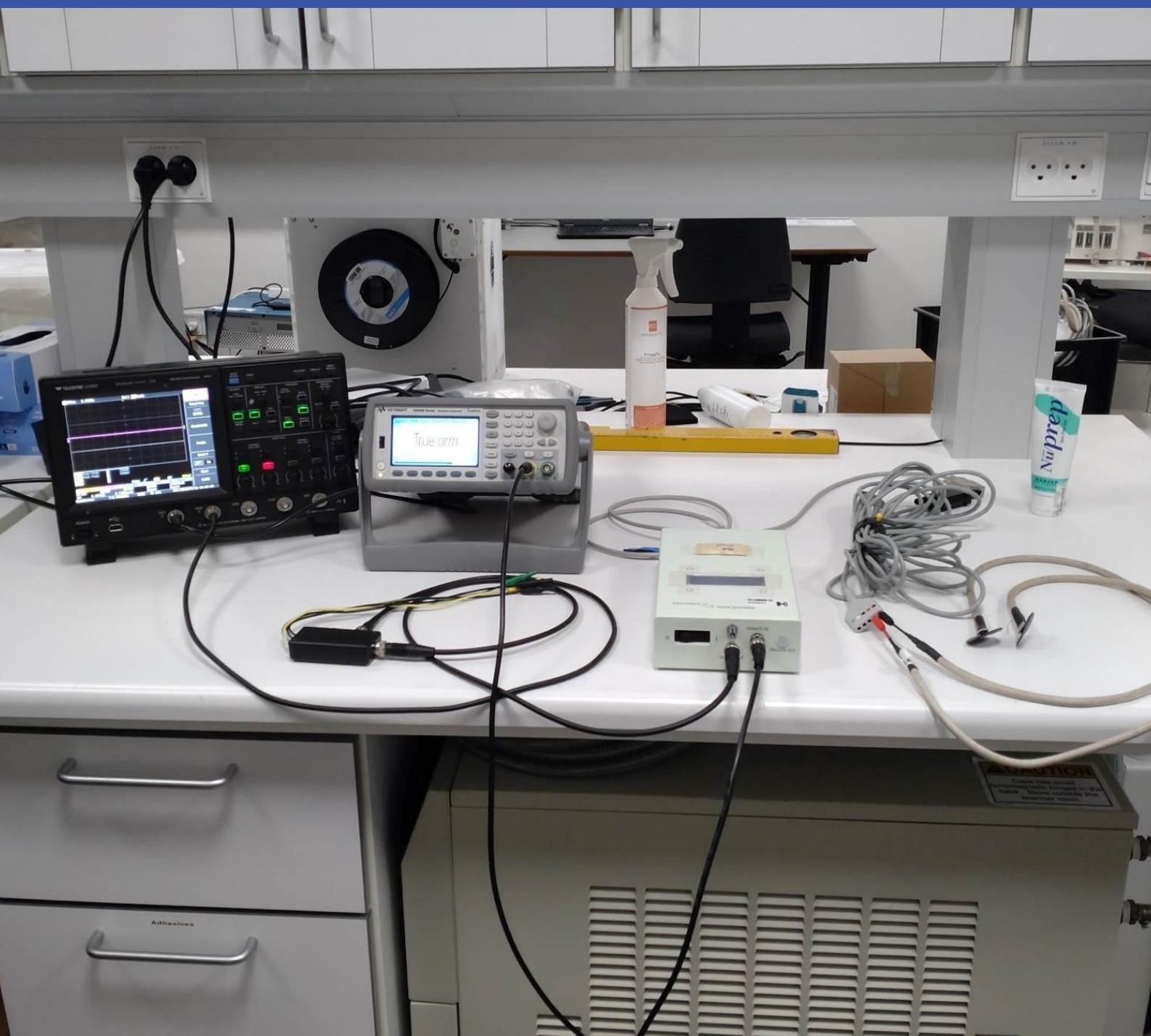


Uncertainty Quantification for conductivity imaging from information about corresponding magnetic fields

Master Thesis

Kenneth Scheel - s174488



Uncertainty Quantification for conductivity imaging from information about corresponding magnetic fields

Master Thesis
January, 2023

By
Kenneth Scheel

Copyright: Reproduction of this publication in whole or in part must include the customary bibliographic citation, including author attribution, report title, etc.

Cover photo: Photo of one half of an experimental MREIT setup, MRI machine excluded. Shown from left to right: An oscilloscope to make sure everything is working. A function generator that controls the amplitude and frequency of the current to be injected. A neuro-signal generator supplies the two attached electrodes with low-frequency current according to the function. The wires on the electrodes are made of rubber, such that they are visible on an MRI scan. Photo by Kenneth Scheel at Danish Research Centre for Magnetic Resonance, Hvidovre Hospital, January 2023

Published by: DTU, Department of Applied Mathematics and Computer Science, Matematiktorvet, Building 303b, 2800 Kgs. Lyngby Denmark
www.compute.dtu.dk

Approval

This thesis has been prepared over five months at the Section for Scientific Computing, Department of Applied Mathematics and Computer Science, at the Technical University of Denmark, DTU, in partial fulfilment for the degree Master of Science in Engineering, MSc Eng.

It is assumed that the reader has a basic knowledge in the areas of partial differential equations, probability theory and functional analysis.

Abstract

MREIT is a promising approach for non-invasive conductivity imaging. The method combines Electrical Impedance Tomography (EIT) and Magnetic Resonance Imaging (MRI). An MR-scanner measures the magnetic field induced by the externally applied current field from the EIT setup. This provides interior information in the form of magnetic field data \mathbf{B} . Knowing the electrical conductivity properties of an organ in a patient is vital in some medical applications, e.g. non-invasive brain stimulation where small currents are applied through electrodes to stimulate brain activity in certain regions. When performing conductivity imaging it might be that an unknown inclusion is detected. How certain can we be that the inclusion is actually there and not just a reconstruction artifact from the algorithm? Can we quantify this uncertainty mathematically?

To answer these questions one may use Bayesian inversion theory which casts the inverse problem as one of statistical inference. All quantities of interest are modeled as random variables. We parameterize the conductivity field in the object to be imaged by $\sigma(p)$, where p are unknown parameters defining the conductivity field. Then the forward problem in MREIT can be modeled as an inhomogeneous elliptic partial differential equation in three dimensions. The Bayesian solution to the inverse problem is a probability distribution called the *posterior*, which we use to infer information about the parameters p given magnetic field data. The forward model in our inverse problem is nonlinear, highly complex, and expensive to evaluate. This makes the posterior intractable. Since we want to reconstruct the conductivity field σ and quantify the standard deviation of σ in the images, it amounts to characterizing the posterior by estimating its moments and statistics. Doing this for our intractable posterior is the main topic of interest.

Acknowledgements

I want to express my utmost gratitude towards my supervisors Prof. Kim Knudsen and postdoc Babak Maboudi Afkham, for their expert guidance and wise ideas during our weekly sparring sessions. I especially want to thank you for all the opportunities you have provided me during the project. It has been a fruitful experience that has taught me many things and shaped my future in multiple ways.

Multiple issues arose along the way which we were not able to solve ourselves. In that regard, I want to thank postdoc Amal Mohammed A. Alghamdi from the CUQI team at DTU Compute for teaching me adjoint state methods for gradient computations, for providing insight on PDE-constrained optimization problems, and for debugging my optimization code. Your support proved valuable multiple times and I am very grateful that you wanted to spend your time helping me.

I also want to thank Axel Thielscher from DTU Health for hosting us in the laboratory at the Danish Research Centre for Magnetic Resonance (DRCMR) at Hvidovre Hospital. It was a very interesting visit where we got to look at an MRI machine and some MREIT equipment, while also learning about the experimental part of MREIT from experts in the field.

My sincere gratitude goes out to the entire CUQI team for allowing me to participate in the weekly seminars and for providing delicious cookies each Tuesday throughout the project period. I should especially thank CUQI for hosting and inviting me along to the IUQ22 workshop in Helsingør at the beginning of the project period, and for allowing me to do a poster presentation on the research goals of this thesis. Going to IUQ22 was a great experience as I got to meet my future Ph.D. supervisor Prof. Aku Seppänen from the University of Eastern Finland, who invited me to Inverse Days 2022 in December to give a short pitch talk on my thesis. In that regard, I want to thank senior researcher Jakob Sauer Jørgensen and all the Ph.D. students from the CUQI team who were with me at Inverse Days for a wonderful time in Finland. Lastly, I want to thank the Villum Foundation, which funds the CUQI project, and the William Demant Foundation for their generous financial contributions to my trip to Finland.

Kenneth Scheel, MSc Engineering, DTU Compute
Author of this thesis

Kim Knudsen, Professor, DTU Compute
Supervisor

Babak Maboudi Afkham, Post-doc, DTU Compute
Co-supervisor

Contents

Preface	ii
Abstract	iii
Acknowledgements	iv
1 Introduction	1
2 Magnetic Resonance Electrical Impedance Tomography	2
2.1 The forward problem	2
2.2 The inverse problem	3
3 Bayesian inverse problems	6
3.1 Markov chain Monte Carlo methods	7
3.2 Chain diagnostics	16
3.3 Posterior inference	20
4 The finite element method and spectral convolution methods	22
4.1 Finite element method for approximating PDE	22
4.2 Spectral method for fast evaluation of magnetic fields	27
5 Computational solution procedure	30
5.1 Building the posterior	30
5.2 Simulation of forward problem	32
5.3 Solution of the Bayesian inverse problem	35
6 Numerical results	41
6.1 Benchmarking	41
6.2 Problem 0: spherical inclusion with known conductivity	44
6.3 Problem 1: two spherical inclusions with known conductivity	47
6.4 Problem 2: rotated ellipsoidal inclusion with unknown conductivity	52
6.5 Problem 3: elliptic inclusion in pure 2D with unknown conductivity	56
6.6 Problem 4: sparse magnetic field data in z -direction	61
6.7 Problem 5: two inclusions with large conductivity contrast	66
7 Discussion	73
7.1 Outlook	75
7.2 Conclusion	76
A Appendix	77
Bibliography	83

1 Introduction

Non-invasive imaging in medical and industrial use can often be framed as an inverse problem. CT scans at the hospital, radar- and sonar technologies, and seismic imaging to name a few, are all examples of real-life applications of inverse problems. The Calderón problem in mathematics asks whether it is possible to determine the electrical conductivity distribution inside an object, by making voltage and current measurements on its boundary [1]. The method of Electrical Impedance Tomography (EIT) was developed to answer this question. In EIT, electrodes are attached to the surface of the object to be imaged and a small low-frequency current is injected. Voltage measurements are made by other electrodes and these are used in a reconstruction algorithm to produce an image. EIT suffers from ill-posedness as only boundary data is available, and only low-resolution conductivity images are achievable [2]. Recently, research has shown that coupled-physics imaging techniques can produce even higher contrast and higher-resolution conductivity images than standard EIT is capable of. One novel method combines EIT and Magnetic Resonance Imaging (MRI). By using an MRI machine to measure the magnetic field induced by the current field from the EIT setup, we gain interior information in the form of magnetic field data [3, 2]. Having interior data available makes the reconstruction problem in MREIT well-posed, compared to the ill-posed reconstruction problem of EIT. Computing a cross-sectional conductivity image of some region of a patient can be vital in medical applications. Current experimental research in MREIT is especially concerned with conductivity imaging of the brain, as the brain consists of several different regions with different conductivity levels and both isotropic and anisotropic properties [4, 5]. In non-invasive brain stimulation, doctors apply small currents through electrodes to stimulate brain activity in certain regions, to either restore or compensate for the lack of function in that region¹. Knowing the conductivity properties of the entire brain allows for more targeted and safer treatment of the patient.

More often than not, data in the real world is inflicted by noise, sparsity, and errors. It could also be that a mathematical model is simplified to such a slightly inaccurate degree. These phenomena mean that many parts of an inverse problem in real life are *uncertain*. We should be aware of these uncertainties and incorporate them into our computational models. Quantifying the uncertainty is therefore of interest as we want to characterize the sensitivity of solutions to inverse problems. This characterization could come in the form of knowing the standard deviation of each pixel in a slice of a patient's conductivity image. To accomplish this task, we use Bayesian inversion theory where one models all quantities of interest as random variables [6]. This allows us to encode our degree of information into the probability distributions, making uncertainty quantification possible by computing standard deviations and credibility intervals. The solution to the inverse problem is also a probability distribution called the *posterior*, which encodes all information about the unknown conductivity field σ in an object given interior magnetic field data. Exploring and characterizing the posterior is our main topic of interest in this thesis. At DTU we have the Computational Uncertainty Quantification for Inverse Problems (CUQI) research initiative funded by the Villum Foundation, which both project supervisors are a part of [7]. The goal of this thesis is to develop a Bayesian computational framework for solving the conductivity field reconstruction problem in MREIT.

¹https://www.hopkinsmedicine.org/physical_medicine_rehabilitation/services/programs/brain-stimulation/treatment.html

2 Magnetic Resonance Electrical Impedance Tomography

The MREIT experiment answers the question of whether it is possible to reconstruct the interior conductivity distribution σ of an object, given noisy measurements of the interior magnetic field. In EIT we only have access to boundary voltage measurements and these are insensitive to internal conductivity perturbations [8]. An MRI scanner is therefore used to measure the interior magnetic field induced by the externally applied current from the electrodes. This provides internal images of the magnetic flux density \mathbf{B} . Hidden in these images is information on the conductivity distribution σ of the object [3]. The inverse problem is to extract that information about σ from noisy observations of \mathbf{B} .

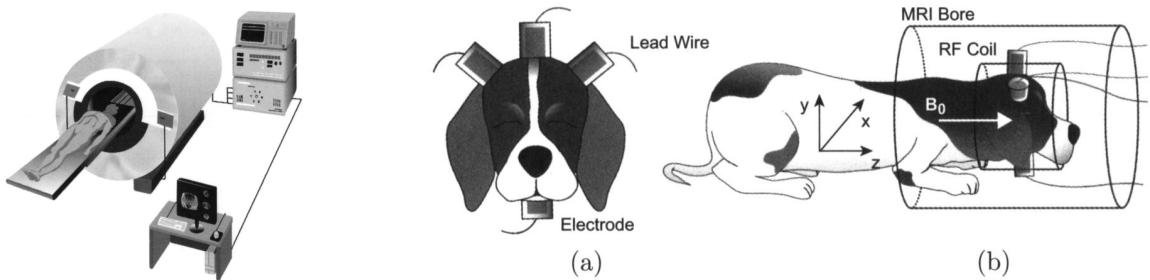


Figure 2.1: Left: Diagram of MREIT system with a human subject. Right: (a) Schematic of electrodes attached to a deceased dog. (b) Schematic of dog placed inside MRI bore. The scanner obtains MR magnitude and magnetic flux density images from slices of the head [3].

In a real setting, the MRI scanner measures only the z -component of the magnetic field \mathbf{B} in a few slices parallel to the z -axis as the coordinate system above indicates [9, 10]. Thus only sparse magnetic flux data is obtained in the z -direction. Obtaining the x - and y -component of the magnetic field would require undesired rotation of the subject inside the scanner. This can be very impractical for human subjects as shown in the left figure above. In a practical setting with living human subjects, only a very small current is injected. A positive and a negative current of ± 2 mA is injected with a frequency of 6 Hz. The current is just strong enough to produce a tingling sensation while not being painful when the electrodes are attached [4].

2.1 The forward problem

The forward problem in MREIT is the mapping from a known conductivity field $\sigma \in L_+^\infty(\Omega)$ to the magnetic field $\mathbf{B} \in [L^2(\Omega)]^3$ measured in the MRI machine. We can decompose the mapping into two stages. One operator $\mathcal{F} : \sigma \mapsto \mathbf{J}$ that maps a conductivity field σ to the induced current density field $\mathbf{J} \in [L^2(\Omega)]^3$. Then the operator $\mathcal{G} : \mathbf{J} \mapsto \mathbf{B}$ which maps \mathbf{J} to the induced magnetic field \mathbf{B} . The space $L_+^\infty(\Omega)$ consists of $L^\infty(\Omega)$ functions which are bounded from above and below by positive constants, i.e. $L_+^\infty(\Omega) = \{v \in L^\infty(\Omega) : v \geq v_0 > 0\}$. The requirement that $\sigma \in L_+^\infty(\Omega)$ means that σ is bounded and greater than 0. We also assume that σ is an isotropic conductivity field, i.e. that the current diffuses at the same rate in all directions. The first stage is very akin to the forward mapping in the EIT problem. It involves solving the Poisson equation for the potential $u \in H^1(\Omega)$ in Ω .

$$\begin{aligned} -\nabla \cdot \sigma \nabla u &= 0 && \text{in } \Omega, \\ u &= f && \text{on } \partial\Omega. \end{aligned} \tag{2.1}$$

The boundary data f is chosen to fit the characteristics of the problem. This we will discuss in section 5.2. The current density field \mathbf{J} inside Ω is given by Ohm's law

$$\mathcal{F} : \sigma(\mathbf{x}) \mapsto \mathbf{J}(\mathbf{x}) = -\sigma(\mathbf{x}) \nabla u(\mathbf{x}). \quad (2.2)$$

The magnetic field \mathbf{B} induced by the current density field \mathbf{J} in Ω are related via Ampère's law $\nabla \times \mathbf{B} = \mathbf{J}$. This expression is equivalent to the Biot-Savart law [11] which determines \mathbf{B} as a function of \mathbf{J}

$$\mathcal{G} : \mathbf{J}(\mathbf{x}) \mapsto \mathbf{B}(\mathbf{x}) = \frac{1}{4\pi} \int_{\Omega} \mathbf{J}(\mathbf{x}') \times \frac{\mathbf{x} - \mathbf{x}'}{|\mathbf{x} - \mathbf{x}'|^3} d\mathbf{x}'. \quad (2.3)$$

Due to the cross-product in Ampère's law the z -component of \mathbf{B} , B_3 , carries no information about the z -component of $\mathbf{J} = (J_1, J_2, J_3)$ [2]. In practice, the magnetic field \mathbf{B} is only measured along slices on the z -axis so we only obtain measurements of B_3 . Obtaining the x - and y - components of the magnetic field would require undesired rotation of the subject inside the bore as can be seen on the coordinate-axes on the dog in figure 2.1. Thus, we can only gain knowledge of (J_1, J_2) from the B_3 measurements. The full forward problem in MREIT is then to compute the magnetic field \mathbf{B} given a known conductivity distribution σ

$$\mathcal{G} \circ \mathcal{F} : \sigma(\mathbf{x}) \mapsto \mathbf{J}(\mathbf{x}) \mapsto \mathbf{B}(\mathbf{x}) = -\frac{1}{4\pi} \int_{\Omega} (\sigma(\mathbf{x}') \nabla u(\mathbf{x}')) \times \frac{\mathbf{x} - \mathbf{x}'}{|\mathbf{x} - \mathbf{x}'|^3} d\mathbf{x}'. \quad (2.4)$$

The forward problem in MREIT is well-posed so we know that solutions exist, given a well-defined conductivity distribution $\sigma \in L_+^\infty(\Omega)$. As long as we work in the correct spaces $u \in H^1(\Omega)$, $\mathbf{B} \in [L^2(\Omega)]^3$, etc. we should have well-defined solutions to all forward problems.

2.2 The inverse problem

The reconstruction problem in MREIT is to determine the electrical conductivity distribution $\sigma(x)$ given noisy magnetic field data B_3^{noise} . The inverse map of the forward problem is the steps $B_3 \mapsto \mathbf{J} \mapsto \sigma$. We want to reconstruct the best estimate of σ given the data, i.e. solve the equation

$$B_3^{\text{noise}} = E[(\mathcal{G} \circ \mathcal{F})(\sigma)], \quad (2.5)$$

where $E[\cdot]$ is some noise operator whether that be additive measurement noise, sparsely sampled data, or both. In this thesis, we consider additive noise in the form of

$$B_3^{\text{noise}} = (\mathcal{G} \circ \mathcal{F})(\sigma) + E, \quad (2.6)$$

where E is a noise realization from some probability distribution. Reconstructing the conductivity field $\sigma(x)$ given the data is not an easy task. The standard deterministic way of solving inverse problems is to apply regularization methods and use optimization to solve the problem

$$\min_{\sigma \in L_+^\infty(\Omega)} \|B_3^{\text{noise}} - (\mathcal{G} \circ \mathcal{F})(\sigma)\|^2 + R(\sigma), \quad (2.7)$$

where $R(\cdot)$ is some suitable regularization term. We have several options available for solving the inverse problem, we list some of them here

1. Deterministic methods via regularization and non-linear optimization [6]
2. Linearize and solve [2]
3. J-substitution method [12, 8]
4. Harmonic B_z algorithm [10, 8]
5. Bayesian inversion [6]

Standard deterministic approaches produce a single point estimate for σ as the solution to the inverse problem given the available data. The J-substitution method has shown promise for providing accurate high-resolution conductivity images in the deterministic case [8]. However, if we want to do uncertainty quantification, a single estimate is not going to do us much good. If we solve the inverse problem using Bayesian inversion, then we reformulate the problem as a statistical inference problem by using Bayesian statistics [6]. In this framework, all quantities are modeled as random variables. Information about each random variable is encoded in a probability distribution. Using probability distributions not only lets us encode which information we have about the quantities of interest; they also reflect our degree of uncertainty about each quantity. This approach is the main subject of the thesis and will be described in more detail next chapter.

2.2.1 Deterministic solution in MREIT via linearization

While the inverse problem of MREIT is well-posed, the experimental setup of the imaging technique has some major drawbacks. The EIT setup by itself only requires some electrodes, a low-frequency power source, and a computer to run the calculations and display the results, while MREIT requires the patient with attached electrodes to be placed inside a very expensive MRI machine for an extended period. The experimental setup in EIT is essentially portable, and since the setup allows for real-time monitoring of lung activity, emergency responders could use the technique for on-scene lung monitoring or stroke detection. Therefore studying the EIT problem, though ill-posed, is both very beneficial and interesting work. A deterministic solution of the EIT reconstruction problem using an iterative Newton's method along with classical regularization techniques has been developed in [2]. The forward mapping in EIT consists of computing $\mathbf{J} = -\sigma \nabla u$ after solving the Poisson eq. in (2.1) for the potential u . This is the first part of the forward mapping in MREIT which we denoted \mathcal{F} . We linearize the problem by computing the Fréchet derivative of the forward operator which in turn is used to derive the Newton procedure. The Fréchet derivative generalizes the notion of differentiability to operators and allows for the linearization of non-linear operators. Let us first define the forward mapping from a conductivity distribution σ to an interior current density field \mathbf{J}

$$\mathcal{F} : \sigma \mapsto \mathbf{J} = -\sigma \nabla u, \quad (2.8)$$

where $u \in H_0^1(\Omega)$ solves the PDE in (2.1). Let \tilde{u} be the solution to (2.1) where σ is replaced by a reference conductivity $\tilde{\sigma}$. If we take $\delta\sigma$ to be in a class of admissible variations and let $\sigma = \tilde{\sigma} + \delta\sigma$, we want to compute a linear operator $d\mathcal{F}(\tilde{\sigma})$ such that

$$\lim_{\delta\sigma \rightarrow 0} \frac{\|\mathcal{F}(\tilde{\sigma} + \delta\sigma) - \mathcal{F}(\tilde{\sigma}) - d\mathcal{F}(\tilde{\sigma})\delta\sigma\|_{L^2(\Omega)}}{\|\delta\sigma\|_{L^\infty(\Omega)}} = 0. \quad (2.9)$$

This usually involves doing a first-order approximation of the form

$$d\mathcal{F}(\tilde{\sigma})\delta\sigma \approx \mathcal{F}(\tilde{\sigma} + \delta\sigma) - \mathcal{F}(\tilde{\sigma}). \quad (2.10)$$

One can show that the Fréchet derivative is given by $d\mathcal{F}(\tilde{\sigma})\delta\sigma := -\tilde{\sigma} \nabla \delta u - \delta\sigma \nabla \tilde{u}$, where $\delta u \in H_0^1(\Omega)$ solves

$$\begin{aligned} \nabla \cdot \tilde{\sigma} \nabla \delta u &= -\nabla \cdot \delta\sigma \nabla \tilde{u} && \text{in } \Omega, \\ \delta u &= 0 && \text{on } \partial\Omega. \end{aligned} \quad (2.11)$$

It can be shown (Appendix 1) that the following estimate holds

$$\|\mathcal{F}(\tilde{\sigma} + \delta\sigma) - \mathcal{F}(\tilde{\sigma}) - d\mathcal{F}(\tilde{\sigma})\delta\sigma\|_{L^2(\Omega)} \leq C \|\delta\sigma\|_{L^\infty(\Omega)}^2. \quad (2.12)$$

The linearized EIT problem can then be obtained by replacing $\mathcal{F}(\tilde{\sigma} + \delta\sigma)$ with the interior current density field data \mathbf{J} in the first order approximation in (2.10). Since \mathbf{J} can be simulated or

measured via the EIT setup we can use the scheme to estimate σ

$$d\mathcal{F}(\tilde{\sigma})\delta\sigma = -\tilde{\sigma}\nabla\delta u - \delta\sigma\nabla\tilde{u} = \mathbf{J} - \mathcal{F}(\tilde{\sigma}). \quad (2.13)$$

Along with the defining PDE for δu in (2.11) we can set up a linear system of equations

$$\begin{bmatrix} \nabla \cdot ([\cdot]\nabla\tilde{u}) & \nabla \cdot (\tilde{\sigma}\nabla[\cdot]) \\ -\nabla\tilde{u} & -\tilde{\sigma}\nabla[\cdot] \end{bmatrix} \begin{bmatrix} \delta\sigma \\ \delta u \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{J} - \mathcal{F}(\tilde{\sigma}) \end{bmatrix}. \quad (2.14)$$

This system is in the form $\mathbf{Ax} = \mathbf{b}$ and can be solved by the iterative Newton approach described in [2]. Adding a Tikhonov term with $\alpha > 0$ for regularization, the final problem is to compute the least squares solution of

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\|_{L^2(\Omega)}^2 + \alpha \|\mathbf{x}\|_{L^2(\Omega)}^2. \quad (2.15)$$

Given some initial conductivity σ_0 the Newton procedure developed in [2] iteratively simulates the Poisson PDE in (2.1) for u^k , sets up and solves the system of equations in (2.14), extracts the variation $\delta\sigma^k$ and lastly updates $\sigma^{k+1} = \sigma^k + \delta\sigma$. The procedure stops when either the update step $\delta\sigma^k$ is small enough say $\|\delta\sigma^k\|_{L^2(\Omega)} < \epsilon$, or when a maximum number of iterations is reached.

3 Bayesian inverse problems

Bayesian inversion theory is another toolbox we have available when we want to solve inverse problems. Casting the problem of determining information about some unknown quantity given observed data, as a statistical inference problem allows us to use the vast theory of probability to solve the problem and give further insights about the solution. The forward problem in MREIT has already been defined in section 2.1 as the operator that maps an electrical conductivity distribution to the z -component of the magnetic field, i.e. $\mathcal{G} \circ \mathcal{F} : \sigma \mapsto B_3$. The inverse problem in MREIT is then to determine the electrical conductivity distribution σ inside a domain Ω from noisy observations of the z -component of the magnetic field B_3 .

$$B_3^{\text{noise}} = y_{\text{obs}} = (\mathcal{G} \circ \mathcal{F})(\sigma) + e, \quad (3.1)$$

where e is some noise realization. Given noisy observations y_{obs} a deterministic solution would give us a single estimate of σ which we can visualize in some way. Inverse problems are often hard to solve since they might suffer from ill-posedness. Solutions to the problem may not exist uniquely and continuous dependence of the solution on the input data is not guaranteed for such problems [13]. To overcome these issues one has to regularize the problem in some way, which is often done by putting some pre-defined restrictions on the solution as in (2.7). We briefly mentioned regularization methods in section 2.2 as the deterministic approach to solving inverse problems, and how such methods yield point estimates as their solutions. If we are interested in doing uncertainty quantification then a single image does not reveal much.

The motivation for finding a way to do uncertainty quantification on the reconstructed conductivity images is the problem of inferring the standard deviation of each pixel in a conductivity image. For medical use, knowing the standard deviation of each pixel could aid doctors in performing more accurate diagnostics and giving better treatment. Say for instance that some unknown inclusion has been detected in a tomographic image of the brain. How certain can we be that this is malignant tissue, benign tissue, or just a reconstruction artifact from the algorithm? To try and answer these questions we use Bayesian inversion theory where one models all quantities of interest as random variables [6]. This lets us encode our degree of information into the underlying probability distributions, thus making uncertainty quantification possible by estimating standard deviations and credibility intervals. Our quantities of interest are the conductivity distribution σ , observed data y_{obs} , and the noise realization e . We will in the following use capital letters for denoting these quantities of interest modeled as random variables. We use the forward model $\mathcal{G} \circ \mathcal{F}$ to relate the random quantities

$$Y = (\mathcal{G} \circ \mathcal{F})(\Sigma) + E, \quad E \sim \mathcal{N}(0, \gamma^2), \quad (3.2)$$

where Σ is the electrical conductivity distribution σ modeled as a random variable, E is the random noise and Y is the noisy B_3 data. We choose to model the additive noise as a Gaussian distribution with zero mean and variance γ^2 . A paper treating the Bayesian formulation of the EIT problem also applied normally distributed noise [14]. The inverse problem is then to determine information about Σ given a realization of some noisy data $Y = y_{\text{obs}}$. Statistically, this quantity can be interpreted as a conditional probability distribution $\pi(\sigma|y_{\text{obs}})$. We call this the *posterior distribution*, and it is essentially the solution to our inverse problem in a statistical sense. This probability distribution is what one uses to infer information about the unknown conductivity distribution Σ . This is in stark contrast to deterministic methods which only produce point estimates as solutions. Such methods use a regularization term to enforce prior knowledge of the problem and make it more well-posed. In the Bayesian framework, all information

about Σ before obtaining data Y can be summarized in a *prior distribution* $\pi_{\text{pr}}(\sigma)$. The forward model and the noise realization that relates Σ and Y is the *likelihood function* $\pi_{\text{like}}(y_{\text{obs}}|\sigma)$, and this acts as a data-fitting term. We determine the posterior density via Bayes' theorem [15]

$$\pi_{\text{post}}(\sigma|y_{\text{obs}}) = \frac{\pi_{\text{like}}(y_{\text{obs}}|\sigma)\pi_{\text{pr}}(\sigma)}{\int \pi_{\text{like}}(y_{\text{obs}}|\sigma)\pi_{\text{pr}}(\sigma) d\sigma} = \frac{\pi_{\text{like}}(y_{\text{obs}}|\sigma)\pi_{\text{pr}}(\sigma)}{\pi(y_{\text{obs}})} \propto \pi_{\text{like}}(y_{\text{obs}}|\sigma)\pi_{\text{pr}}(\sigma) \quad (3.3)$$

The distribution of the likelihood term follows that of the noise E , which means our likelihood is Gaussian since E is. The term in the denominator $\pi(y_{\text{obs}})$ is a scaling factor that makes the posterior integrate to one, as the posterior is proper density. This term is hard to evaluate as it represents a probability of a data point and to evaluate that we must know the probability distribution of the data space. This is normally not something we can compute or something that we are interested in, and it is also constant given a realization $\pi(y_{\text{obs}})$ of the data so it only acts as a scaling. As we shall see in the coming sections we need to compute the maxima of the posterior, and we know that solutions to optimization problems are invariant to the scaling of the objective function. We also need to compute relative probabilities involving fractions with the posterior in both the numerator and denominator, and scaling factors vanish in such fractions. The posterior distribution will however be intractable in most applications, so characterizing it and computing point estimates of reconstructed conductivity distributions is no trivial task.

3.1 Markov chain Monte Carlo methods

How can one characterize an intractable probability distribution without a closed-form expression? Drawing independent samples from such distributions become difficult since we only have the expression in (3.3) to define the posterior. We have access to the prior and the likelihood function in the numerator. However, the term in the denominator is often complicated to evaluate as we have to integrate over all parameters σ . The common way to go about sampling from a posterior density you only know up to a scaling constant is to use Markov chain Monte Carlo (MCMC) methods [16, 6]. In this framework, one produces dependent samples (x_1, x_2, \dots, x_N) from a Markov chain, which is constructed in such a way that we only need to compute fractions of the posterior, while the samples x_i approaches the posterior distribution $\pi(x)$ in the limit. As in (3.3) we only have access to an unnormalized version of the target density $\pi_{\text{post}}(x)$. The core idea behind MCMC methods is to estimate some expected value $\mu = \mathbb{E}[f]_\pi = \int f(x)\pi(x) dx$ by the ergodic average [16]

$$\hat{\mu} \approx \frac{1}{N} \sum_{i=1}^n f(x_i). \quad (3.4)$$

The function f can be any quantity that we wish to estimate using MCMC methods, e.g. the mean, variance, or some other function of the probability distribution. If the samples x_i are of large dimension, then evaluating integrals becomes challenging due to the curse of dimensionality. The caveat of using MCMC lies in the fact that the distribution of x_i only approaches the target distribution π_{post} and is not equal to it. Furthermore, the drawn samples $\{x_i\}_{i=1}^N$ are generally not statistically independent since they are links in a Markov chain. These two flaws essentially go directly against what we mean by independent and identically distributed (i.d.d.) samples, which are needed for computing a good ergodic average.

In order to derive an MCMC method that is guaranteed to reach its target density provided enough samples are drawn, certain conditions must apply to the Markov chains that we sample from. Recall that Markov chains do not remember the past. This means that the distribution of every new random sample X_{i+1} depends only on the immediate predecessor X_i . This can be formalized as the Markov property.

$$\mathbb{P}(X_{i+1}|X_j = x_j, 0 \leq j \leq i) = \mathbb{P}(X_{i+1}|X_i = x_i). \quad (3.5)$$

A Markov chain is then simply a sequence of random variables $\{X_i\}_{i=1}^N$ which satisfy this Markov property. The distribution of each sample in a Markov chain is generally different since the chain evolves over time, however, this does not have to be the case. When the distribution of adjacent states does not vary in time the Markov chain is said to be *time-homogenous*.

$$\mathbb{P}(X_{i+1} = y | X_i = x) = \mathbb{P}(X_1 = y | X_0 = x), \quad (3.6)$$

As such the probability of going from a state x to a state y does not change no matter where we are in the chain. Under certain conditions and if we run the chain for long enough, the chain might reach what is called the *stationary distribution* after some time t . In this situation, all links of the chain X_i for $i > t$ are distributed identically according to some $\pi(x)$ and we have converged to the target distribution in a sense. We want to construct a Markov chain that reaches stationarity after some time, but how exactly does one do that? Let us first introduce some tools from probability theory, namely the concept of a *probability transition kernel*. Let $\mathcal{B} = \mathcal{B}(\mathbb{R}^n)$ denote the collection of all open sets on \mathbb{R}^n . Following the notation in [6], we say that a mapping $P : \mathbb{R}^n \times \mathcal{B} \rightarrow [0, 1]$ is called a probability transition kernel if

1. for each set $B \in \mathcal{B}$, the mapping $\mathbb{R}^n \rightarrow [0, 1]$, $x \mapsto P(x, B)$ is a continuous function;
2. for each point $x \in \mathbb{R}^n$, the mapping $\mathcal{B} \rightarrow [0, 1]$, $B \mapsto P(x, B)$ is a probability distribution.

We say that an ordered sequence of random variables $\{X_i\}_{i=1}^N$ is a *discrete time stochastic process*, and if the random variables $X_i \in \mathbb{R}^n$ satisfy the Markov property in (3.5), then the stochastic process with transition kernel P is called a Markov chain. In the following, let π be the probability distribution of $X_j \in B_j$. We can define a transition kernel P which propagates the chain k steps forward in time from a state x_j .

$$\begin{aligned} P^{(k)}(x_j, B_{j+k}) &= \pi_{X_{j+k}}(B_{j+k} | x_j) \\ &= \int_{\mathbb{R}^n} P(x_{j+k-1}, B_{j+k}) P^{(k-1)}(x_j, B_{j+k-1}) dx_{j+k-1}, \quad k \geq 2, \end{aligned} \quad (3.7)$$

where we define the first instance $P^{(1)}(x_j, B_{j+1}) := P(x_j, B_{j+1})$. The transition kernel is the mapping that propagates the probability distribution of the state X_i to the next link in the Markov chain X_{i+1} , i.e.

$$\pi_{X_{j+1}}(B_{j+1} | x_j) = \int_{\mathbb{R}^n} P(x_j, B_{j+1}) \pi_{X_j}(x_j) dx_j, \quad (3.8)$$

therefore, the transition probability mapping is for computing $\pi_{X_{j+1}} = \pi_{X_j} P$, i.e. we step into the future by multiplying with the transition kernel P on the right. Having taken enough steps into the future with the transition kernel it might be that applying the kernel does not change the distribution, i.e.

$$\pi P = \pi. \quad (3.9)$$

Then the distribution π is said to be *invariant* or *stationary* w.r.t. the transition kernel P , and the above condition is known as a *stationarity condition*. We say that a Markov chain generated by the transition kernel P is *irreducible* if, for any open set B and any initial point x there is a positive probability that the Markov chain visits B . In transition kernel terms it means that for each point $x \in \mathbb{R}^n$ and each $B \in \mathcal{B}$ there exists an integer k such that $P^{(k)}(x, B) > 0$.

It can be the case that some state in the Markov chain X_j is *recurrent* which means that the Markov chain is guaranteed to return to that state at some time with probability 1. Otherwise, the state is called *transient*. Such irreducible kernels can also be *periodic* in the sense that they can produce a Markov chain that is stuck in a loop in state space. In terms of the transition kernel, it means that there is a set of subsets B_1, \dots, B_d of \mathbb{R}^n , such that for all $x \in B_j$ we have that $P(x, B_{j+1 \text{ mod } d}) = 1$. This means that the probability of going from B_j to B_{j+1} is one,

and this cycle will repeat itself after d steps. A transition kernel P which does not generate a periodic Markov chain is simply said to be *aperiodic*.

If all the mentioned conditions apply to a transition kernel P then proposition 3.11 in [6] guarantees the following for all $x \in \mathbb{R}^n$

Proposition 1 *If a transition kernel P generates time-homogenous Markov chains, if it is irreducible and aperiodic, and if π is a stationary distribution w.r.t. P , then*

$$\lim_{N \rightarrow \infty} P^{(N)}(x, B) = \pi(B), \quad \forall B \in \mathcal{B}, \quad (3.10)$$

i.e. the transition kernel reaches the target distribution in the limit as we run the chain for infinitely long.

This result is crucial for estimating the integral of a function f by the ergodic average in (3.4)

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{j=1}^N f(X_j) = \int_{\mathbb{R}^n} f(x) \pi(x) \, dx. \quad (3.11)$$

For the ergodic average to converge properly the samples $\{X_j\}$ must be independent and identically distributed. The proposition above ensures that the samples do reach the target distribution, and the samples are thus identically distributed if we sample for long enough. Independence between the drawn samples of the Markov chain is another issue that we will discuss later. All we need to do is to construct an invariant, irreducible, and aperiodic transition kernel P and generate a Markov chain using that kernel.

