9-21-2015

Heavy Ball with Friction Method for the Elastography Inverse Problem

Oladayo Eluyefa
ome3711@rit.edu

Heavy Ball with Friction Method for the Elastography Inverse Problem

By

Oladayo Eluyefa

A thesis submitted in partial fulfillment of
the requirements for the degree of Master of Science
in Applied and Computational Mathematics
from the School of Mathematical Sciences
College of Science

Rochester Institute of Technology

September 21, 2015

Advisor:	Dr. Akhtar Khan
Co-Advisor:	Dr. Baasansuren Jadamba
Committee members:	Dr. Patricia Clark
	Dr. Bonnie Jacob
Graduate Director:	Dr. Elizabeth Cherry

R·I·T
College of
SCIENCE

Committee Approval:

Akhtar A. Khan — Date
School of Mathematical Sciences
Thesis Advisor

Baasansuren Jadamba — Date
School of Mathematical Sciences
Thesis Co-advisor

Patricia Clark — Date
School of Mathematical Sciences
Committee Member

Bonnie Jacob — Date
Science and Mathematics Department, NTID
Committee Member

Elizabeth Cherry — Date
School of Mathematical Sciences
Director of Graduate Programs

Abstract

The primary objective of this thesis is to develop a fast and efficient computational framework for the nonlinear inverse problem of identifying a variable coefficient in a system of partial differential equation modelling the response of an incompressible elastic object under some known body forces and boundary traction. The main novelty of this contribution is to use, for the first time, of the so-called heavy ball with friction method for inverse problems. The heavy ball with friction dynamical system is a nonlinear oscillator with damping. The key idea is to pose the inverse problem as an optimization problem, derive its optimality system, and then seek the solution through a trajectory of a dynamical system. In this work, we will study four different optimization formulations for the nonlinear inverse problem and thoroughly compare their convergence and numerical performance. Since we use a second-order method, we also investigate a general second-order hybrid and a second-order adjoint method for an efficient computation of the hessian of the output least-squares formulation. The stability of the dynamical system approach with respect to the contamination in the data is thoroughly investigate in the context of a simpler elliptic partial differential equation. The mixed finite element approach is used to discretize the direct as well as the inverse problems.

Keywords: Inverse problems, parameter identification, heavy ball with friction method, optimization, differential equations, dynamical systems.

Dedication

I would like to dedicate this to my family for supporting me unfailingly throughout my studies.

-I love you

Acknowledgement

I would like to thank my family, friends, and mentors for helping me through this learning process, your support was vital to my success.
I am especially grateful to my advisors, Dr. Akhtar Khan and Dr. Baasansuren Jadamba. They have been extremely supportive of me throughout my time in school even before our research began. Their guidance and support has helped me grow and learn valuable skills not only in mathematics, but also in life.
I would also like to thank the School of Mathematical Sciences within the College of Science at RIT for their help throughout my time here. I would like to thank, in particular, Anna Fiorucci, and Carrie Koneski who have been instrumental throughout this journey.
Most importantly, I would like to thank my parents for their encouragement and support. Without them, none of this would be possible. I love my family very much.
Special thanks to my committee members: Prof. Patricia Clark and Dr. Bonnie Jacob, and the former, and current director of graduate programs, Dr. Nathan Cahill, and Dr. Elizabeth Cherry.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter introduces the concept of an inverse problem of parameter identification. It also provides motivation for the research in the field of inverse problems. A literature review is performed on the ideas and concepts leading up to this work. Finally, it states the structure and objectives of the thesis work.

1.1 Motivation

Cancer is becoming one of fastest growing causes of death in the world [39]. Soft tissue cancer is but one form and it is as deadly as any other. The large number of deaths due to soft tissue cancer can be managed through early detection of the tumorous cells within a tissue that could be potentially cancerous. However, the means of identification of these tumorous cells are somewhat lacking. Palpation and ultrasound are some of the procedures currently used in the identification of such cells. Ultrasound relies on the acoustic behavior of tissue based on its physical properties, which is in turn determined by its health. Ultrasound detect differences in tissue stiffness by passing sound waves through the tissue and using the response times as an indication of this property [29]. Palpation is a procedure in which force is manually applied to a region of concern, and differences in the structure of healthy versus unhealthy cells are used to determine if an area has harder "lumps" than its surrounding tissue [19]. A drawback of palpation is that it can only be used near the surface of the skin. Also, deciding which cell regions are classified as harder than others is very subjective.
Recently, elastic imaging has been considered in early tumor identification [1]. Force is applied to the tissue in question, and the axial displacement field of the tissue is retrieved. This displacement field is used to determine the elasticity throughout the tissue, and thus the location for a potentially cancerous tumor.

1.2 The Inverse Problem

An inverse problem arises from an underlying direct problem. Consider a mathematical system that models the heat transfer from a heat source to a metallic body. In this system, we have a heat source, the temperature of the metallic body as a function of position, and the thermal conductivity of the body. In the direct problem, the thermal conductivity and the heat source are the known terms, and the output of the direct problem is the temperature of the body. One possible inverse problem would be to calculate the amount of heat produced from the heat source, provided we know the thermal conductivity of the body, and we have a means of measuring the temperature of the body. This is known as the inverse problem of source identification. The other kind of inverse problem would involve identifying the thermal conductivity of the body given the heat source, and a measurement of the temperature of the body. In most mathematical systems, innate properties such as the thermal conductivity of a body are represented by parameters within the model. As a result, the latter problem is referred to as the inverse problem of parameter identification. Broadly, the inverse problem is a way of non intrusively calculating some innate characteristic of a material within a system that could otherwise not be measured. In the cancer problem, using elastic imaging, the elasticity is the innate characteristic of the tissue that is to be calculated. Some amount of force is applied, which is the source, and the output, is the displacement, which can be measured or observed. Other examples of inverse problems are studied in [12, 22, 23, 25, 32, 36, 46].

1.3 Literature Review

There has been a lot of work leading up to the ideas explored in this thesis both in the field of elasticity imaging with respect to inverse problems, and the use of dynamical systems in optimization problems. This review is in no way exhaustive, but it highlights a few works that have led up to the concepts used directly and extensively in this thesis.

Ultrasound based methods have been extensively used over time when it comes to tumor identification and elasticity imaging. Bertrand[14] and Parker et al.[40], both used ultrasound methods to try to attain the elasticity modulus of a tissue. Bamber at al. [11] used ultrasound techniques to image the compressed breast in order to attain axial displacements. These methods were somewhat idealized as they assumed that both the location, and geometry of the tumor were known at the start of the process.

The idea of using an inverse problem to recover the difference in the strains of a healthy versus an unhealthy tissue was brought forth by Raghavan and Yagle[41]. They applied finite differences when solving the problem. Finite element methods were used by Kallel and Bertrand[31] when they tried to fit the axial displacement of the compressed tissue gotten from ultrasound imaging, in a least squares sense. Doyley et al.[21] used iterative schemes to retrieve the Young's modulus or elasticity modulus. Such work can be see in [27, 33] as well.

In 2003, Oberai et al.[38] studied the identification of the shear modulus in an incompressible elastic material.

More recently, Arnold et al.[7] developed numerical methods for identification of the elasticity modulus. In the same year, Ammari et al.[5] used optimization approaches to tackle the same problem. Doyley has a very extensive survey article on these topics[19]. More can be found in [20, 18, 13, 37, 47]

In the field of continuous methods and its applications to optimization, Bostaris[15] introduced broader curvilinear search paths that used the eigen property of the Hessian matrix in function minimization. Bartholomew-Biggs and Brown[17, 16] studied trajectories based on a system of differential equations to solve an equality constraint optimization problem. Schäffler[42] considered a gradient trajectory method for optimization and fifteen years later Liao[35] considered a gradient based continuous method.

Antipin[6] did work with convex programming problems using continuous gradient projections of both the first and second order. Glazos et al.[24] presented a family of dynamical systems whose steady state solution solves a convex optimization problem. Alvarez and Pérez[4] did work on convex minimization problems using newton type continuous methods. Alvarez[2] then went on to study the dissipative dynamical system and gave results in a Hilbert space setting. He studied the asymptotic behavior of the solution to a particular dynamical system when its convex potential is bounded, and gave conditions for the convergence of the solution to a minimizer of the potential.

Attouch et al.[10] worked on dynamical systems that incorporate the gradient of the functional being minimized into the dissipative dynamical system to yield a solution. They discussed the convergence analysis of the so-called heavy ball with friction problem. They presented conditions for convergence placing constraints on certain terms in the system. Shi[43] developed a multistep method for solving the unconstrained minimization problem which ensured stability of convergence, and linear rate of convergence under specified conditions.

Liao et al.[34] used a gradient based continuous method to solve the minimization problem. In [35], Liao used a continuous method for convex programming problems in which he converted the problems into variational inequalities.

Zhang et al.[48] proposed the continuous Newton type method for unconstrained optimization which is implemented in [44] in the one dimensional case. Attouch and Alvarez[8] use the second order dissipative dynamical system - the heavy ball with friction method - to solve the same unconstrained optimization problem. Jules and Mainge[30] compared the standard proximal point algorithm to the implicit discretization of the dissipative dynamical system by considering a co-coercive operator. The idea that certain parameters in the heavy ball with friction system have large effect on the convergence speed to a minimizer was extensively studied separately in [30] and [10].

1.4 Objectives and Structure

The main goals of this thesis work is to solve the inverse problem of parameter identification using differential equation approaches studied in [45, 48]. It also looks to consider possible extension discussed in [2, 4, 3, 9] with the heavy ball with friction method [2]. Different objective functionals are considered for the optimization scheme and the results are reported.
This thesis is structured as follows: First, we introduce the elastography inverse problem, and we discretize it using mixed finite element methods. We also propose optimization schemes that will be applied in order to solve the inverse problem. We introduce the novel computations of the second order derivative of the output least squares functional. In chapter 3, we discretize the functionals presented in the previous chapter. We introduce the heavy ball with friction method in chapter 4. We discuss other continuous methods, and how we intend to use them in our optimization problem. Chapter 5 reports the results of testing both the new second order derivative computations for the output least squares functional, and the heavy ball with friction method on the elastorgraphy problem. Chapter 6 considers a simpler elliptic partial differential equation. For completeness, we retest the heavy ball with friction method, using different differential equation solvers. Finally, we conduct a noise study to compare both the differential equation techniques and the optimization schemes applied.

Chapter 2

Various Optimization Schemes

In this chapter, we introduce the system of partial differential equations that model the elasticity system. We use mixed finite element methods to derive the weak form of the system, and then formulate the optimization functionals that define the schemes to be used. We also present and derive the novel computation of the second order derivative of the Output Least Squares functional.

2.1 Saddle Point Formulation

Given the domain Ω as a subset of $\mathbb{R}^2$ or $\mathbb{R}^3$ and $\partial\Omega = \Gamma_1 \cup \Gamma_2$ as its boundary, the following system models the response of an isotropic elastic body to the known body forces and boundary traction:

$$-\nabla \cdot \sigma = f \text{ in } \Omega, \tag{2.1a}$$

$$\sigma = 2\mu\epsilon(u) + \lambda \mathrm{div} u\, I, \tag{2.1b}$$

$$u = g \text{ on } \Gamma_1, \tag{2.1c}$$

$$\sigma n = h \text{ on } \Gamma_2. \tag{2.1d}$$

In (2.1), the vector-valued function $u = u(x)$ is the displacement of the elastic body, f is the applied body force, n is the unit outward normal, and $\epsilon(u) = \frac{1}{2}(\nabla u + \nabla u^{\mathrm{T}})$ is the linearized strain tensor. The resulting stress tensor σ in the stress-strain law (2.1b) is obtained under the condition that the elastic body is isotropic and the displacement is sufficiently small so that a linear relationship remains valid. Here μ and λ are the Lamé parameters which quantify the elastic properties of the object.
In this work, our primary objective is to develop a computational framework for the elastography inverse problem of locating soft inclusions in an incompressible object, for example, cancerous tumor in the human body. From a mathematical stand point this inverse problem seeks μ from a measurement

of the displacement vector u under the assumption that the parameter λ is very large. The key idea behind the elatography inverse problem is that the stiffness of soft tissue can vary significantly based on its molecular makeup and varying macroscopic/microscopic structure (see [19]) and such changes in stiffness are related to changes in tissue health. In other words, the elastography inverse problem mathematically mimics the practice of palpation by making use of the differing elastic properties of healthy and unhealthy tissue to identify tumors. In most of the existing literature on elastography inverse problem, the human body is modelled as an incompressible elastic object. Although this assumption simplifies the identification process as there is only one parameter μ to identify, it significantly complicates the computational process as the classical finite element methods become quite ineffective due to the so-called locking effect. One of the few remedies of this situation is by resorting to mixed finite element formulation. We explain this in the following. For the time being, in (2.1), we set $g = 0$. For this case, the space of test functions, denoted by V, is given by:

$$V = \{\bar{v} \in H^1(\Omega) \times H^1(\Omega) : \ \bar{v} = 0 \text{ on } \Gamma_1\}.$$

By using the Green's identity and the boundary conditions (2.1c) and (2.1d), we obtain the following weak form of the elasticity system (2.1): Find $\bar{u} \in V$ such that

$$\int_\Omega 2\mu\epsilon(\bar{u})\cdot\epsilon(\bar{v}) + \int_\Omega \lambda(\operatorname{div}\bar{u})(\operatorname{div}\bar{v}) = \int_\Omega f\bar{v} + \int_{\Gamma_2} \bar{v}h, \quad \text{for every } \bar{v} \in V. \tag{2.2}$$

The mixed finite elements approach then consists of introducing a pressure term $p \in Q = L^2(\Omega)$

$$p = \lambda(\operatorname{div}\bar{u}), \tag{2.3}$$

or equivalently,

$$\int_\Omega (\operatorname{div}\bar{u})q - \int_\Omega \frac{1}{\lambda}pq = 0, \quad \text{for every } q \in Q. \tag{2.4}$$

By using relation (2.3), the weak form (2.2) reads: Find $\bar{u} \in V$ such that

$$\int_\Omega 2\mu\epsilon(\bar{u}) \cdot \epsilon(\bar{v}) + \int_\Omega p(\operatorname{div}\bar{v}) = \int_\Omega f\bar{v} + \int_{\Gamma_2} \bar{v}h, \quad \text{for every } \bar{v} \in V. \tag{2.5}$$

In other words, the problem of finding $\bar{u} \in V$ satisfying (2.2) has now been reformulated as the problem of finding $(\bar{u}, p) \in V \times Q$ satisfying the mixed variational problems (2.4) and (2.5).

In the following we set $B = L_\infty$. Let $A \subset B$ be the set of all feasible coefficients which we assume to be nonempty, closed, and convex. Equations (2.4) and (2.5) can be written as follows:

$$a(\ell, u, v) + b(v, p) \quad = m(v) \quad \forall v \in V \tag{2.6a}$$
$$b(u, q) - c(p, q) \qquad = 0 \quad \forall q \in Q \tag{2.6b}$$

where

$$\begin{aligned} a(\mu, u, v) &= \int_\Omega 2\mu\epsilon(u)\cdot\epsilon(v), \\ b(v, p) &= \int_\Omega p(\mathrm{div} v), \\ b(u, q) &= \int_\Omega (\mathrm{div} u)q, \\ c(p, q) &= \int_\Omega \frac{1}{\lambda} pq, \\ m(v) &= \int_\Omega fv + \int_{\Gamma_2} vh. \end{aligned} \tag{2.7}$$

It is easy to verify that $a : B \times V \times V \to \mathbb{R}$ is trilinear and symmetric with respect to its last two arguments, $b : V \times Q \to \mathbb{R}$ is bilinear, $c : Q \times Q \to \mathbb{R}$ is symmetric and bilinear, and $m : V \to \mathbb{R}$ is a linear continuous map.
We assume that constants, $k_0, k_1, k_2, c_1, c_2 > 0$, exist such that

$$\begin{aligned} a(\ell, v, v) &\geq k_1||v||^2, \\ |a(\ell, u, v)| &\leq k_2||l||||u||||v||, \\ c(q, q) &\geq c_1||q||^2, \\ |c(p, q)| &\leq c_2||p||||q||, \\ |b(v, q)| &\leq k_0||v||||q||, \end{aligned} \tag{2.8}$$

for every $\ell \in A$, $u, v \in V$, $p, q \in Q$. It is important to note that, in the following sections, since the pressure term p is also unknown, we would be stating the weak form in such a way that we look to find $\bar{u} = (u, p)$ for the saddle point problem.

2.2 Output Least Squares

Optimizing the Output Least Squares (OLS) functional is the most common approach to solving inverse problems. It employs the simple idea of minimizing the norm between the solution to the weak form, $\bar{u}$, and some measurement of this solution, $z = (\bar{z}, \hat{z})$.

$$J_{\text{OLS}}(\ell) = \frac{1}{2}||u(\ell) - \bar{z}||_V^2 + \frac{1}{2}||p(\ell) - \hat{z}||_Q^2$$

where $\hat{V} = V \times Q$, for the saddle point problem (SPP).
Inverse problems are highly ill-posed posed, meaning that the existence, uniqueness, and stability of a solution cannot be guaranteed. As a result, regularization is employed in order to attain a stable version. This ultimately yields the following optimization problem:

$$\min_{\ell \in A} J_{\text{OLS}}(\ell) = \frac{1}{2}||\bar{u}(\ell) - z||_{\hat{V}}^2 + \kappa R(\ell) \tag{2.9}$$

where R is the regularization functional, and $\kappa > 0$ is the regularization parameter.
Now that we have derived our OLS functional, we look to compute the first order and second order derivatives of the functional. We compute the first order derivative using the first order adjoint method, and we compute the second order derivative using the second order adjoint method, and the hybrid method.

2.2.1 First Order Derivative

Here, we use the first order adjoint method for the computation of the first derivative of the regularized OLS functional. Recall, the OLS funtional

$$J_{\text{OLS}}(\ell) = \frac{1}{2}||u(\ell) - \bar{z}||_V^2 + \frac{1}{2}||p(\ell) - \hat{z}||_Q^2 + \kappa R(\ell).$$

This functional can be written in terms of the inner products of the spaces in which they exist to get

$$J_{\text{OLS}}(\ell) = \frac{1}{2}\langle u(\ell) - \bar{z}, u(\ell) - \bar{z}\rangle + \frac{1}{2}\langle p(\ell) - \hat{z}, p(\ell) - \hat{z}\rangle + \kappa R(\ell)$$

By using the chain rule, the derivative of J_{OLS} at $\ell \in A$ in the direction $\delta\ell \in A$ is given by

$$DJ_{\text{OLS}}(\ell)(\delta\ell) = \langle Du(\ell)(\delta\ell), u(\ell) - \bar{z}\rangle + \langle Dp(\ell)(\delta\ell), p(\ell) - \hat{z}\rangle + \kappa DR(\ell)(\delta\ell),$$

where $D\bar{u}(\ell)(\delta\ell) = (Du(\ell)(\delta\ell), Dp(\ell)(\delta\ell))$ is the derivative of the coefficient-to-solution map $\bar{u}$ and $DR(\ell)(\delta\ell)$ is the derivative of the regularizer R, both computed at ℓ in the direction $\delta\ell$.
For an arbitrary $\bar{v} = (v, q) \in \hat{V}$, we define the functional $\tilde{J}_{\text{OLS}} : B \times \hat{V} \to \mathbb{R}$ by

$$\tilde{J}_{\text{OLS}}(\ell, \bar{v}) = J_{\text{OLS}}(\ell) + a(\ell, u, v) + b(v, p) + b(u, q) - c(p, q) - m(v).$$

Since $\bar{u}(\ell) = (u(\ell), p(\ell))$ is the solution of saddle point problem (2.6), we have that

$$\tilde{J}_{\text{OLS}}(\ell, \bar{v}) = J_{\text{OLS}}(\ell), \ \forall \bar{v} \in \hat{V}.$$

Consequently, for every $\bar{v} \in \hat{V}$, the following identity holds

$$\frac{\partial \tilde{J}_{\text{OLS}}}{\partial \ell}(\ell, \bar{v})\,(\delta\ell) = DJ_{\text{OLS}}(\ell)\,(\delta\ell)\,, \quad \text{for every } \delta\ell \in A. \tag{2.10}$$

The adjoint method is used to avoid the direct computation of $\delta\bar{u} = D\bar{u}(\ell)(\delta\ell)$, by choosing $\bar{v} \in \hat{V}$ appropriately.

$$\begin{aligned}\frac{\partial \tilde{J}_{\text{OLS}}}{\partial \ell}(\ell, \bar{v})\,(\delta\ell) = {} & \langle Du(\ell)(\delta\ell), u - \bar{z}\rangle + \langle Dp(\ell)(\delta\ell), p - \hat{z}\rangle \\ & + \kappa DR(\ell)(\delta\ell) + a(\delta\ell, u, v) + a(\ell, Du(\ell)(\delta\ell), v) \\ & + b(v, Dp(\ell)(\delta\ell)) + b(Du(\ell)(\delta\ell), q) - c(Dp(\ell)(\delta\ell), q).\end{aligned} \tag{2.11}$$

For $\ell \in A$, let $w(\ell) = (\bar{w}(\ell), p_w(\ell))$ be the unique solution of the saddle point problem

$$a(\ell, \bar{w}, v) + b(v, p_w) = \langle \bar{z} - u, v\rangle\,, \quad \text{for every } v \in V, \tag{2.12a}$$
$$b(\bar{w}, q) - c(p_w, q) = \langle \hat{z} - p, q\rangle\,, \quad \text{for every } q \in Q, \tag{2.12b}$$

where the right-hand sides of (2.12a) and (2.12b) include the solution (u, p) of (2.6) for the given parameter, ℓ, and data, $(\bar{z}, \hat{z})$.
By plugging $\bar{v} = (\bar{w}, p_w)$ in (2.11), we obtain

$$\begin{aligned}\frac{\partial \tilde{J}_{\text{OLS}}}{\partial \ell}(\ell, w)\,(\delta\ell) = {} & \langle Du(\ell)(\delta\ell), u - \bar{z}\rangle + \langle Dp(\ell)(\delta\ell), p - \hat{z}\rangle \\ & + \kappa DR(\ell)(\delta\ell) + a(\delta\ell, u, \bar{w}) + a(\ell, Du(\ell)(\delta\ell), \bar{w}) \\ & + b(\bar{w}, Dp(\ell)(\delta\ell)) + b(Du(\ell)(\delta\ell), p_w) - c(Dp(\ell)(\delta\ell), p_w) \\ = {} & \kappa DR(\ell)(\delta\ell) + a(\delta\ell, u, \bar{w}),\end{aligned}$$

where we used the symmetry of a and c and the fact that w satisfies (2.12). Therefore, using (2.10), we obtain the following formula for the first-order derivative of J_{OLS}:

$$DJ_{\text{OLS}}(\ell)\,(\delta\ell) = \kappa DR(\ell)(\delta\ell) + a(\delta\ell, u, \bar{w}). \tag{2.13}$$

Therefore, in order to compute the derivative $DJ_{\text{OLS}}(\ell)\,(\delta\ell)$ of the OLS functional, given $\ell, \delta\ell \in A$, we first compute $\bar{u}(\ell) = (u(\ell), p(\ell))$ using (2.6). Next, we find $w(\ell) = (\bar{w}(\ell), p_w(\ell))$ by solving the system in (2.12), and finally we evaluate $DJ_{\text{OLS}}(\ell)\,(\delta\ell)$ by using (2.13).

2.2.2 Second Order Derivative

We compute the second order derivative of the OLS functional using two different methods: the Hybrid method and the second-order Adjoint method.

2.2.2.1 Hybrid Method for Second Derivative Computation

We aim to compute the second order derivative of the regularized OLS functional in such a way that we can avoid the computation of the second order derivative of the solution map $\bar{u}$. This method is referred to as the hybrid method because the derivative, $\delta\bar{u}$ is computed using a direct method, and the computation of the second derivative $\delta^2\bar{u}$ is avoided using an adjoint type method.
The hybrid method is based on the following result;
For each ℓ in the interior of A, $\bar{u} = \bar{u}(\ell) = (u(\ell), p(\ell))$ is infinitely differentiable at ℓ. The first derivative $\delta\bar{u} = (\delta u, \delta p) = (Du(\ell)\delta\ell, Dp(\ell)\delta\ell)$ is the unique solution of the saddle point problem:

$$
\begin{aligned}
a(\ell, \delta u, v) + b(v, \delta p) &= -a(\delta\ell, u, v), \quad \text{for every } v \in V, && \text{(2.14a)}\\
b(\delta u, q) - c(\delta p, q) &= 0, \quad \text{for every } q \in Q. && \text{(2.14b)}
\end{aligned}
$$

Let $\delta\ell_2 \in A$ be a fixed direction. For any $\bar{v} = (v, q) \in \hat{V}$, we define

$$
\begin{aligned}
H(\ell, \bar{v}) &= DJ_{\text{OLS}}(\ell)(\delta\ell_2) + a(\ell, Du(\ell)(\delta\ell_2), v) + b(v, Dp(\ell)(\delta\ell_2))\\
&\quad + b(Du(\ell)(\delta\ell_2), q) - c(Dp(\ell)(\delta\ell_2), q) + a(\delta\ell_2, u, v)\\
&= \langle Du(\ell)(\delta\ell_2), u - \bar{z}\rangle + \langle Dp(\ell)(\delta\ell_2), p - \hat{z}\rangle + \kappa DR(\ell)(\delta\ell_2)\\
&\quad + a(\ell, Du(\ell)(\delta\ell_2), v) + b(v, Dp(\ell)(\delta\ell_2)) + b(Du(\ell)(\delta\ell_2), q)\\
&\quad - c(Dp(\ell)(\delta\ell_2), q) + a(\delta\ell_2, u, v).
\end{aligned}
$$

Thus, for every $\bar{v} \in \hat{V}$, we have

$$
\frac{\partial H}{\partial \ell}(\ell, \bar{v})(\delta\ell_1) = D^2 J_{\text{OLS}}(\ell)(\delta\ell_1, \delta\ell_2), \quad \text{for every } \delta\ell_1 \in A. \tag{2.15}
$$

Computing this derivative of H in the direction $\delta\ell_1$ directly, we have

$$
\begin{aligned}
\frac{\partial H}{\partial \ell}(\ell, \bar{v})(\delta\ell_1) &= \left\langle D^2u(\ell)(\delta\ell_1, \delta\ell_2), u - \bar{z}\right\rangle + \langle Du(\ell)(\delta\ell_2), Du(\ell)(\delta\ell_1)\rangle\\
&\quad + \left\langle D^2p(\ell)(\delta\ell_1, \delta\ell_2), p - \hat{z}\right\rangle + \langle Dp(\ell)(\delta\ell_2), Dp(\ell)(\delta\ell_1)\rangle\\
&\quad + \kappa D^2R(\ell)(\delta\ell_1, \delta\ell_2) + a(\delta\ell_1, Du(\ell)(\delta\ell_2), v)\\
&\quad + a(\ell, D^2u(\ell)(\delta\ell_1, \delta\ell_2), v) + b(v, D^2p(\ell)(\delta\ell_1, \delta\ell_2))\\
&\quad + b(D^2u(\ell)(\delta\ell_1, \delta\ell_2), q) - c(D^2p(\ell)(\delta\ell_1, \delta\ell_2), q)\\
&\quad + a(\delta\ell_2, Du(\ell)(\delta\ell_1), v).
\end{aligned} \tag{2.16}
$$

Let $w(\ell) = (\bar{w}(\ell), p_w(\ell))$ be the solution of (2.12), that is,

$$\begin{aligned} a(\ell, \bar{w}, v) + b(v, p_w) &= \langle \bar{z} - u, v \rangle\,, \quad \text{for every } v \in V \\ b(\bar{w}, q) - c(\ell, p_w, q) &= \langle \hat{z} - p, q \rangle\,, \quad \text{for every } q \in Q. \end{aligned}$$

Now make the substitution $\bar{v} = w$ in (2.16), to obtain

$$\begin{aligned} \frac{\partial H}{\partial \ell}(\ell, w)(\delta\ell_1) = {} & \left\langle D^2 u(\ell)(\delta\ell_1, \delta\ell_2), u - \bar{z} \right\rangle + \langle Du(\ell)(\delta\ell_2), Du(\ell)(\delta\ell_1) \rangle \\ & + \left\langle D^2 p(\ell)(\delta\ell_1, \delta\ell_2), p - \hat{z} \right\rangle + \langle Dp(\ell)(\delta\ell_2), Dp(\ell)(\delta\ell_1) \rangle \\ & + \kappa D^2 R(\ell)(\delta\ell_1, \delta\ell_2) + a(\delta\ell_1, Du(\ell)(\delta\ell_2), \bar{w}) \\ & + a(\ell, D^2 u(\ell)(\delta\ell_1, \delta\ell_2), \bar{w}) + b(\bar{w}, D^2 p(\ell)(\delta\ell_1, \delta\ell_2)) \\ & + b(D^2 u(\ell)(\delta\ell_1, \delta\ell_2), p_w) - c(D^2 p(\ell)(\delta\ell_1, \delta\ell_2), p_w) \\ & + a(\delta\ell_2, Du(\ell)(\delta\ell_1), \bar{w}) \\ = {} & \kappa D^2 R(\ell)(\delta\ell_1, \delta\ell_2) + \langle Du(\ell)(\delta\ell_2), Du(\ell)(\delta\ell_1) \rangle \\ & + \langle Dp(\ell)(\delta\ell_2), Dp(\ell)(\delta\ell_1) \rangle + a(\delta\ell_1, Du(\ell)(\delta\ell_2), \bar{w}) \\ & + a(\delta\ell_2, Du(\ell)(\delta\ell_1), \bar{w}). \end{aligned}$$

Therefore from (2.15)

$$\begin{aligned} D^2 J_{\text{OLS}}(\ell)(\delta\ell_1, \delta\ell_2) = {} & \kappa D^2 R(\ell)(\delta\ell_1, \delta\ell_2) + \langle Du(\ell)(\delta\ell_2), Du(\ell)(\delta\ell_1) \rangle \\ & + \langle Dp(\ell)(\delta\ell_2), Dp(\ell)(\delta\ell_1) \rangle + a(\delta\ell_1, Du(\ell)(\delta\ell_2), \bar{w}) \\ & + a(\delta\ell_2, Du(\ell)(\delta\ell_1), \bar{w}). \end{aligned}$$

Arbitrarily, we have

$$\begin{aligned} D^2 J_{\text{OLS}}(\ell)(\delta\ell, \delta\ell) = {} & \kappa D^2 R(\ell)(\delta\ell, \delta\ell) + \langle \delta u, \delta u \rangle \\ & + \langle \delta p, \delta p \rangle + 2a(\delta\ell, \delta u, \bar{w}). \end{aligned} \tag{2.17}$$

We can now compute the derivative $D^2 J_{\text{OLS}}(\ell)(\delta\ell, \delta\ell)$ given $\ell \in A$, $\delta\ell \in A$ by solving (2.6) to compute $u(\ell) = (\bar{u}(\ell), p(\ell))$, then computing $w(\ell) = (\bar{w}(\ell), p_w(\ell))$ by (2.12), and then getting $\delta u = (\delta u, \delta p)$ by (2.14). This gives all the elements required to finally get $D^2 J_{\text{OLS}}(\ell)(\delta\ell, \delta\ell)$ by using (2.17).

2.2.2.2 Adjoint Method for Second Derivative Computation

The second method for computation of the second order derivative of the regularized OLS functional is the second order adjoint method. It is used in order to avoid the computation of the second derivative of the solution map $\bar{u}$. The idea is to apply the results (2.14) and (2.12) twice to avoid the computation of $\delta^2 \bar{u}$.

