a solution can be included in a model to compensate for some of the ill-posedness.

2.3 Compact operators and Hilbert-Schmidt operators

A class of operators that always makes the inverse problem (2.1) ill-posed is the compact linear operators in the following definition from [27]. Compact linear operators model a deconvolution and tomography problem in later chapters.

Definition 2.1 (Compact linear operator) *Let $T\colon V_1 \to V_2$ be a linear, bounded operator between two infinite dimensional and normed Banach spaces V_1 and V_2. For a bounded subset $U \subset V_1$ the closure $\overline{T(U)} \subset V_2$ of $T(U)$ is compact if every sequence in $\overline{T(U)}$ has a subsequence that converges in $\overline{T(U)}$. If $\forall U$, $\overline{T(U)}$ is compact the operator T is said to be a compact linear operator.*

The inverse of a compact linear operator, if it exists, is unbounded [34, Prop. 5.9]. A compact operator K in (2.1) will therefore always make the problem ill-posed by (H3).

The Hilbert-Schmidt integral operators in the next definition from [34] is a class of operators that are always compact. A periodic convolution operator used later to model the deconvolution problem falls in this category.

Definition 2.2 (Hilbert-Schmidt integral operator) *On a bounded set $\Omega \subset \mathbb{R}$ the integral operator $R\colon L^2(\Omega) \to L^2(\Omega)$, defined with $f \in L^2(\Omega)$ and $k \in L^2(\Omega \times \Omega)$ as*

$$(Rf)(x) := \int_\Omega f(t)\, k(x,t)\, \mathrm{d}t,$$

is a Hilbert-Schmidt integral operator *if*

$$\int_\Omega \int_\Omega |k(x,t)|^2 \, \mathrm{d}x \, \mathrm{d}t < \infty.$$

A Hilbert-Schmidt integral operator is a compact operator [34] and an R^{-1} in Definition 2.2, if it exists, is therefore unbounded.

2.4 Singular value expansion

The singular value expansion gives an understanding of the ill-posedess caused by a compact operator. Consider a compact linear operator between two Hilbert spaces $K\colon\ \mathcal{H}_1 \to \mathcal{H}_2$ and its Hilbert adjoint $K^*\colon\ \mathcal{H}_2 \to \mathcal{H}_1$. Let, for $J \subset \mathbb{N}$, $\{\lambda_j\}_{j\in J}$ be a sequence of non-negative increasing eigenvalues of the self-adjoint operator $K^*K\colon\ \mathcal{H}_2 \to \mathcal{H}_2$. Then the singular values $\{\mu_j\}_{j\in J}$ of K are defined in [25, 23], for each $j \in J$, as

$$\mu_j := \sqrt{\lambda_j}. \tag{2.3}$$

There further exist two sequences of orthogonal functions $\{x_j\}_{j\in J} \subset \mathcal{H}_1$ and $\{y_j\}_{j\in J} \subset \mathcal{H}_2$ such that for all $j \in J$

$$Tx_j = \mu_j y_j.$$

Assume that the compact operator K and a given function $g \in \mathcal{H}_2$ makes the equation $g = Kf$ solvable for an $f \in \mathcal{H}_1$. With the singular system $\{\mu_j, x_j, y_j\}_{j \in J}$ of K this assumption is complied, by the *Picard condition*, if and only if

$$g \in \mathcal{N}(K^*)^\perp \quad \text{and} \quad \sum_{j \in J} \left(\frac{|\langle g, y_j \rangle_{\mathcal{H}_2}|}{\mu_j} \right)^2 < \infty.$$

The solution to $g = Kf$ is then given by

$$f = \sum_{j \in J} \frac{\langle g, y_j \rangle_{\mathcal{H}_2}}{\mu_j} x_j. \tag{2.4}$$

By considering the coefficients $\langle g, y_j \rangle_{\mathcal{H}_2} / \mu_j$ in the orthogonal basis $\{x_j\}_{j \in J}$ having contents of increasing frequencies with increasing j, the expansion (2.4) shows how the decay in the magnitudes of the positive singular values μ_j, as $j \to \infty$, will amplify contents in g of higher frequencies. This amplification contributes to the ill-posedness of a given problem in the sense that corrupting g with errors having a flat spectrum will make f in (2.4) unstable.

Chapter 3

Discrete modelling

In practice solutions to inverse problems are computed digitally. The first two sections of this chapter elaborates on assumptions when discretizing a continuum function into a finite elements vector and properties of matrices in systems of linear equations which are used in subsequent chapters. In the last section measures for estimating qualities of inverted discrete functions are introduced.

3.1 Discrete functions

Continuum functions will for $d \in \{1, 2\}$ be considered on $\mathbb{R}^d$ domains in following chapters. Compactly supported functions can be assumed to have support in a set $[0, 1[^d \subset \mathbb{R}^d$.

Functions on $\Omega := [0, 1[\subset \mathbb{R}$ domains are discretized at $n \in \mathbb{N}$ finite equispaced points at $x_j := (j-1)/n$ for $j = 1, 2, ..., n$. By partitioning Ω into n subsets

$$I_j := \begin{cases} \left[\frac{j-1/2}{n}; \frac{j+1/2}{n}\right[& \text{for } j = 1, 2, ..., n-1, \\ \left[0; \frac{1/2}{n}\right[\cup \left[\frac{1-1/2}{n}; 1\right[& \text{for } j = n, \end{cases}$$

each x_j is centered in I_j with periodic boundary conditions such that the point at $x = 1$ corresponds to the point $x = 0$. The motivation for the periodic boundaries is explained later. A continuum function on Ω, discretized to $\bar{f} \in \mathbb{R}^n$, is then defined component-wise as

$$\bar{f}_j := f(x_j) \quad \text{for } j = 1, 2, ..., n.$$

Two dimensional functions on square Ω^2 domains are equivalently discretized at n^2 finite equispaced points

$$\bar{x}_{j,i} := \left(\frac{j-1}{n}, \frac{i-1}{n}\right) \quad \text{for } j, i = 1, 2, ..., n.$$

By partitioning Ω^2 into n^2 subsets $I_{j,i}$ with periodic boundary conditions in both directions, each point $\bar{x}_{j,i}$ is centered in a subset. Then points on $x \in I_{j,1}$ corresponds to

points on $x \in I_{j,0}$, points on $x \in I_{1,j}$ corresponds to points on $x \in I_{0,j}$, and the four corner points corresponds to each other diagonally pairwise. For practical reasons 2D functions are in some cases discretized into a single column vector. That is, $\bar{f} \in \mathbb{R}^{n^2}$ is defined with

$$\bar{f}_{ji} := f(\bar{x}_{j,i}) \quad \text{for } j, i = 1, 2, ..., n.$$

3.2 Discrete operators

Discrete approximations of explicit continuum operators are defined when necessary in later chapters.

With an explicitly defined matrix $A \in \mathbb{R}^{m \times n}$ and two vectors $\bar{g} \in \mathbb{R}^m$, $\bar{f} \in \mathbb{R}^n$, where $\bar{g}$ is given and $\bar{f}$ is unknown, the problem $\bar{g} = A\bar{f}$ represents a system of linear equations which can be written as

$$\begin{aligned} A_{1,1}\bar{f}_1 + A_{1,2}\bar{f}_2 + ... + A_{1,n}\bar{f}_n &= \bar{g}_1 \\ A_{2,1}\bar{f}_1 + A_{2,2}\bar{f}_2 + ... + A_{2,n}\bar{f}_n &= \bar{g}_2 \\ &\vdots \\ A_{m,1}\bar{f}_1 + A_{m,2}\bar{f}_2 + ... + A_{m,n}\bar{f}_n &= \bar{g}_m. \end{aligned} \tag{3.1}$$

When $n = m$ the system (3.1) has a unique solution if A has full rank. In that case A has a unique inverse $A^{-1} \in \mathbb{R}^{n \times n}$ and is called *invertible* [15]. A deconvolution problem is modelled with an invertible $n \times n$ matrix in chapter 5.

The system (3.1) is said to be *underdetermined* if $m < n$ and *overdetermined* if $m > n$. These are the situations where there are respectively less and more equations than unknowns in the system. An underdetermined system has either infinitely many solutions or none and is consequently called *consistent* or *inconsistent* respectively. In chapter 6, a tomography problem is modelled with an underdetermined $m \times n$ matrix.

3.3 Singular value decomposition

As for the continuum operators singular values of matrices provides a tool for investigating the condition of a system (3.1). Using the *singular value decomposition* an operator $A \in \mathbb{R}^{m \times n}$ can be factorized by two unitary matrices $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ and a diagonal matrix $\Sigma \in \mathbb{R}^{m \times n}$ such that

$$A = U\Sigma V^T, \tag{3.2}$$

where for $K := \min(m, n)$ the non-negative and non-increasing sequence $\{s_k\}_{k=1}^K$ of singular values of A constitutes the main diagonal of Σ. Using the largest and the smallest singular values s_1 and s_K the condition number $\kappa(A)$ of A is defined in [32], when $s_K > 0$, as

$$\kappa(A) := \frac{s_1}{s_K}. \tag{3.3}$$

Larger condition numbers indicate worse conditions [24], and if A is square and invertible the condition number is equivalent to $\kappa(A) = \|A\| \, \|A^{-1}\|$. This implies that large operator norms indicate ill conditions.

For $A \in \mathbb{R}^{m \times n}$ let $\bar{g} = \bar{g}^\star + \bar{\varepsilon} \in \mathbb{R}^m$ be a vector containing the true function $\bar{g}^\star \in \mathbb{R}^m$ that is perturbed by an error vector $\bar{\varepsilon} \in \mathbb{R}^m$. Then for $\Omega := \{1, 2, ..., n\}$ the solutions $\bar{f}^\star, \bar{f} \in \mathbb{R}^n$ to $\bar{g}^\star = A\bar{f}^\star$ and $\bar{g} = A\bar{f}$ comply with the bound

$$\|\bar{f}^\star - \bar{f}\|_{\ell^2(\Omega)} \leq \kappa(A) \|\bar{\varepsilon}\|_{\ell^2(\Omega)} \frac{\|\bar{f}^\star\|_{\ell^2(\Omega)}}{\|\bar{g}^\star\|_{\ell^2(\Omega)}}. \tag{3.4}$$

By [24] experience shows that the error $\|\bar{f}^\star - \bar{f}\|_{\ell^2(\Omega)}$ is always close to the upper bound in (3.4). This implies that a higher condition numbers $\kappa(A)$ will amplify the effect of $\|\bar{\varepsilon}\|_{\ell^2(\Omega)}$.

Ill conditions can also be explained by considering an invertible $n \times n$ matrix A in (3.2). The inverted A^{-1} consists of the inverse of the diagonal matrix Σ such that $\{1/s_k\}_{k=1}^n$ appears in A^{-1}. If the largest singular value s_1 is several orders of magnitude greater than the smallest singular value s_K, truncation to finite decimals digital numbers causes significant errors in a naive inversion.

3.4 Model performance

The qualities of regularized inversions carried out in the following chapters are estimated using the measures presented in this section. Testing inverse models by using a known function $\bar{f}^\star \in \ell^2(\Omega)$ on a set $\Omega := \{1, 2, ..., n\}$ satisfying $\|\bar{f}^\star\|_{\ell^2(\Omega)} > 0$ provides a way to compare a regularized solution $\bar{f}^* \in \ell^2(\Omega)$ to the true solution. A typical measure for the quality of an inversion is the ℓ^2-error defined as

$$e := \|\bar{f}^* - \bar{f}^\star\|_{\ell^2(\Omega)}, \tag{3.5}$$

and the relative error defined as

$$e_{\%} := \frac{e}{\|\bar{f}^\star\|_{\ell^2(\Omega)}}. \tag{3.6}$$

The two measures e and $e_{\%}$ are sought minimized for better qualities. For functions representing images on $\mathbb{R}^2$ domains the peak signal-to-noise ratio (PSNR) is commonly used as a quality measure. Using the mean squared error

$$\text{MSE} := \frac{1}{n} \sum_{k=1}^{n} |\bar{f}_k^* - f_k^\star|^2,$$

when $\bar{f}^\star \neq \bar{f}^*$ the PSNR is defined, with $2^B - 1 \geq 1$ being the maximum number of unique values of $\bar{f}^\star$, as

$$\text{PSNR} := 20 \log_{10} \left(\frac{2^B - 1}{\sqrt{\text{MSE}}} \right). \tag{3.7}$$

The PSNR is sought maximized for better qualities and involves the number of unique values of the true solution. In this measure a certain MSE therefore weights more for fewer unique values of the solution. That is, the MSE has less negative effect if the true solution has many unique values.

Better measures does not guarantee truer solutions (which often is also a subjective matter), but are useful tools when testing models.

Chapter 4

Regularization

Regularization replaces an ill-posed problem with a well-posed problem which is expected to have a solution close to the correct sought solution. This method provides a way to extract, ideally, as much information as possible from the true solution in an ill-posed inverse problem. The first section of this chapter elaborates on the concept of sparsity regularization followed, in the two last sections, by a review of the two methods total variation and wavelet based regularization which uses this concept.

4.1 Sparsity regularization

This project focuses on regularization methods that for a finite $J \in \mathbb{N}$ promotes sparsity of a sequence $\{x_j\}_{j=1}^J \subset \mathbb{R}$ associated with the sought function. The sequence $\{x_j\}_{j=1}^J$ is called *sparse* in this respect if a considerable number of elements $x_j = 0$ for $j = 1, 2, ...J$. The regularization methods applied in later chapters all lead to the problem of minimizing, for $\bar{f} \in \mathbb{R}^n$, the Tikhonov type functional

$$\Phi\left(\bar{f}\right) := \left\|A\bar{f} - \bar{g}\right\|^2_{\ell^2(\{1,2,...,m\})} + \alpha \sum_{j=1}^{J} |w_j\, x_j|, \tag{4.1}$$

where $\{w_j\}_{j=1}^J \subset \mathbb{R}$ is a sequence of weights and the matrix $A \in \mathbb{R}^{m\times n}$ models a discrete forward problem corresponding to (2.1) on the form

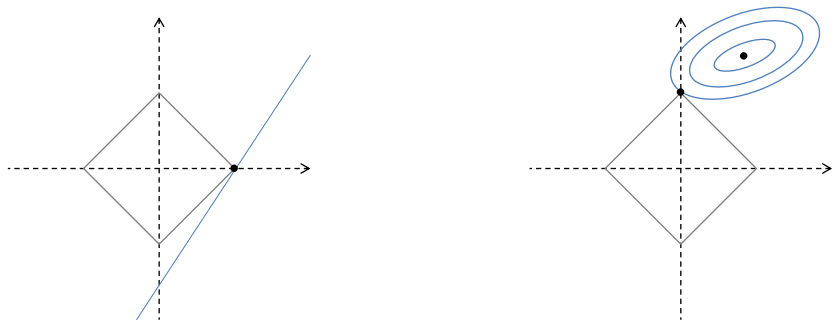
$$\bar{g} = A\bar{f} + \bar{\varepsilon}$$

with vectors $\bar{g}, \bar{\varepsilon} \in \mathbb{R}^m$. The two terms of (4.1) are called the *data fidelity term* and the *penalty term* respectively and the regularization parameter $\alpha > 0$ can be chosen freely, and is used to adjust the weight of the penalty term.

In (4.1) the penalty term corresponds to a weighted ℓ^1-norm of $\{x_j\}_{j=1}^J$ multiplied with α. This choice of penalty term promotes sparsity of the sequence it is applied on [36, 38]. In total variation regularization the ℓ^1-norm is applied on the gradient of the sought function. In wavelet and curvelet based regularization it is applied on weighted coefficients

from an expansion of $\bar{f}$. As explained in the following sections these sparsity promoting methods corresponds to assuming that a sought function contains discontinuities. This is advantageous when recovering edge-containing images.

A geometric argument for the ℓ^1-norm begin sparsity promoting follows for $m, n = 2$. If a system of linear equations $\bar{g} = A\bar{f}$ has at least one solution, the minimization of (4.1) corresponds to minimizing the penalty term subject to $\bar{g} = A\bar{f}$. The situation where this system of equations has an infinitude of solutions is shown in Figure 4.1(a). For a small $r > 0$ the ℓ^1-ball $B_r(0, 0)$ has to coincide with a line of solutions, as r increases, to find a vector $\bar{f} \in \mathbb{R}^2$ minimizing (4.1). In a corner point of the square, which is most likely to coincide with the feasible region first, either of the two coordinates in $\mathbb{R}^2$ is zero valued.



(a) A line of solutions to $\bar{g} = A\bar{f}$.

(b) No solutions to $\bar{g} = A\bar{f}$.

Figure 4.1: Feasible region (blue) and the ℓ^1 ball (grey).

When the system of linear equations has no solutions the *pseudo solution* minimizing the data fidelity term alone is shown as a dot in Figure 4.1(b) in center of the ellipses. The elliptic contours corresponds to curves where the norms of $A\bar{f} - \bar{g}$ are equal and follows from writing the data fidelity term as a quadratic matrix function $\bar{f}^T A^T A\bar{f}$ plus a constant.

The two regularization methods presented in the next sections are both edge-preserving methods [35, 32] where a priori information i.a. is that the sought functions contain discontinuities. By assuming that the functions belong to certain spaces the two methods lead to a sparsity promoting penalty term.

4.2 Total variation regularization

Total variation regularization is a well established edge-preserving method to recover images with edges in ill-posed inverse problems. The method is used in the following chapters to reconstruct edge containing images in ill-posed deconvolution and tomography problems. The *total variation* of a sought function is used as a penalty term to restrict the solution to functions with a sparse and bounded gradient.

4.2.1 Continuum modelling

The total variation of a function $f \in L^1(\Omega)$ is defined as the following in [20].

Definition 4.1 (Total variation)
Consider for $d \in \mathbb{N}$ and $i = 1, 2, ..., d$ an open set $\Omega \subseteq \mathbb{R}^d$ and $\bar{x} = (x_1, x_2, ..., x_n)^T \in \Omega$, a vector $\bar{\phi}$ of once continuously differentiable functions ϕ_i tending to zero outside Ω such that $\bar{\phi}(\bar{x}) = (\phi_1(\bar{x}), \phi_2(\bar{x}), ..., \phi_d(\bar{x}))^T \in C_0^1(\Omega; \mathbb{R}^d)$.
Let $U = \left\{\bar{\phi} \in C_0^1(\Omega; \mathbb{R}^d) \mid \forall \bar{x} \in \Omega \colon |\bar{\phi}(\bar{x})| \leq 1\right\}$ be a set of test-functions. With the divergence $\nabla \cdot \bar{\phi}$ *of $\bar{\phi}$, defined as*

$$\left[\nabla \cdot \bar{\phi}\right](\bar{x}) := \sum_{i=1}^{d} \frac{\partial \phi_i(\bar{x})}{\partial x_i},$$

the total variation $\int_\Omega |[Df](\bar{x})| \, \mathrm{d}\bar{x}$ of a function $f \in L^1(\Omega)$ is then weakly defined as

$$\int_\Omega |[Df](\bar{x})| \, \mathrm{d}\bar{x} := \sup_{\bar{\phi} \in U} \left\{ \int_\Omega f(\bar{x}) \, \left[\nabla \cdot \bar{\phi}\right](\bar{x}) \, \mathrm{d}\bar{x} \right\}. \tag{4.2}$$

A function $f \in L^1(\Omega)$ is said to have *bounded variation* if its total variation (4.2) satisfies

$$\int_\Omega |[Df](\bar{x})| \, \mathrm{d}\bar{x} < \infty,$$

and the space of all such functions will be denoted $BV(\Omega)$ which is Banach with the norm defined as

$$\|f\|_{BV(\Omega)} := \|f\|_{L^1(\Omega)} + \int_\Omega |[Df](\bar{x})| \, \mathrm{d}\bar{x}.$$

The space $BV(\Omega)$ includes i.a. well defined piecewise smooth functions with their derivatives which is advantageous when working with edge containing images. The Sobolev space $W^{1,1}(\Omega)$, defined in [18] as the space of functions in $L^1(\Omega)$ additionally having weak first order derivatives belonging to $L^1(\Omega)$, is a proper subspace of $BV(\Omega)$ and is well suited for discretely approximating assumed piecewise constant functions, and provides a simpler expression of the total variation of a function.

Sobolev spaces uses the weak gradient of functions. With $\bar{x} := (x_1, x_2, ..., x_d)^T \in \mathbb{R}^d$, if for each $i = 1, 2, ..., d$ there exists functions $v_i \in L^1(\Omega)$ satisfying for all test functions $\phi \in C_c^\infty$ that

$$\int_\Omega f(\bar{x}) \, \frac{\partial}{\partial x_i} \phi(\bar{x}) \, \mathrm{d}\bar{x} = -\int_\Omega v_i(\bar{x}) \, \phi(\bar{x}) \, \mathrm{d}\bar{x}$$

the first order gradient Df of f is defined in the weak sense as

$$Df := (v_1, v_2, ..., v_d)^T. \tag{4.3}$$

Using (4.3) the total variation of a function $f \in W^{1,1}(\Omega)$ is defined in [22] as

$$TV(f) := \|Df\|_{L^1(\Omega)} \tag{4.4}$$

and a regularization method can now be defined by using (4.4) as a penalty term. With a given bounded linear operator $K\colon L^2(\Omega) \to L^2(\Omega)$ with trivial null-space, a given function $g \in L^2(\Omega)$ and closed and convex subset $V(\Omega)$ of $L^2(\Omega)$, a regularized solution to the general inverse problem (2.1) is then defined, for $\alpha > 0$, as

$$f^* := \underset{f \in V(\Omega)}{\arg\min} \left\{ \|Kf - g\|^2_{L^2(\Omega)} + \alpha\, TV(f) \right\}. \tag{4.5}$$

By [1, 8] this problem has a unique minimizer when the intersection $V(\Omega) \cap W^{1,1}(\Omega)$ is nonempty. It follows, under the given assumptions, that (4.5) is a well-posed problem.

4.2.2 Discrete modelling

When the discrete approximation of a sough function is assumed to have constant value on each element of a discretization grid the total variation of a discrete 2D function in $\mathbb{R}^{r\times s}$, written with $n := rs$ as $\bar{f} \in \mathbb{R}^n$ by concatenating each column s times, can be approximated by defining component-wise, for $k = 1, 2, ..., n$, a discrete approximate gradient $\tilde{D}$ as

$$\left(\tilde{D}\bar{f}\right)_k := \sqrt{(f[k+m] - f[k])^2 + (f[k+1] - f[k])^2} \tag{4.6}$$

and assuming n-periodicy of $\bar{f}$. Using (4.6) as a penalizer then leads to a discrete regularized solution to $\bar{g} = A\bar{f} + \bar{\varepsilon}$ for a given $A \in \mathbb{R}^{m\times n}$ and a given $\bar{g} \in \mathbb{R}^m$ defined as

$$\bar{f}^* := \underset{\bar{f} \in \ell^2(\{1,2,...,n\})}{\arg\min} \left\{ \|A\bar{f} - \bar{g}\|^2_{(\{1,2,...,m\})} + \alpha \|\tilde{D}\bar{f}\|_{\ell^1(\{1,2,...,n\})} \right\}. \tag{4.7}$$

The sparsity promoting ℓ^1-norm of the gradient $\tilde{D}\bar{f}$ of a sought function $\bar{f}$ implies that some or many elements of the derivatives of $\bar{f}$ are zero indicating that $\bar{f}$ is piecewise constant. Total variation regularization will be used in the following chapters to reconstruct edge containing images in the deconvolution and tomography problem.

4.3 Wavelet based regularization

Wavelet based regularization is used in the following chapters to reconstruct edge containing images in ill-posed deconvolution and tomography problems. The concept is very similar to the one of the main subject, curvelet based regularization, in the sense that in both methods the ℓ^1-norm is used to promote sparsity of expansion coefficients of the sought function. Wavelet based regularization shows edge-preserving properties similar to total variation regularization, especially for the choice of a Haar orthonormal basis.

4.3.1 Continuum modelling

With the Haar wavelet defined below, the magnitudes of the coefficients in a functions expansion is related to jump discontinuities of the function. This feature, as will be demonstrated with the deconvolution problem makes Haar wavelet based regularization have very similar properties to total variation regularization for 1D problems. For 2D problems the Haar wavelet only capture edges in three directions which is much less than curvelets.

The following defines a wavelet on $\mathbb{R}$ as in [10].

