5-6-2024

# Computational Framework for Parameter Identification in the Nearly Incompressible Linear Elasticity System

Rachel Hawks
rh9784@rit.edu

## Recommended Citation

Hawks, Rachel, "Computational Framework for Parameter Identification in the Nearly Incompressible Linear Elasticity System" (2024). Thesis. Rochester Institute of Technology. Accessed from

# Computational Framework for Parameter Identification in the Nearly Incompressible Linear Elasticity System

by

Rachel Hawks

*A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Mathematical Modeling*

School of Mathematical Sciences
College of Science
Rochester Institute of Technology
Rochester, New York
May 6, 2024

# Abstract

The problem being analyzed is based on linear elasticity equations that describe displacement in soft tissue under applied body forces in biomedical applications, specifically in the case of identifying soft tissue cancers. Breast cancer is one of the most common cancers in women and the second leading cause of death among women in the United States. Early detection is a key component in the treatment of cancer and can lead to drastic improvement of patient survival rates. One of the most common methods for early detection of breast cancer is manual examination, also called palpation. Since cancerous tissue is stiffer on average than healthy soft tissue, in the presence of a compressive force, the stiffer regions tend to deform less than the softer regions. Though it is a standard diagnosis technique, there are drawbacks to using palpation to detect the presence of tumors. Elasticity imaging, or elastography, can be employed as a method of tumor identification by using imaging systems such as an ultrasound to measure tissue deformation. The idea behind elastography is to reconstruct material parameters of the tissue, specifically the tissue stiffness, from measurements of the tissue displacement. The degree of tissue displacement will depend on its elasticity since changes in tissue stiffness generally correlate to pathological changes in the tissue. Using this knowledge along with an imaging system, regions that have contrasting material properties can be identified and quantified.

The task of identifying the tissue stiffness parameter is formulated as an optimization problem with a system of partial differential equations (PDEs) as a constraint. The optimization problem of estimating the tissue stiffness parameter, or the shear modulus, is solved using iterative methods that require repeated solving of the underlying PDE system (linear elasticity system). This results in a high computational cost in general and makes these methods less feasible in clinical applications. The primary goal of this work is to develop a computational framework based on finite elements for the identification of a distributed parameter in a system of PDEs where the inverse problem is formulated as an optimiza-

tion problem. We also propose an adaptive mesh refinement framework that provides the resolution needed for the recovery of the spatially varying parameter while improving the computational efficiency.

Chapter 1 introduces the problem, going into both the background and the underlying linear elasticity system. The finite element method (FEM) for the forward version of this problem is constructed in Chapter 2, and numerical experiments are run to show that the underlying linear elasticity system can be solved accurately. Chapter 3 focuses on the inverse problem, where various numerical experiments explore how well the framework captures the behavior of stiff inclusions, as well as why an adaptive mesh is desired over a uniform mesh. Chapter 4 discusses use of measurement data from experiments with tissue phantoms, as well as data smoothing and some quantitative results/analysis of the estimation error. Finally, a model extension with a stochastic tissue stiffness parameter is considered in Chapter 5. A stochastic approximation scheme for identifying a stiffness parameter in a stochastic elasticity system is also discussed, and numerical results are presented.

# Certificate of Approval

## Ph.D Degree Dissertation

The Ph.D. Degree Dissertation of Rachel Hawks has been examined and approved by the Dissertation committee as satisfactory for the dissertation requirement for the Ph.D. degree in Mathematical Modeling

Dr. Basca Jadamba, Ph.D.

Dissertation Advisor, Committee Member

Dr. Patricia Iglesias Victoria, Ph.D.

Dissertation Chair, Committee Member

Dr. Akhtar Khan, Ph.D.

Committee Member

Dr. Miguel Sama, Ph.D.

Committee Member

Dr. Nathan Cahill, Ph.D.

Program Director

# Acknowledgements

First, I would like to thank Dr. Basca Jadamba, my PhD advisor, for her constant support throughout my journey at RIT. Along with her dedication to patiently teaching me everything I needed to know for this dissertation, it was her kindness, her insistence that I take breaks every now and then to rest and spend time with family and friends, as well as the way she always stood up for me, that kept me balanced. She is a major part of why my experience was so positive and I do not know if I can truly express how grateful I am for her. Who knew that one of the very first people I met at RIT, before I had even committed, would become my advisor. I could not have asked for a better mentor.

I also want to acknowledge my committee members, Drs. Patricia Iglesias Victoria, Akhtar Khan, and Miguel Sama. A special mention to Patricia, who was an excellent committee chair. Her kindness and enthusiasm made the defense process much more enjoyable. It was a pleasure to work with all of my committee members, and I am so appreciative of the thoughtful feedback on both my dissertation and dissertation proposal. I feel so very fortunate to have had such a wonderful committee, and I wish to thank them all for their kind words and support. Because of them, I was comforted in knowing that when I walked into my dissertation defense I was walking into a room of friendly faces.

I would also like to thank Dr. David Ross for being another mentor during my time at RIT. Another of the very first people I met before committing here, his support and advice have been invaluable. His belief in me as a mathematician and his willingness to always be on hand for advice truly made such a difference in my experience. I would also like to thank Dr. Darren Narayan who was a pleasure to TA for, always made sure I had time for my research, and was always willing to offer genuine help and guidance.

I want to acknowledge the many professors and mentors I've had from high school all the way through undergrad whose contributions were no less significant. They looked at me and already saw the amazing things I would accomplish, and I am forever grateful for

the encouragement. I would like to thank the DCC Mathematics Department and Math and Science Center for sparking my interest in mathematics and putting me on this path. I would like to whole-heartedly thank the Marist Mathematics Department for all of the fond memories and for creating the type of environment that I still brag to my friends and reminisce about. In particular, Dr. Joseph Kirtland, for having so much faith in my abilities, and Dr. Matthew Glomski, for introducing me to my fascination with mathematical biology and for encouraging me to pursue mathematics into graduate school.

I want to thank all of the friends I made at RIT, you truly made this an incredible experience. All of the time spent together, the support given and received, are memories that I will cherish forever and hope to continue even when we're spread all over the country. Your friendships mean more to me than you know.

Finally, I would like to thank my family for all of their love and support. They are my rock and I truly could not have done any of this without them. There is so much I want to thank them for. My dad, who has always had complete faith in me and pushes me to be the best I can be. My mom, who has talked me through the ups and downs of every moment of this journey and took things off my plate and on to hers when mine was overflowing. My brothers, who have my back no matter what. It's the little things that add up and make the biggest difference, and that does not go unrecognized.

The people around you are what make the difference in your experience, and I have been so fortunate to be surrounded by some wonderful individuals. To anyone who was a positive presence in my life but was not mentioned here by name, I would like them to know that their impact was felt and I am so incredibly grateful for every wonderful moment. I dedicate this dissertation to my parents, without whose support this truly would not have been possible. They helped me in reaching this goal in a million little ways and many big ones. They inspired my love of learning and education, and their belief in me is a constant source of strength I will carry with me always.

# Contents

# Chapter 1

# Introduction

This chapter contains motivations for the study, a brief review of the literature, and an introduction to an isotropic linear elasticity model for displacement as well as the direct and inverse (or parameter identification) problems associated with the model.

## 1.1 Motivation

Early detection is key in the treatment of cancer and improving patient survival rates. The current most common and effective methods for early detection of breast cancer, for example, are palpation and mammography. Though it is a standard diagnosis technique in the medical community, there are several drawbacks to using palpation, or a manual examination, to detect the presence of tumors. Palpation depends strongly on the experience and expertise of the examiner, as well as having tumors that are significantly stiffer than healthy tissue and are close to the surface of the skin. It gives a qualitative assessment of the underlying tissue stiffness, rather than a quantitative one. Tumors that are small or deep within the tissue may be unable to be detected using only palpation. Tumors found deeper in the tissue may also lack the acoustic properties necessary to be detected by ultrasound alone. Another drawback of a manual exam/palpation is that it does not provide recordable results for comparison over time (see [1]). Mammography, on the other hand, exposes the patient to radiation, which is not ideal. To reduce that risk, mammograms are typically only given regularly to certain groups of patients deemed to be at higher risk. The current recommendation is that all women receive annual mammograms beginning at age 40. However, for women with dense breast tissue, mammography may not be a viable screening tool. Ultrasound elastography could be a viable alternative in such cases and does not have the same limitations as mammography. It can be used to give doctors and patients a quantitative assessment for tumor detection not available using current palpation techniques. This research does not look to replace mammograms, but would allow early screening for a wider population (also addressed in [1]).

Elasticity imaging, or elastography, can be employed as a method of tumor identification by using imaging systems such as an ultrasound to measure tissue deformation. The idea behind elasticity imaging is to reconstruct material parameters of the tissue, specifically the tissue stiffness, from measurements of the tissue displacement. When an external, quasi-static compression is applied to the tissue, the images before and after the compression can be compared to measure the internal displacement. For example, as in Figure 1.1, the soft tissue is being compressed by an ultrasound transducer.

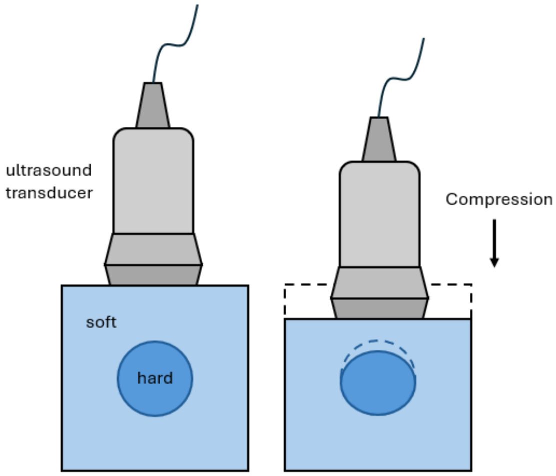


FIGURE 1.1: Tissue being compressed by an ultrasound transducer.

The device sends sound waves and receives echoes as they bounce off the body tissue and organs of the patient. These echoes are used by a computer to create an image, and those images can then be used to generate displacement data. Comparing these pre- and post-compression images to determine displacement is often done by finding the minimum or maximum of a pattern matching function, the most common of which is cross-correlation, as mentioned in [2]. The degree of tissue displacement will depend on its elasticity since changes in tissue stiffness generally correlate to pathological changes in the tissue (see [3]). Cancerous tissue is stiffer on average than healthy soft tissue. Thus, in the presence of a compressive force, the stiffer areas tend to deform less than the softer areas. Using this knowledge along with an imaging system, regions that have contrasting material properties (i.e. healthy versus cancerous tissue) can be identified and quantified.

## 1.2 Literature Review

The field of elastography emerged in the early 1990's, with contributions from many different authors such as Barbone, Gokhale, Oberai, and Ophir, to name a few. There are two types of elastography, static (or quasi-static), where a constant force is applied to the tissue, and dynamic, where the force applied to the tissue varies with time. For this research we are focusing on the isotropic, nearly incompressible, quasi-static elasticity problem. One of the first reported clinical applications of quasi-static elastography was strain imaging of breast tissue as discussed in [4]. Initial methods for solving the elasticity problem focused on strain imaging, which worked under the assumption of uniform internal stress and were obtained from differentiating the displacement data. However, while it can improve diagnosis over ultrasound alone, strain imaging is prone to artifacts in the resulting images, in part, because of the assumption of a uniform internal stress, which is also not valid in clinical settings (see [5] and [6]). Over time, it was realized that there were better, potentially more accurate methods that could be explored. There are two main approaches in elastography, direct and iterative. The direct approach involves solving the strong form of the equilibrium equations, while an iterative method reformulates the problem as a nonlinear optimization problem that uses the weak form of the equilibrium equations (see [6]). In this research we will be taking an iterative approach.

Many researchers from the mathematics and the biomedical engineering communities contributed to the advancement of the problem with deterministic models over the years. A review of the various approaches to the inverse elasticity problem can be found in [7] as well as the different types of elastography in [8]. Barbone and Oberai [9] also gave a comprehensive review of both the mathematics and the computational foundation behind biomechanical imaging in general. As discussed in [10], one of the earliest efforts towards solving the inverse problem came from Raghavan and Yagle [11] in 1994 where they used finite differences and measured strains to reconstruct the elasticity. Others such as Kallel and Bertrand [12] and Doyley and Bamber [13] also had success obtaining good reconstructions using an optimization approach. Work by Goenzen et al. [14] addresses reconstruction of linear and nonlinear elastic parameters for the purpose of identifying tumors in patients with known breast lesions, and Tyagi et al. [15] studied the reconstruction of shear modulus images using force data. Jadamba, Khan, Oberai, and Sama [16] proposed a computational framework with first- and second-order adjoint methods using finite elements for the solution of the elasticity imaging inverse problem. Arnold et al. [17] proposed methods to improve the computational efficiency by mesh adaptation and a clustering technique for the parameters describing tissue stiffness.

In recent years, inverse problems involving stochastic partial differential equations (PDEs) have become increasingly popular, and though the idea of solving a stochastic version of this problem is relatively recent, stochastic PDEs have been around for quite some time. In our case, the motivation behind considering a stochastic model is that the range for the tissue stiffness parameter is varied from patient to patient and this allows a natural uncertainty to be incorporated into the linear elasticity model we consider. On the forward problem front, one such model with nearly incompressible elasticity is considered in [18] where Young's modulus (one of the characteristic measures of tissue properties) is a spatially varying random field. The tissue stiffness parameter is therefore a random variable in such models. We consider a stochastic model similar to this in our inverse problem and use the so-called stochastic approximation method for the solution of the optimization problem under consideration.

The inverse problem of identifying stochastic parameters in stochastic PDEs (SPDEs) from the measurement of the SPDE solution attracted a great deal of attention in recent years. The most commonly adopted approach for this type of problem is a Bayesian approach which conditions a prior distribution on the coefficient function on observations of the PDE solution. The variational approach that is suitable for estimating distributed and spatially correlated parameters in SPDEs consists of formulating a stochastic optimization problem whose solution can give us information on the unknown parameter's stochasticity/statistics. The main advantage of this approach is the access to a wide-ranging collection of efficient and reliable optimization algorithms. Another advantage is a framework for convergence analysis, and easy inclusion of the parameter's structural characteristics into the inversion process. There are two main approaches for obtaining a stochastic optimization formulation in the variational case; the first one is to define an unconstrained stochastic optimization problem and the second one is to introduce a constrained stochastic optimization problem where the underlying PDE is a constraint.

The stochastic approximation method and its variations have been used in many types of optimization problems. An overview of the stochastic approximation method is found in [19] where the focus is on Kiefer–Wolfowitz (KW)-like methods which are gradient-free or stochastic zero-order algorithms, Robbins–Monro (RM)-like methods, and stochastic gradient or stochastic first-order algorithms. Notable recent works where the application is focused on optimization problems with PDEs are by Geiersbach and Pflug [20] and Martin et al [21]. A projected-gradient-type stochastic approximation algorithm was also proposed in [22] for identifying a deterministic parameter in a stochastic partial differential equation. This approach was expanded upon in [23] where a convex stochastic optimization formulation of the underlying problem is introduced and together with a stochastic Galerkin method is applied for the finite element discretization. We aren't aware of other works where stochastic approximation methods are applied along with adaptive mesh refinements in a stochastic elastography inverse problem. Much of the research works in inverse problems in stochastic partial differential equations so far have been concentrated on the various cases where the underlying PDE is scalar. The problem that we study is a vector problem and it involves a system of partial differential equations.

For both the deterministic and the stochastic versions of the model under consideration, we use a finite element method discretization and adaptively refined meshes where the mesh refinement is based on a heuristic error indicator. Adaptive mesh refinements are crucial in improving the resolution and the error in distributed parameter identification. The iterative algorithm that we use for the solution requires repeated solving of the forward problem. Hence, overall computational efficiency is greatly improved by the use of adaptive mesh refinement. Other areas for improvement in the field include data smoothing, an optimal choice of the regularizer and the regularization parameter, and error estimates for the optimization problem.

## 1.3 Isotropic Linear Elasticity System

This section will focus on the model parameters and the equations for the isotropic, nearly incompressible linear elasticity system related to deformation in solid objects under applied loads that are relevant to the application of the inverse problem in the deterministic setting.

If a body force, $f(x, y)$, is applied to an elastic object, there will be some kind of deformation. Now assume that there is a point $(x, y)$ inside the object. When the force is applied, it will move from the point $(x, y)$ to the point $(x + u_1(x, y), y + u_2(x, y))$, where the vector $\overline{\boldsymbol{u}} = \overline{\boldsymbol{u}}(\boldsymbol{x}) = \begin{bmatrix} u_1(x, y) \\ u_2(x, y) \end{bmatrix}$ is the displacement vector. The *forward problem*, also known as the direct problem, is to solve the system to be considered for the displacement vector, $\overline{\boldsymbol{u}}$, i.e. the goal is to know how every point in the domain displaces when the applied forces and material properties are known. Specifically, things that are known in the forward problem are the boundary conditions, the Lamé parameters, $\mu$ and $\lambda$, which are material properties or properties of the body, and the force applied to the body.

This system is considered isotropic because the elastic response is the same in every direction. Under the assumption of an isotropic object, the displacement satisfies the following equations:

$$\begin{aligned} -\nabla \cdot \sigma &= \boldsymbol{f} \quad \text{in} \quad D \\ \overline{\boldsymbol{u}} &= \boldsymbol{g} \quad \text{on} \quad \Gamma_1 \\ \sigma \cdot \boldsymbol{n} &= \boldsymbol{h} \quad \text{on} \quad \Gamma_2 \end{aligned} \tag{1.1}$$

where $D \in \mathbb{R}^2$ is the domain and the boundary, $\partial D$, is partitioned into two parts, $\Gamma_1$ and $\Gamma_2$. Interpretation of the Dirichlet boundary conditions imposed on $\Gamma_1$ is the prescribed displacement on this part of the boundary. For the time being, we confine ourselves to homogeneous Dirichlet boundary conditions, $\boldsymbol{g} = \mathbf{0}$ (the boundary is pinned), on the whole boundary for simplicity.

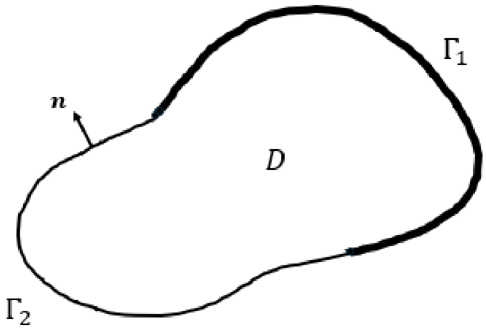


FIGURE 1.2: Generic domain showing a partition of the boundary.

In general, mixed boundary conditions are considered on $\partial D = \Gamma_1 \cup \Gamma_2$ (see Figure 1.2, for example) where $\overline{\boldsymbol{u}}$ is specified on $\Gamma_1$ and the so-called traction boundary condition $\sigma \cdot \boldsymbol{n} = \boldsymbol{h}$ is imposed on $\Gamma_2$ as shown above, where $\boldsymbol{n}$ is the outward normal vector, and $\boldsymbol{h}$ is the rule for how points along the boundary move in the normal direction.

The stress tensor $\sigma$ is a matrix that depends on the displacement, $\overline{\boldsymbol{u}}$. It is given as

$$\sigma = 2\mu\varepsilon_{\overline{\boldsymbol{u}}} + \lambda \operatorname{tr}(\varepsilon_{\overline{\boldsymbol{u}}})I. \tag{1.2}$$

Elastic properties of the object are described by the shear modulus $\mu = \mu(\boldsymbol{x})$, also called the tissue stiffness parameter, and the bulk modulus $\lambda + \mu$. The shear modulus $\mu$ describes the object's response to the shearing motion and the parameter $\lambda$ describes its compressibility.

It is also common to work with Young's Modulus, $E$, and the Poisson ratio, $\nu$, rather than $\mu$ and $\lambda$. Expressions for these parameters are as follows

$$E = \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu}, \quad \nu = \frac{\lambda}{2(\lambda + \mu)} \tag{1.3}$$

$$\mu = \frac{E}{2(1+\nu)}, \quad \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}. \tag{1.4}$$

The linearized strain tensor, $\varepsilon_{\overline{\boldsymbol{u}}}$, is also a matrix and is given as

$$\varepsilon_{\overline{\boldsymbol{u}}} = \frac{1}{2}\left(\nabla\,\overline{\boldsymbol{u}} + (\nabla\,\overline{\boldsymbol{u}})^T\right), \tag{1.5}$$

where $\nabla\,\overline{\boldsymbol{u}}$ is the Jacobian of $\overline{\boldsymbol{u}}$. The strain tensor represents the elastic response of the body. We can use this linearized version if we assume the displacement is reasonably small. Then, since $\varepsilon_{\overline{\boldsymbol{u}}}$ is linearized, $\sigma$ is also linearized. The matrices associated with this component can be considered as well,

$$\nabla\,\overline{\boldsymbol{u}} = \nabla\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \frac{\partial u_1}{\partial x} & \frac{\partial u_1}{\partial y} \\ \frac{\partial u_2}{\partial x} & \frac{\partial u_2}{\partial y} \end{bmatrix}$$

$$\varepsilon_{\overline{\boldsymbol{u}}} = \begin{bmatrix} \frac{\partial u_1}{\partial x} & \frac{1}{2}\left(\frac{\partial u_1}{\partial y} + \frac{\partial u_2}{\partial x}\right) \\ \frac{1}{2}\left(\frac{\partial u_1}{\partial y} + \frac{\partial u_2}{\partial x}\right) & \frac{\partial u_2}{\partial y} \end{bmatrix}.$$

Notice that the trace of $\varepsilon_{\overline{\boldsymbol{u}}}$ is equivalent to the divergence of $\overline{\boldsymbol{u}}$, which is a scalar equal to the sum $\frac{\partial u_1}{\partial x} + \frac{\partial u_2}{\partial y}$. So, equation (1.2) can be rewritten as

$$\sigma = 2\mu\varepsilon_{\overline{\boldsymbol{u}}} + \lambda \mathrm{div}(\overline{\boldsymbol{u}})I, \tag{1.6}$$

and the model equations become

$$-\nabla\cdot(2\mu\varepsilon_{\overline{\boldsymbol{u}}} + \lambda \mathrm{div}(\overline{\boldsymbol{u}})I) = \boldsymbol{f}.$$

To focus specifically on the problem involving tumors in soft tissue, cancerous tissues tend to be stiffer and have larger $\mu$ values than healthy tissues. The parameter $\mu$, the so-called tissue stiffness, is the parameter that needs to be identified in the inverse problem. It is a common practice in the biomedical engineering field to model the human body to be nearly incompressible. This implies that the bulk modulus will be very large compared to the shear modulus ($\lambda$ is considered to be $\infty$ for fully incompressible materials). Nearly incompressible materials are also characterized by a Poisson ratio $\nu = \dfrac{\lambda}{2(\lambda+\mu)}$ close to 0.5. A Poisson ratio value of exactly 0.5 implies that a material is incompressible, meaning the volume does not change when it is compressed, and materials with values approaching 0.5 are considered nearly or approximately incompressible. Rubber, for example, has a Poisson ratio of 0.4999, clay can have a Poisson ratio of 0.30-0.45, steel of 0.27-0.30, and concrete of 0.1-0.2. When $\lambda$ is very large ($\lambda \gg \mu$), it can be shown that $\nabla \cdot \overline{\boldsymbol{u}} \rightarrow 0$ (incompressibility is characterized by $\operatorname{div}(\overline{\boldsymbol{u}}) = 0$).

It is well-known that mesh locking (inaccurate representations of small displacements, or spurious oscillations) may occur when using linear Lagrange finite elements with the formulation described thus far. To prevent mesh locking, a mixed formulation (with elements of the Navier-Stokes equations) with a pressure variable is considered. The pressure variable $p$ is defined by $p = -\lambda \operatorname{div}(\overline{\boldsymbol{u}})$ and the system takes the following form: Find $\boldsymbol{u} = (\overline{\boldsymbol{u}}, p)$ such that

$$-\nabla \cdot (2\mu\varepsilon_{\overline{\boldsymbol{u}}} - pI) = \boldsymbol{f} \text{ in } D \tag{1.7a}$$

$$-\operatorname{div}(\overline{\boldsymbol{u}}) - \frac{1}{\lambda} p = 0 \text{ in } D \tag{1.7b}$$

$$\overline{\boldsymbol{u}} = \mathbf{0} \text{ on } \partial D \tag{1.7c}$$

when the force function $\boldsymbol{f}$ and parameters $\mu$ and $\lambda$ are given. As mentioned previously, we chose a particular set of boundary conditions, which are displayed in equation (1.7c). However, in a general setting we would have mixed boundary conditions such as those given in the system of equations (1.1), where $\overline{\boldsymbol{u}} = \boldsymbol{g}$ on $\Gamma_1$ and $\sigma \cdot \boldsymbol{n} = \boldsymbol{h}$ on $\Gamma_2$ and $\partial D = \Gamma_1 \cup \Gamma_2$. The parameter $\lambda$ can be considered as constant (it is often set to a large value such as $10^6$ in numerical computations).

## 1.4 Inverse Problem

As mentioned previously, the so-called *direct* or *forward* problem associated with the system in Section 1.3 is to find the displacement $\overline{\boldsymbol{u}}$ and pressure $p$, provided that the material parameters $\mu$ and $\lambda$, the force function $\boldsymbol{f}$, and the boundary functions $\boldsymbol{g}$ and $\boldsymbol{h}$ are given either analytically or in a discrete form such as on a finite element mesh. The *inverse problem* associated with the isotropic linear elasticity model revolves around identifying the tissue stiffness parameter $\mu$ from measurement(s) of the displacement $\overline{\boldsymbol{u}}$.

Parameter identification problems with partial differential equations (PDEs) are very common in various application problems that arise in engineering, physics, material science, and medicine. Some of these parameter identification problems focus on identifying a spatially varying parameter (or a set of scalar parameters) in a model that is governed by partial differential equations using measurements of certain quantities in the model. In addition, these parameter identification problems are often formulated as an optimization problem of minimizing a loss (or misfit) function where the underlying PDE is a constraint. Furthermore, the problems are almost always solved by employing numerical methods that provide an estimate of the unknown parameter. The inverse problem associated with the model given in (1.7a)-(1.7c) is to estimate/recover the tissue stiffness parameter $\mu(\boldsymbol{x})$ from measurement(s) of the displacement $\overline{\boldsymbol{u}}(\boldsymbol{x})$, where the force function $\boldsymbol{f}$ and the boundary data $\boldsymbol{g}, \boldsymbol{h}$ are known. We formulate the problem as a minimization problem with a scalar objective functional.

Numerical methods for direct problems using models with PDEs often require some sort of discretization (such as in finite differences, finite elements, finite volume, etc) with a computational mesh. Hence, in this discrete setting, the problem of estimating a spatially varying parameter becomes a large-scale optimization problem with thousands of unknowns as the parameter of interest is also discretized using a computational mesh. Many iterative methods for optimization problems with a PDE constraint require the numerical solution of the underlying PDE to be computed at every step. Therefore, for a high-dimensional optimization problem, it is crucial to have an accurate and efficient method for finding the numerical solution of the direct problem (the discretized PDE problem).

There are many gradient-based iterative methods available, which include methods such as steepest descent (or gradient descent) and LBFGS (limited memory Broyden–Fletcher-Goldfarb–Shanno). In these iterative methods, the updates to the approximation of $\mu$ are computed using the gradient of the objective functional. The flowchart in Figure 1.3 provides a description of a general gradient-based iterative optimization method for solving the inverse problem.

