# Sparsity Regularization for Electrical Impedance Tomography

*Master of Science Thesis*
*by*
*Henrik Garde*

*January 2013*

Department of Mathematics
Technical University of Denmark

**DTU Mathematics**
Department of Mathematics

DTU

# Sparsity Regularization for Electrical Impedance Tomography

**Author:**

Henric Garde ________________________

*Signature*

**Supervisor:**

Kim Knudsen

**DTU Mathematics**
Department of Mathematics

Technical University of Denmark
Matematiktorvet, Building 303 S
2800, Kgs. Lyngby
Denmark

Email: MAT-INSTADM@mat.dtu.dk

| | |
|---|---|
| Project period: | September 1, 2012 - January 25, 2013 |
| Workload: | 30 Honors ECTS |
| Degree: | Master of Science |
| Programme: | Mathematical Modelling and Computation (Honors) |
| Copyright: | © Henrik Garde, 2013 |

# Preface

This thesis was prepared at the Department of Mathematics at the Technical University of Denmark (DTU) in fulfillment of the requirements for acquiring an M.Sc. in Mathematical Modelling and Computation. This thesis is the conclusion of my honors programme at DTU, and the workload is to be reflected by this. The work was conducted from September 2012 to January 2013 under supervision of Associate Professor Kim Knudsen from DTU Mathematics.

The thesis deals with sparsity regularization and total variation regularization for electrical impedance tomography reconstruction, and the use of prior information to improve solutions. The theory is mainly based on the articles Jin and Maass [8], Jin *et al.* [9].

The prerequisites for reading this thesis is familiarity with functional analysis in terms of continuity, boundedness, and Hilbert spaces. A basic knowledge of distribution theory and measure theory is helpful, but not strictly necessary to understand the material. It is recommended to be familiar with Sobolev spaces and their role with regard to partial differential equations (PDE) and weak forms for PDE's. There is a short description in Appendix B giving the basic definitions for Sobolev spaces, and Appendix C lists some convenient results that will be applied throughout the thesis.

Henrik Garde
January, 2013

# Acknowledgements

I would foremost like to thank my thesis supervisor Associate Professor Kim Knudsen from DTU Mathematics, for offering me this subject to work on. I have previously been mostly engaged in purely theoretical projects, and while this project is heavy on theory as well, it has a practical purpose, which makes the project more motivating to work on. Kim has been very encouraging and engaged in the project, and at our weekly meetings we have had some good discussions on what I have been up to.

I would also like to thank Postdoctoral Researcher Martin S. Andersen from DTU Informatics, to provide the idea for looking into total variation regularization, and for taking time out to have a meeting with me.

Finally, I would also like to thank my honors supervisor Professor Ole Christensen at DTU mathematics, for offering me interesting special courses during my education and taking a special interest in the courses that I have taken.

# Abstract

This thesis deals with the inverse problem that arises from electrical impedance tomography (EIT) reconstruction. Here the mathematical foundation is laid for solving the forward EIT problem uniquely, along with continuity and differentiability results for the forward problem.

The inverse EIT problem is investigated in great theoretical detail with respect to regularization techniques, where sparsity and total variation regularization are used to iteratively give approximate solutions to the problem. The use of multiple datasets and partial data are investigated, along with their respective effect on the solution. The idea of using prior information is applied throughout the thesis, in order to improve these approximate solutions, for instance in terms of the gradient used in the iterative algorithm and the bias that is introduced by the regularization.

Furthermore, I have engineered specific basis functions into the solutions of sparsity regularization, by using different parameters for each basis function. This successfully improves the solution to a degree where it is possible to reconstruct sharp edges and the correct contrast, even for very difficult inclusions, something that is otherwise unheard of for a problem as ill-posed and non-linear as EIT. Prior information is also applied in an experiment to have total variation regularization determine an approximation to the support of an inclusion, and use this information to improve the solution from the sparsity regularization.

The iterative methods have been implemented successfully in Python using FEniCS [31], that is based on the finite element method.

# List of Figures

# List of Symbols

| | |
|---|---|
| $\mathcal{A}$ | Admissible set |
| $\mathcal{A}_0$ | Set of admissible inclusions |
| $\alpha_k$ | Regularization parameters for sparsity regularization |
| $\beta$ | Regularization parameter for total variation regularization |
| $C_c^\infty$ | Space of smooth functions with compact support |
| $d$ | Dimension for domain $\Omega$ |
| $F_g$ | Forward map on $\Omega$ |
| $\mathcal{F}_g$ | Forward map on $\partial\Omega$ |
| $g$ | Neumann-data |
| $\Gamma$ | Subset of $\partial\Omega$ |
| $H^{1/2}(\partial\Omega)$ | The space $W^{1/2,2}(\partial\Omega)$ |
| $H^k(\Omega)$ | The space $W^{k,2}(\Omega)$ |
| $H_0^k(\Omega)$ | The space $W_0^{k,2}(\Omega)$ |
| $H^{-1/2}(\partial\Omega)$ | The space $W^{-1/2,2}(\partial\Omega)$ |
| $H^{-k}(\Omega)$ | The space $W^{-k,2}(\Omega)$ |
| $\tilde{H}^1(\Omega)$ | Space of functions $f$ in $H^1(\Omega)$ with $\int Tf ds = 0$ |
| $\tilde{H}^{1/2}(\partial\Omega)$ | Space of functions $f$ in $H^{1/2}(\partial\Omega)$ with $\int f ds = 0$ |

| | |
|---|---|
| $\tilde{H}^{-1/2}(\partial\Omega)$ | Space of distributions $f$ in $H^{-1/2}(\partial\Omega)$ with $\langle f, 1\rangle = 0$ |
| $I_{\max}$ | Maximum no. of iterations |
| $\langle\cdot,\cdot\rangle$ | Inner product or dual pairing |
| $J$ | Discrepancy term |
| $\nabla J$ | Gradient of discrepancy term |
| $\nabla_S J$ | Sobolev gradient of discrepancy term |
| $\lambda$ | Bound for admissible set |
| $L^p$ | Space of $p$-integrable functions |
| $\ell^p$ | Space of $p$-summable sequences |
| $M, \tau$ | Parameters for weak monotonicity |
| $n$ | Outwards pointing unit normal |
| $\lVert\cdot\rVert$ | Norm |
| $\Omega$ | Open bounded domain in $\mathbb{R}^d$ with smooth boundary |
| $\partial\Omega$ | Boundary of $\Omega$ |
| $\omega$ | Noise level |
| $\phi$ | Dirichlet-data |
| $\psi_k$ | Basis functions used for sparsity regularization |
| $\Psi_S$ | Objective function for sparsity regularization |
| $\Psi_{\mathrm{TV}}$ | Objective function for total variation regularization |
| $P_{\mathrm{TV}}$ | Penalty term for total variation regularization |
| $P_{\mathrm{TV},c}$ | Smooth approximation to $P_{\mathrm{TV}}$ |
| $R_{\boldsymbol{\alpha},1}$ | Penalty term for sparsity regularization |
| $s$ | Step size |
| $\mathcal{S}_\beta$ | Soft shrinkage/thresholding operator |
| $\sigma$ | Electrical conductivity |

| | |
|---|---|
| $\sigma_0$ | Background conductivity |
| $\sigma_{C,\tilde{r}}$ | Concentric conductivity |
| $\delta\sigma$ | Inclusion |
| $s_{\min}, s_{\max}$ | Bounds for initial step size |
| $s_{\text{stop}}$ | Stopping criterion for step size |
| $T$ | Trace operator |
| $u$ | Electrical potential |
| $V_h$ | Finite element space |
| $W^{1/2,p}(\partial\Omega)$ | The space given by the trace of $W^{1,p}(\Omega)$ |
| $W^{k,p}(\Omega)$ | Sobolev space for functions with $k$ weak derivatives in $L^p(\Omega)$ |
| $W_0^{k,p}(\Omega)$ | Closure of $C_c^\infty(\Omega)$ in $\|\cdot\|_{W^{k,p}}$ |
| $W^{-1/2,p}(\partial\Omega)$ | Continuous dual space of $W^{1/2,p}(\partial\Omega)$ |
| $W^{-k,p}(\Omega)$ | Continuous dual space of $W_0^{k,p}(\Omega)$ |

# Contents

# 1

# Introduction

Electrical impedance tomography (EIT) is an imaging technique for which electrodes are placed on the surface of the body, emitting small amounts of current through the body (see Figure 1.1). The resulting voltage distribution at the electrodes are then measured and constitutes the data for the mathematical problem. This allows for estimation of the electrical conductivity inside the body, which varies for different parts of the body such as blood, bone, and muscles. The EIT method is especially good at measuring the lung capacity since air has practically no electrical conductivity and therefore the expiration and inspiration state of the lung yields very different conductivities in the same part of the body [27], which by difference imaging leads to estimates of the air distribution in the lungs as seen in Figure 1.2 on the following page.

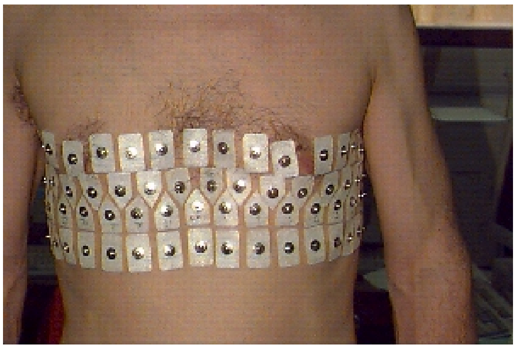


**Figure 1.1:** Electrodes placed on the chest, used for EIT reconstruction. URL: http://en.wikipedia.org/wiki/Electrical_impedance_tomography.

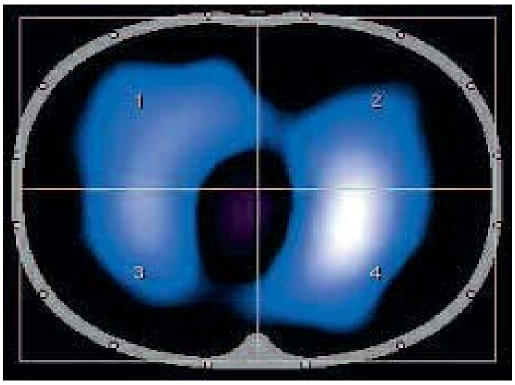


**Figure 1.2:** Air distribution in lungs of a patient, using the device Pulmovista 500. URL: http://campaigns.draeger.com/pulmovista500/en/#introduction.

In biological tissue the electrical conductivity varies from $10^{-4}$ to $10^2 S/m$ (measured in Siemens per metre) [12], the conductivity of a small selection of tissues can be seen in Table 1.1. The fact that the conductivity spans several orders of magnitude, with clear distinction between different types of tissue, hints that it can be used for medical imaging, i.e. that reconstructing the electrical conductivity in the body can produce images of the insides of the body. It should be noted that electrical conductivity depends on the applied frequency, however this is often neglected since the measurements are performed at a fixed frequency [12].

| Tissue | Conductivity (S/m) |
|---|---|
| Wet skin | $3 \cdot 10^{-3}$ |
| Blood | $7 \cdot 10^{-1}$ |
| Fat | $2 \cdot 10^{-2}$ |
| Liver | $5 \cdot 10^{-2}$ |

**Table 1.1:** Conductivities for different types of tissue, at a frequency of 1 kHz [12].

EIT is harmless and the current that is sent through the body is below the threshold of the patients' perception. Unlike many other imaging techniques the actual equipment required to acquire the data is quite cheap. However, due to the highly non-linearity and ill-posedness of the underlying inverse problem, EIT does not perform up to par with other well-established imaging techniques such as CT or MRI when it comes to spatial resolution, and with EIT it is almost impossible to reconstruct any form of sharp edge.

The strengths of EIT, beyond being a harmless and cheap method, is also the speed of the measurements and contrast resolution, making EIT a good candidate for supplementing mammography or MRI in early breast cancer detection, since the former methods have a high false positive rate.

One may argue that EIT is actually not a tomographic method, as the word tomography roughly translates to constructing images in slices, which is actually not the case with EIT, since the current will not be confined to some hypothetical plane. Due to this fact, EIT is intuitively made for 3D-scanning, and solving the inverse problem in 2D can mostly be considered as investigating the mathematical and numerical aspects of the problem, where a proper 3D solution would probably be better suited for practical purposes.

## 1.1 Deriving the Mathematical Model

Now that a small motivation is given for EIT, it is about time to investigate the underlying mathematical model. It is also worth remembering that while we want a good spatial resolution for a reconstruction, this may simply not be possible, so the goal is to get a reasonable spatial resolution and hopefully be able to get a good reconstruction of the contrast.

As mentioned earlier the practical problem is inherently three-dimensional, and therefore $x \in \mathbb{R}^3$ denotes the spatial coordinates in some open bounded domain $\Omega \subset \mathbb{R}^3$, and $t$ the temporal coordinate. In order to model the behaviour of EIT it seems appropriate to start with Maxwell's equations, which are a set of coupled PDE's that can model the propagation of electromagnetic fields. These equations consists of Faraday's law, Ampère's law (with Maxwell's correction term), Gauss' law of magnetism, and Gauss' law,

$$\begin{aligned}
\nabla \times E &= -\partial_t B, && \text{(Faraday's law)} \qquad (1.1)\\
\nabla \times H &= J + \partial_t D, && \text{(Ampère's law)} \qquad (1.2)\\
\nabla \cdot B &= 0, && \text{(Gauss' law of magnetism)} \qquad (1.3)\\
\nabla \cdot D &= \rho. && \text{(Gauss' law)} \qquad (1.4)
\end{aligned}$$

Here $\partial_t$ denotes a partial derivative in $t$. $E(x,t)$ is the electric field intensity, $B(x,t)$ is the magnetic flux density, $D(x,t)$ is the electric displacement field, $H(x,t)$ is the magnetic field intensity, $J(x,t)$ is the free current density, and $\rho(x,t)$ is the free electric charge density [27].

In EIT the current density takes a special form, namely, $J(x,t) = J(x)e^{i\omega t}$ for some temporal angular frequency $\omega$, which in turn yields that all the entities in (1.1) through (1.4) are on the form $F(x,t) = F(x)e^{i\omega t}$ [27]. Inserting this into (1.1) and (1.2) yields

$$\nabla \times E(x)e^{i\omega t} = -\partial_t(B(x)e^{i\omega t}) = -i\omega B(x)e^{i\omega t},$$

thus

$$\nabla \times E(x) = -i\omega B(x), \tag{1.5}$$

and similarly

$$\nabla \times H(x) = J(x) + i\omega D(x). \tag{1.6}$$

Seemingly (1.5) and (1.3) and are not coupled with (1.6) and (1.4), however, this is in fact the case when looking at the electric permittivity $\epsilon(x,\omega)$, magnetic permeability $\mu(x,\omega)$, and the electrical conductivity $\sigma(x,\omega)$

$$D = \epsilon(x,\omega)E, \tag{1.7}$$
$$B = \mu(x,\omega)H, \tag{1.8}$$
$$J = \sigma(x,\omega)E. \tag{1.9}$$

Now investigating $E$ and $B$ about $\mu = 0$ via Taylor expansions yields

$$\begin{aligned} E(x,\omega,\mu) &= E(x,\omega,0) + \partial_\mu E(x,\omega,0)\mu + \tfrac{1}{2}\partial_\mu^2 E(x,\omega,0)\mu^2 + \dots, \\ &= E_0 + \mu E_1 + \mu^2 E_2 + \dots \\ B(x,\omega,\mu) &= B_0 + \mu B_1 + \mu^2 B_2 + \dots \end{aligned}$$

According to Mueller and Siltanen [27] $\mu$ is very small in the human body. Thus ignoring anything but the leading order term for (1.5) and approximating $B_0 = 0$ by setting $\mu = 0$ in (1.8) leads to

$$\nabla \times E_0 = 0. \tag{1.10}$$

Due to (1.10) the electric field is conservative and can therefore be determined by the gradiant of the electrical potential $u$ [12] by

$$E_0 = -\nabla u. \tag{1.11}$$

Since the divergence of a curl is always zero[1], inserting (1.7) and (1.9) into (1.6) and taking the divergence on both sides of the equation, yields

$$\nabla \cdot [(\sigma + i\omega\epsilon)E] = \nabla \cdot (\nabla \times H) = 0. \tag{1.12}$$

[1] from Schaum's Mathematical Handbook of Formulas and Tables.

Now only looking at the zeroth order term $E_0$, and inserting (1.11). For $\omega = 0$ we arrive at

$$0 = \nabla \cdot (\sigma E_0) = -\nabla \cdot (\sigma \nabla u). \tag{1.13}$$

(1.13) is the PDE of interest in EIT. The problem can be formulated theoretically in 2D which is a beneficial representation for various reasons. Some being that it is easier and faster to solve PDE's in lower dimensions, however, it is also much easier to visualize the solutions. Additionally, one should be able to solve the problem in 2D before trying in higher dimensions. Therefore in the remainder of this thesis we only consider the domain $\Omega \subset \mathbb{R}^2$ where $\Omega$ is open and bounded with a smooth boundary. Now by (1.9) the current density that is measured along the boundary is

$$-J|_{\partial\Omega} = \sigma \frac{\partial u}{\partial n}, \tag{1.14}$$

where $n$ is the outward unit normal on the boundary $\partial\Omega$. Applying Green's second identity (Theorem C.2 using $\psi = 1$, $\epsilon = \sigma$ and $\phi = u$) and inserting (1.13),

$$-\int_{\partial\Omega} J ds = \int_{\partial\Omega} \sigma \frac{\partial u}{\partial n} ds = \int_{\Omega} \nabla \cdot (\sigma \nabla u) dx = 0. \tag{1.15}$$

Thus the current density is conserved over the boundary, which reduces to Kirchoff's law of charge conservation.

Now we can formulate the actual model as

$$-\nabla \cdot (\sigma \nabla u) = 0 \text{ in } \Omega, \tag{1.16}$$

$$\sigma \frac{\partial u}{\partial n} = g \text{ on } \partial\Omega, \tag{1.17}$$

for some $g$ representing the applied current density satisfying $\int_{\partial\Omega} g ds = 0$. This model is known as the *continuum model*, since it assumes that we have the Neumann-data $g$ available everywhere on the boundary. Another popular model, is the *complete electrode model* which more precisely models the discrete electrodes that are used for measurements. However, this model will not be considered in this thesis.

In the following thesis it is assumed that real-valued functions are used, this is partly motivated by the simplifications of the theory while most of the results will translate directly by using complex functions. However, it is mostly due to limitations of the software package FEniCS [31] which does not support complex-valued functions. It should also be noted that the real part of any Banach space is still a complete space, thus known theorems regarding general Banach spaces and Hilbert spaces are of course still valid.

The main topic of the thesis is concerning sparsity regularization for EIT reconstruction based on Jin and Maass [8], Jin *et al.* [9], for which it is assumed that the solution can be expanded in a basis using only few basis functions. However, the project has evolved due to ideas I have had on how to apply explicit prior information to the solution, giving the flexibility to include specific basis functions. Furthermore, at a meeting with Martin S. Andersen from DTU informatics he mentioned that total variation regularization is often used to find piecewise constant solutions in image analysis, and I have looked into total variation regularization as an alternative to sparsity regularization. I also investigate the idea of combining sparsity regularization and total variation regularization, thus improving the solution without the use of explicit prior information about the solution.

# 2

# The Forward Problem

The forward EIT problem is stated by (1.16) and (1.17) given the conductivity $\sigma$ and the current distribution $g$, and solving for $u$, which constitutes the data $u|_{\partial\Omega}$ for the inverse problem.

It should be noted that no function spaces have been mentioned yet, but suppose that $\sigma$, $u$, and $g$ are sufficiently smooth and $\sigma \geq C > 0$ for some constant $C$, what sort of solution would one expect from this PDE problem? Here it should be noted that we only have a Neumann-type boundary condition which is insufficient to ensure a unique solution. Hence, to ensure uniqueness we can add the constraint $\int_{\partial\Omega} uds = 0$, which physically speaking imposes a grounding of the electrical potential and therefore makes sense, despite not being part of the original model. How we actually achieve uniqueness from this constraint is shown in a moment using the Lax-Milgram theorem.

Now we may consider what spaces $g$ and $u$ should belong to. It may be too optimistic to directly solve (1.16) and (1.17), and instead we consider the more general weak formulation of the PDE problem. Therefore instead of the usual derivatives we make use of the weak derivatives or distribution derivatives, which means that we should consider Sobolev spaces (see Appendix B on page 145). Now define the following spaces, where $T$ denotes the trace operator from Theorem B.4 and $\langle \cdot, \cdot \rangle$ denotes the dual pairing,

$$\begin{aligned}\tilde{H}^1(\Omega) &:= \{v \in H^1(\Omega) \mid \int_{\partial\Omega} Tvds = 0\},\\ \tilde{H}^{1/2}(\partial\Omega) &:= \{v \in H^{1/2}(\partial\Omega) \mid \int_{\partial\Omega} vds = 0\},\\ \tilde{H}^{-1/2}(\partial\Omega) &:= \{f \in H^{-1/2}(\partial\Omega) \mid \langle f, 1\rangle = 0\}.\end{aligned}$$

Here it should be noted that since $\Omega$ is bounded then the constant function $1 \in H^{1/2}(\partial\Omega) \subset L^2(\partial\Omega)$, and as the continuous dual of $L^2(\partial\Omega)$ can be identified with itself, then $L^2(\partial\Omega) \subset H^{-1/2}(\partial\Omega)$. This means that if $f \in L^2(\partial\Omega) \cap \tilde{H}^{-1/2}(\partial\Omega)$ then $0 = \langle f, 1\rangle = \langle f, 1\rangle_{L^2(\partial\Omega)} = \int_{\partial\Omega} fds$. Hence, utilizing the dual pairing $\langle f, 1\rangle = 0$ we can formalize such a constraint to any of the distributions in $H^{-1/2}(\partial\Omega)$.

In order to determine the space for which $u$ should belong, the PDE needs to be rewritten to its weak form.

Writing up the weak formulation of the PDE problem is mainly a use of Green's first and second identity. Assume that the trial function $u$ and the test function $v$ are sufficiently smooth, multiply (1.16) by $v$, and integrate over $\Omega$. Then using Green's second identity (Theorem C.2), where $n$ denotes the outwards pointing unit normal, yields

$$0 = -\int_\Omega v\nabla \cdot (\sigma\nabla u)dx = -\int_{\partial\Omega} \sigma \left( v\frac{\partial u}{\partial n} - u\frac{\partial v}{\partial n}\right) ds - \int_\Omega u\nabla \cdot (\sigma\nabla v)dx,$$

and by use of Green's first identity on the second integral (Theorem C.1 using $\nabla\phi = \sigma\nabla v$ and $\psi = u$) and inserting the boundary condition (1.17)

$$\begin{aligned}0 &= -\int_{\partial\Omega} \sigma v\frac{\partial u}{\partial n}ds + \int_{\partial\Omega} \sigma u\frac{\partial v}{\partial n}ds - \int_{\partial\Omega} \sigma u\frac{\partial v}{\partial n}ds + \int_\Omega \sigma\nabla v \cdot \nabla udx\\ &= \int_\Omega \sigma\nabla u \cdot \nabla vdx - \int_{\partial\Omega} \sigma\frac{\partial u}{\partial n}vds\\ &= \int_\Omega \sigma\nabla u \cdot \nabla vdx - \int_{\partial\Omega} gvds. \qquad (2.1)\end{aligned}$$

Thus (2.1) gives the weak formulation

$$\int_\Omega \sigma\nabla u \cdot \nabla vdx = \int_{\partial\Omega} gvds, \quad u, v \in V. \qquad (2.2)$$

Since $\int_{\partial\Omega} uds = 0$ and we require first order weak-derivatives, we let $V := \tilde{H}^1(\Omega)$. Thus on the right hand-side of (2.2) we have $\int_{\partial\Omega} gTvds$. So we have $Tv \in \tilde{H}^{1/2}(\partial\Omega)$, i.e. a natural space for $g$ could be $\tilde{H}^{-1/2}(\partial\Omega)$. Thus the right hand-side comprise the dual pairing $\langle g, Tv\rangle$.

Now the definition of the forward problem can be collected.

**Definition 2.1 (EIT Forward Problem)** *Let $\Omega$ be an open bounded domain in $\mathbb{R}^d$ (where $d$ is $2$ or $3$). Let $g \in \tilde{H}^{-1/2}(\partial\Omega)$ and let $c_1 \leq \sigma \leq c_2$ for constants $c_1, c_2 > 0$, then $F_g(\sigma)$ denotes the solution to the following PDE problem and $\mathcal{F}_g(\sigma) := TF_g(\sigma)$ denotes the forward EIT map,*

$$-\nabla \cdot (\sigma \nabla u) = 0 \text{ in } \Omega, \tag{2.3}$$

$$\sigma \frac{\partial u}{\partial n} = g \text{ on } \partial\Omega. \tag{2.4}$$

*Furthermore, the weak formulation of the PDE problem is*

$$\int_\Omega \sigma \nabla u \cdot \nabla v dx = \langle g, Tv\rangle, \quad \forall v \in \tilde{H}^1(\Omega). \tag{2.5}$$

It should be noted that if $g \in L^p(\partial\Omega) \cap \tilde{H}^{-1/2}(\partial\Omega), \;\; p \geq 2$, the right hand-side of (2.5) remains $\int_{\partial\Omega} gTvds$ as in (2.2) due to Theorem B.11.

To show that there exists a unique solution to (2.5), we can use the Lax-Milgram theorem (Theorem C.3), however, the theorem requires that $\tilde{H}^1(\Omega)$ is a Hilbert space.

**Lemma 2.2** *$\tilde{H}^1(\Omega)$ is a closed subspace of $H^1(\Omega)$ and therefore a Hilbert space, and there is the following norm-equivalence for constants $C_1, C_2 > 0$:*

$$C_1 \|v\|_{H^1(\Omega)} \leq \|\nabla v\|_{L^2(\Omega)} \leq C_2 \|v\|_{H^1(\Omega)}, \quad \forall v \in \tilde{H}^1(\Omega).$$

*Therefore $\|v\|_{\tilde{H}^1(\Omega)} := \|\nabla v\|_{L^2(\Omega)}$ defines a norm on $\tilde{H}^1(\Omega)$, and $\langle u, v\rangle_{\tilde{H}^1(\Omega)} := \int_\Omega \nabla u \cdot \nabla v dx$ a corresponding inner product.*

**Proof.** Note that for $G : H^1(\Omega) \to \mathbb{R}$ given by $Gv := \int_{\partial\Omega} Tvds$, then

$$\tilde{H}^1(\Omega) = \{v \in H^1(\Omega) \mid Gv = 0\} = \ker(G).$$

As the trace operator $T$ is linear then $G$ is also linear, and clearly using Theorem B.11 and that $T$ is bounded from $H^1(\Omega)$ to $L^2(\partial\Omega)$

$$|Gv| \leq \int_{\partial\Omega} |Tv| ds \leq C\|Tv\|_{L^2(\partial\Omega)} \leq C\|T\|\|v\|_{H^1(\Omega)}, \quad v \in H^1(\Omega),$$

thus $G$ is linear and bounded and therefore continuous, which means that $\tilde{H}^1(\Omega) = \ker(G)$ is a closed subspace of $H^1(\Omega)$ and therefore also a Hilbert space.

Now for the norm equivalence it is easy to see that

$$\|\nabla v\|^2_{L^2(\Omega)} \leq \|v\|^2_{L^2(\Omega)} + \|\nabla v\|^2_{L^2(\Omega)} = \|v\|^2_{H^1(\Omega)}.$$

For the other inequality we use the alternative version of Poincaré's inequality as shown in Theorem B.6, using that $\int Tvds = 0$ for $v \in \tilde{H}^1(\Omega)$:

$$\|v\|^2_{H^1(\Omega)} = \|v\|^2_{L^2(\Omega)} + \|\nabla v\|^2_{L^2(\Omega)} \leq (1 + \tilde{C}^2)\|\nabla v\|^2_{L^2(\Omega)}.$$

Hence the equivalence has been shown, and since $\|\nabla \cdot\|_{L^2(\Omega)}$ is a semi-norm on $H^1(\Omega)$ then it becomes a norm on $\tilde{H}^1(\Omega)$ due to the equivalence with $\|\cdot\|_{H^1(\Omega)}$. It is also not hard to see that $\langle \nabla \cdot, \nabla \cdot \rangle_{L^2(\Omega)}$ also defines an inner product on $\tilde{H}^1(\Omega)$, again due to the norm-equivalence. □

It is now possible to show that there is in fact a unique solution to (2.5) in the space $\tilde{H}^1(\Omega)$.

**Theorem 2.3 (Uniqueness for Forward PDE Problem)**
*With the setup in Definition 2.1, there exists a unique solution $u \in \tilde{H}^1(\Omega)$ to (2.5).*

**Proof.** Let's identify the forms for the Lax-Milgram theorem (Theorem C.3), from (2.5) we have

$$B(u, v) = Lv, \ \ \forall v \in \tilde{H}^1(\Omega), \tag{2.6}$$

where

$$\begin{aligned} B(u, v) &:= \int_\Omega \sigma \nabla u \cdot \nabla v dx, \\ Lv &:= \langle g, Tv \rangle. \end{aligned}$$

Now using that $c_1 \leq \sigma \leq c_2$ and applying Cauchy-Schwartz' inequality:

$$|B(u, v)| \leq c_2 |\langle u, v \rangle_{\tilde{H}^1(\Omega)}| \leq c_2 \|u\|_{\tilde{H}^1(\Omega)} \|v\|_{\tilde{H}^1(\Omega)},$$

furthermore

$$B(u, u) \geq c_1 \|u\|^2_{\tilde{H}^1(\Omega)}. \tag{2.7}$$

Also $B$ is clearly linear in both entries. $L$ is also linear since both $g$ and $T$ are linear operators and using that $g$ and $T$ are bounded then $L$ is also bounded via Lemma 2.2:

$$|Lv| = |\langle g, Tv\rangle| \leq \|g\|_{H^{-1/2}(\Omega)}\|T\|\|v\|_{H^1(\Omega)} \leq c_3\|g\|_{H^{-1/2}(\Omega)}\|T\|\|v\|_{\tilde{H}^1(\Omega)}. \tag{2.8}$$

So by the Lax-Milgram theorem (Theorem C.3) there exists a unique $u \in \tilde{H}^1(\Omega)$ solving (2.5). □

By use of by (2.6), (2.7), and (2.8) we can determine a stability result for $u$ with respect to $g$,

$$\|u\|^2_{\tilde{H}^1(\Omega)} \leq c_1^{-1}|B(u,u)| = c_1^{-1}|Lu| \leq c_1^{-1}c_3\|g\|_{H^{-1/2}(\Omega)}\|T\|\|u\|_{\tilde{H}^1(\Omega)}, \tag{2.9}$$

thus for $C := c_1^{-1}c_3\|T\|$ then

$$\|u\|_{\tilde{H}^1(\Omega)} \leq C\|g\|_{H^{-1/2}(\Omega)}. \tag{2.10}$$

It should be noted that most software packages, including the one I use in Chapter 4 on page 57, does not directly deal with spaces such as $\tilde{H}^1(\Omega)$, therefore the following corollary will be useful for implementation.

**Corollary 2.4** *Let $u \in H^1(\Omega)$ solve the following weak problem*

$$\langle \sigma\nabla u, \nabla v\rangle_{L^2(\Omega)} = \langle g, Tv\rangle + c\int_{\partial\Omega} Tuds, \quad \forall(c,v) \in \mathbb{R} \times H^1(\Omega), \tag{2.11}$$

*then $u = u^*$ with $u^*$ being the unique solution in Theorem 2.3.*

**Proof.** A case for the test functions is $c \neq 0$ and $v = 0$, then (2.11) becomes

$$\int_{\partial\Omega} Tuds = 0,$$

i.e. the term automatically enforces that $u \in \tilde{H}^1(\Omega)$. Now (2.11) must be satisfied for all $v \in H^1(\Omega)$, and the term $\int_{\partial\Omega} Tuds$ vanishes as discussed above, so it reduces to the form (2.5) with $u \in \tilde{H}^1(\Omega)$. As $\tilde{H}^1(\Omega) \subset H^1(\Omega)$, then (2.11) must be satisfied for all $v \in \tilde{H}^1(\Omega)$, and from Theorem 2.3 the only possible solution is $u^*$.

What remains is to consider $v \in H^1(\Omega) \setminus \tilde{H}^1(\Omega)$, in that case define

$$v_c := |\partial\Omega|^{-1}\int_{\partial\Omega} Tvds,$$

where $|\partial\Omega|$ is the Lebesgue measure of $\partial\Omega$ on the boundary. Clearly $v-v_c \in \tilde{H}^1(\Omega)$ since

$$\int_{\partial\Omega} T(v-v_c)ds = \int_{\partial\Omega} Tvds - |\partial\Omega|^{-1} \int_{\partial\Omega} ds \int_{\partial\Omega} Tvds = 0.$$

Now $v-v_c \in \tilde{H}^1(\Omega)$ ensures that (2.5) is satisfied, thus

$$\begin{aligned}\langle \sigma\nabla u^*, \nabla v\rangle - \langle g, Tv\rangle &= \langle \sigma\nabla u^*, \nabla(v-v_c)\rangle - \langle g, T(v-v_c)\rangle \\ &\quad + \langle \sigma\nabla u^*, \nabla v_c\rangle - \langle g, Tv_c\rangle \\ &= -\langle g, Tv_c\rangle,\end{aligned}$$

where the term $\langle \sigma\nabla u^*, \nabla v_c\rangle$ vanished as $v_c$ is constant, i.e. $\nabla v_c = 0$. Now by the linearity of $g$ (note that in this context it is the distribution $g$), and that $g \in \tilde{H}^{-1/2}(\partial\Omega)$ then

$$\langle \sigma\nabla u^*, \nabla v\rangle - \langle g, Tv\rangle = -|\partial\Omega|^{-1}\langle g, 1\rangle \int Tvds = 0, \quad \forall v \in H^1(\Omega). \tag{2.12}$$

Thus $u^*$ is the only possible solution to (2.11), and $u^*$ is in fact a solution when $(c,v) \in \mathbb{R} \times H^1(\Omega)$. □

As a remark to Corollary 2.4, $g$ may in practice not satisfy $\langle g, 1\rangle = 0$ due to noise in the measurements. In that case one can find $(c_1, u) \in \mathbb{R} \times H^1(\Omega)$ solving

$$\langle \sigma\nabla u, \nabla v\rangle_{L^2(\Omega)} = \langle g, Tv\rangle + c_2 \int_{\partial\Omega} Tuds + c_1 \int_{\partial\Omega} Tvds, \quad \forall (c_2, v) \in \mathbb{R}\times H^1(\Omega). \tag{2.13}$$

Note that adding the term $c_1 \int_{\partial\Omega} Tvds$ does not change the fact that $\int_{\partial\Omega} Tuds$ enforces that $u \in \tilde{H}^1(\Omega)$ due to the case $v = 0$. Furthermore, for $v \in \tilde{H}^1(\Omega)$ then $c_1 \int_{\partial\Omega} Tvds$ vanishes, thus the only possible solution is still $u^*$. Therefore $c_1$ becomes $-|\partial\Omega|^{-1}\langle g, 1\rangle$ from (2.12), which ideally should be close to 0. One may think of $c_1$ and $c_2$ as Lagrange multipliers for the variational problem, enforcing the constraints $\int_{\partial\Omega} Tuds = \int_{\partial\Omega} Tvds = 0$.

The data that is recovered from the measurements is assumed to be on the form

$$\phi := Tu + \epsilon = \mathcal{F}_g(\sigma) + \epsilon. \tag{2.14}$$

Since $Tu \in \tilde{H}^{1/2}(\partial\Omega) \subset L^2(\partial\Omega)$ and $\epsilon$ denotes some perturbation due to noise in the measurements, it is assumed that $\phi \in L^2(\partial\Omega)$. It should be noted that there are two types of data in play here, $\phi$ and $g$, where it is assumed that $g$ is known exactly. In order to distinguish between the two data-types $g$ will often be annotated as Neumann-data due to its role in the boundary condition in (2.4), while $\phi$ may be annotated as the measurement data or Dirichlet-data.

It should be noted that $g \mapsto F_g(\sigma)$ is linear in $g$ for fixed $\sigma$ as seen from (2.5) and bounded via (2.10), however, for fixed $g$ then $F_g$ is a non-linear mapping, as $u$ depends on $\sigma$ via (2.4).

Now the hard problem, which is the inverse EIT problem, is to recover a good approximation to $\sigma$ from the Dirichlet-data $\phi$ and the Neumann-data $g$, i.e. we would like to determine $\sigma$ such that $\mathcal{F}_g(\sigma) \simeq \phi$, but still close to the correct conductivity.

A well-posed problem is a problem that has three properties:

(i) There exists a solution.

(ii) The solution is unique.

(iii) The solution depends continuously on the data.

An ill-posed problem is a problem where one or multiple of the three conditions are not satisfied.

Like the forward problem, the inverse problem is highly non-linear and it is also severely ill-posed, meaning that very small perturbations in the exact data (such as noise) makes it exceptionally hard to recover $\sigma$ or anything that resembles $\sigma$ [24]. Thus solving the inverse problem will have to be done by some sort of regularization, for instance by minimizing a Tikhonov functional [24]. So to be able to construct a decent inversion method we need to first investigate the forward problem.

## 2.1 A Simple Example

Let's see if we can actually construct a $\sigma$ and $g$ that allows us to actually determine $F_g(\sigma)$. Let $\Omega$ be the unit disk in $\mathbb{R}^2$, and let $(r, \theta)$ denote polar coordinates, and let $\tilde{r} \in (0, 1)$ then we can consider the following concentric conductivity with $C > 0$:

$$\sigma_{C,\tilde{r}}(r,\theta) := \begin{cases} 1 + C, & 0 \leq r < \tilde{r}, \\ 1 & \tilde{r} \leq r \leq 1. \end{cases} \tag{2.15}$$

An illustration of such a conductivity is shown in Figure 2.1.

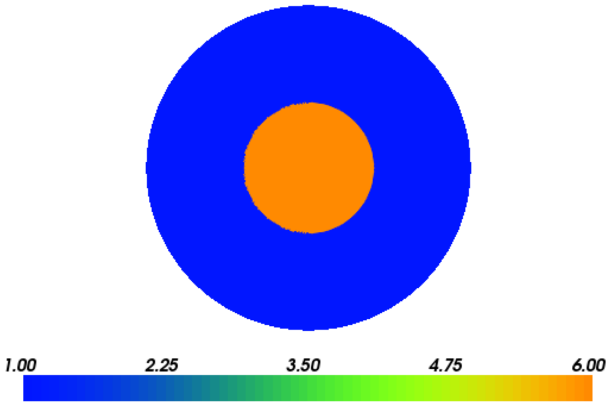


**Figure 2.1:** $\sigma_{C,\tilde{r}}$ for $C = 5$ and $\tilde{r} = 0.4$.

Let $g_N(\theta) := \cos(N\theta)$ be the Neumann-data for some $N \in \mathbb{N}$, and assume that the solution can be separated $u(r, \theta) = R(r)\Theta(\theta)$. Now since $e_\theta \perp n$ with $n$ being the outward normal and $e_\theta$ a unit vector in polar direction of $\theta$, while $e_r \parallel n$ then (2.4) becomes

$$R'(1)\Theta(\theta) = g_N(\theta) = \cos(N\theta),$$

thus we can assume that $\Theta(\theta) := \cos(N\theta)$ and therefore

$$R'(1) = 1. \tag{2.16}$$

Now the requirement $\int_{\partial\Omega} Tuds = 0$ is automatically satisfied in the concentric case when $\int_{\partial\Omega} gds = 0$ as

$$\int_{\partial\Omega} Tuds = R(1) \int_0^{2\pi} \Theta(\theta)d\theta = 0.$$

Now consider the PDE-equation (2.3), let $R := R_1$ for $r < \tilde{r}$ and $R := R_2$ for $r > \tilde{r}$, in both cases we have that $\sigma_{C,\tilde{r}}$ is constant and therefore we have

$$\sigma_{C,\tilde{r}} \Delta u = 0. \tag{2.17}$$

The Laplacian operator is in polar coordinates determined as [16]

$$\begin{aligned} \Delta u &= \frac{1}{r} u_r + u_{rr} + \frac{1}{r^2} u_{\theta\theta} \\ &= \frac{1}{r} R' \Theta + R'' \Theta + \frac{1}{r^2} R \Theta''. \end{aligned}$$

Since $\Theta(\theta) = \cos(N\theta)$ then

$$\Delta u = \left( \frac{1}{r} R' + R'' - \frac{N^2}{r^2} R \right) \Theta,$$

i.e. (2.17) becomes the following, where it is noted that we can choose $\theta$ such that $\Theta \neq 0$:

$$\sigma_{C,\tilde{r}} (r^2 R'' + r R' - N^2 R) = 0.$$

From (2.16) we have $R_2'(1) = 1$, and since we need $u$ and $\sigma_{C,\tilde{r}} u_r$ to be continuous at $r = \tilde{r}$ for (2.3), then $R_2(\tilde{r}) = R_1(\tilde{r})$ and $R_2'(\tilde{r}) = (1+C) R_1'(\tilde{r})$. Furthermore, we do not have a boundary condition at $r = 0$, thus we can impose a boundedness condition assuming that $u$ is bounded near $r = 0$, $|R_1(0)| < \infty$. To summarize we have the following coupled ODE problems:

$$r^2 R_1'' + r R_1' - N^2 R_1 = 0, \quad 0 < r < \tilde{r}, \tag{2.18}$$

$$r^2 R_2'' + r R_2' - N^2 R_2 = 0, \quad \tilde{r} < r < 1, \tag{2.19}$$

$$|R_1(0)| < \infty, \tag{2.20}$$

$$R_2(\tilde{r}) = R_1(\tilde{r}), \tag{2.21}$$

$$R_2'(\tilde{r}) = (1 + C) R_1'(\tilde{r}), \tag{2.22}$$

$$R_2'(1) = 1. \tag{2.23}$$

Looking at the ODE in (2.18) and (2.19), then we recognize this as an Euler equation [16], which has the following solution:

$$\begin{aligned} R_1(r) &= C_1 r^N + C_2 r^{-N}, \quad 0 < r < \tilde{r}, \\ R_2(r) &= C_3 r^N + C_4 r^{-N}, \quad \tilde{r} < r < 1, \end{aligned}$$

where $C_i, \;\; i = 1, 2, 3, 4$ are constants. From (2.20) we can already deduce that $C_2 = 0$. From (2.23) we get

$$1 = N(C_3 - C_4). \tag{2.24}$$

From (2.21) we get

$$\begin{aligned} C_3\tilde{r}^N + C_4\tilde{r}^{-N} &= C_1\tilde{r}^N, \\ C_3 - C_1 + C_4\tilde{r}^{-2N} &= 0. \end{aligned} \tag{2.25}$$

Finally from (2.22) we have

$$\begin{aligned} NC_3\tilde{r}^{N-1} - NC_4\tilde{r}^{-N-1} &= N(1+C)C_1\tilde{r}^{N-1}, \\ (1+C)C_1 - C_3 + C_4\tilde{r}^{-2N} &= 0. \end{aligned} \tag{2.26}$$

Thus (2.24), (2.25), and (2.26) is a linear system of three equations with three unknown:

$$\begin{pmatrix} 0 & N & -N \\ -1 & 1 & \tilde{r}^{-2N} \\ 1+C & -1 & \tilde{r}^{-2N} \end{pmatrix} \begin{pmatrix} C_1 \\ C_3 \\ C_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \tag{2.27}$$

solving (2.27) yields $C_1 = \frac{2}{k_N}$, $C_3 = \frac{C+2}{k_N}$ and $C_4 = \frac{-C\tilde{r}^{2N}}{k_N}$, where $k_N := N(C\tilde{r}^{2N} + C + 2)$.

Thus we have the solution

$$F_{g_N}(\sigma_{C,\tilde{r}})(r,\theta) = \begin{cases} \dfrac{2r^N}{N(C\tilde{r}^{2N}+C+2)}\cos(N\theta), & 0 < r < \tilde{r}, \\ \dfrac{(C+2)r^N - C\tilde{r}^{2N}r^{-N}}{N(C\tilde{r}^{2N}+C+2)}\cos(N\theta), & \tilde{r} < r < 1. \end{cases} \tag{2.28}$$

In Figure 2.2 on the next page an example of (2.28) can be seen, where it is evident that even though the conductivity $\sigma_{C,\tilde{r}}$ is not continuous, the function $F_{g_N}(\sigma_{C,\tilde{r}})$ is.

It is easy to see that we could substitute cos by sin in (2.28) if we used $g_N(\theta) = \sin(N\theta)$.

It was shown in the Theorem 2.3 that solving the PDE by $F_g(\sigma)$ we get a unique solution in $\tilde{H}^1(\Omega)$, however, when considering $\mathcal{F}_g(\sigma) = TF_g(\sigma)$ can we be sure that $\sigma_1 \neq \sigma_2$ does not lead to the same data for the inverse problem? This is an important question when we want to determine the inverse problem, i.e. is the inverse problem well-defined. Of course this boils down to the question of whether $\mathcal{F}_g$ is injective, so let's consider the trace of (2.28), i.e. for $r \to 1$, then

$$\mathcal{F}_{g_N}(\sigma_{C,\tilde{r}})(\theta) = \frac{C+2-C\tilde{r}^{2N}}{N(C\tilde{r}^{2N}+C+2)}\cos(N\theta). \tag{2.29}$$

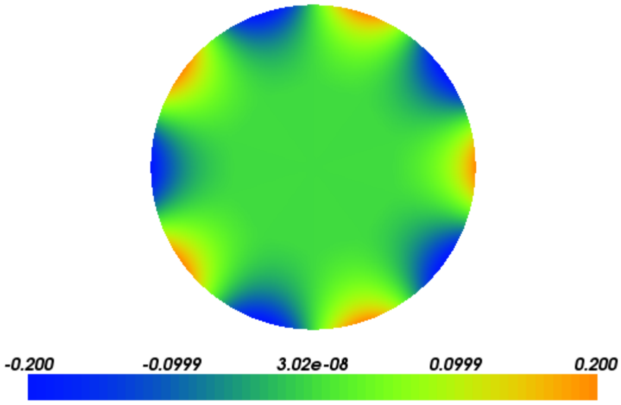


**Figure 2.2:** $F_{g_N}(\sigma_{C,\tilde{r}})$ for $C = 5$, $\tilde{r} = 0.5$, and $N = 5$.

So reducing this to a case of only investigating conductivities of the form (2.15), can we determine $\sigma_{C_1,\tilde{r}_1} \neq \sigma_{C_2,\tilde{r}_2}$ such that $\mathcal{F}_{g_N}(\sigma_{C_1,\tilde{r}_1}) = \mathcal{F}_{g_N}(\sigma_{C_2,\tilde{r}_2})$? Since the cosine-term in (2.29) remains unchanged, let's look at the constant in front of $1/N\cos(N\theta)$:

$$\begin{aligned}\frac{C+2-C\tilde{r}^{2N}}{C\tilde{r}^{2N}+C+2} &= \frac{C+2+C\tilde{r}^{2N}-2C\tilde{r}^{2N}}{C\tilde{r}^{2N}+C+2}\\ &= 1-\frac{2C\tilde{r}^{2N}}{C\tilde{r}^{2N}+C+2}\\ &= 1-\frac{2}{(1+2C^{-1})\tilde{r}^{-2N}+1},\end{aligned}$$

thus the only dependence of $C$ an $\tilde{r}$ in (2.29) is found in $(1+2C^{-1})\tilde{r}^{-2N}$, so let $h_N := (1+2C_1^{-1})\tilde{r}_1^{-2N}$ be fixed and let

$$h_N = (1+2C_2^{-1})\tilde{r}_2^{-2N} \Rightarrow C_2 = \frac{2}{h_N\tilde{r}_2^{2N}-1}. \tag{2.30}$$

Thus if $h_N\tilde{r}_2^{2N}-1 \neq 0$, then (2.30) gives an example where $\mathcal{F}_g$ is not injective, and note that

$$h_N\tilde{r}_2^{2N}-1 = (1+2C_1^{-1})(\tfrac{\tilde{r}_2}{\tilde{r}_1})^{2N}-1,$$

thus for $C_1 > 0$ and $\tilde{r}_2 > \tilde{r}_1$ then $h_N\tilde{r}_2^{2N}-1 > 0$. Hence, for fixed $(C_1,\tilde{r}_1)$ with $C_1 > 0$ then there is an infinity of pairs $(C_2,\tilde{r}_2)$ such that $\mathcal{F}_{g_N}(\sigma_{C_1,\tilde{r}_1}) = \mathcal{F}_{g_N}(\sigma_{C_2,\tilde{r}_2})$. The dependency in (2.30) is illustrated in Figure 2.3 on the following page. Figure 2.6 on page 21 shows three cases with different

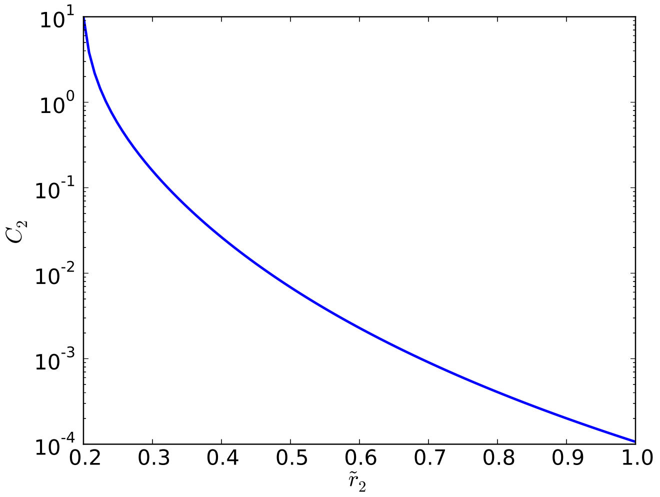


**Figure 2.3:** Plot of (2.30) given $h_N := (1 + 2C_1^{-1})\tilde{r}_1^{-2N}$, with $C_1 := 10$, $\tilde{r}_1 := 0.2$, and $N := 3$.

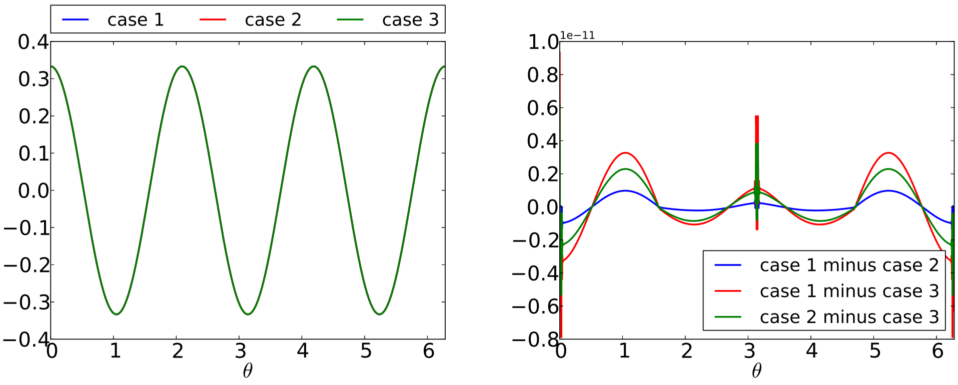
concentric conductivities, that all yields the same forward boundary data $\mathcal{F}_g(\sigma_{C,\tilde{r}})$, and the forward solution on the whole domain $\Omega$ is also shown.

In Figure 2.4 on the facing page the boundary data in the cases from Figure 2.6 on page 21 are shown, and it is seen that the boundary data are identical up to some error of magnitude $10^{-11}$ which is due to rounding errors. From Figure 2.6 it looks like the solutions $F_g(\sigma_{C,\tilde{r}})$ in the three cases are identical everywhere, and not only on the boundary, however that is actually not the case as can be seen in Figure 2.5 on page 20, where the differences are significantly higher than the $10^{-11}$ from the boundary in Figure 2.4b.

So it seems that with a single Neumann-data $g$, then $\mathcal{F}_g$ is not necessarily injective which furthermore shows the ill-posedness of the inverse problem. However, what if we have $g_{N_1}$ and $g_{N_2}$ simultaneously for $N_1 \neq N_2$? Assume that we have $(C_1, \tilde{r}_1)$ and $(C_2, \tilde{r}_2)$ such that $\mathcal{F}_{g_{N_1}}(\sigma_{C_1,\tilde{r}_1}) = \mathcal{F}_{g_{N_1}}(\sigma_{C_2,\tilde{r}_2})$ and $\mathcal{F}_{g_{N_2}}(\sigma_{C_1,\tilde{r}_1}) = \mathcal{F}_{g_{N_2}}(\sigma_{C_2,\tilde{r}_2})$, then

$$(1 + 2C_1^{-1})\tilde{r}_1^{-2N_1} = (1 + 2C_2^{-1})\tilde{r}_2^{-2N_1} \tag{2.31}$$

$$1 + 2C_2^{-1} = (1 + 2C_1^{-1})(\tfrac{\tilde{r}_1}{\tilde{r}_2})^{-2N_1}. \tag{2.32}$$



**(a)** Forward boundary data.

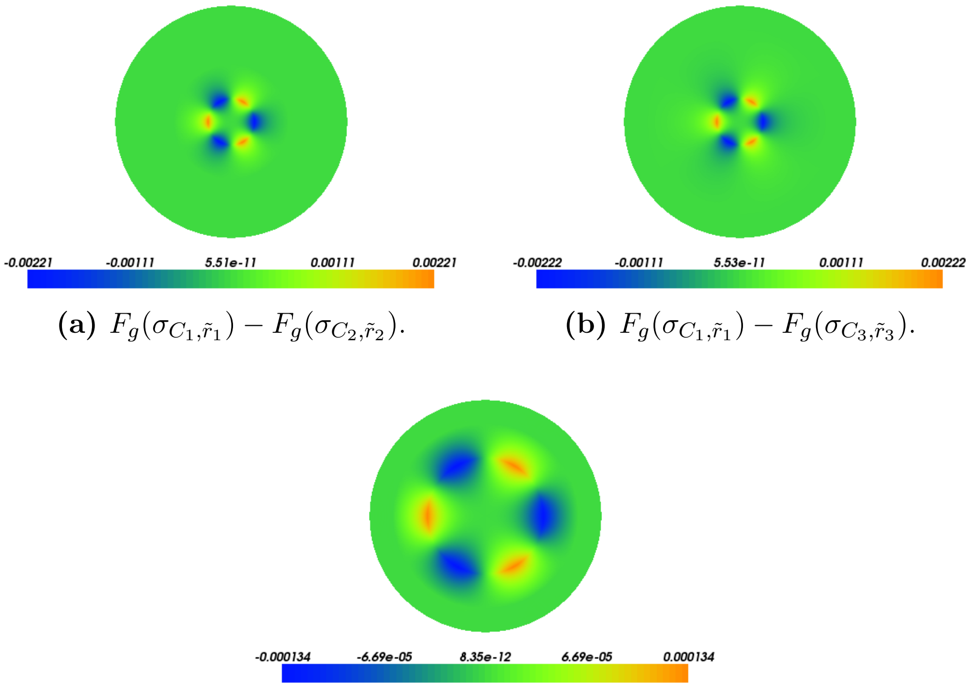
**(b)** Differences in the three cases.

**Figure 2.4:** Forward boundary data $\mathcal{F}_g(\sigma_{C,\tilde{r}})$ from the three cases in Figure 2.6 on page 21.

By substituting (2.32) into the following we get

$$\begin{aligned}(1+2C_1^{-1})\tilde{r}_1^{-2N_2} &= (1+2C_2^{-1})\tilde{r}_2^{-2N_2}\\ (1+2C_1^{-1})(\tfrac{\tilde{r}_1}{\tilde{r}_2})^{-2N_2} &= (1+2C_1^{-1})(\tfrac{\tilde{r}_1}{\tilde{r}_2})^{-2N_1},\end{aligned}$$

as $N_1 \neq N_2$ then the above equation implies that $\tilde{r}_1 = \tilde{r}_2$, which by (2.31) implies that $C_1 = C_2$. So by having multiple Neumann-data, the forward problem became injective (at least with respect to concentric conductivities). The injectivity of the forward operator is an active research subject and is for instance briefly discussed in Kress [10]. In this thesis I will not go further into that subject other than using the *assumption* that with sufficiently many datasets we should be able to recover a good approximation to the sought conductivity $\sigma$. However, when examining the numerical experiments in Chapter 4 on page 57, we find again the phenomenon that an inclusion (i.e. the change from for instance a constant background conductivity) with large support and small amplitude will be mapped by $\mathcal{F}_g$ to similar data as an inclusion with small support and high amplitude. This will serve as a great insight into the behaviour of a regularized solution to the inverse problem.