Define $H : A \times \hat{V} \times \hat{V} \to \mathbb{R}$ by

$$
\begin{aligned}
H(\ell, t, s) = \; & DJ_{\mathrm{OLS}}(\ell)(\delta\ell_2) + a(\ell, u, \bar{t}) + b(\bar{t}, p) + b(u, q_t) - c(p, q_t) - m(\bar{t}) \\
& + a(\ell, \bar{w}\bar{s}) + b(\bar{s}, p_w) + b(\bar{w}, q_s) - c(p_w, q_s) - \langle \bar{z} - u, \bar{s} \rangle - \langle \hat{z} - p, q_s \rangle \\
= \; & \kappa DR(\ell)(\delta\ell_2) + a(\delta\ell_2, u, \bar{w}) + a(\ell, u, \bar{t}) + b(\bar{t}, p) \\
& + b(u, q_t) - c(p, q_t) - m(\bar{t}) + a(\ell, \bar{w}, \bar{s}) + b(\bar{s}, p_w) \\
& + b(\bar{w}, q_s) - c(p_w, q_s) - \langle \bar{z} - u, \bar{s} \rangle - \langle \hat{z} - p, q_s \rangle ,
\end{aligned}
$$

where $\delta\ell_2$ is a fixed direction, $\bar{u} = (u, p)$ is the solution of the saddle point problem (2.6), $w = (\bar{w}, p_w)$ is the solution of the saddle point problem (2.12), $t = (\bar{t}, q_t)$ and $s = (\bar{s}, q_s) \in V$ are arbitrary elements, and (2.13) is used for $DJ_{\mathrm{OLS}}(\ell)(\delta\ell_2)$.
From the above we get that for every $t, s \in \hat{V}$

$$
\frac{\partial H}{\partial \ell}(\ell, t, s)(\delta\ell_1) = D^2 J_{\mathrm{OLS}}(\ell)(\delta\ell_1, \delta\ell_2). \tag{2.18}
$$

The right-hand side derivative of $H(\cdot, \cdot, \cdot)$ at $\ell \in A$ in the direction $\delta\ell_1$ can be computed as follows:

$$
\begin{aligned}
\frac{\partial H}{\partial \ell}(\ell, t, s)(\delta\ell_1) = \; & \kappa D^2 R(\ell)(\delta\ell_1, \delta\ell_2) \; + a(\delta\ell_2, Du(\ell)(\delta\ell_1), \bar{w}) \\
& + a(\delta\ell_2, u, D\bar{w}(\ell)(\delta\ell_1)) + a(\delta\ell_1, u, \bar{t}) + a(\ell, Du(\ell)(\delta\ell_1), \bar{t}) \\
& + b(\bar{t}, Dp(\ell)(\delta\ell_1)) + b(Du(\ell)(\delta\ell_1), q_t) - \; c(Dp(\ell)(\delta\ell_1), q_t) \\
& + a(\delta\ell_1, \bar{w}, \bar{s}) + a(\ell, D\bar{w}(\ell)(\delta\ell_1), \bar{s}) + b(\bar{s}, Dp_w(\ell)(\delta\ell_1)) \\
& + b(D\bar{w}(\ell)(\delta\ell_1), q_s) - c(Dp_w(\ell)(\delta\ell_1), q_s) \\
& + \langle Du(\ell)(\delta\ell_1), \bar{s} \rangle + \langle Dp(\ell)(\delta\ell_1), q_s \rangle .
\end{aligned} \tag{2.19}
$$

Recall (2.14) can be rewritten as

$$
\begin{aligned}
a(\ell, Du(\ell)(\delta\ell_2), v) + b(v, Dp(\ell)(\delta\ell_2)) \quad &= \quad -a(\delta\ell_2, u, v), \quad \forall v \in V, && \text{(2.20a)} \\
b(Du(\ell)(\delta\ell_2), q) - c(Dp(\ell)(\delta\ell_2), q) \quad &= \quad 0, \quad \forall q \in Q, && \text{(2.20b)}
\end{aligned}
$$

Making the substitution $(v, q) = (D\bar{w}(\ell)(\delta\ell_1), Dp_w(\ell)(\delta\ell_1))$ in (2.20) we get

$$
\begin{aligned}
& a(\ell, Du(\ell)(\delta\ell_2), D\bar{w}(\ell)(\delta\ell_1)) + b(D\bar{w}(\ell)(\delta l_1), Dp(\ell)(\delta\ell_2)) \\
& \qquad\qquad = -a(\delta\ell_2, u, D\bar{w}(\ell)(\delta\ell_1)), \\
& b(Du(\ell)(\delta\ell_2), Dp_w(\ell)(\delta\ell_1)) - c(Dp(\ell)(\delta\ell_2), Dp_w(\ell)(\delta\ell_1)) = 0,
\end{aligned}
$$

adding both expressions and using the symmetric property of $a(\cdot, \cdot, \cdot)$ and $c(\cdot, \cdot)$, we obtain

$$
\begin{aligned}
& a(\ell, D\bar{w}(\ell)(\delta\ell_1), Du(\ell)(\delta\ell_2)) + b(Du(\ell)(\delta\ell_2), Dp_w(\ell)(\delta\ell_1)) \\
& + b(D\bar{w}(\ell)(\delta\ell_1), Dp(\ell)(\delta\ell_2)) - c(Dp_w(\ell)(\delta\ell_1), Dp(\ell)(\delta\ell_2)) \\
& \qquad\qquad = -a(\delta\ell_2, u, D\bar{w}(\ell)(\delta\ell_1)).
\end{aligned} \tag{2.21}
$$

Because $w(\ell) = (\bar{w}(\ell), p_w(\ell))$ is the solution of (2.12), the following saddle point problem holds,

$$\begin{aligned} a(\ell, \bar{w}, v) + b(v, p_w) &= \langle \bar{z} - u, v\rangle\,, \quad \text{for every } v \in V \\ b(\bar{w}, q) - c(p_w, q) &= \langle \hat{z} - p, q\rangle\,, \quad \text{for every } q \in Q, \end{aligned}$$

It can be shown that the derivative, $Dw(\ell)(\delta\ell_2) = (D\bar{w}(\ell)(\delta\ell_2), Dp_w(\ell)(\delta\ell_2))$, of $w(\ell)$ in any direction $\delta\ell_2 \in A$ is characterized as the solution of the following saddle point problem:

$$\begin{aligned} a(\ell, D\bar{w}(\ell)(\delta\ell_2), v) &+ b(v, Dp_w(\ell)(\delta\ell_2)) \\ &= -a(\delta\ell_2, \bar{w}, v) - \langle Du(\ell)(\delta\ell_2), v\rangle\,, \quad \forall v \in V, \\ b(D\bar{w}(\ell)(\delta\ell_2), q) &- c(Dp_w(\ell)(\delta\ell_2), q) \\ &= -\langle Dp(\ell)(\delta\ell_2), q\rangle\,, \quad \forall q \in Q. \end{aligned} \tag{2.22}$$

Making the substitution $(v, q) = (Du(\ell)(\delta\ell_1), Dp(\ell)(\delta\ell_1))$ into (2.22), we obtain

$$\begin{aligned} a(\ell, D\bar{w}(\ell)(\delta\ell_2), Du(\ell)(\delta\ell_1)) &+ b(Du(\ell)(\delta\ell_1), Dp_w(\ell)(\delta\ell_2)) \\ = -a(\delta\ell_2, \bar{w}, Du(\ell)(\delta\ell_1)) &- \langle Du(\ell)(\delta\ell_2), Du(\ell)(\delta\ell_1)\rangle \\ b(D\bar{w}(\ell)(\delta\ell_2), Dp(\ell)(\delta\ell_1)) &- c(Dp_w(\ell)(\delta\ell_2), Dp(\ell)(\delta\ell_1)) \\ &= -\langle Dp(\ell)(\delta\ell_2), Dp(\ell)(\delta\ell_1)\rangle\,, \end{aligned}$$

By adding up these equations and using the symmetry of $a(\cdot,\cdot,\cdot)$ and $c(\cdot,\cdot)$, we obtain,

$$\begin{aligned} &a(\ell, Du(\ell)(\delta\ell_1), D\bar{w}(\ell)(\delta\ell_2)) + b(D\bar{w}(\ell)(\delta\ell_2), Dp(\ell)(\delta\ell_1)) \\ &+ b(Du(\ell)(\delta\ell_1), Dp_w(\ell)(\delta\ell_2)) - c(Dp(\ell)(\delta\ell_1), Dp_w(\ell)(\delta\ell_2)) \\ &= -a(\delta\ell_2, \bar{w}, Du(\ell)(\delta\ell_1)) - \langle Du(\ell)(\delta\ell_2), Du(\ell)(\delta\ell_1)\rangle \\ &- \langle Dp(\ell)(\delta\ell_2), Dp(\ell)(\delta\ell_1)\rangle\,. \end{aligned} \tag{2.23}$$

Now we set $s = (\bar{s}, q_s) = (Du(\ell)(\delta\ell_2), Dp(\ell)(\delta\ell_2))$ and
$t = (\bar{t}, q_t) = (D\bar{w}(\ell)(\delta\ell_2), Dp_w(\ell)(\delta\ell_2))$ in (2.19) and put that into (2.21) and (2.23), to get

$$\begin{aligned} \frac{\partial H}{\partial \ell}(\ell, t, s)(\delta\ell_1) = \kappa D^2R(\ell)(\delta\ell_1, \delta\ell_2) &+ a(\delta\ell_2, Du(\ell)(\delta\ell_1), \bar{w}) \\ &+ a(\delta\ell_2, u, D\bar{w}(\ell)(\delta\ell_1)) + a(\delta\ell_1, u, D\bar{w}(\ell)(\delta\ell_2)) \\ &- a(\delta\ell_2, \bar{w}, Du(\ell)(\delta\ell_1)) - \langle Du(\ell)(\delta\ell_2), Du(\ell)(\delta\ell_1)\rangle \\ &- \langle Dp(\ell)(\delta\ell_2), Dp(\ell)(\delta\ell_1)\rangle + a(\delta\ell_1, \bar{w}, Du(\ell)(\delta\ell_2)) \\ &- a(\delta\ell_2, u, D\bar{w}(\ell)(\delta\ell_1)) + \langle Du(\ell)(\delta\ell_1), Du(\ell)(\delta\ell_2)\rangle \\ &+ \langle Dp(\ell)(\delta\ell_1), Dp(\ell)(\delta\ell_2)\rangle \\ = \kappa D^2R(\ell)(\delta\ell_1, \delta\ell_2) &+ a(\delta\ell_1, u, Dw(\ell)(\delta\ell_2)) \\ &+ a(\delta\ell_1, \bar{w}, Du(\ell)(\delta\ell_2)). \end{aligned}$$

Therefore, from (2.18), we obtain the following formula for the second-order derivative of the regularized OLS that has no explicit computation of the second-order derivatives of the solution map:

$$\begin{aligned} D^2 J_{\text{OLS}}(\ell)(\delta\ell_1, \delta\ell_2) = \kappa D^2 R(\ell)(\delta\ell_1, \delta\ell_2) &+ a(\delta\ell_1, u, Dw(\ell)(\delta\ell_2)) \\ &+ a(\delta\ell_1, \bar{w}, Du(\ell)(\delta\ell_2)) \end{aligned}$$

Arbitrarily,

$$\begin{aligned} D^2 J_{\text{OLS}}(\ell)(\delta\ell, \delta\ell) = \kappa D^2 R(\ell)(\delta\ell, \delta\ell) &+ a(\delta\ell, u, Dw(\ell)(\delta\ell)) \\ &+ a(\delta\ell, \bar{w}, Du(\ell)(\delta\ell)). \end{aligned} \tag{2.24}$$

Given $\ell \in A$, $\delta\ell \in A$, we compute $D^2 J_{\text{OLS}}(\ell)(\delta\ell, \delta\ell)$ by first using (2.6) to compute $\bar{u} = (u, p)$, and (2.12) to compute $w = (\bar{w}, p_w)$. Next we use (2.14) to get $\delta\bar{u} = (\delta u, \delta p)$, and (2.22) to get $\delta w = (\delta\bar{w}, \delta p_w)$. This gives all the pieces needed to compute $D^2 J_{\text{OLS}}(\ell)(\delta\ell, \delta\ell)$ using (2.24).

2.3 Modified Output Least Squares

The Modified Output Least Squares, (MOLS), scheme is introduced to cope with some of the problems of the OLS functional like convexity. The MOLS functional is convex,so the local optimality conditions are also global optimality conditions. The MOLS functional does not require the computation of the solution map for the first order derivative. It is known that the convexity allows for higher convergence speed of the MOLS functional over the OLS functional.
The MOLS objective functional uses the weak form of the system as a guide, and is as follows:

$$\begin{aligned} J_{\text{MOLS}}(\ell) = &\frac{1}{2} a(\ell, u(\ell) - \bar{z}, u(\ell) - \bar{z}) + b(u(\ell) - \bar{z}, p(\ell) - \hat{z}) \\ &- \frac{1}{2} c(p(\ell) - \hat{z}, p(\ell) - \hat{z}) \end{aligned} \tag{2.25}$$

where all variables are the same as in the case of the OLS scheme. Also, as with the OLS, this functional is susceptible to ill-posedness and so regularization of some sort is introduced to combat this problem, and provide numerical and computational stability.
Because of the definition of the MOLS functional in (2.25), the computation of the derivative is very simple. The first derivative is free from the computation of the solution map u, which is one of the reasons this method is computationally less expensive that OLS.

Given (2.25), we get that

$$
\begin{aligned}
DJ_{\text{MOLS}}(\ell)(\delta\ell) &= \frac{1}{2}a(\delta\ell, u(\ell)-\bar{z}, u(\ell)-\bar{z}) + a(\ell, \delta u(\ell), u(\ell)-\bar{z}) \\
&\quad - c(\delta p(\ell), p(\ell)-\hat{z}) + b(\delta u(\ell), p(\ell)-\bar{z}) + b(u(\ell)-\bar{z}, \delta p) \\
&= -\frac{1}{2}a(\delta\ell, u(\ell)+\bar{z}, u(\ell)-\bar{z}).
\end{aligned}
$$

Computation of the second order derivative of the MOLS method is as easy as the first order derivative computation. We apply the chain rule to the above form of the first order derivative and obtain

$$
\begin{aligned}
D^2 J_{\text{MOLS}}(\ell)(\delta\ell, \delta\ell) &= -\frac{1}{2}a(\delta\ell, \delta u, u(\ell)-\bar{z}) - \frac{1}{2}a(\delta\ell, u(\ell)+\bar{z}, \delta u) \\
&= a(\ell, \delta u, \delta u) + c(\delta p, \delta p) \qquad (2.26)
\end{aligned}
$$

This then makes the second order derivative of the MOLS functional, a two step procedure. First, compute $\delta u = (\delta u, \delta p)$ by solving (2.14), and then use (2.26) to compute $D^2 J_{\text{MOLS}}(\ell)(\delta\ell, \delta\ell)$

All the details on the MOLS functional can be found in [28].

2.4 Energy Output Least Squares

A variant of the modified OLS is the following energy OLS given below:

$$
J_{\text{EOLS}}(\ell) = \frac{1}{2}a(\ell, u(\ell)-\bar{z}, u(\ell)-\bar{z}) + \frac{1}{2}c(p(\ell)-\hat{z}, p(\ell)-\hat{z}) \qquad (2.27)
$$

All variables used are the same as in the previous sections. Because of the ill-posed nature of the functional, regularization needs to be applied to some level in order to ensure stability, but not so much that it alters the accuracy of the method. This functional is also convex, and this is an advantage over the generic OLS functional.

It follows that the first order derivative of the EOLS functional is given by.

$$
DJ_{\text{EOLS}}(\ell)(\delta\ell) = \frac{1}{2}a(\delta\ell, u-\bar{z}, u-\bar{z}) + a(\delta\ell, u, \bar{w}). \qquad (2.28)
$$

Furthermore, it is known that the second derivative is given by

$$
\begin{aligned}
D^2 J_{\text{EOLS}}(\ell)(\delta\ell, \delta\ell) &= 2a(\delta\ell, \delta u, u-\bar{z}) + a(\ell, \delta u, \delta u) \\
&\quad + c(\delta p, \delta p) + 2a(\delta\ell, \delta u, \bar{w}). \qquad (2.29)
\end{aligned}
$$

All the details on the EOLS functional can be found in [20].

2.5 Equation Error Approach

Another optimization formulation is the so-called equation error approach introduced below. We define $e_1(\mu, \bar{u}) \in V$ and $e_2(\mu, \bar{u}) \in Q$ such that

$$\langle e_1(\mu, \bar{u}), v \rangle = \int_\Omega 2\mu\epsilon(u) \cdot \epsilon(v) + \int_\Omega p(\text{div}v) - \int_\Omega fv - \int_{\Gamma_2} vh \ \ \forall v \in V,$$

$$\langle e_2(\mu, \bar{u}), q \rangle = \int_\Omega (\text{div}u)q - \int_\Omega \frac{1}{\lambda} pq \quad \forall q \in Q.$$

This now allows us to define $e(\mu, \bar{u}) = (e_1(\mu, \bar{u}), e_2(\mu, \bar{u})) \in V \times Q$ such that

$$\langle e(\mu, \bar{u}), \bar{v} \rangle = \int_\Omega 2\mu\epsilon(u) \cdot \epsilon(v) + \int_\Omega p(\text{div}v) - \int_\Omega fv - \int_{\Gamma_2} vh - \int_\Omega (\text{div}u)q + \int_\Omega \frac{1}{\lambda} pq \tag{2.30}$$

where $\bar{v} = (v, q)$.
With this framework set up, it then makes sense to minimize the newly defined function $e(\mu, z)$ with respect to μ, where $z = (\bar{z}, \hat{z})$ is some form of measurement of the solution map $\bar{u}$.
This then makes the EE functional

$$J_{\text{EE}}(\mu) = \frac{1}{2} ||e(\mu, z)||_{\hat{V}}^2 \tag{2.31}$$

where $z = (\bar{z}, \hat{z})$ is some measurement of the solution map u, and pressure p. As with every other functional, this functional also requires a level of regularization for computational stability.
An advantage of this scheme is that it is uniquely solvable in both its continuous, and discrete form. It also produces a convex functional therefore a minimizer is guaranteed to be found. Lastly, this scheme is computationally inexpensive compared to the other forms of the Output Least Squares schemes, as there are no underlying variational problems to be solved. The drawback of this method is that it relies on differentiating the data or measurements entered into the system. As a result, noise or errors in the data can cause solutions to be very inaccurate. In other words, this method is not very robust, and is very sensitive to noise in the data.
The EE functional is uniquely set up such that the computation of its derivatives are defined similarly. The functional (2.31) can be written as

$$J_{\text{EE}}(\mu) = \frac{1}{2} \langle e_1, e_1 \rangle + \frac{1}{2} \langle e_2, e_2 \rangle \tag{2.32}$$

Note that derivatives are taken with respect to μ, and by the definition of e_1, and e_2, we know that e_2 is constant with respect to μ. The first order derivative of the EE functional is the direct derivative of (2.32)

$$DJ_{\text{EE}}(\mu)(\delta\mu) = \langle e_1(\mu, z)\delta\mu, e_{1t}(\mu, z) \rangle \tag{2.33}$$

where

$$\langle e_{1t}(\mu, \bar{u}), v\rangle = a(\mu, \bar{u}, \bar{v}) + b(\bar{v}, p)$$

Similarly, the second order derivative is uniquely defined through direct computation of (2.33) to yield

$$D^2 J_{\mathrm{EE}}(\mu)(\delta\mu, \delta\mu) = \langle e_{1t}(\mu, z)\delta\mu, e_{1t}(\mu, z)\delta\mu\rangle. \tag{2.34}$$

All the details on the EE functional can be found in [18].

Chapter 3

Discretization

In this section, we give the discrete formulations of the objective functionals and their derivatives. We start off by discretizing the spaces in which all our variables and parameters are defined, and then we discretize the functionals of our optimization schemes.

3.1 Finite Element Discretization

We use finite element discretization for both the direct and the inverse problem. We assume that we have a set of finite dimensional subspaces of V and Q, which we represent as $\{V_h\}$, and $\{Q_h\}$. Now, we define $\hat{V}_h = V_h \times Q_h$, where the spaces is the product topology are such that the discrete form of the Babushka-Brezzi condition holds. We also assume B_h is a finite dimensional subspace of B so that we can define a non-empty finite dimensional subspace of feasible coefficients $A_h = B_h \cap A$. Lastly, we assume that there exist component wise projection operations from each space to its choice finite dimensional subspace.
The discrete version of the saddle point problem now becomes

$$\begin{aligned} a(\ell_h, u_h, v) + b(v, p_h) &= m(v) \ \ \forall v \in V_h \\ b(u_h, q) - c(p_h, q) &= 0 \quad \forall q \in Q_h \end{aligned} \tag{3.1}$$

First we define a triangulation $\mathcal{T}_h$ on $\Omega \subset \mathbb{R}^2$, in which L_h is the space of all piecewise continuous polynomials of degree d_ℓ relative to $\mathcal{T}_h$, U_h is the space of all piecewise continuous polynomials of degree d_u relative to $\mathcal{T}_h$, and Q_h is the space of all piecewise continuous polynomials of degree d_q relative to $\mathcal{T}_h$. Next we define the bases for the finite dimensional subspaces L_h, U_h and Q_h by $\{\varphi_1, \varphi_2, \ldots, \varphi_m\}$, $\{\psi_1, \psi_2, \ldots, \psi_n\}$, and $\{\chi_1, \chi_2, \ldots, \chi_k\}$, respectively in order to allow for numerical computation. L_h is now isomorphic to $\mathbb{R}^m$ and for any $\ell \in L_h$, we define $L \in \mathbb{R}^m$ by $L_i = \ell(x_i), \quad i = 1, 2, \ldots, m$, where the nodal basis $\{\varphi_1, \varphi_2, \ldots, \varphi_m\}$ corresponds to the nodes $\{x_1, x_2, \ldots, x_m\}$. Each $L \in \mathbb{R}^m$ now corresponds to $\ell \in L_h$ defined by

$$\ell = \sum_{i=1}^{m} L_i \varphi_i.$$

Similarly, $u \in U_h$ will correspond to $U \in \mathbb{R}^n$, where $\bar{U}_i = u(y_i)$, $i = 1, 2, \ldots, n$, and we define

$$u = \sum_{i=1}^{n} \bar{U}_i \psi_i,$$

where $y_1, y_2, \ldots, y_n$ are the nodes of the mesh defining U_h. Lastly, $q \in Q_h$ will correspond to $Q \in \mathbb{R}^k$, where $Q_i = q(z_i)$, $i = 1, 2, \ldots, k$, and

$$q = \sum_{i=1}^{k} Q_i \chi_i,$$

where $z_1, z_2, \ldots, z_k$ are the nodes of the mesh defining Q_h. The finite dimensional subspaces A_h, U_h, and Q_h are defined relative to the same elements, but the nodes will be different if the condition that $d_\ell \neq d_u \neq d_q$ hold.
Recall that the discrete saddle point problem in (3.1) looks to find the unique $(u_h, p_h) \in V_h \times Q_h$, for each ℓ_h, such that

$$\begin{aligned} a(\ell_h, u_h, v) + b(v, p_h) &= m(v), \quad \forall v \in U_h, && \text{(3.2a)} \\ b(u_h, q) - c(p_h, q) &= 0, \quad \forall q \in Q_h. && \text{(3.2b)} \end{aligned}$$

Now, define $S : R^m \to R^{n+k}$ to be the finite element solution operator that assigns the unique approximate solution $\bar{u}_h = (u_h, p_h) \in U_h \times Q_h$ to each coefficient $\ell_h \in A_h$. Thus $S(L) = U$, where U is defined by

$$K(L)U = F, \tag{3.3}$$

where the stiffness matrix $K(L) \in R^{(n+k)\times(n+k)}$ and the load vector $F \in R^{n+k}$ are given by

$$K(L) = \begin{bmatrix} \widehat{K}_{n\times n}(L) & B^{\mathrm{T}}_{n\times k} \\ B_{k\times n} & -C_{k\times k} \end{bmatrix}$$

with

$$\begin{aligned} \widehat{K}(L)_{i,j} &= a(\ell, \psi_j, \psi_i), \quad i, j = 1, 2, \ldots, n, \\ B_{i,j} &= b(\psi_j, \chi_i), \quad i = 1, 2, \ldots, k, \quad n = 1, 2, \ldots, n \\ C_{i,j} &= c(\chi_j, \chi_i), \quad i, j = 1, 2, \ldots, k, \\ F_i &= m(\psi_i), \quad i = 1, 2, \ldots, n, \\ F_j &= 0, \quad j = n+1, n+2, \ldots, n+k. \end{aligned}$$

It is important to note that

$$\widehat{K}(L)_{ij} = T_{ijk}L_k,$$

where the summation convention is used and T is the tensor defined by

$$T_{ijk} = a(\varphi_k, \psi_i, \psi_j), \quad \text{for every } i, j = 1, \ldots, n, \; k = 1, \ldots, m.$$

For ease of computation, we approximate the components of U_h in a single finite element space $\widetilde{U}_h$ where $U_h = \widetilde{U}_h \times \widetilde{U}_h$. Therefore, if $\{\psi_1, \ldots, \psi_\ell\}$ are the basis of $\widetilde{U}_h$ then the vector-valued basis of U_h can be chosen as

$$\{\psi_i\}_{i=1}^n = \left\{ \begin{bmatrix} \psi_1 \\ 0 \end{bmatrix}, \begin{bmatrix} \psi_2 \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} \psi_\ell \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \psi_1 \end{bmatrix}, \begin{bmatrix} 0 \\ \psi_2 \end{bmatrix}, \cdots, \begin{bmatrix} 0 \\ \psi_t \end{bmatrix} \right\}$$

3.2 Discrete Optimizers

In this section, we propose discrete formulations of the optimization schemes discussed in the previous chapter.

3.2.1 Discrete Output Least Squares

We will now discretize the OLS functional as well as its first order and second order derivatives.
We define the regularized partial OLS functional given by

$$J_{\text{OLS}}(\ell) = \frac{1}{2} \|u(\ell) - \bar{z}\|_V^2 + \kappa R(\ell),$$

where $z = (\bar{z}, \hat{z})$ is the measured data and $u(\ell) = (\bar{u}(\ell), p(\ell))$ is the solution of the following saddle point problem:

$$\begin{aligned} a(\ell, u, v) + b(v, p) &= m(v), \quad \forall v \in V, \\ b(u, q) - c(p, q) &= 0, \quad \forall q \in Q. \end{aligned}$$

The discretized form of the functional becomes

$$J_{\text{OLS}}(L) := \frac{1}{2}(\bar{U} - \bar{Z})^T M(\bar{U} - \bar{Z}) + \kappa R(L),$$

where $M \in \mathbb{R}^{n \times n}$ is defined by

$$M_{ij} = \langle \psi_i, \psi_j \rangle$$

and $U = (\bar{U}, P)$ solves the following linear system

$$\begin{bmatrix} \widehat{K}_{n\times n}(L) & B_{n\times k}^{\mathrm{T}} \\ B_{k\times n} & -C_{k\times k} \end{bmatrix} \begin{bmatrix} \bar{U} \\ P \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix}. \tag{3.4}$$

3.2.1.1 Gradient Computation

We now proceed to give a gradient formula.
We first describe a direct approach.
Recall that the first-order derivative of the regularized OLS is given by

$$DJ_{\text{OLS}}(\ell)(\delta\ell) \ = \langle \delta u, u - \bar{z} \rangle + \kappa DR\,(l)\,(\delta\ell), \tag{3.5}$$

This derivative uses $\delta\bar{u}(\ell) = (\delta u(\ell), \delta p(\ell))$ which is characterized as the unique solution of the following saddle point problem:

$$\begin{aligned} a(\ell, \delta u, v) + b(v, \delta p) &= -a(\delta\ell, u, v), \quad \forall v \in V \\ b(\delta u, q) - c(\delta p, q) &= 0 \quad \forall q \in Q. \end{aligned}$$

The discrete formulation of the above saddle point problem is given by the following linear system

$$K(L)\delta U = \widehat{F}(\delta L), \tag{3.6}$$

where $\widehat{F} \in \mathbb{R}^{n+k}$ is given by

$$\widehat{F}(\delta L) = \begin{pmatrix} -\hat{K}(\delta L)\bar{U} \\ 0 \end{pmatrix},$$

This can be simplified by defining the adjoint stiffness matrix $\mathbb{A}$ which holds the following condition:

$$\widehat{K}(L)\bar{V} = \mathbb{A}(\bar{V})L, \ \ \forall L \in \mathbb{R}^m, \ \ \forall \bar{V} \in \mathbb{R}^n. \tag{3.7}$$

This implies that

$$\widehat{F}(\delta L) = \begin{pmatrix} -\mathbb{A}(\bar{U})(\delta L) \\ 0 \end{pmatrix}. \tag{3.8}$$

And so, the gradient

$$\nabla U = [\nabla_1 U \cdots \nabla_m U] = \begin{bmatrix} \nabla_1 \bar{U} & \cdots & \nabla_m \bar{U} \\ \nabla_1 P & \cdots & \nabla_m P \end{bmatrix} \in \mathbb{R}^{(k+n)\times m}$$

is computed by solving the following m linear equations

$$K(L)\nabla_i U = \widehat{F}(E_i), \qquad i = 1, ..., m, \tag{3.9}$$

which is,

$$\begin{pmatrix} \hat{K}(L) & B^{\mathrm{T}} \\ B & -C \end{pmatrix} \begin{pmatrix} \nabla_i \bar{U} \\ \nabla_i P \end{pmatrix} = \begin{pmatrix} -\mathbb{A}(\bar{U})E_i \\ 0 \end{pmatrix}, \tag{3.10}$$

where $\{E_i\}_{i=1,\ldots,m} \subset \mathbb{R}^m$ denotes the canonical basis of $\mathbb{R}^m$, and $\nabla\bar{U} \in \mathbb{R}^{k\times m}$ denotes the matrix

$$\nabla\bar{U} = [\nabla_1 \bar{U} \cdots \nabla_m \bar{U}],$$

We follow the same notation for $\nabla P \in \mathbb{R}^{n\times m}$.
Therefore, the discretization of (3.5) is as follows:

$$\begin{aligned} DJ_{\text{OLS}}(L)(\delta L) &= \left\langle \delta\bar{U}, \bar{U}-\bar{Z}\right\rangle + \kappa\nabla R\left(L\right)\delta L \\ &= \left(\bar{U}-\bar{Z}\right)^{\mathrm{T}} M\nabla\bar{U}\delta L + \kappa\nabla R\left(L\right)\delta L, \end{aligned}$$

We can then get an explicit form for the gradient of the regularized OLS functional as;

$$\nabla J_{\text{OLS}}(L) = \left(\bar{U}-\bar{Z}\right)^{\mathrm{T}} M\nabla\bar{U} + \kappa\nabla R\left(L\right). \tag{3.11}$$

So in order to compute the gradient of the regularized OLS functional by direct method, we first compute $U = (\bar{U}, P)$ by solving (3.4), then $\nabla\bar{U}$ by solving (3.9), so that we have all the elements in place to compute the gradient. The gradient $\nabla J_{\text{OLS}}(L)$ is computed by using (3.11).