3.1.1 Metropolis-Hastings

Constructing a transition kernel $P(x, B)$ such that it is invariant w.r.t. the density $\pi(x)$ requires that the *detailed balance condition* on the kernel is met [6]. In the following, we will derive the Metropolis-Hastings algorithm which is a standard way of drawing samples from a difficult probability distribution when direct sampling from it is impractical. If the Markov chain is at some point $x \in \mathbb{R}^n$ it can either move to a new point $y \in \mathbb{R}^n$ or stay where it is at x . With this thinking we can split the kernel into two terms, one representing the acceptance of a new sample $X_{j+1} = y$ and one which represents staying at the current step $X_{j+1} = X_j$

$$P(x, B) = \int_B K(x, y) \, dy + r(x) \chi_B(x). \quad (3.12)$$

The first term $K(x, y) \, dy$ can be thought of as the probability of moving from the point $x \in B$ to the infinitesimal set dy at the point y . The second term $r(x) \chi_B(x)$ is the probability of remaining at the point x . If we are not at the point $x \in B$, the only way to reach x is by moving to the point x . The characteristic function $\chi_B(x)$ is equal to one if $x \in B$ and zero if not. It appears since the probability of remaining at x , if we are not at x in the first place, is zero. If we consider the transition kernel for all possible sets $B \in \mathbb{R}^n$ then it is equal to 1, i.e. $P(x, \mathbb{R}^n) = 1$ since \mathbb{R}^n is the whole probability space. The probability of x being in some set in \mathbb{R}^n is naturally 1 since the domain of x is \mathbb{R}^n . Thus we can write

$$r(x) = 1 - \int_{\mathbb{R}^n} K(x, y) \, dy. \quad (3.13)$$

The invariance condition on the distribution π and the transition kernel P means that the following must hold for all $B \in \mathcal{B}$

$$\begin{aligned}\pi P(B) &= \int_{\mathbb{R}^n} \left(\int_B K(x, y) \, dy + r(x)\chi_B(x). \right) \pi(x) \, dx \\ &= \int_B \left(\int_{\mathbb{R}^n} \pi(x)K(x, y) \, dx + r(y)\pi(y) \right) \, dy \\ &= \int_B \pi(y) \, dy.\end{aligned}\tag{3.14}$$

The rewriting in the middle is a consequence of (3.13). The above results further imply that

$$\pi(y)(1 - r(y)) = \int_{\mathbb{R}^n} \pi(x)K(x, y) \, dx.\tag{3.15}$$

Using (3.13) to rewrite the above we arrive at the *balance equation*

$$\int_{\mathbb{R}^n} \pi(y)K(y, x) \, dx = \int_{\mathbb{R}^n} \pi(x)K(x, y) \, dx.\tag{3.16}$$

A stronger condition called the *detailed balance condition* can also apply to K . This condition allows K to automatically satisfy (3.16) if it holds for all $x, y \in \mathbb{R}^n$

$$\pi(y)K(y, x) = \pi(x)K(x, y).\tag{3.17}$$

This condition also ensures that the Markov chain is *reversible* in the sense that the order of the chain does not matter. The probability flow is the same whether we go from $x \rightarrow y$ or $y \rightarrow x$. The goal of the Metropolis-Hastings algorithm is to construct a transition kernel K which satisfies the detailed balance condition (3.17). The idea is to define a function $q(x, y) : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}_+$ called the *proposal distribution*, which we can readily sample from and which integrates to 1 in the second variable, i.e. it is a proper probability density in y given some x

$$\int_{\mathbb{R}^n} q(y|x) \, dy = 1.\tag{3.18}$$

The function q will then define a transition kernel

$$Q(x, A) = \int_A q(y|x) \, dy.\tag{3.19}$$

If the proposal distribution q satisfies the detailed balance condition in (3.17) then we are already done and can simply set $K(x, y) = q(x, y)$ and $r(x) = 0$ in (3.12) to get a well-defined transition probability kernel. If q does not satisfy detailed balance, it instead results in the following inequality for some $x, y \in \mathbb{R}^n$

$$\pi(y)q(x|y) < \pi(x)q(y|x),\tag{3.20}$$

then we modify the kernel by a multiplicative factor α

$$K(x, y) = \alpha(x, y)q(x|y),\tag{3.21}$$

such that we instead satisfy an augmented detailed balance condition

$$\pi(y)\alpha(y, x)q(y|x) = \pi(x)\alpha(x, y)q(x|y).\tag{3.22}$$

This is achieved for all $x, y \in \mathbb{R}^n$ if we define the Metropolis-Hastings kernel $K(x, y) = \alpha(x, y)q(x|y)$ with the following correction term α

$$\alpha(x, y) = \min \left(1, \frac{\pi(y)q(y|x)}{\pi(x)q(x|y)} \right).\tag{3.23}$$

This term is often called the *acceptance ratio* in the literature. A procedure implementing the Metropolis-Hastings method only requires a few steps which we iterate over for each sample we want in the Markov chain.

1. Choose initial value for the Markov chain $x_1 \in \mathbb{R}^n$ and set counter $k = 1$.
2. Draw a candidate $y \in \mathbb{R}^n$ from the proposal distribution $q(y|x_k)$ and compute the acceptance ratio
$$\alpha(x_k, y) = \min \left(1, \frac{\pi(y)q(y|x_k)}{\pi(x_k)q(x_k|y)} \right). \quad (3.24)$$
3. Draw $t \in [0, 1]$ from a uniform density as the probability of accepting the new sample
4. If $t \leq \alpha(x_k, y)$ we update the next sample as $x_{k+1} = y$, else we do not update and set $x_{k+1} = x_k$. Increment $k \leftarrow k + 1$ and go to step 2. Stop the procedure when $k = N$ and the desired number of samples has been drawn.

The method is especially simple to implement if the proposal distribution q is symmetric $q(x|y) = q(y|x)$. Then the acceptance ratio reduces to

$$\alpha(x, y) = \min \left(1, \frac{\pi(y)}{\pi(x)} \right). \quad (3.25)$$

Instead of directly sampling a new value $y \in \mathbb{R}^n$ we can make a *random walk* model. This means that we update the previous step with some random increment we sample from the proposal distribution q , often just white noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$

$$x_{k+1} = \begin{cases} x_k + \epsilon & t \leq \alpha, \\ x_k & t > \alpha. \end{cases} \quad (3.26)$$

This ensures that the proposal distribution is always symmetric since $q(x, y) = g(x - y)$ where g is the Gaussian density. This method produces a Markov chain of dependent samples which can be subsampled further to produce a set of nearly independent samples. Details on why and how to do this will be covered in later sections. First though we will take a look at another common type of sampler called the Gibbs sampler.

3.1.2 Gibbs sampling

If we drop the idea of choosing a proposal distribution q and want to work with the target density π directly, one can instead turn to Gibbs sampling [16, 6]. Usually, we cannot sample $x \in \mathbb{R}^n$ from π directly, this is the sole motivation behind MCMC methods. However, sometimes it is possible to explicitly derive and sample from the conditional distribution of each component x_j , given all the other components $x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n$ of the vector x . We consider each component of the vector by itself. This corresponds to the following partition of \mathbb{R}^n

$$x = [x_1, \dots, x_n] \in \mathbb{R}^n. \quad (3.27)$$

Since we consider the vector x with one element removed it will be convenient to introduce notation for exactly this. Denote by x_{-j} the vector x with the element x_j deleted from the vector.

$$x_{-j} := [x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n] \in \mathbb{R}^{n-1}. \quad (3.28)$$

Thus, in the following when there is a minus in front of the index it denotes the vector x with the single element x_j removed. Isolating the components and sampling each individually conditioned on all the others is called a *single component Gibbs sampler*. Let X be a random variable with distribution π . We can write the probability of component i of X conditioned on all $X_j = x_j$ where $i \neq j$

$$\pi(x_i|x_{-i}) = C_i \pi(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n), \quad (3.29)$$

with C_i being a normalization constant ensuring that we integrate to 1. We can then define a transition kernel $K(x, y)$ similar to the Metropolis-Hastings method [6] for $x, y \in \mathbb{R}^n$

$$K(x, y) = \prod_{i=1}^m \pi(y_i | y_1, \dots, y_{i-1}, x_{i+1}, \dots, x_n), \quad (3.30)$$

while setting $r(x) = 0$ in (3.13). Since we sample from the full conditional target distribution it can be shown that the stationary distribution π is preserved [16]. If we set $r(x) = 0$ in (3.13) it means that the probability of remaining inert at step x_{i+1} is 0, i.e. we never "stand still" at $x_{i+1} = x_i$. Thus we see that every proposal is accepted by the Gibbs sampler which is a major difference from the Metropolis-Hastings sampler. It can be shown that this transition kernel $K(x, y)$ satisfies the sufficient condition for the balance equation in (3.16). However, it does not satisfy the stronger detailed balance condition in (3.17) [p. 101, 6]. We now prove that the transition kernel satisfies the sufficient condition in (3.16). Since we normalized the conditional probability distribution in (3.29) we know that it integrates to 1

$$\int_{\mathbb{R}} \pi(x_i | x_1, \dots, x_{i-1}, y_{i+1}, \dots, y_n) dx_i = 1, \quad \forall i = 1, \dots, n. \quad (3.31)$$

Using this and the transition kernel in (3.30) we can write [6]

$$\begin{aligned} \int_{\mathbb{R}}, K(y, x) dx_n &= \int_{\mathbb{R}} \prod_{i=1}^n \pi(x_i | x_1, \dots, x_{i-1}, y_{i+1}, \dots, y_n) dx_n \\ &= \prod_{i=1}^{n-1} \pi(x_i | x_1, \dots, x_{i-1}, y_{i+1}, \dots, y_n) \int_{\mathbb{R}} \pi(x_n | y_1, \dots, y_{n-1}) dx_n \\ &= \prod_{i=1}^{n-1} \pi(x_i | x_1, \dots, x_{i-1}, y_{i+1}, \dots, y_n). \end{aligned} \quad (3.32)$$

Doing this $n - 1$ times more we see that

$$\int_{\mathbb{R}^n} K(y, x) dx = 1. \quad (3.33)$$

Multiplying both sides of the above equation with $\pi(y)$ leads to

$$\pi(y) \int_{\mathbb{R}^n} K(y, x) dx = \int_{\mathbb{R}^n} \pi(y) K(y, x) dx = \pi(y). \quad (3.34)$$

The first term $i = 1$ in $K(x, y)$ from (3.30) is $\pi(y_1 | x_2, \dots, x_n)$. This is independent of x_1 since the transition kernel conditions on all future steps x_{i+1}, \dots, n . Thus we can use the definition of marginal probability to write

$$\int_{\mathbb{R}} \pi(x) K(x, y) dx_1 = K(x, y) \int_{\mathbb{R}} \pi(x) dx_1 = K(x, y) \pi(x_2, \dots, x_n). \quad (3.35)$$

We insert the product expansion of $K(x, y)$ in the above to get

$$\begin{aligned} \int_{\mathbb{R}} \pi(x) K(x, y) dx_1 &= \left(\prod_{i=1}^n \pi(y_i | y_1, \dots, y_{i-1}, x_{i+1}, \dots, x_n) \right) \pi(x_2, \dots, x_n) \\ &= \left(\prod_{i=2}^n \pi(y_i | y_1, \dots, y_{i-1}, x_{i+1}, \dots, x_n) \right) \times \pi(y_1 | x_2, \dots, x_n) \pi(x_2, \dots, x_n) \\ &= \left(\prod_{i=2}^n \pi(y_i | y_1, \dots, y_{i-1}, x_{i+1}, \dots, x_n) \right) \pi(y_1, x_2, \dots, x_n) \end{aligned}$$

We can then integrate the above expression concerning x_2 , then for x_3 up to and including x_n . This yields the final result

$$\int_{\mathbb{R}^n} \pi(x) K(x, y) dx = \pi(y). \quad (3.36)$$

We multiply both sides of the équation by $\int_{\mathbb{R}^n} K(y, x) dx = 1$ since it integrates to one (3.33)

$$\int_{\mathbb{R}^n} \pi(x) K(x, y) dx = \pi(y) \int_{\mathbb{R}^n} K(y, x) dx. \quad (3.37)$$

Since $\pi(y)$ is independent of x we can put it inside of the integral

$$\int_{\mathbb{R}^n} \pi(x) K(x, y) dx = \int_{\mathbb{R}^n} \pi(y) K(y, x) dx, \quad (3.38)$$

and we thus see that the balance condition in (3.16) is satisfied for the Gibbs transition kernel. We sample each component x_i from the conditional distribution $\pi(x_i|x_{-i})$ and by doing this for each component $i = 1, \dots, n$ we preserve the target distribution π . This also means that each proposal is accepted as we mentioned before, which reduces the computational overhead of the method. The conditionals might not have a closed form though so the Gibbs sampler is not applicable in all situations. Only a few steps are needed to implement the Gibbs sampler

1. Pick an initial value $x_1 \in \mathbb{R}^n$ and set counter $k = 1$.
2. Set $x = x_k$. For all components $1 \leq j \leq n$ we draw a sample $y_j \in \mathbb{R}$ from the distribution $\pi(y_j|y_1, \dots, y_{j-1}, x_{j+1}, \dots, x_n)$.
3. When all components have been sampled we update $x_{k+1} = y$, increment $k \leftarrow k + 1$ and repeat from step 2. Stop the procedure when $k = N$ and the desired number of samples has been drawn.

The sequence in which we sample the individual blocks y_j does not matter. We can sample in a fixed order from 1 to n in which case we denote the sampler by a *fixed scan Gibbs sampler*, or do it in a random order where we denote it by a *random scan Gibbs sampler*. We can show that sampling component x_j from the conditional distribution $\pi(x_j|x_{-j})$ preserves the stationary distribution π [16]. Assume that a sample x has been drawn from the target density π . Replace the j th component x_j of x with some value drawn from the conditional $z \sim \pi(x_j|x_{-j})$. We thus obtain a proposal $y = [x_1, \dots, x_{j-1}, z, x_{j+1}, \dots, x_n]$. We then have from probability theory that

$$\begin{aligned} \pi(y_{-j})\pi(y_j|y_{-j}) &= \pi(y), \\ \pi(x_{-j})\pi(z|x_{-j}) &= \pi(x). \end{aligned} \quad (3.39)$$

We stated that the Gibbs sampler always accepts the proposal which we can think of as being generated from a Metropolis-Hastings step. Since the proposal y comes from the distribution $\pi(y)$ from above we can write the acceptance ratio $\alpha(x, y)$ as

$$\alpha(x, y) = \min \left(1, \frac{\pi(y_{-j})\pi(z|x_{-j})\pi(x_j|x_{-j})}{\pi(x_{-j})\pi(x_j|x_{-j})\pi(z|x_{-j})} \right) = 1. \quad (3.40)$$

We have that y_{-j} and x_{-j} are equal since y is defined to be equal to x with only component x_j changed. Deleting component j from both vectors will make them identical, thus showing that the acceptance ratio is always equal to unity. The main problem with the Gibbs sampler lies in the fact that we have to sample directly from the full conditionals. Sometimes this is not possible and we have to find another way to sample from a difficult conditional. The Metropolis-within-Gibbs method is developed to address this issue. This is discussed in the following section.

3.1.3 Metropolis-within-Gibbs sampling

An interesting idea is to combine the Metropolis-Hastings and the Gibbs methods when we cannot directly sample the j^* th from the conditional distribution. Instead of using a regular Gibbs update for the j^* component of x , we can instead make use of a Metropolis-Hastings step to sample from the conditional distribution [16]. In each iteration k of the Gibbs sampler we will do an MH step instead of sampling directly from $\pi(x_j|x_{-j})$. This procedure takes the following form given an iteration x^k

1. Choose some proposal distribution $q(\cdot|x_{-j^*}^k)$ for component $x_{j^*}^k$ given $x_{-j^*}^k$.

2. Draw $z \sim q(z|x_{-j^*}^k)$ and set $y_j = \begin{cases} x_j^k & j \neq j^* \\ z & j = j^*. \end{cases}$

3. Compute the acceptance ratio

$$\alpha(x^k, y_j) = \min \left(1, \frac{\pi(y_j)q(z|x_{-j^*}^k)}{\pi(x^k)q(x_{j^*}^k|x_{-j^*}^k)} \right). \quad (3.41)$$

4. Draw $t \sim [0, 1]$ from a uniform density as the probability of accepting the new sample.

5. If $t \leq \alpha(x^k, y_j)$ we update the next sample as $x^{k+1} = y_j$, else we do not update and set $x^{k+1} = x^k$.

6. Finish the Gibbs iteration by sampling the rest of the components x_j directly from their full conditionals $\pi(x_j|x_{-j})$.

Note that the Metropolis-Hastings step can be applied to more than just one component of x . We only need to generate one sample in the Metropolis-Hastings step inside the Gibbs sampler. Although one step guarantees the convergence of the method [16], in practice, it is often required to take multiple MH steps for each Gibbs iteration. This is not required though, but it can be beneficial to perform a few extra MH iterations within the Gibbs sampler to reach some form of stationarity and reduce the correlation between the samples. We have mentioned that it can be difficult to find closed-form expressions of the conditional distributions in the Gibbs sampler. We now illustrate how the Gibbs sampler is used for the MREIT problem.

Toy problem: finding conditional distributions for a Gibbs sampler

We consider the toy problem of determining a set of parameters that describes the electrical conductivity distribution σ of a 1 cm cube in \mathbb{R}^3 . A conductive inclusion is embedded in a background medium with lesser conductivity properties. We consider the inclusion to be a ball $B(c, r)$ with a given radius $r \in \mathbb{R}$ and center coordinates $c = (c_x, c_y, c_z) \in \mathbb{R}^3$. Denote by R and C the random variables that model r and c respectively, and assume that R and C are independent. Assume that the background conductivity is 1 S/cm (siemens/centimeter) and that the inclusion has a conductivity of 2 S/cm. We thus define $\sigma(c, r)$ as

$$\sigma(c, r) = \begin{cases} 2 \text{ S/cm} & \text{inside } B(c, r), \\ 1 \text{ S/cm} & \text{outside.} \end{cases} \quad (3.42)$$

Given some measured magnetic field data y we wish to solve the reconstruction problem in MREIT by estimating the parameters (c, r) which completely determines the shape of the electrical conductivity distribution $\sigma(c, r)$.

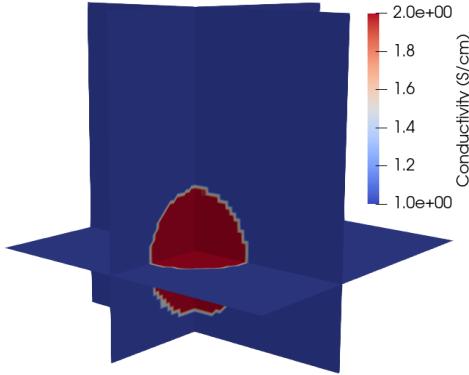


Figure 3.1: Example of a ball as a toy conductivity distribution in 3D parametrized by its radius r and center coordinates c .

Let $\pi(r, c, y)$ be the joint distribution of the parameters. We wish to sample p and c independently given y . This amounts to deriving the following conditional probability distributions: $\pi(r|c, y)$ and $\pi(c|r, y)$. The general form of Bayes' rule states that

$$\pi(x|y) = \frac{\pi(y|x)\pi(x)}{\pi(y)} = \frac{\pi(x, y)}{\pi(y)}. \quad (3.43)$$

We can thus write

$$\pi(c, r|y) = \frac{\pi(y|c, r)\pi(c, r)}{\pi(y)} = \frac{\pi(y|c, r)\pi(c)\pi(r)}{\pi(y)}, \quad (3.44)$$

since r and c are assumed to be independent. We can achieve the conditional density $\pi(c|r, y)$ by dividing both sides of (3.44) by $\pi(r)$. It then follows

$$\pi(c|r, y) = \frac{\pi(c, r|y)}{\pi(r)} = \frac{\pi(y|c, r)\pi(c)\pi(r)}{\pi(y)\pi(r)} = \frac{\pi(y|c, r)\pi(c)}{\pi(y)}.$$

Since y is given as data we know that $\pi(y)$ is a constant and we can thus write

$$\pi(c|r, y) \propto \pi(y|c, r)\pi(c). \quad (3.45)$$

This is the distribution of c conditioned on the radius r and the data y . Similarly, we obtain the conditional density $\pi(c|r, y)$ by dividing (3.44) by $\pi(c)$

$$\pi(r|c, y) = \frac{\pi(c, r|y)}{\pi(c)} = \frac{\pi(y|c, r)\pi(c)\pi(r)}{\pi(y)\pi(c)} = \frac{\pi(y|c, r)\pi(r)}{\pi(y)},$$

which leaves us with the conditional distribution of r given the center c and magnetic field data y

$$\pi(r|c, y) \propto \pi(y|c, r)\pi(r). \quad (3.46)$$

We have thus derived the following two conditional distributions

$$\pi(r|c, y) = \pi(y|c, r)\pi(r), \quad \text{and} \quad \pi(c|r, y) = \pi(y|c, r)\pi(c). \quad (3.47)$$

The two conditional distributions above are the posterior distributions that solve the inverse problem in (3.3). The terms $\pi(r)$ and $\pi(c)$ are the priors and the term $\pi(y|c, r)$ is the likelihood function, which we assume to be Gaussian. We cannot sample from the conditionals directly in the Gibbs sampler, but this is exactly why we have introduced the Metropolis-within-Gibbs step. Since the conditionals in the above are simply the likelihood $\pi(y|c, r)$ multiplied by a uniform

distribution $\pi(r)$ or $\pi(c)$, we have expressions that are proportional to the posterior in (3.3). Thus we can readily use MH updates to sample r and c from the target posterior conditionals in a Metropolis-within-Gibbs sampler for the toy problem.

But how do we make sure that the samples from the Markov chain that we draw using the methods are useful? Remember that the ergodic averages in (3.4) require independent and identically distributed (i.d.d.) samples to converge. Since the links in a Markov chain $\{x_i\}_{i=1}^N$ are inherently dependent one might wonder how we can produce a sequence of i.i.d. samples from this framework. Several techniques exist to analyze the correlation between samples such that we can determine how to subsample the chain to produce a sequence of close-to-independent posterior samples.

3.2 Chain diagnostics

When we have produced a set of Markov chain samples $\{x_i\}_{i=1}^N$ from some sampling method we need to make sure that the samples are statistically valid for estimating averages. Generally, we have two problems to address when we have produced samples x_i

1. How can we decide whether the distribution of the samples x_i has converged to the stationary distribution π ,
2. How can we decide that the samples x_i are fully exploring the stationary distribution π .

To illustrate the above points it will be useful to look at an example. Say we use the Metropolis-Hastings method to sample from the two-dimensional distribution [6]

$$\pi(x) \propto \exp \left(-10(x_1^2 - x_2)^2 - \left(x_2 - \frac{1}{4} \right)^4 \right). \quad (3.48)$$

To the knowledge of the author, there is no closed form for this distribution. We will sample from the distribution using the random walk Metropolis-Hastings method starting from the point $x_0 = [1.5, -0.8]$. To move forward we increment each step $x_{k+1} = x_k + w$, where we choose w as following a Gaussian distribution with zero mean and variance β^2 . We will then vary β to see the effect it has on the sampling. Since we use a random walk model we know that the proposal distribution q is symmetric and thus we only have to evaluate $\pi(x)$ when we compute the acceptance ratio $\alpha(x, y) = \min \left(1, \frac{\pi(y)}{\pi(x)} \right)$ with $\pi(x)$ in (3.48). Doing $N = 5000$ MH iterations for values of $\beta \in \{0.01, 0.05, 0.1, 1\}$ yields the results in figure 3.2 and 3.3. The standard deviation β of the random walk step w can be thought of as a step size in the random walk. For small step sizes $\beta = 0.01$ we see from figure 3.2 that the samples are slow at exploring the target density and for $N = 5000$ iterations we only get to explore a small region of the target distribution. The sampled points are very densely packed and we have a lot of samples, which indicates that a lot of the proposed steps are accepted. This can be confirmed by computing the proportion of accepted samples out of the $K = 5000$ samples, which for $\beta = 0.01$ is equal to 95.42%. When the step length β increases we see that we get to explore the target density more, which is a better result. However, this also means that more of the proposed samples are rejected. With a larger step size, we are more likely to step outside of the support of the density. For $\beta = 0.05$ we get an acceptance ratio of 88.64%. For $\beta = 0.1$ we get 79.24% and lastly for $\beta = 1$ we only get 16.8%. In figure 3.2 we see that there are many more samples (black dots) for $\beta = 0.1$ than for $\beta = 1$. This implies that the acceptance ratio has become large since the step size has caused many samples to fall outside the support of π .

3.2.1 Burn-in and thinning

Continuing with the example target distribution in (3.48) we will now introduce methods of producing close-to-independent samples from a set of Markov chain samples. The contour plots

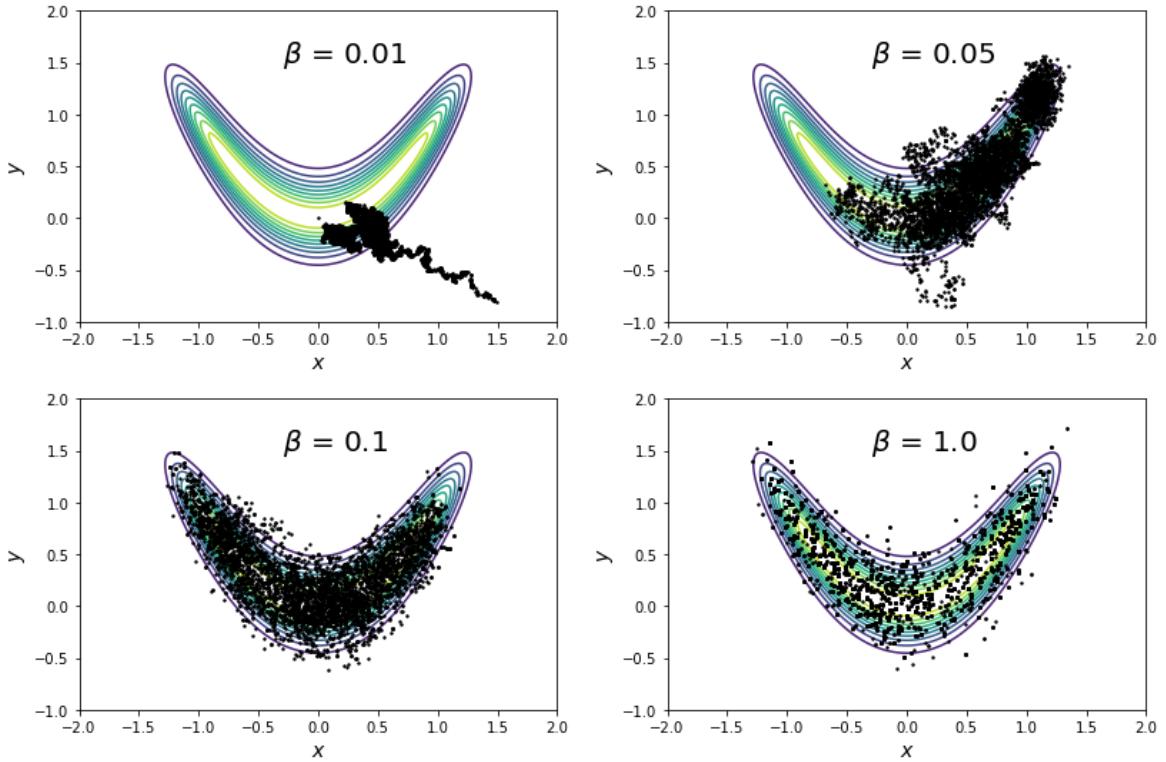


Figure 3.2: Results of the Metropolis-Hastings random walk for different values of β for $N = 5000$ starting from the lower right, on top of a contour plot of $\pi(x)$ from (3.48).

in figure 3.2 reveal how the Metropolis-Hastings method likes to move from regions of small probability to regions of higher probability inside the contours of the density. We initialize the method in a region of very small probability at x_0 . If a new proposal x^* has a higher probability than the current state x_k , then $\pi(x^*)/\pi(x_k)$ is larger than 1, and hence the acceptance ratio $\alpha = 1$ and we are sure to accept the proposal. The samples at the beginning of the chains do not represent the target distribution $\pi(x)$. It is usually a good idea to discard these "warm-up" samples and only use the part of the chain which represents the target distribution better. The process of discarding the first N_b samples is called the *burn-in* in the literature. Plotting individual components of every sample of x as the samples evolve is called a *trace plot*. Examples of trace plots for the x_1 component of all samples in figure 3.2 are shown in figure 3.3. We see a need to apply burn-in since the chains only settle in after a while, especially for small values of β . This is because the Markov chain takes longer to reach the asymptotic behavior. From the trace plots, we can also get an indication of whether the samples x_k have explored the target distribution well. When the samples are near-independent we should not see a lot of correlation between subsequent samples in the traces. If points close to each other share similar values then we know there is a lot of correlation in the samples. We should ideally see a trace plot that looks like random noise as in the bottom right of 3.3. Here the values of adjacent samples of x_1 are very different from each other. This implies that there is not much correlation going on between the samples and we have near-independent samples. With respect to the traces in figure 3.3 we can say that smaller step sizes $\beta \in \{0.01, 0.05, 0.1\}$ lead to a large correlation. We can only say that the correlation length is small for a large step size $\beta = 1$. An eye-ball estimate for a good trace plot is often referred to as a "fuzzy caterpillar". If this is the case then the step size β in the random walk can be said to be appropriate. We refer to the process of the samples fluctuating randomly as *mixing*. The better the mixing the more the trace plots look like fuzzy caterpillars. This is only a rough estimate of the behavior. To ensure the quality of

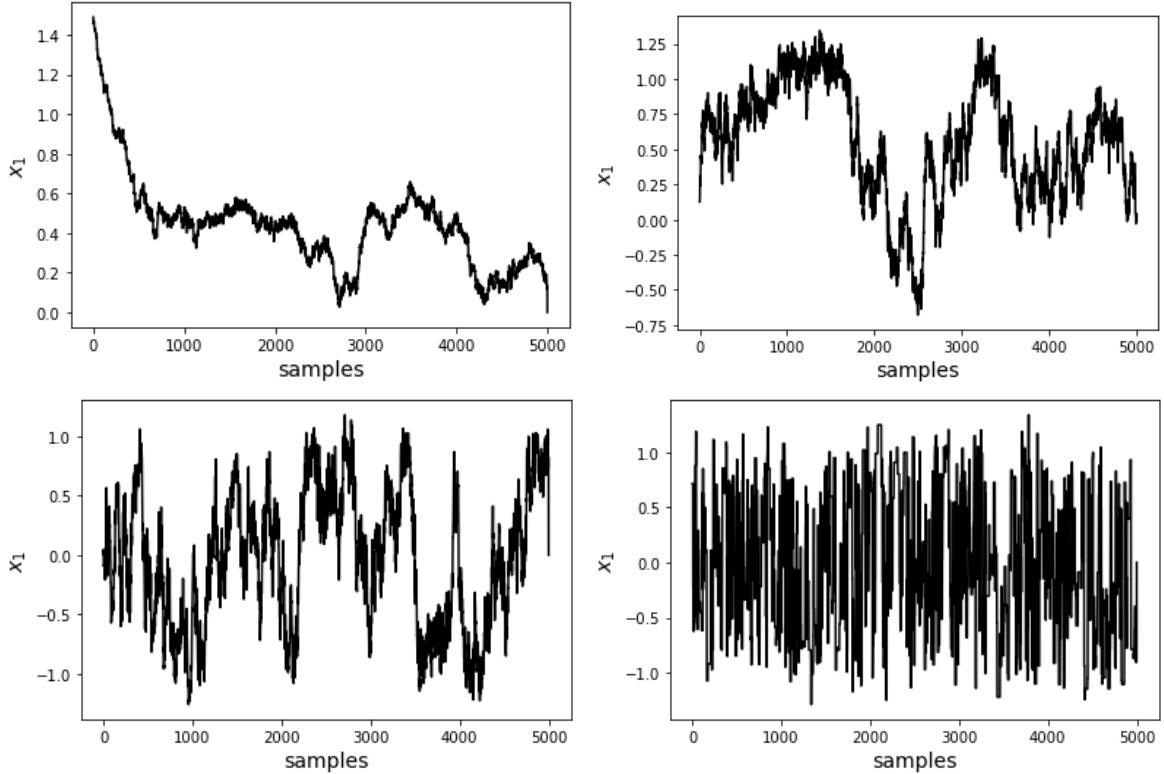


Figure 3.3: Trace plot of the x_1 component of x from the Metropolis-Hastings random walk for different values of β for $N = 5000$.

the samples, one thing we can however do is to only consider every j th sample. This is a process known as *thinning* or *subsampling* the chain. This allows us to substantially reduce the amount of correlation in the final chain allowing the ergodic averages in (3.4) to converge faster. However, it requires us to compute $j \cdot N$ times as many samples in the first place if we desire a final sample size of N .

3.2.2 Effective sample size

We described before how the correlation between consecutive values of an MCMC chain makes the ergodic averages in (3.4) converge slowly. The correlation between samples can be quantified by the *sample auto correlation* function which describes the correlation between samples x_i at lag ℓ for $0 \leq \ell < N$. For a realization of a Markov random field, this can be estimated as

$$\beta_\ell = \frac{1}{N} \sum_{i=1}^{N-\ell+1} (x_i - \bar{x})(x_{i+\ell} - \bar{x}), \quad \text{where } \bar{x} = \sum_{i=1}^N x_i. \quad (3.49)$$

The normalized sample autocorrelation of x_i at lag ℓ is then

$$\rho_\ell = \frac{\beta_\ell}{\beta_0}. \quad (3.50)$$

Continuing with the example in (3.48), we have computed the autocorrelation function ρ_ℓ of both components of x for the Metropolis-Hastings example in (3.48), results are displayed in figure 3.4. We see that there is a large amount of correlation between samples even at lags $\ell = 100$ for $\beta \in \{0.01, 0.05, 0.1\}$, while the autocorrelation function decays rapidly for $\beta = 1$. This shows that the mixing is better for larger step sizes since there is less correlation between consecutive samples in the chain. We want the correlation to be close to zero. The autocorrelation function

for $\beta = 1$ does decay to approximately zero at about lag $\ell = 50$. This indicates that every 50th sample provides an independent sample from the others. Thus it is beneficial to thin the samples and only consider 1 in every 50.

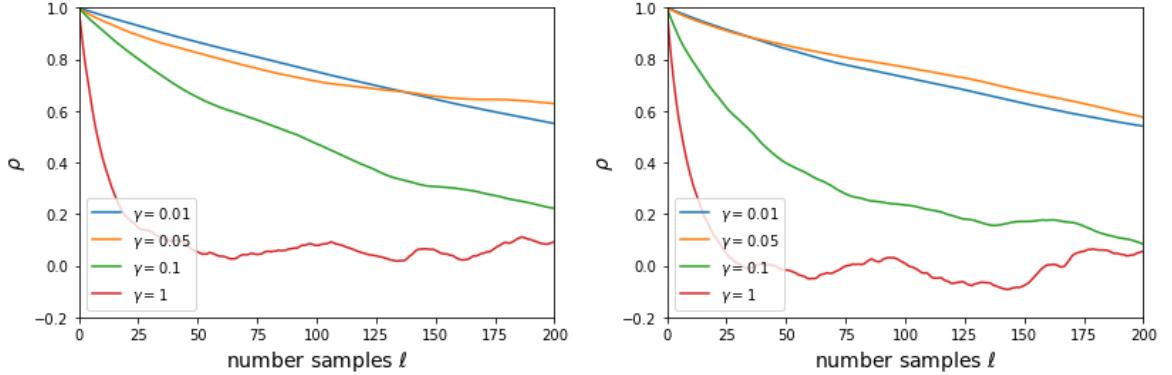


Figure 3.4: Example of autocorrelation function ρ_ℓ as function of lag ℓ for the random walk Metropolis-Hastings example in (3.48) for different values of β . To the left is the autocorrelation of the x_1 component and to the right is the x_2 component.

In this report, we estimate the mean of the posterior using ergodic averages in the form of $\hat{\mu}_N = \frac{1}{N} \sum_{i=1}^N f(x_i)$. What is the variance of the estimate? Since our samples are correlated and not independent this makes a naive estimate of the variance to be biased. Assume the chain has variance σ^2 , then the variance of the MCMC estimator $\hat{\mu}_N$ is [17]

$$\mathbb{V}[\hat{\mu}_N] = \frac{\sigma^2}{N} \tau_f, \quad (3.51)$$

where τ_f is the *integrated autocorrelation time* (IACT) for a stationary process

$$\tau_f = 1 + 2 \sum_{k=1}^{\infty} \rho_k \approx \hat{\tau}_f = 1 + 2 \sum_{k=1}^{N-1} \left(1 - \frac{k}{N}\right) \rho_k, \quad (3.52)$$

with ρ_k as in (3.50) and where $\hat{\tau}_f$ is an unbiased estimator of τ_f . The IACT can be said to be an estimator of how many iterations it takes on average to produce an independent MCMC sample [18], so we want this value to be close to 1. This can be used to compute the effective sample size N_{eff} of a set of MCMC samples of size N

$$N_{\text{eff}} = \frac{N}{1 + 2 \sum_{k=1}^{\infty} \rho_k} \approx \frac{N}{1 + 2 \sum_{k=1}^{N-1} \left(1 - \frac{k}{N}\right) \rho_k}. \quad (3.53)$$

The effective sample size can be used to compare the variance estimated via correlated MCMC samples and the ideal case with the variance computed from completely independent samples. The aim is to obtain an efficient sample size N_{eff} as close to N as possible, since this would mean that the samples are mostly independent. For the example target distribution in (3.48) we drew $N = 5000$ samples with the Metropolis-Hastings method as shown in figure 3.2. We can then compute the effective sample size for each component of x to see how well the sampling performs.

N_{eff}	$\beta = 0.01$	$\beta = 0.05$	$\beta = 0.1$	$\beta = 1$
x_1	10.94	8.33	33.33	190.50
x_2	10.77	17.49	78.44	345.42

Table 3.1: Efficient number of samples for different values of step size β for the samples we drew from the target density $\pi(x)$ in (3.48).

Another metric that can determine how well a chain is mixing is the *mean square jump* distance (MSJ). In an ideal sampling scenario, we want adjacent samples x_i and x_{i-1} to be different from each other for the sake of independence. This is analogous to there being a large "jump" between adjacent samples and the MSJ distance computes the average of the sum of squares between samples.

$$\text{MSJ} = \frac{1}{N} \sum_{i=1}^N \|x_i - x_{i-1}\|_2^2. \quad (3.54)$$

We want this MSJ distance to be as large as possible, the larger the magnitude of MSJ the better the chains are mixing. Another common convergence diagnostic is the *Geweke test* where we split the chain into two parts after discarding the burn-in samples [19]. We consider the first 10% of the chain and the last 50% of the chain. If the Markov chain has reached the stationary distribution we expect the averages of these two parts should be approximately equal. The Geweke test can be found in external Python libraries.

3.3 Posterior inference

The traditional frequentist approach to solving inverse problems uses deterministic methods and only gets a point estimate as the solution. A point estimate is nice since it provides a readily available solution to the inverse problem. Point estimates can be a single estimate of an unknown parameter to be determined, or in our case a tomographic image of a conductivity distribution. In statistical inverse problems, the solution is a probability distribution that requires post-processing to grasp. Computing point estimates from the posterior probability density are therefore very important since we use these point estimates to infer information. Traditional deterministic point estimate solutions can take the form of a regularized solution as in (2.7), or a *maximum likelihood* estimate

$$x_{\text{ML}} = \arg \max_x \pi_{\text{like}}(y_{\text{obs}} | x). \quad (3.55)$$

Recall that the likelihood function worked as a data-fitting term so by only maximizing this we are not including any prior information and therefore not any regularization in the solution. This solution is therefore prone to being very unstable, which is why we use the Bayesian framework and include prior information in the first place. A Bayesian point estimate can be computed by instead maximizing the posterior density, to produce what is called a *maximum a posteriori* (MAP) estimate. This is the highest point of the posterior and can be said to be the most credible point

$$x_{\text{MAP}} = \arg \max_x \pi_{\text{post}}(x | y_{\text{obs}}). \quad (3.56)$$

Note that instead of maximizing the posterior (or likelihood) we usually minimize the negative logarithm of the posterior for a better numerical stability. We can also compute the *posterior mean* or the *conditional mean* which is defined via an integral. Estimating this quantity is one of the central motivations for introducing MCMC methods. When we have a set of MCMC samples $\{x_i\}_{i=1}^N$ we use these to approximate the posterior mean

$$x_{\text{PM}} = \mathbb{E}_{\pi_{\text{post}}}[x] = \int_{\mathbb{R}^n} x \pi_{\text{post}}(x | y) dx \approx \frac{1}{N} \sum_{i=1}^N x_i. \quad (3.57)$$

In the same way, we can estimate the expected value of any function f w.r.t. the posterior

$$\mathbb{E}_{\pi_{\text{post}}}[f] = \int_{\mathbb{R}^n} f(x) \pi_{\text{post}}(x | y) dx \approx \frac{1}{N} \sum_{i=1}^N f(x_i). \quad (3.58)$$

This is especially useful for estimating posterior quantiles by using an indicator function

$$\mathbb{P}[x > a] = \int_a^\infty \pi_{\text{post}}(x|y) dx \approx \frac{1}{N} \sum_{i=1}^N \mathbb{1}_{x_i > a}. \quad (3.59)$$

When we have computed an estimate $\hat{\mu}$ it is interesting to know how certain we can be of it. Can we for example say with 95% certainty that the estimate is contained within some interval? This is possible by using *credibility intervals*. A credibility interval quantifies our certainty that an estimate $\hat{\mu}$ is contained within some interval $a < \hat{\mu} < b$. Credibility intervals quantify the uncertainty in our estimate $\hat{\mu}$ given the sampling we performed to compute $\hat{\mu}$. Since we can compute posterior quantiles we are able to compute credibility intervals. To do this given a set of Markov chain samples $\{x_i\}_{i=1}^N$ we first sort the elements of the set. We then determine the q -th quantile as the value $q/100$ of the way from the minimum to the maximum, interpolating between neighboring values if no exact match is found [20]. A 95% credibility interval thus amounts to computing the 2.5% and 97.5% quantiles. In this report we use the `numpy.percentile` function from the NumPy package.

