Erik Oscar Bøe

Shape Analysis for Inverse Problems

Master's thesis in Applied Physics and Mathematics
Supervisor: Markus Grasmair
June 2021

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

Erik Oscar Bøe

Shape Analysis for Inverse Problems

Master's thesis in Applied Physics and Mathematics
Supervisor: Markus Grasmair
June 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

Abstract

We consider the problem of inverting the Radon transform of a shape, with only sparse data from limited angles available. We propose a regularization functional, where the penalty term is based on the difference in bending energy of an a priori guess and the reconstructed shape. We approximate the ill-posed operator equation by a minimization problem involving the proposed regularization functional. In order to deal with the non-differentiability of the functional, we derive a smooth approximation, before implementing a method to solve the problem numerically. The reconstructed solution is then used as a priori guess in the next minimization of the functional. This is repeated for a fixed number of iterations. Numerical experiments show that, unless we have a very good initial guess of how the shape looks like, the best reconstructions is obtained with some degree of regularization. This result is even more apparent when the available data contains random noise.

Sammendrag

Denne oppgaven undersøker problemet med å invertere Radon-transformasjonen av en form, hvor vi kun har tilgjengelig data fra et fåtall vinkler i et begrenset intervall. For å løse dette foreslår vi en regulariseringsfunksjonal, hvor straffeleddet av funksjonalen er basert på forskjellen i krumningsenergien mellom den rekonstruerte kurven og en a priori gjetning. Den illestilte operatorlikningen approksimeres med et minimeringsproblem som involverer den foreslåtte regulariseringsfunksjonalen. For å håndtere ikke-deriverbarheten til funksjonalen utleder vi en glatt approksimasjon, før vi implementerer en metode for å løse problemet numerisk. Den rekonstruerte løsningen blir så brukt som a priori gjett neste gang vi minimerer funksjonalen, og dette repeteres så for et bestemt antall iterasjoner. De numeriske eksperimentene viser at, med mindre vi har en veldig god initialisering av formen, de beste rekonstruksjonene oppnås med en viss grad av regularisering. Dette resultatet er enda tydeligere når det er tilfeldig støy i den tilgjengelige dataen.

Preface

This report presents the final results of the Master's thesis related to the course TMA4900-Master's thesis. The report is the final work in the Master's degree program, Industrial Mathematics, at the Norwegian University of Science and Technology, NTNU. It is a continuation of the work done in the course TMA4500-Specialization project, where we decided on a specialization topic and investigated theory and methods related to this topic. This thesis applies Tikhonov-Morozov regularization for the reconstruction of shapes, based on available Radon transforms.

I want to give a special thanks to associate professor Markus Grasmair at the Department of Mathematical Sciences for the very valuable, thorough and constructive guidance this semester. It has been very helpful to have someone with such insight into the topic to help me focus my thesis, and present the work in the form of this report.

Contents

Chapter 1

Introduction

An important aspect of object recognition today is segmenting an object in an image. However, the entire image is not always available. In some cases we might only have available data for a transformed version of the image. In this application we will focus on the case where only the Radon transform of the image is available, for a given number of directions, and try to reconstruct the object based on this. Such a situation could occur when an object is inspected with a tomograph. In other words, we wish to solve the inverse problem of

$$\mathcal{R}[\Omega] = f,$$

where $\mathcal{R}$ describes the Radon transform, i.e., the action of the tomograph on the object Ω, and f denotes the available data.

This is an example of an ill-posed problem where the solution does not depend continuously on the data f. In addition, to ensure that biological harm caused by the radiation is kept to a minimum, we might wish to perform the scan for only a limited number of directions. Then, the available data will be insufficient for a unique reconstruction of the object, unless we have some other a priori knowledge about the object. If there is some a priori knowledge available, we can stabilize the reconstruction process by using some kind of regularization. A common approach is to incorporate the a priori knowledge in the initial guess, and then proceed to use an iterative method for matching the available data, but stop well before convergence [1]. Another approach, also described in [1], is to split the operator equation into a functional consisting of a fit-to-data term and a regularization term. This approach is commonly referred to as (generalized) Tikhonov regularization [2]. The fit-to-data term ensures that the solution closely matches the given data, while the regularization term incorporates some a priori knowledge of how the solution should look like.

In this application we will consider the reconstruction of shapes, i.e., objects with a homogeneous density described by binary density functions. This is done, with a Tikhonov regularization approach, in [3], where the shapes considered are star-shaped, and the regularization term is based on integral invariants. In [4] they do not address the exact same problem, but propose a regularization term, based on the Euler-Bernoulli bending energy, for measuring the difference of two shapes. In this application we will follow a similar procedure, for reconstruction of shapes, as in [3], but use the regularization term described in [4]. Furthermore, we generalize the method used in [3] by combining it with an iterative regularization method. This approach is known as Tikhonov-Morozov regularization. Besides, we do not restrict ourselves to star-shaped objects, but also try to reconstruct non star-shaped objects.

In section 2 we describe the shape space, the Radon transform, the bending energy and the optimization problem in a more precise mathematical manner. In section 3 and 4 we describe the discretization and implementation of the problem, while in section 5 we present the numerical experiments that have been conducted.

Chapter 2

Problem formulation

To formulate the problem precisely, consider a given set of functions $f_i \in L^2(\mathbb{R}, \mathbb{R})$ for $i \in \{1, \ldots, N\}$. We then wish to find a shape Ω such that these functions match the Radon transform of Ω for a corresponding set of directions $\sigma_i \in \mathbb{S}^1$, i.e., we wish to find a shape Ω such that,

$$\mathcal{R}_{\sigma_i}[\Omega] = f_i \quad \text{for} \quad i \in \{1, \ldots, N\}.$$

If the number N of directions is small, we cannot guarantee that the problem has a unique solution, and so we have to use some sort of regularization. Regularization is also necessary because of the ill-posedness of the problem, this becomes especially clear if the angles that determine the directions are limited to a narrow interval, as can be seen from the results shown in Figure 5.7. In this application we will use some a priori knowledge of the object, given as a reference shape that covers some essential information about the object, in order to obtain this regularization. This will give us an optimization problem with an objective functional consisting of a fit-to-data term and a regularization term. In the following we describe this more thoroughly.

2.1 Shape description

We consider a shape as a characteristic function of a simply connected and bounded set $\Omega \in \mathbb{R}^2$ with $\mathcal{C}^2$ boundary. To describe a shape we can describe the boundary curve that defines the shape. To achieve this we follow the parameterization procedure described in [4]. Consider the shape manifold of closed curves Γ in $\mathbb{R}^2$, and let Γ be a regular curve of class H^2 and length L. Then there exists a parameterization $\gamma \in H^2_{\text{per}}([0, 1], \mathbb{R}^2)$ of Γ such that $\gamma'(t)$ never vanishes, with $H^2_{\text{per}}([0, 1], \mathbb{R}^2)$ denoting the one-periodic functions in $H^2([0, 1], \mathbb{R}^2)$. Furthermore, without loss of generality with regard to which shapes the curve can describe, we can assume constant speed, i.e., $|\gamma'(t)| = L$. This curve can then be described by a triplet $m = (\theta, L, p)$, where p is the starting point, $\theta \in H^1([0, 1], \mathbb{R})$ is a function describing the angle of the tangent vector to $\gamma(t)$, and L is, as earlier mentioned, the length of the curve. The expression for $\gamma(t)$ then reads

$$\gamma(t) = \gamma_m(t) := p + L \int_0^t (\cos(\theta(\tau)), \sin(\theta(\tau))) \; \mathrm{d}\tau,$$

and since $\gamma \in H^2_{\text{per}}([0, 1], \mathbb{R}^2)$, it satisfies the closing conditions

$$\gamma(0) = \gamma(1), \qquad \gamma'(0) = \gamma'(1).$$

In terms of θ, these are equivalent to the equations

$$\int_0^1 \cos(\theta(t))\mathrm{d}t = \int_0^1 \sin(\theta(t))\mathrm{d}t = 0, \qquad \theta(1) - \theta(0) \in 2\pi\mathbb{Z},$$

where $\frac{\theta(1)-\theta(0)}{2\pi}$ is the turning number of the curve, i.e., the number of loops the curve performs. By further requiring that the turning number of the curve is $+1$, we restrict ourselves to curves that perform a single loop in a counter-clockwise manner, and obtain the angle function space

$$\Theta := \left\{ \theta \in H^1([0,1],\mathbb{R}) \;\middle|\; \int_0^1 \big(\cos(\theta(t)), \sin(\theta(t))\big)\mathrm{d}t = \mathbf{0},\ \theta(1) - \theta(0) = 2\pi \right\}.$$

Thus our space of feasible triplets is defined by

$$\mathcal{M} := \Theta \times \mathbb{R}_{>0} \times \mathbb{R}^2,$$

which implicitly defines the space of feasible curves,

$$\mathcal{S} := \{\gamma_m \mid m \in \mathcal{M}\}.$$

Shapes described by a curve $\gamma_m \in \mathcal{S}$ can be found by considering the winding number of points in $\mathbb{R}^2$. Points that are inside the curve will have winding number 1, while points that are outside the curve will have winding number 0, such that the shape Ω_{γ_m} is given by

$$\Omega_{\gamma_m} := \left\{ x \in \mathbb{R}^2 \mid \frac{1}{2\pi}\int_0^1 \mathrm{d}\phi_{x\gamma_m}(t) = 1 \right\},$$

where $\phi_{x\gamma_m}(t)$ is the signed angle of $\gamma_m(t) - x$.

Note that when reconstructing a domain, one requires a boundary curve that is free of self-intersections. This is not guaranteed for our shape descriptor, as the space $\mathcal{S}$ contains self-intersecting curves, which means that the set of feasible curves is actually only a subset of $\mathcal{S}$. We will address this issue in the next section. Another issue with this parameterization is non-uniqueness: a shape Ω, with a corresponding boundary curve γ, does not give rise to a unique triplet m. To see this, note that for any triplet m we could shift the starting point p arbitrarily along the curve, while also shifting the θ-function accordingly, and still get the same curve. In addition, a shift of the angle function θ by any multiple of 2π would yield the same curve. This can lead to problems if we want to compare shapes based on their parameterizations, as it would require us to determine which parameterizations yield the same shape. In our application, we do not want to compare shapes in this manner, but the regularization term depends on the parameterization that is used for the a priori guess, and thus the parameterization of the reconstructed shape will do so as well. However, the shape itself will not depend on the parameterization, and we can therefore ignore this issue.

2.2 Möbius Energy

To avoid the problem of self-intersecting curves we introduce a popular and widely studied energy that is self-avoiding, known as the Möbius energy [4]. For a curve $\gamma_m \in \mathcal{S}$ this is defined by

$$\mathcal{E}_M^*[\gamma_m] = \int_{\gamma_m}\int_{\gamma_m} \left(\frac{1}{|x-y|^2} - \frac{1}{d_{\gamma_m}(x,y)^2} \right) \mathrm{d}s(x)\mathrm{d}s(y),$$

where $|x - y|$ denotes the Euclidean distance between x and y, $d_{\gamma_m}(x, y)$ denotes the geodesic distance between x and y along γ_m, and integration is performed with respect to the line elements. Self-avoiding in this sense means that finiteness of $\mathcal{E}_M^*[\gamma_m]$ is equivalent to γ_m not having any self-intersections. Observe that the integrand is constructed in such a way that we have singularities precisely when x approaches y, while $d_{\gamma_m}(x, y)$ does not approach zero. As we are considering curves of constant speed L, the Möbius energy can, using the expression in [5, equation 0.3, p. 2], equivalently be computed as

$$\mathcal{E}_M^*[\gamma_m] = L^2 \int_0^1 \int_0^1 \left(\frac{1}{|\gamma_m(t) - \gamma_m(t')|^2} - \frac{1}{d_{\gamma_m}(\gamma_m(t), \gamma_m(t'))^2} \right) \mathrm{d}t\mathrm{d}t'.$$

By including this energy as a penalty term in our objective functional we ensure that the minimizers of the regularized problem are indeed free of self-intersections [4].

The self-avoiding property is more difficult to preserve in numerical experiments, because when conducting numerical integration, only a finite number of points on the curve can be included in the calculation. This means that even for a self-intersecting curve, two points that are included in the calculation of the integral will not have the exact same position. As a result, the self-avoiding property of the Möbius energy will not necessarily hold when performing numerical integration. However, with enough discretization points, the energy will become very large in the case of an intersection. Assuming that this is the case we can make the energy of a self-intersecting curve even larger by redefining the energy to

$$\mathcal{E}_M[\gamma_m] = L^8 \int_0^1 \int_0^1 \left(\frac{1}{|\gamma_m(t) - \gamma_m(t')|^8} - \frac{1}{d_{\gamma_m}(\gamma_m(t), \gamma_m(t'))^8} \right) \mathrm{d}t\mathrm{d}t',$$

which gives us a steeper function, in the sense that the two terms in the integrand grows more rapidly for low distances. This should in turn make the required number of discretization points, needed to ensure that the energy of a self-intersecting curve is sufficiently large, decrease. What is meant by sufficiently large will be explained in section 4.4.

2.3 Bending energy

Before defining the regularization term of the objective functional, we define the concept of the bending energy of a curve. For a curve Γ the Euler-Bernoulli bending energy [4] is given by

$$\int_\Gamma \kappa^2 \mathrm{d}s,$$

where κ denotes the curvature of Γ and $\mathrm{d}s$ is the line element. The bending energy models the energy required to deform a straight line of the same length as Γ into Γ. This measure also has the benefit of being independent of the choice of parameterization of Γ. With our parameterization the curvature is given by $\kappa = \frac{\theta'(t)}{L}$ and the line element is given by $\mathrm{d}s = L\mathrm{d}t$. This results in a bending energy expression,

$$\mathcal{E}_L(m) = \int_0^1 \frac{\theta'(t)^2}{L} \mathrm{d}t,$$

which is dependent on the length L. We choose to proceed with the scale-invariant bending energy, $\mathcal{E}(m) : \Theta \to [0, \infty)$, instead, which is given by

$$\mathcal{E}(m) = \int_0^1 \theta'(t)^2 \mathrm{d}t,$$

i.e., the H^1- seminorm. This gives us a scale-invariant version of the energy needed to deform a straight elastic rod of the same length, L, into the curve described by the triplet m.

For the purpose of the application in this thesis, we wish to describe the bending energy required to deform more general curves into a given curve γ_m. Consider another curve described by the triplet $m^* \in \mathcal{M}$, then the scale-invariant energy required to transform this curve into γ_m is given by

$$\mathcal{E}_b(m, m^*) = \int_0^1 (\theta'(t) - \theta^{*\prime}(t))^2 \mathrm{d}t.$$

This functional will be used as a regularization term in the objective functional, with m^*, and in particular θ^*, representing a reasonable a priori guess of the true solution to the inverse problem. In the absence of a priori knowledge of the shape, a natural selection would be to let m^* describe a circle. In this case we would have $\theta^*(t) = 2\pi t - c$ for some constant c and

$$\mathcal{E}_b(m, m^*) = \int_0^1 (\theta'(t) - 2\pi)^2 \mathrm{d}t.$$

In addition to being scale-invariant, this energy is invariant under translation and rotation of the curve described by m^*. The first is obvious because altering the starting point p^* will not affect the energy, as it does not appear in the expression. To see that it is invariant under rotation note that this would correspond to shifting the θ^*-function by a constant, which again would disappear in differentiation.

2.4 The Radon transform

To describe the problem precisely we need to describe the Radon transform. First, recall that in this application we make the simplification of assuming homogeneous objects and backgrounds, even though this is not realistic for physical objects. Thus, we wish to describe the Radon transform of a shape described by a boundary curve γ_m, for a triplet $m \in \mathcal{M}$. In addition, we let the background intensity be equal to zero and the function value in the shape be equal to one. The Radon transform, for a given direction $\sigma_i \in \mathbb{S}^1$, $\mathcal{R}_{\sigma_i} : \mathcal{S} \to L^2(\mathbb{R}, \mathbb{R})$, is then given by

$$\mathcal{R}_{\sigma_i}[\gamma_m](\alpha) := \int_{\mathbb{R}} \chi_{\Omega_{\gamma_m}}(\alpha \sigma_i^\perp + \tau \sigma_i) \mathrm{d}\mathcal{L}^1(\tau), \tag{2.1}$$

where $\mathcal{L}^1$ denotes the Lebesgue measure, χ the characteristic function and α the perpendicular displacement of the direction vector σ_i. Note that for $\sigma_i = (\sigma_i^1, \sigma_i^2)$ we define $\sigma_i^\perp := (\sigma_i^2, -\sigma_i^1)$, i.e., the direction obtained by rotating σ_i 90 degrees clock-wise. An illustration of the Radon transform, for a given direction σ_i, is shown in Figure 2.1. It is now desireable to find an expression for the Radon transform that depends explicitly of γ_m.

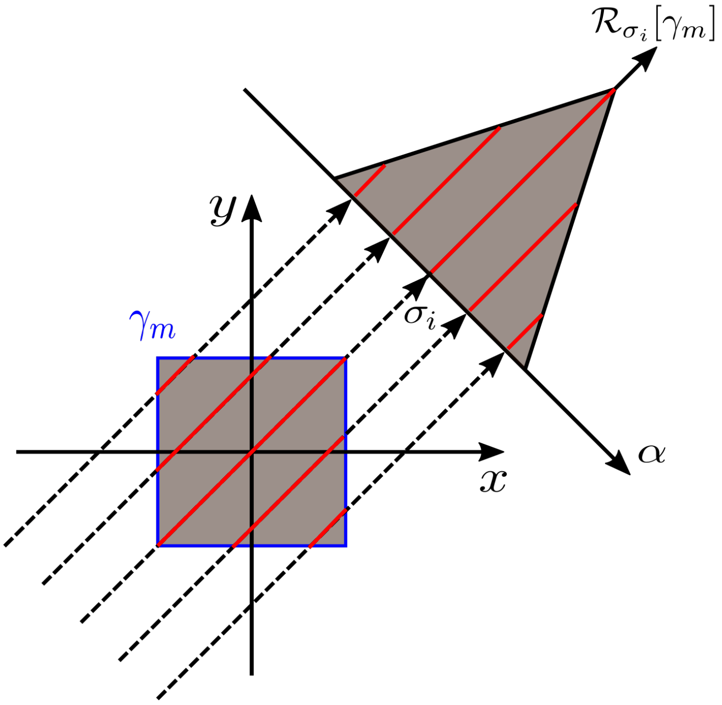

Figure 2.1: A sketch of the Radon transform for a given direction σ_i. Here Ω_{γ_m} is the grey square inside the blue boundary curve γ_m. The stippled lines are given by $\tau\sigma_i + \alpha\sigma_i^\perp$ for $\tau \in \mathbb{R}$ and a given α, where the center line corresponds to $\alpha = 0$. To find the Radon transform for a given α we first find the corresponding line $\tau\sigma_i + \alpha\sigma_i^\perp, \tau \in \mathbb{R}$. Then the Radon transform is the length of the part of this line that lies within Ω_{γ_m} (colored in red).

