DISSERTATION

AN ADAPTIVE ALGORITHM FOR AN ELLIPTIC OPTIMIZATION PROBLEM, AND STOCHASTIC-DETERMINISTIC COUPLING: A MATHEMATICAL FRAMEWORK

Submitted by

Sheldon Lee

Department of Mathematics

In partial fulfillment of the requirements

for the degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Summer 2008

COLORADO STATE UNIVERSITY

July 2, 2008

WE HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER OUR SUPERVISION BY SHELDON LEE ENTITLED "AN ADAPTIVE ALGORITHM FOR AN ELLIPTIC OPTIMIZATION PROBLEM, AND STOCHASTIC-DETERMINISTIC COUPLING: A MATHEMATICAL FRAMEWORK" BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY.

Committee on Graduate Work

Dr. Edwin Chong

Dr. James Liu

Adviser: Dr. Donald Estep

Co-Adviser: Dr. Simon Tavener

Department Head: Dr. Simon Tavener

ABSTRACT OF DISSERTATION

AN ADAPTIVE ALGORITHM FOR AN ELLIPTIC OPTIMIZATION PROBLEM, AND STOCHASTIC-DETERMINISTIC COUPLING: A MATHEMATICAL FRAMEWORK

This dissertation consists of two parts. In the first part, we study optimization of a quantity of interest of a solution of an elliptic problem, with respect to parameters in the data using a gradient search algorithm. We use the generalized Green's function as an efficient way to compute the gradient. We analyze the effect of numerical error on a gradient search, and develop an efficient way to control these errors using a posteriori error analysis. Specifically, we devise an adaptive algorithm to refine and unrefine the finite element mesh at each step in the descent search algorithm. We give basic examples and apply this technique to a model of a healing wound.

In the second part, we construct a mathematical framework for coupling atomistic models with continuum models. We first study the case of coupling two deterministic diffusive regions with a common interface. We construct a fixed point map by repeatedly solving the problems, while passing the flux in one direction and the concentration in the other direction. We examine criteria for the fixed point iteration to converge, and offer remedies such as reversing the direction of the coupling, or relaxation, for the case it does not.

We then study the one dimensional case where the particles undergo a random walk on a lattice, next to a continuum region. As the atomistic region is random, this technique yields a fixed point iteration of distributions. We run numerical tests to study the long term behavior of such an iteration, and compare the results with the deterministic case. We also discuss a probability transition matrix approach, in which we assume that the boundary conditions at each iterations follow a Markov chain.

Sheldon Lee
Department of Mathematics
Colorado State University
Fort Collins, Colorado 80523
Summer 2008

ACKNOWLEDGEMENTS

I would like to thank Don Estep for being an excellent advisor, and a great model of a successful research mathematician. I am thankful for the monetary support that he has provided for me, which enabled me to attend conferences and workshops, as well as allowed me to focus my entire attention on my research. I thank Simon Tavener for investing so much time and energy into my project. As co-advisor, he has been an invaluable resource for the second part of this project. Both Don and Simon provided me with plenty of moral support and enthusiasm. I thank the committee members James Liu and Edwin Chong for their service, and also thank the members of the research team for their support and interest in my work. I thank John Neuberger for spending time helping me to develop my research abilities, and for believing in my ability to succeed at the doctoral level. I thank the faculty I have worked with at Colorado State University, Winona State University, Northern Arizona University, and the University of Wisconsin-Stout for their role in developing my mathematical ability. I thank my student colleagues for providing me with a fun and interesting graduate school experience that I will treasure. I thank my friends and siblings for their support, especially Anne for helping with the final edits. Most importantly, I thank my parents for everything they have done for me. They have a great respect for higher education, and have provided me with unconditional love and support through my years spent in academia.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

This thesis consists of two separate projects, and is thus divided into two parts. This chapter discusses an overview of both problems, and is followed by Parts I and II. Part I consists of Chapters 2 through 5, in which we discuss an adaptive optimization problem. Part II consists of Chapters 6 through 9, where we discuss a mathematical framework for stochastic-deterministic coupling. We end the paper with a Chapter 10, where we give conclusions for both parts.

1.1 An Adaptive Algorithm for an Elliptic Optimization Problem

The first part of the thesis involves solving an elliptic parameter optimization problem. This involves optimizing a quantity of interest obtained from an approximation to the solution of a differential equation. We compute an expression for the gradient of the quantity of interest, then search for a local extrema using a gradient search technique. We utilize *a posteriori* error analysis [22, 18, 3] to correct for the numerical error in the solution, and hence in the gradient. The ideas of *a posteriori* error analysis have made an enormous impact in engineering application over the past few decades, and many of the ideas have been adopted into production codes

at Sandia National Laboratory. In this first project, we use these error techniques to speed up and improve the process of optimizing a quantity of interest.

Consider a wound healing model as in [37], where we have two conservative equations, one for the epithelial cell density per unit area, u_1, and one for the concentration, u_2, of the mitosis-regulating chemical. The three parameters describe the time decay rate of the chemical, the maximum rate of chemical production, and the maximum level of chemical activation of mitosis. We study the problem of optimizing the functional $q(\lambda) = ((u_1, u_2)^T, \psi)_{L^2(\Omega)}$.

The general problem is to optimize $q(u;\lambda) = (u, \psi)$, where u solves

$$\begin{cases} -\nabla \cdot (a\nabla u) = f(u;\lambda), & x \in \Omega, \\ u = g(x), & x \in \partial\Omega. \end{cases} \tag{1.1.1}$$

We use the generalized Green's function, or adjoint, as an efficient way to compute the gradient used for the descent algorithm. If we consider a perturbation of parameter λ, $\tilde{\lambda}$, and the corresponding solution $\tilde{u}$, we linearize about $\tilde{u}$ to obtain the adjoint $\tilde{\phi}$, which satisfies

$$\begin{cases} -\nabla \cdot (a\nabla\tilde{\phi}) - D_u^* f(u;\lambda)\tilde{\phi} = \psi & x \in \Omega, \\ \tilde{\phi} = 0, & x \in \partial\Omega. \end{cases}$$

We then express the gradient $\nabla_\lambda q(\tilde{\lambda})$ using the adjoint,

$$\nabla_\lambda q(\tilde{u};\tilde{\lambda}) \cdot (\lambda - \tilde{\lambda}) = (\nabla_\lambda f(\tilde{u};\tilde{\lambda}) \cdot (\lambda - \tilde{\lambda}), \tilde{\phi})_{L^2(\Omega)}.$$