3.2.1.1.1 Adjoint Approach The first-order derivative by using the first-order adjoint approach reads

$$DJ_{\text{OLS}}(\ell)(\delta\ell) \ = \kappa DR(\ell)(\delta\ell) + a(\delta l, u, \bar{w}), \tag{3.12}$$

where $\bar{u} = (u, p)$ is the solution to the saddle point problem (2.6) and $w = (\bar{w}, q)$ is the solution to the problem (2.12), respectively.
The discrete version of these elements are the vectors $U = (\bar{U}, P)$ which solves (3.4) and $W = (\bar{W}, P_w)$ which solves the linear systems below

$$\begin{bmatrix} \widehat{K}_{n\times n}(L) & B^{\mathrm{T}}_{n\times k} \\ B_{k\times n} & -C_{k\times k} \end{bmatrix} \begin{bmatrix} \bar{W} \\ P_w \end{bmatrix} = \begin{bmatrix} M(\bar{Z}-\bar{U}) \\ 0 \end{bmatrix}. \tag{3.13}$$

Thus for the unregularized first order adjoint computation of the OLS first order derivative in (3.12) we have

$$a(\delta L, \bar{U}, \bar{W}) = \bar{U}^{\mathrm{T}}\widehat{K}(\delta L)\bar{W} = \bar{U}^{\mathrm{T}}\mathbb{A}(\bar{W})\delta L,$$

where $\mathbb{A}$ is the adjoint stiffness matrix. Thus, we arrive at the following discrete version of (3.12)

$$DJ_{\text{OLS}}(L)(\delta L) = \kappa\nabla R(L)(\delta L) + \bar{U}^{\mathrm{T}}\mathbb{A}(\bar{W})\delta L,$$

This means we can get an explicit formula for the gradient of the OLS functional as

$$\nabla J_{\text{OLS}}(L) = \kappa\nabla R(L) + \bar{U}^{\mathrm{T}}\mathbb{A}(\bar{W}). \tag{3.14}$$

The steps then for calculating the gradient of the discrete regularized OLS functional using the first order adjoint method involve computing $U = (\bar{U}, P)$, and $W = (\bar{W}, P_w)$ by solving the systems (3.4) and (3.13) respectively. Now we can compute the gradient, $\nabla J_{\text{OLS}}(L)$, by using (3.14).

3.2.1.2 Hessian Computation

We calculate the discretized second order derivative or Hessian of the regularized OLS functional.

3.2.1.2.1 Hybrid Method Recall the second-order derivative of the regularized OLS

$$D^2 J_{\text{OLS}}(\ell)(\delta\ell, \delta\ell) \; = \kappa D^2 R(\ell)(\delta\ell, \delta\ell) + \langle \delta u, \delta u \rangle + 2a(\delta\ell, \delta u, \bar{w}).$$

By direct discretization, we get

$$\begin{aligned} \nabla^2 J_{\text{OLS}}(L)(\delta L, \delta L) &= \kappa D^2 R(L)(\delta L, \delta L) + \left\langle \delta\bar{U}, \delta\bar{U} \right\rangle + 2a(\delta L, \delta\bar{U}, \bar{W}) \\ &= \delta L^T \nabla^2 R(L) \delta L + \kappa \delta L^T \nabla\bar{U}^{\mathrm{T}} M \nabla\bar{U} \delta L \\ &\quad + 2\delta L^{\mathrm{T}} \nabla\bar{U}^{\mathrm{T}} \hat{K}(\delta L) \bar{W} \\ &= \delta L^T \nabla^2 R(L) \delta L + \kappa \delta L^T \nabla\bar{U}^{\mathrm{T}} M \nabla\bar{U} \delta L \\ &\quad + 2\delta L^{\mathrm{T}} \nabla\bar{U}^{\mathrm{T}} \mathbb{A}(\bar{W}) \delta L. \end{aligned} \tag{3.15}$$

From (3.15), we can get an explicit form for the Hessian.

$$\nabla^2 J_{\text{OLS}}(L) = \kappa \nabla^2 R(L) + \; \nabla\bar{U}^T M \nabla\bar{U} \; + 2\nabla\bar{U}^T \mathbb{A}(\bar{W}). \tag{3.16}$$

The steps to calculate the hessian of the regularized OLS start off with computing $U = (\bar{U}, P)$, and $W = (\bar{W}, P_w)$, by solving the systems (3.4), and (3.13) respectively. Then you compute $\nabla U = (\nabla\bar{U}, \nabla P)$ by solving (3.9). Finally, the hessian, $\nabla^2 J_{\text{OLS}}(L)$, can be computed using (3.16).

3.2.1.2.2 Adjoint Approach The second-order derivative of the regularized OLS by the second-order adjoint approach is given by

$$\begin{aligned} D^2 J_{\text{OLS}}(\ell)(\delta\ell, \delta\ell) &= \kappa D^2 R(\ell)(\delta\ell, \delta\ell) + a(\delta\ell, u, Dw(\ell)(\delta\ell)) \\ &\quad + a(\delta\ell, \bar{w}, Du(\ell)(\delta\ell)). \end{aligned}$$

Recall also the saddle point problem

$$\begin{aligned} &a(\ell, D\bar{w}(\ell)(\delta\ell_2), v) + b(v, Dp_w(\ell)(\delta\ell_2)) = -a(\delta\ell_2, \bar{w}, v) - \langle Du(\ell)(\delta\ell_2), \bar{v} \rangle\,, \quad \forall v \in V, \\ &\qquad b(D\bar{w}(\ell)(\delta\ell_2), q) - c(p_w(\ell)(\delta\ell_2), q) = 0, \quad \forall q \in Q, \end{aligned}$$

Now note that we need to compute Dw, and this can be done by solving the following system of m linear equations.

$$\begin{bmatrix} \widehat{K}_{n\times n}(L) & B^{\mathrm{T}}_{n\times k} \\ B_{k\times n} & -C_{k\times k} \end{bmatrix} \begin{bmatrix} \nabla_i \bar{W} \\ \nabla_i P \end{bmatrix} = \begin{bmatrix} -\mathbb{A}(\bar{U}) E_i - M(\nabla U E_i) \\ 0 \end{bmatrix}. \tag{3.17}$$

The Hessian is thus;

$$\nabla^2 J_{\text{OLS}}(L) = \kappa\nabla^2 R(L) + \nabla\bar{W}^T\mathbb{A}(\bar{U}) + \nabla\bar{U}^T\mathbb{A}(\bar{W}). \tag{3.18}$$

using the same adjoint technique used in the previous Hessian computation. We compute the Hessian for the second-order adjoint approach by computing $U = (\bar{U}, P)$, and $W = (\bar{W}, P)$ using (3.4), and (3.13), respectively. Next, we compute $\nabla U = (\nabla\bar{U}, \nabla P)$ and $\nabla W = (\nabla\bar{W}, \nabla P_w)$ by solving m linear systems each in (3.10) and (3.17), respectively. Now, we compute the hessian, $\nabla^2 J_{\text{OLS}}(L)$, using (3.18).

3.2.2 Discrete MOLS

In this section, we collect discrete formulas for the MOLS, its gradient, and hessian.
We have

$$\begin{aligned} J_{\text{MOLS}}(L) = &\frac{1}{2}(\bar{U}(L)-\bar{Z})^{\mathrm{T}}\widehat{K}(L)(\bar{U}(L)-\bar{Z}) + (\bar{U}(L)-\bar{Z})^{\mathrm{T}}B^{\mathrm{T}}(P(L)-\widehat{Z}) \\ &-\frac{1}{2}(P(L)-\widehat{Z})^{\mathrm{T}}C(P(L)-\widehat{Z}). \end{aligned} \tag{3.19}$$

Moreover,

$$\begin{aligned} \nabla J_{\text{MOLS}}(L) &= -\frac{1}{2}\mathbb{A}(\bar{U}(L)+\bar{Z})^{\mathrm{T}}(\bar{U}(L)-\bar{Z}) \\ &= -\frac{1}{2}\mathbb{A}(\bar{U}(L))^{\mathrm{T}}\bar{U}(L) + \frac{1}{2}\mathbb{A}(\bar{Z})^{\mathrm{T}}\bar{Z}, \\ \nabla^2 J_{\text{MOLS}}(L) &= \ \nabla\bar{U}(L)^{\mathrm{T}}\mathbb{A}(\nabla\bar{U}(L)) + \nabla P(L)^{\mathrm{T}}C\nabla P(L) \end{aligned}$$

3.2.3 Discrete EOLS

We compute the discrete EOLS functional;

$$J_{\text{EOLS}}(L) = \frac{1}{2}(\bar{U}(L)-\bar{Z})^{\mathrm{T}}\widehat{K}(L)(\bar{U}(L)-\bar{Z}) + \frac{1}{2}(P(L)-\widehat{Z})^{\mathrm{T}}C(P(L)-\widehat{Z}).$$

Moreover,

$$\begin{aligned} \nabla J_{\text{EOLS}}(L) &= \frac{1}{2}\left(\bar{U}-\bar{Z}\right)^{\mathrm{T}}\mathbb{A}(\bar{U}-\bar{Z}) + \bar{U}^{\mathrm{T}}\mathbb{A}(\bar{W}(L)) \\ \nabla^2 J_{\text{EOLS}}(L) &= 2^{\mathrm{T}}\nabla\bar{U}^{\mathrm{T}}\mathbb{A}(\bar{U}-\bar{Z}) \ + \nabla\bar{U}^{\mathrm{T}}\hat{K}(L)\nabla\bar{U} \quad + \nabla P^{\mathrm{T}}C\nabla P + 2\nabla\bar{U}^{\mathrm{T}}\mathbb{A}(\bar{W}) \end{aligned}$$

3.2.4 Discrete Equation Error

We can now compute the EE functional

$$\begin{aligned} J_{\text{EE}}(L) = {} & \frac{1}{2}\left(\mathbb{A}(\bar{U})L + B^T P - F\right)^T (K+M)^{-1}\left(\mathbb{A}(\bar{U})L + B^T P - F\right) \\ & + \frac{1}{2}\left(B\bar{U} - CP\right)^T M_Q^{-1}\left(B\bar{U} - CP\right) \end{aligned}$$

Moreover,

$$\begin{aligned} \nabla J_{\text{EE}}(L) &= \mathbb{A}(\bar{U})^T (K+M)^{-1}\left(\mathbb{A}(\bar{U})L + B^T P - F\right), \\ \nabla^2 J_{\text{EE}}(L) &= \mathbb{A}(\bar{U})^T (K+M)^{-1}\left(\mathbb{A}(\bar{U})\right). \end{aligned}$$

Chapter 4

Heavy Ball with Friction Method

4.1 Introduction

In this section, we pose our minimization problem in terms of a dynamical system, so that differential equations based solvers can be used. Iterative techniques are typically used to solve our minimization problem. These solutions at each step of iteration can be put into a sequence, with the sequence limiting to the minimizer of our functional. We can see this sequence as the path of the solution to a dynamical system over artificial time, so that as time approaches infinity, the solution to the dynamical system converges to the minimizer. The next step is to find a suitable dynamical system that closely models such a sequence, and this is what we explore below.

4.2 Continuous Methods

The main focus is the minimization of an objective functional $J(a)$. This implies that, given a suitable trajectory, a continuous method can be used to solve the minimization problem. If the trajectory used is the continuous gradient, then a steepest descent approach is being utilized. That is

$$\begin{aligned} \frac{da}{dt} &= -\nabla J(a) \\ a(t_0) &= a_o \end{aligned} \tag{4.1}$$

In order to solve the minimization problem

$$\min_{\tilde{A}} J(a)$$

we must solve the associated initial value problem. The trajectory of the gradient leads us to a minimizer for the functional $J(a)$. So

$$\{a_n\}_{n\in\mathbb{N}} \to a^* \quad \text{as} \quad n \to \infty$$

where

$$|J(a^*)| \leq J(a) \quad \forall a \in \tilde{A}$$

thus a^* is a minimizer of the functional, and $\tilde{A}$ is the set of feasible values of a.

4.2.1 A Continuous Newton-Type Trajectory

Above, we consider, a trajectory defined solely by the gradient of the functional being minimized. However, another possible trajectory considered by Zhang, Kelley and Liao[48], is the so-called Newton's direction defined by both the gradient and the Hessian of the objective functional.

$$\begin{aligned} \nabla^2 J(a)\frac{da}{dt} &= -\nabla J(a) \\ a(t_0) &= a_0 \end{aligned}$$

where $\nabla^2 J(a)$ is the Hessian of the objective functional $J(a)$. Thus, the initial value problem becomes

$$\begin{aligned} \frac{da}{dt} &= -(\nabla^2 J(a))^{-1}\nabla J(a) \\ a(t_0) &= a_0 \end{aligned} \tag{4.2}$$

This method however can only be applied when the resulting Hessian matrix has no singularity issues. Zhang, Kelley, and Liao proposed a compromise in which the magnitude of the minimum eigenvalue decides what method will be used at each step. The scheme proposed chooses between (4.1) and (4.2) or a convex combination of both.

$$\begin{aligned} \frac{da}{dt} &= g(a) \\ a(t_0) &= a_o \end{aligned}$$

where

$$g(a) = \begin{cases} -(\nabla^2 J(a))^{-1}\nabla J(a) & \text{if} \quad \lambda_{min}(a) > \delta_2 \\ -\alpha(a)(\nabla^2 J(a))^{-1}\nabla J(a) - \beta(a)\nabla J(a) & \delta_1 \leq \lambda_{min}(a) \leq \delta_2 \\ -\nabla J(a) & \lambda_{min}(a) < \delta_1 \end{cases}$$

where $\lambda_{min}(a)$ is the minimum eigenvalue of $\nabla^2 J(a)$, and $\delta_2 > \delta_1 > 0$. We define $\alpha(a)$, and $\beta(a)$ as follows;

$$\begin{aligned} \alpha(a) &= \frac{\lambda_{min}(a) - \delta_1}{\delta_2 - \delta_1} \\ \beta(a) &= 1 - \alpha(a) \\ &= \frac{\delta_2 - \lambda_{min}(a)}{\delta_2 - \delta_1} \end{aligned}$$

This formulation shows that the combined trajectory is a weighted trajectory that leans towards the gradient when $\lambda_{min}(a)$ is closer to δ_1, and leans towards the continuous Newton direction when $\lambda_{min}(a)$ is closer is to δ_2.
Conditions for convergence are presented in [48]. In the simple steepest descent gradient method, the added condition for convergence to a minimizer of $\nabla J(a)$ is that it be Lipschitz continuous in the bounded sets of the Hilbert Space in which it is defined. Extending this, the added requirement for convergence here is that $-(\nabla^2 J(a))^{-1}\nabla J(a)$ be Lipschitz continuous, and this would imply that $g(a)$ is also Lipschitz continuous. This proof is also provided in [48].

4.3 Heavy Ball with Friction Method

Let H be a real Hilbert space and $\Phi : H \to \mathbb{R}$ be a continuously differentiable function with a Lipschitz continuous gradient on the bounded set of H. Attouch et al. in [10] study the nonlinear dissipative dynamical system

$$a'' + \lambda a' + \nabla\Phi(a) = 0$$

In [10], and [2], Attouch and Alvarez concluded that, given the above conditions, the asymptotic behavior of the solution of the dissipative dynamical system has a trajectory that converges weakly towards a minimizer of Φ.
The system models a heavy mass (heavy ball) rolling down a trajectory defined by Φ with the friction term between the mass and the surface of the trajectory being λ, thus the name heavy ball with friction method. Attouch et al.[10] derived the true form of the heavy ball with friction system as

$$\begin{aligned} &a'' + \lambda a' + g\nabla\Phi(a) = 0 \\ &a(0) = a_0 \quad a'(0) = a'_0 \end{aligned} \tag{4.3}$$

where g is some form of downward force, usually a gravitational term. However, they noted that because of the scale and scope of the minimization problem, the physical interpretation of the model with regards to a heavy

mass rolling down a surface, may not make direct sense, but allows for the user to make educated guesses on the behavior of the technique.
It is important to note that, as oppose to the steepest descent techniques, the heavy ball with friction technique is a nonlinear oscillatory technique. It allows for oscillation in its trajectory and as a result, where the steepest descent technique might stop at the first local minimizer it finds, (4.3) allows for oscillation, and as such is better suited for minimization problems with multiple local minimizers. As one might imagine, in the context of a heavy ball rolling down a trajectory, the terms λ, and g would heavily determine the speed at which it would roll, and its ability to handle oscillations. The friction term λ would tend to slow it down to a certain extent, so decreasing the term might seem like a simple solution, but it has a strictly positive restriction. Also, if the friction is not enough, then the ball might roll past a potential minimizer, and this is not optimal. The gravitational term g also plays a roll with regards to the speed of the ball and its ability to handle oscillations in a physical sense. As a result, managing the balance between the friction and the gravitational term is key to this method.
Now (4.3) can be written as a first order system in $H \times H$ as

$$A' = S(A)$$

where

$$A' = \begin{bmatrix} a(t) \\ a'(t) \end{bmatrix} \quad \text{and} \quad S(u,v) = \begin{bmatrix} v \\ -\lambda v - g\nabla\Phi(u) \end{bmatrix}$$

All the previous assumptions of Φ are held, including boundedness from below so that existence and uniqueness of a local solution can be proven by the Cauchy-Lipschitz theorem for the first order initial value problem

$$\begin{aligned} A' &= S(A), \\ A(0) &= A_0 \end{aligned} \tag{4.4}$$

where $A_0 = \begin{bmatrix} a_0 \\ a_0' \end{bmatrix}$
The novel idea here is to combine the advantages of the continuous Newton-type method, its relative speed over the continuous gradient approach, with the improvement of convergence speed of the heavy ball with friction method. This yield the following system from (4.3)

$$\begin{aligned} &a'' + \alpha a' + \beta g(a) = 0 \\ &a(0) = a_0 \quad a'(0) = a_0' \end{aligned} \tag{4.5}$$

where $g(a)$ is as previously defined, and $\alpha, \beta > 0$. The conditions for convergence still hold in this case. Letting $a'(t) = v(t)$, and $a'(0) = v_0$ we can combine (4.4) and (4.5) to yield

$$\begin{aligned} A' &= S(A), \\ A(0) &= A_0 \end{aligned}$$

where $A_0 = \begin{bmatrix} a_0 \\ v_0 \end{bmatrix}$, $A' = \begin{bmatrix} a(t) \\ v(t) \end{bmatrix}$ and $S(a, v) = \begin{bmatrix} v \\ -\alpha v - \beta g(a) \end{bmatrix}$
The heavy ball with friction method is studied extensively in [2, 3, 9, 8, 10, 30]. The choice of the parameters α, and β are guided by their interpretations in the physical model, however, the actual values used would not make sense in that context because of the scope of the minimization problem. Educated, but heuristic choices for the parameters are used during experimentation.

Chapter 5

Numerical Experiments

5.1 Introduction

In this section, we consider two examples for the elasticity problem, in which we try to recover the parameter μ. Because we are considering an incompressible body for our model, we set the value of $\lambda = 10^6$ in the model.
One smooth, and one piece-wise defined example are tested. For the smooth examples, the inverse problem is solved on a 15×15 quadrangular mesh with 289 degrees of freedom.
For the piece-wise example, the inverse problem is solved on a 25×25 quadrangular mesh with 729 degrees of freedom.
The stopping criteria for each problem is set as

$$||\nabla J(a)|| \leq 10^{-12}$$

Recall the elasticity problem.

$$\begin{aligned} -\nabla \sigma &= f \ \text{ in } \ \Omega \\ u &= g \ \text{ on } \ \Gamma_1 \\ \sigma n &= h \ \text{ on } \ \Gamma_2 \\ \text{where} & \\ \sigma &= 2\mu\epsilon_u + \lambda \mathrm{tr}(\epsilon_u) I, \\ \epsilon_u &= \frac{1}{2}\left(\nabla u + \nabla u^T\right) \end{aligned} \tag{5.1}$$

with $\Omega = (0,1) \times (0,1)$, $\partial\Omega = \Gamma_1 \cup \Gamma_2$. Dirichlet conditions hold on Γ_1, which is, for our example, the top boundary, and Neumann boundary conditions hold on Γ_2 which the rest of the boundary.
First, we will be reporting the results from testing the new computations of the second order derivative of the OLS functional. Next, we will show

the results from testing the heavy ball with friction(HBF) technique to solve (5.1) using

- Modified Output Least Squares
- Energy Output Least Squares
- Equation Error

For the elasticity problem, the following examples are used to run experiments and collect numerical data.

- Example 1

$$\mu(x,y) = \left(1 - .12cos(3\pi\sqrt{x^2+y^2}\right)^{-1}$$
$$f(x,y) = \frac{1}{10}\begin{bmatrix}10+x^2\\ y\end{bmatrix}$$
$$g(x,y) = \begin{bmatrix}0\\0\end{bmatrix}$$
$$h(x,y) = \begin{bmatrix}0.5+x^2\\ 0\end{bmatrix}$$

- Example 2

$$\mu(x,y) = \begin{cases} 0.2sin(\pi x) & \text{for } \{(x,y): 0.2 \le x \le 0.4, 0.2 \le y \le 0.4\} \\ 0.5sin(\pi x) & \text{for } \{(x,y): 0.6 \le x \le 0.8, 0.6 \le y \le 0.8\} \\ 0.3sin(\pi x) & \text{for } \{(x,y): 0.2 \le x \le 0.4, 0.6 \le y \le 0.8\} \\ 0.1sin(\pi x) & \text{for } \{(x,y): 0.6 \le x \le 0.8, 0.2 \le y \le 0.4\} \\ 1 & \text{otherwise} \end{cases}$$

$$f(x,y) = \begin{bmatrix}1+0.1x^2\\ 0.1y\end{bmatrix}$$
$$g(x,y) = \begin{bmatrix}0\\0\end{bmatrix}$$
$$h(x,y) = \begin{bmatrix}0.5+x^2\\ 0\end{bmatrix}$$

In the following tables, all errors recorded are the L_2 Errors, CG represent the continuous newton-type first order method, HB represents the second order heavy ball with friction method, ε is the regularization parameter, α, and β are the parameters from the heavy ball with friction method, Iters stands for the number of iterations, Time is measured in seconds, and λ_{min} is the overall minimum eigen value recorded.

5.2 Computations using OLS

The results reported in this section are done so to test the feasibility of the Second Order Adjoint method and the Hybrid method of Hessian computation. We use a second order technique to solve and compare these two methods.

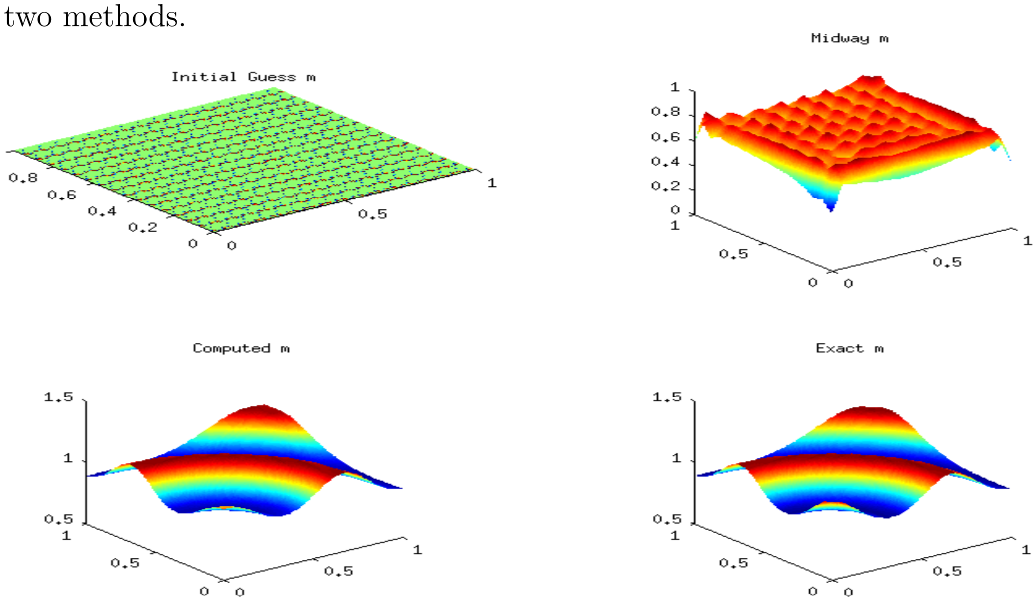

Figure 5.1: Example 1 OLS Hybrid

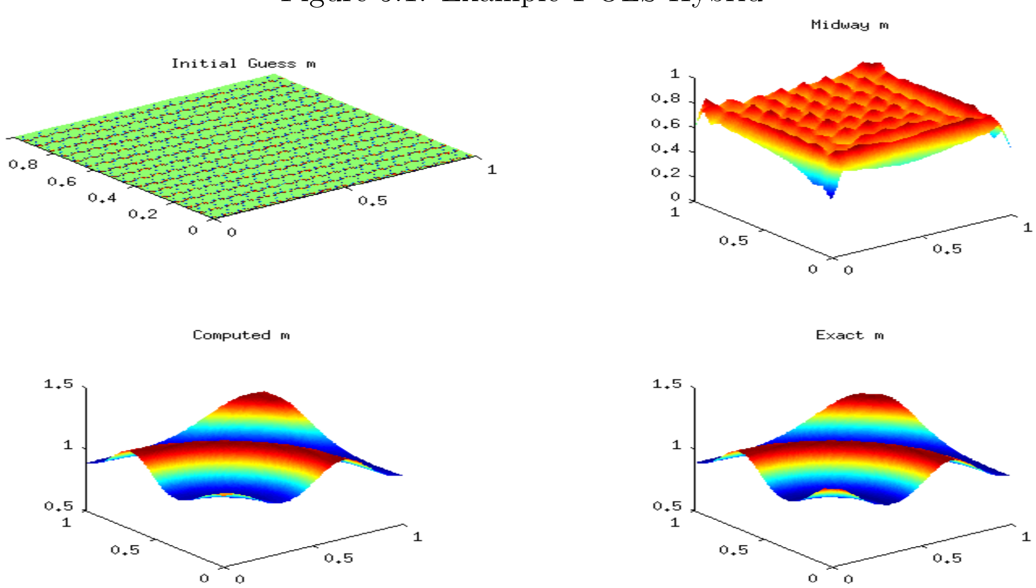

Figure 5.2: Example 1 OLS Adjoint

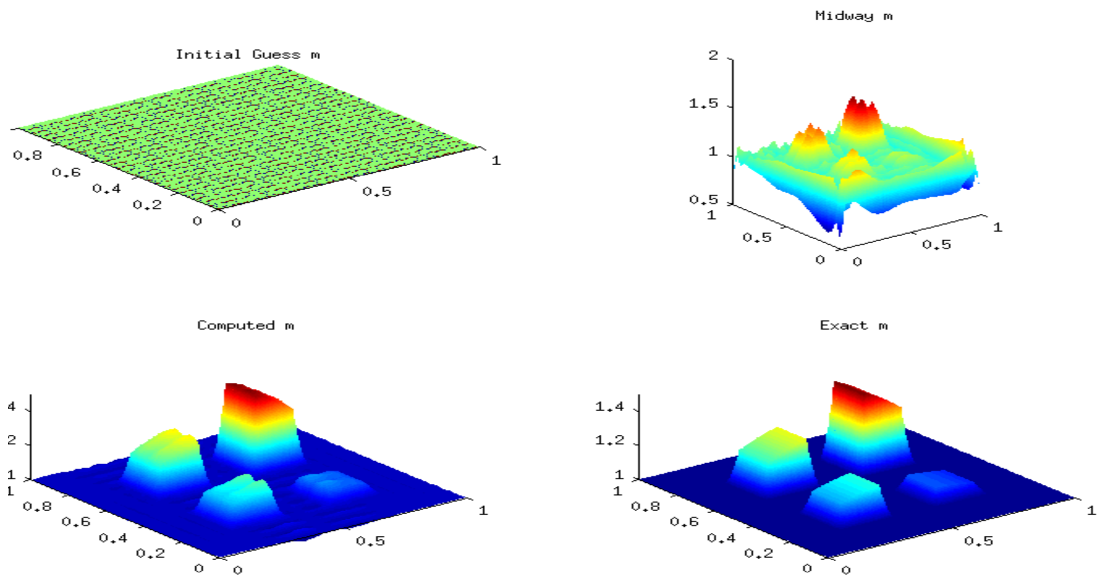

Figure 5.3: Example 2 OLS Hybrid

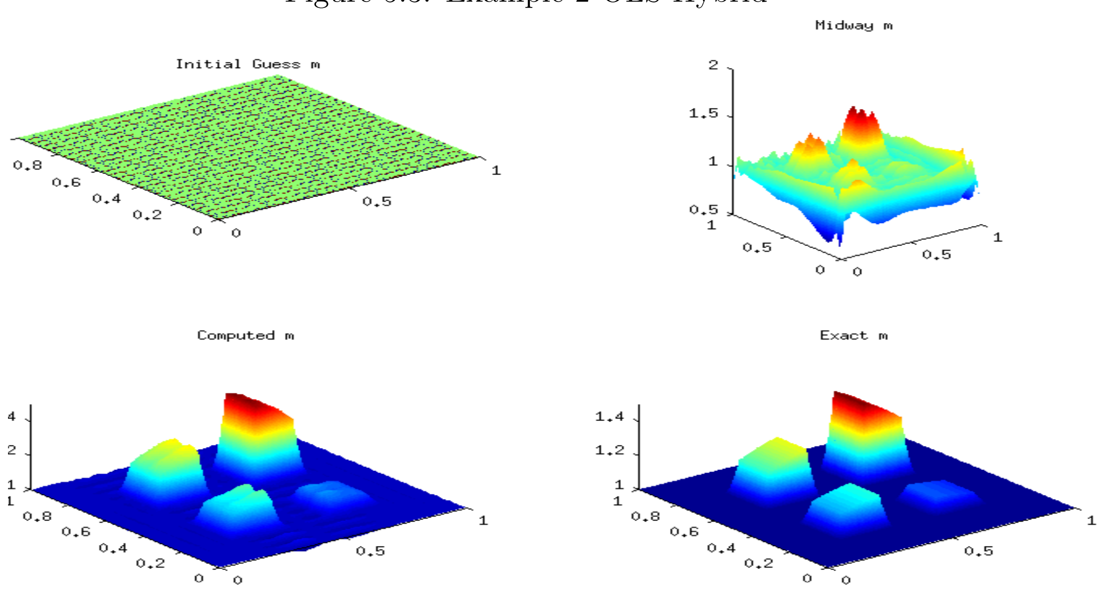

Figure 5.4: Example 2 OLS Adjoint

Table 5.1: OLS Numerical Results

	Method	ε	Iters	Time (s)	Error
1	OLS-H	10^{-6}	15	45.3	$4.68 \cdot 10^{-3}$
	OLS-A	10^{-6}	15	87.9	$4.68 \cdot 10^{-3}$
2	OLS-H	10^{-11}	44	1234.4	$4.37 \cdot 10^{-3}$
	OLS-A	10^{-11}	44	2571.5	$4.37 \cdot 10^{-3}$

5.3 Heavy Ball with Friction Method

Here we will report the results comparing the continuous newton-type method, and the HBF method.