Theorem 2.1 *Denote by $H(x)$ the Heaviside function defined by $H(x) = 0$ for $x < 0$ and $H(x) = 1$ for $x \geq 0$. The operator in (2.1) can, for $\gamma_m \in \mathcal{S}$ a non self-intersecting curve, be written as*

$$\mathcal{R}_{\sigma_i}[\gamma_m](\alpha) := \int_0^1 H\Big(\langle \gamma_m(t), \sigma_i^\perp\rangle - \alpha\Big)\langle \gamma_m'(t), \sigma_i\rangle dt, \tag{2.2}$$

where $\langle\cdot,\cdot\rangle$ denotes the Euclidean inner product on $\mathbb{R}^2$.

Proof. Let $\alpha \in \mathbb{R}$ and $\sigma_i \in \mathbb{S}^1$ be fixed. Then define the projection

$$f_{\sigma_i,\gamma_m}(t) = \langle \gamma_m(t), \sigma_i\rangle$$

of γ_m onto σ_i. First, note that we can write

$$\begin{aligned}&\int_0^1 H\Big(\langle \gamma_m(t), \sigma_i^\perp\rangle - \alpha\Big)\langle \gamma_m'(t), \sigma_i\rangle \mathrm{d}t\\ &= \int_0^1 \operatorname{sgn}\big(\langle \gamma_m'(t), \sigma_i\rangle\big) H\Big(\langle \gamma_m(t), \sigma_i^\perp\rangle - \alpha\Big)\big|\langle \gamma_m'(t), \sigma_i\rangle\big|\mathrm{d}t.\end{aligned}$$

Next, from the one dimensional coarea formula [6, theorem 3.2.3(2), p. 243], we can further write

$$\begin{aligned}&\int_0^1 \operatorname{sgn}\big(\langle \gamma_m'(t), \sigma_i\rangle\big) H\Big(\langle \gamma_m(t), \sigma_i^\perp\rangle - \alpha\Big)\big|\langle \gamma_m'(t), \sigma_i\rangle\big|\mathrm{d}t\\ &= \int_{\mathbb{R}}\bigg(\int_{f^{-1}(\tau)\cap[0,1]} H\Big(\langle \gamma_m(t), \sigma_i^\perp\rangle - \alpha\Big)\operatorname{sgn}(\langle \gamma_m'(t), \sigma_i\rangle)\mathrm{d}\mathcal{H}^0(t)\bigg)\mathrm{d}\tau.\end{aligned}$$

Observe that for every τ such that $f^{-1}(\tau) \cap [0,1]$ is a finite set and $\langle \gamma_m'(t), \sigma_i \rangle \neq 0$ for all $t \in f^{-1}(\tau) \cap [0,1]$, the integrand is well defined. It follows from the coarea formula that this is the case for almost every τ. To see this more clearly recall that γ_m has a finite length L, which can be expressed by the integral

$$L = \int_0^1 |\gamma_m'(t)| \mathrm{d}t = \int_{\mathbb{R}} \left(\int_{f^{-1} \cap [0,1]} \frac{1}{|\langle \gamma_m'(t), \sigma_i \rangle|} \mathrm{d}\mathcal{H}^0(t) \right) \mathrm{d}t.$$

Here the last integral is finite if and only if we have for almost every τ that $f^{-1}(\tau) \cap [0,1]$ is a finite set and $\langle \gamma_m'(t), \sigma_i \rangle \neq 0$ for all $t \in f^{-1}(\tau) \cap [0,1]$.

Now that we have established that the integrand is well defined almost everywhere, observe that for all such τ the integrand will evaluate to either zero or one for any counterclockwise curve. It will evaluate to one for any point inside the curve and to zero for any point outside the curve. To see this, observe that for a given τ^* the integrand can be viewed as the sum of the intersections with $\gamma_m(t)$, when following a ray in direction $\sigma_i^\perp$ from the point $\tau^* \sigma_i + \alpha \sigma_i^\perp$. Sum in this sense means that intersections with $\langle \gamma_m'(t), \sigma_i \rangle < 0$ will subtract one, while intersections with $\langle \gamma_m'(t), \sigma_i \rangle > 0$ will add one. A sketch to illustrate this is shown in Figure 2.2. It follows that we obtain

$$\begin{aligned} &\int_{\mathbb{R}} \left(\int_{f^{-1}(\tau) \cap [0,1]} H\Big(\langle \gamma_m(t), \sigma_i^\perp \rangle - \alpha \Big) \operatorname{sgn}(\langle \gamma_m'(t), \sigma_i \rangle) \mathrm{d}\mathcal{H}^0(t) \right) \mathrm{d}\tau \\ &= \int_{\mathbb{R}} \chi_{\Omega_{\gamma_m}}(\alpha \sigma_i^\perp + \tau \sigma_i) \mathrm{d}\mathcal{L}^1(\tau), \end{aligned}$$

which completes the proof.

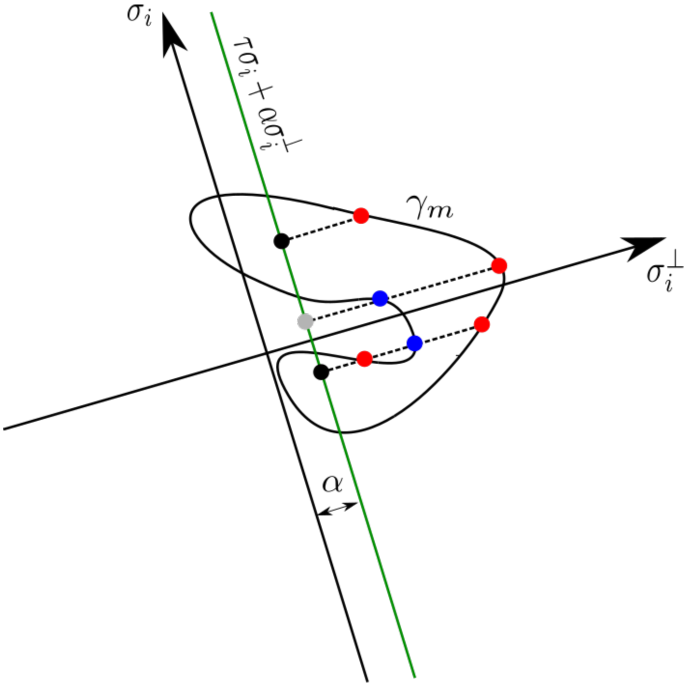

Figure 2.2: A sketch of some curve γ_m, with the line $\tau\sigma_i + \alpha\sigma_i^\perp$, for $\tau \in \mathbb{R}$ and a fixed α. Included are three arbitrary points on this line, two interior points of γ_m in black, and one exterior point in grey. For each point a stapled line is drawn in direction $\sigma_i^\perp$ to find the intersection points with γ_m. The points in red represent the intersections for which $\langle \gamma_m'(t), \sigma_i \rangle$ is positive, while the blue points represent the intersections where this quantity is negative. Recall that $\gamma_m(t)$ is oriented counterclockwise.

2.5 The optimization problem

We can now define the objective functional in a precise manner. Let $m^* \in \mathcal{M}$ be a triplet describing a reference solution $\gamma_{m^*} \in \mathcal{S}$. We then recall the expression for the scale invariant bending energy, and define

$$\mathcal{P}[\gamma_m, \gamma_{m^*}] := \mathcal{E}(m, m^*).$$

Furthermore, for a given $\sigma_i \in \mathbb{S}^1$, let $f_i \in L^2(\mathbb{R}, \mathbb{R})$ be the corresponding measurement data. We can then define the functional

$$\mathcal{D}_{\sigma_i}[\gamma_m] := ||\mathcal{R}_{\sigma_i}[\gamma_m] - f_i||_{L^2}^2,$$

i.e., the squared L^2-norm of the difference between the measurement data and the Radon transform of the shape described by γ_m, for the direction σ_i. Now, for a given regularization parameter $\beta > 0$, a reference solution m^* and a set of directions $\{\sigma_1, \dots, \sigma_N\}$, we define the objective functional $F : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ by

$$F[\gamma_m, \gamma_{m^*}] = \frac{1}{2} \sum_{i=1}^{N} \mathcal{D}_{\sigma_i}[\gamma_m] + \frac{\beta}{2} \mathcal{P}[\gamma_m, \gamma_{m^*}] + \mathcal{E}_M[\gamma_m].$$

Minimizing this functional within the constraints of the curve, γ_m, gives us the optimization problem

$$\begin{aligned} \min_{\gamma_m} \quad & F[\gamma_m, \gamma_{m^*}] \\ \text{s.t.} \quad & \gamma_m(0) = \gamma_m(1), \\ & \gamma_m'(1) = \gamma_m'(0), \end{aligned}$$

which can be expressed in the triplet $m = (p, L, \theta)$ as

$$\begin{aligned} \min_{m=(p,L,\theta)} \quad & F[\gamma_m, \gamma_{m^*}] \\ \text{s.t.} \quad & \int_0^1 \cos(\theta(t))\mathrm{d}t = \int_0^1 \sin(\theta(t))\mathrm{d}t = 0, \\ & \theta(1) - \theta(0) = 2\pi. \end{aligned}$$

Chapter 3

Discretization

3.1 Adjustment of the problem formulation

Before we attempt to solve the optimization problem numerically, we must address certain issues, and make choices regarding discretization and optimization methods. First of all, the optimality condition of the optimization problem involves the derivative of the functional $\mathcal{R}_{\sigma_i}[\gamma_m](\alpha)$, which is not differentiable with respect to γ_m, as γ_m appears as an argument to the Heaviside function in the integrand. For the numerical implementation we will therefore replace the Heaviside function by a smooth approximation defined by

$$H_\epsilon(x) = \frac{1}{1+e^{-2x/\epsilon}}$$

for a small ϵ. Inserting this approximation in the expression for the Radon transform we obtain,

$$\mathcal{R}^\epsilon_{\sigma_i}[\gamma_m](\alpha) := \int_0^1 H_\epsilon\bigg(\langle \gamma_m(t), \sigma_i^\perp\rangle - \alpha\bigg)\langle \gamma_m'(t), \sigma_i\rangle \mathrm{d}t,$$

which gives us the smooth approximate functional,

$$F_\epsilon[\gamma_m, \gamma_{m^*}] = \frac{1}{2}\sum_{i=1}^N ||\mathcal{R}^\epsilon_{\sigma_i}[\gamma_m] - f_i||^2_{L^2} + \frac{\beta}{2}\mathcal{P}[\gamma_m, \gamma_{m^*}] + \mathcal{E}_M[\gamma_m].$$

Using this approximation we wish to determine an upper bound for the resulting error in the functional. To do this we first find an upper bound for the error in the Radon transform. As the curve has a known length L we can find such an upper bound by first calculating the difference between the integral of the original and the mollified Heaviside function, from zero to L. Multiplying this difference by L will then provide an upper bound. The integral of the mollified Heaviside function from zero to L is given by

$$\int_0^L \frac{1}{1+e^{\frac{-2x}{\epsilon}}} = \frac{\epsilon}{2}(\ln(e^{\frac{2L}{\epsilon}}+1) - \ln(2)),$$

while the integral of the original Heaviside function is equal to L. Multiplying the difference of these integrals by L and assuming $L/\epsilon \gg 1$ we get the upper bound

$$L\left|L - \frac{\epsilon}{2}(\ln(e^{\frac{2L}{\epsilon}}+1) - \ln(2))\right| = L\epsilon\ln(\sqrt{2}) + o(\epsilon)$$

Adding this error term to the Radon transform, when calculating the value of the objective functional, will give an error of size $\mathcal{O}(L\epsilon)$. The size of this error will later be compared to the errors resulting from numerical integration, in order to find a reasonable value for ϵ.

To solve the problem of minimizing F_ϵ numerically we must choose a numerical method for solving constrained optimization problems. The approach chosen in this application is to replace the two non-linear constraints by quadratic penalty terms, while the last constraint is built in explicitly. Thus, for a given $\lambda > 0$, and by denoting

$$\begin{aligned} c_1(\theta) &= \int_0^1 \cos(\theta(t))\mathrm{d}t, \\ c_2(\theta) &= \int_0^1 \sin(\theta(t))\mathrm{d}t, \end{aligned}$$

our new problem reads

$$\begin{aligned} \min_{m=(p,L,\theta)} \quad & F_\epsilon[\gamma_m, \gamma_{m^*}] + \frac{\lambda}{2}\sum_{i=1}^{2} c_i(\theta)^2 \\ \text{s.t.} \quad & \theta(1) - \theta(0) = 2\pi. \end{aligned}$$

3.2 Discretization of the optimization variables

An issue when solving the problem numerically is that the optimization problem is infinite dimensional, as the θ variable is infinite dimensional. To avoid this issue we choose a discretization of the interval $[0,1]$ into $n+1$ equally spaced points, $t_j = j/n$ for $j \in \{0,\dots,n\}$. This gives us the set of variables

$$\Theta_{n+1} = \{\theta_0,\dots,\theta_n\} = \{\theta(t_0),\dots,\theta(t_n)\},$$

replacing the infinite dimensional θ variable. Furthermore, we observe that the last constraint of the problem, $\theta(1)-\theta(0) = 2\pi$, can be satisfied automatically by viewing only the variables $\{\theta_0,\dots,\theta_{n-1}\}$ as decision variables and then setting $\theta_n = \theta_0 + 2\pi$. This redefines our set of variables to

$$\Theta_n = \{\theta_0,\dots,\theta_{n-1}\} = \{\theta(t_0),\dots,\theta(t_{n-1})\}.$$

3.3 Discretization of the objective functional

Now that we have the finite dimensional set of variables $m_n = (p, L, \Theta_n)$ we must choose an integration method in order to approximate the functional of the optimization problem. For this purpose we will use the trapezoidal rule, which means that we are viewing the functions as piecewise linear. Thus, we approximate the curve γ_m, in the points $t_j, j \in \{0,\dots,n\}$, by the approximate γ_{m_n} given by

$$\gamma_{m_n}(t_j) = p + L\sum_{i=0}^{j-1}\left(\frac{\cos(\theta_i)+\cos(\theta_{i+1})}{2n}, \frac{\sin(\theta_i)+\sin(\theta_{i+1})}{2n}\right),$$

for $j \in \{1,\dots,n\}$, and $\gamma_{m_n}(t_j) = p$, for $j = 0$. Next, by applying the fundamental theorem of calculus, we find that

$$\gamma_m'(t) = L(\cos(\theta(t)), \sin(\theta(t))),$$

and so a natural choice is to use the approximation

$$\gamma'_{m_n}(t_j) = L(\cos(\theta_j), \sin(\theta_j)),$$

for $j \in \{0, \ldots, n\}$. Here it should be noted that for γ_{m_n} the derivative is not uniquely defined in these points, which is why we have to choose an approximation.

The absolute error of the trapezoidal rule, when integrating a function f from 0 to 1, is given by [7]

$$\frac{|f''(\xi)|}{12n^2}, \tag{3.1}$$

for some $\xi \in [0, 1]$. We obtain an upper bound for this error by choosing the maximum value of $|f''(\xi)|$ on the interval. In the approximation of γ_m, the functions to integrate are $\sin(\theta(\xi))$ and $\cos(\theta(\xi))$, and so the maximum value of the second derivative is dependent on the first and second derivative of θ. Assuming that the curves involved in our problem are somewhat regular, we do not expect these values to be too large. This gives us an expected error in γ_{m_n} of size $\mathcal{O}(L/n^2)$.

3.3.1 Discretization of the fit-to-data term

Now we can define the discretized Radon transform for a given n, α and direction σ_k by

$$\mathcal{R}^{\epsilon}_{\sigma_k}[\gamma_{m_n}](\alpha) = \sum_{i=0}^{n-1} \frac{I_i + I_{i+1}}{2n},$$

where

$$I_i = H_\epsilon\left(\langle \gamma_{m_n}(t_i), \sigma_k^\perp \rangle - \alpha\right) \langle \gamma'_{m_n}(t_i), \sigma_k \rangle.$$

To find an upper bound for the error in the Radon transform, as a result of using the trapezoidal rule, we again use the formula in (3.1). However, for this approximation the second derivative could become very large for some values. The approximated Heaviside function has a maximum value of the second derivative equal to

$$\frac{2}{3\sqrt{3}\epsilon^2}.$$

Differentiating the Heaviside function with respect to t two times also gives us the term $\langle \gamma'_{m_n}(t_i), \sigma_k^\perp \rangle^2$ from the chain rule, which has a maximum value of L^2. The other factor of the integrand is bounded by L, which gives us a maximum value of

$$\frac{2L^3}{3\sqrt{3}\epsilon^2}$$

for the dominating term of the second derivative. This term alone gives us a terrible upper bound for the integration error. However, this is an absolute worst case, and we should expect that for most intervals $[t_i, t_{i+1}]$ the maximum of the first and second derivative of the Heaviside function will be approximately zero. In this case, the upper bound of the integration error is expected to be of the forementioned size $\mathcal{O}(L/n^2)$. Also, as the integrand is bounded in the interval $[0, L]$, the maximum possible integration error for a single interval is bounded by $L/2n$. Since we assume few or no intervals where this is the case, the total integration error is expected to be of this size at most.

