

# Computational Methods for Hybrid Impedance Tomography

M. Sc. Thesis by Christian Kragh



Kongens Lyngby 2016

Technical University of Denmark  
Department of Applied Mathematics and Computer Science  
Richard Petersens Plads, building 324,  
2800 Kongens Lyngby, Denmark  
Phone +45 4525 3031  
[compute@compute.dtu.dk](mailto:compute@compute.dtu.dk)  
[www.compute.dtu.dk](http://www.compute.dtu.dk)

# Abstract

This thesis investigates a hybrid impedance inverse problem where the conductivity in a planar domain is reconstructed from noise-free interior data of the current density and power density types. Two algorithms are implemented for reconstructing  $C^2$ -conductivities. The first algorithm performs well for the current density problem using a single measurement. The second algorithm produces good reconstructions for the power density problem using two measurements. The results indicate that both algorithms are convergent for the current density problem. The second algorithm is the only one which performs well for the power density problem but is only semi-convergent in that case. The effect of highly oscillating Neumann conditions is investigated, where the results indicate that higher frequencies lead to poorer reconstructions. The numerical results moreover indicate that  $L^2$ -regularization for the power density problem does not have any positive impact. Finally, the second algorithm is tested for the reconstruction of a piece-wise constant conductivity, with results that are comparable to the case of a  $C^2$ -conductivity.



# Preface

This Master's thesis was prepared at the Department of Applied Mathematics and Computer Science at the Technical University of Denmark (DTU Compute) as the final project of the Master Programme in Mathematical Modelling and Computation. The work has been carried out during the spring of 2016 from February to July.

The thesis deals with topics of applied functional analysis and partial differential equations. Specifically, it focuses on numerical methods for inverse problems arising in hybrid tomography for conductivity imaging.

It is intended to address fellow students at the graduate level in applied mathematics or engineering mathematics. A prerequisite is familiarity with Functional Analysis and specifically the concept of Hilbert spaces, boundedness and dual spaces. Moreover, some knowledge of theory for partial differential equations is also strongly recommended. That is, acquaintance with the concepts of weak formulations, Sobolev spaces and the Lax–Milgram Theorem is advantageous.

Specifically, students at the Mathematical Modelling and Computation Master Programme at DTU should have the proper prerequisites if they have completed the courses Partial Differential Equations (01246), Functional Analysis (01715) and Advanced Topics in Applied Functional Analysis (01716).

Christian Kragh  
Kgs. Lyngby, July 10, 2016.



# Acknowledgements

I would like to thank my supervisor, Associate Professor Kim Knudsen for his very valuable advice in our weekly meetings and for guiding me in the right direction while also allowing me to work independently. I would also like to thank my co-supervisor Postdoc Stratos Staboulis for our discussions, his very useful inputs and views on the subject.

Finally, I would like to thank Ida Bertelsen for her continuous support during the whole project.





# Contents

<b>Abstract</b>	<b>i</b>
<b>Preface</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Mathematical model . . . . .	3
1.3 Statement of a hybrid inverse problem . . . . .	5
1.4 Thesis outline . . . . .	8
<b>2 The forward problem</b>	<b>9</b>
2.1 Variational form - existence and uniqueness . . . . .	9
2.2 An analytical solution . . . . .	17
2.2.1 Power density data . . . . .	19
<b>3 Methods for solving inverse problems</b>	<b>23</b>
3.1 Continuity and differentiability of the forward map . . . . .	23
3.1.1 Fréchet differentiability of the forward map . . . . .	26
3.1.2 Fréchet derivative of the data map . . . . .	29
3.2 Methods of the Newton-type . . . . .	31
3.2.1 Newton's method in Banach spaces . . . . .	31
3.2.2 A Newton-type method given Power Density data . . . . .	32
3.2.3 A Newton-type method with multiple measurements . . . . .	34
3.2.4 Ellipticity of the linearised problem . . . . .	38
<b>4 Implementation</b>	<b>41</b>
4.1 The Finite Element setup . . . . .	41
4.2 Validation of the forward problem implementation . . . . .	45
4.2.1 Code examples from Python . . . . .	45
4.3 Implementation details of the inverse problem . . . . .	48
4.3.1 Meshes and conductivity phantoms . . . . .	48

---

4.3.2	Parameter choices . . . . .	49
<b>5</b>	<b>Numerical results</b>	<b>53</b>
5.1	Reconstruction of a smooth target conductivity . . . . .	53
5.2	Regularization effects . . . . .	57
5.3	Effects of the choice of boundary conditions . . . . .	58
5.4	Propagation of singularities . . . . .	59
5.5	Reconstruction of a piece-wise constant conductivity . . . . .	61
<b>6</b>	<b>Conclusion and perspectives</b>	<b>63</b>
6.1	Conclusion . . . . .	63
6.2	Perspectives . . . . .	65
	<b>References</b>	<b>65</b>
<b>A</b>	<b>Learning objectives and plan</b>	<b>71</b>
A.1	Final version . . . . .	71
A.1.1	Learning objectives . . . . .	71
A.1.2	Time plan . . . . .	72
A.2	Note on corrections to the original plan . . . . .	73
A.3	Original version . . . . .	74
A.3.1	Learning objectives . . . . .	74
A.3.2	Time plan . . . . .	75
<b>B</b>	<b>Python code</b>	<b>77</b>

# Chapter 1

## Introduction

### 1.1 Motivation

Since the introduction of the X-ray computed tomography (CT) technique in the 1970's, there has been a growing research interest in tomographic methods, leading to many other types of tomography such as PET (positron-electron tomography), SPECT (single photon emission CT), MRI (magnetic resonance imaging), optical- and ultrasound tomography and EIT (electrical impedance tomography).

Originating from Greek as the combination of *tomos* meaning "slice" and *graphia* meaning "description of", the term *tomography* refers to imaging by sectioning. However, it now has a broader definition since in many cases the techniques are not confined to 2D sections and are thus often 3D-based methods. They are however all methods for determining the internal structure of an object from measurements of their corresponding physical modalities. The mathematical problems involved in the methods are so-called *inverse problems* which have been of great theoretical interest in mathematics.

A part from the fact that the methods are based on different physical phenomena and thereby different measurement techniques, they also differ by their costs, the degree of health risk they expose patients to and their ability to detect different biological features.

This led to the idea that coupling different physical modalities might be advantageous and could potentially produce imaging methods with more precision and applicability than the single modality methods. These coupled modality methods are called *hybrid* tomography methods.

Tissue	Conductivity [ $\text{S m}^{-1}$ ]
Skin	$3 \times 10^{-3}$
Fat	$2 \times 10^{-2}$
Liver	$5 \times 10^{-2}$
Blood	$7 \times 10^{-1}$

**Table 1.1:** Conductivities of different types of tissues.

The case of hybrid *impedance* tomography methods have their origins in electrical impedance tomography (EIT) which is a single modality method for reconstructing the conductivity inside an object (e.g. the human body). In EIT, electrodes are attached to the boundary of the object and the resulting voltages on the boundary are measured to provide the data for the reconstruction.

The conductivity is known to be an appropriate indicator of the health of biological tissues and functional processes [12]. Specifically, it provides valuable information of the distribution of air in the lungs of a patient due to the fact that air has a very small conductivity. Moreover, the conductivity of different types of biological tissue can differ by several orders of magnitude making them easily distinguishable provided a high-resolution image. The conductivities of different types of tissue can be seen in Table 1.1.



**Figure 1.1:** A ten days old child with electrodes attached to the chest as an experimental setup for EIT [17].

The original mathematical formulation of the EIT problem was proposed by Calderón in 1980 [8]. It is known to be heavily ill-conditioned and the modulus of continuity is no better than logarithmic [1], meaning that exponential decrease in measurement error only leads to a linear improvement in precision. Due to the logarithmic stability, it is only possible to obtain low resolution reconstructions of the conductivity [27].

Hybrid electrical impedance tomography is then based on the idea of coupling EIT with other physical modalities in order to produce images of higher resolution or to obtain

Symbol	Quantity	Unit	Type
$\mathbf{E}(\mathbf{x}, t)$	Electric field strength	$\text{Vm}^{-1}$	vector field
$\mathbf{B}(\mathbf{x}, t)$	Magnetic Flux density	$\text{Vsm}^{-2} = \text{T}$	vector field
$\mathbf{J}(\mathbf{x}, t)$	Electrical current density	$\text{Am}^{-2}$	vector field
$\sigma(\mathbf{x})$	Electrical conductivity	$\text{AVm}^{-1} = \text{Sm}^{-1}$	scalar field
$\rho(\mathbf{x}, t)$	Electrical charge density	$\text{Asm}^{-3}$	scalar field
$\epsilon(\mathbf{x})$	Electrical permittivity	$\text{AsV}^{-1}\text{m}^{-1} = \text{Fm}^{-1}$	scalar field
$\mu(\mathbf{x})$	Magnetic permeability	$\text{A}^{-1}\text{sVm}^{-1} = \text{Hm}^{-1}$	scalar field

**Table 1.2:** Physical quantities of electrodynamics

added information compared to the single modality case of EIT. The coupling of physical modalities utilised in hybrid EIT is expected provide more stability than that of the original EIT problem [3].

## 1.2 Mathematical model

To provide the physical background of hybrid EIT, we need to go to Maxwell's equations of electrodynamics. In their differential form they read (see for example Griffiths [15]),

$$\begin{aligned} \nabla \cdot \mathbf{E} &= \frac{\rho}{\epsilon_0} & \nabla \cdot \mathbf{B} &= 0 \\ \nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t} & \nabla \times \mathbf{B} &= \mu_0 \left( \mathbf{J} + \epsilon_0 \frac{\partial \mathbf{E}}{\partial t} \right), \end{aligned} \quad (1.1)$$

with the physical quantities given in Table 1.2 and the constant  $\epsilon_0$  and  $\mu_0$  being the vacuum permittivity and permeability respectively. Apart from Maxwell's equations there are a few more important law's of electrodynamics which will be relevant from our perspective. With a continuous charge distribution, the force field pr. unit volume (Lorentz' equation) is given by

$$\mathbf{f} = \rho \mathbf{E} + \mathbf{J} \times \mathbf{B}. \quad (1.2)$$

and the power density is given by

$$H = \mathbf{J} \cdot \mathbf{E}, \quad (1.3)$$

which measures the power [W] of the electric field pr. unit volume. This is of course a scalar quantity and hence is not typeset in bold. A central part of the model is the linear approximation of the relation between the current density  $\mathbf{J}$  and the electric field  $E$ . This is the generalized Ohm's law,

$$\mathbf{J} = \sigma \mathbf{E}, \quad (1.4)$$

which holds for so-called linear media including biological tissue [27]. The conductivity is simply a name for the proportionality constant. The intuition of (1.4) is that in a medium

with a large conductivity, even a small electrical field magnitude can result in a large current density. We shall view the conductivity as a function of space  $\sigma(\mathbf{x})$  and neglect that  $\sigma$  might also depend on time and the frequency of the electrical field as noted by Widlak and Scherzer [27].

Inspecting the third equation in (1.1) (Faraday's law), the partial derivative of the magnetic field with respect to time is present on the right hand side. With a planar wave solution to the magnetic field of the form  $\mathbf{B} = \mathbf{B}_0 \exp(\mathbf{k} \cdot \mathbf{r} - i\omega t)$ , this reduces to

$$\nabla \times \mathbf{E} = -i\omega \mathbf{B} \quad (1.5)$$

from where the *electrostatic approximation* in the low frequency regime, reads

$$\nabla \times \mathbf{E} = 0. \quad (1.6)$$

The approximation is arguably valid ([10], A.2) if

$$\sqrt{\mu_0 \omega \sigma} \ll L \quad (1.7)$$

with  $\omega, \sigma, L$  denoting a reference frequency, conductivity and length respectively. The fact that  $\mathbf{E}$  is rotation-free in  $\mathbb{R}^3$  (a simply connected region) is sufficient to conclude that it can be written as the gradient of a scalar field  $u$  [21],

$$\mathbf{E} = -\nabla u. \quad (1.8)$$

The minus sign is the convention for an energy potential with the force field should point in the direction of largest decrease in the potential.

The fourth Maxwell equation in (1.1) is Ampère's law with Maxwell's correction of including the displacement current  $\epsilon_0 \frac{\partial \mathbf{E}}{\partial t}$ . Taking the divergence on both sides and using that the divergence of a curl is zero, it becomes

$$\nabla \cdot \mathbf{J} + \epsilon_0 \frac{\partial (\nabla \cdot \mathbf{E})}{\partial t} = 0 \quad (1.9)$$

Applying now Gauss' law, the first of Maxwell's equations in (1.1), we get the law of charge conservation,

$$\nabla \cdot \mathbf{J} = -\frac{\partial \rho}{\partial t}, \quad (1.10)$$

Recalling Gauss' Divergence Theorem, this simply corresponds to the fact that the net outward surface current through any closed surface  $\partial\Omega_i$  enclosing a volume  $\Omega_i$  must be due to a change in the charge density,

$$\int_{\partial\Omega_i} \mathbf{J} \cdot \hat{\mathbf{n}} \, ds = \int_{\Omega_i} \nabla \cdot \mathbf{J} \, d\mathbf{x} = -\frac{\partial}{\partial t} \int_{\Omega_i} \rho \, d\mathbf{x}, \quad (1.11)$$

where  $q = \int_{\Omega} \rho d\mathbf{x}$  is the net charge inside the volume  $\Omega$ . Since this must be true for any volume, and we require that there is no local accumulation of charge, we arrive at the equation,

$$\nabla \cdot \mathbf{J} = 0. \quad (1.12)$$

Combining this with Ohm's law (1.4) and (1.8), we obtain

$$\nabla \cdot (\sigma \nabla u) = 0. \quad (1.13)$$

Suppose we have an object which occupies a region  $\Omega \subset \mathbb{R}^3$ . Then the outward current density is measured on the boundary  $\partial\Omega$  to give boundary data,

$$\mathbf{J} \cdot \hat{\mathbf{n}} = \sigma \frac{\partial u}{\partial n} = f \quad \text{on } \partial\Omega. \quad (1.14)$$

Combining (1.13) and (1.14) we arrive at a boundary value problem for determining the potential  $u$  given a known conductivity  $\sigma$  and boundary measurements  $f$ .

$$\begin{aligned} \nabla \cdot (\sigma \nabla u) &= 0 \quad \text{in } \Omega, \\ \sigma \frac{\partial u}{\partial n} &= f \quad \text{on } \partial\Omega, \end{aligned} \quad (1.15)$$

which is the boundary value problem constituting the forward map of EIT. Having obtained a potential  $u$  from (1.15), we can compute  $p$ -power density,

$$H = \sigma |\nabla u|^p, \quad p > 0. \quad (1.16)$$

From (1.8), the relation between electric field  $E$  and the potential  $u$  is given by  $-\nabla u = E$  [15]. In the case  $p = 2$ ,

$$H(\sigma) = \sigma |E|^2 \quad \text{with the SI-unit} \quad [H] = \frac{\text{J}}{\text{s} \cdot \text{m}^3} = \frac{\text{W}}{\text{m}^3} = \frac{\text{power}}{\text{volume}}. \quad (1.17)$$

Hence for  $p = 2$ ,  $H(\sigma)$  corresponds to the power density (the electrical power pr. unit volume). As we shall see shortly, (1.15) and (1.16) are the key ingredients in this project.

### 1.3 Statement of a hybrid inverse problem

The problems arising in (hybrid) tomography are inverse problems. The fundamental objective of an inverse problem is to find  $x$ , typically a model parameter, given some output data  $d$  such that

$$\mathcal{F}(x) = d \quad (1.18)$$

where  $\mathcal{F}$  is the *forward map* which is known more or less explicitly. In that sense, the problem of finding  $x$  is essentially to invert the operator  $\mathcal{F}$  mapping  $x$  into the data  $d$ .

Mathematically,  $\mathcal{F}$  is often a differential or integral operator which is not straightforward to invert. The concept of well-posedness is very central in inverse problem theory and originates from the definition given by Hadamard [16] (1902). Qualitatively it states that a mathematical model should fulfil the three properties:

1. Existence
2. Uniqueness
3. Stability with respect to the data

Many inverse problems are *ill-posed* and often the third condition is not fulfilled. This is the case for EIT, where existence and uniqueness results have been obtained [26], but where solutions depend too sensitively on the data [1]. We will here propose a class of hybrid EIT inverse problem where interior data of the power density type (1.16) is added to the EIT data. To state the problem we first need to introduce a bit of notation.

We consider a open, bounded domain  $\Omega$  in  $\mathbb{R}^2$  or  $\mathbb{R}^3$  and denote by  $\partial\Omega$  the boundary of  $\Omega$  which is assumed to be smooth. Then we introduce the following Sobolev spaces (See Evans [11] for a textbook reference on the well-known spaces  $H^1$ ,  $H^{1/2}$  and their duals),

$$\tilde{H}^1(\Omega) := \left\{ u \in H^1(\Omega) \mid \int_{\partial\Omega} u \, dS = 0 \right\}. \quad (1.19)$$

$$\tilde{H}^{-1/2}(\partial\Omega) := \left\{ f \in H^{-1/2}(\partial\Omega) \mid \int_{\partial\Omega} f \, dS = 0 \right\}, \quad (1.20)$$

where integrals on  $\partial\Omega$  should be understood in the trace sense. Moreover we introduce the space,

$$L_+^\infty(\Omega) := \left\{ \sigma \in L^\infty(\Omega) \mid \sigma(x) > c_1 \text{ a.e. in } \Omega, \quad c_1 > 0 \right\}, \quad (1.21)$$

which requires conductivities to have a global lower bound, corresponding to no material being perfectly insulating. We can now define the inverse problem for the interior data hybrid EIT.



**Problem 1** (Power Density EIT). *Let the true conductivity  $\sigma \in L_+^\infty(\Omega)$  and let  $f_m \in \tilde{H}^{-1/2}(\partial\Omega)$  be given. Then  $u_m \in \tilde{H}^1(\Omega)$  denotes the solution of*

$$\begin{aligned} \nabla \cdot (\sigma \nabla u_m) &= 0 && \text{in } \Omega \\ \sigma \frac{\partial u_m}{\partial n} &= f_m && \text{on } \partial\Omega, \end{aligned} \tag{1.22}$$

*Then the interior data,*

$$H_m(\sigma) = \sigma |\nabla u_m|^p, \quad p > 0 \quad \text{in } \Omega, \tag{1.23}$$

*is given for each measurement  $f_m$ .*

*Reconstruct  $\sigma$ .*

Problem 1 is a more general case of a hybrid inverse problem, where we are not specifically concerned with the method used for obtaining the measurements of the interior data  $H(\sigma)$ . However, we motivate the investigation of this type of problem, by existing tomographic methods that are related to or covered by this formulation.

One example is the magnetic resonance EIT (MREIT) which is based on the physical coupling of the current density field  $\mathbf{J}$  and the magnetic field  $\mathbf{B}$ . The fourth equation in (1.1) (Ampère's law) can be used to express the current density in terms of  $\mathbf{B}$  in the electrostatic approximation,

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} \tag{1.24}$$

An MRI scanner can be used to measure one component of the  $\mathbf{B}$ -field. By rotating the object, one would ideally be able to obtain complete information about  $\mathbf{B}$  and hence  $\mathbf{J}$ . In practice, knowledge of one or two components of  $\mathbf{B}$  is more achievable. MREIT hence is typically concerned with interior measurements of  $\mathbf{J}$ ,  $|\mathbf{J}|$  or only the  $z$ -component of the magnetic field  $B_z$  [27]. Note that  $|\mathbf{J}|$  corresponds exactly to the case  $p = 1$  in our model, since  $\mathbf{J} = \sigma \nabla u$  in linear media. This case is often called Current Density Impedance (CDII) [27].

The case of power density interior data ( $p = 2$ ) appears in different types of ultrasound coupled impedance tomography. In the review by Widlak and Scherzer [27], several methods are mentioned. Impedance-acoustic tomography (IAT) is stated as the conductivity reconstruction from Cauchy data and the power density interior data  $\sigma |\nabla u|^2$ , where the power density is obtained by the ultrasound wave generated by thermal expansion due to Joule heating.

In acousto-electrical tomography (AET) ultrasound beams are used to excite the electrical field and thereby perturbing the conductivity. The measured voltage difference can then be used to compute the power density data  $\sigma|\nabla u|^2$  [27].

Motivated by these hybrid tomography schemes, this project will focus on methods for solving Problem 1 in the cases  $p = 1$  and  $p = 2$  with an emphasis on the latter. However, a few results will also be based on other cases where  $p \in [0.5, 2]$ .

## 1.4 Thesis outline

Having motivated the subject and introduced the main concepts, we are prepared to state the goals of this project and lay out the path leading to these goals.

This thesis surveys computational methods for solving Problem 1 with one and two induced currents and corresponding Neumann measurements. The methods are based on a linearisation of the forward map and a Newton-type method for iterative reconstruction of the conductivity, which is closely related to the algorithms presented by Hoffmann [18], although here applied to the Neumann boundary data whereas in [18], Dirichlet boundary conditions are used. The variational problems are solved using the Finite Element method via the software package **FEniCS** [2].

First, the forward problem of EIT is investigated with emphasis on existence and uniqueness and the formulation of a variational form which is implementable in a Finite Element setting. Subsequently, the methods for solving the inverse problem are presented by deriving the linearisation (Fréchet derivative) of the forward map and variational forms are presented for obtaining updating the conductivity by an increment  $\delta\sigma$ . Then two algorithms are posed as solution procedures to Problem 1. The implementation of the two algorithms is then described, leading up to the presentation of the obtained numerical results. These are discussed in terms of related work. Finally, the perspectives and limitations are discussed and a conclusion of the thesis is presented.

The main objectives of this thesis is to investigate the performance of the two Newton type algorithms from Hoffmann [18] applied to the Neumann boundary condition. In the case  $p = 2$ , we would like to address how does the regularization parameter  $\alpha$  and the choice of Neumann conditions affect the quality of the conductivity reconstructions. Moreover, we will aim to confirm the prediction of propagation of singularities from the thesis by Hoffmann [19] in the case of Neumann boundary conditions. Finally, we will evaluate how the method performs for the reconstruction of a piece-wise constant target conductivity.

# Chapter 2

## The forward problem

This chapter focuses solely on the forward problem of Electrical Impedance Tomography (EIT), which is a Laplace type partial differential equation (PDE) in two or three spatial dimensions. This type of PDE can be solved analytically in some special cases but often requires a numerical approach, e.g. the Finite Element Method (FEM) which will be employed here. Functional analysis also plays a key role in determining conditions for unique solvability of the weak and strong formulations of the boundary value problem, where the weak formulation is especially important regarding the FEM. We here define some concepts that were already mentioned in the introduction. This chapter however goes into a more detail with these concepts.

### 2.1 Variational form - existence and uniqueness

As we saw in Section 1.3 the forward problem of EIT stems from the electrostatic equation. Let  $\Omega$  be a bounded domain in  $\mathbb{R}^2$  or  $\mathbb{R}^3$ , with a smooth boundary  $\partial\Omega$ . Then we consider the boundary value problem from (1.15),

$$\begin{aligned}\nabla \cdot (\sigma \nabla u) &= 0 && \text{in } \Omega \\ \sigma \frac{\partial u}{\partial n} &= f && \text{on } \partial\Omega,\end{aligned}\tag{2.1}$$

with  $\frac{\partial u}{\partial n}$  denoting the outward normal derivative on the boundary.

As one might suspect, this problem is not uniquely solvable for e.g.  $u \in C^2(\Omega)$  since adding

a constant would immediately give a new solution. Therefore, choosing a proper space for  $u$  is important.

In practice, expressing the problem as in (2.1) is not always very convenient. It turns out that a weaker formulation of the problem can be very powerful both for theoretical and numerical reasons. Consider (2.1) where we multiply on both sides by a smooth test function,  $v \in C^\infty(\bar{\Omega})$ , and integrate to get

$$\int_{\Omega} v (\nabla \cdot (\sigma \nabla u)) dx = \int_{\Omega} v (\sigma \Delta u + \nabla \sigma \cdot \nabla u) dx = 0, \quad (2.2)$$

provided  $u$  is sufficiently smooth. It will later become apparent what exactly we mean by this. Applying Green's first identity, (2.2) becomes

$$\int_{\partial\Omega} \frac{\partial u}{\partial n} \sigma v dS - \int_{\Omega} \nabla u \cdot (\sigma \nabla v + v \nabla \sigma) dx + \int_{\Omega} v (\nabla \sigma \cdot \nabla u) dx = 0, \quad (2.3)$$

where two of the terms cancel out to give

$$\int_{\Omega} \sigma \nabla u \cdot \nabla v dx = \int_{\partial\Omega} \frac{\partial u}{\partial n} \sigma v dS. \quad (2.4)$$

Using the Neumann-condition from (2.1), we arrive at

$$\int_{\Omega} \sigma \nabla u \cdot \nabla v dx = \int_{\partial\Omega} f v dS \quad \forall v \in C^\infty(\bar{\Omega}), \quad (2.5)$$

which is what later will form the basis of the weak formulation of (2.1).