We can compute the ML and MAP estimates in (3.55) and (3.56) using nonlinear optimization methods. This does not however say anything about the uncertainty of the point estimates. We can compute an estimate of the posterior mean in (3.57) by using sampling methods, e.g. MCMC. Using the samples we can compute standard deviations and credibility intervals, which is essential when we want to do uncertainty quantification for point estimates. We conclude that it is strictly necessary to explore the posterior if we want to both compute point estimators and do uncertainty quantification for an inverse problem.

4 The finite element method and spectral convolution methods

Since we aim to develop a computational framework for solving the MREIT inverse problem we need some numerical methods for computing the two forward mappings \mathcal{F} and \mathcal{G} from section 2. In this chapter, we will introduce the Finite Element Method (FEM) for numerically solving partial differential equations and a spectral method based on the Fourier transform for fast computation of the Biot-Savart integral in (2.3).

4.1 Finite element method for approximating PDE

For numerically solving the Poisson equation in (2.1) we apply the FEM framework. Here we consider PDE problems in their variational form since this allows one to introduce a finite collection of basis functions, which we will use to represent the solution to the variational problem. Consider an inhomogeneous Poisson equation with homogenous Dirichlet boundary conditions on a domain $\Omega \in \mathbb{R}^d$ where $d \in \{2, 3\}$

$$\begin{aligned} -\nabla \cdot \sigma \nabla u &= f && \text{in } \Omega, \\ u &= 0 && \text{on } \partial\Omega. \end{aligned} \quad (4.1)$$

The solution to this equation is a function $u \in H_0^1(\Omega)$, where the subscript 0 denotes vanishing boundary trace. The vanishing trace condition is due to the homogenous Dirichlet boundary condition. Remember that for a function to be in $H^1(\Omega)$, we need both the function and its first derivative to be square integrable. We look for solutions of the PDE in $H^1(\Omega)$ as we will consider the variational formulation of the problem. And for that, we need both the function u and its gradient ∇u to be L^2 functions, such that the integration by parts formula can be applied. Deriving the variational form of this sort of PDE involves taking a test function v belonging to the Sobolev space $H_0^1(\Omega)$, and multiplying both sides of the PDE with v followed by integration by parts

$$\int_{\Omega} (-\nabla \cdot \sigma \nabla u)v \, d\mathbf{x} = \int_{\Omega} \sigma \nabla u \cdot \nabla v \, d\mathbf{x} - \int_{\partial\Omega} \sigma (\nabla u \cdot \mathbf{n})v \, d\mathbf{s} = \int_{\Omega} fv \, d\mathbf{x}, \quad \forall v \in H_0^1(\Omega), \quad (4.2)$$

where \mathbf{n} is the outward unit normal to $\partial\Omega$. Since $v \in H_0^1(\Omega)$ we know that the boundary term vanishes and we are left with

$$a(u, v) := \int_{\Omega} \sigma \nabla u \cdot \nabla v \, d\mathbf{x} = \int_{\Omega} fv \, d\mathbf{x} := \langle f, v \rangle, \quad \forall v \in H_0^1(\Omega). \quad (4.3)$$

This also explains why $H_0^1(\Omega)$ is the right space to look for a solution to the PDE problem. If v does not vanish on the boundary we would have had a contribution from the boundary term in (4.2), so the condition of vanishing boundary trace is important. The condition of $u, v \in H_0^1(\Omega)$ also has to do with the convergence of the integral $\int_{\Omega} \sigma \nabla u \cdot \nabla v \, d\mathbf{x}$. Since $a(u, v) = \langle f, v \rangle$ has to hold for all $v \in H_0^1(\Omega)$, we can also choose $v = u$. Thus we have the integral $\int_{\Omega} \sigma \nabla u \cdot \nabla u \, d\mathbf{x}$, which is exactly equal to $\|\nabla u\|_{L^2(\Omega)}^2$. If we only had that $u \in L^2(\Omega)$ we would not be able to guarantee that this integral converges, and thus the integration by parts formula would not be applicable. If $u \in H_0^1(\Omega)$ we also have that $\nabla u \in L^2(\Omega)$ and everything works out fine. The relation in (4.3) completely defines the variational problem, find $u \in H_0^1(\Omega)$ such that

$$a(u, v) = \langle f, v \rangle, \quad \forall v \in H_0^1(\Omega). \quad (4.4)$$

The Lax-Milgram theorem insures that the solution to the variational problem 4.4 exists and is unique under certain conditions on the bilinear form $a(u, v)$ and the functional f

Theorem 1 (Lax-Milgram theorem [21]) Let $a(\cdot, \cdot)$ be a bilinear mapping from $[H_0^1(\Omega)]^2 \rightarrow R$ and assume there exists constants $\alpha, \beta > 0$ such that the bilinear form is bounded

$$|a(u, v)| \leq \alpha \|u\| \|v\|, \quad (u, v \in H_0^1(\Omega)), \quad (4.5)$$

and coercive

$$\beta \|u\|^2 \leq a(u, u), \quad (u \in H_0^1(\Omega)). \quad (4.6)$$

Finally, let $g : H^{-1}(\Omega) \rightarrow R$ be a bounded linear functional. Then there exists a unique element $u \in H_0^1(\Omega)$ such that

$$a(u, v) = \langle g, v \rangle \quad (4.7)$$

for all $v \in H_0^1(\Omega)$.

If these conditions are satisfied for all test functions $v \in H_0^1(\Omega)$ then we are guaranteed to have a unique solution $u \in H_0^1(\Omega)$ to the variational problem in (4.4). Now, let V denote the Sobolev space $H_0^1(\Omega)$. Since V is an infinite-dimensional space we cannot immediately use it for numerical computations, since we need a discretized domain. In the FEM framework, one works with a triangulation of the domain Ω . We use a finite set of basis functions as our finite element basis to build our solution, hence the name of the method. The support of each element spans only a few triangles. Let h denote some subdivision of Ω . Doing this lets us construct a finite-dimensional approximation V_h to the infinite-dimensional space V , such that $V_h \subset V$. We can then use the Galerkin method to project the system in (4.4) as a linear system of equations [22]. Let $\{\phi_i \in V_h : i = 1, \dots, N_h\}$ be a collection of basis functions in V_h and expand the solution u in this basis and let U_i be the real values expansion coefficients in the basis

$$u_h = \sum_i U_i \phi_i. \quad (4.8)$$

We choose the test functions ϕ_j to be the basis functions of V_h . A typical example of a finite element basis functions in both 2D and 3D is shown in figure 4.1, more specifically first order Lagrange elements. Such piecewise linear functions take the shape of "triangular hats" in 2D and "pyramids" in 3D. These basis functions are defined over multiple triangles in the triangulation. To increase the accuracy, we can use higher-order basis functions instead of linear ones

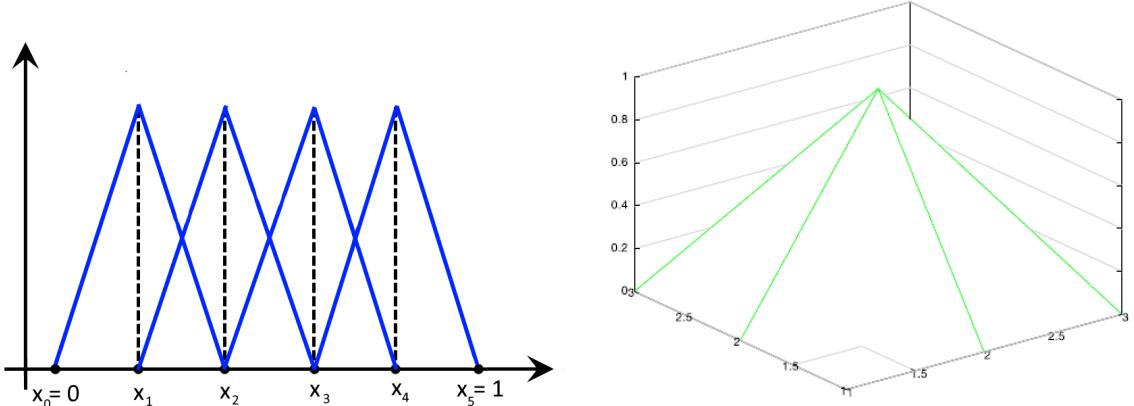


Figure 4.1: Lagrange basis elements for continuous piecewise linear functions in 2D and 3D [22].

The discrete variational problem is then to find $u_h \in V_h$ such that $a(u_h, \phi_j) = \langle f, \phi_j \rangle$ for all $\phi_j \in V_h$. Using the bilinearity of a after inserting the expansion (4.8) yields

$$a\left(\sum_i U_i \phi_i, \phi_j\right) = \sum_i U_i a(\phi_i, \phi_j) = \langle f, \phi_j \rangle, \quad \forall j = 1, \dots, N_h. \quad (4.9)$$

Now define $A_{ij} = a(\phi_i, \phi_j)$ and $F_i = \int_{\Omega} f \phi_i \, dx$ for all $i, j = 1, \dots, N_h$. Considering A_{ij} as elements of a matrix and using that $a(\phi_i, \phi_j) = a(\phi_j, \phi_i)$ for the bilinear form defined in (4.3) we can write the following

$$\begin{aligned} F_j &= \int_{\Omega} f \phi_j \, dx = a(u_h, \phi_j) = a\left(\sum_i U_i \phi_i, \phi_j\right) = \sum_i U_i a(\phi_i, \phi_j) \\ &= \sum_i A_{ij} U_i = \sum_i A_{ji} U_i = (AU)_j, \quad \forall j = 1, \dots, N_h. \end{aligned} \quad (4.10)$$

So we get the linear system

$$\mathbf{AU} = \mathbf{F}, \quad \mathbf{A} \in \mathbb{R}^{N_h \times N_h} \text{ and } \mathbf{U}, \mathbf{F} \in \mathbb{R}^{N_h}. \quad (4.11)$$

Is the solution \mathbf{U} to this linear system unique? To answer this let us again take a look at the Lax-Milgram theorem governing the uniqueness of solutions to the variational form in (4.4). We first need to ensure that the variational form in (4.4) actually satisfies the conditions of the Lax-Milgram theorem. Then we can determine the uniqueness of \mathbf{U} in the linear system above. Boundedness of the bilinear form $a(u, v)$ defined in (4.3) follows from the Cauchy-Schwarz inequality, and coercivity follows from [22, 23]. Thus we know that there exists a unique solution $u \in H_0^1(\Omega)$ to the variational problem in (4.4).

Since the coercivity condition in (4.6) holds, we can show that it ensures that the vector \mathbf{U} exists uniquely. From linear algebra, we know that the solution \mathbf{U} is unique if and only if the only solution to $\mathbf{F} = \mathbf{0}$ is $\mathbf{U} = \mathbf{0}$. This is indeed the case, since if $\mathbf{F} = \mathbf{0}$ it implies that the inhomogeneity f is zero. And if this holds then

$$0 = \langle f, v \rangle = a(u, v) = 0, \quad \forall v \in H_0^1(\Omega). \quad (4.12)$$

Coercivity in Lax-Milgram implies that $\beta \|u\|^2 \leq a(u, u) = 0, \quad \forall u \in H_0^1(\Omega)$. Thus the only solution is the zero solution $u = 0$ which implies that $\mathbf{U} = \mathbf{0}$. Setting up and solving the linear system in (4.11) for a given discretization h and choice of basis $\{\phi_i\}$ is the brunt of the work in most FEM implementations. After obtaining the coefficients \mathbf{U} the discretized solution u_h is computed by (4.8). We will employ the FEM implementation for PDEs supplied by the FEniCS package in Python [24].

At the beginning of the section, we assumed we were dealing with an inhomogenous Poisson equation with homogenous Dirichlet boundary datum. As a short remark, we want to explain how to remedy the opposite situation, i.e. that we instead had a homogenous Poisson equation with inhomogenous Dirichlet data

$$\begin{aligned} -\nabla \cdot \sigma \nabla u &= 0 && \text{in } \Omega, \\ u &= g && \text{on } \partial\Omega. \end{aligned} \quad (4.13)$$

We can rewrite the equation by first considering that $u = g$ on $\partial\Omega$ in the trace sense. The bilinear form in 4.4 has to hold for all $v \in H_0^1(\Omega)$. This means that there must be some function $w \in H^1(\Omega)$ with trace equal to g on $\partial\Omega$ [21]. But then w is a function on the entire Ω with

boundary trace equal to g . We can then define $\tilde{u} = u - w$, which indeed belongs to $H_0^1(\Omega)$ as $\tilde{u} = g - g = 0$ on $\partial\Omega$. This function instead solves the problem

$$\begin{aligned} -\nabla \cdot \sigma \nabla \tilde{u} &= -\nabla \cdot \sigma \nabla g && \text{in } \Omega, \\ \tilde{u} &= 0 && \text{on } \partial\Omega. \end{aligned} \quad (4.14)$$

Which is now an instance of an inhomogenous Poisson equation with homogenous Dirichlet boundary conditions as in (4.1).

4.1.1 Equivalence between 2D and 3D setting

Whether we solve a PDE in 2D or 3D does not change much in the formulation of the variational problem in (4.4). If we look at the homogenous Poisson equation with inhomogenous Dirichlet data in (4.13) and consider both the 2D and the 3D setting, we can prove the equivalence between the two solutions on the unit square and unit cube respectively, if we impose homogeneous Neumann boundary conditions to the top and bottom of the unit cube. The solution in 3D will then correspond to slices of the 2D solution. We thus have two ways of representing the same 3D geometry.

1. Consider a bounded and open 2D domain $\Omega \subset \mathbb{R}^2$ and the conductivity distribution $\sigma \in L_+^\infty(\Omega)$. Then the potential $p \in H_0^1(\Omega)$ solves the 2D problem

$$\begin{aligned} -\nabla \cdot \sigma(x, y) \nabla p(x, y) &= f(x, y) && \text{in } \Omega, \\ p(x, y) &= 0 && \text{on } \partial\Omega. \end{aligned} \quad (4.15)$$

2. Consider an extension of the 2D domain defined by $\Omega \times (0, 1)$. If we define $q \in H_0^1(\Omega \times (0, 1))$ and $\gamma \in L_+^\infty(\Omega \times (0, 1))$ as $q(x, y, z) = p(x, y)$ and $\gamma(x, y, z) = \sigma(x, y)$ for $z \in (0, 1)$, then the corresponding 3D problem is

$$\begin{aligned} -\nabla \cdot \gamma(x, y, z) \nabla q(x, y, z) &= f(x, y) && \text{in } \Omega \times (0, 1), \\ q(x, y, z) &= 0 && \text{on } \partial\Omega \forall z \in (0, 1), \\ \gamma \frac{\partial q}{\partial z} &= 0 && \text{for } z = 0 \text{ and } z = 1 \text{ in } \Omega. \end{aligned} \quad (4.16)$$

We want to show that these two problems are equivalent. Consider a test function $\varphi(x, y, z) \in C_0^\infty(\Omega \times (0, 1))$ and compute the weak formulation of the 3D problem.

$$\begin{aligned} \int_{\Omega \times (0, 1)} (-\nabla \cdot \gamma \nabla q) \varphi \, d\mathbf{x} &= \int_{\Omega \times (0, 1)} \gamma \nabla q \cdot \nabla \varphi \, d\mathbf{x} - \int_{\Omega \cap (0, 1)} (\gamma \nabla q \cdot \mathbf{n}) \varphi \, ds \\ &= \int_{\Omega \times (0, 1)} f \varphi \, d\mathbf{x}. \end{aligned} \quad (4.17)$$

Here \mathbf{n} denotes the normal vector in the z -direction, i.e. $\mathbf{n} = (0, 0, 1)$. Define now a 2D test function $\psi(x, y) \in C_0^\infty(\Omega)$ and let $\varphi(x, y, z) = \psi(x, y)\eta(z)$. The space of 2D test functions $C_0^\infty(\Omega)$ multiplied by some continuous function $\eta(z) \neq 0$ is dense in the space of the 3D test functions $C_0^\infty(\Omega \times (0, 1))$ [25]. Thus we can approximate each 3D test function φ arbitrarily well with some $\psi(x, y)\eta(z)$. We can now write

$$\begin{aligned} &\int_{\Omega \times (0, 1)} \gamma \nabla q \cdot \nabla \varphi \, d\mathbf{x} - \int_{\Omega \cap (0, 1)} (\gamma \nabla q \cdot \mathbf{n}) \varphi \, ds \\ &= \int_{\Omega \times (0, 1)} \gamma \begin{bmatrix} \partial_x q \\ \partial_y q \\ \partial_z q \end{bmatrix} \cdot \begin{bmatrix} \partial_x \varphi \\ \partial_y \varphi \\ \partial_z \varphi \end{bmatrix} \, d\mathbf{x} - \int_{\Omega \cap (0, 1)} \gamma \begin{bmatrix} \partial_x q \\ \partial_y q \\ \partial_z q \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \varphi \, d\mathbf{x} \\ &= \int_{\Omega \times (0, 1)} \gamma (\partial_x q \partial_x \varphi + \partial_y q \partial_y \varphi + \partial_z q \partial_z \varphi) \, d\mathbf{x} - \int_{\Omega \cap (0, 1)} \gamma \partial_z q \varphi \, d\mathbf{x}. \end{aligned} \quad (4.18)$$

We have from the Neumann boundary condition that $\gamma\partial_z q = 0$. Thus the $\partial_z q$ term vanishes in each integral and we are left with

$$\begin{aligned} & \int_{\Omega \times (0,1)} \gamma \nabla q \cdot \nabla \varphi \, d\mathbf{x} - \int_{\Omega \cap (0,1)} (\gamma \nabla q \cdot \mathbf{n}) \varphi \, d\mathbf{s} \\ &= \int_{\Omega \times (0,1)} \gamma (\partial_x q \partial_x \varphi + \partial_y q \partial_y \varphi) \, d\mathbf{x} = \int_{\Omega \times (0,1)} f \varphi \, d\mathbf{x}. \end{aligned} \quad (4.19)$$

Inserting $\varphi = \psi\eta$ yields

$$\begin{aligned} & \int_{\Omega \times (0,1)} \gamma \nabla q \cdot \nabla \varphi \, d\mathbf{x} - \int_{\Omega \cap (0,1)} (\gamma \nabla q \cdot \mathbf{n}) \varphi \, d\mathbf{s} \\ &= \int_{\Omega \times (0,1)} \gamma (\partial_x q \partial_x (\psi\eta) + \partial_y q \partial_y (\psi\eta)) \, d\mathbf{x} = \int_{\Omega \times (0,1)} f \psi \eta \, d\mathbf{x}. \end{aligned} \quad (4.20)$$

Since η is independent of x and y we can pull it out from the differentiations and outside the x - and y -integration. We also insert $\gamma(x, y, z) = \sigma(x, y)$ to get

$$\begin{aligned} &= \int_{\Omega \times (0,1)} \sigma \eta (\partial_x q \partial_x \psi + \partial_y q \partial_y \psi) \, d\mathbf{x} = \int_{\Omega \times (0,1)} f \psi \eta \, d\mathbf{x} \\ &= \int_0^1 \eta \left(\int_{\Omega} \sigma (\partial_x q \partial_x \psi + \partial_y q \partial_y \psi) \, dx dy \right) dz = \int_0^1 \eta \left(\int_{\Omega} f \psi \, dx dy \right) dz. \end{aligned} \quad (4.21)$$

We can now differentiate w.r.t. z and divide by η since $\eta \neq 0$

$$\int_{\Omega} \sigma (\partial_x q \partial_x \psi + \partial_y q \partial_y \psi) \, dx dy = \int_{\Omega} f \psi \, dx dy. \quad (4.22)$$

Since $q(x, y, z) = p(x, y)$ for all $(x, y) \in \Omega$ we can write

$$\int_{\Omega} \sigma (\partial_x q \partial_x \psi + \partial_y q \partial_y \psi) \, dx dy = \int_{\Omega} \sigma (\partial_x p \partial_x \psi + \partial_y q \partial_y \psi) \, dx dy = \int_{\Omega} f \psi \, dx dy. \quad (4.23)$$

Which is the same as writing

$$\int_{\Omega} \sigma \nabla p \cdot \nabla \psi \, dx dy = \int_{\Omega} f \psi \, dx dy. \quad (4.24)$$

Using integrations by parts on the left-hand side we get

$$-\int_{\Omega} (\nabla \cdot \sigma \nabla p) \psi \, dx dy + \int_{\partial\Omega} (\sigma \nabla p \cdot \mathbf{n}) \psi \, ds = \int_{\Omega} f \psi \, dx dy. \quad (4.25)$$

And since $\psi \in C_0^\infty(\Omega)$ we know that ψ vanishes on $\partial\Omega$ so the line integral over $\partial\Omega$ vanishes too. Thus we can write

$$\int_{\Omega} (-\nabla \cdot \sigma \nabla p) \psi \, dx dy = \int_{\Omega} f \psi \, dx dy. \quad (4.26)$$

Differentiating w.r.t. x and y and then dividing by ψ recovers the strong form of the 2D PDE from 4.15, which is what we wanted to show. Thus the 2D and 3D formulations of the Poisson problem with appropriate Dirichlet and Neumann conditions are equivalent. On a discretized $m \times m$ grid we can solve the 2D problem in (4.15) and stack m copies on top of each other to generate the z -discretization in $(0, 1)$ as a "pseudo-3D" solution. The result of this will be equivalent to solving the 3D problem with Neumann boundary conditions in (4.16) on an $m \times m \times m$ grid. Solving a 2D problem in FEM is much faster than solving 3D problems for larger mesh sizes. Therefore it will be beneficial to consider both formulations of the problem for the sake of resolution and computation time.

4.2 Spectral method for fast evaluation of magnetic fields

The presence of a current field \mathbf{J} inside Ω induces a magnetic field \mathbf{B} . The two are related via the Biot-Savart integral

$$\mathbf{B}(\mathbf{x}) = \frac{1}{4\pi} \int_{\Omega} \mathbf{J}(\mathbf{x}') \times \frac{\mathbf{x} - \mathbf{x}'}{|\mathbf{x} - \mathbf{x}'|^3} d\mathbf{x}'. \quad (4.27)$$

Since the forward problem in MREIT requires the evaluation of magnetic fields in large parts of the domain Ω we ought to have a fast and efficient method for evaluating the Biot-Savart integral. In (4.27) we are integrating over a vector in \mathbb{R}^3 , so we can split it up into three Cartesian coordinate functions $\mathbf{B} = (B_1, B_2, B_3)$ using convolutions, which we denote by the $*$ operator [11]

$$\begin{aligned} B_1(\mathbf{x}) &= \frac{1}{4\pi} \int_{\Omega} \frac{(x_3 - x'_3) J_2(\mathbf{x}') - (x_2 - x'_2) J_3(\mathbf{x}')}{|\mathbf{x} - \mathbf{x}'|^3} d\mathbf{x}' \\ &= (G_3 * J_2 - G_2 * J_3)(\mathbf{x}), \\ B_2(\mathbf{x}) &= \frac{1}{4\pi} \int_{\Omega} \frac{(x_1 - x'_1) J_3(\mathbf{x}') - (x_3 - x'_3) J_1(\mathbf{x}')}{|\mathbf{x} - \mathbf{x}'|^3} d\mathbf{x}' \\ &= (G_1 * J_3 - G_3 * J_1)(\mathbf{x}), \\ B_3(\mathbf{x}) &= \frac{1}{4\pi} \int_{\Omega} \frac{(x_2 - x'_2) J_1(\mathbf{x}') - (x_1 - x'_1) J_2(\mathbf{x}')}{|\mathbf{x} - \mathbf{x}'|^3} d\mathbf{x}' \\ &= (G_2 * J_1 - G_1 * J_2)(\mathbf{x}), \end{aligned} \quad (4.28)$$

where the Biot-Savart integral kernels G_n are defined by

$$G_n = \frac{1}{4\pi} \frac{x_n}{|\mathbf{x}|^3}, \quad n = 1, 2, 3. \quad (4.29)$$

Each convolution in (4.28) can be defined by

$$C_{n,m}(\mathbf{x}) = G_n(\mathbf{x}) * J_m(\mathbf{x}), \quad n \neq m. \quad (4.30)$$

Computing the convolution with a naive implementation on a grid with N sampling points has the complexity $O(N^2)$. In the frequency domain, however, the convolution becomes a simple multiplication that only has linear complexity $O(N)$. Denote by F the Fourier transform and by F^{-1} the inverse Fourier transform. Then

$$C_{n,m}(\mathbf{x}) = F^{-1}\{F\{G_n\} \cdot F\{J_m\}\}. \quad (4.31)$$

To compute this we need the Fourier transform of the Biot-Savart kernel $F\{G_n\}$ and the Fourier transform of each component of the current density field $F\{J_m\}$. Only the Fourier transform of the Biot-Savart kernel is available in analytical form, so we use that for better accuracy. We will discuss shortly how we numerically compute the Fourier transform of the current density field $F\{J_m\}$. Let $\mathbf{k} = (k_1, k_2, k_3)$ be the spatial frequency in radian per meter and i the imaginary unit. Then we have for $F\{G_n\}$

$$\begin{aligned} F\{G_n\} &= F\left\{\frac{1}{4\pi} \frac{x_n}{|\mathbf{x}|^3}\right\} = F\left\{-\frac{\partial}{\partial x_n} \frac{1}{4\pi |\mathbf{x}|}\right\} \\ &= -ik_n F\left\{\frac{1}{4\pi |\mathbf{x}|}\right\}. \end{aligned} \quad (4.32)$$

We know that $\frac{1}{4\pi |\mathbf{x}|}$ is the fundamental solution for the negative Laplacian $-\Delta u = 0$. Its Fourier transform is given by [26]

$$F\left\{\frac{1}{4\pi |\mathbf{x}|}\right\} = \frac{1}{|\mathbf{k}|^2}. \quad (4.33)$$

Which then leaves for $F\{G_n\}$

$$F\{G_n\} = -i \frac{k_n}{|\mathbf{k}|^2}. \quad (4.34)$$

For computing the Fourier transform of the discretized current density field $F\{J_m\}$ we must resort to the discrete Fourier transform (DFT) since \mathbf{J} is not available in closed form. We use the Fast Fourier Transform (FFT) from the `scipy.fftpack` library in python to evaluate the DFT of \mathbf{J} . Recall that $\mathbf{J} = \sigma \nabla u$ where u is computed numerically via FEM. Assuming the gradient is discretized on a grid of size N the time complexity for applying FFT is $O(N \log(N))$. The authors of [11] specify two problems to address for the method to work properly. One problem has to do with the regularity of the mesh we are using and the other concerns the fact that convolutions computed via the FFT are not linear convolutions.

1. We must make sure that the current density field \mathbf{J} is sampled on an equidistant and regularly spaced grid. If not then we must interpolate the values to a regular grid first. The frequencies captured by the FFT are directly related to the resolution of the grid, and therefore the resolution must be chosen carefully. The FFT is most efficient when the size of the computational grid is a power of 2, i.e. $n = 2^m$ for some $m \in \mathbb{N}$ [27].
2. When we apply the FFT to compute a convolution of the form

$$g * f = F^{-1}\{F\{g\} \cdot F\{f\}\}$$

the result is a circular convolution since we use an exponential Fourier basis. To fix this, the authors introduce a periodization of the discretized current density field \mathbf{J} . Since the current field \mathbf{J} is zero outside the domain Ω , i.e. it has compact support, we can define a cube $S_1 = (-s/2, s/2)^3$ in \mathbb{R}^3 centered at the origin as the *physical domain* which contains the entire $\Omega \subset S_1$. This cube is the domain where we evaluate the linear convolution in the end. If we consider points $\mathbf{x} \in S_1$, the integral in (4.30) only involves values of the Biot-Savart kernel G_n on a larger *computational domain* $S_2 = (-s, s)^3$ such that $S_1 \subset S_2$. This is illustrated in figure 4.2.

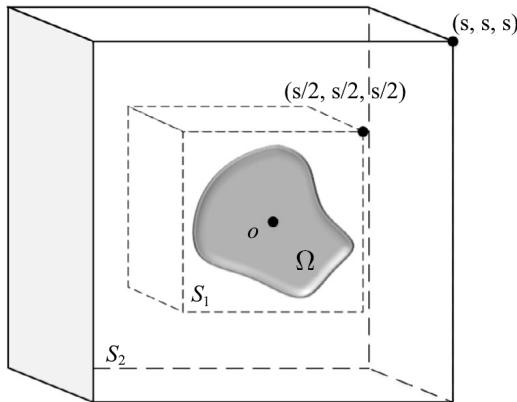


Figure 4.2: The large cube S_2 which determines the periodization of the current density field \mathbf{J} , and the small cube S_1 which shows that \mathbf{J} indeed has compact support in Ω [11].

We can then imagine the entire plane filled with multiple copies of S_2 placed in an orderly square grid, with the original instance of $\Omega \subset S_1 \subset S_2$ at the origin [28]. To compute the discrete linear convolution in the original Ω at the origin, we calculate the FFT in a large region of the plane containing several instances of S_2 where all values of $\mathbf{J} \notin \Omega$ are set to zero. We are now computing a circular convolution on a periodic domain, and

this result yields $F\{\mathbf{J}\}$ on multiple copies of S_2 in a large part of the plane. If we restrict our attention to the original S_2 cube at the origin the result will be identical to a linear convolution in S_2 . The computation of $F\{\mathbf{J}\}$ for all points in the original S_2 is then identical to the linear convolution for all points \mathbf{x} in the inclusion S_1 , and thus also identical to the linear convolution for all $\mathbf{x} \in \Omega \subset S_1$.

After accounting for the two steps above we can compute the discrete convolution in (4.31) where we use the analytical result for $F\{G_n\}$ in (4.34). We evaluate the analytical Fourier transform of the Biot-Savart kernel $F\{G_n\}$ at the spatial frequencies \mathbf{k} which are also present in the FFT computation. There is a singularity in (4.34) at $\mathbf{k} = 0$ which we would like to avoid, and we do this numerically by setting

$$F\{G_n\}[\mathbf{k} = 0] = 0. \quad (4.35)$$

Denote the analytical transform $F\{G_n\}$ evaluated at a set of discrete points by $\overline{F\{G_n\}}$. This can now be multiplied with the numerically computed $\text{FFT}(J_m)$ to produce the convolution in Fourier space. To get back to our original discretized domain Ω we must apply the inverse discrete Fourier transform, which we do via an implementation of the inverse fast Fourier transform (IFFT) from the `scipy.fftpack`. This function utilizes the FFT to compute the inverse discrete transform efficiently. The IFFT and the FFT has the property that for a discrete signal \mathbf{x} , then $\mathbf{x} = \text{IFFT}(\text{FFT}(\mathbf{x}))$. We can now write for the convolution in 4.31

$$C_{n,m}(\mathbf{x}) = \text{IFFT}(\overline{F\{G_n\}} \cdot \text{FFT}(J_m)). \quad (4.36)$$

Efficient computation of the magnetic field $\mathbf{B} = (B_1, B_2, B_3)$ can then be achieved by applying the above Fourier procedure to each $G_n * J_m$ term in (4.28). An algorithm implementing the procedure is presented in listing A.2 in Appendix 2.

5 Computational solution procedure

We model the MREIT reconstruction problem using the Bayesian framework introduced in chapter 3. Exploring and characterizing the posterior distribution which is the solution to our Bayesian inverse problem is the main problem we wish to solve in this thesis.

MCMC methods are often computationally expensive, and therefore an efficient implementation of the forward model $\mathcal{G} \circ \mathcal{F}$ in (2.4) is required for sampling to be feasible in a reasonable time. When sampling we have to evaluate the forward model each time we compute the acceptance probability α in an MCMC iteration (3.24). Constructing the posterior distribution in (3.3) requires specifying a likelihood function. The likelihood contains the forward model along with the noise model. How we parameterize the unknown electrical conductivity distribution σ determines the prior distribution.

5.1 Building the posterior

We choose to model the conductivity distribution σ as a cubic volume conductor. The conductor will act as a medium of constant conductivity, with one or more inclusions of different conductivity properties embedded inside. These inclusions are going to model conductivity anomalies in tissue. The type of parameterization we use for the inclusions should be chosen based on the regularity of the anomalies we want to model, and how many parameters are feasible to use with our forward model. These are related as complex shapes often require many degrees of freedom. We have to evaluate the forward model twice in (3.41) for each parameter we sample, assuming just a single Metropolis step inside each Gibbs iteration. A Naive approach to setting a prior for the conductivity σ is discretizing it into pixels and treating each pixel as an unknown, yielding n^3 unknowns in total. Recall that we have to evaluate the forward model twice each time we compute the acceptance ratio in an MH step in eq. (3.24). Drawing just 1 sample of a pixel-wise σ would require at least $2n^3$ evaluations of the forward model, assuming that all samples are accepted and that we only do a single Metropolis-within-Gibbs iteration. Therefore, we decide to take an alternative approach to model the conductivity field.

Barth et al. propose a low dimensional representation of the conductivity field in [14] by focusing on the location of inclusions, i.e. localized anomalies in the conductivity field. The authothe rs use three types of random fields to describe unknown inclusions, with each prior containing some number of degrees of freedom. We take a more simplified approach and instead look for inclusions parameterized by balls and ellipsoids.

5.1.1 Prior distribution

As we chose σ to be one or more inclusions embedded in a background medium of a lesser conducting material, we need to parameterize the inclusions themselves to determine which parameters to reconstruct. The parameterization of σ for the spherical inclusion in figure 3.1 is a function of two sets of parameters. Modeling inclusions as spheres is a good compromise between complexity and efficiency. Spheres provide information about the inclusions' location and volume while they are described with just a few parameters. Later in this section, we also look into ellipsoids for modeling inclusions which reveals the orientation of an inclusion as well. Assuming a background conductivity value of 1 S/cm and a conductivity value of 2 S/cm inside the inclusion, we parameterize a sphere using its radius $r \in \mathbb{R}$ and its center coordinates $c = (c_x, c_y, c_z) \in \mathbb{R}^3$

$$\sigma(c, r) = \begin{cases} 2 \text{ S/cm} & \text{if } (x - c_x)^2 + (y - c_y)^2 + (z - c_z)^2 \leq r^2, \\ 1 \text{ S/cm} & \text{otherwise.} \end{cases} \quad (5.1)$$

This construction assumes that we already know the electrical conductivity values inside and outside the inclusion. If only the conductivity of the background medium is known we can add an extra independent parameter $\kappa > 0$ to denote the conductivity inside the inclusion. We also consider an ellipsoidal inclusion with radius parameters denoting the semi-axes $r = (r_x, r_y, r_z) \in \mathbb{R}^3$, the center coordinates $c = (c_x, c_y, c_z) \in \mathbb{R}^3$, and unknown conductivity value $\kappa \in \mathbb{R}$.

$$\sigma(c, r, \kappa) = \begin{cases} \kappa \text{ S/cm} & \text{if } \frac{(x-c_x)^2}{r_x^2} + \frac{(y-c_y)^2}{r_y^2} + \frac{(z-c_z)^2}{r_z^2} \leq 1, \\ 1 \text{ S/cm} & \text{otherwise.} \end{cases} \quad (5.2)$$

In the case of multiple inclusions available in the conductivity field, sigma is defined as several spherical inclusions with different conductivity properties. Consider two spherical inclusions parameterized by two radii $r = (r_1, r_2) \in \mathbb{R}^2$, two pairs of center coordinates $c_1 = (c_{1x}, c_{1y}, c_{1z}) \in \mathbb{R}^3$ and $c_2 = (c_{2x}, c_{2y}, c_{2z}) \in \mathbb{R}^3$, and two conductivity levels $\kappa = (\kappa_1, \kappa_2) \in \mathbb{R}^2$

$$\sigma(c, r, \kappa) = \begin{cases} \kappa_1 \text{ S/cm} & \text{if } (x - c_{1x})^2 + (y - c_{1y})^2 + (z - c_{1z})^2 \leq r_1^2, \\ \kappa_2 \text{ S/cm} & \text{if } (x - c_{2x})^2 + (y - c_{2y})^2 + (z - c_{2z})^2 \leq r_2^2, \\ 1 \text{ S/cm} & \text{otherwise.} \end{cases} \quad (5.3)$$

Intersecting inclusions can be handled by this parameterization too. The conductivity level of the region of overlap between the intersecting inclusions will have a value of κ_2 since the conductivity distribution σ is built with an `if`-statement in FEniCS. The advantage of a simple, parametric approach is that the parameters can be assumed to be independent. In section 3.1.3 we discuss how to derive the conditional probability densities for a problem involving two independent parameters r and c . Applying the same technique to include more parameters, e.g. an unknown conductivity κ , is straightforward.

After choosing a parameterization for σ we must describe the probability distribution of its parameters. As a running example, consider a spherical inclusion in σ inside the unit cube as done in section 3.1.3. This inclusion is defined by its center and radius parameters c and r as in (5.1). We will refer to this specific inclusion parameterization for the rest of the chapter. Since we work in the unit cube we impose the constraints that the center coordinate c is contained within $[0, 1]^3$ and that the radius is within $[0, 0.5]$. Therefore, we assign a uniform prior to c and r since we have no prior knowledge of their values apart from the bounds.

$$c \sim \mathcal{U}[0, 1]^3, \quad r \sim \mathcal{U}[0, 0.5]. \quad (5.4)$$

Uniform priors make the conditional distributions in (3.47) take a simple form.

5.1.2 The likelihood function

We assumed that the additive noise E in (3.2) follows a Gaussian distribution with zero mean and variance γ^2 , i.e. $E \sim \mathcal{N}(0, \gamma^2)$. The noisy forward model in (3.2) gives the relation between conductivity σ , noisy magnetic field data y_{obs} and noise E

$$y_{\text{obs}} = (\mathcal{G} \circ \mathcal{F})(\sigma(c, r)) + E \iff E = y_{\text{obs}} - (\mathcal{G} \circ \mathcal{F})(\sigma(c, r)). \quad (5.5)$$

So we conclude that $y_{\text{obs}} - (\mathcal{G} \circ \mathcal{F})(\sigma(c, r))$ is distributed like E . This determines the likelihood function $\pi_{\text{like}}(y_{\text{obs}}|c, r)$ to be Gaussian, i.e.

$$\pi_{\text{like}}(y_{\text{obs}}|c, r) = \frac{1}{\sqrt{2\pi\gamma^2}} \exp\left(-\frac{1}{2} \frac{\|y_{\text{obs}} - (\mathcal{G} \circ \mathcal{F})(\sigma(c, r))\|_2^2}{\gamma^2}\right). \quad (5.6)$$

Assuming that c and r are independent, we can use Bayes' theorem in equation (3.3) to write an expression proportional to the joint posterior distribution

$$\pi_{\text{post}}(c, r|y_{\text{obs}}) \propto \pi(y_{\text{obs}}|c, r)\pi(c, r) = \pi(y_{\text{obs}}|c, r)\pi(c)\pi(r). \quad (5.7)$$

Using the prior distribution for c and r defined in (5.4) and the likelihood from (5.6) above, we can write up the joint posterior in closed-form

$$\pi_{\text{post}}(c, r | y_{\text{obs}}) = \frac{1}{\sqrt{2\pi\gamma^2}} \cdot \frac{1}{1-0} \cdot \frac{1}{0.5-0} \cdot \exp\left(-\frac{1}{2} \frac{\|y_{\text{obs}} - (\mathcal{G} \circ \mathcal{F})(\sigma(c, r))\|^2}{\gamma^2}\right). \quad (5.8)$$

In the Gibbs sampler, we have to use the conditional distributions $\pi(r|c, y_{\text{obs}})$ and $\pi(c|r, y_{\text{obs}})$ respectively, these were derived in (3.47) for the simple case we are considering. We want to obtain the conditional distribution of r given the center c and magnetic field data y_{obs} . Applying the Bayes' rule and (5.6) yields

$$\pi(r|c, y_{\text{obs}}) = \pi_{\text{like}}(y_{\text{obs}}|c, r)\pi(r) = \frac{1}{\sqrt{2\pi\gamma^2}} \cdot \frac{1}{0.5-0} \cdot \exp\left(-\frac{1}{2} \frac{\|y_{\text{obs}} - (\mathcal{G} \circ \mathcal{F})(\sigma(c, r))\|^2}{\gamma^2}\right). \quad (5.9)$$

Similarly, the posterior conditional distribution of c given the radius r and magnetic field data y_{obs} can be written as

$$\pi(c|r, y_{\text{obs}}) = \pi_{\text{like}}(y_{\text{obs}}|c, r)\pi(c) = \frac{1}{\sqrt{2\pi\gamma^2}} \cdot \frac{1}{1-0} \cdot \exp\left(-\frac{1}{2} \frac{\|y_{\text{obs}} - (\mathcal{G} \circ \mathcal{F})(\sigma(c, r))\|^2}{\gamma^2}\right). \quad (5.10)$$

These conditional distributions are also not available in full closed form and are thus intractable. It is infeasible to compute the denominator of Bayes' rule in (3.3) with the above expressions. Thus we must use a Metropolis-Hastings step to sample from them.