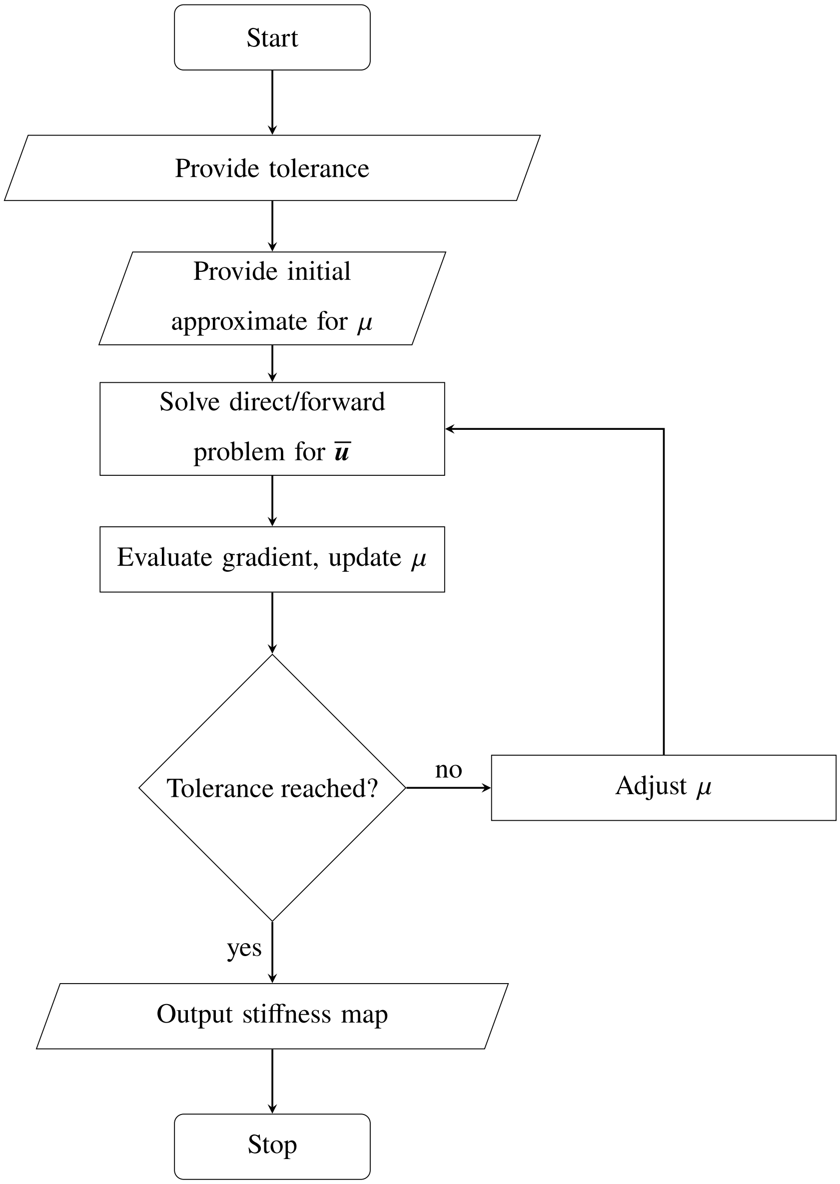


FIGURE 1.3: Process for solving the elastography inverse problem using a general gradient-based optimization algorithm.

The main factors that influence computational complexity for this problem are the high-dimensional discrete direct/forward problem being solved at every iteration and the dimension of the optimization problem which are both directly related to the resolution of the stiffness parameter. Another factor that we need to keep in mind is the accuracy of recovery and the resolution of the often sharp boundary transition between stiff tissue and surrounding soft tissue.

Our main goal is to capture this variation in the tissue stiffness parameter values, for which it is important to have a fine computational mesh capable of recovering this variation. Uniformly refining the computational mesh increases the computational cost drastically at each refinement cycle, as the dimension of the discrete PDE model and the related optimization problem increase with the refinement. Improving computational efficiency of the whole process by employing a suitable optimization method along with an adaptive mesh refinement strategy is one of the main goals of this work.

## 1.5 Adaptive Mesh Refinement

In this section, we provide more details on an important aspect of the problem which is the refinement of the computational mesh. We develop a framework using a finite element method for the solution of both the direct and inverse problems, and the computational mesh here refers to the discretization of the domain into quadrilaterals. The number of quadrilaterals in the domain is directly proportional to the dimension of the underlying constrained optimization problem. Our goal is to keep the dimension of the optimization problem as low as possible without compromising the quality of the estimation. The adaptive mesh refinement strategy is based on refining/coarsening the mesh based on the behavior of the solution (i.e. the computational mesh "adapts" to the solution) through an iterative process. The tissue stiffness parameter $\mu$ often changes rapidly between the cancerous (stiff) areas and the surrounding healthy parts of soft tissues. In order to properly capture this behavior, it may be necessary to have a very fine mesh in places where the solution of these problems are changing rapidly. However, in other places where the solution is relatively constant or not changing rapidly, such a fine mesh would be unnecessary and computationally expensive.

This is also true for the solution of the direct problem, capturing the significant changes in the displacement components. For a successful application of elastography in detecting tumors in soft tissues, we need to have an efficient algorithm for the solution of the problem in question. For the direct problem, which we explore in Chapter 2, the mesh can be refined according to the displacement. The mesh should be more refined where the displacement is changing rapidly. For the inverse problem, the mesh is refined according to the tissue stiffness parameter, $\mu$, because that is truly the quantity of interest. The use of adaptive mesh refinements not only improves the resolution of the tissue stiffness parameter in the whole domain, but also contributes to drastic savings in the the overall computational cost compared to the case where the same uniform mesh is used throughout.

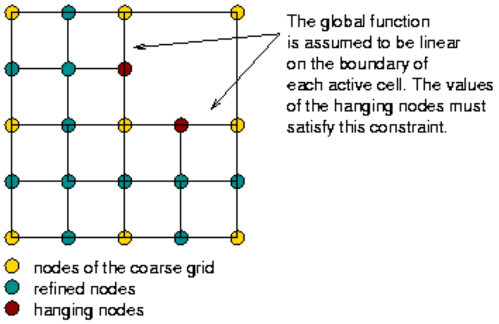


FIGURE 1.4: A mesh with hanging nodes (image from [24]).

We use the finite element library deal.II (see [24]) which offers built-in mesh refinement tools for our numerical experiments. The library uses rectangular elements (hence the tissue stiffness parameter $\mu$ is approximated by bilinear functions on each element/cell of the mesh). This tool refines the rectangular elements by using grids in such a way that neighboring elements will only differ by at most one level of refinement. This may result in what are called "hanging nodes", which are nodes that are part of a more refined grid on one side and a less refined grid on another. This means that the number of edges connected to that node or vertex are unbalanced, see Figure 1.4.

The process of obtaining a numerical solution with mesh refinement is as follows: (a) solve the optimization problem on a mesh, (b) estimate the error on each cell using an error indicator of choice and mark cells with large errors for refinement and with those with smaller errors for coarsening (use thresholds for both), (c) carry out the refinement process and get a new mesh, (c) repeat the process until the maximum number of refinement cycles is reached or the overall value of the total error indicator is sufficiently small. Since we don’t have an error estimator at hand, we opt to use a pre-programmed feature of the deal.II library for the mesh refinements called the "Kelly refinement indicator". In this heuristic error indicator (originally derived for the Poisson equation), a cell error is the integral of the jump of the gradient of the function along the faces of the cell. In the future it may be possible to use a posteriori error estimates to refine the mesh. However, without an exact solution, the previously mentioned criterion is a viable option. Another variation of this process would be to refine/coarsen the finite element mesh each time after a fixed number of iterations are completed.

We can also use different finite element meshes: one to represent the state variable (the displacement $\overline{\boldsymbol{u}}$) and another for the parameter of interest (the tissue stiffness $\mu$), where each mesh could be refined according to their respective error indicators. Another possibility is to use the same finite element mesh for both the state variable and the parameter, and refine the mesh by using an error indicator for the parameter or use some sort of combination of error indicators for the parameter and the state variable.

# Chapter 2

# Finite Element Discretization

In this chapter, we discuss the foundation for the computational framework for solving the forward and inverse problems. The various components of the finite element method (FEM) for the specific problem will be constructed, including weak formulation of the forward problem, computational mesh, basis functions, and discretization. We also present the results of numerical experiments for the forward problem to show that the underlying linear elasticity system can be solved accurately using an adaptive mesh refinement scheme.

## 2.1 Weak Formulation

We now discuss the variational form, or weak form, of the system. This is the form that is used in the finite element discretization. To find the weak form, the system of equations will be multiplied on both sides by test functions and integrated over the domain. Thus, it is necessary to multiply equation (1.7a) by $\overline{\boldsymbol{v}}$ and equation (1.7b) by $q$ and then integrate over the domain, $D$. Focusing on one equation at a time, the system of equations

$$-\nabla \cdot \sigma = \boldsymbol{f}$$

will be considered first. Note that we will derive the weak form for mixed boundary conditions and we choose

$$\overline{\boldsymbol{v}} \in \widehat{V} = \{\overline{\boldsymbol{v}} \in \left[H^1\,(D)\right]^2 : \overline{\boldsymbol{v}} = \mathbf{0} \text{ on } \Gamma_1\},$$

which means, in particular, that both components of $\overline{\boldsymbol{v}}$ belong to the Hilbert space $H^1(D)$. Then, multiplying by the test function and integrating over the domain yields the equation

$$-\int_D (\nabla \cdot \sigma) \cdot \overline{\boldsymbol{v}} = \int_D \boldsymbol{f} \cdot \overline{\boldsymbol{v}}. \tag{2.1}$$

This equation needs to be transformed to take advantage of the symmetry of the stress tensor. The stress tensor is symmetric because the problem is isotropic. Green's Theorem, the multidimensional version of integration by parts, will be used to derive the weak form.

We first need to calculate a few quantities that will help simplify the weak form. In two-dimensions,

$$\begin{aligned}\nabla \cdot (\sigma \overline{\boldsymbol{v}}) = \nabla \cdot \left( \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \right) &= \nabla \cdot \begin{bmatrix} \sigma_{11} v_1 + \sigma_{12} v_2 \\ \sigma_{21} v_1 + \sigma_{22} v_2 \end{bmatrix} \\ &= \frac{\partial}{\partial x} \left( \sigma_{11} v_1 + \sigma_{12} v_2 \right) + \frac{\partial}{\partial y} \left( \sigma_{21} v_1 + \sigma_{22} v_2 \right).\end{aligned}$$

This can be expanded using the product rule.

$$\begin{aligned} &= \frac{\partial \sigma_{11}}{\partial x} v_1 + \sigma_{11} \frac{\partial v_1}{\partial x} + \frac{\partial \sigma_{12}}{\partial x} v_2 + \sigma_{12} \frac{\partial v_2}{\partial x} + \frac{\partial \sigma_{21}}{\partial y} v_1 + \sigma_{21} \frac{\partial v_1}{\partial y} + \frac{\partial \sigma_{22}}{\partial y} v_2 + \sigma_{22} \frac{\partial v_2}{\partial y} \\ &= \left( \frac{\partial \sigma_{11}}{\partial x} + \frac{\partial \sigma_{21}}{\partial y} \right) v_1 + \left( \frac{\partial \sigma_{12}}{\partial x} + \frac{\partial \sigma_{22}}{\partial y} \right) v_2 + \sigma_{11} \frac{\partial v_1}{\partial x} + \sigma_{12} \frac{\partial v_2}{\partial x} + \sigma_{21} \frac{\partial v_1}{\partial y} + \sigma_{22} \frac{\partial v_2}{\partial y} \\ &= \begin{bmatrix} \frac{\partial \sigma_{11}}{\partial x} + \frac{\partial \sigma_{21}}{\partial y} \\ \frac{\partial \sigma_{12}}{\partial x} + \frac{\partial \sigma_{22}}{\partial y} \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial v_1}{\partial x} & \frac{\partial v_2}{\partial x} \\ \frac{\partial v_1}{\partial y} & \frac{\partial v_2}{\partial y} \end{bmatrix} \\ &= \left( \nabla \cdot \sigma^T \right) \cdot \overline{\boldsymbol{v}} + \sigma \cdot \left( \nabla \, \overline{\boldsymbol{v}} \right)^T. \end{aligned}$$

So, $\nabla \cdot (\sigma \overline{\boldsymbol{v}}) = \left( \nabla \cdot \sigma^T \right) \cdot \overline{\boldsymbol{v}} + \sigma \cdot \left( \nabla \, \overline{\boldsymbol{v}} \right)^T$. Since $\sigma$ is symmetric, $\sigma = \sigma^T$ and $\nabla \cdot \sigma = \nabla \cdot \sigma^T$. Also, because the dot product is just a number, $\sigma \cdot \left( \nabla \, \overline{\boldsymbol{v}} \right)^T = \sigma \cdot \nabla \, \overline{\boldsymbol{v}}$. Then, $\sigma \cdot \nabla \, \overline{\boldsymbol{v}}$ could be rewritten as,

$$\begin{aligned} \sigma \cdot \nabla \, \overline{\boldsymbol{v}} &= \frac{1}{2} \sigma \cdot \nabla \, \overline{\boldsymbol{v}} + \frac{1}{2} \sigma \cdot \nabla \, \overline{\boldsymbol{v}} \\ &= \frac{1}{2} \sigma \cdot \nabla \, \overline{\boldsymbol{v}} + \frac{1}{2} \sigma^T \cdot \left( \nabla \, \overline{\boldsymbol{v}} \right)^T \\ &= \frac{1}{2} \sigma \cdot \nabla \, \overline{\boldsymbol{v}} + \frac{1}{2} \sigma \cdot \left( \nabla \, \overline{\boldsymbol{v}} \right)^T \\ &= \sigma \cdot \frac{1}{2} \left( \nabla \, \overline{\boldsymbol{v}} + \left( \nabla \, \overline{\boldsymbol{v}} \right)^T \right) \\ &= \sigma \cdot \varepsilon_{\overline{\boldsymbol{v}}}, \end{aligned}$$

where $\varepsilon_{\overline{\boldsymbol{v}}}$ is the strain. Recall that the strain tensor represents the elastic response of the body. Note that $\varepsilon_{\overline{\boldsymbol{u}}}$ and $\varepsilon_{\overline{\boldsymbol{v}}}$ have similar forms because $\overline{\boldsymbol{v}}$ is the test function. Now,

$$\nabla \cdot (\sigma \overline{\boldsymbol{v}}) = (\nabla \cdot \sigma) \cdot \overline{\boldsymbol{v}} + \sigma \cdot \varepsilon_{\overline{\boldsymbol{v}}}.$$

This equation can then be integrated over $D$,

$$\int_D \nabla \cdot (\sigma \overline{\boldsymbol{v}}) = \int_D (\nabla \cdot \sigma) \cdot \overline{\boldsymbol{v}} + \int_D \sigma \cdot \varepsilon_{\overline{\boldsymbol{v}}}.$$

The goal is to write the first term on the right-hand side in terms of the other two terms because that is the left hand side of equation (2.1), which is where the derivation of the weak form was left. The divergence theorem states $\int_D \nabla \cdot \sigma = \int_{\partial D} \sigma \boldsymbol{n} \, dS$. So, using the divergence theorem, the previous equation becomes

$$\int_{\partial D} (\sigma \overline{\boldsymbol{v}}) \cdot \boldsymbol{n} \, dS = \int_D (\nabla \cdot \sigma) \cdot \overline{\boldsymbol{v}} + \int_D \sigma \cdot \varepsilon_{\overline{\boldsymbol{v}}}.$$

Using again the fact that $\sigma$ is symmetric,

$$\int_{\partial D} (\sigma \boldsymbol{n}) \cdot \overline{\boldsymbol{v}} \, dS = \int_D (\nabla \cdot \sigma) \cdot \overline{\boldsymbol{v}} + \int_D \sigma \cdot \varepsilon_{\overline{\boldsymbol{v}}}.$$

Rearranging the terms yields,

$$-\int_D (\nabla \cdot \sigma) \cdot \overline{\boldsymbol{v}} = \int_D \sigma \cdot \varepsilon_{\overline{\boldsymbol{v}}} - \int_{\partial D} (\sigma \boldsymbol{n}) \cdot \overline{\boldsymbol{v}} \, dS.$$

Next, $\sigma$ can be rewritten in terms of (1.7a) in the first term on the right.

$$-\int_D (\nabla \cdot \sigma) \cdot \overline{\boldsymbol{v}} = \int_D (2\mu \varepsilon_{\overline{\boldsymbol{u}}} - pI) \cdot \varepsilon_{\overline{\boldsymbol{v}}} - \int_{\partial D} (\sigma \boldsymbol{n}) \cdot \overline{\boldsymbol{v}} \, dS$$

$$-\int_D (\nabla \cdot \sigma) \cdot \overline{\boldsymbol{v}} = \int_D 2\mu \varepsilon_{\overline{\boldsymbol{u}}} \cdot \varepsilon_{\overline{\boldsymbol{v}}} - \int_D pI \cdot \varepsilon_{\overline{\boldsymbol{v}}} - \int_{\partial D} (\sigma \boldsymbol{n}) \cdot \overline{\boldsymbol{v}} \, dS$$

Another portion of this equation can be simplified since,

$$\begin{aligned} pI \cdot \varepsilon_{\vec{v}} &= \begin{bmatrix} p & 0 \\ 0 & p \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial v_1}{\partial x} & \frac{1}{2}\left(\frac{\partial v_1}{\partial y} + \frac{\partial v_2}{\partial x}\right) \\ \frac{1}{2}\left(\frac{\partial v_1}{\partial y} + \frac{\partial v_2}{\partial x}\right) & \frac{\partial v_2}{\partial y} \end{bmatrix} \\ &= p\left(\frac{\partial v_1}{\partial x}\right) + p\left(\frac{\partial v_2}{\partial y}\right) \\ &= p\left(\frac{\partial v_1}{\partial x} + \frac{\partial v_2}{\partial y}\right) \\ &= p \operatorname{div}(\overline{\boldsymbol{v}}). \end{aligned}$$

Thus, rearranging slightly and using equation (2.1),

$$\int_D 2\mu\varepsilon_{\overline{\boldsymbol{u}}} \cdot \varepsilon_{\overline{\boldsymbol{v}}} - \int_D p \operatorname{div}(\overline{\boldsymbol{v}}) = \int_D \boldsymbol{f} \cdot \overline{\boldsymbol{v}} + \int_{\partial D} (\sigma \boldsymbol{n}) \cdot \overline{\boldsymbol{v}}\, dS. \tag{2.2}$$

Since $\overline{\boldsymbol{v}} = \boldsymbol{0}$ on $\Gamma_1$ due to the homogeneous Dirichlet boundary condition, the second integral on the right-hand side of equation (2.2) will be zero on $\Gamma_1$. Now, the problem is to find $\overline{\boldsymbol{u}}$ such that the equation

$$\int_D 2\mu\varepsilon_{\overline{\boldsymbol{u}}} \cdot \varepsilon_{\overline{\boldsymbol{v}}} - \int_D p \operatorname{div}(\overline{\boldsymbol{v}}) = \int_D \boldsymbol{f} \cdot \overline{\boldsymbol{v}} + \int_{\Gamma_2} (\sigma \boldsymbol{n}) \cdot \overline{\boldsymbol{v}}\, dS$$

is true for all $\overline{\boldsymbol{v}} \in \widehat{V}$.

Note that the second term on the right only matters with nonhomogeneous traction boundary conditions; the term will vanish if no boundary traction is applied to $\Gamma_2$. In the case where homogeneous Dirichlet boundary conditions are applied on the whole boundary, the test function $\overline{\boldsymbol{v}}$ would belong to the space $\widehat{V} = \left[H_0^1(D)\right]^2$. This boundary condition is also known as a clamped boundary condition because it implies no displacement, or no movement, along the boundary of the object. The weak form, in this case, would be that $\overline{\boldsymbol{u}}$ satisfies

$$\int_D 2\mu\varepsilon_{\overline{\boldsymbol{u}}} \cdot \varepsilon_{\overline{\boldsymbol{v}}} - \int_D p \operatorname{div}(\overline{\boldsymbol{v}}) = \int_D \boldsymbol{f} \cdot \overline{\boldsymbol{v}}$$

for all $\overline{\boldsymbol{v}} \in \left[H_0^1(D)\right]^2$.

Next, the weak form of equation (1.7b) becomes

$$\int_D -q \operatorname{div}(\overline{\boldsymbol{u}}) - \frac{1}{\lambda} pq = 0.$$

Combining the two equations, with mixed boundary conditions, the weak form is

$$\int_D \left(2\mu\varepsilon_{\overline{\boldsymbol{u}}} \cdot \varepsilon_{\overline{\boldsymbol{v}}} - p \operatorname{div}(\overline{\boldsymbol{v}}) - q \operatorname{div}(\overline{\boldsymbol{u}}) - \frac{1}{\lambda} pq\right) = \int_D \boldsymbol{f} \cdot \overline{\boldsymbol{v}} + \int_{\Gamma_2} \boldsymbol{h} \cdot \overline{\boldsymbol{v}}\, dS. \tag{2.3}$$

With homogeneous Dirichlet boundary conditions the second term on the right of (2.3) will disappear. Therefore, the weak formulation of the problem (1.7a)-(1.7c) is to find $\boldsymbol{u} = (\overline{\boldsymbol{u}}, p) \in \widehat{V} \times Q = [H_0^1(D)]^2 \times L^2(D)$ such that

$$\int_D 2\mu\varepsilon_{\overline{\boldsymbol{u}}} \cdot \varepsilon_{\overline{\boldsymbol{v}}} - \int_D p \operatorname{div}(\overline{\boldsymbol{v}}) = \int_D \boldsymbol{f}\, \overline{\boldsymbol{v}} \text{ for all } \overline{\boldsymbol{v}} \in \widehat{V}, \tag{2.4}$$

$$\int_D q \operatorname{div}(\overline{\boldsymbol{u}}) + \int_D \frac{1}{\lambda} pq = 0 \text{ for all } q \in Q, \tag{2.5}$$

where $\widehat{V} = \left\{ \overline{\boldsymbol{v}} \in \left[H^1(D)\right]^2, \overline{\boldsymbol{v}} = \mathbf{0} \text{ on } D \right\}$ and $Q = L^2(D)$. We also define a trilinear form $T$ by

$$T(\mu, \boldsymbol{u}, \boldsymbol{v}) = a(\mu, \overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}) + b(\overline{\boldsymbol{v}}, p) - b(\overline{\boldsymbol{u}}, q) + c(p, q) \tag{2.6}$$

which will be used later, and the weak formulation is rewritten as

$$\begin{aligned} a(\mu, \overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}) + b(\overline{\boldsymbol{v}}, p) &= m(\overline{\boldsymbol{v}}) \\ b(\overline{\boldsymbol{u}}, q) - c(p, q) &= 0 \end{aligned} \tag{2.7}$$

where

$$a(\mu, \overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}) = \int_D 2\mu\varepsilon_{\overline{\boldsymbol{u}}} \cdot \varepsilon_{\overline{\boldsymbol{v}}}, \quad b(\overline{\boldsymbol{v}}, p) = -\int_D p \operatorname{div}(\overline{\boldsymbol{v}})$$
$$c(p, q) = \int_D \frac{1}{\lambda} pq, \quad m(\overline{\boldsymbol{v}}) = \int_D f\overline{\boldsymbol{v}}.$$

Note that $a(\mu, \overline{\boldsymbol{u}}, \overline{\boldsymbol{v}})$, $b(\overline{\boldsymbol{v}}, p)$, and $c(p, q)$ are linear in all of their components, and in order to get the trilinear form shown in equation (2.6) we simply subtract the second equation in (2.7) from the first since adding or subtracting by zero will not make a difference mathematically.

## 2.2 Mesh

Our mesh, or triangulation, will be $\mathcal{T}_h$, where $h$ is the mesh size. Each element is labeled $T_i$ for $i = 1, 2, ..., N_t$, where $N_t$ it the total number of elements (e.g. $T_1$ is element 1). The vertices or nodes are also labeled, $z_i$ for $i = 1, 2, ..., N_p$ where $N_p$ is the total number of vertices. If triangular elements are chosen, the step size can be defined as $h = \max(\text{diam}(T_i))$, or the maximum of the diameters from all of the triangles. The diameter of a triangle is defined by the triangle's longest side. This definition is important if the mesh is made up of triangles of different sizes. For rectangular elements, which are the type of elements we use, the step size is defined as the longest distance between vertices, or the diagonal length.

For second-order boundary value problems (in one dimension), the function $u(x)$ is typically approximated as a piecewise linear function on each element or interval $I_i$. In elliptic PDEs in two dimensions, each component of the displacement vector $u_j(x, y)$ can still be approximated as a piecewise linear function, but this time it will be linear on each triangular element $T_i$, for example, so the approximations consist of triangular planes. The representation of $u_j$ on $T_i$ is $u_j = a_i + b_i x + c_i y$ for $(x, y) \in T_i$. Each triangle will have an equation like this. However, that is a significant amount of unknowns. Fortunately, with the way the mesh was set up, each triangle shares at least one common edge, and the functions must match on the common edges in order to represent $u_j$ continuously. This will reduce the number of unknowns. Thus, because of this continuity restriction, only $N_p$ unknowns (the number of vertices) are needed to represent the function, rather than three unknowns per triangle. Furthermore, if there are Dirichlet boundary conditions along the entire boundary, then those values along the boundary will be known. The explanation and reasoning is similar for rectangular elements. So the number of unknowns in each case would be the number of interior nodes. Similar to the one-dimensional case, all that is needed to completely define the function $u_j$ are the values at the nodes.

## 2.3 Basis Functions

Now the basis functions need to be considered. In two dimensions the basis functions, $\psi_i$, are tent-shaped and have a value of 1 at $z_i$ (vertex $i$) and a value of 0 at every other node. Another way to divide the domain into a mesh is with rectangular, or bilinear, elements instead of triangular ones. These elements can be labeled similarly to the triangular ones. The difference will be in the degrees of freedom. Since a rectangle has four vertices, a polynomial with four degrees of freedom is needed, rather than the three degrees of freedom necessary for triangular elements. The representation of $u_j$ on each element will now have the form $u_j = a_i + b_i x + c_i y + d_i xy$. This is referred to as a bilinear polynomial, and gives the equation for a rectangular plane on each element. This bilinear function will reduce to a linear function of one variable on the edges of the rectangular element. Again, the fact that most of the elements share common edges will reduce the number of unknowns. Only $N_p$ unknowns are still necessary to uniquely represent the function, just as with the triangular elements. In both cases, only the nodal values are needed to completely define the function $u_j$. The basis functions for rectangular elements are similar to those for triangular elements in that they will still have a value of 1 at the node $z_i$, and a value of zero everywhere else, but will be pyramid shaped instead of being tent-shaped.

For quadratic tent-shaped basis functions, $u_j$ is now represented on $T_i$ with the equation $u_j = a_i + b_i x + c_i y + d_i x^2 + e_i xy + f_i y^2$ for $(x, y) \in T_i$. Now, the triangular plane on each element is represented by a quadratic equation of two variables, where each edge of the plane will be a parabola. However, now it is no longer sufficient to represent the element using only the values at the nodes. While a line can be fully represented with two points, a parabola cannot. So, the midpoints of each edge of the element are also needed. Thus, six points are needed to completely define a single element. There will be six unknowns per element, but this can be reduced due to the continuity restriction discussed in Section 2.2.

## 2.4 Direct Problem

In order to implement this problem numerically, the components of the linear system that will be solved by the finite element method must be compiled. As a reminder, the problem being considered is

$$\begin{aligned} -\nabla \cdot (2\mu\varepsilon(\overline{\boldsymbol{u}}) - pI) &= \boldsymbol{f} \;\text{ in }\; D \\ -\operatorname{div}(\overline{\boldsymbol{u}}) - \frac{p}{\lambda} &= 0 \;\text{ in }\; D \\ \overline{\boldsymbol{u}} &= \boldsymbol{0} \;\text{ on }\; \partial D. \end{aligned}$$

We choose a finite element mesh and define finite dimensional subspaces $V_h = \widehat{V}_h \times Q_h$, and $\widetilde{L}_h$, where $\widehat{V}_h$ is the space for $\overline{u}$ and $\overline{v}$, $Q_h$ is the space for $p$ and $q$, and $\widetilde{L}_h$ is the space for $\mu$. Finite dimensional subspaces $\widehat{V}_h$ and $Q_h$ are defined as such that they satisfy the Ladyzhenskaya-Babuska-Brezzi (LBB) conditions. The discrete version of the problem reads: Given $\mu_h \in \widetilde{L}_h$, find $(\overline{\boldsymbol{u}}_h, p_h) \in \widehat{V}_h \times Q_h$ such that for every $\boldsymbol{v}_h = (\overline{\boldsymbol{v}}_h, q_h)$ we have

$$\begin{aligned} \int_D 2\mu_h\varepsilon(\overline{\boldsymbol{u}}_h) \cdot \varepsilon(\overline{\boldsymbol{v}}_h) - \int_D p_h \operatorname{div}(\overline{\boldsymbol{v}}_h) = \int_D f\overline{\boldsymbol{v}}_h \;\text{ for }\; \overline{\boldsymbol{v}}_h \in \widehat{V}_h \\ \int_D q_h \operatorname{div}(\overline{\boldsymbol{u}}_h) + \int_D \frac{1}{\lambda} p_h q_h = 0 \;\text{ for all }\; q_h \in Q_h \end{aligned} \tag{2.8}$$

To simplify notation, the weak form could be written abstractly using the bilinear forms and linear functional defined earlier. Thus, the discrete variational form can be written as

$$\begin{aligned} a(\mu, \overline{\boldsymbol{u}}_h, \overline{\boldsymbol{v}}_h) + b(\overline{\boldsymbol{v}}_h, p) &= m(\overline{\boldsymbol{v}}_h) \\ b(\overline{\boldsymbol{u}}_h, q) - c(p, q) &= 0. \end{aligned} \tag{2.9}$$

The basis functions need to be reconsidered next. In Section 2.3, the basis functions introduced were for the scalar case. This problem involves vector equations, so the basis functions must be adjusted accordingly. In the scalar case, there were $N_f$ basis functions, where $N_f$ is the number of free nodes. A free node is any node not included in a Dirichlet boundary condition. These nodes are considered "free" because, in the forward problem, the value of the function $\overline{\boldsymbol{u}}$ is unknown. The nodes contained within a Dirichlet boundary are considered constrained nodes, $N_c$, because the values of $\overline{\boldsymbol{u}}$ will be known.