We have now stated the alternate formulation (2.5) without really addressing which spaces we should expect  $u$  and  $f$  to belong to. Inspecting (2.5), we would at least require that  $\nabla u$  and  $\nabla v$  exist. That would mean  $u, v \in C^1(\Omega)$ . However, this is more restrictive than necessary. In fact, the partial derivatives of  $u$  and  $v$  entering (2.5) can be replaced by the *weak* partial derivatives. The appropriate spaces for weakly differentiable functions are the Sobolev spaces  $W^{k,p}$  and specifically the Hilbert spaces  $H^k = W^{k,2}$ . See Evans [11], Chap. 5, for a textbook defining weak derivatives and Sobolev spaces. Returning to (2.5), we can write for any  $u, v, \sigma \in C^\infty(\Omega)$ ,

$$\int_{\Omega} \sigma \nabla u \cdot \nabla v dx = \int_{\Omega} \sigma \sum_{i=1}^d u_i v_i dx = \sum_{i=1}^d \int_{\Omega} \sigma u_i v_i dx = \sum_{i=1}^d \int_{\Omega} \sigma D^i u D^i v dx \quad (2.6)$$

where  $u_i$  denotes the (strong) partial derivative with respect to  $u_i$ . The last equality is true because of the approximation result (Th. 5.3.2 in [11]), stating that  $C^\infty(\Omega)$  is dense in  $W^{k,p}(\Omega)$  and thus also in  $H^1(\Omega)$ . Hence, we can relax the condition on  $u$  and  $v$  to  $u, v \in H^1(\Omega) \subset L^2(\Omega)$  when considering (2.5). When writing  $\nabla u$  we then also understand it as the weak gradient of  $u$ .

Inspecting the original PDE (2.1), we immediately encounter an issue with *uniqueness*, since assuming  $u$  is a solution then also  $\tilde{u} = u + c$  is a solution. The same holds for any solution to (2.5). Hence we need some other requirements on  $u$  and possibly  $v$  and  $f$ . One way to obtain uniqueness would be to choose  $u \in H^1(\Omega)/\mathbb{R}$  identifying  $u$  with every function  $u + c$ ,  $c \in \mathbb{R}$ . A more physical requirement however, would be the grounding of  $u$  (on average) on the boundary, demanding  $\int_{\partial\Omega} u \, dS = 0$ . This motivates the introduction of the following auxiliary function spaces,

**Definition 1** (The space  $\tilde{H}^1(\Omega)$ ). *The Sobolev space  $H^1(\Omega)$  consists of functions  $u \in H^1(\Omega)$ , such that*

$$\int_{\partial\Omega} u \, dS = 0, \quad (2.7)$$

equipped with the norm,

$$\|u\|_{\tilde{H}^1(\Omega)} := \|\nabla u\|_{L^2(\Omega)} \quad (2.8)$$

and the corresponding inner product,

$$\langle u, v \rangle_{\tilde{H}^1(\Omega)} := \langle \nabla u, \nabla v \rangle_{L^2(\Omega)}. \quad (2.9)$$

Since now  $u$  is assumed to be in  $H^1(\Omega)$  we have to be careful integrating along the boundary  $\partial\Omega$ . The boundary has dimension  $d - 1$  and is therefore of Lebesgue measure zero. This means  $u$  does not necessarily take values on the boundary. The integral should be understood in the context of traces:

$$\int_{\partial\Omega} u \, dS := \int_{\partial\Omega} Tu \, dS, \quad (2.10)$$

where  $T : W^{1,p} \rightarrow L^p(\partial\Omega)$  is the trace operator (see Evans [11], Theorem 5.5.1). Whenever we integrate an  $H^1(\Omega)$ -function  $u$  along the boundary we will understand it as an integral over  $Tu$  rather than  $u$ .

**Remark 1.** *The norms  $\|\cdot\|_{H^1(\Omega)}$  and  $\|\cdot\|_{\tilde{H}^1(\Omega)}$  are equivalent. That is, there exists constants  $a, b > 0$  such that*

$$a\|u\|_{H^1(\Omega)} \leq \|u\|_{\tilde{H}^1(\Omega)} \leq b\|u\|_{H^1(\Omega)} \quad \forall u \in \tilde{H}^1(\Omega) \subset H^1(\Omega). \quad (2.11)$$

*Proof.* In the case of  $H^1(\Omega)$ , the Sobolev space norm reduces to

$$\|u\|_{H^1(\Omega)}^2 = \|u\|_{L^2(\Omega)}^2 + \|\nabla u\|_{L^2(\Omega)}^2, \quad (2.12)$$

by which the right inequality of (2.11) is clearly fulfilled with  $b = 1$ , since

$$\|u\|_{\tilde{H}^1(\Omega)}^2 = \|\nabla u\|_{L^2(\Omega)}^2 \leq \|u\|_{L^2(\Omega)}^2 + \|\nabla u\|_{L^2(\Omega)}^2 = \|u\|_{H^1(\Omega)}^2. \quad (2.13)$$

The left inequality of (2.11) is obtained using the Poincaré inequality with the average taken on the boundary (see for example [13] B.3), which states that

$$\|u - (u)\|_{L^2(\Omega)} \leq C \|\nabla u\|_{L^2(\Omega)} \quad \forall u \in H^1(\Omega), \quad C > 0. \quad (2.14)$$

where  $(u) := |\partial\Omega|^{-1} \int_{\partial\Omega} u \, dx$ . For an estimate of the left-hand side of (2.14) we note that by definition  $(u) = 0$  for  $u \in \tilde{H}^1(\Omega)$ . Hence,

$$\|u\|_{L^2(\Omega)} \leq C \|\nabla u\|_{L^2(\Omega)}. \quad (2.15)$$

By the definition of the norm on  $\tilde{H}^1(\Omega)$  we then have

$$\|u\|_{H^1(\Omega)} = \|u\|_{L^2(\Omega)} + \|\nabla u\|_{L^2(\Omega)} \leq (C+1)\|\nabla u\|_{L^2(\Omega)} = (C+1)\|u\|_{\tilde{H}^1(\Omega)}, \quad (2.16)$$

which proves the left inequality of (2.11) with  $a = (C+1)^{-1}$ .  $\square$

Moreover,  $\tilde{H}^1(\Omega)$  is in fact closed in  $H^1(\Omega)$  and is thus also a Hilbert space ([13], Lemma 2.2). Requiring  $u, v \in \tilde{H}^1(\Omega)$  is a way to obtain a uniquely solvable problem as we will later see.

Now, we are almost ready to formally define the problem we wish to solve. However, we have only been concerned with the left-hand side of (2.5) so far. As for the right-hand side, we need to consider which space  $f$  should belong to for the integral  $\int_{\partial\Omega} f v \, dS$  to make sense. For this we need to define the trace space on  $\Omega$  and its dual space,

$$T(H^1(\Omega)) := H^{1/2}(\partial\Omega), \quad H^{-1/2}(\partial\Omega) := (H^{1/2})'(\partial\Omega). \quad (2.17)$$

Understanding  $f$  as a functional on  $H^{1/2}(\partial\Omega)$ , we can define  $f : H^{1/2}(\partial\Omega) \rightarrow \mathbb{R}$  as the dual pairing of  $f$  with  $Tv \in H^{1/2}(\partial\Omega)$ ,

$$\langle f, v \rangle = \langle f, Tv \rangle_{L^2(\partial\Omega)} = \int_{\partial\Omega} f Tv \, dS, \quad f \in H^{-1/2}(\Omega). \quad (2.18)$$

Setting  $v = 1 \in H^1(\Omega)$  in (2.5) gives

$$\int_{\partial\Omega} f \, dx = \int_{\Omega} \nabla u \nabla 1 \, dx = 0. \quad (2.19)$$

Hence also  $f$  should integrate zero on the boundary in order to have a legitimate solution of (2.5) and is hence also necessary to have a strong solution in the sense of (2.1). This leads to the definition of the space,

$$\tilde{H}^{-1/2}(\partial\Omega) := \left\{ f \in H^{-1/2}(\partial\Omega) \mid \int_{\partial\Omega} f \, dS = 0 \right\}. \quad (2.20)$$

We have then obtained appropriate function spaces for  $u, v$  and  $f$ . Now, if  $\sigma$  is an (essentially) bounded function,  $\sigma \in L^\infty(\Omega)$ , then

$$\left| \int_{\Omega} \sigma \nabla u \nabla v \, dx \right| \leq \|\sigma\|_{L^\infty(\Omega)} \left| \int_{\Omega} \nabla u \nabla v \, dx \right| < \infty, \quad (2.21)$$

since  $\nabla u$  and  $\nabla v$  belong to  $L^2(\Omega)$  by definition of the Sobolev space  $\tilde{H}^1(\Omega)$ . Shortly, we will also prove boundedness of the integral as a bilinear form. Requiring also that  $\sigma(x) > c_1$  a.e. in  $\Omega$  for  $c_1 > 0$  corresponds to the physical consideration that no region is perfectly insulating ( $\sigma \neq 0$  anywhere). In mathematical terms, the lower and upper global bounds for  $\sigma$  can be obtained by defining the space

$$L_+^\infty(\Omega) := \left\{ \sigma \in L^\infty(\Omega) \mid \sigma(x) > c_1 \text{ a.e. in } \Omega, \quad c_1 > 0 \right\}. \quad (2.22)$$

Now we are able to state a uniquely solvable weak formulation of the forward problem. Let  $u, v \in \tilde{H}^1(\Omega)$ ,  $f \in \tilde{H}^{-1/2}(\partial\Omega)$ , and  $\sigma \in L_+^\infty(\Omega)$ . Then we define the weak formulation of (2.1) as

$$\int_{\Omega} \sigma \nabla u \cdot \nabla v \, dx = \int_{\partial\Omega} f v \, dS \quad \forall v \in \tilde{H}^1(\Omega). \quad (2.23)$$

The left-hand side of (2.23) is the bilinear form,

$$\begin{aligned} a : \tilde{H}^1(\Omega) \times \tilde{H}^1(\Omega) &\rightarrow \mathbb{R}, \quad \text{s.t.} \\ a(u, v) &= \int_{\Omega} \sigma \nabla u \cdot \nabla v \, dx, \end{aligned} \quad (2.24)$$

which is bounded, since

$$\begin{aligned} |a(u, v)| &\leq \|\sigma\|_{L^\infty(\Omega)} \left( \int_{\Omega} |\nabla u|^2 \, dx \int_{\Omega} |\nabla v|^2 \, dx \right)^{\frac{1}{2}} = \|\sigma\|_{L^\infty(\Omega)} \|u\|_{\tilde{H}^1(\Omega)} \|v\|_{\tilde{H}^1(\Omega)} \\ &= A \|u\|_{\tilde{H}^1(\Omega)} \|v\|_{\tilde{H}^1(\Omega)}. \end{aligned} \quad (2.25)$$

The right-hand side of (2.23) is the bounded linear functional  $l : \tilde{H}^1(\Omega) \rightarrow \mathbb{R}$ , such that

$$l(v) = \int_{\partial\Omega} f v \, dx. \quad (2.26)$$

Boundedness of  $l$  is a consequence of  $f$  belong to  $H^{-1/2}(\partial\Omega)$  — the space of bounded linear functionals on  $H^{1/2}(\partial\Omega)$ . Note that one could also obtain boundedness of  $l$  with the restriction  $f \in L^2(\partial\Omega)$ , which would imply

$$|l(v)| = \left| \int_{\partial\Omega} f v \, dx \right| \leq \left( \int_{\partial\Omega} |f|^2 \, dx \int_{\partial\Omega} |v|^2 \, dx \right)^{\frac{1}{2}} = \|f\|_{L^2(\partial\Omega)} \|v\|_{L^2(\partial\Omega)} \quad (2.27)$$

$$= \|f\|_{L^2(\partial\Omega)} \|Tv\|_{L^2(\partial\Omega)} \leq \|f\|_{L^2(\partial\Omega)} \|v\|_{\tilde{H}^1(\Omega)} \quad (2.28)$$

$$\leq C \|f\|_{L^2(\partial\Omega)} \|v\|_{\tilde{H}^1(\Omega)} = B \|v\|_{\tilde{H}^1(\Omega)} \quad (2.29)$$

utilizing the norm equivalence of Remark 1. The problem (2.23) can then be formulated as the variational problem, of finding  $u \in \tilde{H}^1(\Omega)$ , s.t.

$$a(u, v) = l(v) \quad \forall v \in \tilde{H}^1(\Omega), \quad (2.30)$$

In order to obtain uniqueness of (2.23), The Lax–Milgram Theorem (Th. 6.2.1 in Evans [11]) becomes of great use.

**Theorem 1** (Uniqueness). *Let  $\Omega$  be a bounded domain in  $\mathbb{R}^d$  ( $d = 2$  or  $d = 3$ ) with a smooth boundary  $\partial\Omega$  and let  $\sigma \in L_+^\infty(\Omega)$ . Then the problem (2.23) has a unique solution,  $u \in \tilde{H}^1(\Omega)$ .*

*Proof.* First we want to show that  $a(u, v)$  is coercive, i.e. that there exists  $C > 0$  such that

$$|a(u, u)| \geq C\|u\| \quad \forall u \in \tilde{H}^1(\Omega). \quad (2.31)$$

This comes directly from  $\sigma$  being bounded from below, since

$$|a(u, u)| \geq \inf_{x \in \Omega} \{\sigma(x)\} \int_{\Omega} |\nabla u|^2 dx \geq c_1 \int_{\Omega} |\nabla u|^2 dx = c_1 \|u\|_{\tilde{H}^1(\Omega)}. \quad (2.32)$$

Then since  $\tilde{H}^1(\Omega)$  is a Hilbert space,  $a(u, v)$  is a bounded, coercive bilinear form and  $l(v)$  a bounded linear functional on  $\tilde{H}^1(\Omega)$ , the Lax–Milgram Theorem states that there is a unique  $u \in \tilde{H}^1(\Omega)$  such that

$$a(u, v) = l(v) \quad \forall v \in \tilde{H}^1(\Omega), \quad (2.33)$$

which completes the proof.  $\square$

In a numerical setting, the integral condition on functions in  $\tilde{H}^1(\Omega)$  is not very convenient to implement. Hence, one might choose a different but equivalent formulation, which uses  $H^1(\Omega)$ . Instead of solving (2.23), suppose we want to solve for  $(u, c) \in H^1(\Omega) \times \mathbb{R}$ , the problem,

$$A((u, c), (v, d)) = L((v, d)) \quad \forall (v, d) \in H^1(\Omega) \times \mathbb{R}, \quad (2.34)$$

where  $A : (H^1(\Omega) \times \mathbb{R}) \times (H^1(\Omega) \times \mathbb{R}) \rightarrow \mathbb{R}$  is the augmented bilinear form, defined by

$$A((u, c), (v, d)) := \int_{\Omega} \sigma \nabla u \cdot \nabla v dx + d \int_{\partial\Omega} u dS + c \int_{\partial\Omega} v dS. \quad (2.35)$$

The functional  $L : H^1(\Omega) \times \mathbb{R} \rightarrow \mathbb{R}$  extends the functional  $l$  to the product space,

$$L((v, d)) := l(v) = \int_{\partial\Omega} f v dS. \quad (2.36)$$

Using (2.34), we no longer require  $u, v$  or  $f$  to integrate to zero on the boundary. However, we need to prove that there is an equivalence between the problems (2.23) and (2.34).

**Corollary 1** (Uniqueness of the augmented form). *The problem (2.23) has a unique solution  $u \in \tilde{H}^1(\Omega)$  if and only if  $(u, c)$  is a unique solution of (2.34) with  $c = |\partial\Omega|^{-1} \int_{\partial\Omega} f dS$ .*



*Proof.* For the " $\Leftarrow$ "-direction, suppose  $(u, c)$  solves (2.34). By the assumption, (2.34) holds for all  $(v, d) \in H^1(\Omega) \times \mathbb{R}$ . Choosing the trial function  $(v = 0, d \neq 0)$  immediately gives

$$d \int_{\partial\Omega} u \, dS = 0, \quad d \neq 0, \quad \text{implying} \quad \int_{\partial\Omega} u \, dS = 0. \quad (2.37)$$

This shows that  $u \in \tilde{H}^1(\Omega)$ . Then choosing  $v \in \tilde{H}^1(\Omega)$ ,  $d = 0$ , gives

$$\int_{\Omega} \sigma \nabla u \cdot \nabla v \, dx = \int_{\partial\Omega} f v \, dS, \quad \forall v \in \tilde{H}^1(\Omega), \quad (2.38)$$

which shows that  $u$  is a solution of (2.23). By Th. 1, this is unique.

" $\Rightarrow$ ": Now suppose  $u$  solves (2.23). Since  $u \in \tilde{H}^1(\Omega)$ , then  $\int_{\partial\Omega} u \, dS = 0$ . Moreover, for  $v \in \tilde{H}^1(\Omega)$ , (2.34) reduces to (2.23) and hence  $u$  solves (2.34) for all  $(v, d) \in \tilde{H}^1(\Omega) \times \mathbb{R}$ .

Then we only need to show that  $u$  solves (2.34) for  $v \in H^1(\Omega) \setminus \tilde{H}^1(\Omega)$ . Consider  $\tilde{v} = v - v_0$ , where  $v_0 = |\partial\Omega|^{-1} \int_{\partial\Omega} v \, dS$ . Then  $\tilde{v} \in \tilde{H}^1(\Omega)$ , since

$$\int_{\partial\Omega} \tilde{v} \, dS = \int_{\partial\Omega} v - \left( |\partial\Omega|^{-1} \int_{\partial\Omega} v \, dS \right) dS = \int_{\partial\Omega} v \, dS - \int_{\partial\Omega} v \, dS = 0. \quad (2.39)$$

By the fact that (2.23) holds for all  $v \in \tilde{H}^1(\Omega)$ , we obtain

$$\int_{\Omega} \sigma \nabla u \cdot \nabla \tilde{v} \, dx = \int_{\partial\Omega} f \tilde{v} \, dS. \quad (2.40)$$

Since  $v_0$  is a constant, we have  $\nabla v_0 = 0$  a.e., and

$$\int_{\partial\Omega} f \tilde{v} \, dS = \int_{\Omega} \sigma \nabla u \cdot \nabla \tilde{v} \, dx = \int_{\Omega} \sigma \nabla u \cdot \nabla (v - v_0) \, dx = \int_{\Omega} \sigma \nabla u \cdot \nabla v \, dx. \quad (2.41)$$

On the other hand, for (2.34) to be true, we require

$$\int_{\Omega} \sigma \nabla u \cdot \nabla v \, dx = \int_{\partial\Omega} f v \, dS - c \int_{\partial\Omega} v \, dS, \quad (2.42)$$

using that  $\int_{\partial\Omega} u \, dS = 0$  by assumption. Equating (2.41) and (2.42), we get

$$\int_{\partial\Omega} f \tilde{v} \, dS = \int_{\partial\Omega} f v \, dS - c \int_{\partial\Omega} v \, dS. \quad (2.43)$$

Since  $\tilde{v} = v - v_0$ , we have

$$v_0 \int_{\partial\Omega} f \, dS = c \int_{\partial\Omega} v \, dS = c |\partial\Omega| v_0. \quad (2.44)$$

which is fulfilled for all  $v \in H^1(\Omega)$  if only if

$$c = |\partial\Omega|^{-1} \int_{\partial\Omega} f \, dS, \quad (2.45)$$

which guarantees that  $(u, c) \in H^1(\Omega) \times \mathbb{R}$  is unique.  $\square$

Numerically this formulation is advantageous, since we avoid working with the spaces  $\tilde{H}^1$ ,  $\tilde{H}^{1/2}$  and  $\tilde{H}^{-1/2}$  and instead work with  $H^1$ ,  $H^{1/2}$  and  $H^{-1/2}$ . Moreover, we preserve the symmetry of the bilinear form.

As we saw in (2.19),  $\int_{\partial\Omega} f dS = 0$  is a necessary condition for a solution to the strong formulation (2.1) and (2.5), while this is not required for solving the variational forms (2.23) and (2.34). Ideally, we would want our solution to also solve the strong formulation (2.1) or the integral formulation (2.5). At least we should require that  $\int_{\partial\Omega} f dS$  is close to zero. In practice,  $f$  is estimated by measured data on the boundary and there would be some noise, due to which  $\int_{\partial\Omega} f dS \neq 0$ .

Suppose there is a true  $f_0 \in \tilde{H}^{-1/2}(\partial\Omega)$  for which  $\int_{\partial\Omega} f_0 dS = 0$ . For some other measured  $f \in H^{-1/2}(\partial\Omega)$  not necessarily integrating to zero on the boundary, assume that

$$\|f - f_0\|_{L^2(\partial\Omega)} < \delta. \quad (2.46)$$

Then for solutions  $u, u_0 \in \tilde{H}^1(\Omega)$  to (2.23), corresponding to  $f$  and  $f_0$  respectively, we have

$$\|u - u_0\|_{\tilde{H}^1(\Omega)} \leq C\|f - f_0\|_{L^2(\partial\Omega)} = C\delta. \quad (2.47)$$

Considering the variational forms (2.23) for both solutions, we have

$$\int_{\Omega} \sigma \nabla u_0 \cdot \nabla v dx = \int_{\partial\Omega} f_0 v dS, \quad (2.48)$$

$$\int_{\Omega} \sigma \nabla u \cdot \nabla v dx = \int_{\partial\Omega} f v dS, \quad (2.49)$$

for all  $v \in \tilde{H}^1(\Omega)$ . Collecting these, we get

$$\int_{\Omega} \sigma \nabla(u - u_0) \cdot \nabla v dx = \int_{\partial\Omega} (f - f_0)v dS. \quad (2.50)$$

Using  $v = u - u_0 \in \tilde{H}^1(\Omega)$  as a test function and global lower bound  $c_1 > 0$  for  $\sigma \in L_+^\infty(\Omega)$ ,

$$\int_{\Omega} |\nabla(u - u_0)|^2 dx \leq c_1^{-1} \int_{\partial\Omega} (f - f_0)(u - u_0) dS. \quad (2.51)$$

Applying the Cauchy-inequality and the Trace Theorem,

$$\|\nabla(u - u_0)\|_{L^2(\Omega)}^2 \leq c_1^{-1} \|f - f_0\|_{L^2(\partial\Omega)} \|T(u - u_0)\|_{L^2(\partial\Omega)} \quad (2.52)$$

$$\leq C \|f - f_0\|_{L^2(\partial\Omega)} \|u - u_0\|_{H^1(\Omega)} \quad (2.53)$$

Identifying the left-hand side as the squared norm of  $u - u_0$  in  $\tilde{H}^1(\Omega)$  and using the norm equivalence of Remark 1, we obtain

$$\|u - u_0\|_{\tilde{H}^1(\Omega)} \leq C \|f - f_0\|_{L^2(\partial\Omega)} < C\delta, \quad (2.54)$$

thus having obtained a stability result for the approximate solution  $u$ , given a true solution  $u_0$  with measured and true data  $f$  and  $f_0$  respectively. This ensures that there is at most a linear dependence on the measurement noise  $\delta$  in the sense of (2.54).

However, the dependence on noise is much more severe (logarithmic) in the case of the *inverse* EIT problem, which is one of the reasons that hybrid inverse problems such as Problem 1 is investigated, providing more stability.

## 2.2 An analytical solution

In this section, we provide an example of the forward problem (2.1) where it is possible to find an analytical solution. We choose  $\Omega$  to be the unit disk in  $\mathbb{R}^2$ . This region will be used throughout the thesis for examples or implementations. In order to have a specific forward problem to solve, we need to choose  $\sigma \in L_+^\infty(\Omega)$  and  $f \in \tilde{H}^{-1/2}(\Omega)$ . Given that  $\Omega$  is now a circular domain, we naturally express the problem in polar coordinates  $(r, \theta)$ .

We choose  $\sigma$  to be the piece-wise constant, radially symmetric function,

$$\sigma(r, \theta) = \begin{cases} 1 & \text{for } 0 \leq r \leq r_0 \\ 1 + C & \text{for } r_0 < r \leq 1 \end{cases} \quad (2.55)$$

and the Neumann data  $f$  is chosen to be

$$f(\theta) = \cos(k\theta), \quad k \in \mathbb{N}. \quad (2.56)$$

Note that  $\int_{\partial\Omega} f \, ds = 0$  for all  $k \in \mathbb{N}$  as required for a strong solution to exist.

Due to the symmetry in this problem, we can assume that the solution can be written  $u(r, \theta) = R(r)\Theta(\theta)$  by standard separation of variables. Since the outward normal vector is parallel to the radial tangent vector, the Neumann condition is simply

$$\sigma R'(r)\Theta(\theta) = \cos(k\theta). \quad (2.57)$$

This already implies that  $R'(1)\Theta(\theta) = \cos(k\theta)$ . This constant can be fixed by simply setting  $R'(1) = 1$ , which means  $\Theta(\theta) = \cos(k\theta)$ .