5.3.1 Computations using MOLS

We report the results recovered using the MOLS functional for the Elasticity problem. We also report the parameters used, as well as the time, error and other qualitative variables and parameters involved in each example.
For the continuous gradient method we have;

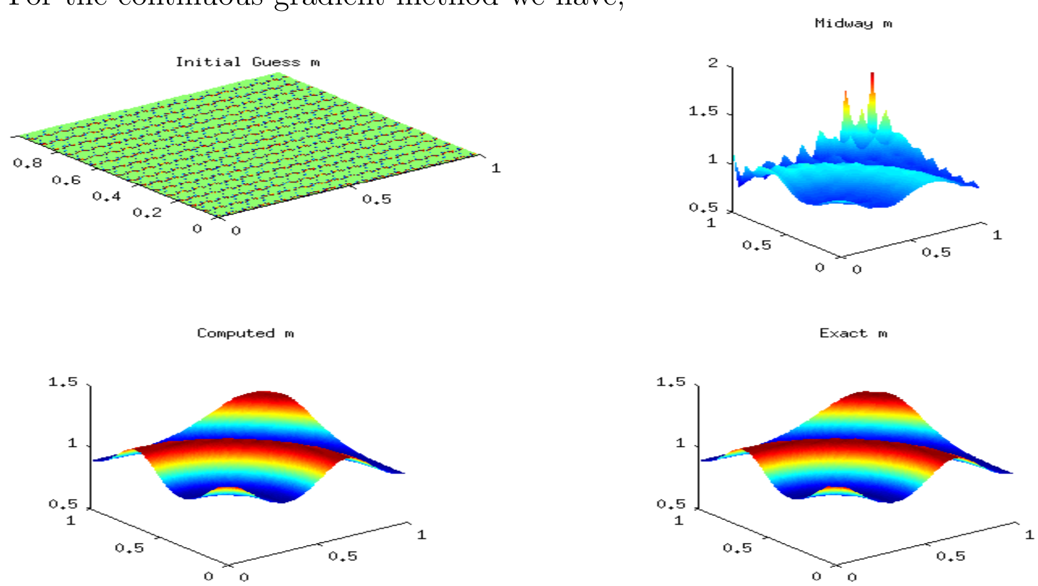

Figure 5.5: Example 1 MOLS CG

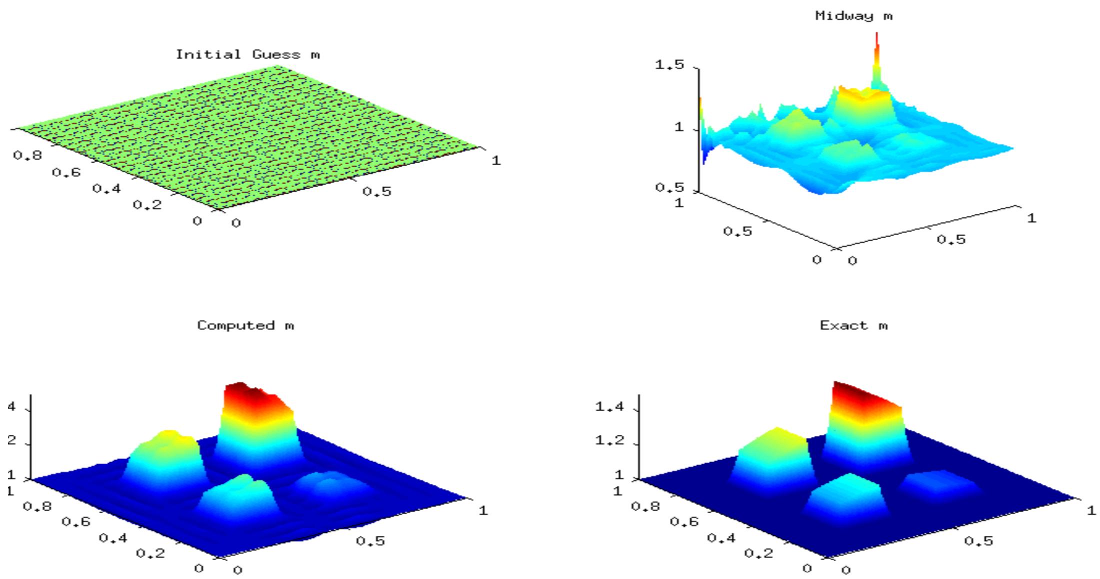

Figure 5.6: Example 2 MOLS CG

For the heavy ball with friction method, we have;

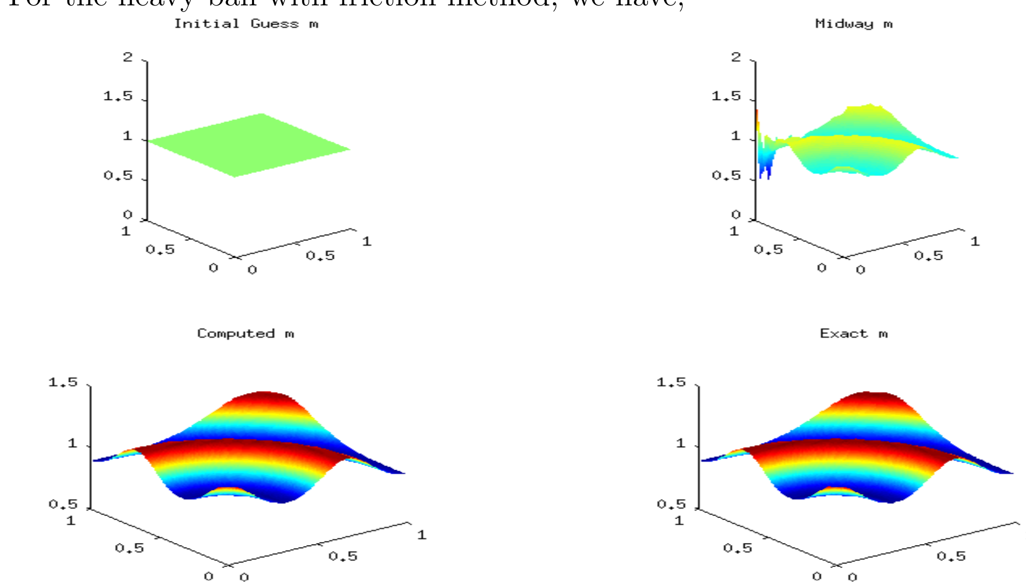

Figure 5.7: Example 1 MOLS HB

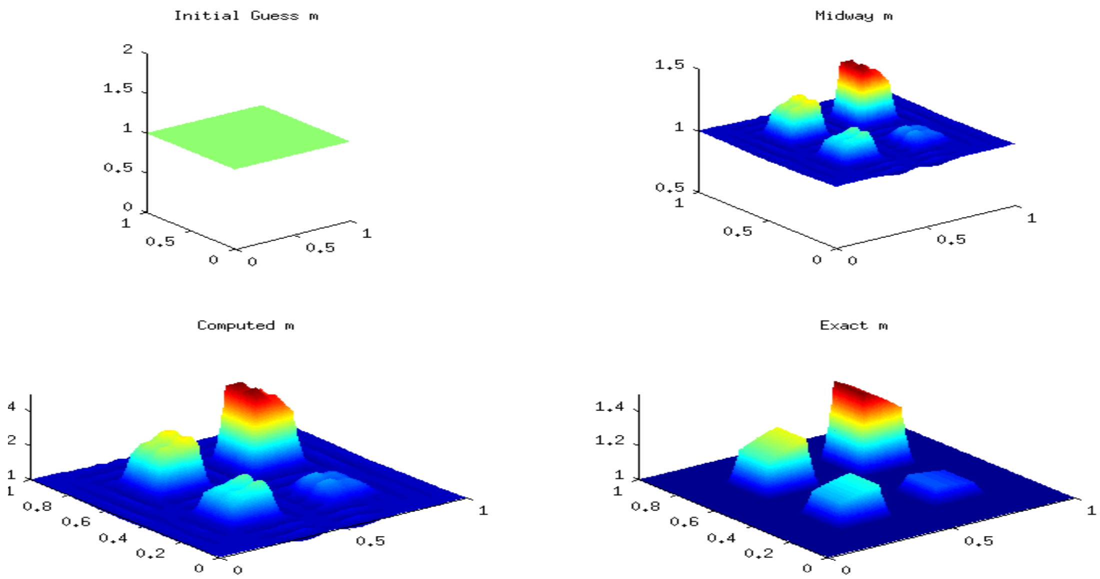

Figure 5.8: Example 2 MOLS HB

Table 5.2: MOLS Numerical Results

	Method	HC	MC	GC	ε	α	β	Time (s)	Iters	λ_{min}
1	CG	13	0	0	10^{-6}	-	-	21.6	13	.1044
	HB	9	0	0	10^{-6}	2.02	1.02	13.8	9	.0985
2	CG	13	0	0	10^{-6}	-	-	182.0	13	.0486
	HB	11	0	0	10^{-6}	2.02	1.02	133.9	11	.0486

For example 1, the L_2 Error recorded for both methods was $3.84 \cdot 10^{-4}$. The error for example 2, was $6.42 \cdot 10^{-3}$.

5.3.2 Computations using EOLS

We report the results recovered using the EOLS scheme. We report the time, error, and other qualitative variables and parameters involved in each example

For the continuous gradient method we have;

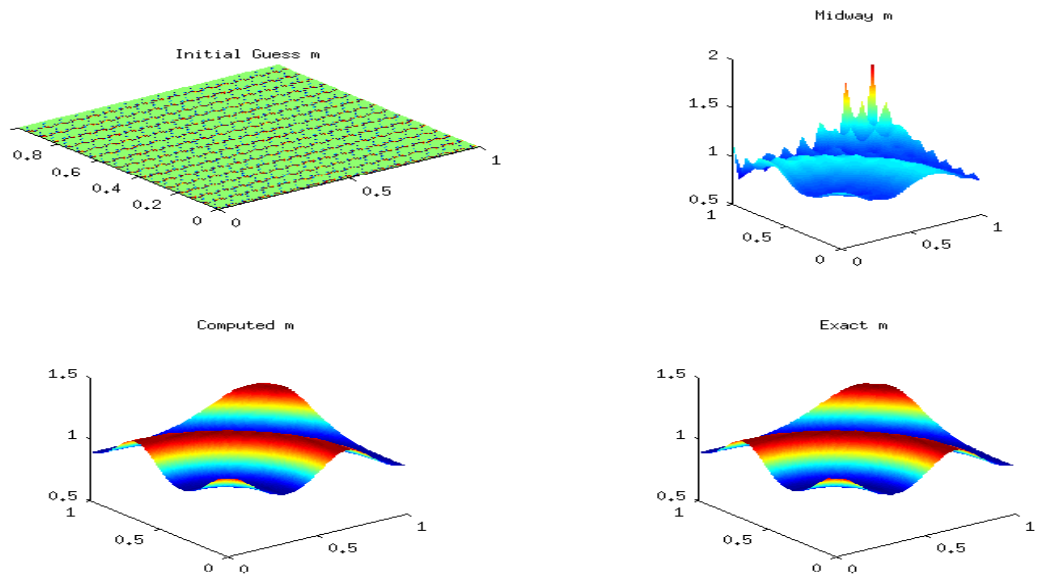

Figure 5.9: Example 1 EOLS CG

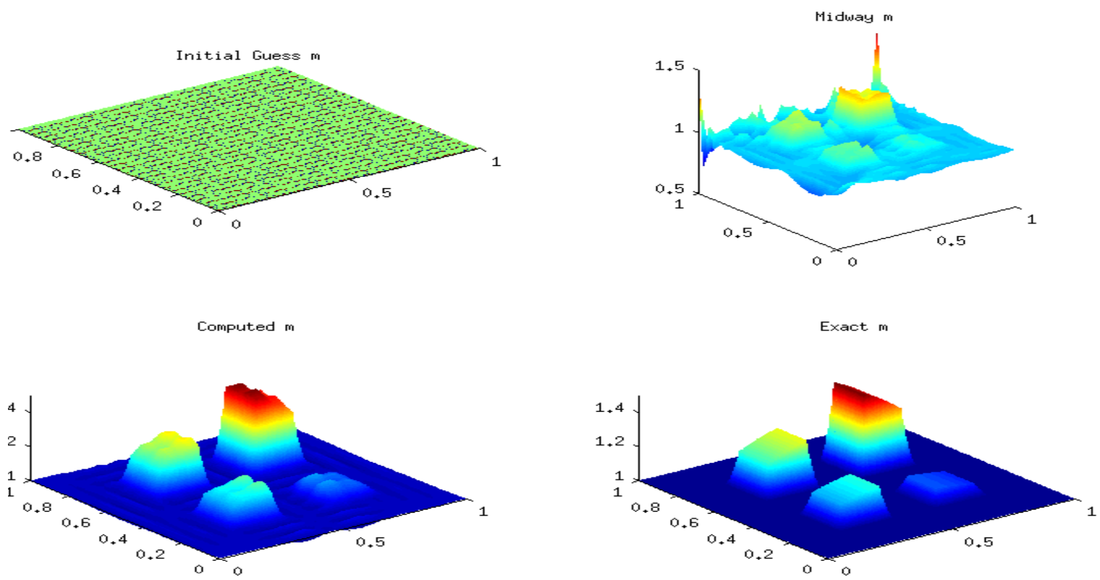

Figure 5.10: Example 2 EOLS CG

For the heavy ball with friction method, we have;

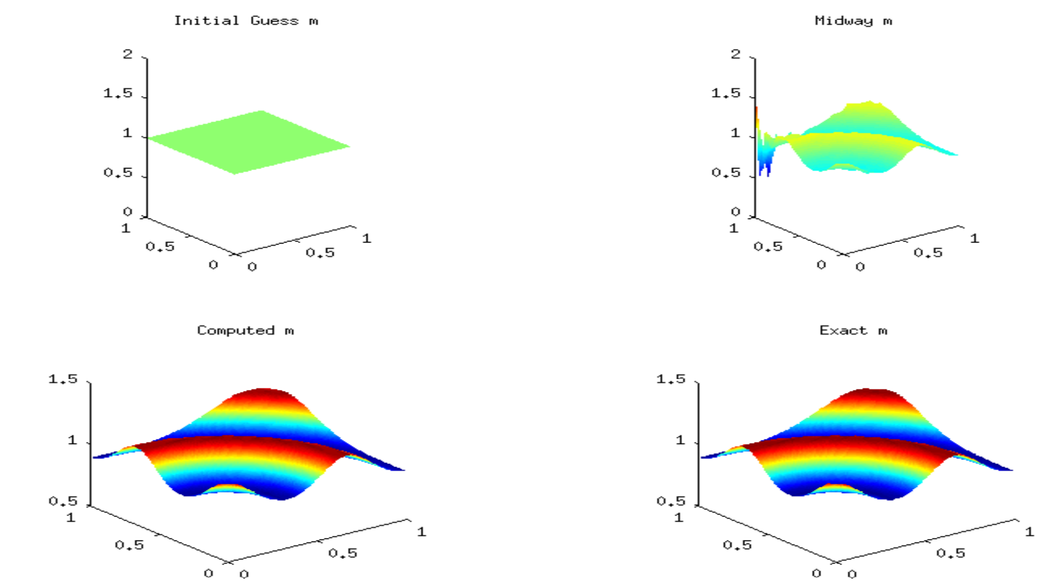

Figure 5.11: Example 1 EOLS HB

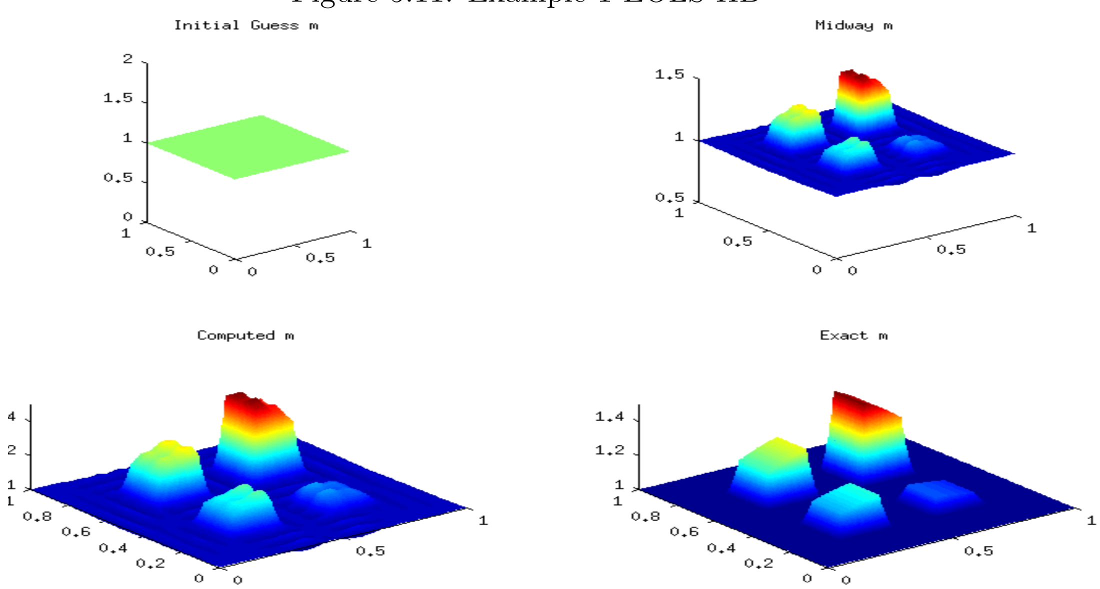

Figure 5.12: Example 2 EOLS HB

Table 5.3: EOLS Numerical Results

	Method	HC	MC	GC	ε	α	β	Time (s)	Iters	λ_{min}
1	CG	13	0	0	10^{-6}	-	-	20.7	13	.1044
	HB	9	0	0	10^{-6}	2.02	1.02	14.1	9	.0985
2	CG	13	0	0	10^{-6}	-	-	183.3	13	.0486
	HB	11	0	0	10^{-6}	2.02	1.02	134.4	11	.0486

For example 1, the L_2 Error recorded for both methods was $3.84 \cdot 10^{-4}$. The error for example 2, was $6.42 \cdot 10^{-3}$

5.3.3 Computations using Equation Error

We report the results obtained from the use of the EE functional for the elasticity problem. We also report the time, error, and other qualitative variables and parameters involved in each example
For the continuous gradient method we have;

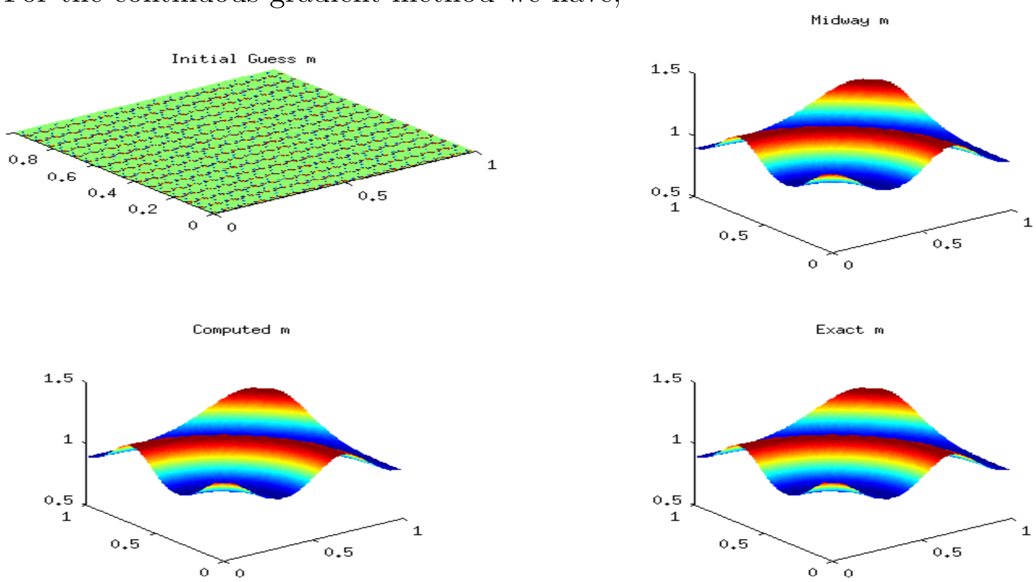

Figure 5.13: Example 1 EE CG

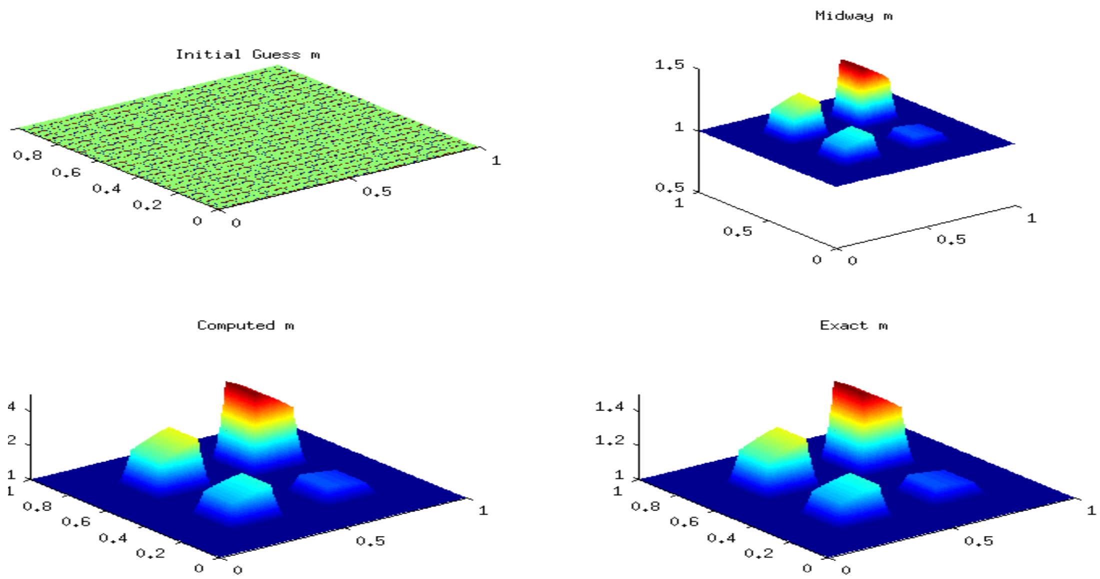

Figure 5.14: Example 2 EE CG

For the heavy ball with friction method, we have;

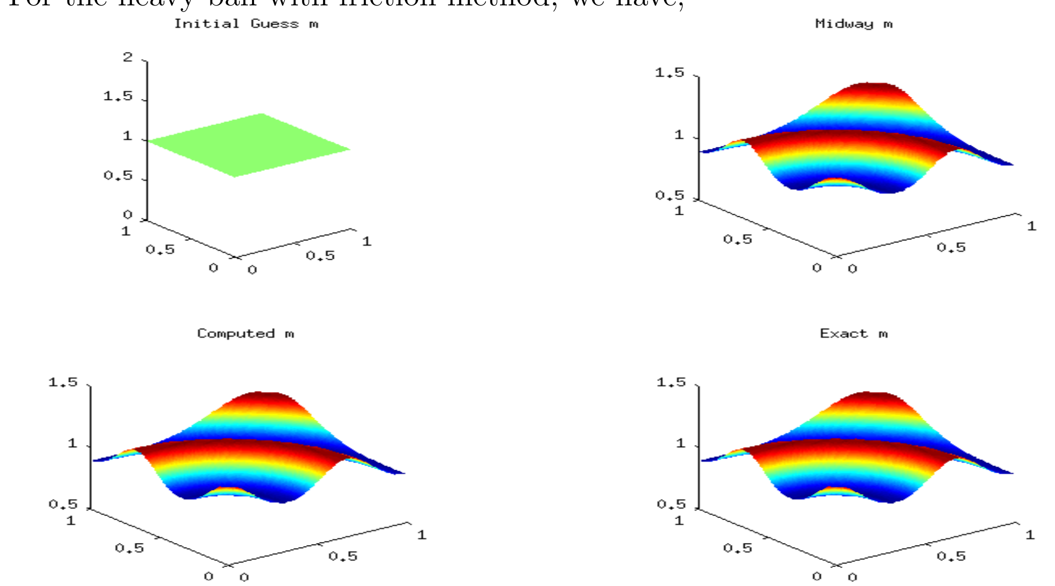

Figure 5.15: Example 1 EE HB

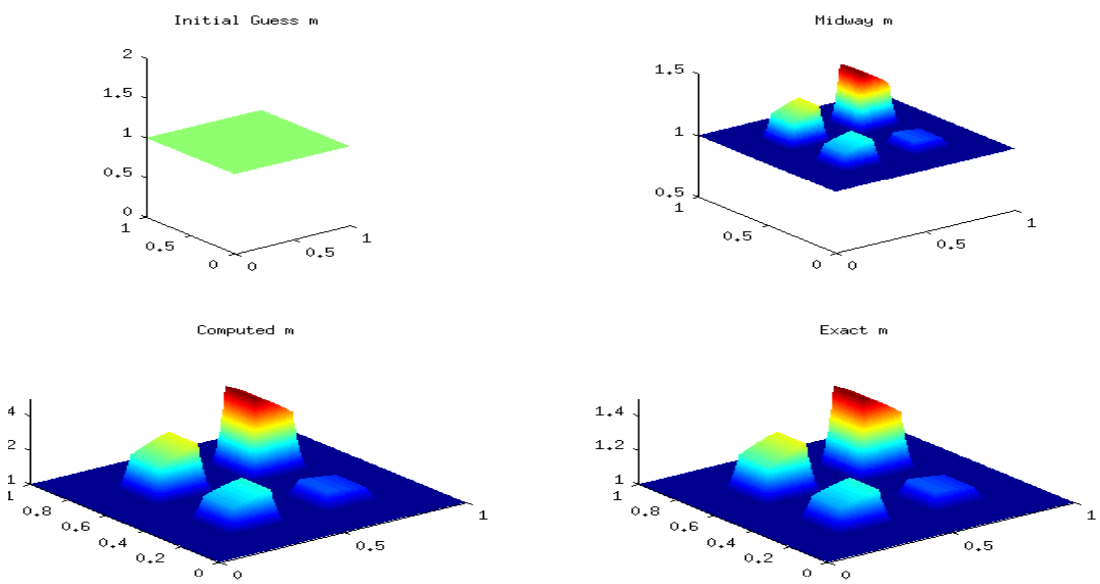

Figure 5.16: Example 2 EE HB

Table 5.4: EE Numerical Results

	Method	HC	MC	GC	ε	α	β	Time (s)	Iters	λ_{min}
1	CG	6	0	0	10^{-6}	-	-	1.0	6	.165
	HB	5	0	0	10^{-6}	2.02	1.02	1.0	5	.165
2	CG	6	0	0	10^{-6}	-	-	3.0	6	.0917
	HB	5	0	0	10^{-6}	2.02	1.02	2.0	5	.0917

For example 1, the L_2 Error recorded for both methods was $5.27 \cdot 10^{-5}$. The error for example 2, was $7.82 \cdot 10^{-5}$

5.4 Choice of Parameters for HBF

The choice of the parameters α, and β are very important when it comes to the number of iterations that the HBF method requires for convergence. It is important to maintain accuracy, and the choice of these parameters tend not to alter this as long as they are within reason. The table below is an example showing different choices of parameters and how they affect the number of iterations required to reach a certain level of accuracy.
The table represents Example 1, using the Equation Error functional, and Euler ODE solver.

Table 5.5: Effect of Parameter Choice in Heavy Ball method

α	β	Iters	Time (s)
2	2	1000	28.2
2.03	1.03	7	1.05
2.02	1.02	5	1.01
2	1	7	1.07
1	1	2557	88.2
0.6	0.1	56	2.64
0.1	0.1	10000	335.4*
0.4	0.06	73	3.74

* - This example was unable to reach required accuracy in 10000 iterations.

In all of these trial, the same level of accuracy is maintained, with the L_2 Error being $5.27 \cdot 10^{-5}$. As we can see the choice of these parameters α, and β, are very important. For every pairing but one, this method seems to be worse than its first order counterpart. This is why the choice of this parameter is vital.

Chapter 6

Performance of Differential Equations Based Solvers for Noisy Data

6.1 Motivation

All of the optimization schemes utilized in this thesis work require data. This data is acquired through measurements made by machines that are prone to errors. This suggests that the data would be subject to a level of noise. As a result, the methods proposed need to be able to handle noise appropriately.

6.2 Objective and Approach

The purpose of this chapter is to compare different differential equation solvers, optimization techniques, and objective functionals. We consider the scalar problem because of its simplicity. It is computationally inexpensive compared to the elasticity problem. The differential equation techniques being considered include:

1. Euler's Method
2. Trapezoidal Method
3. Runge Kutta Method
4. MATLAB's ode113 solver

The objective functionals are tested using these differential equation solvers over varying noise levels in order to compare their robustness.

6.3 Model Problem

To conduct our numerical testing, we will focus on the following simpler BVP:

$$\begin{aligned} -\nabla \cdot (a \nabla u) &= f \;\text{ in }\; \Omega \\ u &= 0 \;\text{ on }\; \partial\Omega \end{aligned} \tag{6.1}$$

with $\Omega = (0,1) \times (0,1)$, and $\partial\Omega$ representing the boundary. Now define the space $V = \{v \in H^1(\Omega) : v = 0 \text{ on } \partial\Omega\}$. The variational form is defined as follows; Find $u \in V$ such that

$$T(a, u, v) = m(v) \quad \forall v \in V \tag{6.2}$$

where

$$\begin{aligned} T(a, u, v) &= \int_\Omega a \nabla u \cdot \nabla v \\ m(v) &= \int_\Omega f \cdot v \end{aligned}$$

We assume that constants, $\alpha, \beta > 0$ exist, such that

$$T(a, u, v) \leq \alpha ||u|| ||v||$$

$$T(a, u, v) \geq \beta ||u||^2,$$

so that by the Lax-Milgram Lemma, the weak form is uniquely solvable. The existence and uniqueness of a solution to (6.2) can also be attained through the Reisz representation theorem, and the assumption that a is bounded, and positive. It is trivial to see that $T(\cdot, \cdot, \cdot)$ is a trilinear map $T : A \times V \times V \to \mathbb{R}$, and is symmetric in its last two arguments. A is a nonempty, closed, and convex subset of a Banach space, B. A is the set of feasible coefficients. $m : V \to \mathbb{R}$ is a continuous linear map.