Since there are two components in the vector $\overline{\boldsymbol{u}}$, the basis functions must have two components as well in order to properly represent the displacement. Thus, there will be twice as many basis functions as the scalar case. The set of basis functions for this problem is $\left\{\overline{\psi}_1, ..., \overline{\psi}_{N_f}, \overline{\psi}_{N_{f+1}}, ..., \overline{\psi}_{2N_f}\right\}$, where

$$\begin{aligned} \overline{\psi}_i &= \begin{bmatrix} \psi_i \\ 0 \end{bmatrix} \\ \overline{\psi}_{N_f+i} &= \begin{bmatrix} 0 \\ \psi_i \end{bmatrix} \end{aligned}$$

for $i = 1, ..., N_f$ and $\psi_i$ is the typical tent- or pyramid-shaped basis function for 2D problems discussed in the previous sections. So, $\overline{\psi}_i$ is the first component of the solution function and $\overline{\psi}_{N_f+i}$ is the second component of the solution function. Since the discrete solution $\overline{\boldsymbol{u}}_h \in \widehat{V}_h$, it can be expressed as a linear combination of the basis functions. So, $\overline{\boldsymbol{u}}_h = \sum_{i=1}^{2N_f} \overline{U}_i \psi_i$ where $\overline{\boldsymbol{U}}$ is the solution vector.

The structure and basis functions for each piece of that linear system will be examined next. The discrete representation for the solution vector, the pressure $p$, and the load vector have the forms

$$\overline{\boldsymbol{U}} = \begin{bmatrix} \overline{U}_1 \\ \vdots \\ \overline{U}_n \end{bmatrix}, \ \boldsymbol{P} = \begin{bmatrix} P_1 \\ \vdots \\ P_k \end{bmatrix}, \ \boldsymbol{F} = \begin{bmatrix} \overline{\boldsymbol{F}} \\ \boldsymbol{0} \end{bmatrix}$$

where the basis functions for $\overline{\boldsymbol{u}}_h$ are $\psi_i$, with $i = 1, ..., n$, and $\overline{\boldsymbol{u}}_h = \sum_{i=1}^{n} \overline{U}_i \psi_i$, the basis functions for $p_h$ are $\chi_i$, with $i = 1, ..., k$, and $\overline{p}_h = \sum_{i=1}^{k} P_i \chi_i$, both $\psi_i$, $\chi_i$ are the basis for the finite dimensional subspace $V_h$ of $V$ corresponding to a chosen finite element mesh, and $\boldsymbol{F}$ is a block vector. One possible combination of basis functions is piecewise quadratic basis functions for the displacement $\overline{\boldsymbol{u}}$ and piecewise linear basis functions for the pressure $p$. This is the combination used in most of the numerical experiments included in this work, but there are other stable combinations. For example, a piecewise quadratic – piecewise constant combination of elements was used for some numerical experiments in one of the later chapters. In order to define the discrete versions of both the objective functional and the gradient in later chapters, we need to discuss the basis functions for one more piece of the linear system. We need a representation $\mu_h = \sum_{i=1}^{m} M_i \phi_i$, where $\phi_i$ are the basis functions corresponding to the finite-dimensional subspace $\widetilde{L}_h$, $M_i$ are the nodal values of the stiffness variable, with $i = 1, ..., m$, and therefore the discrete representation for $\mu_h$ is

$$\boldsymbol{M} = \begin{bmatrix} M_1 \\ \vdots \\ M_m \end{bmatrix}.$$

Note that $\overline{\boldsymbol{U}} \in \mathbb{R}^{2N_f}$, $\boldsymbol{F} \in \mathbb{R}^{2N_f}$, and $K \in \mathbb{R}^{2N_f \times 2N_f}$. For a given vector $\boldsymbol{M}$ representing $\mu_h$, the forward problem is then to find the solution vector $\boldsymbol{U} = [\overline{\boldsymbol{U}}, \boldsymbol{P}]^\top$ of a linear system $K(\boldsymbol{M})\boldsymbol{U} = \boldsymbol{F}$. The stiffness matrix $K$ depends on $\boldsymbol{M}$ and, due to the structure of the basis functions, will be a block matrix defined by

$$K(\boldsymbol{M}) = \begin{bmatrix} A(\boldsymbol{M}) & B \\ B^T & C \end{bmatrix}$$

where we emphasized the dependence of the top-left block $A$ of matrix $K$ on the vector $\boldsymbol{M}$ corresponding to the discretization of $\mu_h$. Note that $A(\boldsymbol{M})$ is $n \times n$, $B$ is $n \times k$, and $C$ is $k \times k$. Entries of the block matrix $K(\boldsymbol{M})$ and the block vector $\boldsymbol{F} = [\overline{\boldsymbol{F}}, 0]^\top$ are defined using the basis functions and forms $a(\cdot,\cdot,\cdot), b(\cdot,\cdot), c(\cdot,\cdot), m(\cdot)$ defined earlier. For example,

$$[A(\boldsymbol{M})]_{ij} = a(\mu_h, \psi_i, \psi_j) \text{ for } i, j = 1, ..., n$$

Thus, the linear system in block matrix form is

$$\begin{bmatrix} A(\boldsymbol{M}) & B \\ B^T & C \end{bmatrix} \begin{bmatrix} \overline{\boldsymbol{U}} \\ \boldsymbol{P} \end{bmatrix} = \begin{bmatrix} \overline{\boldsymbol{F}} \\ \boldsymbol{0} \end{bmatrix}$$

where the entries of the matrix blocks correspond to different pieces of the weak form (equations (2.4) and (2.5)). The matrix $A(\boldsymbol{M})$ corresponds with the first term on the left of (2.4) (the first part of the weak formulation), block $B$ corresponds with the second term on the left of that same equation, $B^T$ with the first term on the left of (2.5) (the second part of the weak formulation), and $C$ with the second term on the left of (2.5). Those entries would then have the form,

$$A_{ij} = \int_D 2\mu\varepsilon_{\bar{\psi}_j} \cdot \varepsilon_{\bar{\psi}_i}, \; B_{ij} = -\int_D \chi_j \text{div}(\bar{\psi}_i), \; C_{ij} = -\frac{1}{\lambda}\int_D \chi_j \chi_i$$

where $\psi_i$ and $\psi_j$ are the basis functions corresponding to the displacement $\overline{\boldsymbol{u}}$ and its test function $\overline{\boldsymbol{v}}$, and $\chi_i$ and $\chi_j$ are the basis functions corresponding to the pressure $p$ and its test function $q$.

This linear system will be solved component by component by using the Schur complement as follows. Expanding of the left-hand side of the system yields the equations

$$A\overline{\boldsymbol{U}} + B\boldsymbol{P} = \overline{\boldsymbol{F}}$$

$$B^T\,\overline{\boldsymbol{U}} + C\boldsymbol{P} = \boldsymbol{0}$$

The first equation will then be solved for $\overline{\boldsymbol{U}}$ and plugged into the second equation to solve for $\boldsymbol{P}$. Thus,

$$\begin{aligned}\overline{\boldsymbol{U}} &= A^{-1}(\overline{\boldsymbol{F}} - B\boldsymbol{P}) \\ B^T A^{-1}(\overline{\boldsymbol{F}} - B\boldsymbol{P}) + C\boldsymbol{P} &= \mathbf{0} \\ B^T A^{-1}\overline{\boldsymbol{F}} - B^T A^{-1} B\boldsymbol{P} + C\boldsymbol{P} &= \mathbf{0} \\ (B^T A^{-1} B - C)\boldsymbol{P} &= B^T A^{-1}\overline{\boldsymbol{F}}\end{aligned}$$

where $(B^T A^{-1} B - C)$ is the Schur complement. Finally, the last equation is solved for $\boldsymbol{P}$, which is then plugged into the equation $\overline{\boldsymbol{U}} = A^{-1}(\overline{\boldsymbol{F}} - B\boldsymbol{P})$ to solve for $\overline{\boldsymbol{U}}$.

## 2.5 Numerical Experiments

In this section, we make a series of numerical experiments that are aimed towards computing the displacements accurately given the specific tissue stiffness parameter $\mu$ and the body force $\boldsymbol{f}$ where the mesh is refined locally by using a heuristic error estimator. All simulations are done using the finite element library deal.II (again, see [24]).

**Experiment 1**. In this example, modified from [25], the domain is $D = [0,1]^2$ and homogeneous Dirichlet boundaries were chosen for the whole boundary. This indicates that there is no movement on the boundaries (i.e. they are pinned). We chose a problem with an analytical solution, so that the error in the solution could be accurately computed. The displacement vector is given by $\overline{\boldsymbol{u}} = \begin{bmatrix} u_1(x,y) \\ u_2(x,y) \end{bmatrix}$ where

$$\begin{aligned}u_1(x,y) &= 0.5\pi\cos(\pi y)\sin(\pi y)\sin^2(\pi x) \\ u_2(x,y) &= -\pi\cos(\pi x)\sin(\pi x)\sin^2(\pi y).\end{aligned}$$

Setting the constant $\nu$, we compute $\mu$ and $\lambda$ as well as the expression for the body force $\boldsymbol{f}(x,y)$ and supply these as data for the problem. The values chosen for these numerical experiments represent the nearly incompressible case, meaning we will choose a Poisson ratio value, $\nu$, approaching 0.5. Components of the displacement $\overline{\boldsymbol{u}}$ as well as a vector view are shown in Figure 2.1.

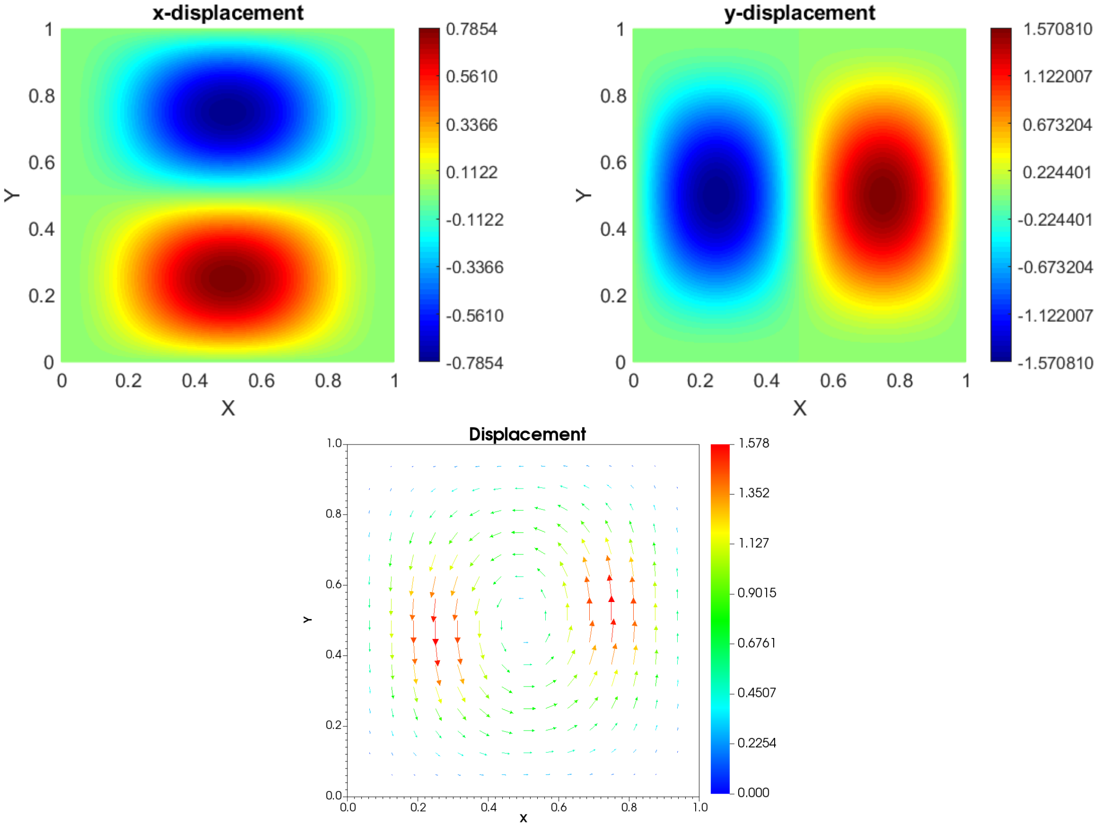


FIGURE 2.1: Components of the displacement vector $\overline{\boldsymbol{u}}$ (top) and a vector view of $\overline{\boldsymbol{u}}$ (bottom) for Experiment 1.

The $L^2$-norm error of the difference of the approximate solution and the exact solution was computed for a range of $\nu$ values. For example, $\nu = 0.4999$ corresponds to $\mu \approx 0.3334$ and $\lambda \approx 1666.4$, and $\nu = 0.49999$ corresponds to $\mu \approx 0.333336$ and $\lambda \approx 16666.4$. Young's modulus $E$ is set to the constant 1 in all computations. Tables 2.1 and 2.2 show the $L^2$-norm errors for the displacement $\overline{\boldsymbol{u}}$ and the error reduction factors for simulations with $\nu = 0.4999$ and $\nu = 0.49999$. The mesh is refined uniformly in each cycle so that the mesh size is decreased by a factor of 2 in each cycle. The purpose of this example is to ensure the accuracy of the solver.

| cycle | # cells | $h/\sqrt{2}$ | # dofs | $L^2$-norm error | factor |
|---|---|---|---|---|---|
| 0 | 64 | 1/8 | 659 | 1.065e+00 | - |
| 1 | 256 | 1/16 | 2467 | 6.503e-02 | 16.37 |
| 2 | 1024 | 1/32 | 9539 | 4.040e-03 | 16.10 |
| 3 | 4096 | 1/64 | 37507 | 2.532e-04 | 15.96 |
| 4 | 16384 | 1/128 | 148739 | 1.645e-05 | 15.39 |

TABLE 2.1: $L^2$-norm errors for the displacement $\overline{\boldsymbol{u}}$ for Experiment 1, where $\nu = 0.4999$.

| cycle | # cells | $h/\sqrt{2}$ | # dofs | $L^2$-norm error | factor |
|---|---|---|---|---|---|
| 0 | 64 | 1/8 | 659 | 1.065e+01 | - |
| 1 | 256 | 1/16 | 2467 | 6.504e-01 | 16.37 |
| 2 | 1024 | 1/32 | 9539 | 4.040e-03 | 16.10 |
| 3 | 4096 | 1/64 | 37507 | 2.532e-03 | 15.96 |
| 4 | 16384 | 1/128 | 148739 | 1.645e-04 | 15.39 |

TABLE 2.2: $L^2$-norm errors for the displacement $\overline{\boldsymbol{u}}$ for Experiment 1, where $\nu = 0.49999$.

We observe that the error decreases, as expected, as the mesh is refined. When using quadratic elements to represent the displacement $\overline{\boldsymbol{u}}$, the $L^2$-norm error in the displacement $\overline{\boldsymbol{u}}$ decreases by a factor of about 16 as $h$ is decreased by a factor of 2. In general, for smooth data and solution, the approximation $\overline{\boldsymbol{u}}_h$ of the displacement $\overline{\boldsymbol{u}}$ satisfies the error estimate

$$||\overline{\boldsymbol{u}} - \overline{\boldsymbol{u}}_h|| \leq Ch^3 \tag{2.10}$$

with a constant $C$ independent of the mesh size $h$ when using $Q_2 - Q_1$ elements (see, for example, [26]) (the notation $Q_2 - Q_1$ simply means that piecewise quadratic elements were used to represent the displacement and piecewise linear elements were used to represent the pressure). However, for this particular example, we observe that the error behaves more like $O(h^4)$. This is due to the superconvergence phenomenon where the finite element approximation of the solution converges to the true solution at a higher rate than the expected. The occurrence of superconvergence may be due to properties of the domain, solution properties, the finite element approximation, or the quadrature formulas used.

For the next several experiments, taken from a tutorial example in deal.II [24], we consider the linear elasticity system on the domain $D = [-1, 1]^2$. The component $f_1(x, y)$ of the force on the body, $\boldsymbol{f}$, is defined by

$$f_1(x, y) = \begin{cases} 1, & \text{if } (x + 0.5)^2 + y^2 \leq 0.2^2 \ \text{ or } \ (x - 0.5)^2 + y^2 \leq 0.2^2 \\ 0, & \text{else} \end{cases}$$

and the component $f_2(x, y)$ is defined by

$$f_2(x, y) = \begin{cases} 1, & \text{if } x^2 + y^2 \leq 0.2^2 \\ 0, & \text{else.} \end{cases}$$

This type of body force implies pressure being uniformly applied on the disks shown in Figure 2.2. For the two disks in the left figure, a unit force in the $x$-direction is applied. For the disk centered at the origin (figure on the right), a unit force is applied in the $y$-direction.

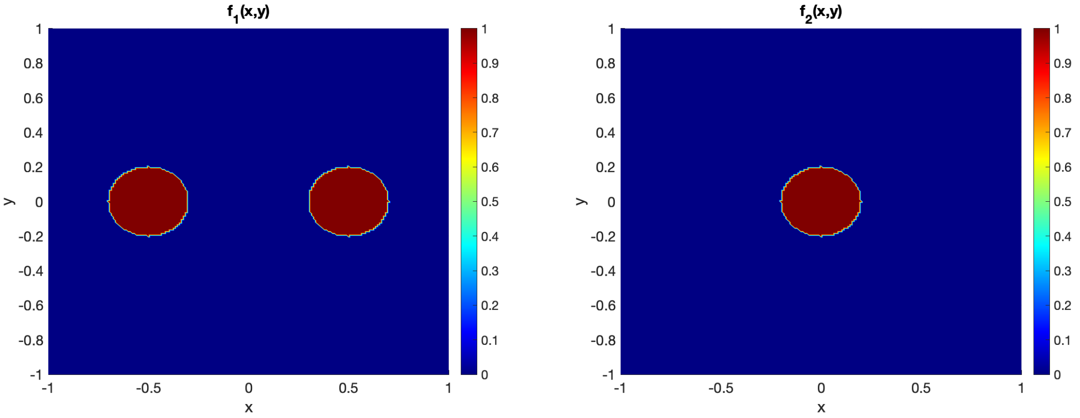


FIGURE 2.2: Components of the body force function $\boldsymbol{f}$.

**Experiment 2**. The tissue stiffness parameter $\mu$ for this example is a constant 1 everywhere in the domain. In Figure 2.3, we show the components of the displacement vector $\overline{\boldsymbol{u}}$ and vector view of the displacement which show the effects of the applied body force $\boldsymbol{f}$.

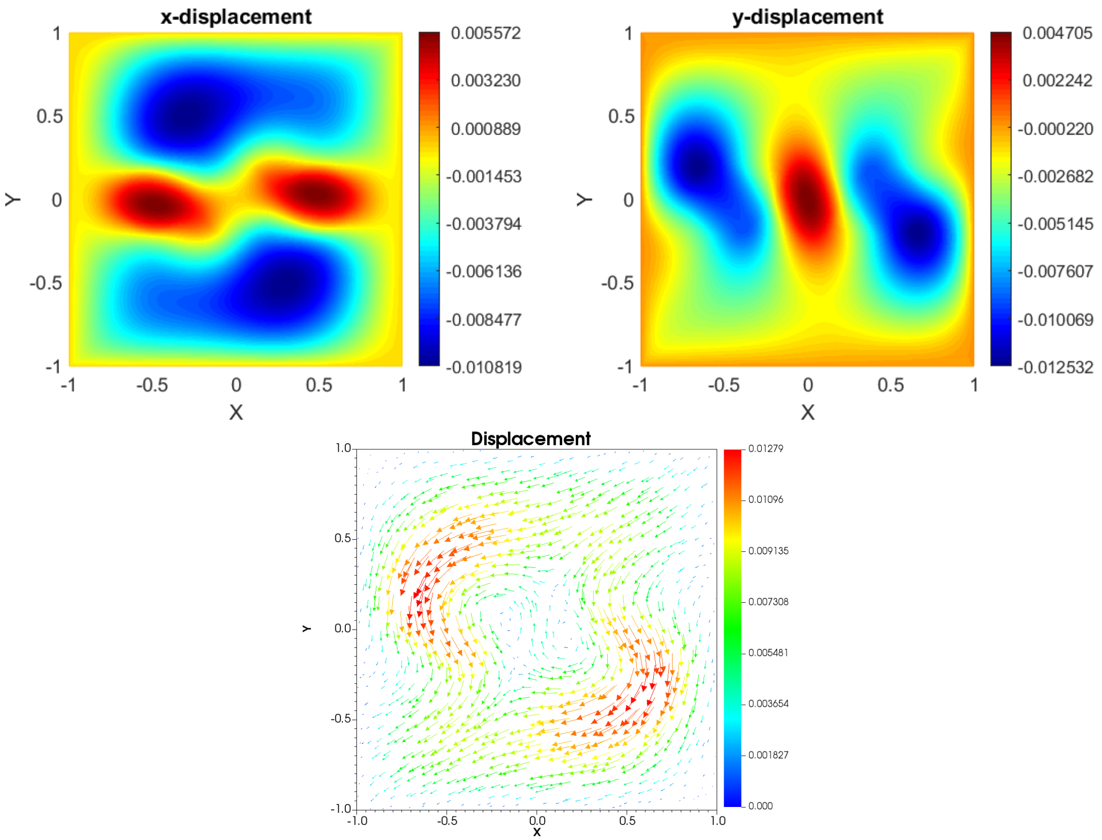


FIGURE 2.3: Components of the displacement vector $\overline{\boldsymbol{u}}$ (top), and a vector view of $\overline{\boldsymbol{u}}$ (bottom) for Experiment 2.

Homogeneous Dirichlet boundary conditions were applied to the entire boundary. For this numerical experiment, we set the Poisson ratio $\nu$ to be a constant 0.4999 over the whole domain, which corresponds to $\lambda = 4999$. The figures are from computations using a uniform $256 \times 256$ mesh on which the displacement and pressure variables are represented by a total of 592387 degrees of freedom. Again, the purpose of this experiment was to show the effects of the body force on a uniform tissue stiffness.

**Experiment 3**. The tissue stiffness parameter $\mu$ for this experiment is given by

$$\mu(x,y) = \begin{cases} 8, & \text{if } x^2 + y^2 \leq 0.25 \\ 1, & \text{else} \end{cases}$$

an image of which can be seen in Figure 2.4. As mentioned previously, cancerous tissues are often stiffer than surrounding healthy tissues, and the values of $\mu$ in cancerous areas can be five to ten times that of those in healthy tissues. On the disk of radius 0.5 at the origin, we take the value of $\mu$ to be 8 in order to resemble this situation. For this experiment, we also have $\nu = 0.4999$ and homogeneous Dirichlet boundary conditions were applied on the entire boundary.

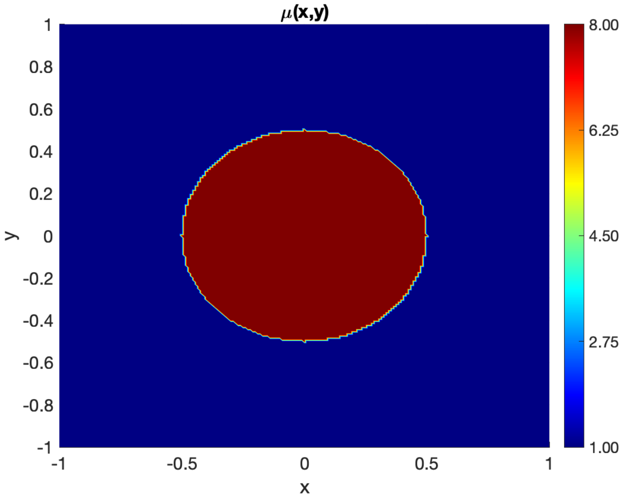


FIGURE 2.4: Tissue stiffness parameter $\mu$ for Experiment 3.

In Figure 2.5, we show the components of the displacement vector $\overline{\boldsymbol{u}}$ in the top left and right figures. Here, we can clearly see the effects of the tissue stiffness parameter $\mu$ in the components of the displacement $\overline{\boldsymbol{u}}$. The simulations were started with 256 elements in the domain (mesh size $\sqrt{2}/16$) and we refine the mesh adaptively 7 times using the heuristic error estimator (or indicator) described in the first chapter.

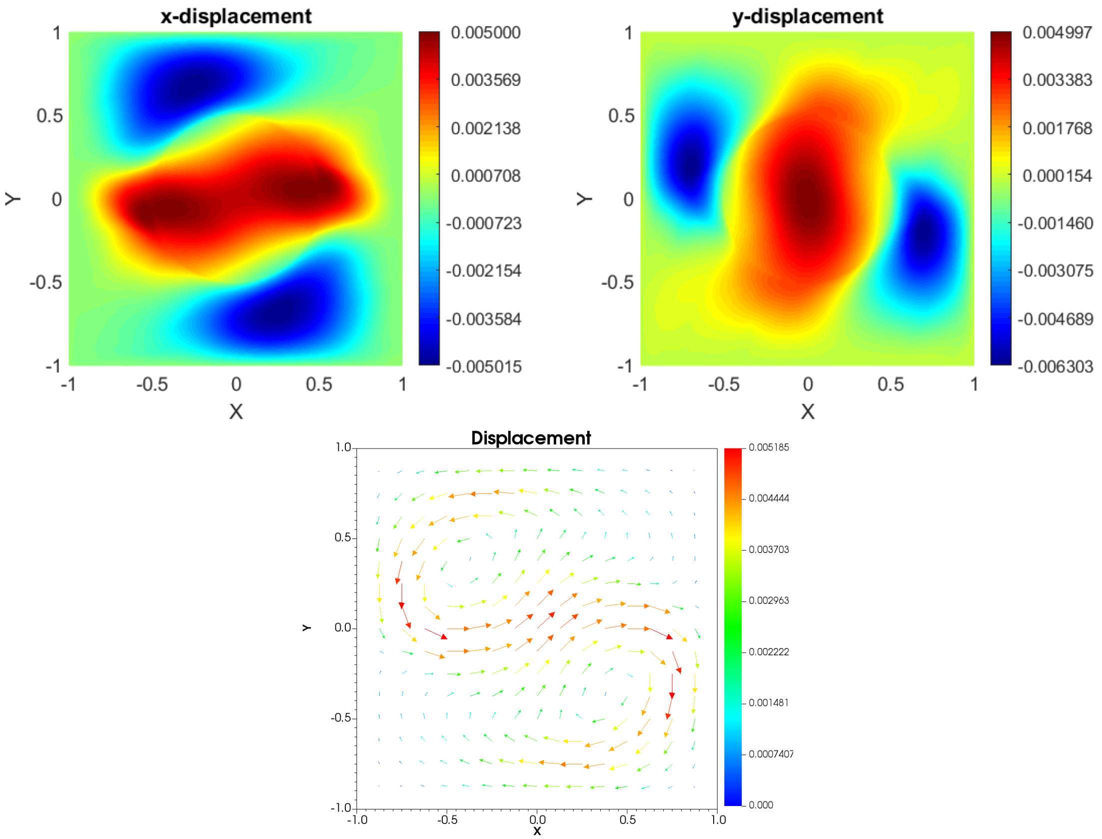


FIGURE 2.5: Components of the displacement vector $\overline{\boldsymbol{u}}$ (top) and a vector view of $\overline{\boldsymbol{u}}$ (bottom) for Experiment 3.

| cycle | # cells | # dofs ($\overline{\boldsymbol{u}}$ and $p$ ) |
|---|---|---|
| 0 | 256 | 2467 |
| 1 | 640 | 6231 |
| 2 | 1600 | 15395 |
| 3 | 3856 | 37323 |
| 4 | 9016 | 86215 |
| 5 | 20872 | 196203 |
| 6 | 47656 | 449483 |
| 7 | 110272 | 1042963 |

TABLE 2.3: Experiment 3: Number of degrees of freedom in each refinement cycle are shown in the last column. The second column shows the number of cells (or square elements) in the mesh.