Then we have also fixed the integral,

$$\int_{\partial\Omega} u \, ds = R(1) \int_{-\pi}^{\pi} \Theta(\theta) \, ds = 0, \quad (2.58)$$

ensuring  $u \in \tilde{H}^1(\Omega)$  provided  $R(r)$  is differentiable. Since  $\sigma$  is constant on each of the two subdomains  $r \leq r_0$  and  $r_0 < r$ , the first part of (2.1) becomes the Laplace-equation on the whole domain. That is

$$\Delta u = 0 \quad \text{in } \Omega, \quad (2.59)$$

where the Laplacian in polar coordinates is given by

$$\Delta = \frac{\partial^2}{\partial r^2} \frac{1}{r^2} + \frac{1}{r} \frac{\partial}{\partial r} + \frac{\partial^2}{\partial \theta^2}. \quad (2.60)$$

Hence, (2.59) becomes

$$R''\Theta + \frac{1}{r}R'\Theta + \frac{1}{r^2}\Theta'' = 0 \quad \text{in } \Omega. \quad (2.61)$$

Now since  $\Theta'' = -k^2 \cos(k\theta) = -k^2\Theta$ , we have

$$\left( R'' + \frac{1}{r}R' - \frac{k^2}{r^2} \right) \Theta = 0 \quad \text{in } \Omega. \quad (2.62)$$

We then split  $R$  into two functions, each defined on a subdomain of  $\Omega$ .

$$R(r, \theta) = \begin{cases} R_1(r) & \text{for } 0 \leq r \leq r_0 \\ R_2(r) & \text{for } r_0 < r \leq 1. \end{cases} \quad (2.63)$$

Requiring continuity of  $R$ , we have  $R_1(r_0) = R_2(r_0)$ . Requiring  $\sigma \nabla u$  to be continuous, we also have  $R_1'(r_0) = (1 + C)R_2'(r_0)$ . Together with the condition  $R'(1) = 1$  from before, we have in total

$$R_1'' + \frac{1}{r}R_1' - \frac{k^2}{r^2} = 0 \quad \text{for } 0 \leq r \leq r_0 \quad (2.64)$$

$$R_2'' + \frac{1}{r}R_2' - \frac{k^2}{r^2} = 0 \quad \text{for } r_0 < r \leq 1. \quad (2.65)$$

$$R_1(r_0) = R_2(r_0) \quad (2.66)$$

$$(1 + C)R_1'(r_0) = R_2'(r_0) \quad (2.67)$$

$$R_2'(1) = 1 \quad (2.68)$$

$$R(r) < \infty \quad \text{in } \Omega. \quad (2.69)$$

The 2nd order ODE's in (2.64) and (2.65) are Euler equations with the known solutions,

$$R_1(r) = ar^k + cr^{-k} \quad (2.70)$$

$$R_2(r) = cr^k + dr^{-k}, \quad a, b, c, d \in \mathbb{R}. \quad (2.71)$$

From the boundedness of  $R_1$  we require  $b = 0$ . The other three boundary conditions give

$$a = c + r_0^{-2k}d \quad (2.72)$$

$$(1 + C)a = c - r_0^{-2k}d \quad (2.73)$$

$$k(c - d) = 1. \quad (2.74)$$

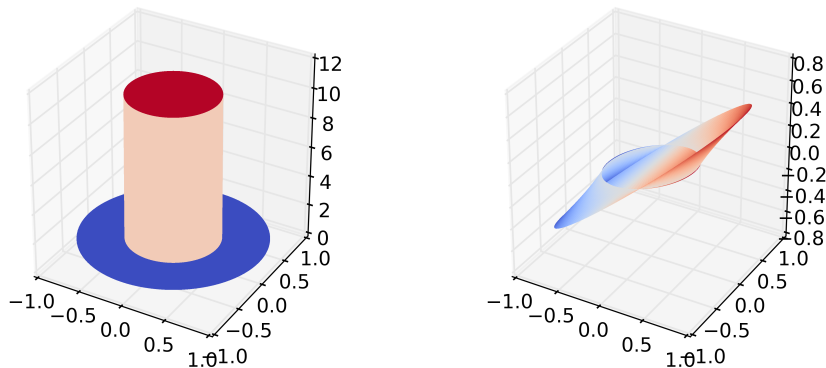
These are linear equations in  $a, c, d$ , and the solution is

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = K \begin{bmatrix} 2 \\ 0 \\ C + 2 \\ -Cr_0^{2k} \end{bmatrix}, \quad K = \frac{1}{k(Cr_0^{2k} + C + 2)}. \quad (2.75)$$

The forward solution then becomes

$$u(r, \theta) = \begin{cases} 2Kr^k \cos(k\theta) & \text{for } 0 \leq r \leq r_0 \\ K[(C + 2)r^k - Cr_0^{2k}r^{-k}] \cos(k\theta) & \text{for } r_0 < r \leq 1. \end{cases} \quad (2.76)$$

In Fig. 2.1, the conductivity is plotted for  $C = 10$ ,  $r_0 = \frac{1}{2}$  (left) along with the solution  $u(r, \theta)$  for  $k = 1$  (right). Note, that in this case,  $f(\theta) = \cos(\theta)$ , which corresponds to



(a)  $\sigma(r, \theta)$  with  $C = 10$

(b) The solution  $u(r, \theta)$  with  $C = 10$ ,  $k = 1$ .

**Figure 2.1:** (a) The piece-wise constant conductivity  $\sigma$  (b) and the solution  $u(r, \theta)$

$f(x) = x$  in cartesian coordinates. While  $\sigma$  is discontinuous at  $r = r_0$ , the solution  $u$  is continuous in all of the domain. However, since  $\sigma \nabla u$  is assured to be continuous, we cannot expect  $\nabla u$  itself to be continuous — again due to the discontinuity in  $\sigma$ .

### 2.2.1 Power density data

In this section, we will compute the interior power density data  $H(\sigma) = \sigma |\nabla u|^p$ ,  $p > 0$  used for the hybrid inverse problem, Problem 1. In the current analysis of the forward

problem, it would be interesting to see how  $H(\sigma)$  behaves given the discontinuity around  $r = r_0$ . Specifically, we would like  $H$  to be well-defined as a mapping into  $L^2(\Omega)$ . In order to examine whether this is the case, we recall that

$$u(r, \theta) = \begin{cases} u_1(r, \theta) = ar^k \cos(k\theta) & \text{for } 0 \leq r \leq r_0 \\ u_2(r, \theta) = (cr^k + dr^{-k}) \cos(k\theta) & \text{for } r_0 < r \leq 1. \end{cases} \quad (2.77)$$

where the constants  $a, c, d \in \mathbb{R}$ . We then compute the gradients in polar coordinates,

$$\nabla u_1 = akr^{k-1} \begin{bmatrix} \cos(k\theta) \\ \sin(k\theta) \end{bmatrix}, \quad \nabla u_2 = kr^{k-1} \begin{bmatrix} (c-d) \cos(k\theta) \\ -(c+d) \sin(k\theta) \end{bmatrix}$$

with the polar tangent vector basis  $\hat{e}_r = (1, 0)^T$ ,  $\hat{e}_\theta = (0, 1)^T$ . The euclidean norm of the gradients then becomes,

$$|\nabla u_1| = akr^{k-1} \quad |\nabla u_2| = \frac{k}{r} \left[ c^2 r^{2k} + d^2 r^{-2k} + 2cd(1 - 2\cos^2(k\theta)) \right]^{1/2}$$

Hence, the power density becomes

$$H(\sigma) = \begin{cases} H_1(\sigma_1) = \sigma_1 |\nabla u_1|^p = a^p k^p r^{p(k-1)} & \text{for } 0 \leq r \leq r_0 \\ H_2(\sigma_2) = \sigma_2 |\nabla u_2|^p \\ \quad = (1+C) \left(\frac{k}{r}\right)^p \left[ c^2 r^{2k} + d^2 r^{-2k} + 2cd(1 - 2\cos^2(k\theta)) \right]^{p/2} & \text{for } r_0 < r \leq 1. \end{cases}$$

In order to evaluate whether  $H(\sigma) \in L^2(\Omega)$  we must consider the two integrals,

$$I_1 = \int_{\Omega_i} |H_1(\sigma_1)|^2 dx, \quad I_2 = \int_{\Omega_o} |H_2(\sigma_2)|^2 dx$$

where  $\Omega_i$  and  $\Omega_o$  denote the inner and outer part of  $\Omega$  respectively. The first integral is rather straight-forward and simply gives

$$I_1 = 2\pi a^{2p} k^{2p} \int_0^{r_0} r^{2p(k-1)} r dr = \frac{\pi (ak)^{2p} r_0^{2p(k-1)+2}}{p(k-1)+1} < \infty \quad \forall p > 0. \quad (2.78)$$

Note that all the constants in front of the integral are strictly larger than zero, where we assume  $k \in \mathbb{N}_+$ . The case  $k = 0$  would simply give  $H_1(\sigma_1) = 0$  in  $\Omega_i$ . The second integral becomes a little more complicated, but we can quite easily find an upper bound. First we find that

$$|H_2(\sigma_2)|^2 \leq (1+C)^2 \left(\frac{k}{r}\right)^{2p} (c^2 r^{2k} + d^2 r^{-2k} + 2cd)^p \quad (2.79)$$

Since  $0 < r_0 < 1$  and  $k, p > 0$ , we also get

$$|H_2(\sigma_2)|^2 \leq (1+C)^2 \left(\frac{k}{r_0}\right)^{2p} (c^2 + d^2 r_0^{-2k} + 2cd)^p, \quad (2.80)$$

which is just a constant upper bound for  $|H_2(\sigma_2)|$ . So in fact  $H_2(\sigma_2) \in L^\infty(\Omega_o)$ . Consequently, we have

$$I_2 \leq 2\pi(1+C)^2 \frac{k^{2p}}{r_0^{2p}} (c^2 + d^2 r_0^{-2k} + 2cd)^p \int_{r_0}^1 r \, dr \quad (2.81)$$

$$= \pi(1-r_0^2)(1+C)^2 \frac{k^{2p}}{r_0^{2p}} (c^2 + d^2 r_0^{-2k} + 2cd)^p < \infty \quad \forall p > 0. \quad (2.82)$$

In total,

$$\|H(\sigma)\|_{L^2(\Omega)} = I_1 + I_2 < \infty \quad \forall p > 0, \quad (2.83)$$

which holds for all possible values of the parameters  $k$ ,  $r_0$ ,  $C$ ,  $a$ ,  $c$  and  $d$ . So it turns out that the power density mapping  $H$  behaves nicely in the sense that it is well-defined as a mapping into  $L^2(\Omega)$  at least for *this choice* of the conductivity and Neumann data. Of course, it cannot necessarily be generalized to any piece-wise conductivity and any smooth Neumann data function  $f$  on the boundary.





## Chapter 3

# Methods for solving inverse problems

This chapter focuses on the strategies used in this thesis for solving hybrid EIT inverse problems given interior data of certain types. Since these problems involve an approximate inversion of a non-linear map between Banach spaces (or other function spaces), this typically requires numerical schemes in practice. This project uses a linearisation approach, where the map of the conductivity  $\sigma$  into the resulting electric potential  $u$ , is linearised. Since linearisations require the knowledge of the derivative of a map, the Fréchet derivative is introduced as the appropriate concept. With the Fréchet derivative of the forward map it is possible to construct a Newton-type method, where each iteration involves solving a linearised problem, in order to approximately solve the conductivity reconstruction problem. Specifically, we will derive methods to solve the Hybrid EIT inverse problem given power density interior data.

### 3.1 Continuity and differentiability of the forward map

The first step in deriving a method for solving the inverse problem is to analyse the forward map, which maps the conductivity  $\sigma \in L^\infty_+(\Omega)$  into the solution  $u \in \tilde{H}^1(\Omega)$  of the forward problem. This gives the following definition.

**Definition 2** (Forward map of EIT). *Let  $\Omega$  be a bounded domain in  $\mathbb{R}^2$  or  $\mathbb{R}^3$ , with a smooth boundary  $\partial\Omega$ . Given  $\sigma \in L_+^\infty(\Omega)$ ,  $f \in \tilde{H}^{-1/2}(\Omega)$ , we define the forward map of the EIT problem as  $U : L_+^\infty(\Omega) \rightarrow \tilde{H}^1(\Omega)$  given by  $U(\sigma) = u$  where  $u \in \tilde{H}^1(\Omega)$  is the unique solution of the problem*

$$\begin{aligned} \nabla \cdot (\sigma \nabla u) &= 0 \quad \text{in } \Omega \\ \sigma \frac{\partial u}{\partial n} &= f \quad \text{on } \partial\Omega, \end{aligned} \tag{3.1}$$

with the corresponding weak formulation,

$$\int_{\Omega} \sigma \nabla u \cdot \nabla v \, dx = \int_{\partial\Omega} f v \, dS \quad \forall v \in \tilde{H}^1(\Omega). \tag{3.2}$$

A fundamental property for the application of iterative methods is the continuity of the forward map. As  $U(\sigma)$  is only implicitly given as the solution of the variational formulation in (3.2), this is our starting point for obtaining a continuity and a differentiability result for  $U$ .

We begin by considering an arbitrary  $\sigma \in L_+^\infty(\Omega)$  and a perturbation  $\delta\sigma \in L^\infty(\Omega)$  such that  $\sigma + \delta\sigma \in L_+^\infty(\Omega)$ , which is of course only guaranteed if  $\delta\sigma$  is "sufficiently" small. To be more precise, it would be sufficient that

$$\|\delta\sigma\|_{L^\infty(\Omega)} < c_1 \tag{3.3}$$

where  $c_1 > 0$  is again the lower bound for  $\sigma$  meaning  $\sigma(x) > c_1$  for a.e.  $x \in \Omega$ . The continuity result is similar to the ones in [13, 20] and can be stated as follows.

**Theorem 2** (Continuity). *Let  $\sigma, \sigma + \delta\sigma \in L_+^\infty(\Omega)$  and  $\delta\sigma \in L^\infty(\Omega)$ . Then there exists a  $C > 0$  independent of  $\delta\sigma$  s.t.*

$$\|U(\sigma + \delta\sigma) - U(\sigma)\|_{\tilde{H}^1(\Omega)} \leq C \|\delta\sigma\|_{L^\infty(\Omega)}. \tag{3.4}$$

*Proof.* By the definition of  $U$  and the fact that the right-hand sides of (3.2) are identical for the two input conductivities,

$$\int_{\Omega} \sigma \nabla U(\sigma) \cdot \nabla v \, dx = \int_{\Omega} (\sigma + \delta\sigma) \nabla U(\sigma + \delta\sigma) \cdot \nabla v \, dx \quad \forall v \in \tilde{H}^1(\Omega), \tag{3.5}$$

implying that

$$\int_{\Omega} \sigma \nabla (U(\sigma) - U(\sigma + \delta\sigma)) \cdot \nabla v \, dx = \int_{\Omega} \delta\sigma \nabla U(\sigma + \delta\sigma) \cdot \nabla v \, dx \quad \forall v \in \tilde{H}^1(\Omega). \tag{3.6}$$

Now, both  $U(\sigma)$  and  $U(\sigma + \delta\sigma)$  belong to  $\tilde{H}^1(\Omega)$ , whereby also  $U(\sigma + \delta\sigma) - U(\sigma) \in \tilde{H}^1(\Omega)$ . Hence, (3.6) holds with  $U(\sigma) - U(\sigma + \delta\sigma)$  as a test function, which means

$$\begin{aligned} \left| \int_{\Omega} \sigma \left| \nabla \left( U(\sigma) - U(\sigma + \delta\sigma) \right) \right|^2 \right| &= \left| \int_{\Omega} \delta\sigma \left[ \nabla U(\sigma + \delta\sigma) \cdot \nabla \left( U(\sigma) - U(\sigma + \delta\sigma) \right) \right] \right| \\ &\leq \|\delta\sigma\|_{L^\infty(\Omega)} \|\nabla U(\sigma + \delta\sigma) \cdot \nabla [U(\sigma) - U(\sigma + \delta\sigma)]\|_{L^1(\Omega)}. \end{aligned} \quad (3.7)$$

By the Cauchy-Schwarz inequality we have

$$\begin{aligned} \|\nabla U(\sigma + \delta\sigma) \cdot \nabla [U(\sigma) - U(\sigma + \delta\sigma)]\|_{L^1(\Omega)} &\leq \|\nabla [U(\sigma) - U(\sigma + \delta\sigma)]\|_{L^2(\Omega)} \\ &= \|U(\sigma + \delta\sigma)\|_{\tilde{H}^1(\Omega)} \|U(\sigma) - U(\sigma + \delta\sigma)\|_{\tilde{H}^1(\Omega)} \end{aligned}$$

and hence,

$$\left| \int_{\Omega} \sigma \left| \nabla \left( U(\sigma) - U(\sigma + \delta\sigma) \right) \right|^2 \right| \leq \|\delta\sigma\|_{L^\infty(\Omega)} \|U(\sigma + \delta\sigma)\|_{\tilde{H}^1(\Omega)} \|U(\sigma) - U(\sigma + \delta\sigma)\|_{\tilde{H}^1(\Omega)}.$$

By the lower bound  $\sigma(x) > c_1$  a.e. in  $\Omega$ , we obtain

$$c_1 \|U(\sigma) - U(\sigma + \delta\sigma)\|_{\tilde{H}^1(\Omega)}^2 \leq \left| \int_{\Omega} \sigma \left| \nabla \left( U(\sigma) - U(\sigma + \delta\sigma) \right) \right|^2 dx \right|. \quad (3.8)$$

Hence, we arrive at

$$\|U(\sigma) - U(\sigma + \delta\sigma)\|_{\tilde{H}^1(\Omega)}^2 \leq c_1^{-1} \|\delta\sigma\|_{L^\infty(\Omega)} \|U(\sigma + \delta\sigma)\|_{\tilde{H}^1(\Omega)} \|U(\sigma) - U(\sigma + \delta\sigma)\|_{\tilde{H}^1(\Omega)},$$

implying that

$$\|U(\sigma) - U(\sigma + \delta\sigma)\|_{\tilde{H}^1(\Omega)} \leq c_1^{-1} \|\delta\sigma\|_{L^\infty(\Omega)} \|U(\sigma + \delta\sigma)\|_{\tilde{H}^1(\Omega)}. \quad (3.9)$$

Using  $v = U(\sigma + \delta\sigma)$  as a test function in (3.2) along with the Cauchy-Schwarz inequality and the Trace Theorem [11] we get the estimate,

$$\|U(\sigma + \delta\sigma)\|_{\tilde{H}^1(\Omega)} \leq c_1^{-1} \|f\|_{L^2(\partial\Omega)}, \quad (3.10)$$

which means

$$\|U(\sigma) - U(\sigma + \delta\sigma)\|_{\tilde{H}^1(\Omega)} \leq c_1^{-2} \|\delta\sigma\|_{L^\infty(\Omega)} \|f\|_{L^2(\partial\Omega)} = C \|\delta\sigma\|_{L^\infty(\Omega)} \quad (3.11)$$

concluding the proof.  $\square$

### 3.1.1 Fréchet differentiability of the forward map

Having obtained a continuity result, we will now proceed to prove differentiability of the forward map in a similar sense to Fréchet differentiability in Banach spaces.

For a function on the real numbers,  $f : \mathbb{R} \rightarrow \mathbb{R}$  we have the notion of the derivative of  $f$  at a point  $x_0$  as the limit,

$$f'(x_0) = \lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h}, \quad (3.12)$$

whenever the limit exists. The concept of Fréchet derivatives generalizes this to maps between Banach spaces [23].

**Definition 3** (Fréchet derivative). *Let  $V$  and  $W$  be Banach spaces and  $U \subset V$  be an open subset of  $V$ . A map,  $f : U \rightarrow W$  is then said to be Fréchet differentiable at  $u \in U$  if there exists a bounded linear operator  $A : V \rightarrow W$  such that*

$$\lim_{\eta \rightarrow 0} \frac{\|f(u + \eta) - f(u) - A\eta\|_W}{\|\eta\|_V} \quad (3.13)$$

and  $A$  is called the Fréchet derivative of  $f$  at  $u \in U$ .

In the following, we want to show that the forward map  $U : L^\infty(\Omega) \rightarrow \tilde{H}^1(\Omega)$  is Fréchet differentiable. The proof is similar to the one given in [19] but is generalized to the Neumann condition, which eventually changes the classical formulation. Also, here the approach takes the variational form as a starting point rather than the classical formulation. Since we know the forward map only implicitly by (3.2), we can only turn to this equation to find a first order derivative of  $U$ . With  $\sigma$  and  $\delta\sigma$  as in Th. 2, we have from (3.5) that

$$\int_{\Omega} \sigma \nabla [U(\sigma + \delta\sigma) - U(\sigma)] \cdot \nabla v \, dx = - \int_{\Omega} \delta\sigma \nabla U(\sigma + \delta\sigma) \cdot \nabla v \, dx \quad (3.14)$$

for all  $v \in \tilde{H}^1(\Omega)$ . By the continuity of  $U$  stated in (3.4), we have

$$\nabla U(\sigma + \delta\sigma) = \nabla U(\sigma) + [\nabla U(\sigma + \delta\sigma) - \nabla U(\sigma)] = \nabla U(\sigma) + O(\|\delta\sigma\|_{L^\infty(\Omega)}). \quad (3.15)$$

This gives us the first order estimate

$$\delta\sigma \nabla U(\sigma + \delta\sigma) = \delta\sigma \nabla U(\sigma) + O\left(\|\delta\sigma\|_{L^\infty(\Omega)}^2\right) \approx \delta\sigma \nabla U(\sigma). \quad (3.16)$$

Hence we arrive at

$$\int_{\Omega} \sigma \nabla [U(\sigma + \delta\sigma) - U(\sigma)] \cdot \nabla v \, dx = - \int_{\Omega} \delta\sigma \nabla U(\sigma) \cdot \nabla v \, dx. \quad (3.17)$$

Writing  $u = U(\sigma)$  and  $\delta u = U(\sigma + \delta\sigma) - U(\sigma)$  we get

$$\int_{\Omega} \sigma \nabla \delta u \cdot \nabla v \, dx = - \int_{\Omega} \delta\sigma \nabla u \cdot \nabla v \, dx \quad \forall v \in \tilde{H}^1(\Omega). \quad (3.18)$$

So to first order, the increment  $\delta u$  can be approximated by solving the variational form (3.18). As we shall see, such a first order approximation is exactly what defines the Fréchet derivative in the weak sense. However, we will also need a strong version of (3.18). As we have already discovered deriving the variational form of the forward problem (3.2), the operator  $\nabla \cdot (\sigma \nabla [\cdot])$  can be understood by its variational form for any test function  $v \in \tilde{H}^1(\Omega)$ . More precisely,

$$\int_{\Omega} [\nabla \cdot (\delta \sigma \nabla u)] v \, dx = \int_{\partial \Omega} \delta \sigma \frac{\partial u}{\partial n} v - \int_{\Omega} \delta \sigma \nabla u \cdot \nabla v \, dx \quad \text{and} \quad (3.19)$$

$$\int_{\Omega} [\nabla \cdot (\sigma \nabla \delta u)] v \, dx = \int_{\partial \Omega} \sigma \frac{\partial \delta u}{\partial n} v - \int_{\Omega} \sigma \nabla \delta u \cdot \nabla v \, dx. \quad (3.20)$$

From the Neumann condition of the (strong) forward problem (3.1), we see that

$$(\sigma + \delta \sigma) \frac{\partial U(\sigma + \delta \sigma)}{\partial n} = \sigma \frac{\partial U(\sigma)}{\partial n} = f. \quad (3.21)$$

This means

$$\sigma \frac{\partial \delta u}{\partial n} = -\delta \sigma \frac{\partial U(\sigma + \delta \sigma)}{\partial n} \approx -\delta \sigma \frac{\partial u}{\partial n} \quad (3.22)$$

by the same first order estimation as before. Using (3.18) and the fact that the boundary terms in (3.19) and (3.20) cancel each other out due to (3.22), we have

$$\int_{\Omega} [\nabla \cdot (\delta \sigma \nabla u)] v \, dx = \int_{\Omega} [\nabla \cdot (\sigma \nabla \delta u)] v \, dx \quad \forall v \in \tilde{H}^1(\Omega). \quad (3.23)$$

Hence, we have a corresponding formulation boundary value problem for  $\delta u$  with a Neumann condition,

$$\begin{aligned} \nabla \cdot (\sigma \nabla \delta u) &= -\nabla \cdot (\delta \sigma \nabla u) \quad \text{in } \Omega, \\ \sigma \frac{\partial \delta u}{\partial n} &= -\delta \sigma \frac{\partial u}{\partial n} \quad \text{on } \partial \Omega, \end{aligned} \quad (3.24)$$

which has the variational form

$$\int_{\Omega} \nabla \delta u \cdot \nabla v \, dx = - \int_{\Omega} \delta \sigma \nabla u \cdot \nabla v \, dx \quad \forall v \in \tilde{H}^1(\Omega), \quad (3.25)$$

given  $\sigma \in L_+^\infty(\Omega)$  and the forward solution at  $\sigma$  denoted  $u = U(\sigma)$ .

Now we are ready to show that  $U$  has a Fréchet derivative given by the solution of (3.25).

**Lemma 1** (Fréchet differentiability of  $U$ ). *Let  $U$  be the forward map from Def. 2. Then  $U$  has a Fréchet derivative in the sense that there exists a bounded linear operator  $dU : L_+^\infty(\Omega) \rightarrow \tilde{H}^1(\Omega)$  such that for  $\delta \sigma \in L^\infty(\Omega)$ ,*

$$\lim_{\delta \sigma \rightarrow 0} \frac{\|U(\sigma + \delta \sigma) - U(\sigma) - dU(\delta \sigma)\|_{\tilde{H}^1(\Omega)}}{\|\delta \sigma\|_{L^\infty(\Omega)}}. \quad (3.26)$$

Given a reference potential  $u \in \tilde{H}^1(\Omega)$  and a reference conductivity  $\sigma \in L_+^\infty(\Omega)$ , the Fréchet derivative in the direction  $\delta\sigma \in L^\infty(\Omega)$  is then given by the solution  $\delta u \in \tilde{H}^1(\Omega)$  to the boundary value problem,

$$\begin{aligned} \nabla \cdot (\sigma \nabla \delta u) &= -\nabla \cdot (\delta\sigma \nabla u) && \text{in } \Omega, \\ \sigma \frac{\partial \delta u}{\partial n} &= -\delta\sigma \frac{\partial u}{\partial n} && \text{on } \partial\Omega, \end{aligned} \quad (3.27)$$

with the corresponding variational form,

$$\int_{\Omega} \nabla \delta u \cdot \nabla v \, dx = - \int_{\Omega} \delta\sigma \nabla u \cdot \nabla v \, dx \quad \forall v \in \tilde{H}^1(\Omega). \quad (3.28)$$

*Proof.* The linearity of  $dU$  comes directly from (3.27) and (3.28) both being linear in  $\delta\sigma$ .