For numerical stability and to avoid computational overflow, we usually take the logarithm of the posterior conditional distributions. This is allowed since we only have to compute fractions involving the posterior in both the numerator and denominator. We can thus express the log of the conditional distributions as

$$\begin{aligned} \log \pi(r|c, y_{\text{obs}}) &\propto -\frac{1}{2} \frac{\|y_{\text{obs}} - (\mathcal{G} \circ \mathcal{F})(\sigma(c, r))\|^2}{\gamma^2}, \quad c \text{ fixed}, \\ \log \pi(c|r, y_{\text{obs}}) &\propto -\frac{1}{2} \frac{\|y_{\text{obs}} - (\mathcal{G} \circ \mathcal{F})(\sigma(c, r))\|^2}{\gamma^2}, \quad r \text{ fixed}. \end{aligned} \quad (5.11)$$

Where we have removed any proportionality constants in the closed-form expressions of the log conditional distributions. These can be omitted since we only have to compute fractions involving the closed-form expression of the log of the conditional distribution in both the numerator and the denominator. We also note that these two expressions are simply equal to the log of the likelihood function $\log(\pi(y_{\text{obs}}|c, r)) = \log(\pi_{\text{like}}(y_{\text{obs}}|c, r))$, but with one of the parameters fixed. We can thus easily construct the log posterior conditional distributions by simply taking the log-likelihood function and fixing one parameter. The conditional distributions are the ones we will sample from via the Metropolis-Hastings step in each Gibbs iteration, for the algorithm presented in section 3.1.3. Extending the technique to include another unknown parameter such as an unknown conductivity value κ is straightforward. We have to include an additional Metropolis step inside the Gibbs sampler since κ is assumed to be independent of c and r .

5.2 Simulation of forward problem

Recall from section 2.1 that the forward problem consisted of two mappings. The first mapping takes the parameterized conductivity distribution $\sigma(p)$, with p being the parameters, to the current density field \mathbf{J} . To obtain the current density, we solve for the electric potential u inside the

unit cube $\Omega = [0, 1]^3$, i.e. solving the Poisson equation with Dirichlet boundary conditions. This is done using FEM on an $n \times n \times n$ grid using the FEniCS platform in Python [24].

$$\begin{aligned} -\nabla \cdot \sigma \nabla u &= 0 && \text{in } \Omega, \\ u(x, y, z) &= x && \text{on } \partial\Omega. \end{aligned} \tag{5.12}$$

For the Dirichlet boundary condition, we choose the function $u(x) = x$. Physically, this means keeping the voltage on the cube's right side at a fixed value of one, with the left side is kept at zero. This construction models that the current applied from the electrodes in the MREIT setup in figure 2.1 runs from right to left. This choice of boundary condition is inspired by the authors of [2]. A short piece of code outlining a FEniCS implementation of the Poisson equation along with the computation of the current density field is shown in listing A.1 in Appendix 2.

After obtaining an approximation of u we compute the gradient $\nabla u(\mathbf{x})$ in FEniCS. We then compute the current density field $\mathbf{J}(\mathbf{x}) = -\sigma(\mathbf{x}) \nabla u(\mathbf{x})$ inside $[0, 1]^3$ and extract the function values on the grid. The magnetic field \mathbf{B} induced by \mathbf{J} is then computed via the Biot-Savart integral [11]

$$\mathbf{B}(\mathbf{x}) = \frac{1}{4\pi} \int_{\Omega} \mathbf{J}(\mathbf{x}') \times \frac{\mathbf{x} - \mathbf{x}'}{|\mathbf{x} - \mathbf{x}'|^3} d\mathbf{x}'. \tag{5.13}$$

We use the spectral-Fourier methods discussed in section 4.2 to evaluate this integral. A Python implementation of the Fourier method for the Biot-Savart integral covered in section 4.2 as written by the authors of [11] is shown in listing A.2 in Appendix 2.

5.2.1 Data generation

When generating synthetic data for an inverse problem we must make sure that we do not commit an *inverse crime*. An inverse crime is a term commonly used for the act of using the same numerical ingredients, e.g. parameters and computational mesh, to generate the data and to invert data [6]. Doing this can make numerical treatment of inverse problems underperform in real-life scenarios. Special care must thus be taken when generating synthetic data for an inverse problem. In this project, we propose to remedy this issue by generating the data on an $(n+2) \times (n+2) \times (n+2)$ fine grid and interpolating it down to an $n \times n \times n$ computational grid for sampling. We must also make sure to scale the additive noise correctly since we wish for the data to have relative noise. Given exact magnetic field data y_{exact} and a realization of standard normal Gaussian noise e , we first normalize the noise by dividing it by its norm. We then multiply the normalized noise with $\|y_{\text{exact}}\|_2/100$ to obtain 1% noise relative to y_{exact} . This can then be scaled by any desired percentage of relative noise. Adding 10% relative noise to y_{exact} to obtain y_{obs} can be achieved by

$$y_{\text{obs}} = y_{\text{exact}} + \|y_{\text{exact}}\|_2 \cdot \frac{e}{\|e\|_2} \cdot \frac{10}{100}. \tag{5.14}$$

In section 4.1.1 we discuss how a two-dimensional and a three-dimensional formulation of the Poisson problem are equivalent. Given appropriate homogenous Neumann boundary conditions in the z -direction for the 3D problem. This produces a solution that is invariant in the z -direction, i.e. where each slice in the z -direction is equivalent to the solution of the 2D problem. In this situation, the gradient of u in the z -direction vanishes. Thus the z -component J_3 of the current density field $\mathbf{J} = (J_1, J_2, J_3) = -\sigma \nabla u$ also vanishes, and we thus cannot get information about the x - and y -components of the magnetic field (B_1, B_2) from the computation in (4.28).

The full-3D model does not need the z -component of the gradient of u . Recall that in an MRI procedure, as depicted in 2.1, one only measures the z -component of the magnetic field B_3 . Thus, from (4.28) we conclude that we only need knowledge of the x - and y -components of \mathbf{J}

to compute magnetic field data. The code snippet implementing the spectral-Fourier method for evaluating the Biot-Savart integral in listing A.2 also reveals that the computation of B_3 does not require knowledge of J_3 . In a sampling method, we can thus omit the computation of B_1 and B_2 for greater efficiency.

Note that the computational efficiency of the two cases is not equivalent. Solving the 2D Poisson equation on an $n \times n$ grid and stacking the solution n times in the z -direction has a computational efficiency of $O(n^2)$, while solving the 3D Poisson equation on an $n \times n \times n$ grid is an $O(n^3)$ computation. Therefore we will consider two versions of the inverse problem:

1. **Pseudo-3D method:** we parameterize $\sigma(p)$ to be a 2D conductivity field and solve the 2D Poisson equation to obtain the electric potential u . We then compute the current $\mathbf{J} = (J_1, J_2) = \sigma \nabla u$. Let $\tilde{\mathbf{J}}$ denote n copies of \mathbf{J} stacked in the z -direction. We then use $\tilde{\mathbf{J}} = (\tilde{J}_1, \tilde{J}_2)$ to compute magnetic field data B_3 via (4.28).
2. **Full-3D method:** we parameterize $\sigma(p)$ to be a 3D conductivity field and solve the 3D Poisson equation to obtain the electric potential u . We then compute $\mathbf{J} = (J_1, J_2) = \sigma \nabla u$ and use these to compute B_3 via (4.28).

We can achieve much better resolution in the inversion for the same computational effort by considering the pseudo-3D method instead of the full-3D method. This approach avoids the heavy computational load of solving PDE in 3D. We compare the computing time between the two approaches for different mesh sizes n in table 5.1. We use the simple circular/spherical inclusion defined in (5.1) as the true conductivity distribution and propagate it through the forward pipeline $\sigma \mapsto \tilde{\mathbf{J}} \mapsto B_3$. The computations were performed on a desktop computer running an Intel(R) Core(TM) i5-3570K CPU processor at 3.40 GHz with 8 GB of RAM.

Mesh size n	pseudo-3D method	full-3D method
8	0.011 s	0.108 s
16	0.021 s	0.984 s
32	0.069 s	12.94 s
64	0.274 s	-

Table 5.1: Full forward model evaluation time in seconds for different mesh sizes

The execution of the full-3D method is interrupted in table 5.1 for $n = 64$, since more memory is required to project the current density $\mathbf{J} = -\sigma \nabla u$ onto the finite element space. The conductivity field σ from (5.1) is shown in figure 5.1 for the pseudo-3D and the full-3D method. The magnetic field data is presented in the middle and bottom rows of figure 5.1 for 10% relative noise. Note in the bottom row how the magnetic field is flowing in the x -direction. This is due to the current field also flowing in the x -direction, more precisely from right to left. With the $u(x, y, z) = x$ boundary condition, we use in eq. (5.12) we have set the potential to $u = 1$ at the right $x = 1$ boundary and $u = 0$ on the left $x = 0$ boundary. Thus the current field \mathbf{J} also points from right to left. A similar approach is used to generate data for other prior parameterizations $\sigma(p)$.

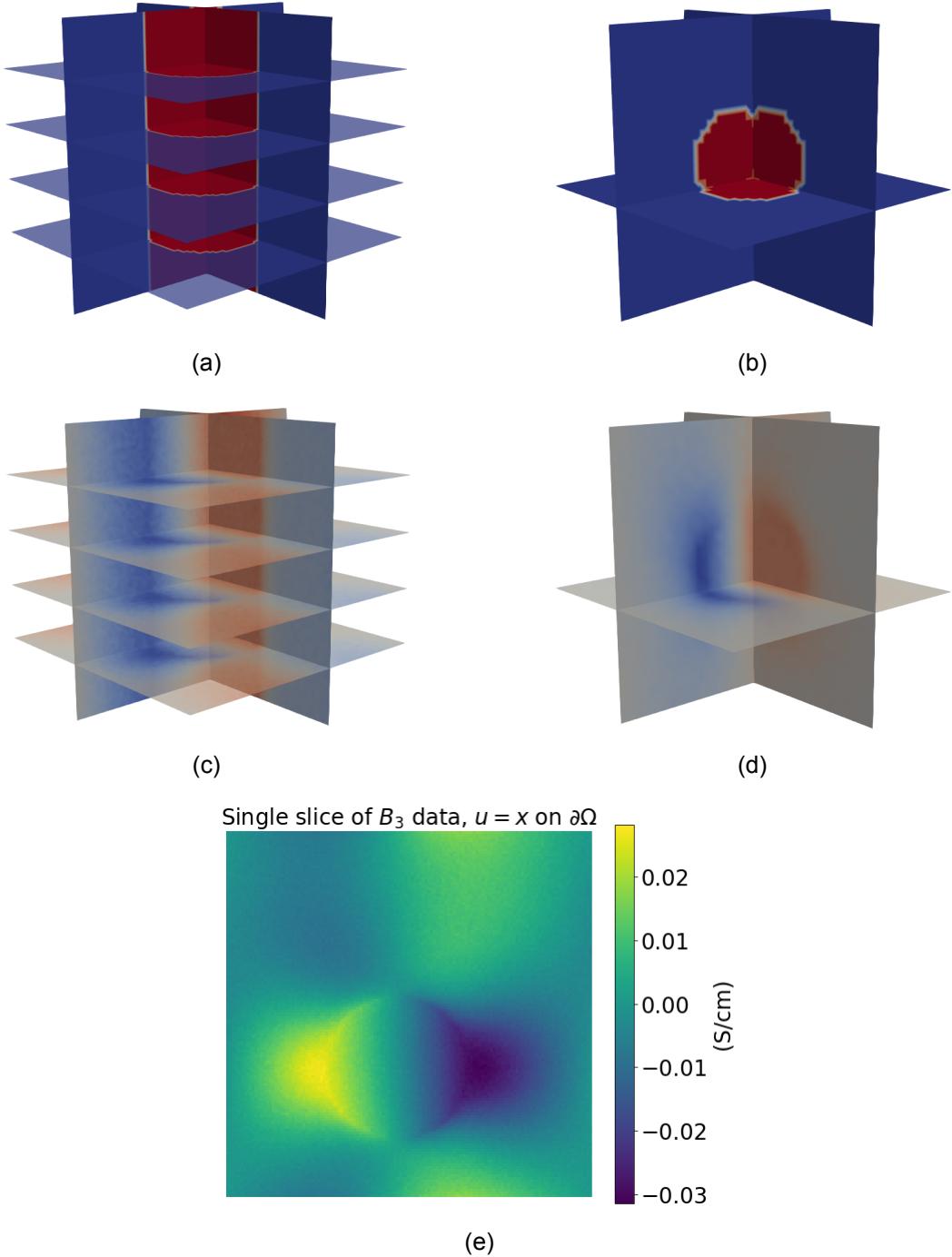


Figure 5.1: Top row: Example conductivity distributions $\sigma(c, r)$ defined by (5.1) for the pseudo-3D model in (a) and the full-3D model in (b). Middle row: magnetic field data from applying the forward model $\mathcal{G} \circ \mathcal{F}$ to the conductivity field $\sigma(c, r)$ for the pseudo-3D model in (a) and the full-3D model (b). Bottom row: (e) single slice of magnetic field data.

5.3 Solution of the Bayesian inverse problem

Once the noisy data y_{obs} has been obtained and the posterior distribution (3.3) has been defined, we are ready to solve the inverse problem in (3.1) via a sampling method. The posterior probability distribution contains information about the unknown conductivity field σ given noisy magnetic field data. We continue with the prior assumption of σ being parameterized by its center c and

radius r , see (5.1). As an example in the figures below, we consider the pseudo-3D method with the z -invariant conductivity distribution parameterized by $(c, r) = (c_x, c_y, r) \in \mathbb{R}^2 \times \mathbb{R}$. The full-3D method is saved for the next chapter. We will now describe our approach to characterizing the posterior distribution and reconstructing the conductivity field for given observed data y_{obs} . We will consider two ways of doing uncertainty quantification on the estimated parameters.

5.3.1 Posterior sampling

Assuming the parameters that determine the geometry of σ are independent, we present the posterior distributions in (5.11). We sample from these distributions using the Metropolis-within-Gibbs sampler as described in section 3.1.3.

Here we explain the Metropolis-within-Gibbs method for the particular example of estimating the two parameters c and r given noisy data y_{obs} . Let $x_k = [c_k, r_k]$ be our state vector at the k 'th Gibbs iteration. Assuming the pseudo-3D method we have $x_k \in \mathbb{R}^2 \times \mathbb{R}$. We choose a proposal model of form $x_{k+1} = x_k + \beta w_k$ where $w_k \sim \mathcal{N}(0, 1)$ is a standard normal increment and β is some step size to be tuned. Since the proposal distribution is symmetric, the computation of the acceptance probability reduces to (3.25). We also assume that bounds for each parameter are provided, such that $c_k \in (c_{\text{low}}, c_{\text{high}})$ and $r_k \in (r_{\text{low}}, r_{\text{high}})$. If the step taken by the random walk model exceeds these bounds or any other constraint which we may supply to the sampler as we wish, we re-sample the step and check the constraints again. The sampling procedure is presented in algorithm 1

The proposal distribution is essentially determined by the step size β which should be tuned for each parameter. The step size should be such that the samples are exploring the posterior well. After sampling we can use the chain diagnostics we discuss in section 3.2 to estimate the performance of the sampling procedure. We can also look at the *acceptance rate* which is the proportion of the proposed samples which are accepted by the sampling method. According to [29] the optimal acceptance ratio is 23.4%. We can tune the step size β such that the observed acceptance rate approaches this value. One such tuning period consists of drawing a number of samples, e.g. $K_t = 200$, where the acceptance rate is computed and β is updated every 10th sample. The update is based on the difference between the observed acceptance rate and the optimal one. The adaptive stepsize procedure is presented in algorithm 2 follow [29]

In the experience of the authors, this adaptive step size scheme in algorithm 2 does not always yield step sizes β which are good for exploring the posterior. Nevertheless, it gives a good indication of which step sizes to use initially. Looking at the parameter traces and doing chain diagnostics can aid in the further manual tuning of step sizes $\beta = [\beta_c, \beta_r]$ for each parameter. Sampling from the posterior requires the following ingredients

- Supply the log-likelihood function $\log \pi_{\text{like}} = -\frac{1}{2} \frac{\|y_{\text{obs}} - (\mathcal{G} \circ \mathcal{F})(\sigma(c, r))\|_2^2}{\gamma^2}$ to construct conditional distributions.
- An initial guess $x_0 = [c_0, r_0]$ for each parameter.
- A step size for each parameter $\beta = [\beta_c, \beta_r]$, either by running the above scheme or guessing.
- Bounds for each parameter $(c_{\text{low}}, c_{\text{high}})$ and $(r_{\text{low}}, r_{\text{high}})$.
- Number of Metropolis iterations $[M_c, M_r]$ to perform in each Gibbs iteration.
- Number of samples K to draw in total.

5.3.2 Uncertainty quantification

When we have obtained posterior samples $\{c_k, r_k\}_{k=1}^K$ of the parameters determining the conductivity distribution $\sigma(c, r)$, we can consider two methods of doing uncertainty quantification

Algorithm 1 Metropolis-within-Gibbs sampler

Require: Step sizes $\beta = [\beta_c, \beta_r]$, number of within-Gibbs iterations $[M_c, M_r]$ and desired sample size K . Supply initial guess (c_0, r_0) , log conditional densities $\log \pi(r_k | c_k, y_{\text{obs}})$ and $\log \pi(c_k | r_k, y_{\text{obs}})$.

```

1:  $k \leftarrow 0$ 
2:  $x_0 \leftarrow [c_0, r_0]$ 
3: while  $k < K$  do
4:   Initialize within-Gibbs chain for the center,  $z_1 \leftarrow c_k$ 
5:   for  $i = 1, \dots, M_c$  do            $\triangleright$  Metropolis-Hastings steps for component ck
6:     Draw  $w_i \sim \mathcal{N}(0, 1)$  and set  $y_c = z_i + \beta_c w_i$ . Redraw if  $y_c \notin (c_{\text{low}}, c_{\text{high}})$ 
7:     Compute acceptance probability

$$\alpha(z_i, y_c) = \min(1, \exp(\log \pi(y_c | r_k, y_{\text{obs}}) - \log \pi(z_i | r_k, y_{\text{obs}})))$$
.
8:     Draw  $t \sim \mathcal{U}[0, 1]$  as the probability of accepting the new sample.
9:     if  $t \leq \alpha(z_i, y_c)$  then
10:       Update within-Gibbs chain  $z_{i+1} = y_c$ 
11:     else
12:       Do not update within-Gibbs chain  $z_{i+1} = z_i$ 
13:     end if
14:   end for
15:   Update outer Gibbs chain and set  $c_{k+1} = z_{M_c}$ 
16:   Initialize within-Gibbs chain for radius,  $s_1 \leftarrow r_k$ 
17:   for  $j = 1, \dots, M_r$  do            $\triangleright$  Metropolis-Hastings steps for component rk
18:     Draw  $w_j \sim \mathcal{N}(0, 1)$  and set  $y_r = s_j + \beta_r w_j$ . Redraw if  $y_r \notin (r_{\text{low}}, r_{\text{high}})$ 
19:     Compute acceptance probability

$$\alpha(s_j, y_r) = \min(1, \exp(\log \pi(y_r | c_{k+1}, y_{\text{obs}}) - \log \pi(s_j | c_{k+1}, y_{\text{obs}})))$$
.
20:     Draw  $t \sim \mathcal{U}[0, 1]$  as the probability of accepting the new sample.
21:     if  $t \leq \alpha(s_j, y_r)$  then
22:       Update within-Gibbs chain  $s_{j+1} = y_r$ 
23:     else
24:       Do not update within-Gibbs chain  $s_{j+1} = s_j$ 
25:     end if
26:   end for
27:   Update outer Gibbs chain and set  $r_{k+1} = s_{M_r}$ 
28:    $k \leftarrow k + 1$ 
29: end while

```

Algorithm 2 Adaptive stepsize procedure

Require: Supply β_c , target acceptance ratio, and initial guess c_0 .

```

1:  $k \leftarrow 0$ 
2:  $v_{\text{acc}} \leftarrow \{1\}$ 
3:  $n_{\text{up}} \leftarrow 0$ 
4: for  $k = 1, \dots, 200$  do
5:   Draw  $w_k \sim \mathcal{N}(0, 1)$  and set  $y_c = c_k + \beta_c w_k$ . Redraw if  $y_c \notin (c_{\text{low}}, c_{\text{high}})$ 
6:   Compute acceptance probability

$$\alpha(c_k, y_c) = \min(1, \exp(\log \pi(y_c | r_k, y_{\text{obs}}) - \log \pi(c_k | r_k, y_{\text{obs}})))$$
.
7:   Draw  $t \sim \mathcal{U}[0, 1]$  as the probability of accepting the new sample.
8:   if  $t \leq \alpha(c_k, y_c)$  then
9:     Update chain  $c_{k+1} = y_c$ 
10:    Append 1 to  $v_{\text{acc}}$ 
11:   else
12:     Do not update chain  $c_{k+1} = c_k$ 
13:     Append 0 to  $v_{\text{acc}}$ 
14:   end if
15:   if  $\text{mod}(k, 10) = 0$  then
16:     Compute current average acceptance rate  $\bar{v}_{\text{acc}}$ , and  $\zeta = \frac{1}{\sqrt{n_{\text{up}}+1}}$ .
17:     Update  $\beta_c \leftarrow \exp(\log(\beta_c) + \zeta(\bar{v}_{\text{acc}} - \text{acc}^*))$ 
18:      $n_{\text{up}} \leftarrow n_{\text{up}} + 1$ 
19:   end if
20: end for

```

(UQ). One involves UQ on the parameter space $(c_k, r_k) \in \mathbb{R}^2 \times \mathbb{R}$ directly, and the other involves UQ on the mapped parameters onto the conductivity space $\sigma \in \mathbb{R}^{n \times n \times n}$ through the parameterization. Mapping the parameters through the parameterization and onto the conductivity space is also commonly known as a pushforward

1. The first approach involves doing UQ on the samples directly and computing the posterior mean of the center coordinates and radius as scalars. We also compute the variance of these parameters. We can then construct the conductivity distribution with the posterior means as parameters $\sigma(\mathbb{E}(c_k), \mathbb{E}(r_k))$, yielding a single estimate of a conductivity distribution. The variance of the sampled chains is used to compute credibility intervals for each parameter.
2. The second approach involves constructing the conductivity distribution that each sample (c_k, r_k) defines, such that we have K different samples of $\sigma(c_k, r_k)$ as conductivity fields of size $n \times n \times n$. We can then compute the posterior mean in conductivity space $\mathbb{E}(\sigma(c_k, r_k))$, and also compute the variance of the samples $\mathbb{V}(\sigma(c_k, r_k))$ pixel-wise. We can visualize the sample mean $\mathbb{E}(\sigma(c_k, r_k))$ as the conductivity distribution estimate, and also the standard deviation $\sqrt{\mathbb{V}(\sigma(c_k, r_k))}$ to get a pixel-by-pixel estimate of the uncertainty.

The circular conductivity prior distribution for the pseudo-3D problem is parameterized by $\sigma(c, r)$ with $c \in \mathbb{R}^2$ and $r \in \mathbb{R}$. This is shown in the top left of figure 5.1. Using this conductivity distribution as an example, we solve the Bayesian inverse problem by sampling from the posterior conditional distributions in (5.11) via the Metropolis-within-Gibbs sampler we described in the previous section. After the sampling is done we perform the UQ analysis described above. We present the posterior mean as our estimate of the conductivity field. The results for uncertainty

quantification on parameter space are shown in figure 5.2. Results for the uncertainty quantification in conductivity space are shown in 5.3, where we plot a single slice of the vertically invariant pseudo-3D solution. We plot the full pseudo-3D solution in the open-source visualization software Paraview. These results are displayed in figure 5.4. 5000 samples were drawn and the first 1000 were discarded as burn-in. From figure 5.2 we see that the true values are

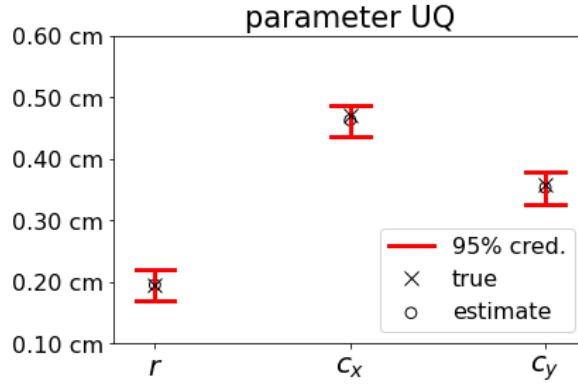


Figure 5.2: Example of UQ analysis for parameter samples (c_k, r_k) .

contained within the confidence intervals for all parameters in this simple case, and the estimated values are close to the true ones in all cases. Since the conductivity parameterization

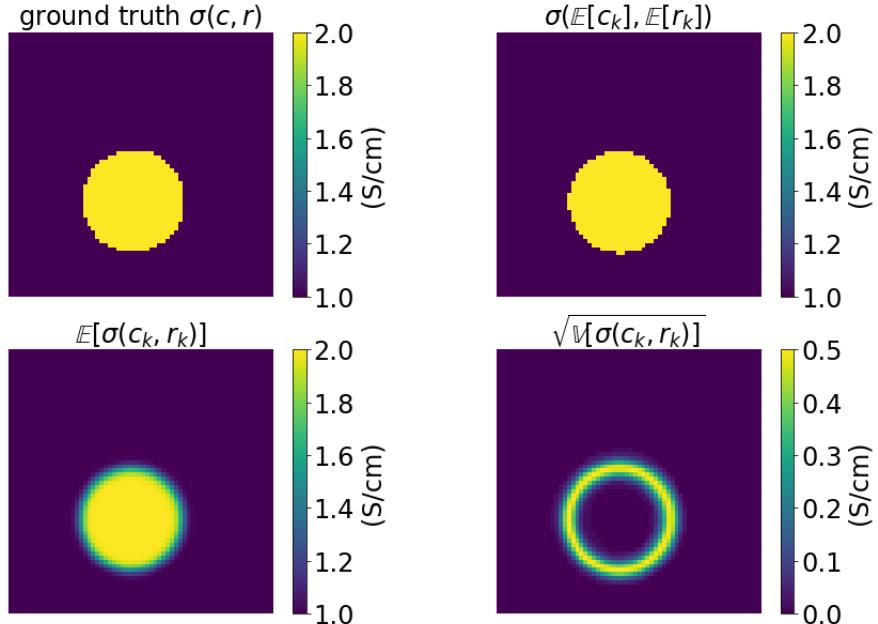


Figure 5.3: Example of UQ analysis for the mapped parameter samples $\sigma(c_k, r_k)$. Top left: true conductivity distribution. Top right: pushforward of parameter sample means. Bottom left: mean of pushforward of parameters. Bottom right: standard deviation of pushforward of parameters.

$\sigma(c, r)$ is piecewise constant, the two conductivity distributions in the top row of figure 5.3 will be so too. For the mean of the parameterization pushforwards in the bottom left row, the conductivity values seem to decay smoothly at the boundary of the inclusion. This is because the circular inclusion in all of the individual $\sigma(c_k, r_k)$ samples are scattered in a small area around the true inclusion. Not all of the sampled inclusions overlap exactly at the boundary. This causes the dip in the mean value of all samples at the boundary, but not in the center of the inclusion as most

samples overlap here. This is also evident from the plot of the standard deviation in the bottom right of the above figure. There is some uncertainty around the boundary of the inclusion, but otherwise, the standard deviation is close to zero inside the inclusion. This is because we assumed the conductivity value of the inclusion to be known with a value of 2 S/cm, so we should not expect a high standard deviation inside of the inclusion. Results similar to the above can be displayed in 3D using Paraview. For the true conductivity distribution and the pushforward

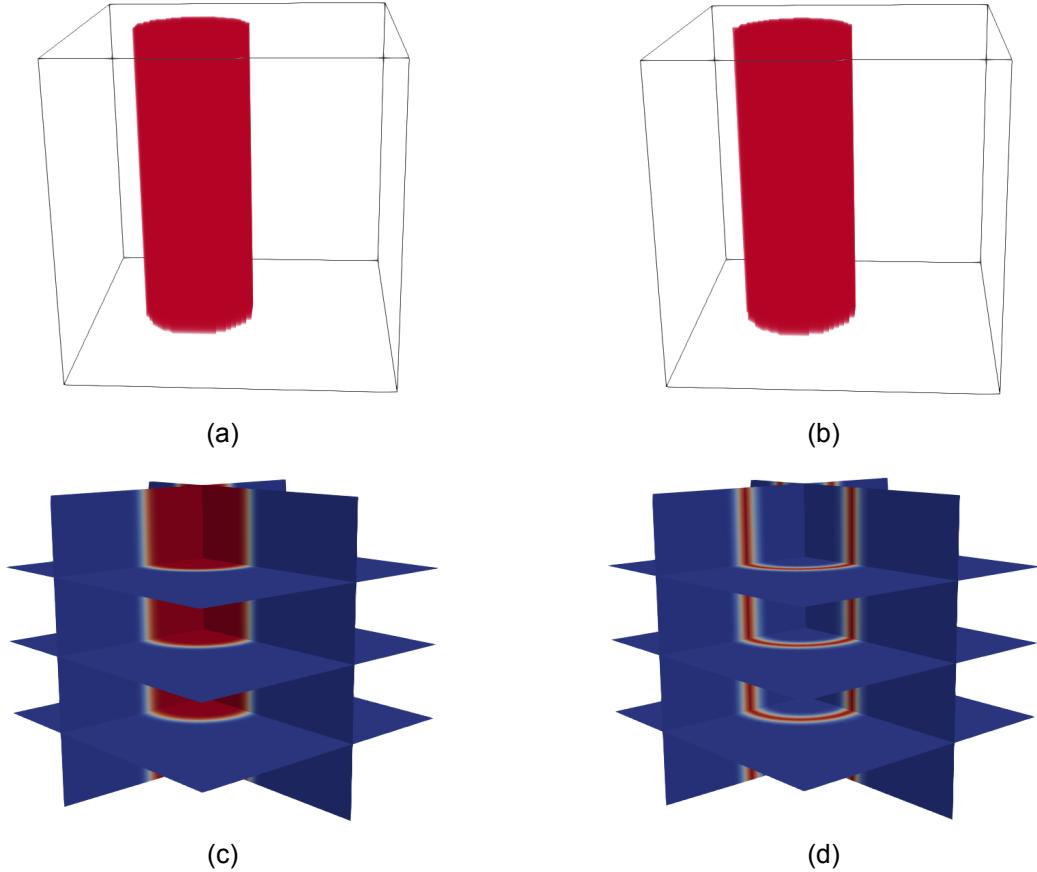


Figure 5.4: Example of UQ analysis for the mapped parameter samples $\sigma(c_k, r_k)$ in Paraview. (a): true conductivity field. (b): pushforward of parameter sample means. (c): mean of pushforward of parameter samples. (d): standard deviation of pushforward of parameter samples.

of the parameter means in the top row of figure 5.4, we can threshold the conductivity values in Paraview since the conductivity values are piecewise constant inside the domain. For the mean and standard deviation of the mapped parameters, we have to visualize via slicing as these are not piecewise constant. We want to be able to see the decay at the inclusion boundary of the mapped parameter mean. Also, the standard deviation inside the inclusion is not the same as on the boundary. Note that the viewpoint is the same in the top row and bottom row respectively. This is chosen such that we can see both inclusions without overlap in the top row, and such that we can see the slices of both inclusions without anything being obstructed in the bottom row. For all of the results in the next chapter, we will do uncertainty quantification in parameter space akin to figure 5.2. For the uncertainty quantification in conductivity space, we will do the Paraview visualizations as shown in figure 5.4, except for the pure two-dimensional case in which we will display results as in 5.3.

6 Numerical results

We apply the computational framework discussed in the previous chapter to reconstruct an inclusion given noisy magnetic field data. We consider several different inclusions of increasing complexity parameterized by an increasing number of parameters. We are thus simulating anomalies with different conductivity properties than the homogenous background medium with constant conductivity value. As shown in table 5.1 we run into memory issues for larger grid sizes for the full-3D model. This is because the system of equations we construct in (4.11) when we use the finite element method becomes too large. This gives rise to very large and very sparse matrices which are difficult to handle. To remedy these issues we use iterative solvers instead of direct solvers for solving the system of equations. We have used the biconjugate gradient stabilized (bicgstab) solver and a successive over-relaxation (sor) preconditioner to gain the best performance in terms of runtime and memory usage [30, 31]. To gain access to more memory and be able to run multiple scripts at once, we use the High-Performance-Computing system at DTU to run all results.

We perform the experiments in a cube with a side length of 1 cm centered at (0.5, 0.5, 0.5) cm. We arbitrarily set the background conductivity value to $\kappa_0 = 1$ S/cm and use larger values for the simulated anomalies. In the sampling, we use 20 Metropolis iterations within each Gibbs iteration to decrease the covariance between adjacent samples. For the FEniCS parameters, we use a function space with linear Lagrange elements (CG1) for solving the Poisson equation in (5.12) for the electric potential u . Consequently, for the gradient of u we can use a vectorized function space with discontinuous constant Lagrange elements (DG0). This is the space that we will project the current density field $\mathbf{J} = -\sigma \nabla u$ onto. We add 10% relative noise to the magnetic field data \mathbf{B} to create the observed data y_{obs} . The options we just described for the solver and the function spaces are the same for all the following numerical experiments.

6.1 Benchmarking

If we want a point estimate of the reconstructed conductivity field for comparison with the MCMC results, we can use one of the techniques for posterior inference discussed in section 3.3. Specifically, we will consider the Maximum a Posteriori (MAP) estimate from equation (3.56). Computing this point requires using an optimization routine to maximize the joint posterior distribution in (5.8). We have to use gradient information for the optimization problem. Due to time constraints, we only consider the MAP estimate in 2D. We will be working with the MREIT forward model $(c, r) \mapsto u \mapsto \mathbf{J} \mapsto B_z$, mapping parameters of the conductivity field $\sigma(c, r)$ to the magnetic field data B_z . Computing gradients of the nonlinear mapping in parameter space $(c, r) \in \mathbb{R}^2 \times \mathbb{R}$ is tricky though, so we resort to working straight in the conductivity space $\sigma \in \mathbb{R}^{n \times n}$. This makes the negative log posterior distribution take the following form with noise variance γ^2

$$-\log \pi_{\text{post}}(\sigma | y_{\text{obs}}) \propto \frac{1}{2} \frac{\|y_{\text{obs}} - (\mathcal{G} \circ \mathcal{F})(\sigma)\|_2^2}{\gamma^2} \quad (6.1)$$

Since we are working directly in conductivity space and not in parameter space, we have to use a regularization term in the cost functional to ensure we find a feasible conductivity field σ . We choose the L^2 norm of σ scaled by a parameter α as the regularization term. The data-discrepancy term in the cost functional is the negative log posterior above. The cost functional can thus be written as

$$F(\sigma) = \frac{1}{2\gamma^2} \|y_{\text{obs}} - (\mathcal{G} \circ \mathcal{F})(\sigma)\|_2^2 + \frac{\alpha}{2} \|\sigma\|_2^2. \quad (6.2)$$

If we think of this cost functional as being equal to a negative log posterior then we can write for the full posterior distribution

$$\pi(\sigma|y_{\text{obs}}) \propto \exp\left(-\frac{1}{2\gamma^2}\|y_{\text{obs}} - (\mathcal{G} \circ \mathcal{F})(\sigma)\|_2^2\right) \cdot \exp\left(-\frac{\alpha}{2}\|\sigma\|_2^2\right). \quad (6.3)$$

So we see that this construction assumes a Gaussian prior on the conductivity field σ with variance equal to $\frac{1}{\alpha}$. To compute the MAP estimate we will minimize the cost functional in (6.2) using the L-BFGS-B algorithm implemented in the `fmin_l_bfgs_b` function from the `scipy` library in Python [32]. Ideally, we should supply the analytical gradient of the cost functional $\frac{dF}{d\sigma}$ to the optimization routine. We compute this as

$$\frac{dF}{d\sigma} = \frac{1}{\gamma^2} \frac{d(\mathcal{G} \circ \mathcal{F})}{d\sigma} (y_{\text{obs}} - (\mathcal{G} \circ \mathcal{F})(\sigma)) + \alpha\sigma. \quad (6.4)$$

So we have to compute the derivative of the full forward mapping $\frac{d(\mathcal{G} \circ \mathcal{F})}{d\sigma}$. This can be done in two stages. We denote by \mathcal{F} the operator which maps a conductivity field σ to a current density field \mathbf{J} . The derivative of this part of the mapping with respect to the conductivity field σ is the linear operator $d\mathcal{F}(\tilde{\sigma})$ from (2.13) in [2], denote this by $\frac{d\mathcal{F}}{d\sigma}$. We can then compute the derivative of the full forward mapping $\frac{d(\mathcal{G} \circ \mathcal{F})}{d\sigma}$. This is done by applying the Biot-Savart map \mathcal{G} to the derivative of the first part of the model $\frac{d\mathcal{F}}{d\sigma}$. This is possible because the Biot-Savart integral is linear and taking derivatives is also a linear operation. We are using that

$$\frac{d(\mathcal{G} \circ \mathcal{F})}{d\sigma} = \mathcal{G}\left(\frac{d\mathcal{F}}{d\sigma}\right). \quad (6.5)$$

The ground truth conductivity field that we are trying to reconstruct is an elliptic inclusion rotated by 45° in the xy -plane centered at $(0.5, 0.5)$, and with semi-axes equal to 0.15 and 0.3. The conductivity value of the inclusion is 2 S/cm and the conductivity level of the background medium is 1 S/cm. The true conductivity field is shown in figure 6.1a to the left in figure 6.1. All computations have been performed on a 64×64 mesh. We have set the regularization parameter to $\alpha = 10^{-5}$. For the stopping criteria in the optimization procedure, we have used a projected gradient tolerance of 10^{-9} and a relative difference factor of 10^4 . The iteration will stop when some component of the projected gradient is below tolerance, or if the relative difference $\frac{f_k - f_{k+1}}{\max\{|f_k|, |f_{k+1}|, 1\}} \leq 10^4 \cdot \epsilon$, where $\epsilon \approx 2.2 \cdot 10^{-16}$ is machine precision. The stopping criteria parameters have been chosen to provide high accuracy [33]. We add 10% noise to the observed magnetic field data y_{obs} and we initialize the optimization with a constant conductivity field σ_0 which takes the value 0.5 S/cm everywhere.

Computing the gradient of the cost functional $\frac{dF}{d\sigma}$ seems to be relatively straightforward in theory. However, in practice, we had a lot of issues with computing it. Our analytical gradient was not comparable with a finite difference approximation of it. Running the L-BFGS-B optimization with the analytical gradient gave poor results, these are shown in figure 6.1b. Only a small part of the boundary of the inclusion is reconstructed correctly. The conductivity value of the small reconstructed part and the value of the background are not far from the true values though. Due to the poor reconstruction using the analytical gradient, we opted to approximate the gradient numerically instead. This can be accomplished by setting the `approx_grad` flag equal to 1 in the call to the L-BFGS-B routine. The results for the approximate gradient are shown in figure 6.1c. The MAP estimate computed with approximate gradients has found the shape of the inclusion quite well, but the conductivity values are underestimated. Also due to time constraints, we did not manage to estimate the covariance matrix of the MAP point. Therefore we do not have any UQ results on this posterior estimate.