We implement a conjugate direction method to find local extrema in parameter space. We then analyze the effect of numerical error on such a gradient search. We take note that different parameter values may require different meshes to achieve accuracy, and that computing a mesh for each parameter

value in the descent search is an expensive proposition. To remedy this, we devise an adaptive algorithm that uses the same mesh, which we refine and unrefine as the parameter changes. Using *a posteriori* techniques, we derive an expression for the gradient error, and show that this represents the change in error from one parameter value to the next. Specifically,

$$\nabla_\lambda q(\tilde{U};\tilde{\lambda})\cdot(\lambda-\tilde{\lambda}) = (\nabla_\lambda f(\tilde{U},\tilde{\lambda})\cdot(\lambda-\tilde{\lambda}),\tilde{\phi}) + \mathcal{R}(U,\phi,\tilde{\phi}), \qquad (1.1.2)$$

where $\tilde{U}$ is the finite element approximation to (1.1.1) with parameter $\tilde{\lambda}$, and ϕ, $\tilde{\phi}$ solve the adjoint problems for parameters λ and $\tilde{\lambda}$ respectively. The term $(\nabla_\lambda f(\tilde{U},\tilde{\lambda})\cdot(\lambda-\tilde{\lambda}),\tilde{\phi})$ is the computable approximate gradient, and the term $\mathcal{R}(U,\phi,\tilde{\phi})$ represents the weak residual for the gradient error. To summarize the algorithm, we first construct an efficient mesh for the first parameter λ_0 by controlling the error in $q(\lambda_0)$, using standard *a posteriori* techniques. For subsequent parameter values, we use (1.1.2) to approximate the gradient, while using $\mathcal{R}(U,\phi,\tilde{\phi})$ to flag elements for refining and unrefining. We give basic examples and apply this technique to the wound healing model.

The first part of the thesis consists of Chapters 2 through 5. In Chapter 2, we discuss background material from the areas of analysis and differential equations. This includes a discussion of the basic function spaces, operator derivatives, adjoint operators, and elliptic operators. In Chapter 3, we discuss the finite element methods and *a posteriori* error estimation used to solve elliptic problems and correct for numerical error. In Chapter 4, we discuss some of the basic gradient search methods used for unconstrained optimization. Here we discuss steepest descent, the conjugate gradient method, and line search methods. In Chapter 5, we discuss the

aforementioned optimization problem in detail, with proofs of the major theorems and numerical results.

1.2 Stochastic-Deterministic Coupling: A Mathematical Framework

In the second part of this thesis, we create a rigorous mathematical framework for coupling atomistic with continuum models.

The continuum region is modeled by differential equations and is solved using standard methods such as finite elements. The atomistic region is modeled by some atomistic simulation method such as molecular dynamics, Monte Carlo, or in our case, Brownian motion. As atomistic simulations are random, we introduce stochastic error into the problem. Controlling the stochastic error of the atomistic simulation could involve increasing the resolution of the simulation, the number of samples, or the length of time that the simulation is run.

If we solve the continuum problem using finite elements, we can reduce the discretization error using standard *a posteriori* techniques. In coupling the continuous problem with an atomistic problem, we typically pass results of the atomistic simulation as parameters into the continuum region. Since the results of the atomistic simulation are uncertain, we introduce stochastic error into the continuum problem. Ideally, we would like to account and correct for both types of errors in an efficient way. A rigorous mathematical theory for achieving this goal is lacking.

We are particularly interested in situations in which there is feedback between the two models. We consider iterative algorithms in which data is passed back and forth between the continuum and the atomistic model.

In some coupling schemes, only an ensemble average of quantities from the stochastic model are passed to the continuum model. This complicates the notion of defining a convergence of iterations, as well as introduces modeling assumptions that may not be valid. Instead, we consider the entire distribution of values from the atomistic simulations. That is, we pass distributions back and forth between the models.

The iterative process of passing the distribution of values at the interface is considered to be a fixed point problem. Using the Banach fixed point theorem, we can obtain necessary conditions for the sequence to converge.

In the case of atomistic coupling through the boundary, we couple a Brownian motion computation on a domain Ω_1, to a continuum problem

$$\begin{cases} -\Delta u = f(u), & x \in \Omega_2, \\ \frac{\partial u}{\partial \eta} = \lambda(x), & x \in \Gamma \subset \partial\Omega_2, \\ u = g(x), & x \in \partial\Omega_2 \setminus \Gamma, \end{cases} \tag{1.2.1}$$

where λ is the random field generated from an atomistic simulation in an adjacent region Ω_1. The coupling is through the common boundary Γ. We solve (1.2.1) using several realizations of λ. The solution u on Γ is also a random vector, which enters back into the atomistic model. The algorithm is as follows.

Make initial guess $u^{(0)}|_\Gamma$.

For $k = 1, 2, ...$

Using $u^{(k-1)}|_\Gamma$ as an initial condition, approximate $\lambda(x)^{(k)}$ using an atomistic technique in Ω_1.

Approximate $u^{(k)}$ by solving (1.2.1).

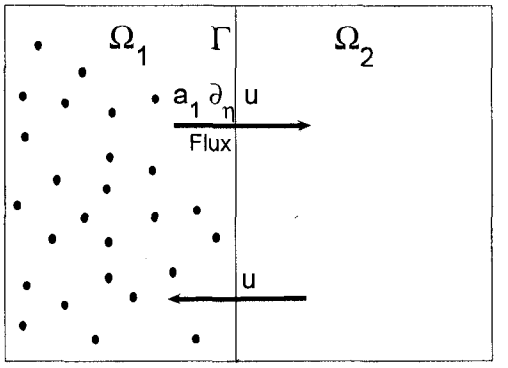

Figure 1.1: We use an atomistic model in Ω_1, and a continuum model in Ω_2. We pass information through Γ, the common interface.

End for

Since λ and $u|_\Gamma$ are random variables, this coupling involves passing distributions back and forth. The natural questions that arise are what does it means for a distribution to converge, and how do we test for convergence?

Chapters 6 through 9 discuss this atomistic to continuum portion of the thesis. In Chapter 6, we include some basic background material from probability theory. We briefly mention the key theorems used to perform statistical analysis on an atomistic simulation. Next, we briefly discuss kernel density estimation. We then include some of the relevant matrix theory needed to study Markov chains. We conclude the chapter with a discussion of stochastic processes and Markov chains. In Chapter 7, we give an overview of molecular dynamics and Brownian motion. These are two of the many techniques that can be used to simulate the diffusion equation in a region at the atomistic level. In Chapter 8, we discuss the problem of coupling diffusion equations across an interface. This chapter consists of the formulation and convergence of the fixed point problem. In Chapter

9, we discuss the stochastic-to-deterministic coupling for a one-dimensional diffusion problem. We formulate the problem of finding the distribution on the interface as the solution to a fixed point problem. We then mention an alternative formulation, in which we describe the problem as a Markov chain. We discuss convergence criteria and show numerical results in both cases.

Part I

An Adaptive Algorithm for an Elliptic Optimization Problem

Chapter 2

BACKGROUND: FUNCTIONAL ANALYSIS AND DIFFERENTIAL EQUATIONS

In our work, we approach error estimation using *a posteriori* techniques, meaning that the error estimates are computed after the solution has been computed. It turns out that stability of the solution of a differential equation greatly influences the effects of perturbation and error. We use duality and adjoint operators to quantify stability *a posteriori*.