To compute the L_2-norm of the difference between the computed Radon transform and the available data, we must discretize the α-domain. Recall that α represents the perpendicular displacement of the direction vector σ_k. It is therefore important that the domain gives a Radon transform that covers the entire object. However, as we operate with images, this domain will be bounded by the size of the image and can be discretized into m evenly distributed points $\{\alpha_1^k, \dots, \alpha_m^k\}$. By further denoting

$$E_i^k = \mathcal{R}_{\sigma_k}^{\epsilon}[\gamma_{m_n}](\alpha_i^k) - f_k(\alpha_i^k),$$

we acquire the approximation of the L^2-norm

$$\sum_{i=1}^{m-1} \frac{(E_i^k)^2 + (E_{i+1}^k)^2}{2(m-1)},$$

where the α-domain has been normalized by the division of $m-1$. For a set of directions, σ_k, $k \in \{1, \dots, N\}$ with corresponding sets of α's $\{\alpha_1^k, \dots, \alpha_m^k\}$ we further obtain an approximation of the first term in our objective functional,

$$\frac{1}{2}\sum_{k=1}^{d} \mathcal{D}_{\sigma_k}[\gamma_{m_n}] = \frac{1}{2}\sum_{k=1}^{N}\sum_{i=0}^{m-1} \frac{(E_i^k)^2 + (E_{i+1}^k)^2}{2(m-1)}.$$

To find an estimate of the error when calculating $\mathcal{D}_{\sigma_k}$, by integrating over α, we could again use the formula in equation (3.1). This would yield an error of similar size as for the approximation of the Radon transform, as the second derivative involves the second derivative of the mollified Heaviside function. However, as we look at a connected domain, and the boundary curve is of length L, the maximum possible integration error is $(L/2)^2/(2m-2)$. One situation that could give an integration error close to this, is, for instance, if the data f_k is the zero function, while the reconstructed shape converges towards a very thin line of length $L/2$, parallel to the direction σ_k. This situation is highly unlikely, but any line segment on the reconstructed shape that is almost parallel to σ_k could potentially give integration errors of this size.

3.3.2 Discretization of the regularization term

To find an approximation of the regularization term in the objective functional, we must decide on a finite difference scheme for approximating the derivative. For this purpose we will use a forward difference scheme, which for our discretization is defined by

$$\delta\theta_i = \frac{1}{\frac{1}{n}}(\theta_{i+1} - \theta_i) = n(\theta_{i+1} - \theta_i), \text{ for } i \in \{0, \dots, n-1\},$$

with $\theta_n = \theta_0 + 2\pi$. Now let θ^* be the angle function of m^*, i.e., our reference solution. Then, by discretizing θ^* in the same manner as θ, we can approximate the second term of the objective functional, $\mathcal{P}[\gamma_m, \gamma_{m^*}]$, by the discretized version

$$\mathcal{P}[\gamma_{m_n}, \gamma_{m_n^*}] = \sum_{i=0}^{n-1} \frac{n}{2}\left((\delta\theta_i - \delta\theta_i^*)^2 + (\delta\theta_{i+1} - \delta\theta_{i+1}^*)^2\right).$$

The error in the forward difference scheme is of size $\mathcal{O}(1/n)$ [8], which leads to a total error in the estimation of $\mathcal{P}$ of size $\mathcal{O}(1/n^2)$. To find an estimate of the integration error we could use the formula in equation (3.1) again, but we do not have an estimate of the second derivative. However, as $\mathcal{P}$ is a penalty term punishing irregularity in the reconstructed solution, we would not expect the second derivatives of θ to be too large, assuming that we have a somewhat regular reference solution. Thus, we expect an integration error of size $\mathcal{O}(1/n^2)$ for this term.

3.3.3 Discretization of the Möbius energy penalty term

To find an approximation of the Möbius energy penalty term we again apply the trapezoidal rule. First we define the integrand, for given $i, j \in \{0, \dots, n\}$, to be

$$J_{i,j} = \begin{cases} \dfrac{1}{|\gamma_{m_n}(t_i) - \gamma_{m_n}(t_j)|^8} - \dfrac{1}{d_{\gamma_{m_n}}(\gamma_{m_n}(t_i), \gamma_{m_n}(t_j))^8} & \text{if } \ i \neq j \\ 0 & \text{else,} \end{cases}$$

where we have included a conditional statement to avoid numerical difficulties with zero division. As we have a double integral we first define the approximation of the innermost integral, for a fixed i, by

$$K_i = \sum_{j=0}^{n-1} \frac{J_{i,j} + J_{i,j+1}}{2n}.$$

Then we can define the approximation to the Möbius energy penalty term as

$$\mathcal{E}_M[\gamma_{m_n}] = L^2 \sum_{i=0}^{n-1} \frac{K_i + K_{i+1}}{2n}.$$

In the calculation of this term we are not too worried about the integration error, as it is mainly the self-avoiding property of this energy that is of importance. However, when we discretize the Möbius energy it is necessary that the grid is adequately fine, in order for the energy to preserve the self-avoiding property. What is meant by adequately fine will be described in more detail in section 4.4. To ensure this, without having to use a much finer grid for the entire discretization, we add k interpolation points between every pair of consecutive points $(\gamma_{m_n}(t_i), \gamma_{m_n}(t_{i+1}))$ on γ_{m_n}, using linear interpolation, before calculating the Möbius energy.

3.3.4 Discretization of the quadratic penalty term

Lastly, we can approximate the penalty terms of our functional by the discretized versions

$$c_1(\Theta_n) = \sum_{i=0}^{n-1} \frac{\cos(\theta_i)}{n}$$
$$c_2(\Theta_n) = \sum_{i=0}^{n-1} \frac{\sin(\theta_i)}{n},$$

where we have used the trapezoidal rule again and exploited that $\sin(\theta_0) = \sin(\theta_n)$ and $\cos(\theta_0) = \cos(\theta_n)$, as $\theta_n = \theta_0 + 2\pi$. In this approximation the second derivatives is again bounded by terms consisting of the first and second derivatives of θ, and the error is thus, based on the same reasoning as before, expected to be of size $\mathcal{O}(1/n^2)$.

In total, for a given n and by defining

$$Q_\epsilon[m_n, m_n^*, \lambda] := F_\epsilon[\gamma_{m_n}, \gamma_{m_n^*}] + \frac{\lambda}{2}\sum_{i=1}^{2} c_i(\Theta_n)^2,$$

we arrive at the discretized problem

$$\min_{m_n=(p,L,\Theta_n)} \quad Q_\epsilon[m_n, m_n^*, \lambda]. \tag{3.2}$$

Chapter 4

Implementation details

4.1 The BFGS method with backtracking line search

In order to solve (3.2) numerically, for a fixed λ, a line search method commonly known as the BFGS (Broyden-Fletcher-Goldfarb-Shanno) method [9] has been implemented. This method is, for a decision variable x and an initial guess x^0, based on updating x iteratively by the formula,

$$x^{k+1} = x^k + \eta_k q_k,$$

where η_k denotes the step length and q_k the direction. For a given direction q_k, the step length η_k is chosen so that for a given set of constants $c_1, c_2 \in (0,1)$, with $c_1 < c_2$, the strong Wolfe conditions [9]

$$\begin{aligned} f(x_k + \eta_k q_k) &\leq f(x_k) + c_1 \eta_k \nabla f(x_k)^T q_k, \\ |\nabla f(x_k + \eta_k q_k)^T q_k| &\geq c_2 |\nabla f(x_k)^T q_k|, \end{aligned}$$

are satisfied. The choice for η_k is obtained using the same algorithm as described in [9, Algorithm 3.5, p. 60].

The direction q_k is determined by considering a quadratic model of the objective function f,

$$m_k(q) = f_k + \nabla f_k^T q + q^T B_k q, \tag{4.1}$$

where B_k is a symmetric positive definite matrix that will be updated in each iteration. The search direction, and minimizer of (4.1), is then given by

$$q_k = B_k^{-1} \nabla f_k.$$

The BFGS method is a quasi-Newton method and the matrix B_k is an approximation to the Hessian at x_k. Since it is the inverse of B_k that is needed to compute the search direction q_k, we focus on finding an approximation of this and denote $H_k = B_k^{-1}$. Instead of computing the inverse Hessian at every iteration, we start with an initial guess, H_0, and then iteratively update H_k by,

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T$$

where

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla f_{k+1} - \nabla f_k \quad \text{and} \quad \rho_k = \frac{1}{y_k^T s_k}.$$

For a step length η_k, satisfying the strong Wolfe conditions, H_k will always be a positive definite matrix, such that q_k is a descent direction [9].

We observe that each iteration of this method avoids the calculation of second derivatives. Thus, for our application, the operation with highest cost is the numerical gradient evaluation, with a computational complexity of $\mathcal{O}(n^3)$. The rate of convergence is super linear, and even though Newton's method converges more rapidly (quadratically), its cost per iteration is much higher due to the calculation of second derivatives, which would have a computational complexity of $\mathcal{O}(n^4)$ if done numerically. It is also worth mentioning that the superlinear/quadratic convergence only holds for sufficiently regular functionals to be minimized. Technically, this is the case here, but, as earlier mentioned, we have very large second and third derivatives of the functional, as a result of the approximated Heaviside function in the integrand for the Radon transform. The Möbius energy could potentially also have large second and third derivatives. Thus, the theoretical advantage of Newton's method might not be visible in practice.

To determine when the BFGS method should terminate, we need to choose a convergence criterion. One way to choose this criterion is by considering the algorithm's progress in terms of reducing the objective function. We wish to terminate the method when it seems to be making almost no improvement in minimizing the objective function. Now, for a given tolerance $T > 0$, we consider an iteration in which

$$\left| \frac{f(x_{k+1}) - f(x_k)}{f(x_k)} \right| < T, \tag{4.2}$$

to be an iteration with almost no improvement. If multiple consecutive iterations yield almost no improvement, we consider the method's progress to be insufficient. Thus, for a fixed constant D, we terminate the method if (4.2) is fulfilled for D consecutive iterations.

4.2 Numerical derivatives

To use the method described above we need to find an expression for the gradient of the objective function $Q_\epsilon[m_n, m_n^*, \lambda]$, with respect to m_n. Now consider m_n to be an $n+3$ dimensional vector,

$$m_n = \{L, p_1, p_2, \theta_0, ..., \theta_{n-1}\},$$

where p_1 denotes the x-coordinate of p and p_2 denotes the y-coordinate. Then, using a forward difference scheme, the partial derivative of $Q_\epsilon[m_n, m_n^*, \lambda]$, with respect to m_n^i, is given by

$$\frac{\partial Q_\epsilon[m_n, m_n^*, \lambda]}{\partial m_n^i} = \lim_{\delta \to 0} \frac{Q_\epsilon[m_n + \delta e_i, m_n^*, \lambda] - Q_\epsilon[m_n, m_n^*, \lambda]}{\delta},$$

where e_i denotes the vector with the i'th element equal to one and the rest equal to zero. This partial derivative can be approximated numerically by choosing a small $\delta > 0$ and then calculating

$$\frac{\partial Q_\epsilon[m_n, m_n^*, \lambda]}{\partial m_n^i} \approx \frac{Q_\epsilon[m_n + \delta e_i, m_n^*, \lambda] - Q_\epsilon[m_n, m_n^*, \lambda]}{\delta}.$$

From this formula we have an approximation of all the partial derivatives, and this can in turn be used to obtain an approximation of the gradient, with respect to m_n.

An important consideration here is the size of the step δ. Choosing a too small δ, when using floating-point arithmetic for calculations, will yield a large rounding error. In fact, the forward difference scheme is ill-conditioned [10] and a too small step will give a value of zero due to cancellation [11]. For larger values of δ the slope of the secant line will be more accurately approximated, but the estimate of the slope of the tangent, using the slope of the secant, will be worse. An idea presented in [12] is to use a δ that is proportional to the variable that we differentiate with respect to. For our problem this would amount to choosing $\delta = km_n^i$, where the proposed value of k is the square root of the machine epsilon for double precision ($2.2 \cdot 10^{-16}$) [12]. This value of δ is small, but does not produce a large rounding error [12]. However, for our problem, the only variable that is related to the size of the functional value is the length L. The size of the angles in Θ_n or the starting point p is not related to the size of the functional. Therefore we choose to scale δ by L, for all decision variables, which gives us the value used in this implementation, $\delta = 1.5L \cdot 10^{-8}$.

4.3 The quadratic penalty method

The method described in section 4.1 considers λ to be a fixed parameter. It follows that we have to decide which values λ should take. As the penalty terms in our objective function represent hard constraints, we must choose a sufficiently large λ to ensure that the constraints are not violated. However, if the value of λ is too big, the penalty terms will completely dominate the problem, and the minimization of the actual objective function will be very slow. To solve this we use the procedure presented in [9, Framework 17.1, p. 501] of first choosing an initial λ, and then, after finding the minimizer m_n^* of $Q_\epsilon[m_n, \lambda]$, increase λ until the constraints are adequately honored by m_n^*. In this application, the updating of λ is done by first choosing an initial λ_0 and then update λ iteratively by

$$\lambda_{i+1} = 10\lambda_i.$$

To determine when to stop the updates and terminate the method, we need a measure of how much the current solution violates the constraints. A natural way to measure this, for the quadratic penalty method, is to consider the sum of the squared values of the violation of the constraints,

$$c_1(\Theta_n)^2 + c_2(\Theta_n)^2.$$

For a given threshold C, we can then state a condition that needs to be satisfied in order for the solution to be accepted, namely

$$c_1(\Theta_n)^2 + c_2(\Theta_n)^2 < C. \tag{4.3}$$

When this condition is satisfied we stop the updating of λ and return the solution.

4.4 Möbius energy

As mentioned in section 3.3.3, we will be using a finer grid when calculating the Möbius energy in each iteration. The purpose of this is to make sure that we never accept a step leading to a self-intersecting curve. To achieve this we make sure that the Möbius energy of any self-intersecting curve is larger than the current objective value. This way, the Wolfe conditions will never be satisfied for a self-intersecting curve.

To make sure that the value of the Möbius energy is larger than the objective value, we must determine a lower bound for the Möbius energy in the case of an intersection.

Such a lower bound could be the contribution to the Möbius energy for a single pair of points on γ_{m_n}. Now, assume that we have a self-intersection and let $\gamma_{m_n}(t_i), \gamma_{m_n}(t_{i+1})$, $\gamma_{m_n}(t_j), \gamma_{m_n}(t_{j+1})$ be the points closest to the intersection as displayed in Figure 4.1. The pairs of points among these four that will give a contribution to the Möbius energy are

$$B = \{(\gamma_{m_n}(t_k), \gamma_{m_n}(t_l)) \mid k \in \{i, i+1\},\ l \in \{j, j+1\}\}.$$

In order to find a lower bound, we now consider the pair in B that gives the largest contribution to the Möbius energy. The contribution from this pair will be the value of the integrand

$$J_{i,j} = \begin{cases} \dfrac{1}{|\gamma_{m_n}(t_i) - \gamma_{m_n}(t_j)|^8} - \dfrac{1}{d_{\gamma_{m_n}}(\gamma_{m_n}(t_i), \gamma_{m_n}(t_j))^8} & \text{if } i \neq j \\ 0 & \text{else,} \end{cases}$$

multiplied by L^8, and divided by n^2 as we apply the trapezoidal rule twice. Here it should be noted that in the case of an intersection we expect the first term of the integrand to be much larger than the second term, for at least one of the pairs in B. For the discretized curve it is theoretically possible to have a small geodesic distance between all pairs in B, but with a large number of discretization points we should in all practical applications expect the geodesic distance between at least one pair to be more than $L/2$. In comparison, a bound for the Euclidean distance between a pair of consecutive points is

$$|\gamma_{m_n}(t_i) - \gamma_{m_n}(t_{i+1})| \leq \frac{L}{n},$$

and so the the maximal Euclidean distance between the elements of a pair in B is $\sqrt{2}L/n$. We therefore expect the ratio between the first and second term of the integrand to be of size $(n/2\sqrt{2})^8$ for at least one pair of points. Hence, for a fairly large n it should be safe to neglect the contribution from the second term, when finding a lower bound.

Now, focusing only on the first term of the integrand, we wish to determine the minimum value of this term for a pair of points in B. This is obtained when the elements of a pair in B have the maximum possible distance between each other, which we recall is $\sqrt{2}L/n$. The first term of the integrand will then evaluate to

$$\left(\frac{n}{\sqrt{2}L}\right)^8 = \frac{n^8}{16L^8},$$

which gives us the contribution to the Möbius energy,

$$\frac{n^8}{16L^8} \cdot \frac{L^8}{n^2} = \frac{n^6}{16}.$$

This is our lower bound for the Möbius energy in the case of an intersection. In order to make sure that this value is larger than the objective value, we have to make sure that n is sufficiently large. By adding k interpolation points between every pair of consecutive points $(\gamma(t_i), \gamma(t_{i+1}))$, as described in section 3.3.3 and displayed in Figure 4.2, we increase n by a factor of $k+1$. This again increases the lower bound by a factor of $(k+1)^6$. In order to ensure that this lower bound is larger than the current objective value we have to ensure that for an objective value K we have

$$\frac{(k+1)^6 n^6}{16} > K,$$

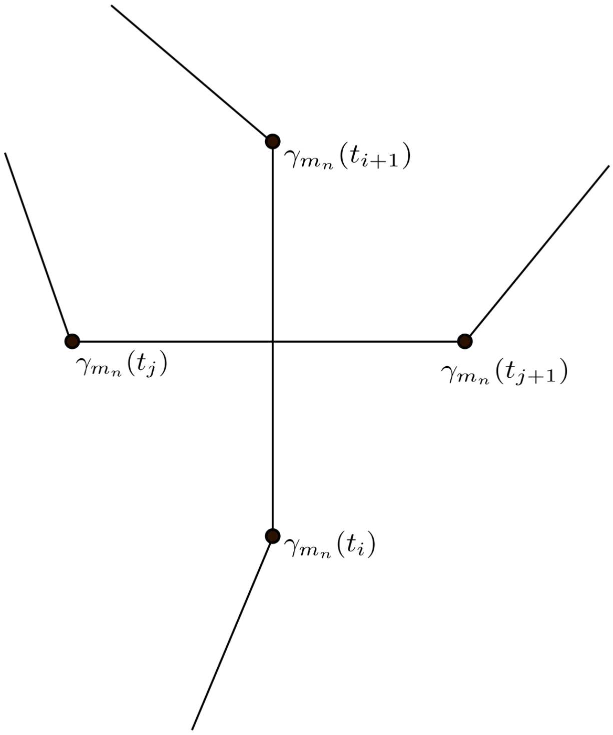

Figure 4.1: A self-intersection of the discretized curve γ_{m_n}, where the crossing line segments are perpendicular.

or equivalently that

$$k > \frac{\sqrt[6]{16K}}{n} - 1.$$

Thus, in our implementation k will be set to the smallest integer value that satisfy this inequality. As one can observe, the exponent ensures that even for very large objective values K, we do not expect k to be infeasibly large when considering computational complexity. This is why we wanted the steeper version of the Möbius energy, as described in section 2.2.