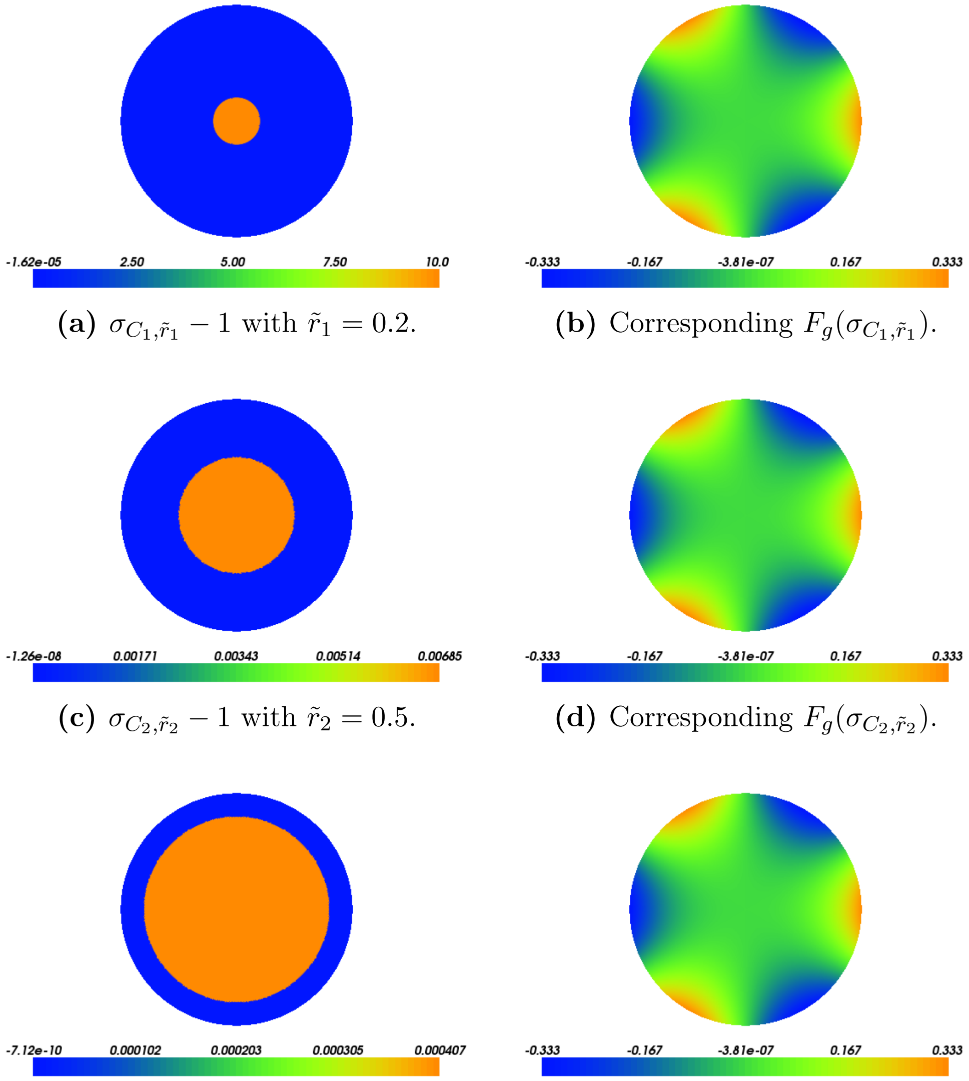


**(a)** $F_g(\sigma_{C_1,\tilde{r}_1}) - F_g(\sigma_{C_2,\tilde{r}_2})$.

**(b)** $F_g(\sigma_{C_1,\tilde{r}_1}) - F_g(\sigma_{C_3,\tilde{r}_3})$.

**(c)** $F_g(\sigma_{C_2,\tilde{r}_2}) - F_g(\sigma_{C_3,\tilde{r}_3})$.

**Figure 2.5:** Differences in the forward data $F_g(\sigma_{C,\tilde{r}})$ in the three cases in Figure 2.6 on the facing page.



**(a)** $\sigma_{C_1,\tilde{r}_1} - 1$ with $\tilde{r}_1 = 0.2$.

**(b)** Corresponding $F_g(\sigma_{C_1,\tilde{r}_1})$.

**(c)** $\sigma_{C_2,\tilde{r}_2} - 1$ with $\tilde{r}_2 = 0.5$.

**(d)** Corresponding $F_g(\sigma_{C_2,\tilde{r}_2})$.

**(e)** $\sigma_{C_3,\tilde{r}_3} - 1$ with $\tilde{r}_3 = 0.8$.

**(f)** Corresponding $F_g(\sigma_{C_3,\tilde{r}_3})$.

**Figure 2.6:** Concentric inclusion that via (2.30) for $N := 3$ yields the same $\mathcal{F}_g(\sigma_{C,\tilde{r}})$. 1 has been subtracted in the plots for the inclusions, such the value of $C$ can be seen from the color bar.

## 2.2 Continuity and Differentiability of the Forward Map

Typically inverse problems are solved iteratively and often by using some sort of regularization to account for the ill-posedness of the problem. Such iterative schemes often make use of the forward problem since it is usually a far more well-posed problem and easily solved numerically. For this reason it is always a good idea to investigate the regularity of the forward problem with respect to continuity and differentiability, in order to apply the forward map in a gradient type iterative method.

Most of the theorems in this section will be concerning $F_g$, however, there will be remarks considering similar results easily derived for $\mathcal{F}_g$ mainly due to the linearity and boundedness of the trace operator.

Firstly we define a special set for which we seek to find $\sigma$, where we assume that there is a known background conductivity $\sigma_0$ and what we in reality seek is an unknown inclusion in the interior of $\Omega$.

**Definition 2.5 (Admissible Set)** *The admissible set $\mathcal{A}$ is for some $\lambda \in (0,1)$ and some known background conductivity $\sigma_0 \in H^1(\Omega)$ given by*

$$\mathcal{A} := \{\sigma \in H^1(\Omega) \mid \lambda \leq \sigma \leq \lambda^{-1} \quad a.e. \text{ in } \Omega, \ T\sigma = T\sigma_0\}$$

*To deal with variations in $\mathcal{A}$ define*

$$\mathcal{A}_0 := \mathcal{A} - \sigma_0 = \{\delta\sigma \in H_0^1(\Omega) \mid \sigma = \sigma_0 + \delta\sigma \in \mathcal{A}\}$$

Definition 2.5 ensures that the conductivities in question are so called inclusions, which means that changes from the known background $\sigma_0$ will have to occur in the interior of $\Omega$. In Jin *et al.* [9] they used $L^\infty$ instead of $H^1$ in the definition of $\mathcal{A}$, however, they later return to $\mathcal{A} \cap H^1$ and the boundedness is already applied with $\lambda \leq \sigma \leq \lambda^{-1}$. So using $H^1$ there are no issues with the statement $T\sigma = T\sigma_0$, while for $L^\infty$ the boundary is not well-defined as it is a set of measure zero. Therefore Jin and Maass [8], Jin *et al.* [9] had to resort to a slightly more convoluted definition, that is a bit more general. In the current thesis we refrain from elaborating further on this given that the authors returned to using $\mathcal{A} \cap H^1$ later on.

**Corollary 2.6** *$\mathcal{A}$, and therefore also $\mathcal{A}_0$, is convex and closed in $H^1(\Omega)$.*

**Proof.** The properties will be shown in two separate steps.

**$\mathcal{A}$ is closed:** Let $\{x_k\} \subseteq \mathcal{A}$ such that $\lim_{k\to\infty}\|x_k - x\|_{H^1(\Omega)} = 0$ for some $x \in H^1(\Omega)$. What remains is to show that $x \in \mathcal{A}$. Firstly it must be shown that $Tx = T\sigma_0$, for which it is applied that $T$ is linear and bounded, and $Tx_k = T\sigma_0, \ \forall k$,

$$\begin{aligned}\|Tx - T\sigma_0\|_{L^2(\partial\Omega)} &= \|Tx - Tx_k\|_{L^2(\partial\Omega)} \\ &\le \|T\|\|x - x_k\|_{H^1(\Omega)} \to 0 \text{ for } k \to \infty,\end{aligned}$$

i.e. we indeed have $Tx = T\sigma_0$. Now since $H^1(\Omega) \hookrightarrow L^2(\Omega)$ (see Definition B.8 on page 148 for the notation on imbeddings) then $x_k \to x$ in $L^2(\Omega)$, which by Theorem C.6 implies that there exists a subsequence $\{x_{k_j}\}$ which converges pointwise to $x$. Since $\lambda \le x_{k_j} \le \lambda^{-1}, a.e. \ \forall k_j$ then it follows from the pointwise convergence that $\lambda \le x \le \lambda^{-1}, a.e.$. Thus $x \in \mathcal{A}$ and $\mathcal{A}$ is therefore closed in $H^1(\Omega)$.

**$\mathcal{A}$ is convex:** Let $x, y \in \mathcal{A}$ and let $t \in [0,1]$, what needs to be shown is that $z := (1-t)x + ty \in \mathcal{A}$. Due to $H^1(\Omega)$ being a vector space then clearly $z \in H^1(\Omega)$. Also as $T$ is linear and $Tx = Ty = T\sigma_0$ then

$$Tz = (1-t)Tx + tTy = (1-t)T\sigma_0 + tT\sigma_0 = T\sigma_0.$$

Finally, as $t \in [0,1]$ then $t, 1-t \ge 0$, thus we have

$$\begin{aligned}z &= (1-t)x + ty \le (1-t)\lambda^{-1} + t\lambda^{-1} = \lambda^{-1} \ \ a.e., \\ z &= (1-t)x + ty \ge (1-t)\lambda + t\lambda = \lambda \ \ a.e.,\end{aligned}$$

i.e. $\lambda \le z \le \lambda^{-1}, \ \ a.e.$ so $z \in \mathcal{A}$. □

The following theorem is stated in several different version in Jin and Maass [8], Jin *et al.* [9], but I apply the one below given in theorem 3.1 in Jin and Maass [8], which again is collected by results in Gallouet and Monier [6], Meyers [11]. The admissible set $\mathcal{A}$ and $\lambda$ are defined in Definition 2.5.

**Theorem 2.7 (Meyers' Gradient Estimate)** *Let $\Omega \subset \mathbb{R}^d$ for $d = 2$ or $d = 3$, where $\Omega$ is open and bounded and has smooth boundary. For any $\sigma \in \mathcal{A}$ there exists a constant $Q$ depending only on $d$ and $\lambda$, such that $\lim_{\lambda\to 1} Q = \infty$ and $\lim_{\lambda\to 0} Q = 2$. For any $q \in (2, Q)$, any $s \in [q - \frac{q}{d}, \infty]$, for $g \in L^s(\partial\Omega) \cap \tilde{H}^{-1/2}(\partial\Omega)$ and for $f \in (L^q(\Omega))^d$, then the solution $u$ to*

$$-\nabla \cdot (\sigma \nabla u) = \nabla \cdot f \quad in \quad \Omega, \qquad \sigma\frac{\partial u}{\partial n} = g \quad on \quad \partial\Omega,$$

*satisfies the following estimate with $C$ only depending on $d, \lambda, \Omega$, and $q$:*

$$\|u\|_{W^{1,q}(\Omega)} \leq C(\|g\|_{L^s(\partial\Omega)} + \|f\|_{L^q(\Omega)}).$$

For the remainder of this section $Q$ will be the constant from Theorem 2.7. Furthermore, $g$ will be fixed and $g \in L^s(\partial\Omega) \cap \tilde{H}^{-1/2}(\partial\Omega)$ for $s \in (q - \frac{q}{d}, \infty]$ where $\frac{1}{q} + \frac{1}{p} = \frac{1}{2}$ and $p$ is chosen in the following theorems. The results also include the case for $p = \infty$, in which case $q = 2$, which is not part of Theorem 2.7. However, if $f = 0$ in Theorem 2.7 then we can let $s \geq 2$, and as in (2.9) we have the following where Cauchy-Schwartz' inequality and Lemma 2.2 are applied

$$\begin{aligned}\lambda\|u\|^2_{\tilde{H}^1(\Omega)} &\leq |Lu| = |\langle g, Tu\rangle_{L^2(\partial\Omega)}| \leq \|g\|_{L^2(\partial\Omega)}\|Tu\|_{L^2(\partial\Omega)} \\ &\leq \|g\|_{L^2(\partial\Omega)}\|T\|\|u\|_{H^1(\Omega)} \leq C_1\|T\|\|u\|_{\tilde{H}^1(\Omega)}\|g\|_{L^2(\partial\Omega)}.\end{aligned}$$

So by Theorem B.11 then

$$\|u\|_{\tilde{H}^1(\Omega)} \leq \lambda^{-1}C_1\|T\|C_2\|g\|_{L^s(\Omega)}.$$

The following theorem shows that $F_g$ is Lipschitz continuous from certain $L^p(\Omega)$-spaces to $\tilde{H}^1(\Omega)$. Note that $\mathcal{A} \subset L^\infty(\Omega)$ thus by Theorem B.11 then $\mathcal{A} \subset L^p(\Omega),\ \ p \in [1, \infty]$.

**Theorem 2.8** *Let $\sigma, \sigma + \eta \in \mathcal{A}$ and $p \in (\frac{2Q}{Q-2}, \infty]$ then there exists a constant $C > 0$ such that*

$$\|F_g(\sigma + \eta) - F_g(\sigma)\|_{\tilde{H}^1(\Omega)} \leq C\|\eta\|_{L^p(\Omega)}.$$

**Proof.** Writing up the weak formulations of $F_g(\sigma)$ and $F_g(\sigma + \eta)$ we get from (2.5):

$$\int_\Omega \sigma\nabla F_g(\sigma)\cdot\nabla v dx = \int_{\partial\Omega} gTv ds = \int_\Omega (\sigma+\eta)\nabla F_g(\sigma+\eta)\cdot\nabla v dx,\ \ \forall v \in \tilde{H}^1(\Omega),$$

thus

$$\int_\Omega \sigma\nabla(F_g(\sigma) - F_g(\sigma + \eta)) \cdot \nabla v dx = \int_\Omega \eta\nabla F_g(\sigma + \eta) \cdot \nabla v dx,\ \ \forall v \in \tilde{H}^1(\Omega). \tag{2.33}$$

Now let $v := F_g(\sigma) - F_g(\sigma + \eta)$ then (2.33) becomes

$$\int_\Omega \sigma|\nabla(F_g(\sigma) - F_g(\sigma+\eta))|^2 dx = \int_\Omega \eta\nabla F_g(\sigma+\eta) \cdot \nabla(F_g(\sigma) - F_g(\sigma+\eta))dx. \tag{2.34}$$

Now by use the generalized Hölder's inequality (Theorem B.10) with $q$ chosen such that $\frac{1}{p}+\frac{1}{q}=\frac{1}{2}$, then from (2.34) we get

$$\begin{aligned}&\int_\Omega \sigma|\nabla(F_g(\sigma)-F_g(\sigma+\eta))|^2 dx\\ &\le \|\eta\|_{L^p(\Omega)}\|\nabla F_g(\sigma+\eta)\|_{L^q(\Omega)}\|\nabla(F_g(\sigma)-F_g(\sigma+\eta))\|_{L^2(\Omega)}. \qquad (2.35)\end{aligned}$$

Due to $\frac{1}{p}+\frac{1}{q}=\frac{1}{2}$ and $p\in(\frac{2Q}{Q-2},\infty]$ then $q\in[2,Q)$, and since $\sigma,\sigma+\eta\in\mathcal{A}$ then $\eta\in L^\infty(\Omega)$ so $\eta\in L^p(\Omega)$ by Theorem B.11.

Now using Theorem 2.7 we have

$$\|\nabla F_g(\sigma+\eta)\|_{L^q(\Omega)}\le\|F_g(\sigma+\eta)\|_{W^{1,q}(\Omega)}\le C_1\|g\|_{L^s(\Omega)}. \qquad (2.36)$$

By (2.35) and (2.36) then

$$\begin{aligned}\lambda\|\nabla(F_g(\sigma)-F_g(\sigma+\eta))\|^2_{L^2(\Omega)}&\le\int_\Omega\sigma|\nabla(F_g(\sigma)-F_g(\sigma+\eta))|^2dx\\ &\le C_1\|g\|_{L^s(\Omega)}\|\nabla(F_g(\sigma)-F_g(\sigma+\eta))\|_{L^2(\Omega)}\|\eta\|_{L^p(\Omega)}, \qquad (2.37)\end{aligned}$$

thus by (2.37) and by setting $C:=\lambda^{-1}C_1\|g\|_{L^s(\Omega)}$ then

$$\|F_g(\sigma)-F_g(\sigma+\eta)\|_{\tilde{H}^1(\Omega)}=\|\nabla(F_g(\sigma)-F_g(\sigma+\eta))\|_{L^2(\Omega)}\le C\|\eta\|_{L^p(\Omega)}.$$

□

Note that by the linearity and boundedness of $T$ and by Lemma 2.2 then it follows trivially from Theorem 2.8 that

$$\begin{aligned}\|\mathcal{F}_g(\sigma+\eta)-\mathcal{F}_g(\sigma)\|_{L^2(\partial\Omega)}&\le\|T\|\|F_g(\sigma+\eta)-F_g(\sigma)\|_{H^1(\Omega)}\\ &\le\|T\|C_1\|F_g(\sigma+\eta)-F_g(\sigma)\|_{\tilde{H}^1(\Omega)}\\ &\le\|T\|C_1C\|\eta\|_{L^p(\Omega)},\end{aligned}$$

thus the same type of continuity result holds for $\mathcal{F}_g$.

Now let's attempt to differentiate $F_g$. This will be done by a linearisation for which we consider the governing equations of $u_1=F_g(\sigma)$ and $u_2=F_g(\sigma+\eta)$ in (2.3) and (2.4) for $\sigma,\sigma+\eta\in\mathcal{A}$, and subtract these equations (while utilizing the constraint $\eta|_{\partial\Omega}=0$ that arises from $\sigma,\sigma+\eta\in\mathcal{A}$):

$$-\nabla\cdot(\sigma\nabla(u_2-u_1))=\nabla\cdot(\eta\nabla u_2),\ \text{in }\Omega, \qquad (2.38)$$

$$\sigma\frac{\partial(u_2-u_1)}{\partial n}=g-g=0,\ \text{on }\partial\Omega. \qquad (2.39)$$

Thus if we set $w := u_2 - u_1$ then we get the following system

$$-\nabla \cdot (\sigma \nabla w) = \nabla \cdot (\eta \nabla u_2), \text{ in } \Omega, \tag{2.40}$$

$$\sigma \frac{\partial w}{\partial n} = 0, \text{ on } \partial\Omega. \tag{2.41}$$

Now we can investigate how $w$ depends on $\eta$, where Theorem 2.8 with $p = \infty$ is applied

$$\|w\|_{\tilde{H}^1(\Omega)} = \|u_2 - u_1\|_{\tilde{H}^1(\Omega)} \leq C\|\eta\|_{L^\infty(\Omega)}.$$

So rewriting the right hand-side of (2.40) we get

$$\nabla \cdot (\eta \nabla u_2) = \nabla \cdot (\eta \nabla (u_1 + w)) = \nabla \cdot (\eta \nabla u_1) + \nabla \cdot (\eta \nabla w),$$

thus the first term $\nabla \cdot (\eta \nabla u_1)$ is of order $O(|\eta|)$ while the second term $\nabla \cdot (\eta \nabla w)$ is of order $O(|\eta|^2)$. Assuming $\eta$ is a small perturbation, we ignore the higher order term, leading to the right hand-side of (2.40) being replaced by $\nabla \cdot (\eta \nabla u_1)$ resulting in the following PDE problem

$$-\nabla \cdot (\sigma \nabla w) = \nabla \cdot (\eta \nabla F_g(\sigma)), \text{ in } \Omega, \tag{2.42}$$

$$\sigma \frac{\partial w}{\partial n} = 0, \text{ on } \partial\Omega. \tag{2.43}$$

It is easy to deduce the weak formulation (2.46) of (2.42) and (2.43) as the PDE problem is very close to that of Definition 2.1, when $g = 0$ and the right hand-side of the PDE is non-zero. So using the weak formulation (2.5) with $g = 0$ and adding the right hand-side of (2.42) multiplied with $v$ and integrated over $\Omega$:

$$\int_\Omega \sigma \nabla w \cdot \nabla v dx = \int_\Omega \nabla \cdot (\eta \nabla F_g(\sigma)) v dx.$$

Now by use of Theorem C.1 (with $\psi = v$ and $\nabla \phi = \eta \nabla F_g(\sigma)$):

$$\int_\Omega \sigma \nabla w \cdot \nabla v dx = -\int_\Omega \eta \nabla F_g(\sigma) \cdot \nabla v dx + \int_{\partial\Omega} v \eta \frac{\partial F_g(\sigma)}{\partial n} ds.$$

What remains is to show that $\int_{\partial\Omega} v\eta \frac{\partial F_g(\sigma)}{\partial n} ds = 0$, however, this is easy as $\sigma, \sigma + \eta \in \mathcal{A}$ i.e. $\sigma|_{\partial\Omega} = \sigma_0|_{\partial\Omega}$ and $(\sigma + \eta))|_{\partial\Omega} = \sigma_0|_{\partial\Omega}$ therefore $\eta|_{\partial\Omega} = 0$ and the term vanishes.

**Definition 2.9** *Let $\sigma, \sigma + \eta \in \mathcal{A}$ then $(F_g)'_\sigma \eta$ denotes the linear mapping of $\eta$ to $w \in \tilde{H}^1(\Omega)$ as the solution of*

$$-\nabla \cdot (\sigma \nabla w) = \nabla \cdot (\eta \nabla F_g(\sigma)), \ \ in\ \Omega, \tag{2.44}$$

$$\sigma \frac{\partial w}{\partial n} = 0, \ \ on\ \partial\Omega. \tag{2.45}$$

*The weak formulation of the above is*

$$\int_\Omega \sigma \nabla (F_g)'_\sigma \eta \cdot \nabla v dx = -\int_\Omega \eta \nabla F_g(\sigma) \cdot \nabla v dx, \ \ \forall v \in \tilde{H}^1(\Omega). \tag{2.46}$$

From (2.46) it is easy to see that $(F_g)'_\sigma$ is linear.

**Theorem 2.10** *With the setup in Definition 2.9, then $(F_g)'_\sigma$ is well-defined, i.e. there is a unique solution to (2.46) in $\tilde{H}^1(\Omega)$.*

**Proof.** Note that the bilinear form for the Lax-Milgram Theorem (Theorem C.3) is identical to that in the proof of Theorem 2.3, thus what remains is to show that $L$ defined by $Lv := -\int_\Omega \eta \nabla F_g(\sigma) \cdot \nabla v dx$ is a linear and bounded functional on $\tilde{H}^1(\Omega)$. Since $\sigma, \sigma + \eta \in \mathcal{A}$, i.e. $\lambda \leq \sigma \leq \lambda^{-1}$ and $\lambda \leq \sigma + \eta \leq \lambda^{-1}$, then we have $|\eta| \leq \lambda^{-1} - \lambda$, so by Cauchy-Schwartz' inequality:

$$\begin{aligned} |Lv| &\leq (\lambda^{-1} - \lambda) |\langle F_g(\sigma), v \rangle_{\tilde{H}^1(\Omega)}| \\ &\leq (\lambda^{-1} - \lambda) \|F_g(\sigma)\|_{\tilde{H}^1(\Omega)} \|v\|_{\tilde{H}^1(\Omega)}, \ \ v \in \tilde{H}^1(\Omega). \end{aligned}$$

Thus as $F_g(\sigma) \in \tilde{H}^1(\Omega)$ then $L$ is bounded, and $L$ is clearly also linear, i.e. by Theorem C.3 there is a unique solution in $\tilde{H}^1(\Omega)$ to (2.46). □

Now we can investigate boundedness of the linearised operator $(F_g)'_\sigma$.

**Theorem 2.11** *Let $\sigma, \sigma + \eta \in \mathcal{A}$ and $p \in (\frac{2Q}{Q-2}, \infty]$ then*

$$\|(F_g)'_\sigma \eta\|_{\tilde{H}^1(\Omega)} \leq C \|\eta\|_{L^p(\Omega)}.$$

**Proof.** The proof is very similar to that of Theorem 2.8. From the weak form (2.46) set $v := (F_g)'_\sigma \eta$:

$$\int_\Omega \sigma |\nabla (F_g)'_\sigma \eta|^2 dx = -\int_\Omega \eta \nabla F(\sigma) \cdot \nabla (F_g)'_\sigma \eta dx. \tag{2.47}$$

Now using Theorem B.10 with $q$ such that $\frac{1}{q}+\frac{1}{p}=\frac{1}{2}$, and as in the proof of Theorem 2.8 this implies that $q \in [2, Q)$:

$$\begin{aligned}\int_\Omega \sigma|\nabla(F_g)'_\sigma\eta|^2 dx &\leq \|\eta\|_{L^p(\Omega)}\|\nabla F(\sigma)\|_{L^q(\Omega)}\|\nabla(F_g)'_\sigma\eta\|_{L^2(\Omega)} \\ &\leq \|\eta\|_{L^p(\Omega)}\|F(\sigma)\|_{W^{1,q}(\Omega)}\|\nabla(F_g)'_\sigma\eta\|_{L^2(\Omega)}. \end{aligned} \tag{2.48}$$

Using Theorem 2.7 then

$$\int_\Omega \sigma|\nabla(F_g)'_\sigma\eta|^2 dx \leq C_1\|g\|_{L^s(\Omega)}\|\nabla(F_g)'_\sigma\eta\|_{L^2(\Omega)}\|\eta\|_{L^p(\Omega)},$$

thus

$$\lambda\|\nabla(F_g)'_\sigma\eta\|^2_{L^2(\Omega)} \leq \int_\Omega \sigma|\nabla(F_g)'_\sigma\eta|^2 dx \leq C_1\|g\|_{L^s(\Omega)}\|\nabla(F_g)'_\sigma\eta\|_{L^2(\Omega)}\|\eta\|_{L^p(\Omega)}. \tag{2.49}$$

So by setting $C := \lambda^{-1}C_1\|g\|_{L^s(\Omega)}$ then (2.49) becomes

$$\|(F_g)'_\sigma\eta\|_{\tilde{H}^1(\Omega)} = \|\nabla(F_g)'_\sigma\eta\|_{L^2(\Omega)} \leq C\|\eta\|_{L^p(\Omega)}.$$ □

Now we will see that $(F_g)'_\sigma$ is sort of a Frechét derivative of $F_g$ at $\sigma$ when $(F_g)'_\sigma$ is considered as a map from certain $L^p(\Omega)$-spaces (see Definition C.4). The notion *sort of* is due to $\mathcal{A}$ or rather $\mathcal{A}-\sigma$ is not a space.

**Theorem 2.12** *Let $\sigma, \sigma+\eta \in \mathcal{A}$ and $p \in (\frac{2Q}{Q-2}, \infty]$ then*

$$\lim_{\|\eta\|_{L^p(\Omega)}\to 0} \frac{\|F(\sigma+\eta)-F(\sigma)-(F_g)'_\sigma\eta\|_{\tilde{H}^1(\Omega)}}{\|\eta\|_{L^p(\Omega)}} = 0. \tag{2.50}$$

**Proof.** Let $u_1 := F_g(\sigma+\eta)$, $u_2 := F_g(\sigma)$ and $u_3 := (F_g)'_\sigma\eta$, then their weak formulations are by Definition 2.1 and Definition 2.9 given by

$$\begin{aligned}\int_\Omega (\sigma+\eta)\nabla u_1\cdot\nabla v dx &= \int_{\partial\Omega} gTv ds, \quad \forall v \in \tilde{H}^1(\Omega), \\ \int_\Omega \sigma\nabla u_2\cdot\nabla v dx &= \int_{\partial\Omega} gTv ds, \quad \forall v \in \tilde{H}^1(\Omega), \\ \int_\Omega \sigma\nabla u_3\cdot\nabla v dx &= -\int_\Omega \eta\nabla u_2\cdot\nabla v dx, \quad \forall v \in \tilde{H}^1(\Omega).\end{aligned}$$

Now define $w \in \tilde{H}^1(\Omega)$ as $w := u_1 - u_2 - u_3$ and insert the above weak forms into the following expression:

$$\begin{aligned}\int_\Omega (\sigma+\eta)\nabla w \cdot \nabla v dx &= \int_{\partial\Omega} gTv ds - \int_{\partial\Omega} gTv ds - \int_\Omega \eta\nabla u_2 \cdot \nabla v dx \\ &\quad + \int_\Omega \eta\nabla u_2 \cdot \nabla v dx - \int_\Omega \eta\nabla u_3 \cdot \nabla v dx \\ &= -\int_\Omega \eta\nabla (F_g)'_\sigma \eta \cdot \nabla v dx, \;\; \forall v \in \tilde{H}^1(\Omega). \qquad (2.51)\end{aligned}$$

Now let $v := w$ in (2.51):

$$\int_\Omega (\sigma+\eta)|\nabla w|^2 dx = -\int_\Omega \eta\nabla (F_g)'_\sigma \eta \cdot \nabla w dx. \qquad (2.52)$$

As in the proof of the previous two theorems we now apply Theorem B.10 with $q$ such that $\frac{1}{p}+\frac{1}{q}=\frac{1}{2}$:

$$\int_\Omega (\sigma+\eta)|\nabla w|^2 dx \leq \|\eta\|_{L^p(\Omega)}\|\nabla(F_g)'_\sigma\eta\|_{L^q(\Omega)}\|\nabla w\|_{L^2(\Omega)},$$

thus as in the proofs of the previous theorems

$$\begin{aligned}\|w\|_{\tilde{H}^1(\Omega)} = \|\nabla w\|_{L^2(\Omega)} &\leq \lambda^{-1}\|\eta\|_{L^p(\Omega)}\|\nabla(F_g)'_\sigma\eta\|_{L^q(\Omega)} \\ &\leq \lambda^{-1}\|\eta\|_{L^p(\Omega)}\|(F_g)'_\sigma\eta\|_{W^{1,q}(\Omega)}. \qquad (2.53)\end{aligned}$$

Notice that the fraction in (2.50) is in fact $\|w\|_{\tilde{H}^1(\Omega)}/\|\eta\|_{L^p(\Omega)}$, thus what remains in order to prove (2.50) is to show that $\|(F_g)'_\sigma\eta\|_{W^{1,q}(\Omega)} \to 0$ when $\|\eta\|_{L^p(\Omega)} \to 0$.

**Showing that $\|(F_g)'_\sigma\eta\|_{W^{1,q}(\Omega)} \to 0$ when $\|\eta\|_{L^p(\Omega)} \to 0$:**

Note that since $\frac{1}{p}+\frac{1}{q}=\frac{1}{2}$ and $p \in (\frac{2Q}{Q-2},\infty]$ then $q \in [2,Q)$. The case $q=2$ has already been covered in Theorem 2.11, so let's assume $q \in (2,Q)$. From Theorem 2.7 then we have

$$\|(F_g)'_\sigma\eta\|_{W^{1,q}(\Omega)} \leq C\|\eta\nabla F_g(\sigma)\|_{L^q(\Omega)}. \qquad (2.54)$$

Now define $\tilde{q} := q+\epsilon \in (q,Q)$ for some arbitrarily small $\epsilon > 0$. Define

$$\tilde{p} := \frac{\tilde{q}q}{\tilde{q}-q},$$

then as $q \in (2, Q)$ we have that $\tilde{p} = \frac{q^2}{\epsilon} + q > \frac{4}{\epsilon} + 2 > 2$, and we have $\frac{1}{\tilde{q}} + \frac{1}{\tilde{p}} = \frac{1}{q}$, thus by Theorem B.10:

$$\|\eta \nabla F_g(\sigma)\|_{L^q(\Omega)} \leq \|\eta\|_{L^{\tilde{p}}(\Omega)} \|\nabla F_g(\sigma)\|_{L^{\tilde{q}}(\Omega)} \leq \|\eta\|_{L^{\tilde{p}}(\Omega)} \|F_g(\sigma)\|_{W^{1,\tilde{q}}(\Omega)}. \tag{2.55}$$

In order to apply Theorem 2.7 to $\|F_g(\sigma)\|_{W^{1,\tilde{q}}(\Omega)}$ it is noted that $\tilde{q} \in (q, Q)$, however, we also need to restrain $s \in [\tilde{q} - \frac{\tilde{q}}{d}, \infty] = [q - \frac{q}{d} + \frac{d-1}{d}\epsilon, \infty]$. Note that $\epsilon$ can be chosen arbitrarily small. Thus for $s \in (q - \frac{q}{d}, \infty]$, we can use Theorem 2.7 which is exactly how $s$ was chosen in the beginning of this section. Hence, applying (2.54) and Theorem 2.7 to (2.55):

$$\|(F_g)'_\sigma \eta\|_{W^{1,q}(\Omega)} \leq C C_1 \|g\|_{L^s(\Omega)} \|\eta\|_{L^{\tilde{p}}(\Omega)}. \tag{2.56}$$

Now let's look at the two cases $\tilde{p} \leq p$ and $\tilde{p} > p$. If $\tilde{p} \leq p$ then by Theorem B.11 we have

$$\tilde{p} \leq p \Rightarrow \|\eta\|_{L^{\tilde{p}}(\Omega)} \leq C_2 \|\eta\|_{L^p(\Omega)}. \tag{2.57}$$

In the case of $\tilde{p} > p$ then $\tilde{p} - p > 0$, and $\sigma, \sigma + \eta \in \mathcal{A}$ entails $|\eta| \leq \lambda^{-1} - \lambda$ so

$$\|\eta\|^{\tilde{p}}_{L^{\tilde{p}}(\Omega)} = \int_\Omega |\eta|^{\tilde{p}} dx = \int_\Omega |\eta|^{\tilde{p}-p} |\eta|^p dx \leq (\lambda^{-1} - \lambda)^{\tilde{p}-p} \|\eta\|^p_{L^p(\Omega)},$$

thus

$$\tilde{p} > p \Rightarrow \|\eta\|^{\tilde{p}}_{L^{\tilde{p}}(\Omega)} \leq C_3 \|\eta\|^p_{L^p(\Omega)}. \tag{2.58}$$

By (2.56), (2.57) and (2.58) we get

$$\lim_{\|\eta\|_{L^p(\Omega)} \to 0} \|(F_g)'_\sigma \eta\|_{W^{1,q}(\Omega)} = 0.$$

□

Now we may consider $(F_g)'_\sigma$ in terms of a derivative in the $H^1$-metric, by imbedding $H^1$ into $L^p$-spaces. Since $H^1(\Omega)$ does not continuously embed into $L^\infty(\Omega)$ for $d = 2$ (Theorem B.9), then the $L^p$-estimates derived in the prior theorems provide more flexibility than the standard result for $L^\infty$.

**Corollary 2.13** *If $d = 2$ then $F_g$ is differentiable in the $H^1$-metric with Frechét derivative $(F_g)'_\sigma$ with perturbations in $\mathcal{A}$. If $d = 3$ then the same holds if $\lambda$ is sufficiently close to 1.*

**Proof.** The proof will consist of imbedding $H^1(\Omega)$ into some of the $L^p$-spaces used in Theorem 2.12.

For $d = 2$ then $H^1(\Omega)$ imbeds continuously into $L^p(\Omega)$ for $p \in [1, \infty)$ by Theorem B.9 and Theorem B.11.

For $d = 3$ then by Theorem B.9 and Theorem B.11 we have that $H^1(\Omega)$ continuously imbeds into $L^p(\Omega)$ for $p \leq 6$. For Theorem 2.7 and Theorem 2.12 we need to have $Q(\lambda) > 2$ and $\frac{2Q(\lambda)}{Q(\lambda)-2} < p \leq 6$, thus we need $Q(\lambda) > 3$. By Theorem 2.7 then $Q$ tends to $\infty$ as $\lambda$ tends to 1, and therefore we have $H^1(\Omega)$ imbedded into $L^p(\Omega)$ for $p \in (\frac{2Q}{Q-2}, 6]$ when $\lambda$ is sufficiently close to 1.

Thus in Theorem 2.12 we can approximate $\|\eta\|_{L^p(\Omega)} \leq C\|\eta\|_{H^1(\Omega)}$, and yield the desired result. □

Now we can easily find the derivative $(\mathcal{F}_g)'_\sigma$ simply by applying that $T$ is linear and bounded and by applying Lemma 2.2. We have that

$$\begin{aligned}&\|\mathcal{F}_g(\sigma+\eta) - \mathcal{F}_g(\sigma) - T(F_g)'_\sigma\eta\|_{L^2(\partial\Omega)}\\ &\qquad \leq \|T\|\|F_g(\sigma+\eta) - F_g(\sigma) - (F_g)'_\sigma\eta\|_{H^1(\Omega)}\end{aligned}$$

thus by Theorem 2.12 and Corollary 2.13 then $(\mathcal{F}_g)'_\sigma = T(F_g)'_\sigma$.

Finally we present the following result, which will play a role in determining the derivative of the discrepancy term in Chapter 3.

**Lemma 2.14** *Let $\sigma, \sigma+\eta \in \mathcal{A}$, and $f \in L^2(\partial\Omega)$, then*

$$\langle f, (\mathcal{F}_g)'_\sigma\eta\rangle_{L^2(\partial\Omega)} = -\langle\nabla F_g(\sigma)\cdot\nabla F_f(\sigma), \eta\rangle_{L^2(\Omega)}. \tag{2.59}$$

**Proof.** The weak form of $F_f(\sigma)$ and $(F_g)'_\sigma\eta$ are by Definition 2.1 and Definition 2.9 given by

$$\int_\Omega \sigma\nabla F_f(\sigma)\cdot\nabla v dx = \int_{\partial\Omega} fTv ds, \quad v \in \tilde{H}^1(\Omega), \tag{2.60}$$

$$\int_\Omega \sigma\nabla(F_g)'_\sigma\eta\cdot\nabla v dx = -\int_\Omega \eta\nabla F_g(\sigma)\cdot\nabla v dx, \quad v \in \tilde{H}^1(\Omega). \tag{2.61}$$

Now let $v := (F_g)'_\sigma\eta$ in (2.60) and $v := F_f(\sigma)$ in (2.61), then as $T(F_g)'_\sigma = (\mathcal{F}_g)'_\sigma$ we get

$$\int_{\partial\Omega} f(\mathcal{F}_g)'_\sigma\eta ds = \int_\Omega \sigma\nabla(F_g)'_\sigma\eta\cdot\nabla F_f(\sigma)dx = -\int_\Omega \eta\nabla F_g(\sigma)\cdot\nabla F_f(\sigma)dx,$$

as a consequence

$$\langle f, (\mathcal{F}_g)'_\sigma\eta\rangle_{L^2(\partial\Omega)} = -\langle\nabla F_g(\sigma)\cdot\nabla F_f(\sigma), \eta\rangle_{L^2(\Omega)}.$$ □

As a remark to Lemma 2.14, we can think of (2.59) in terms of an adjoint operator for $(\mathcal{F}_g)'_\sigma$,

$$((\mathcal{F}_g)'_\sigma)^* f := -\nabla F_g(\sigma) \cdot \nabla F_f(\sigma),$$

and thereby we can formulate

$$\langle f, (\mathcal{F}_g)'_\sigma \eta \rangle_{L^2(\partial\Omega)} = \langle ((\mathcal{F}_g)'_\sigma)^* f, \eta \rangle_{L^2(\Omega)}.$$

# 3

# Solving the Inverse Problem

Having determined important properties of the forward problem, and in particular for $(\mathcal{F}_g)'_\sigma$, we continue by investigating the inverse EIT problem. The conductivity $\sigma$ is assumed to be on the form $\sigma = \sigma_0 + \delta\sigma$ where $\sigma_0$ is a known background conductivity such that $T\sigma = T\sigma_0$, i.e. $T\delta\sigma = 0$, and the inclusion $\delta\sigma$ is the unknown that we seek to determine.

We wish to solve for $\sigma$ from $\mathcal{F}_g(\sigma) \simeq \phi$ given the measured Dirichlet-data $\phi$ and known Neumann-data $g$. Remember that

$$\phi = \mathcal{F}_g(\sigma) + \epsilon,$$

where $\epsilon$ is a perturbation in $L^2(\partial\Omega)$. In order to solve this problem we formulate it as a minimization problem with objective function $\Psi_P$, which attempts to find an inclusion $\delta\sigma$ for which the residual $\phi - \mathcal{F}_g(\sigma_0 + \delta\sigma)$ is small and which also is subject to a penalty term, that introduces a bias to help the method find a good approximation to the correct conductivity.

Now let's define the regularization that is going to be applied in pursuit of determining an approximate solution to the inverse EIT problem. Here it should be reminded that the set $\mathcal{A}_0$ is defined in Definition 2.5, where $\sigma_0$ is some fixed background conductivity.

**Definition 3.1** *Let $\delta\sigma \in \mathcal{A}_0$ and $P : \mathcal{A}_0 \to [0, \infty)$, then $\Psi_P$ is defined as*

$$\Psi_P(\delta\sigma) := J(\sigma_0 + \delta\sigma) + P(\delta\sigma),$$

*where $P$ is called the* penalty term *and $J$ is the* discrepancy term *defined as*

$$J(\sigma) := \frac{1}{2}\|\mathcal{F}_g(\sigma) - \phi\|^2_{L^2(\partial\Omega)}.$$

The functional $\Psi_P$ is the objective function that is sought minimized on $\mathcal{A}_0$ in order to determine an approximation to the exact $\sigma = \sigma_0 + \delta\sigma$.

Note that $J$ corresponds to a least-squares term seeking to minimize the residual. However, naively minimizing the residual will yield a solution which is too affected by noise to be recognized as an approximation to the exact solution. Hence, solving the ill-posed EIT problem requires an alternate approach by including the penalty term. The easiest way to explain this more explicitly is in terms of linear ill-posed problems, say

$$Ax = \tilde{b} = b + \epsilon,$$

where $A$ is a matrix and $b = Ax^*$ is the exact data corresponding to an exact solution $x^*$, and therefore $\tilde{b}$ is the perturbed data. Clearly minimizing the residual corresponds to

$$x = A^{-1}\tilde{b} = A^{-1}b + A^{-1}\epsilon = x^* + A^{-1}\epsilon.$$

We can think of $A^{-1}\epsilon$ as the inverted noise, and if $A$ models an ill-posed problem then $A$ has a large condition number, meaning that $A^{-1}\epsilon$ can become very large and dominate the exact solution $x^*$ even for very small perturbations $\epsilon$.

For non-linear problems we can not split the terms into the exact solution and inverted noise, but the behaviour is similar in the regard that small perturbations will ruin the solution, unless we help the solver in finding an appropriate approximation. This is typically done by help of the penalty term $P$. Thereby we accept that we will not find the correct solution but via regularization we hope to find a good candidate for an approximate solution. Furthermore, due to the perturbation in $\phi$ we can actually not be sure that $\phi$ is even in the range of the forward map, which means that there possibly is no solution to the given data. This is the motivation for using an optimization approach, such that we attempt to find a solution $\sigma$ such that $\mathcal{F}_g(\sigma)$ is close to the Dirichlet-data $\phi$ when projected onto the range of the forward map.

In a typical Tikhonov regularization, one would choose $P(\delta\sigma) = \alpha\|\delta\sigma\|^2$ where $\|\cdot\|$ is an appropriate norm and $\alpha$ will be the regularization parameter. The role of the regularization parameter is to control the balance between minimizing the residual and introducing the bias from the penalty term. It is evident that a good approximation for $\sigma$ is highly dependent on the regularization parameter, and for some problems good candidates can be found using automated approaches [24]. In this thesis we will not use these automatic approaches since the prominent methods are based on linear problems, which EIT is not, and therefore we will determine appropriately good parameters by trial-and-error. However, in this thesis we deal with *toy problems*, i.e. problems where the data is simulated from the exact $\sigma$, so it is not hard to find a good parameter. Choosing the regularization parameter very low and then increasing it steadily will give a decrease in the error for the approximation, until some tipping point where the regularization bias begins to dominate the solution.

## 3.1 Derivative of the Discrepancy

The first step in minimizing $\Psi_P$ using a gradient type iterative algorithm, is to determine a derivative to the discrepancy term. For a definition of the Gâteaux derivative see Definition C.4 in Appendix C.

**Theorem 3.2** *Let $\sigma, \sigma + \eta \in \mathcal{A}$ then the Gâteaux derivative $J'_\sigma$ is given by*

$$J'_\sigma \eta = \langle \nabla J(\sigma), \eta \rangle_{L^2(\Omega)}, \tag{3.1}$$

*where $\nabla J(\sigma)$ is defined as*

$$\nabla J(\sigma) := -\nabla F_g(\sigma) \cdot \nabla F_{\mathcal{F}_g(\sigma)-\phi}(\sigma). \tag{3.2}$$

**Proof.** $J'_\sigma \eta$ is defined by

$$J'_\sigma \eta := \frac{d}{d\epsilon} J(\sigma + \epsilon\eta)|_{\epsilon=0}. \tag{3.3}$$

Thus by use of Corollary 2.13 and Theorem C.5, that $(\mathcal{F}_g)'_\sigma$ is linear, and the product rule for differentiation, then

$$\begin{aligned}
\frac{d}{d\epsilon} J(\sigma + \epsilon\eta) &= \frac{d}{d\epsilon}\frac{1}{2}\int_{\partial\Omega} |\mathcal{F}_g(\sigma + \epsilon\eta) - \phi|^2 ds \\
&= \int_{\partial\Omega} \left[\frac{d}{d\epsilon}(\mathcal{F}_g(\sigma + \epsilon\eta) - \phi)\right](\mathcal{F}_g(\sigma + \epsilon\eta) - \phi) ds \\
&= \int_{\partial\Omega} \left[\frac{d}{d\epsilon}(\mathcal{F}_g(\sigma) + \epsilon(\mathcal{F}_g)'_\sigma\eta + O(\epsilon^2) - \phi)\right](\mathcal{F}_g(\sigma + \epsilon\eta) - \phi) ds \\
&= \int_{\partial\Omega} \left[(\mathcal{F}_g)'_\sigma\eta + O(\epsilon)\right](\mathcal{F}_g(\sigma + \epsilon\eta) - \phi) ds.
\end{aligned}$$

So by (3.3) and Lemma 2.14 then

$$\begin{aligned} J'_\sigma\eta &= \frac{d}{d\epsilon}J(\sigma+\epsilon\eta)|_{\epsilon=0} \\ &= \int_{\partial\Omega}(\mathcal{F}_g)'_\sigma\eta(\mathcal{F}_g(\sigma)-\phi)ds \\ &= \langle\nabla J(\sigma),\eta\rangle_{L^2(\Omega)}. \end{aligned} \tag{3.4}$$

It is evident that (3.4) is what we would expect from the usual differentiation of a composition of functions regardless of how $\mathcal{F}_g$ is defined (as long as it is differentiable), while Lemma 2.14 leads to the less intuitive result that makes use of the definition of $\mathcal{F}_g$. □

The notation used in (3.1) is consistent with the notation in Neuberger [28], where $\nabla J(\sigma)$ denotes the function that via the metric (which in this case is for $L^2$) defines the derivative of the discrepancy term $J$. To relate this notation to the usual gradient in $\mathbb{R}^d$, note that the Gâteaux derivative corresponds to a so called directional derivative, which for $\mathbb{R}^d$ corresponds to $f' = \nabla f \cdot v$ where $\nabla f$ is the usual gradient and $v$ the direction. The directional derivative denotes the derivative along a curve given by the direction $v$. If we let $v$ be normalized such that $|v| = 1$, then it follows from the Cauchy-Schwartz inequality that $|f'| \leq |\nabla f||v| = |\nabla f|$ no matter the direction $v$. If $\nabla f \neq 0$, then this limit is only attained for $v = \pm\nabla f/|\nabla f|$, i.e. $\nabla f$ gives the direction for which there is the steepest ascend in $f$, and similarly $-\nabla f$ denotes the direction for the steepest descend, which for $\mathbb{R}^d$ is a well-known fact. Since the Cauchy-Schwartz inequality can be applied in the exact same way for $\langle\nabla J(\sigma),\eta\rangle_{L^2(\Omega)}$ then $-\nabla J(\sigma)$ denotes the direction for the steepest descend, not in a geometrical sense, but in the sense of the $\|\cdot\|_{L^2(\Omega)}$-metric. Similarly one can look at different metrics for which steepest ascend/descend directions will be different.

Here we will attempt to construct an iterative gradient method, similar to that of a steepest descend method, where we seek to find $\delta\sigma$ such that $\sigma = \sigma_0 + \delta\sigma$. Therefore it seems convenient to use a smoother function $\nabla_s J(\sigma) \in H^1_0(\Omega)$ to define $J'_\sigma$. It is convenient as $\nabla J(\sigma)$ may be non-zero on the boundary which is not very good considering that we wish $T\delta\sigma = 0$ such that $T\sigma = T\sigma_0$, and this may generate numerical problems when applying a steepest descend type algorithm like $\delta\sigma_{i+1} := \delta\sigma_i - s_i\nabla J(\sigma_i)$, where we iterative move in the descend directions using some step size $s_i$.

Now to further justify the use of the $H^1$-metric (on $H_0^1$) which leads to $\nabla_s J(\sigma)$ in Corollary 3.5 it is shown that we can always choose $\mathcal{A}$ such that $J'_\sigma \in (H_0^1(\Omega))' = H^{-1}(\Omega)$.

**Lemma 3.3** *Let $g \in L^s(\Omega)$ where $\Omega \subset \mathbb{R}^d$ is open and bounded and has $C^1$ boundary. Let $p$ and $q$ be conjugate exponents, then $f \mapsto \langle f, g\rangle_{L^2(\Omega)} \in (W^{1,p}(\Omega))'$ in either of the following cases:*

*(i)* $1 \leq p < d$ *and* $s \geq \frac{qd}{q+d}$*,*

*(ii)* $p \geq d$ *and* $s > 1$*.*

**Proof.** First let $1 \leq p < d$ and let $f \in W^{1,p}(\Omega)$ then $f \in L^r(\Omega)$ with $r := \frac{pd}{d-p}$ by Theorem B.9, and $r' := \frac{qd}{q+d}$ is the conjugate exponent to $r$ as

$$\frac{1}{r} + \frac{1}{r'} = \frac{d-p}{pd} + \frac{q+d}{qd} = \frac{qd+pd}{qpd} = \frac{1}{p} + \frac{1}{q} = 1.$$

Thus by Theorem B.11 then $g \in L^{r'}(\Omega)$ for $s \geq r' = \frac{qd}{q+d}$.

The other case is $p \geq d$ and let $f \in W^{1,p}(\Omega)$ then by Theorem B.9 and Theorem B.11 we have $f \in L^r(\Omega)$ for any $1 \leq r < \infty$, and let $r'$ be the conjugate exponent $r' \in (1, \infty]$. Then $s > 1$ will be $s = r'$ for some $r \in [1, \infty)$. Thus $g \in L^{r'}(\Omega)$ for some corresponding $r \in [1, \infty)$.

In both cases we have from Hölder's inequality and the imbedding $W^{1,p}(\Omega) \hookrightarrow L^r(\Omega)$:

$$|\langle f, g\rangle_{L^2(\Omega)}| \leq \|g\|_{L^{r'}(\Omega)} \|f\|_{L^r(\Omega)} \leq C \|g\|_{L^{r'}(\Omega)} \|f\|_{W^{1,p}(\Omega)},$$

thus $f \mapsto \langle f, g\rangle_{L^2(\Omega)}$ is a bounded (and clearly linear) functional on $W^{1,p}(\Omega)$, and therefore contained in $(W^{1,p}(\Omega))'$. □

**Theorem 3.4** *Let $\sigma \in \mathcal{A}$ and $\phi$ be exact Dirichlet-data. If $d = 2$ then $J'_\sigma \in H^{-1}(\Omega)$. If $d = 3$ the same holds if $\lambda$ is sufficiently close to 1 and $g \in L^s(\partial\Omega) \cap \tilde{H}^{-1/2}(\partial\Omega)$ for $s \geq \frac{8}{5}$.*

**Proof.** $F_g(\sigma) \in \tilde{H}^1(\Omega)$ and by Theorem B.4 then $\mathcal{F}_g(\sigma) \in \tilde{H}^{1/2}(\partial\Omega) \subset L^2(\partial\Omega)$, and as $\phi \in \tilde{H}^{1/2}(\partial\Omega)$ then $w := \mathcal{F}_g(\sigma) - \phi \in L^2(\partial\Omega) \cap \tilde{H}^{-1/2}(\partial\Omega)$. Thus using Theorem 2.7 then

$$\|F_w(\sigma)\|_{W^{1,q}(\Omega)} \leq C \|w\|_{L^2(\partial\Omega)}, \tag{3.5}$$

where $q \in (2, Q)$ and satisfies $(1 - \frac{1}{d})q \in [1, 2]$ (the reason for this interval is such that $s = 2$ in Theorem 2.7 satisfies $s \geq q - \frac{q}{d}$), i.e. $q \in (2, Q) \cap [\frac{d}{d-1}, \frac{2d}{d-1}]$. Thus let's use

$$d = 2: \quad q \in (2, \min\{Q, 4\}) \subseteq (2, Q) \cap [2, 4], \tag{3.6}$$

$$d = 3: \quad q \in (2, \min\{Q, 3\}) \subseteq (2, Q) \cap [\tfrac{3}{2}, 3]. \tag{3.7}$$

Now (3.5), (3.6), and (3.7) will be used show that $\nabla J(\sigma)$ belongs to appropriate $L^r$-spaces, in order to use Lemma 3.3. Note that by Lemma 3.3 then $J'_\sigma = \langle \nabla J(\sigma), \cdot \rangle_{L^2(\Omega)} \in (H^1(\Omega))' \subset (H^1_0(\Omega))' = H^{-1}(\Omega)$ if $\nabla J(\sigma) \in L^r(\Omega)$ with $r > 1$ for $d = 2$ and with $r \geq \frac{6}{5}$ for $d = 3$.

First let $d = 2$, since $F_g(\sigma) \in \tilde{H}^1(\Omega)$ then $\nabla F_g(\sigma) \in L^2(\Omega)$. It has already been established in (3.5) and (3.6) that $F_w(\sigma) \in W^{1,q}(\Omega)$ for $q \in (2, \min\{Q, 4\})$, so $\nabla F_w(\sigma) \in L^q(\Omega)$. By Theorem B.10 then

$$\nabla J(\sigma) = -\nabla F_g(\sigma) \cdot \nabla F_w(\sigma) \in L^r(\Omega), \quad \frac{1}{r} = \frac{1}{2} + \frac{1}{q},$$

and as $q > 2$ then $r > 1$, i.e. $J'_\sigma \in H^{-1}(\Omega)$.

Now let $d = 3$. Since we want $\nabla J(\sigma) \in L^{6/5}(\Omega)$, it suffices by Hölder's inequality to show that $\nabla F_g(\sigma), \nabla F_w(\sigma) \in L^{12/5}(\Omega)$. If

$$s \geq \left(1 - \frac{1}{d}\right) \frac{12}{5} = \frac{8}{5},$$

which is presumed in this theorem, then Theorem 2.7 implies $F_g(\sigma) \in W^{1,12/5}(\Omega)$ i.e. $\nabla F_g(\sigma) \in L^{12/5}(\Omega)$. $\nabla F_w(\sigma) \in L^{12/5}(\Omega)$ if $Q > \frac{12}{5}$ by (3.5) and (3.7), which is satisfied for $\lambda$ in $\mathcal{A}$ being sufficiently close to 1, as $\lim_{\lambda \to 1} Q(\lambda) = \infty$ in Theorem 2.7. So for $\lambda$ sufficiently close to 1 and $s \geq \frac{8}{5}$ then $J'_\sigma \in H^{-1}(\Omega)$. □

Now we may use a so called Sobolev gradient $\nabla_s J(\sigma)$ as it is named in Neuberger [28], which will act as a preconditioner for the usual $\nabla J(\sigma)$ in the steepest descent type algorithm by avoiding numerical errors on the boundary.