2.1 Function Spaces and Duality

We briefly recall linear operators. Given normed linear spaces X and Y over $\mathbb{R}$, a linear operator L is a function $L : X \to Y$ such that $L(\alpha x + y) = \alpha L(x) + L(y)$ for all $x, y \in X, \alpha \in \mathbb{R}$. A linear operator is bounded, or continuous if there is a C such that $\|L(x)\|_Y \leq C\|x\|_X$ for all $x \in X$. The space of all continuous operators from X to Y is denoted by $\mathcal{L}(X, Y)$ and is itself a linear space. An important example is the case $Y = \mathbb{R}$, in which case we call $f : X \to \mathbb{R}$ a linear functional on X. Linear functionals are important in our work as we view them as a particular piece of information from a solution of a model. We call this piece of information a quantity of interest. For example, suppose u belongs to some vector

space X and represents the solution to some differential equation. We could then consider the value of u at a point y_0 in the domain Ω to be a quantity of interest. The quantity of interest is a linear functional given by $\int_\Omega u(x)\,\delta_{y_0}(x)\,dx$. The set of bounded linear functionals on a linear space X is given the following definition.

Definition 2.1.1. *Given a normed linear space X, the dual space of X is the set $X^* : \mathcal{L}(X, \mathbb{R})$ of linear, bounded, real valued functions on X.*

We denote this space by X^*, and denote elements by $x^* \in X^*$. The action of an element of the dual $x^*(x)$ is often denoted with $x^*(x) = \langle x^*, x\rangle$. The dual space is a normed linear space under the dual norm

$$\|x^*\|_{X^*} = \sup_{x\neq 0} \frac{\langle x^*, x\rangle}{\|x\|_X}.$$

For example, it turns out that the dual space of L^p is isometrically isomorphic to L^q, where $p^{-1} + q^{-1} = 1$.

Definition 2.1.2. *We define the adjoint of a map $L \in \mathcal{L}(X, Y)$ to be the map $L^* \in (Y^*, X^*)$ if the following bilinear identity is satisfied:*

$$L^*y^*(x) = y^*(Lx), \text{ or } \langle L^*y^*, x\rangle = \langle y^*, Lx\rangle,$$

for all $x \in X, y^* \in Y^*$.

Example 2.1.1. *If $A : \mathbb{R}^n \to \mathbb{R}^m$ is a linear operator, then A is an $m \times n$ matrix. The adjoint of A, $A^* : \mathbb{R}^m \to \mathbb{R}^n$ is the transpose matrix of A, denoted A^T.*

Proposition 2.1.1. *If we consider $L_1, L_2 \in \mathcal{L}(X, Y)$, then the following properties of the adjoint hold:*

- $(L_1 + L_2)^* = L_1^* + L_2^*$
- $(\alpha L)^* = \alpha L^*$
- $(L_1 L_2)^* = L_2^* L_1^*$.

Next, we define Banach spaces and Hilbert spaces, our vector spaces of choice for solving differential equations using the finite element method.

Definition 2.1.3. *A Banach space is a normed linear space that is complete, that is, any Cauchy sequence in V converges under $\|\cdot\|_V$ to an element in V.*

Definition 2.1.4. *An inner product is a function $(\cdot,\cdot) : X \times X \to \mathbb{R}$, such that the following hold*

1. $(x + y, z) = (x, z) + (y, z)$,
2. $(\alpha x) = \alpha(x, y)$,
3. $(x, y) = \overline{(y, x)}$,
4. $(x, x) \geq 0$, *and* $(x, x) = 0$ *if and only if* $x = 0$.

Definition 2.1.5. *A Hilbert space is a Banach space equipped with an inner product, and whose norm is induced by the inner product $\|\cdot\| = \sqrt{(\cdot,\cdot)}$.*

In a Hilbert space, the norm satisfies the following parallelogram law,

$$\|x + y\|^2 + \|x - y\|^2 = 2(\|x\|^2 + \|y\|^2).$$

If H is a Hilbert space, and $v \in H$, then the function $L(x) = (x, v)_H$ defines a linear functional on H. It turns out that all linear functionals on a Hilbert space may be defined that way. Also, the dual space of a Hilbert space H is isometrically isomorphic to H itself. These facts are stated in the following theorem.

Hilbert Space	Inner Product
$\mathbb{R}^n$	$(x, y) = x \cdot y$
$L^2(\Omega)$	$(u, v)_{L^2(\Omega)} = \int_\Omega u\, v\, dx$
$H^s(\Omega), s = 0, 1, ...$	$(u, v)_{H^s(\Omega)} = \sum_{\lvert\alpha\rvert=0}^{s} (\partial^\alpha u(x), \partial^\alpha v)_{L^2(\Omega)}$

Table 2.1: Hilbert spaces and their associated inner products.

Theorem 2.1.1. *(Riesz Representation Theorem) The dual space of a Hilbert space is isomorphic to the Hilbert space itself. Specifically, if $f \in H^\star$, there is a unique $x \in H$ such that*

$$\langle f, y \rangle = (x, y)_H \text{ for all } y \in H,$$

and furthermore, $\|x\|_H = \|f\|_{H^\star}$.

Using the Riesz Representation Theorem, we express the following identity,

$$(u, L^\star v)_H = (Lu, v)_H,$$

to characterize the adjoint of a linear operator L in a Hilbert space H.

2.1.1 Orthogonal Projectors

One of the reasons that Hilbert spaces are so useful is they allow us to define a sense of orthogonality.

Definition 2.1.6. *In a Hilbert space H, we say that u and v are orthogonal if $(u, v)_H = 0$.*

Suppose we want to compute $(f, v)_{L^2(K)}$ for some $v \in V_h$, where K is some element taken from a finite element mesh. Suppose V_h is the space of functions that are polynomials of degree q on each element K. Then we

find a function in $\pi_h f \in V_h$ that is closest to f on average. Specifically, we require that $(f, v)_{L^2(K)} = (\pi_h f, v)_{L^2(K)}$, or

$$(f - \pi_h f, v)_{L^2(K)} = 0,$$

for all $v \in V_h$. In other words, we seek the function $\pi_h f$ such that $f - \pi_h f$ is orthogonal to the space V_h. Such a function $\pi_h f$ is called the L^2 projection of f into V_h.

We show that π_h exists uniquely. To show uniqueness, if there are $u, v \in V_h$ such that $(f - u, w)_H = 0$ and $(f - v, w)_H$ for all $w \in V_h$, then $u - w \in V_h$. Setting $w = u - v$, we obtain $(u - v, u - v)_H = 0$, so $u = v$. Next, given a basis $\{v_j\}_{j=0}^q$ of V_h, we write

$$\pi_h f(x) = \sum_{j=0}^{q} w_j v_j(x),$$

so that $(\pi_h f, w)_H = \sum_{j=0}^q w_j (v_j, w)_H$ for all $w \in V_h$. Applying each basis element in V_h,

$$(f, v_i)_H = \sum_{j=0}^{q} w_j (v_j, v_i)_H \text{ for } i = 0, 1, ..., q,$$

which is a system of $q + 1$ equations and $q + 1$ variables. Since the solution is unique, it must exist.