6.4 Optimization Formulations

6.4.1 Output Least Squares

As with the elasticity problem, optimizing the Output Least Squares (OLS) functional is the most common approach to solving inverse problems. It employs the same idea of minimizing the norm between the solution to the weak form, u, and some measurement of this solution, z.

$$J_{\text{OLS}}(a) = \frac{1}{2} ||u(a) - z||_V^2 \tag{6.3}$$

This functional is usually highly ill-posed and in need of regularization techniques to produce a well-posed version. This yields the following optimization problem:

$$\min_{a \in A} J_{\text{OLS}}(a) = \frac{1}{2}||u(a) - z||_V^2 + \kappa R(a) \tag{6.4}$$

where R is the regularization functional, and $\kappa > 0$ is the regularization parameter.
For the computation of the first derivative of the scalar OLS functional, we use an adjoint method in order to avoid explicit computation of the derivative of the solution map. The adjoint method depends on the following results

$$T(\delta a, u, v) = -T(a, \delta u, v) \tag{6.5}$$

Now, Let $w \in V$ solving the following

$$T(a, w, v) = \langle z - u, v \rangle \quad \forall v \in V. \tag{6.6}$$

Thus,

$$DJ_{\text{OLS}}(a)(\delta a) = \kappa DR(a)(\delta) + T(\delta a, u, w) \tag{6.7}$$

To compute the first order derivative for the regularized OLS functional, first compute $u(a)$ by solving (6.2), and then w by solving (6.6). Finally, the first order derivative, $DJ_{\text{OLS}}(a)(\delta a)$, can now be computed.
Proceeding as before, we have that

$$D^2 J_{\text{OLS}}(a)(\delta a, \delta a) = \kappa D^2 R(a)(\delta a, \delta a) + \langle \delta u, \delta u \rangle + 2T(\delta a, \delta u, w) \tag{6.8}$$

Thus to compute the second order derivative of the regularized OLS functional for the scalar problem, we compute u, the usual way by solving the weak formulation, then compute w using (6.6), and then δu, by solving (6.5). Now we have all we need to compute $D^2 J_{\text{OLS}}(a)(\delta a, \delta a)$ using (6.8)

6.4.2 Modified Output Least Squares

The Modified Output Least Squares, (MOLS), for the scalar problem produces an even simpler functional to work with because of the simplicity of the weak form of the scalar problem. It addresses the same issues as in the elasticity problem.
The MOLS objective functional uses the weak form of the system as a guide, and is as follows:

$$J_{\text{MOLS}}(a) = \frac{1}{2} T(a, u(a) - z, u(a) - z) \tag{6.9}$$

where all variables are the same as in the case of the OLS scheme. Also, as with the OLS, this functional is susceptible to ill-posedness and so regularization of some sort is introduced to combat this problem, and provide numerical and computational stability.
The computation of the first order derivative of the MOLS functional can be done directly without worry of the computation of the derivative of the solution map u. Using the adjoint trick in (6.5)

$$DJ_{\text{MOLS}}(a)(\delta a) = -\frac{1}{2}T(\delta a, u(a)+z, u(a)-z) + \kappa DR(a)(\delta)$$
$$D^2 J_{\text{MOLS}}(a)(\delta a, \delta a) = T(a, \delta u, \delta u) + \kappa D^2 R(a)(\delta a, \delta a)$$

Where δu is gotten by solving (6.5)

6.4.3 Equation Error

The EE functional is defined as follows

$$J_{\text{EE}}(a) = \frac{1}{2}||e(a,z)||_V^2, \tag{6.10}$$

with $e(a,u) \in V$ such that

$$\langle e(a,u), v\rangle = T(a,u,v) - m(v) \tag{6.11}$$

As with every other functional, this functional also requires a level of regularization for computational stability.
An advantage of this scheme is that it is uniquely solvable in both its continuous, and discrete form. It also produces a convex functional, so a minimizer is guaranteed to be found. Lastly, this scheme is computationally inexpensive compared to the other forms of the Output Least Squares schemes, as there are no underlying variational problems to be solved. The drawback of this method is that it relies on differentiating the data entered into the system, and as a result, noise or errors in the data can cause the solution to be rather inaccurate. In other words, this method is not very robust, and is very sensitive to noise in the data.
We have

$$J_{\text{EE}}(a) = \frac{1}{2}\langle e(a,z), e(a,z)\rangle$$

This allows for the derivative of the functional to be

$$DJ_{\text{EE}}(a)(\delta a) = \langle e(a,z)\delta a, e_t(a,z)\rangle + \kappa DR(a)(\delta) \tag{6.12}$$

where

$$\langle e_t(a,z), v\rangle = T(a,z,v) \quad \forall v \in V$$

Using the knowledge applied to compute the first order derivative of the EE functional, we can compute the second order derivative from (6.12) as

$$D^2 J_{\mathrm{EE}}(a)(\delta a, \delta a) = \langle e_t(a,z)\delta a, e_t(a,z)\delta a\rangle + \kappa D^2 R(a)(\delta a, \delta a).$$

The discrete counterparts of the above functionals can be obtained in completely analogous fashion.

6.5 Numerical Experiments

In this section, we explore three representative example of the scalar problem

$$\begin{aligned} -\nabla \cdot (a \nabla u) &= f \ \text{ in } \ \Omega \\ u &= 0 \ \text{ on } \ \partial\Omega \end{aligned} \tag{6.13}$$

where $\Omega = (0,1) \times (0,1)$, and $\partial\Omega$ represents the boundary.
We solve the inverse problem on a 15×15 quadrangular mesh, with 1089 degrees of freedom.
The stopping criteria is the same as with the elasticity example

$$||\nabla J(a)|| \leq 10^{-12}.$$

For the scalar problem, the following examples will be used to test the methods described in previous sections.

- Example 1

$$\begin{aligned} u &= xy(1-x)(1-y) \\ a &= 1 + xy^2 \\ f &= 1 - 6y + 2y^2 - 6x + 8xy + y^3 - 12xy^3 - y^4 + 4xy^4 \\ &\quad + 5xy^2 - 15x^2y^2 + 14x^2y^3 + 2x^2 + 2x^2y + 6x^3y^2 \end{aligned}$$

- Example 2

$$\begin{aligned} u &= x + y - 2xy \\ a &= 2 + sin(2\pi x)sin(2\pi y) \\ f &= 8 - 4\pi cos(2\pi x)sin(2\pi y)(1-x-y) - 4\pi sin(2\pi x)cos(2\pi y) \\ &\quad (1-x-y) + 4sin(2\pi x)sin(2\pi y) \end{aligned}$$

- Example 3

$$\begin{aligned}
u &= 2xy + x^2y - 3x \\
a &= 2 + sin(2\pi xy) \\
f &= -(4y + 8 + 8x + 2\pi cos(2\pi xy)(4xy + 3yx^2 + 2y^2 \\
&\quad + 2xy^2 + 2x^2 + x^3 - 3y - 3x) + 2sin(2\pi xy)(y + 2 + 2x))
\end{aligned}$$

Here we will report the results of extensive experimentation done on the aforementioned scalar examples with respect to noise levels, and the diffential equation techniques previously mentioned.
For completion, we report results using the new second order derivative computations of the OLS functional, as well as the HBF technique using the MOLS, and EE functionals, using different differential equations solvers. Lastly, we report the results from our noise study, using those differential equation solvers.
In the following tables, all errors recorded are the L_2 Errors, CG represent the continuous newton-type first order method, HB represents the second order heavy ball with friction method, ε is the regularization parameter, α, and β are the parameters from the heavy ball with friction method, δ_1 represents the lower threshold for the continuous newton-type method. $\delta_2 = 1000\delta_1$. Iters stands for the number of iterations, Time is measured in seconds, and λ_{min} is the overall minimum eigen value recorded.
With regards to Hessian computation, OLS-H refers to the hybrid computation, and OLS-A refers to the second order adjoint computation.

6.5.1 Computations using OLS

We report the results using a second order algorithm to compare the hybrid, and second order adjoint approaches of Hessian computation. We applied a second order continuous Newton-type approach known as the Pseudo-transient continuation(PTC). This method is a way of implementing the continuous newton-type method and is discussed in [48].

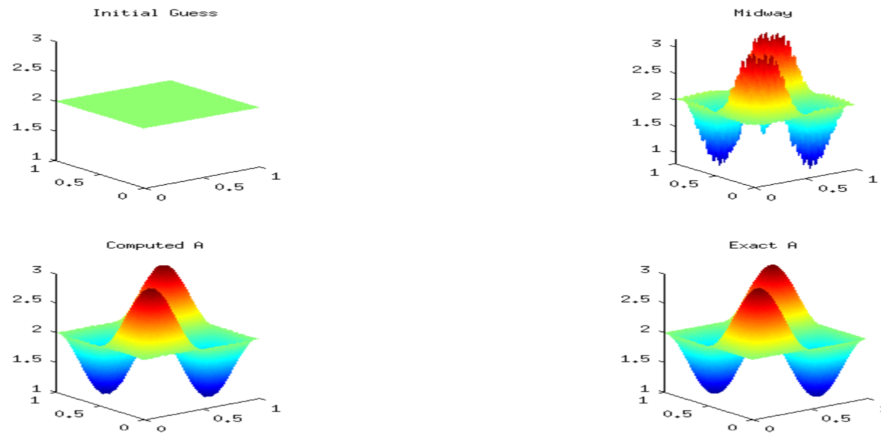

Figure 6.1: Example 2 PTC OLS Hybrid

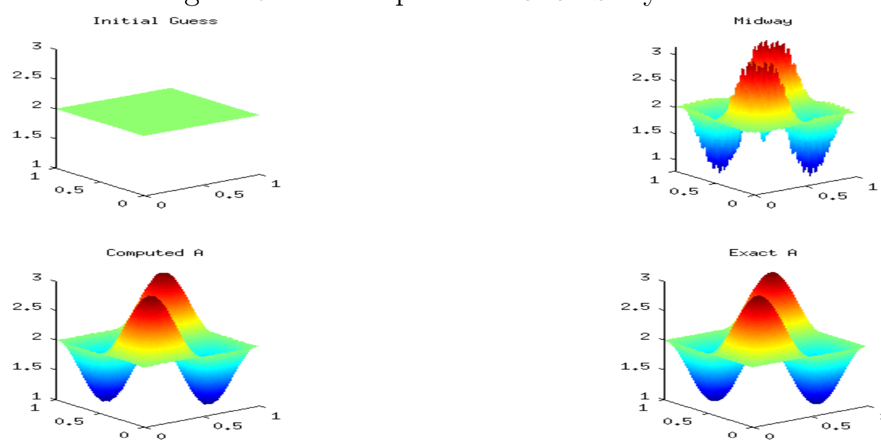

Figure 6.2: Example 2 PTC OLS Adjoint

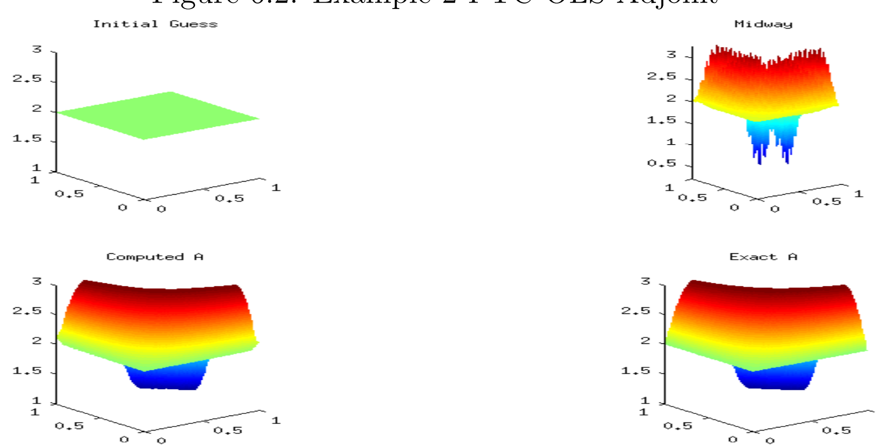

Figure 6.3: Example 3 PTC OLS Hybrid

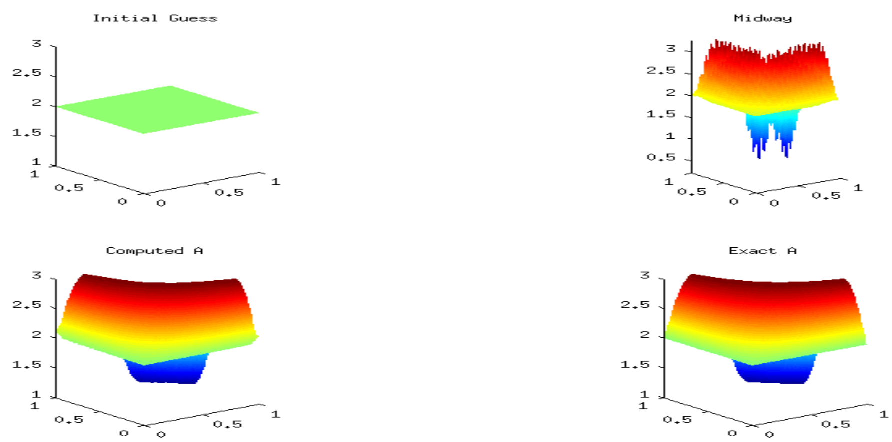

Figure 6.4: Example 3 PTC OLS Adjoint

Table 6.1: Example 2 OLS PTC

	Method	ε	Error	Time	iters
2	OLS-H	10^{-10}	0.0095	740.3	14
	OLS-A	10^{-9}	0.0098	1348.7	12
3	OLS-H	10^{-9}	0.0057	444.0	9
	OLS-A	10^{-9}	0.0057	924.7	8

6.5.2 Computations using MOLS

In this section, we report the results obtain using the Modified Output Least Square scheme to produce the objective functional that is being minimized. We compare the continuous newton-type method and the HBF method using several differential equation solvers.

Example 1 - Continuous Gradient

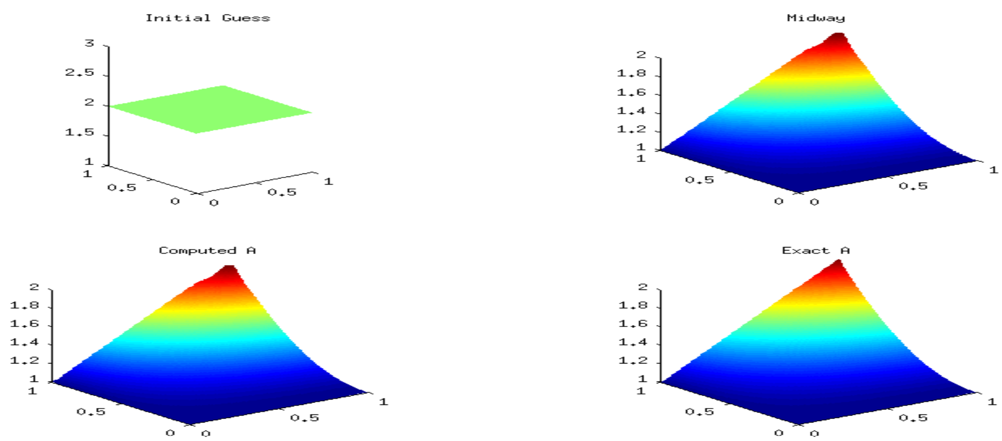

Figure 6.5: Example 1 Euler CG

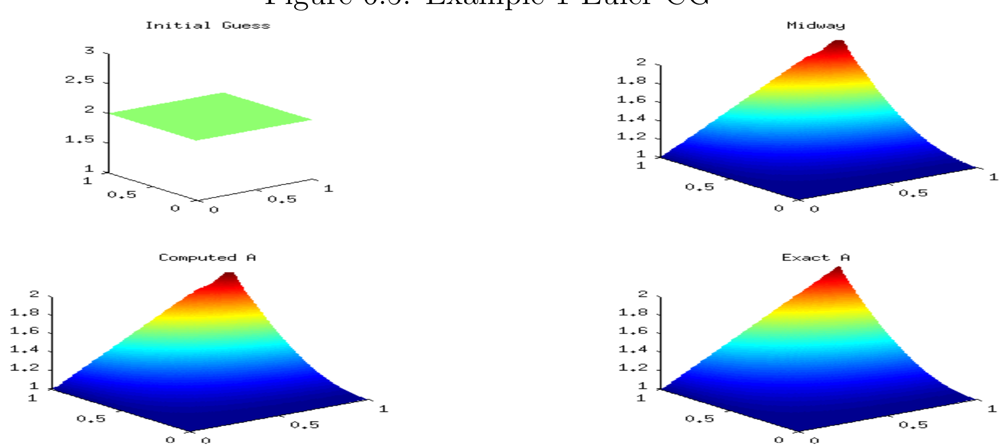

Figure 6.6: Example 1 Trap CG

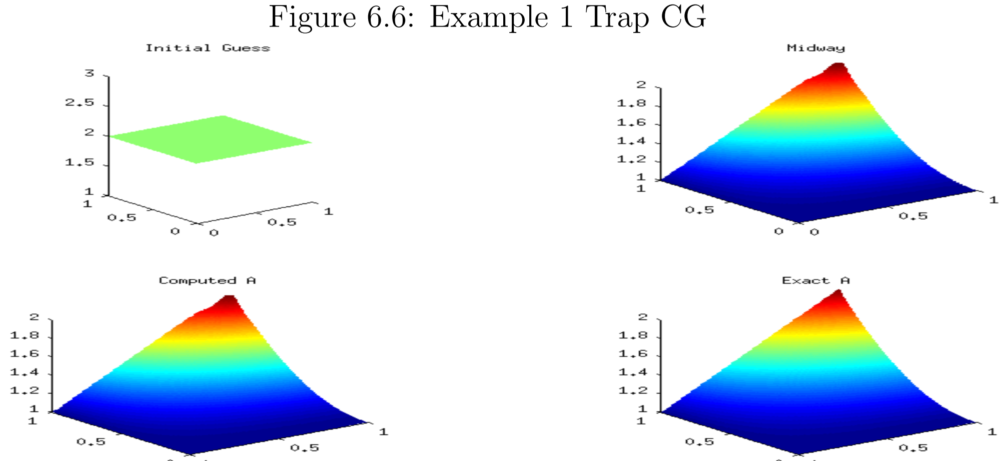

Figure 6.7: Example 1 RK CG

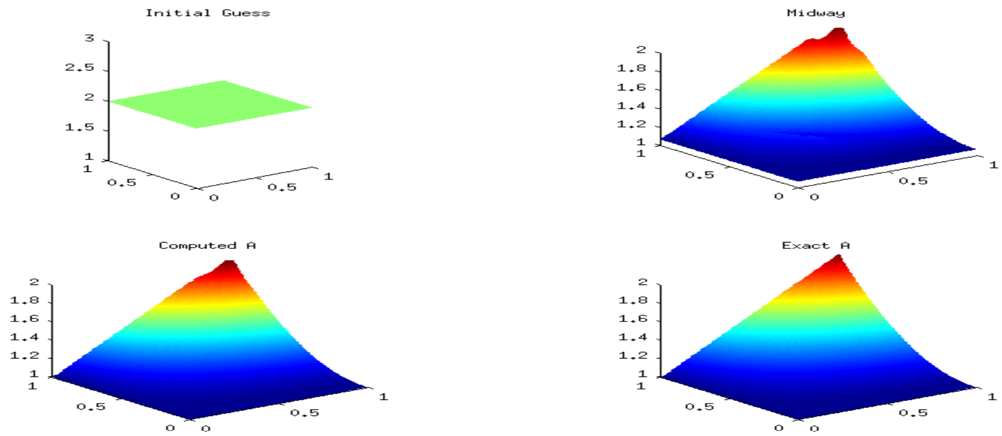

Figure 6.8: Example 1 ode113 CG

Table 6.2: MOLS Example 1 First Order ODE

Method	HC	MC	GC	ε	δ_1	Time(s)	iters	Error	λ_{min}
Euler	0	80	0	10^{-7}	10^{-6}	46.1	80	.0021	$4.01 \cdot 10^{-5}$
Trap	0	90	0	10^{-7}	10^{-6}	121.8	90	.0021	$4.01 \cdot 10^{-5}$
RK	0	88	0	10^{-7}	10^{-6}	230.6	88	.0021	$4.01 \cdot 10^{-5}$
ode113	0	102	0	10^{-7}	10^{-6}	121.5	102	.0021	$4.01 \cdot 10^{-5}$

Heavy Ball with Friction method

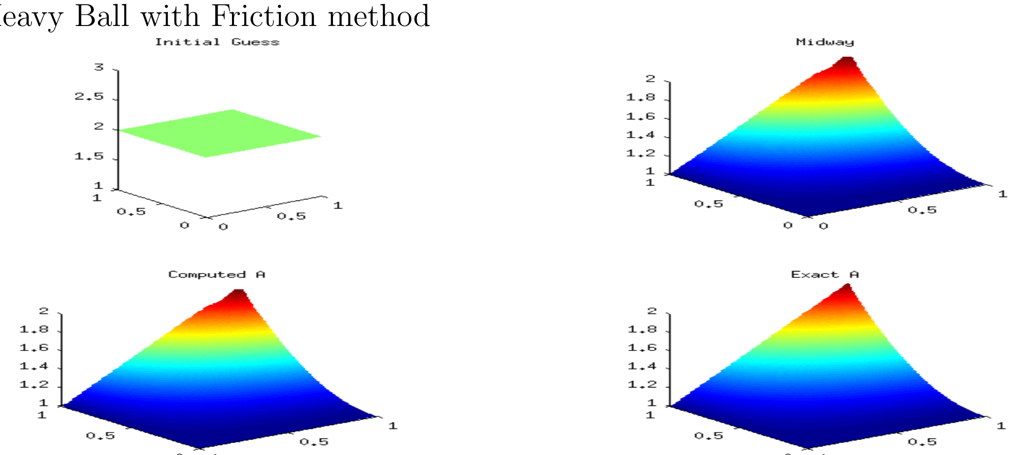

Figure 6.9: Example 1 Euler HB

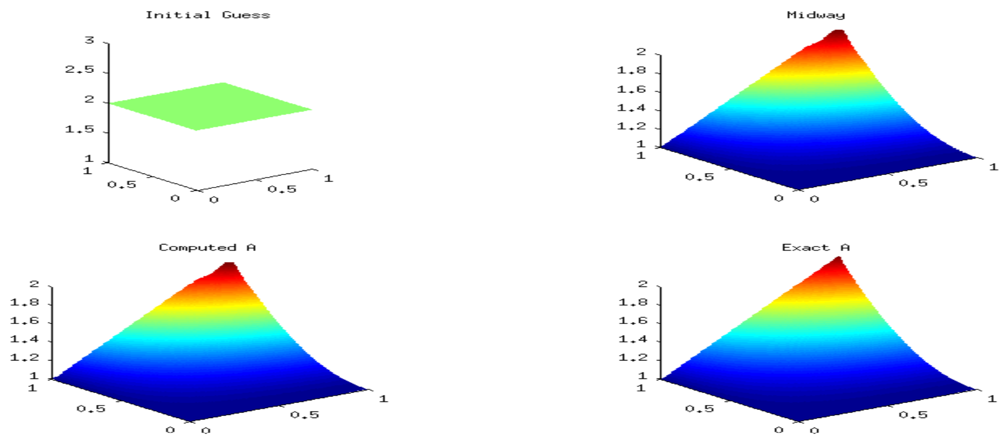

Figure 6.10: Example 1 Trap HB

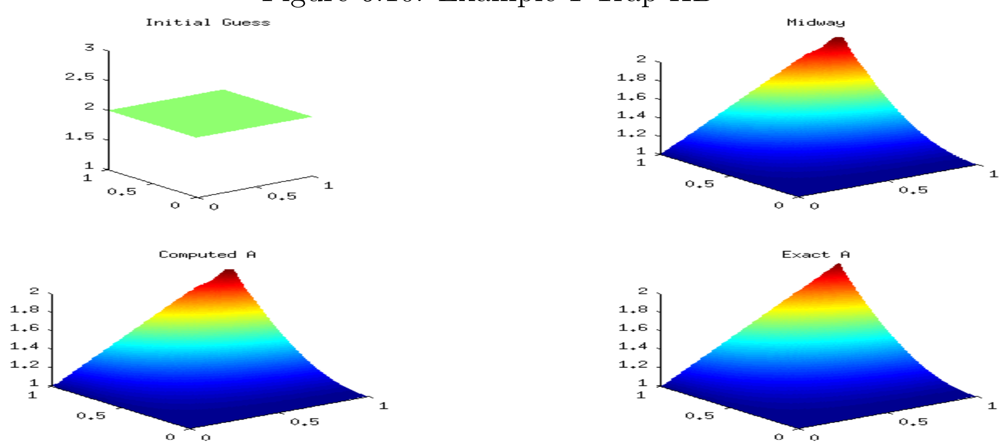

Figure 6.11: Example 1 RK HB

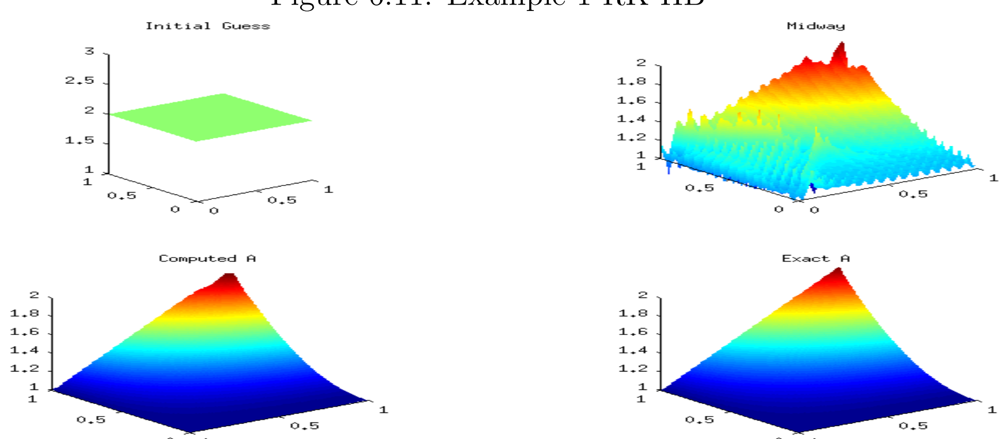

Figure 6.12: Example 1 ode113 HB

Table 6.3: MOLS Example 1 Second Order ODE

Method	HC	MC	GC	α	β	δ_1	Time(s)	iters	λ_{min}
Euler	52	6	0	0.6	0.1	10^{-7}	34.8	58	$3.89 \cdot 10^{-5}$
Trap	57	6	0	0.6	0.11	10^{-7}	81.9	63	$4.01 \cdot 10^{-5}$
RK	54	5	0	0.6	0.11	10^{-7}	153.8	59	$4.08 \cdot 10^{-5}$
ode113	214	0	0	1.9	0.9	10^{-7}	227.4	214	$1.28 \cdot 10^{-4}$

Example 2 - Continuous Gradient

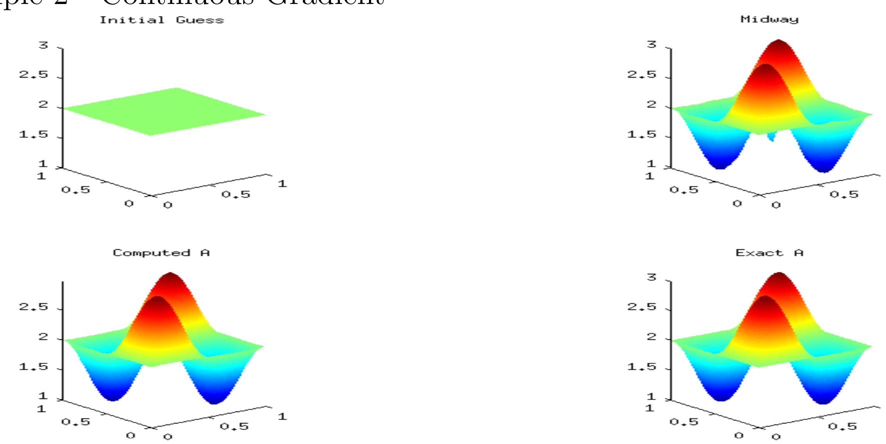

Figure 6.13: Example 2 Euler CG

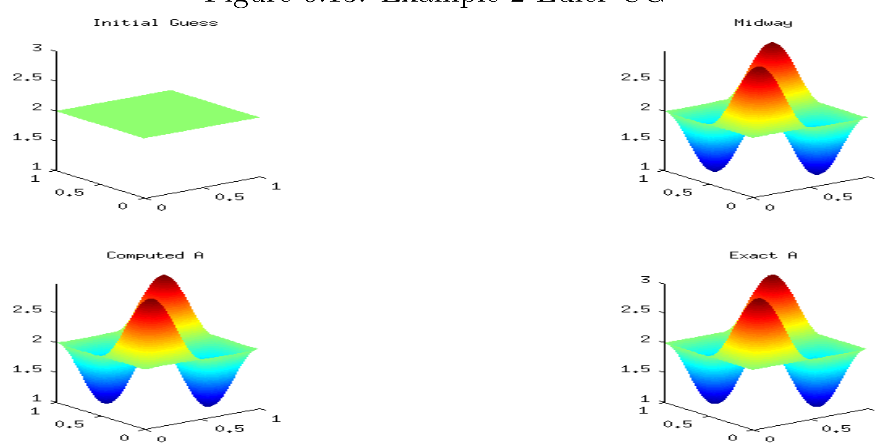

Figure 6.14: Example 2 Trap CG

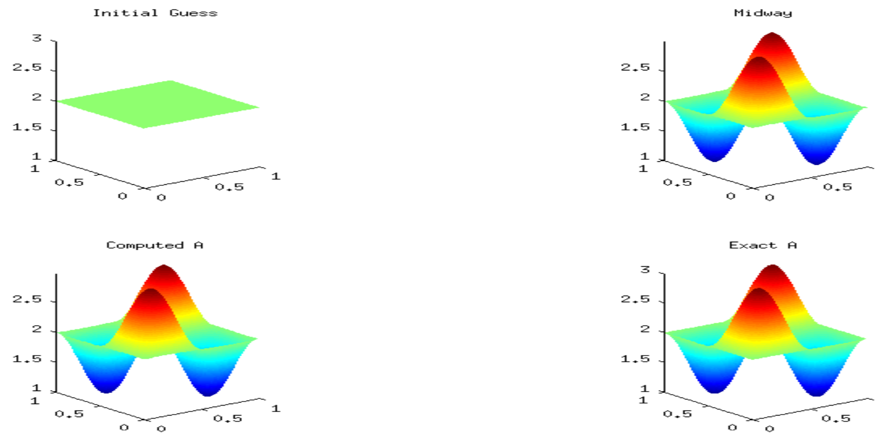

Figure 6.15: Example 2 RK CG

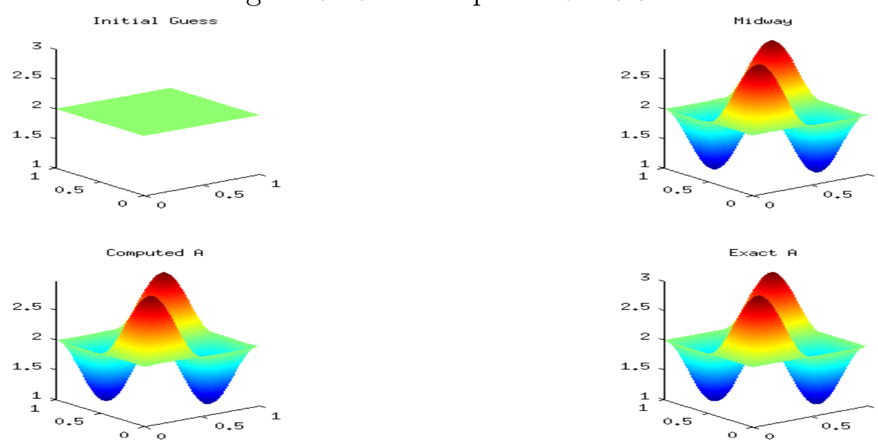

Figure 6.16: Example 2 ode113 CG

Table 6.4: MOLS Example 2 First Order ODE

Method	HC	MC	GC	ε	δ_1	Time(s)	iters	Error	λ_{min}
Euler	8	0	0	10^{-6}	10^{-7}	6.6	8	.0041	$9.10 \cdot 10^{-4}$
Trap	23	0	0	10^{-6}	10^{-7}	31.4	23	.0041	$9.04 \cdot 10^{-4}$
RK	18	0	0	10^{-6}	10^{-7}	51.9	18	.0041	$9.07 \cdot 10^{-4}$
ode113	146	0	0	10^{-6}	10^{-7}	205.9	146	.0041	$9.07 \cdot 10^{-4}$