**Corollary 3.5 (Sobolev Gradient)** *The following PDE problem has a unique solution in $H^1_0(\Omega)$, and this solution will be denoted $\nabla_s J(\sigma)$,*

$$-\Delta u + u = \nabla J(\sigma), \ \textit{in } \Omega, \quad u = 0, \ \textit{on } \partial\Omega. \tag{3.8}$$

*Then* $\nabla_s J(\sigma) = \iota^* \nabla J(\sigma)$ *with* $\iota$ *being the inclusion operator of* $H_0^1(\Omega)$ *into* $L^2(\Omega)$ *(bounded in the* $H^1$*-metric), and thus*

$$J'_\sigma \eta = \langle \nabla J(\sigma), \iota\eta \rangle_{L^2(\Omega)} = \langle \nabla_s J(\sigma), \eta \rangle_{H^1(\Omega)}, \quad \eta \in H_0^1(\Omega). \tag{3.9}$$

**Proof.** The proof follows directly by writing up the weak formulation of (3.8) and applying Riesz' representation theorem [19]. Let $\eta \in H_0^1(\Omega)$ multiply both sides of (3.8) and integrate over $\Omega$, then by Theorem C.1

$$\begin{aligned}\langle \nabla J(\sigma), \eta \rangle_{L^2(\Omega)} &= -\int_\Omega \Delta u \eta dx + \int_\Omega u \eta dx \\ &= \int_\Omega \nabla u \cdot \nabla \eta dx - \int_{\partial\Omega} T\eta \frac{\partial u}{\partial n} ds + \int_\Omega u \eta dx \\ &= \langle u, \eta \rangle_{H^1(\Omega)} - \int_{\partial\Omega} T\eta \frac{\partial u}{\partial n} ds, \\ &= \langle u, \eta \rangle_{H^1(\Omega)}, \end{aligned} \tag{3.10}$$

where $\int_{\partial\Omega} T\eta \frac{\partial u}{\partial n} ds$ vanishes because $T\eta = 0$. As $J'_\sigma \in (H_0^1(\Omega))'$ then the expression (3.10) makes sense.

Since $J'_\sigma \eta = \langle \nabla J(\sigma), \eta \rangle_{L^2(\Omega)} = \langle \nabla J(\sigma), \iota\eta \rangle_{L^2(\Omega)}$ where the embedding $\iota : H_0^1(\Omega) \to L^2(\Omega)$ is a bounded linear operator, then it follows from Riesz' representation theorem [19] that there is a unique adjoint $\iota^* : L^2(\Omega) \to H_0^1(\Omega)$ which is a linear bounded operator satisfying

$$\langle \nabla J(\sigma), \iota\eta \rangle_{L^2(\Omega)} = \langle \iota^* \nabla J(\sigma), \eta \rangle_{H^1(\Omega)},$$

where the unique element $\iota^* \nabla J(\sigma) \in H_0^1(\Omega)$ is denoted $\nabla_s J(\sigma)$. Due to uniqueness $\nabla_s J(\sigma)$ must be the solution in (3.10). □

For the iteration step we need to determine a step size $s_i$, for an algorithm resembling $\delta\sigma_{i+1} = \delta\sigma_i - s_i \nabla_s J(\sigma_i)$. Here we will make use of the popular Barzilai-Borwein (BB) step size rule [1, 9, 13], for which we try to find $s_i$ such that $\frac{1}{s_i}(\delta\sigma_i - \delta\sigma_{i-1}) = \frac{1}{s_i}(\sigma_i - \sigma_{i-1}) \simeq \nabla_s J(\sigma_i) - \nabla_s J(\sigma_{i-1})$. It may not be possible to get equality so it is found in the least-squares sense

$$s_i := \operatorname*{argmin}_s \zeta(s) = \| s^{-1}(\delta\sigma_i - \delta\sigma_{i-1}) - (\nabla_s J(\sigma_i) - \nabla_s J(\sigma_{i-1})) \|^2_{H^1(\Omega)}. \tag{3.11}$$

Step sizes of the form (3.11) has proven to give fast convergence in gradient methods [1].

**Corollary 3.6 (BB Step Size)**
*The solution to* (3.11) *if* $\langle \delta\sigma_i - \delta\sigma_{i-1}, \nabla_s J(\sigma_i) - \nabla J(\sigma_{i-1})\rangle_{H^1(\Omega)} \neq 0$ *is*

$$s_i = \frac{\|\delta\sigma_i - \delta\sigma_{i-1}\|^2_{H^1(\Omega)}}{\langle \delta\sigma_i - \delta\sigma_{i-1}, \nabla_s J(\sigma_i) - \nabla_s J(\sigma_{i-1})\rangle_{H^1(\Omega)}}. \tag{3.12}$$

*Otherwise* $s_i = \infty$.

**Proof.** For simplicity let $\Delta x := \delta\sigma_i - \delta\sigma_{i-1}$ and $\Delta j := \nabla_s J(\sigma_i) - \nabla_s J(\sigma_{i-1})$, and let's assume that $\langle \Delta x, \Delta j\rangle \neq 0$, then

$$\zeta(s) = \|s^{-1}\Delta x - \Delta j\|^2 = s^{-2}\|\Delta x\|^2 + \|\Delta j\|^2 - 2s^{-1}\langle \Delta x, \Delta j\rangle. \tag{3.13}$$

Now differentiating in $s$ yields

$$\zeta'(s) = -2s^{-3}\|\Delta x\|^2 + 2s^{-2}\langle \Delta x, \Delta j\rangle, \tag{3.14}$$

$$\zeta''(s) = 6s^{-4}\|\Delta x\|^2 - 4s^{-3}\langle \Delta x, \Delta j\rangle. \tag{3.15}$$

Solving (3.14) equal zero yields

$$\begin{aligned} 0 &= -2s^{-3}\|\Delta x\|^2 + 2s^{-2}\langle \Delta x, \Delta j\rangle, \\ 0 &= -\|\Delta x\|^2 + s\langle \Delta x, \Delta j\rangle, \\ s &= \frac{\|\Delta x\|^2}{\langle \Delta x, \Delta j\rangle}. \end{aligned} \tag{3.16}$$

Note that $s_i$ is the solution in (3.16). To make sure that $s_i$ is a minimum it is sufficient to check that $\zeta''(s_i) > 0$ since $\zeta$ is continuous at $s_i$ except if $\Delta x = 0$ (which effectively eliminates the optimization problem, and is not relevant), thus

$$\begin{aligned} \zeta''(s_i) &= 6s_i^{-4}\|\Delta x\|^2 - 4s_i^{-3}\langle \Delta x, \Delta j\rangle \\ &= s_i^{-3}(6s_i^{-1}\|\Delta x\|^2 - 4\langle \Delta x, \Delta j\rangle) \\ &= \frac{\langle \Delta x, \Delta j\rangle^3}{\|\Delta x\|^6}(2\langle \Delta x, \Delta j\rangle) \\ &= \frac{2\langle \Delta x, \Delta j\rangle^4}{\|\Delta x\|^6} > 0. \end{aligned}$$

It was initially assumed that $\langle \Delta x, \Delta j\rangle \neq 0$, however, if $\langle \Delta x, \Delta j\rangle = 0$ then it is easy from (3.13) to see that $s = \infty$ is the solution. □

From Corollary 3.6 it is evident that it is important to have a maximum step size, in the case that $\delta\sigma_i - \delta\sigma_{i-1} \perp \nabla_s J(\sigma_i) - \nabla_s J(\sigma_{i-1})$ or almost orthogonal.

Note that Barzilai and Borwein [1] has a small typo as it has $s$ and not $s^{-1}$ in (3.11), however, they arrive at the same result as in Corollary 3.6, which would be incorrect when using $s$ and not $s^{-1}$. It is also evident from the description in Barzilai and Borwein [1] that $s^{-1}$ should be used, since the first type of step size they discuss relates to $\|\Delta x - s\Delta j\|^2$ i.e. that $\Delta x \simeq s\Delta j$, and they go on to say that equivalently one could look at $\|s\Delta x - \Delta j\|^2$, however, this would result in $s\Delta x \simeq \Delta j$, thus it should have been $s^{-1}$.

Note that also Jin *et al.* [9] seems to assume that $s$ and not $s^{-1}$ should be used, which results in what corresponds to $s_i^{-1}$ in Corollary 3.6, and leads to a far too small step size. They also have an error later on (when using weak monotonicity, see Definition 3.7) with reference to Wright *et al.* [13], however, it may be because Wright *et al.* [13] uses a step size of the form $a = \langle \Delta x, \Delta j\rangle / \|\Delta x\|^2$, though the actual step is applied with $1/a$ corresponding with Barzilai and Borwein [1] and Corollary 3.6.

With inspiration from Wright *et al.* [13], then $s_i$ will be initialized by (3.12), after which it is thresholded to lie in $[s_{\min}, s_{\max}]$, where $0 < s_{\min} < s_{\max} < \infty$ are user defined constants. It is noted in Wright *et al.* [13] that BB type step rules converge faster if we do not restrict $\Psi_P$ to decrease in every iteration, and that an occasional increase actually improves the convergence rate. An occasional increase in $\Psi_P$ can be used to avoid places where the method has to take many small steps to ensure the decrease of $\Psi_P$. Therefore one makes sure that the following so called weak monotonicity is satisfied, which compares $\Psi_P(\delta\sigma_{i+1})$ with the most recent $M$ steps.

**Definition 3.7 (Weak Monotonicity)** *Let $\tau \in (0,1)$ and $M \in \mathbb{N}$, then $s_i$ is said to satisfy the weak monotonicity if the following is satisfied:*

$$\Psi_P(\delta\sigma_{i+1}) \leq \max_{i-M+1\leq j\leq i} \Psi_P(\delta\sigma_j) - \frac{\tau}{2s_i}\|\delta\sigma_{i+1} - \delta\sigma_i\|^2_{H^1(\Omega)}. \tag{3.17}$$

As seen for the example in Section 2.1 on page 14 it is advantageous to be able to use multiple datasets $\{(g_k, \phi_k)\}_{k=1}^K$, in order to improve the quality of the reconstruction, and this can simply be done by slightly altering the discrepancy term in Definition 3.1 to

$$J(\sigma) := \sum_{k=1}^{K} w_k J_k(\sigma) = \sum_{k=1}^{K} \frac{w_k}{2} \|\mathcal{F}_{g_k}(\sigma) - \phi_k\|^2_{L^2(\partial\Omega)}, \tag{3.18}$$

i.e. where $J_k$ denotes the usual discrepancy term for the data $(g_k, \phi_k)$. The weights $w_k$ can be used to prioritize the importance of the datasets, i.e. for instance if some datasets have higher importance or are measured more precisely they can be prioritized higher, however, I will just use the weights $w_k = 1, \ \forall k$. It follows straightforwardly that

$$J'_\sigma \eta = \langle \nabla J(\sigma), \eta \rangle_{L^2(\Omega)} = \sum_{k=1}^{K} w_k \langle \nabla J_k(\sigma), \eta \rangle_{L^2(\Omega)}, \tag{3.19}$$

where $\nabla J_k(\sigma) := -\nabla F_{g_k}(\sigma) \cdot \nabla F_{\mathcal{F}_{g_k}(\sigma) - \phi_k}(\sigma)$, i.e. the expression in Theorem 3.2 for the data $(g_k, \phi_k)$. It should be noted that the definition of the Sobolev gradient $\nabla_s J(\sigma)$ is still given by Theorem 3.5, but where $\nabla J(\sigma)$ is the expression given in (3.19). So the theory considered for a single dataset is almost identical to that of multiple datasets.

It is also useful to consider partial/incomplete data. Let $\Gamma \subset \partial\Omega$, and assume that $\operatorname{supp} g_k \subseteq \Gamma$, then we can not be sure that $\operatorname{supp} \phi_k \subseteq \Gamma$ even with no perturbations, as seen in Figure 4.46 on page 107. However, suppose that we are only able to measure $\phi_k$ at $\Gamma$ such that we in reality get $\chi_\Gamma \phi_k$ where $\chi_\Gamma$ is a characteristic function that equals 1 on $\Gamma$ and zero otherwise. Then we can further change the discrepancy term, such that we only seek to minimize the residual on $\Gamma$:

$$J_k(\sigma) := \frac{1}{2} \|\mathcal{F}_{g_k}(\sigma) - \phi_k\|^2_{L^2(\Gamma)} = \frac{1}{2} \|\chi_\Gamma(\mathcal{F}_{g_k}(\sigma) - \phi_k)\|_{L^2(\partial\Omega)}. \tag{3.20}$$

Similarly we get

$$\nabla J_k(\sigma) := -\nabla F_{g_k}(\sigma) \cdot \nabla F_{\chi_\Gamma(\mathcal{F}_{g_k}(\sigma) - \phi_k)}(\sigma), \tag{3.21}$$

i.e. the change to include partial data is very subtle, and we note that if $\Gamma = \partial\Omega$ we obviously get the usual results for complete data.

## 3.2 Sparsity Regularization

The main focus of the thesis is on sparsity regularization, so firstly we must define what is meant by sparsity regularization, which will be a specific penalty term for Definition 3.1.

**Definition 3.8** *Let $\delta\sigma \in \mathcal{A}_0$, and let $\{\psi_k\}_{k\in\mathbb{N}}$ be a basis for $H_0^1(\Omega)$, then the functional $\Psi_S$ is defined as*

$$\Psi_S(\delta\sigma) = J(\sigma_0 + \delta\sigma) + R_{\boldsymbol{\alpha},r}(\delta\sigma),$$

*where for $\delta\sigma = \sum_{k\in\mathbb{N}} c_k\psi_k$, and $\boldsymbol{\alpha} = \{\alpha_k\}_{k\in\mathbb{N}} \subset (0,1]$,*

$$R_{\boldsymbol{\alpha},r}(\eta) := \frac{1}{r}\sum_{k\in\mathbb{N}} \alpha_k|c_k|^r, \quad r \in [1,2].$$

In order to achieve sparsity with respect to the chosen basis $\{\psi_k\}$, i.e. that $\delta\sigma$ can be expanded using only a small number of the basis functions, then $r \in [1,2)$ is usually chosen in order to penalize $\delta\sigma$ with many small coefficients, i.e. $|c_k| < 1$ and rather accept $\delta\sigma$ with few larger coefficients $|c_k| > 1$. In this thesis we only look at the case where $r = 1$, thus we have

$$\Psi_S(\delta\sigma) = J(\sigma) + R_{\boldsymbol{\alpha},1}(\delta\sigma) = \frac{1}{2}\|\mathcal{F}_g(\sigma) - \phi\|^2_{L^2(\partial\Omega)} + \sum_{k\in\mathbb{N}} \alpha_k|c_k|.$$

Now we may be concerned for whether $R_{\boldsymbol{\alpha},1}(\delta\sigma)$ is convergent, for which we have to assume that $\delta\sigma$ is sparse with respect to the chosen basis $\{\psi_k\}$.

**Definition 3.9 (Sparsity w.r.t. Basis)** *Let $\{\psi_k\}$ be a basis for a Banach Space $V$, then we say that $f \in V$ is sparse w.r.t. $\{\psi_k\}$ if $\sum_k |c_k| < \infty$ where $f = \sum_k c_k\psi_k$.*

Note that the role of Definition 3.9 is simply to allow the use of the term $R_{\boldsymbol{\alpha},1}(\delta\sigma)$. Therefore we introduce a so called sparsity constraint, namely, because $R_{\boldsymbol{\alpha},1}(\delta\sigma)$ is not guaranteed to converge in general and the solution is restricted to a certain subset of $\mathcal{A}$, and ideally in terms of sparsity we can find a good approximate solution that only has finitely many non-zero expansion coefficients.

In terms of sparsity, we do not specifically seek the usual $\ell_1$-term given in Definition 3.9, but rather $\sum_k \chi_{\mathbb{R}\setminus\{0\}}(c_k)$ where $\chi_{\mathbb{R}\setminus\{0\}}$ is a characteristic function on the set $\mathbb{R} \setminus \{0\}$, i.e. that sparsity is simply a count of non-zero expansion coefficients. This would lead to the penalty term $\sum_k \alpha_k \chi_{\mathbb{R}\setminus\{0\}}(c_k)$ instead of that in Definition 3.8. The reason that we tend to a sort of pseudo sparsity instead is due to $\sum_k \alpha_k \chi_{\mathbb{R}\setminus\{0\}}(c_k)$ not being convex, which is an essential property for finding a unique minimizer with iterative gradient methods. The penalty term $R_{\boldsymbol{\alpha},1}$ allows solutions where all the expansion coefficients are small but non-zero. However, if the true solution only has a small number of non-zero expansion coefficients, then the discrepancy term along with the penalty term $R_{\boldsymbol{\alpha},1}$ attempt to construct solutions where the non-zero coefficients in the true solution are dominant. A penalty of $R_{\boldsymbol{\alpha},2}$ in the usual Tikhonov regularization is more likely to include coefficients that are not supposed to be non-zero, simply due to the dampening by $|c_k|^2$ for $|c_k| < 1$, which can give a negligible penalty.

Furthermore, it should be noted that there are multiple regularization parameters, namely, one per basis function. In Jin *et al.* [9] they used a single parameter $\alpha_k = \alpha, \ \forall k$ which is quite common in regularization techniques. However, having one parameter per basis function gives a special type of flexibility with respect to applying prior knowledge about the sparsity of the solution, which we will return to in the numerical experiments presented in Chapter 4 on page 57.

Now a new non-linear operator will be introduced, which is known as soft shrinkage/thresholding operator.

**Definition 3.10 (Soft Shrinkage/Thresholding Operator)**
*The soft shrinkage/thresholding operator $\mathcal{S}_\beta : \mathbb{R} \to \mathbb{R}$ is for $\beta \geq 0$ defined as*

$$\mathcal{S}_\beta(x) := \operatorname{sign}(x) \max\{|x| - \beta, 0\}, \quad x \in \mathbb{R}. \tag{3.22}$$

*Suppose that $f \in V$ for some separable Banach space $V$ with basis $\{\psi_k\}$, for which $f = \sum_k c_k \psi_k$. Furthermore, let $\boldsymbol{\beta} := \{\beta_k\} \subset [0, \infty)$, then define $\boldsymbol{\mathcal{S}}_{\boldsymbol{\beta}} : V \to V$ as*

$$\boldsymbol{\mathcal{S}}_{\boldsymbol{\beta}}(f) := \sum_k \mathcal{S}_{\beta_k}(c_k)\psi_k. \tag{3.23}$$

The reason for the name *soft* thresholding, is due to $\mathcal{S}_\beta$ being continuous, unlike the usual *hard* thresholding for which everything with absolute value below a certain constant $\beta$ is set to zero. The difference between soft and hard thresholding can be seen in Figure 3.1 on the facing page.

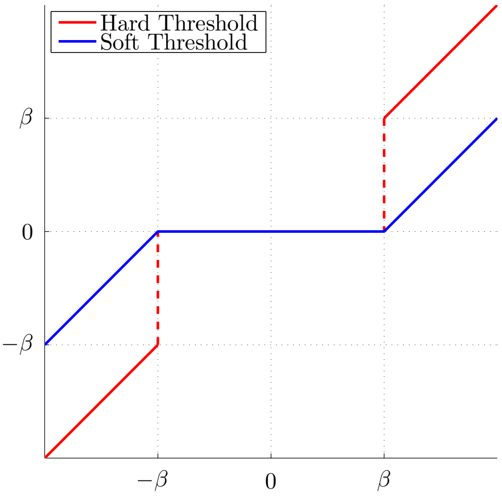


**Figure 3.1:** Soft and hard thresholding functions on the real line.

The operator $\mathcal{S}_\beta$ is non-linear, however, there is the following useful property.

**Lemma 3.11** *Let $c > 0$ then*

$$\mathcal{S}_\beta(cx) = c\mathcal{S}_{\beta/c}(x), \quad x \in \mathbb{R}. \tag{3.24}$$

**Proof.** Here we will go through the three different cases for $x \in \mathbb{R}$.

$c|x| < \beta$: Since $c > 0$ then we have that $|cx| < \beta$, thus $\mathcal{S}_\beta(cx) = 0$. However, we also have that $|x| < \beta/c$, i.e. $\mathcal{S}_{\beta/c}(x) = 0$, hence (3.24) holds.

$cx > \beta$: Since $c > 0$ and $\beta \geq 0$ then $x > 0$, i.e. we have $\mathcal{S}_\beta(cx) = cx - \beta$. Similarly we have that $x > \beta/c$ thus $c\mathcal{S}_{\beta/c}(x) = c(x - \beta/c) = cx - \beta$, and again (3.24) holds.

$cx < -\beta$: In this case $x < 0$ and such that $|cx| > \beta$, i.e. we have $\mathcal{S}_\beta(cx) = -(|cx| - \beta) = cx + \beta$. Similarly $x < -\beta/c$ i.e. $|x| > \beta/c$ so $c\mathcal{S}_{\beta/c}(x) = -c(|x| - \beta/c) = cx + \beta$. □

The non-linearity of $\mathcal{F}_g$ makes it hard to determine a descent direction for $\Psi_S$. Instead we can linearise the discrepancy term $J$ in each iteration, and use an approximation to $\Psi_S$ to propose a gradient step. In order to linearise $J$ it is already now considered that a Barzilai-Borwein (BB) step size rule will be applied, for which we approximate the Hessian $J''_{\sigma_i} \simeq J'_\sigma - J'_{\sigma_i}$ with

$\nabla_s J(\sigma) - \nabla_s J(\sigma_i) \simeq \frac{1}{s_i}(\delta\sigma - \delta\sigma_i)$ where $s_i$ is the step length and $\sigma_i = \sigma_0 + \delta\sigma_i$ and $\delta\sigma_i$ corresponds to the $i$'th iteration in the coming scheme. Thus by Theorem C.5 we have

$$\begin{aligned} J(\sigma) &\simeq J(\sigma_i) + J'_{\sigma_i}(\sigma - \sigma_i) + \frac{1}{2}(J'_\sigma - J'_{\sigma_i})(\sigma - \sigma_i) \\ &= J(\sigma_i) + \langle \nabla_s J(\sigma_i), \delta\sigma - \delta\sigma_i \rangle_{H^1(\Omega)} \\ &\quad + \frac{1}{2}\langle \nabla_s J(\sigma) - \nabla_s J(\sigma_i), \delta\sigma - \delta\sigma_i \rangle_{H^1(\Omega)} \\ &\simeq J(\sigma_i) + \langle \nabla_s J(\sigma_i), \delta\sigma - \delta\sigma_i \rangle_{H^1(\Omega)} + \frac{1}{2s_i}\|\delta\sigma - \delta\sigma_i\|^2_{H^1(\Omega)}. \end{aligned} \tag{3.25}$$

Now given a step length $s_i$ and previous iterate $\delta\sigma_i$, we will determine $\delta\sigma_{i+1}$ by minimizing an approximation of $\Psi_S(\delta\sigma) - \Psi_S(\delta\sigma_i)$ in terms of $\delta\sigma$, thus $\Psi_S$ decreases when $\Psi_S(\delta\sigma_{i+1}) - \Psi_S(\delta\sigma_i) < 0$. Hence from (3.25) we get

$$\begin{aligned} \Psi_S(\delta\sigma) - \Psi_S(\delta\sigma_i) &= J(\sigma) - J(\sigma_i) + R_{\boldsymbol{\alpha},1}(\delta\sigma) - R_{\boldsymbol{\alpha},1}(\delta\sigma_i) \\ &\simeq \langle \nabla_s J(\sigma_i), \delta\sigma - \delta\sigma_i \rangle_{H^1(\Omega)} + \frac{1}{2s_i}\|\delta\sigma - \delta\sigma_i\|^2_{H^1(\Omega)} \\ &\quad + R_{\boldsymbol{\alpha},1}(\delta\sigma) - R_{\boldsymbol{\alpha},1}(\delta\sigma_i). \end{aligned} \tag{3.26}$$

This can be simplified in the following manner.

**Theorem 3.12** *Consider the following optimization problem,*

$$\min_{\delta\sigma \in \mathcal{A}_0} \Upsilon(\delta\sigma) = \frac{1}{2}\|\delta\sigma - (\delta\sigma_i - s_i \nabla_s J(\sigma_i))\|^2_{H^1(\Omega)} + s_i R_{\boldsymbol{\alpha},1}(\delta\sigma). \tag{3.27}$$

*Let $\{\psi_k\}$ be the basis of $H^1_0(\Omega)$ from Definition 3.8, and let $\{v_k\}$ be the expansion coefficients such that $\delta\sigma_i - s_i \nabla_s J(\sigma_i) = \sum_{k\in\mathbb{N}} v_k \psi_k$. Then the following holds:*

*(i) (3.27) is equivalent with minimizing (3.26) w.r.t. $\delta\sigma \in \mathcal{A}_0$.*

*(ii) If $\{\psi_k\}$ are orthogonal in the $H^1$-metric (note they need not be normalized), then $\Upsilon$ is convex on $\mathcal{A}_0$. The solution to (3.27) is:*

$$\delta\sigma_{i+1} = \boldsymbol{\mathcal{S}}_{\boldsymbol{\beta}_i}(\delta\sigma_i - s_i \nabla_s J(\sigma_i)) = \sum_{k\in\mathbb{N}} \mathcal{S}_{\beta_k}(v_k)\psi_k, \tag{3.28}$$

*where $\beta_k := \frac{s_i \alpha_k}{\|\psi_k\|^2_{H^1(\Omega)}}$.*

**Proof.** The proof consists of three steps, one showing (3.27) is equivalent with (3.26), one showing that (3.27) is convex, and finally one that solves (3.27).

**Equivalence of (3.26) and (3.27):** Rewriting $f_1(\delta\sigma) := \frac{1}{2s_i}\|\delta\sigma - (\delta\sigma_i - s_i\nabla_s J(\sigma_i))\|^2_{H^1(\Omega)}$ yields:

$$\begin{aligned} f_1(\delta\sigma) &= \frac{1}{2s_i}\|(\delta\sigma - \delta\sigma_i) + s_i\nabla_s J(\sigma_i)\|^2_{H^1(\Omega)} \\ &= \frac{1}{2s_i}\|\delta\sigma - \delta\sigma_i\|^2_{H^1(\Omega)} + \langle \nabla_s J(\sigma_i), \delta\sigma - \delta\sigma_i\rangle_{H^1(\Omega)} + \frac{s_i}{2}\|\nabla_s J(\sigma_i)\|^2_{H^1(\Omega)}. \end{aligned} \tag{3.29}$$

An optimization problem is equivalent if the objective function is multiplied with a positive constant or if a constant is added. So by (3.29) then (3.27) must be equivalent with the minimization problem with the objective function being:

$$\begin{aligned} \tilde{\Upsilon}(\delta\sigma) &:= \frac{1}{s_i}\Upsilon(\delta\sigma) - \frac{s_i}{2}\|\nabla_s J(\sigma_i)\|^2_{H^1(\Omega)} - R_{\boldsymbol{\alpha},1}(\delta\sigma_i) \\ &= \langle \nabla_s J(\sigma_i), \delta\sigma - \delta\sigma_i\rangle_{H^1(\Omega)} + \frac{1}{2s_i}\|\delta\sigma - \delta\sigma_i\|^2_{H^1(\Omega)} + R_{\boldsymbol{\alpha},1}(\delta\sigma) \\ &\quad - R_{\boldsymbol{\alpha},1}(\delta\sigma_i), \end{aligned}$$

this is precisely the expression in (3.26).

**$\Upsilon$ is convex on $\mathcal{A}_0$:** Because $\{\psi_k\}$ are orthogonal then

$$R_{\boldsymbol{\alpha},1}(x) = \sum_{k\in\mathbb{N}} \frac{\alpha_k}{\|\psi_k\|^2_{H^1(\Omega)}}|\langle x, \psi_k\rangle_{H^1(\Omega)}|,$$

as $\{\psi_k/\|\psi_k\|_{H^1(\Omega)}\}$ is an orthonormal basis of $H^1_0(\Omega)$. The fact that $\{\psi_k\}$ is a basis means that we do not have to worry about convergence issues for the infinite sum when checking for convexity. As a linear combination of convex functions is convex, it suffices to check that each of the summed terms are convex. So let

$$\begin{aligned} f_1(x) &:= \frac{1}{2}\|x - y\|^2_{H^1(\Omega)}, \\ f_2(x) &:= |\langle x, y\rangle_{H^1(\Omega)}|, \end{aligned}$$

then the question of whether $\Upsilon$ is convex boils down to whether $f_1$ and $f_2$ are convex for some $y \in H^1_0(\Omega)$. Let $x_1, x_2 \in \mathcal{A}_0$ and $t \in [0,1]$, then $x := (1-t)x_1 + tx_2 \in \mathcal{A}_0$ as $\mathcal{A}_0$ is convex by Corollary 2.6. By the triangle inequality it follows that

$$\begin{aligned} f_2((1-t)x_1 + tx_2) &= |\langle (1-t)x_1 + tx_2, y\rangle| \\ &\le (1-t)|\langle x_1, y\rangle| + t|\langle x_2, y\rangle| \\ &= (1-t)f_2(x_1) + tf_2(x_2), \end{aligned}$$

thus $f_2$ is convex. Now checking $f_1$:

$$\begin{aligned} f_1((1-t)x_1+tx_2) &= \frac{1}{2}\|(1-t)x_1+tx_2-y\|^2 \\ &= \frac{1}{2}\|(1-t)(x_1-y)+t(x_2-y)\|^2 \\ &= \frac{1}{2}(1-t)^2\|x_1-y\|^2+\frac{1}{2}t^2\|x_2-y\|^2 \\ &\quad +(1-t)t\langle x_1-y,x_2-y\rangle \\ &= (1-t)^2 f_1(x_1)+t^2 f_1(x_2)+(1-t)t\langle x_1-y,x_2-y\rangle. \end{aligned} \tag{3.30}$$

Note that $(1-t)f_1-(1-t)^2f_1=(1-t)tf_1$, thus we have

$$(1-t)f_1(x_1)+tf_1(x_2)-(1-t)^2f_1(x_1)-t^2f_1(x_2)=(1-t)t(f_1(x_1)+f_1(x_2)). \tag{3.31}$$

Now consider $0\leq(a-b)^2=a^2+b^2-2ab$ thus $ab\leq\frac{1}{2}(a^2+b^2)$, this simple inequality along with Cauchy-Schwartz' inequality will now be applied to the following where it is noted that $(1-t)t\geq 0$ as $t\in[0,1]$:

$$\begin{aligned} (1-t)t\langle x_1-y,x_2-y\rangle &\leq (1-t)t\|x_1-y\|\|x_2-y\| \\ &\leq \frac{1}{2}(1-t)t(\|x_1-y\|^2+\|x_2-y\|^2) \\ &= (1-t)t(f_1(x_1)+f_1(x_2)). \end{aligned} \tag{3.32}$$

So from (3.32) and (3.31) then

$$\begin{aligned} &(1-t)^2f_1(x_1)+t^2f_1(x_2)+(1-t)t\langle x_1-y,x_2-y\rangle \\ &\qquad\leq (1-t)^2f_1(x_1)+t^2f_1(x_2)+(1-t)t(f_1(x_1)+f_1(x_2)) \\ &\qquad= (1-t)f_1(x_1)+tf_1(x_2). \end{aligned} \tag{3.33}$$

Now inserting (3.33) into (3.30) yields

$$f_1((1-t)x_1+tx_2)\leq(1-t)f_1(x_1)+tf_1(x_2),$$

thus $f_1$ and thereby also $\Upsilon$ is convex on $\mathcal{A}_0$.

**Solving (3.27) with $\{\psi_k\}$ being an orthogonal basis:** If $\{\psi_k\}$ is an orthogonal basis for $H_0^1(\Omega)$ in the $H^1$-metric, then normalizing $p_k := \frac{\psi_k}{\|\psi_k\|_{H^1(\Omega)}}$, $k\in\mathbb{N}$, yields an orthonormal basis. By Corollary 2.6 then $\mathcal{A}_0$ is closed and convex, and $\Upsilon$ is convex on $\mathcal{A}_0$, thus (3.27) is a convex optimization problem which means that the problem boils down to finding a stationary point.

In order to simplify the notation in the proof let $f := \delta\sigma$, $h := \delta\sigma_i - s_i\nabla_s J(\sigma_i)$, and let $f_k := \langle f, p_k\rangle$ and $h_k := \langle h, p_k\rangle$ (i.e. the expansion coefficients in the $\{p_k\}$-basis [19]). If $\{c_k\}$ are the expansion coefficients of $f$ in the $\{\psi_k\}$-basis, then clearly $c_k = f_k/\|\psi_k\|$. Furthermore, let $\gamma_k := \beta_k\|\psi_k\|$ so we have

$$s_i R_{\boldsymbol{\alpha},1}(f) = \sum_k s_i\alpha_k|c_k| = \sum_k \frac{s_i\alpha_k}{\|\psi_k\|}|f_k| = \sum_k \gamma_k|f_k|.$$

Thus $\Upsilon$ becomes

$$\Upsilon(f) = \frac{1}{2}\|f-h\|^2 + \sum_k \gamma_k|f_k|. \tag{3.34}$$

Note that $\langle f,h\rangle = \sum_k\sum_{\tilde{k}}\langle f_k p_k, h_{\tilde{k}} p_{\tilde{k}}\rangle = \sum_k f_k h_k$ as $\{p_k\}$ is orthonormal. So using the fundamental result for orthonormal bases that $\|f\|^2 = \sum_k |f_k|^2$ [19] then

$$\begin{aligned}\frac{1}{2}\|f-h\|^2 &= \frac{1}{2}\|f\|^2 + \frac{1}{2}\|h\|^2 - \langle f,h\rangle \\ &= \sum_k\left[\frac{1}{2}(f_k^2+h_k^2) - f_k h_k\right].\end{aligned}$$

Insertion into (3.34) gives

$$\Upsilon(f) = \sum_k\left[\frac{1}{2}(f_k^2+h_k^2) + \gamma_k|f_k| - f_k h_k\right] = \sum_k \Upsilon_k(f_k). \tag{3.35}$$

From (3.35) it is evident that minimizing $\Upsilon$ w.r.t. $f$ corresponds to minimizing $\Upsilon_k$ w.r.t. $f_k$ for each $k$. Since the term $|f_k|$ is not differentiable at 0, it is initially assumed that $f_k \neq 0$, so

$$\Upsilon_k'(f_k) = f_k + \gamma_k\,\mathrm{sign}(f_k) - h_k, \quad f_k \neq 0. \tag{3.36}$$

Let $f_k > 0$, solving (3.36) equal zero yields $f_k = h_k - \gamma_k$. Thus a condition for $f_k$ to be stationary for $f_k > 0$ is that $h_k > \gamma_k$. Similarly for $f_k < 0$ we get $f_k = h_k + \gamma_k$ thus we need $h_k < -\gamma_k$ for $f_k < 0$ being stationary. What remains is to consider $|h_k| \leq \gamma_k$, then $f_k$ is not stationary for $f_k > 0$ or $f_k < 0$. Thus we must set $f_k = 0$ as the optimal value is not at a stationary point, and $\pm\infty$ is not a valid option as seen from (3.35). Therefore we can conclude that

$$f_k = \begin{cases} h_k - \gamma_k, & h_k > \gamma_k, \\ h_k + \gamma_k, & h_k < -\gamma_k \\ 0, & |h_k| \leq \gamma_k \end{cases} = \mathcal{S}_{\gamma_k}(h_k). \tag{3.37}$$

Now inserting that $\|\psi_k\|c_k = f_k$, $\|\psi_k\|v_k = h_k$ and $\|\psi_k\|\beta_k = \gamma_k$, and applying Lemma 3.11,

$$c_k = \frac{f_k}{\|\psi_k\|} = \frac{1}{\|\psi_k\|}\mathcal{S}_{\beta_k\|\psi_k\|}(\|\psi_k\|v_k) = \mathcal{S}_{\beta_k}(v_k),$$

which is the result in (3.28). □

From Theorem 3.12 then $v_k = \frac{1}{\|\psi_k\|^2_{H^1(\Omega)}}\langle \delta\sigma_i - s_i\nabla_s J(\sigma_i), \psi_k\rangle_{H^1(\Omega)}$ as we have an orthogonal basis [19], however, we shall later look at bases that are only *almost* orthogonal, i.e. that almost all the basis functions are mutually orthogonal. Therefore the notation in terms of expansion coefficients $v_k$ is useful in order to remain consistent.

## 3.3 Total Variation Regularization

While the main focus for this thesis is on the sparsity regularization, there is another interesting form of regularization that will be investigated called *total variation* (TV) regularization. The reason for looking into this type of regularization is due to its ability to make piecewise constant reconstructions, something that is normally quite hard to do with regularization techniques that usually finds smooth reconstructions. In Chapter 4 on page 57 we will experiment with piecewise constant inclusions and while TV regularization should be tailored for that type of solutions, the ill-posedness of the problem will prove a harsh challenge. Now to actually define TV.

**Definition 3.13 (Total Variation)** *Let $f : \Omega \to \mathbb{R}$ such that $\nabla f \in L^1(\Omega)$, the total variation $P_{\mathrm{TV}}(f)$ of $f$ is*

$$P_{\mathrm{TV}}(f) := \int_\Omega |\nabla f| dx. \tag{3.38}$$

Thus ideally we would like the penalty term $P$ in Definition 3.1 to be $P_{\mathrm{TV}}$, however, everything becomes much easier when the penalty term is Gâteaux differentiable, and since $x \mapsto |x|$ is not differentiable at $x = 0$, so is $P_{\mathrm{TV}}(f)$ not differentiable at $\nabla f = 0$. To accommodate this we introduce a parameter $c > 0$ via

$$P_{\mathrm{TV},c}(f) := \int_\Omega \sqrt{|\nabla f|^2 + c dx}. \tag{3.39}$$

Note that $P_{\mathrm{TV},0} = P_{\mathrm{TV}}$, and to illustrate the behaviour of $P_{\mathrm{TV},c}$ with respect to $c$, it is shown in Figure 3.2 how $\sqrt{|x|^2 + c}$ approximates $|x|$ quite well for sufficiently small $c$, while retaining a smooth corner at $x = 0$.

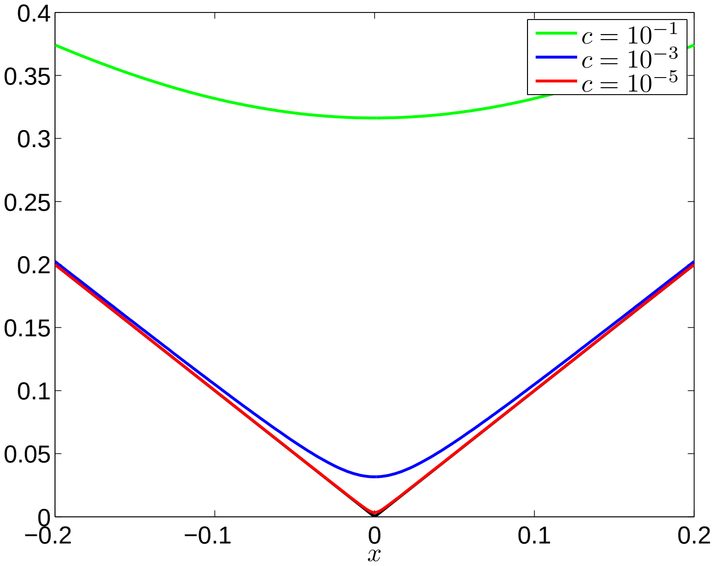


**Figure 3.2:** Approximating $|x|$ (black) by $\sqrt{|x|^2 + c}$ for various values of $c$.

The regularization will be controlled by $\beta$ and $c$ as shown in the following definition, where $\Psi_{\mathrm{TV}}$ is the objective function for the TV regularization.

**Definition 3.14** *Let $\delta\sigma \in \mathcal{A}_0$, and let $\beta > 0$ and $c > 0$ then the functional $\Psi_{\mathrm{TV}}$ is defined as*

$$\Psi_{\mathrm{TV}}(\delta\sigma) = J(\sigma_0 + \delta\sigma) + \beta P_{\mathrm{TV},c}(\delta\sigma).$$

Now let's find some derivatives of $P_{\mathrm{TV},c}$, as they will be used in the linearisation for the gradient method.

**Lemma 3.15 (First and Second Order Derivatives of $P_{\mathrm{TV,c}}$)** *The first and second order Gâteaux derivatives for $P_{\mathrm{TV},c}$ at $\delta\sigma$ are*

$$(P_{\mathrm{TV},c})'_{\delta\sigma}\eta = \int_\Omega f^{-1/2}\nabla\delta\sigma \cdot \nabla\eta dx, \tag{3.40}$$

$$(P_{\mathrm{TV},c})''_{\delta\sigma}\eta = \int_\Omega f^{-1/2}|\nabla\eta|^2 - f^{-3/2}(\nabla\delta\sigma \cdot \nabla\eta)^2 dx. \tag{3.41}$$

*where* $f := |\nabla\delta\sigma|^2 + c$.

**Proof.** Let $f_\epsilon := |\nabla(\delta\sigma + \epsilon\eta)|^2 + c$, then

$$\frac{d}{d\epsilon} f_\epsilon = 2\nabla(\delta\sigma + \epsilon\eta) \cdot \nabla\eta.$$

Starting at the beginning we find $(P_{\text{TV},c})'_{\delta\sigma}$:

$$\begin{aligned}\frac{d}{d\epsilon} P_{\text{TV},c}(\delta\sigma + \epsilon\eta) &= \int_\Omega \frac{d}{d\epsilon}\sqrt{f_\epsilon}dx \\ &= \int_\Omega \frac{\frac{d}{d\epsilon} f_\epsilon}{2\sqrt{f_\epsilon}} dx \\ &= \int_\Omega \frac{\nabla(\delta\sigma + \epsilon\eta) \cdot \nabla\eta}{\sqrt{f_\epsilon}} dx\end{aligned}$$

thus

$$(P_{\text{TV},c})'_{\delta\sigma}\eta = \frac{d}{d\epsilon} P_{\text{TV},c}(\delta\sigma + \epsilon\eta)|_{\epsilon=0} = \int_\Omega f^{-1/2}\nabla\delta\sigma \cdot \nabla\eta dx.$$

Now for the second order derivative:

$$\begin{aligned}\frac{d}{d\epsilon}(P_{\text{TV},c})'_{\delta\sigma+\epsilon\eta}\eta &= \int_\Omega \frac{d}{d\epsilon} f_\epsilon^{-1/2}\nabla(\delta\sigma + \epsilon\eta) \cdot \nabla\eta dx \\ &= \int_\Omega f_\epsilon^{-1/2}|\nabla\eta|^2 - \frac{1}{2} f_\epsilon^{-3/2}\nabla(\delta\sigma + \epsilon\eta) \cdot \nabla\eta \frac{d}{d\epsilon} f_\epsilon dx \\ &= \int_\Omega f_\epsilon^{-1/2}|\nabla\eta|^2 - f_\epsilon^{-3/2}(\nabla(\delta\sigma + \epsilon\eta) \cdot \nabla\eta)^2 dx,\end{aligned}$$

thus

$$\begin{aligned}(P_{\text{TV},c})''_{\delta\sigma}\eta &= \frac{d}{d\epsilon}(P_{\text{TV},c})'_{\delta\sigma+\epsilon\eta}\eta|_{\epsilon=0} \\ &= \int_\Omega f^{-1/2}|\nabla\eta|^2 - f^{-3/2}(\nabla\delta\sigma \cdot \nabla\eta)^2 dx,\end{aligned}$$

giving the expression from (3.41). □

Now similar to (3.26), we get the following using a second order approximation to the TV-term

$$\begin{aligned}\Psi_{\text{TV}}(\delta\sigma) - \Psi_{\text{TV}}(\delta\sigma_i) &= J(\sigma) - J(\sigma_i) + \beta P_{\text{TV},c}(\delta\sigma) - \beta P_{\text{TV},c}(\delta\sigma_i) \\ &\simeq \langle\nabla_s J(\sigma_i), \delta\sigma - \delta\sigma_i\rangle_{H^1(\Omega)} + \frac{1}{2s_i}\|\delta\sigma - \delta\sigma_i\|^2_{H^1(\Omega)} \\ &\quad + \beta(P_{\text{TV},c})'_{\delta\sigma_i}(\delta\sigma - \delta\sigma_i) + \frac{\beta}{2}(P_{\text{TV},c})''_{\delta\sigma_i}(\delta\sigma - \delta\sigma_i).\end{aligned} \tag{3.42}$$

Similar to Theorem 3.12 we can simplify this problem.

**Theorem 3.16** *Consider the following optimization problem,*

$$\min_{\delta\sigma\in\mathcal{A}_0} \Upsilon(\delta\sigma) = \frac{1}{2}\|\delta\sigma-\gamma_i\|^2_{H^1(\Omega)} + \frac{s_i\beta}{2}\int_\Omega f_i^{-1/2}|\nabla\delta\sigma|^2 dx \\ + \frac{s_i\beta}{2}\int_\Omega f_i^{-3/2}(\nabla\delta\sigma_i\cdot\nabla\delta\sigma)(2|\nabla\delta\sigma_i|^2-(\nabla\delta\sigma_i\cdot\nabla\delta\sigma))dx, \tag{3.43}$$

*where $f_i := |\nabla\delta\sigma_i|^2+c$ and $\gamma_i := \delta\sigma_i - s_i\nabla_s J(\sigma_i)$. Then the following holds:*

*(i)* (3.43) *is equivalent with minimizing* (3.42) *w.r.t. $\delta\sigma\in\mathcal{A}_0$.*

*(ii)* *$\Upsilon$ is convex on $\mathcal{A}_0$.*

*(iii)* *The solution to* (3.43) *satisfies*

$$\langle\delta\sigma,\eta\rangle_{H^1(\Omega)} + s_i\beta\int_\Omega f_i^{-1/2}\nabla\delta\sigma\cdot\nabla\eta - f_i^{-3/2}(\nabla\delta\sigma_i\cdot\nabla\delta\sigma)(\nabla\delta\sigma_i\cdot\nabla\eta)dx \\ = \langle\gamma_i,\eta\rangle_{H^1(\Omega)} - s_i\beta\int_\Omega f_i^{-3/2}|\nabla\delta\sigma_i|^2\nabla\delta\sigma_i\cdot\nabla\eta dx,\ \ \forall\eta\in H_0^1(\Omega). \tag{3.44}$$

**Proof.** The convexity of the first term of (3.43) follows from the proof of Theorem 3.12, and similarly can be done for the other squared terms, while the linear term is convex by definition. Furthermore, $\mathcal{A}_0$ is convex by Corollary 2.6. Thus $\Upsilon$ is indeed convex on $\mathcal{A}_0$.

For equivalence between (3.42) and (3.43), the first term of (3.43) follows the proof of Theorem 3.12 along with the multiplication of $s_i$. Thus what remains is to show that $s_i\beta(P_{\text{TV},c})'_{\delta\sigma_i}(\delta\sigma-\delta\sigma_i)+\frac{s_i\beta}{2}(P_{\text{TV},c})''_{\delta\sigma_i}(\delta\sigma-\delta\sigma_i)$ equals the final terms of (3.43) up to multiplication of a constant and addition of a constant. Thus we simply write up and expand the terms:

$$s_i\beta(P_{\text{TV},c})'_{\delta\sigma_i}(\delta\sigma-\delta\sigma_i) + \frac{s_i\beta}{2}(P_{\text{TV},c})''_{\delta\sigma_i}(\delta\sigma-\delta\sigma_i) \\ = s_i\beta\int_\Omega f_i^{-1/2}(\nabla\delta\sigma_i\cdot\nabla\delta\sigma - |\nabla\delta\sigma_i|^2) + \frac{1}{2}f_i^{-1/2}|\nabla(\delta\sigma-\delta\sigma_i)|^2 \\ - \frac{1}{2}f_i^{-3/2}(\nabla\delta\sigma_i\cdot\nabla(\delta\sigma-\delta\sigma_i))^2 dx \\ = \frac{s_i\beta}{2}\int_\Omega f_i^{-1/2}|\nabla\delta\sigma|^2 + f_i^{-3/2}(\nabla\delta\sigma_i\cdot\nabla\delta\sigma)(2|\nabla\delta\sigma_i|^2-(\nabla\delta\sigma_i\cdot\nabla\delta\sigma)) \\ - f_i^{-1/2}|\nabla\delta\sigma_i|^2 - f_i^{-3/2}|\nabla\delta\sigma_i|^4 dx, \tag{3.45}$$

thus removing the constant term $-\frac{s_i\beta}{2}\int_\Omega f_i^{-1/2}|\nabla\delta\sigma_i|^2 + f_i^{-3/2}|\nabla\delta\sigma_i|^4 dx$ yields the expression in (3.43).

Now in order to solve (3.43), we need find a stationary point as the problem is convex, and thus solve $\Upsilon'_{\delta\sigma}\eta = 0,\ \ \forall\eta\in H_0^1(\Omega)$. So first of all $\Upsilon'_{\delta\sigma}$ has to be determined. This will be done in multiple steps for clarity,

$$\frac{d}{d\epsilon}\frac{1}{2}\|\delta\sigma+\epsilon\eta-\gamma_i\|^2_{H^1(\Omega)} = \langle\delta\sigma+\epsilon\eta-\gamma_i,\eta\rangle_{H^1(\Omega)}, \tag{3.46}$$

$$\frac{d}{d\epsilon}\frac{1}{2}|\nabla(\delta\sigma+\epsilon\eta)|^2 = \nabla(\delta\sigma+\epsilon\eta)\cdot\nabla\eta, \tag{3.47}$$

$$\frac{d}{d\epsilon}\nabla\delta\sigma_i\cdot\nabla(\delta\sigma+\epsilon\eta) = \nabla\delta\sigma_i\cdot\nabla\eta, \tag{3.48}$$

$$\frac{d}{d\epsilon}\frac{1}{2}(\nabla\delta\sigma_i\cdot\nabla(\delta\sigma+\epsilon\eta))^2 = (\nabla\delta\sigma_i\cdot\nabla(\delta\sigma+\epsilon\eta))(\nabla\delta\sigma_i\cdot\nabla\eta). \tag{3.49}$$

Now inserting the expressions (3.46) through (3.49) into $\frac{d}{d\epsilon}\Upsilon(\delta\sigma+\epsilon\eta)|_{\epsilon=0}$ yields

$$\begin{aligned}\Upsilon'_{\delta\sigma}\eta = \langle\delta\sigma-\gamma_i,\eta\rangle_{H^1(\Omega)} + s_i\beta\int_\Omega f_i^{-1/2}\nabla\delta\sigma\cdot\nabla\eta + f_i^{-3/2}|\nabla\delta\sigma_i|^2\nabla\delta\sigma_i\cdot\nabla\eta \\ - f_i^{-3/2}(\nabla\delta\sigma_i\cdot\nabla\delta\sigma)(\nabla\delta\sigma_i\cdot\nabla\eta)dx,\ \ \eta\in H_0^1(\Omega).\end{aligned} \tag{3.50}$$

Now solving $\Upsilon'_{\delta\sigma}\eta = 0,\ \ \forall\eta\in H_0^1(\Omega)$ yields (3.44) via (3.50). □

The expression (3.44) seems awfully complicated and seemingly hard to solve, however, we may also recognize that it is similar to a weak formulation of a PDE. Hence it can be solved numerically as such using for instance a finite element method. However, before solving the above problem numerically, it is always convenient to show that there exists a unique solution.

**Corollary 3.17** *The expression in (3.44) has a unique solution $\delta\sigma\in H_0^1(\Omega)$.*

**Proof.** The proof will be an exercise in the Lax-Milgram theorem (Theorem C.3). From (3.44) the forms for Theorem C.3 has already been separated on each side of the equality sign,

$$\begin{aligned}B(\delta\sigma,\eta) &:= \langle\delta\sigma,\eta\rangle_{H^1(\Omega)} + s_i\beta\int_\Omega f_i^{-1/2}\nabla\delta\sigma\cdot\nabla\eta dx \\ &\quad - s_i\beta\int_\Omega f_i^{-3/2}(\nabla\delta\sigma_i\cdot\nabla\delta\sigma)(\nabla\delta\sigma_i\cdot\nabla\eta)dx, \\ L(\eta) &:= \langle\gamma_i,\eta\rangle_{H^1(\Omega)} - s_i\beta\int_\Omega f_i^{-3/2}|\nabla\delta\sigma_i|^2\nabla\delta\sigma_i\cdot\nabla\eta dx.\end{aligned}$$

It is obvious that $B$ is bilinear and that $L$ is linear. Notice that $H_0^1(\Omega)$ is a closed subspace of $\tilde{H}^1(\Omega)$ in the $\|\cdot\|_{\tilde{H}^1(\Omega)}$-norm, thus $H_0^1(\Omega)$ is indeed a Hilbert space when equipped with the $\tilde{H}^1(\Omega)$ inner product from Lemma 2.2. Now to show that $L$ is bounded in $H_0^1(\Omega)$. Here Cauchy-Schwartz' inequality and Lemma 2.2 is applied for the first term in $L$, and for the second term a simple upper bound is applied via the supremum followed by Cauchy-Schwartz' inequality:

$$\begin{aligned} |L(\eta)| &\leq |\langle \gamma_i, \eta\rangle_{H^1(\Omega)}| + s_i\beta\left|\int_\Omega f_i^{-3/2}|\nabla\delta\sigma_i|^2\nabla\delta\sigma_i\cdot\nabla\eta dx\right| \\ &\leq \|\gamma_i\|_{H^1(\Omega)}\|\eta\|_{H^1(\Omega)} + s_i\beta\sup_\Omega\left[f_i^{-3/2}|\nabla\delta\sigma_i|^2\right]|\langle\delta\sigma_i,\eta\rangle_{\tilde{H}^1(\Omega)}| \\ &\leq \left(C\|\gamma_i\|_{H^1(\Omega)} + s_i\beta\|\delta\sigma_i\|_{\tilde{H}^1(\Omega)}\sup_\Omega\left[f_i^{-3/2}|\nabla\delta\sigma_i|^2\right]\right)\|\eta\|_{\tilde{H}^1(\Omega)}. \end{aligned} \tag{3.51}$$

Here it should be noted that $\gamma_i, \delta\sigma_i \in H_0^1(\Omega)$ and the above expression therefore makes sense. What remains for $L$ to be bounded via (3.51) is to show that the expression $\sup_\Omega\left[f_i^{-3/2}|\nabla\delta\sigma_i|^2\right]$ is bounded:

$$\begin{aligned} f_i^{-3/2}|\nabla\delta\sigma_i|^2 &= \frac{|\nabla\delta\sigma_i|^2}{(|\nabla\delta\sigma_i|^2+c)^{3/2}} \leq \frac{|\nabla\delta\sigma_i|^2+c}{(|\nabla\delta\sigma_i|^2+c)^{3/2}} \\ &= (|\nabla\delta\sigma_i|^2+c)^{-1/2} \leq c^{-1/2} < \infty, \end{aligned} \tag{3.52}$$

thus $L$ is a linear and bounded functional on $H_0^1(\Omega)$.