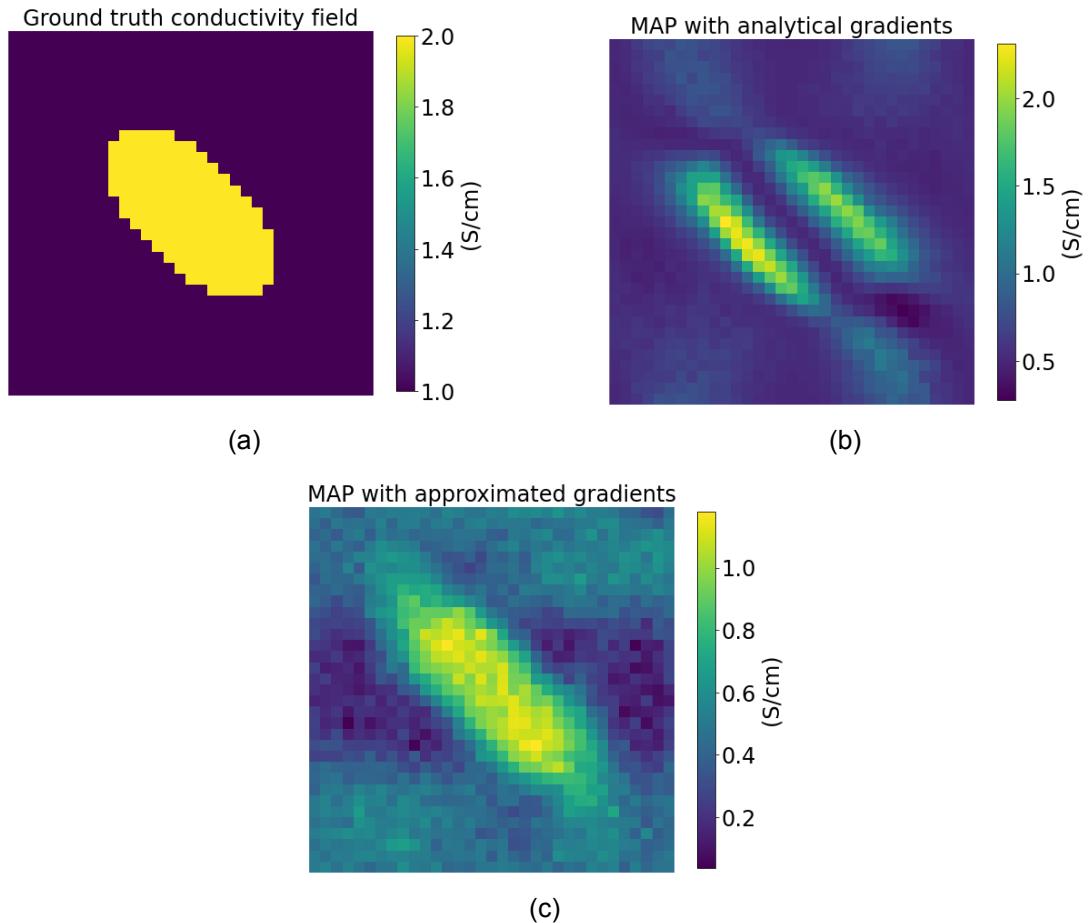


Figure 6.1: Results of the MAP computation in 2D. (a): rotated elliptic inclusion as the ground truth conductivity field σ . (b): MAP estimate with the analytical gradient $\frac{dF}{d\sigma}$ computed in (6.4). (c): MAP estimate with approximate gradients.

6.2 Problem 0: spherical inclusion with known conductivity

We first consider the toy problem we used as a running example in the previous chapter with the conductivity field parameterized by the expression for $\sigma(c, r)$ in (5.1). The ground truth conductivity fields and the noisy magnetic field data are shown in figure 5.1. For the pseudo-3D method, we have to estimate three parameters $(c, r) = ((c_x, c_y), r) \in \mathbb{R}^2 \times \mathbb{R}$. For the full-3D method, we have to estimate four parameters $(c, r) = ((c_x, c_y, c_z), r) \in \mathbb{R}^3 \times \mathbb{R}$. Using the Metropolis-within-Gibbs sampler presented in algorithm 1 we draw 5000 samples (c_k, r_k) from the conditional posterior distributions in (5.10) and (5.9). The parameters shown in table 6.1 were used to construct the true conductivity field and initialize the sampler. The CPU times can

Parameter	Value	Parameter	Value
True center c	$(0.47, 0.36)$ cm	True center c	$(0.47, 0.36, 0.26)$ cm
True radius r	0.195 cm	True radius r	0.195 cm
Conductivity κ	2 S/cm	Conductivity κ	2 S/cm
Data grid	$66 \times 66 \times 66$	Data grid	$10 \times 10 \times 10$
Reconstruction grid	$64 \times 64 \times 64$	Reconstruction grid	$8 \times 8 \times 8$
Initial center c_0	$(0.80, 0.10)$ cm	Initial center c_0	$(0.50, 0.50, 0.50)$ cm
Initial radius r_0	0.10 cm	Initial radius r_0	0.10 cm
Stepsize (β_c, β_r)	$(0.03, 0.06)$	Stepsize (β_c, β_r)	$(0.03, 0.04)$
Bounds $(c_{\text{low}}, c_{\text{high}})$	$(0, 1.0)$ cm	Bounds $(c_{\text{low}}, c_{\text{high}})$	$(0, 1)$ cm
Bounds $(r_{\text{low}}, r_{\text{high}})$	$(0.05, 0.50)$ cm	Bounds $(r_{\text{low}}, r_{\text{high}})$	$(0.05, 0.50)$ cm
Within Gibbs iter.	20	Within Gibbs iter.	20

Table 6.1: Initial parameters. Left: pseudo-3D method. Right: full-3D method.

be read off from the output files generated from executing scripts on the HPC system. With the above parameters, the CPU times are shown in table 6.2 below.

	pseudo-3D method	full-3D method
CPU time	14.25 hours	8 hours

Table 6.2: CPU times on the HPC system for problem 0.

6.2.1 Statistical analysis of posterior samples

Trace plots of the chains for each set of parameters (c, r) are shown for the pseudo-3D method in figure 6.2 and for the full-3D method in figure 6.3.

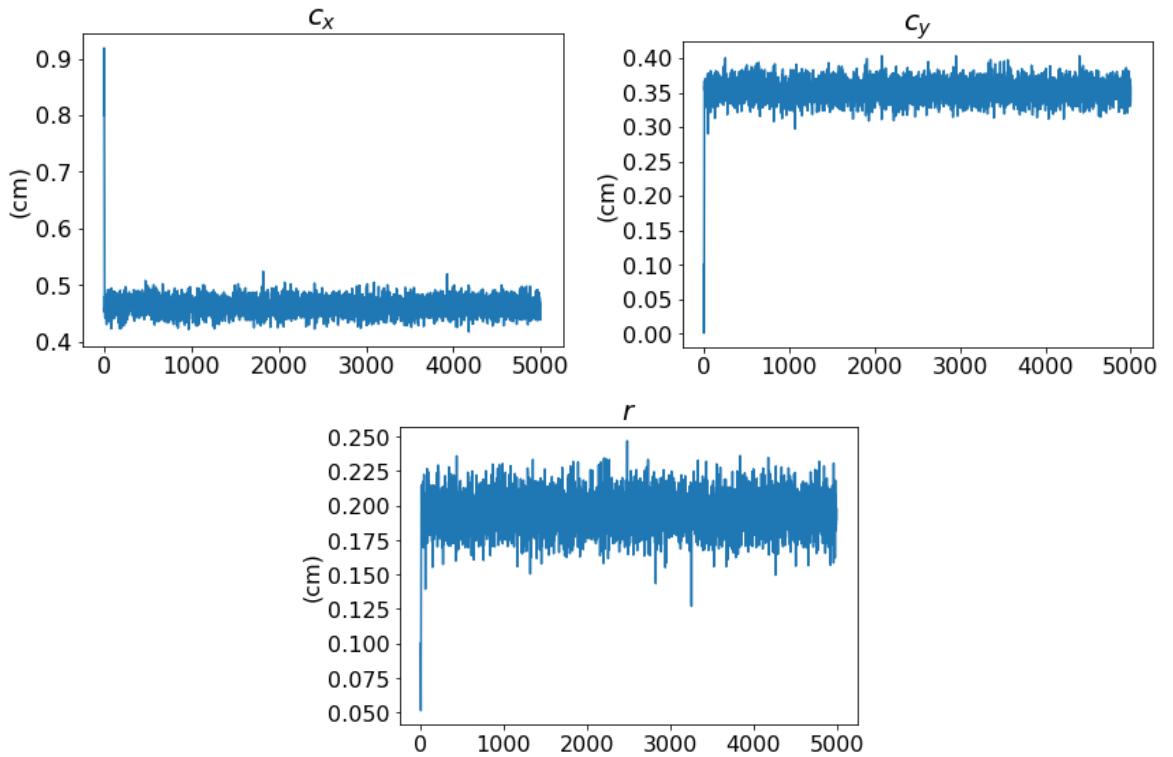


Figure 6.2: Parameter trace of 5000 samples for the pseudo-3D method in problem 0.

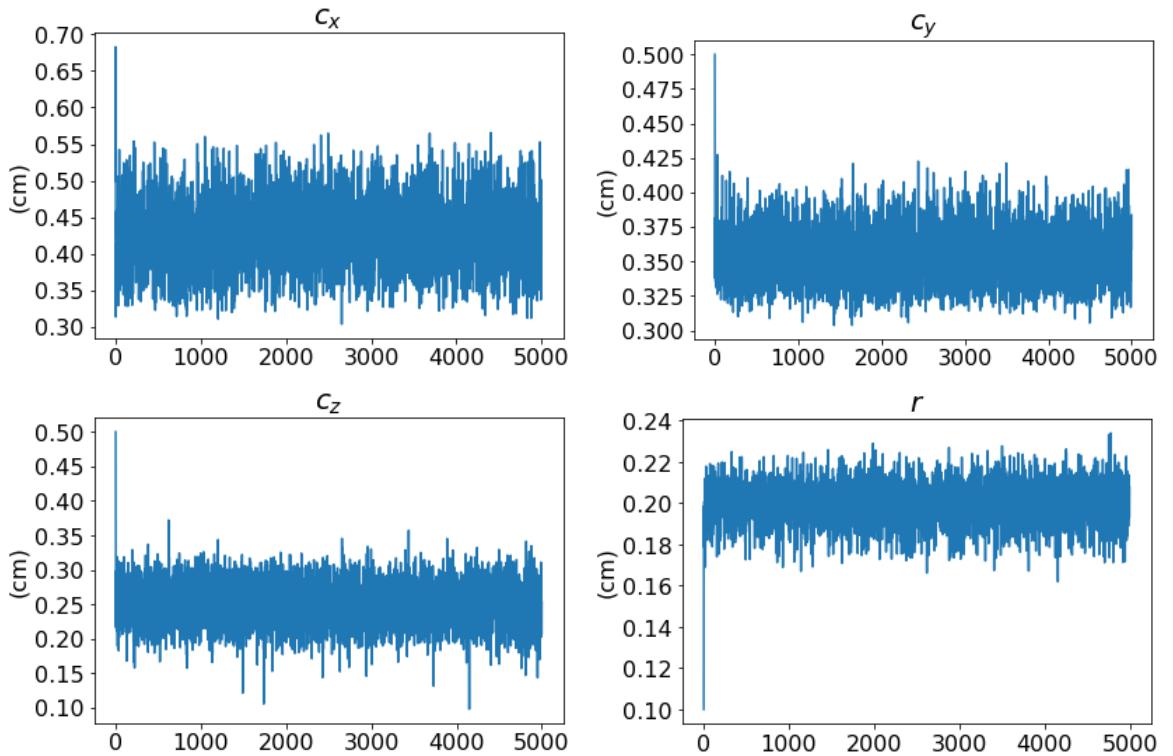


Figure 6.3: Parameter trace of 5000 samples for the full-3D method in problem 0.

Evidently, it takes the sampler a few iterations to converge to the posterior distribution. We discard the first 1000 samples as burn-in and perform diagnostics on the resulting chain of

4000 samples. The chains all look to be mixing well compared with the example of good mixing in the bottom right corner of figure 3.3. For each parameter, we compute the posterior mean $\mathbb{E}[\cdot]$, the integrated autocorrelation time (IACT), and the effective sample size n_{eff} . These are displayed in table 6.3. Recalling the true parameters in table 6.1 we see that the estimates for the pseudo-3D model are more accurate compared to those of the full-3D model. We note that the integrated auto-correlation time (IACT) is larger in the full-3D method which consequently, also leads to a smaller effective sample size n_{eff} . On the burned chains we compute 95% credibility

Diagnostic	Value	Diagnostic	Value
Center IACT	(1.008, 1.019)	Center IACT	(1.823, 1.176, 1.315)
Radius IACT	1.140	Radius IACT	1.561
Center n_{eff}	(3967, 3925)	Center n_{eff}	(2194, 3402, 3042)
Radius n_{eff}	3509	Radius n_{eff}	2562
$\mathbb{E}[c_k]$	(0.463, 0.353) cm	$\mathbb{E}[c_k]$	(0.428, 0.353, 0.250) cm
$\mathbb{E}[r_k]$	0.195 cm	$\mathbb{E}[r_k]$	0.200 cm

Table 6.3: Chain diagnostics for problem 0. Left: pseudo-3D method. Right: full-3D method.

intervals and plot them in figure 6.4 along with the true and estimated parameter values for the UQ analysis. We only display the results for the full-3D model here as the UQ results for the pseudo-3D model were shown in figure 5.2. All credibility intervals seem to contain the true and

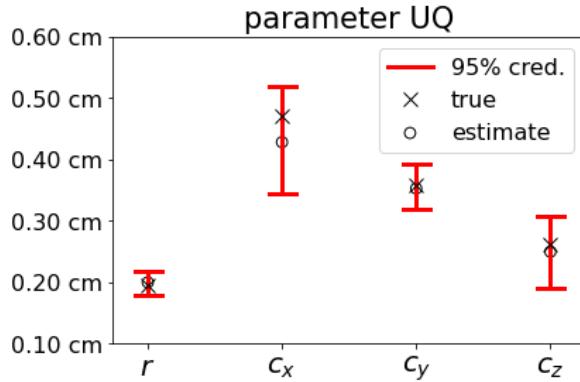


Figure 6.4: Parameter UQ analysis for problem 0. Left: pseudo-3D method. Right: full-3D method.

estimated parameter values. The method is able to accurately estimate the true parameters via the posterior mean. We see that the credibility intervals are a bit larger in the full-3D method, indicating more uncertainty in those estimates.

6.2.2 Conductivity reconstructions

After drawing and post-processing the samples (c_k, r_k) we can estimate the conductivity field by computing posterior means. As described in the previous section we do this in both parameter space and conductivity space. In conductivity space, we compute the parameterization push-forwards $\sigma(c_k, r_k)$ of all $k = 1, \dots, 4000$ samples on a $64 \times 64 \times 64$ grid for visualization purposes. We have already shown the results for the pseudo-3D method in figure 5.4 since we used the current case as a running example in the previous chapter. Results for the full-3D method are shown in figure 6.5.

The true conductivity field $\sigma(c, r)$ in the top left corner and the parameter mean pushforward $\sigma(\mathbb{E}[c_k], \mathbb{E}[r_k])$ in the top right seem to be nearly identical. Note how the uncertainty shows as a sphere of high standard deviation around the surface of the inclusion in the bottom row.

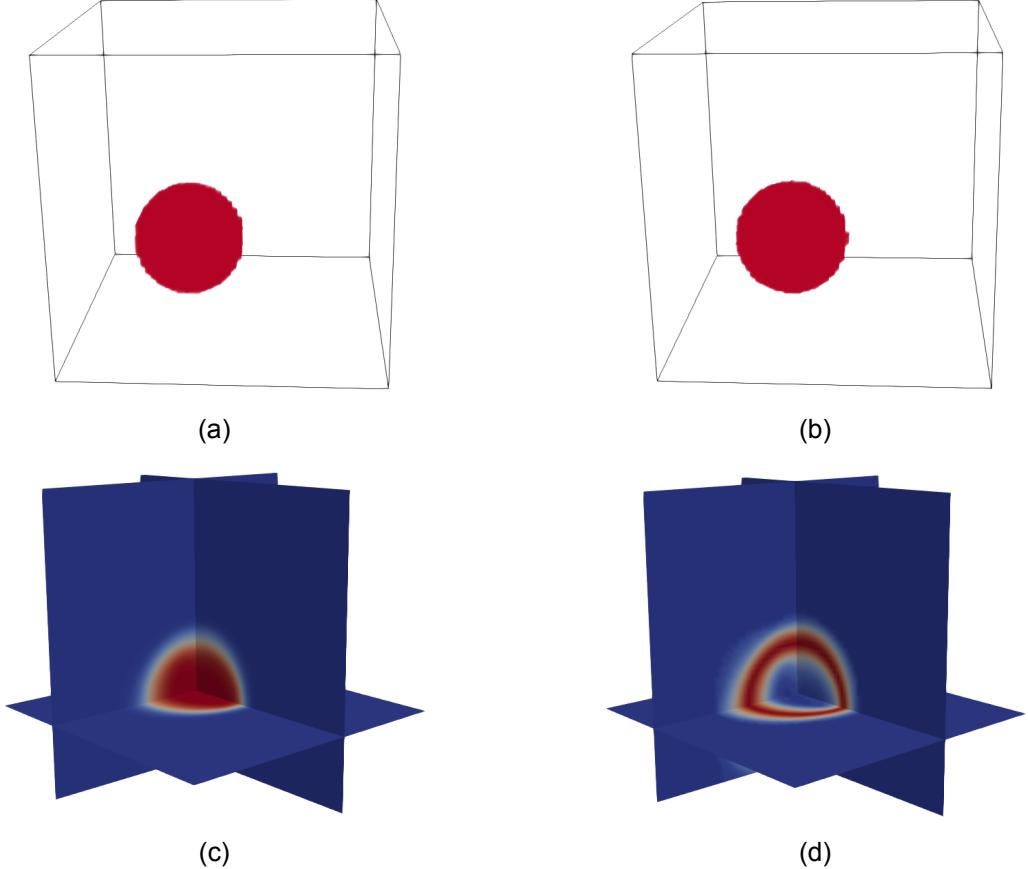


Figure 6.5: Full-3D reconstructions for problem 0. (a): true conductivity field. (b): pushforward of parameter sample means. (c): mean of pushforward of parameter samples. (d): standard deviation of pushforward of parameter samples.

6.3 Problem 1: two spherical inclusions with known conductivity

The problem of reconstructing only a single ball with known conductivity properties was the simplest proof-of-concept case we considered for our method. Next, we consider the case of reconstructing two inclusions with known conductivity properties. We thus have to estimate

$$\text{Pseudo-3D: } (c_1, c_2, r) = ((c_{1x}, c_{1y}), (c_{2x}, c_{2y}), (r_1, r_2)) \in \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2,$$

$$\text{Full-3D: } (c_1, c_2, r) = ((c_{1x}, c_{1y}, c_{1z}), (c_{2x}, c_{2y}, c_{2z}), (r_1, r_2)) \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^2.$$

For shorthand notation, we use $c = (c_1, c_2)$. The conductivity field is parameterized in full-3D by

$$\sigma(c, r) = \begin{cases} 2 \text{ S/m} & \text{if } (x - c_{11})^2 + (y - c_{12})^2 + (z - c_{13})^2 \leq r_1^2, \\ 2 \text{ S/m} & \text{if } (x - c_{21})^2 + (y - c_{22})^2 + (z - c_{23})^2 \leq r_2^2, \\ 1 \text{ S/m} & \text{otherwise.} \end{cases} \quad (6.6)$$

And similarly for the pseudo-3D model with the z -coordinate omitted. The parameters shown in table 6.4 were used to construct the true conductivity field and initialize the sampler.

We draw 5000 samples (c_k, r_k) using the Metropolis-within-Gibbs sampler in algorithm 1. We can use the same posterior conditional distributions as in eq. (5.10) and eq. (5.9) as for the previous case. Since we only have to sample another set of center coordinates and another radius, we can simply add more dimensions to the distributions which we sample from. We do not have to include another MH step in the Gibbs iteration to sample another variable independent

from (c, r) . The CPU times on the HPC system for both cases are shown in table 6.5. The times are comparable to the previous case shown in table 6.5 since we did not have to add another MH step inside the Gibbs iteration.

Parameter	Value	Parameter	Value
True center 1 c_1	(0.80, 0.80) cm	True center 1 c_1	(0.8, 0.8, 0.8)
True center 2 c_2	(0.30, 0.30) cm	True center 2 c_2	(0.3, 0.3, 0.3)
True radii (r_1, r_2)	(0.15, 0.20) cm	True radii (r_1, r_2)	(0.15, 0.2)
Conductivity κ	2 S/cm	Conductivity κ	2 S/cm
Data grid	$66 \times 66 \times 66$	Data grid	$10 \times 10 \times 10$
Reconstruction grid	$64 \times 64 \times 64$	Reconstruction grid	$8 \times 8 \times 8$
Initial center 1	(0.60, 0.60) cm	Initial center 1	(0.6, 0.6, 0.6)
Initial center 2	(0.10, 0.10) cm	Initial center 2	(0.1, 0.1, 0.1)
Initial radii	(0.10, 0.10) cm	Initial radii	(0.1, 0.1)
Stepsize (β_c, β_r)	(0.03, 0.03)	Stepsize (β_c, β_r)	(0.05, 0.07)
Bounds $(c_{\text{low}}, c_{\text{high}})$	(0, 1)	Bounds $(c_{\text{low}}, c_{\text{high}})$	(0, 1)
Bounds $(r_{\text{low}}, r_{\text{high}})$	(0.05, 0.50)	Bounds $(r_{\text{low}}, r_{\text{high}})$	(0.05, 0.5)
Within Gibbs iter.	20	Within Gibbs iter.	20

Table 6.4: Initial parameters for problem 1. Left: pseudo-3D method. Right: full-3D method.

	pseudo-3D method	full-3D method
CPU time	14.1 hours	6.6 hours

Table 6.5: CPU times on the HPC system for problem 1.

6.3.1 Statistical analysis of posterior samples

Trace plots of the chains for each parameter are shown for the pseudo-3D method in figure 6.6 and for the full-3D method in figure 6.8. We discard the first 1000 samples as burn-in and consider only the rest of the chain. For each chain, the posterior mean $\mathbb{E}[\cdot]$, the integrated auto-correlation time (IACT), and the effective sample size n_{eff} are computed. These are displayed in table 6.6. We see in the table that for this problem, the IACT is also the largest in full-3D. The

Diagnostic	Value	Diagnostic	Value
Center 1 c_1 IACT	(1.585, 1.665)	Center 1 c_1 IACT	(7.153, 3.427, 7.127)
Center 2 c_2 IACT	(1.634, 1.391)	Center 2 c_2 IACT	(6.258, 3.489, 5.545)
Radii (r_1, r_2) IACT	(2.081, 1.891)	Radii (r_1, r_2) IACT	(13.303, 10.145)
Center 1 c_1 n_{eff}	(2524, 2403)	Center 1 c_1 n_{eff}	(559, 1167, 561)
Center 2 c_2 n_{eff}	(2448, 2876)	Center 2 c_2 n_{eff}	(639, 1146, 721)
Radii (r_1, r_2) n_{eff}	(1922, 2115)	Radii (r_1, r_2) n_{eff}	(301, 394)
$\mathbb{E}[c_1^k]$	(0.792, 0.791) cm	$\mathbb{E}[c_1^k]$	(0.805, 0.791, 0.784) cm
$\mathbb{E}[c_2^k]$	(0.294, 0.300) cm	$\mathbb{E}[c_2^k]$	(0.258, 0.309, 0.300) cm
$\mathbb{E}[(r_1^k, r_2^k)]$	(0.149, 0.196) cm	$\mathbb{E}[(r_1^k, r_2^k)]$	(0.160, 0.195) cm

Table 6.6: Chain diagnostics for problem 1. Left: pseudo-3D method. Right: full-3D method.

95% credibility intervals are shown in figure 6.7 along with the true and estimated parameters. In the figure, we see how all true parameters fall within the credibility intervals, and the means of each chain of posterior samples seem to be good estimates of the true parameters. We again see that the credibility intervals are wider in the full-3D method which implies more uncertainty in the estimates. The trace plots in figures 6.6 for the pseudo-3D method indicate the chains

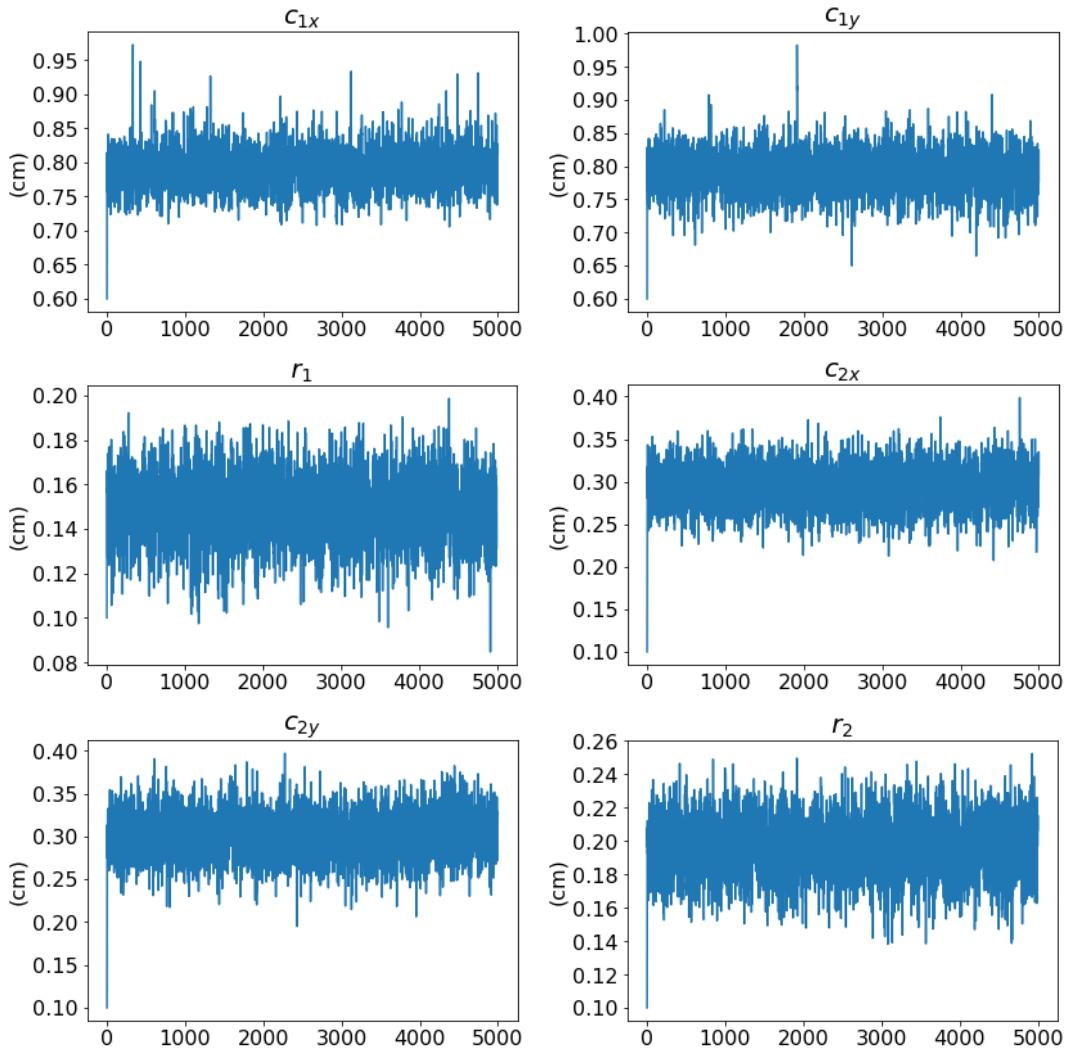


Figure 6.6: Parameter trace of 5000 samples for the pseudo-3D method in problem 1

are mixing well, as they have the desired "fuzzy caterpillar" look. For the full-3D method, however, the Markov chains seem to be more correlated, as the "fuzzy caterpillars" look less dense compared to pseudo-3D.

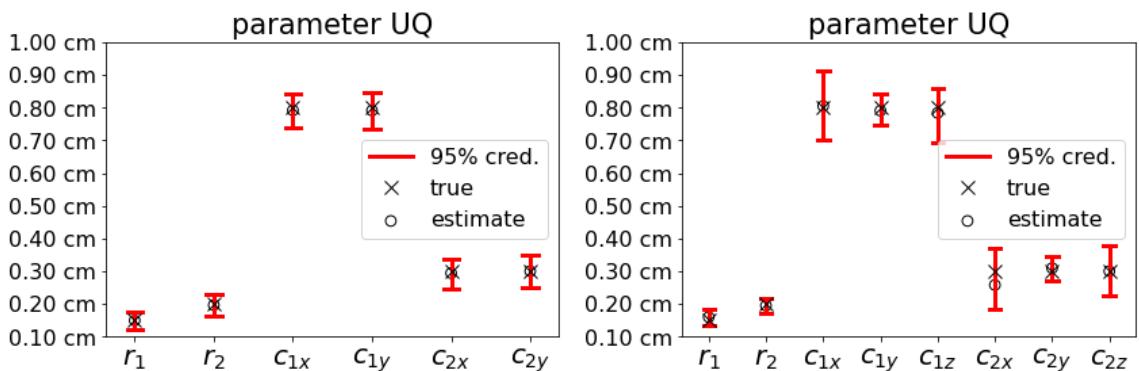


Figure 6.7: Parameter UQ analysis for problem 1. Left: pseudo-3D method. Right: full-3D method.

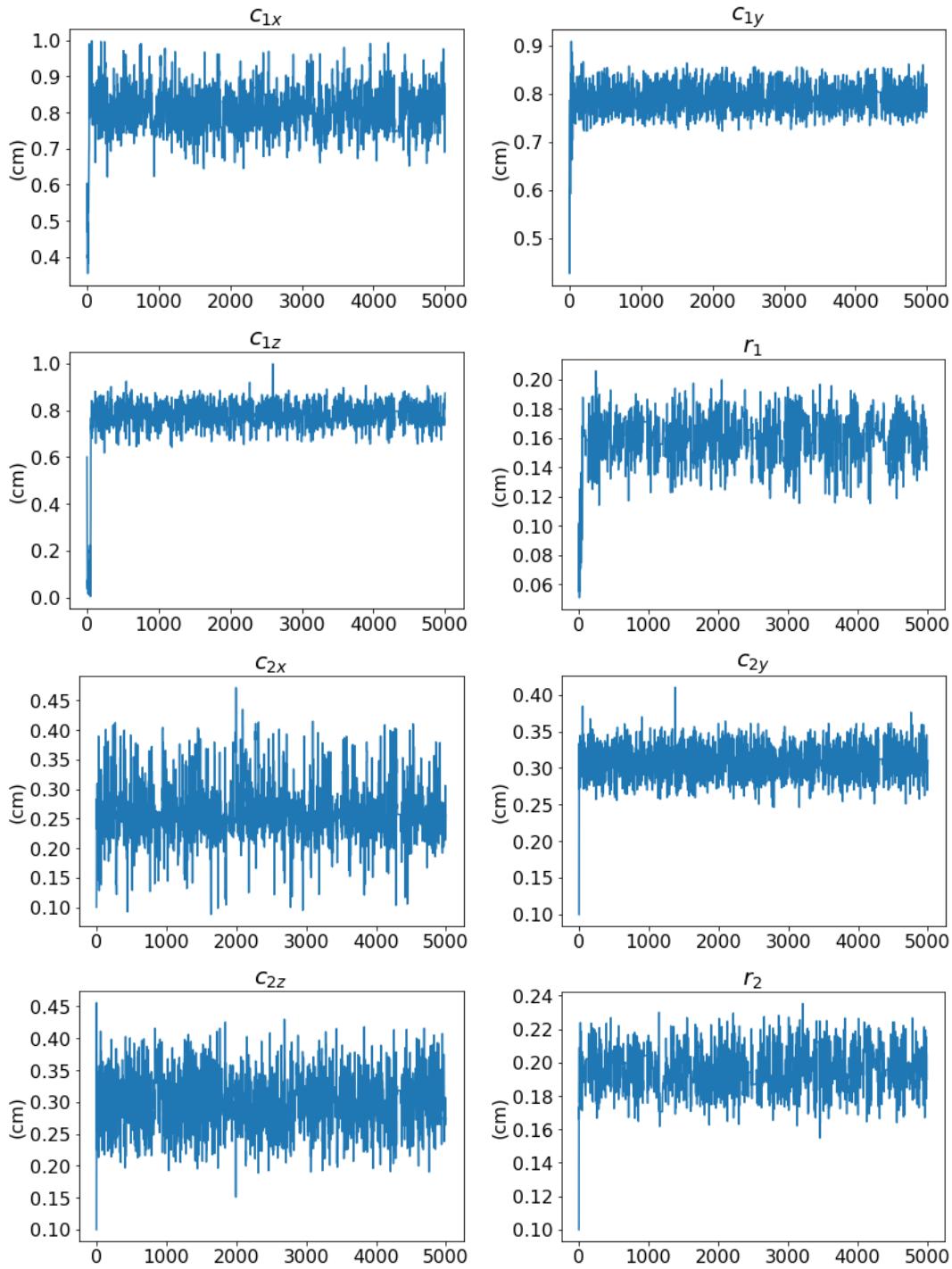


Figure 6.8: Parameter trace of 5000 samples for the full-3D method in problem 1

6.3.2 Conductivity reconstructions

We compute the means of the parameters and map the means to a conductivity field. We also compute the mean conductivity field generated by mapping all of the parameter samples. We also compute the standard deviation in conductivity space. The results are displayed in figures 6.9 and 6.10 for the pseudo-3D and full-3D methods respectively.

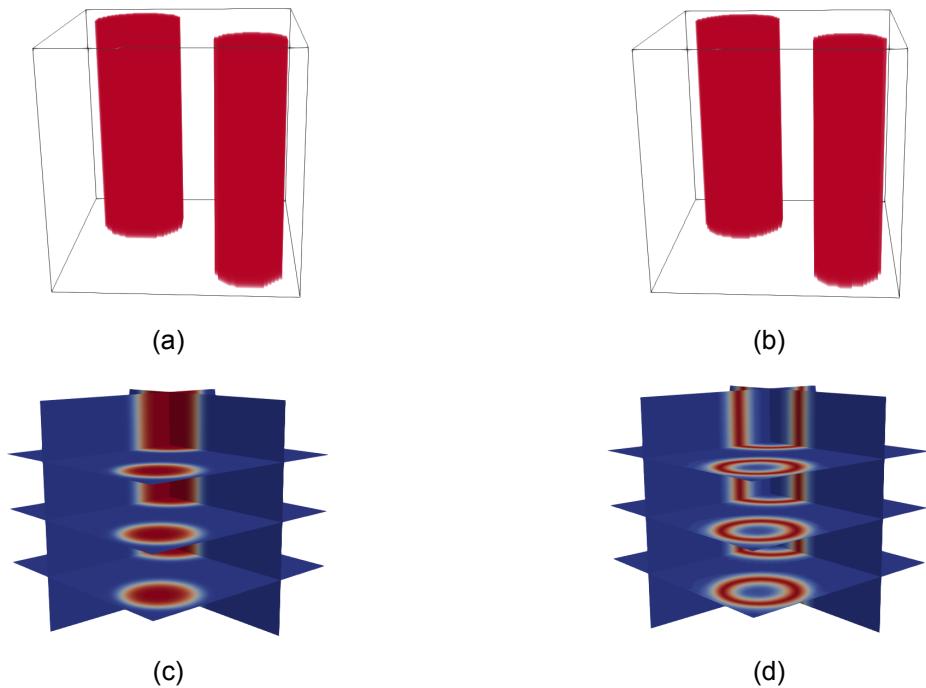


Figure 6.9: Pseudo-3D reconstructions for problem 1. (a): true conductivity field. (b): push-forward of parameter sample means. (c): mean of pushforward of parameter samples. (d): standard deviation of pushforward of parameter samples.

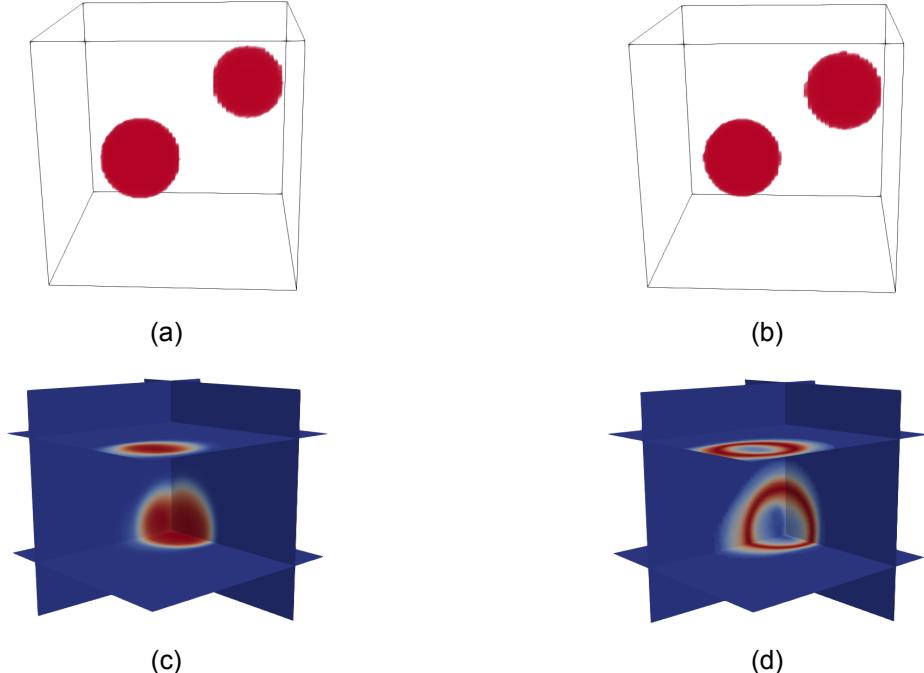


Figure 6.10: Full-3D reconstructions for problem 1. (a): true conductivity field. (b): pushforward of parameter sample means. (c): mean of pushforward of parameter samples. (d): standard deviation of pushforward of parameter samples.

The true conductivity field in the top left and the parameter mean pushforward $\sigma(\mathbb{E}[c_k], \mathbb{E}[r_k])$ in

the top right are close to identical. We see the same trend for the two inclusions as we did for the single inclusion in 5.4, where the uncertainty accumulates at the boundary of the inclusion.

6.4 Problem 2: rotated ellipsoidal inclusion with unknown conductivity

We now consider an ellipsoidal inclusion with unknown conductivity properties that has been rotated by 45° degrees about the z -axis. For the full-3D method, we have to sample $(c, r, \kappa, \delta) = ((c_x, c_y, c_z), (r_x, r_y, r_z), \kappa, \delta) \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R}$ and for the pseudo-3D method we sample $(c, r, \kappa, \delta) = ((c_x, c_y), (r_x, r_y), \kappa, \delta) \in \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R}$. The conductivity field can be parameterized by the following expression in full-3D

$$\sigma(c, r, \kappa, \delta) = \begin{cases} \kappa \text{ S/m} & \text{if } \frac{((x-c_x)\cdot\cos(\delta)+(y-c_y)\cdot\sin(\delta))^2}{r_x^2} + \frac{((y-c_y)\cdot\cos(\delta)-(x-c_x)\cdot\sin(\delta))^2}{r_y^2} + \frac{(z-c_z)^2}{r_z^2} \leq 1, \\ 1 \text{ S/m} & \text{otherwise.} \end{cases} \quad (6.7)$$

With κ being the conductivity level of the inclusion and δ being the rotation angle in the xy -plane about the z -axis. The pseudo-3D method is similar but with the last term involving the z -coordinate dropped. In the particular case of a rotated ellipse, the solution to the reconstruction problem will not be unique unless we impose constraints. Consider an ellipse parameterized by $(c, r) \in \mathbb{R}^2 \times \mathbb{R}^2$ with $c = (c_x, c_y)$ and $r = (r_x, r_y)$, and rotate it by $90^\circ = \frac{\pi}{2}$. We can then build an identical ellipse by instead using $r = (r_y, r_x)$ in the parameterization, as rotation by 90° switches the semi-minor and semi-major axes. To remedy this non-uniqueness problem we introduce a new constraint to the Metropolis-within-Gibbs sampler

$$\text{New constraint: } r_x < r_y. \quad (6.8)$$

We also use the bounds $\delta \in (0, \pi)$ as a constraint on the rotation angle. The parameters shown in table 6.7 were used to construct the true conductivity field and initialize the sampler. We draw

Parameter	Value	Parameter	Value
True center c	(0.43, 0.48) cm	True center c	(0.43, 0.48, 0.41) cm
True radius r	(0.15, 0.30) cm	True radius r	(0.15, 0.30, 0.20) cm
True conductivity κ	2 S/cm	True conductivity κ	2 S/cm
True angle δ	$45 \cdot \frac{\pi}{180} \approx 0.785$	True angle δ	$45 \cdot \frac{\pi}{180} \approx 0.785$
Data grid	$66 \times 66 \times 66$	Data grid	$10 \times 10 \times 10$
Reconstruction grid	$64 \times 64 \times 64$	Reconstruction grid	$8 \times 8 \times 8$
Initial center c_0	(0.1, 0.1) cm	Initial center c_0	(0.1, 0.1, 0.1) cm
Initial radius r_0	(0.1, 0.1) cm	Initial radius r_0	(0.1, 0.1, 0.1) cm
Init. conductivity κ_0	1 S/cm	Init. conductivity κ_0	1 S/cm
Initial angle δ_0	0	Initial angle δ_0	0
Step $(\beta_c, \beta_r, \beta_\kappa, \beta_\delta)$	(0.03, 0.04, 0.05, 0.04)	Step $(\beta_c, \beta_r, \beta_\kappa, \beta_\delta)$	(0.03, 0.04, 0.05, 0.04)
Bounds $(c_{\text{low}}, c_{\text{high}})$	(0, 1) cm	Bounds $(c_{\text{low}}, c_{\text{high}})$	(0, 1) cm
Bounds $(r_{\text{low}}, r_{\text{high}})$	(0.05, 0.5) cm	Bounds $(r_{\text{low}}, r_{\text{high}})$	(0.05, 0.50) cm
Bounds $(\kappa_{\text{low}}, \kappa_{\text{high}})$	(0, 100) S/cm	Bounds $(\kappa_{\text{low}}, \kappa_{\text{high}})$	(0, 100) S/cm
Bounds $(\delta_{\text{low}}, \delta_{\text{high}})$	(0, π)	Bounds $(\delta_{\text{low}}, \delta_{\text{high}})$	(0, π)
Constraint	$r_1 < r_2$ for $r \in \mathbb{R}^2$	Constraint	$r_1 < r_2$ for $r \in \mathbb{R}^3$
Within Gibbs iter.	20	Within Gibbs iter.	20

Table 6.7: Initial parameters. Left: pseudo-3D method. Right: full-3D method.