Heavy Ball with Friction method

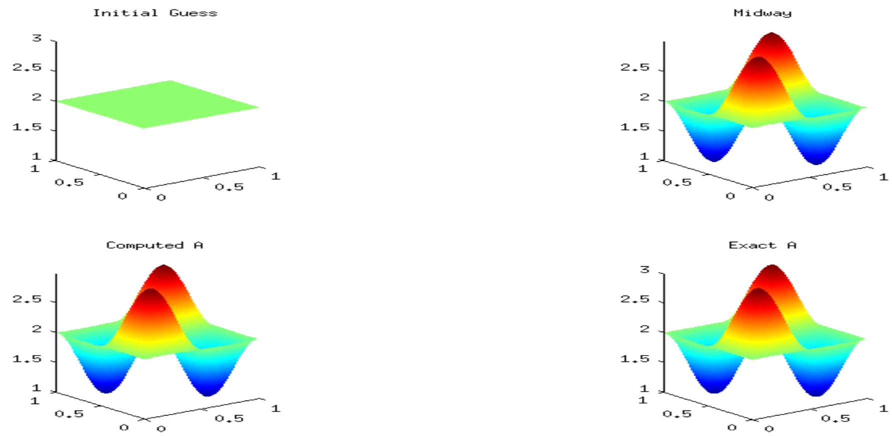

Figure 6.17: Example 2 Euler HB

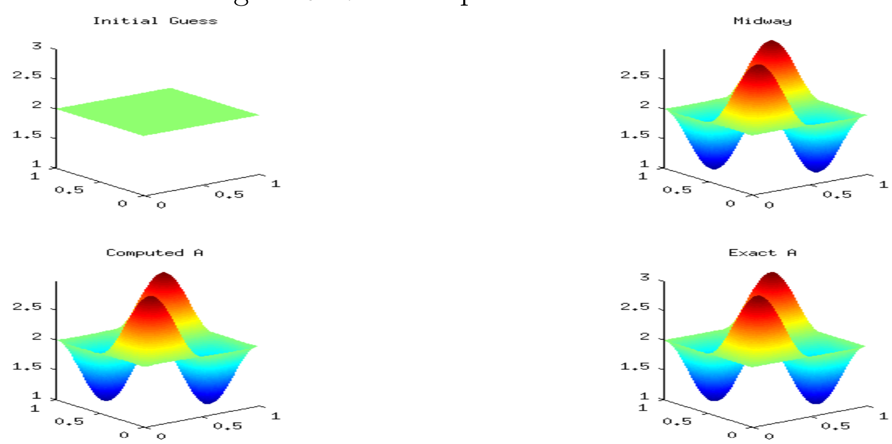

Figure 6.18: Example 2 Trap HB

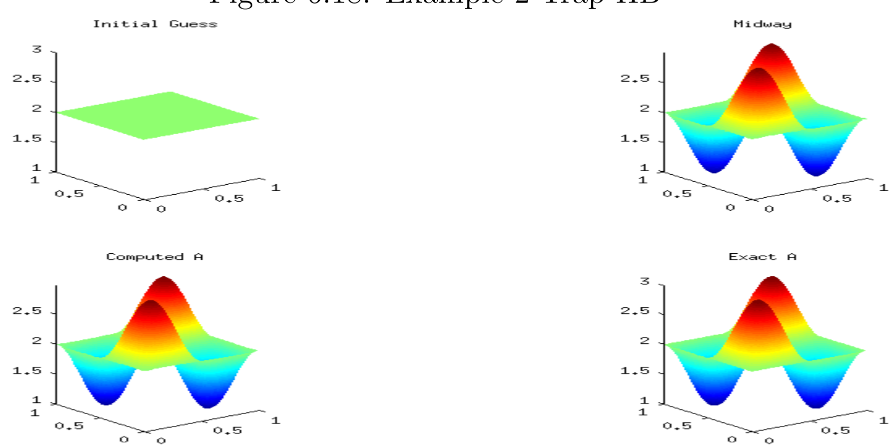

Figure 6.19: Example 2 RK HB

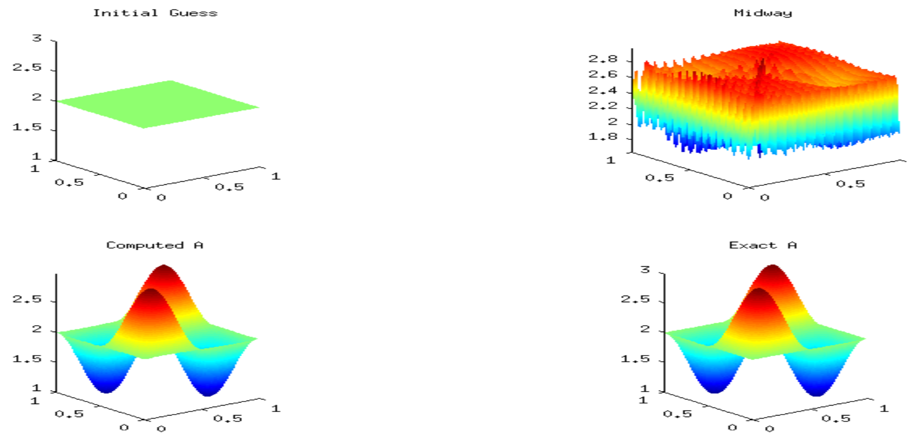

Figure 6.20: Example 2 ode113 HB

Table 6.5: MOLS Example 2 Second Order ODE

Method	HC	MC	GC	α	β	δ_1	Time(s)	iters	λ_{min}
Euler	1	14	0	1.9	.99	10^{-6}	10.8	15	$8.85 \cdot 10^{-4}$
Trap	1	8	0	2	2.1	10^{-6}	14.8	9	$9.11 \cdot 10^{-4}$
RK	0	14	0	2.1	2.1	10^{-6}	39.3	14	$8.98 \cdot 10^{-4}$
ode113	181	0	0	.59	0.1	10^{-7}	243.7	181	$4.34 \cdot 10^{-4}$

Example 3 - Continuous Gradient

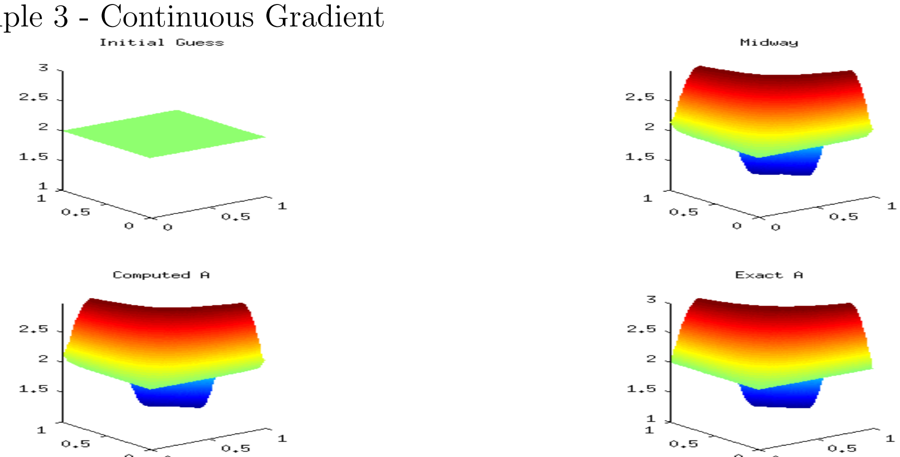

Figure 6.21: Example 3 Euler CG

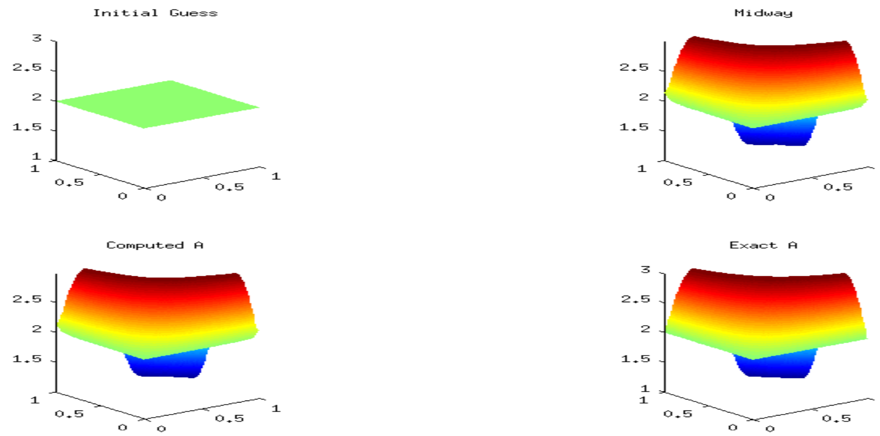

Figure 6.22: Example 3 Trap CG

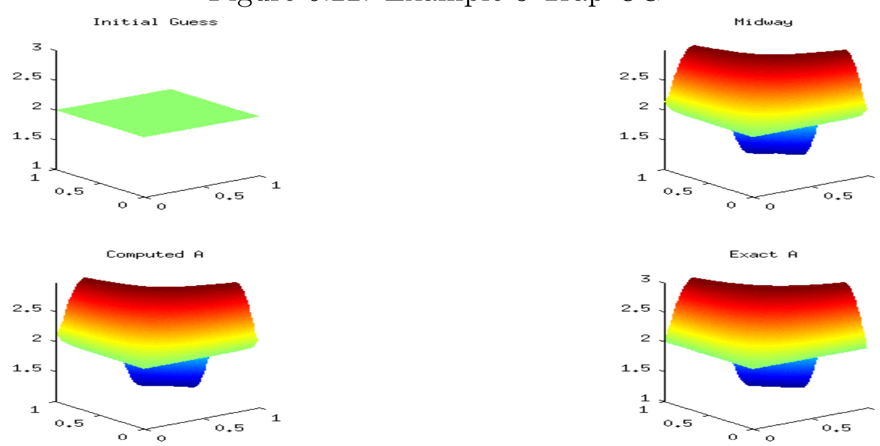

Figure 6.23: Example 3 RK CG

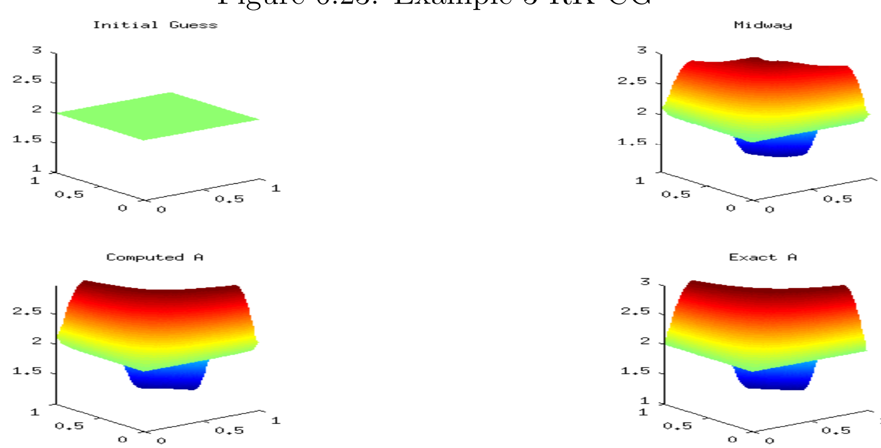

Figure 6.24: Example 2 ode113 CG

Table 6.6: MOLS Example 3 First Order ODE

Method	HC	MC	GC	ε	δ_1	Time(s)	iters	Error	λ_{min}
Euler	0	79	0	10^{-5}	10^{-5}	51.8	79	.0079	$1.7 \cdot 10^{-3}$
Trap	0	86	0	10^{-5}	10^{-5}	114.3	86	.0079	$1.7 \cdot 10^{-3}$
RK	0	84	0	10^{-5}	10^{-5}	224.3	84	.0079	$1.7 \cdot 10^{-3}$
ode113	0	113	0	10^{-5}	10^{-5}	160.7	113	.0079	$1.7 \cdot 10^{-3}$

Heavy Ball with Friction method

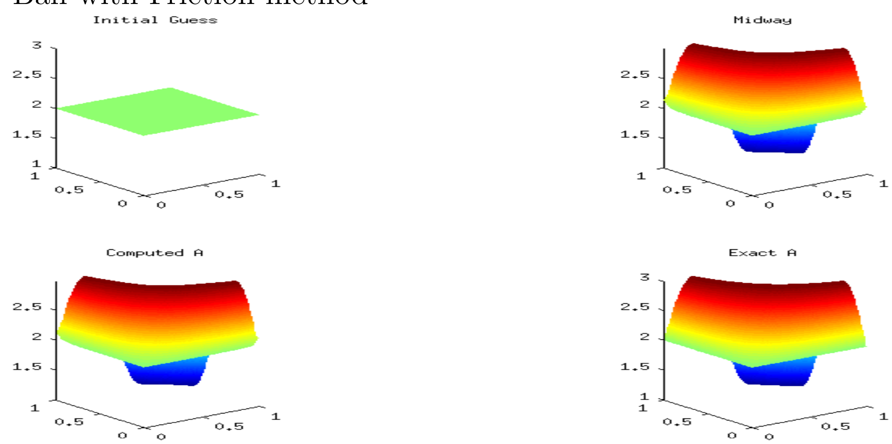

Figure 6.25: Example 3 Euler HB

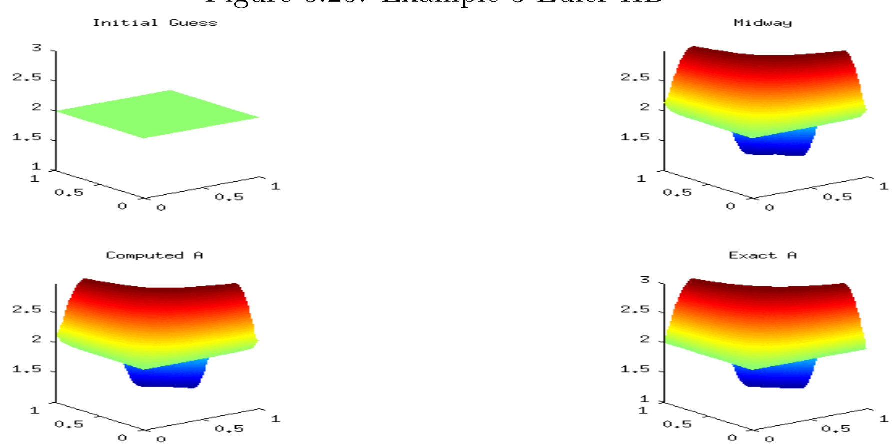

Figure 6.26: Example 3 Trap HB

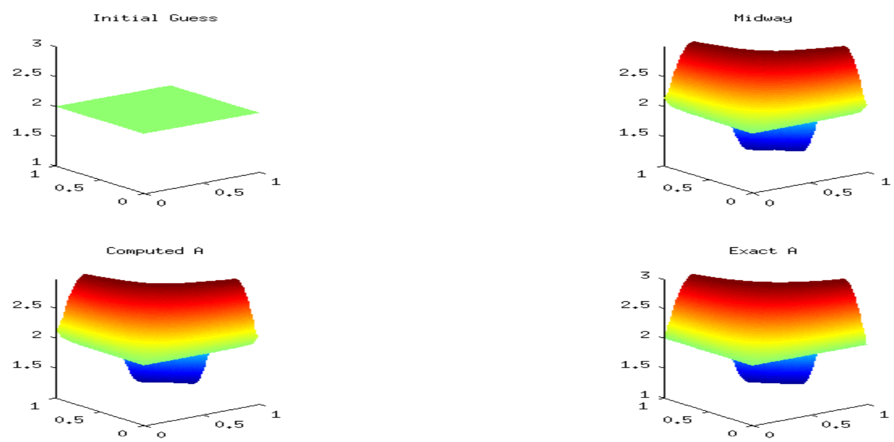

Figure 6.27: Example 3 RK HB

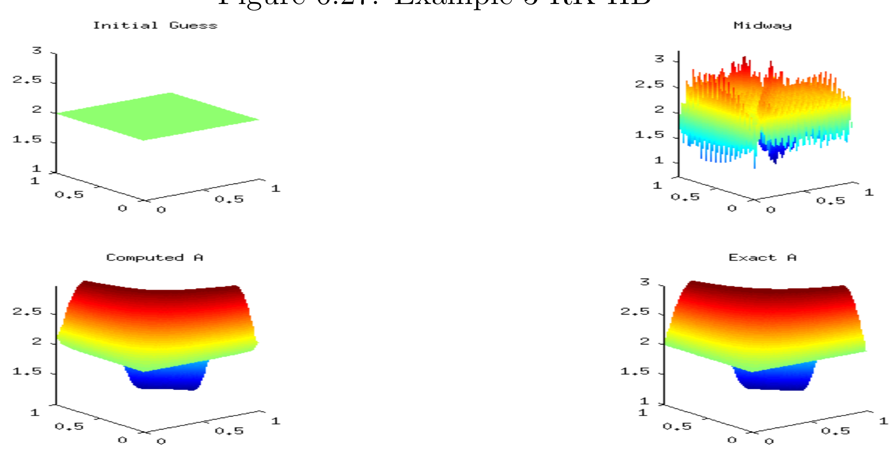

Figure 6.28: Example 3 ode113 HB

Table 6.7: MOLS Example 3 Second Order ODE

Method	HC	MC	GC	α	β	δ_1	Time(s)	iters	λ_{min}
Euler	0	49	0	2	2.7	10^{-5}	27.9	49	$1.9 \cdot 10^{-3}$
Trap	0	25	0	2	5.5	10^{-5}	34.4	25	$1.9 \cdot 10^{-3}$
RK	0	19	0	2	5.6	10^{-5}	60.4	19	$1.8 \cdot 10^{-3}$
ode113	61	185	0	1.2	2.2	10^{-5}	278.8	246	$4.34 \cdot 10^{-4}$

Note that errors were not recorded in the heavy ball tables, as they are maintained throughout the example.

6.5.3 Computations using Equation Error

In this section, we report the results obtained by using the Equation Error scheme to produce the objective functional. We compare the continuous newton-type method and the HBF method using several differential equation solvers.

Example 1 - Continuous Gradient

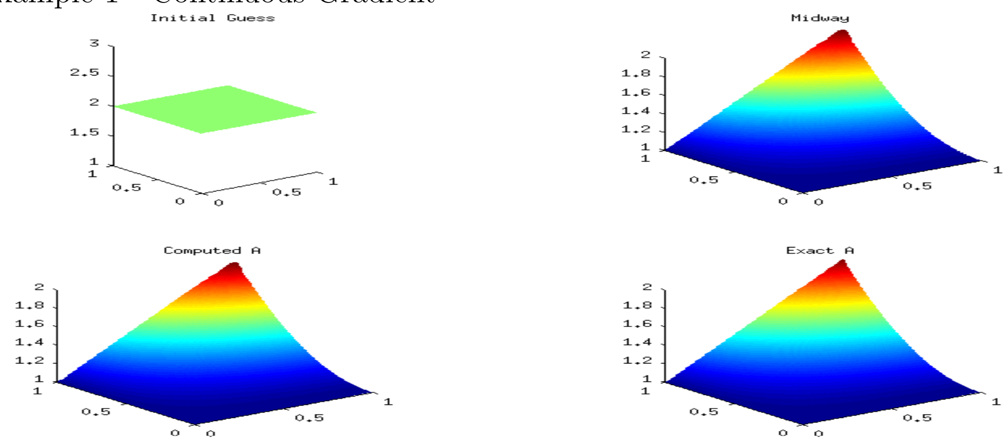

Figure 6.29: Example 1 Euler CG

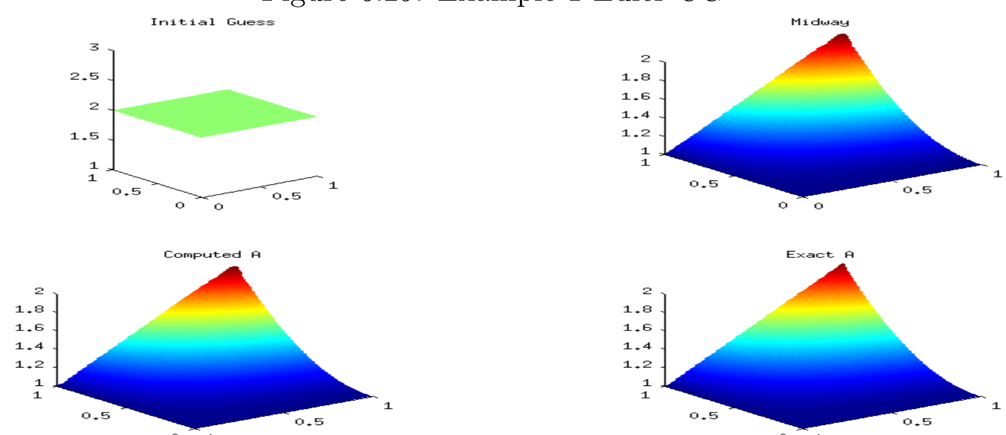

Figure 6.30: Example 1 Trap CG

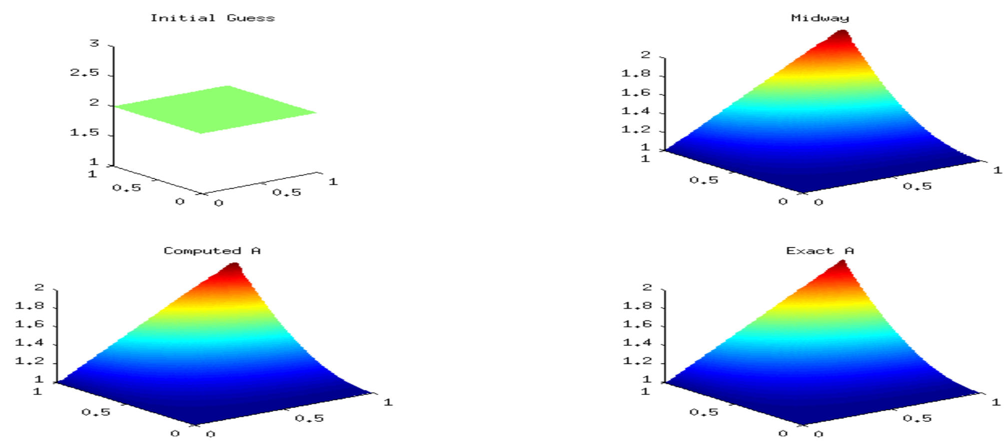

Figure 6.31: Example 1 RK CG

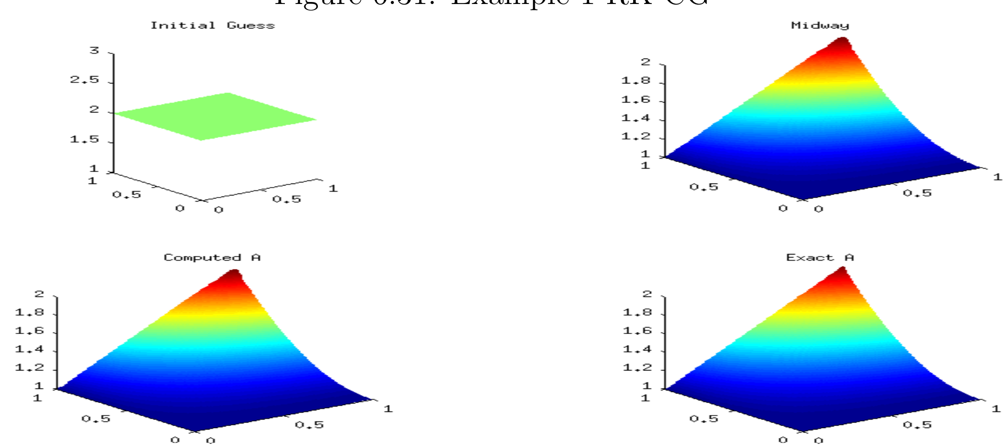

Figure 6.32: Example 1 ode113 CG

Table 6.8: EE Example 1 First Order ODE

Method	HC	MC	GC	ε	δ_1	Time(s)	iters	Error	λ_{min}
Euler	6	0	0	10^{-9}	10^{-9}	3.7	6	$7.40 \cdot 10^{-4}$	$2.41 \cdot 10^{-4}$
Trap	30	0	0	10^{-9}	10^{-9}	31.2	30	$7.40 \cdot 10^{-4}$	$2.41 \cdot 10^{-4}$
RK	21	0	0	10^{-9}	10^{-9}	52.0	21	$7.40 \cdot 10^{-4}$	$2.41 \cdot 10^{-4}$
ode113	123	0	0	10^{-9}	10^{-9}	140.1	123	$7.40 \cdot 10^{-4}$	$2.41 \cdot 10^{-4}$

Heavy Ball with Friction method

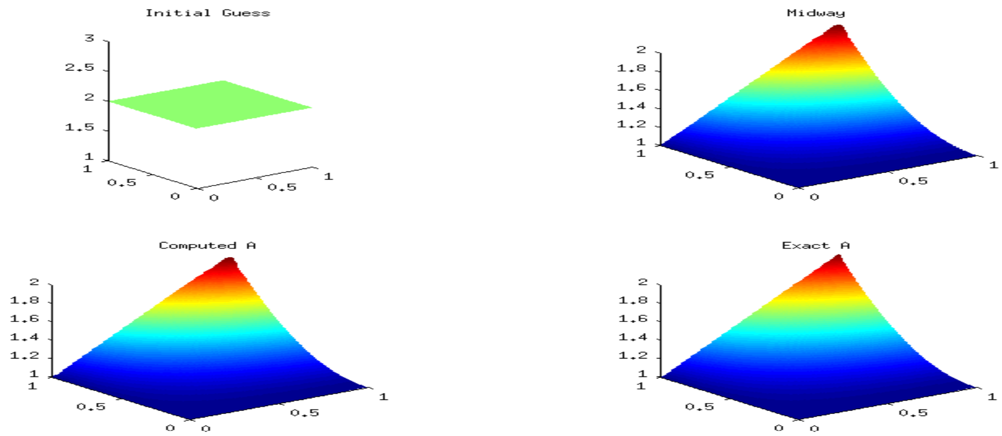

Figure 6.33: Example 1 Euler HB

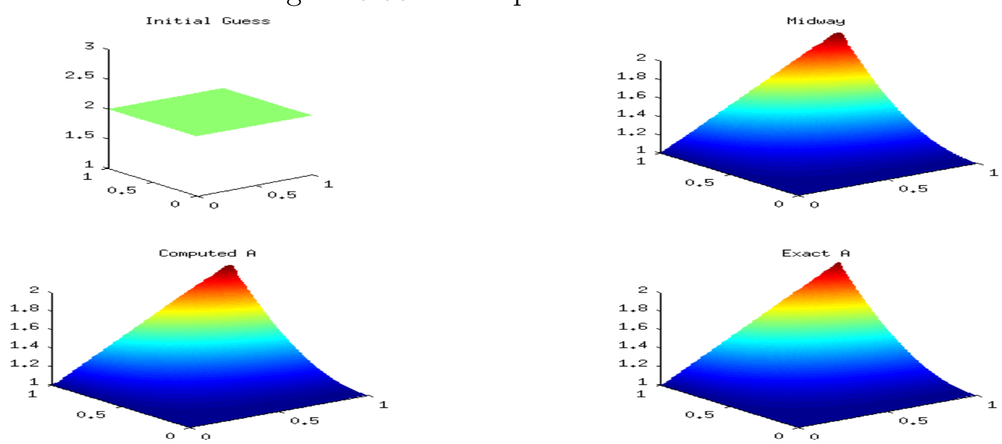

Figure 6.34: Example 1 Trap HB

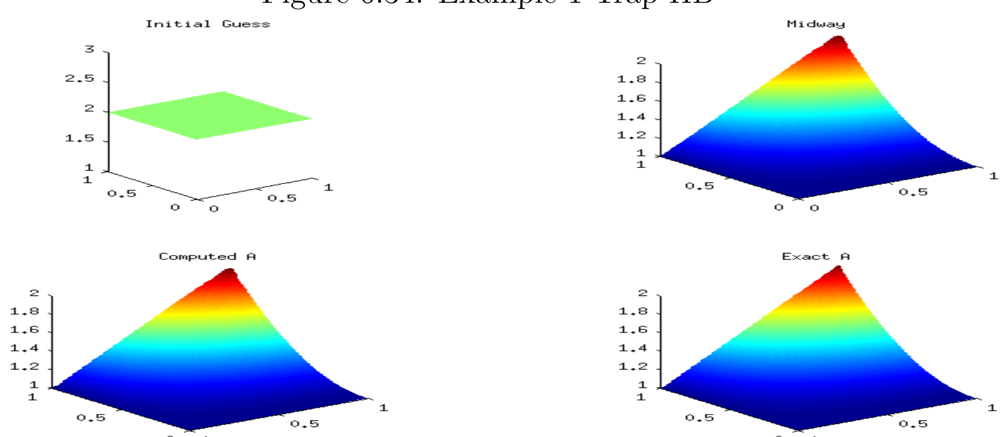

Figure 6.35: Example 1 RK HB

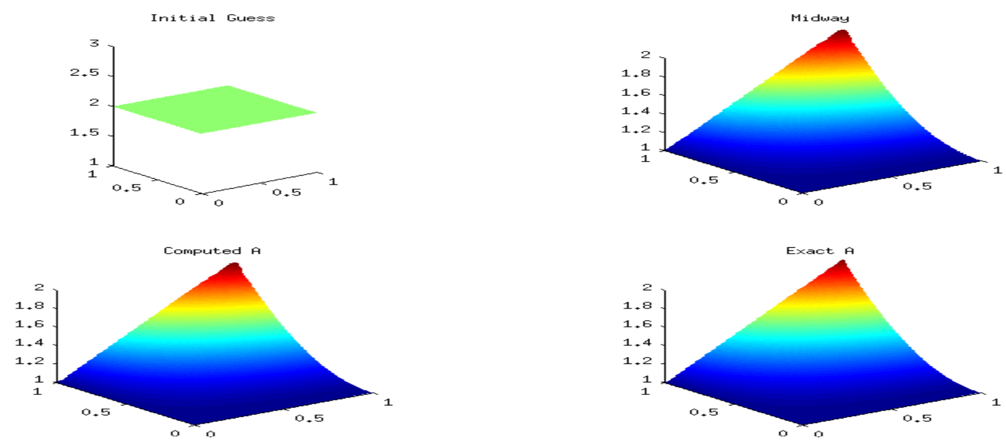

Figure 6.36: Example 1 ode113 HB

Table 6.9: EE Example 1 Second Order ODE

Method	HC	MC	GC	α	β	δ_1	Time(s)	iters	λ_{min}
Euler	7	0	0	2	1	10^{-9}	4.4	7	$2.41 \cdot 10^{-4}$
Trap	9	0	0	1.9	2	10^{-9}	10.9	9	$2.41 \cdot 10^{-4}$
RK	17	0	0	2	1.99	10^{-9}	48.5	17	$2.41 \cdot 10^{-4}$
ode113	107	0	0	.61	0.1	10^{-9}	81.1	107	$2.41 \cdot 10^{-4}$