Now let's turn the attention towards $B$, where Theorem C.3(i) will be shown:

$$\begin{aligned} |B(\delta\sigma,\eta)| \leq |\langle\delta\sigma,\eta\rangle_{H^1(\Omega)}| &+ s_i\beta\left|\int_\Omega f_i^{-1/2}\nabla\delta\sigma\cdot\nabla\eta dx\right| \\ &+ s_i\beta\int_\Omega|f_i^{-3/2}(\nabla\delta\sigma_i\cdot\nabla\delta\sigma)(\nabla\delta\sigma_i\cdot\nabla\eta)|dx. \end{aligned}$$

For the first term we will again make use of Cauchy-Schwartz' inequality and Lemma 2.2. For the second term $\sup_\Omega f_i^{-1/2}$ will be pulled outside and Cauchy-Schwartz' inequality will be applied to the remaining inner product. Lastly for the third term Cauchy-Schwartz' inequality will be applied to get $|f_i^{-3/2}(\nabla\delta\sigma_i\cdot\nabla\delta\sigma)(\nabla\delta\sigma_i\cdot\nabla\eta)| \leq f_i^{-3/2}|\nabla\delta\sigma_i|^2|\nabla\delta\sigma||\nabla\eta|$, and $\sup_\Omega f_i^{-3/2}|\nabla\delta\sigma_i|^2$ is pulled out, and finally Hölder's inequality is applied to the remaining integral:

$$\begin{aligned}|B(\delta\sigma,\eta)| \leq{}& C^2\|\delta\sigma\|_{\tilde{H}^1(\Omega)}\|\eta\|_{\tilde{H}^1(\Omega)} + s_i\beta\sup_\Omega f_i^{-1/2}|\langle\delta\sigma,\eta\rangle_{\tilde{H}^1(\Omega)}| \\ &+ s_i\beta\sup_\Omega\left[f_i^{-3/2}|\nabla\delta\sigma_i|^2\right]\|\nabla\delta\sigma\|_{L^2(\Omega)}\|\nabla\eta\|_{L^2(\Omega)} \\ ={}& \left(C^2 + s_i\beta\sup_\Omega\left[f_i^{-1/2} + f_i^{-3/2}|\nabla\delta\sigma_i|^2\right]\right)\|\delta\sigma\|_{\tilde{H}^1(\Omega)}\|\eta\|_{\tilde{H}^1(\Omega)}. \end{aligned} \tag{3.53}$$

What remains in order to show Theorem C.3(i) from (3.53) is that the expression $\sup_\Omega\left[f_i^{-1/2} + f_i^{-3/2}|\nabla\delta\sigma_i|^2\right]$ is bounded, which is done by using (3.52) and that $f_i^{-1/2} \leq c^{-1/2} < \infty$.

Now to show Theorem C.3(ii):

$$B(\delta\sigma,\delta\sigma) = \|\delta\sigma\|^2_{H^1(\Omega)} + s_i\beta\int_\Omega f_i^{-1/2}|\nabla\delta\sigma|^2dx - s_i\beta\int_\Omega f_i^{-3/2}(\nabla\delta\sigma_i\cdot\nabla\delta\sigma)^2dx.$$

For the first term Lemma 2.2 will be applied. For the third term due to the negativity sign, that $s_i\beta > 0$, and that the integrand is non-negative, then Cauchy-Schwartz' inequality can be applied as a $\geq$ inequality for $\nabla\delta\sigma_i\cdot\nabla\delta\sigma$:

$$\begin{aligned}B(\delta\sigma,\delta\sigma) \geq{}& C\|\delta\sigma\|^2_{\tilde{H}^1(\Omega)} + s_i\beta\int_\Omega\left[f_i^{-1/2} - f_i^{-3/2}|\nabla\delta\sigma_i|^2\right]|\nabla\delta\sigma|^2dx \\ \geq{}& \left(C + s_i\beta\inf_\Omega\left[f_i^{-1/2} - f_i^{-3/2}|\nabla\delta\sigma_i|^2\right]\right)\|\delta\sigma\|^2_{\tilde{H}^1(\Omega)}. \end{aligned} \tag{3.54}$$

Since the constant $C$ from Lemma 2.2 is positive, what remains in order to show Theorem C.3(ii) from (3.54) is to show that $\inf_\Omega\left[f_i^{-1/2} - f_i^{-3/2}|\nabla\delta\sigma_i|^2\right]$ is bounded from below and does not make the expression in front of $\|\delta\sigma\|^2_{\tilde{H}^1(\Omega)}$ negative:

$$f_i^{-1/2} - f_i^{-3/2}|\nabla\delta\sigma_i|^2 \geq f_i^{-1/2} - f_i^{-3/2}(|\nabla\delta\sigma_i|^2 + c) = f_i^{-1/2} - f_i^{-3/2}f_i = 0,$$

thus Theorem C.3(ii) holds.

Hence we can conclude the proof using Theorem C.3, that (3.44) has a unique solution $\delta\sigma \in H^1_0(\Omega)$. □

# 4

# Numerical Experiments Using the Finite Element Method

For the numerical experiments with the previously explained algorithms, the finite element method (FEM) will be applied, which is a very popular method for solving PDE's. While this thesis is not focusing on how FEM works in practice, there will be a small introduction on some properties that will be useful for understanding the solutions from this method.

The elements I will be using are so called continuous Galarkin elements of first degree, which means that the domain $\Omega$ is discretized in a triangle mesh, and FEM solutions will be continuous piecewise linear interpolations between correct values defined at the triangle nodes. This gives a finite dimensional closed space $V_h \subseteq H^1(\Omega)$ (since continuous piecewise linear functions on a bounded domain are weak-differentiable), so solving a PDE with this FEM corresponds to a projection from the solution in $H^1(\Omega)$ and onto $V_h$.

Let $\psi_k \in V_h$ be the function that is zero at all nodes except at the $k$'th node where it has the value 1, which can be written as $\psi_k(x_j) = \delta_{k,j}$ with $\delta_{k,j}$ being Kronecker's delta, and $x_j$ be the location of the $j'th$ node. Note that the support of $\psi_k$ is restricted to the polygon comprised of the neighbouring nodes, see Figure 4.1 on the next page.

$\{\psi_k\}$ is a basis for $V_h$, as any linear combination of piecewise linear functions, is also a piecewise linear function, and the nodes are the only degrees of freedom in $V_h$. Let $u \in V_h$ then we have the following exceptionally useful

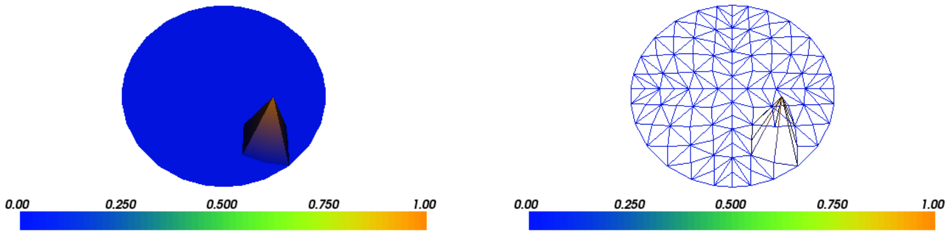


**Figure 4.1:** 3D plot of a single basis function for the FEM basis as a surface plot and wireframe.

property

$$u(x) = \sum_k u(x_k)\psi_k(x), \quad x \in \Omega,$$

where $x_k$ is the location of the $k$'th node. Thus any function in $V_h$ is expanded via its own discretized function at the nodes. It should also be noted that something as simple as a product of two functions in $V_h$ or taking the absolute value of such functions will sometimes lead to functions outside $V_h$, as they can only be determined on a finer mesh, as seen in Figure 4.2 where additional nodes would be required at the zeroes in order to accurately determine the absolute value of a function. For this reason

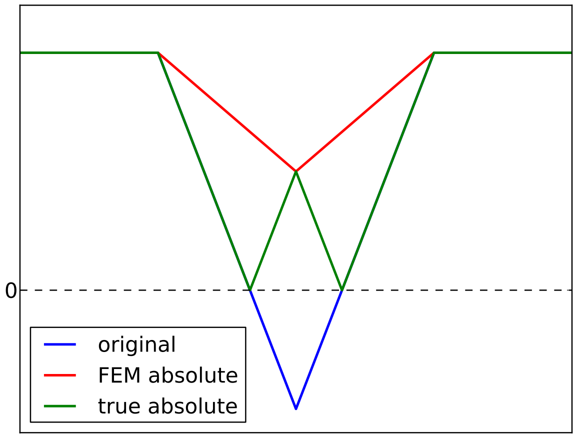


**Figure 4.2:** 1D case of absolute value of a FEM-function on a very coarse grid.

we will approximate such operations such that they remain in $V_h$ and are exact on the mesh nodes, i.e.

$$\begin{aligned} |f(x)| &:= \sum_k |f(x_k)|\psi_k(x), \quad x \in \Omega, f \in V_h, \\ (f_1 f_2)(x) &:= \sum_k f_1(x_k) f_2(x_k) \psi(x), \quad x \in \Omega, f_1, f_2 \in V_h. \end{aligned}$$

Note how $\{\psi_k\}$ is firstly only a basis for $V_h$ which is only a small subspace of $H^1(\Omega)$. Secondly $\{\psi_k\}$ is not orthogonal, namely, due to an overlapping support of the non-negative basis functions corresponding to neighbouring nodes. Thus we have an almost orthogonal basis, as any one basis function is orthogonal to all other basis functions except for the basis functions at neighbouring nodes.

In Jin *et al.* [9] they use the FEM basis anyway assuming orthonormality, since their result for Theorem 3.12 relies on an orthonormal basis. However, $\{\psi_k\}$ is not even an orthogonal basis, and only a basis for a finite dimensional subspace of $H_0^1(\Omega)$. So let's try instead to find an approximate solution to (3.27) using only the assumption that we use $V_h$ and not anything about orthogonality.

**Theorem 4.1 (FEM Approximate Step)** *Given $\delta\sigma_i, \nabla_s J(\sigma_i) \in V_h$ and $R_{\boldsymbol{\alpha},1}$ is given for $V_h$ in terms of the $\{\psi_k\}$-basis. Let $\{x_k\}$ denote the locations of the mesh nodes. Then*

$$\min_{\delta\sigma \in V_h} \Upsilon(\delta\sigma) = \frac{1}{2}\|\delta\sigma - (\delta\sigma_i - s_i \nabla_s J(\sigma_i))\|^2_{H^1(\Omega)} + s_i R_{\boldsymbol{\alpha},1}(\delta\sigma), \tag{4.1}$$

*is* approximately *solved by*

$$\delta\sigma_{i+1} := \sum_k \mathcal{S}_{s_i \alpha_k / \|\psi_k\|_{L^1(\Omega)}}(\delta\sigma_i(x_k) - s_i \nabla_s J(\sigma_i)(x_k))\psi_k, \tag{4.2}$$

*i.e.*

$$\delta\sigma_{i+1}(x_k) = \sum_k \mathcal{S}_{s_i \alpha_k / \|\psi_k\|_{L^1(\Omega)}}(\delta\sigma_i(x_k) - s_i \nabla_s J(\sigma_i)(x_k)).$$

**Proof.** Let $f := \delta\sigma$ with $f_k := \delta\sigma(x_k)$ and $h := \delta\sigma_i - s_i \nabla_s J(\sigma_i)$ with $h_k := \delta\sigma_i(x_k) - s_i \nabla_s J(\sigma_i)(x_k)$. Similarly $f_{x,k}$ etc. will denote $\frac{d}{dx} f(x_k)$, where $f$ here denotes the underlying smoother function that has been discretized.

Assuming $d = 2$ (the proof is similar for $d = 3$), then

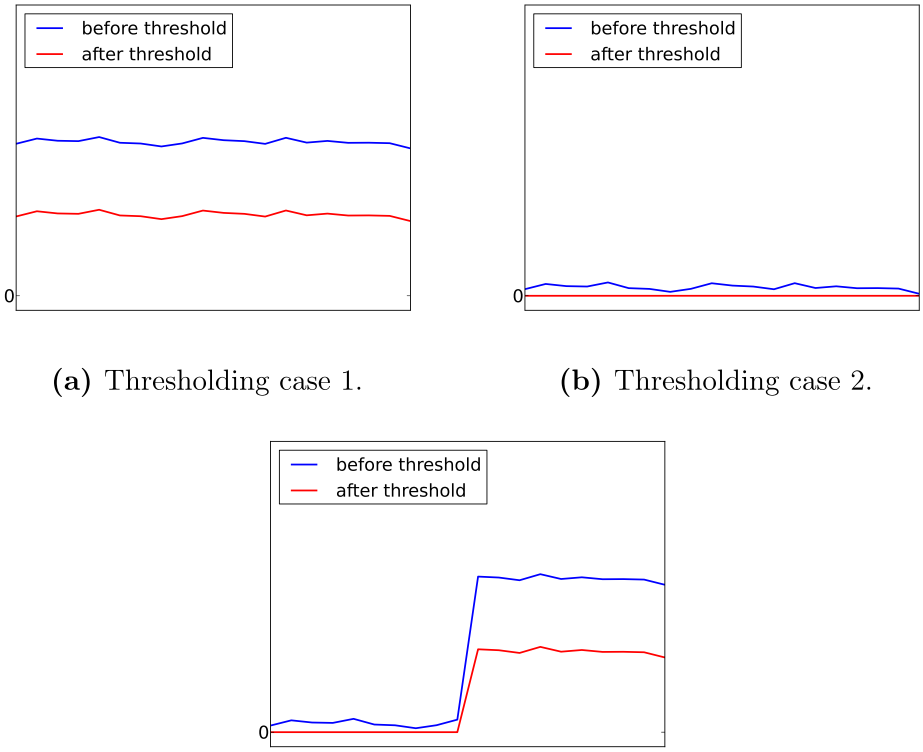
$$\begin{aligned}
\Upsilon(f) &= \frac{1}{2}\|f-h\|^2_{H^1(\Omega)} + \sum_k s_i\alpha_k|f_k| \\
&= \frac{1}{2}\int_\Omega |f-h|^2 dx + \frac{1}{2}\int_\Omega |\nabla f - \nabla h|^2 dx + \sum_k s_i\alpha_k|f_k| \\
&= \frac{1}{2}\int_\Omega \sum_k (f_k - h_k)^2 \psi_k dx \\
&\quad + \frac{1}{2}\int_\Omega \sum_k \left[(f_{x,k}-h_{x,k})^2 + (f_{y,k}-h_{y,k})^2\right]\psi_k dx + \sum_k s_i\alpha_k|f_k| \\
&= \frac{1}{2}\sum_k \left[(f_k-h_k)^2 + (f_{x,k}-h_{x,k})^2 + (f_{y,k}-h_{y,k})^2\right]\|\psi_k\|_{L^1(\Omega)} \\
&\quad + \sum_k s_i\alpha_k|f_k| \\
&= \sum_k \Upsilon_k(f_k, f_{x,k}, f_{y,k}).
\end{aligned}$$

Thus in accordance with the proof of Theorem 3.12 we attempt to minimize each $\Upsilon_k$, however, here we need to do so w.r.t. $f_k$, $f_{x,k}$ and $f_{y,k}$. It is evident that the only stationary point w.r.t. $f_{x,k}$ and $f_{y,k}$ are for $f_{x,k} = h_{x,k}$ and $f_{y,k} = h_{y,k}$. While minimizing solely w.r.t. $f_k$ follows the proof of Theorem 3.12 and yields the result in (4.2).

So in the case that $h$ is not truncated at a node and its neighbouring nodes by the soft-shrinkage, then we have that $\nabla f = \nabla h$ locally since $f = h$ locally up to subtraction with a constant, and it is therefore also optimal in the $f_{x,k}$ and $f_{y,k}$ sense, see Figure 4.3a on the next page.

If $h$ is truncated at a node and its neighbouring nodes, then $f = 0$ and therefore $\nabla f = 0$ locally, however, since $s_i\alpha_k$ is typically very small (even for large step sizes) this implies that $h$ must be locally close to zero for the truncation. Since we have a discretization and use piecewise linear interpolation between the nodes, then $h$ being close to zero implies that $h$ is very flat, i.e. $\nabla h \simeq 0$ and therefore approximately optimal for $f_{x,k}$ and $f_{y,k}$, see Figure 4.3b on the facing page.

In the case that $h$ is truncated at a node, but not for all the neighbouring nodes, then there is an error in this particular $\Upsilon_k$, see Figure 4.3c on the next page. However, in a heuristic sense this is a much smaller error compared to what was done in Jin *et al.* [9] where such approximations would



**(a)** Thresholding case 1.

**(b)** Thresholding case 2.

**(c)** Thresholding case 3.

**Figure 4.3:** Three cases in 1D for the soft-shrinkage threshold.

occur at every node, since it was assumed that $\{\psi_k\}$ was orthonormal (and even an orthonormal basis for $H_0^1(\Omega)$ and not the finite dimensional FEM subspace). □

Thus while Theorem 4.1 is heuristic, it gave better solutions compared to the implementation I made based on Jin *et al.* [9] where $\{\psi_k\}$ was assumed to be an orthonormal basis. Also note that assuming that $\{\psi_k\}$ is orthonormal does not take the area of the support for the basis functions into account as seen in Theorem 3.12, where the normalization term would vanish.

Under any circumstances the parameters $\alpha_k$ will have to be determined for a fixed mesh, since a refinement of a mesh would only slightly change the discrepancy term while the penalty term would be drastically increased, by simply adding more terms to the summation. This is illustrated in

Figure 4.4 where a simple splitting of the mesh triangles yields a great increase in the number of nodes (this is just an example to make the point about using a fixed mesh for the experiments, and not on how to in practice refine a mesh. In practice when refining a mesh by splitting triangles, there is typically also edge flipping involved to make the mesh more even, such as in Delaunay triangulation). Therefore a fixed mesh will be used throughout this section for the computations.

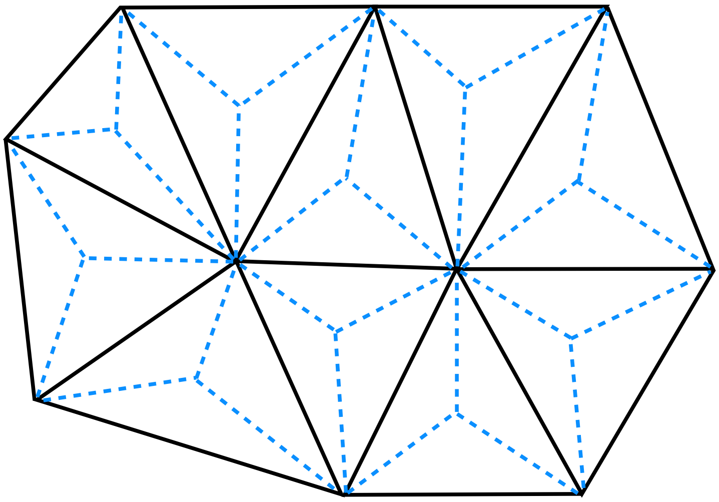
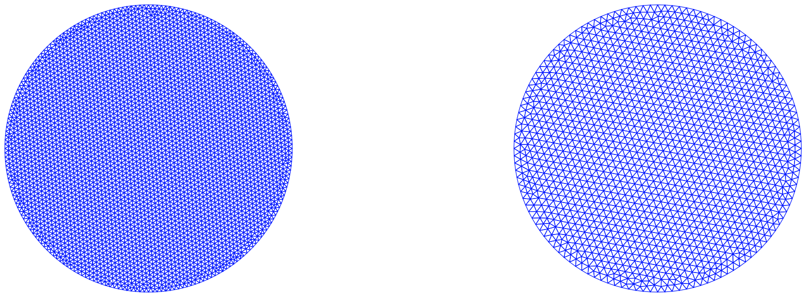


**Figure 4.4:** Illustration of introducing new nodes into a triangulated mesh, by splitting each triangle. Black lines indicate old edges, while the dashed blue lines are the edges from the new nodes.

Since the data used for this thesis will be simulated numerically, this will be done by solving the forward problem with the exact conductivity using a very fine mesh, and afterwards it is interpolated onto a much coarser mesh where noise will be applied, see Figure 4.5 on the next page for the mesh. The reason for simulating the data on a different mesh, is to avoid the so-called inverse crimes, which occurs when the data is determined via the same mesh as the solver is applying. Inverse crimes often gives unnaturally good reconstructions that are not comparable with what you would normally get from real-world data.

To summarize the algorithm that will be applied, pseudo code of the algorithm is presented in Algorithm 1 and Algorithm 2. It is worth noting that a minimum $s_{\min}$ and a maximum $s_{\max}$ initial step size is used. The maximum step size is to avoid exceptionally large step sizes from (3.12)



**(a)** Fine mesh with 8152 triangles. **(b)** Coarse mesh with 2880 triangles.

**Figure 4.5:** Fine and coarse mesh that will be used for numerical experiments.

when $\langle \delta\sigma_i - \delta\sigma_{i-1}, \nabla_s J(\sigma_i) - \nabla J(\sigma_{i-1})\rangle_{H^1(\Omega)} \simeq 0$, and the minimum step size is to check if a slightly larger step size is acceptable if the computed BB step size is very small (and if $s_{\min}$ is not acceptable it will automatically be reduced).

**Algorithm 1** Algorithm for choosing step size.

```
procedure StepSize
    Compute $s_i$ by (3.12)
    Threshold $s_i$ by $s_i := \min\{\max\{s_i, s_{\min}\}, s_{\max}\}$
    while $s_i$ not accepted do
        Compute $\delta\sigma_{i+1}$ using one of the following:
        (3.28) (sparsity)
        (4.2) (sparsity with FEM basis)
        (3.44) (TV)
        if $s_i$ and $\delta\sigma_{i+1}$ satisfies the weak monotonicity (3.17) then
            Accept $s_i$ and $\delta\sigma_{i+1}$
        else
            $s_i := s_i/2$
        end if
    end while
end procedure
```

**Algorithm 2** General framework for the iterative algorithm.

```
procedure EITSOLVER
    Set $i := 0$
    Set $\delta\sigma_i := 0$
    Compute $\Psi_P(\delta\sigma_i)$
    while stopping criteria not reached do
        Set $\sigma_i := \sigma_0 + \delta\sigma_i$
        Compute $\nabla J(\sigma_i)$ by (3.2)
        Compute $\nabla_s J(\sigma_i)$ as the solution of (3.8)
        Compute $s_i$ and $\delta\sigma_{i+1}$ by Algorithm 1
        Compute $\Psi_P(\delta\sigma_{i+1})$
        Set $i := i + 1$
        Check stopping criteria
    end while
    Return $\sigma := \sigma_0 + \delta\sigma_i$
end procedure
```

## 4.1 Preparing for the Numerical Experiments

In the following, EIT reconstruction using either sparsity regularization with the FEM basis or TV regularization are tested in various scenarios to evaluate their respective strengths and weaknesses. Some parameters, however, will be fixed throughout all the experiments, $\tau := 10^{-5}$ and $M := 5$ for the weak monotonicity in Definition 3.7. The parameter $c$ for TV regularization is fixed at $c := 10^{-5}$ since this value produced good results. It should be noted that a too high value of $c$ will generate bad approximations to the true total variation, cf. (3.38) via (3.39), whilst a too small value will make the method unstable due to a singularity in the derivative of the TV penalty term (Lemma 3.15). Furthermore, in sparsity regularization we will use $\alpha_k := \alpha, \ \forall k$, i.e. the same parameter for all basis functions, in all tests except in Section 4.9 on page 115 where we exploit having different parameters to incorporate prior information into the solution. The minimum and maximum initial step sizes for each iteration are $s_{\min} = 1$ and $s_{\max} = 1000$.

The stopping criteria are a maximum number of iteration $I_{\max} := 200$ and lower bound on the step size $s_{\text{stop}} := 10^{-3}$. The use of $s_{\text{stop}}$ has proven quite effective, because once the step size drops this low, numerical experiments have shown that the method does not change significantly even after hundreds of additional iterations, strongly indicative of it being stuck at a local minimum. Another auspicious stopping criteria would be the change in the objective function $\Psi_P$, however, I have chosen not to apply this as the method has given much improved solutions (especially in terms of contrast) even when $\Psi_P$ has only been reduced very little.

The weights $\{w_k\}$ in (3.19) have been included in the implementation, however, for these experiments they are all chosen as $w_k := 1, \ \forall k$, i.e. all the datasets are weighted equally. Furthermore, $\Gamma := \partial\Omega$ in (3.20), except in Section 4.8 on page 106 and Section 4.9 on page 115 where there will be experiments with partial data. The domain $\Omega$ used for the experiments will be the unit disk in $\mathbb{R}^2$.

The Neumann-data that will be used for the following tests are:

$$\begin{aligned} g_N &:= \cos(N\theta), \\ \tilde{g}_N &:= \sin(N\theta), \end{aligned} \tag{4.3}$$

for various $N \in \mathbb{N}$, where $\theta \in [0, 2\pi]$ is the angular variable for polar coordinates in $\Omega$. It is easily checked that this form of Neumann-data satisfies the condition $\int_{\partial\Omega} g ds = 0$.

The perturbation $\epsilon$ in the data $\phi := \mathcal{F}_g(\sigma) + \epsilon$ will be applied to the nodal points on the boundary via

$$\epsilon(x_\ell) := \omega \max_{k=1,\ldots,K} \max_{x_j \in \{x_j\}} |[\mathcal{F}_{g_k}(\sigma)](x_j)| \zeta_\ell, \quad x_\ell \in \{x_j\} \subset \partial\Omega, \tag{4.4}$$

where $K$ denotes the number of Neumann-datasets $g_k$ (which could be a combination of the types in (4.3)), $\{x_j\}$ denotes the nodel points on the boundary, $\omega \in [0, 1]$ denotes the noise level in terms of $L^\infty$, and $\zeta_\ell$ is normally $\mathcal{N}(0, 1)$ distributed. This is in accordance with the perturbations used in Jin *et al.* [9].

Remember that we seek to find $\sigma = \sigma_0 + \delta\sigma$, where $\sigma_0$ is a known background conductivity. The background conductivity $\sigma_0$ will be constant in all tests, since the true challenge lies in determining $\delta\sigma$ and the role of $\sigma_0$ is quite minimal as it is assumed to be known. In all tests but one in Section 4.5 on page 94 I will use the background conductivity $\sigma_0 := 1$ in $\Omega$.

For solving the PDE's numerically I have used the Python based software package FEniCS [31], and despite the existence of tools in FEniCS to generate mesh for simple domains such as the unit disk, I found that the mesh was quite non-uniform. Therefore I have used the software Gmsh [32] to generate the mesh.

Numerous test of different scenarios will be presented each highlighting specific properties of the solution. Of these tests, I will emphasize Section 4.8 on page 106 and Section 4.9 on page 115 and encourage the reader to focus on these sections, where we look into using partial data and further use of prior information.

From Figure 4.6 it is evident that the sparsity and TV solvers take just about the same time to run each iteration. The TV solver is negligibly slower, due to having to solve the weak problem in (3.44). The main computational burden lies with the number of datasets that is applied, i.e. if $K$ datasets are used then it will need to solve $K$ weak problems for each evaluation of $\Psi_P$, and $K+1$ weak problems for finding $\nabla_s J(\sigma)$ via (3.19) and (3.9) when the forward solutions from the evaluation of $\Psi_P$ is reused. $\Psi_P$ and $\nabla_s J(\sigma)$ have to be evaluated at least once per iteration, and $\Psi_P$ has to further be evaluated if a step size reduction is required. So $(2+Z)K+1$ weak problems have to be solved in an iteration for the sparsity solver and $(2+Z)K+2$ weak problems for the TV solver, where $Z$ denotes the number of times the step size has to be reduced (which usually is $0-1$ times).

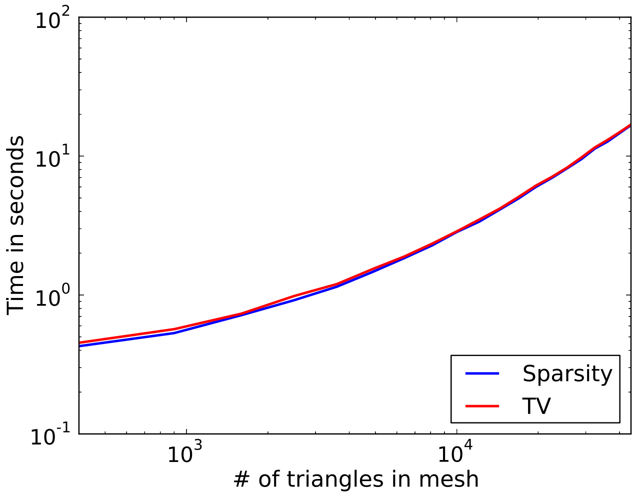


**Figure 4.6:** Computation time for one iteration, for increasingly refined mesh using 10 datasets.

From Figure 4.6 the computation time for one iteration is almost increasing as a straight line in the double-logarithmic scale, i.e. if $C_T$ is the computation time and $N_T$ the no. of triangles in the mesh, then

$$\log_{10}(C_T) \simeq a \log_{10}(N_T) + \log_{10}(b) = \log_{10}(bN_T^a) \Rightarrow C_T \simeq bN_T^a.$$

Fitting a first order polynomial to $(\log_{10}(N_T), \log_{10}(C_T))$ lead to the values $(a, b) = (0.8405, 1.518 \cdot 10^{-3})$ for sparsity and $(a, b) = (0.8303, 1.712 \cdot 10^{-3})$ for TV. So it is quite evident that sparsity and TV are equal in speed per iteration, however, it should be noted that TV tends to use far more iterations than sparsity regularization.

## 4.2 Validating the Implementation of the Forward Map

In Section 2.1 on page 14 we saw a case for which the solution of the forward problem could be determined via known functions in (2.28). This can be used to validate the implementation for the forward problem, which is an implementation of the weak formulation (2.13). While the actual forward map is $\mathcal{F}_g$ and is determined on $\partial\Omega$, I will instead look at $F_g$ which is determined on $\Omega$ since this map is involved in also determining the derivative of the discrepancy term.

For the test I have used a concentric conductivity $\sigma_{C,\tilde{r}}$ (see (2.15)) with radius $\tilde{r} := 0.5$ and $C := 5$, furthermore, $N := 5$ for $g_N := \cos(N\theta)$. In Figure 4.7 on the next page the errors in the $H^1(\Omega)$-norm are given for increasingly refined mesh, and it is evident that the absolute and relative errors decrease with the refinement in the mesh, in a manner resembling a straight line in the double-logarithmic scale. Similar to what was done for Figure 4.6 on the facing page we can determine the rate of decay for the errors. So let $u^*$ be the exact solution and $u$ the numerical approximation, and $N_T$ the no. of triangles in the mesh, then

$$\begin{aligned} \|u^* - u\|_{H^1(\Omega)} &\simeq 1.626 N_T^{-0.4627}, \\ \frac{\|u^* - u\|_{H^1(\Omega)}}{\|u^*\|_{H^1(\Omega)}} &\simeq 2.077 N_T^{-0.4646}. \end{aligned}$$

Thus quadrupling the no. of triangles in the mesh approximately halves the absolute and relative errors.

The reason for the jagged curves in Figure 4.7 on the next page is that I used the built-in mesh for the unit circle in FEniCS [31] to easily scale

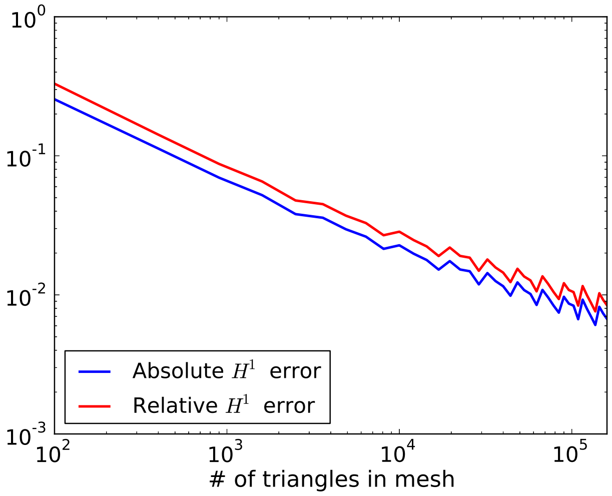
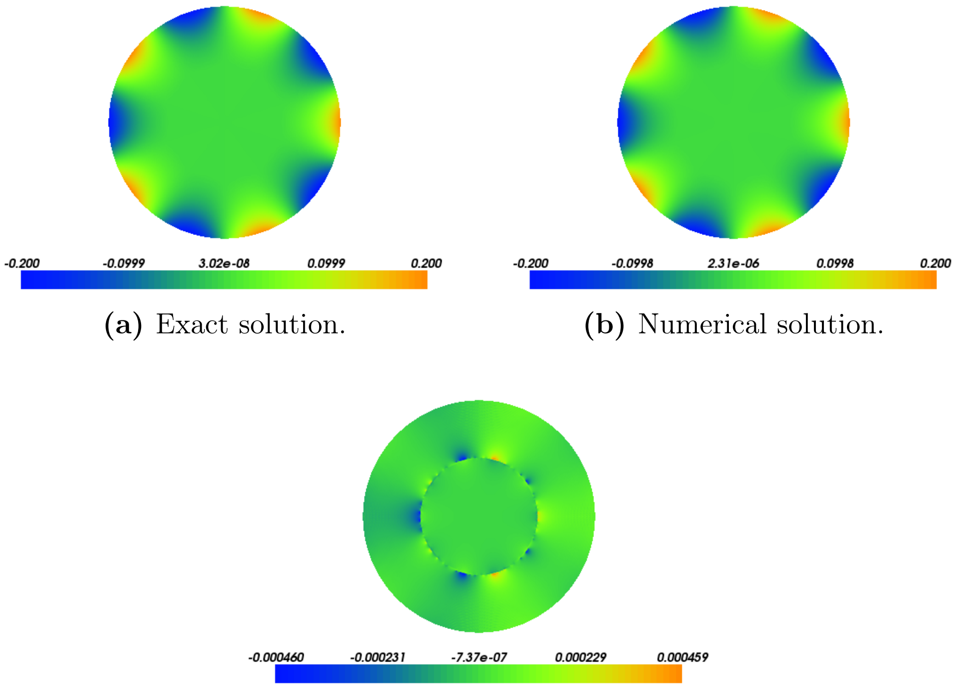


**Figure 4.7:** Absolute and relative errors in the $H^1$-norm, for increasingly refined mesh.

the triangles in the mesh, however, as aforementioned the mesh is not very uniform in this tool, and some configurations of the elements are better suited for the numerical computations.

To elucidate where the errors occur in the implementation, the solution at the finest mesh in Figure 4.7 is shown in Figure 4.8 on the next page, where one can see an imprint of the concentric conductivity with radius 0.5, and it is near the discontinuity of the conductivity that the error is largest. This makes sense as we deal with piecewise linear elements, so no matter how small the elements are, there will always be an error in approximating the discontinuity with a continuous function, which will show up in the numerical solution of the forward map.



**(a)** Exact solution.

**(b)** Numerical solution.

**(c)** Difference.

**Figure 4.8:** Exact and numerical solution of the forward case, and the residual.

## 4.3 Determining Appropriate Regularization Parameters, Number of Datasets, and Noise Level

Choosing a good value for the regularization parameters is key to achieving good reconstructions from regularized solutions.

This section will go through tests to determine good candidates for regularization parameters for sparsity and TV regularization, as well as investigating the influence on the solution by the number of Neumann-datasets applied. Finally, the solutions from sparsity and TV regularization are investigated for different noise levels to see how robust the methods are towards noise.

## Regularization Parameters

To get an idea of the magnitude of a good regularization parameter, I have tested a range of different parameter for the sparsity and TV regularization with noise levels $\omega \in \{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. This is performed using the conductivity phantom in Figure 4.9 and 6 datasets of Neumann-data, $g_N$ and $\tilde{g}_N$ from (4.3) with $N = 1, 2, 3$. It should first of all be noted that the phantom is actually quite difficult to reconstruct in itself, as it is discontinuous which is often something that regularization techniques have huge problems in reconstructing, and the range of the contrast is quite large for such a small inclusion.

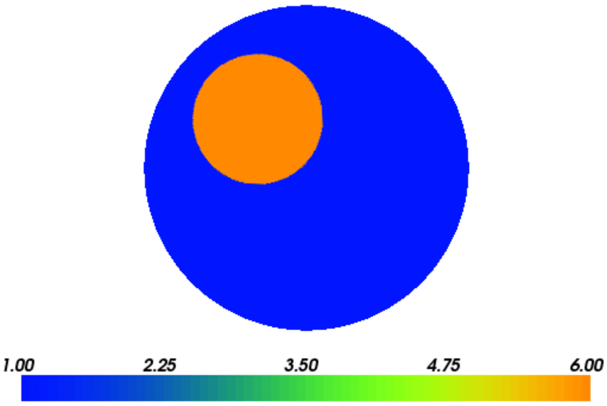
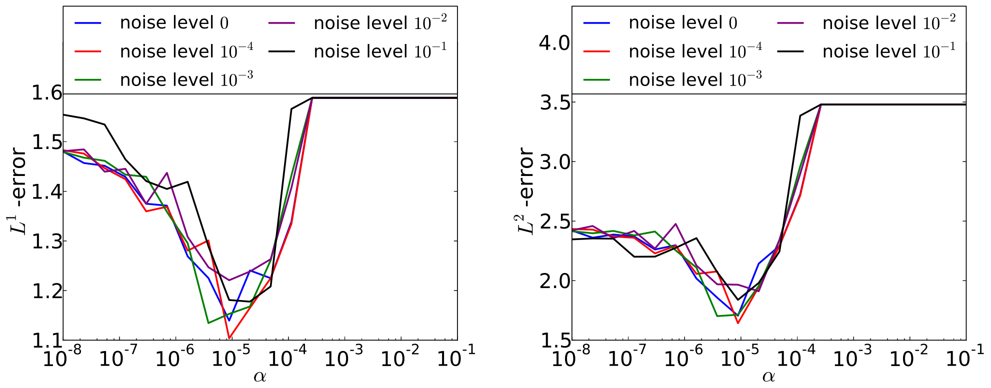


**Figure 4.9:** Conductivity phantom.

In Figure 4.10 on the next page the $L^1$ and $L^2$ absolute errors are shown for the reconstructions with different parameters of $\alpha$, using a few iterations for each reconstruction. The reason for using both $L^1$ and $L^2$ errors, is due to $L^1$ giving much larger errors if the support is not well determined (i.e. if there are small unnecessary oscillations), and $L^2$ errors will be large if the contrast of the inclusion itself is not well reconstructed. It seems that a good choice of parameter is $\alpha := 2 \cdot 10^{-5}$, which will be used for most of the following tests. However, since sparsity regularization using the FEM basis is more sensitive to the regularization parameter, there will be some situations where a different parameter will be applied and this will explicitly be stated.

Similarly, we can investigate the TV regularization, and as seen in Figure 4.11 on the facing page the curve is quite flat for low values of $\beta$, so

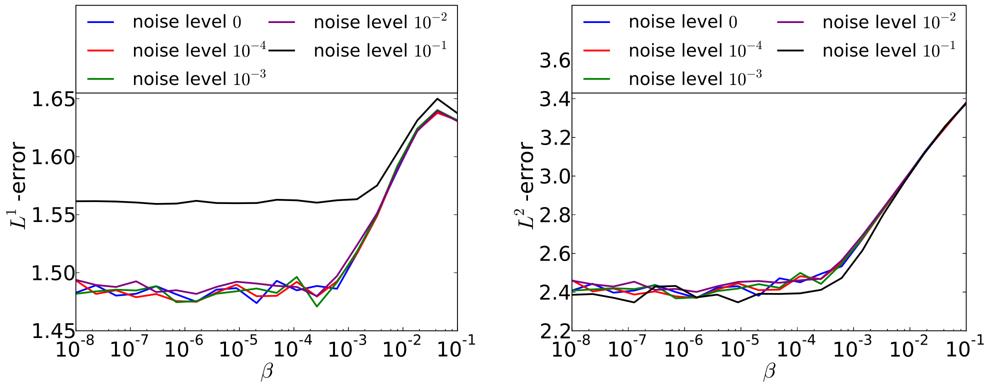


**(a)** Absolute $L^1$ error.

**(b)** Absolute $L^2$ error.

**Figure 4.10:** Sparsity reconstructions for different noise levels, using a maximum of 50 iterations, for varying regularization parameter $\alpha$.

to promote a piecewise constant solution the largest value of $\beta$ that yields a small error is applied. A good parameter is $\beta := 6 \cdot 10^{-4}$, and TV regularization is actually quite stable in terms of the regularization parameter when it comes to different inclusions or mesh, so the parameter will only rarely be changed in the coming tests.

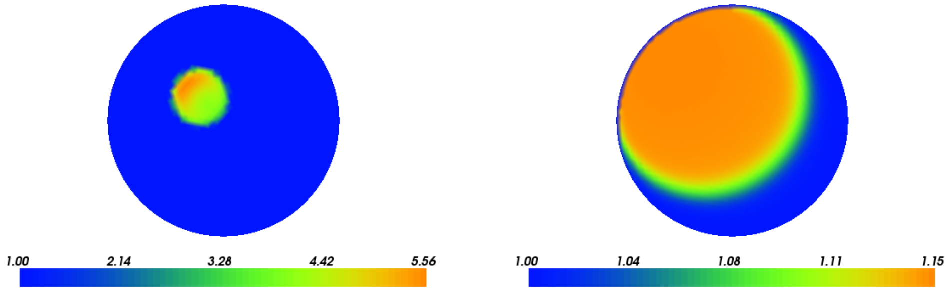


**(a)** Absolute $L^1$ error.

**(b)** Absolute $L^2$ error.

**Figure 4.11:** TV reconstructions for different noise levels, using a maximum of 50 iterations, for varying regularization parameter $\beta$.

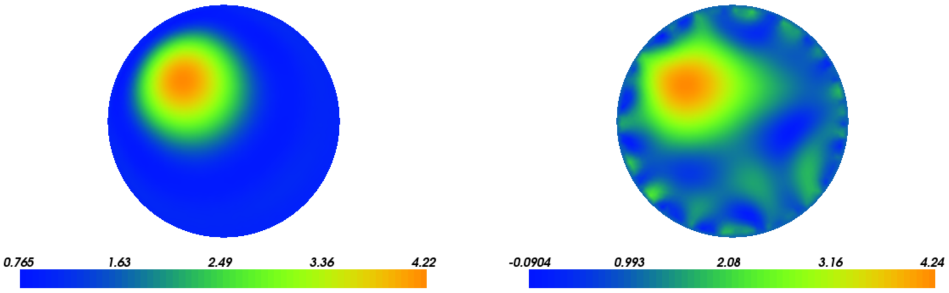
The effects of over regularization, i.e., where the regularization parameter is chosen too large can be seen in Figure 4.12 on the next page. Over regularization in sparsity regularization leads to a very small support, which is quite intuitive with the penalty term that penalizes non-zero values at the nodal points. TV regularization display the opposite case, namely, that the support is vastly overestimated.



**(a)** Sparsity. **(b)** TV.

**Figure 4.12:** Over regularization in sparsity and TV reconstructions using parameters $\alpha = 10^{-4}$ and $\beta = 10^{-1}$. The noise level is $10^{-3}$ and 10 datasets were applied.

In Figure 4.13 we see the effect that noise has on the solution when no regularization is applied. Here it should also be noted that 10 datasets were applied, which significantly improves the solution. It appears that with no noise the method performs okay, however, it is seen that the inclusion itself has negative values as the background conductivity in Figure 4.9 on page 70 is $\sigma_0 := 1$.



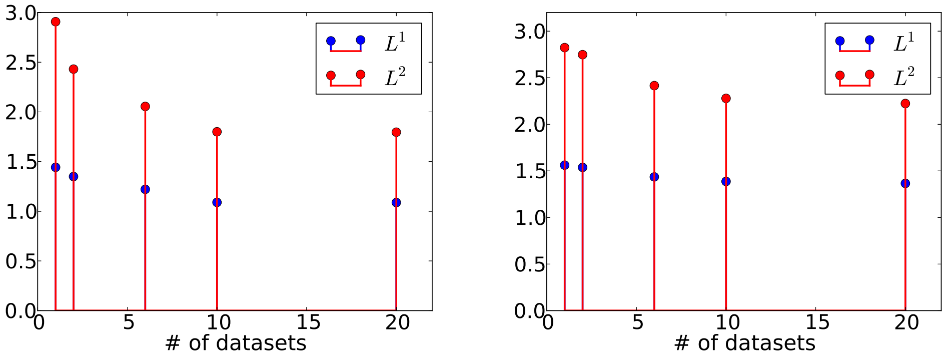
**(a)** Noise level 0. **(b)** Noise level $10^{-1}$.

**Figure 4.13:** Reconstruction using no regularization.

We shall see in Figure 4.24 on page 82 and Figure 4.25 on page 83 that the parameters $\alpha = 2 \cdot 10^{-5}$ and $\beta = 6 \cdot 10^{-4}$ found in this section leads to much better solutions.

## Number of Datasets

Given that good candidates for the regularization parameters have been found, the influence of the number of datasets will now be investigated. In this test the noise level will be fixed at $10^{-3}$, and the tests will be conducted using 1, 2, 6, 10, and 20 datasets.

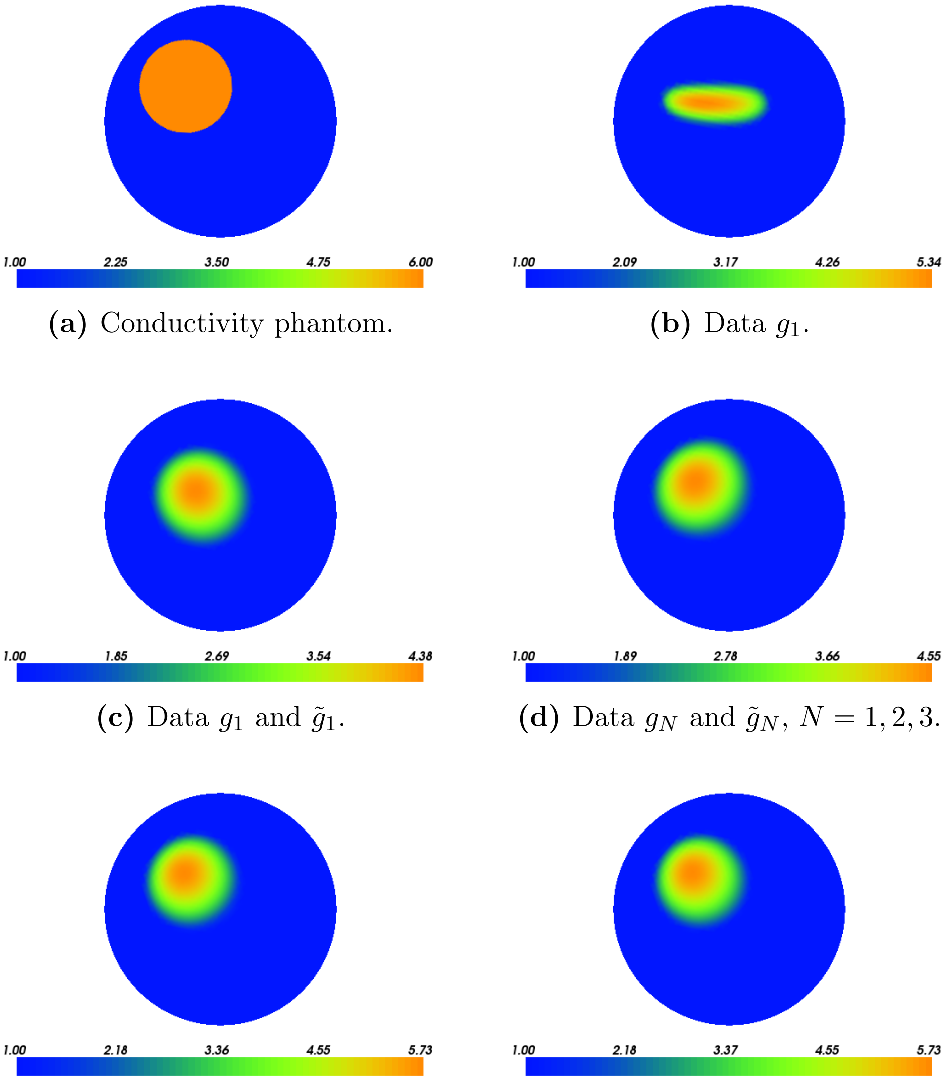


**(a)** Sparsity.

**(b)** TV.

**Figure 4.14:** Absolute error in sparsity and TV reconstructions for different number of Neumann-data, corresponding with Figure 4.15 on the following page and Figure 4.16 on page 75.

In Figure 4.15 on the following page the results for sparsity reconstruction can be seen, where especially the case with 1 dataset is quite bad, which corresponds well with the example in Section 2.1 on page 14 where it was shown that there are cases where 1 dataset is not enough to uniquely invert the data. As expected the results improve with increasing number of datasets, and the same is the case for TV regularization as seen in Figure 4.16 on page 75. However, the doubling from 10 to 20 datasets does not seem to improve much, which is also evident in Figure 4.14 where the insignificant decrease in error is completely unjustified by the vast increase in the number of numerical computations. Therefore the number of datasets used in the rest of the tests here and in the following sections will be fixed at 10.



**(a)** Conductivity phantom.

**(b)** Data $g_1$.

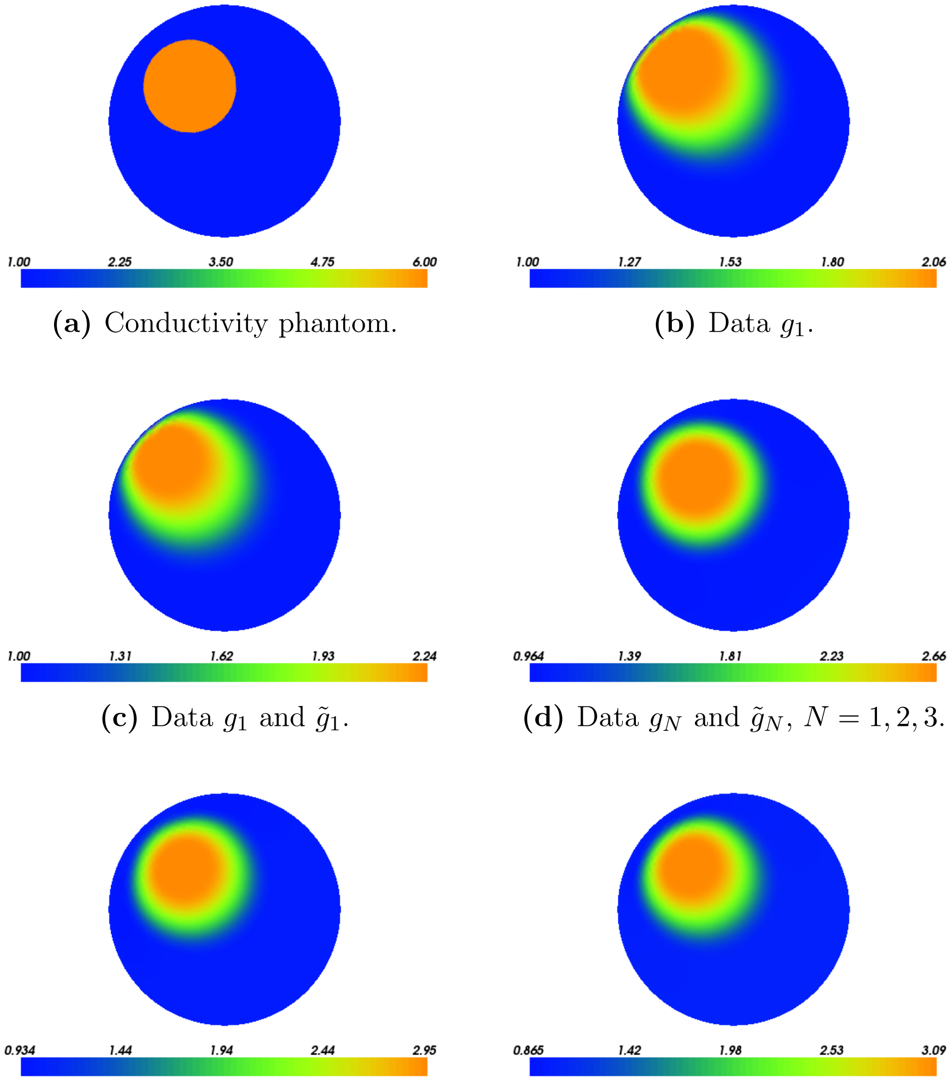
**(c)** Data $g_1$ and $\tilde{g}_1$.

**(d)** Data $g_N$ and $\tilde{g}_N$, $N = 1, 2, 3$.

**(e)** Data $g_N$ and $\tilde{g}_N$, $N = 1, \ldots, 5$.

**(f)** Data $g_N$ and $\tilde{g}_N$, $N = 1, \ldots, 10$.

**Figure 4.15:** Sparsity reconstructions for different number of Neumann-data (4.3) with noise level $10^{-3}$ and regularization parameter $\alpha := 2 \cdot 10^{-5}$.



**(a)** Conductivity phantom.

**(b)** Data $g_1$.

**(c)** Data $g_1$ and $\tilde{g}_1$.

**(d)** Data $g_N$ and $\tilde{g}_N$, $N = 1, 2, 3$.

**(e)** Data $g_N$ and $\tilde{g}_N$, $N = 1, \dots, 5$.

**(f)** Data $g_N$ and $\tilde{g}_N$, $N = 1, \dots, 10$.

**Figure 4.16:** TV reconstructions for different number of Neumann-data (4.3) with noise level $10^{-3}$ and regularization parameter $\beta := 6 \cdot 10^{-4}$.

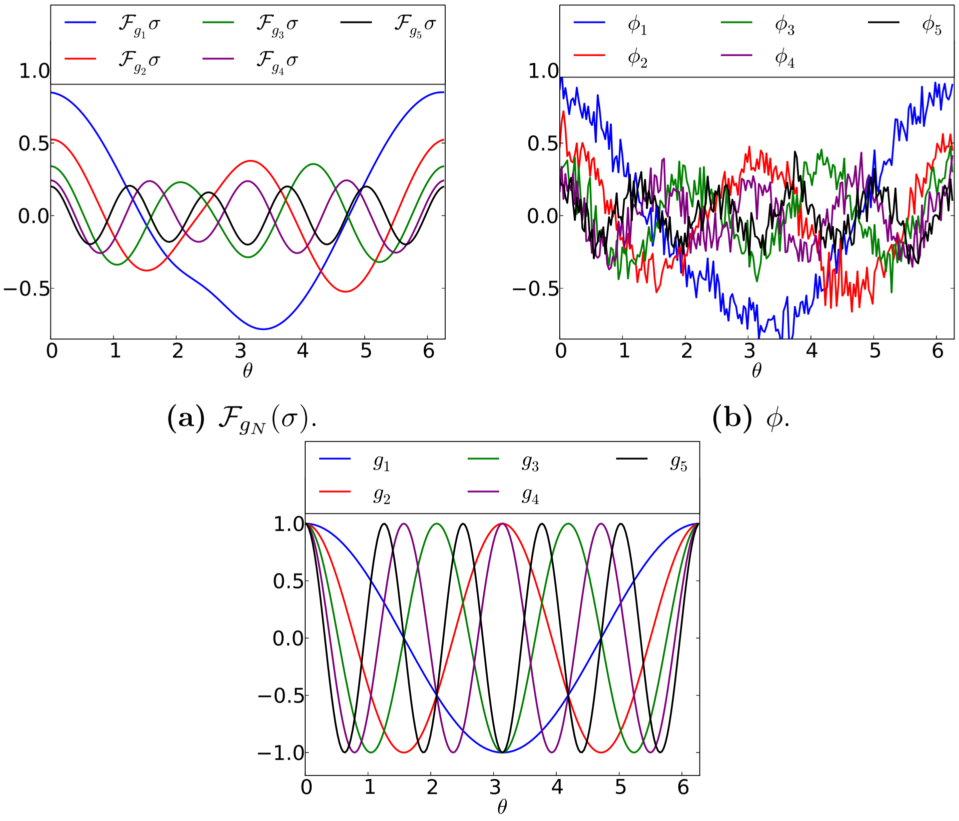
## Noise Level

Now the reconstructions for various noise levels will be investigated. In Figure 4.24 on page 82 it evident that sparsity reconstruction is quite robust, as very similar reconstructions are found for the different noise levels. It is seen that the support is slightly smaller for the reconstructions than the phantom, and this will be characteristic for the sparsity regularization throughout the tests. However, the contrast is quite close to that of the phantom which is excellent given that the contrast is actually quite large for such a small inclusion. For the noise level of $10^{-1}$ we do see that it is slightly more difficult to reconstruct the circular shape of the inclusion.

Figure 4.25 on page 83 shows the reconstructions from TV regularization for various noise levels, and here it is evident that apart from the noise level of $10^{-1}$, the reconstruction are almost identical, and for the noise level of $10^{-1}$ we get quite an awful reconstruction. Unlike the sparsity regularization, the support of the TV reconstruction appear to be slightly larger than the phantom. However, when it comes to the contrast, the TV regularization does not perform quite as well as the sparsity regularization. Furthermore, the TV reconstructions are constant in an area corresponding well with the support of the phantom yet there is still a smooth transition near the boundary of the inclusion, so the purpose of finding piecewise constant solutions with this technique is only partially fulfilled.