The projection $\pi_h f$ turns out to be the closest function to f in the L^2 sense. That is,

$$\|f - \pi_h f\|_{L^2(K)} \leq \|f - v\|_{L^2(K)} \text{ for all } v \in V_h.$$

2.2 Gâteaux and Frechét Deriviatives

In the previous section we reviewed the adjoint operator, which is defined for linear operators. As we solve problems with nonlinearities in our

projects, we are left with the problem of how to define the adjoint. In order to obtain an adjoint for a nonlinear differential operator, we will linearize the problem using operator derivatives. These derivatives are discussed briefly in this section, and may be found in detail in [2, 38].

2.2.1 Basic Definitions

Recall that a real-valued function $f : \mathbb{R} \to \mathbb{R}$ is differentiable at x if there exists an $a = f'(x)$ such that

$$\lim_{t \to 0} \frac{1}{t} (f(x+t) - f(x) - a\,t) = 0.$$

We extend this definition to operators in higher dimensions using norms as follows.

Definition 2.2.1. *Let $F : D \subset \mathbb{R}^n \to \mathbb{R}^m$. We say that F is Gâteaux differentiable at $u \in D^0$ if there exists an operator $A \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$ such that*

$$\lim_{h \to 0} \frac{1}{h} \|F(u + hv) - F\,u - h\,A\,v\| = 0 \text{ for all } v \in \mathbb{R}^n. \tag{2.2.1}$$

We think of Av as the deriviative of F in the direction v, and we sometimes write $F'(u)v = Av$. Such an A is unique, for suppose A_1 and A_2 satisfy (2.2.1). Then,

$$\begin{aligned} &\|(A_1 - A_2)v\| \\ &\qquad \leq \left\| \frac{F(u + h\,v) - F(u)}{h} - A_1\,v \right\| + \left\| \frac{F(u + h\,v) - F(u)}{h} - A_2\,v \right\| \\ &= \frac{1}{h} \|F(u + h\,v) - F(u) - h\,A_1\,v\| + \frac{1}{h} \|F(u + h\,v) - F(u) - h\,A_2\,v\|, \end{aligned}$$

which converges to 0 as $h \to 0$. Since v is arbitrary, $\|A_1 - A_2\| = 0$, or $A_1 = A_2$. Note that the statement $A\,v = \lim_{h \to 0} \frac{1}{h}(F(u + h\,v) - F(u))$ for all v follows from (2.2.1). However, the limit on the right hand side existing

for all v does not imply the existence of the Gâteaux deriviative, as the following example demonstrates. Let $f : \mathbb{R}^2 \to \mathbb{R}$ be defined by

$$f(x) = \begin{cases} 0, & x = 0, \\ \frac{x_1 x_2^2}{x_1^2 + x_2^4}, & x \neq 0. \end{cases}$$

Then we show that $\lim_{t \to 0} \frac{1}{t}(f(th) - f(0)) = \frac{h_2^2}{h_1}$, which clearly cannot be written in the form Ah for $A \in \mathcal{L}(\mathbb{R}^2, \mathbb{R})$. Hence, f is not Gâteaux differentiable at $x = 0$.

In general, Gâteaux differentiability does not imply continuity. For example, let $f : \mathbb{R}^2 \to \mathbb{R}$ be defined by $f(x) = \begin{cases} 0 & x_2 = 0 \\ (\frac{x_1^2 x_2}{x_1^4 + x_2^2})^2 & x_2 \neq 0 \end{cases}$. Then we show that f has Gâteaux deriviative $(0,0)^T$ at 0; however, f is not continuous at 0. One can overcome this problem by using Frechét deriviatives.

Definition 2.2.2. *Let $F : D \subset \mathbb{R}^n \to \mathbb{R}^m$, where D is an open subset of $\mathbb{R}^n$. We say that F is Frechét differentiable at $u \in D$ if there exists an operator $A \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$ such that*

$$\lim_{\|h\| \to 0} \frac{\|F(u+h) - F(u) - Ah\|}{\|h\|} = 0. \tag{2.2.2}$$

Note that the above limit must exist for all sequences $\{h_n\}_{n=1}^{\infty}$ of nonzero elements of $\mathbb{R}^n$ such that $h_n \to 0$. If the limit exists, we call A the Frechét derivative of F at u, and write $F'(u) = A$. We also note that (2.2.2) is equivalent to $F(u+h) - F(u) - Ah = \mathbf{o}(\|h\|)$, or

$$Ah = F(u+h) - F(u) + \mathbf{o}\|h\|, \tag{2.2.3}$$

where $g(h) = \mathbf{o}\|h\|$ if and only if $\frac{g(h)}{\|h\|} \to 0$ as $\|h\| \to 0$.

Proposition 2.2.1. *If F is Frechét differentiable, then F is also Gâteaux differentiable.*

The converse to the above statement is not true. For example, let

$$f(x) = \begin{cases} 0 & x = 0 \\ \frac{x_2(x_1^2+x_2^2)^{\frac{3}{2}}}{(x_1^2+x_2^2)^2+x_2^2} & x \neq 0. \end{cases}$$

One may show that f is Gâteaux differentiable at 0, but is not Frechét deriviative at 0.

Finally, the following fact illustrates a main advantage of the Frechét deriviative over the Gâteaux deriviative:

Proposition 2.2.2. *If F is Frechét differentiable, then F is continuous.*

Theorem 2.2.1. *(Chain Rule) If $F : D_f \subset \mathbb{R}^n \to \mathbb{R}^m$ has a Gâteaux derivative at x, and $G : D_G \subset \mathbb{R}^m \to \mathbb{R}^p$ has a Frechét derivative at Fx, then the composite mapping $H = G \circ F$ has a Gâteaux derivative at x, and*

$$H'(x) = G'(Fx)F'(x).$$

If, in addition, $F'(x)$ is a Frechét derivative, then $H'(x)$ is a Frechét derivative as well.