Example 2 - Continuous Gradient

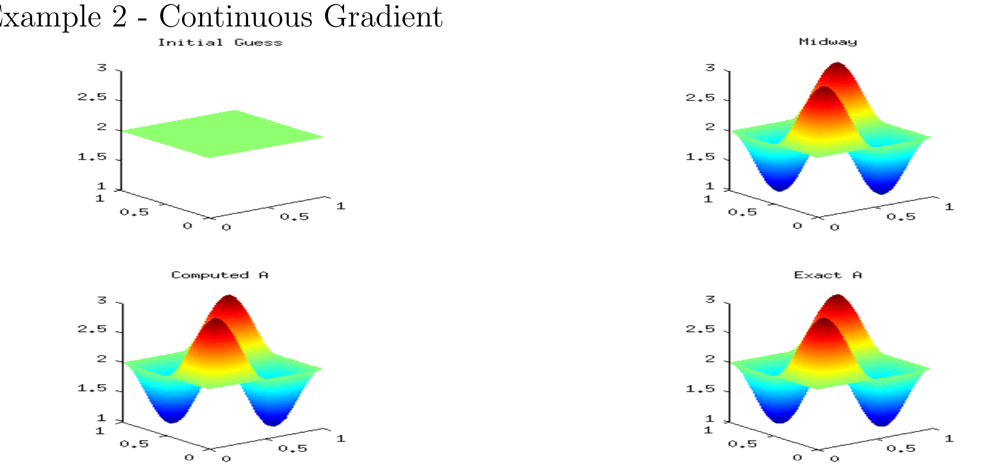

Figure 6.37: Example 2 Euler CG

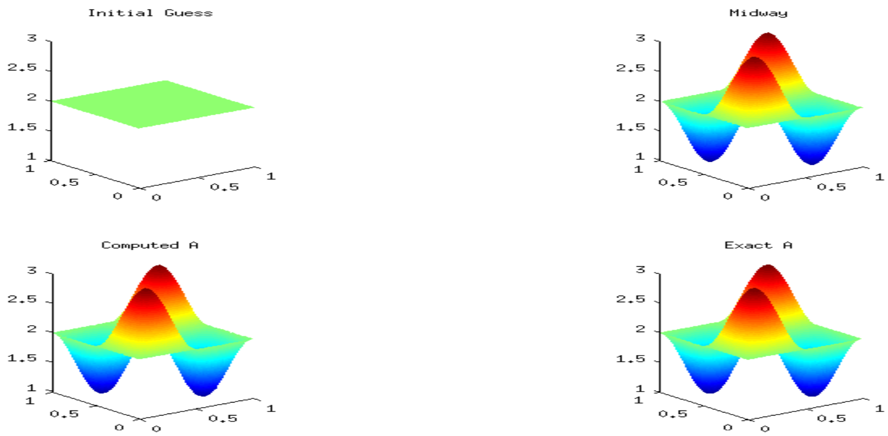

Figure 6.38: Example 2 Trap CG

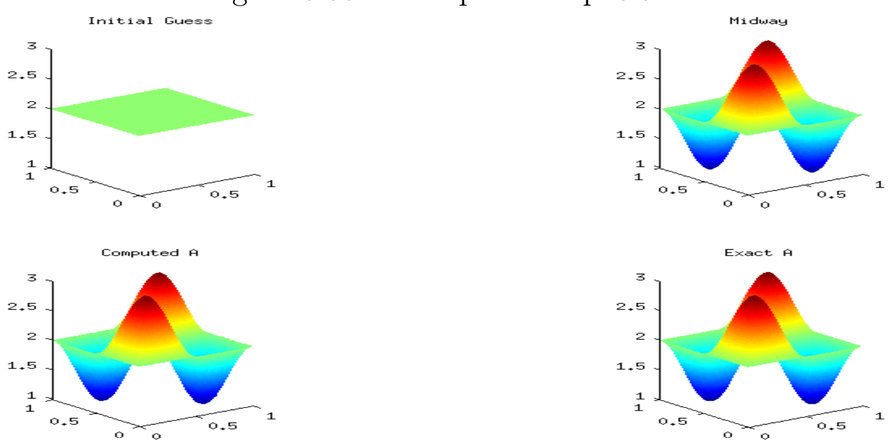

Figure 6.39: Example 2 RK CG

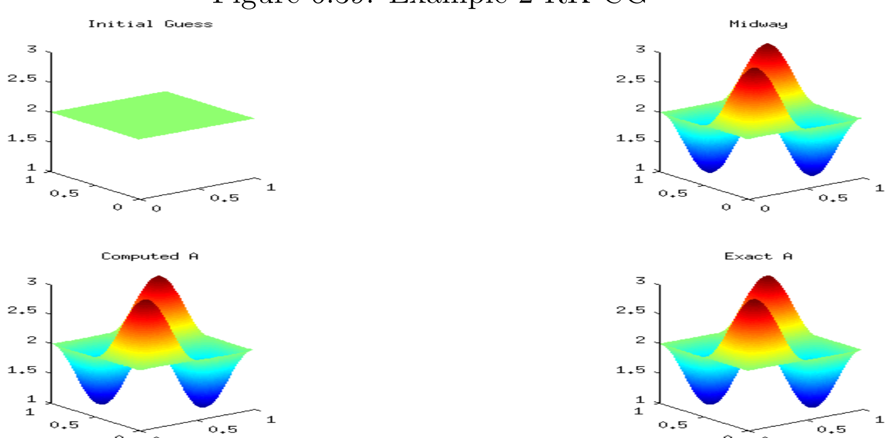

Figure 6.40: Example 2 ode113 CG

Table 6.10: EE Example 2 First Order ODE

Method	HC	MC	GC	ε	δ_1	Time(s)	iters	Error	λ_{min}
Euler	5	0	0	10^{-9}	10^{-9}	4.0	5	.0014	$1.8 \cdot 10^{-3}$
Trap	22	0	0	10^{-9}	10^{-9}	23.6	22	.0014	$1.8 \cdot 10^{-3}$
RK	15	0	0	10^{-9}	10^{-9}	39.4	15	.0014	$1.8 \cdot 10^{-3}$
ode113	121	0	0	10^{-9}	10^{-9}	158.4	121	.0014	$1.8 \cdot 10^{-3}$

Heavy Ball with Friction method

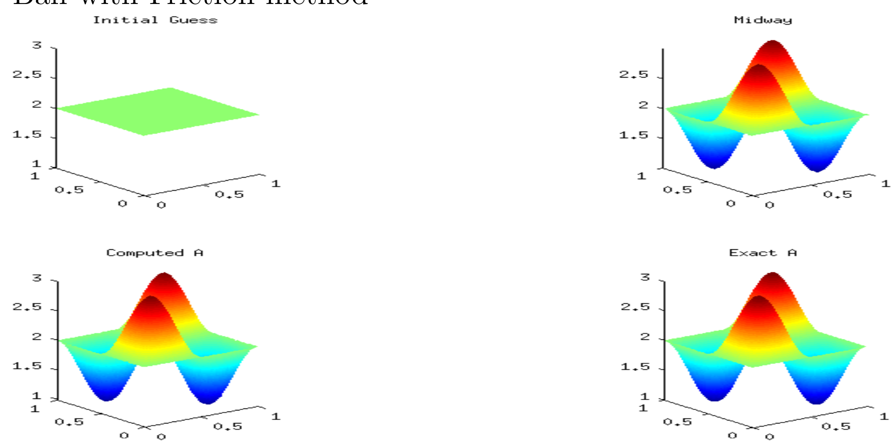

Figure 6.41: Example 2 Euler HB

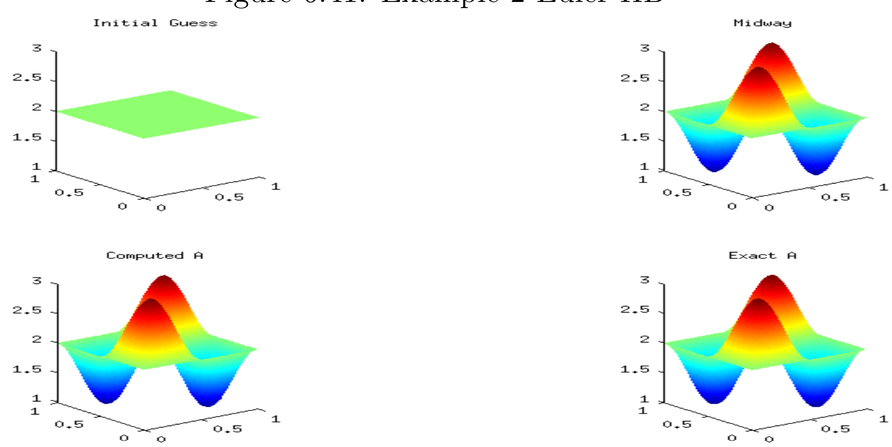

Figure 6.42: Example 2 Trap HB

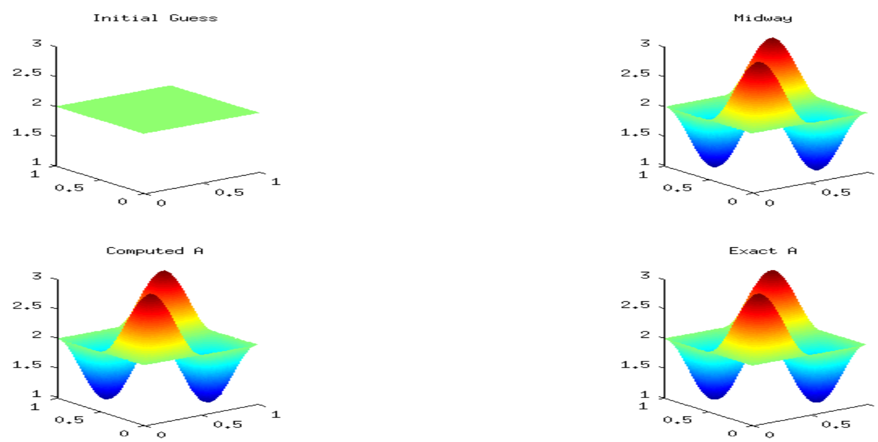

Figure 6.43: Example 2 RK HB

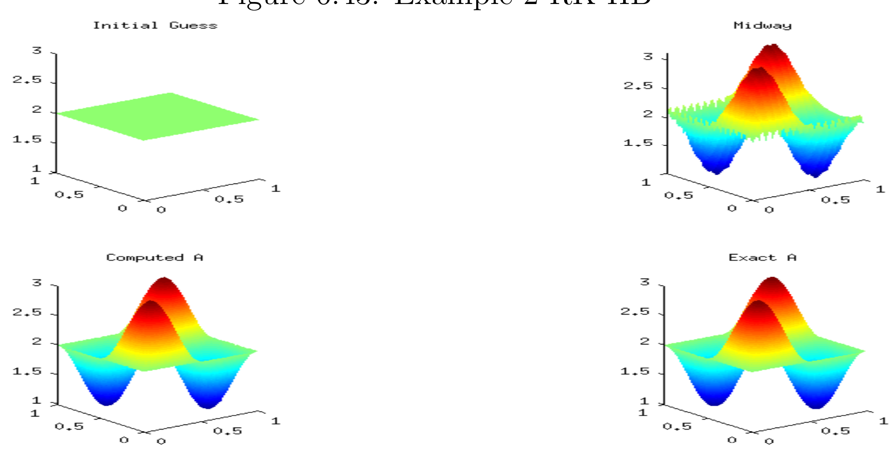

Figure 6.44: Example 2 ode113 HB

Table 6.11: EE Example 2 Second Order ODE

Method	HC	MC	GC	α	β	δ_1	Time(s)	iters	λ_{min}
Euler	5	0	0	2.02	1.02	10^{-9}	4.5	5	0.0018
Trap	7	0	0	2	2.1	10^{-9}	11.2	7	0.0018
RK	15	0	0	1.92	1.94	10^{-9}	39.7	15	0.0018
ode113	103	0	0	0.59	0.1	10^{-9}	145.7	103	0.0018

Example 3 - Continuous Gradient

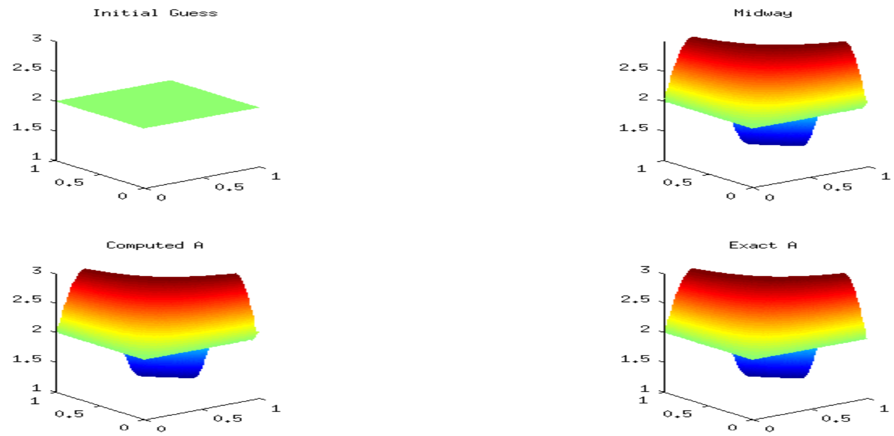

Figure 6.45: Example 2 Euler CG

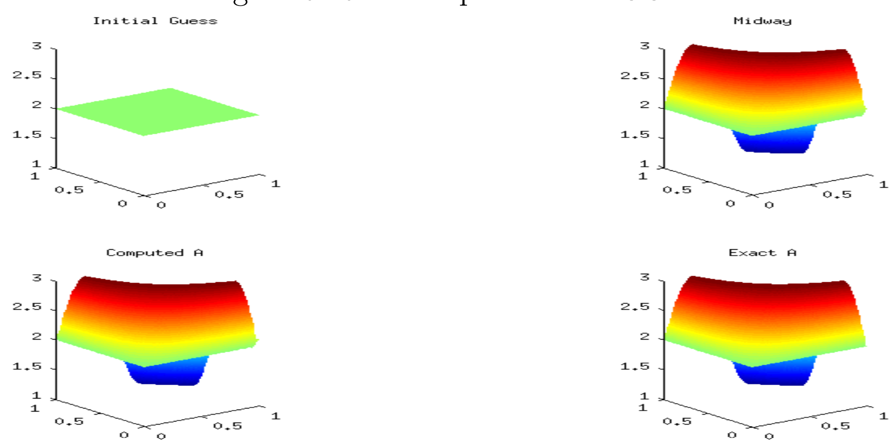

Figure 6.46: Example 3 Trap CG

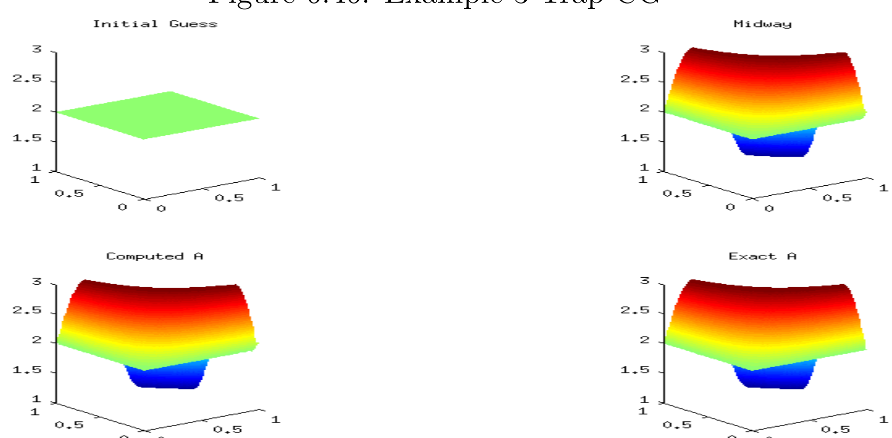

Figure 6.47: Example 3 RK CG

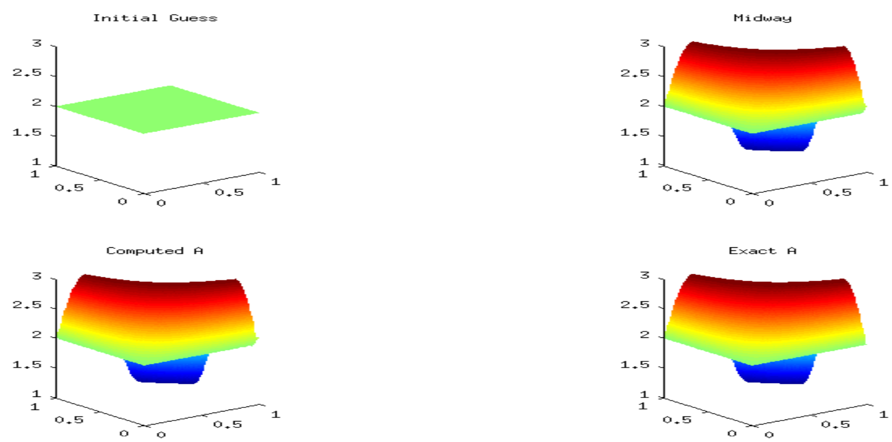

Figure 6.48: Example 3 ode113 CG

Table 6.12: EE Example 3 First Order ODE

Method	HC	MC	GC	ε	δ_1	Time(s)	iters	Error	λ_{min}
Euler	4	0	0	10^{-9}	10^{-9}	3.1	4	.0024	$3.4 \cdot 10^{-3}$
Trap	20	0	0	10^{-9}	10^{-9}	27.9	20	.0024	$3.4 \cdot 10^{-3}$
RK	14	0	0	10^{-9}	10^{-9}	32.2	14	.0024	$3.4 \cdot 10^{-3}$
ode113	114	0	0	10^{-9}	10^{-9}	141.0	114	.0024	$3.4 \cdot 10^{-3}$

Heavy Ball with Friction method

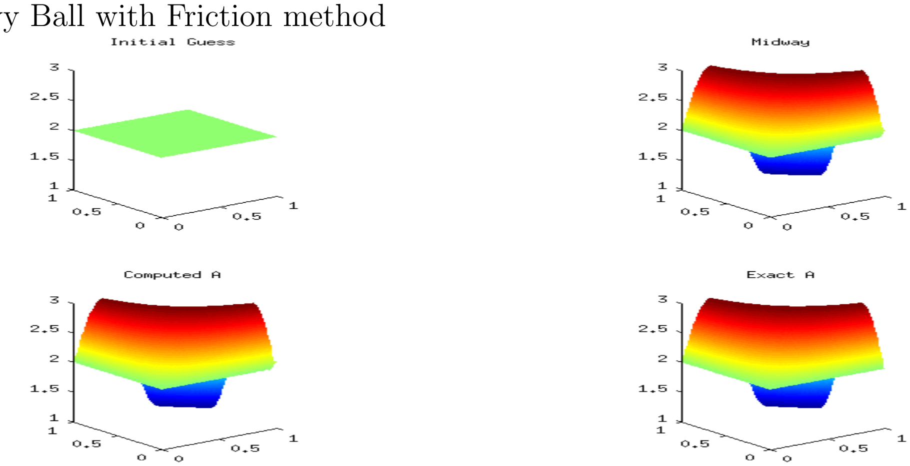

Figure 6.49: Example 3 Euler HB

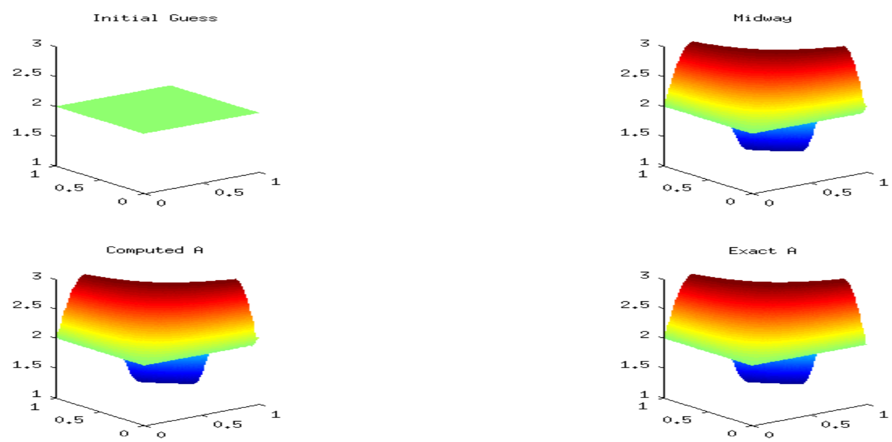

Figure 6.50: Example 3 Trap HB

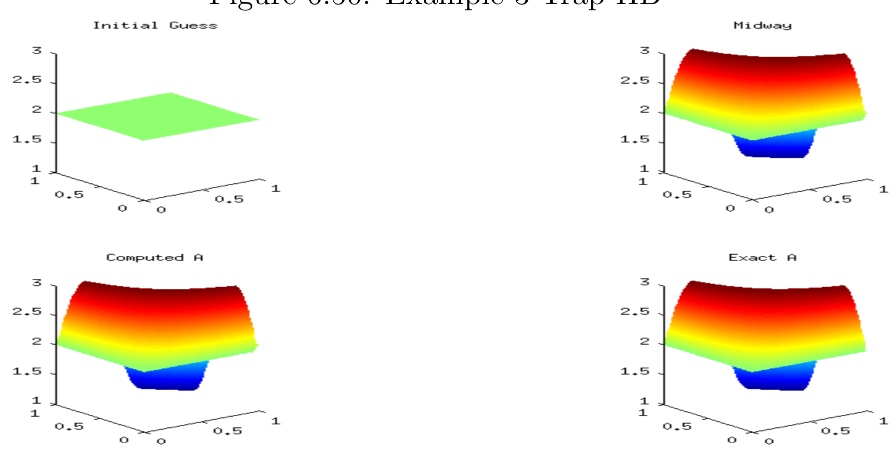

Figure 6.51: Example 3 RK HB

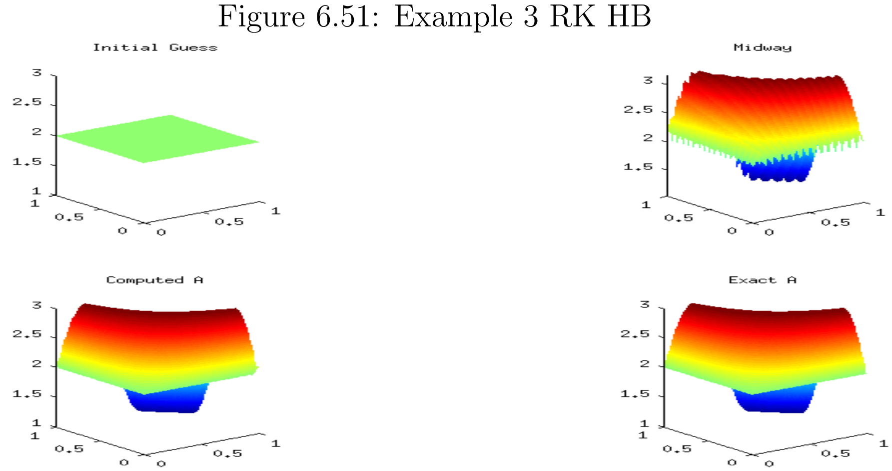

Figure 6.52: Example 3 ode113 HB

Table 6.13: EE Example 3 Second Order ODE

Method	HC	MC	GC	α	β	δ_1	Time(s)	iters	λ_{min}
Euler	5	0	0	2.02	1.02	10^{-9}	3.8	5	0.0034
Trap	5	0	0	2	2	10^{-9}	7.0	5	0.0034
RK	13	0	0	2.29	1.4	10^{-9}	35.1	13	0.0034
ode113	102	0	0	0.61	0.1	10^{-9}	87.6	102	0.0034

Note that errors were not recorded in the heavy ball tables, as they are maintained throughout each examples.

6.6 A Comparative Analysis of the Noisy Data

Here, we compare the MOLS and EE schemes using several differential equation techniques. Next, we compare the functionals themselves using the same differential equation solver. We add noise using a random number generator. First, we solve the direct problem, which produces a result with an amount of error. Next, we add a random number, with the appropriate magnitude of error, to each element of the solution of the direct problem. This becomes the noisy data that is used to solve the inverse problem.

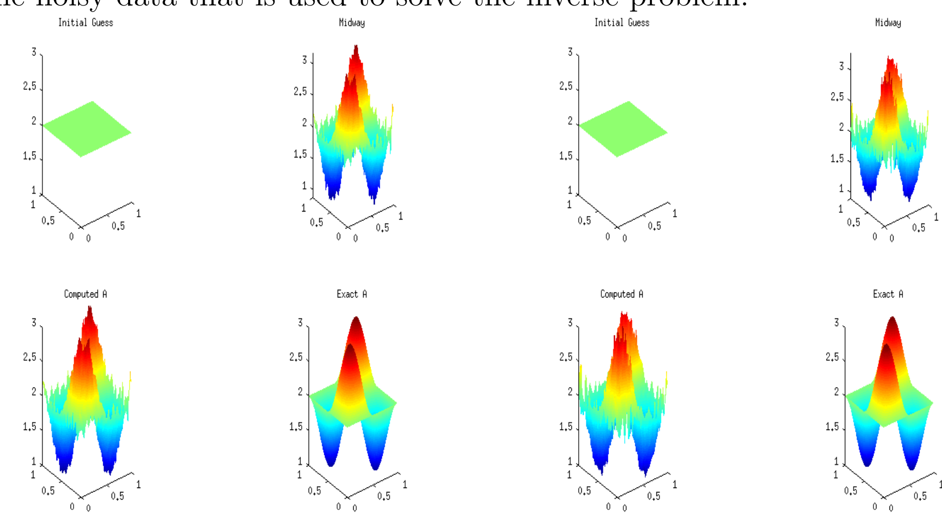

Figure 6.53: Example 2 MOLS vs EE Noise = 10^{-3}

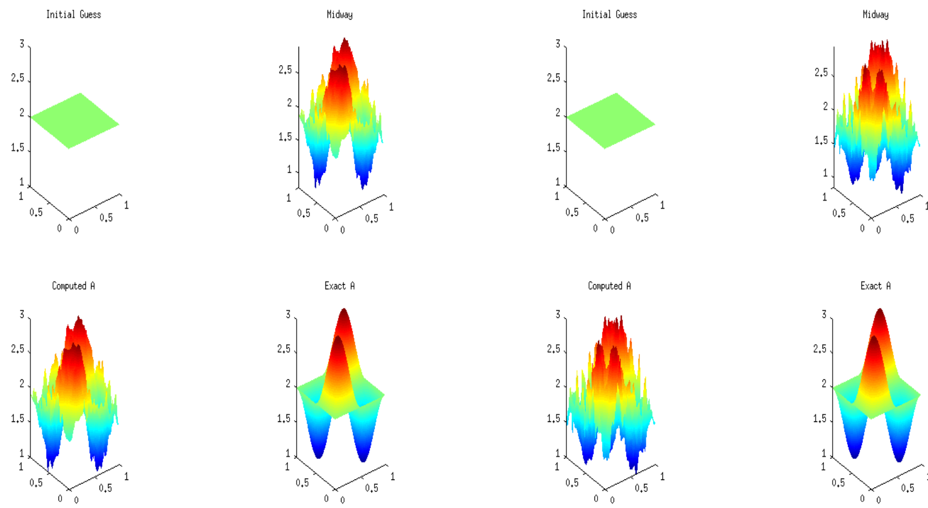

Figure 6.54: Example 2 MOLS vs EE Noise $= 10^{-2}$

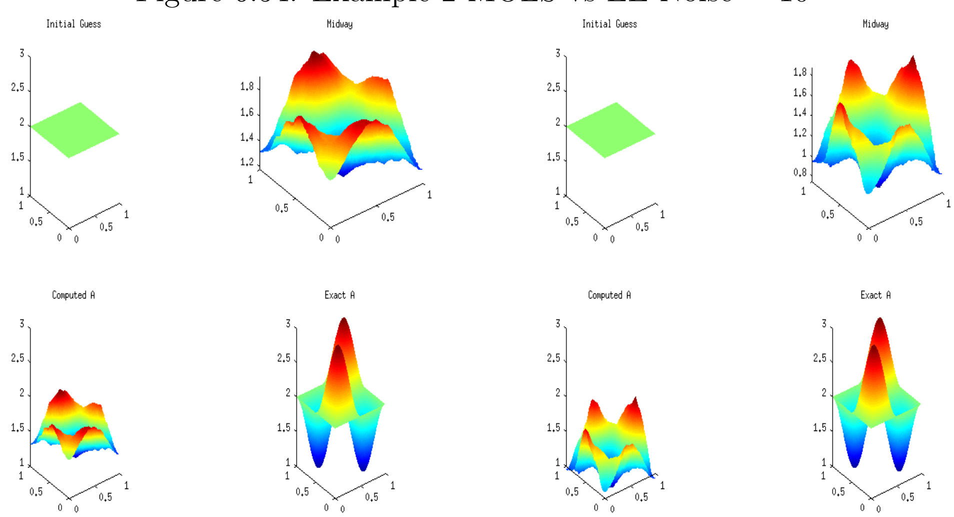

Figure 6.55: Example 2 MOLS vs EE Noise $= 5 \cdot 10^{-2}$

Table 6.14: Example 2 - Euler Noise Study

Scheme	Noise	ε	Error	Time (s)	iters
MOLS	$5 \cdot 10^{-2}$	10^{-2}	0.6015	7.8	7
	10^{-2}	10^{-4}	0.1494	4.2	7
	10^{-3}	10^{-6}	0.0733	4.7	8
EE	$5 \cdot 10^{-2}$	10^{-2}	0.9225	5.0	6
	10^{-2}	10^{-4}	0.2376	5.1	7
	10^{-3}	10^{-6}	0.0900	2.7	4

Table 6.15: Example 2 - Trapezoid Noise Study

Scheme	Noise	ε	Error	Time (s)	iters
MOLS	$5 \cdot 10^{-2}$	10^{-2}	0.6050	66.9	39
	10^{-2}	10^{-4}	0.1453	36.9	38
	10^{-3}	10^{-6}	0.0732	24.7	25
EE	$5 \cdot 10^{-2}$	10^{-2}	0.9331	58.8	40
	10^{-2}	10^{-4}	0.1453	29	37
	10^{-3}	10^{-6}	0.0905	23.0	27

Table 6.16: Example 2 - Runge Kutta Noise Study

Scheme	Noise	ε	Error	Time (s)	iters
MOLS	$5 \cdot 10^{-2}$	10^{-2}	0.6003	73.4	20
	10^{-2}	10^{-4}	0.1592	40.7	20
	10^{-3}	10^{-6}	0.0675	59.7	27
EE	$5 \cdot 10^{-2}$	10^{-2}	0.8937	68.7	21
	10^{-2}	10^{-4}	0.2452	38.1	20
	10^{-3}	10^{-6}	0.0948	34.2	19

Table 6.17: Example 2 - ODE113 Noise Study

Scheme	Noise	ε	Error	Time (s)	iters
MOLS	$5 \cdot 10^{-2}$	10^{-2}	0.5924	271.0	153
	10^{-2}	10^{-4}	0.1436	140.5	131
	10^{-3}	10^{-6}	0.0697	157.4	154
EE	$5 \cdot 10^{-2}$	10^{-2}	0.9313	220.5	127
	10^{-2}	10^{-4}	0.2443	96.3	117
	10^{-3}	10^{-6}	0.0914	105.3	119