Definition 4.2 (Wavelet) *A sequence $\{\psi_j\}_{j\in\mathbb{N}} \subset L^2(\mathbb{R})$ is a basis for $L^2(\mathbb{R})$ if for all $f \in L^2(\mathbb{R})$ there exist unique scalars $\{c_j\}_{j\in\mathbb{N}}$ such that $\|f - \sum_{j\in\mathbb{N}} c_j\psi_j\|_{L^2(\mathbb{R})} \leq \epsilon$ for all $\epsilon > 0$. The sequence $\{\psi_j\}_{j\in\mathbb{N}}$ is an orthonormal system if $\langle\psi_j, \psi_k\rangle_{L^2(\mathbb{R})} = 0$ whenever $j \neq k$. A function $\psi \in L^2(\mathbb{R})$ is a wavelet if the sequence of functions*

$$\left\{2^{j/2}\psi\left(2^j x - k\right)\right\}_{j,k\in\mathbb{Z}}$$

form an orthonormal basis for $L^2(\mathbb{R})$.

Thus, any function $f \in L^2(\mathbb{R})$ can be written using a ψ complying with Definition 4.2 as the infinite term expansion

$$f = \sum_{j,k\in\mathbb{Z}} \langle f, \psi_{j,k}\rangle_{L^2(\mathbb{R})}\, \psi_{j,k}. \tag{4.8}$$

Multi resolution analysis is a technique that can be used to generate wavelets using, as a starting point, a scaling function $\phi \in L^2(\mathbb{R})$ that captures low frequencies and meets certain criteria. With this method and the translation operator $T_a\colon\ L^2(\Omega) \to L^2(\mathbb{R})$ defined on $x \in \mathbb{R}$ as $(T_a f)(x) := f(x - a)$ for an $a \in \mathbb{R}$, an expansion corresponding to (4.8) becomes

$$f = \sum_{k\in\mathbb{Z}} \langle f, T_k\phi\rangle_{L^2(\mathbb{R})} T_k\phi + \sum_{j\in\mathbb{N}}\sum_{k\in\mathbb{Z}} \langle f, \psi_{j,k}\rangle_{L^2(\mathbb{R})}\psi_{j,k}. \tag{4.9}$$

The translation parameter k shifts the wavelet and scaling function throughout $\mathbb{R}$ and the scaling parameter j dilates the wavelets making their graphs taller and narrower for increasing j. This causes the coefficients to have contents from f at higher frequencies for increasing j.

For a compactly supported f, or compactly supported ψ and ϕ, the k-summations in (4.9) reduces to finite terms. In practice an approximated or 'low-pass'-filtrated version of f can then be reconstructed using partial sums in j. The magnitudes of wavelet coefficients in an expansion decay with increasing j at rates dependent on the number of *vanishing moments* of the used wavelet [10].

Definition 4.3 (Vanishing moments) *On* $\mathbb{R}$ *a function* ψ *is said to have* $M \in \mathbb{N}$ vanishing moments *if for* $j = 1, ..., M$

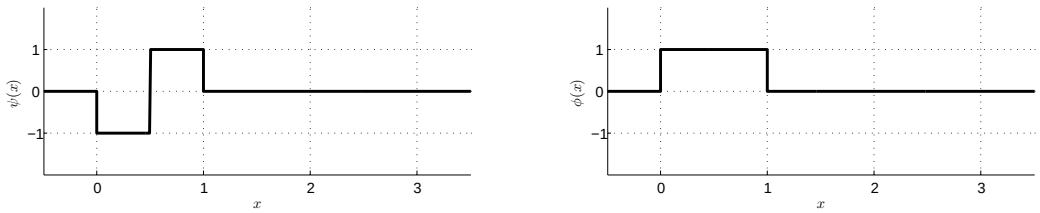
$$\int_{\mathbb{R}} x^{j-1}\, \psi(x)\, \mathrm{d}x = 0.$$

The number of vanishing moments of a wavelet gives information about the expected sparsity of the expansion coefficients and how much information is discarded when the expansion (4.9) is truncated to finite termed sums.

The compactly supported and simple discontinuous Haar wavelet and scaling function, with one vanishing moment, shown in Figure 4.2, are defined as

$$\psi(x) := \begin{cases} 1 & \text{if } 0 \leq x < \frac{1}{2}, \\ -1 & \text{if } \frac{1}{2} \leq x < 1, \\ 0 & \text{otherwise,} \end{cases} \qquad \phi(x) := \chi_{[0,1[}(x). \tag{4.10}$$

When the Haar wavelet and scaling function (4.10) are used in the expansion (4.9) coefficients $\langle f, T_k \phi \rangle_{L^2(\mathbb{R})}$ and $\langle f, \psi_{j,k} \rangle_{L^2(\mathbb{R})}$ of large values indicates a jump in f at a translation k. This feature makes the Haar wavelet well suited to detect discontinuities in a function.

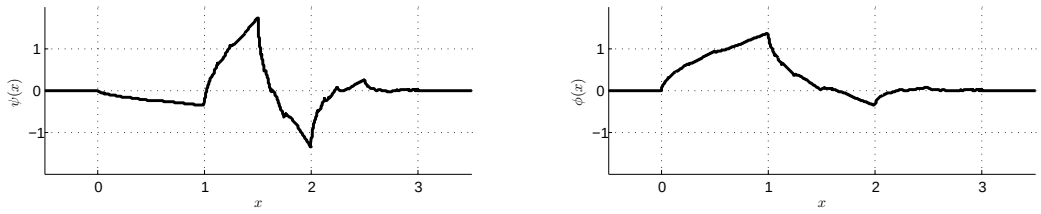


(a) Wavelet. (b) Scaling function.

Figure 4.2: Haar wavelet and scaling function.

For comparison the compactly supported and continuously differentiable Daubechies 2 (DB2) wavelet, which has two vanishing moments, is considered. The family of Daubechies wavelets is known to perform well in i.a. image compression. The DB2 wavelet and scaling function, which cannot be written explicitly in mathematical terms, are shown in Figure 4.3. With two vanishing moments this wavelet is expected to give rise to sparser expansions than the Haar wavelet. This is advantageous when truncating the expansion to finite sums, but on the other hand the DB2 wavelet does not perform as well as the Haar in edge detecting.

With regard to regularization the Besov smoothness space is an applicable space for wavelet decomposed functions [32, 26]. For $d \in \mathbb{N}$, on an open set $\Omega \subset \mathbb{R}^d$, the definition of Besov spaces of functions in $L^p(\Omega)$ follows [14]. Define for a step-length $h > 0$ and $r \in \mathbb{N}$ the rth order difference operator as



(a) Wavelet. (b) Scaling function.

Figure 4.3: Daubechies 2 wavelet and scaling function.

$$(\Delta_h^r f)(x) := \begin{cases} (f(x+h) - f(x))^r & \text{for } x, x+h, x+2h, ..., x+rh \in \Omega, \\ 0 & \text{otherwise.} \end{cases} \tag{4.11}$$

The rth order modulus of smoothness ω_r^p of a function $f \in L^p(\Omega)$ is defined, with a bound t for h, as

$$\omega_r^p(f,t) := \sup_{|h| \leq t} \|\Delta_h^r f\|_{L^p(\Omega)}.$$

A Besov space is Banach when $1 \geq p, q \leq \infty$ and consists of functions with common smoothness $r > s$. A function $f \in B_{p,q}^s(\Omega)$ if $f \in L^p(\Omega)$ and

$$\left(\int_{\mathbb{R}_0^+} t^{-s}\, \omega_r^p(f,t)^q \, \frac{\mathrm{d}t}{t} \right)^{1/q} < \infty. \tag{4.12}$$

Wavelets provide a base for $B_{p,q}^s(\Omega)$ [11]. With the choices $p = q = s = 1$ the norm of an $f \in B_{1,1}^1(\Omega)$, provided that the wavelet and scaling function are once continuously differentiable, is defined in [31] in terms of its expansion coefficients as

$$\|f\|_{B_{1,1}^1(\Omega)} := \sum_{k \in \mathbb{Z}} |\langle f, T_k \phi \rangle_{L^2(\Omega)}| + \sum_{j \in \mathbb{N}} \sum_{k \in \mathbb{Z}} 2^{j/2} |\langle f, \psi_{j,k} \rangle_{L^2(\Omega)}|. \tag{4.13}$$

The Haar wavelet and scaling functions (4.10) are not once continuously differentiable and does therefore not make (4.13) well defined by Meyers proofs in [31]. In spite of a theoretical proof of when or when not the Haar wavelet is applicable to (4.13) the Haar wavelet is used in i.a. [32, 26] as well as in this project.

Note that the norm (4.13) corresponds to a weighted ℓ^1-norm of the wavelet coefficients. For a fixed s different values of $r > s$ give equivalent Besov norms [14].

Using (4.13) as a regularization penalty term then restricts a sought function of an inverse problem to be in $B_{1,1}^1(\Omega)$. That is, the function is in $L^1(\Omega)$ and have modulus of smoothness $r > s = 1$. With a given bounded linear operator $K \colon L^2(\Omega) \to L^2(\Omega)$ with trivial null-space, a given function $g \in L^2(\Omega)$, and closed and convex subset $V(\Omega)$ of

$L^2(\Omega)$, a regularized solution to the general inverse problem (2.1) is defined, for $\alpha > 0$, as

$$f^* := \underset{f \in V(\Omega)}{\arg\min} \left\{ \frac{1}{2\sigma^2} \|Kf - g\|^2_{L^2(\Omega)} + \alpha \|f\|_{B^1_{1,1}(\Omega)} \right\} \tag{4.14}$$

with the variance $\sigma^2 > 0$ of the errors ε in (2.1). By [11] this problem has a unique minimizer when $V(\Omega) \cap B^1_{1,1}(\Omega)$ is nonempty. The regularized problem (4.14) is therefore well-posed under the given assumptions. The variance factor on the data fidelity term in (4.14) follows from a Bayesian statistical deduction in [26] where the posterior probability of f under condition of g is sought maximized using likelihood with a priori information on σ.

The problem (4.14) can be expressed in terms of a wavelet decomposed function such that an equivalent problem solves for the sought functions wavelet coefficients. Let $\{c_\phi\}$ denote the finite number of scaling coefficients from a wavelet decomposed compactly supported function and let $c_{j,k}$ be a wavelet coefficient at translate k and scale j. Using the infinite sequence

$$c := \{c_i\}_{i \in \mathbb{N}} := \{c_\phi\} \cup \left\{ \{c_{j,k}\}_{k=0}^{2^j - 1} \right\}_{j \in \mathbb{N}} \in \ell^2(\mathbb{N}) \tag{4.15}$$

of wavelet coefficients with the corresponding basis $\{u_i\}_{i \in \mathbb{N}}$ of functions $u_i \in L^2(\mathbb{R})$ the basis reconstruction operator $R \colon \ell^2(\mathbb{N}) \to L^2(\mathbb{R})$ is then defined as

$$Rc := \sum_{i \in \mathbb{N}} c_i u_i \tag{4.16}$$

such that Rc by Definition 4.2 converges to f in the L^2-norm. The problem (4.14) can with R, a sequence $\{w_i\}_{i \in \mathbb{N}}$ containing the $2^{j/2}$ weights and by (4.13) be expressed in terms of the wavelet coefficients of the sought function as

$$c^* := \underset{c \in \ell^2(\mathbb{N})}{\arg\min} \left\{ \frac{1}{2\sigma^2} \|KRc - g\|^2_{L^2(\Omega)} + \alpha \sum_{i \in \mathbb{N}} |w_i c_i| \right\}. \tag{4.17}$$

When R in (4.16), as in practice, truncates the coefficients (4.15) after $J \in \mathbb{N}$ finite scales $j = 0, 1, ..., J-1$, the functions $\{u_i\}_{i=1}^{2^J}$ constitutes a basis for a finite-dimensional subspace of the Banach space $L^2(\mathbb{R})$ which Rc then belongs to.

4.3.2 Discrete modelling

Turning to the discrete problem a wavelet decomposition operator and reconstruction operator is introduced. In practice the discrete wavelet transform is performed by discretely convolving the transformed function with a filter uniquely determined by the choice wavelet. Let $f \in L^2(\mathbb{R})$ have compact support in $[0, 1[\subset \mathbb{R}$ and $\bar{f} \in \ell^2(\Omega)$ on $\Omega := \{1, 2, ..., n\} \subset \mathbb{N}$ denote the discretization of f. Note that the expansion (4.9) of f in this case only gives rise to one coefficient $\langle f, T_0 \phi \rangle$ from the scaling function. The

discrete wavelet transform of $\bar{f}$ at finite scales $j = 0, 1, ..., J-1 \subset \mathbb{N}_0$ then gives rise to the follow set of scalar coefficients

$$\{c_i\}_{i=1}^{2^J} := \{c_\phi\} \cup \left\{ \{c_{j,k}\}_{k=0}^{2^j-1} \right\}_{j=0}^{J-1},$$

where c_ϕ is the single coefficient generated by the scaling function and $c_{j,k}$ are the coefficients generated by the wavelets. The decomposition operator $D\colon\ \mathbb{R}^n \to \mathbb{R}^{2^J}$ that maps $\bar{f}$ into the sequence of coefficients $\{c_i\}_{i=1}^{2^J}$ at the J first scales in a wavelet decomposition is defined implicitly such that

$$D\bar{f} = \{c_i\}_{i=1}^{2^J}. \tag{4.18}$$

Similarly, the reconstruction operator $R\colon\ \mathbb{R}^{2^J} \to \mathbb{R}^n$ maps coefficients $\{c_i\}_{i=1}^{2^J}$ into a finite scales j reconstruction $\bar{f}_J$ of $\bar{f}$ and is defined such that

$$R\{c_i\}_{i=1}^{2^J} = \bar{f}_J.$$

By defining the diagonal $2^J \times 2^J$ weight matrix W, such that the sequence of weights $\{\{2^{j/2}\}_{k=0}^{2^j-1}\}_{j=0}^{J-1}$ appears in the diagonal, as

$$W := \begin{pmatrix} 2^{0/2} & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 2^{1/2} & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 2^{1/2} & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 2^{2/2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 2^{(J-1)/2} & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 2^{(J-1)/2} \end{pmatrix}, \tag{4.19}$$

the Besov norm (4.13) of $\bar{f} \in \ell^2(\Omega)$, for $\Omega := \{1, 2, ..., n\}$, can be approximated, using the ℓ^1-norm, by

$$\|WD\bar{f}\|_{\ell^1(\{1,2,...,2^J\})}.$$

With a given matrix $A\colon\ \ell^2(\{1, 2, ..., n\}) \to \ell^2(\{1, 2, ..., m\})$ and a given function $\bar{g} \in \ell^2(\{1, 2, ..., m\})$ the discrete wavelet based regularized solution $\bar{f}^*$ is then defined as

$$\bar{f}^* := \underset{\bar{f} \in \ell^2(\{1,2,...,n\})}{\arg\min} \left\{ \frac{1}{2\sigma^2} \|A\bar{f} - \bar{g}\|^2_{\ell^2(\{1,2,...,m\})} + \alpha \|WD\bar{f}\|_{\ell^1(\{1,2,...,2^J\})} \right\}, \tag{4.20}$$

The weights W in (4.20) penalizes discontinuities (large coefficients) at higher scales j more than at lower scales. The errors in $\bar{g}$ are assumed to be approximately white Gaussian noise and for a sought deterministic function with spectres tending to zero in the limits, the relative noise content will be larger at higher frequencies. This favours a

solution $\bar{f}^*$ in (4.20) with less high-frequency content. This matter also indicates that a truncation of the wavelet expansion can be justified.

On $\mathbb{R}^2$ domains a discrete wavelet transform is obtained by first applying the 1D transform on each column of a matrix representing the discrete 2D function. The 1D transform is then applied on each row of the resultant matrix of the first 1D transform [39]. At each scale j this gives rise to three sets of coefficients in addition to the approximation coefficients from the scaling function. With the edge detecting feature of the Haar wavelet, the three set of coefficients corresponds to edges in respectively the horizontal, vertical and diagonal directions.

Chapter 5

Deconvolution

This chapter introduces the *convolution operator* which is used to model a forward operator for an ill-posed inverse problem of deconvolution. A composed specific inverse problem of de-blurring a 1D digital image is considered to explore the edge-preserving features of total variation- and wavelet based regularization and demonstrate their utilities.

5.1 Continuum modelling

With a given function $k \in L^1(\mathbb{R})$ the linear convolution operator $P_k \colon L^2(\mathbb{R}) \to L^2(\mathbb{R})$ is defined for $f \in L^2(\mathbb{R})$ in [16, 10] as

$$(P_k f)(x) := \int_{\mathbb{R}} f(x-t)\, k(t)\, \mathrm{d}t \tag{5.1}$$

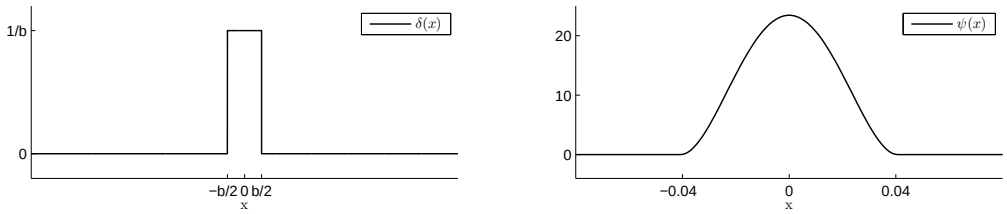
for almost all $x \in \mathbb{R}$. From measure theory technicalities requires L^2-functions to for instance be piecewise continuous for the convolution to be pointwise well-defined. The blurred version of a 1D image f can be modelled as $P_k f$ using a *point spread function* (PSF) as k. To illustrate the concept consider the δ-distribution on $x \in \mathbb{R}$ defined by Dirac to be

$$\int_{\mathbb{R}} \delta(x)\, \mathrm{d}x := 1 \quad \text{and} \quad \delta(x) := 0 \text{ for } x \neq 0, \tag{5.2}$$

and define on $x \in \mathbb{R}$, with a spreading constant $0 < a < 1/2$, a point spread function $\psi_a \in L^2(\mathbb{R})$ from [32], constructed by a fourth degree polynomial, as

$$\psi_a(x) := \begin{cases} \frac{(x+a)^2\,(x-a)^2}{\int_{-a}^{a}(t+a)^2\,(t-a)^2\,\mathrm{d}t} & \text{for } -a \leq x \leq a, \\ 0 & \text{for } |x| > a, \end{cases} \tag{5.3}$$

and satisfying $\int_{\mathbb{R}} \psi_a(x)\, \mathrm{d}x = 1$. The δ-function can in the limit of a $b \to 0, b > 0$ be considered as a tall and narrow rectangular shape with height $1/b$ and width b about $x = 0$ as in Figure 5.1a.



(a) Visualization of δ as $b \to 0$.

(b) PSF, $a := 0.04$.

Figure 5.1: Distributions.

The *sampling property* of δ implies that $\int_{\mathbb{R}} f(x)\,\delta(x)\,\mathrm{d}x = f(0)$ and $(P_\delta f)(x) = f(x)$ for a continuous f [28]. That is, the convolution of f with δ replicates f. The PSF in a sense also reproduces a function which it is convolved with, but with energies of each point spread over neighbouring points. This feature causes a blurring effect where edges are smoothened as in Figure 5.2.

(a) Original (f).

(b) Blurred ($P_{\psi_a} f$) with $a := 0.04$.

Figure 5.2: Example - blurring of a 1D function.

In a convolution $P_{\psi_a} f$ the PSF distributes the content of a compactly supported f such that $P_{\psi_a} f$ is supported on a slightly larger set on $\mathbb{R}$ than f is, depending on the choice of a. To retain support of sought functions in the deconvolution model, the functions are thought of as being 1-periodic and a cyclic convolution is considered. This way energy distributed to domains $x < 0$ or $x > 1$ by a convolution gets its impact on domains $x < 1$ or $x \geq 0$ respectively.

Define for $x \in [0,1[\subset \mathbb{R}$ and $t \in [0,1[\subset \mathbb{R}$ one period of a t-translatable version of a 1-periodic version of $f \in L^2([0,1[)$ as

$$f^\circ(x,t) := \begin{cases} f(x+1-t) & \text{for } x-t \in [-1,0[, \\ f(x-t) & \text{for } x-t \in [0,1[\end{cases} \in L^2([0,1[\times[0,1[), \tag{5.4}$$

satisfying that

$$\int_0^1 \int_0^1 f^\circ(x,t)\,\mathrm{d}x\,\mathrm{d}t = \int_0^1 f(x)\,\mathrm{d}x, \tag{5.5}$$

and for $t \in [0,1[\subset \mathbb{R}$ define one period of a 1-periodic version of the PSF as

$$\psi_a^\circ(t) := \begin{cases} \psi_a(x) & \text{for } t \in [0, \frac{1}{2}[, \\ \psi_a(x-1) & \text{for } t \in [\frac{1}{2}, 1[\end{cases} \in L^1([0,1[). \tag{5.6}$$

Then the operator $K\colon L^2([0,1[) \to L^2([0,1[)$, defined as

$$(Kf)(x) := \int_0^1 f^\circ(x,t)\, \psi_a^\circ(t)\, \mathrm{d}t \tag{5.7}$$

for almost all $x \in [0,1[\subset \mathbb{R}$, corresponds to a cyclic convolution of 1-periodic versions of f and ψ, and will be used to model a continuum forward operator for the de-blurring problem at hand with the general model (2.1).

The commutative property of the convolution of two functions in $L^1([0,1[)$ allows the operator K to be considered applied on the PSF. With $f \in L^2([0,1[)$ the estimate

$$\int_0^1 \int_0^1 |\psi^\circ(x-t)|^2 \, \mathrm{d}x \, \mathrm{d}t < \infty$$

of the PSF then implies that K in (5.7) is a Hilbert-Schmidt operator as in Definition 2.2. The forward operator K is thus expected to make an inverse problem ill-posed by (H3).

The unbounded inverse of a convolution operator appears when considering the related Fourier transform. With $f, k \in L^1(\mathbb{R})$ and P_k in (5.1) the Fourier transform $\widehat{(\cdot)}$ of $P_k f$ is given by

$$\widehat{(P_k f)}(\gamma) = \hat{f}(\gamma)\, \hat{k}(\gamma).$$

The inverse Fourier transform of $\hat{f}$, provided that it is well defined, is then naively expressed as

$$f(x) = \int_{\mathbb{R}} \frac{\widehat{(P_k f)}(\gamma)}{\hat{k}(\gamma)}\, e^{2\pi i x \gamma} \, \mathrm{d}\gamma.$$

When $\hat{k}(\gamma) \to 0$ for $|\gamma| \to \infty$, which is the case for the PSF, this inversion is unbounded.

5.2 Discrete modelling

The circular integral (5.7) is approximated using numerical quadrature. With the PSF and sought function discretized as explained in section (3.1), $\bar{f} \in \mathbb{R}^{2n}$ denotes a vector containing two periods of the sought function in (5.4) and $\bar{\psi} \in \mathbb{R}^n$ denotes the PSF in (5.6). The discrete convolution is then defined element-wise as

$$(\bar{f} * \bar{\psi})_k := \frac{1}{n} \sum_{l=1}^{n} f[k+n-l]\, \psi[l] \quad \text{for } k = 1, 2, ..., n \tag{5.8}$$

Defining a circulant $n \times n$ matrix $A\colon \mathbb{R}^n \to \mathbb{R}^n$ as

$$A := \frac{1}{n}\begin{pmatrix} \psi[1] & \psi[n] & \psi[n-1] & \cdots & \psi[3] & \psi[2] \\ \psi[2] & \psi[1] & \psi[n] & \cdots & \psi[4] & \psi[3] \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \psi[n-1] & \psi[n-2] & \psi[n-3] & \cdots & \psi[1] & \psi[n] \\ \psi[n] & \psi[n-1] & \psi[n-2] & \cdots & \psi[2] & \psi[1] \end{pmatrix}, \tag{5.9}$$

the discrete convolution (5.8) can be written as $A\bar{f}$ such that the discrete forward model corresponding to (2.1) for the deconvolution problem becomes

$$\bar{g} = A\bar{f} + \bar{\varepsilon}. \tag{5.10}$$

From [21] a circulant $n \times n$ matrix as (5.9) has the eigenvalues

$$\lambda_k = \sum_{j=1}^{n} \psi[j] e^{-2\pi kj/n} \quad \text{for} \quad k = 1, 2, ..., n.$$

Since, unless $\psi[k] = 0$ for all $k = 1, 2, ..., n$, $\sum_{j=1}^{n} \psi[j] > 0$, the eigenvalues of A in (5.9) must be $\lambda_k \neq 0$ for all $k = 1, 2, ..., n$. This implies that A is non-singular and invertible.