The number of cells and total degrees of freedom needed to represent both $\overline{\boldsymbol{u}}$ and $p$ are shown in Table 2.3. At each refinement cycle, 30% of the cells are marked for refinement and 3% of the cells are marked for coarsening. We note that the heuristic error estimator used in the computations works well for the problem. The mesh is more refined in the areas where the displacement is changing rapidly (near the boundary of the disk where the tissue stiffness parameter abruptly changes) which this is exactly what we would like to have. Computational meshes for the last 4 refinement cycles are shown in Figure 2.6.

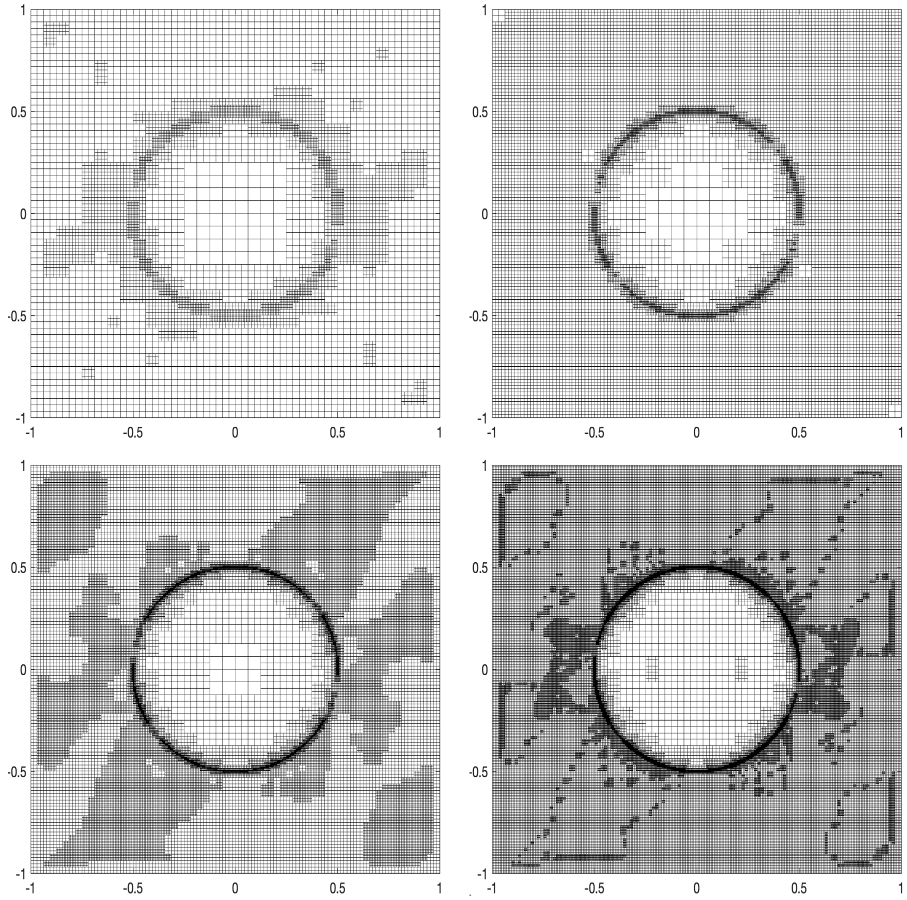


FIGURE 2.6: Adaptively refined meshes for Experiment 3.

**Experiment 4**. For this experiment, we changed the tissue stiffness $\mu$ in the following way and repeated the computations. We choose the tissue stiffness parameter as

$$\mu(x,y) = \begin{cases} 8, & \text{if } (x+0.25)^2 + (y-0.25)^2 \leq 0.4^2 \text{ or } (x-0.5)^2 + (y+0.5)^2 \leq 0.2^2 \\ 1, & \text{else.} \end{cases}$$

This choice of $\mu$ corresponds to two circular regions of stiffer tissues (to resemble a situation where multiple tumors of different sizes are present) in the domain (see Figure 2.7).

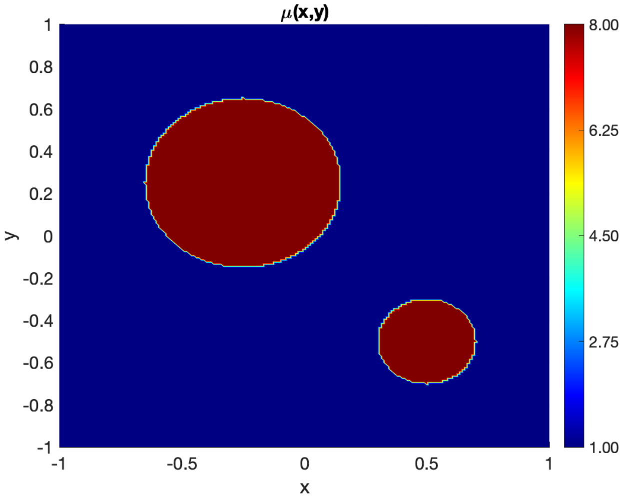


FIGURE 2.7: Tissue stiffness parameter $\mu$ for Experiment 4.

At the start of the computation, we have a coarse, uniform mesh that consists of 1024 cells (a $32 \times 32$ mesh). The mesh is then refined adaptively 5 times. Table 2.4 shows the number of cells in the mesh and the corresponding degrees of freedom needed to represent $\overline{\boldsymbol{u}}$ and $p$ for each refinement cycle.

Components of the displacement $\overline{\boldsymbol{u}}$ and the adaptively refined mesh for one of the last refinement cycles are shown in Figure 2.8. Again, we can see the effects of the tissue stiffness parameter $\mu$ in the displacement components, and the local refinements are present in the parts of the domain where we expect them to be.

| cycle | # cells | # dofs ($\overline{\boldsymbol{u}}$ and $p$ ) |
|---|---|---|
| 0 | 1024 | 9539 |
| 1 | 2668 | 25361 |
| 2 | 6880 | 66343 |
| 3 | 14488 | 132593 |
| 4 | 31504 | 292721 |
| 5 | 62476 | 571921 |

TABLE 2.4: Experiment 4: Number of degrees of freedom (dofs) in each refinement level are shown in the last column. The second column shows the number of cells in the mesh.

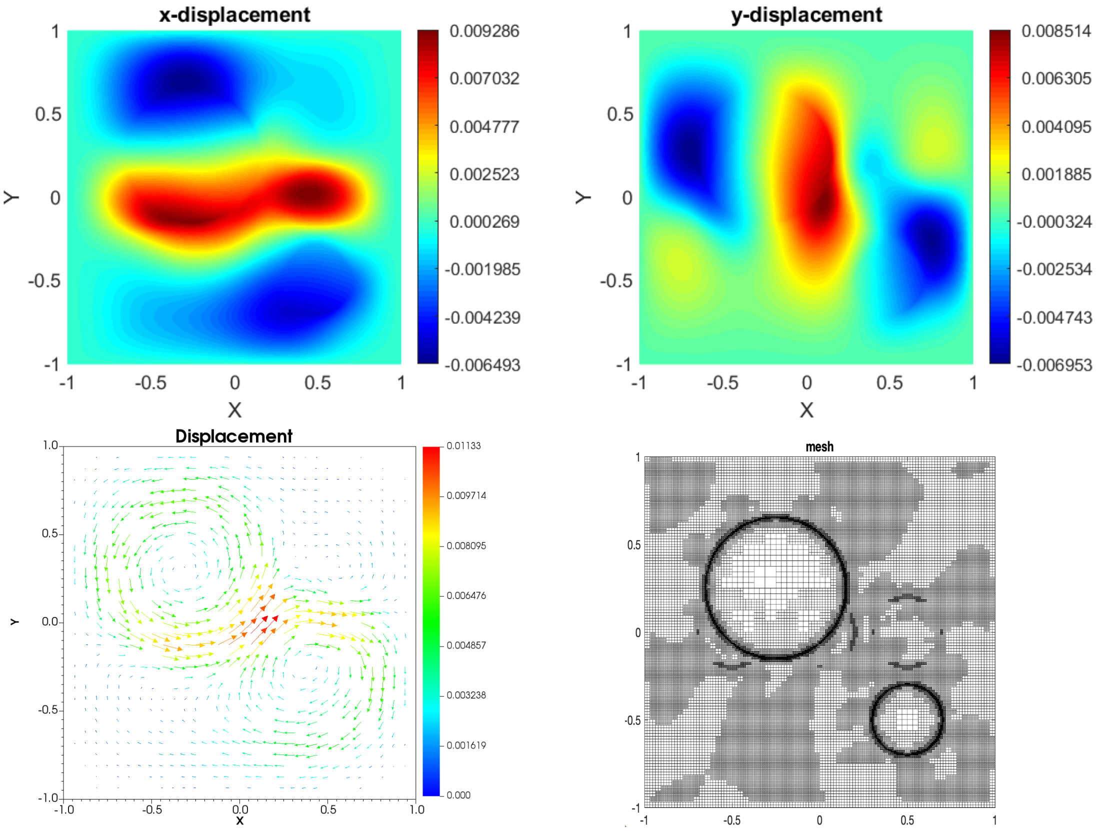


FIGURE 2.8: Components of the displacement vector $\overline{\boldsymbol{u}}$ (top), a vector view of $\overline{\boldsymbol{u}}$ (bottom left), and the mesh (bottom right) for Experiment 4.

**Experiment 5**. This experiment is almost the same as Experiment 3. The only change that is made is in the boundary conditions. In this case, homogeneous Dirichlet boundary conditions are applied on the left, top, and right boundaries, and a traction boundary condition $\sigma \cdot \boldsymbol{n} = [0, -0.0001]^T$ is applied to the bottom boundary. The traction boundary condition specifies the applied pressure on the boundary in the normal direction. Figure 2.9 shows results of a simulation with $\nu = 0.4999$. We observe the movement of the points along the bottom boundary caused by the force applied to it in the figures. The adaptively refined mesh for this example is also shown in the bottom right.

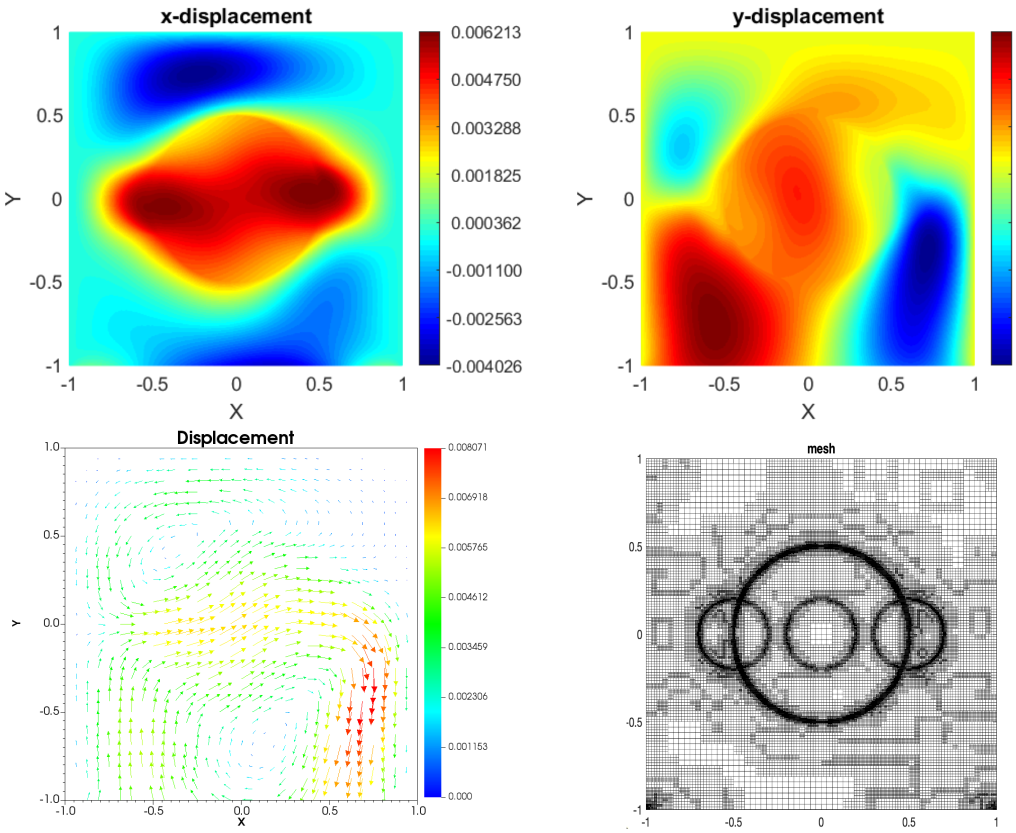


Figure 2.9: Components of the displacement vector $\overline{\boldsymbol{u}}$ (top), a vector view of $\overline{\boldsymbol{u}}$ (bottom left), and the mesh (bottom right) for Experiment 5.

**Experiment 6**. We then modify Experiment 5 slightly by applying the traction boundary conditions $\sigma \cdot \boldsymbol{n} = [0, -0.0001]^T$ on both left and right boundaries. Homogeneous Dirichlet boundary conditions are imposed on the top and bottom boundaries (meaning they are pinned). Figure 2.10 shows results of a simulation with $\nu = 0.4999$. Effects of the traction on the left and the right boundaries are now clearly seen in the figures showing the displacement components (top row), as well as the vector view of the displacement (bottom left). The adaptively refined mesh is displayed in the figure on the bottom right.

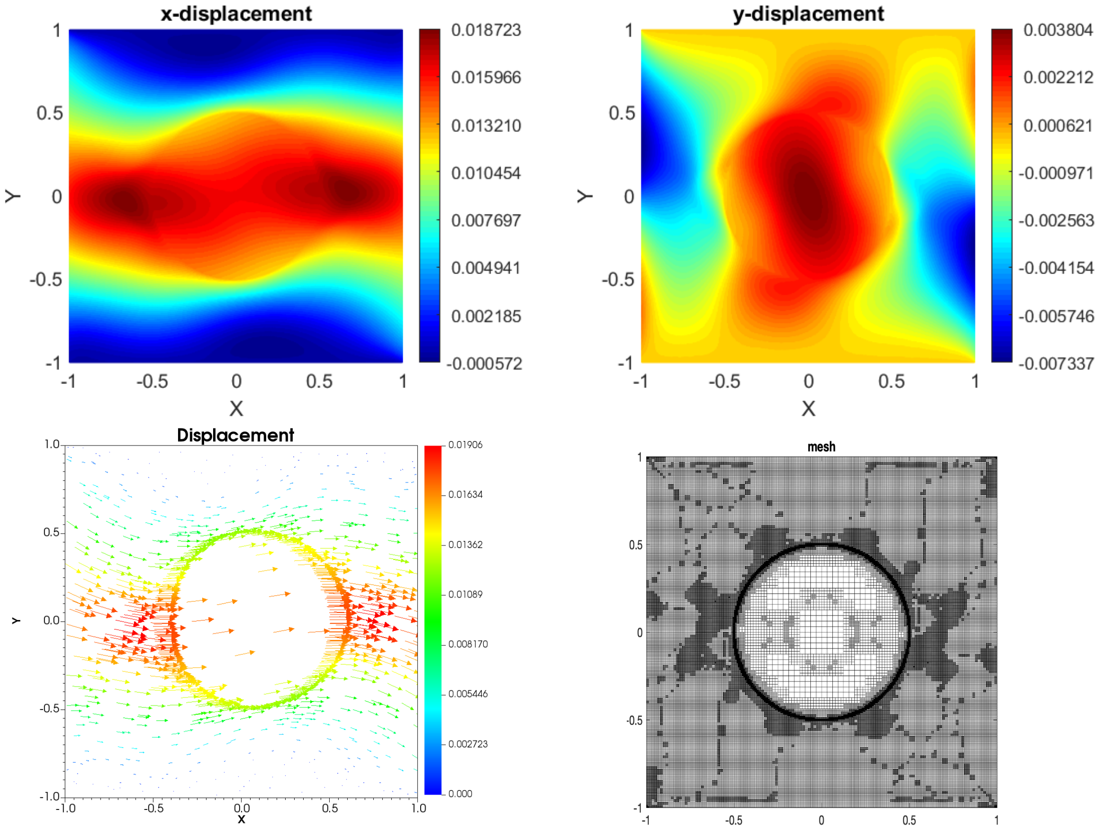


FIGURE 2.10: Components of the displacement vector $\overline{\boldsymbol{u}}$ (top), a vector view of $\overline{\boldsymbol{u}}$ (bottom left), and the mesh (bottom right) for Experiment 6.

An additional experiment is explored as an example of how traction boundary conditions such as those seen in Experiments 5 and 6 could affect the computational mesh. Figure 2.11 shows the results of what the displacement in the computational mesh would look like when a traction boundary condition similar to that of Experiment 5 is applied to the bottom boundary in another numerical experiment. For this particular example, we have a setup with no body force present and there is a circular stiff inclusion at the center of the domain. The top boundary is pinned and there is no traction (also called zero traction) on the left and right boundaries, meaning those sides are free to move. Traction at the bottom boundary is given by $\sigma \cdot \boldsymbol{n} = [0, 0.5]^T$. The figure on the left shows the computational mesh that is refined adaptively. The figure on the right shows the computational mesh displaced by the vector field $\overline{\boldsymbol{u}}$ (the displacement in the elasticity system).

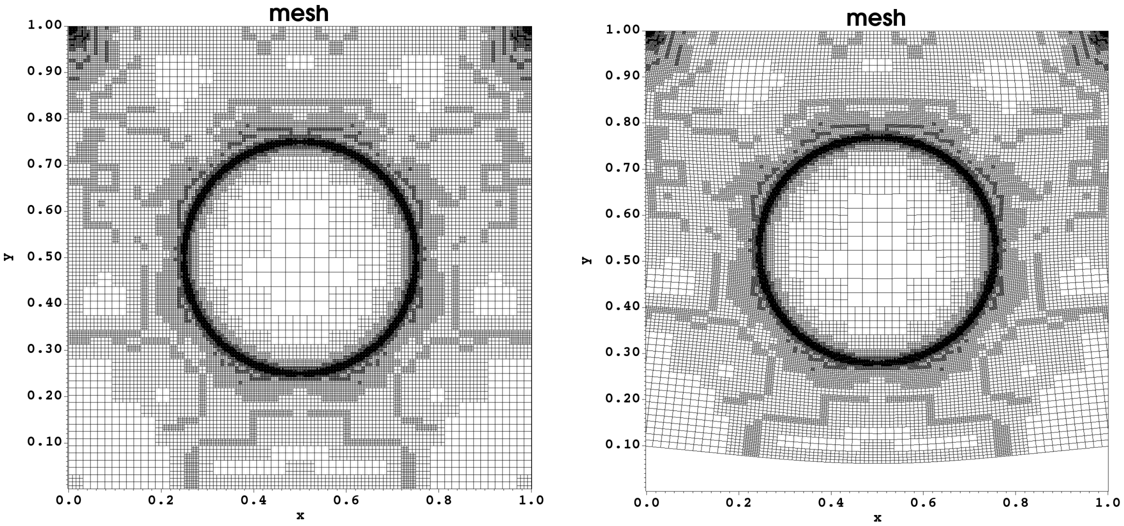


FIGURE 2.11: Mesh representation of the displacement.

# Chapter 3

# Elastography Inverse Problem

In this chapter, we discuss the optimization approach for the parameter identification problem. We include descriptions of two different objective functionals, a characterization of the derivative, and discrete formulas for implementation for a choice of objective functional. We also discuss stochastic variations of the gradient descent method for the problem and include the results of numerical experiments with synthetic data.

## 3.1 Objective Functionals

In parameter identification problems, minimization of the so-called output least-squares (OLS) functional is the most commonly used approach for solving parameter identification problems. The optimization problem is to find $\mu = \mu(\boldsymbol{x}) \in \widetilde{L}$ such that

$$J_1(\mu) = \frac{1}{2}\|\overline{\boldsymbol{u}}(\mu) - \overline{z}\|^2_{\overline{V}} + \frac{1}{2}\|p(\mu) - \hat{z}\|^2_Q \tag{3.1}$$

is minimized. Here, another possible formulation is to find $\mu \in \widetilde{L}$ such that the modified output least-squares (MOLS) objective functional

$$J(\mu) = \frac{1}{2}T(\mu, \boldsymbol{u}(\mu) - \boldsymbol{z}, \boldsymbol{u}(\mu) - \boldsymbol{z}), \tag{3.2}$$

where the trilinear form $T$ is defined as in (2.6), is minimized. In (3.1) and (3.2), $\boldsymbol{z} = (\overline{z}, \hat{z})$ stands for a measurement of $(\overline{\boldsymbol{u}}, p)$, $\boldsymbol{u}(\mu) - z = (\overline{\boldsymbol{u}}(\mu) - \overline{z}, p(\mu) - \hat{z})$, where $\boldsymbol{u}(\mu)$ is the solution to the direct problem (2.4)-(2.5) for a specific $\mu$, and $\widetilde{L}$ is the set of feasible parameters. Due to the ill-posedness of the problem, a regularized version

$$J_\beta(\mu) = J(\mu) + \frac{1}{2}\beta R(\mu) = \frac{1}{2}T(\mu, \boldsymbol{u}(\mu) - \boldsymbol{z}, \boldsymbol{u}(\mu) - \boldsymbol{z}) + \frac{1}{2}\beta R(\mu),$$

where $R(\mu)$ is a regularization term and $\beta$ is the regularization parameter, is used in practice. For the numerical experiments in this chapter, we will use $H^1$-norm regularization, i.e $R(\mu) = \|\mu\|_{H^1}$. Using (2.6) to rewrite $J(\mu)$, we get

$$\begin{aligned} J(\mu) &= \frac{1}{2}[a(\mu, \overline{\boldsymbol{u}}(\mu) - \overline{z}, \overline{\boldsymbol{u}}(\mu) - \overline{z}) + b(\overline{\boldsymbol{u}}(\mu) - \overline{z}, p(\mu) - \hat{z}) \\ &\qquad - b(\overline{\boldsymbol{u}}(\mu) - \overline{z}, p(\mu) - \hat{z}) + c(p(\mu) - \hat{z}, p(\mu) - \hat{z})] \\ &= \frac{1}{2}a(\mu, \overline{\boldsymbol{u}}(\mu) - \overline{z}, \overline{\boldsymbol{u}}(\mu) - \overline{z}) + \frac{1}{2}c(p(\mu) - \hat{z}, p(\mu) - \hat{z}). \end{aligned}$$

Thus, the MOLS objective functional is given explicitly as

$$J(\mu) = \frac{1}{2}\int_D 2\mu\varepsilon_{(\overline{\boldsymbol{u}}(\mu)-\overline{z})} \cdot \varepsilon_{(\overline{\boldsymbol{u}}(\mu)-\overline{z})} + \frac{1}{2}\int_D \frac{1}{\lambda}(p(\mu) - \hat{z})^2. \tag{3.3}$$

The regularized discrete optimization problem is

$$\arg\min_{\mu_h \in \widetilde{L}_h} J(\mu_h) = J(\mu_h) + \frac{1}{2}\beta R(\mu_h)$$

$$J_\beta(\mu_h) = \frac{1}{2}\int_D 2\mu_h|\varepsilon_{(\overline{\boldsymbol{u}}_h-\overline{z})}|^2 + \frac{1}{2}\int_D \frac{1}{\lambda}|p(\mu_h) - \hat{z}|^2 + \frac{1}{2}\beta R(\mu_h),$$

where $(\overline{\boldsymbol{u}}_h(\mu_h), p(\mu_h))$ is the solution to (2.8) for the parameter $\mu_h$, $R(\mu_h)$ is a regularization term, and $\beta$ is a regularization parameter.

## 3.2 Derivative Characterization

Next, we move on to a discussion of the gradient of the objective functional. To get a derivative characterization we must differentiate (2.7) with respect to $\mu$. Using the product rule gives

$$\begin{aligned} a(\delta\mu, \overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}) + a(\mu, \delta\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}) + b(\overline{\boldsymbol{v}}, \delta p) &= 0 \\ b(\delta\overline{\boldsymbol{u}}, q) - c(\delta p, q) &= 0 \end{aligned}$$

Note that $\delta\boldsymbol{u} = (\delta\overline{\boldsymbol{u}}, p) = (D\overline{\boldsymbol{u}}(\mu)\delta\mu, Dp(\mu)\delta\mu)$, where $Dp(\mu)\delta\mu$ is the derivative of $p$ with respect to $\mu$ in the direction $\delta\mu$ and the notation $\delta\overline{\boldsymbol{u}}$ is the shorthand way of writing $D\overline{\boldsymbol{u}}(\mu)\delta\mu$.