Boundedness can be obtained by considering the variational form (3.28),

$$\int_{\Omega} \sigma \nabla \delta u \cdot \nabla v \, dx = - \int_{\Omega} \delta\sigma \nabla \tilde{u} \cdot \nabla v \, dx \quad \forall v \in \tilde{H}^1(\Omega). \quad (3.29)$$

Using  $v = \delta u$  as a test function, we get

$$\|\nabla \delta u\|_{L^2(\Omega)}^2 \leq c_1^{-1} \left| \int_{\Omega} \sigma |\nabla \delta u|^2 \, dx \right| = c_1^{-1} \left| \int_{\Omega} \delta\sigma \nabla \tilde{u} \cdot \nabla \delta u \, dx \right| \quad (3.30)$$

$$\leq c_1^{-1} \|\delta\sigma\|_{L^\infty(\Omega)} \|\nabla \tilde{u} \cdot \nabla \delta u\|_{L^1(\Omega)}, \quad (3.31)$$

again with  $c_1$  as the lower bound of  $\sigma$ . Applying the Cauchy-Schwarz inequality, we find

$$\|\nabla \delta u\|_{L^2(\Omega)}^2 \leq c_1^{-1} \|\delta\sigma\|_{L^\infty(\Omega)} \|\nabla u\|_{L^2(\Omega)} \|\nabla \delta u\|_{L^2(\Omega)}, \quad (3.32)$$

whereby

$$\|\delta u\|_{\tilde{H}^1(\Omega)} = \|\nabla \delta u\|_{L^2(\Omega)} \leq c_1^{-1} \|\delta\sigma\|_{L^\infty(\Omega)} \|\nabla u\|_{L^2(\Omega)} = C \|\delta\sigma\|_{L^\infty(\Omega)}, \quad (3.33)$$

which proves that  $dU$  is bounded as an operator from  $L^\infty(\Omega)$  to  $\tilde{H}^1(\Omega)$ .

To prove the limit in (3.26) we define  $u_1 = U(\sigma)$ ,  $u_2 = U(\sigma + \delta\sigma)$ . Then we get from the definition of  $U$  that

$$\nabla \cdot [(\sigma + \delta\sigma) \nabla (u_2 - u_1 - \delta u)] = -\nabla \cdot (\delta\sigma \nabla \delta u), \quad (3.34)$$

which has the weak formulation,

$$\int_{\Omega} (\sigma + \delta\sigma) \nabla (u_2 - u_1 - \delta u) \cdot \nabla v \, dx = - \int_{\Omega} \delta\sigma \nabla \delta u \cdot \nabla v \, dx, \quad \forall v \in \tilde{H}^1(\Omega). \quad (3.35)$$

Since  $u_2 - u_1 - \delta u \in \tilde{H}^1(\Omega)$  we can choose this as a test-function. Then we get

$$\int_{\Omega} (\sigma + \delta\sigma) |\nabla(u_2 - u_1 - \delta u)|^2 dx = - \int_{\Omega} \delta\sigma \nabla\delta u \cdot \nabla(u_2 - u_1 - \delta u) dx. \quad (3.36)$$

Then since  $\sigma + \delta\sigma \in L_+^\infty(\Omega)$ , and hence is bounded from below by the constant  $c_1 > 0$ ,

$$c_1 \|u_2 - u_1 - \delta u\|_{\tilde{H}^1(\Omega)}^2 \leq \|\nabla\delta u \cdot \nabla(u_2 - u_1 - \delta u)\|_{L^1(\Omega)} \|\delta\sigma\|_{L^\infty(\Omega)} \quad (3.37)$$

$$\leq \|\delta u\|_{\tilde{H}^1(\Omega)} \|u_2 - u_1 - \delta u\|_{\tilde{H}^1(\Omega)} \|\delta\sigma\|_{L^\infty(\Omega)}. \quad (3.38)$$

By (3.33), we have that

$$\|\delta u\|_{\tilde{H}^1(\Omega)} \leq C_1 \|\delta\sigma\|_{L^\infty(\Omega)}, \quad \text{for some } C_1 > 0. \quad (3.39)$$

Hence,

$$\|u_2 - u_1 - \delta u\|_{\tilde{H}^1(\Omega)} \leq c_1^{-1} C_1 \|\delta\sigma\|_{L^\infty(\Omega)}^2 = C \|\delta\sigma\|_{L^\infty(\Omega)}^2 \quad (3.40)$$

and evaluating the quotient in (3.26),

$$\frac{\|u_2 - u_1 - \delta u\|_{\tilde{H}^1(\Omega)}}{\|\delta\sigma\|_{L^\infty(\Omega)}} \leq C \|\delta\sigma\|_{L^\infty(\Omega)} \rightarrow 0 \text{ for } \delta\sigma \rightarrow 0. \quad (3.41)$$

□

### 3.1.2 Fréchet derivative of the data map

The map that we actually want to invert in terms of the hybrid inverse problem (Problem 1) is the one taking the target conductivity into the power density data. That is the map  $H : L_+^\infty(\Omega) \rightarrow L^2(\Omega)$ , given by

$$H(\sigma) = \sigma |\nabla u|^p, \quad p > 0, \quad (3.42)$$

where  $u$  is given by the forward map of  $\sigma$ ,  $u = U(\sigma)$ . In order to derive an iterative method based on a linearisation of  $H$  we are interested in deriving the Fréchet derivative of  $H$ . First we write out  $H(\sigma)$  as the product,

$$H(\sigma) = h_1(\sigma) h_2(\sigma), \quad h_1(\sigma) = \sigma, \quad h_2(\sigma) = |\nabla U(\sigma)|^p, \quad (3.43)$$

For Fréchet derivatives, the rules of differentiation (the product rule, the sum rule and the chain rule) from the usual differential calculus hold [4] (Sec. 5.3). This means we can write

$$dH(\delta\sigma) = \left( h_1 \frac{\partial h_2}{\partial \sigma} + h_2 \frac{\partial h_1}{\partial \sigma} \right) \delta\sigma \quad (3.44)$$

where  $dH(\delta\sigma)$  is the Fréchet derivative of  $H$  in the direction  $\delta\sigma$ . For the derivative of  $h_2$ ,

$$\frac{\partial h_2}{\partial\sigma} = \frac{\partial}{\partial\sigma} (|\nabla u|^2)^{\frac{p}{2}} = \frac{p}{2} (|\nabla u|^2)^{\frac{p}{2}-1} \frac{\partial|\nabla u|^2}{\partial\sigma} = \frac{p}{|\nabla u|^{2-p}} \frac{\partial|\nabla u|^2}{\partial\sigma}. \quad (3.45)$$

To compute the last term  $\frac{\partial|\nabla u|^2}{\partial\sigma}$  we write for  $d = 3$  (equivalent for  $d = 2$ ),

$$\frac{\partial|\nabla u|^2}{\partial\sigma} = \frac{\partial}{\partial\sigma} (\nabla u \cdot \nabla u) = \frac{\partial}{\partial\sigma} (u_x^2 + u_y^2 + u_z^2) \quad (3.46)$$

$$= 2 \left( u_x \frac{\partial u_x}{\partial\sigma} + u_y \frac{\partial u_y}{\partial\sigma} + u_z \frac{\partial u_z}{\partial\sigma} \right) = 2(\nabla u \cdot \nabla dU), \quad (3.47)$$

where  $dU$  is the operator representing the Fréchet derivative of the forward map from Lemma 1. Collecting this we obtain,

$$\frac{\partial h_2}{\partial\sigma} \delta\sigma = p \frac{\nabla u \cdot \nabla dU}{|\nabla u|^{2-p}} \delta\sigma = p \frac{\nabla u \cdot \nabla \delta u}{|\nabla u|^{2-p}}, \quad (3.48)$$

where  $\delta u$  is the directional (Fréchet) derivative  $\delta u = dU(\delta\sigma)$ .

Finally we get from  $H(\sigma) = h_1(\sigma)h_2(\sigma)$ , that

$$dH(\delta\sigma) = \left( \frac{\partial h_1}{\partial\sigma} h_2 + \frac{\partial h_2}{\partial\sigma} h_1 \right) \delta\sigma = |\nabla u|^p \delta\sigma + p\sigma \frac{\nabla u \cdot \nabla \delta u}{|\nabla u|^{2-p}} \quad (3.49)$$

$$= |\nabla u|^p \left( \delta\sigma + p\sigma \frac{\nabla u \cdot \nabla \delta u}{|\nabla u|^2} \right). \quad (3.50)$$

So we might assume that  $dH$  represents a sort of derivative wrt.  $\sigma$ . Now we will prove that this is in fact the case, in the context of Fréchet derivatives, under certain assumptions.

**Theorem 3** (Fréchet differentiability of  $H$ ). *Let  $\sigma \in L_+^\infty(\Omega)$  and  $u \in \tilde{H}^1(\Omega)$  denote the solution of the forward problem (2) with  $\sigma \in L_+^\infty(\Omega)$ . Moreover assume that  $|\nabla u|$  is bounded from below by a positive constant and that  $|\nabla u|^p \in L^2(\Omega)$  for  $p > 0$ . Then the map,  $H : L_+^\infty(\Omega) \rightarrow L^2(\Omega)$ , mapping  $\sigma$  into the data,*

$$H : \sigma \mapsto \sigma |\nabla u|^p, \quad p > 0, \quad (3.51)$$

*has a Fréchet derivative in the sense that there exists a bounded linear operator  $dH : L^\infty(\Omega) \rightarrow L^2(\Omega)$  such that*

$$\lim_{\delta\sigma \rightarrow 0} \frac{\|H(\sigma + \delta\sigma) - H(\sigma) - dH(\delta\sigma)\|_{L^2(\Omega)}}{\|\delta\sigma\|_{L^\infty(\Omega)}}. \quad (3.52)$$

*provided  $\sigma + \delta\sigma \in L_+^\infty(\Omega)$ . The Fréchet derivative in the direction  $\delta\sigma$  is then denoted  $\delta h$  and is given by*

$$\delta h := dH(\delta\sigma) = |\nabla u|^p \left( \delta\sigma + p\sigma \frac{\nabla u \cdot \nabla \delta u}{|\nabla u|^2} \right). \quad (3.53)$$



*Proof.* The linearity of  $dH$  can be seen directly from (3.53), since  $\delta u$  is linear in  $\delta h$  from Lemma 1. For boundedness, we note that

$$\|\delta h\|_{L^2(\Omega)} \leq \|\delta\sigma|\nabla u|^p\|_{L^2(\Omega)} + \left\| p\sigma \frac{\nabla u \cdot \nabla \delta u}{|\nabla u|^2} \right\|_{L^2(\Omega)} \quad (3.54)$$

$$\leq \|\delta\sigma\|_{L^\infty(\Omega)} \|\nabla u\|_{L^2(\Omega)} + |p| \|\sigma\|_{L^\infty(\Omega)} \|\nabla u\|_{L^2(\Omega)}^{-1} \|\nabla \delta u\|_{L^2(\Omega)}, \quad (3.55)$$

using that  $|\nabla u|^p \in L^2(\Omega)$  by assumption. Now since we also assumed that  $|\nabla u|^{-1} \in L^\infty(\Omega)$  and that  $dU$  is bounded from Lemma 1,

$$\|\nabla \delta u\|_{L^2(\Omega)} = \|\delta u\|_{\tilde{H}^1} \leq C_1 \|\delta\sigma\|_{L^\infty}, \quad C_1 > 0, \quad (3.56)$$

we get that

$$\|\delta h\|_{L^2(\Omega)} \leq \|\delta\sigma\|_{L^\infty(\Omega)} \|\nabla u\|_{L^2(\Omega)} + |p| \|\sigma\|_{L^\infty(\Omega)} \|\nabla u\|_{L^\infty(\Omega)}^{-1} \|\delta u\|_{\tilde{H}^1} \quad (3.57)$$

$$\leq \left( \|\nabla u\|_{L^2(\Omega)} + C_1 |p| \|\sigma\|_{L^\infty(\Omega)} \|\nabla u\|_{L^\infty(\Omega)}^{-1} \right) \|\delta\sigma\|_{L^\infty(\Omega)} \quad (3.58)$$

$$= C \|\delta\sigma\|_{L^\infty(\Omega)}, \quad C > 0. \quad (3.59)$$

We are left to show that (3.52) is satisfied. By the derivation in (3.43)-(3.50), we obtain an expansion to first order of  $H(\sigma)$ ,

$$H(\sigma + \delta\sigma) - H(\sigma) = |\nabla u|^p \left( \delta\sigma + p\sigma \frac{\nabla u \cdot \nabla \delta u}{|\nabla u|^2} \right) + O(|\delta\sigma|^2). \quad (3.60)$$

This means that

$$\|H(\sigma + \delta\sigma) - H(\sigma) - dH(\delta\sigma)\|_{L^2(\Omega)} \leq C \|\delta\sigma\|_{L^2(\Omega)}^2, \quad (3.61)$$

implying that (3.52) is satisfied.  $\square$

## 3.2 Methods of the Newton-type

Now we turn our attention to the methods for solving the power density inverse problem, Prob. 1. In this section we will derive a Newton-type algorithm, using the linearisation of  $H$  (the forward data map) from before. This section is largely based on the work by Hoffmann[18].

### 3.2.1 Newton's method in Banach spaces

Suppose  $f : X \rightarrow Y$  is mapping between Banach spaces  $X$  and  $Y$ . Then Newton's method for solving the equation,  $f(x) = y$  for  $x \in X$ , may be stated as the iterative process of

solving at each step  $k \in \mathbb{N}$ ,

$$x_{k+1} = x_k - dF^{-1}(y - f(x_k)), \quad (3.62)$$

where  $dF$  denotes the Fréchet derivative of  $f$ . With the increment  $y - f(x_k)$  being approximated by the linear term, we could also write

$$y - f(x) \approx dF(x_{k+1} - x_k) = dF(\delta x), \quad (3.63)$$

using our previous notation.

Under certain conditions, among them that the starting point  $x_0$  lies within a certain neighbourhood of the solution, this iterative process converges to a solution of  $f(x) = y$ . That is,  $f(x_k) \rightarrow y$  in  $Y$  for  $k \rightarrow \infty$ . See [6], Theorem on p. 829.

However, the method is restrictive in the sense that it requires  $dF$  to be invertible at all the points in the chosen neighbourhood of the solution. Moreover, it requires  $X$  and  $Y$  to be Banach spaces which, as we say in the previous section, is not the case for  $L_+^\infty(\Omega)$ . Nonetheless, the general idea is still applicable for our purpose.

### 3.2.2 A Newton-type method given Power Density data

Let us consider Problem 1, where we want to reconstruct  $\sigma \in L_+^\infty(\Omega)$  from the power density data  $H(\sigma)$ .

In this case we have a non-linear map  $H : L_+^\infty(\Omega) \rightarrow L^2(\Omega)$ , with the Fréchet-like derivative given in Th. 3. However, we cannot expect  $dH$  to be invertible, so the method stated in (3.62) is not necessarily well-defined. However, we can use the equivalent of (3.63), being

$$H(\sigma) - H(\sigma_k) = dH|_\sigma(\delta\sigma_k). \quad (3.64)$$

The Fréchet derivative of  $H$  thus gives a linear approximation to the difference between the current approximation at step  $k$ ,  $H(\sigma_k)$ , and the true data  $H(\sigma)$ . Hence, we want to solve (3.64) for  $\delta\sigma$  to perform a Newton-type step, equivalent to computing the increment  $x_{k+1} - x_k$  in (3.62). Implicitly,  $dH$  depends on the Fréchet derivative of  $U$  as we saw in Section 3.1.2. So in order to find the direction  $\delta\sigma$  we can solve (3.64) and the PDE defining  $dU$ , (3.27), simultaneously.

Now we can define a Newton-type step for the inverse problem: Let  $\sigma_k \in L_+^\infty(\Omega)$  and  $u_k \in \tilde{H}^1(\Omega)$  be given reference functions, we want to solve the system for  $\delta\sigma_k \in L^\infty(\Omega)$ ,

$\delta u_k \in \tilde{H}^1(\Omega)$  at a step  $k \in \mathbb{N}$ ,

$$\nabla \cdot (\sigma_k \nabla \delta u_k) = -\nabla \cdot (\delta \sigma_k \nabla u_k) \quad \text{in } \Omega \quad (3.65)$$

$$\sigma_k \frac{\partial(\delta u_k)}{\partial n} = -\delta \sigma_k \frac{\partial u_k}{\partial n} \quad \text{on } \partial\Omega, \quad (3.66)$$

$$|\nabla u_k|^p \left( \delta \sigma_k + p \sigma_k \frac{\nabla u_k \cdot \nabla \delta u_k}{|\nabla u_k|^2} \right) = H(\sigma) - H(\sigma_k) \quad \text{in } \Omega. \quad (3.67)$$

In order to solve (3.65)-(3.67) in a Finite Element-setting, we need to convert the system to its weak formulation. Multiplying each equation by test functions,  $v_1 \in \tilde{H}^1(\Omega)$ ,  $v_2 \in L^\infty(\Omega)$ , and using the additivity of weak formulations, we obtain the following variational problem.

**Variational Problem 1 (A Newton-type step).**

Given  $\sigma_k \in L_+^\infty(\Omega)$ ,  $u_k \in \tilde{H}^1(\Omega)$ ,  $p > 0$ , solve for  $\delta \sigma_k \in L^\infty(\Omega)$ ,  $\delta u_k \in \tilde{H}^1(\Omega)$ ,

$$\begin{aligned} & \int_{\Omega} (\delta \sigma_k \nabla u_k + \sigma_k \nabla \delta u_k) \cdot \nabla v_1 \, dx + \int_{\Omega} |\nabla u_k|^p \left( \delta \sigma_k + p \sigma_k \frac{\nabla u_k \cdot \nabla \delta u_k}{|\nabla u_k|^2} \right) v_2 \, dx \\ & + \alpha \int_{\Omega} \delta u_k v_1 + \delta \sigma_k v_2 \, dx = \int_{\Omega} (H(\sigma) - H(\sigma_k)) v_2 \, dx \quad (3.68) \\ & \forall v_1 \in \tilde{H}^1(\Omega), v_2 \in L^\infty(\Omega), \end{aligned}$$

where  $0 < \alpha \ll 0$  is a regularization parameter, which in this case enforces a penalty on the  $L^2$ -norm of the trial functions, using test cases  $v_1 = \delta u_k$ ,  $v_2 = \delta \sigma_k$ .

Choosing an initial  $\sigma_0 \in L_+^\infty(\Omega)$ , one can obtain an initial  $u_0$  as the forward map  $U(\sigma_0)$ . From here the  $k$ 'th step in (3.68) can be performed successively until  $\|\sigma_{k+1} - \sigma_k\|_{L^\infty(\Omega)} < T$  for some predefined error tolerance  $0 < T \ll 1$ . This results in the Newton-type method given in Algorithm 1.

---

**Algorithm 1** Newton-type algorithm for one measurement  $M = 1$

---

**Require:** The domain  $\Omega$ , the Neumann data  $f \in \tilde{H}^{-1/2}(\partial\Omega)$  and the interior data  $H(\sigma)$ .

Define  $\sigma_0 \in L_+^\infty(\Omega)$ , a tolerance  $T > 0$  and maximum iteration number  $K$ .

Set  $k = 0$ .

**while**  $k < K$  and  $\varepsilon < T$  **do**

Solve for  $u_k$  the forward problem,

$$\int_{\Omega} \sigma_k \nabla u_k \cdot \nabla v \, dx = \int_{\partial\Omega} f v \, dS \quad \forall v \in \tilde{H}^1(\Omega).$$

Solve for  $\delta u_k \in \tilde{H}^1(\Omega)$ ,  $\delta\sigma_k \in L^\infty(\Omega)$  the variational problem,

$$\begin{aligned} & \int_{\Omega} (\delta\sigma_k \nabla u_k + \sigma_k \nabla \delta u_k) \cdot \nabla v_1 \, dx + \int_{\Omega} |\nabla u_k|^p \left( \delta\sigma_k + p\sigma_k \frac{\nabla u_k \cdot \nabla \delta u_k}{|\nabla u_k|^2} \right) v_2 \, dx \\ & + \alpha \int_{\Omega} \delta u_k v_1 + \delta\sigma_k v_2 \, dx = \int_{\Omega} (H(\sigma) - H(\sigma_k)) v_2 \, dx \quad \forall v_1 \in \tilde{H}^1(\Omega), v_2 \in L^\infty(\Omega). \end{aligned}$$

Compute  $H(\sigma_k)$ .

Set

$$\sigma_{k+1} = \sigma_k + \delta\sigma_k \quad \text{and compute} \quad \varepsilon = \|\delta\sigma_k\|_{L^2(\Omega)}.$$

Set  $k = k + 1$ .

**end while**

**return**  $\sigma_k$

---

### 3.2.3 A Newton-type method with multiple measurements

The Newton-type method of Algorithm 1 uses only one measurement given by  $H(\sigma)$ . Now we want to derive a method corresponding to Algorithm 1 which can handle multiple measurements of the interior data.

By a *measurement* we shall mean the pair of the  $m$ 'th Neumann data and the power density data,  $\{f^m, H^m(\sigma)\}$ , where  $H^m(\sigma)$  is the data resulting from the forward map (3.2) with  $f = f^m$ . The superscript notation is to not confuse with the iteration index  $k$ , used as a subscript for other variables. Suppose we have  $M$  such measurements. For

the given  $f^m \in \tilde{H}^{-1/2}(\partial\Omega)$  we get a forward solution  $u_k^m \in \tilde{H}^1(\Omega)$  at the step  $k$ . Given the forward solution and a reference conductivity  $\sigma_k \in L_+^\infty(\Omega)$ , the want to solve for  $(\delta\sigma_k, \delta u_k^m) \in L^\infty(\Omega) \times \tilde{H}^1(\Omega)$  for the  $m$ 'th measurement,

$$\nabla \cdot (\sigma_k \nabla \delta u_k^m) = -\nabla \cdot (\delta\sigma_k \nabla u_k^m) \quad \text{in } \Omega \quad (3.69)$$

$$\sigma_k \frac{\partial(\delta u_k^m)}{\partial n} = -\delta\sigma_k \frac{\partial(u_k^m)}{\partial n} \quad \text{on } \partial\Omega, \quad (3.70)$$

$$|\nabla u_k^m|^p \left( \delta\sigma_k + p\sigma_k \frac{\nabla u_k \cdot \nabla \delta u_k^m}{|\nabla u_k^m|^2} \right) = H^m(\sigma) - H^m(\sigma_k) \quad \text{in } \Omega. \quad (3.71)$$

Now for each of the  $M$  measurements we want to solve (3.69)-(3.71) which gives two equations with a Neumann boundary condition on  $\delta u_k^m$ . Hence in all we get  $2M$  equations. However for each measurement we need to solve for the same  $\delta\sigma_k$ , which means we have  $M+1$  variables, not  $2M$ . This can be written as a  $2M \times (M+1)$  matrix equation, which is only a square system for  $M=1$ , as was the case in Algorithm 1. In the case  $M > 1$  we will have an overdetermined system, which we can collect into

$$\begin{cases} \mathbf{A}\mathbf{x} = \mathbf{b} & \text{in } \Omega \\ \mathbf{D}\mathbf{x} = \mathbf{0} & \text{on } \partial\Omega. \end{cases} \quad (3.72)$$

$$\mathbf{A} = \begin{bmatrix} \nabla \cdot ([\cdot] \nabla u_k^m) & \nabla \cdot (\sigma_k [\cdot]) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \nabla \cdot ([\cdot] \nabla u_k^M) & 0 & \cdots & \nabla \cdot (\sigma_k [\cdot]) \\ |\nabla u_k^1|^p & p\sigma_k \frac{\nabla u_k^1 \cdot \nabla [\cdot]}{|\nabla u_k^1|^{2-p}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ |\nabla u_k^M|^p & 0 & \cdots & p\sigma_k \frac{\nabla u_k^M \cdot \nabla [\cdot]}{|\nabla u_k^M|^{2-p}} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ H^1(\sigma) - H^1(\sigma_k) \\ \vdots \\ H^M(\sigma) - H^M(\sigma_k) \end{bmatrix} \quad (3.73)$$

$$\mathbf{D} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ [\cdot] \frac{\partial u}{\partial n} & \sigma \frac{\partial [\cdot]}{\partial n} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ [\cdot] \frac{\partial u}{\partial n} & 0 & 0 & \sigma \frac{\partial [\cdot]}{\partial n} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \delta\sigma_k \\ \delta u_k^1 \\ \vdots \\ \delta u_k^M \end{bmatrix}, \quad (3.74)$$

where  $\mathbf{x} \in L^\infty(\Omega) \times [\tilde{H}^1(\Omega)]^M$ , denoting by  $V$  the space  $L^\infty(\Omega) \times [\tilde{H}^1(\Omega)]^M$ .  $\mathbf{A}$  is then a  $2M \times (M+1)$  matrix,  $\mathbf{x}$  is a  $(M+1) \times 1$  vector and  $\mathbf{b}$  is a  $2M \times 1$  vector. Since the system is overdetermined, we would not expect it to be satisfied exactly. Rather we will look for a minimizer,

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in V} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_{L^2(\Omega)} \quad (3.75)$$

This minimization problem can be solved by a Least-Squares Finite Element Method (see [7] for a reference). Since  $\mathbf{A}$  is a second order operator which would require a  $H^2$ -conforming Finite Element Method. This is generally not something one would prefer in the numerical setting. Hence, we transform the problem into a first order system,

$$\begin{cases} \hat{\mathbf{A}}\hat{\mathbf{x}} = \mathbf{b} & \text{in } \Omega \\ \mathbf{G}\mathbf{x} = \hat{\mathbf{x}} & \text{in } \Omega \end{cases} \quad (3.76)$$

where  $\hat{\mathbf{A}}$  and  $\mathbf{G}$  are defined by

$$\mathbf{A} = \hat{\mathbf{A}}\mathbf{G}, \quad \mathbf{G} = \text{diag}(1, \nabla, \dots, \nabla), \quad (3.77)$$

where  $\mathbf{G}$  essentially is just the operator that transforms the variables  $\delta u_k^m$  to their gradients  $\delta w_k^m := \nabla \delta u_k^m$ , such that  $\hat{\mathbf{A}}$  is a first order operator. As a technicality, one also needs to solve the forward problem as a first order problem. This gives the variational form of the forward problem for  $u_k^m \in \tilde{H}^1(\Omega)$ ,  $w_m^k \in H_{\text{div}}^1(\Omega)$ ,

$$\begin{aligned} \int_{\Omega} \sigma_k w_k^m \cdot \nabla v \, dx + \int_{\Omega} w_k^m \cdot \phi \, dx &= \int_{\partial\Omega} f v \, dS + \int_{\Omega} \nabla u_k^m \cdot \phi \, dx \\ \forall (v, \phi) &\in \tilde{H}^1(\Omega) \times H_{\text{div}}(\Omega). \end{aligned} \quad (3.78)$$

Here, we have defined the vector space,

$$H_{\text{div}}(\Omega) := \{w \in [L^2(\Omega)]^d \text{ s.t. } \nabla \cdot w \in L^2(\Omega)\} \quad (3.79)$$

Finally, we arrive at the least squares variational form of (3.76).