5.3 Demonstrations

Consider a testing function $f^\star$ defined on $x \in [0, 1[\subset \mathbb{R}$ for the purpose as

$$\begin{aligned} f^\star(x) :=& 1\chi_{]0.1;0.2]}(x) + 0.3\chi_{]0.2;0.3]}(x) + 0.8\chi_{]0.35;0.4]}(x) + \\ & 9(x - 0.5)\chi_{]0.5;0.6]}(x) + 0.6\sin(5\pi(x - 0.7))\chi_{]0.7;0.9]}(x), \end{aligned} \tag{5.11}$$

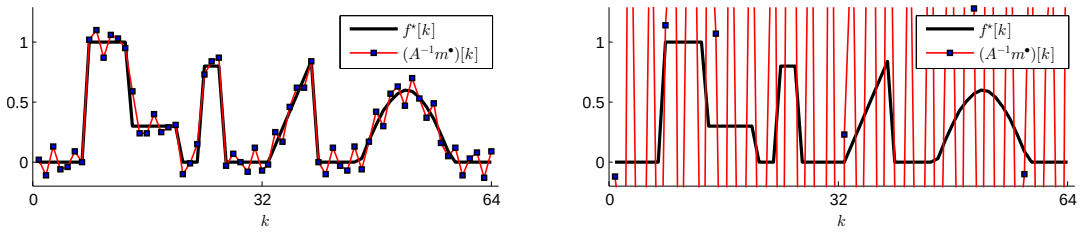
and the PSF with spreading constant $a := 0.04$. Let $\bar{f}^\star, \bar{\psi} \in \mathbb{R}^{64}$ be discrete versions of $f^\star$ and ψ respectively. The continuous functions are thought as being periodic and the discrete vectors constitute one single period. Figure 5.2 shows $\bar{f}^\star$ and the discrete convolution between $\bar{f}^\star$ and $\bar{\psi}$. With $n = 64$ the forward operator $A \in \mathbb{R}^{n\times n}$ from (5.9) is used to model the 1D de-blurring problem. As expected A was verified to have full rank, and the inverse A^{-1} is expected to provide unstable solutions by the continuum theory of a compact convolution operator. The condition number (3.3) of A was computed to be $\kappa(A) \approx 1/0.0016 \approx 628.32$.

Simulated measurements $\bar{g}^\bullet$ are constructed consistently in all demonstrations below. The vector $\bar{g}^\bullet \in \mathbb{R}^n$ is defined as one period of the convolution $\bar{f} * \bar{\psi}$ computed with MATLABs `integral()` function that approximates an analytic integral using adapted quadrature and interpolated from $n = 1000$ to $n = 64$ discrete points. Finally, Guassian white noise with a specified variance is added to $\bar{g}^\bullet$. The simulated data is intentionally not computed as $\bar{g}^\bullet = A\bar{f}$. This is done to avoid so-called *inverse crime* where testing of

inverse models is done under unrealistically good circumstances. For the same reason, the data is interpolated, and the PSF spreading constant a used in A is not exactly equal to the true one ($|a - \tilde{a}| = 0.001$).

5.3.1 Naive reconstruction

The vector $\bar{f}^* \in \mathbb{R}^n$ is reconstructed naively using A^{-1} from (5.9) such that $\bar{f}^* := A^{-1}\bar{g}^\bullet$. This is done both without and with errors added to $\bar{g}^\bullet$. In the latter case the errors are white Gaussian noise with a standard deviation corresponding to $2\% \cdot \max \bar{f}$.



(a) No noise.

(b) 2% noise.

Figure 5.3: Deconvolution - naive reconstruction.

As a result of the inverse crime initiatives, even without errors the naive reconstruction in Figure 5.3(a) still deviates from the true function as expected. With 2% noise, as seen in Figure 5.3(b), the reconstruction is useless as expected.

5.3.2 Total variation regularized reconstruction

Using the finite-difference $n \times n$ matrix L corresponding to (4.6) for the 1D case, defined as

$$L := \frac{1}{\Delta x} \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \\ 1 & 0 & 0 & \cdots & 0 & -1 \end{pmatrix}, \tag{5.12}$$

the total variation regularized solution (4.7) is computed using quadratic programming. Consider the circular convolution operator $A \in \mathbb{R}^{n \times n}$ from (5.9), the identity matrix $I \in \mathbb{R}^{n \times n}$, the vectors $\bar{0}$, $\bar{1} \in \mathbb{R}^n$ having all elements equal to zero or one respectively, and a matrix $\bar{\bar{0}} \in \mathbb{R}^{n \times n}$ having all elements equal to zero. Set $\bar{v}_+ - \bar{v}_- := L\bar{f}$ such that $\bar{v}_+$, $\bar{v}_- \in \mathbb{R}_0^{+n}$ and define

$$\bar{x} := \begin{pmatrix} \bar{f} \\ \bar{v}_+ \\ \bar{v}_- \end{pmatrix} \in \mathbb{R}^{3n}, \quad Q := \begin{pmatrix} 2A^TA & \bar{\bar{0}} & \bar{\bar{0}} \\ \bar{\bar{0}} & \bar{\bar{0}} & \bar{\bar{0}} \\ \bar{\bar{0}} & \bar{\bar{0}} & \bar{\bar{0}} \end{pmatrix} \in \mathbb{R}^{3n\times 3n},$$

$$\bar{c} := \begin{pmatrix} -2A^T\bar{g}^\bullet \\ \alpha\bar{1} \\ \alpha\bar{1} \end{pmatrix} \in \mathbb{R}^{3n}, \quad K := \begin{pmatrix} L \\ -I \\ I \end{pmatrix} \in \mathbb{R}^{3n\times n}.$$

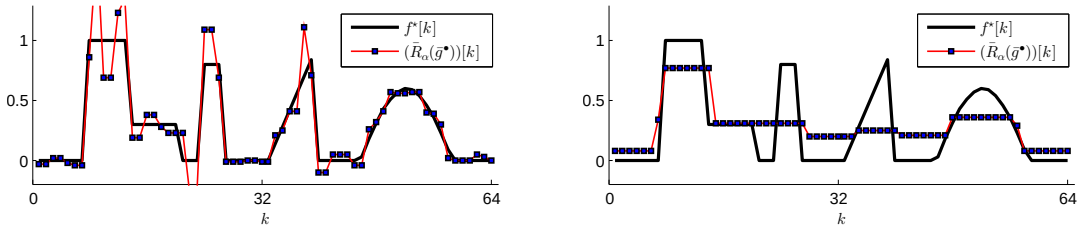
Then the total variation regularization problem (4.7) can be written on the form

$$\bar{R}_\alpha(\bar{g}^\bullet) := \arg\min_{\bar{x}} \left\{ \frac{1}{2}\bar{x}^TQ\bar{x} + \bar{c}^T\bar{x} \right\} \tag{5.13}$$

with the two constraints

$$K^T\bar{x} = \bar{0} \in \mathbb{R}^{3n}, \quad \bar{x} \geq \begin{pmatrix} -\infty\bar{1} \\ \bar{0} \\ \bar{0} \end{pmatrix} \in \mathbb{R}^{3n}.$$

The problem (5.13) can be solved with i.a. MATLABs `quadprog()` command. Using the simulated data $\bar{g}^\bullet$ with 2% noise, Figure 5.4 shows reconstructions $\bar{R}_\alpha(\bar{g}^\bullet)$ with total variation regularization for two choices of α (high and low). Comparing to Figure 5.3(b) it is easy to see improvements in the regularized inversion. As expected, the solution have less total variation and tends to become more 'piecewise constant'-like for higher α.



(a) $\alpha := 0.001$, $e_\% \approx 0.36$.

(b) $\alpha := 1$, $e_\% \approx 0.51$.

Figure 5.4: Total variation, low and high α.

By increasing α with 0.001 in each iteration, $\alpha = 0.0233$ was found to give the smallest error e from (3.5) in a realization of this concrete problem. The regularized solution for this choice of α is shown in Figure 5.5 with the relative error $e_\%$ from (3.6). The reconstruction in Figure 5.5 shows much better results than the naive reconstruction in 5.3(b). It is also evident how edges are preserved and the constant regions are reconstructed very well. The two smooth non-constant intervals unfortunately gets a 'staircase' shape from total variation regularization.

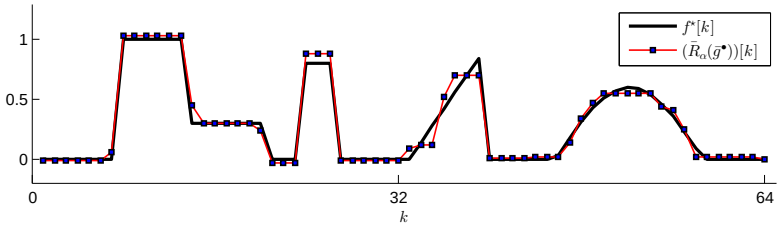


Figure 5.5: Total variation regularization, $\alpha := 0.031$, $e_{\%} \approx 0.12$.

5.3.3 Wavelet based regularized reconstruction

The simulated data $\bar{g}^\bullet$ with 2% noise is used to reconstruct an approximate version of the true function $f^\star$ with wavelet based regularization. The known variance σ^2 of the noise added to $\bar{m}^\bullet$ was used as a priori information. A reformulation of the discrete regularized solution $\bar{R}_\alpha(\bar{g}^\bullet)$ in (4.20) to quadratic programming form follows.

The number of wavelet scales was chosen to be $J := 6$ such that $2^J = n$. From (5.9) $A \in \mathbb{R}^{n\times n}$ is the convolution operator, the decomposition operator $D \in \mathbb{R}^{n\times n}$ from (4.18) maps a function into its finite sequence of wavelet coefficients at $J \in \mathbb{N}$ scales, and $W \in \mathbb{R}^{n\times n}$ from (4.13) is the Besov-norm $2^{j/2}$-scales weight matrix. The identity matrix is denoted $I \in \mathbb{R}^{n\times n}$, the vectors $\bar{0}$, $\bar{1} \in \mathbb{R}^n$ have all elements equal to zero or one respectively, and the matrix $\bar{\bar{0}} \in \mathbb{R}^{n\times n}$ have all elements equal to zero. Set $\bar{\alpha}_k := \alpha$ for all $k = 1, 2, ..., n$, and $\bar{v}_+ - \bar{v}_- := T\bar{f}$ such that $\bar{v}_+, \bar{v}_- \in \mathbb{R}_0^{+n}$ and define

$$\bar{x} := \begin{pmatrix} \bar{f} \\ \bar{v}_+ \\ \bar{v}_- \end{pmatrix} \in \mathbb{R}^{3n}, \quad Q := \begin{pmatrix} \frac{1}{\sigma^2}A^TA & \bar{\bar{0}} & \bar{\bar{0}} \\ \bar{\bar{0}} & \bar{\bar{0}} & \bar{\bar{0}} \\ \bar{\bar{0}} & \bar{\bar{0}} & \bar{\bar{0}} \end{pmatrix} \in \mathbb{R}^{3n\times 3n},$$

$$\bar{c} := \begin{pmatrix} -\frac{1}{\sigma^2}A^T\bar{g}^\bullet \\ \alpha\bar{1} \\ \alpha\bar{1} \end{pmatrix} \in \mathbb{R}^{3n}, \quad K := \begin{pmatrix} WD \\ -I \\ I \end{pmatrix} \in \mathbb{R}^{3n\times n}$$

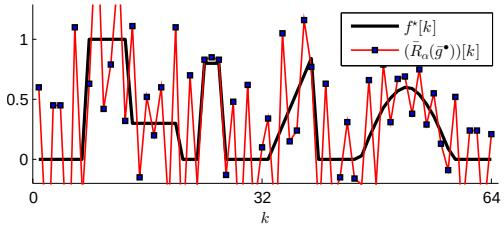
Then the problem (4.20) can be written on the quadratic programming form as

$$\bar{R}_\alpha(\bar{g}^\bullet) := \arg\min_{\bar{x}} \left\{ \frac{1}{2}\bar{x}^TQ\bar{x} + \bar{c}^T\bar{x} \right\}, \tag{5.14}$$

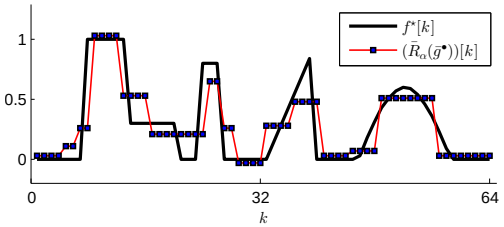
with the two constraints

$$K^T\bar{x} = 0 \in \mathbb{R}^{3n}, \quad \bar{x} \geq \begin{pmatrix} -\infty\bar{1} \\ 0 \\ 0 \end{pmatrix} \in \mathbb{R}^{3n}.$$

Using the Haar wavelet and Daubechies 2 (DB2) wavelet the reconstructions are shown in Figure 5.8 and Figure 5.9 with the relative error $e_{\%}$ from (3.6). The MATLAB tool `wavedec()` is used to decompose the function into its wavelet coefficients (acting as the

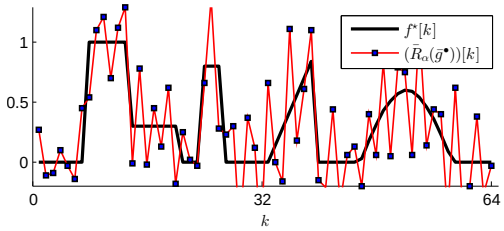


(a) $\alpha := 0.01$, $e_{\%} > 1$.

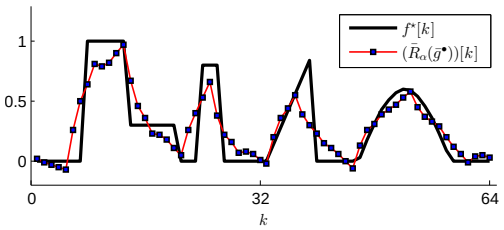


(b) $\alpha := 100$, $e_{\%} \approx 0.44$.

Figure 5.6: Haar wavelet, low and high α.



(a) $\alpha := 0.01$, $e_{\%} \approx 0.76$.



(b) $\alpha := 100$, $e_{\%} \approx 0.40$.

Figure 5.7: DB2 wavelet, low and high α.

operator D) and for both wavelets, reconstructions with a low and high choice of α is shown in Figure 5.6 and Figure 5.7.

The choice of α with the lowest error e in (3.5) was found iteratively at steps $\Delta\alpha = 0.01$ for both wavelets, and their reconstructions are shown in Figure 5.8 and Figure 5.9.

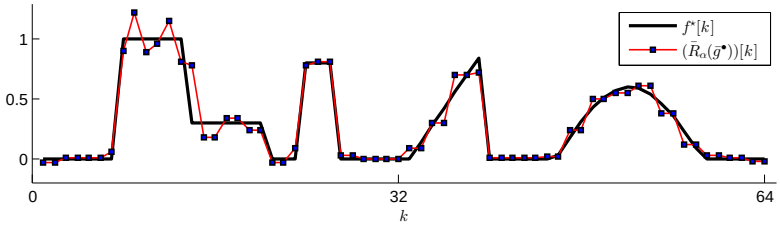


Figure 5.8: Haar wavelet, $\alpha := 0.63$, $e_{\%} \approx 0.21$.

Both wavelets reconstructs the edges, but do not perform as well as total variation regularization on the constant intervals. The DB2 performs best on the two smooth non-constant intervals. The Haar wavelet tends to recover this region as staircases as total variation regularization.

The magnitudes of wavelet coefficients of the reconstructions in Figure 5.8 and Figure 5.9 are investigated in Figure 5.10.

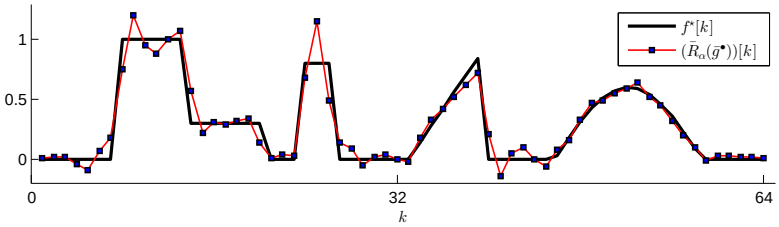
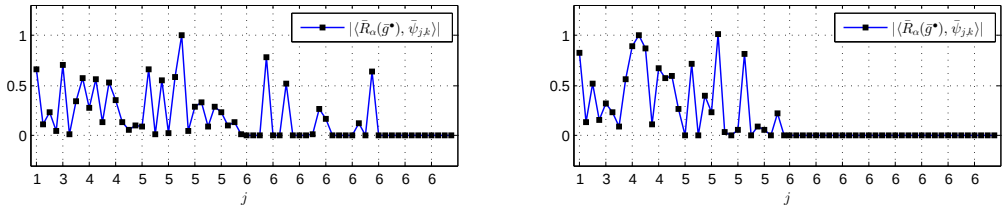


Figure 5.9: DB2 wavelet, $\alpha := 6.9$, $e_{\%} \approx 0.23$.



(a) Haar.

(b) DB2.

Figure 5.10: Wavelet coefficients magnitudes of regularized solutions.

The DB2 solution has sparser coefficients than the Haar solution in compliance with the number of vanishing moments for each wavelet basis. In addition, the Haar solution was reconstructed using a smaller regularization parameter α which causes the sparsity promoting ℓ_1-norm of the coefficients to have less weight in the Haar case.

Chapter 6

Computed tomography

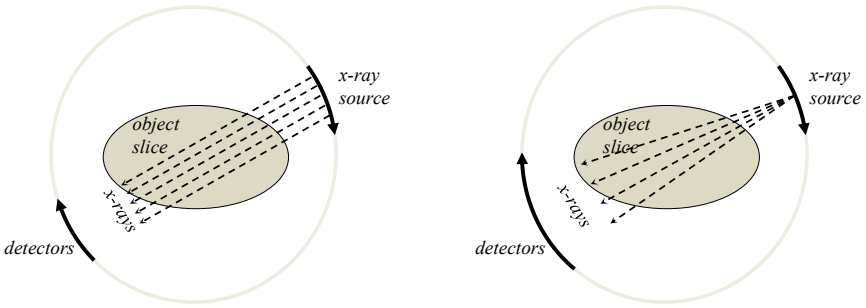
Computed tomography (CT) reconstructs images in two or three dimensions of the interior of a physical object by e.g. measuring x-ray intensity loss through the object at different angles. Especially in the medical industry, CT is intensively used at clinics and hospitals around the world for diagnosis and screening of patients. The following sections presents mathematical modelling for inverting 2D tomographic images and reconstructions are demonstrated using the edge-preserving total variation and wavelet based regularization methods. The same CT problem is used for curvelet based regularization in a later chapter.

6.1 X-rays and computed tomography

X-rays are electromagnetic waves which are explained as the result of combining an unsteady electric field acting as a magnetic source and an unsteady magnetic field acting as an electric source [41]. These two fields sustain each other and causes energy in the form of electromagnetic radiation (e.g. visible light, radiowaves or x-rays) to propagate through a medium at the speed of light. Energy of electromagnetic radiation is carried by the elementary particles photons and the energy level of a photon is proportional to the frequency of the radiation. X-ray photons are therefore of higher energies than light photons, allowing x-rays to penetrate matters that are impermeable to visible light. Different materials attenuate x-rays at different magnitudes making it possible to construct an image with color intensities corresponding to materials of different attenuation coefficients in the interior of an object.

A variety of geometric configurations of CT scanners exist for different purposes. The *parallel beam* and *fan beam* radiations in Figure 6.1 are two examples of CT scanner setups. In a *full angle* or *limited angle* 2D scan, the source and detectors are rotated 180° or less, respectively, in the circular path around the object.

Roughly speaking, higher x-ray intensity and more measured projections in a scan gives reconstructions of higher quality. On the other hand, a patient in a medical scan is exposed to a dose of x-ray radiation which is known to be a risk for long termed cell changes or cancer development. This dilemma motivates the research in mathematical



(a) Parallel beam.

(b) Fan beam.

Figure 6.1: CT scanner geometries.

models that can reconstruct good quality images based on less measurements.

The MATLAB generated Shepp-Logan head phantom in Figure 6.2 represents a simple simulated tomographic image of a cross section of a human head. The image is created with analytic expressions of ellipses and circles, making it easy to generate phantoms in different image sizes. The Shepp-Logan phantom will be used to test different regularization methods in tomographic inverse models.

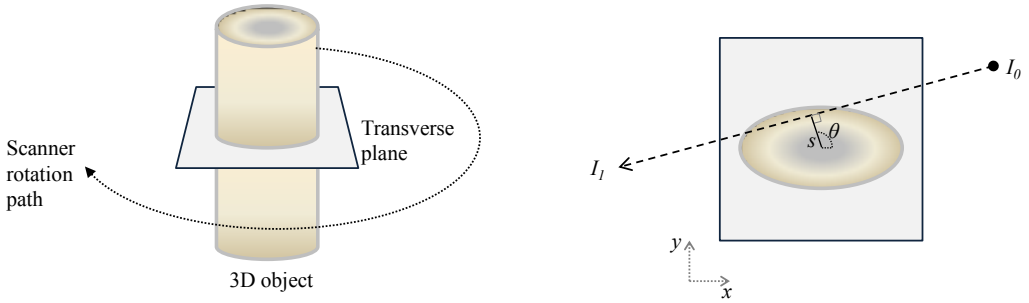
Figure 6.2: Shepp-Logan phantom.

6.2 Continuum modelling

Intensity of x-rays sent through an object is typically measured by counting photons in small areas on a detector plate. One x-ray beam is modelled as a straight line with a measured intensity I_1 and a known source intensity I_0 (I_0 corresponds to the intensity measured with no object present in the scanner). The interesting information is of course the intensities between I_0 and I_1.

Consider on the unit square $\Omega := [0,1[\times[0,1[\subset \mathbb{R}^2$ a function $f \in L^2(\Omega)$ returning atten-

uation coefficients $f(x, y)$ of materials at different x, y-coordinates in a cross sectional plane of an object. The transverse plane square in Figure 6.3 represents the support of f.



(a) Transverse plane.

(b) Single x-ray beam at arbitrary rotation and displacement.

Figure 6.3: Transverse plane of 3D object.

At a fixed $\tilde{y}$ and variable x the function $f(x, \tilde{y})$ represents attenuation coefficients along a straight horizontal line. With $I(x)$ denoting the intensities along this line, the boundary intensities are modelled as $I(0) = I_0$ and $I(1) = I_1$ and in between, a relative loss of intensity at a small distance $\mathrm{d}x$ is modelled relative to f as

$$-\frac{\mathrm{d}I(x)}{I(x)} = f(x, \tilde{y})\,\mathrm{d}x. \tag{6.1}$$

Integrating (6.1) then gives a known value (by I_0 and I_1) of the line integral:

$$\int_0^1 f(x, \tilde{y})\,\mathrm{d}x = \log_e(I_0) - \log_e(I_1). \tag{6.2}$$

The line integral (6.2) can be defined along arbitrary lines in the xy-plane, described by a displacement value $s \in \mathbb{R}$ (shortest distance from origin to the line) and an angle $\theta \in [0, 2\pi[$ (between the positive abscissa and the line with length s) as in Figure 6.3(b). Define the set U of Cartesian points on a line on $\mathbb{R}^2$ as

$$U = \{(x, y) \colon x\cos\theta + y\sin\theta = s\}.$$

and a vector $\bar{v} = (x, y) \in U$. In [12, 7] the *Radon transform* defined as

$$(\mathfrak{R}f)(s, \theta) := \int_U f(\bar{v})d\bar{v}. \tag{6.3}$$

is a compact operator that maps $L^2(\Omega)$ into a weighted L^2-space. A so-called *sinogram* as in Figure 6.4 shows a function in the Radon domain. Sinograms will be used in the following sections to depict simulated measurements from a CT scan.

The singular values (2.3) of $\mathfrak{R}$ in (6.3) is given with $j \in \mathbb{N}_0$ by

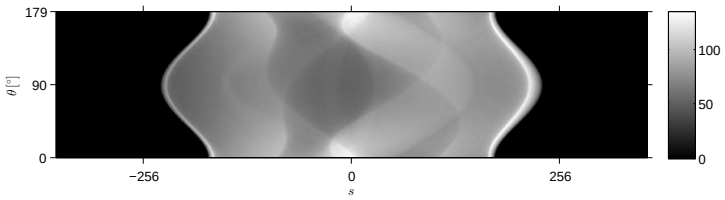


Figure 6.4: Sinogram of Shepp-Logan phantom in Figure 6.2.

$$\mu_{j,k} = \sqrt{\frac{4\pi}{j+1}}$$

with the multiplicity $k = 0, 1, ..., j$ [33, 7]. This is a not a very fast decay of the singular values, and the ideal Radon transform operator in (6.3) is therefore only expected to be only mildly ill-posed.

Roughly speaking, the continuous Radon transform (6.3) can be seen as an ideal model for the parallel beam setup in Figure 6.1(a). Johann Radon showed that if f is continuous and compactly supported, then $\mathfrak{R}f$ is uniquely determined from a complete integration along all lines on the support of f [13]. It has also been showed that a unique f cannot be determined in an inversion, if the number of projections are finite.