5000 samples $(c_k, r_k, \kappa_k, \delta_k)$ using a variation of the Metropolis-within-Gibbs sampler described in algorithm 1. This time we have to modify the posterior conditional distributions in (5.10) and

(5.9) to make them fit the current case. This is not complicated as the conditional distribution w.r.t. one variable is simply the joint posterior with all the other variables fixed. The new log posterior conditionals in the current case are

$$\begin{aligned}\log \pi(c|r, \kappa, \delta, y_{\text{obs}}) &= -\frac{1}{2} \frac{\|y_{\text{obs}} - (\mathcal{G} \circ \mathcal{F})(\sigma(c, r, \kappa, \delta))\|^2}{\gamma^2}, \quad (r, \kappa, \delta) \text{ fixed}, \\ \log \pi(r|c, \kappa, \delta, y_{\text{obs}}) &= -\frac{1}{2} \frac{\|y_{\text{obs}} - (\mathcal{G} \circ \mathcal{F})(\sigma(c, r, \kappa, \delta))\|^2}{\gamma^2}, \quad (c, \kappa, \delta) \text{ fixed}, \\ \log \pi(\kappa|c, r, \delta, y_{\text{obs}}) &= -\frac{1}{2} \frac{\|y_{\text{obs}} - (\mathcal{G} \circ \mathcal{F})(\sigma(c, r, \kappa, \delta))\|^2}{\gamma^2}, \quad (c, r, \delta) \text{ fixed}, \\ \log \pi(\delta|c, r, \kappa, y_{\text{obs}}) &= -\frac{1}{2} \frac{\|y_{\text{obs}} - (\mathcal{G} \circ \mathcal{F})(\sigma(c, r, \kappa, \delta))\|^2}{\gamma^2}, \quad (c, r, \kappa) \text{ fixed}.\end{aligned}\tag{6.9}$$

As we now have double the amount of variables to sample we need to add two more MH steps to the Metropolis-within-Gibbs sampler described in section 5.3.1. This will inevitably make the sampling take at least double the amount of time compared to the previous case, and also use much more memory. The CPU times on the HPC system for system are shown in table 6.8

	pseudo-3D method	full-3D method
CPU time	53 hours	35.1 hours

Table 6.8: CPU times on the HPC system for system for problem 2.

6.4.1 Statistical analysis of posterior samples

Trace plots of the chains for each parameter are shown for the pseudo-3D method in figure 6.12 and for the full-3D method in figure 6.13. We discard the first 1000 samples as burn-in and perform diagnostics on the resulting chain of 4000 samples. For each parameter, we compute the posterior mean $\mathbb{E}[\cdot]$, the integrated autocorrelation time (IACT), and the effective sample size n_{eff} . These are displayed in table 6.9. We also compute the 95% credibility intervals and

Diagnostic	Value	Diagnostic	Value
Center IACT	(1.417, 1.321)	Center IACT	(3.614, 1.890, 3.783)
Radius IACT	(16.20, 11.00)	Radius r IACT	(31.27, 20.45, 23.54)
Conductivity IACT	39.33	Conductivity IACT	92.68
Angle IACT	17.29	Angle IACT	77.78
Center n_{eff}	(2822, 3027)	Center n_{eff}	(1107, 2116, 1057)
Radius n_{eff}	(247, 364)	Radius n_{eff}	(128, 196, 170)
Conductivity n_{eff}	102	Conductivity n_{eff}	43
Angle n_{eff}	231	Angle n_{eff}	51
$\mathbb{E}[c_k]$	(0.420, 0.478) cm	$\mathbb{E}[c_k]$	(0.402, 0.475, 0.410) cm
$\mathbb{E}[(r_k)]$	(0.145, 0.295) cm	$\mathbb{E}[r_k]$	(0.172, 0.300, 0.182) cm
$\mathbb{E}[(\kappa_k)]$	2.149 S/cm	$\mathbb{E}[(\kappa_k)]$	2.407 S/cm
$\mathbb{E}[(\delta_k)]$	0.743	$\mathbb{E}[(\delta_k)]$	0.865

Table 6.9: Chain diagnostics for problem 2. Left: pseudo-3D method. Right: full-3D method.

plot them in figure 6.11 along with the true and estimated parameter values for the UQ analysis.

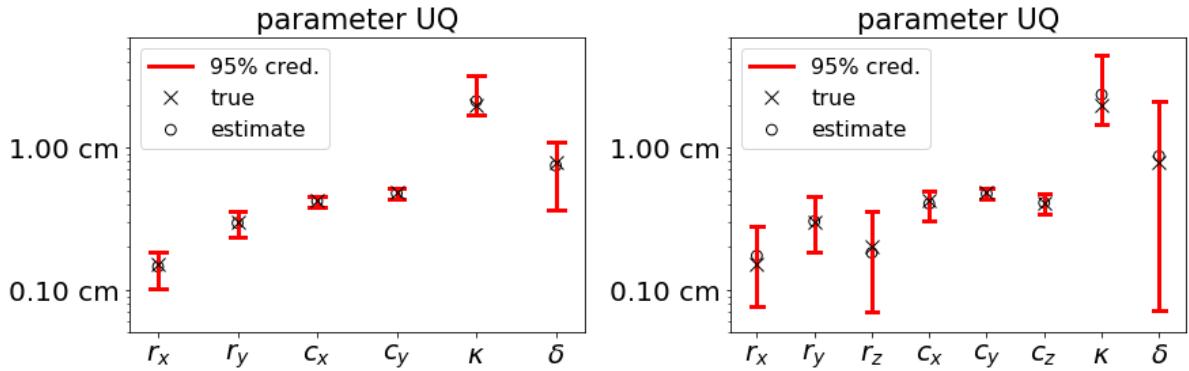


Figure 6.11: Parameter UQ analysis for problem 2. Left: pseudo-3D method. Right: full-3D method.

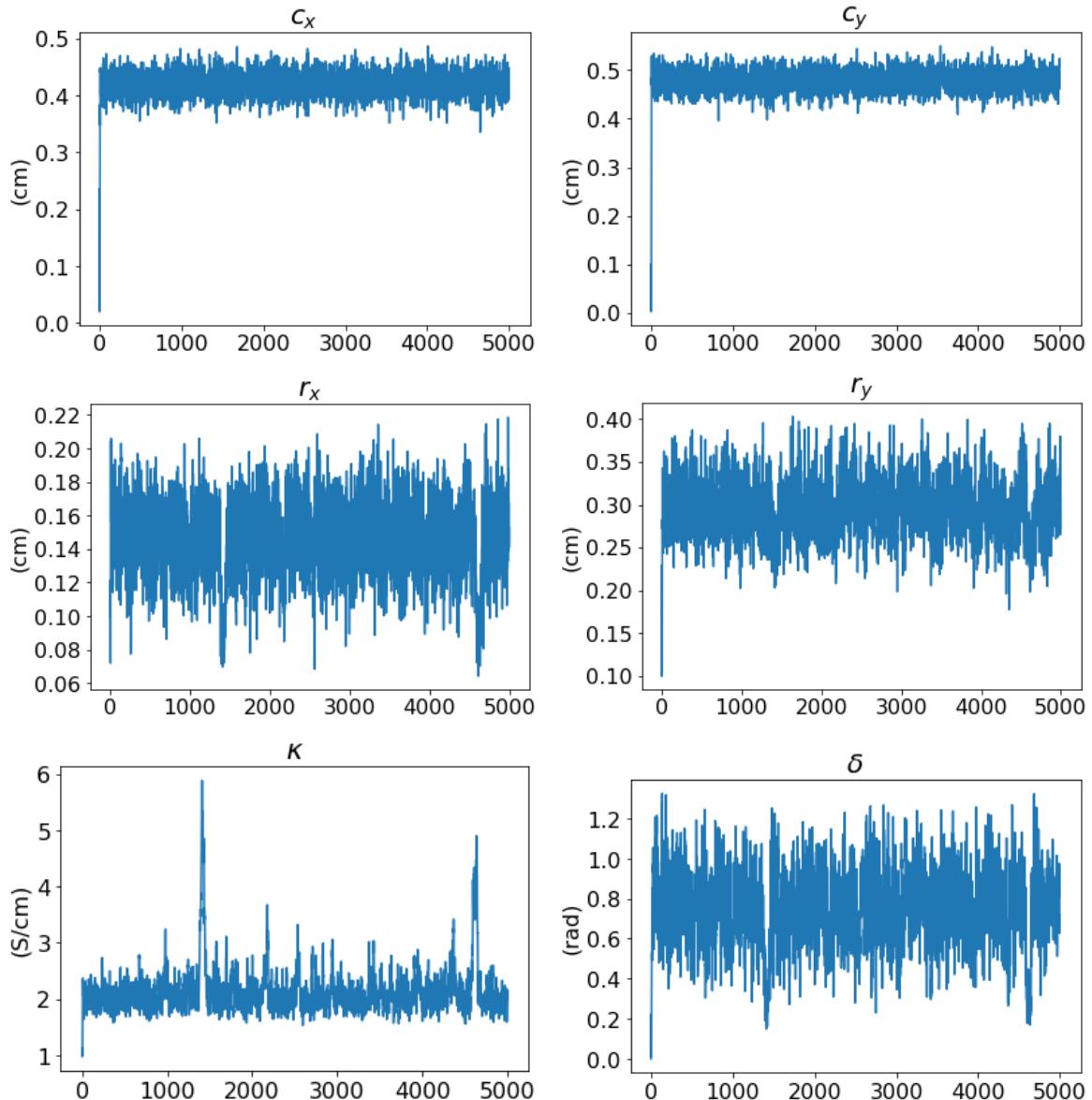


Figure 6.12: Parameter trace of 5000 samples for the pseudo-3D method in problem 2.

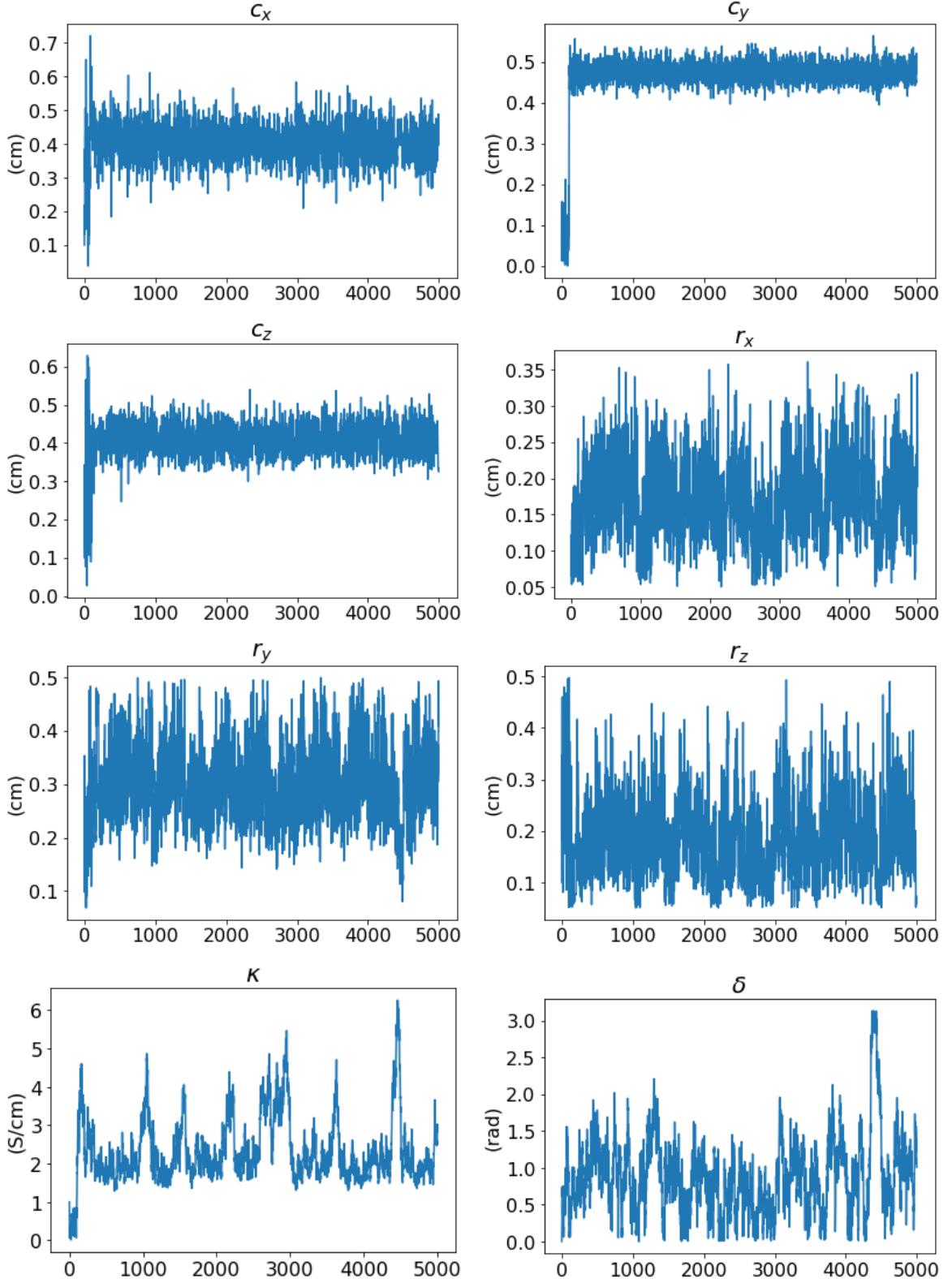


Figure 6.13: Parameter trace of 5000 samples for the full-3D method in problem 2.

Figure 6.11 reveals that all the true parameters are contained within the 95% credibility intervals and that the estimated parameters are not far off. However, the credibility intervals for the full-3D method are much larger compared to the pseudo-3D method. We note that the credi-

bility intervals for the x -component of the radius r_x and the rotation angle δ are quite wide for both the pseudo-3D model and the full-3D model. We already touched upon how there are some non-uniqueness issues between these two parameters, which we propose to remedy by introducing the constraint. As this issue was expected we are not surprised to see it reflected in the sampling. In the trace plots in figures 6.12 and 6.13 we see that the chains for the center coordinate c are mixing very well in both cases, resembling the fuzzy caterpillar behavior. However, the traces for the radius parameter r , conductivity value κ , and rotation angle δ are not as dense, indicating bad mixing of the chains for these parameters. The poorer sampling for (r, κ, δ) is also evident from the chain diagnostics shown in table 6.9, where the IACT is somewhat small only for the center coordinates. For this problem, we also see a difference in the performance of our method between the pseudo-3D and the full-3D method. However, this difference in performance between the pseudo-3D and the full-3D method is no coincidence as we shall discuss later.

6.4.2 Conductivity reconstructions

After post-processing the samples $(c_k, r_k, \kappa_k, \delta_k)$ we estimate the conductivity field by computing posterior means. Results for the pseudo-3D problem are shown in figure 6.14 and results for the full-3D method are shown in figure 6.15. We again see a high standard deviation around the boundary of the inclusion, as was the trend for the previous results. This time however since the conductivity value κ is also assumed to be unknown, we also see a high standard deviation inside the inclusion. Each sample κ_k is a little different and this introduces the interior uncertainty when we compute the standard deviation over a set of samples.

We note that similar results can be achieved by grouping the angle parameter δ with the radius parameter r into a single shape parameter $\phi = (r, \delta)$, as the two cannot be said to be fully independent due to the non-uniqueness issues we discussed earlier. This is handled by instead sampling $(c, \phi, \kappa) \in \mathbb{R}^3 \times \mathbb{R}^4 \times \mathbb{R}$ for the full-3D method, and $(c, \phi, \kappa) \in \mathbb{R}^2 \times \mathbb{R}^3 \times \mathbb{R}$ for the pseudo-3D method.

6.5 Problem 3: elliptic inclusion in pure 2D with unknown conductivity

We consider the pure 2D problem of estimating the defining parameters of a non-rotated elliptic inclusion with an unknown conductivity value, given only a single slice of magnetic field data. Since the pseudo-3D method is invariant in the z -direction the same information is contained in each vertical slice. We are simulating the case where the MRI machine only takes one cross-sectional scan of the object to be imaged. The parameterization of the inclusion is

$$\sigma(c, r, \kappa) = \begin{cases} \kappa \text{ S/m} & \text{if } \frac{(x-c_1)^2}{r_1^2} + \frac{(y-c_2)^2}{r_2^2} \leq 1, \\ 1 \text{ S/m} & \text{otherwise.} \end{cases} \quad (6.10)$$

We have to estimate 5 parameters $(c, r, \kappa) = ((c_x, c_y), (r_x, r_y), \kappa) \in \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}$. The log conditional densities that we use in the Metropolis-within-Gibbs sampler in algorithm 1 are

$$\begin{aligned} \log \pi(r|c, \kappa, y_{\text{obs}}) &= -\frac{1}{2} \frac{\|y_{\text{obs}} - (\mathcal{G} \circ \mathcal{F})(\sigma(c, r, \kappa))\|^2}{\gamma^2}, \quad (c, \kappa) \text{ fixed,} \\ \log \pi(c|r, \kappa, y_{\text{obs}}) &= -\frac{1}{2} \frac{\|y_{\text{obs}} - (\mathcal{G} \circ \mathcal{F})(\sigma(c, r, \kappa))\|^2}{\gamma^2}, \quad (r, \kappa) \text{ fixed,} \\ \log \pi(\kappa|c, r, y_{\text{obs}}) &= -\frac{1}{2} \frac{\|y_{\text{obs}} - (\mathcal{G} \circ \mathcal{F})(\sigma(c, r, \kappa))\|^2}{\gamma^2}, \quad (c, r) \text{ fixed.} \end{aligned} \quad (6.11)$$

The parameters shown in table 6.10 were used to construct the true conductivity field and initialize the sampler. We draw 5000 posterior samples (c_k, r_k, κ_k) . With those parameters, the CPU time was 18.8 hours.

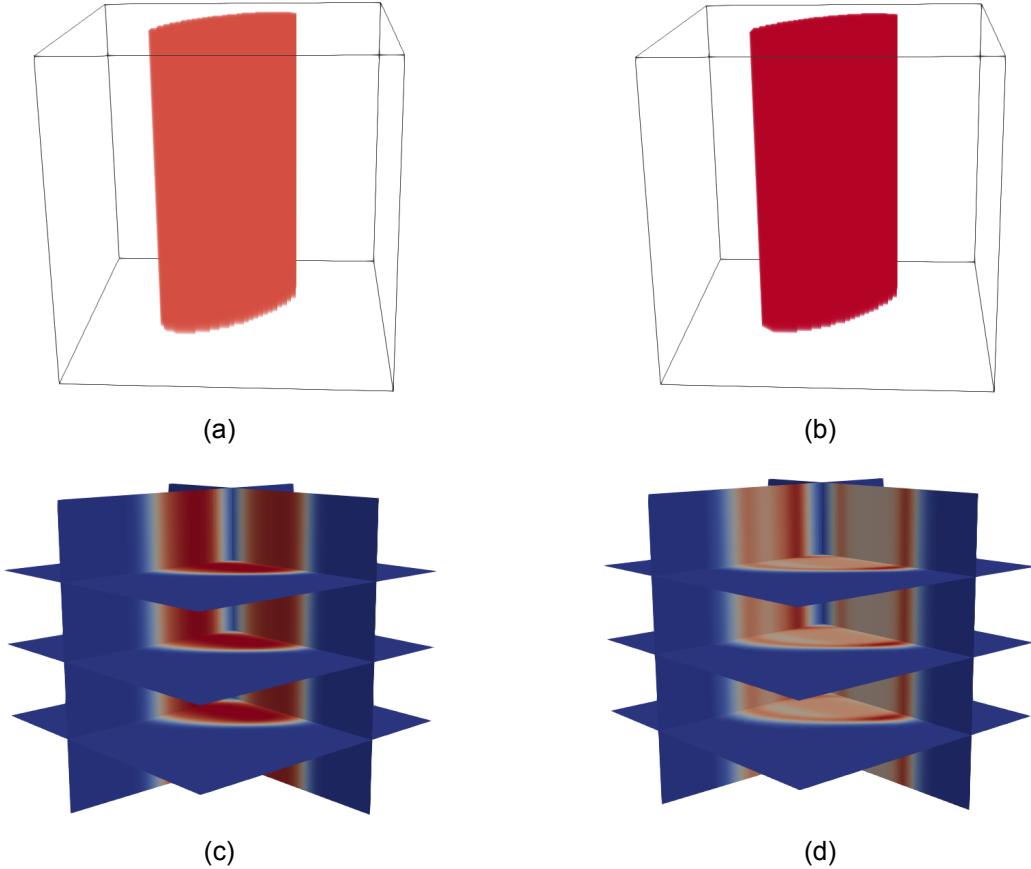


Figure 6.14: Pseudo-3D reconstructions for problem 2. (a): true conductivity field. (b): push-forward of parameter sample means. (c): mean of pushforward of parameter samples. (d): standard deviation of pushforward of parameter samples.

Parameter	Value
True center c	(0.5, 0.5) cm
True radius r	(0.2, 0.3) cm
True conductivity κ	2 S/cm
Data grid	$66 \times 66 \times 66$
Reconstruction grid	$64 \times 64 \times 64$
Initial center c_0	(0.1, 0.1) cm
Initial radius r_0	(0.1, 0.1) cm
Init. conductivity κ_0	1 S/cm
Stepsize $(\beta_c, \beta_r, \beta_\kappa)$	(0.03, 0.04, 0.05)
Bounds $(c_{\text{low}}, c_{\text{high}})$	(0, 1) cm
Bounds $(r_{\text{low}}, r_{\text{high}})$	(0.05, 0.5) cm
Bounds $(\kappa_{\text{low}}, \kappa_{\text{high}})$	(0, 100) S/cm
Within Gibbs iter.	20

Table 6.10: Initial parameters for pure 2D case in problem 3.

6.5.1 Statistical analysis of posterior samples

Trace plots of the chains for each parameter are shown for the pure 2D case in figure 6.16. We discard the first 1000 samples as burn-in and perform diagnostics on the resulting chain

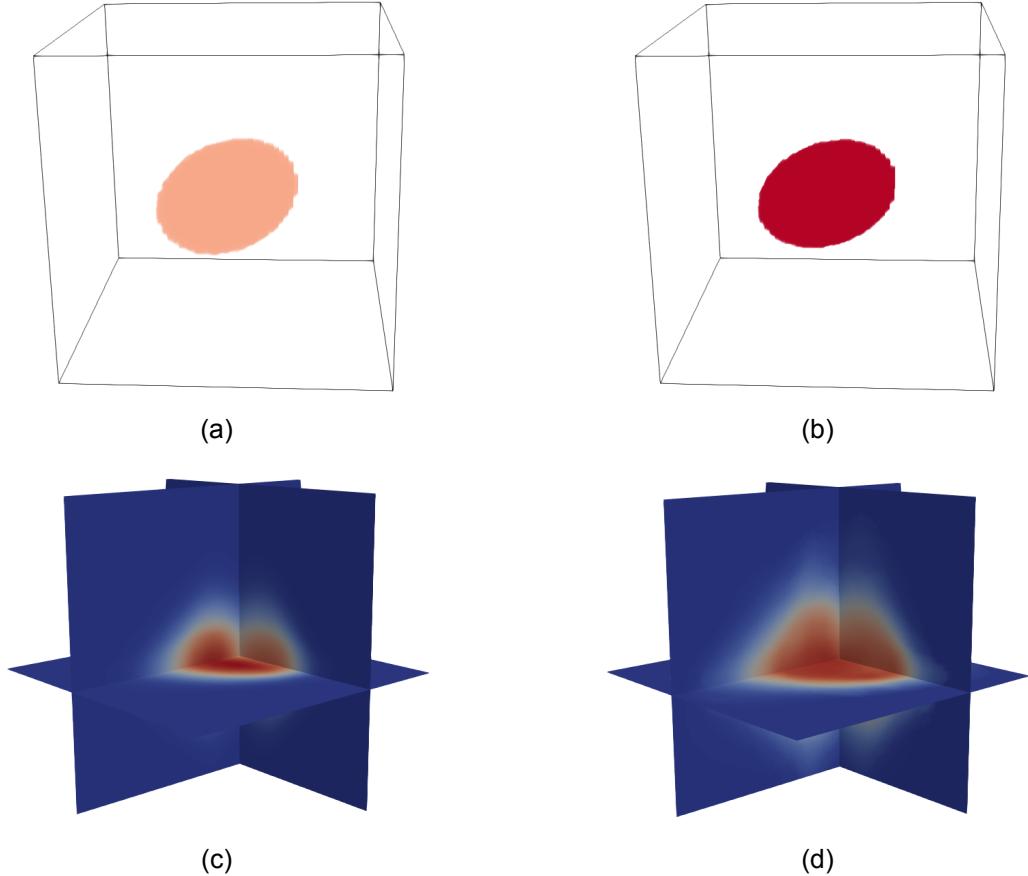


Figure 6.15: Full-3D reconstructions for problem 2. (a): true conductivity field. (b): pushforward of parameter sample means. (c): mean of pushforward of parameter samples. (d): standard deviation of pushforward of parameter samples.

of 4000 samples. For each parameter, we compute the posterior mean $\mathbb{E}[\cdot]$, the integrated autocorrelation time (IACT), and the effective sample size n_{eff} . These are displayed in table 6.11. We see that the IACT for the x -component of the radius parameter r_x and the conductivity value κ is much larger than the IACT of the other parameters. The trace plot for the x -component of the radius in figure 6.16 also reveal that the sampling went poorly, likewise with the trace plot for the conductivity value κ . In figure 6.17 we display the 95% credibility intervals and plot them with the true and estimated parameters. We note that the credibility intervals for r_x and κ are much larger compared to the others, which indicate higher uncertainty for these two parameters.

Diagnostic	Value
Center IACT	(1.136, 1.122)
Radius IACT	(51.51, 1.925)
Conductivity IACT	95.00
Center n_{eff}	(3520, 3565)
Radius n_{eff}	(78, 2078)
Conductivity n_{eff}	42
$\mathbb{E}[c_k]$	(0.490, 0.493) cm
$\mathbb{E}[r_k]$	(0.176, 0.302) cm
$\mathbb{E}[\kappa_k]$	2.353 S/cm

Table 6.11: Chain diagnostics for the pure 2D case in problem 3.

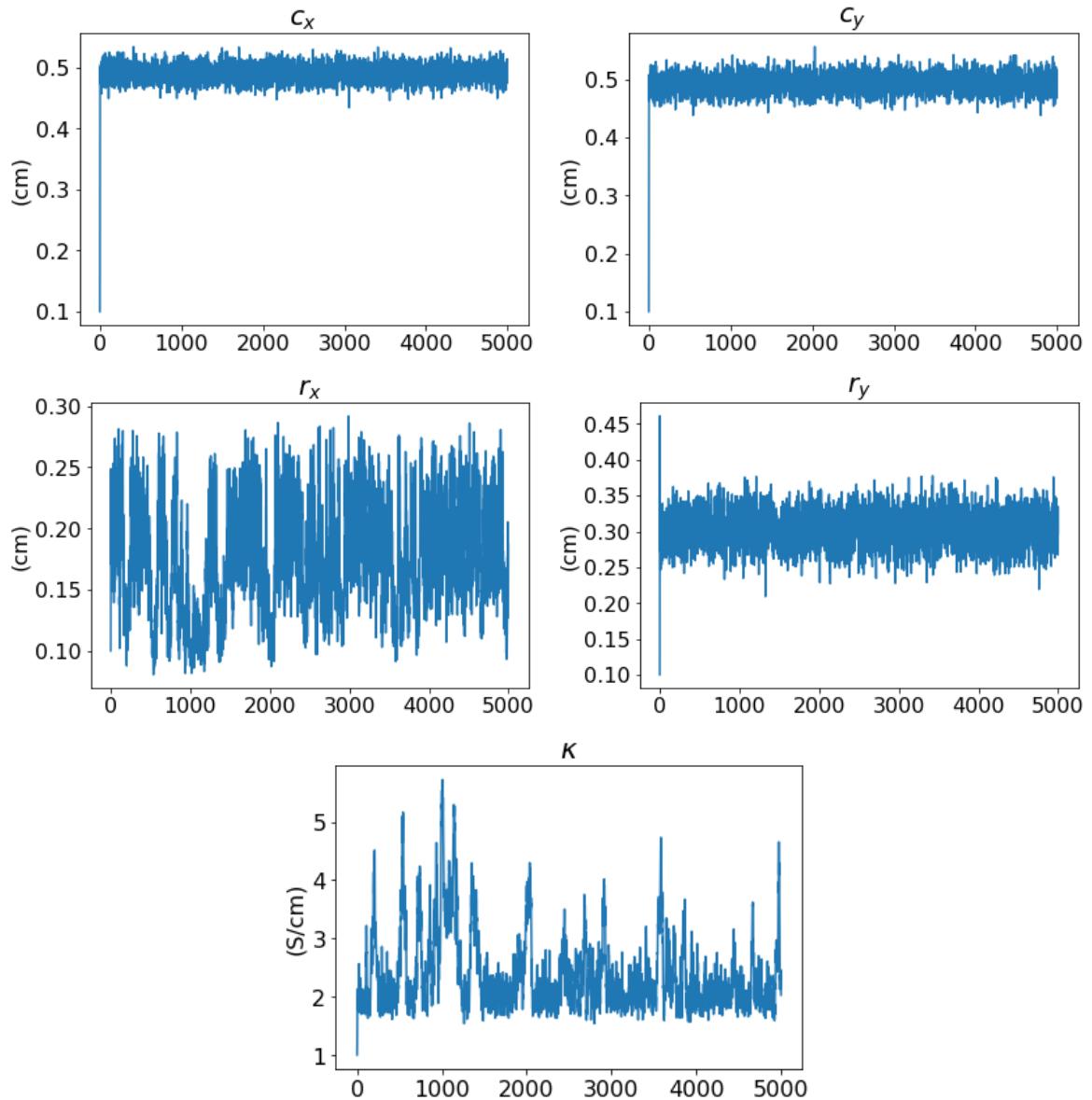


Figure 6.16: Parameter trace of 5000 samples for the pure 2D case in problem 3.

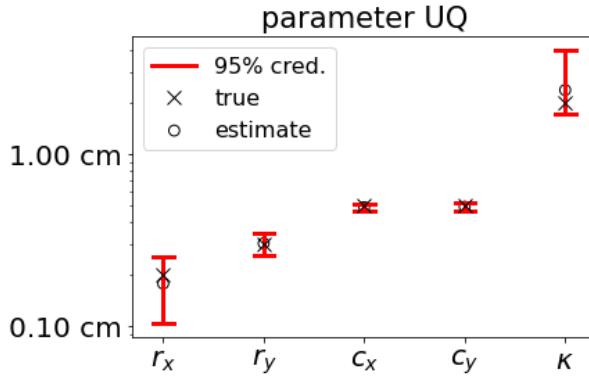


Figure 6.17: Parameter UQ analysis for the pure 2D case in problem 3.

6.5.2 Conductivity reconstructions

After drawing and post-processing the samples (c_k, r_k, κ_k) we can estimate the conductivity field by computing posterior means. We do this in both parameter space and conductivity space. Results for the pure 2D problem are shown in figure 6.18 where we plot a single slice in the z -direction. As the conductivity value parameter κ is also an unknown in this problem we expect nonzero standard deviation inside the inclusion, as well as around the boundary of the inclusion. This seems also to be the case in figure 6.18.

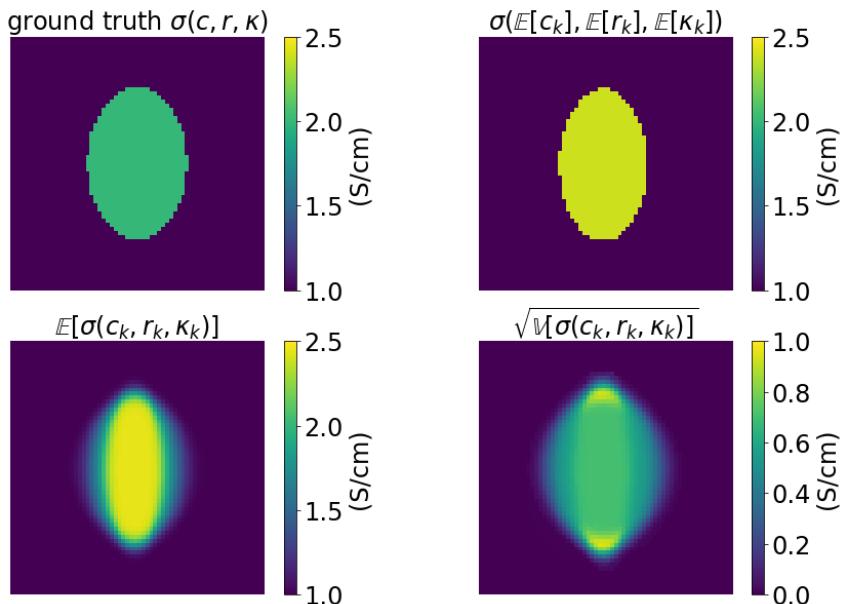


Figure 6.18: Pure 2D reconstructions for problem 3. Top left: true conductivity field. Top right: pushforward of parameter sample means. Bottom left: mean of pushforward of parameter samples. Bottom right: standard deviation of pushforward of parameter samples.

To the bottom right, we see that the standard deviation $\sqrt{V[\sigma(c_k, r_k, \kappa_k)]}$ extends further in the x -direction than in the y -direction when comparing with the mapped parameter means to the bottom left $E[\sigma(c_k, r_k, \kappa_k)]$. One can say that the "band of uncertainty" is thicker in the x -direction than in the y -direction. This makes sense when we have figure 6.17 in mind. The credibility intervals for the r_x estimate are wider than those for the r_y estimate, so more uncertainty in the x -direction is expected. Recall that we used the boundary condition $u = x$ on $\partial\Omega$ in (5.12). We discuss later if this has an influence on the wide band of uncertainty along the x -axis.

6.6 Problem 4: sparse magnetic field data in z -direction

When an MRI scan is performed multiple cross-sectional images of the object in question are taken and sampled together as one. For the setup described in figure 2.1 this implies that the magnetic field data is sparse in the z -direction, meaning that the resolution in the z -direction is coarser than in the xy -plane [9, 10]. The case of sparse z -data is simulated numerically by only including every 2nd or every 3rd slice of the magnetic field data in the z -direction, thus simulating 50% and 33% available B_z data. In figure 6.18 in the previous section, we showed that good results can be achieved in pure 2D with only a single slice of magnetic field data. As the pseudo-3D method is z -invariant it is not interesting to consider a sparse z -discretization of that case, hence we omit it. We consider the same rotated ellipsoidal inclusion as in problem 2, parameterized by the expression in equation (6.7). We are thus going to be sampling 8 parameters $(c, r, \kappa, \delta) = ((c_x, c_y, c_z), (r_x, r_y, r_z), \kappa, \delta) \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R}$. The parameters shown in table 6.12 were used to construct the true conductivity field and initialize the sampler. We

Parameter	Value
True center c	$(0.43, 0.48, 0.41)$ cm
True radius r	$(0.15, 0.30, 0.20)$ cm
True conductivity κ	2 S/cm
True angle δ	$45 \cdot \frac{\pi}{180} \approx 0.785$
Data grid	$10 \times 10 \times 10$
Reconstruction grid	$8 \times 8 \times 8$
Initial center c_0	$(0.5, 0.5, 0.5)$ cm
Initial radius r_0	$(0.1, 0.1, 0.1)$ cm
Init. conductivity κ_0	1 S/cm
Initial angle δ_0	0
Step $(\beta_c, \beta_r, \beta_\kappa, \beta_\delta)$	$(0.03, 0.04, 0.05, 0.04)$
Bounds $(c_{\text{low}}, c_{\text{high}})$	$(0, 1)$ cm
Bounds $(r_{\text{low}}, r_{\text{high}})$	$(0.05, 0.50)$ cm
Bounds $(\kappa_{\text{low}}, \kappa_{\text{high}})$	$(0, 100)$ S/cm
Bounds $(\delta_{\text{low}}, \delta_{\text{high}})$	$(0, \pi)$
Constraint	$r_1 < r_2$ for $r \in \mathbb{R}^2$
Within Gibbs iter.	20

Table 6.12: Initial parameters for problem 4. The same initial parameters were used for initializing the sampler for both the 50% data and 33% data case.

draw 5000 posterior samples $(c_k, r_k, \kappa_k, \delta_k)$ from the same posterior conditional distributions as in equation (6.9) using the sampler in algorithm 1. The only difference is that the forward model in the current case only returns partial B_z data. The CPU times on the HPC system are shown in table 6.13. We expect that the 50% data case takes a bit longer as there is more data available.

	50% data case	33% data case
CPU time	35.4 hours	35.2 hours

Table 6.13: CPU time on the HPC for problem 4.

6.6.1 Statistical analysis of posterior samples

Trace plots of the chains for each parameter are shown for the 50% B_z data case in figure 6.20 and the 33% B_z data case in figure 6.21. Judging from the mixing of each Markov chain it seems that the sampling for the center coordinates c went very well. The sampling for the radius

parameters was a bit worse, while the sampling for the conductivity value κ and rotation angle δ went poorly. Judging from the trace plots of the rotation angle parameter δ it seems that the chain visits several modes of the distribution. Just as in problem 2 (figures 6.12 and 6.13) the sampler has some issues when it comes to estimating the conductivity level κ of the inclusion and its rotation angle δ . As expected the mixing looks better for the 50% data case since more information is available. Different inclusions could look more similar in the 33% data case due to more information being left out. We discard the first 1000 samples as burn-in and perform diagnostics on the resulting chain of 4000 samples. The posterior mean $\mathbb{E}[\cdot]$, the integrated autocorrelation time (IACT), and the effective sample size n_{eff} are computed and displayed in table 6.14.

Diagnostic	Value	Diagnostic	Value
Center IACT	(3.165, 1.786, 2.836)	Center IACT	(14.00, 3.884, 4.879)
Radius IACT	(11.59, 13.33, 24.87)	Radius r IACT	(59.72, 27.25, 25.00)
Conductivity IACT	56.12	Conductivity IACT	218.5
Angle IACT	109.5	Angle IACT	265.6
Center n_{eff}	(1264, 2239, 1411)	Center n_{eff}	(286, 1030, 820)
Radius n_{eff}	(345, 300, 160)	Radius n_{eff}	(67, 147, 160)
Conductivity n_{eff}	71	Conductivity n_{eff}	18
Angle n_{eff}	37	Angle n_{eff}	15
$\mathbb{E}[c_k]$	(0.403, 0.476, 0.412) cm	$\mathbb{E}[c_k]$	(0.412, 0.474, 0.410) cm
$\mathbb{E}[(r_k)]$	(0.184, 0.300, 0.196) cm	$\mathbb{E}[r_k]$	(0.168, 0.296, 0.181) cm
$\mathbb{E}[(\kappa_k)]$	2.170 S/cm	$\mathbb{E}[(\kappa_k)]$	2.506 S/cm
$\mathbb{E}[(\delta_k)]$	1.168	$\mathbb{E}[(\delta_k)]$	1.467

Table 6.14: Chain diagnostics for problem 4. Left: 50% B_z data. Right: 33% B_z data.