A possible implementation issue when using this approach is rounding errors when the distance between consecutive points gets very low. Let W_i denote the set of the k interpolation points between $(\gamma_{m_n}(t_i), \gamma_{m_n}(t_{i+1}))$ for a given i. These points could be very close to each other if k is large. However, all of these points lie on a straight line, so the integrand should evaluate to zero. In order to avoid rounding errors we will therefore set the integrand to zero explicitly for all pair of points contained in the same set W_i.

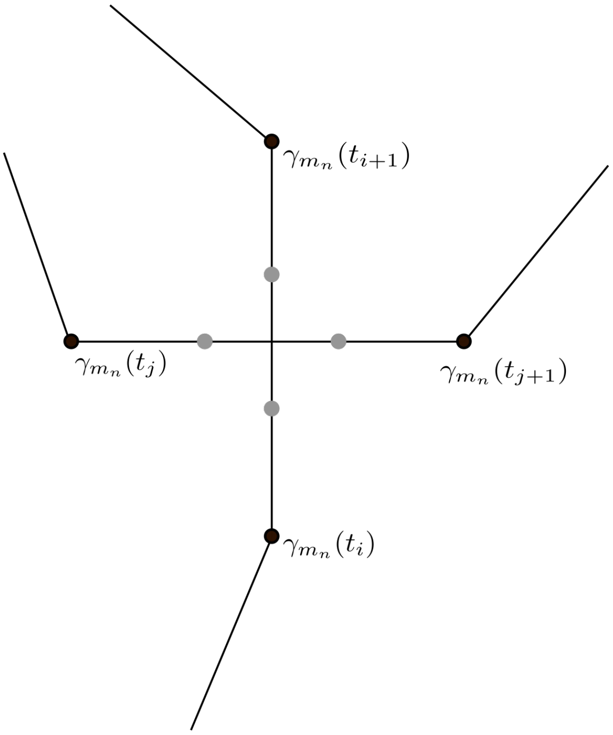

Figure 4.2: A self-intersection of the discretized curve γ_{m_n} with $k = 2$ extra interpolation points between each pair of consecutive points. The points in black are the original points on γ_{m_n}, while the points in grey are the extra interpolation points that will be added before calculating the Möbius energy.

4.5 Iterative Tikhonov-Morozov method

When the algorithm in 4.1 terminates for a λ such that the reconstructed solution satisfies the constraint condition in (4.3), the method terminates and returns the reconstructed solution. This solution is expected to be a better guess of the true solution than the initial guess. It therefore makes sense to assume that if we update the initial guess to the current solution, and run the method again, we would obtain better results. This approach is known as the iterative Tikhonov–Morozov method [13]. Assume we repeat the updating procedure for a fixed number of r times, for a fixed regularization parameter β, this will correspond to using a regularization parameter β/r in one iteration. The idea behind this method is that we can use a stronger regularization in each iteration, and in this sense solve an easier problem, as the quadratic term is more dominant in the objective functional.

A natural question is why we do not run this method all the way to convergence. A simplified explanation is that, ignoring the constraints of our problem and the Möbius penalty term, we have a problem of the form

$$\min_u \frac{1}{2}||F(u) - v||^2 + \frac{1}{2\tau}||u - u_k||^2$$

in each step. The solution u_{k+1} to this problem will satisfy

$$F'(u_{k+1})^*(F(u_{k+1}) - v) + \frac{1}{\tau}(u_{k+1} - u_k) = 0,$$

or equivalently

$$\frac{1}{\tau}(u_{k+1}) - u_k) = -F'(u_{k+1})^*(F(u_{k+1}) - v).$$

This can be viewed as using the implicit Euler method for approximating a flow of the form

$$\partial_t u = -F'(u)^*(F(u) - v).$$

If we assume that for sufficiently small steps the explicit and implicit Euler method behave similarly, we can use the explicit Euler method to get an idea of how the method will behave. A step with the Explicit Euler method is given from the optimality condition

$$F'(u_k)^*(F(u_k) - v) + \frac{1}{\tau}(u_{k+1} - u_k) = 0,$$

which results in the step

$$u_{k+1} = u_k - \tau F'(u_k)^*(F(u_k) - v).$$

We recognize this as the gradient descent method for the residual

$$\frac{1}{2}||F(u) - v||^2.$$

The gradient descent method tries to find a critical point (in general a minimum) of the least squares problem, ideally a solution of the equation $F(u) = v$. In the presence of noise, however, this equation will in general have no solution. As we have presence of measurement and integration errors, trying to run the method until convergence will make it diverge. We expect that our method behaves similarly, and this is why we do not want to run the iterations all the way to convergence.

4.6 Specific algorithm

Altogether we obtain the method described by the pseudo-code in Algorithm 1.

Algorithm 1: The full method

```
Set $m_n^0 = (p^0, L^0, \Theta_n^0)$ as initial guess;
Set $m_n^* = m_n^0$ as reference solution;
$\lambda = \lambda_0$;
repeat $r$ times
    Compute $m_n^{\min}$ as the minimizer of $Q[\cdot, m_n^*, \lambda]$, with the BFGS method;
    while $m_n^{min}$ violates the constraints do
        $\lambda = 10\lambda$;
        Set $m_n^{\min}$ as initial guess;
        Compute $m_n^{\min}$ as the minimizer of $Q[\cdot, m_n^*, \lambda]$, with the BFGS method;
    end
    Set $m_n^0 = m_n^{\min}$ as initial guess;
    Set $m_n^* = m_n^{\min}$ as reference solution;
    $\lambda = \lambda_0$;
end
```

4.6.1 Computational complexity

Having defined the discretization and algorithm we can determine the computational complexity of the method. To do this we first consider the complexity of a single iteration of the BFGS-method. In each step of the BFGS-method the most expensive operation is calculating the numerical gradient, which is done at least one time, and potentially many times, when searching for a step that satisfies the Wolfe conditions. To calculate the numerical gradient we must calculate the objective functional $\mathcal{O}(n)$ times. Finding the first term of the objective functional involves calculating γ_{m_n} from m_n ($\mathcal{O}(n)$), and from γ_{m_n} calculating the Radon transform for all $\alpha's$ ($\mathcal{O}(mn)$). Computing the Bernoulli energy term is of complexity $\mathcal{O}(n)$, while calculating the Möbius energy term is of complexity $\mathcal{O}(n^2(k+1)^2)$. Thus, calculating the objective functional have a computational complexity of $\mathcal{O}(n + mn + n + n^2(k+1)^2) = \mathcal{O}(mn + n^2(k+1)^2)$, which gives us a complexity of $\mathcal{O}(mn^2 + n^3(k+1)^2)$ for the calculation of the numerical gradient. The computational complexity of one iteration of the BFGS method is expected to be of this size, but the number of calculations of the numerical gradient, that are needed to find a step that satisfies the Wolfe conditions, is dependent on the specific problem and iterate. The total number of performed iterations of the BFGS method is naturally also dependent of the specific problem.

4.6.2 Programming language

The algorithm has been implemented in Python 3.8, a link to the Github repository is provided in the beginning of Chapter 5. To speed up the computations, the Numba package was used. This package translates Python functions to optimized machine code at runtime using the industry-standard LLVM compiler library [14]. Numba-compiled numerical algorithms in Python can approach the speeds of C or FORTRAN [14].

Chapter 5

Numerical experiments

In the numerical experiments we test how the implemented method performs for different data sets and reference solutions. We test how the method performs for a range of values for the regularization parameter, and how different sets of angles affect the reconstructed solution. In addition, we test how the method performs when the data is perturbed by random noise. The code is available in the Github repository `https://github.com/ErikBoee/ShapeAnalysisForInverseProblems.git`, for anyone interested in either verifying the experiments presented here, or performing their own.

5.1 General setup for the experiments

5.1.1 Discretization points

Throughout the experiments we will use a fixed grid for the t- and α-domain. In order to define the grids we need to decide values for the number of discretization points, n and m. As we have seen, the errors in the functional resulting from integration errors are expected to scale with L/n and L^2/m in a worst case scenario, while the computational complexity contains a term proportional to mn^2 and a term proportional to n^3. Thus, the error terms are of the same order, but the error term from the integration over α has a higher proportionality constant. It therefore makes sense to choose n and m to be of the same order, but with m larger than n. This is also consistent with the fact that the computational complexity of the problem scales better with m.

The first parameter we consider is the number of points in the discretization of the t-domain $[0, 1]$. Naturally, we would like this parameter to be large enough to avoid significant errors from the numerical integration, but we also have to consider the computational complexity and decide on a value that do not make the running time of the method infeasibly high. As mentioned in section 4.6.1, there are some factors that make it hard to get an accurate estimate of the running time, but we expect it to be $\mathcal{O}(mn^2 + n^3)$. It therefore makes sense to perform some tests with low values of n, and make an estimate of the running time for larger values, based on how we expect the running time to scale when increasing n. As discussed in the former paragraph we want m to be of the same order as, but larger than, n, we therefore set $m = 3n$ in the experiments, so that the expected running time is $\mathcal{O}(n^3)$.

After some initial testing, setting the number of points in the discretization to be $n = 100$ appeared to be the best option. The running time for solving a problem with this value was between 10 and 48 hours. Increasing n significantly above this value would yield infeasibly large running times, and even though the error bound of $L/2n$ is not very low in a worst case scenario we do expect the error to scale with L/n^2 for most curves.

Next, we must determine a domain and discretization of the α's. In order to avoid inverse crimes, as described in [2], where the data is generated by the same method as what is used in the reconstruction process, we wish to use Python's built-in function for the Radon transform to generate the data. For an image with $m \times m$ pixels, this function discretizes and determines the α-domain to be m evenly spaced points, where the distance from the first to the last point is equal to the width and height of the image. Here the height and width of a pixel is considered to be one, such that the image has height and width equal to m. The center point of the domain corresponds to the line, in direction σ_k, passing through the center pixel of the image. As we wish to compare the Radon transforms of our reconstructed solution with this data, we choose the same domain and discretization for the α's in our experiments. As earlier mentioned, the value of m was set to $3n = 300$. In addition to the forementioned error considerations, it is important to have a sufficiently fine grid in order for the Radon transforms from the true solution to accurately describe the shape. The built-in function in Python counts pixels, and if the grid is coarse this could deviate much from what is found by the analytical expression in (2.2).

5.1.2 Heaviside approximation

As discussed in section 3.1 we use a smooth approximation for the Heaviside function, which leads to an error in the objective functional bounded by $\mathcal{O}(L\epsilon)$. On the other hand, the regularity of the functional increases with ϵ, and we want this value to be as large as possible, without imposing a significant error. We therefore want ϵ to be of the same order as $1/n$ and $1/m$. A choice of $\epsilon = 7 \cdot 10^{-3}$ satisfies this requirement and was used in the experiments.

5.1.3 Convergence of the method

To determine when to terminate the method we need to decide the value of four parameters, the tolerance, T, and number of consecutive iterations, D, for terminating the BFGS method, the tolerance, C, for not increasing λ further, and the number of iterations, r, performed in the iterative Tikhonov-Morozov method. For the parameter C it makes sense to choose a value of the size ϵ^2, as ϵ can be viewed as the amount of blur in the curve. Thus, if the constraint violation is smaller than this it should not give a noticeable error in the results. The reason for choosing ϵ^2 is that C is a bound for the squared value of the constraint violation. In these experiments C is set equal to 10^{-6}. The values of T and D are based on observations from preliminary experiments, where the method was ran without a stopping criterion. Here it was observed that when the method made a relative improvement in the objective function of less than 10^{-5} for more than 5 consecutive iterations, we did not observe any noticeable further improvement when continuing to run the method. Therefore, T was set to 10^{-5} and D was set to 5. A similar procedure was used when deciding the value of r, a selection of $r = 10$ appeared to be small enough to avoid divergence of the method, and simultaneously large enough to ensure that further iterations did not give a significant improvement in the reconstructed curve. Here it should be mentioned that the optimal number of iterations is dependent on the specific problem, and the regularization parameter β. For $\beta = 0$ it does not make sense

Fixed parameters	
ϵ	7×10^{-3}
λ_0	100
r	10
c_1	1×10^{-3}
c_2	0.9
T	1×10^{-5}
D	5
C	1×10^{-6}
n	100
Pixels	300

Table 5.1: Values of the fixed parameters in the numerical experiments.

to update the reference solution at all, as it does not affect the solution, while with strong regularization it would probably make sense to run more iterations. For further work it would be interesting to look at more intelligent methods for determining the number of iterations, instead of using a fixed number.

5.1.4 Fixed parameters

In addition to the forementioned parameters, some additional parameters will be fixed for all experiments. These parameters are λ_0, used in the first iteration of the quadratic penalty method, and c_1, c_2 used in the Wolfe conditions, when determining an acceptable step length. The values of the fixed parameters are listed in Table 5.1.

5.1.5 Construction of experiments

To construct the numerical experiments we first define the triplet, m_n^{sol}, describing the boundary curve of the shape we wish to reconstruct. This boundary curve will later be referred to as the true solution. From the boundary curve we construct an image with 300×300 pixels that contains the characteristic function describing the shape. Then, for a given set of angles, the built-in Python function generates the Radon transforms. A visualization of how a given angle is related to the direction of the Radon transform is presented in Figure 5.1. The Radon transforms are then perturbed by random noise. To generate this noise, for a particular Radon transform f, we first find the maximum element of the Radon transform, i.e.,

$$f_{\max} = \max_{\alpha} f(\alpha).$$

Then, for each α, $f(\alpha)$ is perturbed by a noise value drawn from a normal distribution with zero mean and standard deviation equal to $\mu f_{\max}$, for a noise parameter μ. Next, we define another triplet, m_n^{ref}, which represents the reference solution, i.e., the a priori knowledge of how we expect the boundary curve to look like. For all examples we let the initial guess of m_n, m_n^0, equal the reference solution, m_n^{ref}. Note that the position of the starting point p, and the length of the curve L is not assumed to be known. However, we should be able to get a reasonable estimate from the Radon transforms, and thus a good initial guess of these variables is not expected to influence the results. On the other hand, correct rotation of the initial guess is not assumed to be known either, and could be more difficult to estimate. We should therefore be careful when evaluating the performance of the method for the experiments with very good initial guesses. That said, this will be equal for every reconstruction, and so the comparison of the different reconstructions should be valid.

For each experiment we wish to test different sets of angles and different values of the regularization parameter β, and the noise parameter μ. We have chosen to look at four sets of angles, three sets of respectively $4, 8$ and 16 angles evenly distributed over the entire interval from 0 to 180 degrees, and one set with 5 angles evenly distributed over the more narrow interval from 67.5 to 112.5 degrees. We will refer to these sets as A_4, A_5, A_8 and A_{16}, where the subscript is equal to the number of angles. Each set gives different information about the shape. Here we would expect that for the sets with angles evenly distributed over the entire interval, a larger number of angles leads to better reconstructions. Furthermore, for A_5 we expect that there may be parts of the shape that have hardly any influence on the measurements. Thus the reconstructions are significantly more difficult with angles in A_5 than with angles in one of the other sets. For the noise parameter we have looked at the values 0 and 0.15, where the former corresponds to no noise, and the latter to a substantial amount of noise. For the regularization parameter we have considered, for each set of angles, the values $0N, 0.01N, 0.1N$ and N, where N represents the number of angles in the set. The reason for scaling with N is that when we increase the number of angles, we increase the number of fit-to-data terms in the objective functional. It therefore makes sense to adjust the regularization parameter in order to avoid too much change in the ratio between the regularization term and the sum of the fit-to-data terms. Here it should be noted that we always have the regularization from the Möbius energy penalty term, which prefers circles over more complicated forms, but this effect is expected to be rather weak.

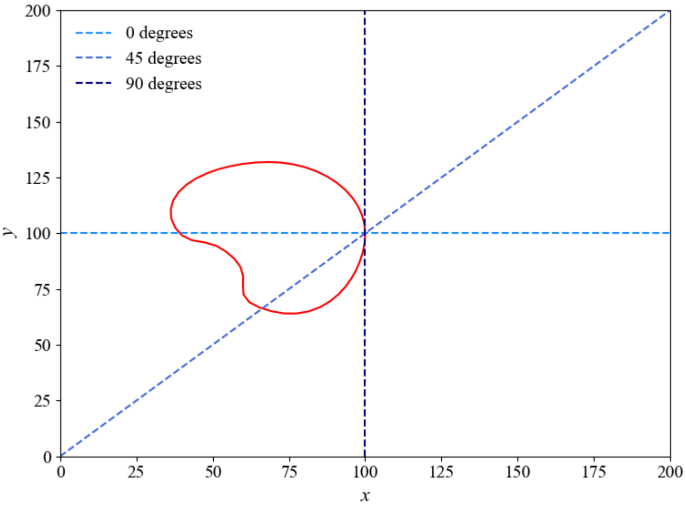

Figure 5.1: Directions of the Radon transform corresponding to the angles 0, 45 and 90 degrees.