Inverting the Radon transform is related to the two dimensional Fourier transform $\hat{f}$ of $f \in L^2(\Omega)$ defined (with the usual extension from L^1) as

$$\hat{f}(\omega_x, \omega_y) := \int_{\mathbb{R}} \int_{\mathbb{R}} f(x, y)\, e^{-i2\pi(x\omega_x + y\omega_y)}\, \mathrm{d}x\, \mathrm{d}y,$$

and the Fourier inversion

$$f(x, y) = \int_{\mathbb{R}} \int_{\mathbb{R}} \hat{f}(\omega_x, \omega_y)\, e^{i2\pi(x\omega_x + y\omega_y)}\, \mathrm{d}\omega_x\, \mathrm{d}\omega_y. \tag{6.4}$$

The *central slice theorem* says that the Fourier transform of a projected line $\mathcal{L}(s, \theta)$ in the domain f equals a projected line $\hat{\mathcal{L}}(\theta)$ (same θ) trough origin in the Fourier transform of f. Denote the Fourier transform of $\mathfrak{R}f$ with respect to s as

$$\widehat{(\mathfrak{R}f)}(s, \theta) := \int_{\mathbb{R}} (\mathfrak{R}f)(s, \theta)\, e^{-its}\, \mathrm{d}t.$$

Then the Fourier slice theorem is defined in [32] as

$$\widehat{(\mathfrak{R}f)}(r, \theta) = \hat{f}(r\cos\theta, r\sin\theta). \tag{6.5}$$

Using (6.5) on the inverse Fourier transform (6.4) in polar coordinates reveals an explicit expression of $f(x, y)$ given $\mathfrak{R}f$ as

$$f(x,y) = \int_0^{\pi} \int_{\mathbb{R}} \widehat{(\mathfrak{R}f)}(r,\theta)\, e^{i2\pi s(x\cos\theta + y\sin\theta)}\, |s|\, \mathrm{d}s\, \mathrm{d}\theta. \tag{6.6}$$

Considering (6.6) as Fourier transforming $\mathfrak{R}f$ in r, multiplying with the radial parameter $|s|$, and then transforming back to the spatial domain shows that the inversion acts as a ramp filter which amplifies higher frequencies that are oriented along straight lines through origin. *Filtered backprojection* is the common name for the Radon inversion (6.6).

Another interesting relation to the Radon transform is the backprojection/layergram operator $\mathfrak{B}$ which maps a radon transformed function into integrals over all lines coinciding with a particular point in a polar coordinate system. With $x\cos(\theta) + y\sin(\theta) = s$, the backprojection operator is defined as

$$\left(\mathfrak{B}(\mathfrak{R}f)\right)(x,y) := \int_0^{\pi} (\mathfrak{R}f)(s,\theta)\, d\theta.$$

On $\mathbb{R}^2$-domains, the backprojection operator is related to the Hilbert adjoint $\mathfrak{R}^*$ of the Radon transform by the identity [13]

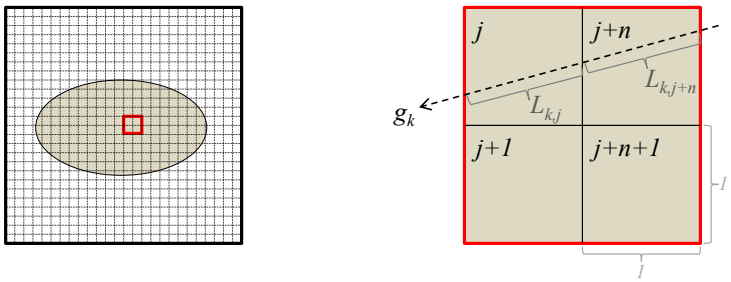
$$\mathfrak{R}^* = 2\mathfrak{B}. \tag{6.7}$$

The relation 6.7 is used later to model the transpose operator of a discrete approximate version of the Radon transform. The transpose operator is used by an iterative soft-thresholding algorithm which has the advantage that inversion can be carried out without defining matrices explicitly.

6.3 Discrete modelling

The following shows how a discrete approximate model can be constructed corresponding to the parallel beam geometry considered in the continuous case. A square containing the two dimensional object slice is considered as the area to be reconstructed as in Figure 6.5(a). This area is divided into $n \times n$ equally sized squares (pixels), each normalized to area 1^2. The red cut-out, magnified in Figure 6.5(b), illustrates how a single measured attenuation value g_k can be modelled. In the ith row and the pth column, each pixel is numbered by $j = i + n(p-1)$. Assuming that each pixel represents an area of constant x-ray attenuation (homogeneous material) or that a pixel represents an average value, f_j is the attenuation value in the jth pixel. The length that the kth x-ray beam intersects with the jth pixel is denoted $L_{k,j}$, and g_k is the measured attenuation value of the kth beam.

If the kth beam does not intersect a pixel j, $L_{k,j}$ is defined to be zero and thus each measured attenuation value g_k can be written as



(a) Object in squared $n \times n$ grid.

(b) Modelling one measurement.

Figure 6.5: Discrete CT model.

$$g_k = \sum_{j=1}^{n^2} f_j L_{k,j}.$$

The values of a forward operator A in this model is therefore only dependent on the geometric conditions between the scanner and object. Considering a vector of measurements $\bar{g} \in \mathbb{R}^m$ and a vector $\bar{f} \in \mathbb{R}^n$, a matrix $A \in \mathbb{R}^{m \times n}$ that models an inverse problem on the form $\bar{g} = A\bar{f} + \bar{\varepsilon}$ can be defined as

$$A := \begin{pmatrix} L_{1,1} & L_{1,2} & \cdots & L_{1,n-1} & L_{1,n} \\ L_{2,1} & L_{2,2} & \cdots & L_{2,n-1} & L_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ L_{m-1,1} & L_{m-1,2} & \cdots & L_{m-1,n-1} & L_{m-1,n} \\ L_{m,1} & L_{m,2} & \cdots & L_{m,n-1} & L_{m,n} \end{pmatrix}. \tag{6.8}$$

6.4 Demonstrations

With $n = 512^2$ a $\sqrt{n} \times \sqrt{n}$ pixels Shepp-Logan phantom $\bar{\bar{f}}^\star \in \mathbb{R}^{\sqrt{n} \times \sqrt{n}}$ was generated by MATLABs `phantom()` command. The vector $\bar{f}^\star \in \mathbb{R}^n$ denotes a column-wise concatenation of $\bar{\bar{f}}^\star$. For $i = 1, 2, ..., 180$ and $j = 1, 2, ..., 724 \approx (\sqrt{2}\sqrt{n})$, the angles $\theta_i \in \{0, 1, ..., 179\}$ and displacements $s_j \in \{-362, -361, ..., 360, 361\}$ was used to generate the discrete forward operator $A \in \mathbb{R}^{(724 \cdot 180) \times 512^2}$ in (6.8) using DTU AIRtools. A simulated sinogram $\bar{g}^\bullet \in \mathbb{R}^{(724 \cdot 30)}$, shown in Figure 6.6, was generated using A on $\bar{f}^\star$, and for $i = 1, 2, ..., 30$ interpolating to angles $\tilde{\theta}_i \in \{0, 6, ..., 174\}$, and finally adding Gaussian white noise. The 2D version of $\bar{g}^\bullet$ will be denoted $\bar{\bar{g}}^\bullet \in \mathbb{R}^{724 \times 30}$.

This problem is underdetermined with more than twelve times more pixels to be reconstructed than measurement values available. It becomes difficult to compute eigenvalues and singular values when matrices are of sizes at this scale. From section 3.2, the problem $\bar{g} = A\bar{f}$ has either zero or infinitely many solutions. Theory of the continuum full data Radon transform suggested that the problem is mildly ill-posed [33, 7].

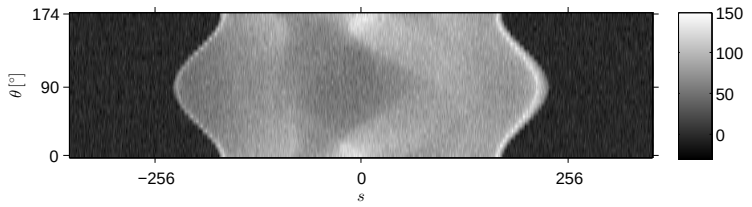


Figure 6.6: Sinogram of $\bar{\bar{g}}^{\bullet}$ with 2% noise.

Solving the CT demonstrations in the following sections requires the forward operator A and its transpose A^T applied on a vector. The MATLAB function `radon(x)` is used instead of $A\bar{x}$ from (6.8) and `iradon(x)` with filter `'None'` instead of $A^T\bar{x}$. This way, the solver is free of using large matrices, and the reconstructions are not made with the exact same model as the data is simulated with.

The `radon()` function models the problem slightly different than (6.8). Each pixel in the reconstructed area is divided into four equally sized squared subpixels which each either contribute fully to a single measurement (if the beam coincides with center of a subpixel) or evenly to two neighbouring measurements.

Figure 6.7 shows the absolute difference between two sinograms computed with the two different methods. At the 30 projection angles $\tilde{\theta}_i$ the transform is done in both cases on a 512×512 pixels Shepp-Logan phantom with no noise. The small difference in the two methods are seen in the sinogram.

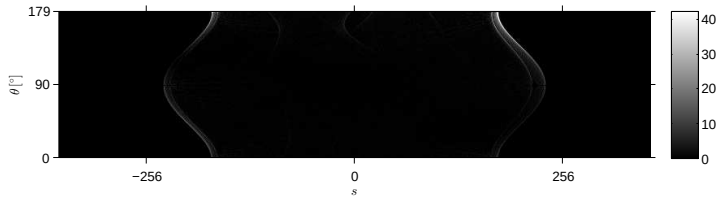
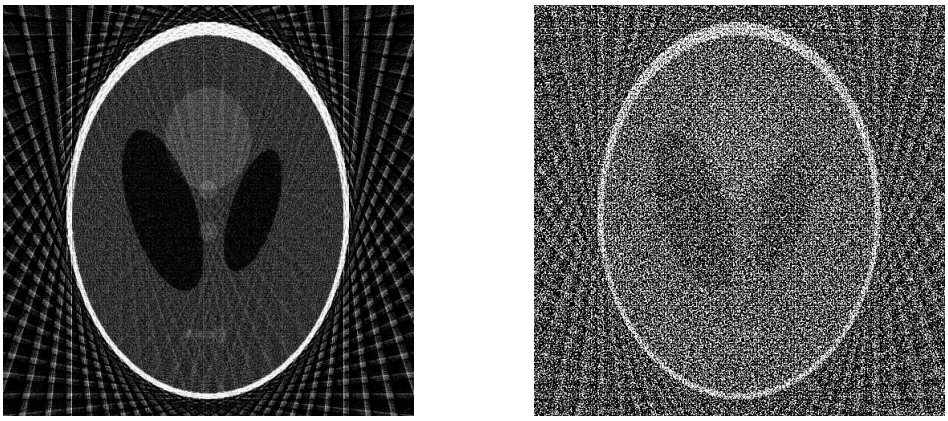


Figure 6.7: Absolute difference in two sinograms using MATLABs `radon()` and A in (6.8).

6.4.1 Filtered backprojection

MATLAB implements a discrete version of the filtered backprojection (6.6) using the discrete fast Fourier transform. Inverting the Radon transform is in a sense a naive approach to reconstruct CT images, but with sufficiently many x-ray projections and a sufficiently small norm of ε in (2.1), the problem is well-posed enough to use the filtered backprojection method. Figure 6.8 shows two reconstructions with different noise-levels added to the simulated measurements. With a standard deviation of $\sigma = 0.5 \max \bar{g}$, the filtered backprojection works very well.

In practice the filtered backprojection is the first choice of method when the CT problem



(a) $e_{\%} \approx 0.41$, PSNR ≈ 29.39dB. (b) $e_{\%} > 1$, PSNR ≈ 16.29dB.

Figure 6.8: Filtered backprojection, (a) 0.5% noise, (b) 2% noise.

is well-posed enough. The method is easy to implement, not memory requiring, fast to execute and reliable.

6.4.2 Total variation regularization

The total variation regularized solution (4.7) is approximated using the iterative Barzilai and Borwein gradient descent method [2, 22]. This method assumes homogeneous Neumann boundary conditions of the sought function $\bar{f} \in \mathbb{R}^n$ and the penalty term is made continuously differentiable by approximating the absolute value $|\cdot|$ in the ℓ^1-norm with $\sqrt{(\cdot)^2 + \beta}$ for a small $\beta > 0$. With the discrete vector-gradient $\tilde{D}$ from (4.6), the objective functional to be minimized takes the form

$$\Phi_{\alpha,\beta}(\bar{f}) := \|A\bar{f} - \bar{g}\|^2_{(\{1,2,\ldots,m\})} + \alpha \sum_{k=1}^{n} \sqrt{((\tilde{D}\bar{f})[k])^2 + \beta}. \tag{6.9}$$

The gradient $\nabla\Phi_{\alpha,\beta}(\bar{f})$ can then be determined. From [22] the gradient of the data-fidelity term is given by

$$\nabla\left(\|A\bar{f} - \bar{g}\|^2_{(\{1,2,\ldots,m\})}\right) = 2A^T(A\bar{f} - \bar{g}), \tag{6.10}$$

and the gradient of the penalty-term is given component-wise by

$$\frac{\partial}{\partial \bar{f}_k}\left(\alpha \sum_{k=1}^{n} \sqrt{((\tilde{D}\bar{f})[k])^2 + \beta}\right) = \frac{2\bar{f}_k - \bar{f}_{k+\sqrt{n}} - \bar{f}_{k+1}}{\sqrt{(\bar{f}_{k+\sqrt{n}} - \bar{f}_k)^2 + (\bar{f}_{k+1} - \bar{f}_k)^2 + \beta}} + \frac{\bar{f}_k - \bar{f}_{k-\sqrt{n}}}{\sqrt{(\bar{f}_k - \bar{f}_{k-\sqrt{n}})^2 + (\bar{f}_{k-\sqrt{n}+1} - \bar{f}_{k-\sqrt{n}})^2 + \beta}} + \frac{\bar{f}_k - \bar{f}_{k-1}}{\sqrt{(\bar{f}_k - \bar{f}_{k-1})^2 + (\bar{f}_{k+\sqrt{n}-1} - \bar{f}_{k-1})^2 + \beta}}. \quad (6.11)$$

For iterations $i = 0, 1, ..., I$, a starting guess $\bar{f}^0 := \bar{0}$ and the adaptive step sizes

$$\lambda_{i+1} := \frac{(\bar{f}^{i+1} - \bar{f}^i)^T(\bar{f}^{i+1} - \bar{f}^i)}{(\bar{f}^{i+1} - \bar{f}^i)^T(\nabla\Phi_{\alpha,\beta}(\bar{f}^{i+1}) - \nabla\Phi_{\alpha,\beta}(\bar{f}^i))},$$

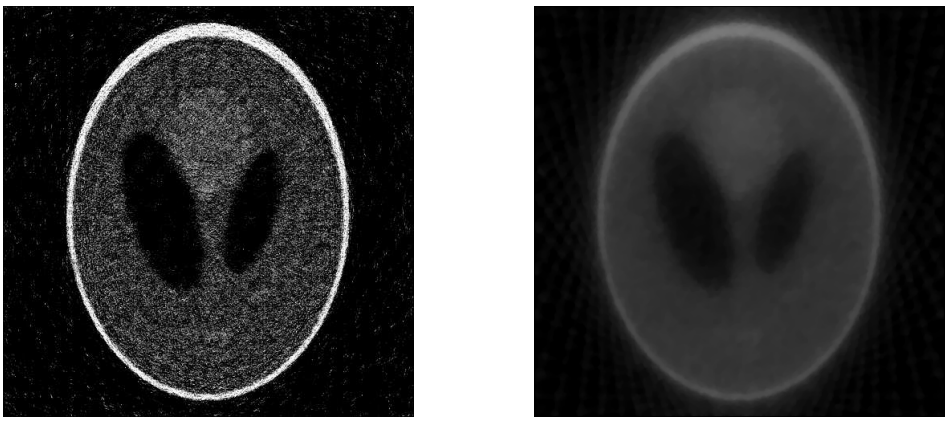
the iterative solutions in the Borwein-Barzilai method is given by

$$\bar{f}^{i+1} := P(\bar{f}^i - \lambda_i \nabla L_\beta(\bar{f}^i)) \quad (6.12)$$

with the operator P projecting a vector to its assumed non-negative region and defined as

$$\left(P(\bar{f})\right)_k := \begin{cases} \bar{f}_k & \text{if } \bar{f}_k \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

The iterations in (6.12) converges to a solution of the total variation regularized inverse problem. The result of reconstructing the simulated data $\bar{g}^\bullet$ with a low and high choice of the regularization parameter α are shown in Figure 6.9. As for 1D total variation the reconstruction becomes more 'piecewise constant'-like for higher α.

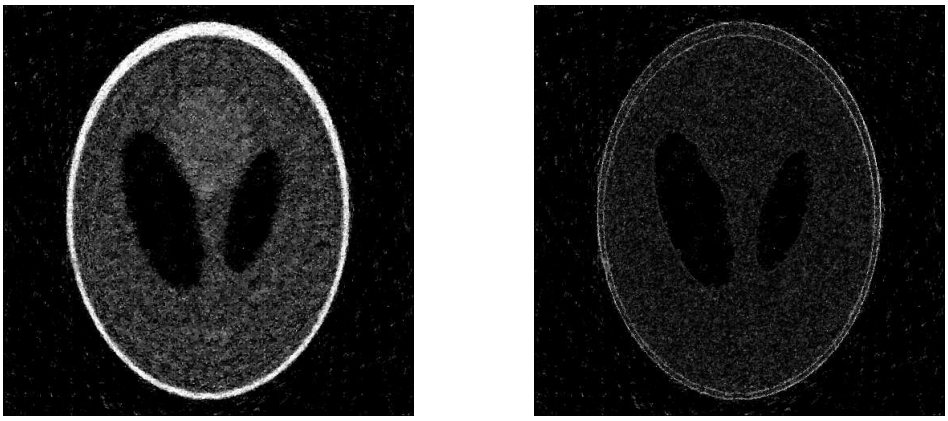


(a) $\bar{R}_{\alpha,\beta}(\bar{g}^\bullet)$, $\alpha := 0.001$, $e_{\%}\approx 0.17$.

(b) $\bar{R}_{\alpha,\beta}(\bar{g}^\bullet)$, $\alpha := 20$, $e_{\%}\approx 0.34$.

Figure 6.9: Total variation, low and high α.

A choice of α giving the lowest error $e_{\%}$ is shown in Figure 6.10 together with the absolute difference between the reconstruction and the true solution.



(a) $\bar{R}_{\alpha,\beta}(\bar{g}^{\bullet})$. (b) $|\bar{f}^{\star} - \bar{R}_{\alpha,\beta}(\bar{g}^{\bullet})|$.

Figure 6.10: Total variation, 2% noise, $\alpha := 1.48$, $e_{\%} \approx 0.12$, PSNR ≈ 35.42dB.

The convergence question of the Barzilai Borwein is more sophisticated than many other iterative methods because of the variable step-sizes. The method does not guarantee a strictly monotone decrease of the objective functional after each iteration. Figure 6.11 shows the convergence, during 200 iterations, of the regularized solution shown in Figure 6.10.

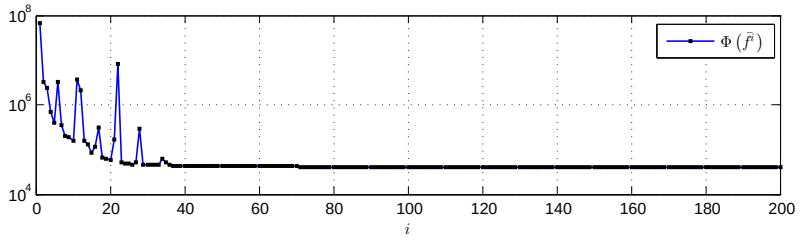


Figure 6.11: Logarithmic convergence plot from solving $\bar{R}_{\alpha,\beta}(\bar{g}^{\bullet})$ in Figure 6.10.

6.4.3 Wavelet based regularization

An iterative soft-thresholding algorithm is used on the wavelet based regularized solution (4.20). With the tomography forward operator $A \in \mathbb{R}^{m\times n}$, the wavelet decomposition operator $D \in \mathbb{R}^{2^J\times n}$ from (4.18) and the Besov $2^{j/2}$ scales weight-matrix $W \in \mathbb{R}^{2^J\times 2^J}$ from (4.19), the objective functional takes the form

$$\Phi_{\alpha}\left(\bar{f}\right) := \frac{1}{2}\left\|A\bar{f} - \bar{g}^{\bullet}\right\|^2_{\ell^2(\{1,2,\ldots,m\})} + \alpha\sigma^2\left\|WD\bar{f}\right\|_{\ell^1(\{1,2,\ldots,2^J\})}. \tag{6.13}$$

With the wavelet reconstruction operator R from (4.16) and the α-dependent soft-thresholding operator, defined component-wise for a vector $\bar{x} \in \mathbb{R}^n$, as

$$(S_\tau(\bar{x}))_k := \begin{cases} \bar{x}_k - \tau_k \mathtt{sign}(\bar{x}_k) & \text{for } |\bar{x}_k| \geq \tau_k, \\ 0 & \text{for } |\bar{x}_k| < \tau_k, \end{cases} \tag{6.14}$$

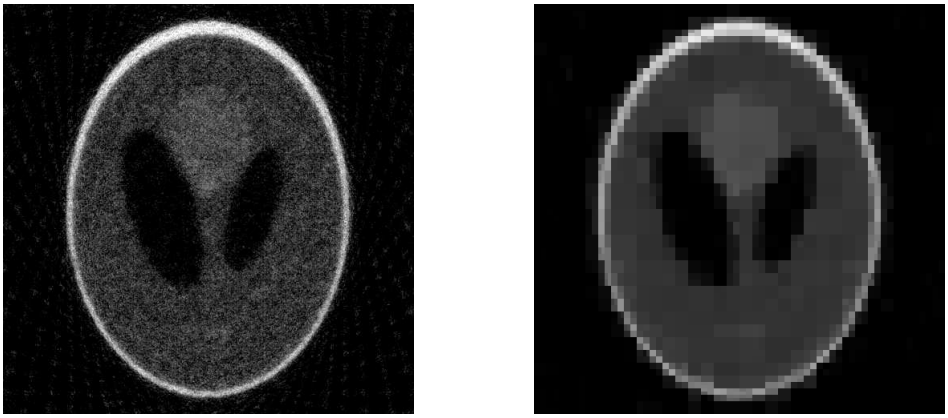
a minimizer $\bar{f}^*$ of (6.13), for $\tau_k := \alpha\,\sigma^2\,W_{k,k}$, then has to satisfy [11, 37]

$$\bar{f}^* = (RS_\tau D)\left(\bar{f}^* + A^T\left(A\bar{f}^* - \bar{g}^\bullet\right)\right). \tag{6.15}$$

With a step-size $\lambda \in \mathbb{R}$ satisfying $0 < \lambda < 2/\|A\|^2$ the iterations

$$\bar{f}^{i+1} = (RS_{\alpha\sigma^2}D)\left(\bar{f}^i + \lambda A^T\left(\bar{g}^\bullet - A\bar{f}^i\right)\right)$$

converges to $\bar{f}^*$ in (6.15) [4]. With $\lambda := 10^{-4}$, the Haar and DB2 wavelets are used to reconstruct the simulated CT data $\bar{f}^\star \in \mathbb{R}^n$. Figure 6.12 shows the reconstructions with a low and high choice of α using the Haar wavelet.



(a) $\bar{R}_\alpha(\bar{g}^\bullet)$, $\alpha := 0.001$, $e_{\%}{\approx}0.18$.

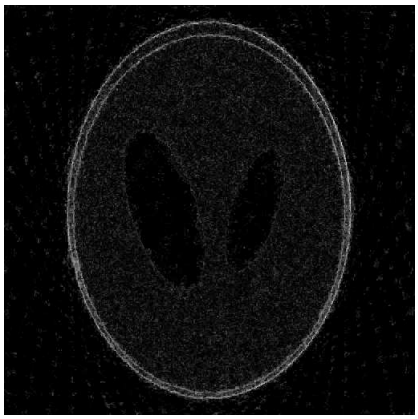
(b) $\bar{R}_\alpha(\bar{g}^\bullet)$, $\alpha := 20$, $e_{\%}{\approx}0.22$.

Figure 6.12: Haar wavelet, low and high α.

In Figure 6.13 the reconstruction with the choice of α giving the lowest relative error is shown together with the absolute difference between the reconstruction and the true image. The convergence of the iterative algorithm from the reconstruction in Figure 6.13 is shown in Figure 6.14.