The two regularization techniques seem to perform quite well in terms of determining the support of the inclusion, considering the difficulty of the inclusion and the severe ill-posedness of the problem, and the two techniques also seem to excel at reconstructing different properties of the solution. That sparsity regularization underestimates the support and TV regularization overestimates the support may also explain why sparsity regularization gives a higher contrast than TV regularization. In Section 2.1 on page 14 we saw that for concentric conductivities a smaller support and higher contrast leads to similar Dirichlet-data as a larger support and smaller contrast, and this may very well also be the case for non-concentric conductivities.

The exact data $\mathcal{F}_{g_N}(\sigma)$ along with perturbed data are shown in Figure 4.17 on the next page, and it is evident that there is a dent in $\mathcal{F}_{g_N}(\sigma)$ near $\theta = \frac{3\pi}{4} \simeq 2.36$ (the angular variable on the boundary) which is where the inclusion in Figure 4.9 on page 70 is closest to the boundary. The dent is most noticeable for $g_1$ and this small dent is how the inclusion is identified in the data. When we look at the perturbed data in Figure 4.17b it is not surprising that TV regularization has difficulty in reconstructing with this data, however, it is quite amazing that the sparsity regularization performs this well.



**(a)** $\mathcal{F}_{g_N}(\sigma)$.

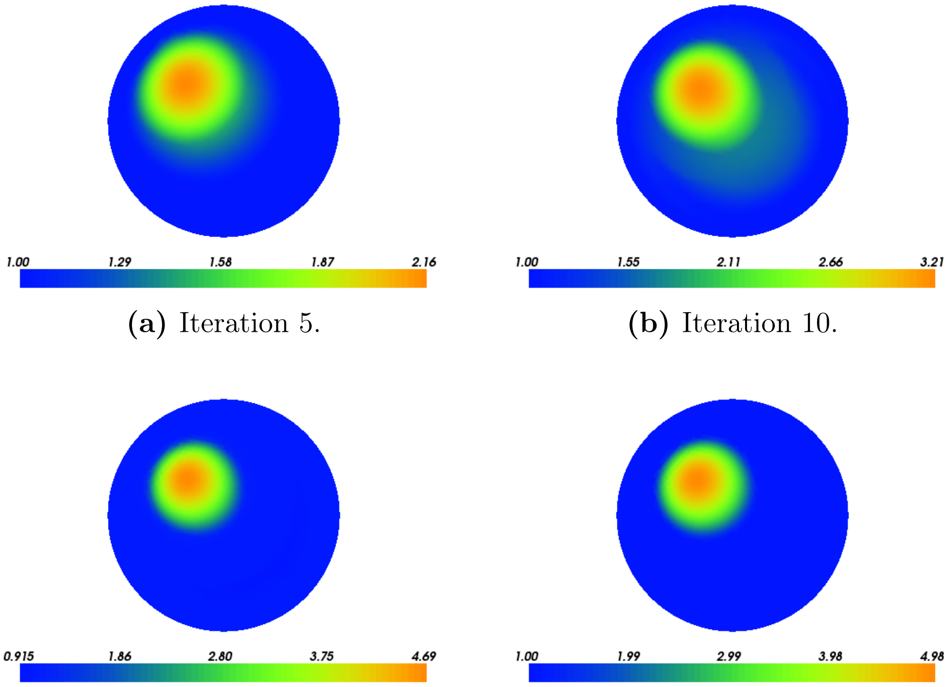
**(b)** $\phi$.

**(c)** $g_N$.

**Figure 4.17:** Exact and perturbed Dirichlet-data with noise level $10^{-1}$ corresponding to the Neumann-data $g_N$ in (4.3) with $N = 1, \ldots, 5$.

To illustrate the performance of the iterative method some of the iterates are shown in Figure 4.18 on the following page for sparse regularization and Figure 4.19 on page 79 for TV regularization, where it is apparent how the support and contrast improve throughout the iterations.

To further show the performance of the iterative method, the $L^1$ and $L^2$ errors are shown in Figure 4.20 on page 80 for sparsity regularization, and the errors for TV regularization are depicted in Figure 4.21 on page 80. In both cases an overall decay is seen in the errors. It is also seen that sparsity



**(a)** Iteration 5.

**(b)** Iteration 10.

**(c)** Iteration 30.

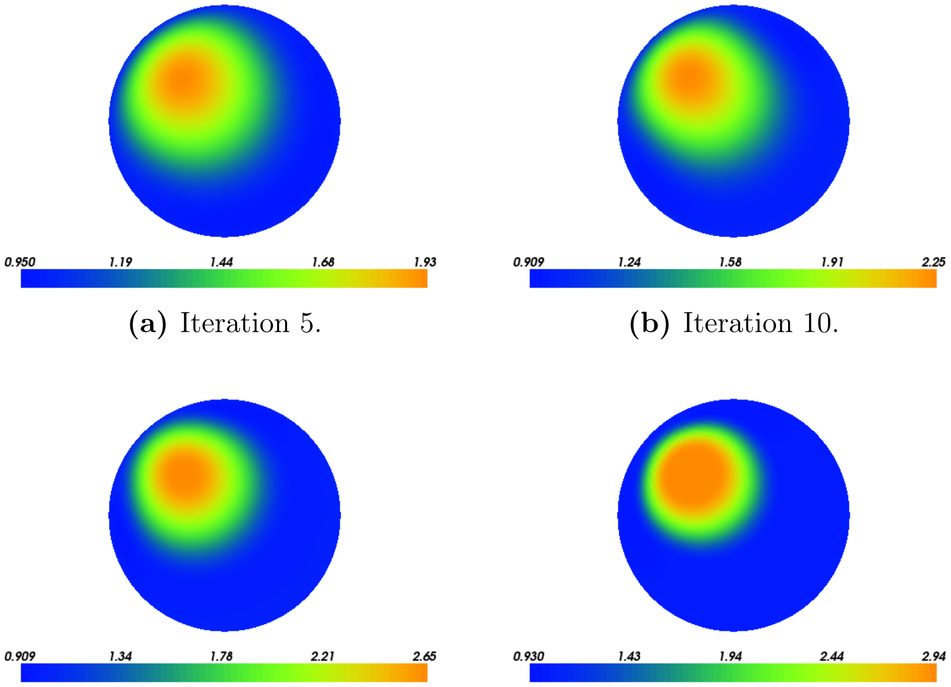
**(d)** Final iteration.

**Figure 4.18:** Sparsity reconstruction at different iterations, with noise level $10^{-3}$ and $\alpha := 2 \cdot 10^{-5}$.

regularization tends to take much fewer iterations while TV regularization makes use of all total 200 iterations. For the case of noise level 0 and noise level $10^{-4}$ there is a sudden drop in the error for sparsity regularization, after which it is increased again, this may very well be due to semi-convergence. Semi-convergence is common in iterative methods since we do not have the exact data. The method may come very close to the exact solution (corresponding to data with no perturbation) and then diverge away from it again towards the biased solution from the regularization [24].

By looking at the objective functions $\Psi_S$ and $\Psi_{\text{TV}}$ in Figure 4.22 on page 81 we see an overall decay, which shows that the algorithm is in fact trying to minimize these functionals. There is also the occasional increase which is allowed due to the use of the weak monotonicity (3.17).

Finally looking at the step size at different iterations in Figure 4.23 on page 81, it is evident that it varies quite a lot, and for sparsity regulariza-



**(a)** Iteration 5.

**(b)** Iteration 10.

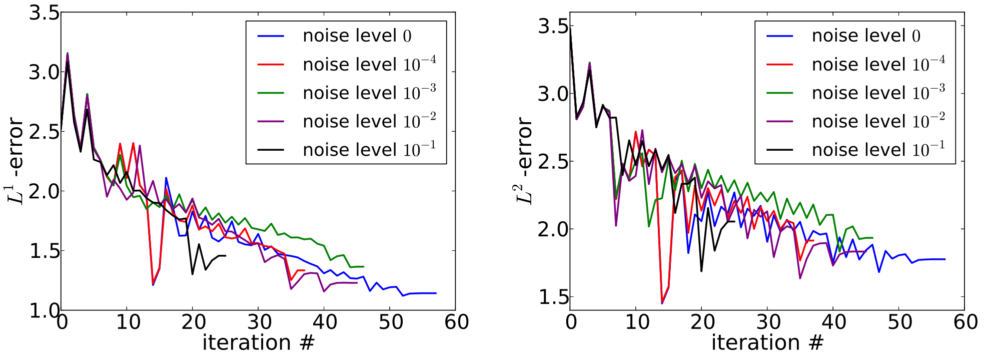
**(c)** Iteration 30.

**(d)** Final iteration.

**Figure 4.19:** TV reconstruction at different iterations, with noise level $10^{-3}$ and $\beta := 6 \cdot 10^{-4}$.

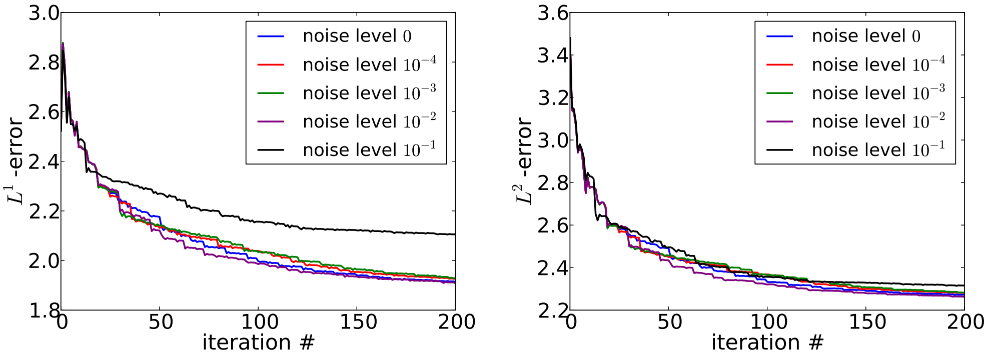
tion it suddenly drops to $s_{\text{stop}} = 10^{-3}$ the stopping criterion, when a local minimum has been found. All we can say about the step size for TV is that it varies quite a lot and that it is not reduced below the initial minimum of $s_{\min} = 1$.

From the above tests for different noise levels it may be too optimistic to get good results for the noise level $10^{-1}$ (at least for total variation), and the following tests will all be conducted using the noise level $10^{-3}$.



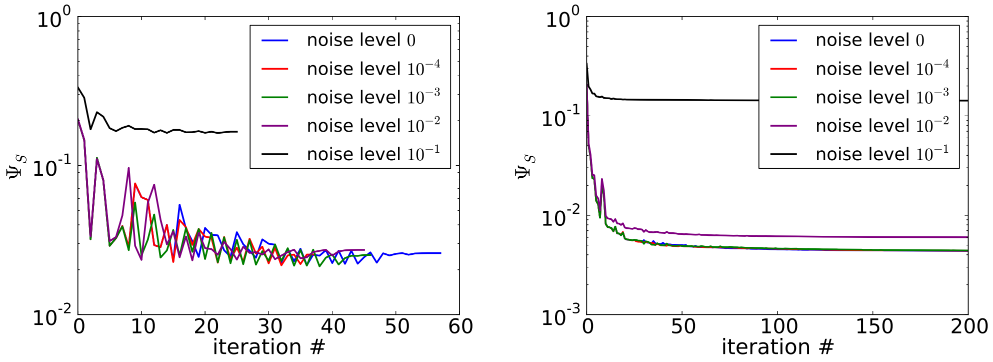
**(a)** $L^1$ error. **(b)** $L^2$ error.

**Figure 4.20:** Error for sparsity reconstruction at different noise levels with $\alpha := 2 \cdot 10^{-5}$.



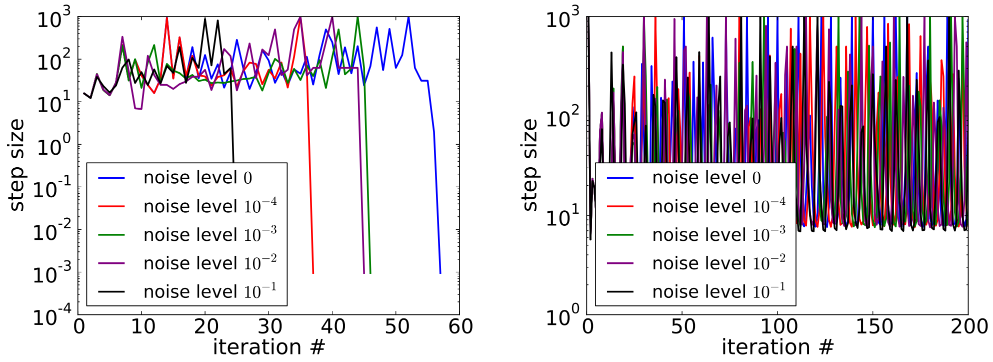
**(a)** $L^1$ error. **(b)** $L^2$ error.

**Figure 4.21:** Error for TV reconstruction at different noise levels with $\beta := 6 \cdot 10^{-4}$.



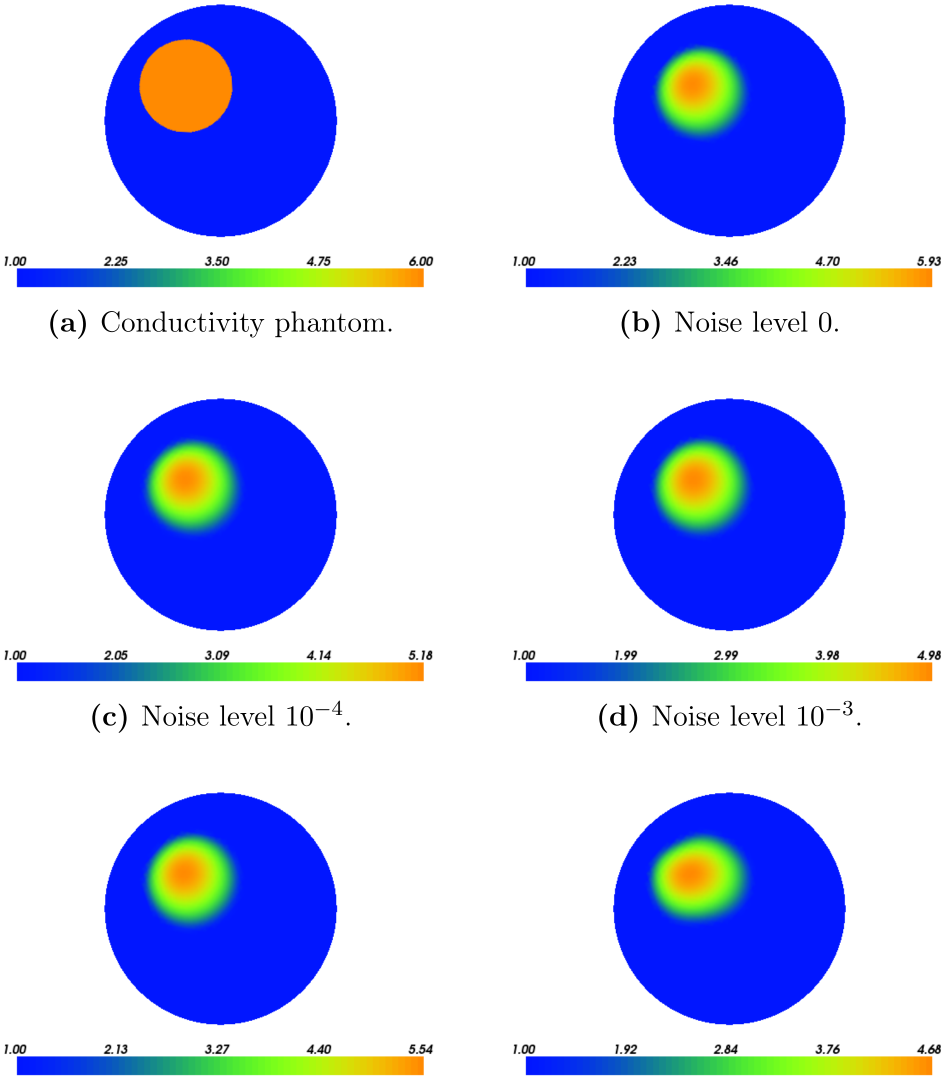
**(a)** Sparsity. **(b)** TV.

**Figure 4.22:** Objective functional for sparsity and TV reconstruction at different noise levels with parameters $\alpha := 2 \cdot 10^{-5}$ and $\beta := 6 \cdot 10^{-4}$.



**(a)** Sparsity. **(b)** TV.

**Figure 4.23:** Step size for sparsity and TV reconstruction at different noise levels with parameters $\alpha := 2 \cdot 10^{-5}$ and $\beta := 6 \cdot 10^{-4}$.



**(a)** Conductivity phantom.

**(b)** Noise level 0.

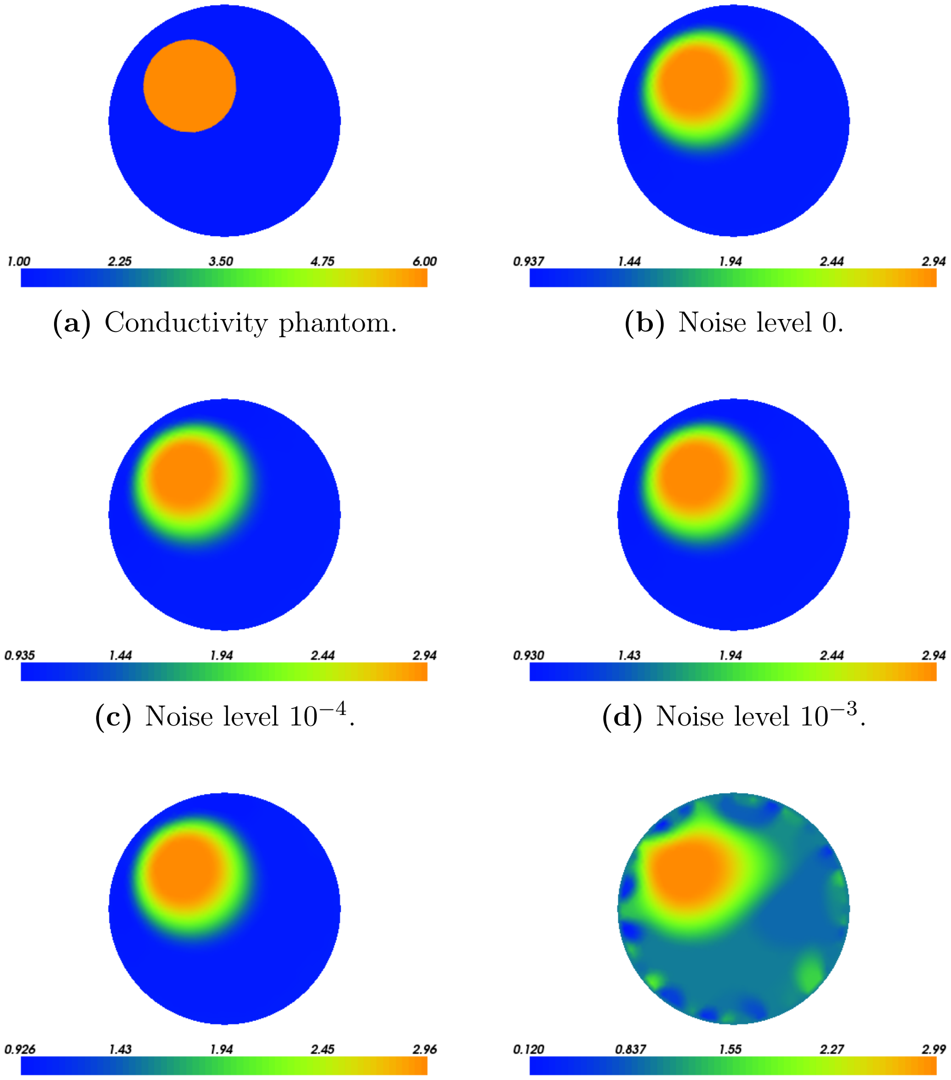
**(c)** Noise level $10^{-4}$.

**(d)** Noise level $10^{-3}$.

**(e)** Noise level $10^{-2}$.

**(f)** Noise level $10^{-1}$.

**Figure 4.24:** Sparsity reconstructions for different noise levels with regularization parameter $\alpha := 2 \cdot 10^{-5}$.



**(a)** Conductivity phantom.

**(b)** Noise level 0.

**(c)** Noise level $10^{-4}$.

**(d)** Noise level $10^{-3}$.

**(e)** Noise level $10^{-2}$.

**(f)** Noise level $10^{-1}$.

**Figure 4.25:** TV reconstructions for different noise levels with regularization parameter $\beta := 6 \cdot 10^{-4}$.

## 4.4 Reconstructions for Various Locations, Sizes, and Amplitudes

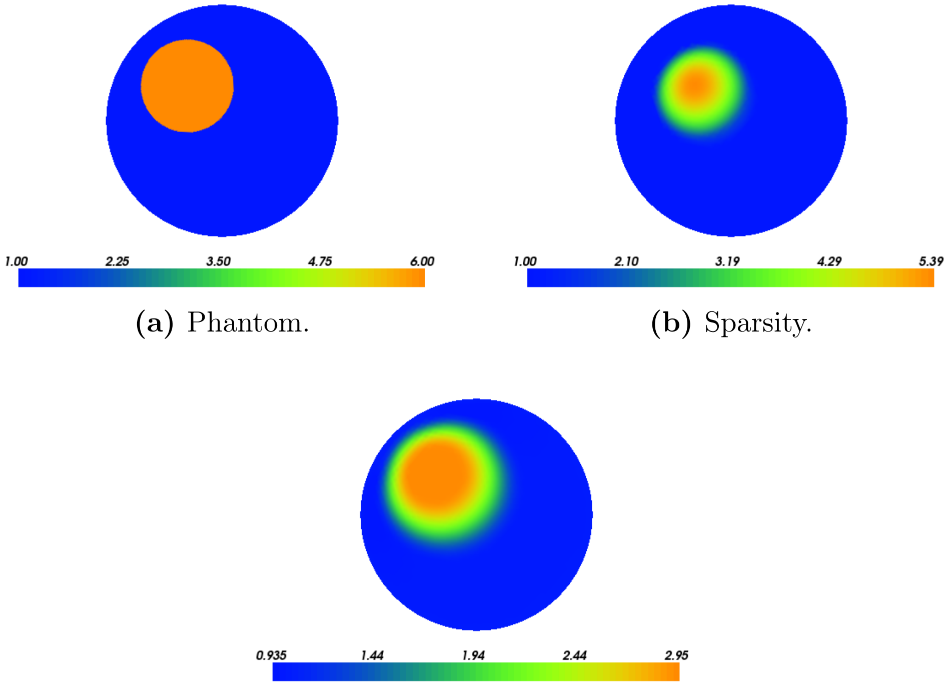
In this section, reconstructions based on simple inclusions are investigated in terms of the location, size, and amplitude of the inclusion $\delta\sigma$. The reason for looking into these cases is because these parameters for the inclusion have a lot to say on how well the inclusion is reconstructed in the solution. The inclusions here will be disks with the format $(c_x, c_y, C, r)$ where $(c_x, c_y)$ will be the centre, $C$ will be the amplitude of the inclusion $\delta\sigma$, and $r$ the radius.

### Location

Here reconstructions of an inclusion is investigated as it is moved from close to the boundary and towards the centre of the domain. Since the domain is the unit disk which is symmetric in the angular direction, it is only interesting to investigate the position with regard to the distance from the boundary.

From Figure 4.26 through 4.28 on pages 85–87 it is seen that the support of the inclusion is reconstructed in much the same way, no matter the distance from the boundary, and the circular shape of the inclusion is also well reconstructed in all three cases. The contrast, however, is much better reconstructed for the inclusions closer to the boundary, and this is quite intuitive, because the closer the inclusion is to the boundary the more significantly it can affect the boundary data.

In Figure 4.8a on page 69 we have already seen that the forward map has a smoothing effect, even on discontinuous conductivities, and this is partly what makes the inverse problem very ill-posed as inclusion are difficult to detect in the data.

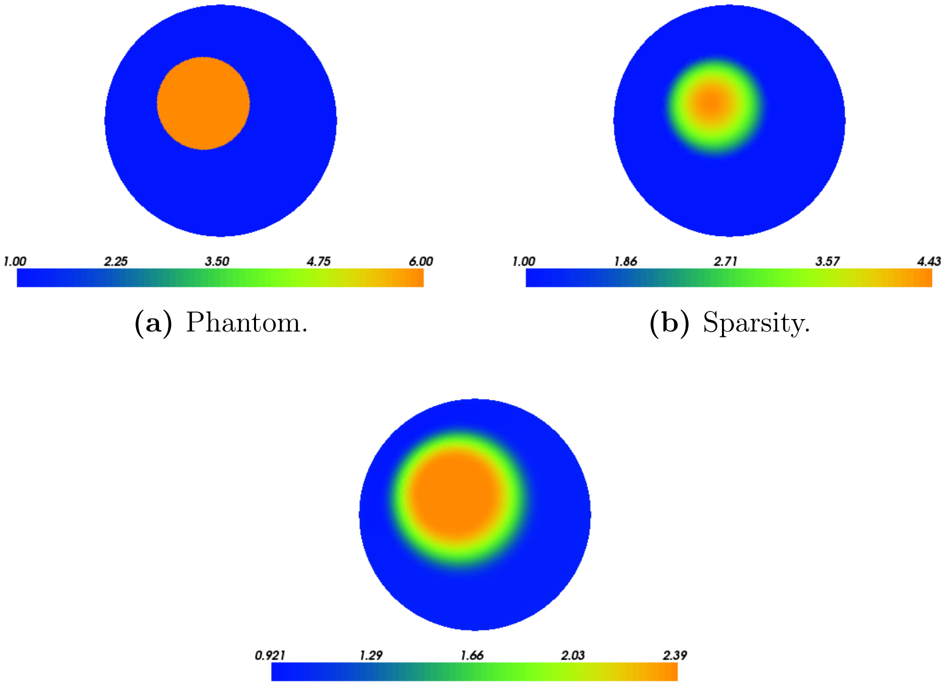


**(a)** Phantom.

**(b)** Sparsity.

**(c)** TV.

**Figure 4.26:** Sparsity and TV reconstruction of phantom $(c_x, c_y, C, R) = (-0.3, 0.3, 5, 0.4)$. $\alpha = 2 \cdot 10^{-5}$ and $\beta = 6 \cdot 10^{-4}$.

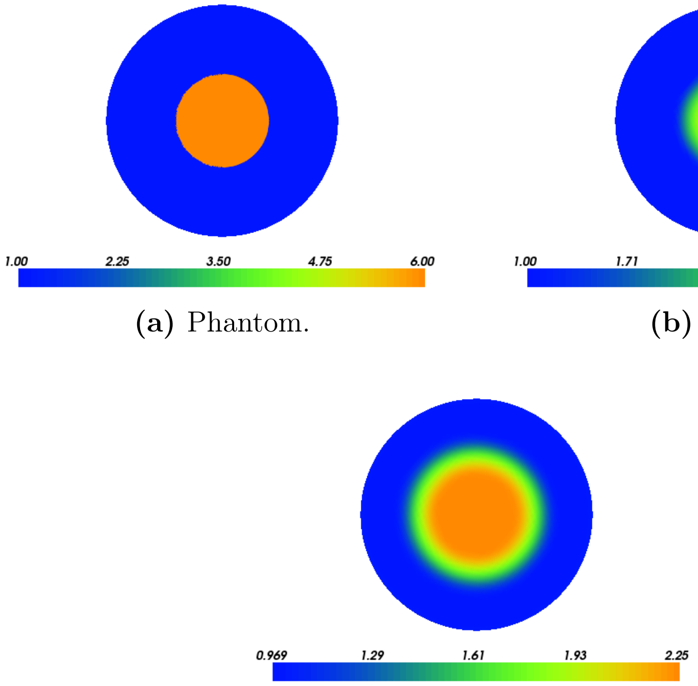


**(a)** Phantom.

**(b)** Sparsity.

**(c)** TV.

**Figure 4.27:** Sparsity and TV reconstruction of phantom $(c_x, c_y, C, R) = (-0.15, 0.15, 5, 0.4)$. $\alpha = 2 \cdot 10^{-5}$ and $\beta = 6 \cdot 10^{-4}$.



**(a)** Phantom.

**(b)** Sparsity.

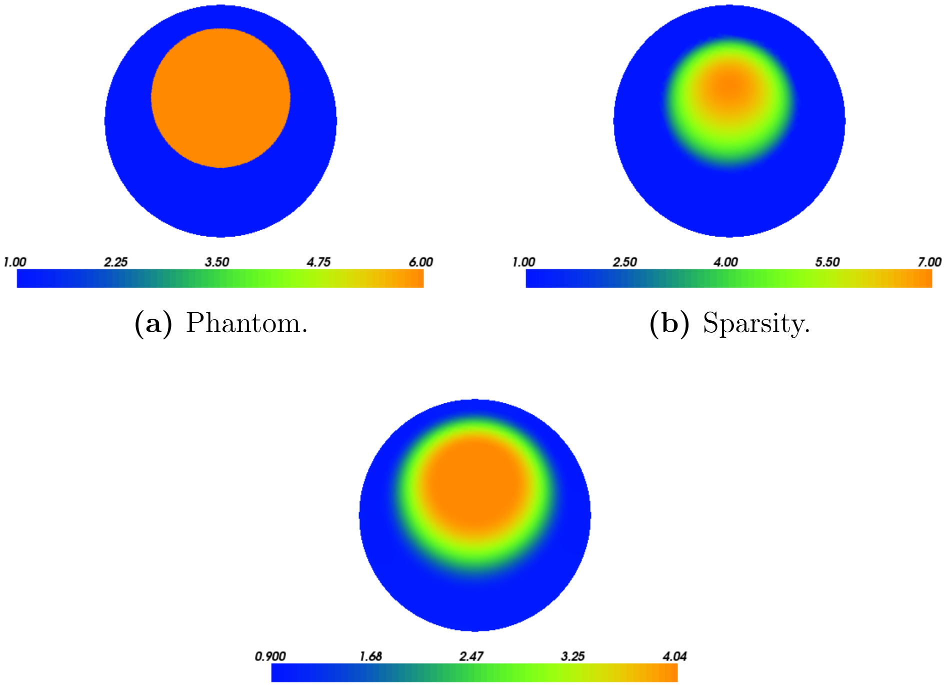
**(c)** TV.

**Figure 4.28:** Sparsity and TV reconstruction of phantom $(c_x, c_y, C, R) = (0, 0, 5, 0.4)$. $\alpha = 2 \cdot 10^{-5}$ and $\beta = 6 \cdot 10^{-4}$.

## Size

Here reconstructions are performed for inclusions with the same amplitude but varying size of the support. We saw in the test above that the distance of the inclusion to the boundary has an effect on the contrast in the reconstruction, therefore all the inclusions here are constructed such that they have the same distance to the boundary.

In Figure 4.29 through 4.31 on pages 88–90 it is seen that the shape of the inclusion and its support are reconstructed well, independent of the size of the inclusion. From Figure 4.29 the contrast is reconstructed better than in Figure 4.30 and Figure 4.31, and for TV regularization there is also a significant improvement in Figure 4.30c over Figure 4.31c.



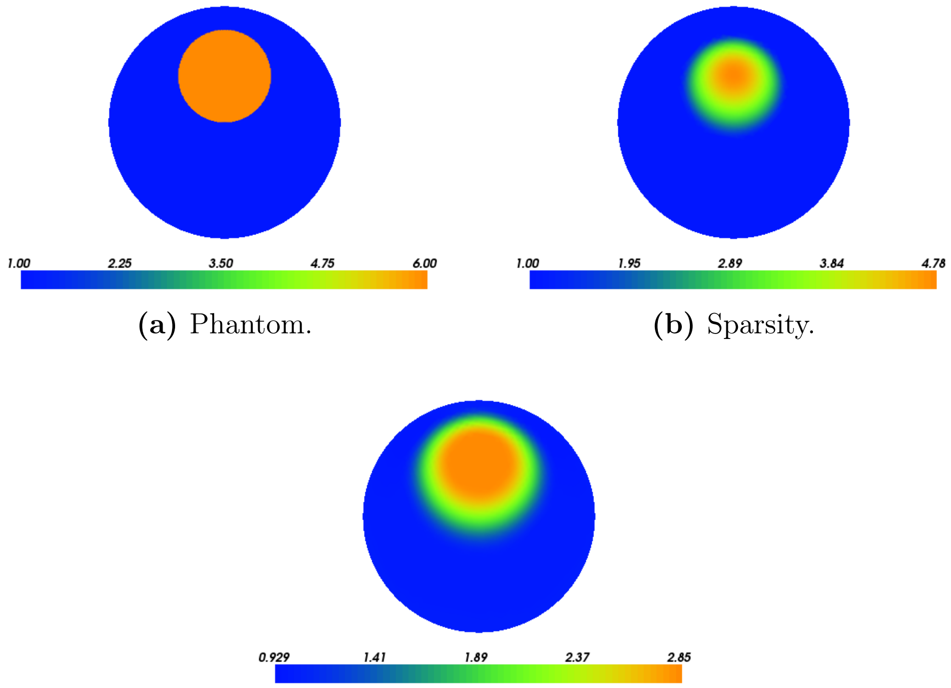
**(a)** Phantom.

**(b)** Sparsity.

**(c)** TV.

**Figure 4.29:** Sparsity and TV reconstruction of phantom $(c_x, c_y, C, R) = (0, 0.2, 5, 0.6)$. $\alpha = 2 \cdot 10^{-5}$ and $\beta = 6 \cdot 10^{-4}$.

In Figure 4.29b there is even an overestimation of the contrast in the middle of the reconstructed inclusion, and a more reasonable contrast for the rest of the inclusion. Such overestimations are quite common in regularization



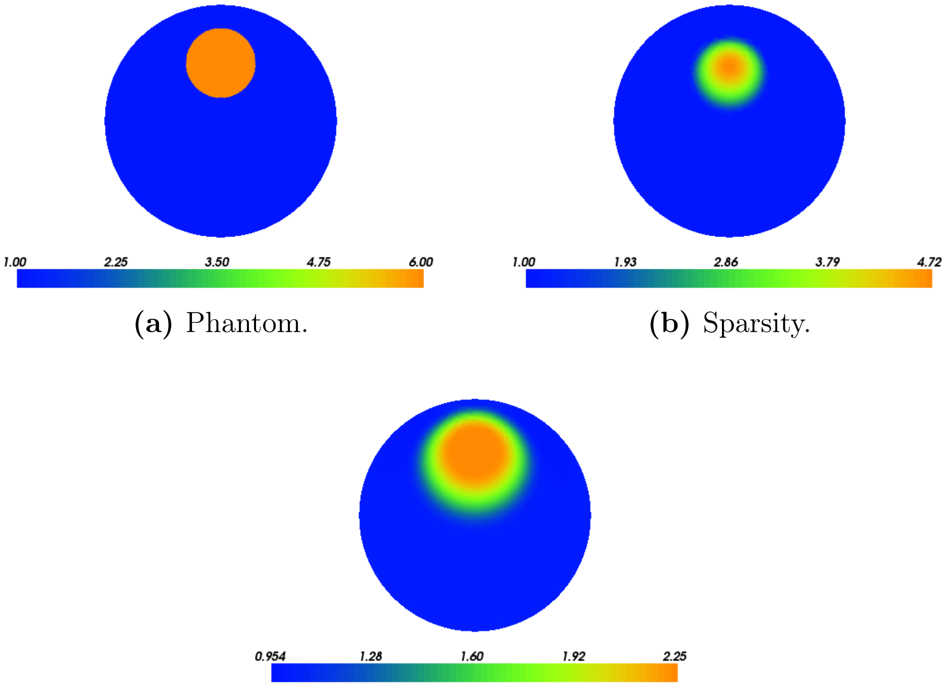
**(a)** Phantom.

**(b)** Sparsity.

**(c)** TV.

**Figure 4.30:** Sparsity and TV reconstruction of phantom $(c_x, c_y, C, R) = (0, 0.4, 5, 0.4)$. $\alpha = 2 \cdot 10^{-5}$ and $\beta = 6 \cdot 10^{-4}$.

techniques (at least for easier and often linear problems) when attempting to approximate a discontinuous solution with a continuous reconstruction, and it is similar to the Gibbs phenomenon that occurs in Fourier series [18]. This explains why it is easier to reconstruct inclusions with larger support, as most of the error occurs near the boundary of the inclusion, and a larger support allows the method to better approximate the interior of the inclusion.



**(a)** Phantom.

**(b)** Sparsity.

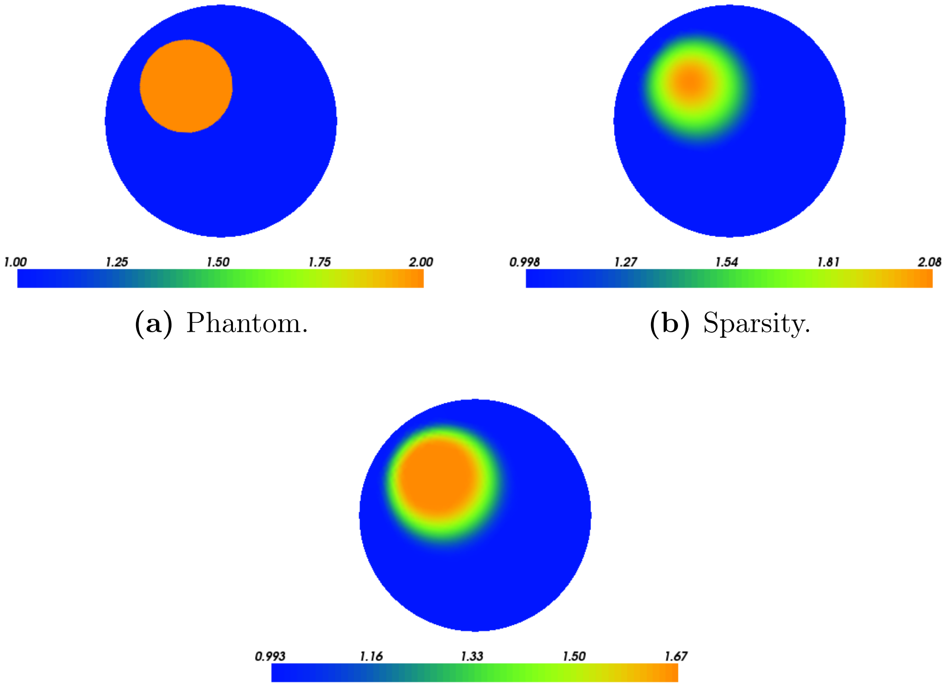
**(c)** TV.

**Figure 4.31:** Sparsity and TV reconstruction of phantom $(c_x, c_y, C, R) = (0, 0.5, 5, 0.3)$. $\alpha = 2 \cdot 10^{-5}$ and $\beta = 6 \cdot 10^{-4}$.

## Contrast

Now varying the contrast of an inclusions is tested, where the position and size is fixed. In Figure 4.32 through 4.34 on pages 91–93 we see the very intuitive results, namely, that a smaller amplitude of $\delta\sigma$ is easier to reconstruct. In Figure 4.34 we even see that the reconstructions do not come anywhere near the correct contrast, however, the contrast is also unreasonable large compared to the size of the inclusion, and the case in Figure 4.33 is already a very difficult case, where the sparsity regularization actually reconstructs the contrast quite well. In general TV regularization does not reconstruct the contrast very well, even in the case in Figure 4.32.

Again the support and circular shape of the inclusion are reconstructed reasonably well, and as a conclusion to this section it appears that the support and shape of the reconstruction are quite robust to the location, size, or amplitude of the inclusion. However, the contrast of the reconstruction appears to be affected in all three cases, and is by far the most difficult



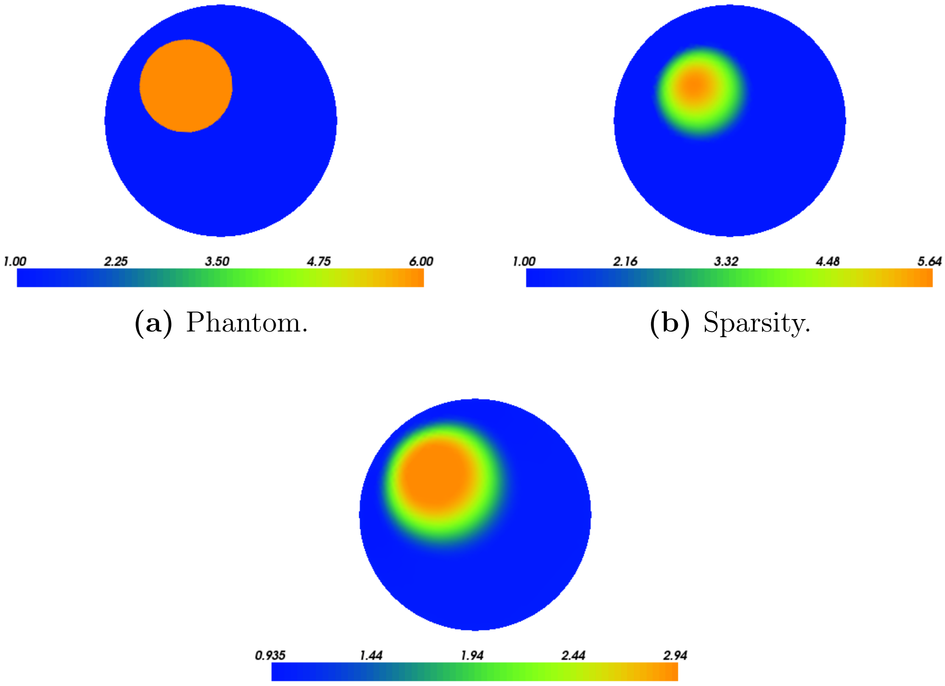
**(a)** Phantom.

**(b)** Sparsity.

**(c)** TV.

**Figure 4.32:** Sparsity and TV reconstruction of phantom $(c_x, c_y, C, R) = (-0.3, 0.3, 1, 0.4)$. $\alpha = 5 \cdot 10^{-7}$ and $\beta = 6 \cdot 10^{-4}$.

property to determine correctly.

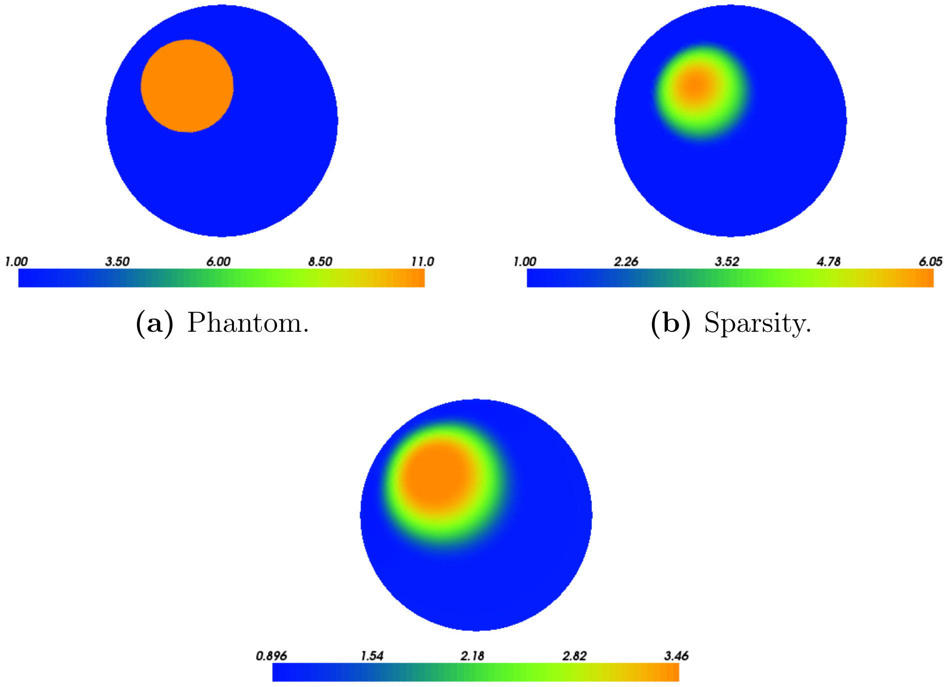


**(a)** Phantom.

**(b)** Sparsity.

**(c)** TV.

**Figure 4.33:** Sparsity and TV reconstruction of phantom $(c_x, c_y, C, R) = (-0.3, 0.3, 5, 0.4)$. $\alpha = 2 \cdot 10^{-5}$ and $\beta = 6 \cdot 10^{-4}$.



**(a)** Phantom.

**(b)** Sparsity.

**(c)** TV.

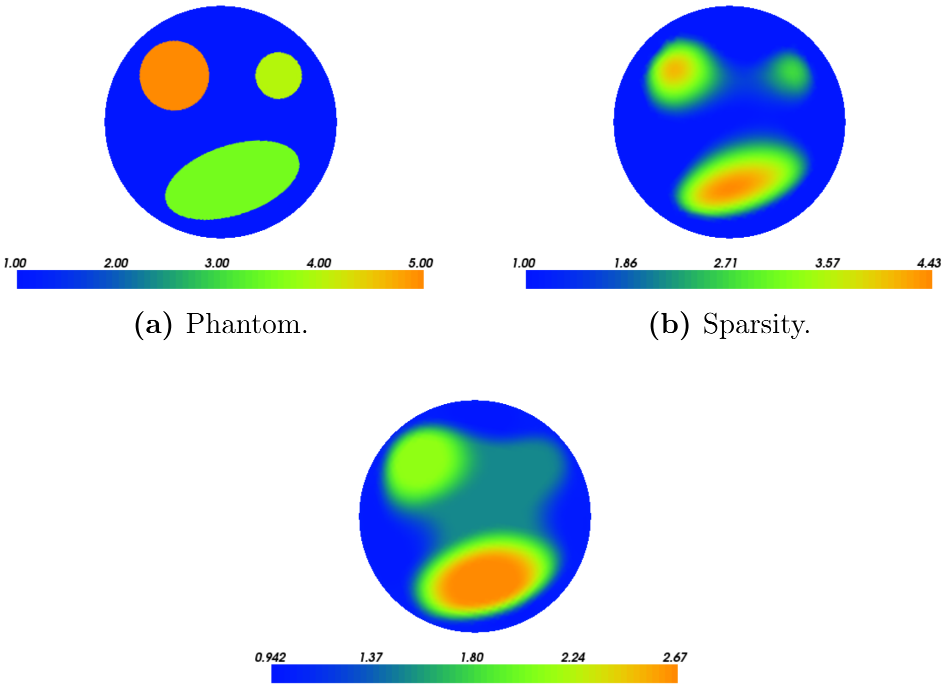
**Figure 4.34:** Sparsity and TV reconstruction of phantom $(c_x, c_y, C, R) = (-0.3, 0.3, 10, 0.4)$. $\alpha = 2 \cdot 10^{-5}$ and $\beta = 6 \cdot 10^{-4}$.

## 4.5 Reconstructions of Difficult Inclusions

In this section we will look at more difficult types of inclusions, which will consist of multiple inclusions in different shapes and with different amplitude to see if the method is able to separate the inclusions. There is also an example of a ring-type inclusion which is donut-shaped, as holes in the inclusions can be very difficult to reconstruct [9].

### Multiple Inclusions

The first case of multiple inclusions is depicted in Figure 4.35a with two circular inclusion and an ellipse. The small inclusion has an amplitude of 3, the larger circular inclusion has an amplitude of 4, and the ellipse an amplitude of 2.5, and as usual the background conductivity is $\sigma_0 = 1$.



**(a)** Phantom.

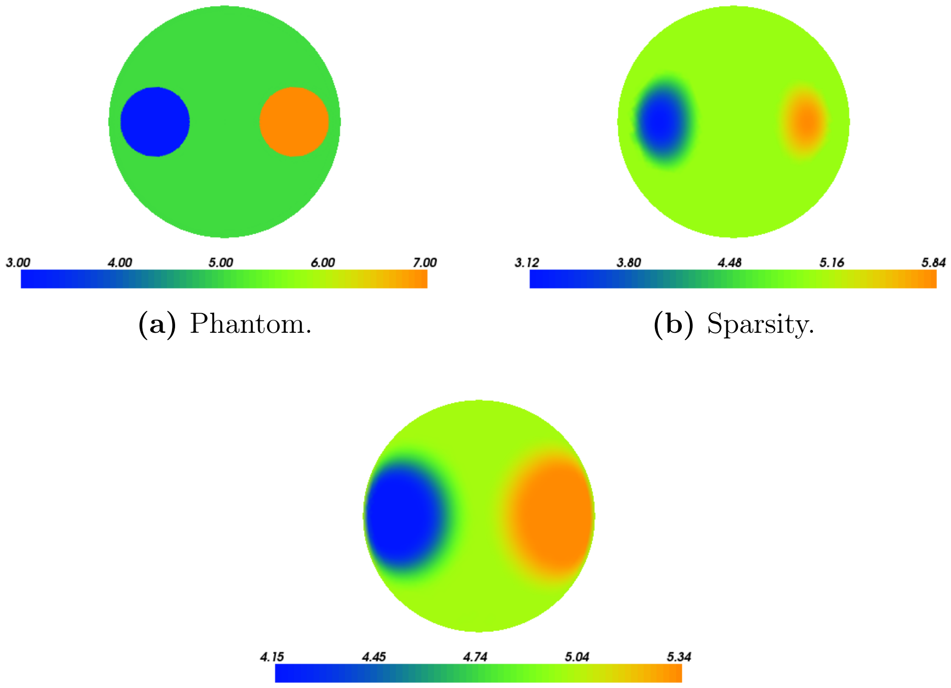
**(b)** Sparsity.

**(c)** TV.

**Figure 4.35:** Sparsity and TV reconstruction of multiple inclusions. $\alpha = 10^{-6}$ and $\beta = 6 \cdot 10^{-4}$.

From Figure 4.35 it is seen that both sparsity and TV regularization have great difficulty in determining the correct contrast, and they both want the ellipse inclusion to have the largest amplitude most likely due to it being the inclusion with the largest support. It is also evident that TV regularization has difficulty in separating the inclusions, which is not the case for sparsity where there is a clear separation. It should also be noted that it was necessary to reduce the parameter $\alpha$ to $10^{-6}$ to get a proper result for sparsity regularization.

In Figure 4.35 we saw multiple inclusions, however, they are all positive inclusions so it may also be interesting to see what happens if there is a sign change. In Figure 4.36a we have two circular inclusions, the one on the left has a contrast of $-2$ while the one on the right has a contrast of 2. Since we can not allow the conductivity to be negative, a background conductivity of $\sigma_0 = 5$ has been used.



**(a)** Phantom.

**(b)** Sparsity.

**(c)** TV.

**Figure 4.36:** Sparsity and TV reconstruction of multiple inclusions. $\alpha = 10^{-7}$ and $\beta = 10^{-4}$.

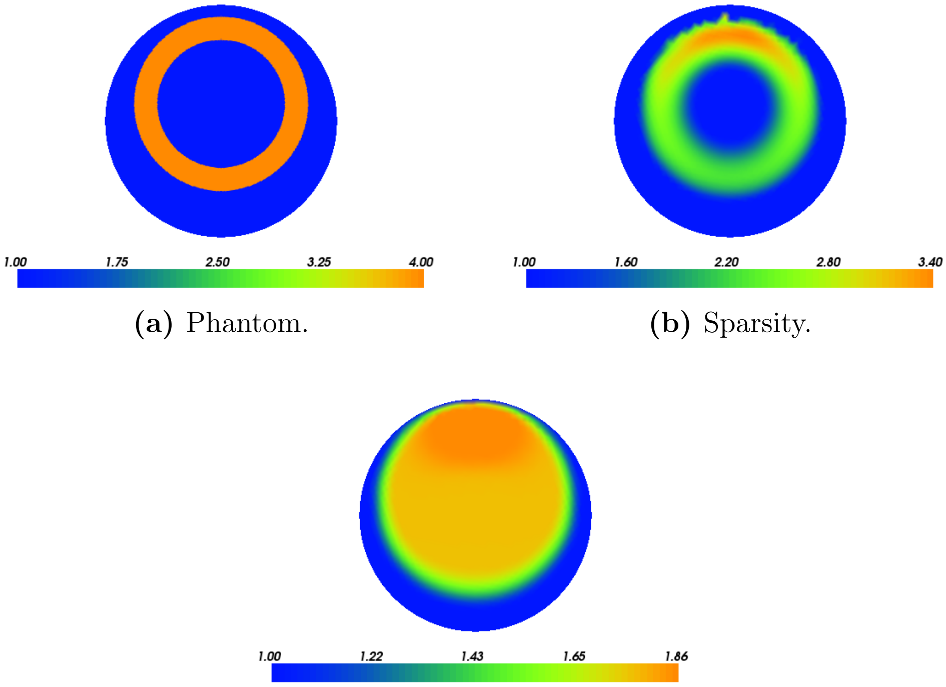
In Figure 4.36 it is evident that both methods determines the sign of the inclusions correctly, and both methods separates the inclusions, though TV regularization leads to a much too large support. The most interesting observation is probably that both methods favour the negative inclusion in terms of contrast. Sparsity regularization almost reconstructs the contrast for the negative inclusion perfectly while it hardly reconstructs the contrast of the positive inclusion. It should also be noted that both of the regularization parameters had to be reduced significantly to give proper results, and it seems that both methods have difficulty in dealing with multiple inclusions when these inclusions have different amplitudes.

In Jin *et al.* [9] they also looked at inclusions with different signs, however, they used a much smaller amplitude for the negative inclusion (only 0.5), in order to also be able to reconstruct a small positive inclusion. There is no immediate indication from the theory for the method of why the negative inclusion should be favoured, but we also saw in Figure 4.35 on page 94 that some inclusions are favoured over the others for no apparent reason. While beyond the scope of this thesis, it is definitely a subject that is worth investigating further in the future.

## Ring-Type Inclusion

The following test was conducted because a similar test was found in Jin *et al.* [9], where it was quite evident that ring-shaped (donut-shaped) conductivities are very difficult to reconstruct, mainly because of the hole in the inclusion.

In Figure 4.37 on the facing page we see sparsity and TV reconstructions of such a ring inclusion, and the most glaring observation is probably that TV regularization is not able to reconstruct the hole at all, but acts as if the inclusion is solid, and changing the regularization parameter did not improve the result. The same was actually the case with sparsity regularization for the usual $\alpha = 2 \cdot 10^{-5}$, but reducing the parameter to $\alpha = 10^{-6}$ it actually reconstructs the hole, and in general it reconstructs the ring-shape quite well. In Jin *et al.* [9] the reconstruction had the hole but the shape was not quite as circular as in Figure 4.37b on the next page. In both cases we see that the contrast is best reconstructed near the boundary, which has already been discussed in the previous tests.



**(a)** Phantom.

**(b)** Sparsity.

**(c)** TV.

**Figure 4.37:** Sparsity and TV reconstruction of ring-type inclusion. $\alpha = 10^{-6}$ and $\beta = 6 \cdot 10^{-4}$.

## 4.6 Reconstructions of Smooth Inclusions

Until now we have only investigated reconstructions of phantoms that are piecewise constant, and we have seen that in general the reconstructions have been quite smooth. Therefore it could be interesting to see if a smoother phantom will lead to better reconstructions. Most regularization techniques are better at reconstructing smooth solutions, and has difficulty in reconstructing discontinuities, however, both sparsity and TV regularization does not favour smooth solutions over discontinuous solutions and in particular TV regularization actually prefers discontinuous solutions over general smooth solutions. So intuitively we should actually expect that sparsity and TV regularization perform worse for smooth solutions.

The different inclusions in the previous tests consists of disks, ellipses, and a ring, all of which are determined by a centre and a parameter $r$ which is the radius for the circular inclusions, similarly the ellipse is just a circle that has been scaled in one or both directions. So the usual piecewise constant inclusions can be converted into smoother inclusions by a smooth transition from some amplitude $C$ down to 0 that depends on $r$, i.e. the distance from the centre of the inclusion. Thus we can determine a polynomial $Q$ in $r$ such that $Q(a) = C$ for the amplitude $C$ of the inclusion and $Q(b) = 0$ to make the transition in the range $r \in [a, b]$. Since we know that the inclusions used are constant outside the transition, their derivatives should be zero, i.e. $Q^{(q)}(a) = Q^{(q)}(b) = 0$ for $q = 1, \dots, D$ for $D$ continuous derivatives,

$$\begin{aligned} Q(a) &= C, \\ Q(b) &= 0, \\ Q^{(q)}(a) &= 0, \quad q = 1, \dots, D, \\ Q^{(q)}(b) &= 0, \quad q = 1, \dots, D. \end{aligned} \tag{4.5}$$

This leads to $2D + 2$ constraints which means that $2D + 2$ coefficients or a $2D+1$ order polynomial can be determined uniquely via the linear system of equations (4.5). In Figure 4.38 I have shown such a polynomial for $D = 5$.