5.1.6 Performance

In order to determine how well the method performs we wish to measure the difference of the reconstructed solution and the true solution. To do this we will define a metric on the space $\mathcal{M}$. We start by denoting the distance from a point $p \in \mathbb{R}^2$ to a triplet $m \in \mathcal{M}$ as the minimum distance from p to the curve defined by m, i.e.,

$$d_p(\gamma_m) = \min_{t \in [0,1]} |p - \gamma_m(t)|.$$

Assume further that we have two triplets $m_1, m_2 \in \mathcal{M}$, then we define the distance from γ_{m_1} to γ_{m_2} as the maximum distance from a point on γ_{m_1} to γ_{m_2}, i.e.,

$$d_{\gamma_{m_1}}(\gamma_{m_2}) = \max_{p \in \gamma_{m_1}} d_p(\gamma_{m_2}).$$

We are now ready to define the metric on $\mathcal{M}$ as,

$$d(m_1, m_2) = \max(d_{\gamma_{m_1}}(\gamma_{m_2}), d_{\gamma_{m_2}}(\gamma_{m_1})), \text{ for } m_1, m_2 \in \mathcal{M}.$$

In order to use this metric to measure the difference between the true solution m_n^{sol} and the reconstructed solution m_n^{rec}, we have to define a discretized version of this metric. This is done in a straight forward manner by considering the distance between each point on the discretized curves. This results in the following minimum distance from a point,

$$d_p(\gamma_{m_n}) = \min_{t \in \{t_0, \dots, t_n\}} |p - \gamma_{m_n}(t)|.$$

In the expression for distance between curves we interpret $p \in \gamma_{m_n}$ as $p = \gamma_{m_n}(t_i)$ for some $i \in \{0, ..., n\}$. To make this discretized version more accurate we use linear interpolation to add extra points between two consecutive points on γ_{m_n}, in the same manner as when calculating the Möbius energy. Since we only calculate this metric once for each experiment, we can afford to add many points. The number of points added between every pair of consecutive points was set to n, giving a total of n^2 points. It is worth mentioning that here we are only looking at pointwise differences and thus completely disregard the regularity of the solutions. Therefore, a reconstruction with a relatively small distance to the true solution might in practice be relatively poor due to its irregularity, which we will see examples of in the experiments.

5.2 Simple shapes

5.2.1 Summary of results

The first experiment we present contains rather simple shapes, which are modifications of a circle and displayed in Figure 5.2. The values of the metric described in section 5.1.6, for the different sets and parameters, are shown in Table 5.2. In terms of this metric we observe that for the sets A_4 and A_5 the best results were obtained with a regularization parameter of $0.1N$, regardless of whether the data contained noise. This was also the situation for the sets with more angles, A_8 and A_{16}, when the data contained noise. However, when there was no noise present, the best results for these sets were achieved with no regularization. This makes sense, as we expect more angles and less noise to make the reconstruction process more stable, which in turn should reduce the need for

Regularization	A_4	A_8	A_{16}	A_5	A'_4	A'_8	A'_{16}	A'_5
0.0	4.64	**2.14**	**1.64**	15.53	9.4	7.82	9.9	21.7
$0.01N$	3.22	2.25	1.8	13.21	8.42	6.71	7.14	17.03
$0.1N$	**2.68**	3.42	1.92	**11.26**	**7.37**	**3.42**	**5.65**	**10.6**
N	5.38	5.11	4.54	11.66	10.35	7.5	7.92	17.49

Table 5.2: A table showing the difference between the reconstructed solution m_n^{rec} and the true solution m_n^{sol}, measured in $d(m_n^{\text{rec}}, m_n^{\text{sol}})$, for the first experiment (shapes in Figure 5.2). The prime indicates that the data set contained noise.

regularization. It is also worth mentioning that the reconstructions with angles in A_5 give a much higher value than the other reconstructions. This could indicate that some parts of the true solution have hardly any influence on the Radon transform from these angles, which makes it hard to reconstruct. This will be discussed in more detail in section 5.2.3.

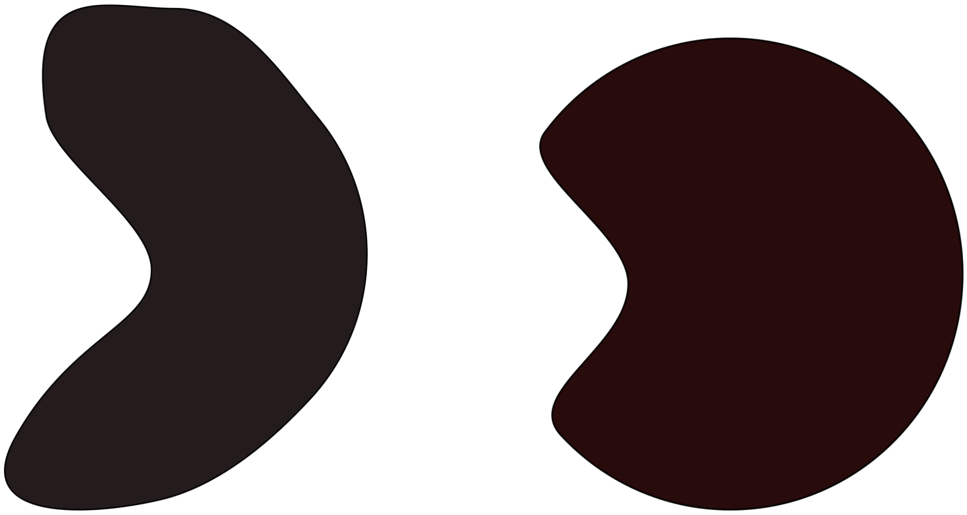

Figure 5.2: On the left we see the true solution from which the Radon transforms are generated, on the right we see the reference solution.

5.2.2 Importance of regularization when noise is present in the data

The data in Table 5.2 tells us the furthest distance from a point on one of the curves to the other curve. To make a more thorough evaluation of the method we want to study the reconstructions. A visualization of the reconstructions with angles in A_8 and no noise is presented in Figure 5.3. As one can observe, the reconstruction with no regularization matches the overall shape of the true solution well. However, in the rightmost part of the curve we observe that the reconstruction is far less smooth than the true solution. In the other reconstructions we see that the curves are smooth, but the overall shape matches the true solution more poorly as the regularization parameter increases.

In Figure 5.4 we show a visualization of the same reconstructions, with angles in A_8, but this time with noisy data. As one can observe the reconstruction with no regularization is in this case highly irregular. Using the weakest regularization, $\beta = 0.08$, we observe that the reconstruction is far more smooth, but approximates the true solution poorly in some regions. When $\beta = 0.8$ we observe a much better reconstruction, as the overall shape still matches the true solution well, and the local errors are far less apparent. In the last reconstruction the regularization again seem to be too strong, as the overall shape does not match the true solution as well as for the other reconstructions.

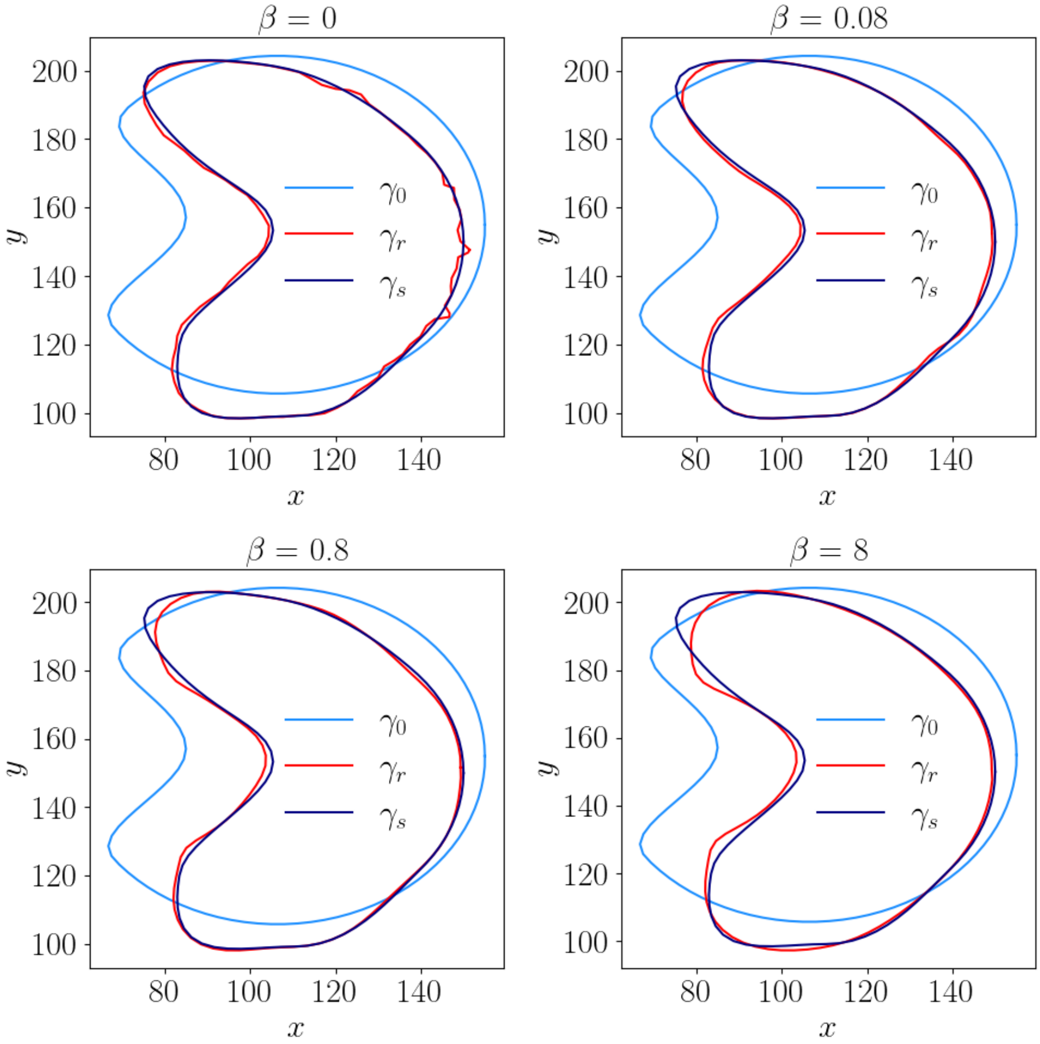

Figure 5.3: A plot showing the reconstructed curves for different values of the regularization parameter. In this plot we see the reconstructed curves for angles in A_8 and data containing no noise. The curves presented in each subplot are the true solution, γ_s, the reconstructed solution, γ_r, and the initial guess, γ_0, which coincides with the reference solution. We observe that most of the reconstructions are very good, but with no regularization we see some irregularity in the rightmost part of the reconstructed curve. For the strongest regularization the overall fit is not as good as for the other reconstructions.

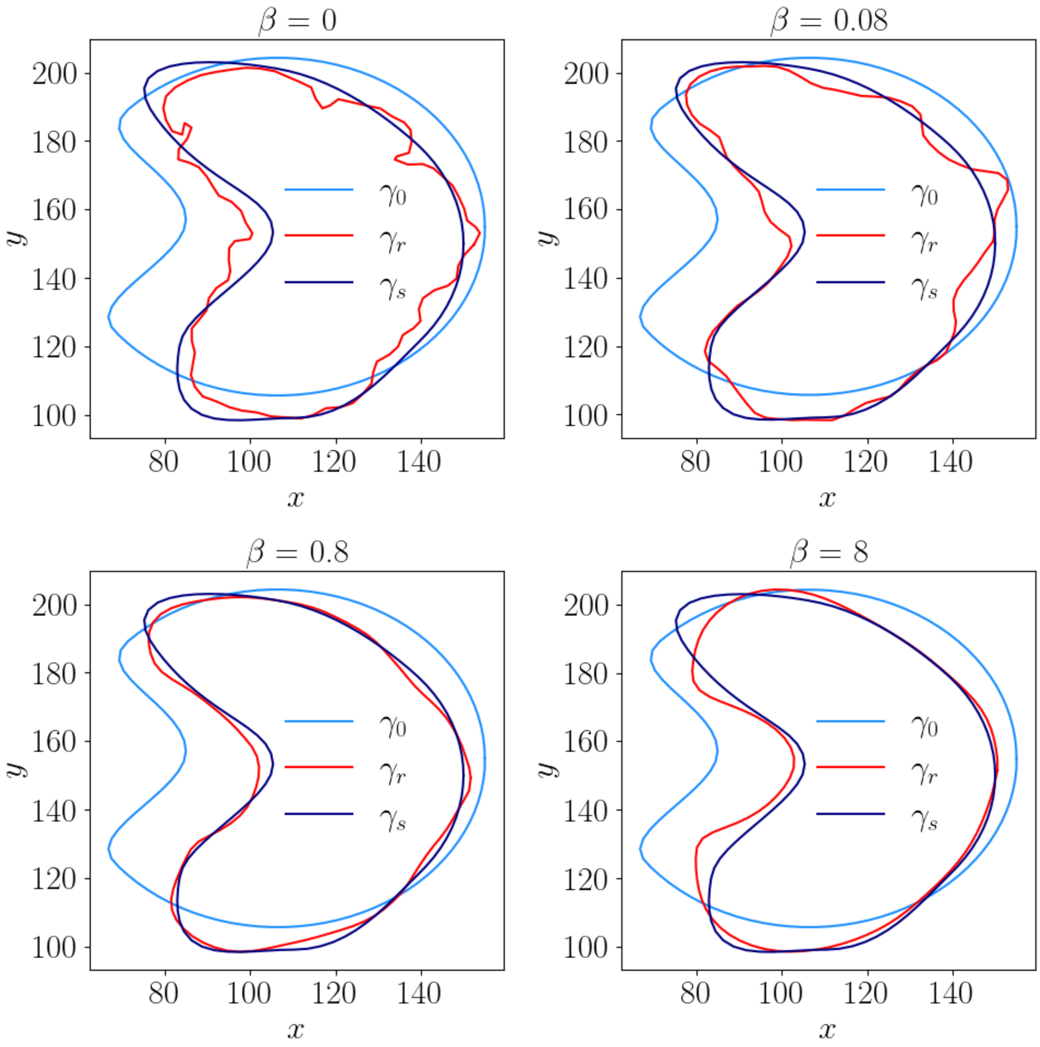

Figure 5.4: A plot showing the reconstructed curves for different values of the regularization parameter. In this plot we see the reconstructed curves for angles in A_8 and data containing noise with $\mu = 0.15$. The curves presented in each subplot are the true solution, γ_s, the reconstructed solution, γ_r, and the initial guess, γ_0, which coincides with the reference solution. Here we observe that the reconstructions with some amount of regularization are much more robust to noise in the data, with the best reconstruction obtained for $\beta = 0.8$.

5.2.3 Difficulties with angles in A_5

In the reconstructions using the angle sets A_4, A_8 and A_{16} similar results were observed, in the sense that all reconstructions matched the overall shape of the true solution quite well, and the best reconstructions corresponded to the lowest values in Table 5.2. For the reconstructions using angles in A_5, on the other hand, we did not observe good reconstructions, a visualization is displayed in Figure 5.5. None of the reconstructions match the overall shape of the true solution very well, which indicates that the available Radon transforms do not give enough information about the object. Recall that all angles in A_5 are close to 90 degrees, which could make some parts of the curve difficult to reconstruct. In figure 5.6 we present the Radon transforms for the true solution and the reconstructed solution, with $\beta = 0.05$. As one can observe, the Radon transforms of the reconstructed curve is not very different from the Radon transforms of the true solution, and some of the differences between the curves appears to be hard to discover based on these Radon transforms.

To investigate this hypothesis further, the same experiment was conducted, but with both the true solution and the initial guess rotated 90 degrees. The reconstructions are presented in Figure 5.7. As one can observe, the reconstructions are much better than in the former experiment, especially when weak regularization is utilized. This indicates that most of the information about these shapes was in Radon transforms corresponding to angles that were not included in A_5 for the original experiment. In Figure 5.8 we present the original and rotated solution, along with the directions corresponding to the angles in A_5.

From a purely theoretical view, if we have angles in an arbitrary small interval $(-l, l)$ for some $l > 0$, a perfect reconstruction is possible [2]. In practice, on the other hand, it is observed that when the angle of view decreases from 180 degrees towards zero, the reconstruction problem becomes increasingly ill-posed and makes reconstruction very challenging [2]. However, in chapter 9.2 of [2] they state that some features of the shape are expected to be reconstructed quite well, and roughly speaking the boundaries that are tangential to the directions in A_5 should be reconstructed well.

It is hard to tell from Figure 5.8 if the directions are closer to being tangential to the boundaries, of the solution, in the rightmost plot. However, we do observe that the reconstructions of the original solution in Figure 5.5 was especially poor on the left hand side of the shape, where the boundaries are not tangential to any of the angles. Another explanation, for why the reconstructions of the rotated solution are better, could be that the entire shape is seen from directions in A_5. In comparison, the original solution has a concave part that is not seen from any of the directions in A_5.