Figure 6.15 shows the reconstructions using the DB2 wavelet for a low and high choice of α.

(a) $\bar{R}_\alpha(\bar{g}^\bullet)$.



(b) $|\bar{f}^\star - \bar{R}_\alpha(\bar{g}^\bullet)|$.

Figure 6.13: Haar wavelet, $\alpha := 0.5$, $e_{\%} \approx 0.14$, $\texttt{PSNR} \approx 36.73\texttt{dB}$.

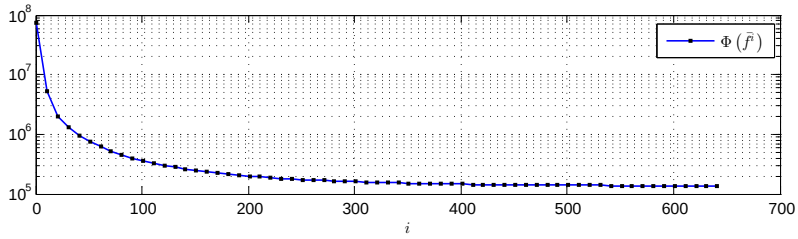
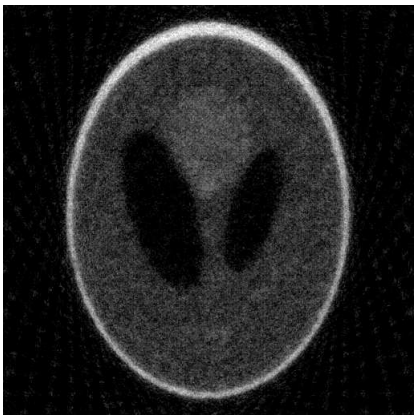


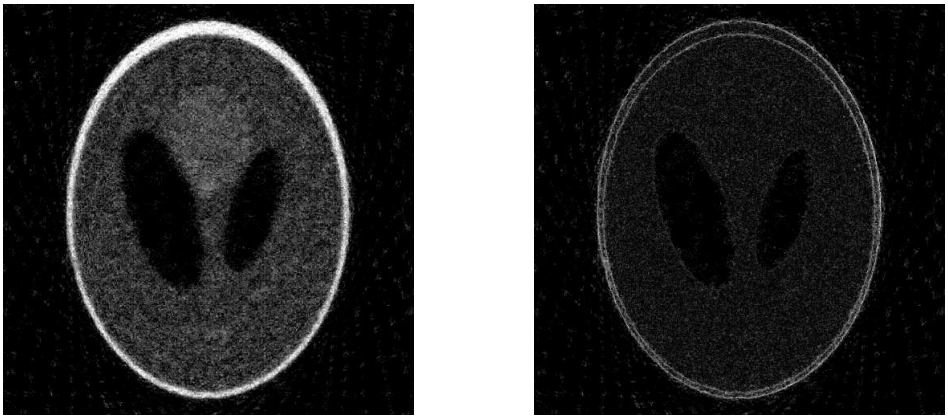
Figure 6.14: Logarithmic convergence plot from solving $\bar{R}_\alpha(\bar{g}^\bullet)$ in Figure 6.13.



(a) $\bar{R}_\alpha(\bar{g}^\bullet)$, $\alpha := 0.001$, $e_{\%}\approx 0.20$.

(b) $\bar{R}_\alpha(\bar{g}^\bullet)$, $\alpha := 20$, $e_{\%}\approx 0.24$.

Figure 6.15: DB2 wavelet, low and high α.



(a) $\bar{R}_\alpha(\bar{g}^\bullet)$. (b) $|\bar{f}^\star - \bar{R}_\alpha(\bar{g}^\bullet)|$.

Figure 6.16: DB2, 2% noise, $\alpha := 1$, $e_{\%} \approx 0.5$, PSNR ≈ 35.53dB.

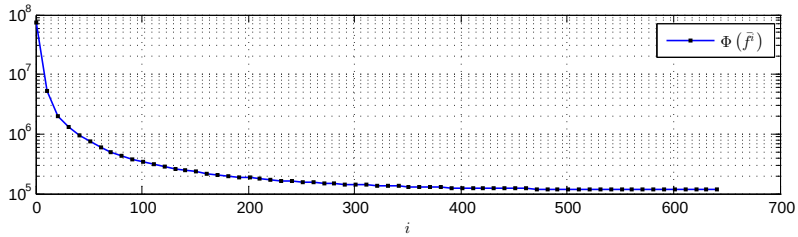


Figure 6.17: Logarithmic convergence plot from solving $\bar{R}_\alpha(\bar{g}^\bullet)$ in Figure 6.16.

Chapter 7

Curvelets

Before turning to the main subject of curvelet based regularization, this chapter reviews the second generation curvelet transform based on the work of Candés and Donoho in [6, 5]. The continuum curvelet transform is examined in the first section, followed by a review of the fast discrete curvelet transform in section 7.2. Finally, some important properties of curvelets with respect to regularization are stated in section 7.3.

7.1 Curvelet transform

The curvelet transform allows a function $f \in L^2(\mathbb{R}^2)$ to be expressed in terms of an infinite linear combination of scalar coefficients and curvelet functions. Curvelets does not constitute a basis like wavelets, but a *frame* for $f \in L^2(\mathbb{R}^2)$ as defined below from [9].

Definition 7.1 (Frame) *In a non-trivial and separable Hilbert space $\mathcal{H}$ an infinite sequence of functions $\{\varphi_i\}_{i\in\mathbb{N}} \subset \mathcal{H}$ is a frame for $\mathcal{H}$ if there exists frame bounds $A, B > 0$ such that for all $f \in \mathcal{H}$*

$$A \|f\|_{\mathcal{H}}^2 \leq \sum_{i\in\mathbb{N}} |\langle f, \varphi_i\rangle_{\mathcal{H}}|^2 \leq B \|f\|_{\mathcal{H}}^2. \tag{7.1}$$

Curvelets are constructed to be a *tight* frame which has equal bounds $B = A$ and further to have the normalized bound $A = 1$. Therefore, for a sequence of curvelets $\{\varphi_i\}_{i\in\mathbb{N}} \subset L^2(\mathbb{R}^2)$, (7.1) reduces to the *Parseval* relation

$$\sum_{i\in\mathbb{N}} |\langle f, \varphi_i\rangle_{L^2(\mathbb{R}^2)}|^2 = \|f\|_{L^2(\mathbb{R}^2)}^2, \quad \forall f \in L^2(\mathbb{R}^2). \tag{7.2}$$

A tight frame of curvelets $\{\varphi_i\}_{i\in\mathbb{N}} \subset L^2(\mathbb{R}^2)$ allows a function $f \in L^2(\mathbb{R}^2)$ to be expanded as

$$f = \sum_{i\in\mathbb{N}} \langle f, \varphi_i\rangle_{L^2(\mathbb{R}^2)}\, \varphi_i \tag{7.3}$$

with equality meaning that for all $\epsilon > 0$

$$\|f - \sum_{i\in\mathbb{N}} \langle f, \varphi_i\rangle_{L^2(\mathbb{R}^2)}\, \varphi_i\|_{L^2(\mathbb{R}^2)} < \epsilon.$$

In contrast to an orthonormal basis of wavelets, the curvelets expansion might have non-unique coefficients $\langle f, \varphi_i\rangle_{L^2(\mathbb{R}^2)}$.

Curvelet functions are constructed by dividing the 2D Fourier domain into windows as in Figure 7.1. A sequence of curvelets will from here be denoted with the indexing $\{\varphi_{j,\ell,\bar{k}}\}_{j,\ell,\bar{k}}$, where $j \in \mathbb{N}_0 \cup \{-1\}$ is a scaling parameter, $\ell \in \mathbb{N}_0$ is a rotation parameter and $\bar{k} \in \mathbb{Z}^2$ is a translation parameter.

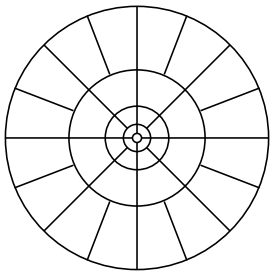


Figure 7.1: Supports of curvelets in the 2D frequency domain.

The center circle in Figure 7.1 acts as a low-pass filter similar to the scaling function in wavelets and is defined to at scale $j = -1$. Away from the center circle the frequency domain is divided into parabolic windows (wedges) of length $\approx 2^j$ and width $\approx 2^{j/2}$ for $j \in \mathbb{N}_0$. Each wedge is constructed mathematically by considering a radial window $W(r), r \in [1/2, 2]$ and an angular window $V(t), t \in [-1, 1]$. The windows are thought of as smooth, real-valued and non-negative functions that satisfy the admissibility conditions

$$\sum_{j\in\mathbb{Z}} W^2\left(2^j r\right) = 1, \quad r \in [1/2, 2] \tag{7.4}$$

and

$$\sum_{\ell\in\mathbb{Z}} V^2\left(t - \ell\right) = 1, \quad t \in [-1, 1]. \tag{7.5}$$

At each scale $j \geq 0$, generating curvelets $\hat{\varphi}_{j,0,\bar{0}}$ having support in a wedge in the frequency domain is then defined in polar coordinates $r \in [1/2, 2]$ and $\theta \in [0, 2\pi[$, using the floor operator $\lfloor\cdot\rfloor\colon\ \mathbb{R} \to \mathbb{Z}$ rounding down to nearest integer, as

$$\hat{\varphi}_{j,0,\bar{0}}\left(r, \theta\right) := 2^{-3j/4}\, W\left(2^{-j} r\right)\, V\left(2^{\lfloor j/2\rfloor}\frac{\theta}{2\pi}\right). \tag{7.6}$$

To avoid working with complex valued numbers the generating curvelets are thought of as being symmetric such that $\hat{\varphi}_{j,0,\bar{0}}(r, \theta) = \hat{\varphi}_{j,0,\bar{0}}(r, \theta + \pi)$.

The generating curvelets are rotated and translated in the spatial domain. Consider the rotation matrix R_θ defined via

$$R_\theta := \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}, \quad R_\theta^{-1} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \tag{7.7}$$

the equispaced angles $\{\theta_\ell\}_\ell$ at each j, defined with

$$\theta_\ell := (2\pi)2^{-\lfloor j/2 \rfloor}\ell,$$

and the operator $K_{\theta_\ell,\bar{k}}\colon\ \mathbb{R}^2 \to \mathbb{R}^2$ that rotates and translates coordinates in $\mathbb{R}^2$ by θ_ℓ and $\bar{k} = (k_x, k_y)$, defined as

$$K_{\theta_\ell,\bar{k}}(x,y) := R_{\theta_\ell}\left((x,y) - R_{\theta_\ell}^{-1}\left(2^{-j}k_x, 2^{-j/2}k_y\right)\right),$$

In terms of the generating curvelets and $K_{\theta_\ell,\bar{k}}$ a curvelet at each rotation ℓ and translation $\bar{k}$ is then described for $j \geq 0$ by

$$\varphi_{j,\ell,\bar{k}}(x,y) = \varphi_{j,0,\bar{0}}\left(K_{\theta_{j,\ell},\bar{k}}(x,y)\right).$$

At scale $j = -1$ the low-pass function with support in the center circle in Figure 7.1 is constructed with a radial window W_{-1} in the frequency domain and defined such that

$$|W_{-1}(r)|^2 + \sum_{j\in\mathbb{N}_0} \left|W(2^{-j}r)\right|^2 = 1.$$

Then the generating curvelet at the coarsest scale $j = -1$, without orientation, is defined in the frequency domain, as

$$\hat{\varphi}_{-1,0,\bar{0}}(r,\theta) = W_{-1}(r).$$

The translates of $\varphi_{-1,0,\bar{k}}$ then complete the frame $\{\varphi_I\}_I$ when

$$I = \left\{(-1,0,\bar{k})\colon \bar{k} \in \mathbb{Z}^2\right\} \cup \left\{(j,\ell,\bar{k})\colon j \in \mathbb{N}_0, -2^{\lfloor j/2 \rfloor} \geq \ell < 2^{\lfloor j/2 \rfloor}, \bar{k} \in \mathbb{Z}^2\right\}.$$

Figure 7.2 shows examples in the spatial domain of a curvelet at different scales and orientations. The curvelet at scale $j = -1$ shown in (a) stands out since the curvelet here is without orientation. At the finer scales $j \geq 0$ the curvelets have significant support in lengths approximately equal to their width squared. This elongated shape allows the curvelets to capture directions of discontinuities along curves. At a glance, scaling captures different frequencies and rotation captures different directions.

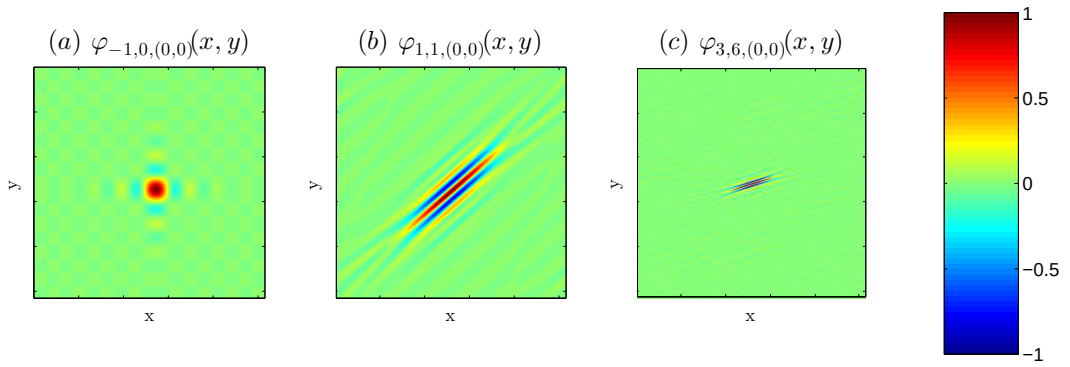


Figure 7.2: Curvelets at different scales j and rotations ℓ

7.2 Discrete curvelets

The round shapes of the wedges in Figure 7.1 and the use of angles and π does not adapt very well to a discrete $\mathbb{R}^2$ domain, and in the discrete curvelet transform the frequency domain is divided into sheared wedges instead as in Figure 7.3.

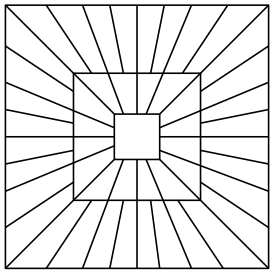


Figure 7.3: Supports of discrete curvelets in the 2D frequency domain.

A function $\phi\colon \mathbb{R} \to \mathbb{R}$ obeying that $0 \leq \phi \leq 1$ defines a window $\tilde{W}_j$ on $(\omega_x, \omega_y) \in \mathbb{R}^2$ in the frequency domain for each scale $j \geq 0$ by

$$\tilde{W}_j(\omega_x, \omega_y) := \sqrt{\left(\phi\left(2^{1-j}\omega_x\right)\,\phi\left(2^{1-j}\omega_y\right)\right)^2 - \left(\phi\left(2^{-j}\omega_x\right)\,\phi\left(2^{-j}\omega_y\right)\right)^2}.$$

With the angular window V satisfying (7.5), the window $V_j(\omega_x, \omega_y)$ is defined for each $j \geq 0$ as

$$V_j(\omega_x, \omega_y) := V\left(2^{\lfloor j/2 \rfloor}\,\frac{\omega_y}{\omega_x}\right),$$

and a shear window $\tilde{U}_j(\omega_x, \omega_y)$ at scale j can then described by

$$\tilde{U}_j(\omega_x, \omega_y) := \tilde{W}_j(\omega_x, \omega_y)\, V_j(\omega_x, \omega_y).$$

Instead of the rotation matrix (7.7) a shear matrix is used in the discrete case defined as

$$S_\theta := \begin{pmatrix} 1 & 0 \\ -\tan\theta & 1 \end{pmatrix}. \tag{7.8}$$

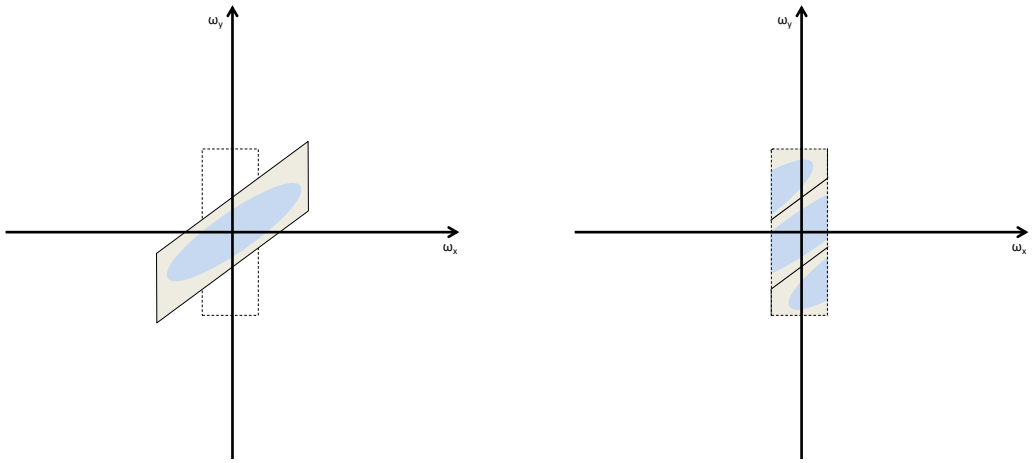
Here the family of angles θ_ℓ are defined such that the slopes $\tan\theta_\ell$ are equispaced rather than the angles θ_ℓ, and are defined for $\ell \in \{-2^{\lfloor j/2 \rfloor}, -2^{\lfloor j/2 \rfloor}+1, ..., 2^{\lfloor j/2 \rfloor}-1\}$ by

$$\tan\theta_\ell := \ell\, 2^{\lfloor j/2 \rfloor}.$$

With (7.8) and $\bar{\omega} = (\omega_x, \omega_y)$ the discrete windows at each scale j and shear ℓ then takes the form

$$\tilde{U}_{j,\ell}(\bar{\omega}) := \tilde{W}_j(\bar{\omega})\, V_j(S_{\theta_\ell}\, \bar{\omega}). \tag{7.9}$$

In practice the discrete curvelet transform can be applied to a function by multiplying, at each j and ℓ, the discrete Fourier transform of the function with the window (7.9) on the same discrete grid. The resulting windowed Fourier transformed is then considered periodic and the center window is wrapped around origin as in Figure 7.4. The inverse Fourier transform is then applied to the wrapped function at each scale and shear to obtain the frame coefficients to be translated throughout the spatial domain.



(a) Before wrapping.

(b) After wrapping.

Figure 7.4: Wrapping of curvelets.

The `MATLAB` toolbox CurveLab 2.1.3 from http://curvelet.org implements the discrete curvelet transform and is used to demonstrate curvelet based regularization numerically in the next chapter.

7.3 Some properties of curvelets

Curvelets posses properties that motivates their application to certain types of functions described as *functions that are C^2 except for discontinuities along piecewise C^2 curves.* The class of all such functions is denoted $\mathcal{C}$. The Shepp-Logan phantom constructed by ellipses and circles comply with $\mathcal{C}$.

In [6] curvelets are argued to *provide an essentially optimal sparse representation of* $f \in \mathcal{C}$. In particular it is shown that for $f \in \mathcal{C}$, a partial reconstructed approximation $f_n = \sum_{i_n} \langle f, \varphi_i \rangle_{L^2(\mathbb{R}^2)} \varphi_i$ using the n largest terms in (7.3), indexed i_n, will approximate the true function with an error bounded as

$$\|f - f_n\|_{L^2}^2 \leq C \cdot n^{-2} \cdot (\log n)^3 \text{ as } n \to \infty, \tag{7.10}$$

for a constant $C > 0$. The estimate (7.10) is useful since it in practice only is possible to deal with sums of a finite number of terms. In Figure 7.5 the error decay is compared to the corresponding n-largest term approximation for wavelets which is of order n^{-1}.

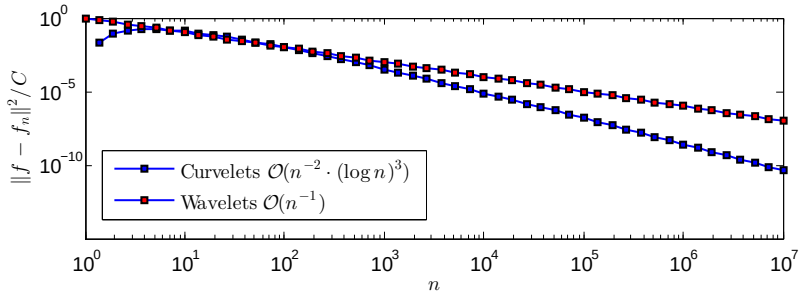


Figure 7.5: Error convergence for $f \in \mathcal{C}$ using n largest terms

In (7.3) - the frame expansion of f using curvelets - the coefficients $\langle f, \varphi_i \rangle_{L^2(\mathbb{R}^2)}$ are the inner products between the function f and the curvelets at different parameters $i = (j, \ell, k)$. Curvelets are constructed to provide the significant coefficients in the cases where i coincide such that at these particular scales, orientations and translations, the curvelets are aligned with an edge of the image. Following an illustrative heuristic argument from [6, 17] on the utility of curvelets, Figure 7.6 shows three situations of a curvelet and an arbitrary edge in the spatial domain.

In the first situation (a) where the curvelet is aligned with the edge *and* its significant support covers the edge, the coefficient will become large in magnitude. In contrast, the other situations (b) and (c) don't comply with the same two properties, and the coefficients in these cases will ideally be close to zero.

Finally, for an understanding of the smoothness of curvelet decomposed functions and comparison with the wavelet based regularization, in [3] a decomposition space consisting

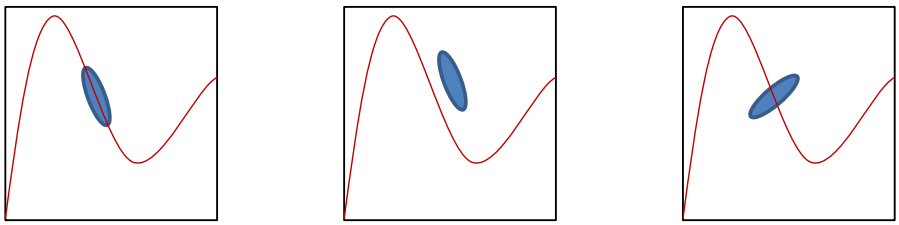


Figure 7.6: Three situations of a curvelet (blue ellipse) and an edge (red curve)

of functions $f \in \mathcal{C}$ is proved embedded continuously in the Besov space $B^s_{p,q}(\mathbb{R}^2)$, defined in Chapter 4.3, with the choices $p = q = 2/3$ and $s = 5/4 + \epsilon$ for all $\epsilon > 0$.

Chapter 8

Curvelet based regularization

The curvelet based regularization method presented in this chapter is based on the work of Jürgen Frikel in [19]. The concept is very similar to the one of wavelet based regularization examined in Chapter 4.3. The article [19] focuses on discrete modelling and the first section of this chapter presents the main topics. For ideas to continuum curvelet regularization modelling, decomposition spaces and embeddings in Besov smoothness spaces the reader is referred to e.g. [30, 29, 40, 3]. In the final section the curvelet based regularization method is applied on the CT problem from Chapter 6.4.

8.1 Modelling

With a curvelet reconstruction operator $R\colon \ell^2(\{1,2,...,N\}) \to \ell^2(\{1,2,...,n\})$ defined for a sequence of curvelet coefficients $\bar{c} := \{c_i\}_{i=1}^N \in \ell^2(\{1,2,...,N\})$ as

$$R\bar{c} := \sum_{i=1}^{N} c_i \, \varphi_i,$$

a forward operator $A \in \mathbb{R}^{m\times n}$ and a weight matrix $W \in \mathbb{R}^{m\times n}$ having appropriate weights in the diagonal, a curvelet based regularized solution is modelled as

$$\bar{f}^* := \underset{\bar{c}\in\ell^2(\{1,2,...,N\})}{\arg\min} \left\{ \frac{1}{2}\|AR\bar{c} - \bar{g}\|^2_{\ell^2(\{1,2,...,m\})} + \alpha\|W\bar{c}\|_{\ell^1(\{1,2,...,M\})} \right\}. \tag{8.1}$$

In [19] a sequence of weights $\tau_{j,\ell}$ depending on the curvelet scale j and rotation ℓ is suggested defined as

$$\tau_{j,\ell} := 2^{j-J}\sigma\sqrt{2\log_e N_{j,\ell}}$$

with J being the finest curvelet scale used (largest j) and $N_{j,\ell}$ is the number of curvelet coefficients at a certain scale and rotation. As in wavelet based regularization σ denotes standard deviation of the errors.