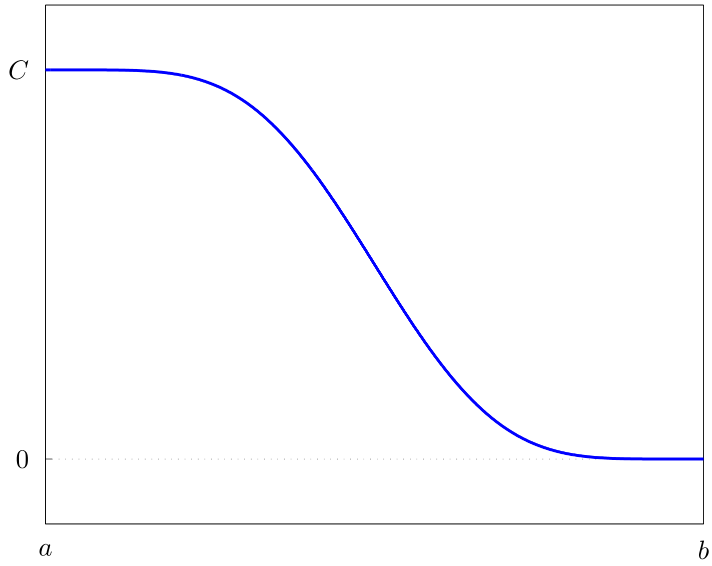


**Figure 4.38:** 11th order polynomial solving (4.5) for $D = 5$.

In Figure 4.40 on page 100 the usual non-smoothed phantoms can be seen along with their smoothed version, also using $D = 5$ for the transitions. Because the transitions increase the support, I have taken the liberty to decrease the size of the smoothed phantoms slightly.

The actual smoothing may be a little easier to see in 3D, so an example with the multiple inclusion is shown in Figure 4.39.

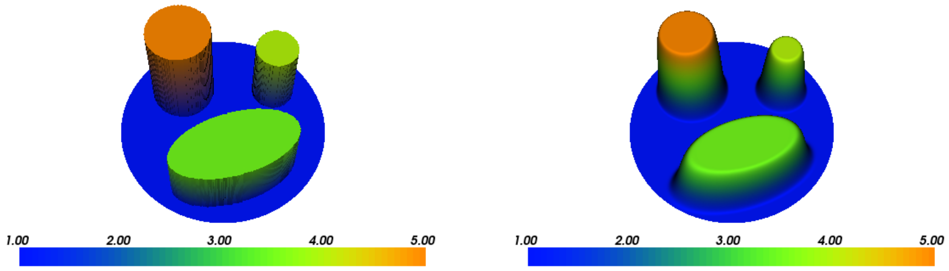


**Figure 4.39:** Discontinuous and smoothed versions of phantom with multiple inclusions.

Now reconstructions are done based on the smooth phantoms in Figure 4.40 on the next page, and the results can be seen in Figure 4.41 on page 101, where in general the reconstructions are very similar to those of the non-smoothed phantoms (see Figure 4.33 on page 92, Figure 4.35 on page 94, and Figure 4.37 on page 97 for the non-smoothed reconstructions). As explained in the beginning of this section it is not surprising to see that the methods perform slightly worse for smooth solutions. Especially the reconstruction of the ring-inclusion is much worse, and the hole in the sparsity reconstruction in Figure 4.41e is barely visibly (even after fine tuning the regularization parameter).

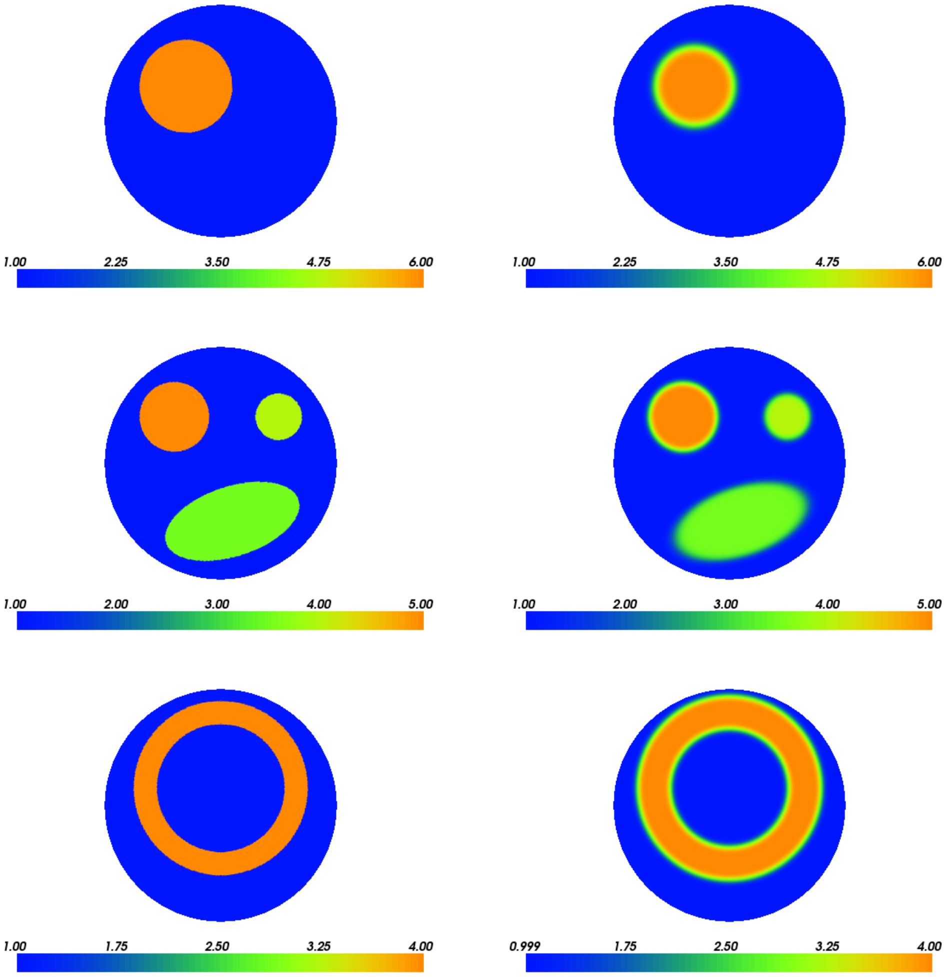
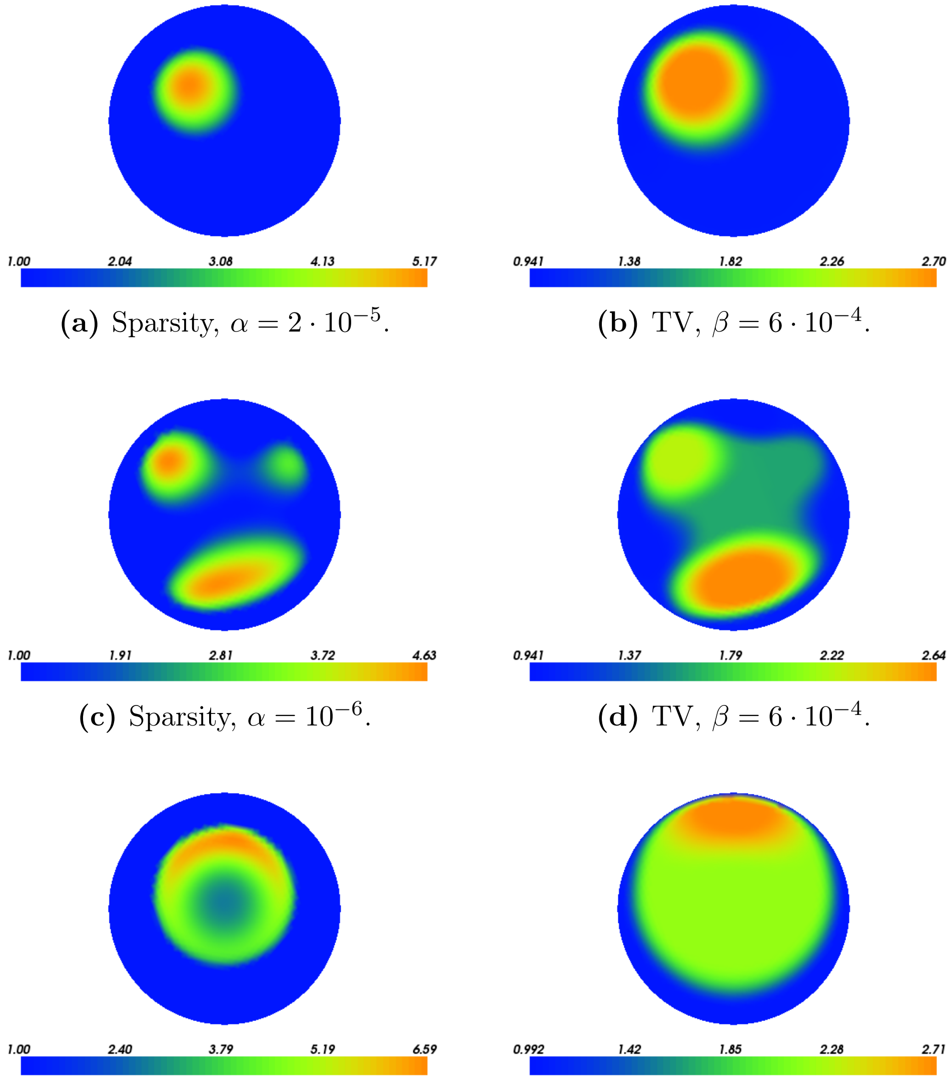


**Figure 4.40:** Discontinuous and smoothed versions of phantoms.



**(a)** Sparsity, $\alpha = 2 \cdot 10^{-5}$.

**(b)** TV, $\beta = 6 \cdot 10^{-4}$.

**(c)** Sparsity, $\alpha = 10^{-6}$.

**(d)** TV, $\beta = 6 \cdot 10^{-4}$.

**(e)** Sparsity, $\alpha = 3.5 \cdot 10^{-5}$.

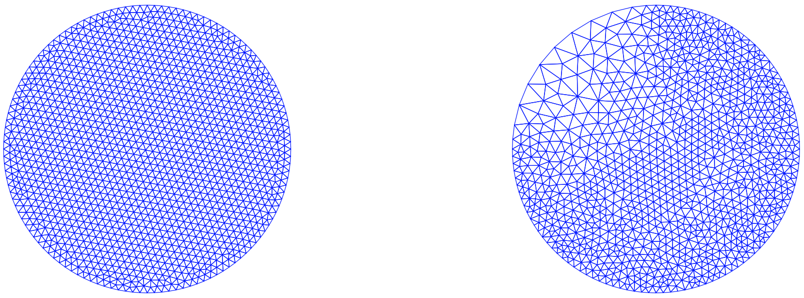
**(f)** TV, $\beta = 6 \cdot 10^{-4}$.

**Figure 4.41:** Sparsity and TV reconstructions for smoothed phantoms in Figure 4.40 on the preceding page.

## 4.7 Reconstructions on a Non-Uniform Mesh

In the previous tests a very uniform mesh has been used, so now we shall see how well sparsity and TV regularization performs on a very non-uniform mesh. The reason for doing this, is because it may be useful to know if you can modify the method such that you start out with a very coarse mesh, and as you approximate your inclusion you refine your mesh near the inclusion, such that you have a fine mesh at the inclusion and a coarse mesh outside the support of the inclusion. It is an idea to reduce the computational workload by only focusing on the area with the inclusion.

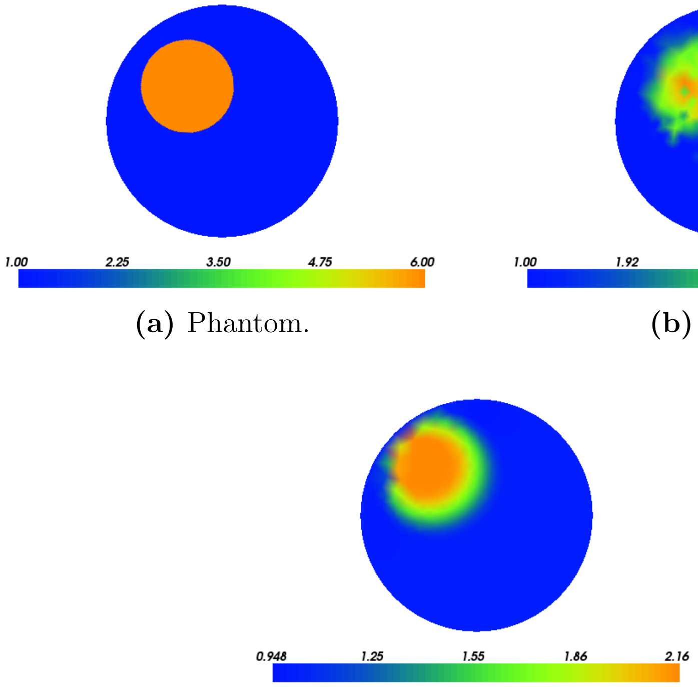
So first off, we need a non-uniform mesh, and I have constructed such a mesh in Gmsh [32] as can be seen in Figure 4.42, where the mesh is very coarse in the top left part of the domain while it is more like the usual uniform mesh towards the bottom right part of the domain. It is also seen that the boundary of the mesh is non-uniformly distributed, which means that this will give only few nodal points where the data is given in the top left.



**(a)** Uniform mesh. **(b)** Non-uniform mesh.

**Figure 4.42:** The usual uniform mesh along with the non-uniform mesh.

The reconstructions on the non-uniform mesh can be seen in Figure 4.43 through 4.45 on pages 103–105, and it is evident that TV regularization performs very well on the non-uniform mesh, and of course slightly better in Figure 4.45c where the mesh is finer. However, the reconstruction in Figure 4.43c is actually quite good considering how coarse the mesh is. The sparsity regularization is not performing very well, no matter where the inclusion is positioned, and the solution is in general quite grainy. The graininess in the sparsity solution is due to the different sizes of the triangles, as the penalty term leads to a normalization from the FEM-basis functions in Theorem 4.1, so the different FEM-basis functions will be penalized differently depending on the size of the elements.



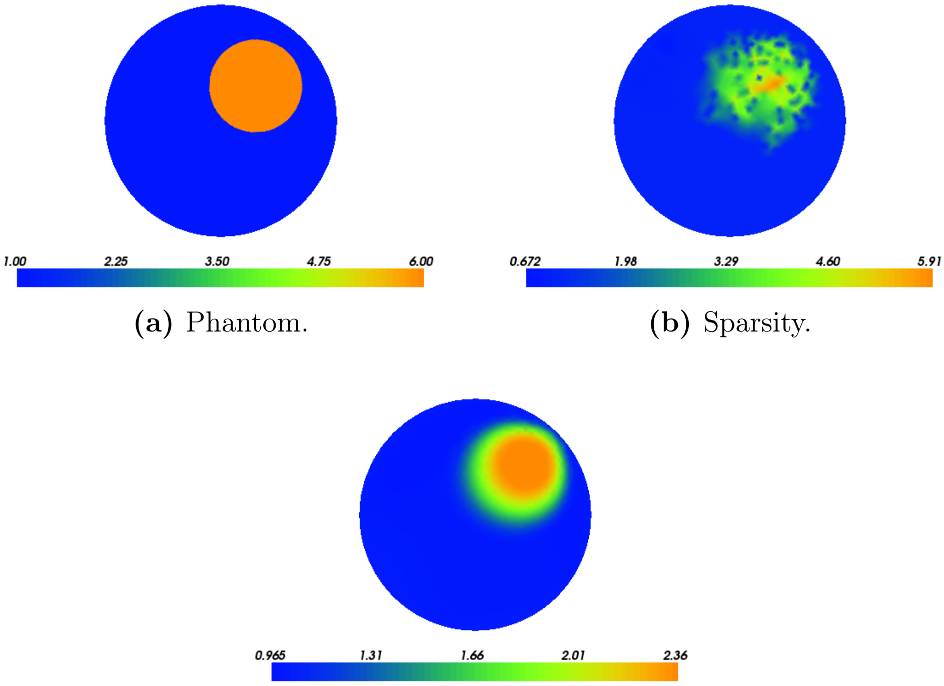
**(a)** Phantom.

**(b)** Sparsity.

**(c)** TV.

**Figure 4.43:** Sparsity and TV reconstruction on non-uniform mesh. $\alpha = 10^{-6}$ and $\beta = 6 \cdot 10^{-4}$.

So for further research into these methods, a method that adaptively refines the mesh near the inclusion should probably not be applied to sparsity regularization, unless one finds a way to straighten out the solution on such a mesh. TV regularization, however, seems very well fitted for adaptively changing the mesh, and since the TV-regularization tends to use far more iterations than sparsity regularization, it may be a good idea in order to speed up the computations.

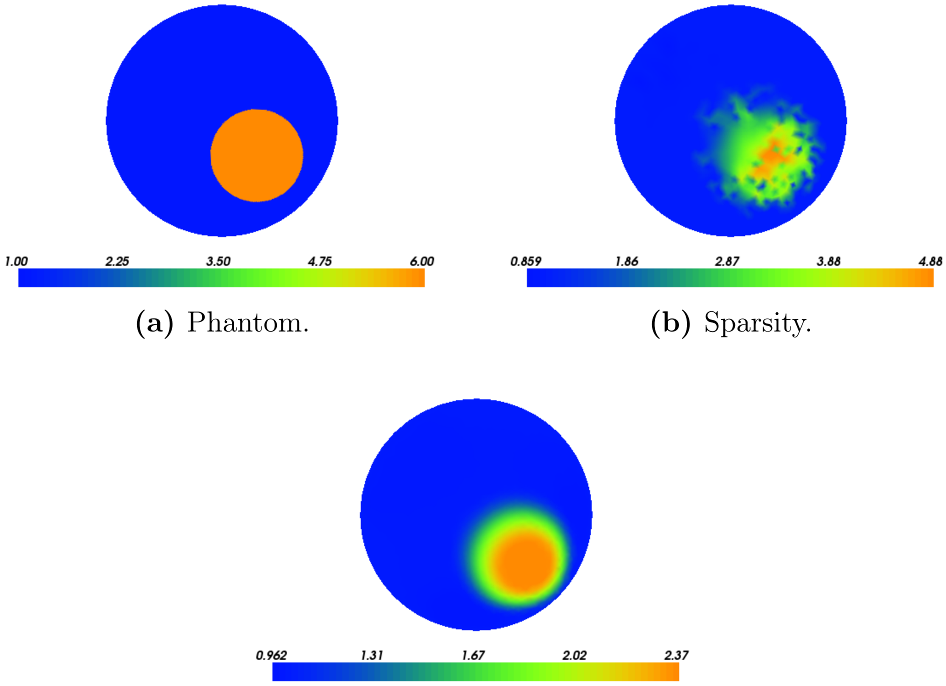


**(a)** Phantom.

**(b)** Sparsity.

**(c)** TV.

**Figure 4.44:** Sparsity and TV reconstruction on non-uniform mesh. $\alpha = 10^{-6}$ and $\beta = 6 \cdot 10^{-4}$.



**(a)** Phantom.

**(b)** Sparsity.

**(c)** TV.

**Figure 4.45:** Sparsity and TV reconstruction on non-uniform mesh. $\alpha = 10^{-6}$ and $\beta = 6 \cdot 10^{-4}$.

## 4.8 Reconstructions Based on Partial/Incomplete Data

Now we shall investigate the use of partial data, which means that we are only able to measure data on part of the boundary $\Gamma \subset \partial\Omega$. This is very relevant in practice, for instance in 3D medical imaging, since it is not very practical to cover the entire patient with electrodes. Instead the electrodes are for instance placed on the chest (i.e. for imaging the lungs), which means that there is no data available in the cross-section of the body.

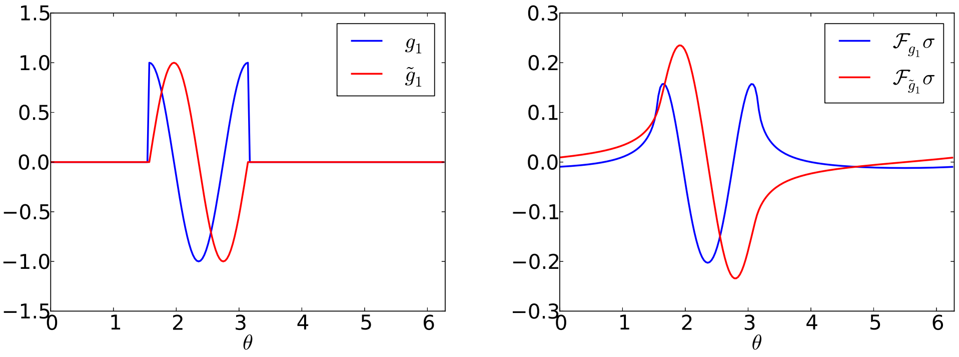
It was shown in (3.20) how the method is implemented to solve problems using partial data, so it is assumed that $\Gamma$ is known and that the Neumann-data has support on $\Gamma$. The implementation can make use of any $\Gamma \subset \partial\Omega$. As $\Omega$ is the unit disk, then for the sake of this test we can simplify the problem by letting $\Gamma$ be determined by the circular arc with polar angle $\theta \in [\theta_1, \theta_2]$, $0 < \theta_1 < \theta_2 < 2\pi$. The corresponding Neumann-data that will be used, are scaled and translated versions of (4.3) to ensure that the Neumann-data have support in $[\theta_1, \theta_2]$ while maintaining $N$ oscillations:

$$\begin{aligned} g_N &:= \chi_{[\theta_1,\theta_2]}(\theta) \cos\left(\frac{2\pi}{\theta_2 - \theta_1} N(\theta - \theta_1)\right), \\ \tilde{g}_N &:= \chi_{[\theta_1,\theta_2]}(\theta) \sin\left(\frac{2\pi}{\theta_2 - \theta_1} N(\theta - \theta_1)\right). \end{aligned} \tag{4.6}$$

Where the term $\chi_{[\theta_1,\theta_2]}(\theta)$ is a characteristic function on $[\theta_1, \theta_2]$. By the definition in (4.6) then it becomes equal to (4.3) for $\theta_1 = 0$ and $\theta_2 = 2\pi$. It also satisfies the condition that $\int_{\partial\Omega} g ds = 0$.

From Figure 4.46 on the next page it is evident that even though the Neumann-data is supported on $[\theta_1, \theta_2]$, this is generally not the case for the Dirichlet-data $\phi$. Therefore there will be two different cases in the following tests, namely, using the complete data $\phi$ and the truncated data $\hat{\phi} := \chi_{[\theta_1,\theta_2]}\phi$ which is the more realistic case, i.e. that we only measure the Dirichlet-data on $[\theta_1, \theta_2]$ where we know the Neumann-data.

There will be two types of tests in this section, one looking at reconstructions with varying size of the interval $[\theta_1, \theta_2]$ determining $\Gamma$, and another where $\Gamma$ is fixed and the location of the inclusion varies.



**(a)** $g_1$ and $\tilde{g}_1$.

**(b)** Corresponding $\phi$.

**Figure 4.46:** Neumann-data $g_1$ and $\tilde{g}_1$ along with corresponding Dirichlet data (no perturbations), with $[\theta_1, \theta_2] = [\frac{\pi}{2}, \pi]$. The phantom used is from Figure 4.47.

## Size of $\Gamma$

Here the size of the interval $[\theta_1, \theta_2]$ that determines $\Gamma$ will be varied, and we shall try to reconstruct the inclusion in Figure 4.47. In Figure 4.48 on the next page is an illustration of the three cases of $\Gamma$ considered here.

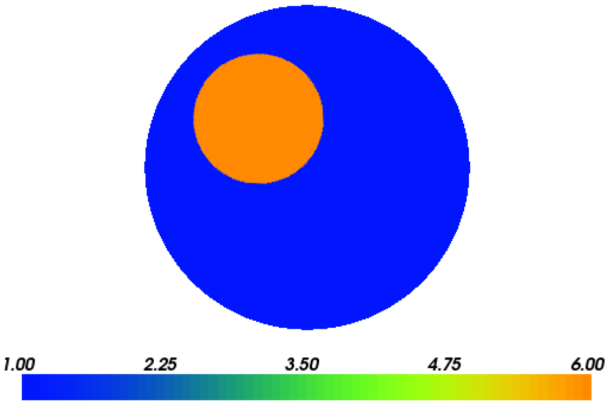
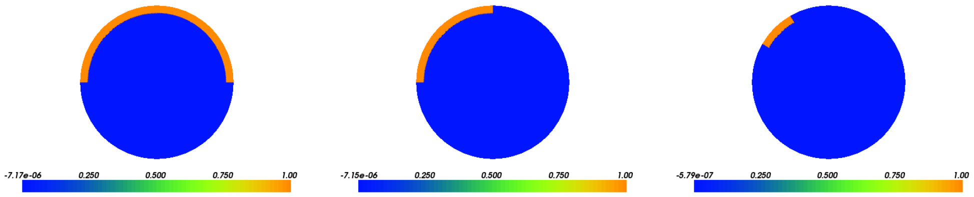


**Figure 4.47:** Phantom.

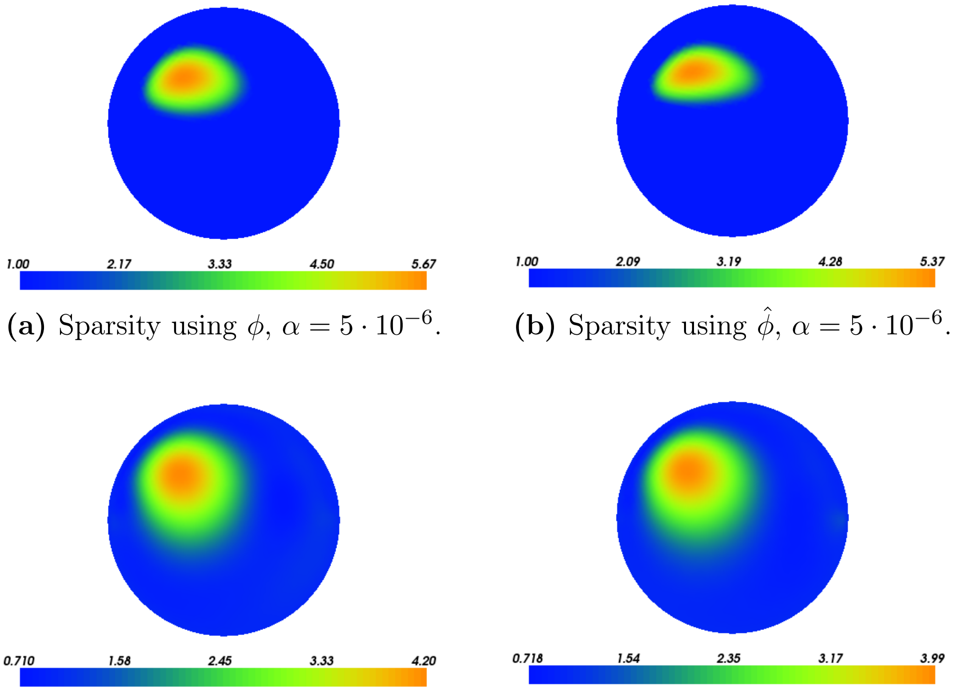
In Figure 4.49 on the following page the results are found using $\Gamma$ in Figure 4.48a on the next page. It is evident that the shape of the inclusion is not reconstructed as well, as when (4.3) are used for the Neumann-data, and not surprisingly the reconstruction using the complete data $\phi$ yields



**(a)** $[\theta_1, \theta_2] = [0, \pi]$. **(b)** $[\theta_1, \theta_2] = [\frac{\pi}{2}, \pi]$. **(c)** $[\theta_1, \theta_2] = [\frac{2\pi}{3}, \frac{5\pi}{6}]$.

**Figure 4.48:** Illustration of the location of $\Gamma$ in three cases.

better reconstructions than the incomplete data $\hat{\phi}$ in terms of the contrast. But with $\Gamma$ being half of $\partial\Omega$ the results do not differ that much whether full or partial Dirichlet-data are used. The most significant change is that the shapes of the inclusions are not reconstructed as well when only partial Neumann-data is applied.

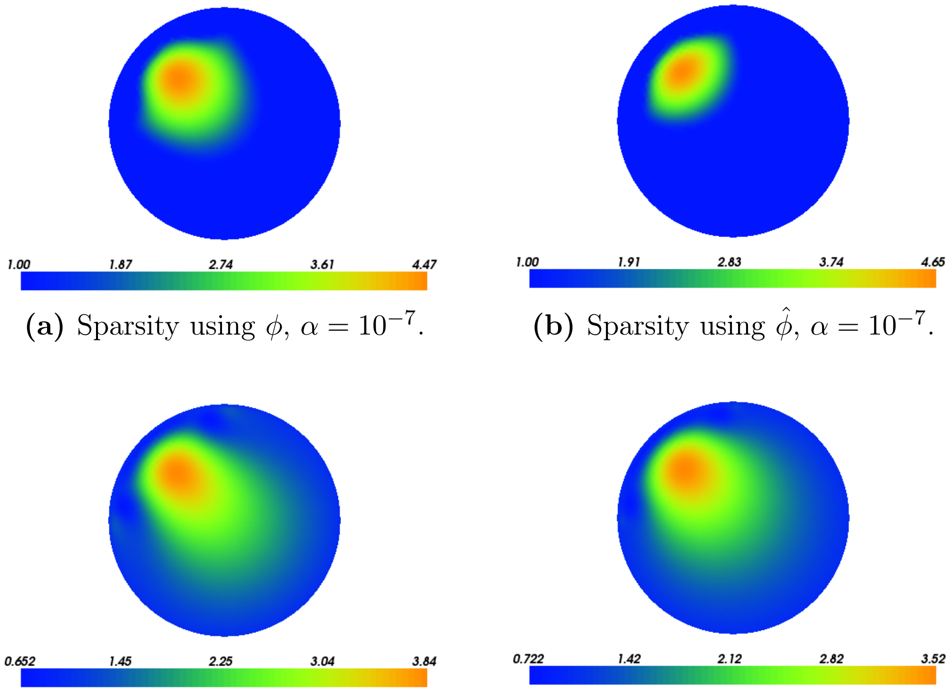


**(a)** Sparsity using $\phi$, $\alpha = 5 \cdot 10^{-6}$. **(b)** Sparsity using $\hat{\phi}$, $\alpha = 5 \cdot 10^{-6}$.

**(c)** TV using $\phi$, $\beta = 5 \cdot 10^{-6}$. **(d)** TV using $\hat{\phi}$, $\beta = 5 \cdot 10^{-6}$.

**Figure 4.49:** Sparsity and TV reconstruction with full data $\phi$ and partial data $\hat{\phi}$, using the partial Neumann-data in (4.6), where $\Gamma$ is given in Figure 4.48a.

In Figure 4.50 we see the result using $\Gamma$ in Figure 4.48b on the preceding page, and the shape of the inclusion is further warped. It is seen that the part of the inclusion farthest away from $\Gamma$ is reconstructed much worse than for full Neumann-data, and it is also that part of the inclusion where the shape is warped the most. Intuitively this is what we could expect from only using data on part of the boundary, namely, that the reconstruction is better for the part of the inclusion which is closest to the data. These results are further emphasized in Figure 4.51 on the following page where $\Gamma$ now is given by Figure 4.48c on the preceding page and is indeed very small. Here it is very evident that using the partial Dirichlet-data $\hat{\phi}$ gives much worse results, and from Figure 4.51b that the reconstructed support is much too small.



**(a)** Sparsity using $\phi$, $\alpha = 10^{-7}$.

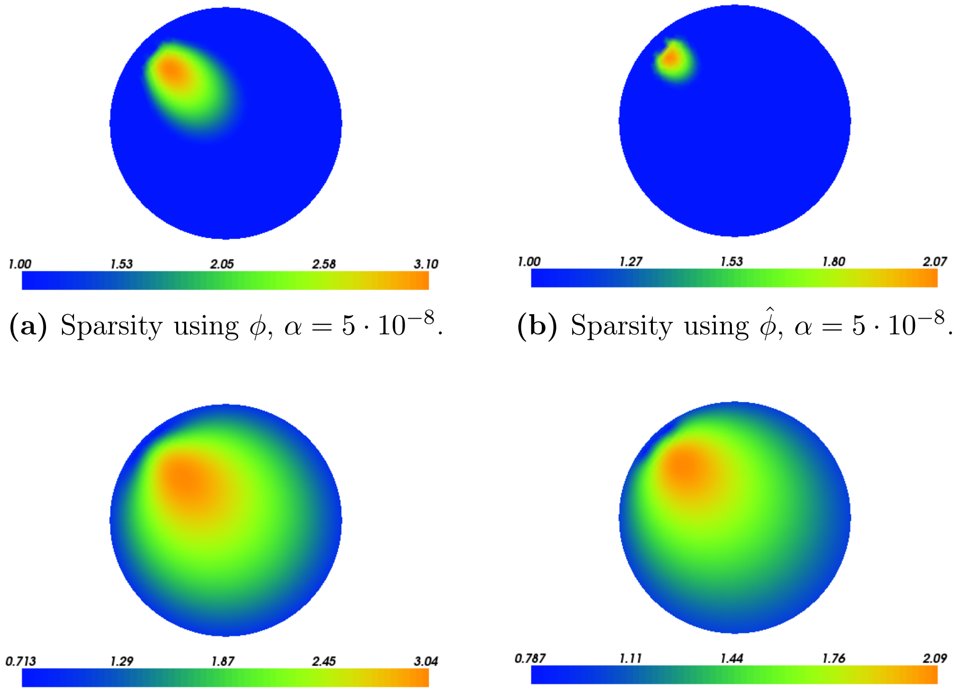
**(b)** Sparsity using $\hat{\phi}$, $\alpha = 10^{-7}$.

**(c)** TV using $\phi$, $\beta = 10^{-6}$.

**(d)** TV using $\hat{\phi}$, $\beta = 10^{-6}$.

**Figure 4.50:** Sparsity and TV reconstruction with full data $\phi$ and partial data $\hat{\phi}$, using the partial Neumann-data in (4.6), where $\Gamma$ is given in Figure 4.48b on the preceding page.

In general TV regularization is not performing very well with partial Neumann-data, and the support is vastly overestimated. Sparsity regularization is performing much better in terms of finding the support as well as determining the contrast quite well considering the lack of data, even for partial Dirichlet-data $\hat{\phi}$, except in Figure 4.51b. It is also noticed that the regularization parameters have to be reduced along with $\Gamma$ becoming smaller, since the discrepancy term consists of an integral over a smaller domain, and the balance between the discrepancy term and the penalty term needs to be evened out.



**(a)** Sparsity using $\phi$, $\alpha = 5 \cdot 10^{-8}$.

**(b)** Sparsity using $\hat{\phi}$, $\alpha = 5 \cdot 10^{-8}$.

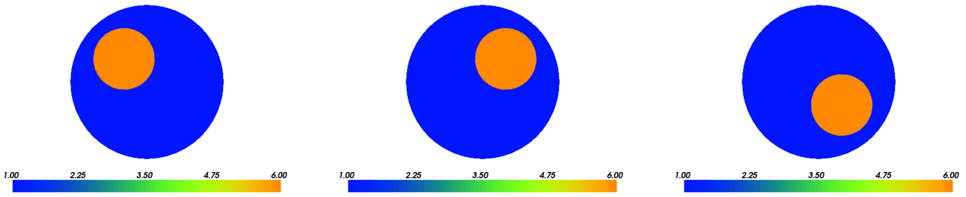
**(c)** TV using $\phi$, $\beta = 10^{-6}$.

**(d)** TV using $\hat{\phi}$, $\beta = 10^{-6}$.

**Figure 4.51:** Sparsity and TV reconstruction with full data $\phi$ and partial data $\hat{\phi}$, using the partial Neumann-data in (4.6), where $\Gamma$ is given in Figure 4.48c on page 108.

## Position of the Inclusion

Here we will use a fixed $\Gamma$ as shown in Figure 4.48b on page 108, and reconstruct inclusions given by the three cases in Figure 4.52.



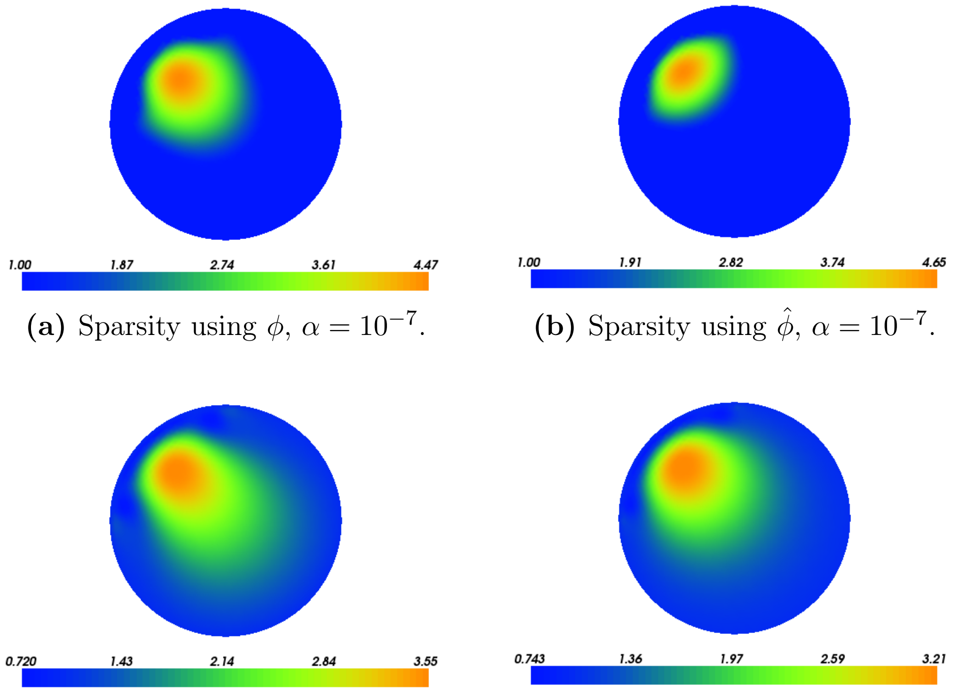
**(a)** Case 1. **(b)** Case 2. **(c)** Case 3.

**Figure 4.52:** Phantoms in three cases.

From Figure 4.53 through 4.55 on pages 112–114 reconstruction are done using the phantoms in Figure 4.52. Intuitively, the reconstructions are better the closer they are to the Neumann-data, and that is certainly also the case. However, when applying the full Dirichlet-data $\phi$ to the sparsity regularization, we see that the reconstructions actually approximate the support quite well (when considering the limited data), even in Figure 4.54 and Figure 4.55.

When only using the partial Dirichlet-data $\hat{\phi}$, the quality of the reconstructions rapidly decay as the inclusion is moved away from $\Gamma$. In Figure 4.55b and Figure 4.55d we see that these solutions do not at all resemble the sought inclusion, and there are even some artefacts at the boundary of $\Gamma$ as a result of the discontinuities in the Neumann-data via $\tilde{g}_N$ (see Figure 4.46 on page 107).

Thus we can conclude that in general TV regularization is less favoured when it comes to the use of partial Neumann-data, even when full Dirichlet-data is applied. Sparsity regularization performs quite well for partial Neumann-data and to some degree also when partial Dirichlet-data are applied, however, in that case it should only be used to reconstruct inclusions near $\Gamma$.



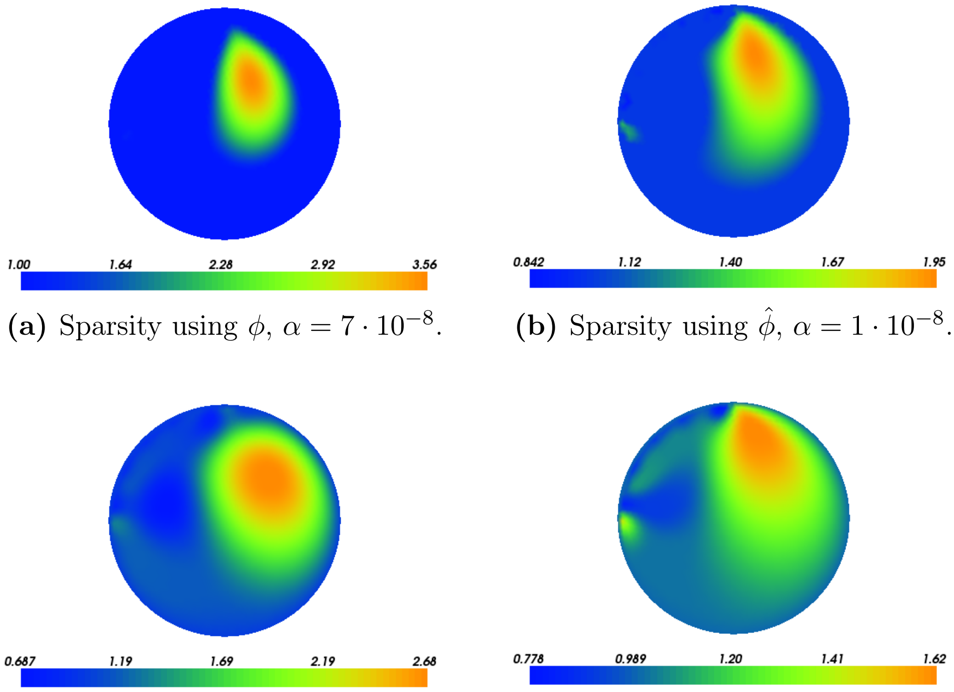
**(a)** Sparsity using $\phi$, $\alpha = 10^{-7}$.

**(b)** Sparsity using $\hat{\phi}$, $\alpha = 10^{-7}$.

**(c)** TV using $\phi$, $\beta = 10^{-5}$.

**(d)** TV using $\hat{\phi}$, $\beta = 10^{-5}$.

**Figure 4.53:** Sparsity and TV reconstruction of the inclusion in Figure 4.52a on the previous page with full data $\phi$ and partial data $\hat{\phi}$ using the partial Neumann-data in (4.6), where $\Gamma$ is given in Figure 4.48b on page 108.



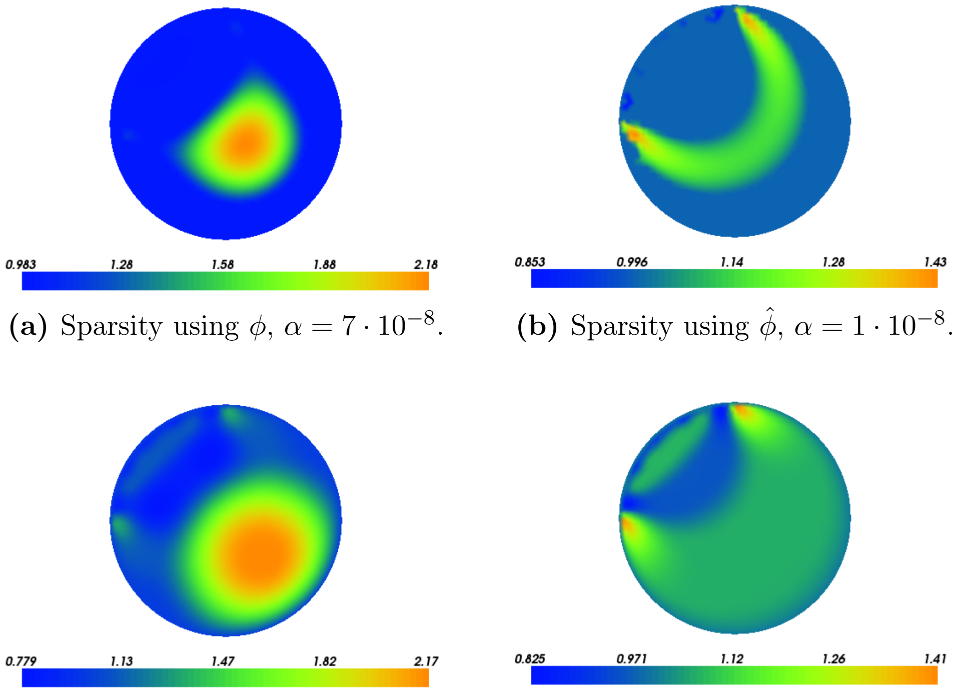
**(a)** Sparsity using $\phi$, $\alpha = 7 \cdot 10^{-8}$.

**(b)** Sparsity using $\hat{\phi}$, $\alpha = 1 \cdot 10^{-8}$.

**(c)** TV using $\phi$, $\beta = 10^{-5}$.

**(d)** TV using $\hat{\phi}$, $\beta = 10^{-5}$.

**Figure 4.54:** Sparsity and TV reconstruction of the inclusion in Figure 4.52b on page 111 with full data $\phi$ and partial data $\hat{\phi}$ using the partial Neumann-data in (4.6), where $\Gamma$ is given in Figure 4.48b on page 108.



**(a)** Sparsity using $\phi$, $\alpha = 7 \cdot 10^{-8}$.

**(b)** Sparsity using $\hat{\phi}$, $\alpha = 1 \cdot 10^{-8}$.

**(c)** TV using $\phi$, $\beta = 10^{-5}$.

**(d)** TV using $\hat{\phi}$, $\beta = 10^{-5}$.

**Figure 4.55:** Sparsity and TV reconstruction of the inclusion in Figure 4.52c on page 111 with full data $\phi$ and partial data $\hat{\phi}$ using the partial Neumann-data in (4.6), where $\Gamma$ is given in Figure 4.48b on page 108.

## 4.9 Reconstructions Based on Prior Information

The title of this section is a bit misleading, since all the reconstructions performed so far are based on prior information about the solution. This is in the form of the bias that is introduced in the regularization, which for sparsity regularization is that the solution is sparse in the chosen basis, and for TV regularization that the solution is approximately piecewise constant. Even the use of the Sobolev gradient $\nabla_s J(\sigma)$ (Corollary 3.5) uses the prior information that the inclusion is zero on the boundary of the domain.

What is meant by the title of this section is the use of prior information regarding the distinct basis functions used in sparsity regularization. Instead of using the same regularization parameter $\alpha_k := \alpha,\ \ \forall k$, we will now scale the parameters for the basis functions based on prior information that they are included in the basis expansion of the solution. One can scale each basis function individually based on the precision of the prior information, however, I make a simple case where

$$\alpha_k := \mu_k \alpha. \tag{4.7}$$

Here $\alpha$ is a parameter similar to what has been used so far, and for basis functions with no prior information $\mu_k := 1$, i.e. they are penalized in the usual way. But for basis functions where I have prior information, I will use $\mu_k \in [0,1)$ and thereby penalize these basis functions less, thus encouraging the method to include them in the solution.

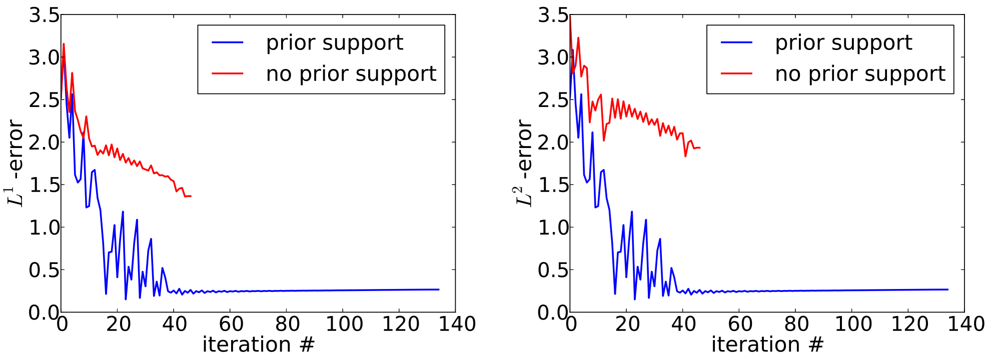
Since I use the FEM basis with the sparsity regularization, prior information on the FEM basis functions corresponds to prior information on the support of the inclusions (since each FEM basis function determines the support near each nodal point). Thereby we can help the method by indicating where the inclusion should be located, and the main goal is to be able to get a very precise reconstruction of the contrast and hopefully also some sharper edges in the reconstruction. Here it should be stressed that without the use of prior information it is currently hopeless to get sharp edges in EIT reconstruction.

There will be three different types of tests in this section, firstly we shall look at the use of exact prior information of the support of the inclusions, secondly we investigate the results when incomplete or completely false prior information is applied. The third test will concern an idea I got during the tests, where I used TV regularization to approximate the support of inclusions, and use this information as prior information in sparsity regularization.

## Exact Prior Information

In the previous tests I have used phantoms, that can be applied to any grid. However, to more easily see the result of the prior information that is applied for the sparsity regularization, I instead show the phantoms projected onto the coarse grid which is used in the iterative algorithm. For this reason the *true solution* will be more jagged near the boundary of the inclusion due to the size of the triangles in the mesh. Because it can be hard to determine how well the edges in the reconstruction are determined, I have also included plots of the difference between the true solutions and the reconstructions.

From Figure 4.57 on the next page and Figure 4.58 on page 118 we see the same prior information used, but where the scaling of the prior information in Figure 4.57 is $\mu_k = 10^{-2}$ and in Figure 4.58 it is $\mu_k = 0.5$. Evidently these results supports the intuition that a smaller penalty leads to a stronger enforcement of the prior information. Furthermore, we see that the contrast and support along with sharp edges are almost exactly reconstructed in Figure 4.57.

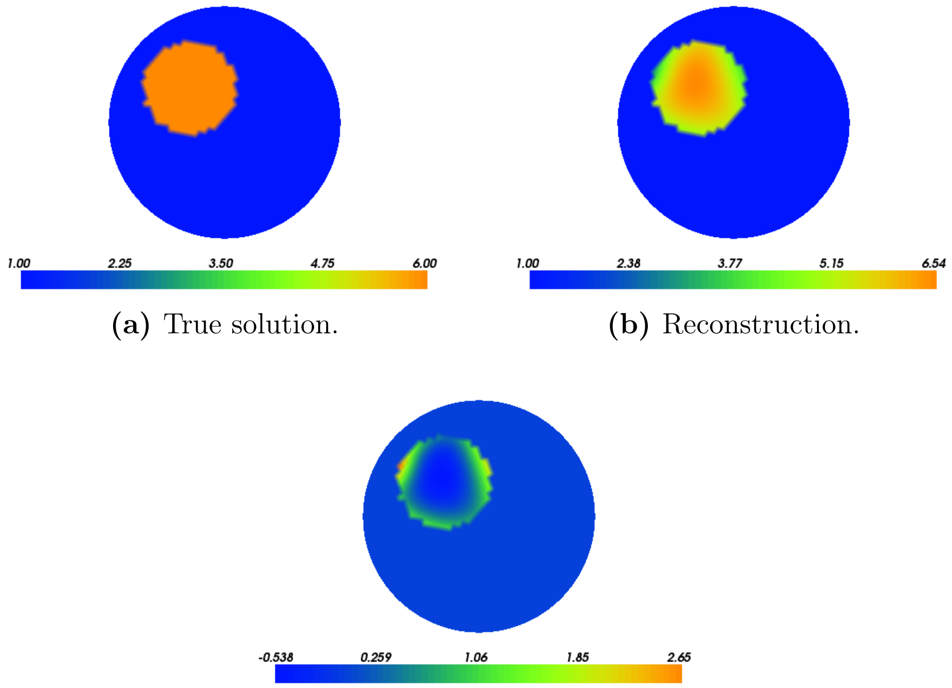


**(a)** Absolute $L^1$ error.

**(b)** Absolute $L^2$ error.

**Figure 4.56:** Errors at different iterations, with and without the use of prior information on the support for the case in Figure 4.57 on the facing page.

It is also evident from Figure 4.56 that using the exact prior information for the support of the inclusion leads to a much faster convergence. The



**(a)** True solution.

**(b)** Reconstruction.

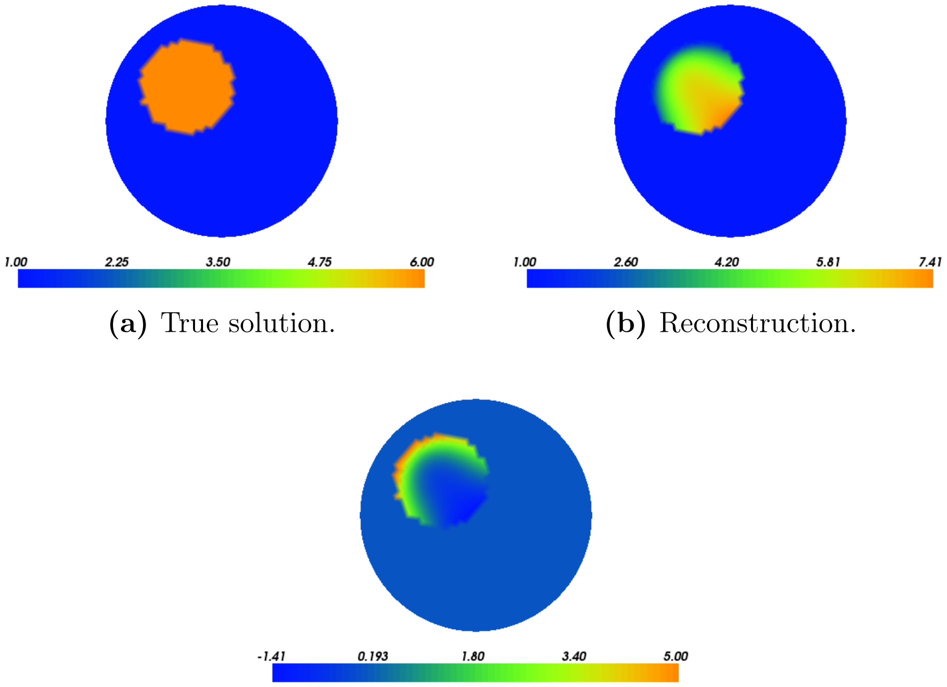
**(c)** Difference.

**Figure 4.57:** Sparsity reconstruction using exact prior information. $\mu_k = 10^{-2}$ for prior information, and $\alpha = 2 \cdot 10^{-5}$.

methods do, however, keep going for quite a lot of iterations after the initial convergence, without too much improvement.

In Figure 4.59 on page 119 reconstruction with partial data close to $\Gamma$ is seen to also be reconstructed very successfully. However, in Figure 4.60 on page 120 it is also seen that even exact prior information for the support can not remedy the use of partial data when the inclusion is not close to $\Gamma$, and it seems like there is not much that can be done to further improve that case.

We saw in Section 4.5 on page 94 how difficult multiple inclusions with different amplitudes and ring-type inclusion are to reconstruct, so it is very satisfying to see how well they can be reconstructed using prior information in Figure 4.61 through 4.63 on pages 121–123. The support is correctly determined and the contrast is quite close, and correctly distributed in the cases with multiple inclusions. It is still a little hard to reconstruct the



**(a)** True solution.

**(b)** Reconstruction.

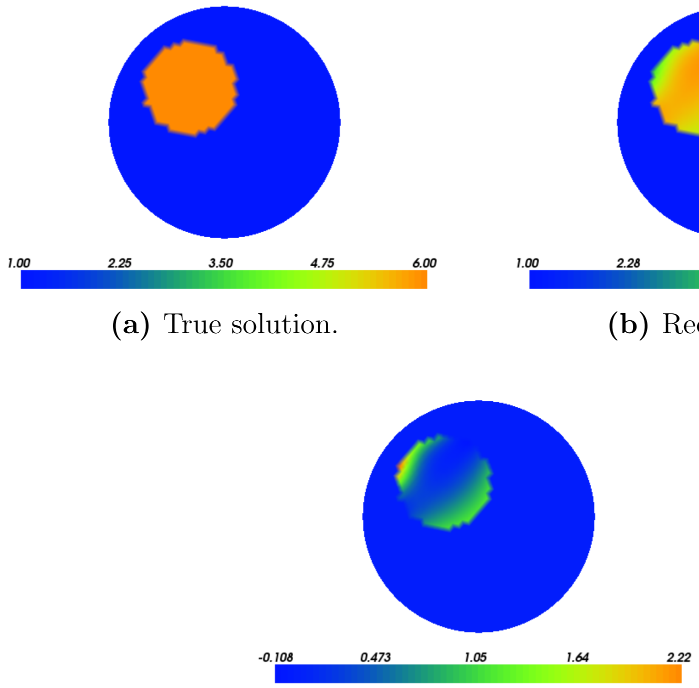
**(c)** Difference.