2.2.2 The Mean Value Theorem

Recall the Mean Value Theorem for scalar functions: If $\varphi : [a, b] \subset \mathbb{R} \to \mathbb{R}$ is continuous on $[a, b]$ and differentiable on (a, b), then there exists a $t \in (a, b)$ such that $\varphi(b) - \varphi(a) = \varphi'(t)(b - a)$. It is not too difficult to make sense of a Mean Value Theorem for functionals:

Proposition 2.2.3. *If $F : D \subset \mathbb{R}^n \to \mathbb{R}$ is Gâteaux differentiable at all points in a convex set $D_0 \subset D$, then for each $x, y \in D_0$ there exists a $t \in [0, 1]$ such that $f(y) - f(x) = f'(x + t(y - x))(y - x)$.*

There are several ways to define a Mean Value Theorem for functions $F : \mathbb{R}^n \to \mathbb{R}^m$. The first approach is one in which we apply the previous proposition to each component functional of F. For $F = (f_1, ...f_m)^T$, we apply Proposition 2.2.3 to get $t_1, ..., t_m \in \mathbb{R}$ such that

$$Fy - Fx = \begin{bmatrix} f_1'(x + t_1(y - x)) \\ \vdots \\ f_m'(x + t_m(y - x)) \end{bmatrix} (y - x).$$

Another alternative gives an upper bound for $\|Fy - Fx\|$ in terms of $F'(x)$.

Proposition 2.2.4. *If $F : D \subset \mathbb{R}^n \to \mathbb{R}^m$ is Gâteaux differentiable at all points in a convex set $D_0 \subset D$, then for all $x, y \in D_0$ we have*

$$\|Fy - Fx\| \leq \sup_{t \in [0,1]} \|F'(x + t(y - x))\| \|x - y\|.$$

Corollary 2.2.1. *If $\|F'(x)\| \leq M < \infty$ for all $x \in D_0$ then F is Lipschitz continuous in D_0.*

The third approach to the mean value theorem is based on the Integral Mean Value Theorem. Let $G : [a, b] \subset \mathbb{R} \to \mathbb{R}^m$. Thus, $G = [g_1, \ldots, g_m]$, where each $g_i : [a, b] \subset \mathbb{R} \to \mathbb{R}$. Define $\int_a^b G(t)dt = \begin{bmatrix} \int_a^b g_1(t)dt \\ \vdots \\ g_1(t)dt \end{bmatrix}$. Then letting $F : \mathbb{R}^n \to \mathbb{R}^m$, where $F = [f_1(x) \ldots f_m(x)]^T$, we have $f_i(y) - f_i(x) = \int_0^1 f_i'(x + t(y - x))(y - x)dt$ for each i. To justify this formula we simply take $\phi_i(s) = f_i(x + s(y - x))$ and apply the Fundamental Theorem of Calculus to $\phi(s)$. Provided that each f_i is Riemann integrable, under proper conditions we write

$$Fy - Fx = \int_0^1 F'(x + t(y - x))(y - x)dt.$$

2.2.3 Fixed Point Theory

Given a map F, a fixed point x satisfies $Fx = x$. A common problem is to guarantee that a map has a fixed point, and that the iterative map $x_n = Fx_{n-1}$ converges to a fixed point.

Theorem 2.2.2. *(Banach Fixed Point Theorem) Let X be a Banach space, and M be a closed nonempty subset of X. Let $F : M \to M$ be a map satisfying $\|Fx - Fy\|_X \leq q\|x - y\|_X$ for $0 < q < 1$. Then there is a unique fixed point $x^* \in M$. Furthermore, the sequence $x_n = Fx_{n-1}$ converges to x^* for any initial $x_0 \in M$.*

Provided the Frechét derivative F' exists, we may use the Banach Fixed Point Theorem to obtain:

$$\|F'(x)\| = \left\| \lim_{y \to x} \frac{Fx - Fy}{x - y} \right\| = \lim_{y \to x} \frac{\|Fx - Fy\|}{\|x - y\|} \leq q < 1.$$

2.3 The Adjoint Operator

We recall that for a Hilbert space H, and a linear operator L on H, we use the identity

$$(u, L^*v)_H = (Lu, v)_H,$$

to define the adjoint operator L^*. To obtain the adjoint of a differential operator L, we want to move the derivatives away from u onto v. To do this, we recall the Divergence Theorem, $\int_\Omega \nabla \cdot \mathbf{F} dx = \int_{\partial\Omega} \mathbf{F} \cdot \mathbf{n}\, dS(x)$, where $\mathbf{n}$ is the outward normal pointing unit vector across the surface boundary. By setting $\mathbf{F} = v\nabla u$ and applying the product rule, we obtain Green's First Identity,

$$\int_\Omega v \Delta u \, dx = \int_{\partial\Omega} v\, \partial_\eta u \, dS(x) - \int_\Omega \nabla v \cdot \nabla u \, dx, \tag{2.3.1}$$

where $\partial_\eta u = \nabla u \cdot \mathbf{n}$. Equation (2.3.1) is a generalization of integration by parts to higher dimensions. Also, if $\Omega \subset \mathbb{R}^n$ and there is a matrix $A(x) \in \mathbb{R}^{n\times n}$, then we generalize (2.3.1) to get

$$\int_\Omega \nabla \cdot (A\nabla u)\, v\, dx = \int_{\partial\Omega} v\, A\nabla u \cdot \mathbf{n}\, dS(x) - \int_\Omega A\nabla u \cdot \nabla v\, dx. \tag{2.3.2}$$

2.3.1 A Linear Example

Consider the boundary value problem,

$$\begin{cases} Lu = -\nabla \cdot (A\nabla u) + \mathbf{b} \cdot \nabla u + c\, u = f, & x \in \Omega \\ u = 0, & x \in \partial\Omega. \end{cases} \tag{2.3.3}$$

The goal is to find an operator L^* such that $(Lu, v) = (L^*, v)$ for all $v \in L^2(\Omega)$. We multiply both sides by v and integrate over Ω to obtain

$$(Lu, v) = \int_\Omega (-\nabla \cdot (A\nabla u)\, v + \mathbf{b} \cdot \nabla u\, v + c\, u\, v)\, dx,$$

where we have assumed that v is zero on the boundary $\partial\Omega$. We apply (2.3.2) to the first term and integration by parts to the second term,

$$(Lu, v) = \int_\Omega (A\nabla u \cdot \nabla v - u\, \mathbf{b} \cdot \nabla v + c\, u\, v)\, dx.$$

Applying (2.3.2) again,

$$(Lu, v) = \int_\Omega u(-\nabla \cdot (A\nabla v) - u\, \mathbf{b} \cdot \nabla v + c\, u\, v)\, dx = (u, L^* v).$$

We thus obtain the adjoint equation,

$$\begin{cases} L^* v = -\nabla \cdot (A\nabla v) - \mathbf{b} \cdot \nabla v + c\, v = \psi, & x \in \Omega \\ v = 0 & x \in \partial\Omega. \end{cases} \tag{2.3.4}$$

Remark 2.3.1. *From the previous example and in view of Proposition 2.1.1, we see that the operator $Lu = -\nabla \cdot (a\nabla u) + c\, u$ is self-adjoint. That is, $L = L^*$.*

Suppose we want to evaluate $u(y_0)$ for some $y_0 \in \Omega$. We solve the adjoint problem (2.3.4) by setting the data $\psi = \delta_{y_0}$. Thus, we obtain

$$u(y_0) = (u, \delta_{y_0}) = (u, L^*v) = (Lu, v) = (f, v).$$

This is called the method of Green's function.