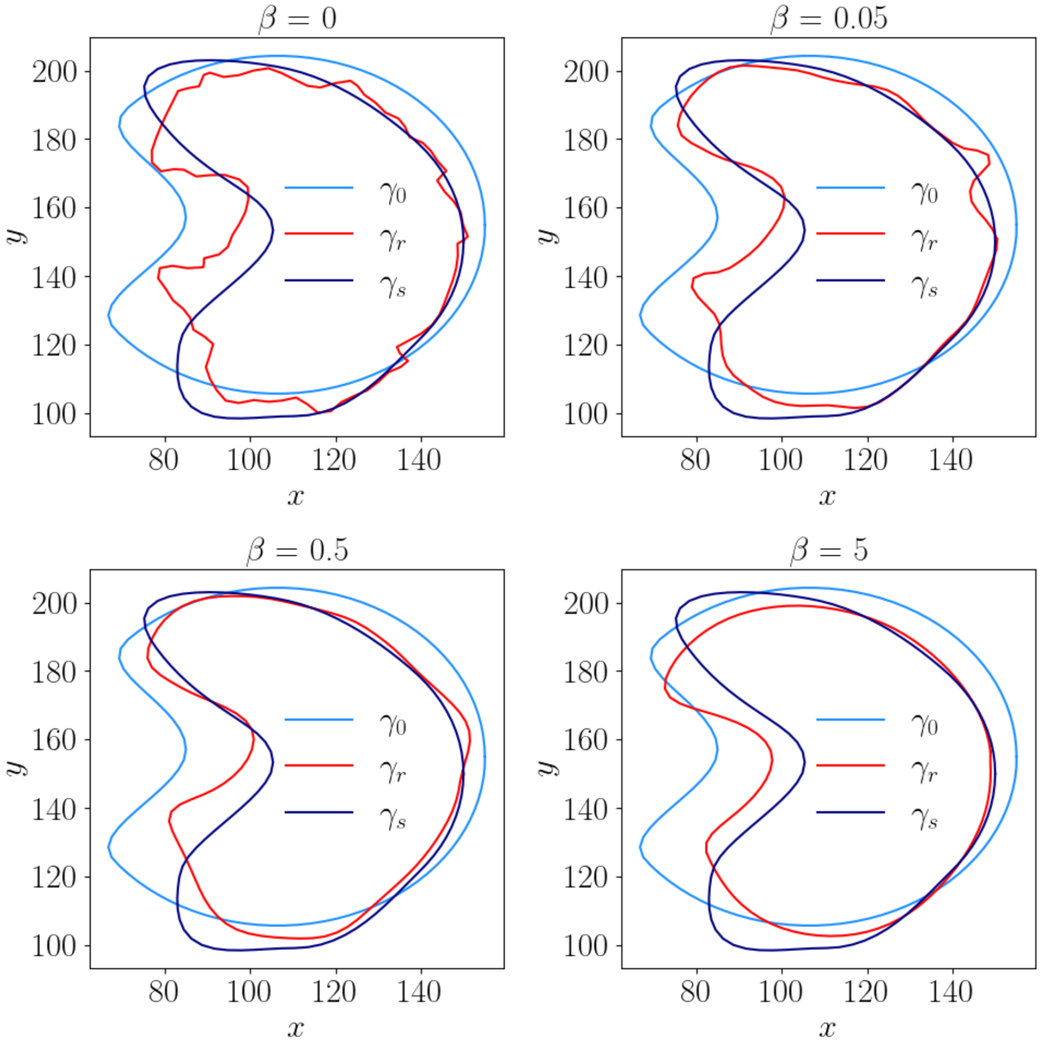

Figure 5.5: A plot showing the reconstructed curves for different values of the regularization parameter. In this plot we see the reconstructed curves for angles in A_5 and data containing no noise. The curves presented in each subplot are the true solution, γ_s, the reconstructed solution, γ_r, and the initial guess, γ_0, which coincides with the reference solution. We obtain poor reconstructions, due to angles in the narrow interval from 67.5 to 112.5 degrees. The reconstructions obtained with some regularization are smoother.

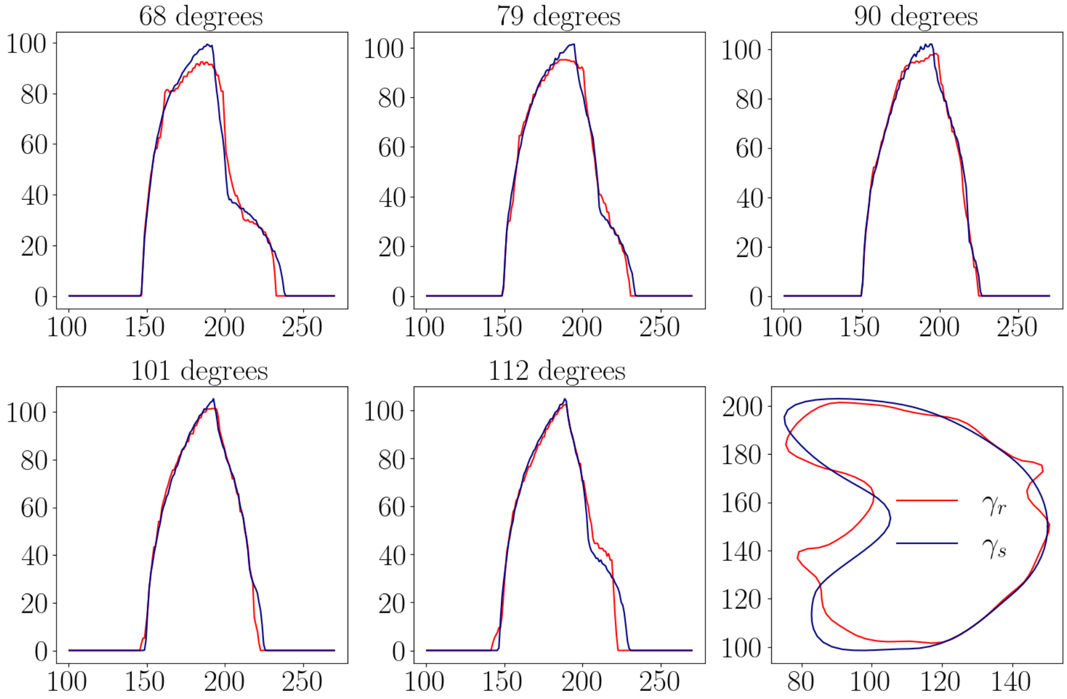

Figure 5.6: A visualization of the true solution, the reconstructed solution and the Radon transforms, when there is no noise in the data, the angles are in A_5, and we have $\beta = 0.05$. In the bottom right plot we see the reconstructed curve, γ_r, in red and the true solution, γ_s in blue. In the rest of the plots we see the Radon transforms corresponding to the angle in the title. The red graphs represent the Radon transforms of the reconstructed curve, while the blue graphs represent the Radon transforms of the true solution. Here we observe that the Radon transforms of the reconstructed solution are not too different from the Radon transforms of the true solution. However, the reconstruction is still poor.

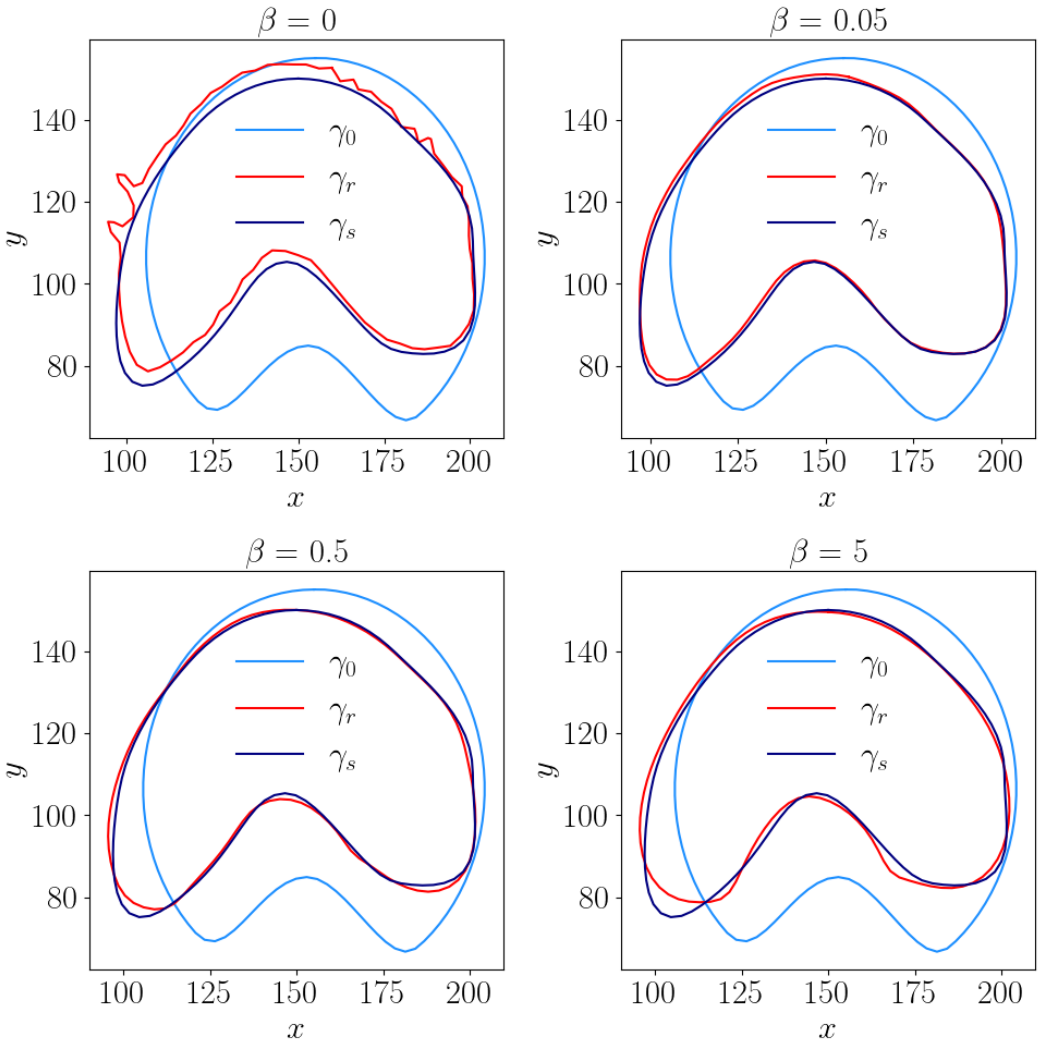

Figure 5.7: A plot showing the reconstructed curves for different values of the regularization parameter. In this plot we see the reconstructed curves for angles in A_5 and data containing no noise. The curves presented in each subplot are the true solution, γ_s, the reconstructed solution, γ_r, and the initial guess, γ_0, which coincides with the reference solution. Here we see rather good approximations even though we have angles in the narrow interval from 67.5 to 112.5 degrees, however, the reconstruction with no regularization creates some spurious oscillations at certain parts of the boundary.

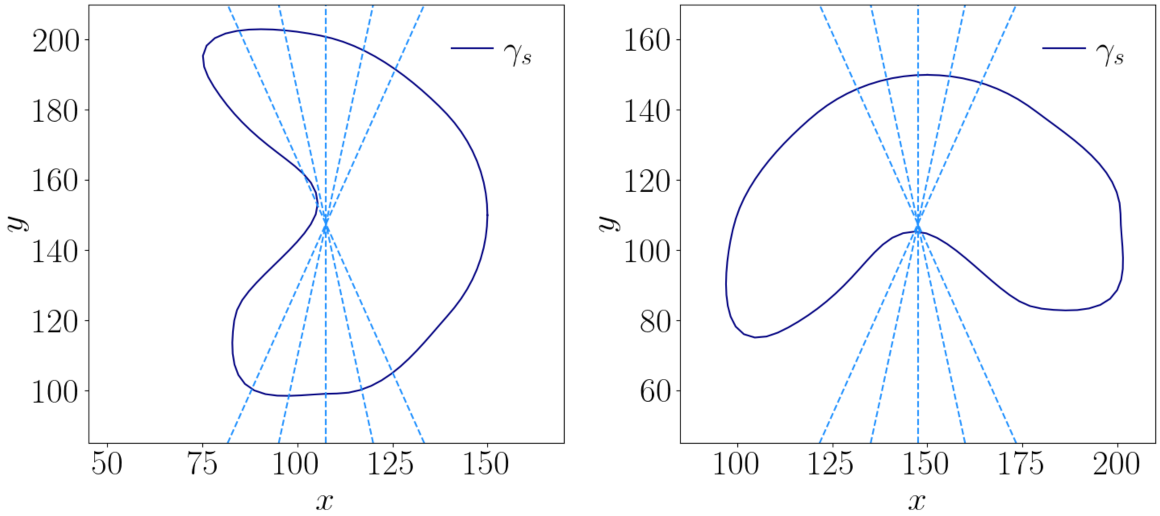

Figure 5.8: A plot showing the original solution of the experiment on the left, and the rotated solution on the right. In both plots the directions corresponding to the angles in A_5 are included.

5.3 More complicated shapes

In the next experiment we look at more complicated shapes. The reference solution is still close to the true solution, and yields a good initial guess, the shapes are shown in Figure 5.9. In Table 5.3 we present the results. For the experiments with no noise, we see that regularization only improved the reconstructions when the available Radon transforms corresponded to angles in A_4. For the sets with more angles the reconstructions were stable enough without any regularization. However, when the data contained noise we observe that for all angle sets, the best reconstructions were obtained with some degree of regularization. It is also interesting to see that when the data sets contained noise the results was far better with regularization, while the results without regularization was only marginally better for the data sets without noise.

As mentioned earlier, the metric in Table 5.3 does not describe the reconstructions fully, and it would be interesting to see what reconstructions led to these results. In Figure 5.10 we present the reconstructions for angles in A_{16} without noise. As one can observe, the points furthest from the other line appear to be at the bottom left corner in each reconstruction. Except from this part of the curves, the reconstructions are good, and the ones obtained with regularization are smoother than the one obtained without regularization. Next, we look at the reconstructions when there is noise present in the data. The reconstructions are presented in Figure 5.11. Here we clearly see that the reconstructions obtained with weak or no regularization are terrible. The reconstructions with stronger regularization are much better. However, we see that the curvature of these reconstructions look very much like the initial guess, and could be dependent on a good initial guess. To investigate this further, we perform an experiment with a very poor initial guess.

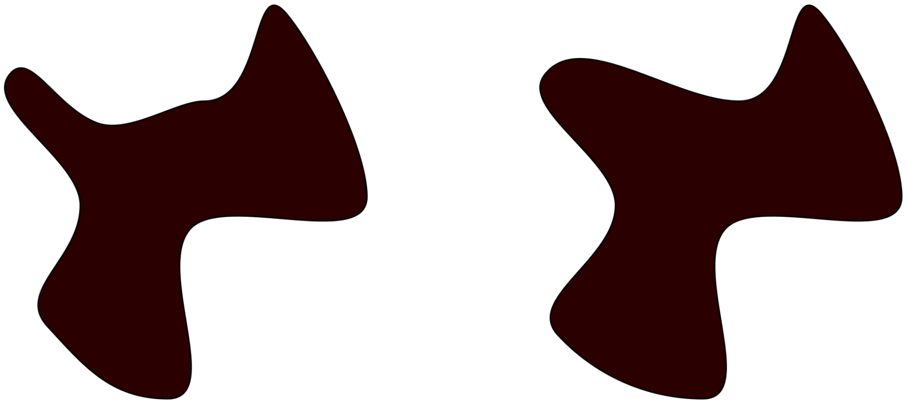

Figure 5.9: On the left we see the true solution from which the Radon transforms are generated, on the right we see the reference solution.

Regularization	A_4	A_8	A_{16}	A_5	A'_4	A'_8	A'_{16}	A'_5
0.0	5.21	**1.99**	**1.87**	**2.66**	11.42	13.58	14.55	12.13
$0.01N$	3.24	3.82	3.15	3.34	9.31	**5.29**	9.66	9.77
$0.1N$	**3.04**	2.16	2.09	4.44	**5.87**	5.85	6.17	10.45
N	4.76	3.25	3.01	5.44	6.58	5.96	**4.99**	**7.42**

Table 5.3: A table showing the difference between the reconstructed solution m_n^{rec} and the true solution m_n^{sol}, measured in $d(m_n^{\text{rec}}, m_n^{\text{sol}})$, for the second experiment (shapes in Figure 5.9). The prime indicates that the data set contained noise.

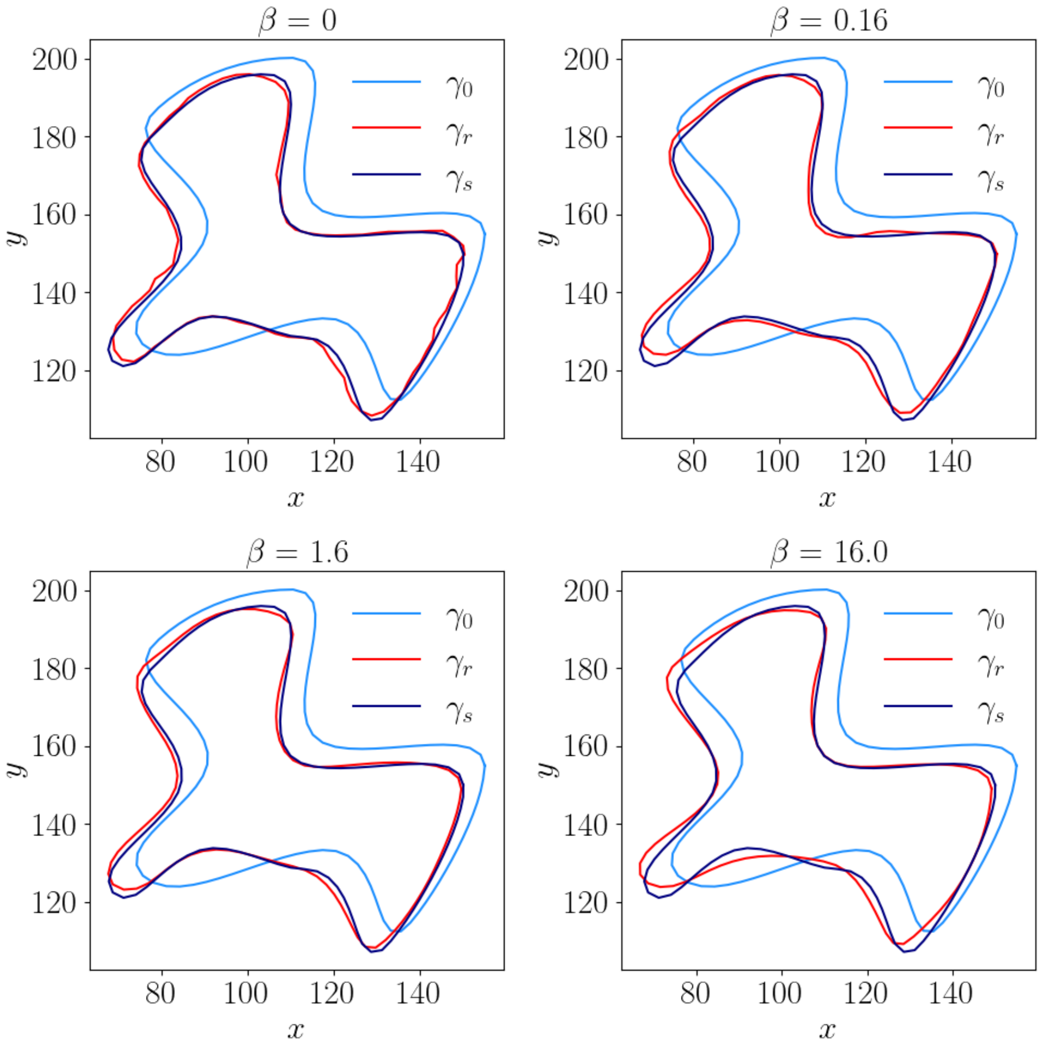

Figure 5.10: A plot showing the reconstructed curves for different values of the regularization parameter. In this plot we see the reconstructed curves for angles in A_{16} and data containing no noise. The curves presented in each subplot are the true solution, γ_s, the reconstructed solution, γ_r, and the initial guess, γ_0, which coincides with the reference solution. Here we observe the reconstructions of a more complicated shape. The reconstructions are rather good, but we observe some irregularity in the reconstruction with $\beta = 0$, and the regularization with $\beta = 16$ appears to be too strong.