**Figure 4.58:** Sparsity reconstruction using exact prior information. $\mu_k = 0.5$ for prior information, and $\alpha = 2 \cdot 10^{-5}$.

correct contrast for the ring-type inclusion, however, it is vastly improved.

Finally I have tried to use prior information on the non-uniform mesh in Figure 4.64 on page 124, and while it does remove some of the grain from the solution compared to Figure 4.43b on page 103, it is still not desirable to properly use this algorithm on a non-uniform mesh when considering that the exact support is known.

Comparing with the previous tests, it is quite evident how powerful prior information can be in determining better solutions. However, in this case exact prior information was applied, and it is rare that we in practice know precisely where the support of an inclusion is located.

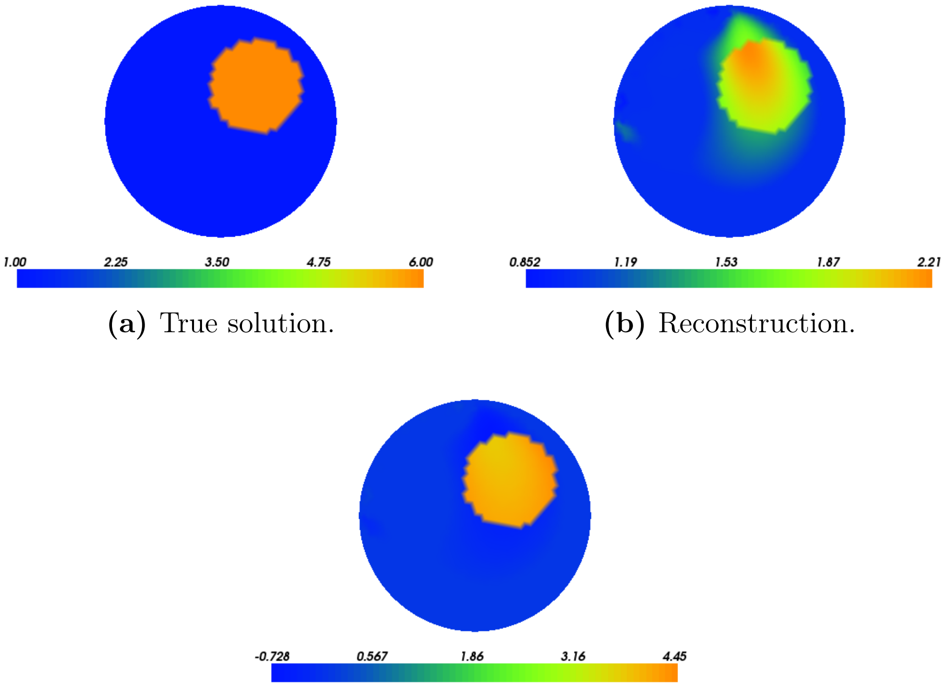


**(a)** True solution.

**(b)** Reconstruction.

**(c)** Difference.

**Figure 4.59:** Sparsity reconstruction for partial Neumann- and Dirichlet-data corresponding with Figure 4.53b on page 112. Exact prior information is applied. $\mu_k = 10^{-2}$ for prior information, and $\alpha = 10^{-7}$.

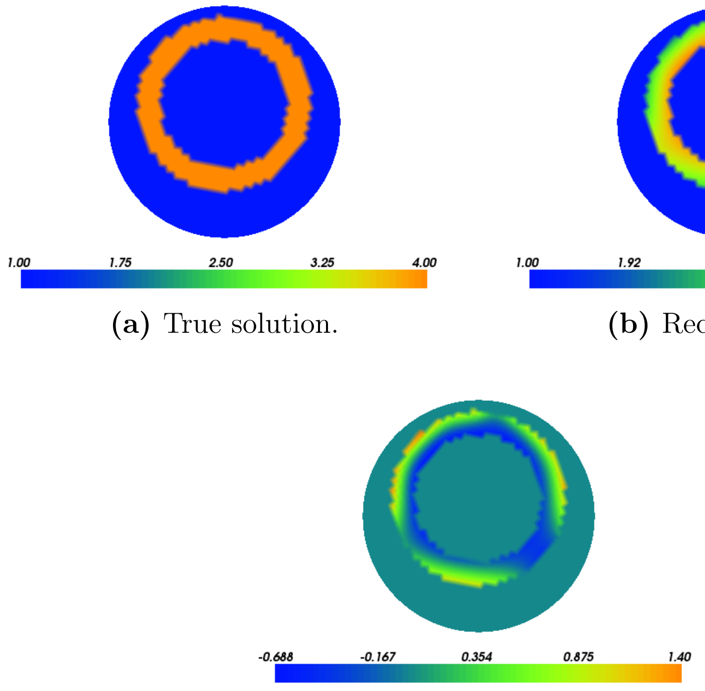


**(a)** True solution.

**(b)** Reconstruction.

**(c)** Difference.

**Figure 4.60:** Sparsity reconstruction for partial Neumann- and Dirichlet-data corresponding with Figure 4.54b on page 113. Exact prior information is applied. $\mu_k = 10^{-2}$ for prior information, and $\alpha = 10^{-8}$.

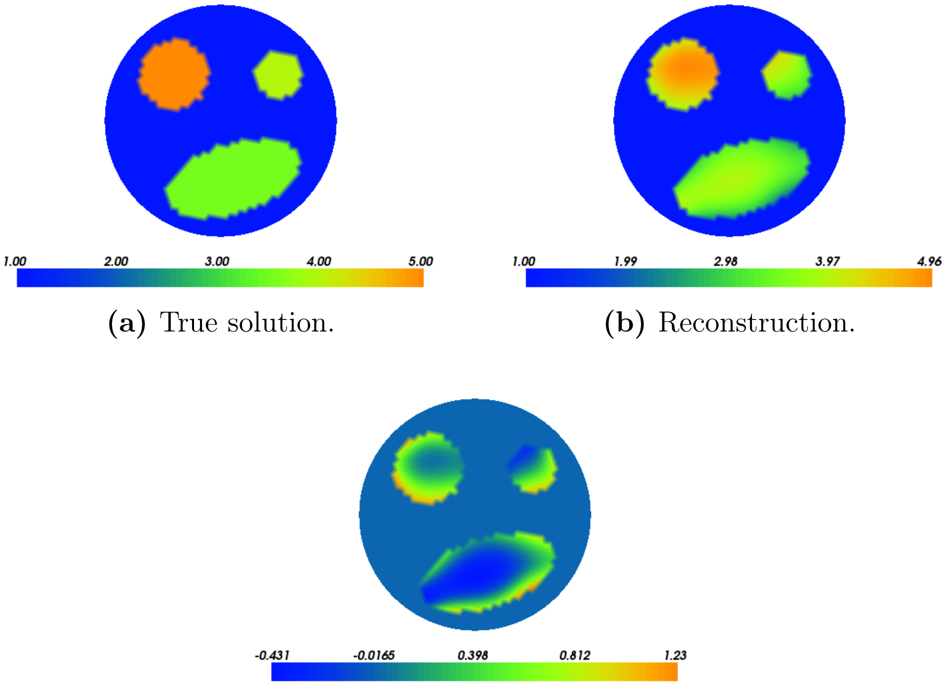


**(a)** True solution.

**(b)** Reconstruction.

**(c)** Difference.

**Figure 4.61:** Sparsity reconstruction using exact prior information. $\mu_k = 10^{-2}$ for prior information, and $\alpha = 10^{-6}$.

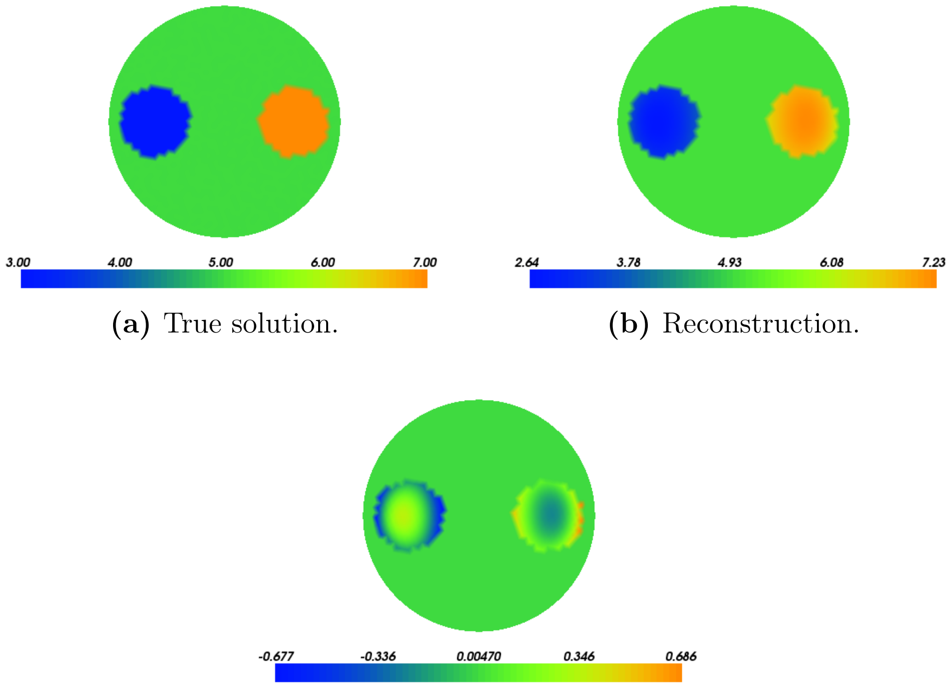


**(a)** True solution.

**(b)** Reconstruction.

**(c)** Difference.

**Figure 4.62:** Sparsity reconstruction using exact prior information. $\mu_k = 10^{-2}$ for prior information, and $\alpha = 10^{-6}$.

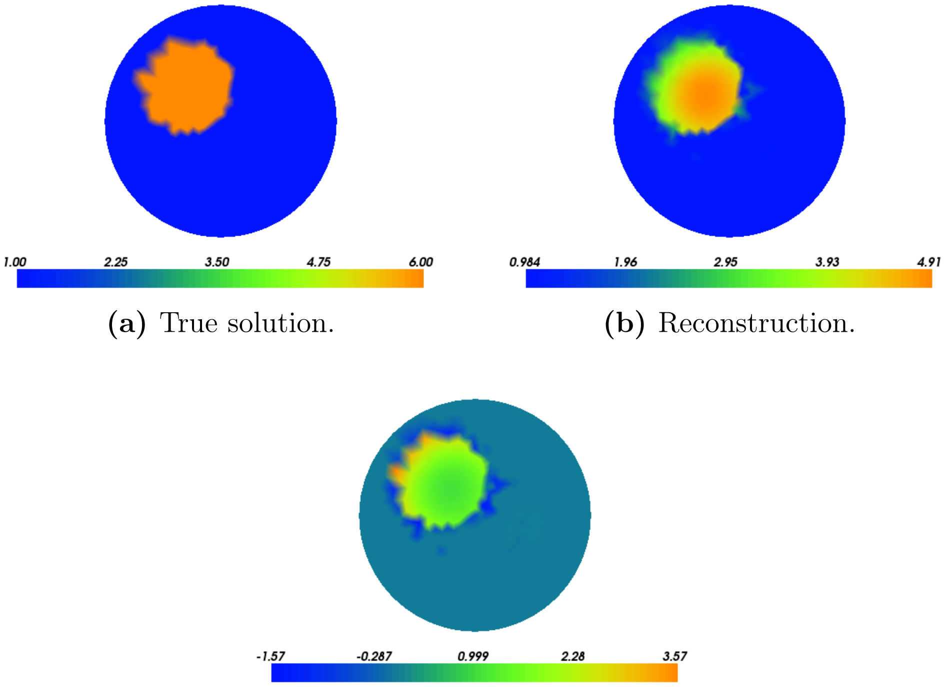


**(a)** True solution.

**(b)** Reconstruction.

**(c)** Difference.

**Figure 4.63:** Sparsity reconstruction using exact prior information. $\mu_k = 10^{-2}$ for prior information, and $\alpha = 10^{-7}$.



**(a)** True solution.

**(b)** Reconstruction.

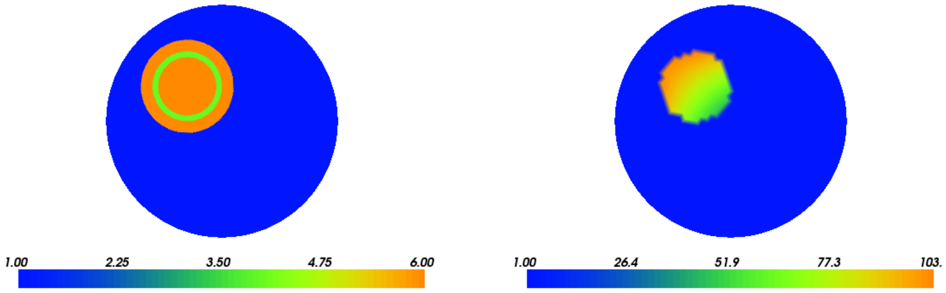
**(c)** Difference.

**Figure 4.64:** Sparsity reconstruction on a non-uniform mesh corresponding with Figure 4.43b on page 103. Exact prior information is applied. $\mu_k = 10^{-2}$ for prior information, and $\alpha = 10^{-6}$.

## Incomplete or False Prior Information

Now let's investigate some cases where the applied prior information is either partially correct or false. To illustrate the applied prior information, I have marked an outline with a green color on the plots of the phantoms in the following figures, and everything contained within this outline (including the width of the outline itself) is assumed to be the support of the inclusion.

From Figure 4.65 it is quite evident that even though the prior information that is given is correct, i.e. the inclusion is non-zero in the given location, the reconstruction is characterized with only having an inclusion where the prior information is given, furthermore, the contrast is exceptionally large in the reconstruction. To give an intuitive explanation of this phenomenon, we can refer to Section 2.1 on page 14 where we saw for the concentric case that smaller support and larger contrast can give similar Dirichlet-data as larger support and smaller contrast. The bias introduced by the prior information for the too small support may have lead to a local minimum in the objective function where the Dirichlet-data is sufficiently similar using the small support and the large contrast.



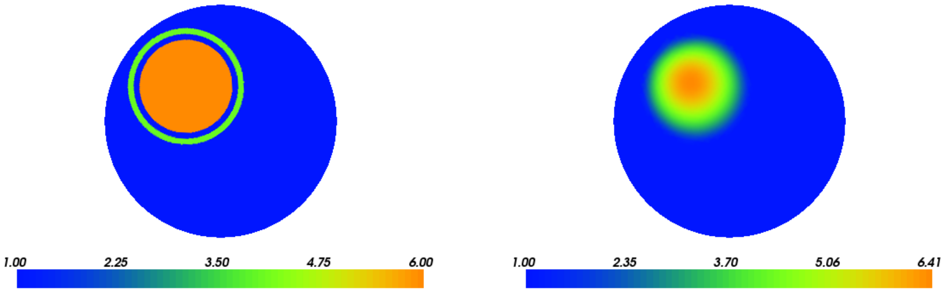
**(a)** Phantom and prior outline.

**(b)** Reconstruction.

**Figure 4.65:** Sparsity regularization using prior information with correct centre, but too small support. $\mu_k = 10^{-2}$ for prior information, and $\alpha = 2 \cdot 10^{-5}$.

In Figure 4.66 on the next page we see that assuming that the support is larger than it really is, we actually get a reconstruction very similar to what we got without the explicit use of prior information on the support. The contrast is better reconstructed than without the explicit prior information, but otherwise not much of an improvement. It is also evident, that in this case we do not see the prior information being enforced as heavily as in

Figure 4.65, since the support is not reconstructed larger than the actual phantom, and we again get the smoother transition near the boundary of the inclusion as we saw in earlier tests.



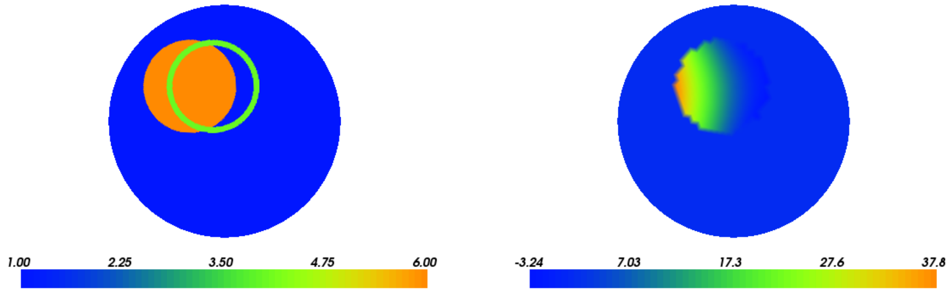
**(a)** Phantom and prior outline. **(b)** Reconstruction.

**Figure 4.66:** Sparsity regularization using prior information with correct centre, but too large support. $\mu_k = 10^{-2}$ for prior information, and $\alpha = 2 \cdot 10^{-5}$.

For Figure 4.67 on the facing page and Figure 4.68 on page 128 the size of the support is correct, however, the centre of the inclusion is assumed to be shifted. In both cases we see a much too larger contrast, and the contrast is largest in the direction of where the true inclusion is supposed to be. In Figure 4.67 there is an overlap between the prior information and the phantom, and it is also evident that the majority of the reconstructed inclusion lies in this overlap.

For Figure 4.68b there is a small inclusion where the true inclusion is supposed to be, but the prior information is applied even if it is completely false and the main part of the reconstruction lies where the prior information was prescribed, however, with the contrast pointing in the direction of the true phantom. Increasing the scaling $\mu_k$ to 0.5 for the prior information in Figure 4.68c on page 128 improves the result slightly, but increasing $\mu_k$ to 0.9 for the prior information gives a much better result, as seen in Figure 4.68d on page 128. This again follows the intuition that the prior information is enforced according to how low the scaling $\mu_k$ is.

Thus it can be concluded that the use of prior information in this way can be extremely powerful in determining a better solution that also has sharp edges reconstructed. However, if the prior information is incorrect,



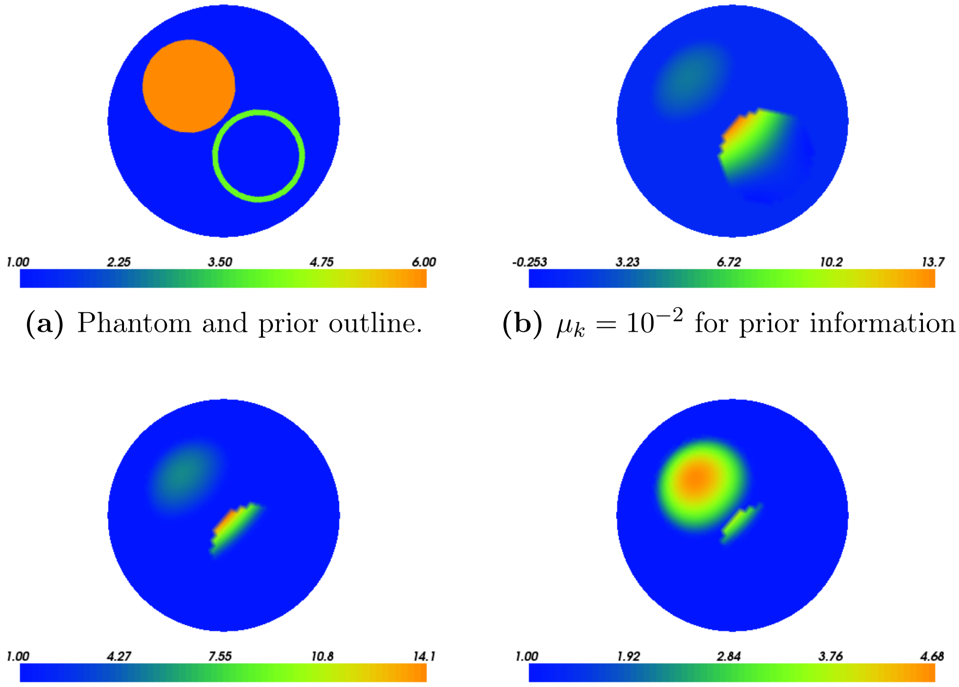
**(a)** Phantom and prior outline.

**(b)** Reconstruction.

**Figure 4.67:** Sparsity regularization using prior information with correct size of inclusion, but with wrong centre, and with overlap between phantom and prior information. $\mu_k = 10^{-2}$ for prior information, and $\alpha = 2 \cdot 10^{-5}$.

the method tends to apply the prior information anyway, which can lead to worse reconstructions compared to when no prior information on the support is used. Therefore using prior information in this way, one should probably always also make a reconstruction with no prior information on the support as a sanity check of whether the result makes sense. Furthermore, estimating the support slightly larger yields much better results than estimating the support too small, and it can be used without too much risk. Finally, we should always scale the prior information according to the correctness of the prior information, i.e. for well-determined prior information we can allow a small $\mu_k$ to enforce the prior information, while prior information with only doubtful precision we should have $\mu_k$ close to 1 to avoid the above cases.

We may also observe that using false prior information can lead to situation where many iterations are used, and sometimes many very small step sizes are applied as seen in Figure 4.69 on page 129.



**(a)** Phantom and prior outline.

**(b)** $\mu_k = 10^{-2}$ for prior information.

**(c)** $\mu_k = 0.5$ for prior information.

**(d)** $\mu_k = 0.9$ for prior information.

**Figure 4.68:** Sparsity regularization using prior information with correct size of inclusion, but with wrong centre, and with no overlap between phantom and prior information. $\alpha = 2 \cdot 10^{-5}$.

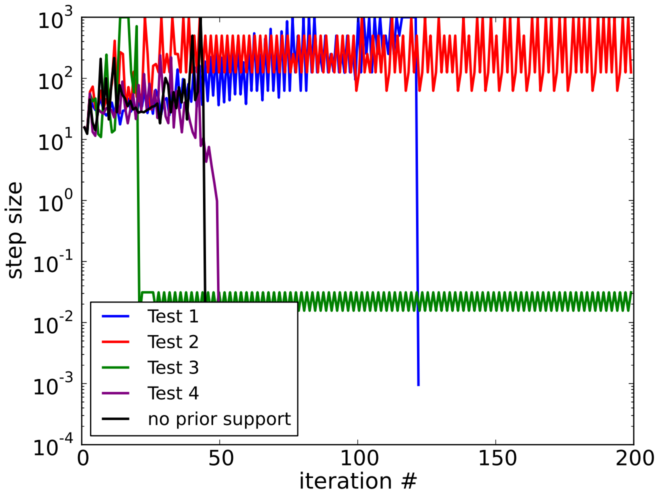


**Figure 4.69:** Step sizes in the iterations for the above four tests compared to a similar reconstruction using no prior information on the support.

## Combining TV and Sparsity Regularization

As we saw in the previous tests on the use of false prior information, a slightly larger support is preferable to a too small support in the applied prior information. We have also seen how TV regularization in many cases estimates the support of the inclusions quite well, yet lacks the contrast from sparsity regularization. Therefore I got the idea of making a simple experiment by estimating the support using TV regularization and use this as prior information in the sparsity regularization. In this way we actually do not make use of any explicit assumptions on the location of the support, because this information is gained from the TV regularization.

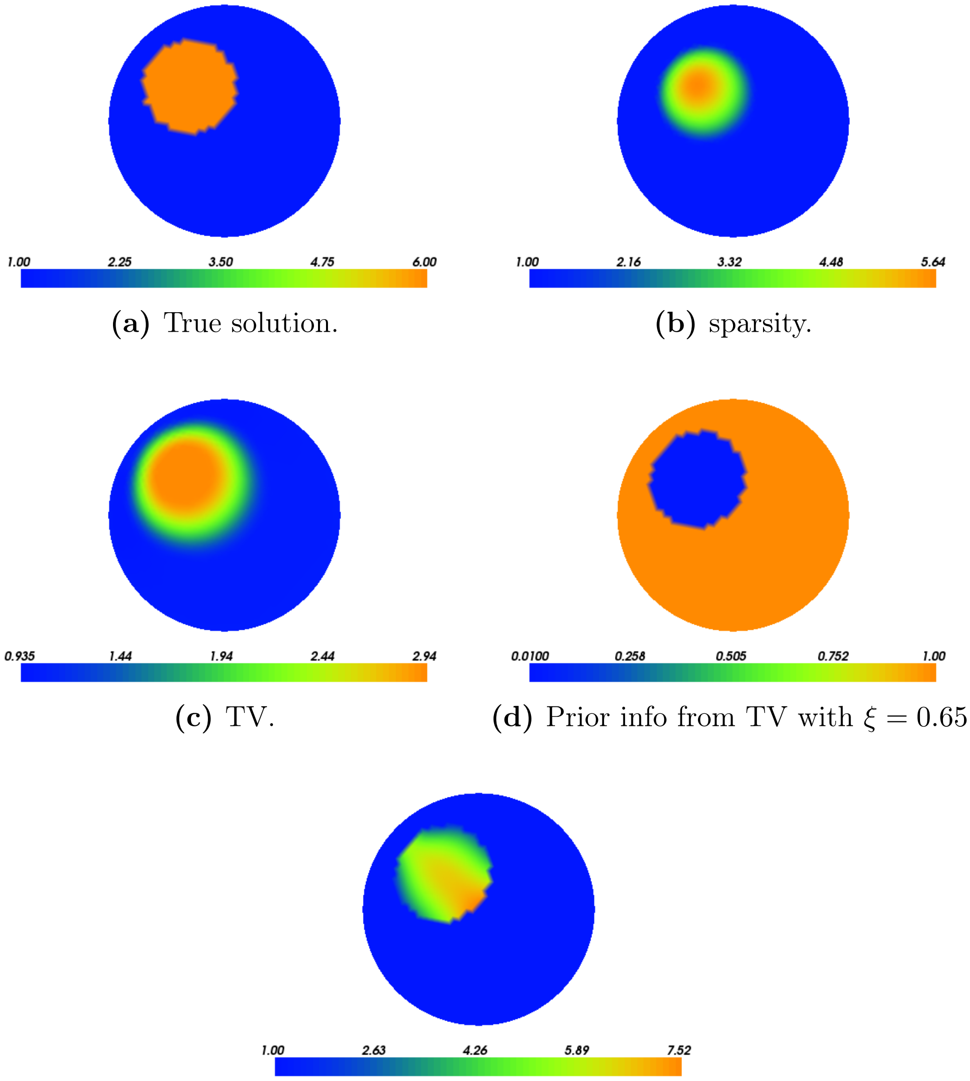
To get the prior information from the TV reconstruction $\sigma_{\mathrm{TV}}$, I have used the cases with positive inclusions, and given a parameter $\xi \in (0,1)$ then the prior information in terms of $\mu_k$ is approximated by:

$$\mu_k := \begin{cases} 1, & \sigma_{\mathrm{TV}}(x_k) \leq \xi \|\sigma_{\mathrm{TV}}\|_{L^\infty(\Omega)}, \\ 10^{-2}, & \sigma_{\mathrm{TV}}(x_k) > \xi \|\sigma_{\mathrm{TV}}\|_{L^\infty(\Omega)}, \end{cases} \tag{4.8}$$

where $\{x_k\}$ are the nodal points. This means that large values in the reconstruction from TV regularization leads to a penalty with $\alpha_k = 10^{-2}\alpha$, while small values in the TV reconstructions leads to the usual penalization $\alpha_k = \alpha$.

In Figure 4.70 on the next page we see the usual sparsity and TV solutions along with the prior information extracted from TV regularization, and finally the solution using the prior information. The result first of all has sharp edges and the support is quite well reconstructed, and the contrast is quite well reconstructed as well except at the bottom right part of the inclusion where the contrast was overestimated. In this way we have actually achieved a better reconstruction, without actually having more initial information about the solution.

Figure 4.71 on page 132 shows the case with multiple inclusions, and first of all it is noticed that the smallest inclusion is not determined well with the TV regularization. From the final reconstruction using the prior information from the TV reconstruction, it is observed that the two inclusion found by TV regularization are reconstructed quite well and with sharp edges and a better distribution of the contrast compared to the usual sparsity reconstruction. The small inclusion that was undetectable by the TV reconstruction was of course not as well reconstructed since no prior information was applied at that location. However, there is still an inclusion



**(a)** True solution.

**(b)** sparsity.

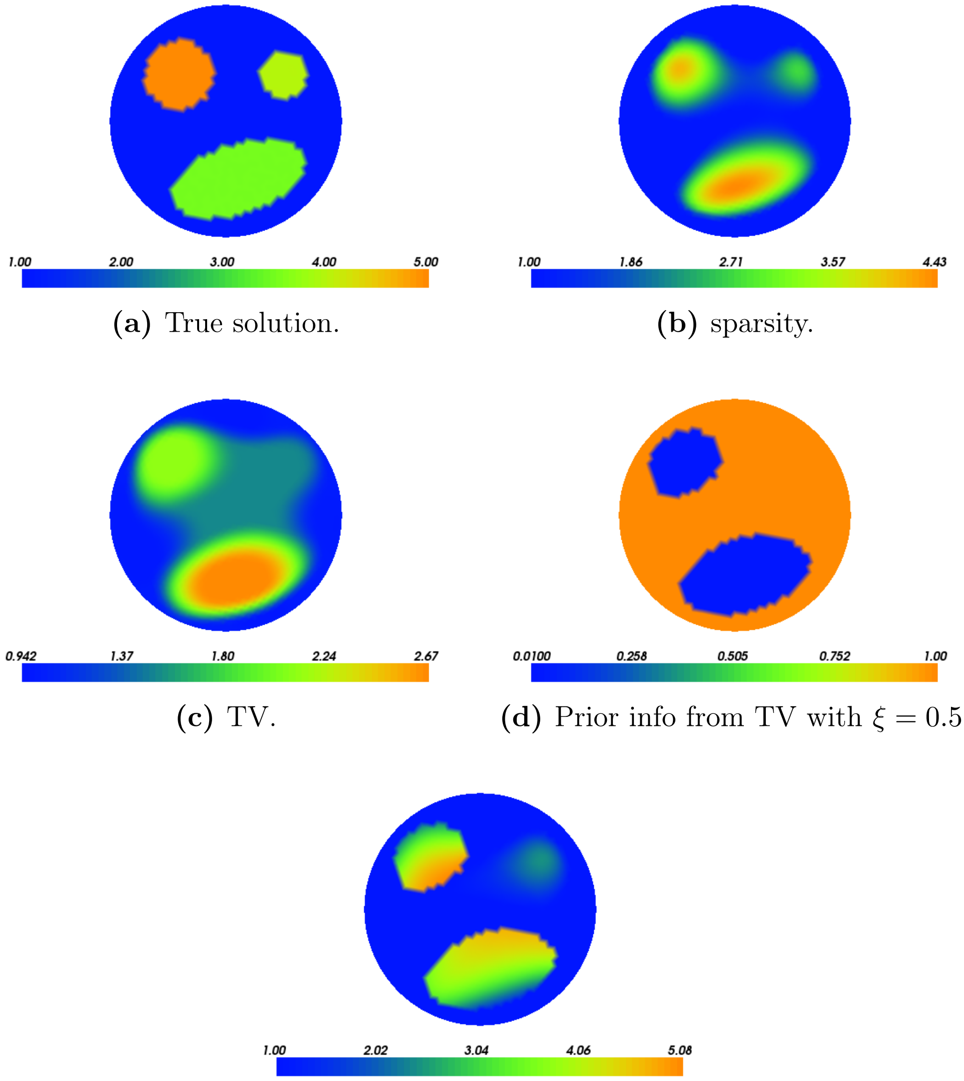
**(c)** TV.

**(d)** Prior info from TV with $\xi = 0.65$.

**(e)** Sparsity with prior from TV.

**Figure 4.70:** Sparsity regularization using prior information on the support of the inclusion based on TV reconstruction. $\alpha = 2 \cdot 10^{-5}$.

reconstructed but it is not as good as the usual sparsity reconstruction. So in this case we get a trade off in this sense of getting better reconstructions of the two larger inclusions and a worse reconstruction of the small inclusion.



**(a)** True solution.

**(b)** sparsity.

**(c)** TV.

**(d)** Prior info from TV with $\xi = 0.5$.

**(e)** Sparsity with prior from TV.

**Figure 4.71:** Sparsity regularization using prior information on the support of the inclusion based on TV reconstruction. $\alpha = 10^{-6}$.

One could consider maybe using both sparsity and TV regularization to get an idea of the support of the inclusions, which would also include the small inclusion in this case. One may also look into a more sophisticated way of attaining the prior information from the TV reconstruction rather than a simple threshold.

It should also be noted that there are likely better and faster methods than TV regularization to get an estimate of the support of an inclusion.

# 5 Discussion

In this chapter there will be a short discussion on ideas and further interesting ways one could continue investigating these methods beyond what was presented in this thesis.

The first subject I would like to discuss is the use of a different basis than the FEM basis, because as seen in Theorem 3.12, we could use any orthogonal basis for $H_0^1(\Omega)$ (orthogonal in the $H^1$-metric). So the question is what kinds of properties that can be extracted by having a sparse representation from a different basis. Take for instance a wavelet basis for $L^2(\Omega)$ for which we could emphasize the small scales of the solution, and in that way emphasize discontinuities. Though it is not completely clear if there are similar orthonormal bases for $H_0^1(\Omega)$.

When speaking of promoting discontinuities in the solution, then TV seemed like a good candidate, however, as we have seen in Chapter 4 the contrast is severely reduced by the bias introduced from the regularization. An interesting idea would be to combine sparsity and TV regularization, such that we had the objective function

$$\psi(\delta\sigma) := J(\sigma_0 + \delta\sigma) + R_{\boldsymbol{\alpha},1}(\delta\sigma) + \beta P_{\mathrm{TV},c}(\delta\sigma).$$

Hopefully, the sparsity regularization could help the TV regularization obtain the correct contrast, and that TV regularization could help the sparsity regularization near the boundary of the inclusion, such that we get a piecewise constant inclusion with the correct contrast. However, it may also turn out that the two regularization techniques hopelessly counter each other, and due to multiple bias for the different regularizations we get something worse than either sparsity or TV regularization alone, something that is simply too difficult to predict without actually trying it out.

Another interesting idea, would be whether the sparsity regularization parameters could be changed adaptively, i.e. that in each iteration we could change the parameters based on how the current iterate looks, and for instance emphasize the nodes where the iterate has larger values. In that way we could automatically approximate the support of the solution without explicitly giving prior information about the location of the support. This leads to several questions, where probably the most important one is whether the problem is even well-defined, since now we deal with an optimization problem where the objective function changes in each iteration, and is this even possible to solve? It would definitely put a stop to the use of the weak monotonicity in its current form (Definition 3.7) because we would no longer be able to compare the objective function between different iterations. Instead this may be achieved by saving the most recent iterates $\{\delta\sigma_k\}_{k=i-M+1}^{i}$, and then compare with the new objective function $\Psi_i$ computed for the previous iterates:

$$\Psi_i(\delta\sigma_{i+1}) \leq \max_{i-M+1\leq j\leq i} \Psi_i(\delta\sigma_j) - \frac{\tau}{2s_i}\|\delta\sigma_{i+1} - \delta\sigma_i\|_{H^1(\Omega)}^2.$$

Another way to get an adaptive method would be to start out with a very coarse mesh and then refine it in each iteration near where the inclusion is non-zero, such that we in the end get a very fine mesh near the support of the inclusion, and a coarse mesh where the inclusion is zero. This is actually a well understood technique that has been used in other applications to for instance speed up computationally tough problems. However, in this particular problem we again arrive at the notion that we have to change the regularization parameter in each iteration, since the sparsity penalty term depends a lot on the discretization and how many nodes that are present in the mesh. So in this case we would also have to solve the issues with changing the objective function in each iteration. Furthermore, as seen in Section 4.7 on page 102 sparsity regularization performs very poorly on non-uniform mesh, so adaptively refining the mesh locally is probably not a good idea in the method's current state. However, for TV regularization it was evident in Section 4.7 that it performs quite well on non-uniform mesh, and there will not be problems with the regularization parameter if the mesh is altered. TV regularization also tends to use far more iterations than sparsity regularization, so reducing the computational burden by initially using a coarse mesh could significantly improve the speed of the method.

The above discussions leads to a new subject, namely, how we can generalize the sparsity regularization parameter such that a good choice of a parameter on one mesh translates to a similarly good parameter on a finer/coarser mesh. The idea would be to have a specific choice of the parameter, and then scale it depending on the number of nodes in the mesh or the average size of the elements. It is a highly undesirable quality of the FEM basis that the regularization parameters depends so much on the mesh, and not only on the actual problem. Therefore a good and simple way to generalize this dependency will greatly help the method become more transparent and easy to apply. I suppose that it would also help in regards of determining an automated approach in finding a good regularization parameter.

Now one of the more interesting subjects is the use of prior information to help solve the problem. My initial idea would be to use a fixed parameter $\alpha$ for the sparsity regularization when no prior information is known, and then scale it down depending on the quality of the prior information to penalize it less, i.e.

$$\alpha_k := \mu_k \alpha.$$

Here $0 < \mu_k \leq 1$ (and possibly a larger lower bound to avoid getting to close to 0) will control how the specific basis function should be penalized. For prior information that is not well-determined we should have $\mu_k$ close to 1 such that it is close to no prior information at all. If the prior information is very well-determined i.e. we are almost completely sure that the prior information is true, then we can have $\mu_k$ become close to the lower bound to penalize the basis function less and therefore promote it in the solution. Again this very practical approach leads to a very difficult question: How can we quantify how well-determined prior information is? It surely depends on what type of prior information we talk about i.e. what basis that is used in the sparsity regularization. Another question is also to what degree it actually makes sense to use the prior information, for instance if we aim to use EIT for medical imaging, then it would be doctors who need to interpret the reconstructions, and using some prior information may bias the reconstruction in a way that other important details are less apparent or completely gone from the reconstruction.

Finally, it should be noted that the use of prior information in the sparsity regularization can easily be translated to other problems than EIT, as it simply depends on changing the discrepancy term for the actual problem and choosing an appropriate basis. While the $\ell_1$-penalty term is commonly used, it would be interesting to see some new sparsity based methods using interesting bases to promote characteristic properties in otherwise difficult problems, by applying explicit prior information about the basis functions.

# 6

# Conclusion

I have been investigating the results from Jin and Maass [8], Jin *et al.* [9] in great detail, and provided several proofs that were omitted in the articles. I have furthermore been skeptical towards some of the more heuristic arguments in the articles and provided more rigorous arguments that actually alters the method slightly. In particular in terms of the step size in Corollary 3.6, the normalization that occurs in the sparsity steps in Theorem 3.12, and the entire result of Theorem 4.1 by using the properties of the FEM basis to derive an approximate solution rather than simply assume that the FEM basis is orthonormal (and a basis for $H_0^1(\Omega)$), which is far from the truth.

Many of the results had not been proved and only some were stated in Jin and Maass [8], Jin *et al.* [9], including all the results in the first part of Chapter 2, and all the results in Chapter 3 and Chapter 4. All the results regarding total variation regularization were done from scratch. As for the results in Section 2.2 I have greatly expanded on the details of the proofs, as details were very sparse and it was implied that the reader should fill out the gaps. I have also expanded on the theory that was presented in Jin and Maass [8], Jin *et al.* [9] by generalizing the use of several regularization parameters in sparsity regularization.

I got the idea of using a regularization parameter for each basis function in order to apply explicit prior information to include specific basis functions in the solution. This was in practice applied via the FEM basis to give prior information on the support of inclusions, and provided correct prior information this leads to exceptionally good reconstructions, something that is not possible to do with EIT without the use of prior information. I have also had the idea to combine total variation regularization and sparsity regularization, thus improving solutions without the use of further prior information, by simply applying the solution from total variation regularization to determine an approximation to the support for the inclusions. Furthermore, I have generalized the methods to allow for the use of partial data, something that in general always will be the case for practical 3D reconstructions of the human body.

I touched on many advanced topics during the thesis, including weak formulations of PDE's, Gâteaux and Fréchet derivatives and their use in Taylor expansions as well as their use in gradient descend methods. I have also been introduced to sparsity and total variation regularization, and their respective strengths and weaknesses.

I have also successfully implemented both the sparsity and total variation regularization for EIT, which includes solving multiple PDE's numerically. Furthermore, I have formulated the forward problem via (2.13), such that it easily can be implemented using the space $H^1(\Omega)$ instead of $\tilde{H}^1(\Omega)$. The implementation has been done in the programming language Python using the software package FEniCS [31], both of which I have had no prior experience with. It was observed that sparsity regularization does indeed attempt to find a solution which is sparse in the applied basis, here the FEM basis. The total variation regularization attempts to find a solution that is constant at the location of the inclusion, however, it does turn out that the transition at the boundary of the inclusion is not discontinuous in the solution but rather smooth. The smooth transition is most likely due to the use of the smoother gradient via the Sobolev gradient $\nabla_s J(\sigma)$, however, the use of this gradient also ensured the stability near the boundary, by enforcing that the solution equals the background conductivity on the boundary.

I have also programmed many different tests to thoroughly investigate the properties of the solutions in different scenarios, including different sizes of inclusions with different amplitudes, and observed how the reconstruction is affected by the distance of the inclusion to the boundary of the domain. Using uniform and non-uniform mesh it was observed how badly sparsity regularization performs on the non-uniform mesh while total variation regularization is almost indifferent. It was seen that sparsity regularization in general performs best on reconstructing the contrast, while total variation regularization performs slightly better for reconstructing the support. For partial data it was seen that sparsity regularization performs quite well, as long as there are data near the inclusion, while total variation regularization vastly overestimates the support.

For the use of explicit prior information of the support of the inclusions, it was possible to make near-perfect reconstructions with sparsity regularization when exact information was provided. However, it was also seen how unstable the solution is towards false prior information, and the scaling of the regularization parameters must therefore in practice be determined from the precision of the prior information, i.e. how sure we are that the prior information is correct. By using total variation to determine an approximation to the support of inclusions in the solution, it was used as prior information in sparsity regularization, giving a good approximation to the support in the sparsity solution, but it also improved the contrast of the solution significantly. I have also discussed the possibility of having both sparsity and total variation penalty terms in the objective function for the optimization problem, however, that is beyond the scope of this thesis but still an interesting idea to maybe be able to simultaneously approximate the contrast and the support well.

# Appendix A

# Multi-Index Notation

In many situations in $\mathbb{R}^d$ it becomes clumsy to write things out completely, and a more compact notation such as multi-index can help out. A multi-index is a $d$-tuple $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_d)$ where $\alpha_j \in \mathbb{N}_0 := \mathbb{N} \cup \{0\},\ j = 1, 2, \ldots, d$. One may write that $\alpha \in {\mathbb{N}_0}^d$ i.e. the $d-1$ Cartesian products of $\mathbb{N}_0$. The following shows how multi-index notation is used:

$$|\alpha| := \sum_{j=1}^{d} \alpha_j, \tag{A.1}$$

$$\alpha! := \prod_{j=1}^{d} \alpha_j!, \tag{A.2}$$

$$x^\alpha := \prod_{j=1}^{d} x_j^{\alpha_j}, \quad x \in \mathbb{R}^n, \tag{A.3}$$

$$\partial^\alpha := \frac{\partial^{\alpha_1}}{\partial x^{\alpha_1}} \frac{\partial^{\alpha_2}}{\partial x^{\alpha_2}} \cdots \frac{\partial^{\alpha_d}}{\partial x^{\alpha_d}}. \tag{A.4}$$

There is also a version of the product rule for differentiation in terms of the multi-index [21]:

$$\partial^\alpha (fg) = \sum_{\beta+\gamma=\alpha} \frac{\alpha!}{\beta!\gamma!} (\partial^\beta f)(\partial^\gamma g), \tag{A.5}$$

where $\beta$ and $\gamma$ are multi-indices.

# *Appendix* B

# Sobolev Spaces

This chapter will mainly be a set of definitions and results taken from Adams and Fournier [15], Evans [20], and will be used as reference for the notation and definitions of many of the spaces in this report, as they are instrumental in determining the more general weak solutions to a PDE. In the following it is assumed that $\Omega$ is an open subset of $\mathbb{R}^d$.

It should be noted that there is a bijection between a subspace $V$ of the space of distributions $\mathcal{D}'(\Omega)$ and the space $L^1_{\text{loc}}(\Omega)$ of locally integrable functions [21], given by

$$\langle f, \phi \rangle := \int_\Omega f \phi dx, \;\; \phi \in C_c^\infty(\Omega),$$

where $f$ on the right hand-side is a function $f \in L^1_{\text{loc}}(\Omega)$ and on the left hand-side we have the dual pairing of the corresponding distribution $f \in V$ and the test function $\phi \in C_c^\infty(\Omega)$. Due to the identification of the spaces $V$ and $L^1_{\text{loc}}(\Omega)$ it is common to use the same symbol for the function and the corresponding distribution.

**Definition B.1 (Weak Derivative)** *A weak derivative is another name for the distribution derivative, but for functions in $L^1_{\text{loc}}(\Omega)$. So the $\alpha$'th weak partial derivative of $u \in L^1_{\text{loc}}(\Omega)$ is (if it exists) the unique function $v \in L^1_{\text{loc}}(\Omega)$ satisfying*

$$\int_\Omega u \partial^\alpha \phi dx = (-1)^{|\alpha|} \int_\Omega v \phi dx, \;\; \forall \phi \in C_c^\infty(\Omega),$$

*and we write $v = \partial^\alpha u$.*

It should be noted that such terms as bijection between $V$ and $L^1_{\text{loc}}(\Omega)$, and uniqueness of a weak derivative, is up to a set of measure zero. However, there is no loss of generality in omitting this detail, since the $L^p$-spaces and $L^1_{\text{loc}}(\Omega)$ in reality consists of equivalence classes of functions that are equal up to a set of measure zero, and writing $f \in L^1_{\text{loc}}(\Omega)$ it is understood that $f \in [f]$ with $[f]$ being such an equivalence class.

**Definition B.2 (Sobolev Spaces $\boldsymbol{W^{k,p}}$)** *The Sobolev spaces are defined as follows*

$$W^{k,p}(\Omega) := \{u \in L^p(\Omega) \mid \partial^\alpha u \in L^p(\Omega),\ \ \forall |\alpha| \leq k\},\ \ k \in \mathbb{N}_0, p \in [1, \infty].$$

*The $W^{k,p}$-spaces are often equipped with the following norms for which they become Banach spaces,*

$$\|u\|_{W^{1,p}(\Omega)} := \begin{cases} \left( \displaystyle\sum_{|\alpha| \leq k} \|\partial^\alpha u\|^p_{L^p(\Omega)} \right)^{1/p}, & p \in [1, \infty), \\ \displaystyle\sum_{|\alpha| \leq k} \|\partial^\alpha u\|_{L^\infty(\Omega)}, & p = \infty. \end{cases}$$

For $p = 2$ one usually writes $H^k(\Omega) := W^{k,2}(\Omega)$ and these spaces are Hilbert spaces when equipped with the inner product

$$\langle u, v \rangle_{H^k(\Omega)} := \sum_{|\alpha| \leq k} \langle \partial^\alpha u, \partial^\alpha v \rangle_{L^2(\Omega)},\ \ u, v \in H^k(\Omega).$$

In the case of $H^1(\Omega)$ where there is no hint of mixed derivatives, one may write the inner product as

$$\langle u, v \rangle_{H^1(\Omega)} = \langle u, v \rangle_{L^2(\Omega)} + \langle \nabla u, \nabla v \rangle_{L^2(\Omega)},\ \ u, v \in H^1(\Omega),$$

where $\langle \nabla u, \nabla v \rangle_{L^2(\Omega)} = \int_\Omega \nabla u \cdot \overline{\nabla v} dx$. One can then write $\|u\|^2_{H^1(\Omega)} = \|u\|^2_{L^2(\Omega)} + \|\nabla u\|^2_{L^2(\Omega)}$.

**Definition B.3 ($\boldsymbol{W_0^{k,p}}$)** *The spaces $W_0^{k,p}(\Omega)$ are defined as the closure of $C_c^\infty(\Omega)$ in the $\|\cdot\|_{W^{k,p}(\Omega)}$-norm.*

Similarly we write $H_0^k(\Omega) := W_0^{k,2}(\Omega)$.

**Theorem B.4 (Trace Theorem)** *Assume that $\Omega$ is open and bounded and $\partial\Omega$ is $C^1$, then there exists a bounded linear operator $T : W^{1,p}(\Omega) \to L^p(\partial\Omega)$ such that*

*(i)* $Tu = u|_{\partial\Omega}$ *if* $u \in W^{1,p}(\Omega) \cap C(\overline{\Omega})$.

*(ii)* $\|Tu\|_{L^p(\partial\Omega)} \leq C\|u\|_{W^{1,p}(\Omega)}, \;\; \forall u \in W^{1,p}(\Omega)$, *where $C$ only depends on $p$ and $\Omega$.*

Using the trace theorem the $W_0^{1,p}(\Omega)$-spaces can be characterized in the following way, assuming that $\Omega$ satisfies the assumptions.

**Theorem B.5 (Characterization of $\boldsymbol{W_0^{1,p}}$)** *Assume that $\Omega$ is open and bounded and $\partial\Omega$ is $C^1$ then*

$$W_0^{1,p}(\Omega) = \{u \in W^{1,p}(\Omega) \mid Tu = 0\}.$$

From Theorem B.4 it is evident that $T(W^{1,p}(\Omega)) \subseteq L^p(\partial\Omega)$, however, it turns out that $T$ is not surjective, thus it may be convenient to name the range of $T$.

**Definition B.6 ($\boldsymbol{W^{1/2,p}}$)** *Assume $\Omega$ is open and bounded and $\partial\Omega$ is $C^1$, then $W^{1/2,k}(\partial\Omega)$ is defined as*

$$W^{1/2,p}(\partial\Omega) := T(W^{1,p}(\Omega)).$$

*A norm on the $W^{1/2,p}(\partial\Omega)$ space could for instance be*

$$\|u\|_{W^{1/2,p}(\partial\Omega)} := \inf\{\|v\|_{W^{1,p}(\Omega)} \mid v \in W^{1,p}(\Omega), \;\; Tv = u\}.$$

Once again we write $H^{1/2}(\partial\Omega) := W^{1/2,2}(\partial\Omega)$.

The continuous duals of the Sobolev spaces are often used when solving PDE's for instance when applying the Lax-Milgram theorem (Theorem C.3), therefore the continuous duals have a special notation.

**Definition B.7** *The continuous dual spaces of the Sobolev spaces are annotated as follows:*

$$\begin{aligned} W^{-k,p}(\Omega) &:= (W_0^{k,p}(\Omega))', \;\; k \in \mathbb{N}_0, \\ W^{-1/2,p}(\partial\Omega) &:= (W^{1/2,p}(\partial\Omega))'. \end{aligned}$$

As one might expect we write $H^{-k}(\Omega) := W^{-k,2}(\Omega)$ and $H^{-1/2}(\partial\Omega) := W^{-1/2,2}(\partial\Omega)$.

## B.1 Inequalities and Imbeddings

First let's define an imbedding.

**Definition B.8 (Imbedding)** *Let $X$ and $Y$ be normed spaces, then we say that $X$ is* imbedded *in $Y$ and write $X \hookrightarrow Y$ if the following holds*

*(i) $X$ is a subspace of $Y$,*

*(ii) the inclusion operator $\iota : X \to Y$ that maps $\iota x = x$, $\forall x \in X$ is continuous or equivalently (as $\iota$ is linear) bounded.*

The following version of the Sobolev imbedding theorem is taken from Adams and Fournier [15]. It should be noted that in Adams and Fournier [15] a requirement for the Sobolev imbedding theorem is that the domain $\Omega$ satisfies the so called *cone condition*, however, it is later remarked that if $\Omega$ is bounded then it is sufficient that $\Omega$ is a Lipschitz domain. However in order to stay on topic I have opted to use the less general assumption that $\Omega$ is bounded and has $C^1$ boundary.

**Theorem B.9 (Sobolev Imbedding Theorem)**
*Let $\Omega$ be an open bounded domain in $\mathbb{R}^d$ with $C^1$ boundary, and let $1 \leq p < \infty$, then the following imbeddings apply.*

*(i) If $k < \frac{d}{p}$ then*

$$W^{k,p}(\Omega) \hookrightarrow L^q(\Omega), \;\; p \leq q \leq dp/(d-kp).$$

*(ii) If $k > \frac{d}{p}$ or if $k = d$ and $p = 1$ then*

$$W^{k,p}(\Omega) \hookrightarrow L^q(\Omega), \;\; p \leq q \leq \infty.$$

*(iii) If $k = \frac{d}{p}$ then*

$$W^{k,p}(\Omega) \hookrightarrow L^q(\Omega), \;\; p \leq q < \infty.$$

*If $W^{k,p}$ is substituted with $W_0^{k,p}$ then $\Omega$ can be an arbitrary domain in $\mathbb{R}^d$. The operator norm $\|\iota\|$ for the inclusion operator only depends on $\Omega$, $d$, $k$, $p$, and $q$.*

Hölder's inequality is well-known and can often be quite helpful, however, the following generalized Hölder's inequality will prove to be very useful when investigating the continuity and differentiability of the forward map for EIT.

**Theorem B.10 (Generalized Hölder's Inequality)** *Let $m \in \mathbb{N}$ and $r \in [1,\infty)$, and let $p_j \in [1,\infty]$, $j = 1,2,\dots,m$ such that*

$$\sum_{j=1}^{m} \frac{1}{p_j} = \frac{1}{r}, \tag{B.1}$$

*where $\frac{1}{p_j}$ is identified with 0 if $p_j = \infty$. For $u_j \in L^{p_j}(\Omega)$, $j = 1,2,\dots,m$ and $u := \prod_{j=1}^m u_j$ then $u \in L^r(\Omega)$ and*

$$\|u\|_{L^r(\Omega)} \leq \prod_{j=1}^{m} \|u_j\|_{L^{p_j}(\Omega)}.$$

**Proof.** Firstly the $p_j$'s are ordered such that $p_1 \leq p_2 \leq \dots \leq p_m$. The proof will be by induction, and since the case $m = 1$ is trivial i.e. $p_m = r$, it is assumed that $m \geq 2$. For simplicity the $L^p(\Omega)$-norm is denoted $\|\cdot\|_p$.

**Case 1:** $p_m = \infty$

Since $p_m = \infty$ then $\sum_{j=1}^{m-1} \frac{1}{p_j} = \frac{1}{r}$. Pulling the $\operatorname{ess\,sup}_{x\in\Omega}|u_m(x)|$ outside $\|\prod_{j=1}^m u_j\|_r$ yields

$$\|\prod_{j=1}^{m} u_j\|_r \leq \|\prod_{j=1}^{m-1} u_j\|_r \|u_m\|_\infty = \|\prod_{j=1}^{m-1} u_j\|_{1/\left(\sum_{j=1}^{m-1}\frac{1}{p_j}\right)} \|u_m\|_{p_m}. \tag{B.2}$$

**Case 2:** $p_m < \infty$

Due to the ordering of the $p_j$'s then $p_1, p_2, \dots, p_{m-1} < \infty$ so $p_m > r$ in order for (B.1) to hold, i.e. $p_m \in (1,\infty)$. Now define

$$p := \frac{p_m}{p_m - r}, \quad q := \frac{p_m}{r}.$$

So as $p_m \in (1,\infty)$ and $p_m > r$ then $p,q \in (1,\infty)$ and $\frac{1}{p}+\frac{1}{q}=1$ i.e. the usual Hölder's inequality can be applied:

$$\||\prod_{j=1}^{m} u_j|^r\|_1 = \||\prod_{j=1}^{m-1} u_j|^r|u_m|^r\|_1 \leq \||\prod_{j=1}^{m-1} u_j|^r\|_p\||u_m|^r\|_q,$$

thus

$$\|\prod_{j=1}^{m-1} u_j\|_r \leq \||\prod_{j=1}^{m-1} u_j|^r\|_p^{1/r}\||u_m|^r\|_q^{1/r} = \|\prod_{j=1}^{m-1} u_j\|_{pr}\|u_m\|_{qr}. \tag{B.3}$$

As $qr = p_m$ and as $pr = \frac{p_m r}{p_m - r} = 1/(\frac{1}{r}-\frac{1}{p_m}) = 1/(\sum_{j=1}^{m-1}\frac{1}{p_j})$ then (B.3) becomes

$$\|\prod_{j=1}^{m-1} u_j\|_r \leq \|\prod_{j=1}^{m-1} u_j\|_{1/\left(\sum_{j=1}^{m-1}\frac{1}{p_j}\right)}\|u_m\|_{p_m}. \tag{B.4}$$

So in both cases we get (B.2) and (B.4), thus by induction of this equation $m-1$ times yields

$$\|\prod_{j=1}^{m-1} u_j\|_r \leq \|u_m\|_{p_m}\|u_{m-1}\|_{p_{m-1}}\cdots\|u_2\|_{p_2}\|u_1\|_{1/(1/p_1)} = \prod_{j=1}^{m}\|u_j\|_{p_j}, \tag{B.5}$$

thereby proving the theorem. □

Now we can easily prove the following theorem.