Rearranging the first equation above gives the characterization for the parameter-to-solution map,

$$\begin{aligned} a(\mu, \delta\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}) + b(\overline{\boldsymbol{v}}, \delta p) &= -a(\delta\mu, \overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}) \\ b(\delta\overline{\boldsymbol{u}}, q) - c(\delta p, q) &= 0. \end{aligned}$$

Now we need the derivative of (3.3), $J(\mu)$, with respect to $\mu$ in the continuous setting.

$$\begin{aligned} DJ(\mu)\delta\mu =& \frac{1}{2}a(\delta\mu, \overline{\boldsymbol{u}}(\mu) - \overline{z}, \overline{\boldsymbol{u}}(\mu) - \overline{z}) + \frac{1}{2}a(\mu, \delta\overline{\boldsymbol{u}}, \overline{\boldsymbol{u}}(\mu) - \overline{z}) + \frac{1}{2}a(\mu, \overline{\boldsymbol{u}}(\mu) - \overline{z}, \delta\overline{\boldsymbol{u}}) \\ &+ \frac{1}{2}c(\delta p, p(\mu) - \hat{z}) + \frac{1}{2}c(p(\mu) - \hat{z}, \delta p) \\ =& \frac{1}{2}a(\delta\mu, \overline{\boldsymbol{u}}(\mu) - \overline{z}, \overline{\boldsymbol{u}}(\mu) - \overline{z}) + a(\mu, \delta\overline{\boldsymbol{u}}, \overline{\boldsymbol{u}}(\mu) - \overline{z}) + c(\delta p, p(\mu) - \hat{z}). \end{aligned}$$

Using the differentiation of (2.7) given above, $a(\mu, \delta\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}) = -a(\delta\mu, \overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}) - b(\overline{\boldsymbol{v}}, \delta p)$ and $c(\delta p, q) = b(\delta\overline{\boldsymbol{u}}, q)$. Keeping in mind that the test functions are now $\overline{\boldsymbol{u}}(\mu) - \overline{z}$ and $p(\mu) - \hat{z}$, $DJ(\mu)\delta\mu$ can be rewritten as

$$\begin{aligned} DJ(\mu)\delta\mu =& \frac{1}{2}a(\delta\mu, \overline{\boldsymbol{u}}(\mu) - \overline{z}, \overline{\boldsymbol{u}}(\mu) - \overline{z}) - a(\delta\mu, \overline{\boldsymbol{u}}(\mu), \overline{\boldsymbol{u}}(\mu) - \overline{z}) \\ &- b(\overline{\boldsymbol{u}}(\mu) - \overline{z}, \delta p) + b(\delta\overline{\boldsymbol{u}}, p(\mu) - \hat{z}). \end{aligned}$$

Next, the $\frac{1}{2}$ can be brought inside since the function is linear. Then, since the function is linear in each argument those terms can be combined.

$$\begin{aligned} DJ(\mu)\delta\mu =& a(\delta\mu, \frac{1}{2}(\overline{\boldsymbol{u}}(\mu) - \overline{z}), \overline{\boldsymbol{u}}(\mu) - \overline{z}) - a(\delta\mu, \overline{\boldsymbol{u}}(\mu), \overline{\boldsymbol{u}}(\mu) - \overline{z}) \\ &- b(\overline{\boldsymbol{u}}(\mu) - \overline{z}, \delta p) + b(\delta\overline{\boldsymbol{u}}, p(\mu) - \hat{z}) \\ =& a(\delta\mu, -\frac{1}{2}\overline{\boldsymbol{u}}(\mu) - \frac{1}{2}\overline{z}, \overline{\boldsymbol{u}}(\mu) - \overline{z}) - b(\overline{\boldsymbol{u}}(\mu) - \overline{z}, \delta p) + b(\delta\overline{\boldsymbol{u}}, p(\mu) - \hat{z}). \end{aligned}$$

Pulling the $\frac{1}{2}$ out in front again gives the continuous version of the gradient.

$$DJ(\mu)\delta\mu = -\frac{1}{2}a(\delta\mu, \overline{\boldsymbol{u}}(\mu) + \overline{z}, \overline{\boldsymbol{u}}(\mu) - \overline{z}) - b(\overline{\boldsymbol{u}}(\mu) - \overline{z}, \delta p) + b(\delta\overline{\boldsymbol{u}}, p(\mu) - \hat{z}).$$

## 3.3 Discrete Formulas

The discrete version of the objective functional is

$$J(\boldsymbol{M}) = \frac{1}{2}\left[\overline{\boldsymbol{U}}(\boldsymbol{M}) - \overline{\boldsymbol{Z}}\right]^T A(\boldsymbol{M})\left[\overline{\boldsymbol{U}}(\boldsymbol{M}) - \overline{\boldsymbol{Z}}\right] + \frac{1}{2}\left[P(\boldsymbol{M}) - \widehat{Z}\right]^T C\left[P(\boldsymbol{M}) - \widehat{Z}\right]. \quad (3.4)$$

The discrete version of the gradient is

$$\begin{aligned} DJ(\boldsymbol{M})\delta M = &-\frac{1}{2}\left[\overline{\boldsymbol{U}}(\boldsymbol{M}) + \overline{\boldsymbol{Z}}\right]^T A(\delta M)\left[\overline{\boldsymbol{U}}(\boldsymbol{M}) - \overline{\boldsymbol{Z}}\right] \\ &+\frac{1}{2}\left[P(\boldsymbol{M}) - \widehat{Z}\right]^T B\nabla\overline{\boldsymbol{U}}(\boldsymbol{M})\delta M - \left[\overline{\boldsymbol{U}}(\boldsymbol{M}) - \overline{\boldsymbol{Z}}\right]^T B\nabla P(\boldsymbol{M})\delta M \end{aligned} \tag{3.5}$$

where $\nabla\overline{\boldsymbol{U}}(\boldsymbol{M})\delta M$ is just $\delta\overline{\boldsymbol{u}}$, $\nabla P(\boldsymbol{M})\delta M$ is just $\delta p$, $DJ(\boldsymbol{M})\delta M$ can be written as $\nabla J\delta M$, and the regularization term is ignored for now because of its simplicity.

We want to extract the direction, $\delta M$, from the equation for the gradient to get $\nabla J$ alone. And since the $\delta M$ is already extracted in the second two terms, only the first term needs to be manipulated. This manipulation is done using the the so-called adjoint stiffness matrix $S = S(\overline{\boldsymbol{U}})$ defined by

$$A(\boldsymbol{M})\overline{\boldsymbol{U}} = S(\overline{\boldsymbol{U}})\boldsymbol{M} \ \ \forall\boldsymbol{M} \in \mathbb{R}^m \text{ and } \forall\, \overline{\boldsymbol{U}} \in \mathbb{R}^n.$$

Note that $A(\boldsymbol{M})$ is $n\times n$, $\overline{\boldsymbol{U}}$ is $n\times 1$, and $\boldsymbol{M}$ is $m\times 1$, meaning $S(\overline{\boldsymbol{U}})$ is $n\times m$. When manipulating the adjoint stiffness matrix, we only need to work with the upper left block, $A$, not the entire stiffness matrix $K$. The $A(\delta M)\left[\overline{\boldsymbol{U}}(\boldsymbol{M}) - \overline{\boldsymbol{Z}}\right]$ part of the first term will be transformed using the adjoint stiffness matrix to get $A(\delta M)\left[\overline{\boldsymbol{U}}(\boldsymbol{M}) - \overline{\boldsymbol{Z}}\right] = S(\overline{\boldsymbol{U}}(\boldsymbol{M}) - \overline{Z})\delta M$. Plugging this into the formula for $\nabla J\delta M$ will yield the gradient formula since the $\delta M$'s will cancel out on both sides. Thus,

$$\begin{aligned} \nabla J(\boldsymbol{M}) = &-\frac{1}{2}\left[\overline{\boldsymbol{U}}(\boldsymbol{M}) + \overline{\boldsymbol{Z}}\right]^T S(\overline{\boldsymbol{U}}(\boldsymbol{M}) - \overline{\boldsymbol{Z}}) \\ &+\frac{1}{2}\left[P(\boldsymbol{M}) - \widehat{Z}\right]^T B\nabla\overline{\boldsymbol{U}}(\boldsymbol{M}) - \left[\overline{\boldsymbol{U}}(\boldsymbol{M}) - \overline{\boldsymbol{Z}}\right]^T B\nabla P(\boldsymbol{M}). \end{aligned} \tag{3.6}$$

## 3.4 Stochastic Gradient Descent Type Methods

We consider several variations of the (stochastic) gradient descent method (see, for example, [27]) for the solution of the optimization problem in this section. Our goal is to investigate the applicability of these methods to our problem. The first variation of these types of methods is given in Algorithm 1.

**Algorithm 1** A variation of the gradient descent method.

```
Choose an initial guess $M^{(0)}$ and a positive step size $\alpha$.
while the stopping criterion is not satisfied do
    $M^{(k+1)} = M^{(k)} - \alpha \nabla J(M^{(k)})$
end while
```

The difference between this algorithm and the commonly used gradient descent (or steepest descent) is the choice of the step size $\alpha$. Normally, a step size $\alpha_k$ is chosen (by using some kind of line search algorithm) so that the maximum possible decrease is achieved in the functional that is being minimized. However, in this algorithm, the step size is kept constant. When dealing with large-scale problems such as ours, where the dimension of the optimization problem is in the tens or hundreds of thousands, savings in the computational costs associated with the line search step can be quite significant.

Other variations of the stochastic gradient descent method that help reduce the computational cost are the stochastic (batch) gradient descent method. The idea here is to use a partial gradient $\nabla J_p(\boldsymbol{M}^{(k)})$ instead of the (full) gradient $\nabla J(\boldsymbol{M}^{(k)})$ when making an update of the iterate $\boldsymbol{M}^{(k+1)}$. At every iteration, we can compute one randomly chosen component of the gradient or a set of components of the gradient and use that to update the iterates. In this case, there are existing examples (linear and nonlinear optimization problems) where the method is applied successfully at least for small- to medium-scale problems. The step-sizes $\alpha_k$ would be decreasing in every iteration. There are other versions of these algorithms as well. For example, in a semi-stochastic version of the gradient descent method, a certain (randomly chosen) number of components are computed at every iteration and the full gradient is computed once in every fixed number of iterations to speed up the convergence. We plan to investigate applications of this and other variations of stochastic gradient descent to large-scale problems such as ours in the future.

## 3.5 Numerical Results

In this section, we present results of some numerical experiments with both continuous and discontinuous stiffness parameters. Figure 3.1 shows examples of such stiffness parameters. The image on the left shows a situation where the stiffness parameter $\mu$ has a smooth transition between tissue stiffness values; the two peaks represent stiffer tissue relative to the areas surrounding them.

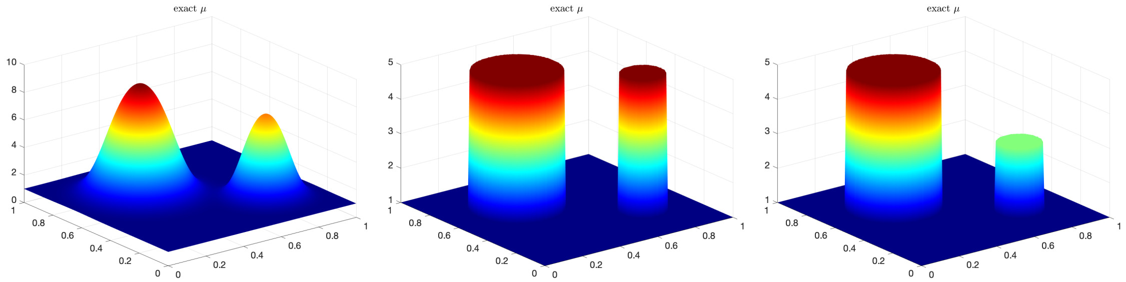


FIGURE 3.1: Smooth and discontinuous tissue stiffness parameters.

The other two images correspond to the situation where there are two stiffer inclusions within the domain and where the interfaces between the tissue regions with different values (5 for stiffer tissues and 1 for soft tissues for the image in the center, and 1 for soft tissues and 5 and 2 for stiffer inclusions in the right image) are sharp. Clinically, a discontinuous tissue stiffness parameter represents a tumor that has defined edges and a continuous stiffness parameter represents a tumor that is still growing into the surrounding healthy tissue.

### 3.5.1 Continuous transition on a uniform mesh

We create an analytical example to test the application of a stochastic gradient descent method for recovering a tissue stiffness parameter where the values of the parameter transition smoothly between areas of stiffer tissue and softer tissue. The stiffness parameter given in this subsection represents a tissue region with two round, stiffer inclusions (shown on the left in Figure 3.1).

We choose the parameter as

$$\mu(x,y) = 15 - 8\tanh\left(\frac{(x-0.35)^2+(y-0.65)^2}{0.04}\right) - 6\tanh\left(\frac{(x-0.75)^2+(y-0.3)^2}{0.02}\right)$$

and choose a displacement field by

$$\begin{aligned} u_1(x,y) &= \sin(2\pi y)(-1+\cos(2\pi x)) + \frac{1}{1+\lambda}\sin(\pi x)\sin(\pi y) \\ u_2(x,y) &= \sin(2\pi x)(1-\cos(2\pi y)) + \frac{1}{1+\lambda}\sin(\pi x)\sin(\pi y). \end{aligned}$$

The force field $\boldsymbol{f}$ is then computed from the constitutive equations using the $\mu$ and $\overline{\boldsymbol{u}}$ provided above. The pressure $p$ is computed using equation (1.7b) where $\lambda = \dfrac{2\nu}{1-2\nu}$ with a Poisson ratio of $\nu = 0.49995$. Homogeneous Dirichlet boundary conditions are used on the whole boundary of the computational domain $[0,1]^2$. We have step size $\alpha = 0.01$ as in Algorithm 1, and regularization parameter $\beta = 10^{-5}$ for the modified output least-squares objective functional.

Figure 3.2 shows the exact and recovered tissue stiffness parameters. The top left image in Figure 3.2 corresponds to the exact parameter representation on a uniform $32 \times 32$ mesh and the remaining figures show the estimated parameter at various stages of the iterative process on the same uniform mesh. The mesh has 1024 cells and 1089 nodes and, with the use of linear Lagrange elements for $\mu$, the dimension of the constrained optimization problem is 1089. We observe that the quality of the approximation is excellent for the gently varying (smooth) parameter (shown in Figure 3.1 on the left) on this relatively coarse mesh. The regions of stiffer tissue are clearly distinguishable after just 500 iterations (top right) and this could be sufficient for diagnostic purposes. For an accurate recovery/estimation, however, we provided a maximum number of 20000 iterations along with a tolerance of $10^{-8}$ for the $L^2$-norm of the gradient of the objective functional. The bottom right image corresponds to the estimated parameter after 10000 iterations and it is nearly indistinguishable from the exact parameter shown in the top left.

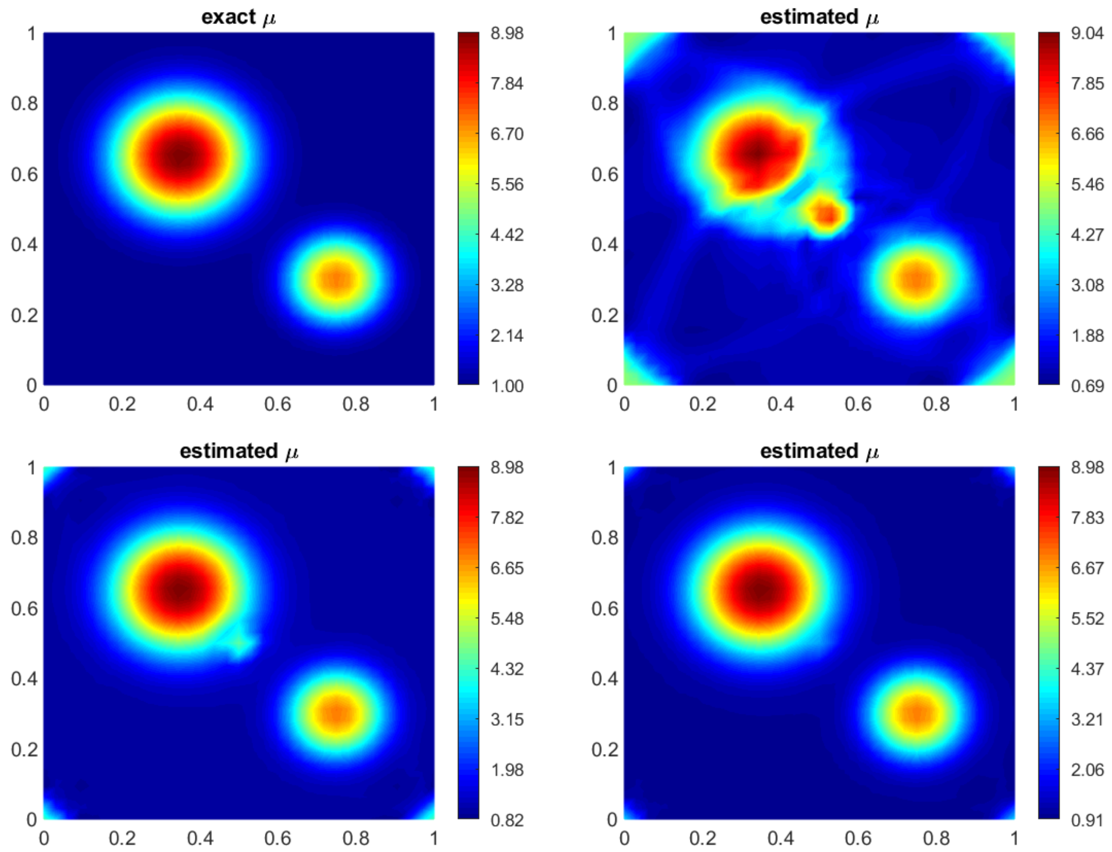


FIGURE 3.2: Exact stiffness parameter (top left), estimated parameter after 500 iterations (top right), estimated parameter after 5000 iterations (bottom left), and estimated parameter after 10000 iterations (bottom right).

### 3.5.2 Continuous transition on adaptive mesh

Using the same continuous inclusion as in subsection 3.5.1, the results of computation on adaptively refined meshes are explored. The mesh is refined using the heuristic error indicator for the tissue stiffness parameter (see Section 1.5). We start the iteration with a coarse, uniform mesh and specify the maximum number of refinement cycles. At every refinement cycle, approximately 30% of the cells were marked for refinement and 3% of the cells were marked for coarsening. Frequency of the refinement is a run-time parameter that is provided. Figure 3.3 shows the resulting adaptive meshes overlaid on the tissue stiffness maps for some intermediate iterations.

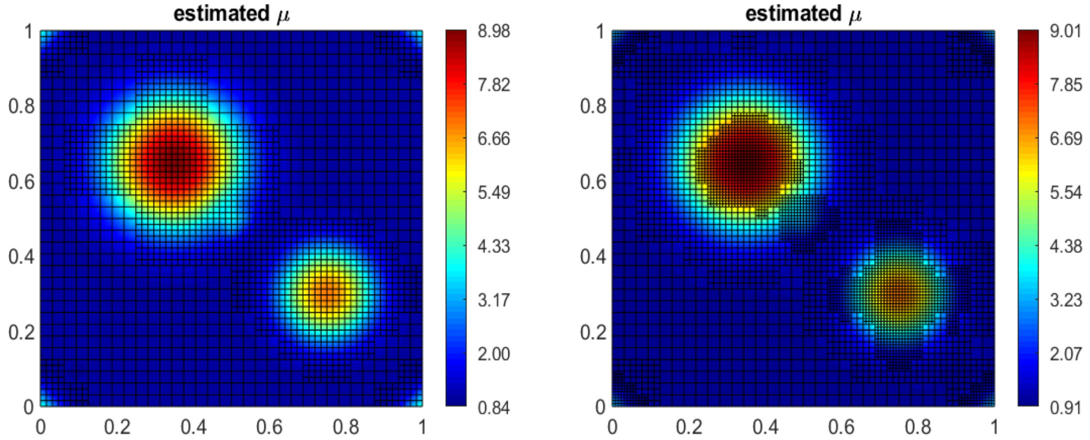


FIGURE 3.3: Adaptively refined meshes in two consecutive refinement cycles.

For a smooth stiffness parameter, we observe that the estimated stiffness parameter is qualitatively a very close match to the exact stiffness parameter (see the upper left panel of Figure 3.2). In Figure 3.3, the mesh is more refined around the boundaries of the stiffer regions as expected. The minimum cell diameters $h_{\text{min}}$ are 0.0442 and 0.0221 respectively. The key takeaway here is that we can obtain a very good estimate of the parameter on a relatively coarse mesh if changes in the parameter values are gradual. Obtaining the same level of resolution for a uniform mesh would be much more computationally expensive (we would have more degrees of freedom to represent the parameter as the mesh would need to be sufficiently fine on the entire domain).

### 3.5.3 Discontinuous transition on a uniform mesh

For this numerical experiment, we test the algorithm for an example with a discontinuous tissue stiffness parameter. The goal is to examine the performance and accuracy of the method for parameters that correspond to cases where the stiffer regions are more localized, i.e. transitions between soft and hard tissues are sharp. For the first experiment, we choose a tissue stiffness parameter $\mu$ given as

$$\mu(x,y) = \begin{cases} 5, & (x-0.35)^2 + (y-0.65)^2 \leq 0.2^2 \text{ or } (x-0.75)^2 + (y-0.3)^2 \leq 0.1^2 \\ 1, & \text{else.} \end{cases}$$

A surface view of the function is shown in the middle of Figure 3.1, and there are two circular stiff inclusions within the domain.

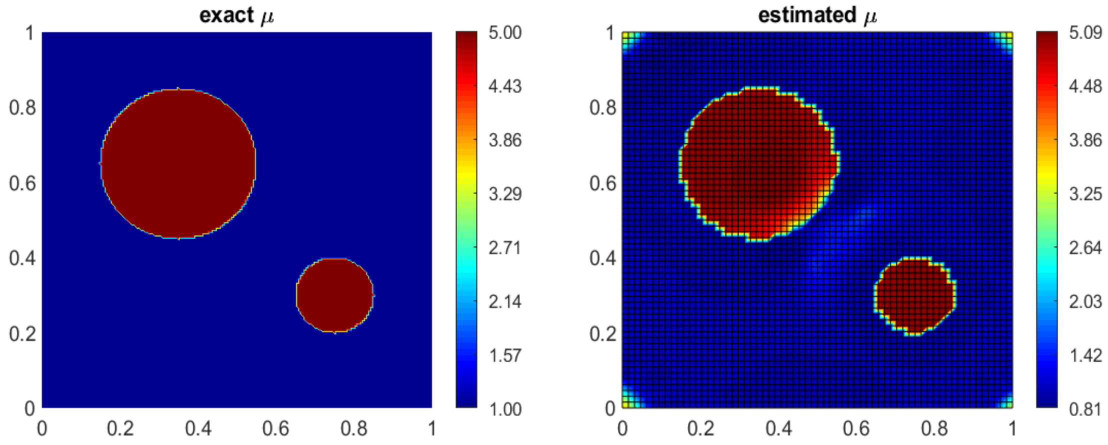


Figure 3.4: Exact parameter (left) and the estimated parameter after 20000 iterations.

Figure 3.4 shows the exact parameter and the estimated parameter side by side. The exact parameter is shown on a $200 \times 200$ mesh. For the estimated parameter, the mesh has 4096 cells (corresponding to a $64 \times 64$ mesh) and mesh size $h = \sqrt{2}/64$. For linear Lagrange elements, this corresponds to 4225 degrees of freedom for the representation of the tissue stiffness parameter. We notice that while the use of the coarse mesh leads to a good identification of the parameter, the discontinuous transition around the boundaries of the stiffer inclusion isn't captured well. We would have to use a very fine mesh to accurately capture this transition, which would lead to a significant increase in the dimension of the optimization problem. A dramatic increase in the overall computational time would be unavoidable in this situation. To mitigate this, our goal in the next subsection is to show the results of computations where the mesh is adaptively refined as the iterative process progresses, i.e. the computational mesh will be adapted to the estimated stiffness parameter. At each refinement/coarsening cycle, the mesh is locally refined in the regions where the parameter values change significantly and coarsened where changes in parameter values are relatively small.

### 3.5.4 Discontinuous transition on adaptive mesh

The computation is explored for the same discontinuous inclusion as in subsection 3.5.3, now on adaptively refined meshes. For adaptive refinements, the goal is to be able to capture the discontinuous transition accurately by using significantly fewer degrees of freedom compared to a computation with a very fine uniform mesh. This is achieved by the local refinement strategy described in Section 1.5. Figure 3.5 shows the estimation process, and computational meshes corresponding to each of these images are shown in Figure 3.6.

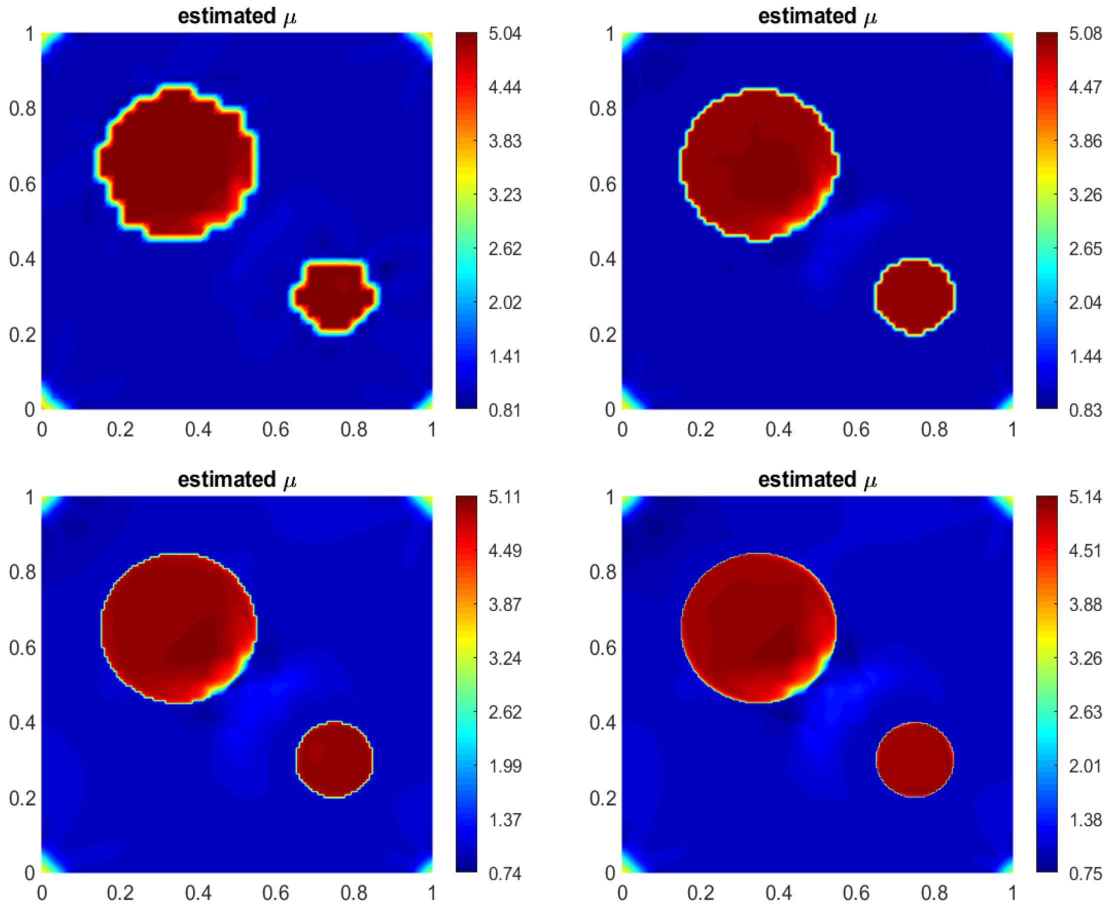


FIGURE 3.5: Approximations of a discontinuous stiffness parameter on adaptively refined meshes after 5000 iterations (top left), 10000 iterations (top right), 15000 iterations (bottom left), and 20000 iterations (bottom right).

The number of cells for each mesh in Figure 3.6 are 1957 (top left), 4627 (top right), 10903 (bottom left), and 25759 (bottom right) respectively. In the top right image in the figure corresponding to 10000 iterations, we have 4627 cells, 2141 degrees of freedom used in representing the parameter, and a minimum cell diameter corresponding to $h_{\min} = 0.0221$. A similar accuracy can be achieved by using a uniform mesh of the same cell diameter ($h = \sqrt{2}/64 \approx 0.0221$) and would have required $64 \times 64 = 4096$ cells and 4225 degrees of freedom. Thus, the dimension of the optimization problem is halved when using an adaptive, rather than a uniform, mesh.

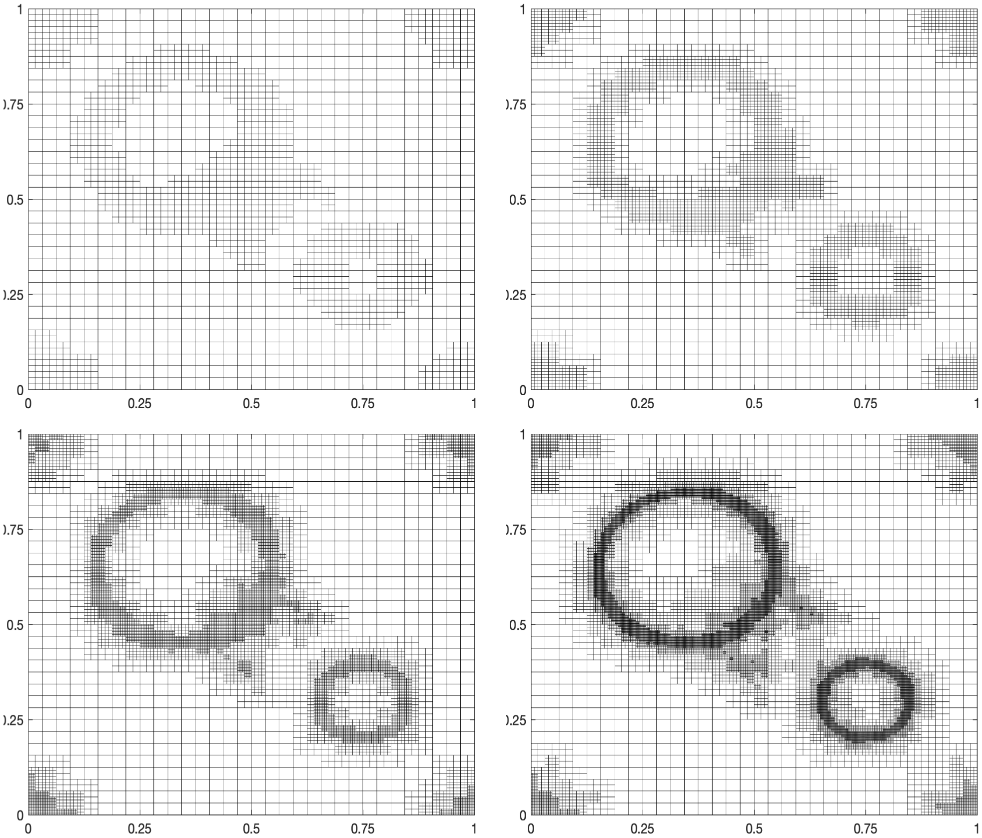


FIGURE 3.6: Successively refined meshes. Number of cells in each mesh: 1957 (top left), 4627 (top right), 10903 (bottom left), 25759 (bottom right).