2.3.2 A Semilinear Example

As the adjoint is defined for linear operators, we may construct a unique adjoint to linear differential equations as done in the previous example. Suppose we have a problem with a nonlinear right-hand side,

$$\begin{cases} -\nabla \cdot (A\nabla u) = f(u), & x \in \Omega \\ u = 0, & x \in \partial\Omega. \end{cases} \tag{2.3.5}$$

Since the right hand side depends on the solution u, we need to include it as part of the operator. That is, we let

$$Lu = -\nabla \cdot (A\nabla u) - f(u).$$

The problem is that if f is nonlinear, then L is not linear with respect to u. Provided that the nonlinear operator is a map between Banach spaces with a convex domain, we may choose one of several ways to define the adjoint. In general, we want to estimate a linear functional of the error $e = u - U$, where U is an approximation to the true solution u. To obtain a linear operator when L is Frechét differentiable, we use the Integral Mean Value Theorem to write

$$L(u) - L(U) = \int_0^1 L'(tu + (1-t)U)\, dt(u - U),$$

where L' is the Jacobian of L. We define the "average" Jacobian

$$\overline{L'} = \int_0^1 L'(tu + (1-t)U)\, dt,$$

a linear operator which is, in a sense, the average of $L'(u)$ and $L'(U)$. We take the adjoint of the nonlinear operator L to be the adjoint of the linear operator $\overline{L'}$ using the standard identity $(\overline{L'}e, \phi) = (e, \overline{L'}^*\phi)$. This adjoint operator is impractical because it requires the exact solution u. To define a computable adjoint, we instead simply linearize L about U, i.e., replace $\overline{L'}$ with $L'(U)$. To take this approach for (2.3.5), we take the Gâteaux derivative as follows.

$$\begin{aligned} L'(U)(v) &= \lim_{h\to 0} \frac{1}{h}(L(U+hv) - LU) \\ &= \lim_{h\to 0} \frac{1}{h}\left(-\nabla\cdot(A\nabla(U+hv)) - f(U+hv) + \nabla\cdot(A\nabla U) + f(U)\right) \\ &= \lim_{h\to 0} \frac{1}{h}\left(-h\nabla\cdot(A\nabla v) - D_u f(U)\, h\, v + \mathbf{o}(h^2)\right) \\ &= -\nabla\cdot(A\nabla v) - D_u f(U)\, v. \end{aligned} \tag{2.3.6}$$

We define the adjoint of L as the adjoint of the linear operator $L'(U)$. That is, we search for $(L'(U))^*$ such that $(L'(U)v, w) = (v, (L'(U))^* w)$. By Remark 2.3.1, with $c = -f'(U)$, we see that $L'(U)$ is self adjoint. Hence,

$$(L'(U)(w))^* = -\nabla\cdot(A\nabla w) - D_u f(U)\, w. \tag{2.3.7}$$

2.4 The Diffusion Equation

To motivate the diffusion equation, consider a domain $\Omega \subset \mathbb{R}^3$ containing a contaminant with concentration $u(x,t)$ at position $x \in \Omega$ and time $t > 0$. Let $F(x,t)$ be the flow rate of the medium, which is a vector quantity $F = (F_1, F_2, F_3)$, where $F_i(x,t)$ is the rate of flow in the x_i direction at position x and time t.

We start with the following basic conservation principle. In a region Ω, the total change in mass of a contaminant during the interval $t_1 < t < t_2$

is equal to the total outward flux of the contaminant across the boundary of the region,

$$\int_\Omega u(x,t_2)dx - \int_\Omega u(x,t_1)\,dx = -\int_{t_1}^{t_2}\int_{\partial\Omega} F(x,t)\cdot\eta\,d\sigma(x)\,dt.$$

Using the Fundamental Theorem on the left, and the Divergence Theorem on the right,

$$\int_{t_1}^{t_2}\int_\Omega u_t(x,t)\,dx\,dt = -\int_{t_1}^{t_2}\int_\Omega \nabla\cdot F\,dx\,dt,$$

or

$$\int_{t_1}^{t_2}\int_\Omega (u_t + \nabla\cdot F)\,dx\,dt = 0,$$

for all $\Omega, (t_1,t_2)$. Then,

$$u_t + \nabla\cdot F = 0 \text{ for almost every } x.$$

According to Fick's Law, particles flow from regions of high concentration to regions of low concentration depending on ∇u. In other words, the flow rate is proportional to the rate of change of u,

$$F = -D(x)\nabla u.$$

Substituting this into the above, we obtain

$$u_t - \nabla\cdot(D(x)\nabla u) = 0.$$

If we consider the addition of a particle source $f(x,t)$, the diffusion equation is given by

$$\partial_t u(x,t) - \nabla\cdot(D(x)\nabla u(x,t)) = f(x,t). \tag{2.4.1}$$

If $D(x)$ is constant, we get a simplified case called the heat equation,

$$\partial_t u - D\Delta u = f(x,t).$$

We also note that if $u(x,t)$ represents the temperature in a region, then the above derivation is equally valid, and in this case $D(x)$ represents the thermal conductivity. In order for the problem to be solvable, we generally specify an initial condition $u(x,0) = u_0(x)$ and boundary conditions $u(x,t) = u_1(x,t)$ or $D\partial_\eta u(x,t) = u_1(x,t)$ for $x \in \partial\Omega, t \geq 0$.

In the case of particle diffusion, u denotes concentration of a contaminant, and D the corresponding diffusion coefficient. To demonstrate, consider the initial value problem

$$\begin{cases} u_t(t,x) - \frac{1}{2}Du_{xx}(t,x) = 0, & t > 0, x \in \mathbb{R} \\ u(0,x) = \delta_0(x). \end{cases} \tag{2.4.2}$$

This problem models all of the particles lying at $x = 0$ at time $t = 0$. The coefficient D controls how quickly the particles disperse from $x = 0$. The solution to (2.4.2) is

$$u(t,x) = \frac{1}{\sqrt{2\pi Dt}} \exp\left(-\frac{x^2}{2Dt}\right),$$

a Gaussian with mean zero and standard deviation $\sqrt{Dt}$. Hence, the particles spread out with increasing time. In addition, increasing the diffusion coefficient D results in the particles spreading out at a faster rate.

If we expect the concentration $u(x,t)$ to settle to some equilibrium value $u(x)$ after a long time, we set the derivative $\partial_t u = 0$ to obtain the stationary state equation,

$$-\nabla \cdot (D(x)\nabla u(x)) = f(x). \tag{2.4.3}$$

We note that (2.4.3) is an example of an elliptic equation and will be a topic of discussion for much of this thesis.