**Variational Problem 2 (Least-squares).** In correspondance to the first-order version of the system, we define the space,

$$V := H^1(\Omega) \times [\tilde{H}^1(\Omega)]^M \times H^1(\Omega) \times [H_{\text{div}}(\Omega)]^M. \quad (3.80)$$

Again  $\sigma_k \in L_+^\infty(\Omega)$ ,  $(\nabla u_k^1, \nabla u_k^2, \dots, \nabla u_k^M) \in [H_{\text{div}}^1]^M$ ,  $p > 0$  are given. We then want to find the norm-minimizing solution,

$$\arg \min_{(\mathbf{x}, \hat{\mathbf{x}}) \in V} \|\hat{\mathbf{A}}\hat{\mathbf{x}} - \mathbf{b}\|_{L^2(\Omega)} + \|\mathbf{G}\mathbf{x} - \hat{\mathbf{x}}\|_{L^2(\Omega)}. \quad (3.81)$$

This results in the least-squares variational form [LSFEM [7], Th. 3.1] for  $(\mathbf{x}, \hat{\mathbf{x}}) \in V$

$$\int_{\Omega} (\hat{\mathbf{A}}\hat{\mathbf{x}})^T \hat{\mathbf{A}}\hat{\mathbf{v}} + (\mathbf{G}\mathbf{x} - \hat{\mathbf{x}})^T (\mathbf{G}\mathbf{v} - \hat{\mathbf{v}}) + \alpha \mathbf{x}^T \mathbf{v} \, dx = \int_{\Omega} \mathbf{b}^T \hat{\mathbf{A}}\hat{\mathbf{v}} \, dx \quad \forall (\mathbf{v}, \hat{\mathbf{v}}) \in V, \quad (3.82)$$

where the term  $\alpha \int_{\Omega} \mathbf{x}^T \mathbf{v} \, dx$  imposes an  $L^2$ -penalty on  $\mathbf{x}$  due to the test case  $\mathbf{v} = \mathbf{x} = [\delta\sigma_k, \delta u_k^1, \dots, \delta u_k^M]^T$ .

---

**Algorithm 2** Least-squares Newton-type algorithm with  $M$  measurements

---

**Require:** The domain  $\Omega$ , the Neumann data

$$(f^1, \dots, f^M) \in [\tilde{H}^{-1/2}(\partial\Omega)]^M, \text{ and the interior data } H^1(\sigma), \dots, H^M(\sigma).$$

Define  $\sigma_0 \in L_+^\infty(\Omega)$ , a tolerance  $T > 0$  and maximum iteration number  $K$ .

Set  $k = 0$ .

**while**  $k < K$  and  $\varepsilon < T$  **do**

**for**  $m = 1 : M$  **do**

    Solve for  $w_k^m = \nabla u_k^m \in H_{\text{div}}(\Omega)$  the forward problem as a first order PDE,

$$\int_{\Omega} \sigma_k w_k^m \cdot \nabla v \, dx + \int_{\Omega} w_k^m \cdot \phi \, dx = \int_{\partial\Omega} f^m v \, dS + \int_{\Omega} \nabla u_k^m \cdot \phi \, dx$$

$$\forall (v, \phi) \in \tilde{H}^1(\Omega) \times H_{\text{div}}(\Omega).$$

    Compute  $H^m(\sigma_k)$ .

**end for**

Solve for  $(\mathbf{x}, \hat{\mathbf{x}}) \in V$ , the variational problem,

$$\int_{\Omega} (\hat{\mathbf{A}}\hat{\mathbf{x}})^T \hat{\mathbf{A}}\hat{\mathbf{v}} + (\mathbf{G}\mathbf{x} - \hat{\mathbf{x}})^T (\mathbf{G}\mathbf{v} - \hat{\mathbf{v}}) + \alpha \mathbf{x}^T \mathbf{v} \, dx = \int_{\Omega} \mathbf{b}^T \hat{\mathbf{A}}\hat{\mathbf{v}} \, dx \quad \forall (\mathbf{v}, \hat{\mathbf{v}}) \in V,$$

Set

$$\sigma_{k+1} = \sigma_k + \delta\sigma_k \quad \text{and compute} \quad \varepsilon = \|\delta\sigma_k\|_{L^2(\Omega)}.$$

Set  $k = k + 1$ .

**end while**

**return**  $\sigma_k$

---

Solving (3.82) we get the desired increment  $\delta\sigma_k$  for the  $k$ 'th step in a least-squares type Newton method, given in Algorithm 2.

The presented algorithms are both implemented using `Python` with the Finite-Element library software `FEniCS` [2]. The implementation details are described in the following chapter.

### 3.2.4 Ellipticity of the linearised problem

In this section we will briefly discuss ellipticity of the PDE system (3.72) defining the Newton step for with an arbitrary number of measurements  $M$ . This is more of a heuristic approach — the underlying theory of ellipticity for this type of problems is beyond the scope of this thesis. Hence, we will not go into detail with the proofs but rather state the results which will be useful for explaining the numerical results presented in Chapter 5.

Elliptic PDE's have the property that provided smooth coefficients and boundary conditions, the solution is also smooth [11], Th. 6.3. This is important, since this otherwise we would not expect a Continuous Galerkin-type Finite Element Method to be applicable. An important concept related to ellipticity is the principal symbol of the differential operator entering the PDE. The principal symbol is a scalar representation of a differential operator, which essentially replaces each partial derivative of order  $\alpha$  by a variable  $\xi^\alpha$ .

The principal symbol of the the Fréchet derivative  $dH$  as given in (3.53) for each measurement  $m$  is given for  $(x, \xi) \in \Omega \times \mathbb{R}^n$  by

$$P_m = |\nabla u_m|^p \left( 1 - p \frac{(\nabla u_m \cdot \xi)^2}{|\nabla u_m|^2 |\xi|^2} \right) \quad (3.83)$$

That this operator actually represents the scalar equations of the PDE system in (3.65)-(3.67) requires a derivation given in [19], Sec. 4.4.1.

The main result we want to refer here, is that the corresponding scalar operator to the operator  $\mathbf{A}$  in (3.73) is elliptic if and only if  $P_m = 0$  for some  $(x, \xi) \in \Omega \times \mathbb{R}^d$ . For a single measurement, assuming  $\nabla u_m \neq 0$ , loss of ellipticity occurs for pairs  $(x, \xi) \in \Omega \times \mathbb{R}^d$  where

$$\frac{(\nabla u_m \cdot \xi)^2}{|\nabla u_m|^2 |\xi|^2} = \frac{1}{p} \quad (3.84)$$

This is the same as

$$\cos^2(\langle \nabla u_m, \xi \rangle) = \frac{1}{p} \quad (3.85)$$

where  $\langle \cdot, \cdot \rangle$  denotes the angle between the vectors. For one measurement, this equation has no solutions for  $p < 1$  and the scalar operator is elliptic in that case (Th. 4.13 in [19]). For all other  $p \geq 1$ , at each  $x$  there is an angle for which (3.85) is fulfilled and hence the scalar operator is not elliptic for  $p \geq 1$ . For multiple measurements  $M$ , loss of ellipticity occurs when (3.84) is fulfilled at  $(x, \xi) \in \Omega \times \mathbb{R}^d$  for all  $m \in \{1, 2, \dots, M\}$ .

The last thing we will note regards the propagation of singularities. Theorem 4.17 in [19] states that a solution  $\delta\sigma$  to the scalar version of (3.72) has singularities propagating in the *perpendicular* direction to the direction of loss of ellipticity. That is, for example with  $p = 1$  if  $\nabla u$  is in the  $x$ -direction, then loss of ellipticity occurs when  $\xi$  is parallel to  $\nabla u$  by (3.85). This means that singularities will then propagate in the  $y$ -direction.



---

In the case of two measurements  $M = 2$  for  $p = 2$ , (3.85) is fulfilled for both gradient fields  $\nabla u_1$  and  $\nabla u_2$  provided they both make an angle of 45 degrees with some direction  $\xi$  at a point  $x \in \Omega$ . This implies that whenever the gradient fields become perpendicular to each other there will be two directions, corresponding to the two solutions of (3.85), where ellipticity is lost. These characteristic directions are observed in the numerical results in Chap. 5.



# Chapter 4

## Implementation

This chapter describes the numerical implementation of the methods from Chapter 3. So far, we have only considered continuous formulations of the power density inverse problem (1). Even though Algorithms 1 and 2 are designed to be implemented in a numerical setting, they consist of a sequence of weak formulations of boundary value imposed PDEs, which are all continuous problems. In practice, it is often necessary to solve these types of problems numerically, which the Finite Element Method (FEM) is suitable for.

In this chapter, we will describe the type of Finite Element Method which is used and the implementation of the method using the software package **FEniCS** [2]. Then we will present examples of the solutions obtained with the method and verify them with convergence tests on problems with known solutions.

### 4.1 The Finite Element setup

Rather than going through the whole Finite Element Method we will highlight some of the important properties and the setup used for our specific purpose. For a reference on the Finite Element Method see for example [14].

The first step in the FEM is to identify the weak formulation of problem we want to solve. In this thesis we have essentially four variational problems. To provide an overview, we will briefly summarize them here. With  $\sigma \in L^{\infty}_+(\Omega)$  and  $f \in \tilde{H}^{-1/2}(\partial\Omega)$  they are given as follows.

**The forward problem.** Solve for  $u_k \in \tilde{H}^1(\Omega)$

$$\int_{\Omega} \sigma \nabla u_k \cdot \nabla v \, dx = \int_{\partial\Omega} f v \, dx \quad \forall v \in \tilde{H}^1(\Omega) \quad (4.1)$$

**The first-order forward problem.** For each  $m$ , solve for  $(u_k^m, w_k^m)$  in  $\tilde{H}^1(\Omega) \times H_{\text{div}}^1(\Omega)$ ,

$$\int_{\Omega} \sigma_k w_k^m \cdot \nabla v \, dx + \int_{\Omega} w_k^m \cdot \phi \, dx = \int_{\partial\Omega} f v \, dS + \int_{\Omega} \nabla u_k^m \cdot \phi \, dx \quad (4.2)$$

$$\forall (v, \phi) \in \tilde{H}^1(\Omega) \times H_{\text{div}}(\Omega).$$

Let the variables be given as in Sec. 3.2. Then we have the Newton steps for the two presented algorithms:

**The Newton step for Algorithm 1.** Solve for  $(\delta\sigma, \delta u) \in L^\infty(\Omega) \times \tilde{H}^1(\Omega)$ ,

$$\int_{\Omega} (\delta\sigma_k \nabla u_k + \sigma_k \nabla \delta u_k) \cdot \nabla v_1 \, dx + \int_{\Omega} |\nabla u_k|^p \left( \delta\sigma_k + p\sigma_k \frac{\nabla u_k \cdot \nabla \delta u_k}{|\nabla u_k|^2} \right) v_2 \, dx$$

$$+ \alpha \int_{\Omega} \delta u_k v_1 + \delta\sigma_k v_2 \, dx = (H(\sigma) - H(\sigma_k)) v_2 \quad (4.3)$$

$$\forall v_1 \in \tilde{H}^1(\Omega), v_2 \in L^\infty(\Omega),$$

**The Newton step for Algorithm 2.** Solve for  $(\mathbf{x}, \hat{\mathbf{x}}) \in V$ ,

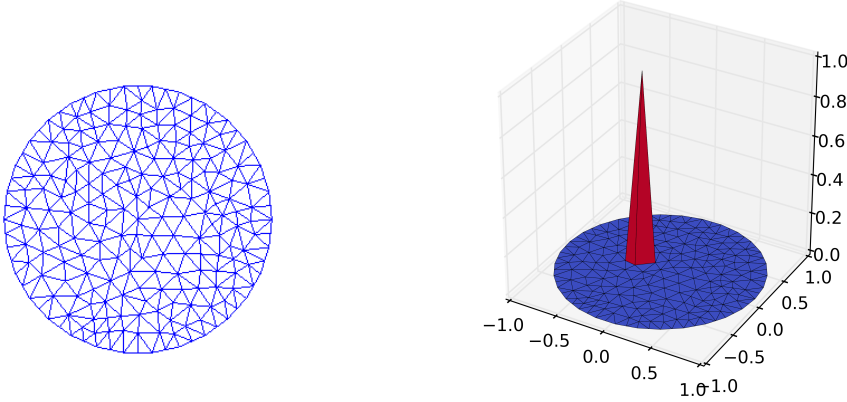
$$\int_{\Omega} (\hat{\mathbf{A}}\hat{\mathbf{x}})^T \hat{\mathbf{A}}\hat{\mathbf{v}} + (\mathbf{G}\mathbf{x} - \hat{\mathbf{x}})^T (\mathbf{G}\mathbf{v} - \hat{\mathbf{v}}) + \alpha \mathbf{x}^T \mathbf{v} \, dx = \int_{\Omega} \mathbf{b}^T \hat{\mathbf{A}}\hat{\mathbf{v}} \, dx \quad \forall (\mathbf{v}, \hat{\mathbf{v}}) \in V, \quad (4.4)$$

$$V = H^1(\Omega) \times [\tilde{H}^1(\Omega)]^M \times H^1(\Omega) \times [H_{\text{div}}(\Omega)]^M,$$

with  $\hat{\mathbf{A}}, \mathbf{G}$  defined as in Section 3.2.3.

Note that the forward problem (4.1) is the one used in Algorithm 1 whereas the first-order version (4.2) is employed in Algorithm 2 in order to avoid second order terms of  $u_k^m$  in the Newton-step (4.4). To implement these variational forms in a FEM-setting, we must choose a discretization of the domain  $\Omega$  (a mesh) and function spaces that can be represented on the discretization.

In all the numerical experiments in this thesis we choose  $\Omega$  to be the unit disk in  $\mathbb{R}^2$  as in Chapter 2 for the analytical example. The most common choice of representation of a domain in  $\mathbb{R}^2$  in the FEM, is one consisting of triangular elements. Such a triangulation



(a) A mesh with the number of nodes: 223 and the number of elements: 384.

(b) A Continuous Galerkin basis function on the unit disk.

**Figure 4.1:** (a) A coarse mesh on the unit disk (b) and a Continuous Galerkin basis function.

or *mesh* is shown in Fig. 4.1a. We then need to define a suitable function space on the mesh where we want to look for solutions to the discretized problem.

For the forward problems (4.1) and (4.2), we need to conform the spaces  $\tilde{H}^1(\Omega)$  and  $H_{\text{div}}(\Omega)$ . The former is simply  $H^1(\Omega)$  with an integral condition  $\int_{\partial\Omega} Tu \, dS = 0$ , which naturally calls for piece-wise linear function space. The latter is a vector space where each entry is weakly differentiable. Hence, each entry can be conformed by a piece-wise linear function.

For the Newton step for Algorithm 1 (4.3), we solve for functions in the spaces  $L^\infty(\Omega)$  and  $\tilde{H}^1(\Omega)$ . Generally, we would need to allow discontinuous function to approximate functions in  $L^\infty(\Omega)$ . However, in these numerical experiments we primarily investigate the reconstruction  $C^2$ -conductivities for which piece-wise linear functions are appropriate.

For the Newton step for Algorithm 2 (4.4), we encounter  $H^1(\Omega)$ ,  $\tilde{H}^1(\Omega)$  and  $H_{\text{div}}^1(\Omega)$ , which can all be approximated by piece-wise linears or piece-wise linears in each entry in the case of the vector function space  $H_{\text{div}}^1(\Omega)$ . The transformation into a first-order system (see Section 3.2.3) ensures that we avoid the Sobolev space  $H^2$  which is known to be impractical to implement — see Section 2.2.2 in [7].

To summarize, the chosen finite dimensional subspace for the discretization is the space of

piecewise linear functions on the mesh. These are defined by their value at each node and a linear interpolation between the three nodes on each triangular element. This is known as the space of Continuous Galerkin (CG1) functions. A basis for this space on a mesh with  $N$  nodes is the collection of continuous, piecewise linear functions  $\{v_i\}_{i=1}^N$  having the property,

$$v_i(x_j) = \delta_{ij}. \quad (4.5)$$

with  $x_i$  denoting the coordinate of the  $i$ 'th node and  $\delta_{ij}$  being the Kronecker delta. Since these are all  $H^1$ -functions on  $\Omega$ , they constitute a basis for a  $N$ -dimensional subspace of  $H^1(\Omega)$ ,

$$V_N := \text{span} \{v_i\}_{i=1}^N \subset H^1(\Omega). \quad (4.6)$$

A CG1 basis function can be seen in Fig. 4.1b. The interpolation  $u(x)$  of a continuous function onto the mesh, is then given as the convenient expansion,

$$u(x) = \sum_{i=1}^N u(x_i)v_i(x_i). \quad (4.7)$$

The three stated problems are all linear in the trial functions, and can generally be expressed as variational forms for each trial function

$$a(u, v) = l(v) \quad \forall v \in V. \quad (4.8)$$

where  $V$  is the corresponding trial function space,  $a$  is a bilinear form and  $l$  is a bounded linear functional. In the finite-dimensional setting we set up  $N_{\text{total}}$  equations by solving instead for  $u_i \in V_h$ ,

$$a(u_i, v_i) = l(v_i) \quad \forall v_i \in V_h, \quad (4.9)$$

where  $V_h$  represents the total finite-dimensional space. In Table 4.1,  $V_h$  is given for each variational problem. Now, each of the variational problems of the form (4.9) can be assembled to a system of  $N_{\text{total}}$  linear equations,

$$Ax = b, \quad (4.10)$$

which can be solved by an efficient matrix solver embedded in FEM software packages, such as FEniCS.

Note that the FEM spaces are only dependent on  $M$  for Algorithm 2, since Algorithm 1 is only designed for  $M = 1$ . However, it is quite clear that Algorithm 2 is more expensive. However, for the implementation we only use  $M = 1$  and  $M = 2$  for which Algorithm 2 is implementable on a workstation computer. The advantage of Algorithm 2 is that it transforms the 2nd order least squares problem in to a first order problem, which makes it more suitable for a CG1-setting. Note also, that the matrices  $\hat{\mathbf{A}}$  and  $\mathbf{G}$  are quite sparse, making Algorithm 2 more efficient than one would suppose from the dimensionality of  $V_h$ .

The forward problem (Alg. 1):	$V_h = V_N$
The forward problem, 1st order (Alg. 2):	$V_h = V_N \times [V_N]^2$ (solved $M$ times)
Newton step (Algorithm 1):	$V_h = V_N \times V_N$
Newton step (Algorithm 2):	$V_h = V_N \times (V_N)^M \times V_N \times (V_N)^{2M}$ .

**Table 4.1:** Finite-dimensional subspaces used for the two algorithms.  $V_N$  is the  $N$ -dimensional Continuous Galerkin (CG1) space on the mesh.

## 4.2 Validation of the forward problem implementation

For the implementation of the Finite Element Method we employ the software package `FEniCS` [2], which is a very intuitive Finite Element software package for both `Python` and `C++`. We will present a few pieces of examples of the `Python` code. The majority of the code for the implementation of both the forward and inverse problems presented can be seen in Appendix B.

In this section, we will validate the implementation of the forward problem 4.1 by comparing with the analytical solution from Section 2.2. Hence we consider on the unit disk in polar coordinates the forward problem with the piece-wise constant conductivity,

$$\sigma(r, \theta) = \begin{cases} 1 & \text{for } 0 \leq r \leq r_0 \\ 1 + C & \text{for } r_0 < r \leq 1 \end{cases} \quad (4.11)$$

and the Neumann data  $f \in \tilde{H}^{-1/2}(\partial\Omega)$  given by

$$f(\theta) = \cos(k\theta), \quad k \in \mathbb{N} \quad \text{on } \partial\Omega. \quad (4.12)$$

### 4.2.1 Code examples from Python

As discussed in Sec. 2.1, in order to enforce the solution to be in  $\tilde{H}^1(\Omega)$  we need to add two terms to the bilinear form. The resulting augmented variational problem is then to solve for  $(u, c) \in H^1(\Omega) \times \mathbb{R}$ ,

$$\int_{\Omega} \sigma \nabla u \cdot \nabla v \, dx + d \int_{\partial\Omega} u \, dS + c \int_{\partial\Omega} v \, dS = \int_{\partial\Omega} f v \, dS \quad \forall (v, d) \in H^1(\Omega) \times \mathbb{R}. \quad (4.13)$$

Using the main `Python`-library `dolfin` from the `FEniCS` package one can implement the problem. A mesh such as the one in Fig. 4.1a is defined. The space  $H^1(\Omega) \times \mathbb{R}$  is then

conformed by  $V_N \times \mathbb{R}$ . This function space is defined in the code as

---

```
V = Functionspace(mesh, "CG", 1)
R = FunctionSpace(mesh, "R", 0)
W = V*R
```

---

The bilinear form is very intuitively implemented as

---

```
a = sigma*inner(grad(u), grad(v))*dx

# Additional terms enforcing the integral condition int(u) ds = 0
constraints = (v*c + u*d)*ds

# Add the terms to the variational form
a = a + constraints
```

---

And the linear functional is equivalently defined.

---

```
L = f*v*ds
```

---

Then the `dolfin`-function `solve()` is used to compute the solution of the variational problem.

---

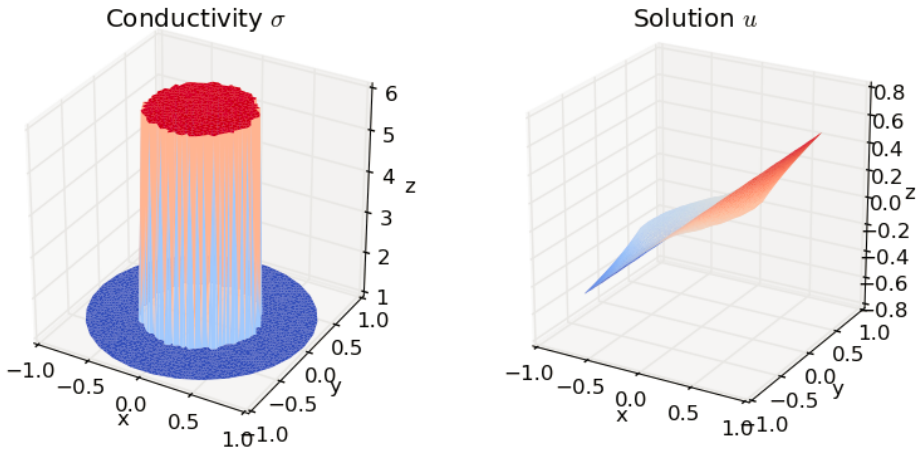
```
z = Function(W)
solve(a == L, z)
```

---

In Fig. 4.2 one can see the conductivity (left) with  $C = 10$ ,  $r_0 = \frac{1}{2}$  and the forward FEM solution, with a mesh size of  $N = 1809$ . This is the FEM solution corresponding to the analytical solution plotted in Fig. 2.1b, Sec. 2.2. By a visual comparison it seems that the FEM solution is an approximation of the true one. To verify the method explicitly, we need to evaluate the norm of the difference of the true and approximate solutions in a suitable function space as a function of the mesh-size  $N$  or alternatively as a function of the node-to-node distance  $h$ . Since this distance varies a bit on the mesh-types used for the unit disk, we prefer to use  $N$  as a measure of the mesh resolution. Note, that  $h$  is of length dimension and hence of order  $N^{-1/2}$  since the area of each element is of order  $N^{-1}$ . That is,

$$A_i = O(h^2) = O\left(\frac{1}{N}\right) \quad \text{and hence} \quad h = O\left(\frac{1}{\sqrt{N}}\right) \quad (4.14)$$





**Figure 4.2:** Left: The piece-wise constant conductivity  $\sigma$ . Right: The solution  $u$  with  $f(\theta) = \cos(\theta)$ . Both projected to piecewise linears on a mesh with 1809 vertices.