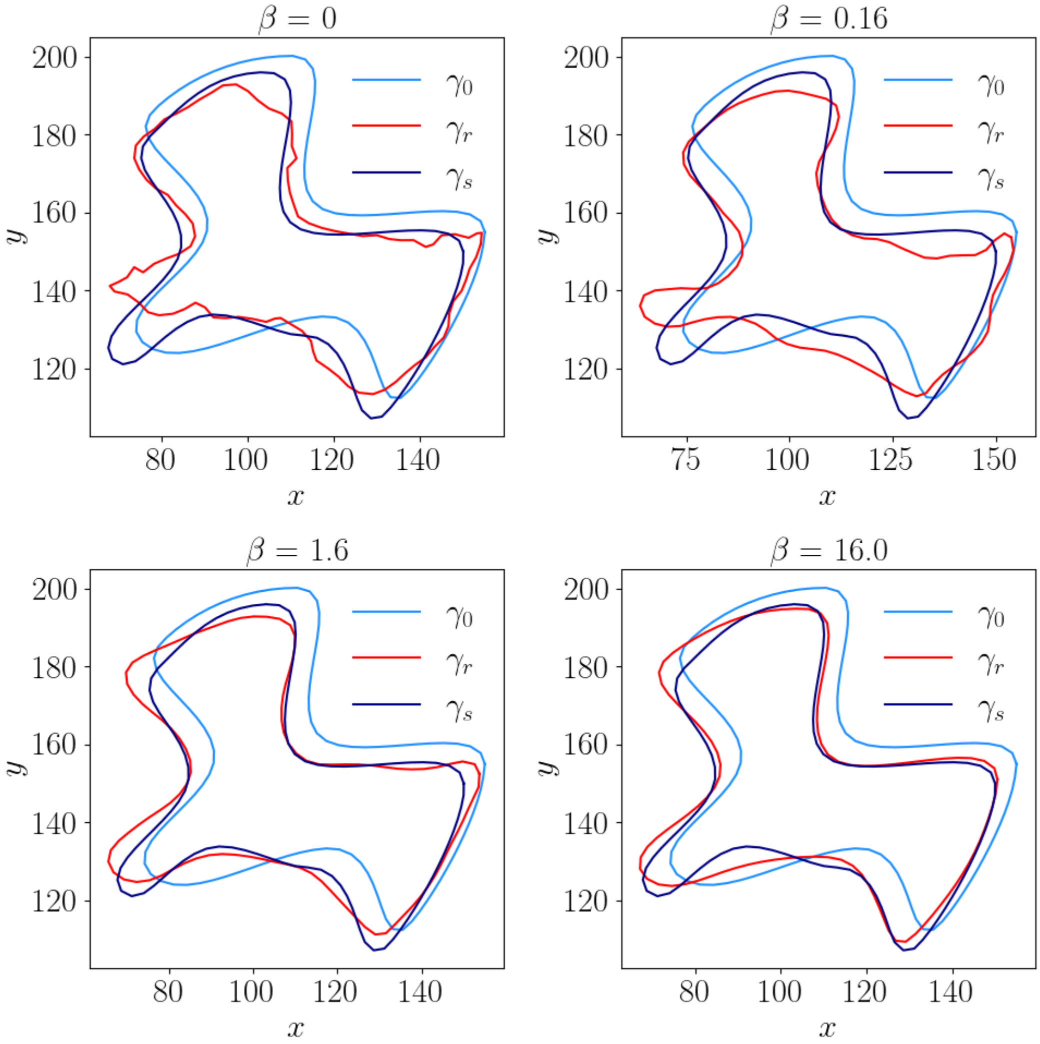

Figure 5.11: A plot showing the reconstructed curves for different values of the regularization parameter. In this plot we see the reconstructed curves for angles in A_{16} and data containing noise with $\mu = 0.15$. The curves presented in each subplot are the true solution, γ_s, the reconstructed solution, γ_r, and the initial guess, γ_0, which coincides with the reference solution. Here we see that with noise present in the data, strong regularization is needed. However, this could be dependent on a good initial guess, which we have here.

5.4 Poor initial guess

In this section we present an experiment where the initial guess is a circle, while the true solution is more complex. The shapes are displayed in Figure 5.12. The values of the metric described in section 5.1.6, for the different sets and parameters, are shown in Table 5.4. For such a poor initial guess we do not expect very good results, especially when there is noise present. This is also reflected in the results, as we observe that the metric is above 10 for all reconstructions with noise. In Figure 5.13 we present the reconstructions for angles in A_8 and noise in the data. As one can observe, the reconstructions with weak regularization gives a poor reconstruction that is highly irregular and contains artifacts. For $\beta = 0.8$ the reconstruction is much smoother, and this is maybe the best reconstruction of the overall shape. However, at the rightmost part of the figure we see an artifact. The strongest regularization gives a rather poor reconstruction of the overall shape, but it does not create artifacts. In addition, the reconstruction with strong regularization is not as sensitive to noise in the data, and is in this sense more robust. For the reconstructions with no noise in the data we see that for angles in A_8 and A_{16} the values are a bit more promising for some reconstructions. In Figure 5.14 we show the reconstructions for angles in A_{16}. As one can observe, the reconstruction with no regularization is poor, while the one with $\beta = 0.16$ approximates the true solution well for almost the entire curve, but deviates at the bottom. For $\beta = 1.6$ this problem is eliminated, and we have obtained a very good reconstruction. The reconstruction with the strongest regularization is similar to the reconstruction with strongest regularization in Figure 5.13, it is a rather poor approximation of the overall shape, but robust.

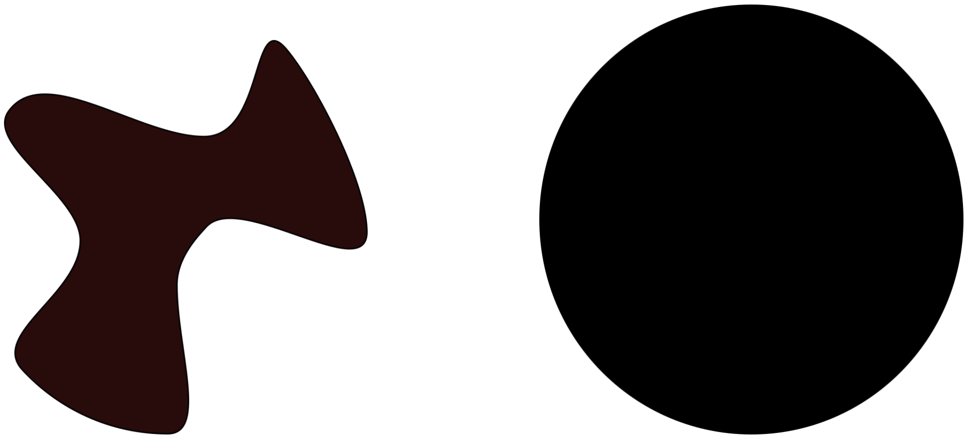

Figure 5.12: On the left we see the true solution from which the Radon transforms are generated, on the right we see the reference solution.

Regularization	A_4	A_8	A_{16}	A_5	A'_4	A'_8	A'_{16}	A'_5
0.0	13.5	13.43	12.36	10.65	13.83	12.83	12.63	**16.82**
$0.01N$	**12.01**	**6.69**	11.64	14.88	**12.4**	11.75	**11.62**	24.2
$0.1N$	26.64	7.89	**2.81**	**9.03**	12.95	**11.68**	20.2	18.88
N	32.26	9.32	10.05	14.61	14.02	14.38	22.92	21.79

Table 5.4: A table showing the difference between the reconstructed solution m_n^{rec} and the true solution m_n^{sol}, measured in $d(m_n^{\text{rec}}, m_n^{\text{sol}})$, for the second experiment (shapes in Figure 5.9). The prime indicates that the data set contained noise.

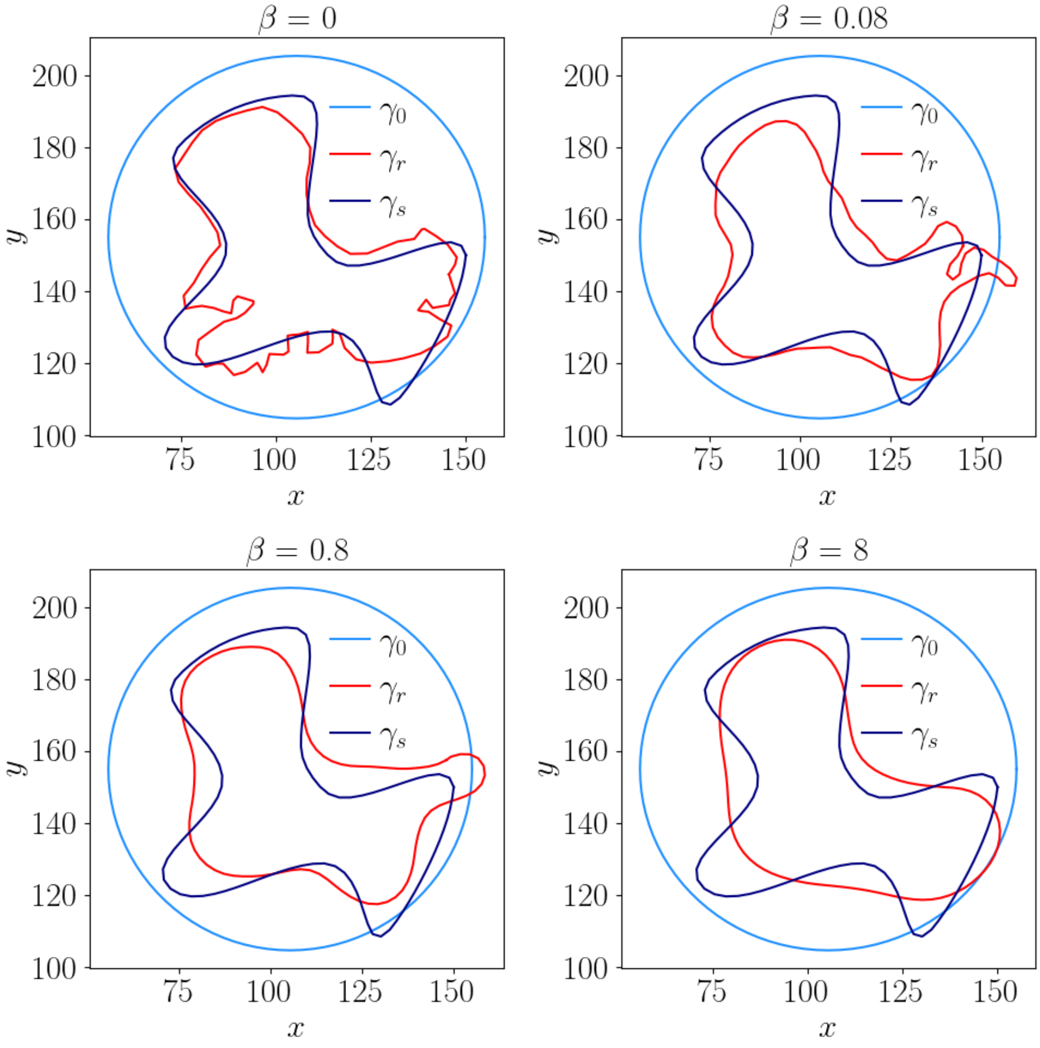

Figure 5.13: A plot showing the reconstructed curves for different values of the regularization parameter. In this plot we see the reconstructed curves for angles in A_8 and data containing noise with $\mu = 0.15$. The curves presented in each subplot are the true solution, γ_s, the reconstructed solution, γ_r, and the initial guess, γ_0, which coincides with the reference solution. With a poor initial guess and noise in the data, it is hard to obtain a good reconstruction, but the reconstructions with stronger regularization are the best attempts.

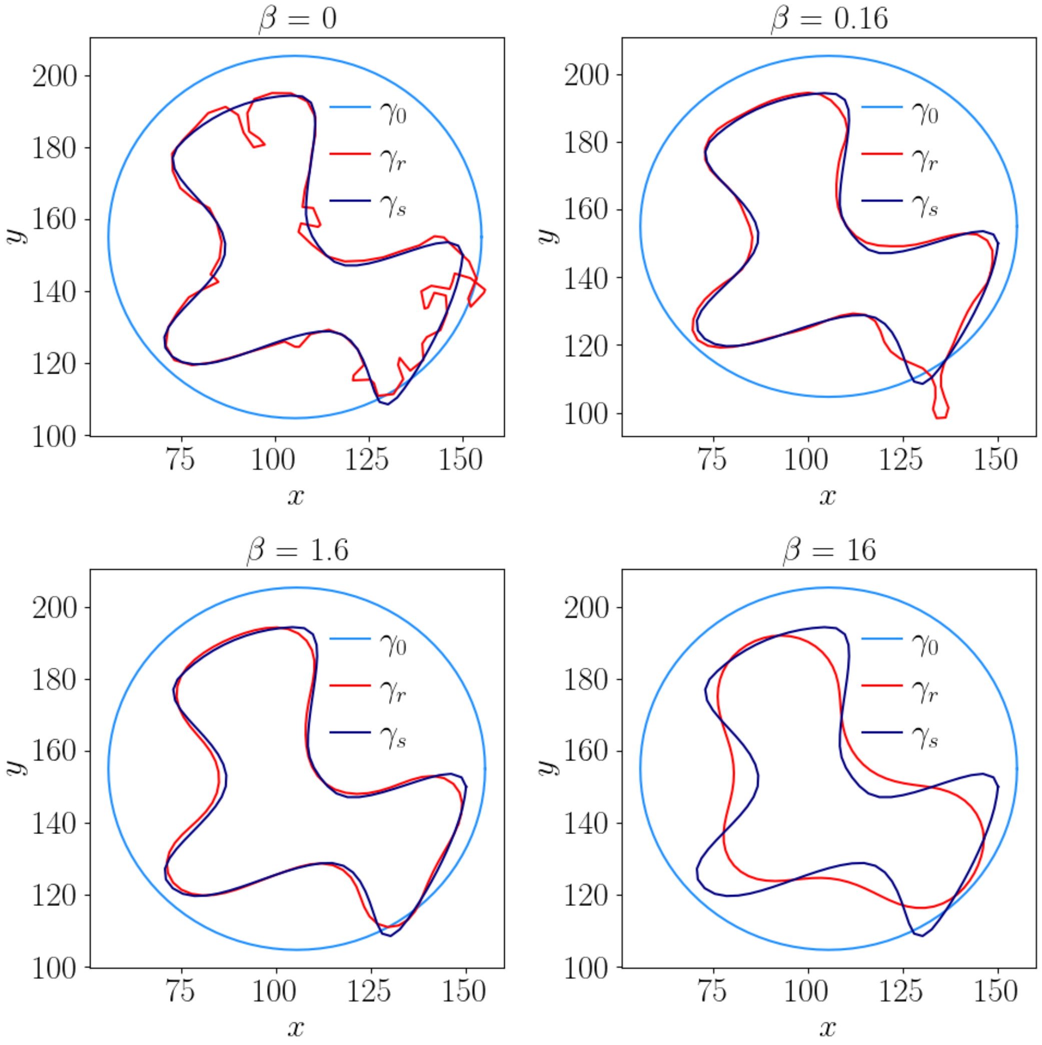

Figure 5.14: A plot showing the reconstructed curves for different values of the regularization parameter. In this plot we see the reconstructed curves for angles in A_{16} and data containing no noise. The curves presented in each subplot are the true solution, γ_s, the reconstructed solution, γ_r, and the initial guess, γ_0, which coincides with the reference solution. Here we have a very poor initial guess, and observe that quite strong regularization is needed to avoid a very irregular reconstruction. However, with too strong regularization the overall fit is rather poor.

5.5 Other experiments

A total of 9 experiments were conducted, the shapes, and the tables with results, for the experiments not discussed in this chapter are presented in Appendix A. As earlier mentioned we should keep in mind that this metric only gives an indication of how well the reconstruction is. A reconstruction with a large distance to the true solution will, however, be poor in at least some areas, but a reconstruction with a relatively small distance to the true solution might in practice also be relatively poor, due to its irregularity. This was often the case for the reconstructions with $\beta = 0$, which gave highly irregular curves. Stronger regularization gave smoother curves, but sometimes at the cost of how well the reconstruction matched the overall shape of the true solution.

Chapter 6

Conclusion

This thesis has introduced a regularization functional for the reconstruction of shapes, which is based on the Euler-Bernoulli bending energy. We have considered the inverse problem of reconstructing a shape from its Radon transform, given only for a finite number of directions. To solve the problem numerically, we implemented the BFGS-method with a line search based on the Wolfe conditions.

In the numerical experiments we tried to reconstruct a shape from an initial guess, with four different levels of regularization. This was done with different amounts of information available, where the information is the Radon transforms of the true solution for a given set of angles. Three of the sets contained angles evenly distributed between 0 and 180 degrees, with respectively 4, 8 and 16 angles, while he last set contained 5 angles evenly distributed in the more narrow interval from 67.5 to 112.5 degrees. In addition, we compared these reconstructions with the reconstructions obtained when random noise was added to the Radon transforms. The quality of the reconstructions varied much for the different experiments. It was observed that, for some shapes, it was important that the available Radon transforms corresponded to angles in the entire interval from 0 to 180 degrees, as the Radon transforms for angles between 67.5 to 112.5 degrees failed to give enough information. It was also seen that, when the initial guess was very poor, more angles were needed to obtain good reconstructions, and the reconstructions with 8 or 16 available Radon transforms were much better than the ones with 4. The same was observed when noise was added to the data.