The problem (8.1) has by [19, 11, 37] at least one minimizer and using the soft-thresholding operator S_τ in (6.14), the fixed point iterations

$$\bar{c}^{i+1} = S_{\tau^i}\left(\bar{c}^i - \lambda_i (AR)^T \left(AR\bar{c}^i - \bar{g}\right)\right) \tag{8.2}$$

for $i \in \mathbb{N}_0$ converges to a minimizer with step-sizes λ_i that satisfy

$$0 < \lambda_i < 2/\|AR\|^2.$$

Alternatively to the soft-thresholding operator a *firm-thresholding* operator from [37] is used with the advantage that a higher penalty weight α can be used. This makes it possible to use fewer iterations. The firm-thresholding operator is defined with a constant ρ satisfying that

$$0 < \tau/2 \leq \rho \leq \tau$$

as

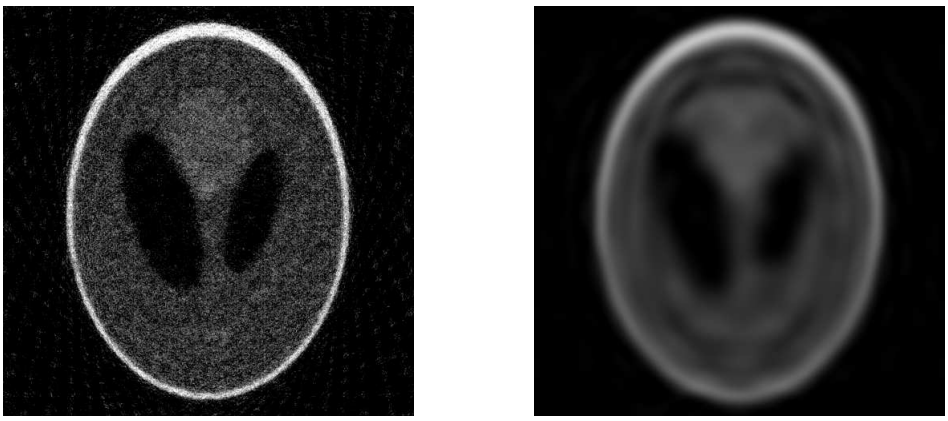
$$(S_{\rho,\tau}(\bar{x}))_k := \begin{cases} \bar{x}_k - (2\rho - \tau) & \text{for} \quad \bar{x}_k \geq \tau, \\ 2(\bar{x}_k - \rho) & \text{for} \quad \rho < \bar{x}_k < \tau, \\ 0 & \text{for} \quad -\rho \leq \bar{x}_k \leq \rho, \\ 2(\bar{x}_k + \rho) & \text{for} \quad -\tau < \bar{x}_k < -\rho, \\ \bar{x}_k + (2\rho - \tau) & \text{for} \quad \bar{x}_k \leq -\tau. \end{cases} \tag{8.3}$$

Unlike the soft-thresholding method the firm-thresholding does not shrink expansion coefficients in an iteration if coefficients are sufficiently large in magnitudes. The parameter ρ tunes this setting and is chosen to be the smallest allowed value $\rho = \tau_k/2$.

8.2 Demonstrations

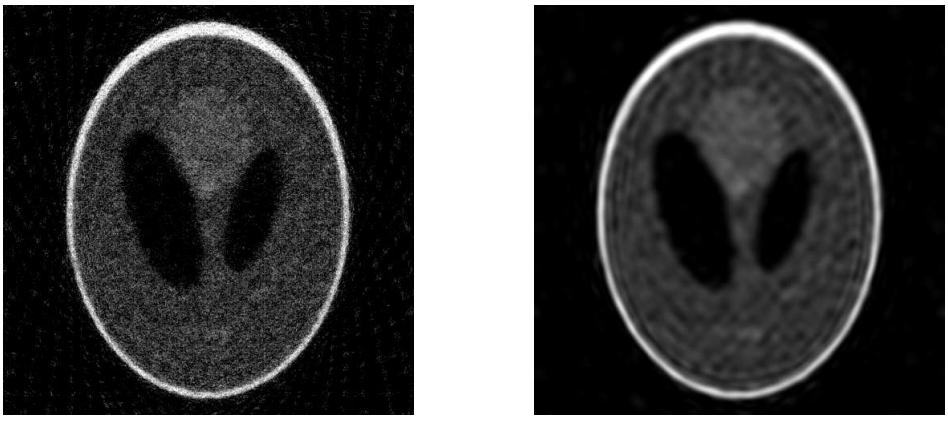
In this final section the fixed point iterations (8.2) are used with both the soft- and firm-thresholding operators (6.14) and (8.3) to reconstruct the CT problem from Chapter 6 for comparison with total variation and wavelet based regularization.

Figures 8.1 and 8.2 show reconstructions with a low and high choice of α using the soft- and firm-thresholding operator respectively. Using the same high $\alpha := 30$ with both methods, the advantage of the firm-thresholding is evident.



(a) $\bar{R}_\alpha(\bar{g}^\bullet)$, $\alpha := 0.001$, $e_\%{\approx}0.15$. (b) $\bar{R}_\alpha(\bar{g}^\bullet)$, $\alpha := 30$, $e_\%{\approx}0.28$.

Figure 8.1: Soft-thresholding, low and high α.



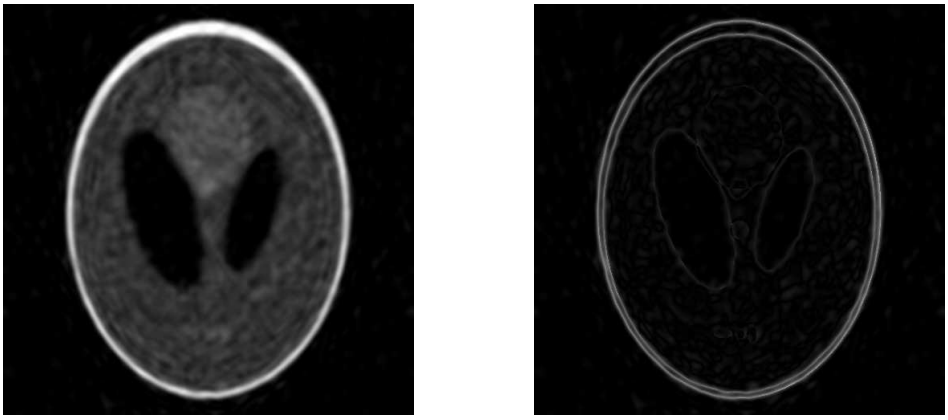
(a) $\bar{R}_\alpha(\bar{g}^\bullet)$, $\alpha := 0.001$, $e_\%{\approx}0.15$. (b) $\bar{R}_\alpha(\bar{g}^\bullet)$, $\alpha := 30$, $e_\%{\approx}0.15$.

Figure 8.2: Firm-thresholding, low and high α.

Choosing $\alpha := 1$ in Figures 8.3 and 8.5 the curvelet based regularization gives rise to reconstructions that give relative errors $e_\%$ that are low. Running the fixed point algorithm for 600 iterations takes roughly twenty minutes on a standard laptop PC from

2005, and finding the best choice of α iteratively is therefore not a good option. As expected the reconstructions are much less pixelated than the wavelet reconstructions due to the higher level of orientation with curvelets. The relative errors are close to the best total variation reconstruction which was ≈ 0.12.

Convergence plots from reconstructions shown in Figures 8.3 and 8.5 are shown in Figures 8.4 and 8.6.



(a) $\bar{R}_\alpha(\bar{g}^\bullet)$. (b) $|\bar{f}^\star - \bar{R}_\alpha(\bar{g}^\bullet)|$.

Figure 8.3: Soft-thresholding, $\alpha := 1$, $e_{\%} \approx 0.15$, $\texttt{PSNR} \approx 35.47\texttt{dB}$.

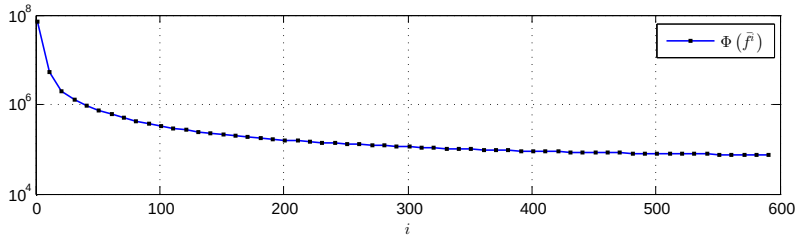
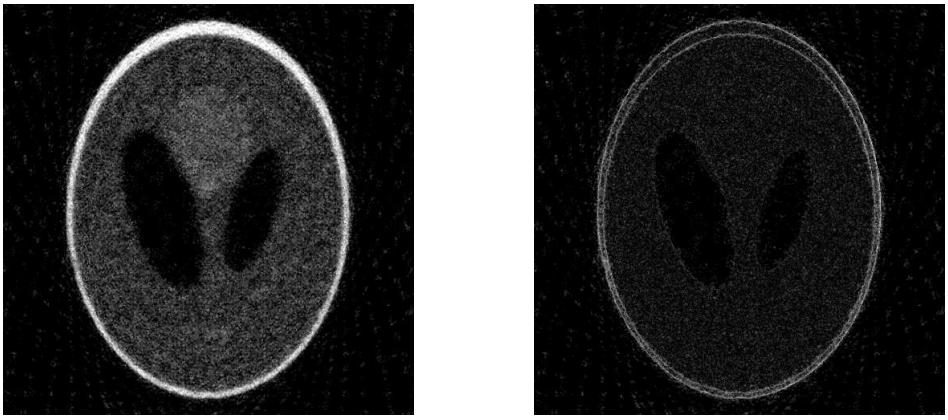


Figure 8.4: Logarithmic convergence plot from solving $\bar{R}_\alpha(\bar{g}^\bullet)$ in Figure 8.3.



(a) $\bar{R}_\alpha(\bar{g}^\bullet)$. (b) $|\bar{f}^\star - \bar{R}_\alpha(\bar{g}^\bullet)|$.

Figure 8.5: Firm-thresholding, $\alpha := 1$, $e_{\%} \approx 0.14$, $\texttt{PSNR} \approx 37.35\texttt{dB}$.

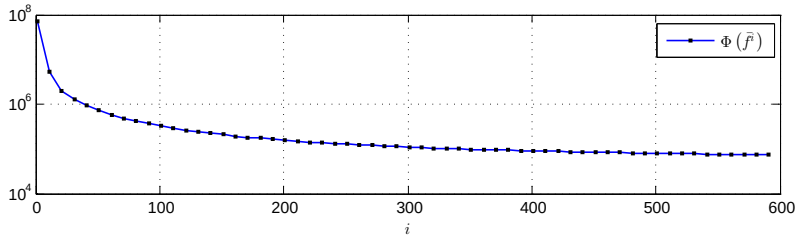


Figure 8.6: Logarithmic convergence plot from solving $\bar{R}_\alpha(\bar{g}^\bullet)$ in Figure 8.5.

Chapter 9

Conclusions

The different regularization methods used in this project have shown to provide ways to deal with ill-posed inverse problems of image de-blurring and computed tomography. Properties of the operator modeling a forward problem determines to a great extend the quality of the reconstructions.

The parameter α permits infinitely many choices of solutions to a regularized inversion. Different suggested methods to automate a choice of α exists with different pros and cons. The subject has not been investigated thoroughly in this project, but it is a matter that has to be considered in any commercial usage of the regularization methods. The complexity and scales of the different methods are also matters worth considering. Problems in 2D or higher easily grow very large and can require major computational power. Not all commercial situations allow a reconstruction to take several minutes.

The choice of regularization method can be based on assumptions or a priori knowledge of a sought solution to compensate for some of the ill-posedness in inverse problems with the intention to improve the quality of a reconstruction. Reconstructions carried out in this project was assumed to be edge-containing images and the different regularization methods has shown to permit a way to include this assumption by using sparsity promoting ℓ^1-norm as a penalty term.

Uniqueness of a regularized solution is not a completely settled subject. Depending on the null-space of the forward operator there might be infinitudes of solutions. The matters of the sparsity promoting norm discussed in Chapter 4 argues that the probability of non-uniqueness is infinitely little, but nevertheless there is a probability. Whether the regularizing in this case provides a well-posed problem can therefore be discussed.

A great challenge in regularization of images is the ability to reconstruct small objects. Small objects are difficult to distinguish from high oscillatory noise which of course is sought removed in a reconstruction. None of the three used methods where found able to reconstruct the small objects of the Shepp-Logan phantom in a convincing way in the specific CT problem.

The well established total variation regularization method promotes sparsity of a sought

solutions gradient. This method gave rise to the best reconstructions in terms of the relative error for both the 1D de-blurring problem and the 2D tomography problem. The total variation was also the easiest method to implement and the most efficient in terms of computation time.

In both wavelet and curvelet based regularization the expansion coefficients of a sought function is assumed sparse which corresponds to assuming an edge containing image. The wavelet based method is applicable to both 1D and 2D problems, but has limited orientation properties in 2D. Curvelets apply to 2D (and 3D) problems and captures much more orientation than wavelets. The edge preservation is evident for both methods in the reconstructions carried out. In terms of the relative error, both the wavelet and curvelet based methods performed close to as well as the total variation method. Subjectively, the qualities of reconstructions using curvelets are arguably better than the wavelet based reconstructions.

Altogether strong arguments for choosing the curvelet based regularization method over total variation regularization has not been established in this project, but the use of curvelets demonstrates great potential for further research and ideas in the field of regularization of edge-containing images.

Appendix A

List of symbols

$\#\{X\}$	The cardinality of X (number of elements in X).
$X^{\perp}$	Orthogonal to X.
$\mathcal{D}(X)$	The domain of X.
$\mathcal{R}(X)$	The range of X.
χ_Ω	The characteristic function defined on Ω to be $\chi_\Omega := 1$ for $x \in \Omega$ and zero otherwise.
$\|\cdot\|_V$	Norm on a vector space V.
$\langle\cdot,\cdot\rangle_V$	Inner product on a vector space V.
$\mathcal{H}$	A Hilbert space.
A^*	The Hilbert adjoint of a bounded linear operator $A\colon\ \mathcal{H}_1 \to \mathcal{H}_2$.
A^T	The transpose of matrix A.
$C^r(\Omega)$	The set of continuous and r times differentiable functions on Ω.
$L^p(\Omega)$	The norm-complete Lebesgue space, normed for $1 \leq p < \infty$ as $\|f\|_{L^p(\Omega)} := (\int_\Omega \|f\|^p dx)^{1/p}$, and defined as $L^p(\Omega) := \{f : \Omega \to \mathbb{C} \mid \int_\Omega \|f(x)\|^p dx < \infty\}$. Hilbert space for the choice $p = 2$ with the inner product $\langle f, g\rangle_{L^2(\Omega)} := \int_\Omega f(x)\overline{g(x)}dx$.
$\ell^p(\mathbb{N})$	The norm-complete discrete Lebesgue space, normed for $1 \leq p < \infty$ as $\|\bar{f}\|_{L^p(\Omega)} := (\sum_{k\in\mathbb{N}} \|\bar{f}_k\|^p)^{1/p}$, and defined as $\ell^p(\mathbb{N}) := \{\bar{f} \mid \sum_{k\in\mathbb{N}} \|\bar{f}_k\|^p < \infty\}$. Hilbert space for the choice $p = 2$ with the inner product $\langle \bar{f}, \bar{g}\rangle_{\ell^2(\mathbb{N})} := \sum_{\mathbb{N}} \bar{f}_k\overline{\bar{g}_k}$.
$\arg\min_f \{\Phi(f)\}$	The element f that minimizes a functional $\Phi(f)$.
$\lfloor\cdot\rfloor$	The floor operator rounding a scalar down to nearest integer.

Appendix B

Fourier transform

The Fourier transform is widely used and has been a very well established tool in engineering and physics for many years. It allows a function to be represented in terms of its frequency spectrum. On $\mathbb{R}$ the Fourier transform $\hat{f}\colon \mathbb{R} \to \mathbb{C}$ of a function $f \in L^1(\mathbb{R})$ is defined for $\omega \in \mathbb{R}$ as

$$\hat{f}(\omega) := \int_{\mathbb{R}} f(x) e^{-2\pi i x\omega} dx,$$

with the inversion

$$f(x) = \int_{\mathbb{R}} \hat{f}(\omega) e^{2\pi i x\omega} d\omega$$

for almost all $x \in \mathbb{R}$. The usual extension of the Fourier transform allows continuous and compactly supported functions to be transformed in $L^2(\Omega)$ as well [10, Th.7.2.2]. The transformed function $\hat{f}$ is complex-valued even though f is real. Usually this is described as $\hat{f}$ containing frequency information about magnitudes in the real part $\Re(\hat{f})$ and phases in the imaginary part $\Im(\hat{f})$. In brief the magnitudes tell *how much* the different frequency components are present and the phases tell *where* they are present. In order to reconstruct a function from its Fourier-transform both informations are needed. And $\mathbb{R}^2$ and for an $f \in L^1(\mathbb{R}^2)$ the Fourier transform $\hat{f}\colon \mathbb{R}^2 \to \mathbb{C}^2$ of $f \in L^1(\mathbb{R}^2)$ is for $(\omega_1, \omega_2) \in \mathbb{R}^2$ defined as

$$\hat{f}(\omega_1, \omega_2) := \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i2\pi(\omega_1 x + \omega_2 y)} dx\, dy,$$

with the inversion

$$f(x, y) = \int_{\mathbb{R}} \int_{\mathbb{R}} \hat{f}(\omega_x, \omega_y) e^{i2\pi(x\omega_x + y\omega_y)} d\omega_x\, d\omega_y$$

for almost all $(x, y) \in \mathbb{R}^2$.

Notating the Fourier transform as an operator $\mathcal{F}\colon L^1(\mathbb{R}^d) \to \mathcal{R}(\mathcal{F})$, the important translation and dilation relations can be stated. For $\bar{a} \in \mathbb{R}^d$ the translation operator $T_{\bar{a}}$ is

defined for $f \in L^2(\mathbb{R}^d)$ on $\bar{x} \in \mathbb{R}^d$ as $(T_{\bar{a}} f)(\bar{x}) := f(\bar{x} - \bar{a})$. The translation relation [10, Th.7.1.2]

$$(\mathcal{F} T_{\bar{a}} f)(\bar{\omega}) = \hat{f}(\bar{\omega}) e^{-2\pi i \bar{a} \cdot \bar{\omega}}$$

says that translation of a function f changes phases in the Fourier domain. With dilation constants $c_1, c_2, ..., c_d > 0$ and the matrices $C := \texttt{diag}(c_k)$ and $C^{-1} := \texttt{diag}(c_k^{-1})$, the dilation operator D_C defined as $(D_C f)(\bar{x}) := f(C\bar{x})$ is related to the Fouier transform by

$$(\mathcal{F} D_{C^{-1}} f)(\bar{\omega}) = |\prod_{k=1}^{d} c_k| \hat{f}(C\bar{\omega}).$$

This relation says that scaling f to smaller sizes results in higher frequencies in $\hat{f}$ (and vice versa).

Appendix C

CT reconstructions (full size)

C.1 Total variation

C.1.1 Reconstruction from Figure 6.9(b)

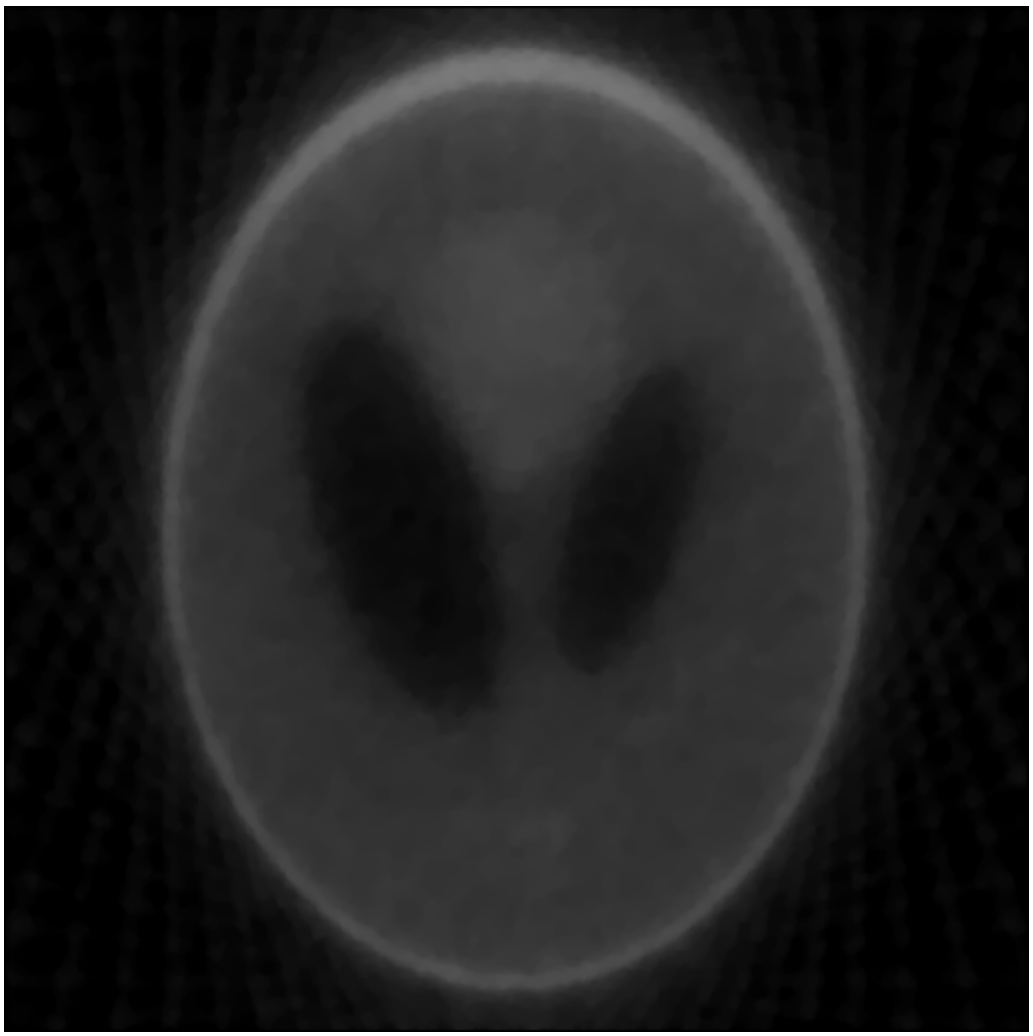


Figure C.1: Total variation, $\alpha := 20$, $e_{\%} \approx 0.34$.

C.1.2 Reconstruction from Figure 6.10(a)

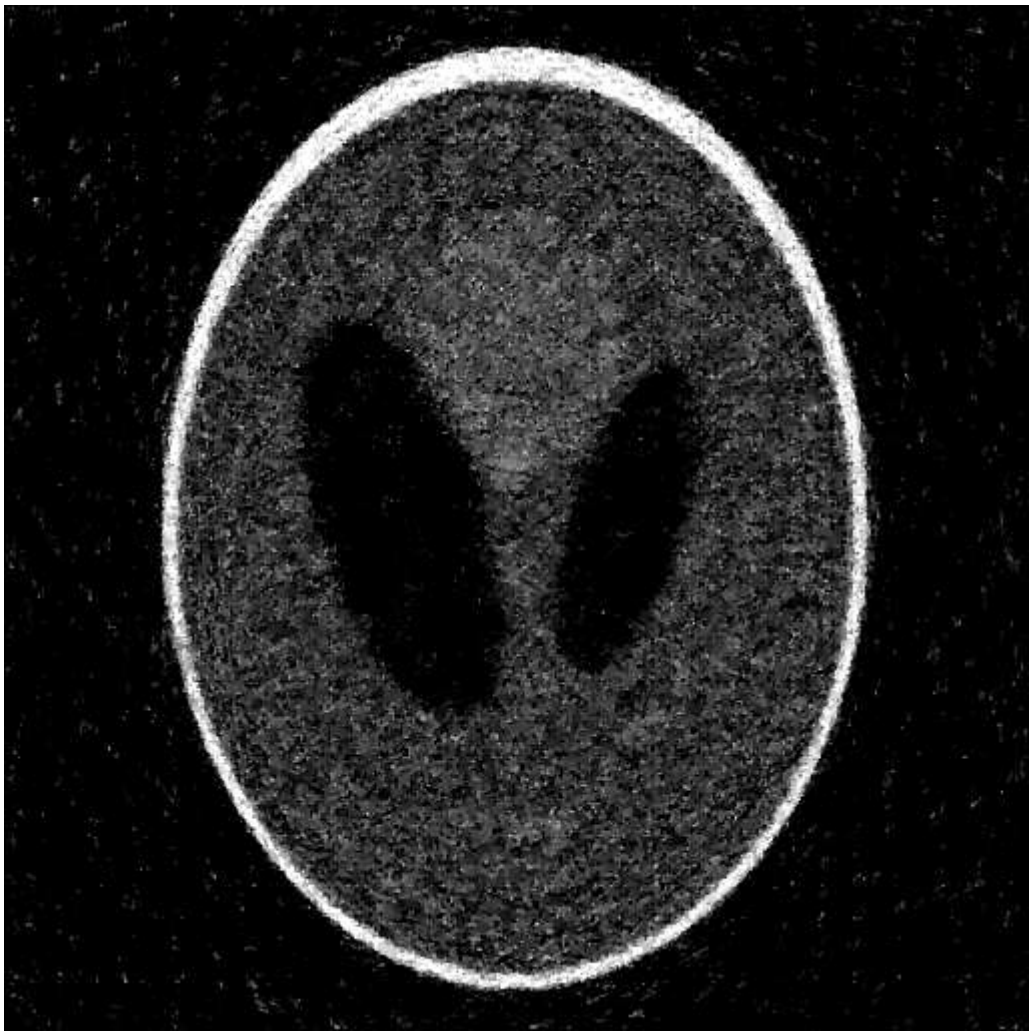


Figure C.2: Total variation, $\alpha := 1.48$, $e_{\%} \approx 0.12$.