2.5 Elliptic Problems

Consider the differential operator having the form

$$Lu = -\sum_{i,j=1}^{n} \partial_{x_j}(a_{ij}(x)\partial_{x_i}) + \sum_{i=1}^{n} b_i(x)\partial_{x_i}u + c(x)u, \tag{2.5.1}$$

or equivalently,

$$Lu = -\nabla \cdot (A(x)\nabla u) + \mathbf{b}(x) \cdot \nabla u + c(x)u.$$

We say L is an elliptic operator if the matrix $A = [a_{ij}(x)]$ is a real positive definite, symmetric matrix. In particular, for arbitrary $\xi \in \mathbb{R}^n$, L satisfies the ellipticity condition

$$\sum_{i,j=1}^{n} a_{ij}(x)\xi_i\xi_j > 0. \tag{2.5.2}$$

In general we make the stronger assumption of uniform ellipticity,

$$\sum_{i,j=1}^{n} a_{ij}(x)\xi_i\xi_j > C|\xi|^2. \tag{2.5.3}$$

If A is symmetric and in addition (2.5.3) holds for all $\xi \in \mathbb{R}^n$, we say that L is uniformly elliptic. If this holds, then the smallest eigenvalue of A is greater than or equal to C for all x.

For example, if $a_{ij} = \delta_{ij}, b_i = 0, c = 0$, then $L = -\Delta$, a uniformly elliptic operator. An equation of the form $-\Delta u = f$ is called Poisson's equation, and an equation of the form $-\Delta u = 0$ is called Laplace's equation.

2.5.1 Applications of Elliptic Problems

Poisson's equation has numerous applications in areas such as electrostatics, elasticity, fluid mechanics, and statistical physics.

In electrostatics, we study the electrostatic potential, or voltage in a region $\Omega \subset \mathbb{R}^3$. From the Maxwell equations we have $\nabla \cdot E = \rho$ in Ω, where

$E(x)$ is the electric field. From Faraday's law, E is a conservative field, so that $E = \nabla u$ for some scalar electric potential u. This leads to the Poisson equation $\Delta u = \rho$. In addition, we assume that $u|_{\partial\Omega} = c$ for some constant c, so that $\partial\Omega$ is a perfectly conducting surface.

In fluid mechanics, we have a velocity field $F(x)$ of fluid in a region Ω. We assume that the fluid is rotation-free, so that the field is conservative. Then we have $u = \nabla F$ for some velocity potential. If the fluid is incompressible, then $\nabla \cdot u = 0$, and we obtain the Laplace equation $\Delta u = 0$.

We may also apply the Laplace equation to Brownian motion. Consider a randomly moving particle within a region Ω that moves at random until it hits the boundary Γ. We divide the boundary so that $\Gamma = \Gamma_1 \cup \Gamma_2$, and let $u(x)$ be the probability that a particle starting at x ends up on Γ_1. Then it turns out that u solves

$$\begin{cases} \Delta u = 0 & x \in \Omega, \\ u = 1, & x \in \Gamma_1 \\ u = 0, & x \in \Gamma_2. \end{cases}$$

2.5.2 Sobolev Spaces

Before continuing, we review some basic facts about Sobolev spaces. These turn out to be useful in studying partial differential equations, and are the spaces we use in solving finite element problems. The vector space $L^2(\Omega)$ is a Hilbert space defined by

$$L^2(\Omega) = \left\{ u(x) | \int_\Omega |u(x)|^2 \, dx < \infty \right\},$$

where the integral is defined in the Lebesgue sense. The $L^2(\Omega)$ inner product is defined as

$$(u, v)_{L^2(\Omega)} = \int_\Omega u(x)\overline{v(x)} \, dx,$$

and the induced norm is

$$\|u(x)\|_{L^2(\Omega)} = \left(\int_\Omega |u(x)|^2 \, dx \right)^{1/2}.$$

Before introducing more general Sobolev spaces, we first review the notion of a weak derivative. Let $C_C^\infty(\Omega)$ be the space of infinitely differentiable functions $\phi : \Omega \to \mathbb{R}$ with compact support in Ω. Functions $\phi \in C_C^\infty(\Omega)$ are often called test functions.

Definition 2.5.1. *A function $v : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ belongs to the space $L^1_{loc}(\Omega)$ if for every $x \in \Omega$, there is an open neighborhood N containing x such that $\bar{N} \subset \Omega$ and v is integrable on N.*

Definition 2.5.2. *Suppose $u, v \in L^1_{loc}(\Omega)$, and let α be a multiindex. We say that v is the α^{th} weak partial derivative of u, and write $D^\alpha u = v$, if*

$$\int_\Omega u \, D^\alpha \phi \, dx = (-1)^{|\alpha|} \int_\Omega v \, \phi \, dx \text{ for all } \phi \in C_C^\infty(\Omega).$$

One may show that weak derivatives are unique up to a set of measure zero.

Example 2.5.1. *Let $\Omega = (0, 1)$, and define*

$$u = \begin{cases} x & 0 < x \leq 1 \\ 1 & 1 < x < 2, \end{cases} \qquad v = \begin{cases} 1 & 0 \leq x \leq 1 \\ 0 & 1 < x < 2. \end{cases}$$

Note that u is not differentiable at $x = 1$; however, u has weak derivative v in the weak sense. To show this, let $\phi \in C_C^\infty(\Omega)$. Then,

$$\begin{aligned} \int_0^2 u \, \phi' \, dx = \int_0^1 x \, \phi' \, dx + \int_1^2 \phi' \, dx &= - \int_0^1 \phi \, dx + \phi(1) + \phi(2) - \phi(1) \\ &= - \int_0^1 \phi \, dx = - \int_0^2 v \, \phi \, dx, \end{aligned}$$

as desired.

For $s \in \mathbb{Z}^+$, we define the Sobolev space $H^s(\Omega)$ to be the set of functions such that the weak derivative D^α exists and is in $L^2(\Omega)$ for each multiindex α with $|\alpha| \leq k$. That is,

$$H^s(\Omega) = \{u(x) | \partial^\alpha u \in L^2(\Omega), \text{ for all } |\alpha| \leq s\},$$

where $\alpha = (\alpha_1, ..., \alpha_n)$. It may be shown that H^s is a Hilbert space, with the inner product and norms defined to be

$$\|u\|_{H^s(\Omega)} = \left(\sum_{|\alpha|=0}^{s} \|\partial^\alpha u(x)\|^2_{L^2(\Omega)} \right)^{1/2},$$

$$(u, v)_{H^s(\Omega)} = \sum_{|\alpha|=0}^{s} (\partial^\alpha u(x), \partial^\alpha v)_{L^2(\Omega)}.$$

Note in particular, that $L^2(\Omega) = H^0(\Omega)$. Also, for $s = 1$, we have

$$\|u(x)\|_{H^1(\Omega)} = \left(\|u(x)\|^2_{L^2(\Omega)} + \sum_{j=1}^{n} \|\partial_{x_j} u(x)\|^2_{L^2(\Omega)} \right)^{1/2}.$$

We define $H^1_0(\Omega)$ as the set of functions $u \in H^1(\Omega)$ such that $u|_{\partial\Omega} = 0$. The set $H^1_0(\Omega)$ is a subspace of $H^1(\Omega)$ and shares its norm and inner product.