Similarly, for the bottom left image corresponding to 15000 iterations, we have $h_{\min} \approx 0.0055$ and 11741 degrees of freedom. A uniform mesh with $h = \sqrt{2}/256 \approx 0.0055$ would have $256 \times 256 = 65536$ cells and would have required 66049 degrees of freedom to represent the parameter. So, at this point in the iterative process, using an adaptive mesh improves the degree of the optimization problem by a factor of about 6.

Our next numerical experiment is the recovery of a discontinuous parameter given by

$$\mu(x,y) = \begin{cases} 5, & (x-0.35)^2 + (y-0.65)^2 \leq 0.2^2 \\ 2, & (x-0.75)^2 + (y-0.3)^2 \leq 0.1^2 \\ 1, & \text{else.} \end{cases}$$

The exact parameter is shown in Figure 3.1 (right) and in Figure 3.7 (top left).

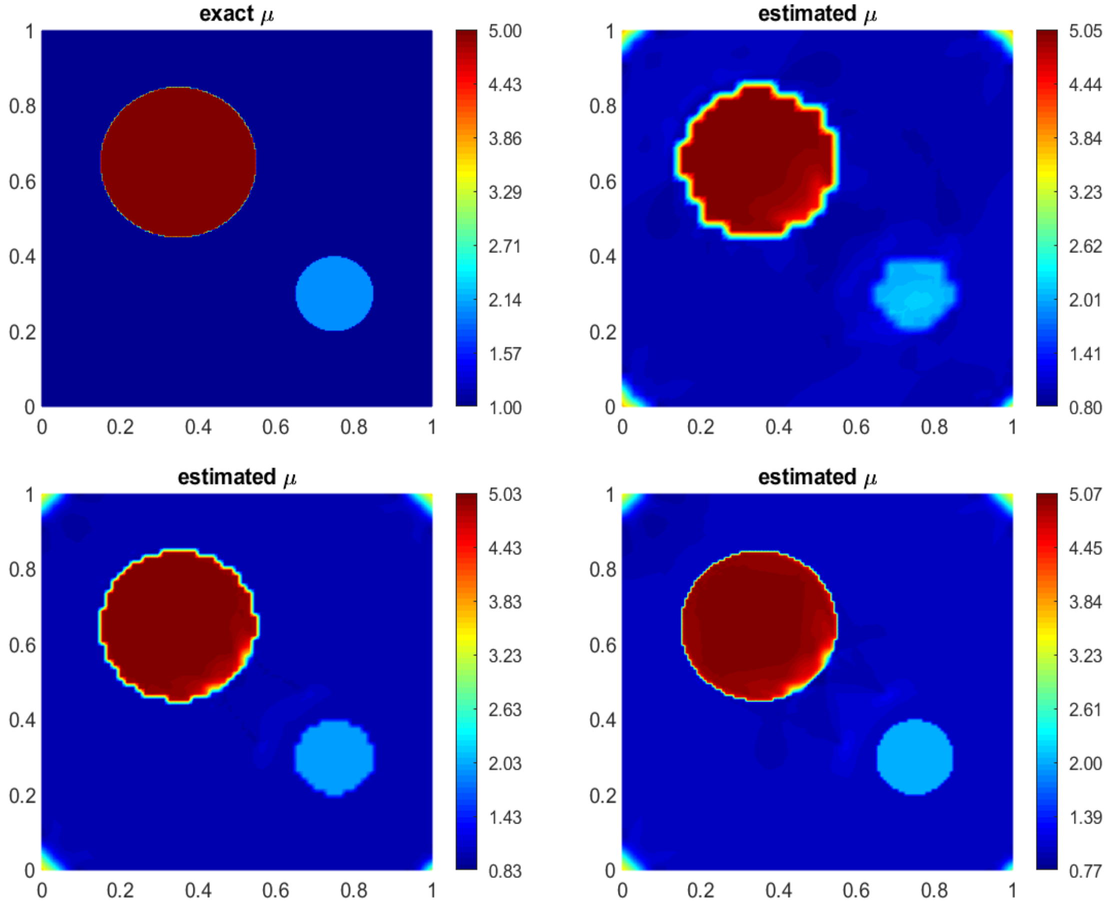


FIGURE 3.7: Exact stiffness parameter (top left) and approximations for another discontinuous inclusion on adaptively refined meshes after 5000 iterations (top right), 10000 iterations (bottom left), and 15000 iterations (bottom right).

The results of the simulation are shown in Figure 3.7 at various stages of the iterative process and it is evident that the combination of the stochastic gradient method and adaptive mesh refinement works very well for detecting inclusions stiffer than the surrounding regions, including those with varying degrees of stiffness. As can be seen from Figures 3.5 and 3.7, we are able to capture the solution behaviour in the regions of transition much more accurately with an adaptive mesh than with a uniform mesh. We also note in this last set of experiments that the tissue stiffness parameter has a value close to that of the soft tissue in the region where the smaller inclusion is located and our method is able to distinguish between these regions that have similar values quite well.

# Chapter 4

# Displacement Data, Error Analysis

In this chapter, we will discuss the use of measurement data and the dependence of the estimation error on the computational parameters. In particular, we examine the efficacy of our methods on more realistic data that is obtained in a lab where the experiments were conducted using tissue-mimicking phantoms.

## 4.1 Displacement Measurement Data and Numerical Results

As mentioned previously, we would like to see how our solution methods handle more realistic data. The displacement data used in these experiments comes from the Image Processing and Characterization of Tissues (IMPACT) lab, https://users.encs.concordia.ca/ impact/database/. An extensive description of the process of obtaining the ultrasound images and displacement data is found in Tehrani and Rivaz [28] and Rivaz et al [29]. The phantoms used for data collection were 40 mm × 32 mm tissue-mimicking breast phantoms which contained hard inclusions with Young's Modulus values at least twice that of the surrounding tissue. In particular, the values of Young's Modulus representing the surrounding healthy tissue were about 20 kPa, the hard inclusions ranged from 45-60 kPa depending on the experiment, and the Poisson ratio used by the authors was 0.49.

The displacement data from the database is separated into axial and lateral components, with both data sets having size $10 \times 256 \times 2048$, where the dimensions are $256 \times 2048$ and the other 10 layers of the data are increasing compression levels with average strain ranging from 0% – 4.5%. Displacement in the phantom was measured by comparing the current image to the first image taken, meaning the first of the ten layers of the data set has zero displacement because it is being compared to itself and the tenth layer of the data set should have the most displacement. Figure 4.1 shows the displacement components corresponding to a strain of 4.5% (layer 10) for both the lateral field ($x$-displacement) and axial field ($y$-displacement). Note that the axial direction is oriented down into the tissue or tissue-mimicking phantom, parallel to the beam of the ultrasound, and the lateral direction is side to side, perpendicular to the ultrasound beam. Thus, when we read the displacement data into our code, we used the lateral field as the $x$-component of the displacement, $u_1(x, y)$, and the axial field as the $y$-component of the displacement, $u_2(x, y)$. The figures included in this section correspond to a 64 × 128 compressed version of the original data of size 256 × 2048 in order to make the size of the problem more reasonable.

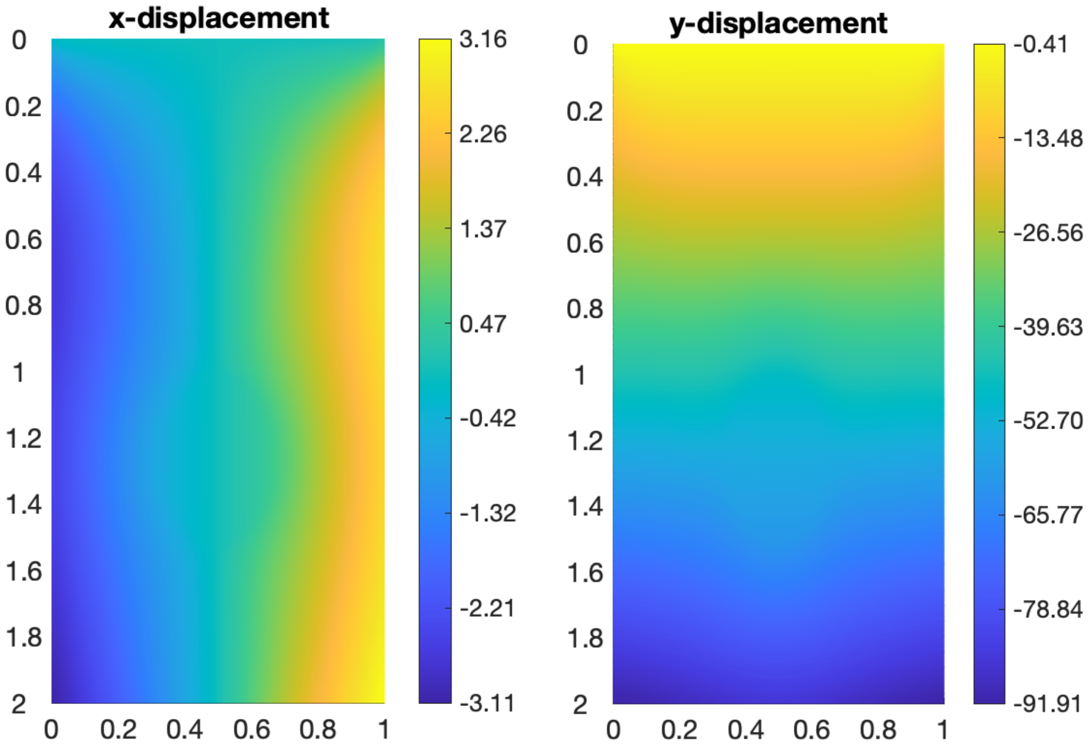


FIGURE 4.1: Displacement data (from IMPACT lab) for model experiment 2 with a single stiff inclusion.

Note that for the $y$-axis on both plots the labels are inverted, with a value of 0 at the top and a value of 2 at the bottom. This is because in [28], the axial direction ($y$-axis) represents the depth into the tissue or phantom, meaning the top of the figure corresponds to the surface of the phantom and the labeled values on the $y$-axis increase as you go deeper into the phantom. Additionally, as the boundary conditions were not provided in the work cited, we chose a set of boundary conditions which we consider to be physiologically realistic. Zero traction boundary conditions were imposed on the left and the right boundaries, meaning the tissue on those boundaries is free to move. A zero Dirichlet condition is considered for the bottom, indicating that there is no movement along that boundary (i.e. it is pinned). For the top boundary, a traction condition was chosen to mimic a uniform compression of the surface of the tissue phantom. As such, the vertical component of the function $\boldsymbol{h}$ was set to a small constant value.

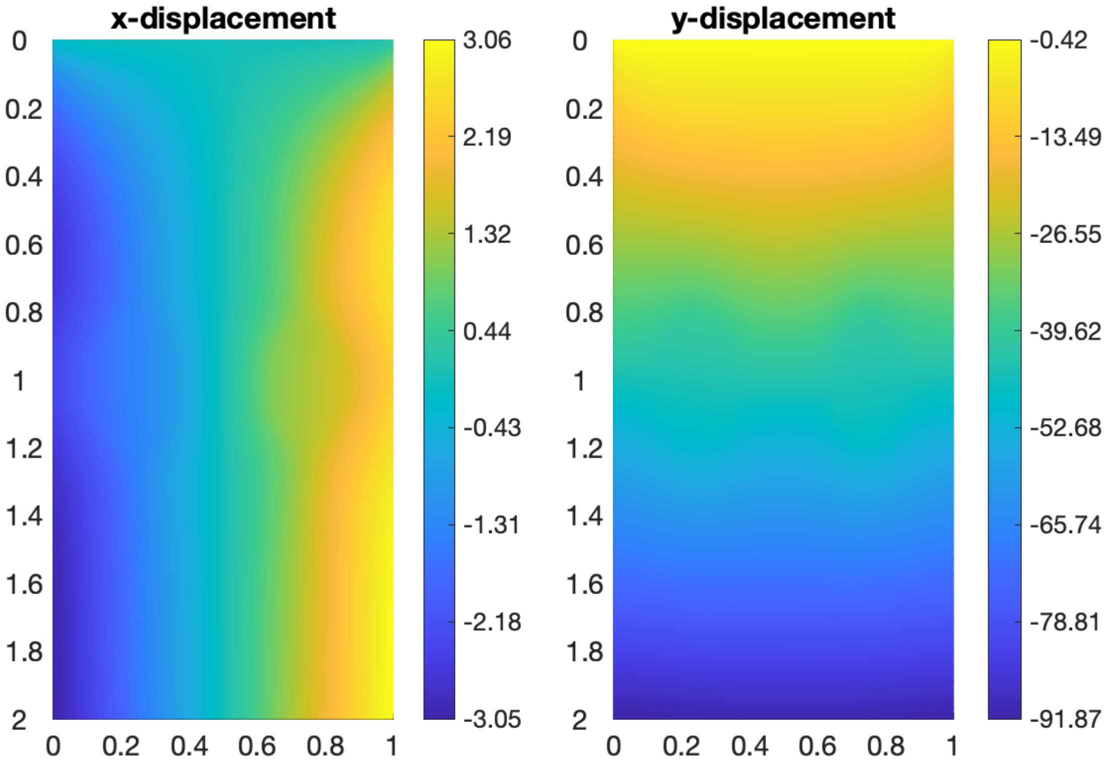


FIGURE 4.2: Displacement data (from IMPACT lab) for model experiment 10 with two stiff inclusions.

Figure 4.1 shows the displacement data obtained from a measurement with a tissue phantom that has one stiff inclusion centered approximately at $(0.5, 1.25)$ and Figure 4.2 shows the displacement components from a phantom with two stiffer inclusions located approximately at $(0.25, 1)$ and $(0.75, 1)$.

Figure 4.3 shows the results of numerical experiments using these two sets of data (one and two stiff inclusions respectively). The image on the left shows the stiff inclusion in the phantom corresponding to the displacement data shown in Figure 4.1, and the image on the right shows the two stiff inclusions corresponding to the displacement data shown in Figure 4.2. The inclusions are clearly identified and the location of the stiffer tissues in each case match those shown on the IMPACT lab webpage (referred to earlier).

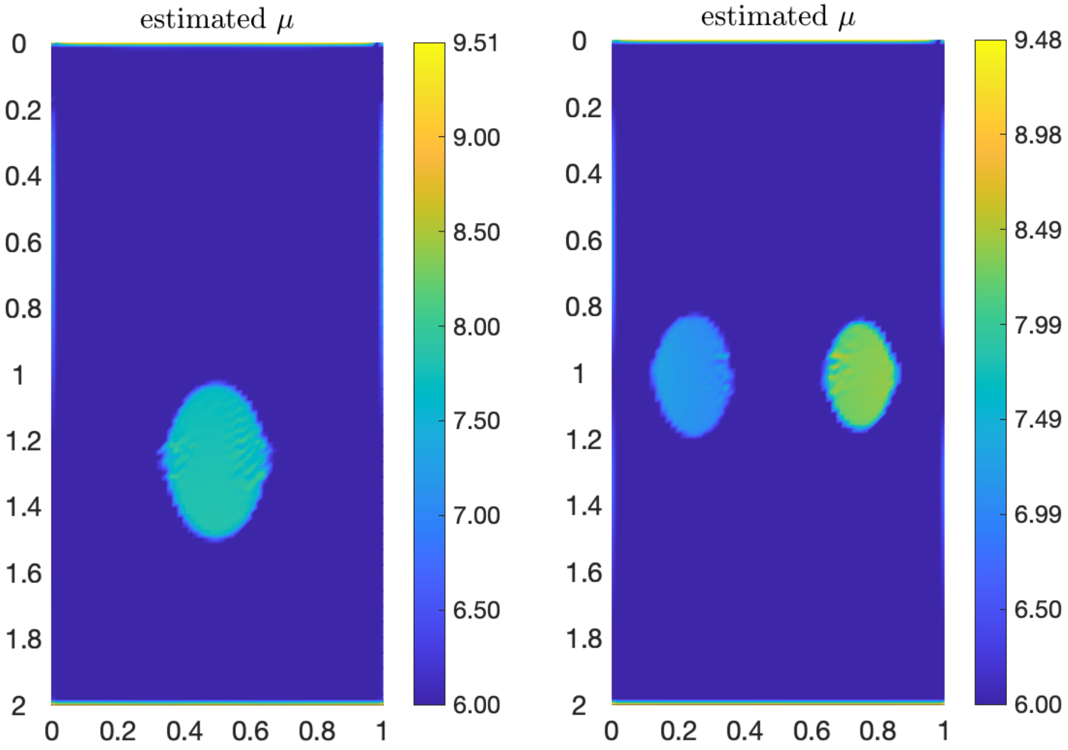


FIGURE 4.3: Estimated stiffness parameter $\mu$ corresponding to the displacement data shown in Figures 4.1 and 4.2 respectively.

## 4.2 Data Smoothing

One aspect of parameter identification methods that is relevant to the application is the presence of noise in the data. If it is known that the data is contaminated with a significant amount of noise, there are smoothing methods that can be applied to the measurement data before the iterative process starts. Fortunately, there are many data smoothing methods/techniques that could be used for this pre-processing of the data and to help mitigate problems in the identification. Among them are the methods of Moving Averages (MA), Local Polynomials (LP), and kernel smoothers. The most commonly used method for data smoothing is Moving Averages, its extension Moving Weighted Average (MWA), and their modifications. These algorithms involve replacing each data point with the average of all the data points within some window. The larger the window, the stronger the smoothing effect. An extension of this method is to use a Moving Weighted Average where the data points in the center of the window are given larger weights (i.e. original data is followed more closely). One disadvantage of the method is that it is not as robust in the presence of outliers.

Local Polynomials is another common method where the data within a window is fit to some polynomial, typically of first- or second-degree. The level of smoothing can be controlled both by the size of the window and the degree of the polynomial. Though LP handles patterns in the data better than MA or MWA, it also does not handle outliers in a robust manner. A popular extension of the LP method is Locally Weighted Scatterplot Smoothing (LOWESS or LOESS). The idea behind the LOWESS [30] method is to fit the data in the window to a polynomial and then assign weights to those points such that the weights get smaller as the data points get farther from the point being estimated. The reasoning behind such an idea is points close to one another are more likely to be related. Though the flexibility of this method is desirable, one other drawback of LOWESS is its computational expense.

The MWA and LOWESS algorithms can be considered kernel smoothers. Kernel smoothing is a general data smoothing technique that is defined by weighting the data points within a window according to some shape function, or kernel. Typically, the kernel assigns larger weights to data points that are close by and decreases smoothly towards assigning smaller weights to points that are farther away. Another potential method touted specifically for inverse problems is data smoothing by regularization such as in [31] and [32].

In a problem similar to the one that we consider, Albocher et al. [33] consider two methods of smoothing for an incompressible linear elasticity inverse problem. They consider both an averaging method and a down-sampling method where points are removed to reduce the influence of the noise. While information is lost with the second method, one way to improve those results is to interpolate the remaining points to obtain a repopulated estimate. Through experimentation they determined that the averaging method provided much better results which were reasonably close to a target solution [33]. They also obtained good results when employing a smoothing method along with adding a regularization term.

Finally, research for smoothing methods specific to ultrasound data is advancing. For example, de Araujo et al. [34] presented an averaging method for ultrasound images that yielded promising results. The idea behind this method is to identify the pixels along the border of an area of interest. These pixels would be subjected to a lower level of smoothing than the pixels in the rest of the image, a technique called low-pass filtering. The authors claim that this method outperforms other methods commonly used because it is able to adapt to the contents of the image. As indicated by many authors, the efficacy of the smoothing algorithm may depend on the type of problem it is being applied to. Thus, it will be necessary to explore which of these methods will best fit our problem of interest. Based on the preliminary literature study, however, an averaging method would be a reasonable place to start. The displacement data used in computations in Section 4.1 is obtained from measurements in a lab, and therefore contains measurement errors (noise). The results shown in Figure 4.3 capture the characteristics of the underlying phantom quite well and it is difficult to conclude if or how the computational framework, or the choice of the objective functional, is influenced by the measurement noise in the data.

## 4.3 Quantitative Analysis

This set of analyses were performed for the numerical example in Section 3.5.1 where the transition between the background tissue and the inclusion was continuous and the problem was solved using a uniformly refined mesh (see Figure 3.2). Simply looking at the figures to determine the quality of the approximation, while informative, is a very qualitative approach. It is helpful to also have more quantitative analyses of how the computational framework is performing, as well as being able to compare various modifications of that framework.

One of the most common quantitative measures to examine is error in the recovered coefficient. It is sufficient to use an error measure for the recovered parameter such as

$$E = \|\mu - \mu_h\|_{L^2}.$$

Besides the mesh size or minimum cell diameter, $h$, there are a few other computational parameters that contribute to the error in our framework. These are the regularization parameter, $\beta$, and the step size in the stochastic gradient algorithm, $\alpha$. In order to quantify contributions of these parameters in the total error $E$, we carry out multiple numerical experiments on uniform meshes for a problem with an analytical solution. In addition to error, we record the number of iterations, degrees of freedom (dofs), and the computational times. Several experiments were run, the first of which was varying the mesh size, and the error versus the number of iterations was plotted.

In Figure 4.4, we display the number of iterations needed and the $L^2$-norm error. In this set of experiments, the only parameter that changes is the mesh size $h$. The code was run until a norm of the objective functional gradient falls below $5 \cdot 10^{-6}$ (a stopping criterion); due to this criterion, the total number of iterations was different for various mesh sizes.

Without a theoretical error estimate, it is difficult to predict and quantify the contributions of a chosen set of parameters in the total error. Nevertheless, we collect our observations from the numerical experiments conducted. In Figure 4.4, we notice that the decrease in error is more rapid and a smaller error is achieved for coarser meshes when compared at the same number of iterations. This is not what is expected intuitively, however, we need to take into account that the errors in the solution are not only due to the discretization of the PDE itself but also from the optimization method used in the recovery of the parameter. As the mesh gets finer, it is typically the case that more regularization (i.e. a larger regularization parameter $\beta$) is necessary. In this particular set of experiments, the regularization parameter is kept constant at $\beta = 10^{-5}$ regardless of the varying mesh size and this would explain the aforementioned error behaviour.

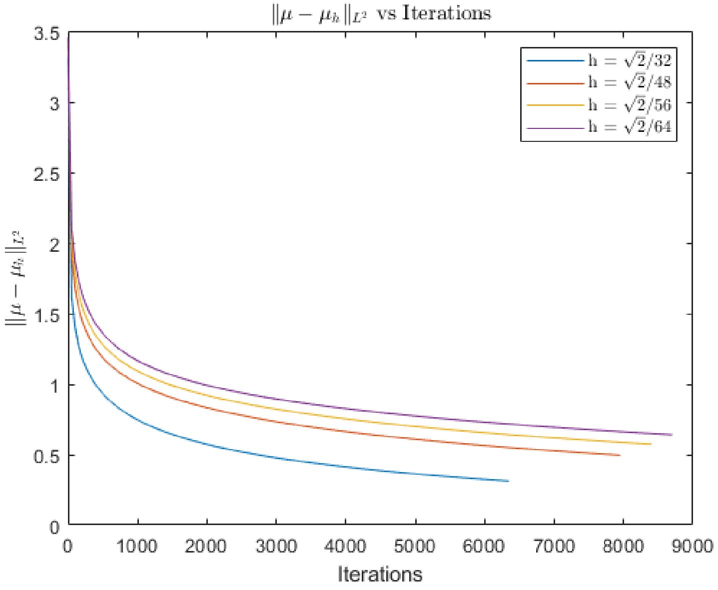


FIGURE 4.4: Error vs iteration plot for 32-64 subdivisions of the mesh with piecewise quadratic elements for $\overline{\boldsymbol{u}}$ and piecewise linear elements for $p$.

We repeat the same set of experiments (varying mesh sizes and all other parameters kept the same) using another stable finite element combination for the mixed formulation of the system, piecewise quadratic elements for the displacement $\overline{\boldsymbol{u}}$ and piecewise constant elements for the pressure $p$ (a $Q_2 - Q_0$ combination). The results of this experiment are shown in Figure 4.5. Recall that all numerical experiments in Chapters 2 and 3, as well as those resulted in Figure 4.4 used piecewise quadratic elements for $\overline{\boldsymbol{u}}$ and piecewise linear elements for $p$ (the so-called $Q_2 - Q_1$ combination). With the piecewise constant representation of the pressure variable, we reduce the dimension of the forward problem that we solve in every iteration, which leads to savings in computational time.

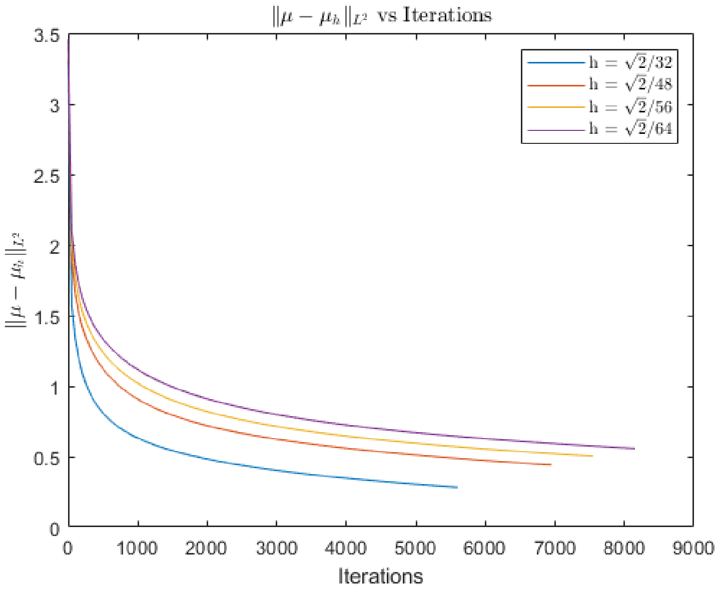


FIGURE 4.5: Error vs iteration plot for 32-64 subdivisions of the mesh with piecewise quadratic elements for $\overline{\boldsymbol{u}}$ and piecewise constant elements for $p$.

Comparisons of the two combinations of elements are shown more clearly in Figures 4.6-4.8 where the error vs iteration curves are directly compared for various mesh sizes.

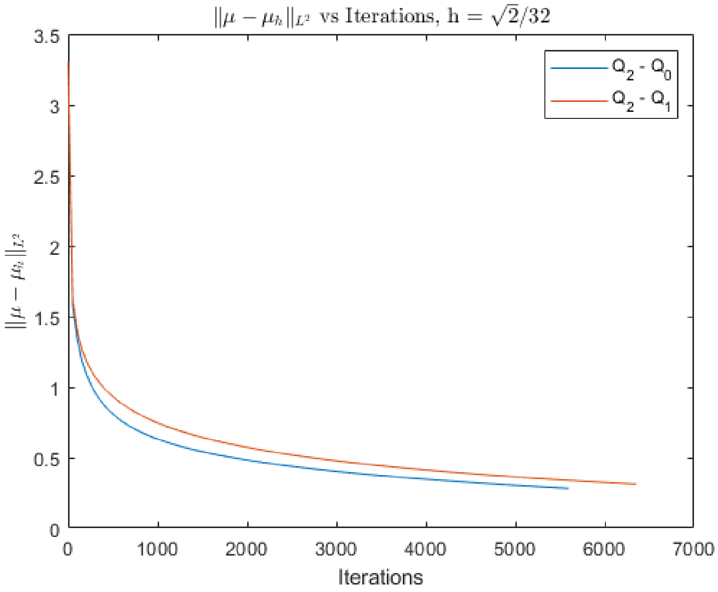


FIGURE 4.6: Error vs iteration plot for 32 subdivisions of the mesh (a 32 × 32 mesh) comparing the two different combinations of elements, $Q_2 - Q_1$ and $Q_2 - Q_0$.

We see that the combination with $Q_0$ elements for the pressure routinely needs fewer iterations and leads to a smaller error in the solution than the combination with $Q_1$ elements.