**Theorem B.11** *Let $\Omega$ be bounded, then $L^{p_2}(\Omega) \hookrightarrow L^{p_1}(\Omega)$ for any $1 \leq p_1 \leq p_2 \leq \infty$, and $\|\iota\|$ depends only on $p_1, p_2$ and $\Omega$.*

**Proof.** The case $p_1 = p_2$ is trivial so let $p_1 < p_2$ such that $p_1 \in [1,\infty)$ and $p_2 \in (p_1,\infty]$. Now define $p_3 \in [1,\infty)$ such that $\frac{1}{p_1} = \frac{1}{p_2}+\frac{1}{p_3}$. Now let $u \in L^{p_2}(\Omega)$ then by Theorem B.10

$$\|u\|_{L^{p_1}(\Omega)} = \|u\cdot 1\|_{L^{p_1}(\Omega)} \leq \|u\|_{L^{p_2}(\Omega)}\|1\|_{L^{p_3}(\Omega)} = |\Omega|^{1/p_3}\|u\|_{L^{p_2}(\Omega)} < \infty,$$

where $|\Omega|$ is the Lebesgue measure of $\Omega$, which is finite as $\Omega$ is bounded. As $u \in L^{p_2}(\Omega)$ was arbitrary then $L^{p_2}(\Omega) \subseteq L^{p_1}(\Omega)$. The inclusion is bounded with $\|\iota\| \leq |\Omega|^{1/p_3} = |\Omega|^{1/p_1 - 1/p_2}$, and equality is achieved for $u = 1$ on $\Omega$, i.e. $\|\iota\| = |\Omega|^{1/p_1-1/p_2}$. □

**Theorem B.12** *Let $\Omega$ be bounded, then $W^{k,p_2}(\Omega) \hookrightarrow W^{k,p_1}(\Omega)$ for any $1 \leq p_1 \leq p_2 \leq \infty$, and $\|\iota\|$ depends only on $k$, $p_1$, $p_2$ and $\Omega$.*

**Proof.** As in the proof of Theorem B.11 the case $p_1 = p_2$ is trivial, so initially assume that $p_1 < p_2 < \infty$. Instead of working with the norm defined in Definition B.2 we introduce the following equivalent norm[1],

$$\|u\|'_{W^{k,p}(\Omega)} = \sum_{|\alpha|\leq k} \|\partial^\alpha u\|_{L^p(\Omega)}, \quad u \in W^{k,p}(\Omega).$$

So by use of Theorem B.11 let $u \in W^{k,p_2}(\Omega)$,

$$\|u\|_{W^{k,p_1}(\Omega)} \leq C_1 \|u\|'_{W^{k,p_1}(\Omega)} = C_1 \sum_{|\alpha|\leq k} \|\partial^\alpha u\|_{L^{p_1}(\Omega)} \leq C_1 \sum_{|\alpha|\leq k} C_\alpha \|\partial^\alpha u\|_{L^{p_2}(\Omega)},$$

so by letting $C_2 := C_1 \max_{|\alpha|\leq k} C_\alpha$ then

$$\|u\|_{W^{k,p_1}(\Omega)} \leq C_2 \sum_{|\alpha|\leq k} \|\partial^\alpha u\|_{L^{p_2}(\Omega)} = C_2 \|u\|'_{W^{k,p_2}(\Omega)} \leq C_3 \|u\|_{W^{k,p_2}(\Omega)}.$$

Now consider the case $p_1 < p_2 = \infty$, then we can use almost the same approach as above:

$$\|u\|_{W^{k,p_1}(\Omega)} \leq C_1 \sum_{|\alpha|\leq k} \|\partial^\alpha u\|_{L^{p_1}(\Omega)} \leq C_2 \sum_{|\alpha|\leq k} \|\partial^\alpha u\|_{L^{\infty}(\Omega)} = C_2 \|u\|_{W^{k,\infty}(\Omega)}.$$

□

The following version of Poincaré's inequality is not the most common one, since the term $u_\Omega$ is usually equal to $u_\Omega = |\Omega|^{-1} \int_\Omega u dx$, i.e. the average of $u$ over $\Omega$, however the following version will be more useful in this report as we will deal with situations where $\int_{\partial\Omega} Tu ds = 0$.

**Theorem B.13 (Poincaré's Inequality)** *Let $\Omega$ be a bounded connected open subset of $\mathbb{R}^d$ with $\partial\Omega$ being $C^1$. Let $p \in [1, \infty]$ then there exists a constant $C > 0$ depending only on $d, p$ and $\Omega$, such that*

$$\|u - u_\Omega\|_{L^p(\Omega)} \leq C \|\nabla u\|_{L^p(\Omega)}, \quad \forall u \in W^{1,p}(\Omega), \tag{B.6}$$

*where $u_\Omega := |\partial\Omega|^{-1} \int_{\partial\Omega} Tu ds$.*

**Proof.** Let $u \in W^{1,p}(\Omega)$ and let $u_\Omega := |\partial\Omega|^{-1} \int_{\partial\Omega} Tu ds$. The proof will be by contradiction, so assume that (B.6) is not true, then there exists a sequence $\{u_k\}_{k\in\mathbb{N}} \subset W^{1,p}(\Omega)$ satisfying

$$\|u_k - (u_k)_\Omega\|_{L^p(\Omega)} > k \|\nabla u_k\|_{L^p(\Omega)}, \quad k \in \mathbb{N}. \tag{B.7}$$

[1] en.wikipedia.org/wiki/Sobolev_space

Now define

$$v_k := \frac{u_k - (u_k)_\Omega}{\|u_k - (u_k)_\Omega\|_{L^p(\Omega)}}.$$

Then we have $\|v_k\|_{L^p(\Omega)} = 1, \ \ k \in \mathbb{N}$, and

$$(v_k)_\Omega = \frac{1}{\|u_k - (u_k)_\Omega\|_{L^p(\Omega)}}((u_k)_\Omega - |\partial\Omega|^{-1}|\partial\Omega|(u_k)_\Omega) = 0, \ \ k \in \mathbb{N}. \tag{B.8}$$

Furthermore, by (B.7) then

$$\begin{aligned} \|\nabla v_k\|_{L^p(\Omega)} &= \left\| \nabla \left( \frac{u_k}{\|u_k - (u_k)\|_{L^p(\Omega)}} - \frac{(u_k)_\Omega}{\|u_k - (u_k)\|_{L^p(\Omega)}} \right) \right\|_{L^p(\Omega)} \\ &= \frac{\|\nabla u_k\|_{L^p(\Omega)}}{\|u_k - (u_k)\|_{L^p(\Omega)}} < \frac{1}{k}, \ \ k \in \mathbb{N}. \end{aligned} \tag{B.9}$$

By the remark on p. 289 in Evans [20] there exists a subsequence $\{v_{k_j}\}_{j\in\mathbb{N}} \subseteq \{v_k\}_{k\in\mathbb{N}}$ and $v \in L^p(\Omega)$ such that $\lim_{j\to\infty}\|v_{k_j} - v\|_{L^p(\Omega)} = 0$, thus $\|v\|_{L^p(\Omega)} = 1$ and by (B.9) then $\|\nabla v\|_{L^p(\Omega)} = 0$, i.e. $v \in W^{1,p}(\Omega)$. Thus $\nabla v = 0$ *a.e.*, so $v = C$ *a.e.* for some constant $C$.

Due to the trace theorem (Theorem B.4) then $Tv = C$ since $\tilde{v} := C$ equals $v$ almost everywhere. By (B.8) then $v_\Omega = 0$ which implies that $C = 0$ so $v = 0$ *a.e.* which is a *contradiction* with $\|v\|_{L^p(\Omega)} = 1$. □

# Appendix C

# Various Theorems and Lemmas

The following two identities, also known as Green's first and second identity, are part of any introductory PDE book with respect for itself for instance Asmar [16], Strauss [30] or Evans [20], however, the slightly more general version presented in Theorem C.2 of the second Green's identity is usually shown with $\epsilon = 1$, however, it is a simple corollary to the divergence theorem on the field $F = \psi\epsilon\nabla\phi - \phi\epsilon\nabla\psi$.

**Theorem C.1 (Green's First Identity)** *Let $\Omega$ be a connected region in $\mathbb{R}^d$ with outwards pointing normal $n$, let $\phi \in C^2(\Omega)$ and $\psi \in C^1(\Omega)$ then*

$$\int_\Omega (\psi\nabla^2\phi + \nabla\psi \cdot \nabla\phi) dx = \oint_{\partial\Omega} \psi \frac{\partial\phi}{\partial n} ds.$$

**Theorem C.2 (Green's Second Identity)** *Let $\Omega$ be a connected region in $\mathbb{R}^d$ with outwards pointing normal $n$, let $\phi, \psi \in C^2(\Omega)$ and $\epsilon \in C^1(\Omega)$ then*

$$\int_\Omega \left[\psi\nabla \cdot (\epsilon\nabla\phi) - \phi\nabla \cdot (\epsilon\nabla\psi)\right] dx = \oint_{\partial\Omega} \epsilon \left(\psi\frac{\partial\phi}{\partial n} - \phi\frac{\partial\psi}{\partial n}\right) ds.$$

The following theorem is a fundamental, but very powerful result from functional analysis.

**Theorem C.3 (Lax-Milgram Theorem)** *Let $H$ be a Hilbert space with norm $\|\cdot\|$, let $B : H \times H \to \mathbb{R}$ be bilinear, and let $L \in H'$ (where $H'$ is the continuous dual space of $H$). Assume that there exists constants $C_1, C_2 > 0$ such that* *(i)* $|B(u,v)| \leq C_1\|u\|\|v\|, \;\; \forall u, v \in H,$

*(ii)* $B(u,u) \geq C_2\|u\|^2, \;\; \forall u \in H.$

*Then there exists a unique* $u^* \in H$ *such that*

$$B(u^*, v) = \langle L, v\rangle, \;\; \forall v \in H,$$

*where* $\langle L, v\rangle$ *denotes the dual pairing of* $L$ *and* $v$*.*

The following definition of Gâteuax and Fréchet derivatives, and the following theorem, are taken from Zorn [14], Griffel [22][1]. It is possible to define higher order Gâteaux and Fréchet derivatives using different directions, however this will not be applied in this thesis, so the notation is simplified by using the same direction in every order of the derivatives.

**Definition C.4 (Gâteaux and Fréchet Derivatives)** *Let* $X$ *and* $Y$ *be normed spaces, with* $U \subseteq X$ *being open and* $F : X \to Y$*.*

*(i)* $F$ *is said to be Gâteaux differentiable at* $x \in U$ *with derivative* $F'_x : X \to Y$*, if the following limit exists:*

$$F'_x(\eta) := \lim_{\epsilon \to 0} \frac{F(x + \epsilon\eta) - F(x)}{\epsilon} = \frac{d}{d\epsilon} F(x + \epsilon\eta)|_{\epsilon=0}, \;\; \forall \eta \in X.$$

*A higher order Gâteaux derivative in the direction* $\eta$ *can be determined similarly by*

$$F_x^{(n)}(\eta) := \frac{d}{d\epsilon_1} \cdots \frac{d}{d\epsilon_n} F(x + \sum_{k=1}^{n} \epsilon_k \eta)|_{\epsilon_1 = \cdots = \epsilon_n = 0}.$$

*(ii)* $F$ *is said to be Fréchet differentiable at* $x \in U$ *with derivative* $F'_x = L_x$*, if there exists a bounded linear operator* $L_x : X \to Y$ *such that*

$$\lim_{\|\eta\|_X \to 0} \frac{\|F(x+\eta) - F(x) - L_x\eta\|_Y}{\|\eta\|_X} = 0.$$

[1] the remainder term for Theorem C.5 is from planetmath.org/encyclopedia/taylorpolynomialsinbanachspaces.html

*If $F$ is Fréchet differentiable then it is also Gâteaux differentiable, and the derivatives coincide. The converse is only true if the Gâteaux derivative is linear and bounded.*

**Theorem C.5 (Taylor Expansions in Gâteaux Derivatives)** *Suppose that $F : X \to Y$ with $X, Y$ being Banach spaces, and that $F$ is $k$ times continuously Gâteaux differentiable. Let $U \subseteq X$ be an open subset such that $x + t\eta \in U$ for $t \in [0, 1]$, then*

$$F(x+\eta) = F(x) + F'_x(\eta) + \cdots + \frac{1}{(k-1)!} F_x^{(k-1)}(\eta) + R_x^k(\eta),$$

*with the remainder term being*

$$R_x^k(\eta) = \frac{1}{(k-1)!} \int_0^1 (1-t)^{k-1} F_{x+t\eta}^{(k)}(\eta) dt.$$

**Theorem C.6** *Let $\{f_j\}$ be a sequence in $L^p(\mathcal{X}, \mathbb{E}, \mu)$ for some measure space $(\mathcal{X}, \mathbb{E}, \mu)$.*

*(i) If $\{f_j\}$ converges to $f \in L^p(\mathcal{X}, \mathbb{E}, \mu)$ in the $L^p$-sense, then there is a subsequence $\{f_{j_k}\}$ which converges pointwise a.e. to $f$.*

*(ii) If $\{f_j\}$ converges to $g_1 \in L^p(\mathcal{X}, \mathbb{E}, \mu)$ in the $L^p$-sense, and pointwise a.e. to $g_2$, then $g_1 = g_2$ a.e..*

*Appendix D*

# Source Code of Implementation

This appendix will contain the source code for the implemented algorithms for solving the inverse EIT problem. The language used is Python and the main library used is `dolfin` from the FEniCS project [31]. It should be noted that in Python the use of backslash \ denotes that the code continues on the next line, and it is used to make code fit on the page.

## D.1 Shared Functions

This section will contain the source code of the file `SharedEITFunctions.py`, that contains functions that are shared by the sparsity and TV solvers.

```python
"""
Functions that are shared between the different EIT solvers
"""

from dolfin import *
import numpy as np
import shelve # used to easily load/save stuff from/to files

"""
Solves the Forward EIT problem with Neumann condition:
      -div(sigma*grad(u)) = 0
              sigma*du/dn = g
          int trace(u) ds = 0

Input:
    Mesh 'mesh'   mesh on which the problem is solved
Function 'sigma'  electrical conductivity
Function 'g'      Neumann-data
```

```

Output:
Function 'u'      forward solution
"""
def FSolver(mesh,sigma,g):
    # Define function space
    V = sigma.function_space()
    R = FunctionSpace(mesh, "R", 0)
    W = V * R

    # Define trial and test functions
    (u, c) = TrialFunctions(W)
    (v, d) = TestFunctions(W)

    # Define forms and langrange multipliers
    LangMult = (c*v + u*d)*ds # forces int u ds = int v ds = 0
    B = sigma*inner(grad(u), grad(v))*dx + LangMult
    L = g*v*ds

    # Compute solution
    w = Function(W)
    solve(B == L, w)
    (u, c) = w.split(deepcopy = True)

    return u



"""
Computes the discrepancy gradient \nabla J(sigma)

Input:
    Mesh 'mesh'    mesh on which the problem is solved
Function 'sigma'   electrical conductivity
Function 'g'       Neumann-data
Function 'phi'     Dirichlet-data
Function 'F'       forward solution F_g(sigma)
Function 'Tg'      threshold function for partial data,
                   default = Constant(1.0)

Output:
Function 'Jprime'  gradient of discrepancy
"""
def Jsolver(mesh,sigma,g,phi,F,Tg = Constant(1.0)):
    # Solve adjoint problem
    adjointdata = (F-phi)*Tg
    utilde = FSolver(mesh,sigma,adjointdata)
    Jprime = -inner(grad(utilde),grad(F))

    return Jprime


"""
Computes the Sobolev gradient \nabla_s J(sigma)

Input:
    Mesh 'mesh'    mesh on which the problem is solved
Function 'Jprime'  usual gradient of discrepancy

```

```
Output:
Function 'Js'      sobolev gradient of discrepancy
"""
def Jsobolev(mesh,Jprime):
    # Dirichlet BC
    V = FunctionSpace(mesh,"CG",1)
    def Js_boundary(x, on_boundary):
        return on_boundary

    bc = DirichletBC(V,Constant(0.0),Js_boundary)
    # Define variational problem
    Js = TrialFunction(V)
    v = TestFunction(V)
    B = inner(grad(Js),grad(v))*dx + Js*v*dx
    L = Jprime*v*dx
    # Solve
    Js = Function(V)
    solve(B == L,Js,bc)

    return Js



"""
A simple implementation of the H1 inner product, used to shorten
code length.
"""
def H1IP(u,v):
    return assemble((u*v + inner(grad(u),grad(v)))*dx)


"""
Computes a list of arrays, giving a correspondance between nodes
and cells in the mesh. nodecells[k] will be an array of cell-id's
for node 'k'.

Input:
Mesh 'mesh'       mesh on which the problem is solved

Output:
List 'nodecells'  node-to-cell connectivity
"""
def node2cell(mesh):
    N = mesh.num_vertices()
    C = mesh.cells()
    CN = len(C)
    nodecells = list([] for k in xrange(N))
    for k in xrange(CN):
        c = C[k]
        nodecells[c[0]].append(k)
        nodecells[c[1]].append(k)
        nodecells[c[2]].append(k)

    return nodecells



"""
Computes an array containing the L1-norm of the FEM basis
```

```
functions.

Input:
Mesh ’mesh’   mesh on which the problem is solved

Output:
List ’A’      list of L1-norms, in order of nodes in mesh
"""
def BasisL1Norm(mesh):
    # get node-cell connectivity
    nodecells = node2cell(mesh)
    V = FunctionSpace(mesh,’CG’,1)
    N = mesh.num_vertices()
    A = np.zeros(N) # for storing the L1-norms
    v_array = np.zeros(N)
    v = Function(V)
    for k in xrange(N):
        # get cells connected to node ’k’ and the ’k’th basis
        # function
        nc = nodecells[k]
        v_array[k] = 1.0
        v.vector()[:] = v_array
        # get domain for cells connected to node ’k’ for the
        # support of the ’k’th basis function
        domain = CellFunction("uint",mesh)
        domain.set_all(0)
        for j in xrange(len(nc)):
            domain.set_value(nc[j],1)
        # only integrate on the support of basis function for
        # efficiency
        dx = Measure("dx")[domain]
        A[k] = assemble(v*dx(1))
        v_array[k] = 0.0

    return A



"""
Simply makes a version of arctan2 that returns in range [0,2*pi]
instead of [-pi,pi].
"""
def arctan3(y,x):
    a = np.arctan2(y,x)
    N = np.size(a)
    if N == 1:
        if a < 0:
            a += 2*pi
    else:
        for i in xrange(N):
            if a[i] < 0:
                a[i] += 2*pi

    return a



"""
Constructs expression for ’g’ giving a scaled and translated
```

```
version of either a sine or cosine, i.e.
g = cos(2*pi/(theta2-theta1)*N*(theta-theta1)), and truncated
such that it is zero outside [theta1,theta2].

Input:
   Boolean 'iscos'              True for cosine, False for sine
  Constant 'N'                  number of periods
 Constants 'theta1','theta2'    determines the range
                                [theta1,theta2], default is
                                [0,2*pi]

Output:
Expression 'g'
"""
def g_exp(iscos,N,theta1 = 0.0, theta2 = 2*pi):
    class g(Expression):
        def eval(self, values, x):
            theta = arctan3(x[1],x[0])
            if theta < theta1 or theta > theta2:
                values[0] = 0
            elif iscos == True:
                values[0] = np.cos(2*pi/(theta2-theta1)*N* \
                            (theta-theta1))
            else:
                values[0] = np.sin(2*pi/(theta2-theta1)*N* \
                            (theta-theta1))

    return g()



"""
Function that thresholds the boundary at a range [theta1,theta2],
default is [0,2*pi]. Only works for unit circle domain.

Input:
     Mesh 'mesh'             mesh on which the problem is solved
Constants 'theta1','theta2'  default values are 0.0 and 2*pi

Output:
Function 'Tg'                threshold function
"""
def ThresholdFunction(mesh,theta1 = 0.0,theta2 = 2*pi):
    V = FunctionSpace(mesh,'CG',1)
    Tg = Function(V)
    N = mesh.num_vertices()
    coords = mesh.coordinates()
    Tg_array = np.zeros(N)
    for k in xrange(N):
        c = coords[k]
        # check if on boundary
        if abs(c[0]**2+c[1]**2-1.0) < 1E-5:
            a = arctan3(c[1],c[0])
            if a >= theta1 and a <= theta2:
                Tg_array[k] = 1.0

    Tg.vector()[:] = Tg_array
    return Tg

```

```

"""
Obtains the boundary for the unit circle case, and orders it in
terms of the angle.

Input:
    Mesh 'mesh'    mesh on which the problem is solved
Function 'u'       function from which the boundary is extracted

Output:
    List 'ang'     angular values
    List 'val'     amplitude
"""
def GetBoundary(mesh,u):
    u_array = u.vector().array()
    coords = mesh.coordinates()
    angles = arctan3(coords[:,1],coords[:,0])
    T = []
    N = len(angles)
    for i in xrange(N):
        c = coords[i]
        # check if on boundary
        if abs(c[0]**2+c[1]**2-1.0) < 1E-5:
            T.append((angles[i],u_array[i]))

    dtype = [('angles','double'),('uval','double')]
    A = np.array(T,dtype)
    A = np.sort(A,order='angles') # order points correctly
    ang = A.getfield('double',0)
    val = A.getfield('double',8)

    return ang,val


"""
Simulate data for the EIT-problem using cosines and sines on the
boundary, on a mesh for the unit disk, and with given noise
level.

To avoid inverse crimes the forward problem is solved on a very
fine mesh, and then interpolated onto the given coarser mesh.

For 'k' datasets of cosines/sines use an array 'cos_array'/
'sin_array' of length 'k', containing the number of periods for
the specific datasets.

'sigma', exact data, noisy data, noiselvl, 'cos_array',
'sin_array', 'theta1', and 'theta2' will be saved to the file
given by 'filename'. Use the library 'shelve' to retrieve it.

The functions will be saved as arrays.

Input:
       Expression 'sigma'            the exact conductivity
             Mesh 'mesh'             coarse mesh
             Mesh 'fine_mesh'        fine mesh used to avoid
                                     inverse crime
         Constant 'noiselvl'         the noise level for the
                                     additive white noise, in
                                     terms of L^infty
List of constants 'cos_array'        Constants n that determines
                                     the periods of the cosine
                                     Neumann-data
List of constants 'sin_array'        Constants n that determines
                                     the periods of the sine
                                     Neumann-data
           String 'filename'         name for the file where the
                                     data is stored
        Constants 'theta1','theta2'  scale and translate the
                                     Neumann-data such that it is
                                     supported in [theta1,theta2],
                                     default is [0,2*pi])
"""
def SimulateEITData(sigma,mesh,fine_mesh,noiselvl,cos_array,\
                    sin_array,filename,theta1=0.0,theta2=2*pi):
    # allow interpolation between different meshes
    parameters["allow_extrapolation"] = True
    V = FunctionSpace(mesh,'CG',1)
    Vfine = FunctionSpace(fine_mesh,'CG',1)
    sigmafine = project(sigma,Vfine)
    sigmafun = project(sigma,V)
    NumVert = mesh.num_vertices()
    u_exact_array = []
    u_noise_array = []
    maxval = 0
    for j in xrange(2):
        if j == 0:
            iscos = True
            n_array = cos_array
        else:
            iscos = False
            n_array = sin_array
        k = len(n_array)
        for i in xrange(k):
            # solve and save exact data
            g = g_exp(iscos,n_array[i],theta1,theta2)
            u_fine = FSolver(fine_mesh,sigmafine,g)
            u = Function(V)
            # down sample to coarse mesh
            u = interpolate(u_fine,V)
            u_array = u.vector().array()
            u_exact_array.append(u_array)
            # get max value of boundary
            ang,val = GetBoundary(mesh,u)
            maxval = max([abs(val).max(),maxval])

    # now adding noise according to
    # noiselvl*max_k max_x u(on bdry)
    for i in xrange(len(u_exact_array)):
        noise = noiselvl*maxval*np.random.normal(0,1,NumVert)
        u_noise_array.append(u_exact_array[i] + noise)

    # save it all to a file
    data = shelve.open(filename)
    data["noiselvl"] = noiselvl
    data["sigma"] = sigmafun.vector().array()
    data["u_exact_array"] = u_exact_array
    data["u_noise_array"] = u_noise_array
    data["cos_array"] = cos_array
    data["sin_array"] = sin_array
    data["theta1"] = theta1
    data["theta2"] = theta2
    data.close()
```

## D.2 Sparsity Solver

This section will contain the source code of the file EITSparsitySolver.py that contains the implementation of the sparsity solver.

```
"""
Source code for the sparsity solver for EIT
"""

from dolfin import *
import numpy as np
from SharedEITFunctions import *


"""
Computes Tikhonov functional Psi(delta_sigma) using the finite
element basis.

Input:
               Mesh 'mesh'          mesh on which the problem is
                                    solved
   List of Constant 'alpha_array'   reg. parameters
           Function 'sigma0'        background
           Function 'delta_sigma'   inclusion
List of expressions 'g_array'       Neumann-datasets
       List of data 'phi_array'     Dirichlet-datasets
  List of functions 'F_array'       used to store forward solution
           Function 'Tg'            threshold function for partial
                                    data, default = Constant(1.0)
  List of constants 'w_array'       optional weights for
                                    discrepancy, default = 1

Output:
             Double 'Psi'           evaluation of the Tikhonov
                                    functional
"""
def PsiEval(mesh,alpha_array,sigma0,delta_sigma,g_array,\
            phi_array,F_array,Tg = Constant(1.0),w_array=None):
    print "Evaluating Tikhonov functional"
    N = len(g_array)
    V = sigma0.function_space()
    sigma = Function(V)
    sigma.vector()[:] = sigma0.vector().array() + \
                        delta_sigma.vector().array()
    # Set default values of weights if not given
    if w_array == None:
        w_array = np.ones(N)

    # Find discrepancy
    J = 0
    for k in xrange(N):
        # Solve forward problem
        u = FSolver(mesh,sigma,g_array[k])
        F_array[k] = u
        # Evaluate part of discrepancy
        udiff = Function(V)
        udiff.vector()[:] = u.vector().array() - \
                            phi_array[k].vector().array()
        J = J + w_array[k]*assemble(udiff*udiff*Tg*ds)
    J = 0.5*J

    # Compute l1 penalty-term
    penalty = abs(alpha_array*delta_sigma.vector().array()).sum()

    return J + penalty




"""
Soft shrinkage operator.

Input:
         Function 'u'
List of constants 'beta_array'   parameters corresponding to each
node

Output:
         Function 'v'            the soft shrinkage of 'u'
"""
def SoftShrinkage(u,beta_array):
    V = u.function_space()
    v = Function(V)
    # makes a copy of the array from u (not the pointer to the
    # actual array in u)
    v_array = u.vector().array()
    # perform soft shrinkage at nodes
    for k in xrange(len(v_array)):
        vk = v_array[k]
        v_array[k] = np.sign(vk)*np.max([abs(vk) - \
                     beta_array[k],0])
    v.vector()[:] = v_array

    return v



"""
Computes step size for iteration, and returns the soft shrinkage
for that step.

Input:
           Function 'sigma0'            background
               Mesh 'mesh'              mesh on which the problem
                                        is solved
           Function 'delta_sigma'       inclusion
           Function 'delta_sigma_prev'  inclusion from previous
                                        iterate
           Function 'Js'                sobolev gradient of J
           Function 'Js_prev'           sobolev gradient of J
                                        from previous iterate
  List of functions 'psi_array'         evaluations of Psi(sigma)
                                        up to current iteration
  List of functions 'g_array'           Neumann-datasets
  List of functions 'phi_array'         Dirichlet-datasets
  List of functions 'F_array'           used to store forward
                                        solution
               List 'param'             parameters [s_min,s_max,
                                        M,tau,q,alpha_array,
                                        s_stop,I_max,beta_array]
                                        in that exact order
           Function 'Tg'                threshold function for
                                        partial data, default =
                                        Constant(1.0)
  List of constants 'w_array'           optional weights for
                                        discrepancy, default = 1

Output:
             Double 's'                 step size
           Function 'soft_step'         next iterate
"""
def StepSize(sigma0,mesh,delta_sigma,delta_sigma_prev,Js,\
             Js_prev,psi_array,g_array,phi_array,F_array,param,\
             Tg=Constant(1.0),w_array=None):
    # unpacking parameters
    s_min = param[0]
    s_max = param[1]
    M = param[2]
    tau = param[3]
    q = param[4]
    alpha_array = param[5]
    beta_array = param[8]
    s_stop = param[6]
    V = sigma0.function_space()

    # we want to look back M steps, but initially we only have N
    N = len(psi_array)
    if N < M:
        M = N

    # initialize step, if N = 1 then we are at first iteration
    if N == 1:
        # at first iteration we have no step size estimate
        s = s_max
    else:
        diff_sigma = Function(V)
        diff_Js = Function(V)
        diff_sigma.vector()[:] = delta_sigma.vector().array() - \
                                delta_sigma_prev.vector().array()
        diff_Js.vector()[:] = Js.vector().array() - \
                              Js_prev.vector().array()
        IPsigmaJs = H1IP(diff_sigma,diff_Js)
        IPsigma2 = H1IP(diff_sigma,diff_sigma)
        # avoid division with zero in special cases
        if IPsigma2 == 0.0:
            s = s_min
        elif IPsigmaJs == 0.0:
            s = s_max
        else:
            s = H1IP(diff_sigma,diff_sigma)/ \
                H1IP(diff_sigma,diff_Js)
        # make initial step size lie in [s_min, s_max]
        s = np.min([np.max([s_min, s]), s_max])

    # check weak monotonicity
    accept = False
    psi_max = np.array(psi_array[N-M:N]).max()
    delta_step = Function(V)
    while accept == False:
        delta_step.vector()[:] = delta_sigma.vector().array() - \
                                 s*Js.vector().array()
        soft_step = SoftShrinkage(delta_step,s*beta_array)
        diff = Function(V)
        diff.vector()[:] = soft_step.vector().array() - \
                           delta_sigma.vector().array()
        upperbound = psi_max - 0.5*tau/s*H1IP(diff,diff)
        psi_step = PsiEval(mesh,alpha_array,sigma0,soft_step,\
                           g_array,phi_array,F_array,Tg,w_array)
        print "step size: ", s
        # if weak monotonicity holds we accept the step
        if psi_step <= upperbound or s <= s_stop:
            accept = True
        # otherwise we decrease s by factor q
        else:
            print "Reducing step size"
            s = s/q
    # append psi evaluated at next step
    print "Psi: ", psi_step
    psi_array.append(psi_step)
    return s, soft_step



"""
Iterative solver that minimizes the Tikhonov functional
Psi(delta_sigma), and produces a candidate for the inverse
EIT-problem for the continuum model:

      -div(sigma*grad(u)) = 0
              sigma*du/dn = g
                 int u ds = 0

The method supports multiple datasets (g_k,phi_k), where g_k is
the Neumann-data and phi is the measured Dirichlet-data
corresponding to trace(u)+noise.

The FEM elements used are Lagrange elements of degree 1.

If 'alpha_array' has length 1, then it is extended to an array of
correct length with this value at all entries.

Input:
         Function 'sigma0'      background
             Mesh 'mesh'        mesh on which the problem is
                                solved
List of functions 'g_array'     Neumann-datasets
List of functions 'phi_array'   Dirichlet-datasets
```

```
             List 'param'       parameters [s_min,s_max,M,tau,q,
                                alpha_array,s_stop,I_max], in
                                this exact order
                                s_min,s_max : bounds for initial
                                              step sizes
                                      M,tau : parameters for
                                              weak monotonocity
                                          q : factor for reducing
                                              step sizes
                                alpha_array : reg. parameters
                                     s_stop : stopping criteria
                                              for step sizes
                                      I_max : maximum no. of
                                              iterations
         Function 'Tg'          threshold function for partial
                                data, default = Constant(1.0)
List of constants 'w_array'     optional weights for discrepancy,
                                default = 1

Output:
         Function 'sigma'       final solution
   List of double 'psi_array'   evaluations for Psi for each
                                iteration
   List of double 's_array'     step sizes for each iteration
   List of arrays 'sigma_array' solution in each iteration
"""
def SparsitySolver(sigma0,mesh,g_array,phi_array,param,\
                   Tg = Constant(1.0),w_array = None):
    # initialize
    K = len(g_array)
    N = mesh.num_vertices()
    # Set default values of weights if not given
    if w_array == None:
        w_array = np.ones(K)
    stop = False
    alpha_array = param[5]
    # Make sure that alpha_array has correct length
    if np.size(alpha_array) == 1:
        alpha_array = alpha_array*np.ones(N)
        param[5] = alpha_array
    # make beta_array, given by alpha_array weighted by FEM
    # basis L1-norm
    beta_array = alpha_array/BasisL1Norm(mesh)
    param.append(beta_array)
    s_stop = param[6]
    I_max = param[7]
    V = sigma0.function_space()
    sigma = Function(V)
    delta_sigma = Function(V)
    delta_sigma_prev = Function(V)
    Js = Function(V)
    Js_prev = Function(V)
    psi_array = []
    s_array = []
    sigma0array = sigma0.vector().array()
    sigma_array = []
    # make list for storing forward solution at each iteration
    F_array = list([] for k in xrange(K))
    # evaluate Psi at start point
    psi0 = PsiEval(mesh,alpha_array,sigma0,delta_sigma,g_array,\
                   phi_array,F_array,Tg,w_array)
    psi_array.append(psi0)
    i = 0
    # begin iterating
    while stop == False:
        i = i+1
        print "Iteration: ", i
        sigma.vector()[:] = sigma0array + \
                            delta_sigma.vector().array()
        sigma_array.append(sigma.vector().array())
        # computing discrepancy gradient
        Jprime = Function(V)
        for k in xrange(K):
            Jprime = Jprime + w_array[k]*Jsolver(mesh,sigma,\
                                        g_array[k],phi_array[k],\
                                        F_array[k],Tg)
        # computing Sobolev gradient
        Js_prev.vector()[:] = Js.vector().array()
        Js = Jsobolev(mesh,Jprime)
        # compute step size and next step
        s, soft_step = StepSize(sigma0,mesh,delta_sigma,\
                                delta_sigma_prev,Js,Js_prev,\
                                psi_array,g_array,phi_array,\
                                F_array,param,Tg,w_array)
        s_array.append(s)
        delta_sigma_prev.vector()[:] = delta_sigma.vector(). \
                                       array()
        delta_sigma.vector()[:] = soft_step.vector().array()
        # check stopping conditions
        if i == I_max or s <= s_stop:
            stop = True
    # compute final sigma
    sigma.vector()[:] = sigma0array + \
                        delta_sigma.vector().array()
    sigma_array.append(sigma.vector().array())
    return sigma, psi_array, s_array, sigma_array
```

## D.3 Total Variation Solver

This section will contain the source code of the file EITTVSolver.py that contains the implementation of the TV solver.

```
"""
Source code for the total variation solver for EIT
"""

from dolfin import *
import numpy as np
from SharedEITFunctions import *

"""
Computes Tikhonov functional Psi(sigma) using the finite element
basis.

Input:
```

```
               Mesh 'mesh'         mesh on which the problem is
                                   solved
          Constants 'beta,c'       reg. parameters
           Function 'sigma0'       background
           Function 'delta_sigma'  inclusion
List of expressions 'g_array'      Neumann-datasets
       List of data 'phi_array'    Dirichlet-datasets
  List of functions 'F_array'      used to store forward solution
           Function 'Tg'           threshold function for partial
                                   data, default = Constant(1.0)
  List of constants 'w_array'      optional weights for
                                   discrepancy, default = 1

Output:
             Double 'Psi'          evaluation of the Tikhonov
                                   functional
"""
def PsiEval(mesh,beta,c,sigma0,delta_sigma,g_array,phi_array,\
            F_array,Tg = Constant(1.0),w_array=None):
    print "Evaluating Tikhonov functional"
    N = len(g_array)
    V = sigma0.function_space()
    sigma = Function(V)
    sigma.vector()[:] = sigma0.vector().array() + \
                        delta_sigma.vector().array()
    # Set default values of weights if not given
    if w_array == None:
        w_array = np.ones(N)

    # Find discrepancy
    J = 0
    for k in xrange(N):
        # Solve forward problem
        u = FSolver(mesh,sigma,g_array[k])
        F_array[k] = u
        # Evaluate part of discrepancy
        udiff = Function(V)
        udiff.vector()[:] = u.vector().array() - \
                            phi_array[k].vector().array()
        J = J + w_array[k]*assemble(udiff*udiff*Tg*ds)
    J = 0.5*J

    # Compute TV penalty-term
    penalty = beta*assemble(sqrt(inner(grad(delta_sigma), \
              grad(delta_sigma))+Constant(c))*dx)

    return J + penalty



"""
Computes the TV step given a parameter 'beta' or s*beta as in the
algorithm.

Input:
 Function 'delta_sigma'  current iterate
 Function 'gamma'        delta_sigma-s_i*nabla_s J(sigma_i)
Constants  'beta,c'      reg. parameters

```

```
Output:
Function 'TV_step'       the new TV step
"""
def TVStep(delta_sigma,gamma,beta,c):
    V = delta_sigma.function_space()
    def bdry(x, on_boundary):
        return on_boundary

    bc = DirichletBC(V,Constant(0.0),bdry)
    v = TestFunction(V)
    u = TrialFunction(V)
    # make variables to shorten code
    f = inner(grad(delta_sigma),grad(delta_sigma))+Constant(c)
    f1 = Constant(beta)/sqrt(f)
    f2 = f1/f
    # setting up the forms
    B = u*v*dx + inner((Constant(1)+f1)*grad(u),grad(v))*dx - \
        f2*inner(grad(delta_sigma),grad(u))* \
        inner(grad(delta_sigma),grad(v))*dx
    L = gamma*v*dx + inner((grad(gamma)-f2* \
        inner(grad(delta_sigma),grad(delta_sigma))* \
        grad(delta_sigma)),grad(v))*dx
    # solving
    TV_step = Function(V)
    solve(B == L,TV_step,bc)

    return TV_step






"""
Computes step size for iteration, and returns the next TV step.

Input:
           Function 'sigma0'            background
               Mesh 'mesh'              mesh on which the problem
                                        is solved
           Function 'delta_sigma'       inclusion
           Function 'delta_sigma_prev'  inclusion from previous
                                        iterate
           Function 'Js'                sobolev gradient of J
           Function 'Js_prev'           sobolev gradient of J
                                        from previous iterate
  List of functions 'psi_array'         evaluations of Psi(sigma)
                                        up to current iteration
  List of functions 'g_array'           Neumann-datasets
  List of functions 'phi_array'         Dirichlet-datasets
  List of functions 'F_array'           used to store forward
                                        solution
               List 'param'             parameters [s_min,s_max,
                                        M,tau,q,beta_array,
                                        s_stop,I_max,c], in that
                                        exact order
           Function 'Tg'                threshold function for
                                        partial data,
                                        default = Constant(1.0)
  List of constants 'w_array'             optional weights for
                                          discrepancy, default = 1

Output:
             Double 's'                   step size
           Function 'TV_step'             next iterate
"""
def StepSize(sigma0,mesh,delta_sigma,delta_sigma_prev,Js,\
             Js_prev,psi_array,g_array,phi_array,F_array,param,\
             Tg=Constant(1.0),w_array=None):
    # unpacking parameters
    s_min = param[0]
    s_max = param[1]
    M = param[2]
    tau = param[3]
    q = param[4]
    beta = param[5]
    c = param[8]
    s_stop = param[6]
    V = sigma0.function_space()

    # we want to look back M steps, but initially we only have N
    N = len(psi_array)
    if N < M:
        M = N

    # initialize step, if N = 1 then we are at first iteration
    if N == 1:
        # at first iteration we have no step size estimate
        s = s_max
    else:
        diff_sigma = Function(V)
        diff_Js = Function(V)
        diff_sigma.vector()[:] = delta_sigma.vector().array() - \
                                delta_sigma_prev.vector().array()
        diff_Js.vector()[:] = Js.vector().array() - \
                              Js_prev.vector().array()
        IPsigmaJs = H1IP(diff_sigma,diff_Js)
        IPsigma2 = H1IP(diff_sigma,diff_sigma)
        # avoid division with zero in special cases
        if IPsigma2 == 0.0:
            s = s_min
        elif IPsigmaJs == 0.0:
            s = s_max
        else:
            s = H1IP(diff_sigma,diff_sigma)/ \
                H1IP(diff_sigma,diff_Js)
        # make initial step size lie in [s_min, s_max]
        s = np.min([np.max([s_min, s]), s_max])

    # check weak monotonicity
    accept = False
    psi_max = np.array(psi_array[N-M:N]).max()
    delta_step = Function(V)
    while accept == False:
        delta_step.vector()[:] = delta_sigma.vector().array() - \
                                 s*Js.vector().array()
        TV_step = TVStep(delta_sigma,delta_step,s*beta,c)
        diff = Function(V)
        diff.vector()[:] = TV_step.vector().array() - \
                           delta_sigma.vector().array()
        upperbound = psi_max - 0.5*tau/s*H1IP(diff,diff)
        psi_step = PsiEval(mesh,beta,c,sigma0,TV_step,g_array,\
                           phi_array,F_array,Tg,w_array)
        print "step size: ", s
        # if weak monotonicity holds we accept the step
        if psi_step <= upperbound or s <= s_stop:
            accept = True
        # otherwise we decrease s by factor q
        else:
            print "Reducing step size"
            s = s/q
    # append psi evaluated at next step
    print "Psi: ", psi_step
    psi_array.append(psi_step)
    return s, TV_step




"""
Iterative solver that minimizes the Tikhonov functional
Psi(sigma), and produces a candidate for the inverse EIT-problem
for the continuum model:

      -div(sigma*grad(u)) = 0
              sigma*du/dn = g
                 int u ds = 0

The method supports multiple datasets (g_k,phi_k), where g_k is
the Neumann-data and phi is the measured Dirichlet-data
corresponding to trace(u)+noise.

The FEM elements used are Lagrange elements of degree 1.

Input:
         Function 'sigma0'      background
             Mesh 'mesh'        mesh on which the problem is
                                solved
List of functions 'g_array'     Neumann-datasets
List of functions 'phi_array'   Dirichlet-datasets
             List 'param'       parameters [s_min,s_max,M,tau,q,
                                beta_array,s_stop,I_max,c], in
                                this exact order
                                s_min,s_max : bounds for initial
                                              step sizes
                                      M,tau : parameters for weak
                                              monotonocity
                                          q : factor for reducing
                                              step sizes
                               beta_array,c : reg. parameters
                                     s_stop : stopping criteria
                                              for step sizes
                                      I_max : maximum no. of
                                              iterations
         Function 'Tg'          threshold function for partial
                                data, default = Constant(1.0)
List of constants 'w_array'     optional weights for discrepancy,
```

```python
                                default = 1

Output:
         Function 'sigma'       final solution
   List of double 'psi_array'   evaluations for Psi for each
                                iteration
   List of double 's_array'     step sizes for each iteration
   List of arrays 'sigma_array' solution in each iteration
"""
def TVSolver(sigma0,mesh,g_array,phi_array,param,\
             Tg = Constant(1.0),w_array = None):
    # initialize
    K = len(g_array)
    # Set default values of weights if not given
    if w_array == None:
        w_array = np.ones(K)
    stop = False
    beta = param[5]
    c = param[8]
    s_stop = param[6]
    I_max = param[7]
    V = sigma0.function_space()
    sigma = Function(V)
    delta_sigma = Function(V)
    delta_sigma_prev = Function(V)
    Js = Function(V)
    Js_prev = Function(V)
    psi_array = []
    s_array = []
    sigma0array = sigma0.vector().array()
    sigma_array = []
    # make list for storing forward solution at each iteration
    F_array = list([] for k in xrange(K))
    psi0 = PsiEval(mesh,beta,c,sigma0,delta_sigma,g_array,\
                   phi_array,F_array,Tg,w_array)
    psi_array.append(psi0)
    i = 0
    # begin iterating
    while stop == False:
        i = i+1
        print "Iteration: ", i
        sigma.vector()[:] = sigma0array + \
                            delta_sigma.vector().array()
        sigma_array.append(sigma.vector().array())
        # computing discrepancy gradient
        Jprime = Function(V)
        for k in xrange(K):
            Jprime = Jprime + w_array[k]*Jsolver(mesh,sigma,\
                                        g_array[k],phi_array[k],\
                                        F_array[k],Tg)
        # computing Sobolev gradient
        Js_prev.vector()[:] = Js.vector().array()
        Js = Jsobolev(mesh,Jprime)
        # compute step size and next step
        s, TV_step = StepSize(sigma0,mesh,delta_sigma,\
                              delta_sigma_prev,Js,Js_prev,\
                              psi_array,g_array,phi_array,\
                              F_array,param,Tg,w_array)
        s_array.append(s)
        delta_sigma_prev.vector()[:] = delta_sigma.vector(). \
                                       array()
        delta_sigma.vector()[:] = TV_step.vector().array()
        # check stopping conditions
        if i == I_max or s <= s_stop:
            stop = True
    # compute final sigma
    sigma.vector()[:] = sigma0array + \
                        delta_sigma.vector().array()
    sigma_array.append(sigma.vector().array())
    return sigma, psi_array, s_array, sigma_array
```

## D.4 Sample Code

This section contains a small sample of code that shows how to use the above implementations to simulate data, load the data, run the sparsity solver, and finally visualize the result.

```
"""
Test the implementation for inverse EIT
"""

from dolfin import *
import numpy as np
import matplotlib.pyplot as py
import shelve # used to easily load/save stuff from/to files
import EITSparsitySolver as eit


# parameters
fine_mesh = Mesh("../mesh/finemesh.xml")
mesh = Mesh("../mesh/mediummesh.xml")
V = FunctionSpace(mesh,'CG',1)
datafile = "data"
resultfile = "result"
M = 5
q = 2
s_stop = 1E-3
s_min = 1.0
s_max = 1000.0
I_max = 200
tau = 1E-5
alpha = 2E-5
param = [s_min,s_max,M,tau,q,alpha,s_stop,I_max]
theta1 = 0.0
theta2 = 2*pi
Tg = eit.ThresholdFunction(mesh,theta1,theta2)
w_array = None # using default weights

# simulate data
r = 0.4
cx = -0.3
cy = 0.3
C = 5
class sigmafun(Expression):
    def eval(self, values, x):
        if np.sqrt((x[0]-cx)**2+(x[1]-cy)**2) < r:
            values[0] = 1+C
        else:
            values[0] = 1

sigma = sigmafun()
cos_array = [1,2,3,4,5]
sin_array = [1,2,3,4,5]
noiselvl = 1E-3

eit.SimulateEITData(sigma,mesh,fine_mesh,noiselvl,cos_array,\
                    sin_array,datafile)

# load data
data = shelve.open(datafile)
sigma = Function(V)
sigma.vector()[:] = data["sigma"]
u_exact_array = data["u_exact_array"]
u_noise_array = data["u_noise_array"]
cos_array = data["cos_array"]
sin_array = data["sin_array"]
data.close()
g_array = []
phi_array = []
for j in range(0,2):
    if j == 0:
        n_array = cos_array
        iscos = True
        p = 0
    else:
        n_array = sin_array
        iscos = False
        p = len(cos_array)
    k = len(n_array)
    for i in range(0,k):
        phi = Function(V)
        phi.vector()[:] = u_noise_array[i+p]
        g = eit.g_exp(iscos,n_array[i],theta1,theta2)
        phi_array.append(phi)
        g_array.append(g)
sigma0 = interpolate(Constant(1.0),V)

# attempt reconstruction
# we do not want 100's of "solving problem" logs
set_log_active(False)
sigma_approx,psi_array,s_array,sigma_array = \
    eit.SparsitySolver(sigma0,mesh,g_array,phi_array,param,Tg,\
                       w_array)

# visualize result
plot(sigma_approx,warpscalar=False,title='sigma approx')
plot(sigma,warpscalar=False,title='exact sigma')
interactive()
```

# Bibliography

## Articles

[1] Barzilai, J. and Borwein, J. M. *Two-point step size gradient methods*. IMA J. Numer. Anal., 8(1):141–148, 1988. ISSN 0272-4979. doi: 10.1093/imanum/8.1.141. URL: http://dx.doi.org.globalproxy.cvt.dk/10.1093/imanum/8.1.141.

[2] Bonesky, T., Bredies, K., Lorenz, D. A., and Maass, P. *A generalized conditional gradient method for nonlinear operator equations with sparsity constraints*. Inverse Problems, 23(5):2041–2058, 2007. ISSN 0266-5611. doi: 10.1088/0266-5611/23/5/014. URL: http://dx.doi.org.globalproxy.cvt.dk/10.1088/0266-5611/23/5/014.

[3] Borsic, A., Graham, B., Adler, A., and Lionheart, W. *In vivo impedance imaging with total variation regularization*. IEEE Trans Med Imaging, 29(1):44–54, 2010 Jan. doi: 10.1109/TMI.2009.2022540.

[4] Bredies, K., Lorenz, D. A., and Maass, P. *A generalized conditional gradient method and its connection to an iterative shrinkage method*. Comput. Optim. Appl., 42(2):173–193, 2009. ISSN 0926-6003. doi: 10.1007/s10589-007-9083-3. URL: http://dx.doi.org.globalproxy.cvt.dk/10.1007/s10589-007-9083-3.

[5] Daubechies, I., Defrise, M., and De Mol, C. *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*. Comm. Pure Appl. Math., 57(11):1413–1457, 2004. ISSN 0010-3640. doi: 10.1002/cpa.20042. URL: http://dx.doi.org.globalproxy.cvt.dk/10.1002/cpa.20042.

[6] Gallouet, T. and Monier, A. *On the regularity of solutions to elliptic equations.* Rend. Mat. Appl. (7), 19(4):471–488 (2000), 1999. ISSN 1120-7183.

[7] Gehre, M., Kluth, T., Lipponen, A., Jin, B., Seppänen, A., Kaipio, J. P., and Maass, P. *Sparsity reconstruction in electrical impedance tomography: an experimental evaluation.* J. Comput. Appl. Math., 236(8):2126–2136, 2012. ISSN 0377-0427. doi: 10.1016/j.cam.2011.09.035. URL: http://dx.doi.org.globalproxy.cvt.dk/10.1016/j.cam.2011.09.035.

[8] Jin, B. and Maass, P. *An analysis of electrical impedance tomography with applications to Tikhonov regularization.* ESAIM: Control, Optimisation and Calculus of Variations, 1(1), 2011.

[9] Jin, B., Khan, T., and Maass, P. *A reconstruction algorithm for electrical impedance tomography based on sparsity regularization.* Internat. J. Numer. Methods Engrg., 89(3):337–353, 2012. ISSN 0029-5981. doi: 10.1002/nme.3247. URL: http://dx.doi.org.globalproxy.cvt.dk/10.1002/nme.3247.

[10] Kress, R. *Inverse problems and conformal mapping.* Complex Var. Elliptic Equ., 57(2-4):301–316, 2012. ISSN 1747-6933. doi: 10.1080/17476933.2011.605446. URL: http://dx.doi.org.globalproxy.cvt.dk/10.1080/17476933.2011.605446.

[11] Meyers, N. G. *An $L^p$-estimate for the gradient of solutions of second order elliptic divergence equations.* Ann. Scuola Norm. Sup. Pisa (3), 17:189–206, 1963.

[12] Widlak, T. and Scherzer, O. *Hybrid tomography for conductivity imaging.* Inverse Problems, 28(8):084008, 28, 2012. ISSN 0266-5611. doi: 10.1088/0266-5611/28/8/084008. URL: http://dx.doi.org.globalproxy.cvt.dk/10.1088/0266-5611/28/8/084008.

[13] Wright, S. J., Nowak, R. D., and Figueiredo, M. A. T. *Sparse reconstruction by separable approximation.* IEEE Trans. Signal Process., 57(7):2479–2493, 2009. ISSN 1053-587X. doi: 10.1109/TSP.2009.2016892. URL: http://dx.doi.org.globalproxy.cvt.dk/10.1109/TSP.2009.2016892.

[14] Zorn, M. A. *Characterization of analytic functions in Banach spaces.* Ann. of Math. (2), 46:585–593, 1945. ISSN 0003-486X.

# Books

[15] Adams, R. A. and Fournier, J. J. F. *Sobolev spaces*, volume 140 of *Pure and Applied Mathematics (Amsterdam)*. Elsevier/Academic Press, Amsterdam, second edition, 2003. ISBN 0-12-044143-8.

[16] Asmar, N. *Partial differential equations with Fourier series and boundary value problems*. Pearson Prentice Hall, 2005. ISBN 9780131480964.

[17] Christensen, O. *Frames and bases*. Applied and Numerical Harmonic Analysis. Birkhäuser Boston Inc., Boston, MA, 2008. ISBN 978-0-8176-4677-6.

[18] Christensen, O. *Differentialligninger og uendelige rækker*. Department of Mathematics (DTU), 2009.

[19] Christensen, O. *Functions, spaces, and expansions*. Applied and Numerical Harmonic Analysis. Birkhäuser Boston Inc., Boston, MA, 2010. ISBN 978-0-8176-4979-1.

[20] Evans, L. C. *Partial differential equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, second edition, 2010. ISBN 978-0-8218-4974-3.

[21] Folland, G. B. *Real analysis*. Pure and Applied Mathematics (New York). John Wiley & Sons Inc., New York, second edition, 1999. ISBN 0-471-31716-0.

[22] Griffel, D. H. *Applied functional analysis*. Ellis Horwood Ltd., Chichester, 1981. ISBN 0-85312-226-1.

[23] Hansen, E. *Measure Theory*. University of Copenhagen. Department of mathematical sciences, 2009. ISBN 9788791927447.

[24] Hansen, P. C. *Discrete inverse problems*, volume 7 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2010. ISBN 978-0-898716-96-2.

[25] Hansen, V. L. *Functional analysis*. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2006. ISBN 981-256-686-4.

[26] Hansen, V. L. *Mathematics 3: Entrance to Advanced Mathematics*. Department of Mathematics (DTU), 2008.

[27] Mueller, J. L. and Siltanen, S. *Linear and Nonlinear Inverse Problems with Practical Applications*. *not yet published*, 2012.

[28] Neuberger, J. W. *Sobolev gradients and differential equations*, volume 1670 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, second edition, 2010. ISBN 978-3-642-04040-5.

[29] Rudin, W. *Real and complex analysis*. McGraw-Hill Book Co., New York, third edition, 1987. ISBN 0-07-054234-1.

[30] Strauss, W. A. *Partial differential equations*. John Wiley & Sons Ltd., Chichester, second edition, 2008. ISBN 978-0-470-05456-7.

## Software

[31] Alnæs, M. S., Hake, J., Hoffman, J., Jansson, J., Kirby, R. C., Logg, A., Mardal, K.-A., Massing, A., Narayanan, H., Ring, J., Rognes, M., Selim, K., Scott, R., Wells, G. N., and Ølgaard, K. B. The fenics project: Automated solution of differential equations by the finite element method, 2011. URL: http://www.fenicsproject.org.

[32] Geuzaine, C. and Remacle, J.-F. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities, July 2012. URL: http://geuz.org/gmsh/.