Choosing the  $L_2$ -norm of the relative error,

$$\varepsilon = \frac{\|u_{\text{true}} - u_{\text{FEM}}\|_{L^2(\Omega)}}{\|u_{\text{true}}\|_{L^2(\Omega)}}, \quad (4.15)$$

approximated in  $V_N$ , the order of convergence can be defined as a largest  $k > 0$  satisfying

$$\|\varepsilon\|_{L^2(\Omega)} \leq C \left(N^{-1/2}\right)^k \quad (4.16)$$

for some  $C > 0$  and for all  $N$  in a large enough range. This is equivalent to

$$\|\varepsilon\|_{L^2(\Omega)} \leq C_h \langle h \rangle^k, \quad (4.17)$$

with  $\langle h \rangle$  denoting the average node-to-node distance. The  $L^2$ -error is chosen throughout since it is a natural error in a numerical setting, being equivalent to the least-squared error for vectors in  $\mathbb{R}^N$ . In Fig. 4.3 the relative error  $\varepsilon$  can be seen as a function of  $N$  plotted against a slope of order  $O(N^{-1/2}) = O(h)$ . The plot indicates that with respect to the  $L^2$ -norm, the convergence rate is roughly of order  $N^{-1/2}$  which is equivalent to  $O(h)$ . In the sense of (4.16) this indicates first order convergence. Moreover, it validates the implementation of the forward problem in the sense that it does converge to the true solution as  $h \rightarrow 0$ .

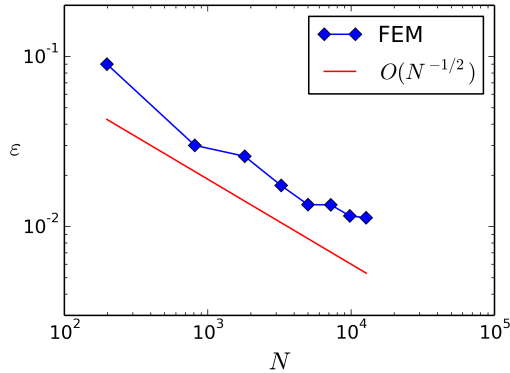


Figure 4.3: The relative  $L^2$ -error as a function of  $N$ .

### 4.3 Implementation details of the inverse problem

In this section we will go through some details in the setup of the implementation of Algorithms 1 and 2. These details are mostly related to the choices of test examples for numerical experiments and the choices of parameters in the two algorithms.

#### 4.3.1 Meshes and conductivity phantoms

As mentioned earlier, we will continue with the unit disk  $\mathbb{D} \subset \mathbb{R}^2$  as the domain for our numerical investigations. We will introduce two different mesh sizes to be used applied in the algorithms:

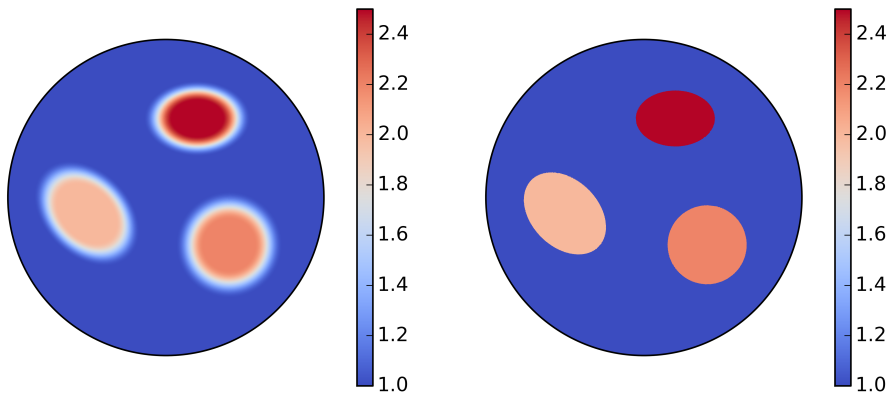
1. A fine mesh with  $N = 19899$  vertices for simulating interior data  $H(\sigma)$ .
2. A coarser mesh with  $N = 7211$  vertices for computing reconstructions with Algorithm 1 and 2.

We should note that it is important to simulate the interior data on another mesh than the one used for reconstruction. In practice, the interior data would be accessible on another mesh than the one used for FEM triangulation and should then be interpolated onto the FEM mesh. Using the same mesh for both data simulation and reconstruction falls into the category of an "inverse crime".

For simulating the data  $H(\sigma)$ , the forward problem is solved, given a chosen *phantom* or true conductivity  $\sigma$ . This can be done on a relatively large mesh, since it only requires solving the forward problem once. The coarser mesh is then used for the inverse problem algorithms.

The target conductivity phantoms are based on a piece-wise constant conductivity similar to the one used in the previous examples. However, applying CG1-based methods, we cannot be sure to obtain convergence of the algorithms without some added regularity. Hence, our starting point will be a smoothed version of the piece-wise constant conductivity  $\sigma_{\text{smooth}} \in C^2(\Omega)$  by smoothing in the region close to the discontinuity with a fifth order polynomial.

Most of the implementations will be tested on this conductivity phantom. In the last part of Chapter 5, we will however test the methods for the actual piece-wise constant phantom  $\sigma_{\text{PC}} \in L_+^\infty(\Omega)$ . The two conductivity phantoms are plotted in Fig. 4.4 as 2D-plots with a cool-to-warm color scale representing min-to-max function values in the range. These phantoms are inspired by those used in [18] and [25] with three elliptic inclusions.



(a) The conductivity phantom with fifth order polynomial smoothing. (b) The piece-wise constant conductivity phantom.

**Figure 4.4:** The conductivity phantoms used for testing.

### 4.3.2 Parameter choices

In the two implemented algorithms there are a range of parameters to be set. Two parameters have already been discussed — the mesh size and the chosen target conductivities.

For each algorithm we need to define an initial conductivity,  $\sigma_0$  which should be as general as possible. For the numerical tests, we choose the constant conductivity

$$\sigma_0(x) = 1 \quad \text{in } \Omega. \quad (4.18)$$

Note that the background value of the phantoms in Fig. 4.4 is also 1. Hence, the initial conductivity already has the correct value outside the inclusions and on the boundary. In the implementation of Algorithm 1 and 2, this means that one can add a Dirichlet condition on  $\delta\sigma$ , ensuring that perturbations only take place in the interior of  $\Omega$ . This technical step is done to add stability to the methods. The same assumption is used in [18] and [13]. It should be noted that this did not change the qualitative behaviour of the methods, although it slightly improved the error rates.

In the testing of the methods we additionally have the following parameters:

1. The power  $p > 0$  entering the data  $H(\sigma) = \sigma|\nabla u|^p$  which will take values in the set  $p \in \{0.5, 0.75, 1.0, 2.0\}$ .
2. The regularization parameter  $\alpha$  entering both algorithms, enforcing an  $L^2$ -penalty on the solution. The interval for testing is  $\alpha \in [0, 0.1]$ .
3. The number of interior data measurements  $M \in \{1, 2\}$ .
4. The frequency  $n$  entering the Neumann data  $f_1 = \cos(n\theta)$ ,  $f_2 = \cos(n\theta) + \sin(n\theta)$  with the interval  $n \in \{1, 2, \dots, 6\}$ .
5. The maximal iteration number  $K \in \{1, 20\}$ .

The numerical experiments can be divided into five smaller parts:

### Part 1: Reconstructions of the smooth conductivity

For a fixed  $\alpha = 10^{-4}$  and measurements  $f_1 = \cos(\theta) = x$ ,  $f_2 = \cos(\theta) + \sin(\theta) = x + y$ , we produce reconstructions of  $\sigma_{\text{smooth}}$  with  $M = 1$  and  $p \in \{0.5, 0.75, 1.0\}$  for Algorithm 1 and with  $M = 2$ ,  $p \in \{0.75, 1.0, 2.0\}$  for Algorithm 2. We use the iteration number  $k$  as a variable and examine the behaviour as function of  $k$  with  $p$  as a parameter.

### Part 2: Regularization effects

For a fixed  $p = 2.0$  and  $M = 2$  with  $f_1 = \cos(\theta) = x$ , and  $f_2 = \cos(\theta) + \sin(\theta) = x + y$ , we produce reconstructions of  $\sigma_{\text{smooth}}$  with Algorithm 2 for a varying regularization  $\alpha \in [0, 0.1]$  to investigate the effect of the regularization.

### Part 3: Neumann data dependence

For a fixed  $p = 2.0$ ,  $\alpha = 10^{-4}$  and  $M = 2$  we set  $f_1 = \cos(n\theta)$ , and  $f_2 = \cos(n\theta) + \sin(n\theta)$ , and vary  $n$  in the range  $n \in \{1, 2, \dots, 6\}$ . We then produce reconstructions of  $\sigma_{\text{smooth}}$  with Algorithm 2 for chosen values of  $n$  to evaluate how the quality of the reconstructions depends on the  $n$ .

### Part 4: Propagation of singularities

For a fixed  $\alpha = 10^{-4}$  and one measurement  $M = 1$  with  $f_1 = \cos(\theta) = x$ , we reconstruct  $\sigma_{\text{smooth}}$  for  $p = 1.1$ ,  $p = 1.3$ ,  $p = 1.5$ ,  $p = 2.0$  with Algorithm 2 using only one iteration. Since  $\nabla u$  is in the same direction as the Neumann data after the first iteration (the  $x$ -direction in this case), one can predict the angle of propagation of singularities as the orthogonal direction to the loss of ellipticity direction, as discussed in Sec. 3.2.4. The propagation direction is to be confirmed by the reconstructions.

Moreover, when using two measurements, the ellipticity is lost at each point where  $\nabla u_1$  and  $\nabla u_2$  are orthogonal, as we saw in Sec. 3.2.4. Using orthogonal Neumann boundary conditions  $f_1 = \cos(\theta) = x$ , and  $f_2 = \sin(\theta) = y$  will lead to orthogonal gradients after one iteration with Algorithm 2. One should consequently see singularities propagating in the angle of  $45^\circ$  to the  $x$ -axis.

### Part 5: Reconstruction of the piece-wise constant conductivity

As a final test of our methods, we will attempt to reconstruct the piece-wise constant conductivity shown in Fig. 4.4b in the case of power density interior data ( $p = 2$ ). This is done for  $\alpha \in \{0, 0.1, 1\}$  and  $M = 2$  using Algorithm 2. The case  $\alpha = 1$  is only an illustrative one, to indicate how a far too large regularization parameter affects the results. The Neumann data is again chosen to be  $f_1 = \cos(\theta) = x$ , and  $f_2 = \cos(\theta) + \sin(\theta) = x + y$ .



# Chapter 5

## Numerical results

In this chapter, we will present the most important results from the numerical experiments done in the project. All the results are obtained by implementing Algorithm 1 and 2 with the setup described in Chapter 4.

### 5.1 Reconstruction of a smooth target conductivity

We first present the reconstructions of the smooth conductivity phantom in Fig. 5.1

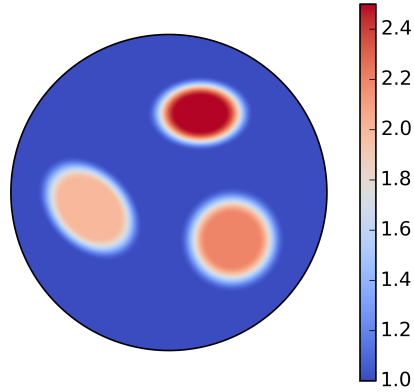
#### Algorithm 1

In Fig. 5.2 the reconstructions obtained with Algorithm 1 after 10 Newton iterations are plotted for  $p = 0.5$ ,  $p = 0.75$  and  $p = 1.0$ . Here, the regularization parameter  $\alpha = 10^{-4}$  and one measurement is used with  $f_1 = x$  as the Neumann condition on the boundary. Each reconstruction is terminated after  $K = 10$  iterations where the criterion

$$\|\sigma_{k+1} - \sigma_k\|_{L^2(\Omega)} = \|\delta\sigma\|_{L^2(\Omega)} \leq 10^{-6} \quad (5.1)$$

is satisfied. Note, that the color scale is from smallest value (blue) to largest value (red) and hence the colors do not directly represent the same values as in the color scale for the phantom in 5.1. Rather, the color scale indicates whether the reconstruction adequately captures the features of the true phantom.

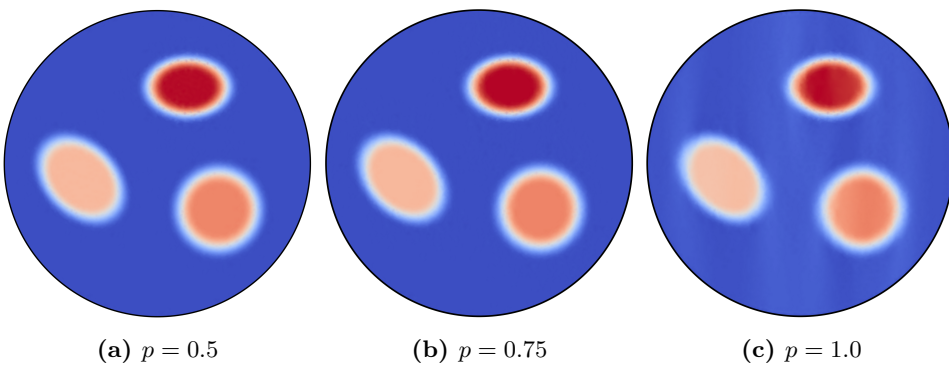
We observe that for  $p = 0.5$  and  $p = 0.75$  the reconstruction are as good as indistinguishable from the phantom. For  $p = 1.0$  we observe that the reconstruction is slightly blurred along



**Figure 5.1:** The conductivity phantom with fifth order polynomial smoothing.

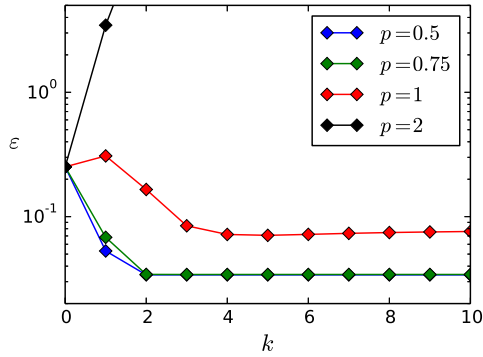
the  $y$ -direction. For  $p = 1.0$  the loss of ellipticity from (3.84) is in the  $x$ -direction, at least after one iteration, where  $\nabla u$  is parallel to the  $x$ -axis. This is due to the Neumann condition  $f_1 = x$  for  $x \in \partial\Omega$  and the initial conductivity  $\sigma_0$  being constant. As we saw in Section 3.2.4, singularities propagate in the orthogonal direction to the loss of ellipticity direction, which in this case is the  $y$ -direction.

In Fig. 5.3 we examine the convergence of Algorithm 1 for the same setting as before, with different parameter values of  $p$ . Here we plot the relative  $L^2$ -error of the difference



**Figure 5.2:** Reconstructions with Algorithm 1 where  $M = 1$ ,  $\alpha = 10^{-4}$ .





**Figure 5.3:** The  $L_2$ -error as a function of the iteration number  $k$  for Algorithm 1 with  $M = 1$  measurements

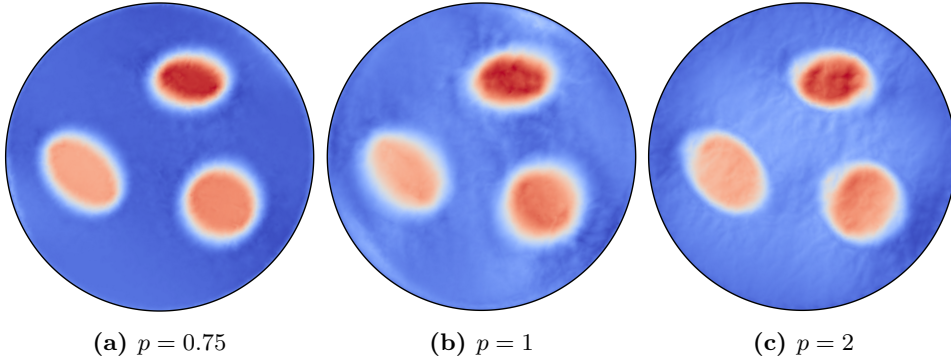
between the true conductivity and the reconstructed one,

$$\varepsilon = \frac{\|\sigma_k - \sigma_{\text{true}}\|_{L^2(\Omega)}}{\|\sigma_{\text{true}}\|_{L^2(\Omega)}}. \quad (5.2)$$

Fig. 5.3 shows that Algorithm 1 converges to a stable solution for all three values of  $p$ . Moreover, the loss of ellipticity for  $p = 1.0$  affects the errors, as the relative  $L^2$ -error converges to a larger value for  $p = 1.0$ . This is due to the error induced by the propagation of singularities parallel to the  $y$ -axis. The smallest relative error is approximately 0.43 for both  $p = 0.5$  and  $p = 0.75$ . For  $p = 1$  it is 0.72. One should note that the reconstruction problem for  $p = 0.5$  is not very challenging since the data  $H(\sigma) = \sigma|\nabla u|^{1/2}$  already resembles the conductivity  $\sigma$ , due to the fact that the power  $p = \frac{1}{2}$ , damps relative changes in  $|\nabla u|$ .

### Algorithm 2

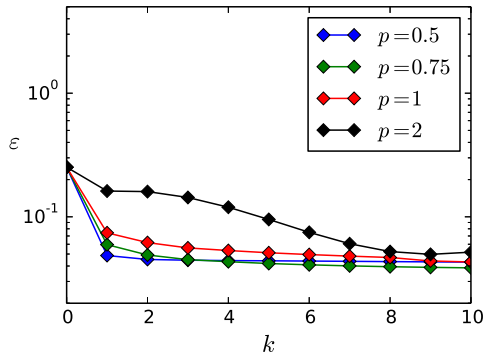
Having now obtained good reconstructions for  $p \leq 1$ , we will turn to the case of two measurements  $M = 2$  to provide the needed stability to obtain reconstructions for  $p = 2$  (power density case) where the PDE system (3.72) becomes non-elliptic. For  $p = 2$  and  $M = 2$  loss of ellipticity occurs at points  $x \in \Omega$  for which  $\nabla u_1$  and  $\nabla u_2$  are orthogonal. The Neumann data is chosen to be  $f_1 = \cos(\theta) = x$  and  $f_2 = \cos(\theta) + \sin(\theta) = x + y$ . Seeing that orthogonal Neumann-directions would lead to orthogonal gradients after the first iteration (loss of ellipticity), while too little angular separation between the directions of  $f_1$  and  $f_2$  would imply less added information, a separation of 45 degrees appears to be a reasonable choice. In Fig. 5.4 we view the reconstructions obtained by Algorithm



**Figure 5.4:** Reconstructions with Algorithm 2 where  $M = 2$ ,  $\alpha = 10^{-4}$ .

2 with two measurements. Again the method is stopped after  $K = 10$  iterations. The reconstructions are fairly close to the true phantom, but for  $p = 1$  and  $p = 2.0$  there are some noisy regions. For  $p = 2$  this noise will induce more errors in the next iterations and eventually lead to semi-convergence: After some number  $k$  iterations, the error starts to increase. For this reason, it is difficult to use a convergence criterion for  $p = 2$  as the one in (5.1).

In Fig. 5.5 we examine the behaviour of the errors for different  $p$ . We observe that for  $p \leq 1$  the method converges to a stable solution. For  $p = 2.0$  the error behaves quite differently but comes quite close to the true solution after 9 and 10 iterations. It is however only semi-convergent, as we will see in the next section. The smallest errors obtained are approximately 0.04 for  $p \leq 1$  and 0.064 for  $p = 2$ .



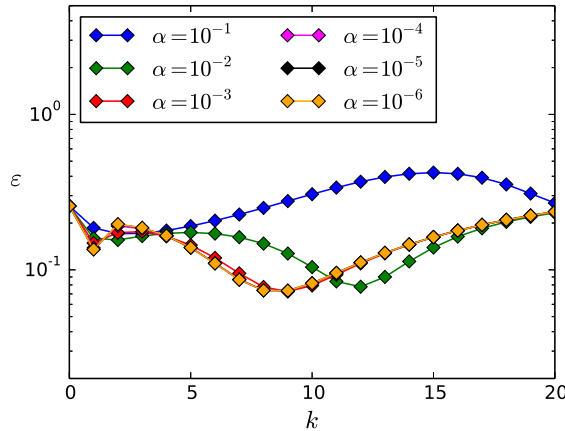
**Figure 5.5:** The  $L_2$ -error as a function of the iteration number  $k$  for Algorithm 2 with  $M = 2$  measurements.

## 5.2 Regularization effects

In this section we investigate solely the power density case  $p = 2$  with Algorithm 2 and specifically how the regularization affects the performance of the method.

The regularization parameter  $\alpha$  enforces an  $L^2$ -regularization on the norm of  $\delta\sigma$  and  $\delta u$ . Having a too large regularization would lead to reconstructions that do not fit the data well enough, since the solved problem in that case differs too much from the true one. A too small regularization would presumably lead to amplification of errors resulting from the Finite Element approximation.

In Fig. 5.6 the relative  $L^2$ -error is plotted as a function of the iteration number, for different values of  $\alpha$  and  $p = 2.0$  and two measurements  $M = 2$ . Firstly, we observe that the method is only semi-convergent for all the chosen values of  $\alpha > 0$ . For all  $\alpha \leq 10^{-3}$  the method obtains a minimal relative error after 9 iterations of  $\varepsilon_{\min} \approx 0.072$ . With  $\alpha = 10^{-2}$  a minimum is obtained after 12 iterations with the value  $\varepsilon_{\min} \approx 0.078$ . For  $\alpha = 10^{-1}$  the method is never really close to the solution and obtains a minimum after 1 iteration of  $\varepsilon_{\min} \approx 0.18$ . For each  $\alpha \leq 10^{-3}$  the behaviour is almost identical — it is



**Figure 5.6:** The  $L_2$ -error as a function of the iteration number  $k$  with varying  $\alpha$  using Algorithm 2 with  $M = 2$  and  $p = 2$ .

only just possible to separate the  $\alpha = 10^{-3}$  curve from the last three, which take almost identical values. This suggests that as long as the regularization is sufficiently small, one will obtain the same results. Hence, one would assume that the optimal value is in fact  $\alpha = 0$  implying that there is no actual need for regularization. The explanation for this might be that the LSFEM variational form already regularizes the solution by using two

measurements and finding an  $L^2$ -norm minimizing solution. As noted by Hoffmann [19], using a regularization parameter  $\alpha \in [10^{-12}, 10^{-3}]$  for this algorithm does not change the *qualitative* properties of the results. This can be confirmed here in a more quantitative way as Fig. 5.6 shows that there is negligible difference in the errors for  $\alpha \leq 10^{-3}$ . Moreover, the tests showed the same results for  $\alpha = 0$ . This does however not rule out regularization in general but merely indicates that  $L^2$ -regularization is not suitable for this problem. One should note that noise is not included and regularization could become important in that case. Other types of regularization (e.g. Total Variation) could be suitable for the problem even without added noise.

The indication of  $L^2$ -regularization being unnecessary is consistent with the conclusions of Capdeboscq [9] where a similar problem is solved with three measurements and the computational results indicate that the problem is not severely ill-posed and that Tikhonov regularization is not needed.

### 5.3 Effects of the choice of boundary conditions

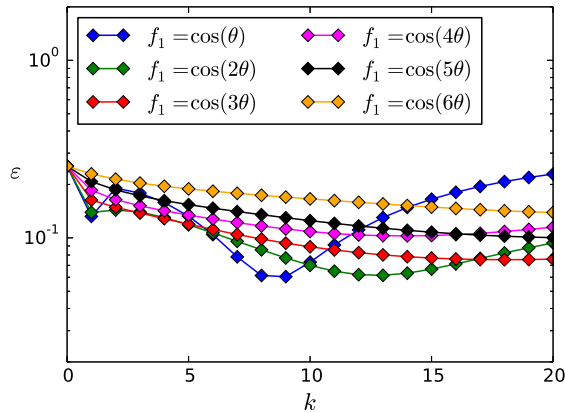
We now investigate the effects of changing the frequency  $n$  entering the Neumann boundary conditions for the forward problem with

$$f_1(\theta) = \cos(n\theta) \tag{5.3}$$

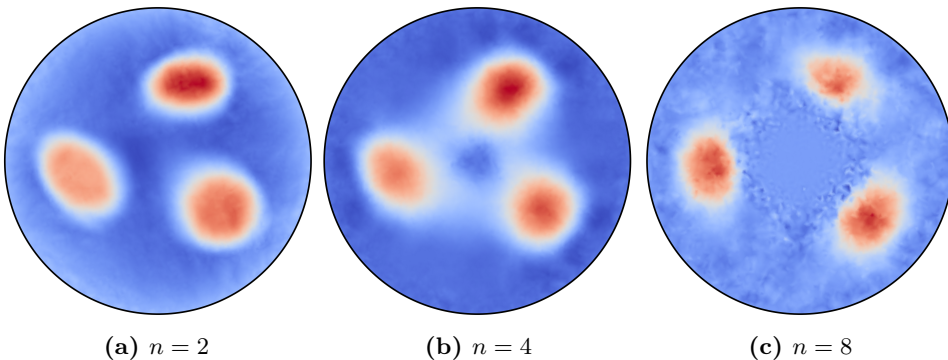
$$f_2(\theta) = \cos(n\theta) + \sin(n\theta). \tag{5.4}$$

In Fig. 5.7 we see the relative  $L^2$ -error of the reconstruction as a function of the iteration number  $k$  and with the frequency  $n$  as a parameter. Note that for  $n = 1$ ,  $f_1 = \cos(\theta)$  corresponds to  $f_1 = x$  and is the same boundary condition as used for the previous reconstructions. We find again that the smallest error is obtained at  $k = 9$  and that the method is only semi-convergent. In the case of  $n > 1$  the methods appear to be more stable but further away from the true target conductivity.