As the integrated auto-correlation times show it is easier for our method to draw independent samples of the center coordinates than the other parameters, independent of how much data is used. The full data case was problem 2, and from table 6.9 we see similar results with the IACT for the pseudo-3D and the full-3D method. We compute 95% credibility intervals and plot them in figure 6.19 along with the true and estimated parameters. The credibility intervals contain the true parameters in all cases, and it is only the x -component of the radius parameter r_x and the rotation angle δ which is poorly estimated in both partial data cases. This issue was also present in problem 2, see figure 6.11. We believe it is caused by the non-uniqueness issues we had to remedy with the new constraint in section 6.4. Lastly, we note that the chain diagnostics in table 6.14 clearly show that the 50% data case performs better than the 33% data case.

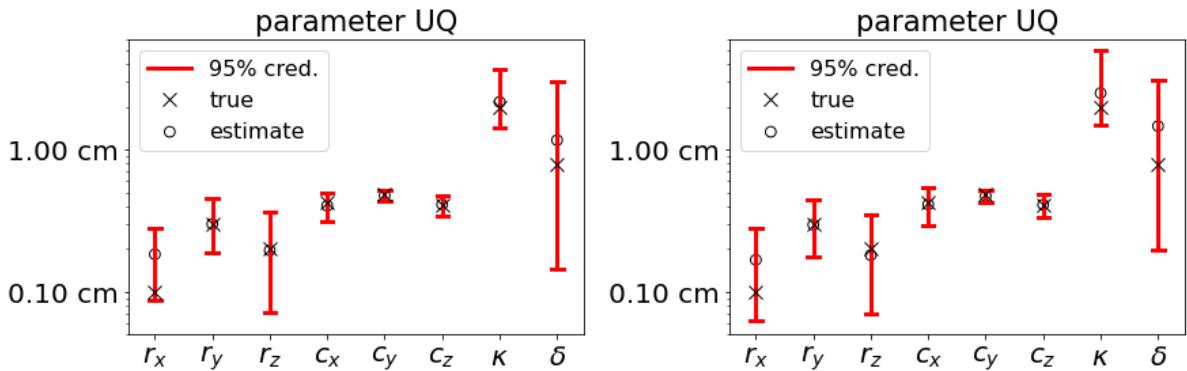


Figure 6.19: Parameter UQ analysis for problem 4. Left: 50% available B_z data. Right: 33% available B_z data.

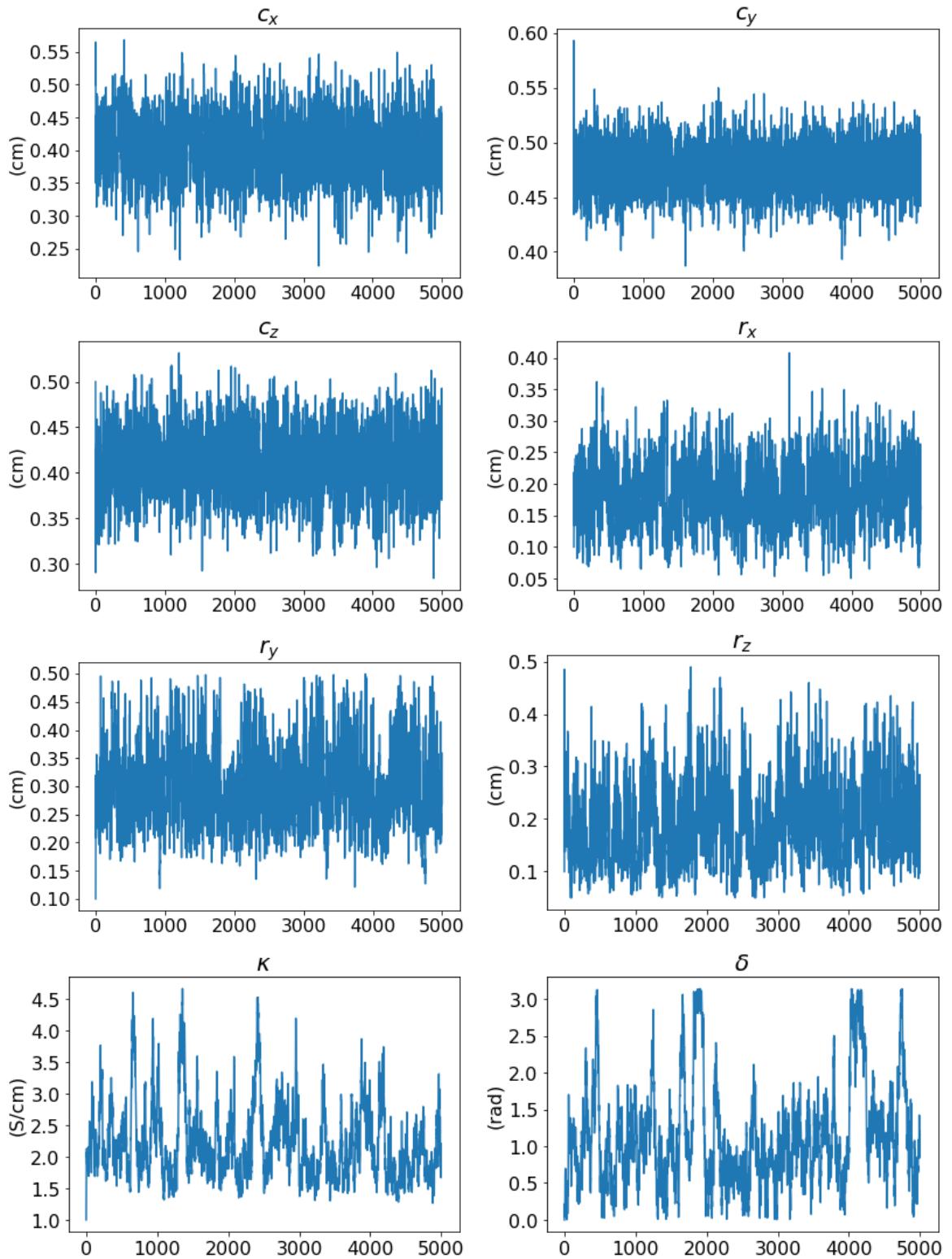


Figure 6.20: Parameter trace of 5000 samples for the 50% available B_z data case in problem 4.

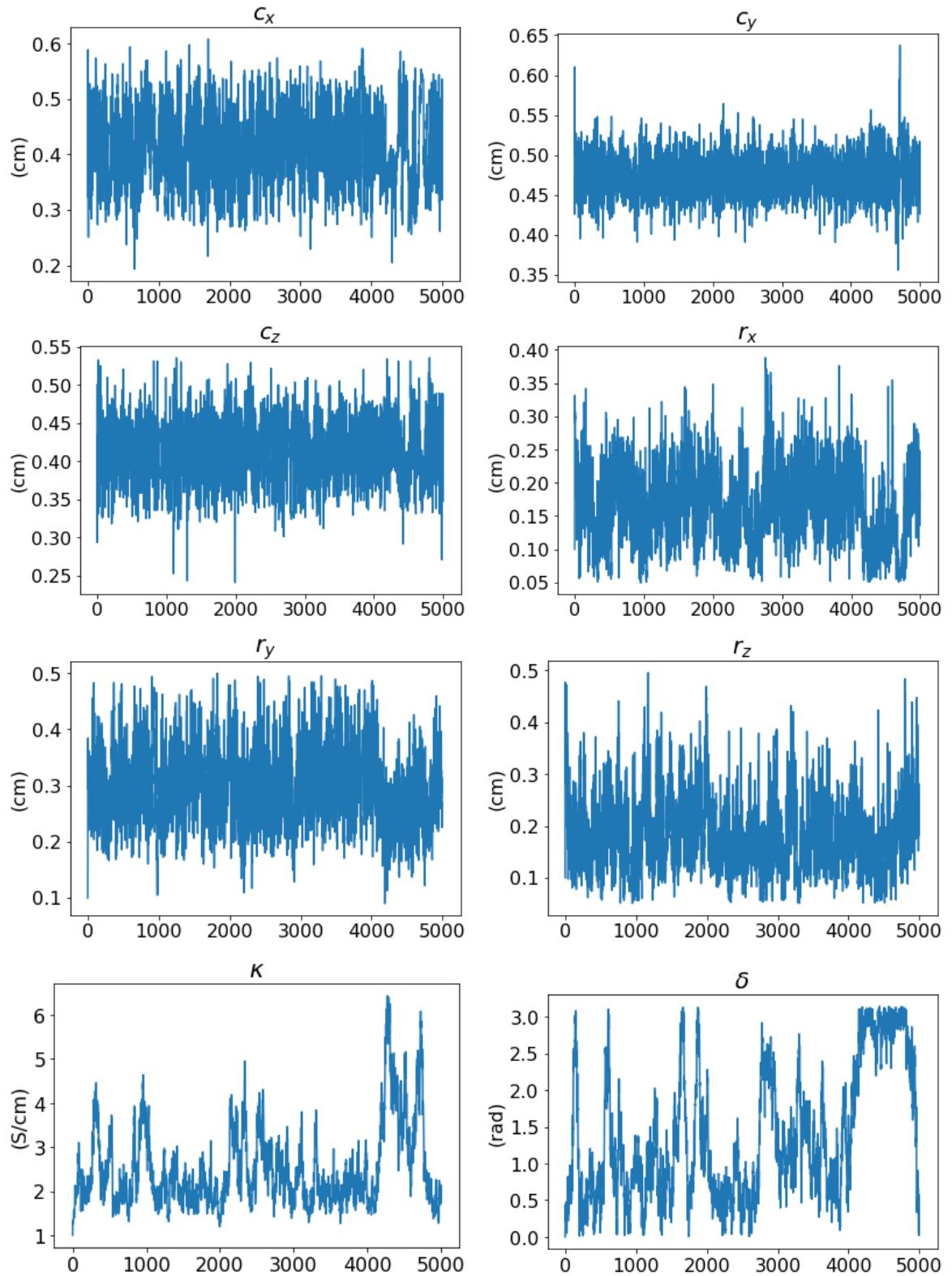


Figure 6.21: Parameter trace of 5000 samples for the 33% available B_z data case in problem 4.

6.6.2 Conductivity reconstructions

After post-processing the samples $(c_k, r_k, \kappa_k, \delta_k)$ for the two different z -resolution cases, we estimate the conductivity field by the posterior means. Results for the 50% B_z data case are shown in figure 6.22 and results for the 33% B_z data case are shown in figure 6.23. The two posterior means for the 50% and 33% data cases look quite similar. Comparing the top row of each figure it can be seen that there is a larger difference between the true and estimated conductivity values for the 33% data case. Comparing the bottom right plot of each figure, we see that there is a larger area of uncertainty with a higher standard deviation around the inclusion for the 33% data case.

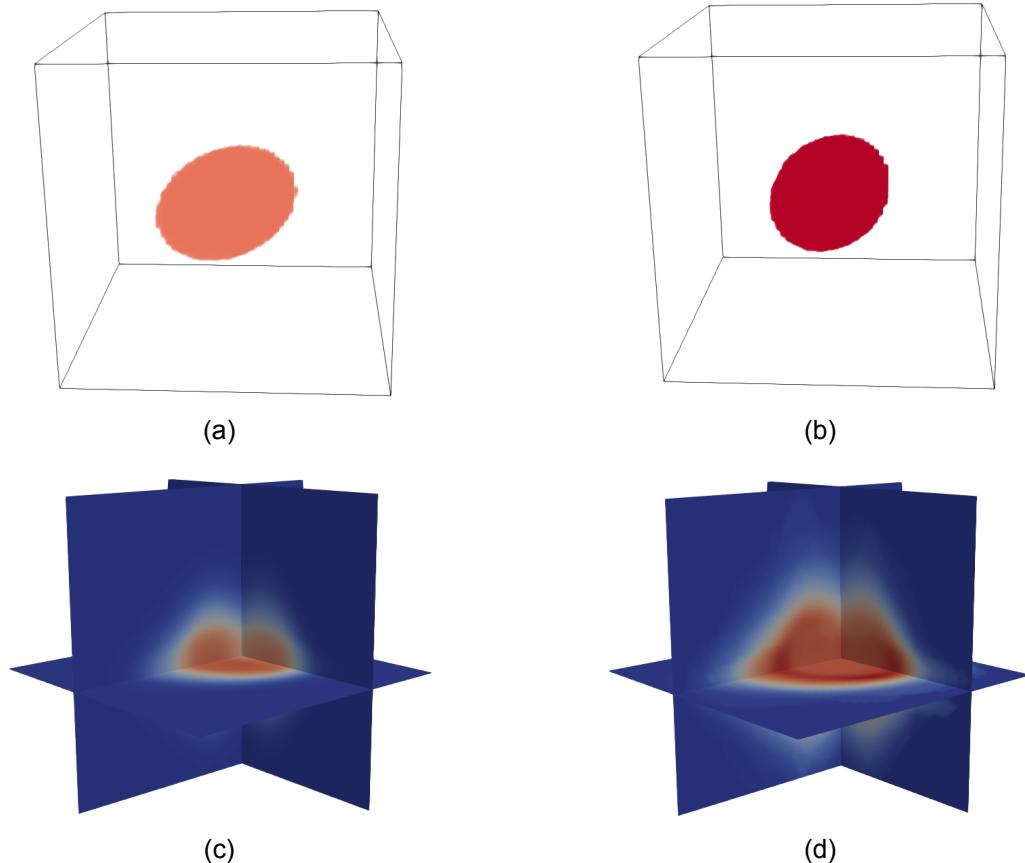


Figure 6.22: 50% data reconstructions for problem 4. (a): true conductivity field. (b): push-forward of parameter sample means. (c): mean of pushforward of parameter samples. (d): standard deviation of pushforward of parameter samples.

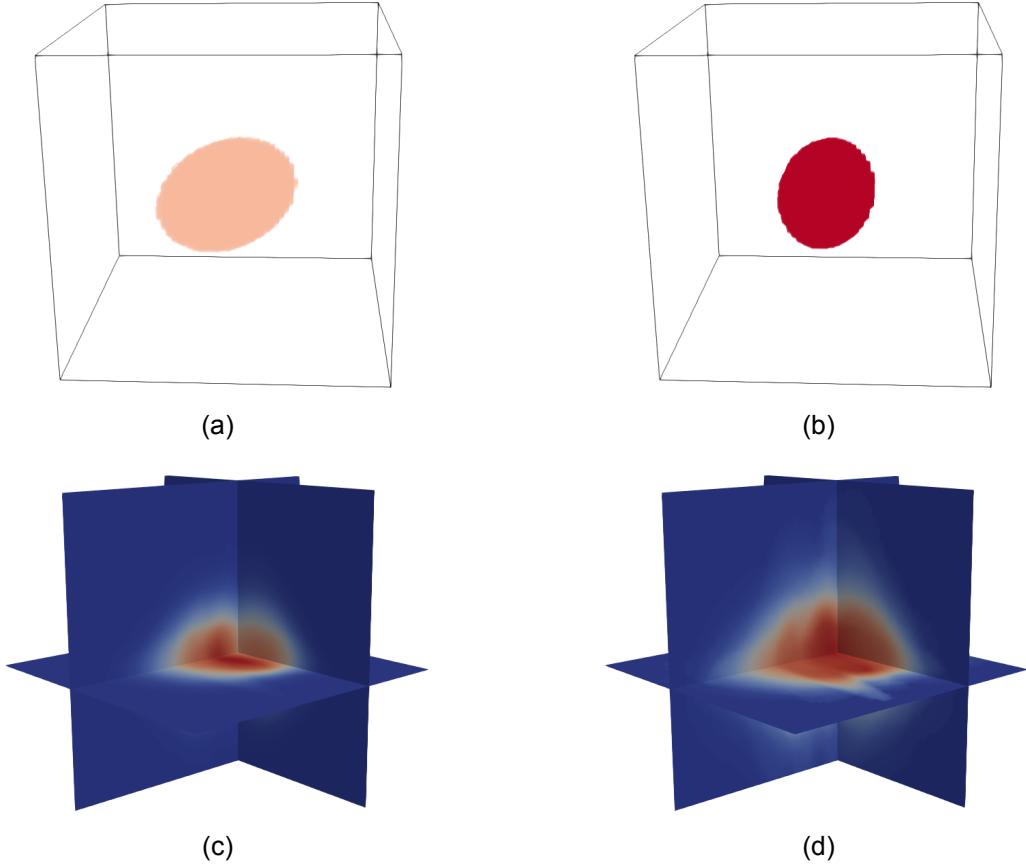


Figure 6.23: 33% data reconstructions for problem 4. (a): true conductivity field. (b): push-forward of parameter sample means. (c): mean of pushforward of parameter samples. (d): standard deviation of pushforward of parameter samples.

6.7 Problem 5: two inclusions with large conductivity contrast

Consider the problem of estimating the conductivity levels of two inclusions where one is more conducting than the other. The motivation for including this experiment is that the conductivity properties of healthy and malignant tissue are different. One paper considers the reconstruction problem in EIT with inclusions shaped as balls with conductivity levels 10 times higher than the background [34]. The parameters for this problem are the center coordinates c , the radius r , and the conductivity level κ for each of the two inclusions. We thus have to estimate

$$\text{Pseudo-3D: } (c_1, c_2, r, \kappa) = ((c_{1x}, c_{1y}), (c_{2x}, c_{2y}), (r_1, r_2), (\kappa_1, \kappa_2)) \in \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2,$$

$$\text{Full-3D: } (c_1, c_2, r, \kappa) = ((c_{1x}, c_{1y}, c_{1z}), (c_{2x}, c_{2y}, c_{2z}), (r_1, r_2), (\kappa_1, \kappa_2)) \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^2 \times \mathbb{R}^2.$$

The parameterization below defines the conductivity field. We use $c = (c_1, c_2)$ for brevity

$$\sigma(c, r, \kappa) = \begin{cases} \kappa_1 \text{ S/m} & \text{if } (x - c_{1x})^2 + (y - c_{1y})^2 + (z - c_{1z})^2 \leq r_1^2, \\ \kappa_2 \text{ S/m} & \text{if } (x - c_{2x})^2 + (y - c_{2y})^2 + (z - c_{2z})^2 \leq r_2^2, \\ 1 \text{ S/m} & \text{otherwise.} \end{cases} \quad (6.12)$$

Where (κ_1, κ_2) are chosen to create high contrast between the two inclusions and the background conductivity. We choose the conductivity levels $(\kappa_1, \kappa_2) = (3, 10)$ S/cm to simulate the event that one inclusion is a cancerous anomaly, given a background conductivity of 1 S/cm. The inclusion with index 2 is the malignant one that we hope to be able to reconstruct. The

parameters shown in table 6.15 were used for the true conductivity field and to initialize. Note that we for the full-3D method have used 30 within-Gibbs iterations to improve the sampling.

Parameter	Value	Parameter	Value
True center 1 c_1	(0.8, 0.8) cm	True center 1 c_1	(0.7, 0.7, 0.7) cm
True center 2 c_2	(0.3, 0.3) cm	True center 2 c_2	(0.3, 0.3, 0.3) cm
True radii (r_1, r_2)	(0.15, 0.2) cm	True radii (r_1, r_2)	(0.15, 0.2) cm
Conductivities (κ_1, κ_2)	(3, 10) S/cm	Conductivities (κ_1, κ_2)	(3, 10) S/cm
Data grid	$66 \times 66 \times 66$	Data grid	$10 \times 10 \times 10$
Reconstruction grid	$64 \times 64 \times 64$	Reconstruction grid	$8 \times 8 \times 8$
Initial center 1	(0.6, 0.6) cm	Initial center 1	(0.5, 0.5, 0.5) cm
Initial center 2	(0.1, 0.1) cm	Initial center 2	(0.1, 0.1, 0.1) cm
Initial radii	(0.1, 0.1) cm	Initial radii	(0.1, 0.1) cm
Init. conductivities	(1, 1) S/cm	Init. conductivities	(1, 1) S/cm
Stepsize ($\beta_c, \beta_r, \beta_\kappa$)	(0.01, 0.02, 0.02)	Stepsize (β_c, β_r)	(0.05, 0.07)
Bounds ($c_{\text{low}}, c_{\text{high}}$)	(0, 1) cm	Bounds ($c_{\text{low}}, c_{\text{high}}$)	(0, 1) cm
Bounds ($r_{\text{low}}, r_{\text{high}}$)	(0.05, 0.5) cm	Bounds ($r_{\text{low}}, r_{\text{high}}$)	(0.05, 0.5) cm
Bounds ($\kappa_{\text{low}}, \kappa_{\text{high}}$)	(0, 100) S/cm	Bounds ($\kappa_{\text{low}}, \kappa_{\text{high}}$)	(0, 100) S/cm
Within Gibbs iter.	20	Within Gibbs iter.	30

Table 6.15: Initial parameters for problem 5. Left: pseudo-3D method. Right: full-3D method.

We draw 5000 posterior samples with the Metropolis-within-Gibbs sampler described in algorithm 1. We use the log posterior conditionals shown in eq. 6.11 with the dimensions of the parameters tailored to this problem. The CPU times for both cases are shown in table 6.16.

	pseudo-3D method	full-3D method
CPU time	16.1 hours	12.9 hours

Table 6.16: CPU times on the HPC system for problem 5.

6.7.1 Statistical analysis of posterior samples

Trace plots for each parameter are shown for the pseudo-3D method in figure 6.24 and for the full-3D method in figure 6.25. Looking at the parameter traces we see several bad, but interesting, things happening. The Markov chains of the parameters defining the high conductivity inclusion 2 are mixing better than the Markov chains of the parameters defining the low conductivity inclusion 1. The chain mixing of the center coordinates of inclusion 2 is decent once burn-in has been performed. The chains for c_{2x} and c_{2y} seem to converge to the true value of (0.3, 0.3) cm. These observations also apply to the full-3D method in figure 6.25.

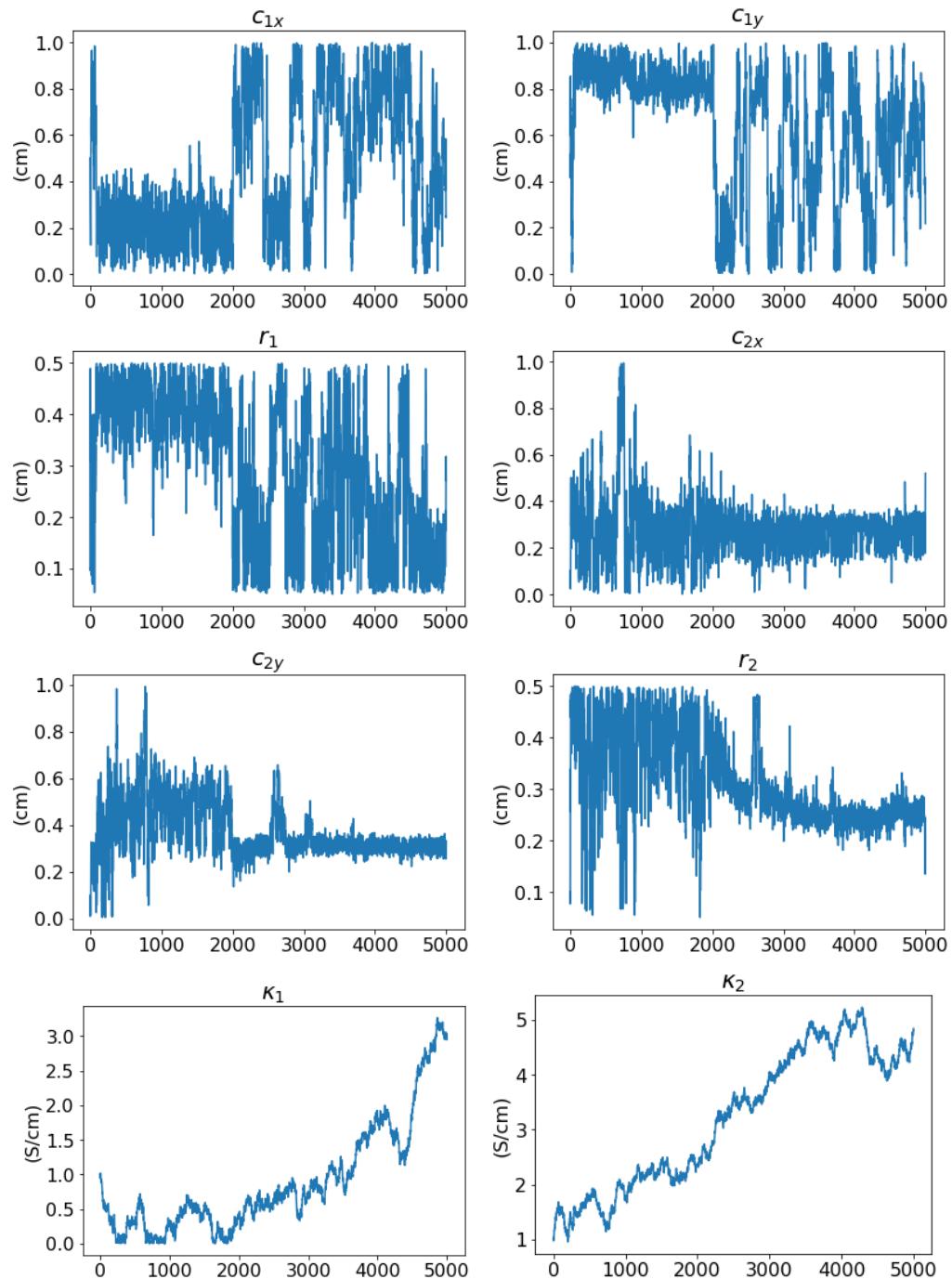


Figure 6.24: Parameter trace of 5000 samples for the pseudo-3D method in problem 5

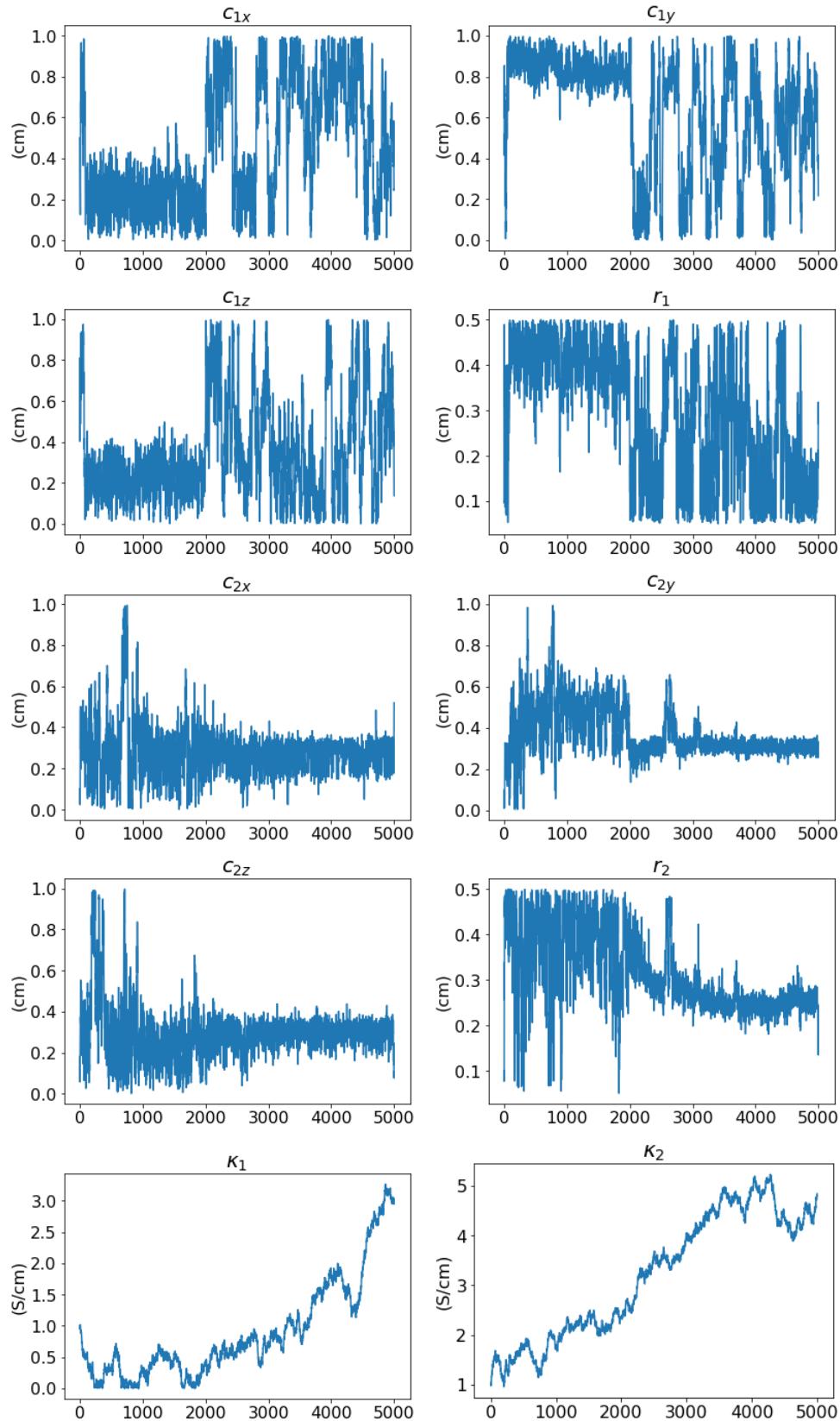


Figure 6.25: Parameter trace of 5000 samples for the full-3D method in problem 5

For both the pseudo-3D and the full-3D method, the trace plots for the center coordinates of

the low conductivity inclusion with index 1, i.e. c_{1x} , c_{1y} , and c_{1z} jump between different modes of the posterior. For this inclusion, we see that the two modes are around 0.3 and 0.8 which are also the two true values of the center coordinates. The magnetic field emitted by the lesser conducting inclusion is weak compared to the more conducting one with index 2, i.e. inclusion 2 dominates the magnetic field data. Therefore the field from inclusion 1 blend in with the background compared to the strong field from inclusion 2. This makes the sampler believe that there are two conducting inclusions at around the same place, or a larger inclusion if the two overlap. This is most prevalent for the full-3D method in figure 6.25. There is not much bimodality in the distributions of the parameters for the second, more conducting inclusion. The mixing of the conductivity parameter chains (κ_1, κ_2) is very poor in both pseudo-3D and full-3D. We discard the first 1000 samples as burn-in and perform diagnostics on the resulting chain. The posterior mean $\mathbb{E}[\cdot]$, the integrated autocorrelation time (IACT), and the effective sample size n_{eff} are computed for each chain and displayed in table 6.17. We also compute the 95% cred-

Diagnostic	Value	Diagnostic	Value
Center 1 c_1 IACT	(40.95, 32.15)	Center 1 c_1 IACT	(337.9, 158.3, 97.25)
Center 2 c_2 IACT	(3.826, 3.564)	Center 2 c_2 IACT	(7.865, 294.9, 39.86)
Radii (r_1, r_2) IACT	(153.9, 289.2)	Radii (r_1, r_2) IACT	(182.4, 454.9)
Conductivity 1 IACT	513.7	Conductivity 1 IACT	460.9
Conductivity 2 IACT	511.4	Conductivity 2 IACT	543.2
Center 1 $c_1 n_{\text{eff}}$	(98, 124)	Center 1 $c_1 n_{\text{eff}}$	(12, 25, 41)
Center 2 $c_2 n_{\text{eff}}$	(1045, 1122)	Center 2 $c_2 n_{\text{eff}}$	(509, 14, 100)
Radii (r_1, r_2) n_{eff}	(26, 14)	Radii (r_1, r_2) n_{eff}	(22, 9)
Conductivity 1 n_{eff}	8	Conductivity 1 n_{eff}	9
Conductivity 2 n_{eff}	8	Conductivity 2 n_{eff}	7
$\mathbb{E}[c_1^k]$	(0.796, 0.775) cm	$\mathbb{E}[c_1^k]$	(0.499, 0.576, 0.373) cm
$\mathbb{E}[c_2^k]$	(0.292, 0.300) cm	$\mathbb{E}[c_2^k]$	(0.264, 0.358, 0.279) cm
$\mathbb{E}[(r_1^k, r_2^k)]$	(0.143, 0.207) cm	$\mathbb{E}[(r_1^k, r_2^k)]$	(0.267, 0.298) cm
$\mathbb{E}[(\kappa_1^k, \kappa_2^k)]$	(3.756, 7.541) S/cm	$\mathbb{E}[(\kappa_1^k, \kappa_2^k)]$	(1.054, 3.615) S/cm

Table 6.17: Chain diagnostics for problem 5. Left: pseudo-3D method. Right: full-3D method.

ibility intervals and plot them in figure 6.26 along with the true and estimated parameters. The

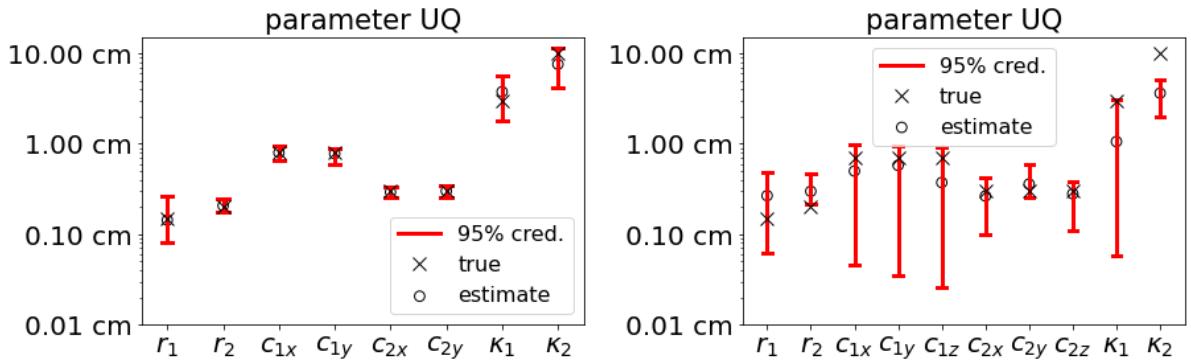


Figure 6.26: Parameter UQ analysis for problem 5. Left: pseudo-3D method. Right: full-3D method.

parameter uncertainty quantification reveals that all parameters fall within the 95% credibility intervals for the pseudo-3D method. For the full-3D method, all but one parameter falls within its 95% credibility interval and that is the conductivity level of the cancerous inclusion κ_2 . We already saw in the previous problems that estimating a single conductivity value was hard, and

it was often the parameter with the most uncertainty involved. Estimating two different conductivity levels is harder, which is confirmed by the abnormally large IACT in table 6.17 for both the pseudo-3D and full-3D method. We also see this from the poor chain mixing for (κ_1, κ_2) in figures 6.24 and 6.25. These observations lead us to suspect that high-conductivity anomalies are reconstructed better than low-conductivity anomalies when in the presence of each other.

6.7.2 Conductivity reconstructions

After post-processing the samples (c_k, r_k, κ_k) we estimate the conductivity field by computing posterior means. This is done in parameter space and conductivity space. Results for the pseudo-3D problem are shown in figure 6.27 and results for the full-3D method are shown in figure 6.28. For the pseudo-3D method, the pushforward of the parameter means in the top right resembles the true conductivity field quite well, apart from the underestimated conductivity level of the two inclusions. We see in the plot to the right in the bottom row, how the standard deviation is largest at the boundary of the highly conductive inclusion, but also nonzero in the interior since we also sample the conductivity level. The two inclusions have been reconstructed quite well in this case, especially the highly conductive one which is what we hoped for.

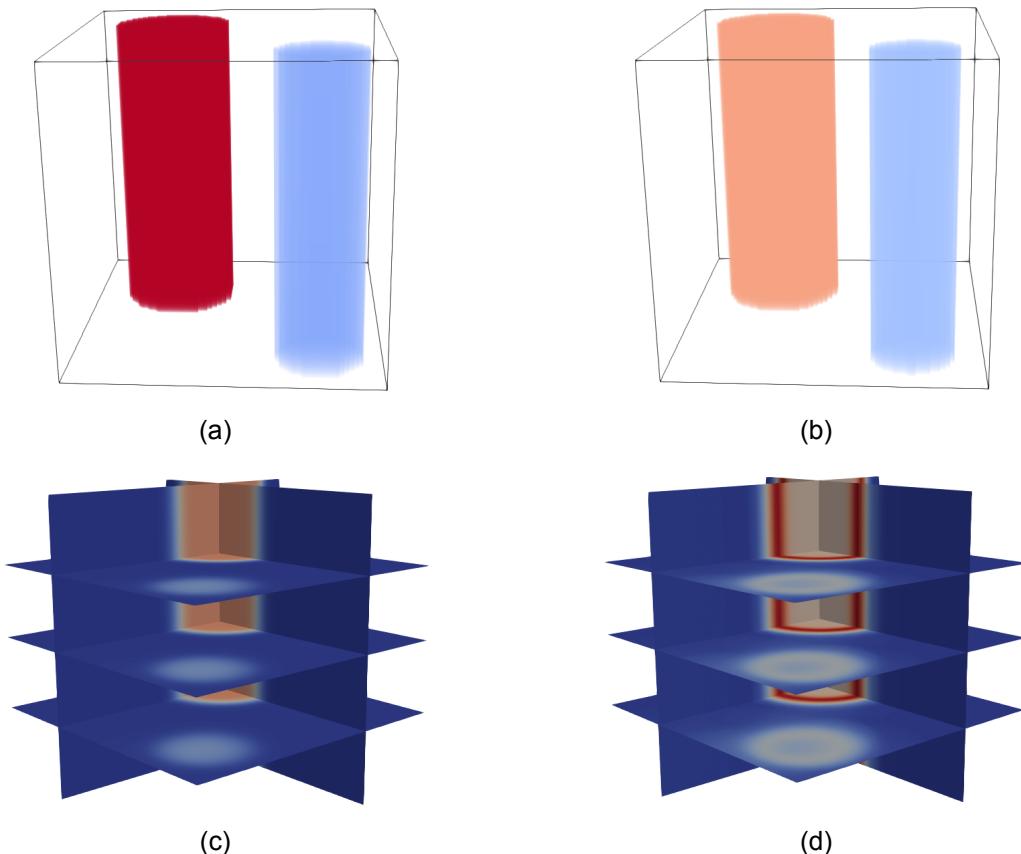


Figure 6.27: Pseudo-3D reconstructions for problem 5. (a): true conductivity field. (b): pushforward of parameter sample means. (c): mean of pushforward of parameter samples. (d): standard deviation of pushforward of parameter samples.

For the full-3D method in figure 6.28 we see that none of the inclusions are reconstructed to a satisfactory degree. In the plot of the pushforward of the parameter means to the top left we see that the more conductive gray inclusion is somewhat near the position of the highly conductive red inclusion in the true conductivity field to the top left. The plots in the bottom row reveal how the conductivity of the blue ball in the top right plot, has a conductivity value very close

to the background. In the bottom right plot, we can see the remnants of the less conductive blue ball as a white shade of nonzero standard deviation, right next to the red area indicating the uncertainty region of the more conductive gray inclusion. We again see that the uncertainty accumulates at the boundaries of the inclusions.

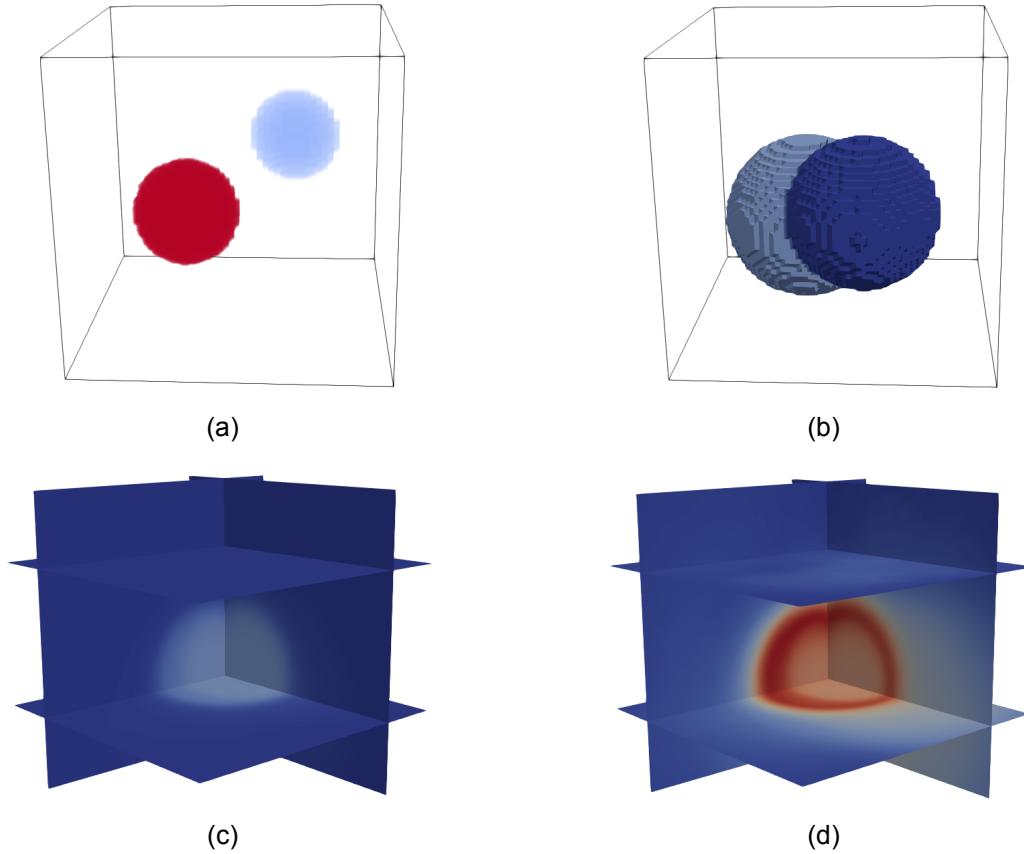


Figure 6.28: Full-3D reconstructions for problem 5. (a): true conductivity field. (b): pushforward of parameter sample means. (c): mean of pushforward of parameter samples. (d): standard deviation of pushforward of parameter samples.