In the experiments we observed that the optimal amount of regularization depended on several factors, including of course the shape of the true solution and the initial guess. If the shapes were simple, we had a good initial guess, and sufficient available information, with no random noise, the best reconstructions were obtained with no regularization. However, in almost all experiments, no regularization resulted in a very irregular reconstruction. The reconstructions containing some regularization were more visually appealing. The strongest level of regularization appeared to be too strong in most experiments, as the reconstructions did not match the overall shape of the true solution very well. However, in some experiments the other reconstructions were very poor, due to either noise, lack of available information or poor initial guesses, so that this gave the best reconstruction anyway. The best reconstructions in most experiments was obtained with β equal to $0.01N$ or $0.1N$, this gave reconstructions that were smooth and also matched the overall shape of the true solution quite well.

For further work it would be interesting to look at the possibilities for improving the iterative regularization method. More specifically, it should be possible to use more intelligent methods for determining the number of iterations, instead of using a fixed number.

References

[1] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.

[2] J. L. Mueller and S. Siltanen. *Linear and nonlinear inverse problems with practical applications*, volume 10 of *Computational Science & Engineering*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2012, pages xiv+351. ISBN: 978-1-611972-33-7. DOI: 10.1137/1.9781611972344. URL: https://doi.org/10.1137/1.9781611972344.

[3] T. Fidler, M. Grasmair, and O. Scherzer. Shape reconstruction with a priori knowledge based on integral invariants. *SIAM J. Imaging Sci.*, 5(2):726–745, 2012. DOI: 10.1137/110824735. URL: https://doi.org/10.1137/110824735.

[4] J. Eckhardt, R. Hiptmair, T. Hohage, H. Schumacher, and M. Wardetzky. Elastic energy regularization for inverse obstacle scattering problems. *Inverse Problems*, 35(10):104009, 20, 2019. ISSN: 0266-5611. DOI: 10.1088/1361-6420/ab3034. URL: https://doi.org/10.1088/1361-6420/ab3034.

[5] M. H. Freedman, Z.-X. He, and Z. Wang. Möbius energy of knots and unknots. *Ann. of Math. (2)*, 139(1):1–50, 1994. ISSN: 0003-486X. DOI: 10.2307/2946626. URL: https://doi.org/10.2307/2946626.

[6] H. Federer. *Geometric measure theory.* Springer, 2014.

[7] K. E. Atkinson. *An introduction to numerical analysis.* John Wiley & Sons, Inc., New York, second edition, 1989, pages xvi+693. ISBN: 0-471-62489-6.

[8] P. J. Olver. *Introduction to partial differential equations.* Undergraduate Texts in Mathematics. Springer, Cham, 2014, pages xxvi+635. ISBN: 978-3-319-02098-3; 978-3-319-02099-0. DOI: 10.1007/978-3-319-02099-0. URL: https://doi.org/10.1007/978-3-319-02099-0.

[9] J. Nocedal and S. J. Wright. *Numerical optimization.* Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006, pages xxii+664. ISBN: 978-0387-30303-1.

[10] B. Fornberg. Numerical differentiation of analytic functions. *ACM Trans. Math. Software*, 7(4):512–526, 1981. ISSN: 0098-3500. DOI: 10.1145/355972.355979. URL: https://doi.org/10.1145/355972.355979.

[11] W. Squire and G. Trapp. Using complex variables to estimate derivatives of real functions. *SIAM Rev.*, 40(1):110–112, 1998. ISSN: 0036-1445. DOI: 10.1137/S003614459631241X. URL: https://doi.org/10.1137/S003614459631241X.

[12] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes.* Cambridge University Press, Cambridge, third edition, 2007, pages xxii+1235. ISBN: 978-0-521-88068-8. The art of scientific computing.

[13] O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier, and F. Lenzen. *Variational methods in imaging*, volume 167 of *Applied Mathematical Sciences*. Springer, New York, 2009, pages xiv+320. ISBN: 978-0-387-30931-6.

[14] S. Lam, A. Pitrou, and S. Seibert. *Numba: a LLVM-based Python JIT compiler*. Association for Computing Machinery, 2015, pages 1–6. ISBN: 978-1-450-34005-2. URL: `https://doi.org/10.1145/2833157.2833162`.

Appendix A

Additional experiments

A.1 Experiment 4

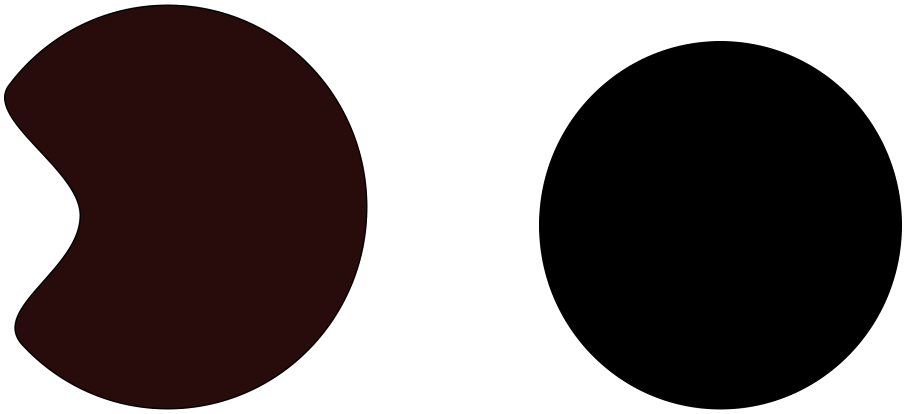

Figure A.1: On the left we see the true solution from which the Radon transforms are generated, on the right we see the reference solution.

Regularization	A_4	A_8	A_{16}	A_5	A'_4	A'_8	A'_{16}	A'_5
0.0	4.67	**2.18**	3.32	**8.22**	8.61	**4.83**	4.43	13.69
$0.01N$	**2.46**	4.28	**2.07**	15.44	**6.7**	5.65	**3.72**	15.11
$0.1N$	7.82	4.27	5.48	14.62	10.1	10.56	8.09	**9.23**
N	11.47	7.15	8.6	13.48	12.94	7.76	7.36	17.81

Table A.1: A table showing the difference between the reconstructed solution m_n^{rec} and the true solution m_n^{sol}, measured in $d(m_n^{\text{rec}}, m_n^{\text{sol}})$, for the second experiment (shapes in Figure A.1). The prime indicates that the data set contained noise.

The reconstructions with angles in A_4, A_8 and A_{16} matched the overall shape of the true solution quite well, when weak or no regularization was utilized. The reconstructions obtained with no regularization had a more irregular boundary, this was most apparent when the data contained noise. The reconstructions with angles in A_5 were significantly worse, which could be because of the concave part of the true solution, as it cannot be seen from these directions.

A.2 Experiment 5

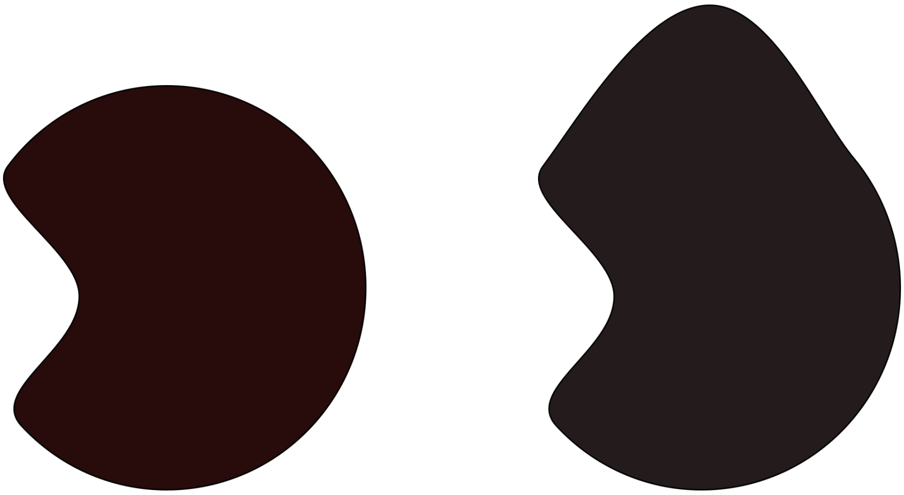

Figure A.2: On the left we see the true solution from which the Radon transforms are generated, on the right we see the reference solution.

Regularization	A_4	A_8	A_{16}	A_5	A'_4	A'_8	A'_{16}	A'_5
0.0	2.68	2.32	4.57	**2.94**	6.52	8.82	5.53	7.77
$0.01N$	4.96	**1.99**	2.53	5.77	5.53	**3.92**	**2.76**	12.19
$0.1N$	**2.54**	2.66	2.47	4.58	7.47	5.2	5.4	**7.73**
N	2.78	3.1	**2.41**	3.44	**4.31**	7.25	8.23	10.95

Table A.2: A table showing the difference between the reconstructed solution m_n^{rec} and the true solution m_n^{sol}, measured in $d(m_n^{\text{rec}}, m_n^{\text{sol}})$, for the second experiment (shapes in Figure A.2). The prime indicates that the data set contained noise.

The reconstructions obtained with some degree of regularization had smoother boundaries. Reconstructions with angles in A_5 were almost as good as the other reconstructions, when there was no noise in the data, which indicates that the information from these angles was sufficient. However, when the data contained noise, the reconstructions in A_5 were significantly worse than the other reconstructions.

A.3 Experiment 6

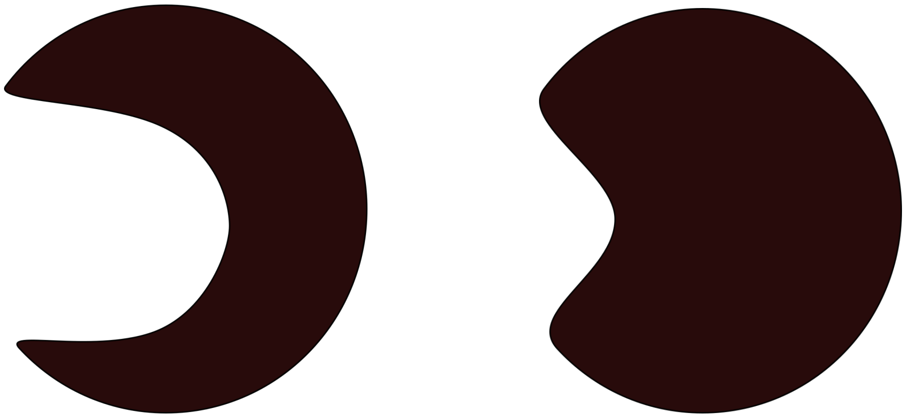

Figure A.3: On the left we see the true solution from which the Radon transforms are generated, on the right we see the reference solution.

Regularization	A_4	A_8	A_{16}	A_5	A'_4	A'_8	A'_{16}	A'_5
0.0	11.59	32.15	16.33	24.34	23.24	28.37	24.29	26.6
$0.01N$	**10.61**	21.25	**13.68**	25.45	**15.13**	**22.02**	31.3	**17.64**
$0.1N$	13.26	23.57	13.72	21.71	17.48	33.21	**16.63**	30.4
N	23.57	**21.15**	21.92	**18.18**	26.26	23.07	23.98	22.14

Table A.3: A table showing the difference between the reconstructed solution m_n^{rec} and the true solution m_n^{sol}, measured in $d(m_n^{\text{rec}}, m_n^{\text{sol}})$, for the second experiment (shapes in Figure A.3). The prime indicates that the data set contained noise.

The true solution was difficult to reconstruct, due to the pointy part of the boundary. The best reconstructions were obtained with some regularization, the reason for the large values in the table is that no reconstructions matched the pointy part.

A.4 Experiment 7

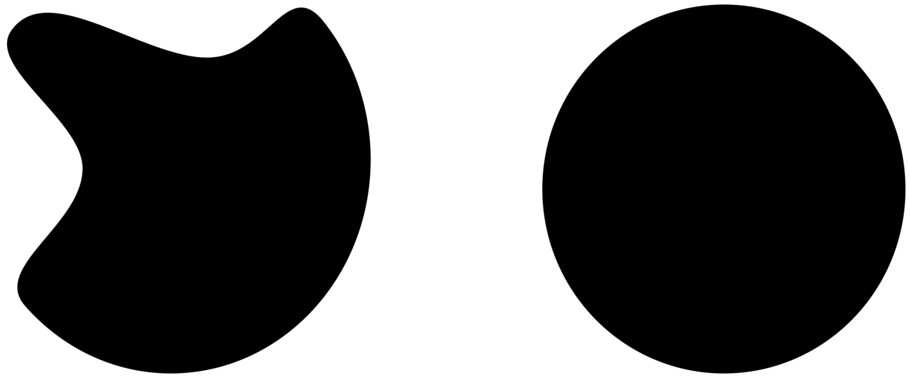

Figure A.4: On the left we see the true solution from which the Radon transforms are generated, on the right we see the reference solution.

Regularization	A_4	A_8	A_{16}	A_5	A'_4	A'_8	A'_{16}	A'_5
0.0	8.42	7.24	10.32	**2.57**	15.16	9.17	**5.58**	**12.95**
$0.01N$	**2.56**	**5.37**	**2.53**	19.62	8.66	11.48	8.04	14.86
$0.1N$	7.63	5.56	5.36	23.45	**4.97**	**8.23**	15.45	18.61
N	12.78	14.51	8.51	17.86	18.68	12.49	17.28	19.39

Table A.4: A table showing the difference between the reconstructed solution m_n^{rec} and the true solution m_n^{sol}, measured in $d(m_n^{\text{rec}}, m_n^{\text{sol}})$, for the second experiment (shapes in Figure A.4). The prime indicates that the data set contained noise.

Artifacts, created when no regularization was used, gave some seemingly random values for the metric corresponding to these reconstructions. The reconstructions with some regularization had smoother boundaries, thus the metric gives a more realistic result for these reconstructions.

A.5 Experiment 8

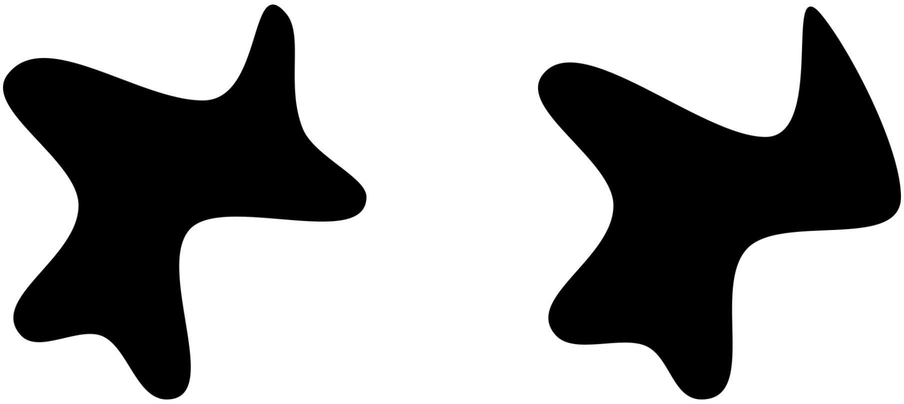

Figure A.5: On the left we see the true solution from which the Radon transforms are generated, on the right we see the reference solution.

Regularization	A_4	A_8	A_{16}	A_5	A'_4	A'_8	A'_{16}	A'_5
0.0	6.81	**2.85**	**4.28**	**2.94**	7.14	13.15	11.08	**8.11**
$0.01N$	**5.41**	9.96	7.67	6.12	10.81	11.68	10.81	10.71
$0.1N$	6.85	4.39	4.35	11.09	**6.69**	7.66	6.14	9.58
N	5.49	5.62	6.54	9.17	7.27	**5.71**	**6.01**	8.23

Table A.5: A table showing the difference between the reconstructed solution m_n^{rec} and the true solution m_n^{sol}, measured in $d(m_n^{\text{rec}}, m_n^{\text{sol}})$, for the second experiment (shapes in Figure A.5). The prime indicates that the data set contained noise.

The best reconstructions were obtained without regularization when there was no noise in the data, as the overall shape matched the true solution better. However, with noise present, these reconstructions became highly irregular, and the best reconstructions were obtained with some regularization.

A.6 Experiment 9

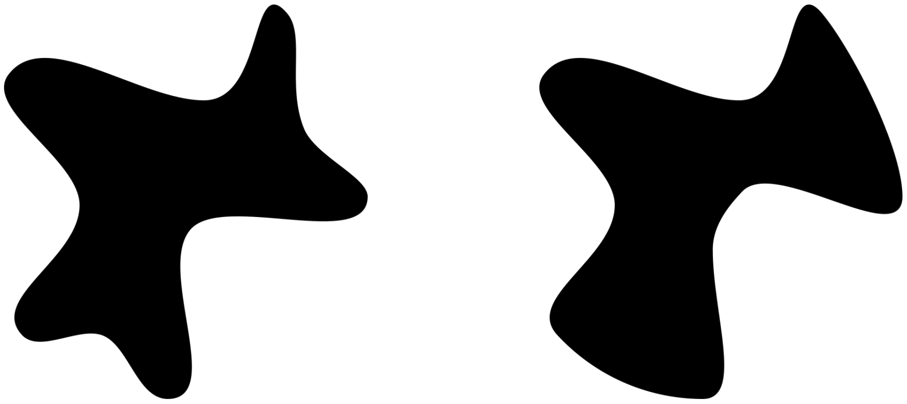

Figure A.6: On the left we see the true solution from which the Radon transforms are generated, on the right we see the reference solution.

Regularization	A_4	A_8	A_{16}	A_5	A'_4	A'_8	A'_{16}	A'_5
0.0	**3.11**	3.94	3.09	7.74	8.24	**5.86**	7.35	9.74
$0.01N$	6.76	3.76	**1.51**	3.77	9.7	6.42	10.11	8.76
$0.1N$	3.61	**2.32**	2.42	**3.33**	**6.1**	8.89	**6.36**	**5.51**
N	5.79	4.69	6.62	4.9	9.24	8.16	8.53	7.55

Table A.6: A table showing the difference between the reconstructed solution m_n^{rec} and the true solution m_n^{sol}, measured in $d(m_n^{\mathrm{rec}}, m_n^{\mathrm{sol}})$, for the second experiment (shapes in Figure A.6). The prime indicates that the data set contained noise.

The best results were in general obtained with some degree of regularization. In the columns where the lowest value of the metric was obtained without regularization, the boundary of this reconstruction was irregular, and the reconstruction corresponding to the second lowest value of the column was more visually appealing.

NTNU

Norwegian University of
Science and Technology