It seems plausible that the higher oscillations in the Neumann data cause loss of information around the inclusions, since the forward solution has to capture more information (in terms of a more rapidly changing function as opposed to a constant one) from the boundary condition. This eventually leads to reconstructions which cannot describe the behaviour around the inclusion boundaries. In Fig. 5.8 we show the reconstructions for  $n = 2$ ,  $n = 4$  and  $n = 8$  after 3 iterations which is sufficient to illustrate the effect. Clearly, the more rapid oscillations with  $n = 4$  and  $n = 8$  induce some errors in the region close to the center of disk. Especially for  $n = 8$  the reconstruction appears quite pixelated, even though it is computed on the same mesh as the two others. The errors could however also be a result of the fact that a more rapidly oscillating function is not approximated well by piece-wise linear functions, unless the mesh is very fine.



**Figure 5.7:** The  $L_2$ -error as a function of the iteration number  $k$  with  $f_1 = \cos(n\theta)$ ,  $f_2 = \cos(n\theta) + \sin(n\theta)$  for different  $n$  using Algorithm 2.



**Figure 5.8:** Reconstructions with different values of  $n$  using Algorithm 2 where  $M = 2$ ,  $\alpha = 10^{-4}$ .

## 5.4 Propagation of singularities

We here investigate the result from Hoffmann [19], discussed in Section 3.2.4, concerning propagation of singularities. The result states that there are certain directions where singularities propagate and these are the perpendicular directions to the loss of ellipticity directions. In Sec. 3.2.4 we saw that the pairs of points and directions where ellipticity is lost is given by  $(x, \xi) \in \Omega \times \mathbb{R}^2$  where

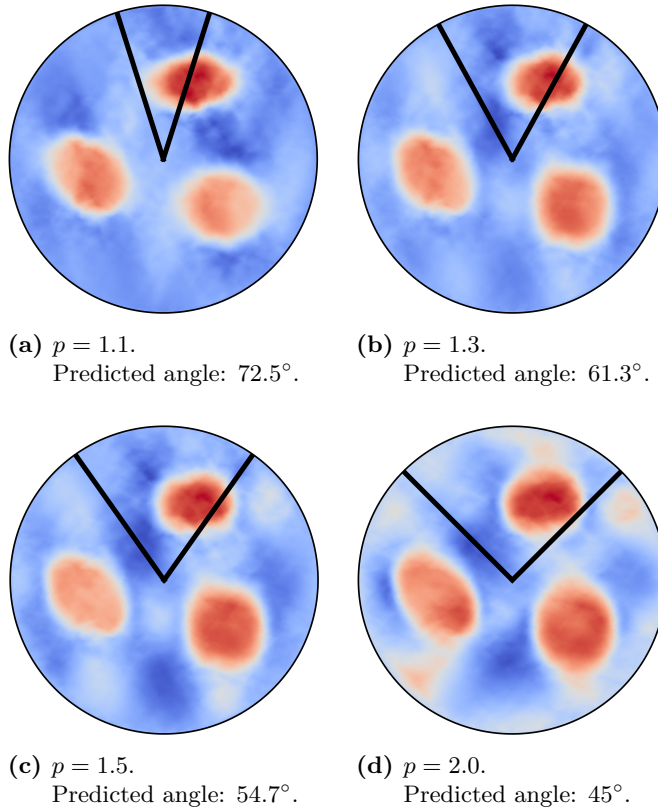
$$\cos^2(\langle \nabla u, \xi \rangle) = \frac{1}{p}. \quad (5.5)$$

In the first Newton iteration the background conductivity is  $\sigma_0 = 1$  for all  $x \in \mathbb{D}$ . With one Neumann boundary condition  $f_1 = x$ , the gradient of the forward solution is then everywhere parallel to the  $x$ -axis. One can then calculate the angle of loss of ellipticity specific values of  $p$  as  $\cos^2(p^{-1})$ . The direction for propagation of singularities is the perpendicular direction, which gives the directions

$$p = 1.1 : \quad \theta = 72.45^\circ \qquad p = 1.3 : \quad \theta = 61.29^\circ \qquad (5.6)$$

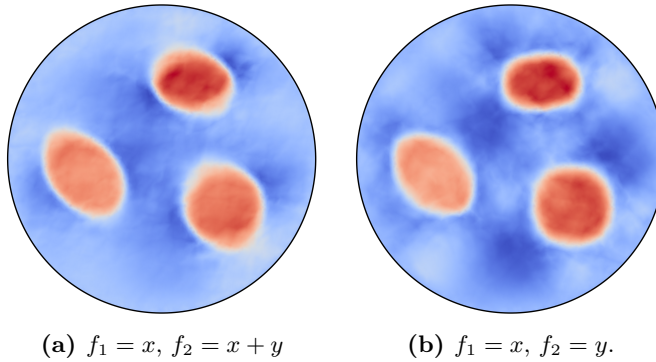
$$p = 1.5 : \quad \theta = 54.73^\circ \qquad p = 2.0 : \quad \theta = 45.00^\circ \qquad (5.7)$$

In Fig. 5.9 the reconstructions after one iteration are plotted using Algorithm 2 and  $f_1 = x$



**Figure 5.9:** Reconstructions with Algorithm 2 after 1 iteration with one measurement  $f = x = \cos(\theta)$ .

as the only Neumann condition. The predicted angles of the propagation of singularities are shown with a black guideline. The characteristic directions are very visible in the reconstructions and the direction is very close to the predicted one. Of course, this is only



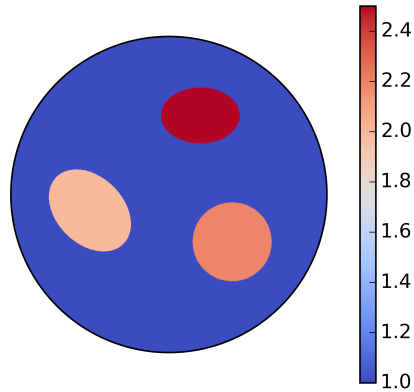
**Figure 5.10:** Reconstructions with Algorithm 2 after 1 iteration with  $p = 2$ .

after one iteration where we know that  $\nabla u$  will be in the  $x$ -direction. In the next iteration it is already hard to say where the loss of ellipticity will occur.

Using two Neumann boundary conditions, there might still be points where ellipticity is lost, i.e. where the condition (3.85) is fulfilled for both gradient fields  $\nabla u_1$  and  $\nabla u_2$  simultaneously. For the case  $p = 2$ , this can happen whenever the gradient fields are orthogonal — i.e. there exists a direction  $\xi$  which makes an angle of  $45^\circ$  with both fields. Choosing orthogonally oriented Neumann boundary conditions  $f_1 = x$  and  $f_2 = y$  ensures that this will happen after one iteration. Conversely, if one chooses  $f_1 = x$  and  $f_2 = x + y$ , the gradient fields will not be orthogonal at any point after one iteration. In Fig. 5.10 we observe this phenomenon where the reconstruction for orthogonally oriented boundary conditions in Fig. 5.10b has a distinct characteristic direction of  $45^\circ$  to the  $x$ -axis corresponding to the propagation of singularities along this direction, while there is no such direction in the case of  $f_2 = x + y$  in Fig. 5.10a. Consequently, the choice of boundary conditions separated by  $45^\circ$  degrees is preferable in terms of avoiding propagations of singularities.

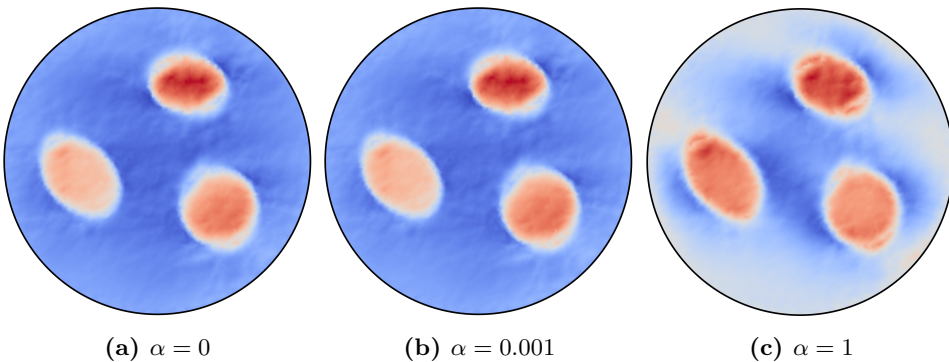
## 5.5 Reconstruction of a piece-wise constant conductivity

Finally, we evaluate for  $p = 2$  (the power density problem) using Algorithm 2, the reconstructions of a piece-wise constant conductivity. The phantom is shown in Fig. 5.11. Fig. 5.12 shows reconstructions for three different values of the  $L^2$ -regularization parameter  $\alpha$  to also illustrate that even for piece-wise constant conductivities, the regularization parameter should be as small as possible in accordance with what we found in Sec. 5.2.



**Figure 5.11:** The piece-wise constant conductivity phantom.

We observe that the reconstructions for  $\alpha = 0$  and  $\alpha = 0.001$  are roughly of the same quality as the one obtained for the smooth conductivity from Fig. 5.4c (with  $\alpha = 10^{-4}$ ). For  $\alpha = 1$ , the reconstruction becomes more erroneous, but is in fact closer to the true phantom than one would expect from the level of regularization. Clearly, the reconstructions for the piece-wise constant conductivity do not represent the discontinuities in  $\sigma$  perfectly. The boundaries of the inclusions are somewhat blurred in each of the examples in Fig 5.12. Qualitatively, the reconstructions are still very resemblant of the true piece-wise constant conductivity of Fig. 5.11. The algorithm was again found semi-convergent and the errors evolve similarly to those of Fig. 5.6 with a smallest relative  $L^2$ -error of approximately 0.10.



**Figure 5.12:** Reconstructions of the piece-wise constant conductivity after 5 iterations, for  $p = 2$ , using Algorithm 2 where  $M = 2$  and different values of  $\alpha$ .



# Chapter 6

## Conclusion and perspectives

### 6.1 Conclusion

In this thesis, we have presented methods for solving a class of hybrid inverse problems with interior data of the current density type (CDII) and the power density type, which arises in for example acousto-electrical tomography (AET) and impedance acoustic tomography (IAT).

The thesis presents two algorithms which are designed to solve inverse problem using a single measurement (Algorithm 1) and several measurements (Algorithm 2) respectively. The algorithms are based on a linearisation of the forward map and a coupled PDE for determining both the Fréchet derivative  $\delta u$  and the increment direction  $\delta\sigma$  simultaneously. These algorithms appear in [18] using a Dirichlet boundary condition on the electrical potential, while in this thesis, they are applied to the Neumann boundary condition.

The algorithms are applied to reconstruct both  $C^2$ -conductivities and piece-wise constant conductivities on the unit disk, both with three elliptic inclusions to be reconstructed. In [18] the same type of algorithms are applied to the  $C^2$ -case while the experiments on piece-wise constant conductivities are an addition presented in this thesis. Both algorithms are implemented with the Finite Element software **FEniCS** [2].

Algorithm 1 successfully solves the problem with 1 measurement for  $p \leq 1$ , with errors being smaller than 0.07 in the relative  $L^2$ -norm. Algorithm 2 solves the problem with 2 measurements for the significantly more challenging problem of  $p = 2$  with a relative  $L^2$ -error of approximately 0.072. With the chosen mesh-size for the reconstructions (7211

vertices) the methods are only semi-convergent in the numerical sense — they come close to the target conductivity around 10 Newton-type iterations but after sufficiently many iterations they eventually diverge, assumably due to the accumulation of errors in the FEM–approximation in each step.

Additional investigations have been carried out solely for the power density case ( $p = 2$ ), where Algorithm 2 (as opposed to Algorithm 1) provides good reconstructions by employing two boundary measurements.

For the reconstruction from the power density data, the effects of the  $L^2$ –regularization parameter  $\alpha$  have been examined. The results clearly indicate that there is no effective gain in adding  $L^2$ –regularizing terms. Small values  $\alpha < 10^{-4}$  all produce approximately the same error at each iteration while larger values of  $\alpha$  lead to increased errors. In Hoffmann [18] it is mentioned that choosing  $\alpha \in [10^{-12}, 10^{-3}]$  does not change the qualitative properties of the methods used. This is consistent with what is found here, with the exception that we conclude that there regularization term could be set to exactly zero or any value  $< 10^{-12}$ . Moreover, it is consistent with Capdeboscq [9], where it is noted that Tikhonov-regularization is not necessary for a similar power density problem.

We also investigate the effects of Neumann boundary conditions of the type  $f_1 = \cos(n\theta)$ ,  $f_2 = \cos(n\theta) + \sin(n\theta)$  for different values of  $n$ . The numerical results indicate that a larger frequency  $n$  leads to poorer reconstructions with the mesh-size used for reconstructions (7211 vertices).

A result concerning the propagation directions of singularities in non-elliptic cases, which is presented in [19], is confirmed here for the Neumann case and slightly different conductivity phantoms. In the case of a single measurement, the reconstructions after 1 iteration show quite clearly the characteristic directions which are consistent with the theoretically predicted ones.

Additionally, in the case of two measurements with perpendicular gradients after 1 iteration, a characteristic propagation direction of 45 degrees with the  $x$ -axis is found. This is also consistent with the theory presented in [19], whereas the numerical evidence for this is not presented there.

Finally, reconstructions are presented for the case of piece-wise constant conductivities with two measurements and power density interior data. The reconstructions are still reasonable even though piece-wise linear basis functions are strictly not suitable for the recovery of a piece-wise constant target conductivity. The relative  $L^2$ –error takes a minimal value of approximately 0.10.

## 6.2 Perspectives

Here we present a discussion on how the methods used in this project could be modified and which other approaches could be taken.

The first thing to note is that the model used here does not include noise in the data. This might be done as in [9] where up to 10% Gaussian noise is added. Including noise would allow us to numerically evaluate the stability of the power density problem, especially in the case of two measurements.

This immediately leads us to the discussion of regularization, since it becomes more relevant when more noise is added. It would be natural to evaluate whether  $L^2$ -regularization would be necessary in the case of noisy data. Another interesting subject would be to apply other types of regularization. An approach could be to build a method applying Total Variation regularization, which penalizes the  $L^1$ -norm of  $\nabla\delta\sigma$  and is generally preferable in order to preserve discontinuous change in the conductivity [28].

A third consideration is that the methods presented here could be tested on other types of conductivity phantoms. The examples used in the numerical experiments are quite artificial in the sense that they do not appropriately represent the physical reality that they attempt to model. The simulated conductivities could indeed be constructed such that they represent the complexity of a the conductivity inside e.g. the human body — an example would be the conductivity phantoms used by Bal *et. al.* [5] for the power density problem. In addition to this, one should also note that 2D reconstructions are of course not realistic representations of the intrinsically three-dimensional phenomenon electrical current inside a physical object.

Finally, future investigations could deal with the implementation of similar algorithms applied to inverse problems with different types of interior data. The original idea of this project was to consider convoluted data types where only the convolution of the power density data would be accessible. In the case of Gaussian convolutions this would essentially be adding a deblurring problem to the one presented here. A method could then consist of the solving a deblurring inverse problem to obtain the interior data  $H(\sigma)$  and hereafter apply the reconstruction algorithms for the interior data. Alternatively one could perform the linearisation directly on the convoluted data and apply a Newton-type algorithm via this linearisation.



# References

- [1] Giovanni Alessandrini. Stable determination of conductivity by boundary measurements. *Applicable Analysis*, 27(1-3):153–172, 1988.
- [2] Martin S. Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E. Rognes, and Garth N. Wells. The fenics project version 1.5. *Archive of Numerical Software*, 3(100), 2015.
- [3] Habib Ammari. *An Introduction to Mathematics of Emerging Biomedical Imaging*, volume 62 of *Mathématiques et Applications*. Springer, Berlin, 2008.
- [4] Kendall Atkinson and Weimin Han. *Theoretical Numerical Analysis: A Functional Analysis Framework*. Texts in Applied Mathematics. Springer New York, 2009.
- [5] Guillaume Bal, Wolf Naetar, Otmar Scherzer, and John Schotland. The levenberg-marquardt iteration for numerical inversion of the power density operator. *Journal of Inverse and Ill-Posed Problems*, 21(2):265–280, 4 2013.
- [6] Robert G. Bartle. Newton’s method in banach spaces. *Proceedings of the American Mathematical Society*, 6(5):827–831, 1955.
- [7] Pavel B. Bochev and Max D. Gunzburger. *Least-Squares Finite Element Methods*, volume 166 of *Applied Mathematical Sciences*. Springer-Verlag New York, 175 Fifth Avenue, New York, USA, 2009.
- [8] Alberto P. Calderón. On an inverse boundary value problem. *Seminar on Numerical Analysis and its Applications to Continuum Physics*, pages 65–73, 1980.
- [9] Yves Capdeboscq, Jérôme Fehrenbach, Frédéric De Gournay, and Otared Kavian. Imaging by modification: Numerical reconstruction of local conductivities from corresponding power density measurements. *SIAM Journal on Imaging Sciences*, 2(4):1003–1030, 2009.
- [10] Margaret Cheney, David Isaacson, and Jonathan C. Newell. Electrical impedance tomography. *SIAM Review*, 41(1):85–101, 1999.

- 
- [11] Lawrence C. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, Rhode Island, 1998.
- [12] Sami Gabriel, R. W. Lau, and C. Gabriel. The dielectric properties of biological tissues: II. measurements in the frequency range 10 Hz to 20 GHz. *Physics in Medicine and Biology*, 41(11):2251, 1996.
- [13] Henrik Garde. Sparsity Regularization for Electrical Impedance Tomography. Master's thesis, Technical University of Denmark, Kgs. Lyngby, Denmark, 2013.
- [14] Mark S. Gockenbach. *Understanding and Implementing the Finite Element Method*. SIAM, 2006.
- [15] David J. Griffiths. *Introduction to Electrodynamics*. Prentice Hall, 1999.
- [16] Jacques Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton University Bulletin*, 13:49–52, 1902.
- [17] Sina Heinrich, Holger Schiffmann, Alexander Frerichs, Adelbert Klockgether-Radke, and Inéz Frerichs. Body and head position effects on regional lung ventilation in infants: an electrical impedance tomography study. *Intensive Care Medicine*, 32(9):1392–1398, 2006.
- [18] Kristoffer Hoffmann and Kim Knudsen. Iterative reconstruction methods for hybrid inverse problems in impedance tomography. *Sensing and Imaging*, 15, 2014.
- [19] Kristoffer Hoffmann and Kim Knudsen. *Reconstruction Methods for Inverse Problems with Partial Data*. PhD thesis, 2015.
- [20] Bangti Jin and Peter Maass. An analysis of electrical impedance tomography with applications to Tikhonov regularization. *ESAIM: Control, Optimisation and Calculus of Variations*, 18:1027–1048, 10 2012.
- [21] Per W Karlsson and Vagn Lundsgaard Hansen. *Matematisk Analyse 2: Funktioner af flere reelle variable*. Institut for Matematik, DTU, 1998.
- [22] Peter D. Lax and Arthur N. Milgram. Parabolic equations. *Annals of Mathematics Studies*, 33:167–190, 1954.
- [23] Michael J. Cloud Leonid P. Lebedev, Iosif I. Vorovich. *Functional Analysis in Mechanics*. Springer Monographs in Mathematics. Springer New York, 2012.
- [24] Norman G. Meyers. An  $L^p$ -estimate for the gradient of solutions of second order elliptic divergence equations. *Annali della Scuola Normale Superiore di Pisa - Classe di Scienze*, 17(3):189–206, 1963.
- [25] Jennifer L. Mueller and Samuli Siltanen. *Linear and Nonlinear Inverse Problems with Practical Applications*. Computational Science and Engineering. SIAM, 2012.

- 
- [26] John Sylvester and Gunther Uhlmann. A global uniqueness theorem for an inverse boundary value problem. *Annals of Mathematics*, 125(1):153–169, 1987.
- [27] Thomas Widlak and Otmar Scherzer. Hybrid tomography for conductivity imaging. *Inverse Problems*, 28(8):084008, 2012.
- [28] Zhou Zhou, Gustavo Sato dos Santos, Thomas Dowrick, James Avery, Zhaolin Sun, Hui Xu, and David S Holder. Comparison of total variation algorithms for electrical impedance tomography. *Physiological Measurement*, 36(6):1193, 2015.
-





# Appendix A

## Learning objectives and plan

### A.1 Final version

#### A.1.1 Learning objectives

In addition to the overarching learning objectives, stated in the MSc in Mathematical Modelling and Computation (DTU) Programme Specification (<http://sdb.dtu.dk/2015/20/378>), a student completing this project will be able to:

- Understand the topics of functional analysis related to PDE's: The weak formulation of the forward problem, application of the Lax–Milgram Theorem for variational problems.
- Use Python for scientific computing, specifically understand how to use the software package, FEniCS, for solving the forward and inverse problems in Hybrid Tomography using the Finite Element Method.
- Theoretically understand and implement iterative methods for inverse problems. This includes: How to derive a continuity and differentiability (Fréchet derivative) of the forward map and use it to construct a Newton-type algorithm.
- Apply FEniCS to Hybrid Tomography (inverse) problems and discuss the results in relation to known literature.

### A.1.2 Time plan

The main objectives to be completed (writing process excluded) are the following:

1. Literature reading: Functional analysis topics, research papers on hybrid tomography.
2. Theory and implementation (FEniCS) for the forward problem.
3. Computation of interior data  $H(\sigma, u)$  with FEniCS.
4. Theoretical understanding of the inverse problem.
5. Implementation of inverse algorithms given interior data.

Writing will be done in parallel with these tasks, according to the more specific time plan shown in Fig A.1.

Christian Kragh, Lyngby, July 10, 2016.

## A.2 Note on corrections to the original plan

The learning objectives have been modified such that they correspond to the actual subjects that were dealt with in the project. Moreover, they contain more specific elements than the original one which can be seen in Appendix A.3.

The time plan was modified, since the work with implementations of the power density problem demanded more time than was expected, due to the fact that both the programming language `Python` and the FEM software `FEniCS` were subjects that I was not previously familiar with.

## A.3 Original version

### A.3.1 Learning objectives

In addition to the overarching learning objectives, stated in the MSc in Mathematical Modelling and Computation (DTU) Programme Specification (<http://sdb.dtu.dk/2015/20/378>), a student completing this project will be able to:

- Understand the topics of functional analysis related to PDE's and the Finite Element method: Sobolev spaces, Poincaré inequality, the Lax-Milgram Theorem, the weak formulation.
- Use Python for scientific computing, specifically understand how to use the software package, **FEniCS** for solving the forward and inverse problems in Hybrid Tomography.
- Theoretically understand and implement optimization methods for inverse problems. This includes calculus of variations, functional (Fréchet) derivatives, Gauss-Newton and similar methods, deconvolution algorithms.
- Apply **FEniCS** to Hybrid Tomography (inverse) problems and discuss the results in relation to known literature.

### A.3.2 Time plan

The main objectives to be completed (writing process excluded) are the following:

1. Literature reading: Functional analysis topics, research papers on tomography.
2. Theory and implementation (FEniCS) for the forward problem.
3. Computation of internal data  $H(\sigma, u)$  and convolutions of  $H$  with FEniCS.
4. Theoretical understanding of the inverse problem.
5. Implementation of inverse (optimization) algorithms given internal data.
6. Implementation of inverse (optimization) algorithms given convolutions of  $H$  (internal data).

Writing will be done in parallel with these tasks, according to the more specific time plan shown in Fig A.1.

Christian Kragh, Lyngby, February 24, 2016.