C.2 Wavelet based regularization

C.2.1 Reconstruction from Figure 6.12(b)

Figure C.3: Haar wavelet, $\alpha := 20$, $e_{\%} \approx 0.22$.

C.2.2 Reconstruction from Figure 6.13(a)

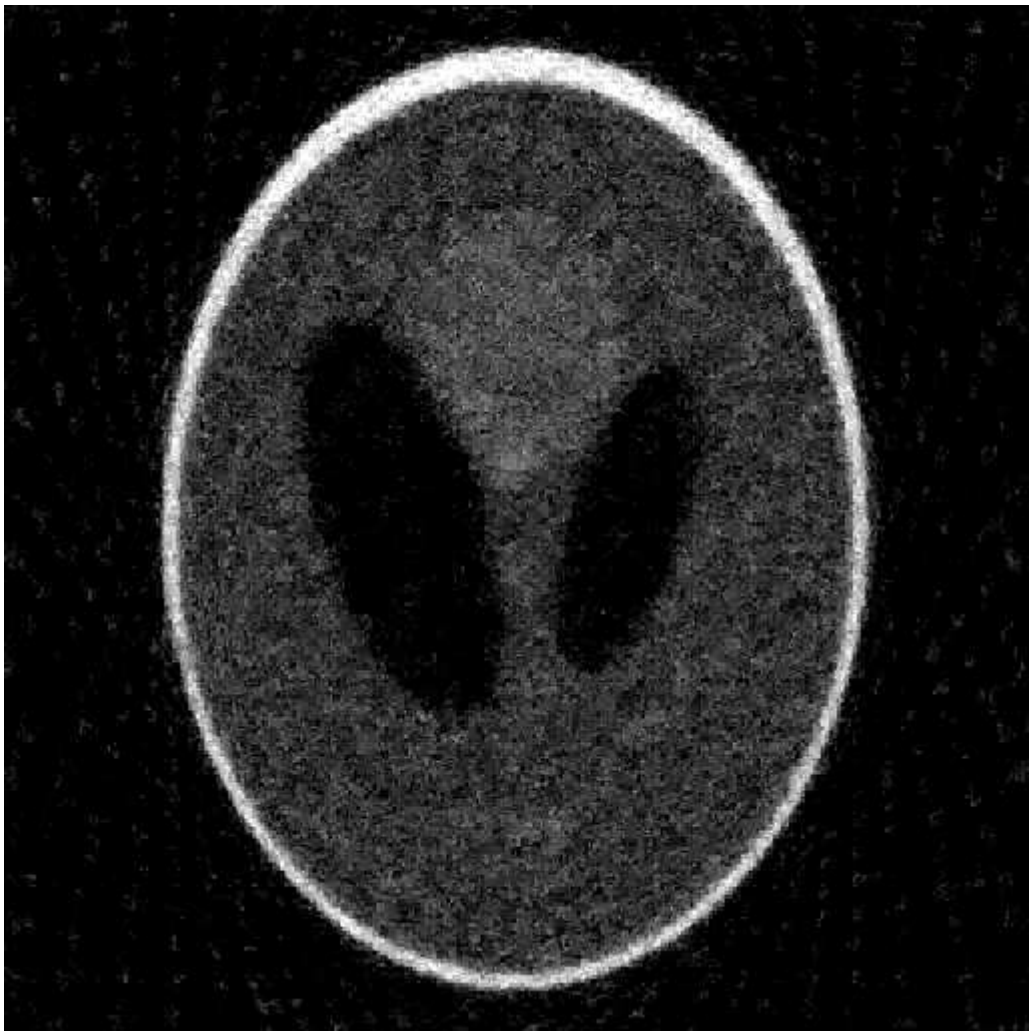


Figure C.4: Haar wavelet, $\alpha := 0.5$, $e_{\%} \approx 0.14$.

C.3 Curvelet based regularization

C.3.1 Reconstruction from Figure 8.1(b)

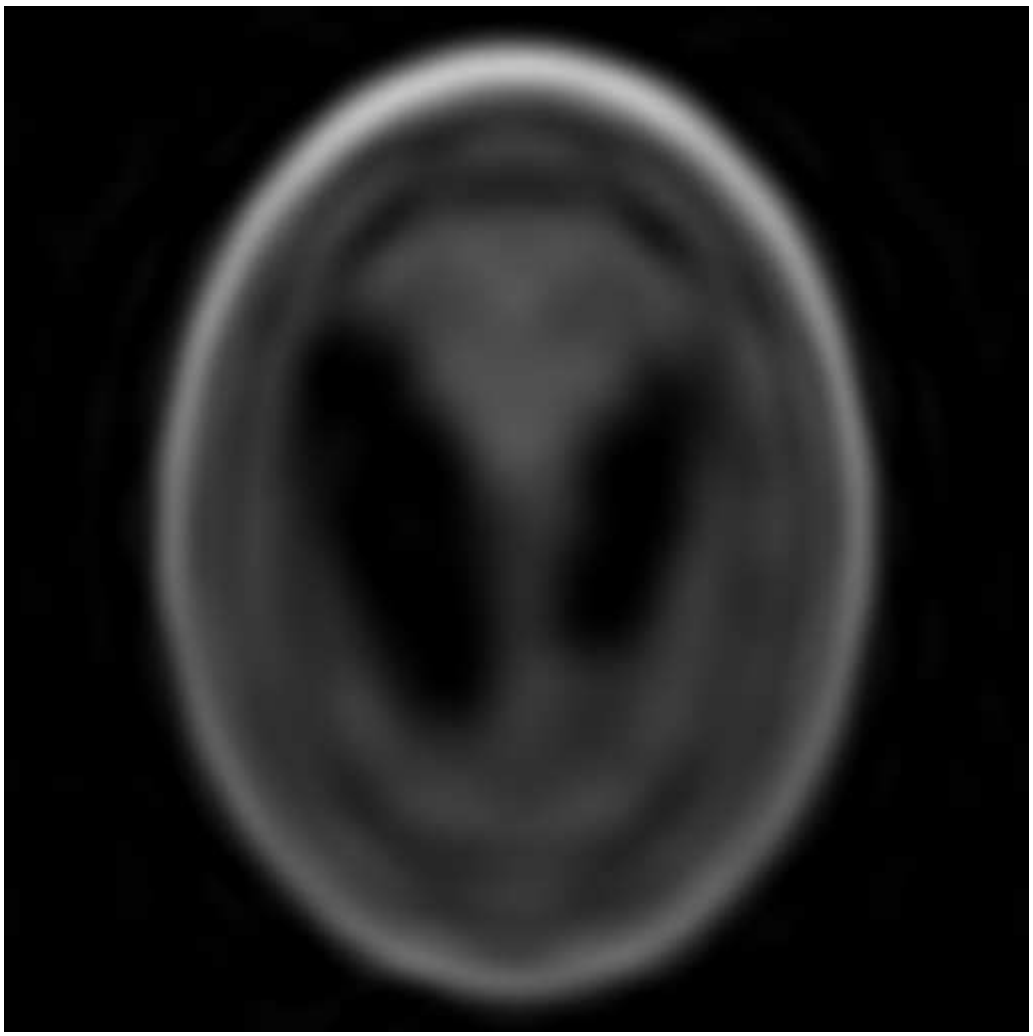


Figure C.5: Soft-thresholding, $\alpha := 30$, $e_{\%} \approx 0.28$.

C.3.2 Reconstruction from Figure 8.2(b)

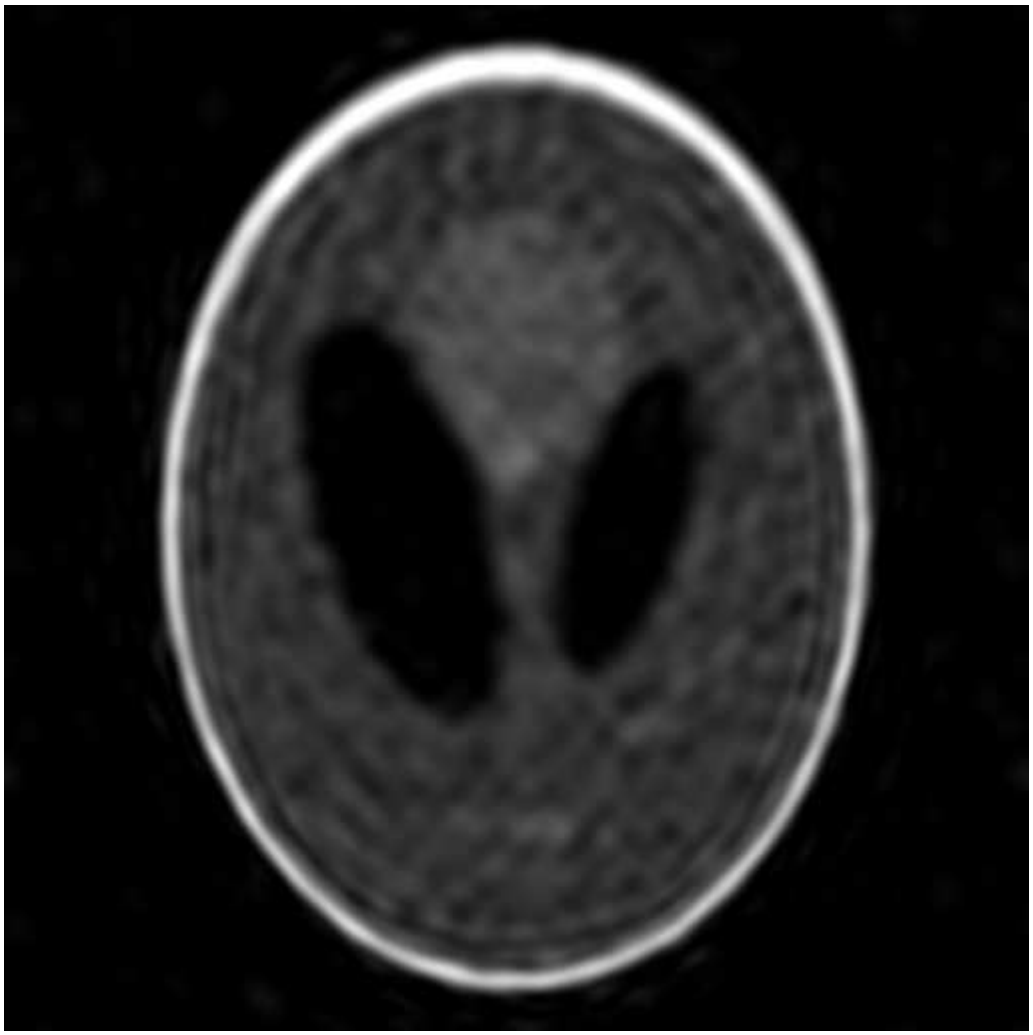


Figure C.6: Firm-thresholding, $\alpha := 30$, $e_{\%} \approx 0.15$.

C.3.3 Reconstruction from Figure 8.3(a)

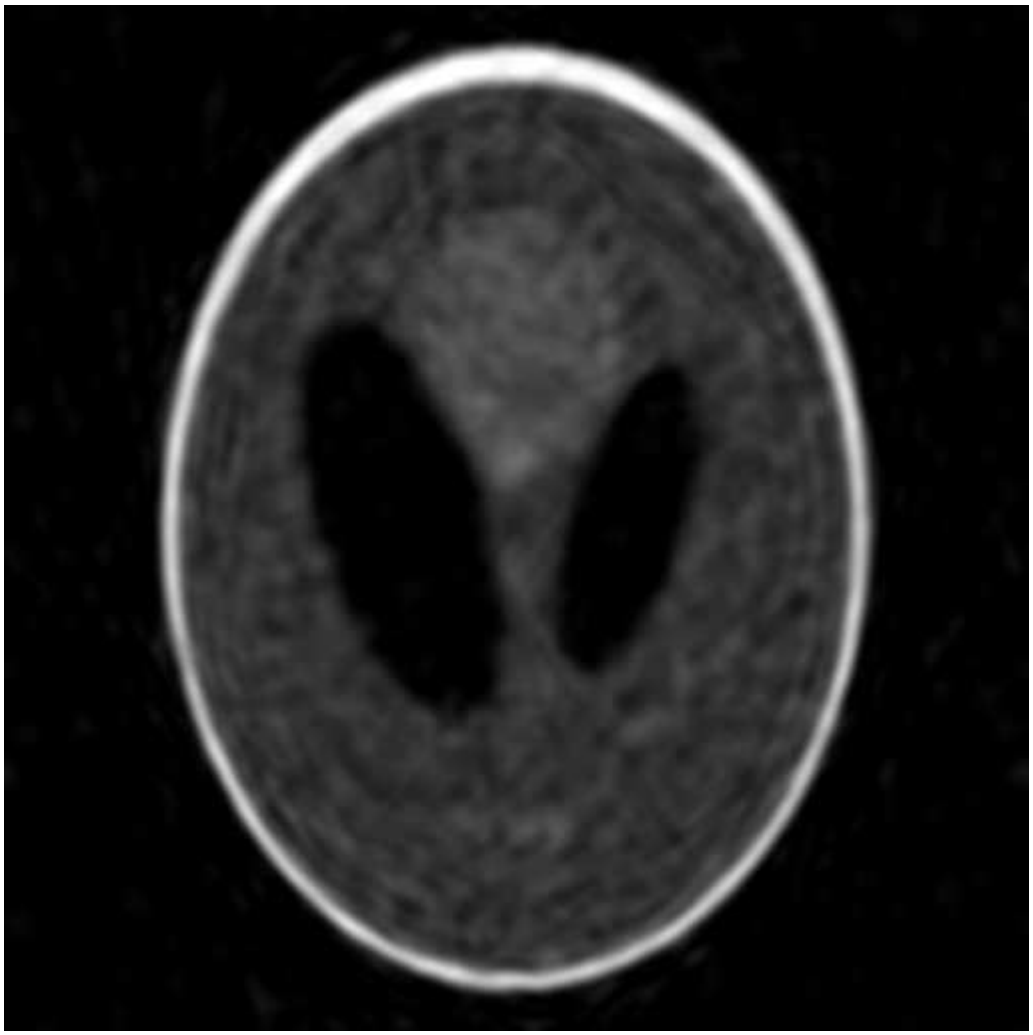


Figure C.7: Soft-thresholding, $\alpha := 1$, $e_{\%} \approx 0.15$.

C.3.4 Reconstruction from Figure 8.5(a)

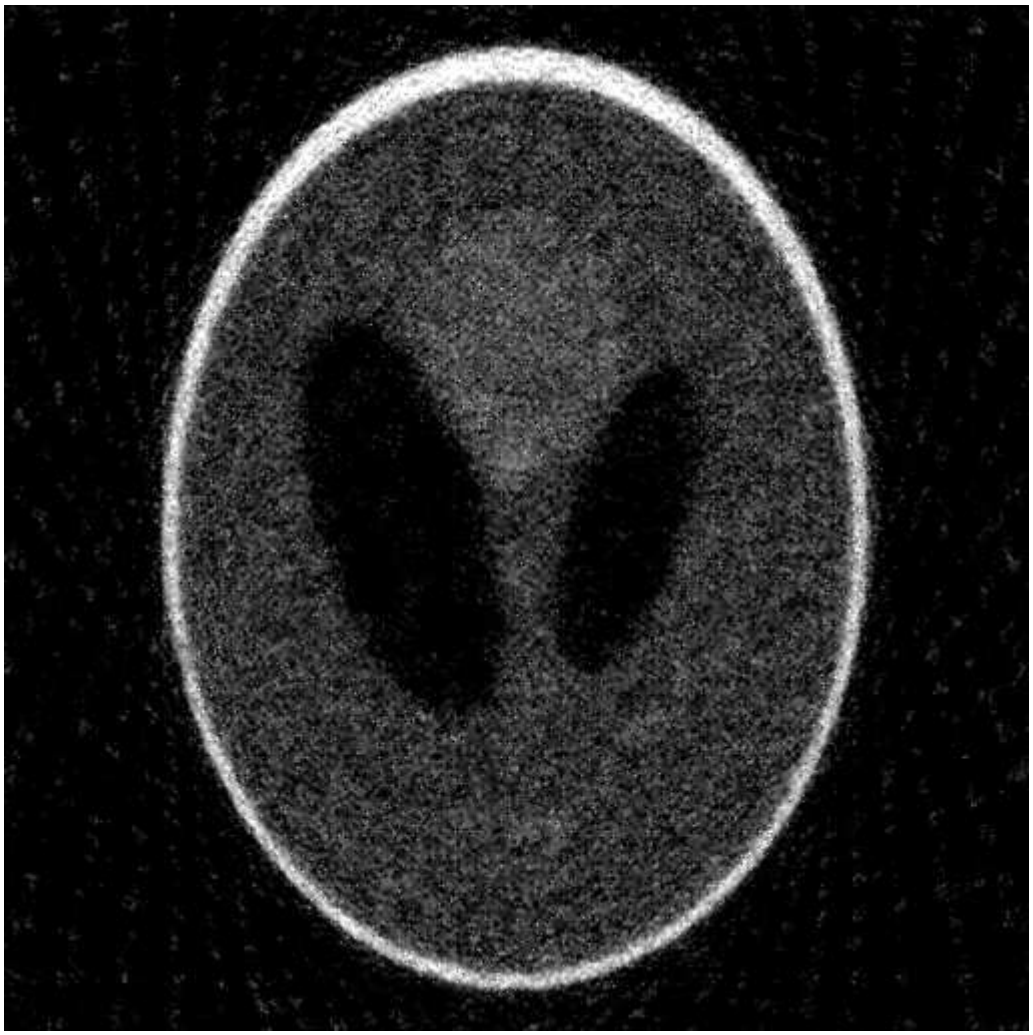


Figure C.8: Firm-thresholding, $\alpha := 1$, $e_{\%} \approx 0.14$.

Appendix D

MATLAB code

D.1 MATLAB code (deconvolution)

D.1.1 Generate deconvolution testfunction (5.11)

```
function [f] = testfunction(x)

% Creates 1D piecewise continuous 1-periodic function f(x)
% suitable for testing deconvolution problems.
%
%   [f] = testfunction(x)
%
% Input:
%   x           Scalar or vector of one or more x-values.
%
% Output:
%   f           The function value at each x-value.
%

% Jacob Larsen
% May/June, 2013, DTU Compute.

f = zeros(1,length(x)); % Reset/preallocate variable for f.

for i=1:length(x) % Loop over number of x-values in input.

    xi = x(i); % Current iteration value.

    % Make function 1-periodic.
    if (xi>0)
        xi = xi-floor(xi);
    elseif (xi<0)
        xi = xi+abs(floor(xi));
    end
    xi=abs(xi);

    % Generate output f-value dependent on x-value.
    if (xi>0.1) && (xi<=0.2)
        f(i) = 1;
    elseif (xi>0.2) && (xi<=0.3)
        f(i) = 0.3;
```

```
    elseif (xi>0.35) && (xi<=0.4)
        f(i) = 0.8;
    elseif (xi>0.5) && (xi<=0.6)
        f(i) = 9*(xi-0.5);
    elseif (xi>0.7) && (xi<=0.9)
        f(i) = 0.6*sin(pi*(xi-0.7)/0.2);
    else
        f(i) = 0;
    end

end
```

D.1.2 Generate point spread function (5.3)

```
function [psi] = psifunction(x,a)

% Creates 1D 1-periodic point spread function psi(x)
% with spreading constant a.
%
%   [psi] = psifunction(x,a)
%
% Input:
%   x           Scalar or vector of one or more x-values.
%   a           Spreading constant (0<a<0.5).
%
% Output:
%   psi         The function value at each x-value.
%

% Jacob Larsen
% May/June, 2013, DTU Compute.


psi = zeros(1,length(x)); % Reset/preallocate variable for psi.

Ca = 15/(16*(a^5)); % Analytic expression for normalization constant.
% Define one period of function supported on -a<=x<=a.
psi0 = @(x) Ca*(x+a).^2 .* (x-a).^2;

for i=1:length(x) % Loop over number of x-values in input.

    xi = x(i); % Current iteration value.

    % Make function 1-periodic.
    if (xi>0)
        xi = xi-floor(xi);
    elseif (xi<0)
        xi = xi-ceil(xi);
    end

    % Generate output psi-value dependent on x-value.
    if (xi>=-a) && (xi<=a)
        psi(i) = psi0(xi);
    elseif (xi>=1-a) && (xi<=1+a)
        psi(i) = psi0(xi-1);
    elseif (xi>=-1-a) && (xi<=-1+a)
        psi(i) = psi0(xi+1);
```

```
    else
        psi(i) = 0;
    end

end
```

D.1.3 Generate simulated measurements for a deconvolution problem

```
function [m,sigma] = simulate_convolution_data(nin,nout,a,noisepct)

% Creates simulated data for a 1D deconvolution problem. The
% measurement vector m is made of a piecewise continuous function
% that is convolved with a point spread function (PSF). Then it is
% interpolated to a smaller grid and Gaussian noise is added.
%
%   [m] = testfunction(nin,nout,a,noisepct)
%
% Input:
%   nin           Size of original generated vector m.
%   nout          Size of output vector m after interpolation.
%   a             Spreading constant in PSF
%   noisepct      Std. of noise added to m is noisepct/100*max(m)
%
% Output:
%   m             Output simulated measurement vector.
%   sigma         Std. of the noise added to m.

% Jacob Larsen
% May/June, 2013, DTU Compute.

dx = 1/nin;
dxout = 1/nout;
fn = testfunction(0:dx:1-dx); % Generate piecewise continuous testfunction.
M = zeros(nin,1); % Reset/preallocate measurement vector on original grid.

% Convolution, integrate once for each value of tau.
q = 1;
for t=0:dx:1-dx
    tau = t;
    Q = @(x)(testfunction(tau-x).*psifunction(x,a)); % Integrand.
    M(q) = integral(Q,0,1); % Add to M
    q=q+1;
end

% Interpolate to new grid
xold = linspace(0,1-dx,nin);
xnew = linspace(0,1-dxout,nout);
m = interp1(xold,M,xnew,'spline');

% Add noise
sigma = noisepct/100*max(m);
noise = sigma.*randn(1,nout);
m = m+noise;
```

D.1.4 Generate circular convolution matrix A from (5.9)

```
function [A] = circ_conv_matrix(m,n,p)

% Generates a circular convolution matrix of size m x n that
% acts as a convolution operator between the given point spread function
% p (size m), and a function (size n) that it is acting on.
%
%   [A] = circ_conv_matrix(m,n,p)
%
% Input:
%   m           Size of vector p.
%   n           Size of vector to act on.
%   p           Vector with point spread function
%
% Output:
%   A           The circular convolution matrix
%

% Jacob Larsen
% May/June, 2013, DTU Compute.

dx = 1/n;
v = (length(p)-1)/2; % Support of p.
A = zeros(m,n); % Reset/preallocate A.
prot = p(end:-1:1); % Backwards version of p.

% Generate A row-wise by circular shifting p.
pnow = zeros(n,1);
pnow(1:length(prot)) = prot;
pnow = circshift(pnow,-v);
for i=1:m
    A(i,:) = pnow;
    pnow = circshift(pnow,1);
end

% Multiply with discrete factor.
A = dx*A;
```