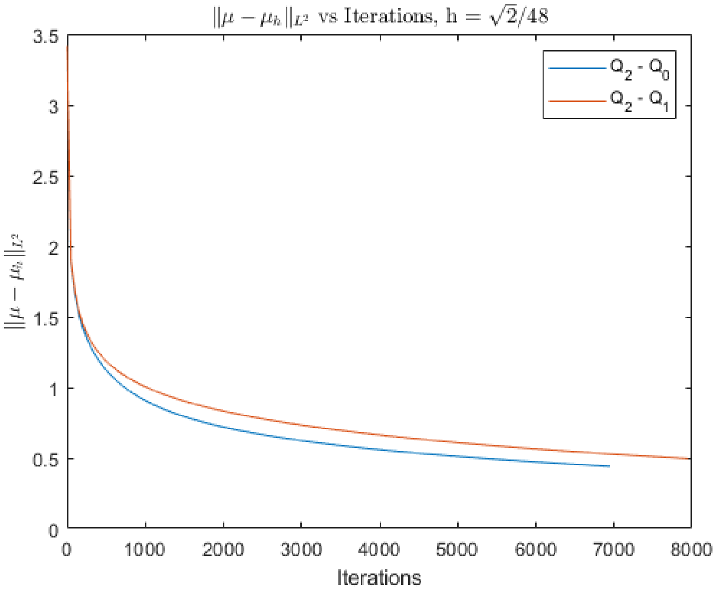


FIGURE 4.7: Error vs iteration plot for 48 subdivisions of the mesh (a 48 × 48 mesh) comparing the two different combinations of elements, $Q_2 - Q_1$ and $Q_2 - Q_0$.

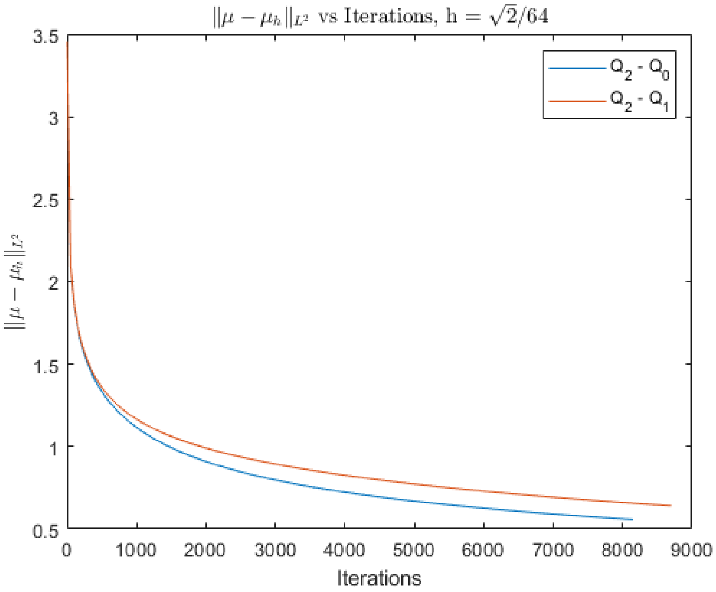


FIGURE 4.8: Error vs iteration plot for 64 subdivisions of the mesh (a 64 × 64 mesh) comparing the two different combinations of elements, $Q_2 - Q_1$ and $Q_2 - Q_0$.

Tables 4.1 and 4.2 show the computational times along with the degrees of freedom for the pressure variable after a fixed number of iteration steps are completed for the $Q_2 - Q_1$ and $Q_2 - Q_0$ element combinations respectively for various subdivisions of the mesh.

| subdivisions | # dofs ($p$) | CPU time (s) |
|---:|---:|---:|
| 32 | 1089 | 15309.17 |
| 48 | 2401 | 38019.75 |
| 56 | 3249 | 53219.09 |
| 64 | 4225 | 71480.37 |

TABLE 4.1: Degrees of freedom (dofs) for the pressure and computational time (seconds) after 5000 iterations for the piecewise quadratic/piecewise linear ($Q_2 - Q_1$) element combination for various mesh sizes.

| subdivisions | # dofs ($p$) | CPU time (s) |
|---:|---:|---:|
| 32 | 1024 | 12814.77 |
| 48 | 2304 | 30821.35 |
| 56 | 3136 | 43501.81 |
| 64 | 4096 | 57999.97 |

TABLE 4.2: Degrees of freedom (dofs) for the pressure and computational time (seconds) after 5000 iterations for the piecewise quadratic/piecewise constant ($Q_2 - Q_0$) element combination for various mesh sizes.

Table 4.3 gives a direct comparison between these two element combinations for one mesh size. The degrees of freedom used in representing the pressure variable decrease from 1089 to 1024 when switching from a piecewise linear to a piecewise constant representation and we observe a decrease of about 16% in the total computational time after a fixed number of (5000) iterations are completed.

| element combination | # dofs ($p$) | CPU time (s) |
|---:|---:|---:|
| $Q_2 - Q_1$ | 1089 | 15309.17 |
| $Q_2 - Q_0$ | 1024 | 12814.77 |

TABLE 4.3: Degrees of freedom (dofs) for the pressure and computational time (seconds) after 5000 iterations for the two combinations of elements on a $32 \times 32$ mesh.

Next, we explore the effects of varying the regularization parameter, $\beta$, for two different (fixed) mesh sizes. Notice in Figures 4.9 and 4.10 that, though it requires more iterations overall to achieve a certain prescribed accuracy (i.e. the process is terminated after the norm of the gradient of the objective functional falls below a fixed tolerance), the simulation with the largest $\beta$ value, $\beta = 5 \cdot 10^{-5}$, leads to the smallest error when compared with the trends for the other values of $\beta$ after the same number of iterations are completed. This is consistent with the expectations stated previously, that a larger regularization parameter yields better results for finer meshes.

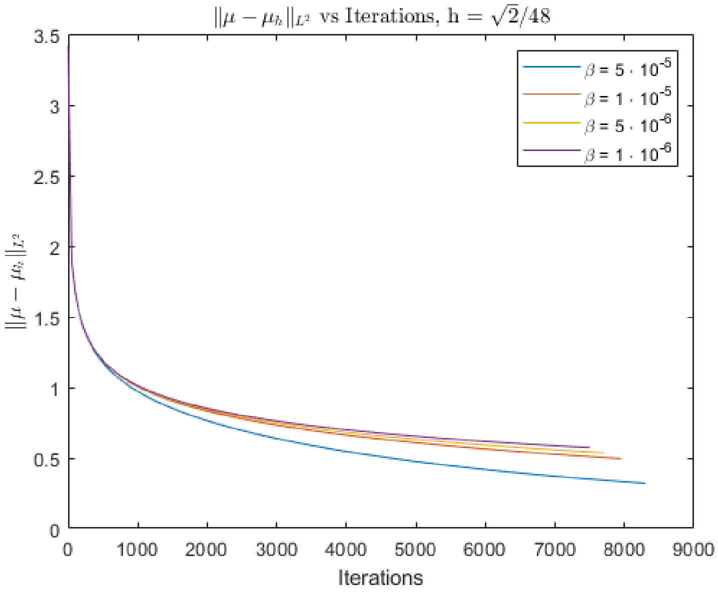


FIGURE 4.9: Error vs iteration plot for 48 subdivisions of the mesh (a $48 \times 48$ mesh) for various $\beta$ values for the $Q_2 - Q_1$ element combination.

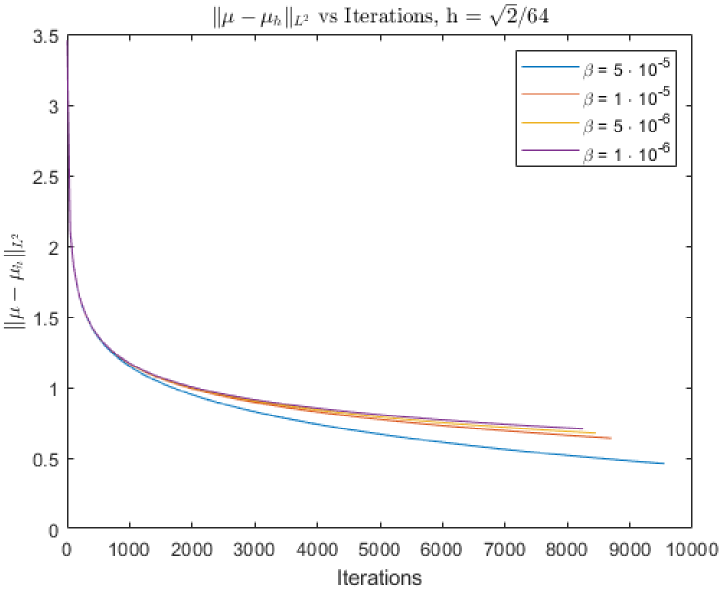


FIGURE 4.10: Error vs iteration plot for 64 subdivisions of the mesh (a 64 × 64 mesh) for various $\beta$ values for the $Q_2 - Q_1$ element combination.

The results discussed here are a preliminary study on understanding the accuracy of the solver from the problem parameters as well as the computational parameters. A more thorough and systematic theoretical and numerical study that contributes to understanding the complex interplay of these computational parameters in the error is necessary.

# Chapter 5

# Stochastic Model of Elastography

This chapter is focused on a stochastic linear elasticity system and the inverse problem of identifying a random stiffness parameter in the system. The goal is to develop an adaptive finite element solution framework for the identification of a distributed parameter in a system of stochastic partial differential equations. A convex optimization framework is applied in this setting, and we use a stochastic approximation method with finite element discretization for the numerical solution of the problem. A description of a finite element discretization that fits the optimization framework and stochastic approximation scheme used in the numerical solution is given. We also propose an adaptive mesh refinement framework that provides the resolution needed for the recovery of the spatially varying parameter while improving computational efficiency.

## 5.1 An Example of a Stochastic Model

The motivation behind using a stochastic parameter for tissue stiffness is that the parameter can be different for each patient. This would allow natural uncertainty to be inserted into the model, as the parameter would be allowed to take on a range of values. Using a statistical method would give a probability distribution for the parameter, rather than a single fixed value. The authors of [18] considered a model of nearly incompressible elasticity where Young's modulus $E$ is a spatially varying random field given by

$$E(\boldsymbol{x}, \boldsymbol{r}) = e_0(\boldsymbol{x}) + \sum_{k=1}^{M} e_k(\boldsymbol{x}) r_k$$

where $\boldsymbol{r} = [r_1, r_2, \ldots, r_M]$ and $r_k$, $k = 1, \ldots, M$ are random parameters whose distributions are known. The tissue stiffness parameter $\mu$ is therefore a random variable in this model (see also equations (1.3) and (1.4)). The authors analyzed the model and introduced a solution strategy that uses the stochastic Galerkin method (see also [25] where the authors presented a posteriori error estimates for the stochastic Galerkin formulation of the problem).

To turn the PDE into a stochastic PDE (SPDE), the stiffness parameter will now be $\mu(x, y, \omega)$, meaning $\mu$ will depend on the random variable $\omega$. Since $\mu$ depends on this random variable, and $\overline{\boldsymbol{u}}$ depends on $\mu$, the displacement will depend on $\omega$ as well, $\overline{\boldsymbol{u}}(x, y, \omega)$. For example, we can assume that the tissue stiffness parameter $\mu$ has the expansion

$$\mu = \mu_1(x, y) + \mu_2(x, y)Y_1(\omega) + \mu_3(x, y)Y_2(\omega)$$

where $Y_1(\omega)$ and $Y_2(\omega)$ are random variables and $\mu_1, \mu_2, \mu_3$ are the deterministic parts of $\mu$. Due to the appearance of random variables $Y_1(\omega)$ and $Y_2(\omega)$ in the stiffness parameter and the force function $\boldsymbol{f}$, we have a random displacement $\overline{\boldsymbol{u}}(x, y, Y_1(\omega), Y_2(\omega))$ in the model. We assume that the distributions of random variables $Y_1(\omega)$ and $Y_2(\omega)$ are known. Then the inverse problem that we have is to recover $\mu_i(\boldsymbol{x})$, $i = 1, 2, 3$ in the system.

The stochastic model with homogeneous Dirichlet boundary conditions in its original form is given by

$$\nabla \cdot \sigma(\omega, \boldsymbol{x}) = \boldsymbol{f}(\omega, \boldsymbol{x}) \qquad \text{on } \Omega \times D \tag{5.1}$$

$$\overline{\boldsymbol{u}}(\omega, \boldsymbol{x}) = \boldsymbol{0} \qquad \text{on } \partial D \tag{5.2}$$

where $\sigma = 2\mu(\omega, \boldsymbol{x})\varepsilon_{\overline{\boldsymbol{u}}}(\omega, \boldsymbol{x}) + \lambda \mathrm{div}(\overline{\boldsymbol{u}}(\omega, \boldsymbol{x}))I$, $D$ is the domain, and $\Omega$ is the set of elementary events in the probability space.

There are two types of variational forms that could be used for solving the stochastic forward problem, a pathwise formulation and an integral formulation. In the integral formulation, we would have a double integral where the inner integral is with respect to the space variable just like the weak problem for the deterministic case and the outer integral is with respect to the probability measure. The pathwise version utilizes a fixed realization of the random variable $\omega$, and it is considered a deterministic problem parameterized by the random variable $\omega$. This is the version we need for the type of solution methods we are proposing.

## 5.2 Stochastic Forward Problem

A commonly used numerical method for solving stochastic PDEs is the sampling-based Monte-Carlo method. For this method, $s$ number of realizations of the random variables $r_k$ are generated, and for each realization the solution $\overline{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{r})$ is obtained where the data are the (random) material parameters and the force $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{r})$ (as well as boundary conditions). Similar to the deterministic case, the stochastic version of the model equations can be written as

$$-\nabla \cdot (2\mu(\omega, \boldsymbol{x})\varepsilon_{\overline{\boldsymbol{u}}}(\omega, \boldsymbol{x}) + \lambda \mathrm{div}(\overline{\boldsymbol{u}}(\omega, \boldsymbol{x}))I) = \boldsymbol{f}(\omega, \boldsymbol{x}) \tag{5.3}$$

where the solution $\overline{\boldsymbol{u}}$, and the data $\mu$ and $\boldsymbol{f}$ belong to appropriate spaces. The data $\mu = \mu(\omega, \boldsymbol{x})$ (or the Young modulus $E = E(\omega, \boldsymbol{x})$) and $\boldsymbol{f}(\omega, \boldsymbol{x})$ are random fields with specified distributions and our task is to find $\overline{\boldsymbol{u}}(\omega, \boldsymbol{x})$ such that the equations hold (almost everywhere) in the weak sense. Equation (5.3) can easily be rewritten using the equations from the mixed formulation where the pressure $p = p(\omega, \boldsymbol{x})$ is also random. In the Monte Carlo Finite Element Method (MCFEM), if we have, for example, a deterministic $\boldsymbol{f} = \boldsymbol{f}(\boldsymbol{x})$, we transform the weak problem to a deterministic one using realizations of the shear modulus $\mu(\omega, \boldsymbol{x})$.

The main idea of MCFEM is that given i.i.d samples of approximate shear modulus $\mu_r = \mu(\omega_r, \boldsymbol{x})$, $r = 1, \ldots, Q_1$, we generate $Q_1$ i.i.d samples of $\overline{\boldsymbol{u}}_{h,r} = \overline{\boldsymbol{u}}_h(\omega_r, \boldsymbol{x})$ of the finite element solution $u_h(\omega, \boldsymbol{x})$ by solving $Q_1$ variational problems. The mean $\mathbb{E}[\overline{\boldsymbol{u}}(\omega, \boldsymbol{x})]$ is then estimated by

$$\frac{1}{Q_1}\sum_{r=1}^{Q_1} \overline{\boldsymbol{u}}_{h,r}.$$

Other methods that are commonly used by researchers include stochastic Galerkin and stochastic collocation methods (see, for example, [35]).

We discuss the results of a numerical experiment where we chose an example where Young's modulus $E = 1$ in a deterministic problem with a known solution (see Experiment 1 in Chapter 2) is changed to a random Young's modulus given as

$$E = 1 + r_1$$

where $r_1$ is a random variable distributed uniformly on $[-0.1, 0.1]$. Recall that the relation between the shear modulus (tissue stiffness parameter) and Young's modulus $E$ is given by $\mu = \dfrac{E}{2(1+\nu)}$. Figure 5.1 shows the approximation of the mean $\mathbb{E}[\overline{\boldsymbol{u}}(\omega, \boldsymbol{x})]$, where we used 100 samples of $r_1$ for the computation of the approximate mean. The results of the numerical experiment have shown that the mean of the tissue stiffness parameter agrees nicely with the solution of the corresponding deterministic problem whose solutions are known.

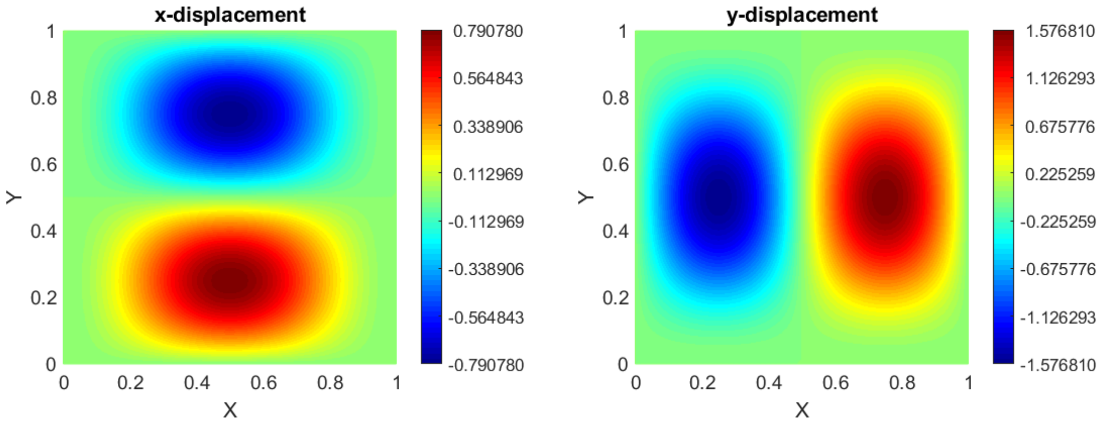


FIGURE 5.1: Componentwise view of an approximation of $\mathbb{E}[\overline{\boldsymbol{u}}(\omega, \boldsymbol{x})]$ computed using MCFEM with 100 samples.

## 5.3 Stochastic Optimization Problem

We describe an optimization formulation for the inverse problem with a stochastic linear elasticity system in this section. We consider the case where the tissue stiffness parameter is considered as a random field $\mu(\omega, \boldsymbol{x})$, where $\omega$ is a random variable from a probability space with the set of elementary events $\Omega$.

The forward problem is to find a random field $\overline{\boldsymbol{u}} : \Omega \times D \to \mathbb{R}$ satisfying (5.1) - (5.2) almost surely for given functions $\boldsymbol{f}(\omega, \boldsymbol{x})$ and $\mu(\omega, \boldsymbol{x})$. The following discrete form of the forward problem will be needed for computations: Find $\boldsymbol{U} = \boldsymbol{U}(\omega, \boldsymbol{M})$ that satisfies $K(\boldsymbol{M})\boldsymbol{U}(\omega, \boldsymbol{M}) = \boldsymbol{F}(\omega)$ where the block matrix $K = K(\boldsymbol{M})$ is defined by

$$K(\boldsymbol{M}) = \begin{bmatrix} A(\boldsymbol{M}) & B \\ B^T & C \end{bmatrix}.$$

The matrix is the parameterized version of the deterministic case since we use a pathwise formulation of the problem in the solution method we propose. We consider the trilinear form $T(\mu, \boldsymbol{u}(\mu) - \boldsymbol{z}, \boldsymbol{u}(\mu) - \boldsymbol{z})$ where $\boldsymbol{u}(\mu) = \boldsymbol{u}(\mu)(\omega, \boldsymbol{x})$ is the solution to the forward problem for a given parameter $\mu(\omega, \boldsymbol{x})$. Using (2.6), we get

$$\begin{aligned} \mathbb{J}(\mu) &= a(\mu, \overline{\boldsymbol{u}}(\mu) - \overline{z}, \overline{\boldsymbol{u}}(\mu) - \overline{z}) + b(\overline{\boldsymbol{u}}(\mu) - \overline{z}, p(\mu) - \hat{z}) \\ &\quad - b(\overline{\boldsymbol{u}}(\mu) - \overline{z}, p(\mu) - \hat{z}) + c(p(\mu) - \hat{z}, p(\mu) - \hat{z}) \\ &= a(\mu, \overline{\boldsymbol{u}}(\mu) - \overline{z}, \overline{\boldsymbol{u}}(\mu) - \overline{z}) + c(p(\mu) - \hat{z}, p(\mu) - \hat{z}) \\ &= \int_D 2\mu \varepsilon_{\overline{\boldsymbol{u}}(\mu) - \overline{z}} \cdot \varepsilon_{\overline{\boldsymbol{u}}(\mu) - \overline{z}} + \int_D \frac{1}{\lambda}(p(\mu) - \hat{z})^2. \end{aligned}$$

We define a set of admissible parameters $\mu$ by

$$\mathbb{M} = \left\{ \mu \in L^2(\Omega, D), 0 < \mu_0 < \mu(\omega, \boldsymbol{x}) \le \mu_1 < \infty \ \text{ a.e. in } \Omega \times D \right\}.$$

Then the optimization problem is formulated as follows: Find $\mu \in \mathbb{M}$ such that the modified output least-squares (MOLS) functional $\mathbb{J}$ defined by

$$\mathbb{J}(\mu) := \frac{1}{2}\mathbb{E}\left[ \int_D 2\mu \varepsilon_{\overline{\boldsymbol{u}}(\mu) - \overline{z}} \cdot \varepsilon_{\overline{\boldsymbol{u}}(\mu) - \overline{z}} + \int_D \frac{1}{\lambda}(p(\mu) - \hat{z})^2 \right] \tag{5.4}$$

is minimized. Here, $\mathbb{E}$ stand for the expectation with respect to the probability space and $z(\omega, \boldsymbol{x}) = (\overline{z}, \hat{z})$ are the data (measurements of $(\overline{\boldsymbol{u}}, p)$). The regularized version of the modified output least-squares functional needed for stable identification is given by

$$\mathbb{J}_\beta(\mu) := \frac{1}{2}\mathbb{E}\left[ \int_D 2\mu \varepsilon_{\overline{\boldsymbol{u}}(\mu) - \overline{z}} \cdot \varepsilon_{\overline{\boldsymbol{u}}(\mu) - \overline{z}} + \int_D \frac{1}{\lambda}(p(\mu) - \hat{z})^2 + \frac{\beta}{2}R(\mu) \right], \tag{5.5}$$

where $R(\mu)$ is the regularization term (to be specified later), and $\beta$ is a regularization parameter.

Given $r$ random variables $\xi_t : \Omega \mapsto \Gamma_t$, for $t = 1, \dots, r$, a function $v \in L^2(\Omega; L^2(D))$ of the form $v(\xi(\omega), \boldsymbol{x})$ for $\boldsymbol{x} \in D$ and $\omega \in \Omega$, where $\xi = (\xi_1, \xi_2, \dots, \xi_r) : \Omega \mapsto \Gamma \subset \mathbb{R}^r$ and $\Gamma := \Gamma_1 \times \Gamma_2 \cdots \times \Gamma_r$ is called a finite-dimensional noise (see [35]). Most solution methods for stochastic PDEs require an assumption that is called the finite-dimensional noise assumption involving a finite number of real-valued random variables. With the assumption, the expectation is expressed as an integral over an observation space. If a random field $v(\xi, \boldsymbol{x})$ is a finite-dimensional noise, a change of variables can be made for evaluating expectations. Using finite-dimensional noise representations for the random fields involved helps transform the stochastic system into a high-dimensional parametric system. Under the assumption, the random field $\mu$ takes the form

$$\mu(\omega, \boldsymbol{x}) = \mu_0(\boldsymbol{x}) + \sum_{t=1}^{r} \mu_t(\boldsymbol{x}) Y_t(\omega) = \sum_{t=0}^{r} \mu_t(\boldsymbol{x}) Y_t(\omega)$$

where $Y_0(\omega) = 1$ by convention and $Y_t(\omega)$, $t = 0, 1, ..., r$ are real-valued random variables. With the finite dimensional expansion for $\mu$, the variational form can be written as

$$\begin{aligned} \mathbb{E}\left[\int_D 2\left(\sum_{t=0}^{r} \mu_t(\boldsymbol{x}) Y_t(\omega)\right) \varepsilon_{\overline{\boldsymbol{u}}} \cdot \varepsilon_{\overline{\boldsymbol{v}}} - \int_D p \operatorname{div}(\overline{\boldsymbol{v}})\right] &= \mathbb{E}\left[\int_D \boldsymbol{f}\overline{\boldsymbol{v}}\right], \\ \mathbb{E}\left[\int_D q \operatorname{div}(\overline{\boldsymbol{u}}) + \int_D \frac{1}{\lambda} pq\right] &= 0, \end{aligned}$$

where $\overline{\boldsymbol{u}} = \overline{\boldsymbol{u}}(\omega, \boldsymbol{x})$, $p = p(\omega, \boldsymbol{x})$, $\overline{\boldsymbol{v}} = \overline{\boldsymbol{v}}(\omega, \boldsymbol{x})$, and these functions live in appropriate Bochner spaces. Therefore, $\mathbb{J}(\mu) = \mathbb{E}\left[\mathbb{J}(\omega, \mu)\right]$ where

$$\mathbb{J}(\omega, \mu) := \frac{1}{2} \sum_{t=0}^{r} Y_t(\omega) \int_D 2\mu_t(\boldsymbol{x}) \varepsilon_{\overline{\boldsymbol{u}}(\mu) - \overline{z}} \cdot \varepsilon_{\overline{\boldsymbol{u}}(\mu) - \overline{z}} + \int_D \frac{1}{\lambda}(p(\mu) - \hat{z})^2. \tag{5.6}$$

Furthermore, it can be shown that $D\mathbb{J}(\mu)(\delta\mu) = \mathbb{E}[T(\omega, \mu)(\delta\mu)]$ where

$$\begin{aligned} T(\omega, \mu)(\delta\mu) := \sum_{t=0}^{r} &\left[-\frac{1}{2} a(\delta\mu_t(\boldsymbol{x}), \overline{\boldsymbol{u}}(\mu) + \overline{z}, \overline{\boldsymbol{u}}(\mu) - \overline{z})\right] Y_t(\omega) \\ &- b(\overline{\boldsymbol{u}}(\mu), \delta p) + b(\delta\overline{\boldsymbol{u}}, p(\mu) - \hat{z}). \end{aligned}$$

Consequently, $\nabla_\mu \mathbb{J}(\mu) = \nabla_\mu \mathbb{E}\left[\mathbb{J}(\omega, \mu)\right] = \mathbb{E}\left[\nabla_\mu \mathbb{J}(\omega, \mu)\right]$ and

$$\nabla_\mu \mathbb{J}(\omega, \mu) = \nabla_\mu T(\omega, \mu) = \left(\frac{\partial T}{\partial \mu_t}(\omega, \mu)\right)_{t=0,\dots,r}$$

where we have

$$\frac{\partial T}{\partial \mu_t}(\omega, \mu)(\cdot) = \left[-\frac{1}{2}\int_D a(\cdot, \overline{\boldsymbol{u}}(\mu) + \overline{z}, \overline{\boldsymbol{u}}(\mu) - \overline{z})\right] Y_t(\omega) \tag{5.7}$$

for $t = 0, ..., r$.

## 5.4 Stochastic Approximation Scheme

In this section, we discuss details of the discretization that are needed for a numerical implementation. Discretization of the forward problem is standard and found in existing literature, therefore we focus on the relevant details for the inverse problem. The stochastic approximation method that we consider uses the pathwise formulation of the forward problem and we provide a brief description of the discrete version needed for numerical implementation. We propose a regularized projected stochastic approximation scheme and provide discrete formulas for the objective functional along with its gradients needed for implementation.