7 Discussion

As we discuss briefly in section 2.2 there is a number of methods available when solving the MREIT reconstruction problem. However, if one wishes to do uncertainty quantification on the reconstructed conductivity fields then the Bayesian framework is a natural choice. When solving an inverse problem in the Bayesian setting there are a few ways to do UQ. In this thesis, we compute standard deviations and credibility intervals of a set of samples we draw from the posterior density via MCMC methods. MCMC methods are computationally expensive and often infeasible for large-scale problems because of the many forward model evaluations. This is why we had to resort to such small grid sizes in the full-3D model. Another way of doing UQ is to compute an estimate of the posterior covariance matrix around a given point in the solution space, usually taken to be the mode of the posterior, i.e. the MAP estimate. Computing the exact covariance is very costly for large dimensional parameter spaces and is often intractable without further approximation. Some authors suggest using a low-rank approximation of the posterior covariance matrix [35].

By using MCMC methods we describe the posterior from a set of samples that are expensive to compute. Another way of avoiding the heavy computational load of evaluating an expensive posterior is to instead approximate the posterior with a simpler probability distribution. One method is the Laplace approximation which fits an un-normalized Gaussian density to the posterior. The Gaussian is centered at the mode and the covariance matrix is taken to be the negative inverse of the Hessian of the log posterior at the mode [36, 37]. We also mention variational Bayesian methods that approximate the posterior with a simpler proposal distribution [37]. This involves solving an optimization problem where some difference measure between the posterior and the proposal distribution is minimized. The density one obtains via the variational Bayesian method is locally optimal and is an analytical approximation of the posterior.

The simple parametric approach we use for the prior distribution assumes quite a lot of information. Ultimately we want an approach where we do not have to specify the number of inclusions, their shapes, or the conductivity levels beforehand. We only know that the conductivity field σ is bounded between two positive constants, that it is a discontinuous field, and that there is a contrast between different materials in an object. A prior with this behavior must be able to generate discontinuous, stochastic fields. Continuous stochastic fields can be generated with a Karhunen-Loëve (KL) expansion, while discontinuity can be introduced by using level sets [38, 39] to threshold the field. Using such a prior involves more variables than our parametric approach. UQ on the sampled Karhunen-Loëve expansion parameters does not have an immediate interpretation as UQ on samples of radius or conductivity parameters does.

We discuss in section 5.2 how we choose the boundary conditions in the Poisson equation in eq. (5.12). Inspired by [2] we chose the Dirichlet condition that $u(x, y, z) = x$ on $\partial\Omega$. This has the effect that the current density field points from left to right in the x -direction. This is due to the potential being set to $u = 1$ on the right side $x = 1$, and $u = 0$ on the left side $x = 0$. More realistic choices could have been made to simulate the electrodes attached to the surface of the conducting object in MREIT. We mention the Complete Electrode Model (CEM) used in [14], which defines short pairs of segments on opposite sides of $\partial\Omega$ as electrodes. On these segments, the normal component of the current field \mathbf{J} must integrate to some fixed value I_j since we are applying a fixed amount of current through the electrodes to induce the magnetic field. On the segments representing the electrodes, they also impose the condition that $\nabla u \times \mathbf{n} = 0$, i.e. no electricity escapes out of the electrodes in a direction normal to the

surface. On the remaining boundary not covered by electrodes, we have $\sigma_{\partial n}^{\partial u} = 0$ on $\partial\Omega$, i.e. no current goes in or out of the domain.

In figures 6.4, 6.7, 6.11 and especially figure 6.26 we see a large difference between the width of the credibility intervals of the pseudo-3D method compared to the full-3D method. There seems to be more uncertainty involved with the parameter estimates for the full-3D method. The trace plots for the full-3D method in figures 6.8, 6.13, 6.21, 6.20 and 6.25 reveal that the chains in the pseudo-3D model are mixing better. The trace plots for the full-3D method are not as dense and do not show the same level of "fuzzy caterpillar" behavior as the chains for the pseudo-3D method does. The chain diagnostics show how the integrated auto-correlation time (IACT) is generally much larger in the full-3D method, which consequently leads to smaller effective sample sizes. These discrepancies between the performance of the pseudo-3D method compared to the full-3D method can mostly be ascribed to the *curse of dimensionality*. Since there are more degrees of freedom in the full-3D method, it means we are sampling from a posterior distribution of higher dimensions. This makes the sampling way harder, as the volume of the space in which the sampling is performed grows exponentially with each added dimension. This makes it harder for the sampler to move around the support of the posterior as there is more space to cover. Also, as seen in table 6.7 the achievable resolution for the full-3D method is not as high as for the pseudo-3D method. Having a smaller number of points to represent your computational domain also leaves room for discretization errors.

The benchmark MAP estimate can be compared with the results for the rotated ellipsoidal inclusion in problem 2 in section 6.4. We achieved the best results by using approximate gradients in figure 6.1c. If we compare the MAP against a single slice of the mean of the mapped parameters for the pseudo-3D method in figure 6.14c, we see that the MAP estimate captures the shape and orientation of the inclusion as well the sampling method does, at least by visual inspection. However, the conductivity values for the MAP estimate are underestimated. This issue could be caused by our lack of an exact gradient for the cost functional in the call to the optimization algorithm. The run time for computing the MAP estimate was substantially smaller than the time used for sampling with an MCMC method. Using the L-BFGS-B algorithm [32] we computed the MAP estimate on a 32×32 grid in about 5 minutes on the HPC system.

In problem 2 in section 6.4 we discuss how there are some non-uniqueness issues related to estimating the semi-axes and the rotation angle for a rotated ellipsoidal inclusion. When the rotation was in the xy -plane about the z -axis, we had to introduce a constraint on the x - and y -components of the semi-axes. We briefly mention that if the rotation was about the y -axis instead, i.e. in the xz -plane, then we would have to introduce the constraint $r_x < r_z$. Similarly, we would introduce $r_y < r_z$ if the rotation was about the x -axis in the yz plane. The inequality sign can be flipped in all constraints, as long as they are not allowed to switch values in the sampling since this is what caused the non-uniqueness issue.

In figure 6.18 for problem 3 we observed that the band of uncertainty is wider in the x -direction around the boundary 2D elliptic inclusion. This might have something to do with the boundary condition $u(x, y, z) = x$. As stated above the current flows from right to left. This has an impact on the magnetic field data since it determines its orientation. If we instead use the Dirichlet boundary condition $u(x, y, z) = y$, then the magnetic field data is rotated by 90 degrees because the current flows perpendicular to the previous case (see figure A.3 in appendix 3). We see from figure 5.1 that the magnetic field spreads out in the x -direction, by consequence of the current flow direction. This spreading could make the magnetic field data insensitive to perturbing the x -component of the radius of the elliptic inclusion by a small amount. Two inclusions with slightly different r_x values can produce similar magnetic field data, as the information is blurred in the direction of change.

In section 6.6 we explored what effect sparse magnetic field data in the z -direction would have on the reconstructions. The dense data case was covered in section 6.4. Comparing figures 6.11 and 6.19 we see that there is not much difference in the estimates or the uncertainty of the parameters. Only for the rotation angle do we see a particularly bad estimate from the 33% data case, while the dense data case is very close to the true one. Due to the heavy computational load of using MCMC methods, we could only do the 3D sampling on an 8^3 grid. In the two experiments with 50% and 33% data, we are using 4 and 3 slices in total respectively. This setup has the potential for massive discretization errors.

In problem 5 we see from figure 6.26 that the full-3D model is much more uncertain about the parameters compared with the pseudo-3D model. The estimated parameters in the full-3D model are not as close to the true parameters as for the pseudo-3D model. Since the cancerous inclusion is more conducting its magnetic field is stronger compared to the magnetic field from the lesser conducting inclusion. With high contrast in the conductivity level, the magnetic field from the low conductivity inclusion is harder to distinguish from the background. Since we only see high uncertainty for the full-3D model we believe that the curse of dimensionality is also an issue here. The small computational grid in 3D does not have a positive impact either as two different inclusions can look similar if they are downsampled enough. In section 6.3 for problem 1 we investigated a very similar case to problem 5, but where we knew the conductivity levels of the two inclusions. From figure 6.7 we see that the two inclusions are identified with much more certainty if we already know the conductivity levels.

7.1 Outlook

Using a Metropolis-within-Gibbs sampler to solve the Bayesian inverse problem is possible because we can derive the conditional distributions of all parameters defining the conductivity field. This was simple as we assumed independence between the parameters. Hence a component-wise Gibbs sampler was relatively simple to implement. If we consider more complicated parameterizations of the conductivity field then we have to use more advanced sampling methods. More advanced methods such as Hamiltonian Monte Carlo methods can avoid the random walk behavior of standard MCMC methods such as Metropolis-Hastings and Gibbs sampling, by taking informed steps rather than random steps [40]. This lowers the correlation between samples in the Markov chain as the sampler can use the gradient information to move to a high probability area of the distribution. These methods require that we supply gradients of the posterior density w.r.t. the parameters, and these can be very tough to compute. One such sampling method is the No-U-Turn Sampler (NUTS) which is state-of-the-art in Hamiltonian Monte Carlo methods [41]. We only got to consider gradients of the posterior very late in the project period in conjunction with computing the MAP estimate for benchmarking. As such time did not permit us to apply gradient information to improve our sampling methods.

One way of reducing the sampling time of an MCMC method is to make the forward model cheaper to evaluate. Most computations in our forward model involve solving PDEs in FEniCS. Solving the large linear system of equations in (4.11) for the expansion coefficients in the finite element basis is computationally very costly, which is why we resort to iterative solvers in our method. However, the solving procedure can also be done in parallel as FEniCS can utilize the Message Passing Interface (MPI). The linear system is split into several parts where each part is solved in parallel on separate cores [42].

We have assumed that the conductivity is constant inside the inclusions of the isotropic conductivity field σ . It would be interesting to relax that assumption and consider anisotropic conductivity fields, as several materials in engineering and several types of tissue in the human body are anisotropic. Muscle fibers, tendons, and white matter in the brain are examples of soft tissue that show anisotropic properties [43, 5]. We mentioned non-invasive brain stimulation as a po-

tential application of MREIT in the introduction in section 1, so relaxing the isotropic constraint is especially important for this case.

Ideally, we wanted to benchmark our Bayesian method of solving the MREIT reconstruction problem against a deterministic method well-rooted in the literature. One such method is the harmonic B_z algorithm [10]. We are only benchmarking the results from the MCMC sampling against a MAP estimate which we compute in section 6.1. However, the harmonic B_z algorithm assumes the Complete Electrode Model as boundary conditions which we describe in the previous section [44]. Due to this issue, it was infeasible for us to implement the harmonic B_z algorithm in due time.

7.2 Conclusion

This research aims to develop a computational framework for solving and doing uncertainty quantification on the reconstruction problem in Magnetic Resonance Electrical Impedance Tomography (MREIT). We have used statistical inversion theory to solve the inverse problem in a probabilistic sense [6]. The solution is a probability distribution we construct with Bayes' theorem called the posterior distribution, which is then used to infer hidden information about the conductivity field. We have taken the approach of parameterizing the conductivity field σ as spherical and ellipsoidal inclusions embedded in a background medium of constant conductivity value. The process of inferring and doing uncertainty quantification on the defining parameters of the inclusions involves using MCMC methods to estimate moments and statistics of the posterior distribution. The numerical results show that our sampling-based method is able to reconstruct up to two inclusions of varying shapes and sizes with both known and unknown conductivity levels κ , even with 10% relative noise to the magnetic field data. Our uncertainty quantification for the mapped parameters in conductivity space reveals how the pixel-wise standard deviation accumulates at the boundaries of the inclusions. If the conductivity level is unknown we also see uncertainty in the interior of the inclusions in the form of nonzero standard deviation.

Using our method, the credibility intervals and chain diagnostics of the Markov chains show that the two hardest parameters to draw independent samples of, are the conductivity levels κ and rotation angle δ of the elliptic inclusions. Our results show that there is increased uncertainty involved with solving the reconstruction problem in full 3D and with sparse data. The curse of dimensionality increases the difficulty of the sampling. We are able to achieve more robust results by using the pseudo-3D method as the reconstruction grid is finer and the dimensions of the posterior distributions are lower. To generalize our results to a broader class of inclusions and conductivity fields, future studies should address the case of using random fields as a prior distribution. While parameterizing conductivity fields as inclusions in a background medium limits the generalizability of the results, our method does demonstrate that pixel-wise uncertainty quantification for the MREIT conductivity reconstruction problem is possible for a simplified setup.

A Appendix

Appendix 1 - Fréchet derivative estimate

We want to show that the following estimate holds for the Fréchet derivative we computed in section 2.2.1

$$\|\mathcal{F}(\tilde{\sigma} + \delta\sigma) - \mathcal{F}(\tilde{\sigma}) - d\mathcal{F}(\tilde{\sigma})\delta\sigma\|_{L^2(\Omega)} \leq C\|\delta\sigma\|_{L^\infty(\Omega)}^2. \quad (\text{A.1})$$

We linearize the problem by computing the Fréchet derivative of the forward operator which in turn is used to derive the Newton procedure. We have already defined the forward mapping from a conductivity distribution σ to an interior current density field $\mathbf{J} = -\sigma\nabla u$. Let \tilde{u} be the solution to (2.1) where σ is replaced by a reference conductivity $\tilde{\sigma}$. If we take a perturbation $\delta\sigma$ such that $\tilde{\sigma} + \delta\sigma \in L_+^\infty(\Omega)$ we want to compute a linear operator $d\mathcal{F}(\tilde{\sigma})$ such that [45]

$$\lim_{\delta\sigma \rightarrow 0} \frac{\|\mathcal{F}(\tilde{\sigma} + \delta\sigma) - \mathcal{F}(\tilde{\sigma}) - d\mathcal{F}(\tilde{\sigma})\delta\sigma\|_{L^2(\Omega)}}{\|\delta\sigma\|_{L^\infty(\Omega)}} = 0. \quad (\text{A.2})$$

If this holds then the linear operator $d\mathcal{F}(\tilde{\sigma})$ is denoted by the Fréchet derivative. We are doing a linearization of the forward mapping around $\tilde{\sigma}$. Computing the derivative involves doing a first-order approximation of the form

$$\mathcal{F}(\tilde{\sigma} + \delta\sigma) - \mathcal{F}(\tilde{\sigma}) \approx d\mathcal{F}(\tilde{\sigma})\delta\sigma. \quad (\text{A.3})$$

We have the expression $\sigma = \tilde{\sigma} + \delta\sigma$, and we use this to compute an identity with $u - \tilde{u}$

$$\begin{aligned} \nabla \cdot \tilde{\sigma}\nabla(u - \tilde{u}) &= \nabla \cdot \tilde{\sigma}\nabla u - \nabla \cdot \tilde{\sigma}\nabla \tilde{u} = \nabla \cdot \tilde{\sigma}\nabla u - 0 = \nabla \cdot \tilde{\sigma}\nabla u \\ \nabla \cdot (\tilde{\sigma} - \sigma)\nabla u &= \nabla \cdot \tilde{\sigma}\nabla u - \nabla \cdot \sigma\nabla u = \nabla \cdot \tilde{\sigma}\nabla u - 0 = \nabla \cdot \tilde{\sigma}\nabla u. \end{aligned}$$

Which yields the following PDE problem for $u - \tilde{u}$ since u and \tilde{u} both solve the Poisson equation with boundary datum f

$$\begin{aligned} \nabla \cdot \tilde{\sigma}\nabla(u - \tilde{u}) &= \nabla \cdot (\tilde{\sigma} - \sigma)\nabla u && \text{in } \Omega, \\ u - \tilde{u} &= f - f = 0 && \text{on } \partial\Omega. \end{aligned} \quad (\text{A.4})$$

We have that $\delta u \approx u - \tilde{u}$ to first order in δu . Doing the approximation $\nabla u \approx \nabla \tilde{u}$ on the right-hand-side above yields a PDE problem for $\delta u \in H_0^1(\Omega)$

$$\begin{aligned} \nabla \cdot \tilde{\sigma}\nabla \delta u &= -\nabla \cdot \delta\sigma\nabla \tilde{u} && \text{in } \Omega, \\ \delta u &= 0 && \text{on } \partial\Omega. \end{aligned} \quad (\text{A.5})$$

So (A.5) is the PDE problem for δu . Doing the Fréchet derivative computation in (A.3) yields

$$\begin{aligned} \mathcal{F}(\tilde{\sigma} + \delta\sigma) - \mathcal{F}(\tilde{\sigma}) &= -(\tilde{\sigma} + \delta\sigma)\nabla(\tilde{u} + \delta u) - (-\tilde{\sigma}\nabla \tilde{u}) \\ &= -\tilde{\sigma}\nabla \tilde{u} - \tilde{\sigma}\nabla \delta u - \delta\sigma\nabla \tilde{u} - \delta\sigma\nabla \delta u + \tilde{\sigma}\nabla \tilde{u} \\ &= -\tilde{\sigma}\nabla \delta u - \delta\sigma\nabla \tilde{u} - \delta\sigma\nabla \delta u. \end{aligned} \quad (\text{A.6})$$

If we define $d\mathcal{F}(\tilde{\sigma})\delta\sigma := -\tilde{\sigma}\nabla \delta u - \delta\sigma\nabla \tilde{u}$, we can write

$$\begin{aligned} \mathcal{F}(\tilde{\sigma} + \delta\sigma) - \mathcal{F}(\tilde{\sigma}) - (-\tilde{\sigma}\nabla \delta u - \delta\sigma\nabla \tilde{u}) &= -\delta\sigma\nabla \delta u \\ \mathcal{F}(\tilde{\sigma} + \delta\sigma) - \mathcal{F}(\tilde{\sigma}) - d\mathcal{F}(\tilde{\sigma})\delta\sigma &= -\delta\sigma\nabla \delta u. \end{aligned} \quad (\text{A.7})$$

Taking the L^2 norm on both sides yields

$$\|\mathcal{F}(\tilde{\sigma} + \delta\sigma) - \mathcal{F}(\tilde{\sigma}) - d\mathcal{F}(\tilde{\sigma})\delta\sigma\|_{L^2(\Omega)} = \|\delta\sigma\nabla\delta u\|_{L^2(\Omega)} \leq \|\delta\sigma\|_{L^\infty(\Omega)} \|\nabla\delta u\|_{L^2(\Omega)}. \quad (\text{A.8})$$

We wish for the bound above to only be expressed only in terms of $\|\delta\sigma\|_{L^\infty(\Omega)}$ so we ought to explore the term $\|\nabla\delta u\|_{L^2(\Omega)}$. Remember that δu was the solution to an inhomogenous Poisson problem in (A.5) with a term containing $\delta\sigma$ on the right-hand-side. Generally, we can ask whether the solution v to a general Poisson equation

$$\begin{aligned} \nabla \cdot \sigma \nabla v &= g && \text{in } \Omega, \\ v &= 0 && \text{on } \partial\Omega, \end{aligned} \quad (\text{A.9})$$

can be bounded in terms of the functional on the right-hand-side g . For a general second-order elliptic PDE in Ω we look for solutions $u \in H_0^1(\Omega)$. Based on section 5.9.1 in [21] we can denote the dual space of $H_0^1(\Omega)$ by $H^{-1}(\Omega)$. The functional g on the right-hand-side of the PDE belongs to $H^{-1}(\Omega)$, if it is a bounded linear functional on the space $H_0^1(\Omega)$. To see this we need to consider the variational form of the problem which is derived by multiplying both sides of the PDE with a test function $w \in H_0^1(\Omega)$ and integrating by parts on the left-hand-side

$$a(v, w) := \int_{\Omega} \sigma \nabla v \cdot \nabla w \, d\mathbf{x} = \int_{\Omega} gw \, d\mathbf{x} := \langle g, w \rangle, \quad \forall w \in H_0^1(\Omega). \quad (\text{A.10})$$

The equation above defines the variational problem, find $v \in H_0^1(\Omega)$ such that

$$a(v, w) = \langle g, w \rangle, \quad \forall w \in H_0^1(\Omega). \quad (\text{A.11})$$

Certain conditions on the bilinear form $a(v, w)$ must be met in order for a unique solution to exist. This result is the Lax-Milgram theorem which we introduced in chapter 4. The coercivity condition in (4.6) implies that for a solution v of the variational problem in (A.11) we have for some $\beta > 0$

$$\begin{aligned} \beta \|v\|_{H_0^1(\Omega)}^2 &\leq a(v, v) \\ \implies \beta \|v\|_{H_0^1(\Omega)} &\leq \frac{a(v, v)}{\|v\|_{H_0^1(\Omega)}} = \frac{|\langle g, v \rangle|}{\|v\|_{H_0^1(\Omega)}} \leq \sup_{v \neq 0 \in H_0^1(\Omega)} \frac{\langle g, v \rangle}{\|v\|_{H_0^1(\Omega)}} = \|g\|_{H^{-1}(\Omega)}. \end{aligned} \quad (\text{A.12})$$

So we can actually bound a solution v to the Poisson equation in terms of the H^{-1} norm of the right-hand-side g , which will prove to be useful. We can take the functional H^{-1} norm on both sides of the PDE in (A.5)

$$\|\nabla \cdot \tilde{\sigma} \nabla \delta u\|_{H^{-1}(\Omega)} = \|-\nabla \cdot \delta\sigma \nabla \tilde{u}\|_{H^{-1}(\Omega)}, \quad (\text{A.13})$$

with the norm on $H^{-1}(\Omega)$ defined as

$$\|g\|_{H^{-1}(\Omega)} = \sup_{\|v\|_{H_0^1(\Omega)}=1} |\langle g, v \rangle|, \quad (\text{A.14})$$

where $\langle \cdot, \cdot \rangle$ denotes the dual pairing between $H^{-1}(\Omega)$ and $H_0^1(\Omega)$. Computing the $H^{-1}(\Omega)$ norm on the right hand side in (A.13) along with integration by parts and Cauchy-Schwartz yields

$$\begin{aligned} \|-\nabla \cdot \delta\sigma \nabla \tilde{u}\|_{H^{-1}(\Omega)} &= \sup_{\|v\|=1} |\langle -\nabla \cdot \delta\sigma \nabla \tilde{u}, v \rangle| = \sup_{\|v\|=1} \left| \int_{\Omega} (-\nabla \cdot \delta\sigma \nabla \tilde{u}) v \, d\mathbf{x} \right| \\ &= \sup_{\|v\|=1} \left| \int_{\Omega} \delta\sigma \nabla \tilde{u} \nabla v \, d\mathbf{x} \right| \leq \sup_{\|v\|=1} \int_{\Omega} |\delta\sigma \nabla \tilde{u} \nabla v| \, d\mathbf{x} \\ &\leq \sup_{\|v\|=1} \|\delta\sigma\|_{L^\infty(\Omega)} \|\nabla \tilde{u}\|_{L^2(\Omega)} \|\nabla v\|_{L^2(\Omega)}. \end{aligned} \quad (\text{A.15})$$

Recall the definition of the Sobolev norm $\|u\|_{H_0^1(\Omega)}$ in terms of L^2 norms of u and ∇u

$$\|u\|_{H_0^1(\Omega)} = \left(\|u\|_{L^2(\Omega)}^2 + \|\nabla u\|_{L^2(\Omega)}^2 \right)^{1/2}. \quad (\text{A.16})$$

Thus we have the valid estimate $\|\nabla u\|_{L^2(\Omega)} \leq \|u\|_{H_0^1(\Omega)}$ which we can use in (A.15)

$$\begin{aligned} \| -\nabla \cdot \delta\sigma \nabla \tilde{u} \|_{H^{-1}(\Omega)} &\leq \sup_{\|v\|=1} \|\delta\sigma\|_{L^\infty(\Omega)} \|\nabla \tilde{u}\|_{L^2(\Omega)} \|\nabla v\|_{L^2(\Omega)} \\ &\leq \sup_{\|v\|=1} \|\delta\sigma\|_{L^\infty(\Omega)} \|\tilde{u}\|_{H_0^1(\Omega)} \|v\|_{H_0^1(\Omega)} \\ &\leq \|\delta\sigma\|_{L^\infty(\Omega)} \|\tilde{u}\|_{H_0^1(\Omega)}. \end{aligned} \quad (\text{A.17})$$

Now since $\tilde{\sigma} \in L_+^\infty(\Omega)$ is a reference conductivity we know it is bounded from above by a positive constant M . Since the Poisson equation is elliptic we have by elliptic regularity that the reference voltage \tilde{u} is also bounded from above by a constant $\|\tilde{u}\|_{H_0^1(\Omega)} \leq C$, where C depends on the bound on $\tilde{\sigma}$ and the boundary conditions. We thus finally have for the $\| -\nabla \cdot \delta\sigma \nabla \tilde{u} \|_{H^{-1}(\Omega)}$ estimate

$$\| -\nabla \cdot \delta\sigma \nabla \tilde{u} \|_{H^{-1}(\Omega)} \leq C \|\delta\sigma\|_{L^\infty(\Omega)}. \quad (\text{A.18})$$

This implies for the Poisson problem in (A.5), that we can bound the solution δu in terms of the right-hand-side

$$\|\delta u\|_{H_0^1(\Omega)} \leq \| -\nabla \cdot \delta\sigma \nabla \tilde{u} \|_{H^{-1}(\Omega)}. \quad (\text{A.19})$$

Using the result we derived in (A.18) we conclude that

$$\|\delta u\|_{H_0^1(\Omega)} \leq \| -\nabla \cdot \delta\sigma \nabla \tilde{u} \|_{H^{-1}(\Omega)} \leq C \|\delta\sigma\|_{L^\infty(\Omega)}. \quad (\text{A.20})$$

We will then apply this result to the Fréchet derivative estimate in (A.8) which yields

$$\begin{aligned} \|\mathcal{F}(\tilde{\sigma} + \delta\sigma) - \mathcal{F}(\tilde{\sigma}) - d\mathcal{F}(\tilde{\sigma})\delta\sigma\|_{L^2(\Omega)} &\leq \|\delta\sigma\|_{L^\infty(\Omega)} \|\nabla \delta u\|_{L^2(\Omega)} \\ &\leq \|\delta\sigma\|_{L^\infty(\Omega)} \|\delta u\|_{H_0^1(\Omega)} \\ &\leq C \|\delta\sigma\|_{L^\infty(\Omega)}^2, \end{aligned} \quad (\text{A.21})$$

which shows that $d\mathcal{F}(\tilde{\sigma})\delta\sigma \approx \mathcal{F}(\tilde{\sigma} + \delta\sigma) - \mathcal{F}(\tilde{\sigma})$ to second order in $\delta\sigma$.

Appendix 2 - forward model implementation

```
1 def solver(c,r):
2     class sigmafun(UserExpression):
3         if( (x[0] - c[0])**2 + (x[1] - c[1])**2 + (x[2] - c[2])**2 <= r +
4             DOLFIN_EPS ):
5             values[0] = sigma_1
6         else:
7             values[0] = sigma_0
8     # Initialize conductivity sigma
9     femel = V.ufl_element()
10    sigma_tru = sigmafun(element=femel)
11    sigma_true = project(sigma_tru, V, solver_type='bicgstab', preconditioner_type='
12        sor')
13    # Define functions and variational form
14    u = TrialFunction(V)
15    v = TestFunction(V)
16    a = inner(sigma*grad(u),grad(v))*dx      # lhs of the weak formulation
17    L = Constant(0)*v*dx
18    # Solve the system
19    u = Function(V)
20    solve(a == L, u, bcs, solver_parameters = {'linear_solver':'bicgstab', '
21        preconditioner':'sor'})
22    sigma = Function(V)
23    sigma = interpolate(sigma_true, V)
24    # Compute the current density field
25    J_val = project(-sigma*grad(u), V_grad, solver_type='bicgstab',
26        preconditioner_type='sor').compute_vertex_values(mesh)
27    Jx = J_val[0:(n+1)**3].reshape(n+1, n+1, n+1)
    Jy = J_val[(n+1)**3:2*(n+1)**3].reshape(n+1, n+1, n+1)
    # Computes the magnetic field
    Bz = FFTCONV(Jx,Jy,n)
    return Bz
```

Listing A.1: FEniCS code for conductivity to magnetic field mapping

Figure A.1: Procedure for simulating the Poisson equation given parameters for a conductivity distribution.

```

28 def FFTCONV(Jx,Jy,n):
29     ''' Calculates the magnetic field caused by a current density distribution Jx,
30         Jy, and Jz '''
31     #Fourier transform of the Biot-Savart kernel
32     h = 1/n
33     k = 2*np.pi*np.fft.fftfreq(Jx.shape[0],h)
34     kx, ky, kz = np.meshgrid(k,k,k,indexing='ij',sparse=True)
35     kz[0,0,0] = 1e-9
36     K=1j/(kx**2+ky**2+kz**2)
37     K[0,0,0]=0
38
39     Jx_fft = np.fft.rfftn(Jx)
40     Byh = np.multiply((K*kz)[:, :, 0:len(K)//2+1], Jx_fft)
41     Bzh = np.multiply(-(K*ky)[:, :, 0:len(K)//2+1], Jx_fft)
42     del Jx_fft
43
44     Jy_fft = np.fft.rfftn(Jy)
45     Bxh = np.multiply(-(K*kz)[:, :, 0:len(K)//2+1], Jy_fft)
46     Bzh += np.multiply((K*kx)[:, :, 0:len(K)//2+1], Jy_fft)
47     del Jy_fft
48
49     Jz_fft = np.fft.rfftn(Jz)
50     Bxh += np.multiply((K*ky)[:, :, 0:len(K)//2+1], Jz_fft)
51     Byh += np.multiply(-(K*kx)[:, :, 0:len(K)//2+1], Jz_fft)
52     del Jz_fft
53
54     Bxh = np.fft.irfftn(Bxh)
55     Byh = np.fft.irfftn(Byh)
56     Bzh = np.fft.irfftn(Bzh)
57
58     return Bxh, Byh, Bzh

```

Listing A.2: Fourier method for fast evaluation of Biot-Savart

Figure A.2: Procedure for fast evaluation of magnetic fields via the Fourier transform.

Appendix 3 - magnetic field data from using a different boundary condition in 2D

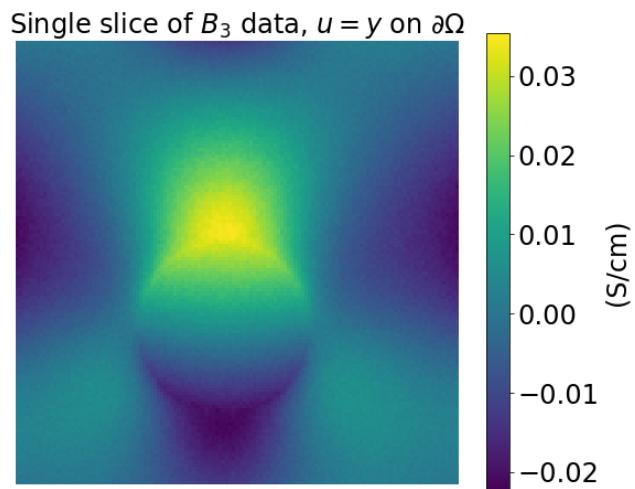


Figure A.3: Magnetic field data for a disk-inclusion where we instead use $u(x, y, z) = y$ as the boundary condition in the Poisson equation in (5.12).

Bibliography

- [1] Alberto-P. Calderón. *On an inverse boundary value problem*. In *Seminar on Numerical Analysis and its Applications to Continuum Physics (Rio de Janeiro, 1980)*, pages 65–73. Soc. Brasil. Mat., 1980.
- [2] H. Yazdanian and K. Knudsen. *Numerical conductivity reconstruction from partial interior current density information in three dimensions*. Inverse Problems, 37(10), 2021. URL: <https://doi.org/10.1088/1361-6420/ac1e81>.
- [3] J. K. Seo and E. J. Woo. *Magnetic Resonance Electrical Impedance Tomography (MREIT)*. SIAM Review, 53(1)., 2011.
- [4] Axel Thielsscher, Lars G. Hanson and Kristoffer H. Madsen. *Interview from lab visit at Danish Research Centre for Magnetic Resonance, Hvidovre Hospital*. <https://www.drcmr.dk/>. Visit date: 2023-01-17. 2023.
- [5] *Anisotropy in radiology*. URL: <https://radiopaedia.org/articles/anisotropy>.
- [6] Kaipio J. and Somersalo E. *Statistical and Computational Inverse Problems*. Springer Applied Mathematical Sciences, Volume 160, 2005. URL: <https://doi.org/10.1007/b138659>.
- [7] *Computational Uncertainty Quantification for Inverse problems - research initiative*. URL: <https://www.compute.dtu.dk/english/cuqi>.
- [8] “Magnetic Resonance Electrical Impedance Tomography”. In: *An Introduction to Mathematics of Emerging Biomedical Imaging*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 169–175. ISBN: 978-3-540-79553-7. DOI: 10.1007/978-3-540-79553-7_9. URL: https://doi.org/10.1007/978-3-540-79553-7_9.
- [9] Hasan H. Eroğlu et al. “On the reconstruction of magnetic resonance current density images of the human brain: Pitfalls and perspectives”. In: *NeuroImage* 243 (2021). ISSN: 1053-8119. DOI: <https://doi.org/10.1016/j.neuroimage.2021.118517>.
- [10] Suk Oh et al. “Conductivity and current density image reconstruction using Harmonic Bz algorithm in magnetic resonance electrical impedance tomography”. In: *Physics in medicine and biology* 48 (2003). DOI: 10.1088/0031-9155/48/19/001.
- [11] Thielsscher A. Yazdanian H. Saturnino G. B. and Knudsen K. *Fast evaluation of the Biot-Savart integral using FFT for electrical conductivity imaging*. Journal of Computational Physics, 411, 2020. URL: <https://doi.org/10.1016/j.jcp.2020.109408>.
- [12] Ohin Kwon et al. “Magnetic resonance electrical impedance tomography (MREIT): Simulation study of J-substitution algorithm”. In: *IEEE transactions on bio-medical engineering* 49 (2002). DOI: 10.1109/10.979355.
- [13] J. Hadamard. “Sur les problèmes aux dérivés partielles et leur signification physique”. In: *Princeton University Bulletin* 13 (1902), pp. 49–52.
- [14] Matthew M. Dunlop and Andrew M. Stuart. “The Bayesian Formulation of EIT: Analysis and Algorithms”. In: *Inverse Problems and Imaging* 10 (2016).
- [15] Jim Pitman. *Probability*. Springer Texts in Statistics, Springer New York, 1993. URL: <https://doi.org/10.1007/978-1-4612-4374-8>.
- [16] Art B. Owen. *Monte Carlo theory, methods and examples*. 2013.
- [17] M. Betancourt. “A conceptual introduction to Hamiltonian Monte Carlo”. In: *arXiv* (2017).
- [18] *IAT documentation*. URL: <https://search.r-project.org/CRAN/refmans/LaplaceDemon/html/IAT.html>.
- [19] J. Geweke. *Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments*. In *Bayesian Statistics 4*, Oxford University Press., 1992.

- [20] *numpy.percentile documentation*. URL: <https://numpy.org/doc/stable/reference/generated/numpy.percentile.html>.
- [21] L. C. Evans. *Partial differential equations*. Providence, R.I.: American Mathematical Society, 2010.
- [22] L. Ridgway Scott. *Introduction to Automated Modeling using FEniCS, Release 0.3*. University of Chicago, 2017.
- [23] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. Vol. 15. Texts in Applied Mathematics. Springer, 2008. ISBN: 9780387759333. DOI: 10.1007/978-0-387-75934-0.
- [24] G. N. Wells et. al. *The FEniCS Project Version 1.5*. Archive of Numerical Software 3, 2015. URL: <https://fenicsproject.org/>.
- [25] DH Griffel. *Applied Functional Analysis*, Other: Revised edition published in 1985. Ellis Horwood, 1981.
- [26] P. Kythe. *Fundamental Solutions for Differential Operators and Applications*. Springer Science & Business Media, 2012.
- [27] *scipy.fftpack.fft documentation*. URL: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.fftpack.fft.html>.
- [28] S. Siltanen K. Knudsen J. Mueller. *Numerical solution method for the dbar-equation in the plane*. Journal of Computational Physics, Volume 198, Issue 2, 2004. URL: <https://doi.org/10.1016/j.jcp.2004.01.028>.
- [29] Chris Sherlock and Gareth Roberts. “Optimal scaling of the random walk Metropolis on elliptically symmetric unimodal targets”. In: *Bernoulli* 15.3 (2009), pp. 774–798. DOI: 10.3150/08-BEJ176.
- [30] H. A. van der Vorst. “Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems”. In: *SIAM Journal on Scientific and Statistical Computing* 13.2 (1992), pp. 631–644. DOI: 10.1137/0913035.
- [31] David M. Young. “Iterative methods for solving partial difference equations of elliptic type”. In: *Transactions of the American Mathematical Society* 76 (1954), pp. 92–111.
- [32] Richard H. Byrd et al. “A Limited Memory Algorithm for Bound Constrained Optimization”. In: *SIAM Journal on Scientific Computing* 16.5 (1995), pp. 1190–1208. DOI: 10.1137/0916069.
- [33] *scipy.fmin_l_bfgs_b documentation*. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.fmin_l_bfgs_b.html.
- [34] Andrea Barth et al. “Detecting stochastic inclusions in electrical impedance tomography”. In: *Inverse Problems* 33.11 (2017). DOI: 10.1088/1361-6420/aa8f5c.
- [35] Peter Benner, Yue Qiu, and Martin Stoll. “Low-rank computation of posterior covariance matrices in Bayesian inverse problems”. In: (Mar. 2017).
- [36] Robert E. Kass, Luke Tierney, and Joseph (Jay) B. Kadane. “The validity of posterior expansions based on Laplace’s method”. In: 1990.
- [37] David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Copyright Cambridge University Press, 2003.
- [38] Matthew Dunlop, Marco Iglesias, and Andrew Stuart. “Hierarchical Bayesian Level Set Inversion”. In: *Statistics and Computing* 27 (2017). DOI: 10.1007/s11222-016-9704-8.
- [39] William Reese, Arvind K. Saibaba, and Jonghyun Lee. *Bayesian Level Set Approach for Inverse Problems with Piecewise Constant Reconstructions*. 2021. DOI: 10.48550/ARXIV.2111.15620.
- [40] M. J. Betancourt and Mark Girolami. “Hamiltonian Monte Carlo for Hierarchical Models”. In: (2013). DOI: 10.48550/ARXIV.1312.0906.
- [41] Matthew D. Hoffman and Andrew Gelman. “The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo”. In: (2011). DOI: 10.48550/ARXIV.1111.4246.

- [42] M. Sussman. “FEniCS, part I”. In: *Lecture slides from the course "math3040_2014 Topics in Scientific Computing"*. Department of Scientific Computing, Florida State University, 2014.
- [43] Zhanxiong Wu et al. “A review of anisotropic conductivity models of brain white matter based on diffusion tensor imaging”. In: *Medical & Biological Engineering & Computing* 56 (June 2018), pp. 1–8. DOI: 10.1007/s11517-018-1845-9.
- [44] Kiwan Jeon et al. “MREIT conductivity imaging based on the local harmonic Bz algorithm: Animal experiments”. In: *Journal of Physics: Conference Series* 224.1 (2010). URL: <https://dx.doi.org/10.1088/1742-6596/224/1/012079>.
- [45] H. Cartan. *Differential Calculus*. Hermann, 1983. ISBN: 9780901665140.

Technical
University of
Denmark

Matematiktorvet, Building 303b
2800 Kgs. Lyngby
Tlf. 4525 1700

www.compute.dtu.dk