D.1.5 Solve 1D inverse problem using Total Variation Regularization

```
function [f] = tvreg1d(m,A,alpha)

% Total Variation Regularization of a 1D linear inverse
% problem on the form m=Af, using quadratic programming.
%
%
%   [f] = tvreg1d(m,A,alpha)
%
% Input:
%   m             Measurement vector to be regularized.
%   A             Forward operator.
%   alpha         Regularization parameter.
%
% Output:
%   f             The regularized solution.
%

% Jacob Larsen
```

```
% May/June, 2013, DTU Compute.



% solve argmin { (1/2)x^T.Q.x + c^T.x }

n = length(m);

% Reset/preallocate Q and c.
Q = zeros(n*3,n*3);
c = zeros(n*3,1);

% Set Q and C as defined.
Q(1:n,1:n) = 2*(A'*A);
c(1:n) = -2*A'*m';
c(n+1:end) = alpha;

% Define K which is the difference matrix L extended with
% the unit matrix I twice.
K = zeros(n*3,n*3);
for i=1:n-1
    K(i,i) = -1;
    K(i,i+1) = 1;
    K(i,n+i) = -1;
    K(i,2*n+i) = 1;
end
K(n,1) = 1; K(n,n)=-1;
K(n,n+i) = -1;
K(n,2*n+i) = 1;

% Define lower bound for x
lb = zeros(3*n,1);
lb(1:n) = -inf;

% Solve quadradic problem
opts = optimset('Algorithm','interior-point-convex','MaxIter',1000);
x = quadprog(Q,c,[],[],K,zeros(n*3,1),lb,[],[],opts);
f = x(1:n);
```

D.1.6 Generate wavelet decomposition matrix (4.18)

```
function [Binv] = wavedecomp(n,wavename)

% Generate wavelet decomposition (nXn) matrix Binv using MATLABs
% wavedec() command on each row.
%
%   [Binv] = wavedecomp(N,wavename)
%
% Input:
%   n             Size of matrix.
%   wavename      Name of wavelet in MATLABs toolbox eg. ('haar','db2').
%
% Output:
%   Binv          The nXn wavelet decomposition matrix.
%

% Jacob Larsen
```

```
% May/June, 2013, DTU Compute.


dwtmode('per'); % Assume periodic functions.
Binv = spalloc(n,n,50*n); % Preallocate sparse matrix.

% Loop over each row in matrix.
for i=1:n
    % Genereate dirac-delta vector.
    z = zeros(1,n);
    z(i) = 1;
    % Compute wavelets on dirac delta vector
    [C,L] = wavedec(z,ceil(log2(n)),wavename);
    % Add to matrix as one row
    Binv(:,i) = C;
end
```

D.1.7 Solve 1D inverse problem using wavelet based regularization

```
function [f,C] = wavereg1d(m,A,sigma,alpha,wavename)

% Wavelet based regularization of a 1D linear inverse
% problem on the form m=Af, using quadratic programming.
%
%   [f] = wavereg1d(m,sigma,alpha,wavename)
%
% Input:
%   m             Measurement vector to be regularized.
%   A             Forward operator.
%   sigma         Standard deviation of noise.
%   alpha         Regularization parameter.
%   wavename      Name of wavelet in MATLABs toolbox eg. ('haar','db2').
%
% Output:
%   f             The regularized solution.
%

% Jacob Larsen
% May/June, 2013, DTU Compute.

n = length(m);

% Reset/preallocate Q and c.
Q = zeros(n*3,n*3);
c = ones(n*3,1);

% Set Q and C as defined.
Q(1:n,1:n) = (1/(sigma^2))*(A'*A);
c(1:n) = -(1/(sigma^2))*A'*m;
c(n+1:end) = alpha;

% Compute weight matrix with 2^(j/2) in the diagonal.
w = zeros(n,n);
w = [1];
for j=0:ceil(log2(n))-1
    w = [w; 2^(j/2)*ones(2^j,1)];
end
```

```
W = diag(w);
clear w;

% Generate wavelet decomposition matrix.
Binv = wavedecomp(n,wavename);

% Set equality constraints and lower bound.
Aeq = [W*Binv -eye(n,n) eye(n,n)];
Beq = sparse(zeros(n,1));
lb = sparse([-inf*ones(n,1); zeros(2*n,1)]);

% Call quadprog()
opts = optimset('Algorithm','interior-point-convex','MaxIter',1000);
x = quadprog(Q,c,[],[],Aeq,Beq,lb,[],[],opts);

f = x(1:n);
C = x(n+1:2*n) - x(2*n+1:3*n);
```

D.2 MATLAB code (CT)

D.2.1 Generate simulated measurements for a CT problem

```
function [m,sigma] = simulate_ct_data(n,thetain,thetaout,noisepct)

% Simulates measured data for a CT inverse problem by generating
% a nxn Shepp-Logan phantom, Radon transforming it, interpolating
% to a smaller grid, and adding Gaussian noise.
%
%
%   [m] = simulate_ct_data(n,thetain,thetaout,noisepct)
%
% Input:
%   n             Size of the nxn pixels phantom image.
%   thetain       Original theta-grid to Radon transform on, e.g 0:1:179.
%   thetaout      Grid on measurements after interpolation, e.g 0:3:179.
%   noisepct      Std. of noise added to m is noisepct/100*max(m).
%
% Output:
%   m             Output simulated measurement vector.
%   sigma         Std. of the noise added to m.
%

% Jacob Larsen
% May/June, 2013, DTU Compute.

% Generate nxn Shepp-Logan phantom.
P = phantom(n);
% Normalize image so smallest color intensity is 0 and largest is 255.
P = 255/max(max(P))*P;
% Convert to uint8 to make sure it is only whole integer values.
P = uint8(P);
% Convert back to double with values between 0 and 1.
P = double(P)/255;

% Radon transform on P.
[mdummy,Xp]=radon(P,thetain);

A = paralleltomo(n,thetain);
mt = A*P(:);
mt = reshape(mt,length(mt)/length(thetain),length(thetain));
z = size(mdummy,1)-size(mt,1);
m = zeros(size(mdummy));
m(3:end-2,:) = mt;

% Interpolate to smaller theta-grid.
[toldx,toldy] = meshgrid(thetain,1:1:size(m,1));
[tnewx,tnewy] = meshgrid(thetaout,1:1:size(m,1));
m = interp2(toldx,toldy,m,tnewx,tnewy,'spline');

% Add noise.
sigma = noisepct/100*max(m(:));
m = m+ randn(size(m))*sigma^2;
```

D.2.2 Solve 2D inverse CT problem using Total Variation Regularization

```
function [f,convergence] = tvreg2d(n,m,theta,alpha,beta,lambda0,f0,ite)

% Total Variation Regularization of a 2D linear inverse CT
% problem on the form m=Af, using Barzilai-Borwein iterations.
%
%
%   [f,convergence] = tvreg2d(A,m,alpha,beta,lambda0,f0,ite)
%
% Input:
%   n             Size of nxn image.
%   m             Measurement vector to be regularized.
%   theta         CT angles (e.g. 0:1:179).
%   alpha         Regularization parameter.
%   beta          Smoothening parameter, small e.g 10^(-5).
%   lambda0       First step length, small e.g 10^(-5).
%   f0            Starting guess for a solution, e.g a zero vector.
%   ite           Number of iterations to run.
%
% Output:
%   f             The regularized solution.
%   convergence   Array of values of the objective functional for
%                 each iteration.
%

% Jacob Larsen
% May/June, 2013, DTU Compute.

% Set inital step length
lambda = lambda0;

convergence = [];

% Make 3 peroids of f to calculate gradient term.
ff = reshape(f0,n,n);
ff = repmat(ff,1,3);
f = ff(:);

% Compute gradient of penalty-term element wise.
grad = zeros(n^2,1);
for j = n^2+1:2*n^2
    dif_penalty = ( 2*f(j)-f(j+n)-f(j+1) ) / ...
        (sqrt( (f(j+n)-f(j))^2 + (f(j+1)-f(j))^2  ) + beta )   ...
        + ( f(j)-f(j-n) ) / ...
        ( sqrt ( (f(j)-f(j-n))^2 + (f(j-n+1)-f(j-n))^2 ) + beta ) ...
        + ( f(j)-f(j-1) ) / ...
        ( sqrt( (f(j)-f(j-1))^2 + (f(j+n-1)-f(j-1))^2 ) + beta  ) ;
    gradnew(j-n^2) = alpha*dif_penalty;
end

% Resize back to one period of f and set boundary
% conditions.
f = f(1:n^2);
f = reshape(f,n,n);
```

```
f(:,1) = 0; f(:,end) = 0;
f(1,:) = 0; f(end,:) = 0;
f = f(:);

% Gradient of entire functional to be minimized.
%grad = alpha*grad + 2*A'*(A*f-m);
ff = reshape(f,n,n);
mr = radon(ff,theta);

% Next step
grad2 = 2*2*iradon(mr-m,theta,'Linear','None',1,n);
grad2 = grad2(:);
grad = grad + grad2;
clear grad2;
fold = f;

P = phantom(512);
P(P<0)=0;

% Look ite times
for it=1:ite

    % Next f (next iteration).
    f = fold-lambda*grad;

    % Set f elementwise to feasible region (f>=0).
    f(f<0)=0;

    % Make 3 periods of f.
    ff = reshape(f,n,n);
    ff = repmat(ff,1,3);
    f = ff(:);

    % Compute gradient of penalty term using new f.
    gradnew = zeros(n^2,1);
    for j = n^2+1:2*n^2
        dif_penalty = ( 2*f(j)-f(j+n)-f(j+1) ) / ...
            (sqrt( (f(j+n)-f(j))^2 + (f(j+1)-f(j))^2  ) + beta )   ...
            + ( f(j)-f(j-n) ) / ...
            ( sqrt ( (f(j)-f(j-n))^2 + (f(j-n+1)-f(j-n))^2 ) + beta ) ...
            + ( f(j)-f(j-1) ) / ...
            ( sqrt( (f(j)-f(j-1))^2 + (f(j+n-1)-f(j-1))^2 ) + beta  ) ;
        gradnew(j-n^2) = alpha*dif_penalty;
    end

    % Resize f to one period.
    f = f(1:n^2);
    f = reshape(f,n,n);
    f(:,1) = 0; f(:,end) = 0;
    f(1,:) = 0; f(end,:) = 0;
    f = f(:);

    % Gradient of entire functional to be minimized.
    ff = reshape(f,n,n);
    mr = radon(ff,theta);

    % Next step
    grad2 = 2*2*iradon(mr-m,theta,'Linear','None',1,n);
```

```
    grad2 = grad2(:);
    gradnew = gradnew + grad2;
    clear grad2;

    % Compute step-length lambda for next iteration.
    lambda = ((f-fold)' * (f-fold)) / ((f-fold)' * (gradnew-grad) );

    % Save objective functional value for current iteration.
    convergence = [convergence; norm(mr-m,2)^2 + alpha*norm(diff(ff),1)];

    % Set new iteration.
    grad = gradnew;
    fold = f;
end
```

D.2.3 Solve 2D inverse CT problem using Wavelet Regularization

```
function [f,C,convergence] = wavereg2d(n,m,theta,alpha,lambda ...
    ,f0,ite,wavename,sigma)

% Wavelet Based Regularization of a 2D linear inverse CT
% problem on the form m=Af, using soft-thresholding iterations.
%
%
%   [f,C,convergence] = tvreg2d(A,m,alpha,beta,lambda0,f0,ite)
%
% Input:
%   n             Size of nxn image.
%   m             Measurement vector to be regularized.
%   theta         CT angles (e.g. 0:1:179).
%   alpha         Regularization parameter.
%   lambda        Step-size, small e.g 10^(-5).
%   f0            Starting guess for a solution, e.g a zero vector.
%   ite           Number of iterations to run.
%   wavename      Name of wavelet to use (eg. 'haar' or 'db2').
%   sigma         Std. of noiselevel.
%
% Output:
%   f             The regularized solution.
%   C             Wavelet coefficients of f.
%   convergence   Array of values of the objective functional for
%                 each iteration.
%

% Jacob Larsen
% May/June, 2013, DTU Compute.



convergence = [];
% Set wavelet settings: Periodic
dwtmode('per');

% Number of max wavelet scales j.
J = wmaxlev([n n], wavename);

% Coefficients of starting guess.
```

```
[COld,S] = wavedec2(f0,J,wavename);

% Generate weight vector.
w = [ones(S(1,1)*S(1,2),1)];
for j=2:length(S)-1
    le = S(j);
    tau = (2^(j/2));
    w = [w; tau.*ones(3*(le^2),1)];
end
w = (alpha).*w;

% Set first f to starting guess.
f = f0;

for it=1:ite

    % One iteration step.
    fold = waverec2(COld,S,wavename);
    mr = radon(fold,theta);
    BP = 2 * iradon((m-mr),theta,'Linear','None',1,n);
    ftemp = fold + lambda*BP;
    ftemp(ftemp<0)=0;
    [CNew,S] = wavedec2(ftemp,J,wavename);

    % Run soft-thresholding.
    for j=1:length(CNew)
        CNew(j) = max(0, abs(CNew(j))-(lambda/(sigma^2))*w(j) ) ...
            * sign(CNew(j));
    end

    % Set new coefficients
    COld = CNew;

    % Save objective functional value for current iteration.
    convergence = [convergence; norm(mr-m,2)^2 + norm(w'.*CNew,1)];
    it

end

% Output
C = CNew;
f = waverec2(C,S,wavename);
```

D.3 MATLAB code (curvelets)

D.3.1 Solve 2D inverse CT problem using Curvelet Regularization with Soft Thresholding Algorithm

```
function [f,C,convergence] = curvereg2d(n,m,theta,alpha,lambda ...
    ,f0,ite,sigma)

% Curvelet Based Regularization of a 2D linear inverse CT
% problem on the form m=Af, using soft-thresholding iterations.
%
% REQUIRES Curvelab-2.1.3 (download from http://www.curvelet.org)
%
%
%   [f,C,convergence] = tvreg2d(A,m,alpha,beta,lambda0,f0,ite)
%
% Input:
%   n             Size of nxn image.
%   m             Measurement vector to be regularized.
%   theta         CT angles (e.g. 0:1:179).
%   alpha         Regularization parameter.
%   lambda        Step-size, small e.g 10^(-5).
%   f0            Starting guess for a solution, e.g a zero vector.
%   ite           Number of iterations to run.
%   sigma         Std. of Gaussian noise in m.

%
% Output:
%   f             The regularized solution.
%   C             Curvelet coefficients of f.
%   convergence   Array of values of the objective functional for
%                 each iteration.
%

% Jacob Larsen
% May/June, 2013, DTU Compute.

convergence = [];
is_real = 1;

% Coefficients of starting guess.
COld = fdct_wrapping(f0, is_real);

% Multiply with step-length.
alpha = alpha*lambda;
rho = rho*lambda;

% Number of scales j=1,2,...,J.
J = length(COld);

% Set first f to starting guess.
fold = f0;

% Loop for each iteration
for it=1:ite
```

```
    % One iteration step.
    mr = radon(fold,theta);
    BP = 2 * iradon((m-mr),theta,'Linear','None',1,n);
    ftemp = fold + lambda*BP;
    ftemp(ftemp<0)=0;
    CNew = fdct_wrapping(ftemp, is_real);

    cc = [];

     %Run soft-thresholding.
        for scale=1:J
            for rot=1:length(CNew{scale})
                tau = alpha*2^(scale-length(CNew))* ...
                sigma*sqrt(2*log(length(CNew{scale}{rot}(:))) );
                CNew{scale}{rot}(:) = max(0, abs(CNew{scale}{rot}(:)) ...
                 - tau ) .* sign(CNew{scale}{rot}(:));
                cc = [cc; CNew{scale}{rot}(:)];
            end
        end


    % Reconstruction of f from coefficients.
    fnew = ifdct_wrapping(CNew, is_real);
    fold = fnew;

    % Save objective functional value for current iteration.
    convergence = [convergence; norm(mr-m,2)^2 + norm(cc,1)];
    if it >= 2
        % Output convergence during iterations
        it
        convergence(end)-convergence(end-1)
    end

end

% Output
C = CNew;
f = ifdct_wrapping(C, is_real);
```

D.3.2 Solve 2D inverse CT problem using Curvelet Regularization with Firm Thresholding Algorithm

```
function [f,C,convergence] = curvereg2d(n,m,theta,alpha,lambda ...
    ,f0,ite,sigma,rhofact)

% Curvelet Based Regularization of a 2D linear inverse CT
% problem on the form m=Af, using firm-thresholding iterations.
%
% REQUIRES Curvelab-2.1.3 (download from http://www.curvelet.org)
%
%
%   [f,C,convergence] = tvreg2d(A,m,alpha,beta,lambda0,f0,ite)
%
% Input:
%   n             Size of nxn image.
%   m             Measurement vector to be regularized.
%   theta         CT angles (e.g. 0:1:179).
```

```
%   alpha          Regularization parameter.
%   lambda         Step-size, small e.g 10^(-5).
%   f0             Starting guess for a solution, e.g a zero vector.
%   ite            Number of iterations to run.
%   sigma          Std. of Gaussian noise in m.

%
% Output:
%   f              The regularized solution.
%   C              Curvelet coefficients of f.
%   convergence    Array of values of the objective functional for
%                  each iteration.
%

% Jacob Larsen
% May/June, 2013, DTU Compute.

convergence = [];
is_real = 1;

% Coefficients of starting guess.
COld = fdct_wrapping(f0, is_real);

% Multiply with step-length.
alpha = alpha*lambda;

% Number of scales j=1,2,...,J.
J = length(COld);

% Set first f to starting guess.
fold = f0;

% Loop for each iteration
for it=1:ite

    % One iteration step.
    mr = radon(fold,theta);
    BP = 2 * iradon((m-mr),theta,'Linear','None',1,n);
    ftemp = fold + lambda*BP;
    ftemp(ftemp<0)=0;
    CNew = fdct_wrapping(ftemp, is_real);

    cc = [];

    % Run firm-thresholding.
    for scale=1:J
        for rot=1:length(CNew{scale})
            tau = alpha*2^(scale-length(CNew))* ...
            sigma*sqrt(2*log(length(CNew{scale}{rot}(:))) );
            rho = tau*rhofact;

            for k1=1:size(CNew{scale}{rot},1)
                for k2=1:size(CNew{scale}{rot},2)
                    x = CNew{scale}{rot}(k1,k2);
                    if x >= tau
                        CNew{scale}{rot}(k1,k2) = x - (2*rho-tau);
                    elseif (x > rho) && (x < tau)
                        CNew{scale}{rot}(k1,k2) = 2*(x-rho);
```

```
                    elseif (x >= -rho) && (x <= rho)
                        CNew{scale}{rot}(k1,k2) = 0;
                    elseif (x > -tau) && (x < -rho)
                        CNew{scale}{rot}(k1,k2) = 2*(x+rho);
                    elseif (x <= -tau)
                        CNew{scale}{rot}(k1,k2) = x + (2*rho-tau);
                    end
                end
            end
            cc = [cc; CNew{scale}{rot}(:)];
        end
    end

    % Reconstruction of f from coefficients.
    fnew = ifdct_wrapping(CNew, is_real);
    fold = fnew;

    % Save objective functional value for current iteration.
    convergence = [convergence; norm(mr-m,2)^2 + norm(cc,1)];
    if it >= 2
        % Output convergence during iterations
        it
        convergence(end)-convergence(end-1)
    end

end

% Output
C = CNew;
f = ifdct_wrapping(C, is_real);
```

Appendix E

Bibliography

[1] S. Bartels. Total variation minimization with finite elements. *SIAM*, pages 1163–1180, 2012.

[2] J Barzilai and J Borweinn. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8:141–148, January 1988.

[3] L. Borup, M. Nielsen, and Aalborg Universitet. Institut for Matematiske Fag. *Frame Decomposition of Decomposition Spaces*. Report: Institut for Matematiske Fag. Department of Mathematical Sciences, Aalborg University, 2006.

[4] Kristian Bredies and Dirk A. Lorenz. Linear convergence of iterative soft-thresholding. *University of Bremen, Center for Industiral Mathematics*, 2008.

[5] Emmanuel Candés, Laurent Demanet, David Donoho, and Lexing Ying. Fast discrete curvelet transforms, 2005.

[6] Emmanuel J. Candés and David L. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise c^2 singularities. *COMM. ON PURE AND APPL. MATH*, pages 219–266, 2002.

[7] A Caponnetto and M Bertero. Tomography with a finite set of projections: singular value decomposition and resolution. *Inverse Problems*, pages 1191–1295, 1997.

[8] E. Casas, K. Kunisch, and C. Pola. Regularization by functions of bounded variation and applications to image enhancement. *Appl. Math. Optim.*, pages 229–257, 1999.

[9] Ole Christensen. *Frames and bases, an introductory course*. Applied and numerical harmonic analysis. Birkhäuser, 2008.

[10] Ole Christensen. *Functions, Spaces, and Expansions*. Applied and numerical harmonic analysis. Birkhäuser, 2010.

[11] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57:1413–1457, 2004.

[12] M. E. Davison. A singular value decomposition for the radon transform in n-dimensional euclidean space. *Numerical Functional Analysis and Optimization*, 3:321–340, 1981.

[13] S.R. Deans. *The Radon Transform and Some of Its Applications*. Krieger Publishing Company, 1993.

[14] Ronald A. Devore and Robert C. Sharpley. Besov spaces on domains in r d, 1993.

[15] Jens Eising. *Lineær ALGEBRA*. Danmarks Tekniske Universitet, 1999.

[16] R. Ellis and I. Gohberg. *Orthogonal systems and convolution operators*. Operator Theory: Advances and Applications Series. Birkhauser Verlag GmbH, 2003.

[17] Brian Eriksson. The very fast curvelet transform.

[18] L. C. Evans. *Partial Differential Equations*. AMS, 2010.

[19] Jurgen Frikel. Sparse regularization in limited angle tomography. Technical Report arXiv:1109.0385, Sep 2011.

[20] Giusti. *Minimal Surfaces and Functions of Bounded Variation*. Monographs in Mathematics. Birkhäuser Boston, 1984.

[21] Robert M. Gray. Toeplitz and circulant matrices: A review. *Deptartment of Electrical Engineering, Stanford University*, 2001.

[22] K. Hämäläinen, L. Harhanen, A. Hauptmann, A. Kallonen, E. Niemi, and S. Siltanen. Total variation regularization for large-scale x-ray tomography. *Department of Physics, University of Helsinki, Finland*, 2013.

[23] P.C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. Mathematical Modeling and Computation. Society for Industrial and Applied Mathematics, 1998.

[24] Per Christian Hansen. *Discrete Inverse Problems, Insight and Algorithms*. SIAM, 2010.

[25] A. Kirsch. *An introduction to the mathematical theory of inverse problems*. Applied mathematical sciences. Springer New York, 2011.

[26] V. Kolehmainen, K. Niinimäki M. Lassas, and S. Siltanen. Sparsity-promoting bayesian inversion. *Inverse Problems, Volume 28, Number 2*, 2012.

[27] E. Kreyszig. *Introductory Functional Analysis with Applications*. Wiley Classics Library. Wiley, 1989.

[28] B.P. Lathi. *Signal processing and linear systems*. Berkeley Cambridge Press, 1998.

[29] Guojun Liu and Xianchu Feng. Curvelets-based iterative regularization and inverse scale methods. *Chinese Journal of Electronics*, 2009.

[30] Guojun Liu, Weiwei Wang, and Xianchu Feng. Curvelets shrinkage and variational method dependend on decomposition spaces. *Chinese Journal of Electronics*, 2010.

[31] Y. Meyer and D.H. Salinger. *Wavelets and Operators:*. Number vb. 1 in Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1995.

[32] Jennifer L. Mueller and Samuli Siltanen. *Linear and Nonlinear Inverse Problems with Practical Applications*, volume 10 of *Computational science and engineering*. SIAM, 2012.

[33] Frank Natterer. *The Mathematics of Computerized Tomography*. SIAM, 1990.

[34] M. Pedersen. *Functional analysis in applied mathematics and engineering*. Studies in Advanced Mathematics Series. Chapman & Hall/CRC Press, 2000.

[35] David Strong and Tony Chan. Edge-preserving and scale-dependent properties of total variation regularization. *Inverse Problems*, 19(6):S165, 2003.

[36] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58:267–288, 1996.

[37] Sergey Voronin and Hugo J. Woerdeman. A new iterative firm-thresholding algorithm for inverse problems with sparsity constraints. *Applied and Computational Harmonic Analysis*, 35:151–164, 2013.

[38] Emmanuel J. CandwsÂ·and Michael B. Wakin and Stephen P. Boyd. Enhancing sparsity by reweighted l1 minimization. *J Fourier Anal Appl*, 14:877–905, 2008.

[39] David F. Walnut. *An Introduction to Wavelet Analysis*. Birkhäuser, 2002.

[40] Liang Xiao, Li-Li Huang, and Badrinath Roysam. Image variational denoising using gradient fidelity on curvelet shrinkage. *EURASIP J. Adv. Signal Process*, 2010:2:1–2:16, February 2010.

[41] Hugh D. Young and Roger A. Freedman. *University Physics, 12th edition*. Pearson, 2008.