Due to the expansion $\mu(\omega, \boldsymbol{x}) = \mu_0(\boldsymbol{x}) + \sum_{t=1}^{r} \mu_t(\boldsymbol{x}) Y_t(\omega) = \sum_{t=0}^{r} \mu_t(\boldsymbol{x}) Y_t(\omega)$, the matrix $A(\boldsymbol{M})$ can be decomposed as follows: For $i, j = 1, \ldots, n$,

$$\begin{aligned} A(\boldsymbol{M}(\omega))_{i,j} &= a(\mu(\omega, \boldsymbol{x}), \psi_j, \psi_i) = a\left(\sum_{t=0}^{r} \mu_t(\boldsymbol{x}) Y_t(\omega), \psi_j, \psi_i\right) \\ &= \sum_{t=0}^{r} Y_t(\omega) a(\mu_t(\boldsymbol{x})\ \psi_j, \psi_i) = \sum_{t=0}^{r} Y_t(\omega) A(\boldsymbol{M}_t), \end{aligned} \tag{5.8}$$

where

$$(A(\boldsymbol{M}_t))_{i,j} := a(\mu_t(\boldsymbol{x}), \psi_j, \psi_i) \text{ for } i, j = 1, \ldots, n.$$

With $\boldsymbol{M}_t$, the discrete representation of $\mu_t(\boldsymbol{x})$, we will use a stochastic approximation scheme in terms of the spatial components $\boldsymbol{M}^{(k)} = (\boldsymbol{M}_0^{(k)}, ..., \boldsymbol{M}_r^{(k)})$. From (5.7), we get the discrete form of the gradient

$$\frac{\partial T}{\partial \boldsymbol{M}_t}(\omega, \boldsymbol{M}) = -\frac{1}{2}\left[\boldsymbol{S}\left(\overline{\boldsymbol{U}}(\omega, \boldsymbol{M}) + \overline{\boldsymbol{Z}}(\omega)\right)\right]^T \left[\overline{\boldsymbol{U}}(\omega, \boldsymbol{M}) - \overline{\boldsymbol{Z}}(\omega)\right] Y_t(\omega) \tag{5.9}$$

for $t = 0, \ldots, r$ where $\boldsymbol{S} \in \mathbb{R}^{n \times m}$ is the adjoint stiffness matrix defined by the condition that

$$\boldsymbol{S}(\boldsymbol{V})\widetilde{\boldsymbol{M}} = A(\widetilde{\boldsymbol{M}})\boldsymbol{V} \text{ for every } \boldsymbol{V} \in \mathbb{R}^n \text{ and } \widetilde{\boldsymbol{M}} \in \mathbb{R}^m.$$

We emphasize that the $\widetilde{\boldsymbol{M}}$ used here only for the purposes of this definition is a vector of dimension $m$ and is different from the vector $\boldsymbol{M}$ we've been using elsewhere, which has dimension $(r+1)m$. Block vectors $\overline{\boldsymbol{Z}}$ and $\overline{\boldsymbol{U}}(\boldsymbol{M})$ stand for displacement data and the computed/simulated displacement (corresponding to a given parameter represented by $\boldsymbol{M}$) respectively. Given the random vector $(Y_0(\omega), ..., Y_r(\omega))$, we will use the notation

$$\boldsymbol{M}^{(k)}(\omega) = \sum_{t=0}^{r} \boldsymbol{M}_t^{(k)} Y_t(\omega).$$

for the corresponding realization. We propose the following projected-gradient-type stochastic approximation scheme, applicable for general variational inequalities. For a similar stochastic optimization problem, stemming from a scalar PDE model, the modified output least-squares objective functional is known to be convex in the interior of the set $\mathbb{M}$ (see [36] and [37]) and a necessary and sufficient optimality condition for the stochastic optimization problem is a stochastic variational inequality. For the variational inequality of finding $\mu \in \mathbb{M}$ such that $\langle F(\mu), v - \mu \rangle \geq 0$ for all $v \in \mathbb{M}$, the scheme takes the following form:

(i) Start with $\mu_0 \in \mathbb{M}$ with $\mathbb{E}\left[\|\mu_0\|^2\right] < \infty$.

(ii) At step $k$, compute $\mu_{k+1} \in \mathbb{M}$ by

$$\mu_{k+1} = P_{\mathbb{M}}\left[\mu_k - \alpha_k \left(F(k_n) + \beta_k \mu_k + \omega_k\right)\right] \tag{5.10}$$

where $\alpha_k$ is the variable step size, $\beta_k$ is the regularization parameter, $\omega_k$ is a realization of the random variable $\omega$, and $P_{\mathbb{M}}$ is the projection onto the set of admissible parameters $\mathbb{M}$. Sequences for step lengths $\{\alpha_k\}$, regularization parameters $\{\beta_k\}$, and the sample rate $\{s_k\} \subset \mathbb{N}$, are updated at every iteration. Additional details of the regularized stochastic approximation method can be found in [22] and [38], including the conditions on the parameter sequences and a proof of convergence of the following algorithm:

1. Choose an initial guess $\boldsymbol{M}^{(0)}$, step length $\alpha_0$, sample rate $s_0$, and initial samples $\{\omega_j^0\}_{j=1}^{s_0}$ of the random variable $\omega$.
2. Generate a random vector $(1, y_1(\omega_k), ..., y_r(\omega_k))$.
3. Compute $\boldsymbol{U}^{(k)} = \boldsymbol{U}(\omega_k, \boldsymbol{M}^{(k)})$ by solving the system

$$K(\boldsymbol{M}^{(k)}(\omega_k))\boldsymbol{U}^{(k)} = \boldsymbol{F}(\omega_k).$$

4. Given $\boldsymbol{M}^{(k)}$ in the admissible set $\mathbb{M}$, generate samples $\{\omega_j^k\}_{j=1}^{s_k}$ of $\omega$ and compute $\boldsymbol{M}^{(k+1)} \in \mathbb{M}$ by

$$\boldsymbol{M}^{(k+1)} = P_{\mathbb{M}}\left[\boldsymbol{M}^{(k)} - \frac{\alpha_k}{s_k}\sum_{j=1}^{s_k} G_{\beta_k}\left(\omega_j^k, \boldsymbol{M}^{(k)}\right)\right], \tag{5.11}$$

   where $G_{\beta_k}$ is the discrete variant of the gradient of the regularized MOLS objective functional. Here, $P_{\mathbb{M}}$ stands for the projection into the admissible parameter set.

Note that in the process above, we modify the classical stochastic gradient scheme by sampling the regularized gradient at a given sampling rate. As for the regularizer, for example, we can take the derivative of the discrete form of the $H^1$-norm regularizer and add to the gradient components given in (5.9).

## 5.5 A Numerical Example

In this section, we address some elements of the adaptive mesh refinement we use in computations and present a numerical example. We created a simple analytical example to test the accuracy and correctness of the parameter estimation process. The components of the exact displacement $\overline{\boldsymbol{u}}$ are chosen as

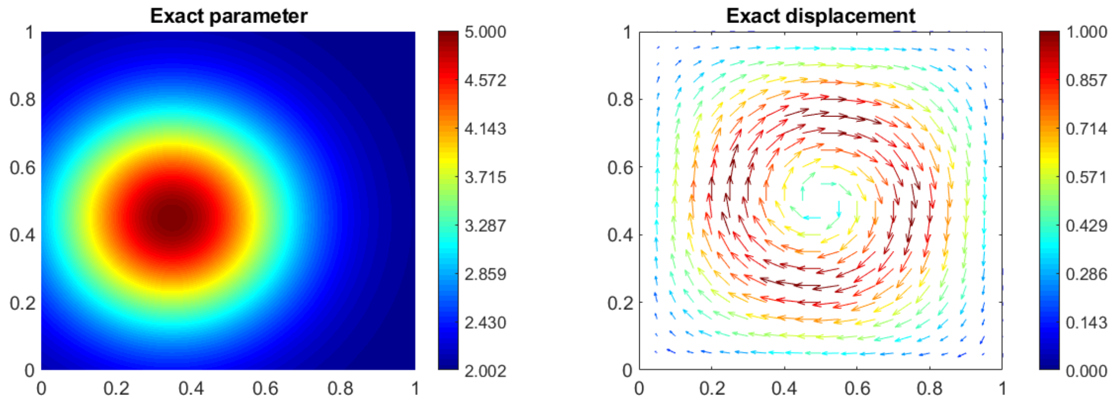
$$u_1(x,y) = Y_1(\omega)\sin(2\pi y)(-1+\cos(2\pi x)) + \frac{Y_2(\omega)}{1+\lambda}\sin(\pi x)\sin(\pi y) \tag{5.12}$$

$$u_2(x,y) = Y_1(\omega)\sin(2\pi x)(1-\cos(2\pi y)) + \frac{Y_2(\omega)}{1+\lambda}\sin(\pi x)\sin(\pi y) \tag{5.13}$$

with $\lambda = \dfrac{2\nu}{1-2\nu}$ where $\nu$ is the Poisson ratio (chosen to be close to 0.5 to indicate near incompressibility),

$$\mu(x,y) = 2 + 3\exp(-10(x-0.35)^2 - 10(y-0.45)^2),$$

and compute the corresponding force vector $\boldsymbol{f}$ using equation (1.1). Figure 5.2 shows the exact parameter $\mu$ and the mean of $\overline{\boldsymbol{u}}$. The spatial domain $D$ is $[0,1]^2$ and we have $r = 0$ (i.e. the parameter to be identified is deterministic). Random variables $Y_1(\omega)$ and $Y_2(\omega)$ are uniformly distributed on $[0,1]$.



(A) Exact parameter $\mu$

(B) Mean displacement vector

FIGURE 5.2: Parameter $\mu$, mean displacement of $\overline{\boldsymbol{u}}$ (i.e. $Y_1(\omega) = Y_2(\omega) = 0.5$).

Our initial approximation is the block vector $M^{(0)}$,

$$\boldsymbol{M}^{(0)} = \begin{bmatrix} \boldsymbol{M}_1 \\ \boldsymbol{M}_2 \\ \boldsymbol{M}_3 \end{bmatrix}$$

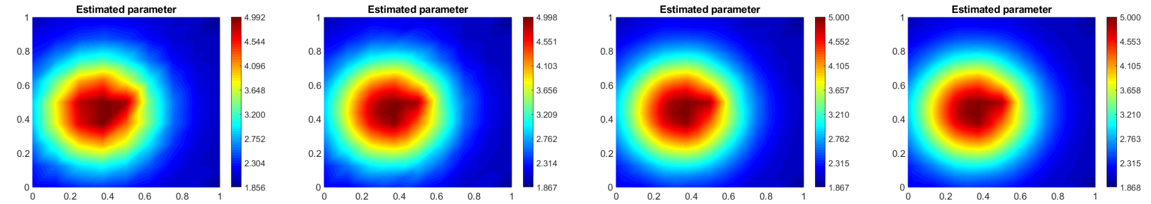
where the blocks $\boldsymbol{M}_1, \boldsymbol{M}_2, \boldsymbol{M}_3$ correspond to the initial approximations of $\mu_1(\boldsymbol{x})$, $\mu_2(\boldsymbol{x})$, and $\mu_3(\boldsymbol{x})$ respectively. At every iteration, the gradient of the objective functional at $\boldsymbol{M}^{(k)}$ is computed using $s_k$ realizations of the random variable $\omega$ and is then used in computing the next iterate $\boldsymbol{M}^{(k+1)}$. At iteration $k$, we generate a set of realization $\{\omega_j\}_{j=1}^{s_k}$ of the random variable, and we compute the step length $\alpha_k$. For the step lengths $\alpha_k$, we use the initial step length $\alpha_0$ provided at the beginning of the iterations and make sure that the sequence $\alpha_k$ satisfies

$$\sum_{n\in\mathbb{N}} \alpha_n = \infty, \quad \sum_{n\in\mathbb{N}} \alpha_n^2 < \infty.$$

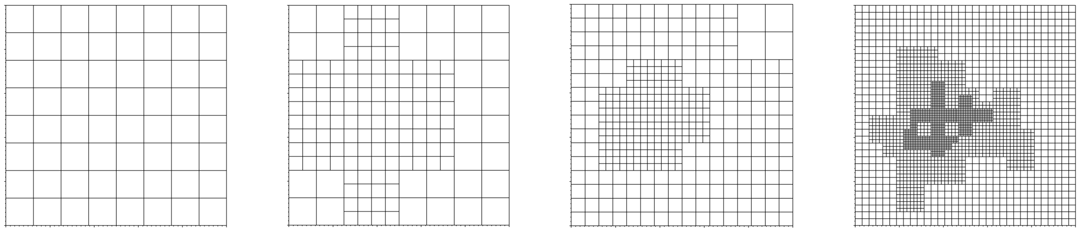
The projection to the admissible set is necessary so the stiffness parameter values stay between the two positive scalar bounds $\mu_0$ and $\mu_1$. Displacement data $\overline{\boldsymbol{Z}}$ needed in the gradient computation (5.9) are nodal interpolants of the analytical solution $\overline{\boldsymbol{u}}$ without any additional noise. The constraint set we choose in the projection step is $\{\boldsymbol{M} \in \mathbb{R}^m | \mu_0 \le \boldsymbol{M} \le \mu_1\}$ where $\mu_0$ and $\mu_1$ are constant upper and lower bound vectors. We choose $\nu = 0.49995$

which corresponds to a $\lambda$ value of 9999. In each mesh refinement cycle, 3% of the cells are marked for coarsening and 30% of the cells are marked for refinement, and the refinement process is completed in a way that any two neighboring cells in the mesh do not differ by more than one level of refinement. For simplicity, we used the same finite element mesh for both the displacement and the parameter, and an error indicator based on the parameter values only. In the stochastic approximation algorithm (see Section 5.4), we used the sample rate $s_k = 1$ and regularization parameter $\beta_k = 10^{-6}$ in every iteration. The algorithm is set to terminate when the $L^2$-norm of the gradient falls below $10^{-7}$.

Figure 5.3 shows the estimated parameter $\mu_h$ for three cycles of refinement (the initial mesh is uniform), along with the computational meshes which are refined adaptively every 1000 iterations.



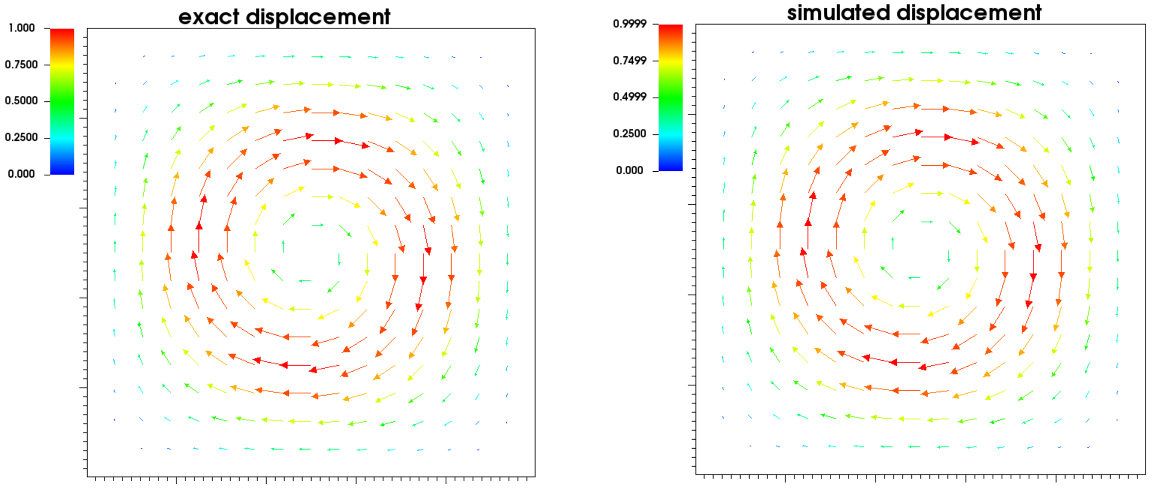
(a) cycle 0 (b) cycle 1 (c) cycle 2 (d) cycle 3



(e) cycle 0 (f) cycle 1 (g) cycle 2 (h) cycle 3

Figure 5.3: Estimated parameter $\mu_h$ for 3 refinement cycles and the computational meshes.

Figure 5.4 shows the simulated displacement vector $\overline{\boldsymbol{u}}_h(\mu_h)$ together with the exact displacement vector on a coarse ($16 \times 16$) mesh where $Y_1(\omega)$ and $Y_2(\omega)$ are set to their mean values (0.5 in each case, see formulas (5.12) and (5.13)). Table 5.1 shows the number of cells, degrees of freedom (dofs) for the parameter $\mu$, and minimum cell diameter for each refinement cycle of the mesh.



(A) Exact displacement $\overline{\boldsymbol{u}}(\mu)$ (B) Simulated displacement $\overline{\boldsymbol{u}}_h(\mu_h)$

FIGURE 5.4: Simulated and exact displacements for $Y_1(\omega) = Y_2(\omega) = 0.5$ on a coarse grid.

| cycle | # cells | # dofs ($\mu$) | minimum cell diameter |
|---|---|---|---|
| 0 | 64 | 81 | 0.1768 |
| 1 | 160 | 191 | 0.0884 |
| 2 | 400 | 449 | 0.0442 |
| 3 | 1024 | 1113 | 0.0221 |
| 4 | 2752 | 2937 | 0.0110 |
| 5 | 7492 | 7951 | 0.0055 |

TABLE 5.1: Number of cells in the mesh, degrees of freedom (dofs) for $\mu$, and minimum cell diameters for 5 refinement cycles.

# Conclusion

Here, we include discussions of the work completed so far and mention some thoughts on future directions for this work. All of our codes were written in C++ using the finite element library deal.ii and post-processing was completed using both MATLAB and VisIt. We studied a parameter identification problem in deterministic and stochastic linear elasticity systems that stem from the recovery of the tissue stiffness parameter from measurements of the displacement in the tissue under prescribed forces. In order to properly model the isotropic linear elasticity problem where $\lambda$ is large, a mixed formulation is necessary. A formulation which was derived by incorporating an additional equation with a new pressure variable to the existing linear elasticity system. A mixed finite element method with an adaptive mesh refinement framework within the proposed method provides the accuracy and resolution needed for the recovery of the tissue stiffness parameter.

Numerical experiments described in Chapter 2 show the accuracy and efficiency of the forward problem solver which is crucial when solving the inverse problem using iterative methods. We demonstrated the differences in displacement fields due to various body forces, boundary conditions, and locations of stiffer regions. In Chapter 5 we compared solutions of the forward problem from the deterministic model with those from the stochastic model by comparing the mean displacement field in the stochastic model with the displacement field in the deterministic model.

We have also explored variations of the stochastic gradient-descent method for tissue stiffness identification problems with a deterministic model, results of which were shown in Chapter 3. We demonstrated that our framework, which combines the stochastic gradient method with adaptive mesh refinement, is able to to estimate the tissue stiffness parameters accurately for both smoothly and sharply varying transitions. Future directions could include exploring and analyzing other variations of the stochastic gradient descent method for inverse problems with deterministic and stochastic models.

In Chapter 4 we demonstrated that the computational framework is able to handle a parameter identification problem where measurement (displacement) data is collected from experiments with tissue-mimicking phantoms in a lab. Ultrasound data such as this, however, tends to be noisy, so we also explored some data smoothing methods. Some quantitative results and analyses of the estimation error were presented as well in an initial study for the purpose of better understanding how the problem parameters and computational parameters affect the accuracy of the solver.

In Chapter 5 we studied the parameter identification problem in a stochastic linear elasticity system and set up a general framework for solving the elasticity inverse problem in a stochastic setting. Preliminary results of a larger study underway are presented in order to demonstrate the feasibility of the approach.

In terms of other future directions for this work, we would like to continue to examine the efficacy of our methods on more realistic data both through synthetically generated data from tissue-mimicking phantoms and through any available clinical data. In particular, a study of what boundary conditions yield the most physiologically realistic results would be useful. A systematic study of the interplay of various parameters involved in the algorithm such as the sequence of step lengths $\alpha_k$, stopping criteria, number of degrees of freedom of the computational mesh, the sequence of regularization parameters $\beta_k$, and their effects on the convergence of the algorithm, as well as the quality of the numerical solution, should also be undertaken.

# Bibliography

[1] Lorraine G. Olson, Robert D. Throne, Adam J. Nolte, Allison Crump, Kaelyn Griffin, Tianhong Han, Nicolae Iovanac, Thomas Janssen, Michael Jones, Xiaoyin Ling, and Michael Samp. An inverse problem approach to stiffness mapping for early detection of breast cancer: tissue phantom experiments. Inverse Probl. Sci. Eng., 27(7):1006–1037, 2019.

[2] Minhua Lu, Yanan Tang, Ruichao Sun, Tianfu Wang, Siping Chen, and Rui Mao. A real time displacement estimation algorithm for ultrasound elastography. Computers in Industry, 69:61–71, 2015.

[3] Sung Jae Kwon and Mok Kun Jeong. Advances in ultrasound elasticity imaging. Biomed Eng Lett, 7(2):71–79, 2017.

[4] Tomy Varghese. Quasi-static ultrasound elastography. Ultrasound Clin, 4(3):323–338, 2009.

[5] Isabell M. Franck and P. S. Koutsourelakis. Sparse variational Bayesian approximations for nonlinear inverse problems: applications in nonlinear elastography. Comput. Methods Appl. Mech. Eng., 299:215–244, 2016.

[6] Walaa Khaled, Helmut Ermert, Stefan Reichling, and Otto Bruhns. The inverse problem of elasticity: A reconstruction procedure to determine the shear modulus of tissue. Proceedings - IEEE Ultrasonics Symposium, 2005.

[7] M. M. Doyley. Model-based elastography: a survey of approaches to the inverse elasticity problem. Physics in Medicine and Biology, 57(3):35–73, 2012.

[8] M. M. Doyley and K. J. Parker. Elastography: general principles and clinical applications. Ultrasound clinics, 9(1):1–11, 2014.

[9] Paul E. Barbone and Assad A. Oberai. A review of the mathematical and computational foundations of biomechanical imaging. In Computational Modeling in Biomechanics, pages 375–408. Dordrecht: Springer Netherlands, 2010.

[10] Nathan D. Cahill, Baasansuren Jadamba, Akhtar A. Khan, Miguel Sama, and Brian C. Winkler. A first-order adjoint and a second-order hybrid method for an energy output least-squares elastography inverse problem of identifying tumor location. Bound. Value Probl., 2013:19, 2013. Id/No 263.

[11] K.R. Raghavan and A.E. Yagle. Forward and inverse problems in elasticity imaging of soft tissues. IEEE Transactions on Nuclear Science, 41(4):1639–1648, 1994.

[12] F. Kallel and M. Bertrand. Tissue elasticity reconstruction using linear perturbation method. IEEE Transactions on Medical Imaging, 15(3):299–313, 1996.

[13] Marvin Doyley, P. Meaney, and Jeffrey Bamber. Evaluation of an iterative reconstruction method for quantitative elasticity. Physics in medicine and biology, 45:1521–40, 2000.

[14] S. Goenezen, J. F. Dord, Z. Sink, P. E. Barbone, J. Jiang, T. J. Hall, and A. A. Oberai. Linear and nonlinear elastic modulus imaging: an application to breast cancer diagnosis. IEEE transactions on medical imaging, 31(8):1628–1637, 2012.

[15] M. Tyagi, S. Goenezen, P. E. Barbone, and A. A. Oberai. Algorithms for quantitative quasi-static elasticity imaging using force data. International journal for numerical methods in biomedical engineering, 30(12):1421–1436, 2014.

[16] Baasansuren Jadamba, Akhtar A. Khan, Assad A. Oberai, and Miguel Sama. First-order and second-order adjoint methods for parameter identification problems with an application to the elasticity imaging inverse problem. Inverse Probl. Sci. Eng., 25(12):1768–1787, 2017.

[17] Alexander Arnold, Stefan Reichling, Otto T Bruhns, and Jörn Mosler. Efficient computation of the elastography inverse problem by combining variational mesh adaption and a clustering technique. Phys Med Biol, 55(7):2035–56, 2010.

[18] Arbaz Khan, Catherine E. Powell, and David J. Silvester. Robust preconditioning for stochastic galerkin formulations of parameter-dependent nearly incompressible elasticity equations. SIAM Journal on Scientific Computing, 41(1):A402–A421, 2019.

[19] M. Chau and M. C. Fu. An overview of stochastic approximation. In Michael C. Fu, editor, Handbook of simulation optimization, volume 216. New York, NY: Springer, 2015.

[20] Caroline Geiersbach and Georg Ch. Pflug. Projected stochastic gradients for convex constrained problems in Hilbert spaces. SIAM J. Optim., 29(3):2079–2099, 2019.

[21] M. Martin, S. Krumschield, and F. Nobile. Analysis of stochastic gradient methods for pde-constrained optimal control problems with uncertain parameters. Preprint, pages 1–39, 2018.

[22] Rachel Hawks, Baasansuren Jadamba, Akhtar A. Khan, Miguel Sama, and Yidan Yang. A variational inequality based stochastic approximation for inverse problems in stochastic partial differential equations. In Nonlinear analysis and global optimization, pages 207–226. Cham: Springer, 2021.

[23] Baasansuren Jadamba, Akhtar A. Khan, Miguel Sama, Hans-Jorg Starkloff, and Christiane Tammer. A convex optimization framework for the inverse problem of identifying a random parameter in a stochastic partial differential equation. SIAM/ASA J. Uncertain. Quantif., 9:922–952, 2021.

[24] Daniel Arndt, Wolfgang Bangerth, Marco Feder, Marc Fehling, Rene Gassmöller, Timo Heister, Luca Heltai, Martin Kronbichler, Matthias Maier, Peter Munch, Jean-Paul Pelteret, Simon Sticko, Bruno Turcksin, and David Wells. The `deal.II` library, version 9.4. Journal of Numerical Mathematics, 30(3):231–246, 2022.

[25] Arbaz Khan, Alex Bespalov, Catherine E. Powell, and David J. Silvester. Robust a posteriori error estimation for parameter-dependent linear elasticity equations. Math. Comput., 90(328):613–636, 2021.

[26] Alexandre Ern and Jean-Luc Guermond. Theory and practice of finite elements., volume 159 of Appl. Math. Sci. New York, NY: Springer, 2004.

[27] George Papamakarios. Comparison of modern stochastic optimization algorithms. University of Edinburgh, Edinburgh, 2014.

[28] Ali K.Z. Tehrani and Hassan Rivaz. Displacement estimation in ultrasound elastography using pyramidal convolutional neural network. IEEE Trans. Ultrason., Ferroelect., Freq. Contr., 67(12):2629–2639, 2020.

[29] Hassan Rivaz, Emad M. Boctor, Michael A. Choti, and Gregory D. Hager. Real-time regularized ultrasound elastography. IEEE Trans. Med. Imaging, 30(4):928–945, 2011.

[30] Itay Hen, Anat Sakov, Neri Kafkafi, Ilan Golani, and Yoav Benjamini. The dynamics of spatial behavior: how can robust smoothing techniques help? Journal of Neuroscience Methods, 133(1-2):161–172, 2004.

[31] Chong Gu. Smoothing noisy data via regularization: statistical perspectives. Inverse Probl., 24(3):20, 2008. Id/No 034002.

[32] William Menke and Zachary Eilon. Relationship between data smoothing and the regularization of inverse problems. Pure Appl. Geophys., 172(10):2711–2726, 2015.

[33] U. Albocher, P. E. Barbone, M. S. Richards, A. A. Oberai, and I. Harari. Approaches to accommodate noisy data in the direct solution of inverse problems in incompressible plane strain elasticity. Inverse Probl. Sci. Eng., 22(8):1307–1328, 2014.

[34] Alex Araujo, Chris Constantinou, and Joao Tavares. Relationship between data smoothing and the regularization of inverse problems. Pure Appl. Geophys., 172(10):2711–2726, 2015.

[35] Gabriel J. Lord, Catherine E. Powell, and Tony Shardlow. An introduction to computational stochastic PDEs. Camb. Texts Appl. Math. Cambridge: Cambridge University Press, 2014.

[36] B. Jadamba, A. A. Khan, G. Rus, M. Sama, and B. Winkler. A new convex inversion framework for parameter identification in saddle point problems with an application to the elasticity imaging inverse problem of predicting tumor location. SIAM J. Appl. Math., 74(5):1486–1510, 2014.

[37] Baasansuren Jadamba, Akhtar A. Khan, Michael Richards, and Miguel Sama. A convex inversion framework for identifying parameters in saddle point problems with applications to inverse incompressible elasticity. Inverse Probl., 36(7):25, 2020. Id/No 074003.

[38] Baasansuren Jadamba, Akhtar A. Khan, Miguel Sama, and Yidan Yang. An iteratively regularized stochastic gradient method for estimating a random parameter in a stochastic PDE. A variational inequality approach. J. Nonlinear Var. Anal., 5(6):865–880, 2021.