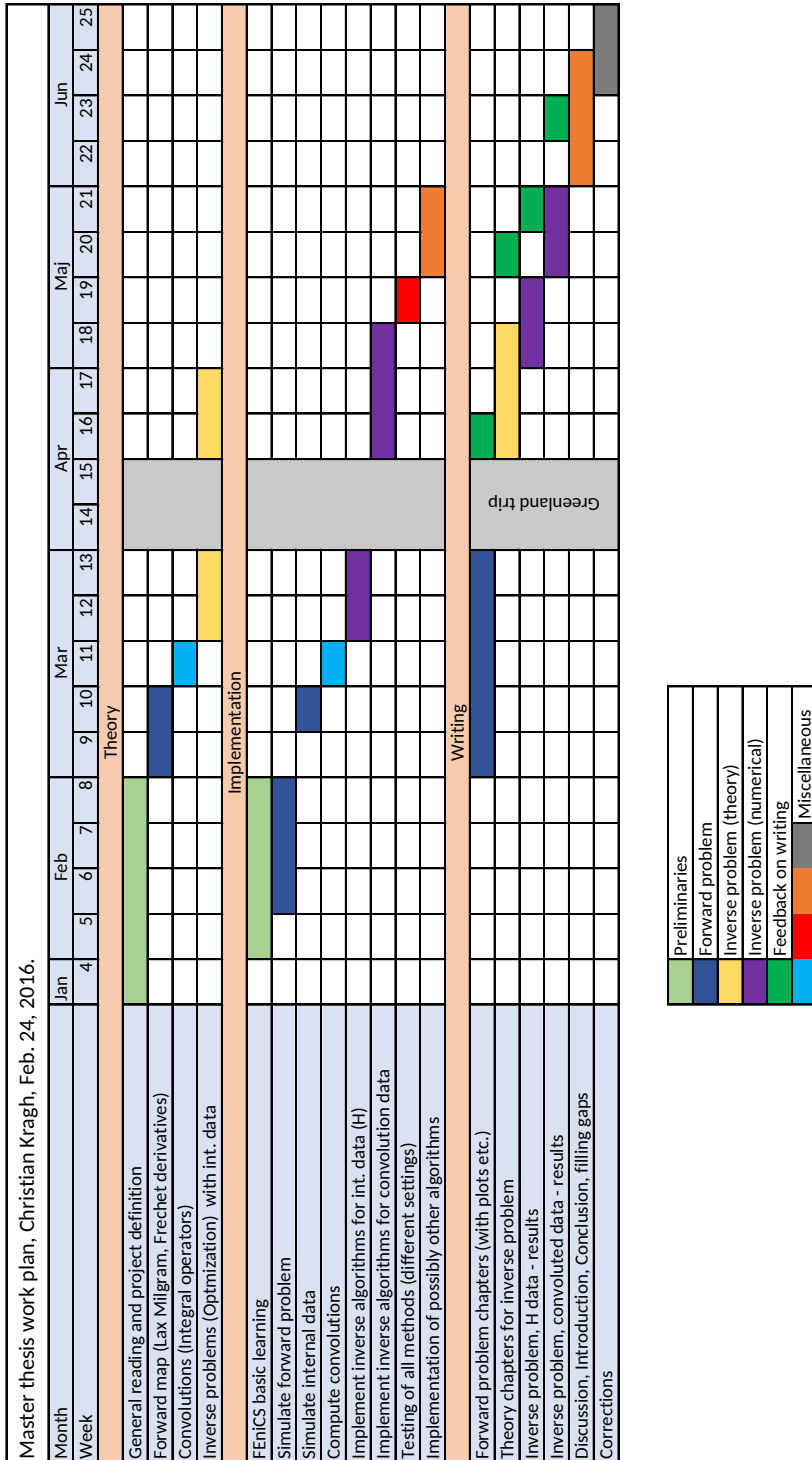


Figure A.1: Time plan for the Master Thesis with start date January 25 2016, to be submitted June 25 2016.

# Appendix B

## Python code

### Algorithm1.py

---

```
1 from dolfin import *
2 import numpy as np
3 import mshr
4
5 def NewtonIter(mesh,u0,sigma0,p,alpha,Hdiff,V,f,bc=True):
6
7 # Define function space
8 R = FunctionSpace(mesh, "R", 0) # constant space
9 W = MixedFunctionSpace([V, V,R]) # product space
10
11 # Define trial and test functions
12 (deltas,deltau,c) = split(TrialFunction(W))
13 (phi,psi,d) = split(TestFunction(W))
14 dx1 = dx(mesh)
15 ds1 = ds(mesh)
16
17 def boundary(x, on_boundary):
18     return on_boundary
19
20 bc1=DirichletBC(W.sub(0), 0.0, boundary)
21
22
23 # q1 = (div( deltaz*grad(u0)) + div(sigma0*deltaw) )* \
24 #       (div( v3* grad(u0))+ div(sigma0*v4) )*dx1
25 q1 = inner( deltas*grad(u0)+sigma0*grad(deltau),grad(psi))*dx1
26 # Power density |grad u|^p
```

```

27     u0sqr=inner(grad(u0),grad(u0))
28     pwd = u0sqr**(p/2)
29
30     if abs(p-2.0) < 0.001:
31         q2 = (deltas*u0sqr + p*sigma0*\
32             inner(grad(u0),grad(deltau)))*phi*dx1
33     else:
34         q2 = (deltas*pwd + p*sigma0*( pwd/u0sqr )*\
35             inner(grad(u0),grad(deltau)))*phi*dx1
36
37     reg = alpha*(deltas*phi + deltau*psi)*dx1
38
39     L = Hdifff*phi*dx1
40     LagrM = (c*psi + d*deltau)*ds1
41     q3 = -deltas*(f/sigma0)*psi*ds1
42     #     q3=0
43     a = q1+q2+reg+LagrM # remove q3 if delta_sigma = 0 on boundary
44
45     # Solve the system
46     w = Function(W)
47
48     if bc == True:
49         solve(a == L,w,bc1)
50     elif bc == False:
51         a = a+q3
52         solve(a == L,w)
53         #     solve(a == L,w,bc1)
54         # Alternative condition delta_sigma=0 on boundary
55     (x1,x2,c) = w.split(deepcopy = True)
56
57     return x1

```

---

## Algorithm2.py

```

1  from dolfin import *
2  import numpy as np
3  import mshr
4
5  def NewtonIterFull(mesh,w0,sigma0,p,alpha,Hdifff,V,Vg,w2=0,Hdifff2=0):
6
7      # Define function space
8      R = FunctionSpace(mesh, "R", 0) # constant space
9
10     if w2==0:
11         #Define trial and test functions

```



---

```

12     W = MixedFunctionSpace([V, V, V,Vg,R])
13     (deltas,deltau,deltaz,deltaw,c) = split(TrialFunction(W))
14     (v1,v2,v3,v4,d) = split(TestFunction(W)) # product space
15     deltau2 = 0
16     deltaw2 = 0
17     v5=0
18     v6=0
19 else:
20     W = MixedFunctionSpace([V, V, V,Vg,V,Vg,R,R])
21     (deltas,deltau,deltaz,deltaw,deltau2,deltaw2,c,c2)\
22     = split(TrialFunction(W))
23     (v1,v2,v3,v4,v5,v6,d,d2) = split(TestFunction(W))
24
25 dx1 = dx(mesh)
26 ds1 = ds(mesh)
27
28 def boundary(x, on_boundary):
29     return on_boundary
30
31 bc1=DirichletBC(W.sub(0), 0.0, boundary)
32 bc2=DirichletBC(W.sub(2), 0.0, boundary)
33
34 def Q1(deltaz,deltaw,w0):
35     q1 = inner(grad(deltaz),w0)\
36         + deltaz*div(w0) \
37         + inner(grad(sigma0),deltaw) \
38         + sigma0*div(deltaw)
39     return q1
40
41 q11 = Q1(deltaz,deltaw,w0)*Q1(v3,v4,w0)
42
43 w0sqr = inner(w0,w0)
44 pwd = w0sqr**(p/2)
45
46 def Q2(deltaz,deltaw,w0,w0sqr,new=False):
47
48     if p == 2:
49         q2 = deltaz*w0sqr + p*sigma0*inner(w0,deltaw)
50     else:
51         q2 = deltaz*pwd + p*sigma0*( pwd/w0sqr )*\
52             inner(w0,deltaw)
53     return q2
54
55 q21 = Q2(deltaz,deltaw,w0,w0sqr)*Q2(v3,v4,w0,w0sqr)
56
57
58 def G(deltau,deltaw,v2,v4):

```

```

59     g = inner( grad(deltau)-deltaw,grad(v2)-v4)
60     return g
61 g = (deltas-deltaz)*(v1-v3)
62 g = g+G(deltau,deltaw,v2,v4)
63 #
64 #     if w2!=0:
65 #         g = g+G(deltau2,deltaw2,v5,v6)
66 #
67 #     g = g
68
69 def Reg(deltas,deltau1):
70     reg = alpha*(deltas*v1 + deltau1*v2)
71 #     reg = alpha*(inner(grad(deltas),grad(v1)) + deltau1*v2)
72     return reg
73 reg = Reg(deltas,deltau)
74
75 def Lfun(Hdiff,v3,v4,w0,w0sqr,pwd):
76     l1 = Hdiff*( v3 *pwd + p*sigma0*( pwd/w0sqr )\
77     *inner(w0, v4 ))
78
79     return l1
80
81 L1 = Lfun(Hdiff,v3,v4,w0,w0sqr,pwd)
82
83 LagrM = (c*v2 + d*deltau)
84
85 if w2 !=0:
86     w2sqr = inner(w2,w2)
87     pwd2 = w2sqr**(p/2)
88
89     q12 = Q1(deltaz,deltaw2,w2)*Q1(v3,v6,w2)
90     q22 = Q2(deltaz,deltaw2,w2,w2sqr)*Q2(v3,v6,w2,w2sqr)
91     g = g+G(deltau2,deltaw2,v5,v6)
92     reg = reg + alpha*deltau2*v5
93     L2 = Lfun(Hdiff2,v3,v6,w2,w2sqr,pwd2)
94     LagrM = LagrM + c2*v5 + d2*deltau2
95 else:
96     q12 = 0
97     q22 = 0
98     L2 = 0
99
100 LagrM = LagrM*ds
101 #     reg = alpha*sqrt(sqrt(inner(grad(deltas),grad(v1))))
102 a = (q11+q12+q21+q22+g+reg)*dx1+LagrM
103 L = (L1+L2)*dx1
104 #     anew = q1+q2new+g+reg+LagrM
105 # Solve the system

```

---

```

106     w = Function(W)
107     solve(a == L, w, [bc1, bc2])
108     if w2 == 0:
109         (x1, x2, x3, x4, c1) = w.split(deepcopy = True)
110     else:
111         (x1, x2, x3, x4, x5, x6, c1, c2) = w.split(deepcopy = True)
112     return x1

```

---

## forward\_functions.py

---

```

1  from dolfin import *
2  import numpy as np
3  import mshr
4  """
5  Functions for solving the forward problem and computing internal data
6  """
7
8  """
9  Solver for the forward problem with a given conductivity sigma
10 and a current density f on the boundary of the domain.
11
12     div(sigma*grad(u)) = 0
13     sigma*du/dn       = f
14
15 Input:
16     Mesh 'mesh'           the finite element mesh
17     Function 'sigma'      the conductivity defined in the domain
18     Function 'f'          current density on the boundary
19
20 Output:
21     Function 'u'          solution of the forward problem
22 """
23 def EITForwardSolve(mesh, sigma, f, V):
24
25     # Define function space
26     V = FunctionSpace(mesh, "CG", 1)
27     R = FunctionSpace(mesh, "R", 0) # ----- constant space
28     W = V * R # ----- product space
29
30     # Define trial and test functions
31     (u, c) = TrialFunctions(W)
32     (v, d) = TestFunctions(W)
33
34     # Define forms and Lagrange multipliers
35     lagrMult = (v*c + u*d)*ds

```

```

36     a = sigma*inner(grad(u), grad(v))*dx + lagrMult
37     L = f*v*ds
38
39     # Solve the system
40     w = Function(W)
41     solve(a == L, w)
42     (u, c) = w.split(deepcopy = True)
43
44     return u
45
46 def EITForwardSolve1storder(mesh, sigma, f, V, Vg):
47
48     # Define function space
49     # V = FunctionSpace(mesh, "CG", 1)
50     R = FunctionSpace(mesh, "R", 0) # ----- constant space
51     W = MixedFunctionSpace([V, Vg, R]) # ----- product space
52     # Define trial and test functions
53     (u, w, c) = split(TrialFunction(W))
54     (v, phi, d) = split(TestFunction(W))
55
56     # Define forms and Lagrange multipliers
57     lagrMult = (v*c + u*d)*ds
58
59     a = sigma*inner(w, grad(v))*dx + lagrMult
60     a = a + inner(w-grad(u), phi)*dx
61     L = f*v*ds
62
63     # Solve the system
64     z = Function(W)
65     solve(a == L, z)
66     (u, w, c) = z.split(deepcopy = True)
67
68     return u, w, c
69
70 def ComputeH(sigma, grad_u, p):
71     # solve(a == L, H)
72     H = sigma*dot(grad_u, grad_u)**(p/2)
73     return H
74
75 def ComputeHnew(sigma, u, p, V):
76     v = TestFunction(V)
77     w = TrialFunction(V)
78     #
79     a = inner(w, v)*dx
80     L = inner(sigma*dot(grad(u), grad(u)), v)*dx
81     H = Function(V)
82     solve(a == L, H)

```

```
83     return H
84
85
86 def project1(sigma,V):
87
88     v = TestFunction(V)
89     w = TrialFunction(V)
90     #
91     a = inner(w, v)*dx
92     L = inner(sigma, v)*dx
93     z = Function(V)
94     problem = LinearVariationalProblem(a,L,z)
95     solver = LinearVariationalSolver(problem)
96     solver.parameters["krylov_solver"]["relative_tolerance"] = 5e-6
97     solver.parameters["krylov_solver"]["monitor_convergence"] = True
98     solver.solve()
99     return z
100
101 def ComputeGradient(mesh,u):
102     V_g = VectorFunctionSpace(mesh, 'Lagrange', 1)
103     V = FunctionSpace(mesh, 'Lagrange', 1)
104     project(grad(u),V_g)
105     v = TestFunction(V_g)
106     w = TrialFunction(V_g)
107     a = inner(w, v)*dx
108     L = inner(grad(u), v)*dx
109     grad_u = Function(V_g)
110     solve(a == L, grad_u)
111     return grad_u
112
113
114 def get_centers():
115     eps = 0.2
116     cx1 = -0.5
117     cy1 = -0.1
118     r1 = 0.3
119
120     cx2 = 0.2
121     cy2 = 0.5
122     r2 = 0.25
123
124     cx3 = 0.4
125     cy3 = -0.3
126     r3 = 0.25
127
128     C1 = 1
129     C2 = 1.5
```

---

```

130     C3 = 1.2
131
132     ## Test values
133     #   C1 = 0
134     #   C2 = 0
135     #   C3 = 1
136     #   eps = 0.2
137
138     C = [C1,C2,C3]
139     cx = [cx1,cx2,cx3]
140     cy = [cy1,cy2,cy3]
141     rad = [r1,r2,r3]
142     return C,cx,cy,rad,eps
143
144 def polsmooth(r,eps):
145     pol = -6*r**5/(eps**5)+ \
146         15*r**4/(eps**4)-10*r**3/(eps**3)+1
147     return pol

```

---

## callAlgorithms.py

---

```

1  """
2  Script that calls Algorithm 1 and 2
3  and performs the Newton iterations
4  """
5
6  from scipy import misc
7
8  from dolfin import *
9  import numpy as np
10 import mshr
11
12 #
13 ## Import the EIT-functions from another module
14 ## including the forward-solver functions for the
15 ## interior data computation.
16
17 from forward_functions import *
18 from Algorithm1 import *
19 from Algorithm2 import *
20 from EITplotter import *
21
22 ## Compute the interior data by solving the forward problem
23 ## on the disk of radius r.
24

```

```
25 import time
26 #
27 time1 = time.time()
28
29 alpha = 0
30 pval = [0.5,0.75,1.0,1.5,2.0]
31 M = 1
32 K = 20
33 n = 1
34 jrange = range(2,3)
35
36 #parameters["num_threads"] = 2
37 #parameters["form_compiler"]["cpp_optimize"] = True
38
39 err = np.zeros((5,K+1))
40 #
41 res1 = 100
42 res2 = 60
43
44 res1true = 120
45 res2true = 90
46
47 # Define the Finite Element triangulization
48 mesh_rad=1.0
49 omega1 = mshr.Circle(Point(0.,0.),mesh_rad,res1)
50 mesh1 = mshr.generate_mesh(omega1, res2, "cgal")
51
52 omega_true = mshr.Circle(Point(0.,0.),mesh_rad,res1true)
53 mesh_true = mshr.generate_mesh(omega_true, res2true, "cgal")
54
55
56 V1 = FunctionSpace(mesh1,"CG",1)
57 Vtrue = FunctionSpace(mesh_true,"CG",1)
58
59 # Define boundary input
60 f1 = Expression('x[0]')
61 f2 = Expression('x[0]+x[1]')
62
63 f1str = 'x'
64 f2str = 'x+y'
65
66 class NeumannCos(Expression):
67     def eval(self, values, x):
68         theta = np.arctan2(x[1],x[0])
69         values[0] = cos(n*theta)
70
71 class NeumannSin(Expression):
```

```

72     def eval(self, values, x):
73         theta = np.arctan2(x[1],x[0])
74         values[0] = sin(n*theta)
75
76 class NeumannSinCos(Expression):
77     def eval(self, values, x):
78         theta = np.arctan2(x[1],x[0])
79         values[0] = sin(n*theta)+cos(n*theta)
80
81 # Define the target conductivity. Here we choose a conductivity which
82 # is piece-wise constant, with a different constant value inside two
83 # ellipses within the domain
84
85 C,cx,cy,rad,eps1 = get_centers()
86 cx1,cy1,r1 = cx[0],cy[0],rad[0]
87 cx2,cy2,r2 = cx[1],cy[1],rad[1]
88 cx3,cy3,r3 = cx[2],cy[2],rad[2]
89 C1,C2,C3 = C[0],C[1],C[2]
90
91 #f1 = NeumannCos()
92 #f2 = NeumannSinCos()
93
94 # Piece-wise constant conductivity
95 class conductivityFun(Expression):
96     def eval(self, values, x):
97         X1 = x[0]-cx1
98         Y1 = x[1]-cy1
99
100         R1 = np.sqrt(1.5*X1**2 + 1.5*Y1**2 + X1*Y1)
101         R2 = np.sqrt( (x[0] - cx2)**2 + 2*(x[1] - cy2)**2 )
102         R3 = np.sqrt( (x[0] - cx3)**2 + (x[1] - cy3)**2 )
103 #         if np.sqrt( 2*(x[0] - cx1)**2 + (x[1] - cy1)**2 ) < r1:
104         if R1 < r1:
105             values[0] = 1 + C1
106         elif R2 < r2:
107             values[0] = 1 + C2
108         elif R3 < r3:
109             values[0] = 1 + C3
110         else:
111             values[0] = 1
112
113 class conductivityExp(Expression):
114     def eval(self, values, x):
115         R1 = 2*(x[0] - cx1)**2 + (x[1] - cy1)**2
116         R2 = (x[0] - cx2)**2 + 2*(x[1] - cy2)**2
117         R3 = (x[0] - cx3)**2 + (x[1] - cy3)**2
118

```



---

```

119     steepness = 13
120     values[0] = 1+2*np.exp(-steepness*R1)+\
121                 1*np.exp(-steepness*R2)\
122                 +1.5*np.exp(-steepness*R3)
123
124 class conductivitySmooth(Expression):
125     def eval(self, values, x):
126         X1 = x[0]-cx1
127         Y1 = x[1]-cy1
128         eps=eps1
129         R1 = np.sqrt(1.5*X1**2 + 1.5*Y1**2 + X1*Y1)
130         R2 = np.sqrt( (x[0] - cx2)**2 + 2*(x[1] - cy2)**2 )
131         R3 = np.sqrt( (x[0] - cx3)**2 + (x[1] - cy3)**2 )
132
133         values[0] = 1
134         r1new = r1+0.5*eps
135         r2new = r2+0.5*eps
136         r3new = r3+0.5*eps
137         if R1 < r1new:
138             values[0] = 1 + C1*polsmooth(R1-(r1new-eps),eps)
139             if R1 < (r1new-eps):
140                 values[0] = 1+C1
141         if R2 < r2new:
142             values[0] = 1 + C2*polsmooth(R2-(r2new-eps),eps)
143             if R2 < (r2new-eps):
144                 values[0] = 1 + C2
145         #         values[0] = 1
146         if R3 < r3new:
147             values[0] = 1 + C3*polsmooth(R3-(r3new-eps),eps)
148             if R3 < (r3new-eps):
149                 values[0] = 1+ C3
150
151 sigmaFun = conductivitySmooth()
152 sigma_true = project(sigmaFun,Vtrue)
153 #sigmaFun=Expression('1')
154
155 l1 = LagrangeInterpolator()
156 sigma_trueV1 = Function(V1)
157 l1.interpolate(sigma_trueV1, sigma_true)
158
159 norm_sigma_true = norm(sigma_trueV1, 'L2')
160
161 #plot(sigma_true)
162 u_true1 = EITForwardSolve(mesh_true, sigma_true, f1, Vtrue)
163 u_true2 = EITForwardSolve(mesh_true, sigma_true, f2, Vtrue)
164 #
165

```

```

166 #
167 for j in xrange:
168     p = pval[j]
169     print('p=')
170     print(p)
171     #grad_u_true = ComputeGradient(mesh,sigma,u_true)
172
173     Htrue1 = ComputeH(sigma_true,grad(u_true1),p)
174     Htrue2 = ComputeH(sigma_true,grad(u_true2),p)
175
176     parameters['allow_extrapolation'] = True
177     Htrue1 = project(Htrue1,Vtrue)
178     Htrue2 = project(Htrue2,Vtrue)
179     parameters['allow_extrapolation'] = False
180
181     HV1 = Function(V1)
182     HV2 = Function(V1)
183     ll.interpolate(HV1,Htrue1)
184     ll.interpolate(HV2,Htrue2)
185     Htrue1=HV1
186     Htrue2=HV2
187
188 #     sigma_true = sigma_trueV1
189 #     sigma_true = project(sigma_true,V1)
190
191 # =====
192 # Newton-type algorithm for inverse problem
193 # =====
194
195 C1 = 0
196 C2 = 0
197 C3 = 0
198 sigma0 = conductivityFun()
199 sigma0 = interpolate(sigma0,V1)
200
201 Vg = VectorFunctionSpace(mesh1, 'CG', 1)
202 err[j,0]=norm(project((sigma_trueV1-sigma0),V1),'L2')\
203             /norm_sigma_true
204 print err[j,0]
205
206 for k in range(1,K+1):
207     u1 = EITForwardSolve(mesh1,sigma0,f1,V1)
208     u2 = EITForwardSolve(mesh1,sigma0,f2,V1)
209
210 # First-order forward prob
211 #     u1,w1,c1 = EITForwardSolve1storder(mesh1,sigma0,f1,V1,Vg)
212 #     u2,w2,c2 = EITForwardSolve1storder(mesh1,sigma0,f2,V1,Vg)

```

```

213
214     # Newton iteration
215     H01 = ComputeH(sigma0,grad(u1),p)
216     Hdiff1 = Htrue1-H01
217     H02 = ComputeH(sigma0,grad(u2),p)
218     Hdiff2 = Htrue2-H02
219
220     # 1 measurement - Algorithm 1
221     if M==1:
222         dsigma = NewtonIter(mesh1,u1,sigma0,p,alpha,
223                             Hdiff1,V1,f1,bc=True)
224
225     # 2 measurements
226     if M==2:
227         dsigma = NewtonIterFull(mesh1,w1,sigma0,p,alpha,
228                                 Hdiff1,V1,Vg,w2,Hdiff2)
229
230     sigma0 = sigma0+dsigma
231
232     # Compute L2-norm of error
233     parameters['allow_extrapolation'] = True
234     norm_sigma_true = norm(sigma_trueV1,'L2')
235     sigma_diff = project((sigma_trueV1-sigma0),V1)
236     dsig_norm = norm(dsigma,'L2')
237     print(dsig_norm)
238     err[j,k]=norm(project((sigma_trueV1-sigma0),V1),'L2')\
239                /norm_sigma_true
240     print(err[j,k])
241     parameters['allow_extrapolation'] = False
242
243     # if alpha < 0.1:
244     # alpha = alpha*10
245
246     time2 = time.time()
247
248     elapsed = time2-time1
249
250     print("Elapsed time:")
251     print(elapsed)
252
253     #V1 = FunctionSpace(mesh1,"CG",1)
254     #
255     #z1 = sigmaV.compute_vertex_values()
256     sigmaV = project(sigma0,V1)
257     sigmaVtrue = Function(Vtrue)
258     ll.interpolate(sigmaVtrue,sigmaV)
259

```

```
260 z0 = sigmaVtrue.compute_vertex_values()
261 z1 = sigmaV.compute_vertex_values()
262 z2 = sigma_true.compute_vertex_values()
263 z3 = sigma_trueV1.compute_vertex_values()
264
265 print("Test succeeded.")
266 #r1,theta1,data1 = polardata(sigmaV,mesh1,mesh_rad)
267 #r2,theta2,data2 = polardata(sigma_true,mesh_true,mesh_rad)
268 #
269 params = np.array([p,n,M,K,alpha,res1,res1true,f1str,f2str,\
270                  C[0],C[1],C[2],res2,res2true])
271 np.save('params.npy',params)
272 np.save('err.npy',err)
273 np.save('sigmaVtrue.npy',z0)
274 np.save('sigmaV1.npy',z1)
275 np.save('sigma_trueVtrue.npy',z2)
276 np.save('sigma_trueV1.npy',z3)
277 #np.save('r1',r1)
278 #np.save('theta1',theta1)
279 #np.save('data1',data1)
```

---