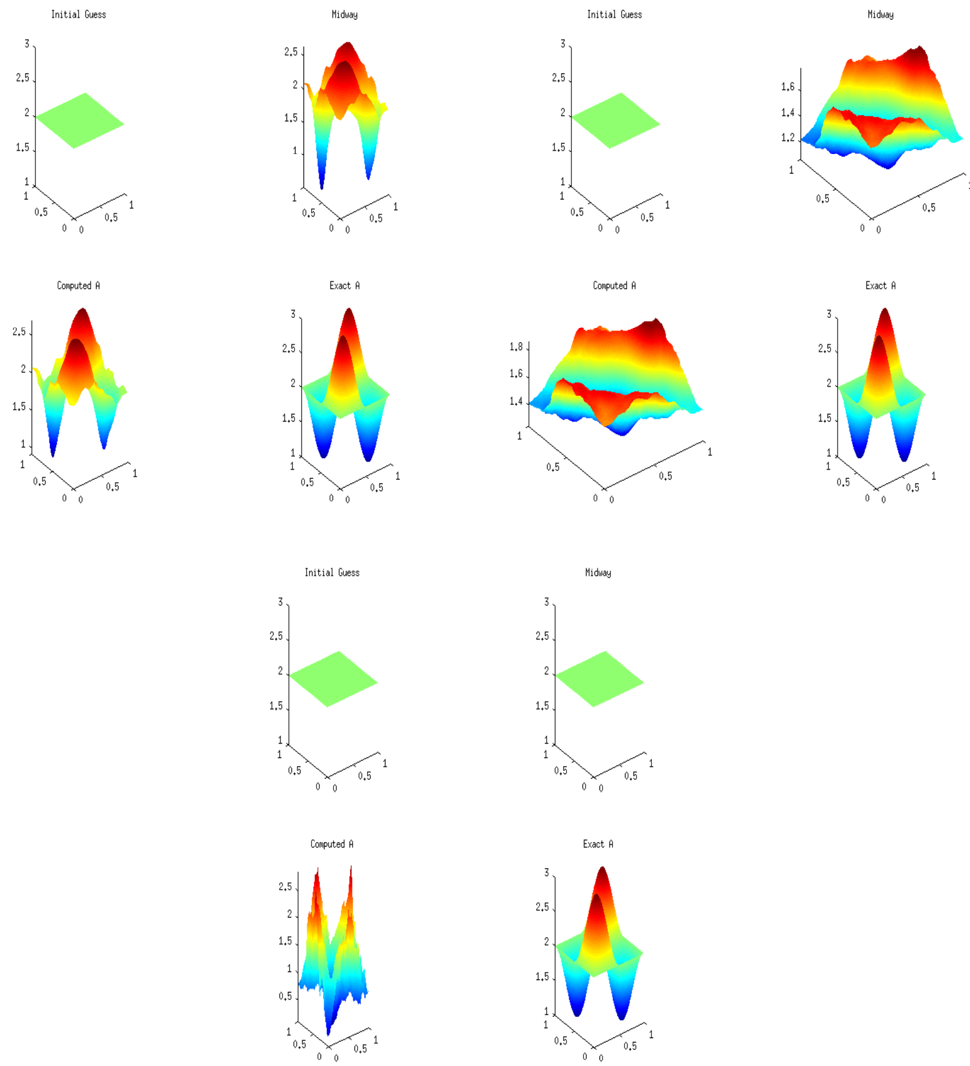

Figure 6.56: Example 2 OLS vs. MOLS vs. EE Noise $= 5 \cdot 10^{-2}$

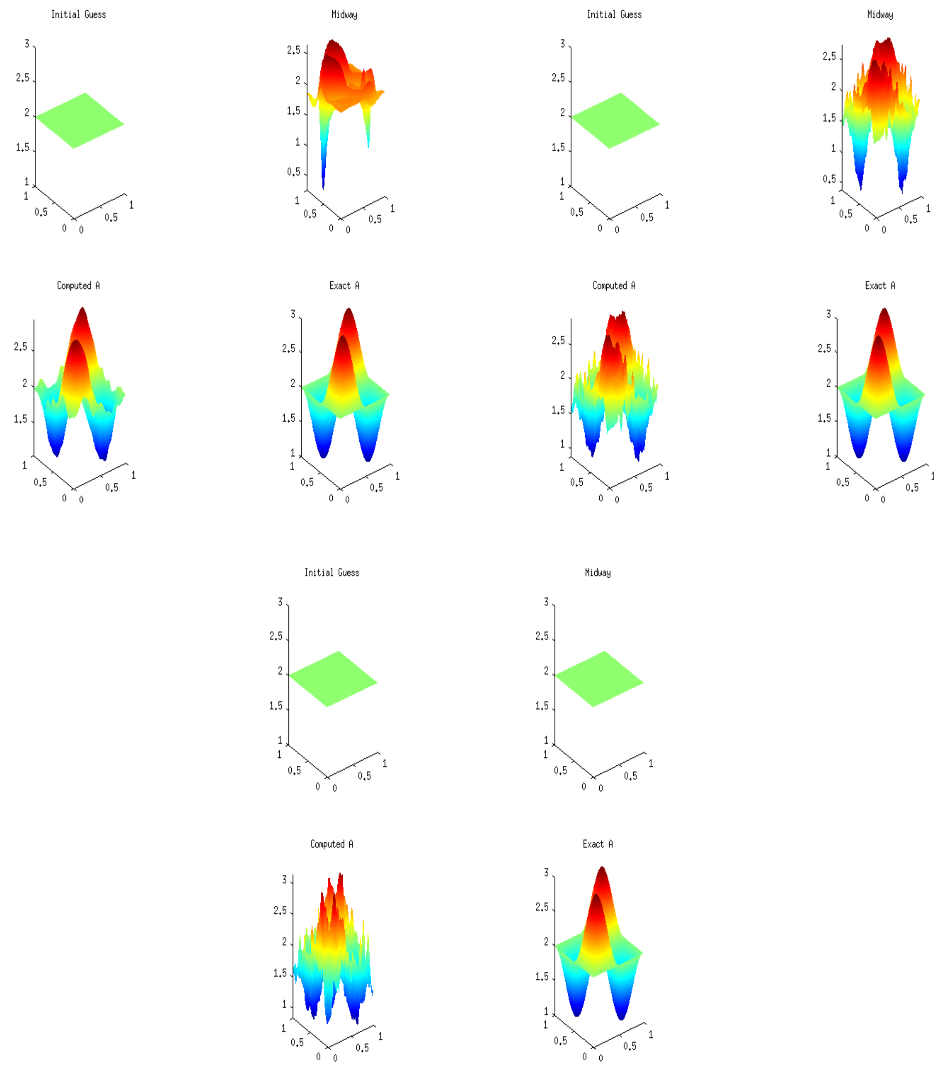

Figure 6.57: Example 2 OLS vs. MOLS vs. EE Noise = 10^{-2}

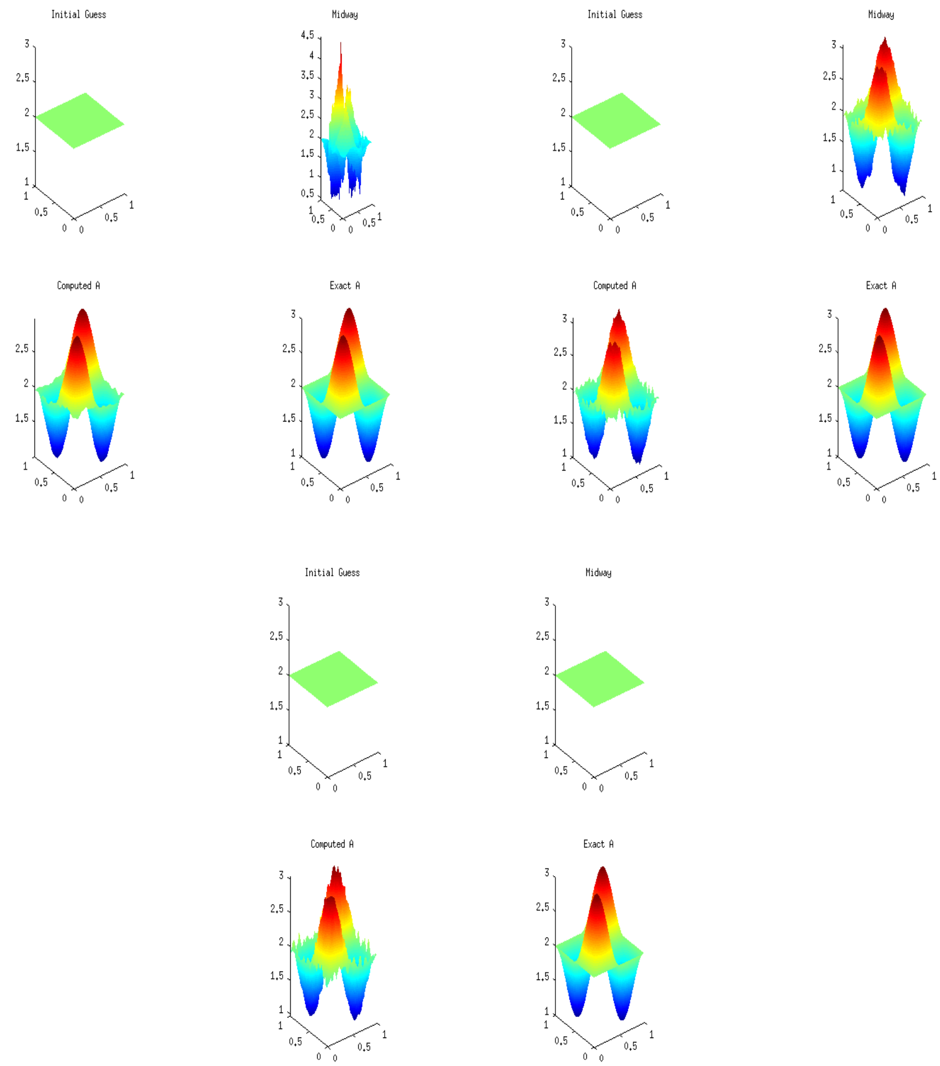

Figure 6.58: Example 2 OLS vs. MOLS vs. EE Noise = 10^{-3}

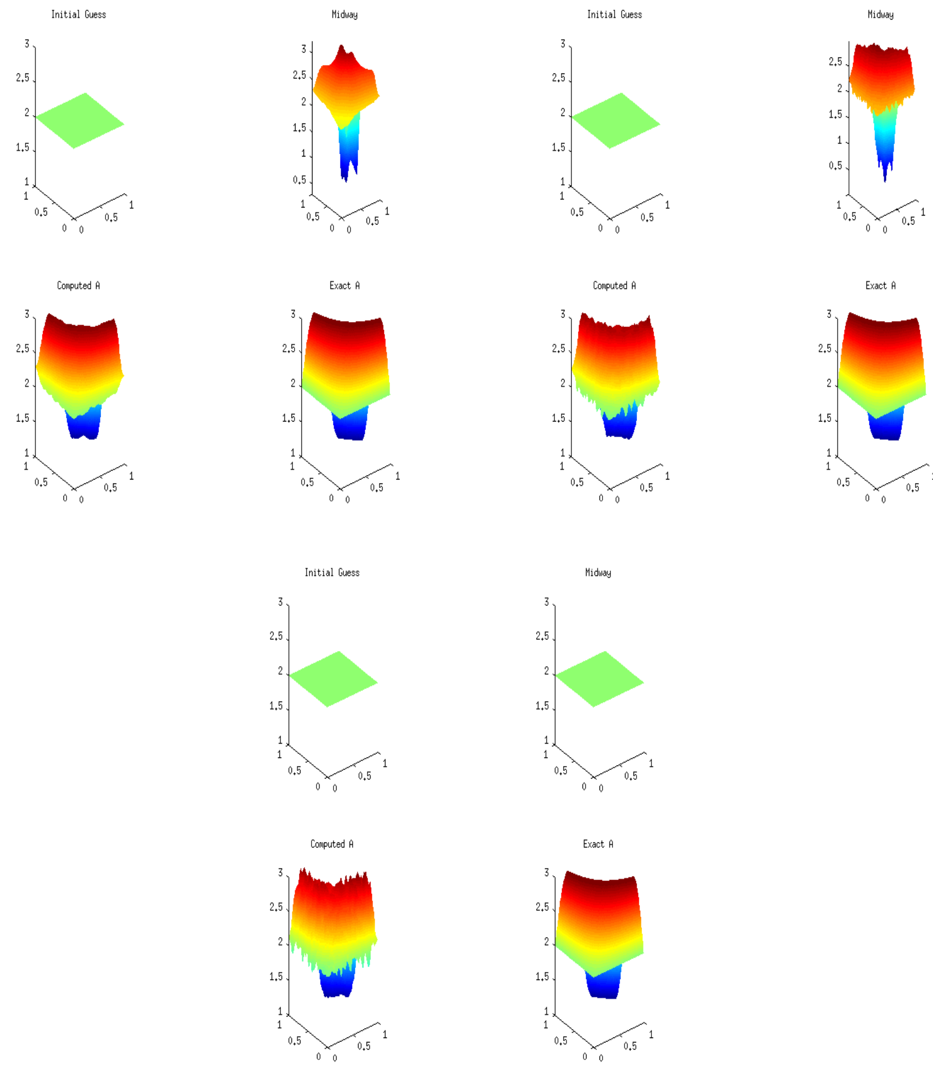

Figure 6.59: Example 3 OLS vs. MOLS vs. EE Noise = 10^{-1}

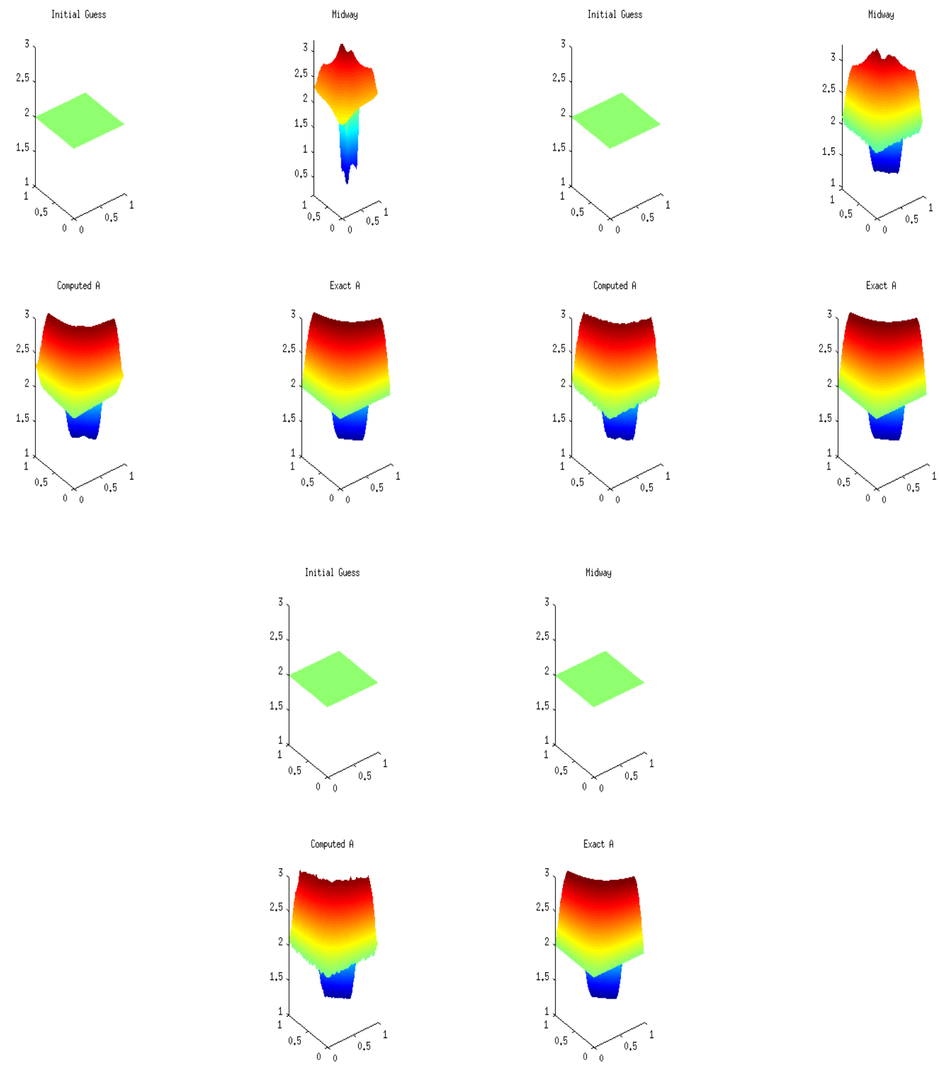

Figure 6.60: Example 3 OLS vs. MOLS vs. EE Noise = 10^{-2}

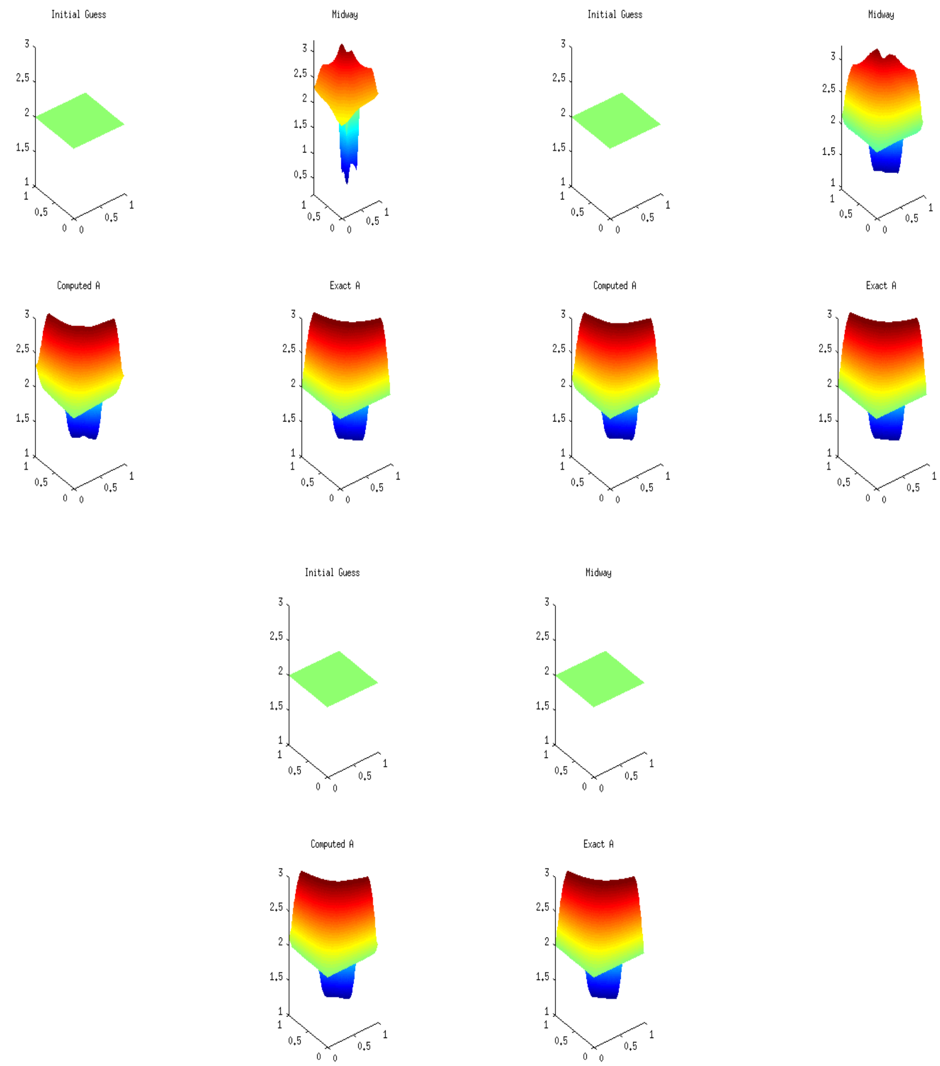

Figure 6.61: Example 3 OLS vs. MOLS vs. EE Noise = 10^{-3}

Table 6.18: Example 2 - Scheme Comparison

Scheme	Noise	ε	Error	Time (s)	iters
OLS	0.05	10^{-6}	0.1719	29.1	5
	0.01	10^{-7}	0.0761	46.2	7
	0.001	10^{-8}	0.0229	62.1	10
	0.0001	10^{-9}	0.0078	72.8	10
MOLS	0.05	10^{-2}	0.5982	3.9	5
	0.01	10^{-4}	0.1516	4.7	6
	0.001	10^{-5}	0.0342	4.6	6
	0.0001	10^{-6}	0.0082	5.2	7
EE	0.05	10^{-3}	1.0246	1.3	2
	0.01	10^{-4}	0.2527	1.4	2
	0.001	10^{-5}	0.0421	1.3	2
	0.0001	10^{-6}	0.0094	1.3	2

Table 6.19: Example 3 - Scheme Comparison

Scheme	Noise	ε	Error	Time (s)	iters
OLS	0.1	10^{-7}	0.0322	58.1	10
	0.01	10^{-7}	0.0309	55.8	9
	0.001	10^{-7}	0.0311	55.5	9
	0.0001	10^{-7}	0.0311	52.6	9
MOLS	0.1	10^{-4}	0.0411	6.2	9
	0.01	10^{-5}	0.0107	4.3	6
	0.001	10^{-5}	0.0080	4.3	6
	0.0001	10^{-5}	0.0079	4.3	6
EE	0.1	10^{-4}	0.0441	0.85	2
	0.01	10^{-5}	0.0123	0.87	2
	0.001	10^{-5}	0.0051	0.87	2
	0.0001	10^{-5}	0.0050	0.87	2

Chapter 7

Concluding Remarks

In this thesis work, we presented a new computational method for solving the inverse problem of parameter identification. Mixed finite element methods were applied in order to discretize and solve the problem. This method was used in order to overcome the locking effect associated with the classical finite element methods, when applied to the elastography problem. We develop an optimization framework for our inverse problem, and use different schemes including the output least squares(OLS), modified output least squares(MOLS), energy output least squares(EOLS), and equation error(EE) schemes. We proposed the use of second order techniques to solve these optimization schemes, and this led to the novel computation methods for the second order derivative of the OLS functional. We develop a hybrid, and second order adjoint method for the computation of the second order derivative, based on the first order adjoint technique. We also provide discrete frameworks for each optimization scheme, as well as discrete formulations of all the required terms for optimization.

In order to solve our optimization problem, we propose the use of dynamical systems. This is a well known application of continuous methods,and it has been previously applied to inverse problems. However, the continuous newton type method had only been applied once(see [44]), and the heavy ball with friction(HBF) method has not been applied to solving optimization problems in the context of an inverse problem. The combination of the ideas from the continuous newton type method, and the HBF method is a new idea in the field of optimization. We study how the parameters in the HBF method can be used to improve convergence speed over the continuous newton type method.

We report first, the results from testing the new OLS second order derivative computations. A second order method for optimization is used. For completeness, we test these computations in the simpler scalar example, as well as the more complicated elasticity problem. The hybrid method solves only

$m + 2$ linear systems, as oppose to the second order adjoint method that solves $2m + 2$ linear systems. This means that even in the simpler scalar problem, when the second order adjoint method took less iterations to reach a solution, it took longer computational time than the hybrid method.
Next, we tested the HBF method as compared to the continuous newton type method. In both the scalar and elasticity cases, we showed that, with appropriately chosen parameters in the HBF method, it ensures increased convergence speed. To illustrate the importance of this, we shows how the choice of the parameters in the HBF method can grossly affect the convergence speed of the method for better or worse. (see Table 5.5)
Lastly, we conducted a noise study comparing different differential equation solvers. We noted that when using the MOLS, and EE schemes, the range of errors across differential equation solvers remained about the same. This seems to imply that the solvers has no say in the accuracy of the method. Next, we compared the OLS, MOLS, and EE schemes. We noted that for very little error levels, the EE scheme was the most efficient method because of its formulation, however, as the noise levels were increased, we noted that this method quickly became the worst method. This is because the EE scheme requires the computation of the derivative of the data, which is very noisy. The OLS scheme on the other hand, while slower than the other schemes, had a smaller range of errors as the noise level increased. As a result, the OLS method handled the noise the best. The MOLS method handled noise a lot better than the EE technique but not as well as the OLS scheme.
The choice of the parameters in the HBF technique is a very important factor in how the method works, and its convergence speed. As a result, this is the direction that this thesis work points towards for further research. Extensive study of the relationship between the parameters and the problems, and between the parameters themselves is very important in order to improve and increase the use of this method.

Bibliography

[1] Miguel Aguiló, Wilkins Aquino, John C Brigham, Mostafa Fatemi, et al. An inverse problem approach for elasticity imaging through vibroacoustics. *Medical Imaging, IEEE Transactions on*, 29(4):1012–1021, 2010.

[2] Felipe Alvarez. On the minimizing property of a second order dissipative system in hilbert spaces. *SIAM Journal on Control and Optimization*, 38(4):1102–1119, 2000.

[3] Felipe Alvarez, Hedy Attouch, Jérôme Bolte, and P Redont. A second-order gradient-like dissipative dynamical system with hessian-driven damping.: Application to optimization and mechanics. *Journal de mathématiques pures et appliquées*, 81(8):747–779, 2002.

[4] F Alvarez D and JM Pérez C. A dynamical system associated with newton's method for parametric approximations of convex minimization problems. *Applied mathematics & optimization*, 38:193–217, 1998.

[5] Habib Ammari, Pierre Garapon, and François Jouve. Separation of scales in elasticity imaging: a numerical study. *J. Comput. Math*, 28(3):354–370, 2010.

[6] Anatoly Sergeevich Antipin. Minimization of convex functions on convex sets by means of differential equations. *Differential Equations*, 30(9):1365–1375, 1994.

[7] 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. *Physics in Medicine and Biology*, 55(7):2035, 2010.

[8] H Attouch and F Alvarez. The heavy ball with friction dynamical system for convex constrained minimization problems. optimization (namur, 1998), 25–35. *Lecture Notes in Econom. and Math. Systems*, 481.

[9] Hedy Attouch and F Alvarez. *The heavy ball with friction dynamical system for convex constrained minimization problems.* Springer, 2000.

[10] Hedy Attouch, Xavier Goudou, and Patrick Redont. The heavy ball with friction method, i. the continuous dynamical system: global exploration of the local minima of a real-valued function by asymptotic analysis of a dissipative dynamical system. *Communications in Contemporary Mathematics*, 2(01):1–34, 2000.

[11] Jeffrey C Bamber, Leanore De Gonzalez, David O Cosgrove, Phillip Simmons, J Davey, and JA McKinna. Quantitative evaluation of real-time ultrasound features of the breast. *Ultrasound in medicine & biology*, 14:81–87, 1988.

[12] H Thomas Banks and Karl Kunisch. *Estimation techniques for distributed parameter systems*. Springer Science & Business Media, 2012.

[13] Elena Beretta, Eric Bonnetier, Elisa Francini, and Anna L Mazzucato. Small volume asymptotics for anisotropic elastic inclusions. *Inverse Prob. Imaging*, 6:1–23, 2011.

[14] M Bertrand, J Meunier, M Doucet, and G Ferland. Ultrasonic biomechanical strain gauge based on speckle tracking. In *Ultrasonics Symposium, 1989. Proceedings., IEEE 1989*, pages 859–863. IEEE, 1989.

[15] Charalampos A Botsaris. Differential gradient methods. *Journal of Mathematical Analysis and Applications*, 63(1):177–198, 1978.

[16] AA Brown and MC Bartholomew-Biggs. Ode versus sqp methods for constrained optimization. *Journal of optimization theory and applications*, 62(3):371–386, 1989.

[17] AA Brown and Michael C Bartholomew-Biggs. Some effective methods for unconstrained optimization based on the solution of systems of ordinary differential equations. *Journal of Optimization Theory and Applications*, 62(2):211–224, 1989.

[18] Erin Crossen, Mark S Gockenbach, Baasansuren Jadamba, Akhtar A Khan, and Brian Winkler. An equation error approach for the elasticity imaging inverse problem for predicting tumor location. *Computers & Mathematics with Applications*, 67(1):122–135, 2014.

[19] MM Doyley. Model-based elastography: a survey of approaches to the inverse elasticity problem. *Physics in medicine and biology*, 57(3):R35, 2012.

[20] MM Doyley, B Jadamba, AA Khan, M Sama, and B Winkler. A new energy inversion for parameter identification in saddle point problems

with an application to the elasticity imaging inverse problem of predicting tumor location. *Numerical Functional Analysis and Optimization*, 35(7-9):984–1017, 2014.

[21] MM Doyley, PM Meaney, and JC Bamber. Evaluation of an iterative reconstruction method for quantitative elastography. *Physics in Medicine and Biology*, 45(6):1521, 2000.

[22] Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.

[23] Richard S Falk. Error estimates for the numerical identification of a variable coefficient. *Mathematics of Computation*, 40(162):537–546, 1983.

[24] Michael P Glazos, Stefen Hui, and Stanislaw H Zak. Sliding modes in solving convex programming problems. *SIAM Journal on Control and Optimization*, 36(2):680–697, 1998.

[25] Mark S Gockenbach and Akhtar A Khan. Identification of lamé parameters in linear elasticity: a fixed point approach. *Journal of Industrial and Management Optimization*, 1(4):487, 2005.

[26] MS Gockenbach, B Jadamba, and AA Khan. Equation error approach for elliptic inverse problems with an application to the identification of lamé parameters. *Inverse Problems in Science and Engineering*, 16(3):349–367, 2008.

[27] Timothy P Harrigan and Elisa E Konofagou. Estimation of material elastic moduli in elastography: a local method, and an investigation of poisson's ratio sensitivity. *Journal of biomechanics*, 37(8):1215–1221, 2004.

[28] B Jadamba, AA 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 Journal on Applied Mathematics*, 74(5):1486–1510, 2014.

[29] B Jadamba, AA Khan, and M Sama. Inverse problems of parameter identification in partial differential equations. *Mathematics in Science and Technology*, pages 228–258, 2011.

[30] Florence Jules and Paul-Emile Maingé. Numerical approach to a stationary solution of a second order dissipative dynamical system. *Optimization*, 51(2):235–255, 2002.

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

[32] Ian Knowles. Parameter identification for elliptic problems. *Journal of computational and applied mathematics*, 131(1):175–194, 2001.

[33] Elisa E Konofagou, Timothy P Harrigan, Jonathan Ophir, and Thomas A Krouskop. Poroelastography: imaging the poroelastic properties of tissues. *Ultrasound in medicine & biology*, 27(10):1387–1397, 2001.

[34] Li-Zhi Liao, Liqun Qi, and Hon Wah Tam. A gradient-based continuous method for large-scale optimization problems. *Journal of Global Optimization*, 31(2):271–286, 2005.

[35] LZ Liao. A continuous method for convex programming problems. *Journal of optimization theory and applications*, 124(1):207–226, 2005.

[36] Tao Lin and Edgardo Ramirez. A numerical method for parameter identification of a boundary value problem. *Applicable Analysis*, 69(3-4):349–379, 1998.

[37] Hatef Mehrabian, Gordon Campbell, and Abbas Samani. A constrained reconstruction technique of hyperelasticity parameters for breast cancer assessment. *Physics in medicine and biology*, 55(24):7489, 2010.

[38] Assad A Oberai, Nachiket H Gokhale, and Gonzalo R Feijóo. Solution of inverse problems in elasticity imaging using the adjoint method. *Inverse Problems*, 19(2):297, 2003.

[39] World Health Organization et al. Cancer. fact sheet n 297. 2013, 2011.

[40] KJ Parker, SR Huang, RA Musulin, and RM Lerner. Tissue response to mechanical vibrations for sonoelasticity imaging. *Ultrasound in medicine & biology*, 16(3):241–246, 1990.

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

[42] Stefan Schäffler and Hubert Warsitz. A trajectory-following method for unconstrained optimization. *Journal of Optimization Theory and Applications*, 67(1):133–140, 1990.

[43] Z-J Shi. Convergence of multi-step curve search method for unconstrained optimization. *Journal of Numerical Mathematics jnma*, 12(4):297–309, 2004.

[44] Corinne Teravainen. Continuous methods for elliptic inverse problems. 2014.

[45] Daniel A Tortorelli and Panagiotis Michaleris. Design sensitivity analysis: overview and review. *Inverse problems in Engineering*, 1(1):71–105, 1994.

[46] Curtis R Vogel. *Computational methods for inverse problems*, volume 23. Siam, 2002.

[47] David JN Wall, Peter Olsson, and Elijah EW Van Houten. On an inverse problem from magnetic resonance elastic imaging. *SIAM Journal on Applied Mathematics*, 71(5):1578–1605, 2011.

[48] Lei-Hong Zhang, CT Kelley, and Li-Zhi Liao. A continuous newton-type method for unconstrained optimization. *Pacific Journal of Optimization*, 4(2):259–277, 2008.