Remark 2.5.1. *By Poincare's inequality [23],*

$$\|u\|^2_{H^1_0(\Omega)} \leq c\|\nabla u\|^2_{L^2(\Omega)}. \tag{2.5.4}$$

Consequently, it may be shown that $\|\nabla u\|_{L^2(\Omega)}$ is an equivalent norm for the set $H^1_0(\Omega)$.

2.5.3 Existence and Uniqueness

Consider the elliptic boundary value problem

$$\begin{cases} -\nabla \cdot (A(x)\nabla u) + \mathbf{b}(x) \cdot \nabla u + c(x)u = f(x), & x \in \Omega, \\ u = 0 & x \in \partial\Omega, \end{cases} \tag{2.5.5}$$

where $\Omega \subset \mathbb{R}^n$, and $A = a\, I_n$. To obtain uniform ellipticity, we require that $a \geq a_0 > 0$.

Definition 2.5.3. *Given a vector space V, a bilinear form is a function $B : V \times V \to \mathbb{R}$, that is linear in each argument.*

Definition 2.5.4. *We say that $u \in H_0^1(\Omega)$ is a weak solution to (2.5.5) if*

$$B(u,v) \equiv \int_\Omega A\nabla u \cdot \nabla v + \mathbf{b} \cdot \nabla u\, v + c\, u\, v\, dx = (f,v), \tag{2.5.6}$$

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

Definition 2.5.5. *A bilinear map B is bounded if there exists a constant $\alpha > 0$ such that*

$$|B(u,v)| \leq \alpha \|u\|_H \|v\|_H \text{ for all } u, v \in H.$$

Definition 2.5.6. *A bilinear map B is coercive if there exists a constant $\beta > 0$ such that*

$$\beta \|u\|_H^2 \leq B(u,u) \text{ for all } u \in H. \tag{2.5.7}$$

For example, if $Lu = \Delta u$, then (2.5.7) means that

$$\beta \|u\|_{L^2(\Omega)}^2 \leq \|\nabla u\|_{L^2(\Omega)}^2,$$

which requires the derivative to be bounded away from zero.

Theorem 2.5.1. *(Lax-Milgram Lemma) Let H be a real Hilbert space, and $B : H \times H \to \mathbb{R}$ be a coercive, bounded, bilinear mapping on H. If $f : H \to \mathbb{R}$ is a bounded linear functional on H (i.e. $f \in H^*$), then there exists a unique element $u \in H$ such that*

$$B(u,v) = \langle f, v \rangle \text{ for all } v \in H.$$

Furthermore there is a constant C, independent of f, such that

$$\|u\|_H \leq C \|f\|_{H^*}.$$

This theorem relies on the Riesz Representation Theorem applied to the mapping $v \rightarrow B(u, v)$.

To apply the Lax-Milgram Lemma to (2.5.5) to get an existence result, we need the following theorems.

Theorem 2.5.2. *Let $B(u, v)$ be the bilinear form defined in (2.5.6). Then there is an $\alpha > 0$ such that*

$$|B(u, v)| \leq \alpha \|u\|_{H_0^1(\Omega)} \|v\|_{H_0^1(\Omega)}.$$

The proof of the following result is given in Appendix A.

Theorem 2.5.3. *If $-\frac{1}{2}\nabla \cdot \mathbf{b} + c \geq 0$, then B is coercive.*

We look for solutions $u \in H_0^1(\Omega)$. We therefore require that $f \in H^{-1}(\Omega) = (H_0^1(\Omega))^*$, the dual space of $H_0^1(\Omega)$. Then provided that $-\frac{1}{2}\nabla \cdot \mathbf{b} + c \geq 0$, the Lax-Milgram lemma (2.5.1) guarantees that there is a unique $u \in H_0^1(\Omega)$ satisfying

$$B(u, v) = \langle f, v \rangle \text{ for all } v \in H_0^1(\Omega).$$

We list some basic regularity results for the case that f has higher regularity than $H^{-1}(\Omega)$.

Theorem 2.5.4. *Suppose $a, b, c \in C^\infty(\Omega), f \in C^\infty(\Omega)$, and $u \in H_0^1(\Omega)$ is a weak solution to (2.5.5). Then $u \in C^\infty(\Omega)$.*

Theorem 2.5.5. *Suppose that $a \in C^1(\overline{\Omega}), b_i, c \in L^\infty(\Omega), f \in L^2(\Omega)$, and $u \in H_0^1(\Omega)$ is a weak solution to (2.5.5). Then $u \in H^2(\Omega)$, and*

$$\|u\|_{H^2(\Omega)} \leq C \left(\|f\|_{L^2(\Omega)} + \|u\|_{L^2(\Omega)} \right),$$

for some constant C depending on . In addition, if $u \in H_0^1(\Omega)$ is the unique weak solution, then

$$\|u\|_{H^2(\Omega)} \leq C \|f\|_{L^2(\Omega)}.$$

Theorem 2.5.6. *(Weak Maximum Principle) If $\overline{\Omega}$ is compact, and $u \in C^2(\Omega) \cup C(\overline{\Omega})$ satisfies $\Delta u \geq 0$ in Ω, then*

$$\max_{x \in \overline{\Omega}} u(x) = \max_{x \in \partial\Omega} u(x).$$

The theorem is sometimes used to give a uniqueness result, for if $u = v$ on $\partial\Omega$, then $u = v$ on all of $\overline{\Omega}$.

The Weak Maximum Principle also gives a useful continuous dependence result. Consider the problem

$$\begin{cases} -\Delta u = f(x), & x \in \Omega, \\ u = g(x), & x \in \partial\Omega, \end{cases}$$

where f, g are continuous. Define the perturbed problem,

$$\begin{cases} -\Delta w = f(x) & x \in \Omega \\ w = h(x) & x \in \partial\Omega. \end{cases}$$

By the maximum principle,

$$\max_{x \in \overline{\Omega}} (u - w) = \max_{x \in \partial\Omega} (g - h).$$

Consequently, if $|g - h| \leq \epsilon$ for all $x \in \partial\Omega$, then $|u - w| \leq \epsilon$ for all $x \in \overline{\Omega}$. This says that u depends continuously on the Dirichlet condition on the boundary.

Chapter 3

BACKGROUND: FINITE ELEMENTS AND ERROR ESTIMATION

3.1 The Finite Element Method

The finite element method [11, 28, 17] is one of the standard methods for approximating the solution of a differential equation. It gives a system of equations which may be solved using a computer to produce an approximate solution. The method is very powerful in that it can be used for a variety of domains with complex geometries. It is also desirable because we can use *a posteriori* error estimation to improve the error in a quantity of interest while minimizing the extra computational effort required to do so.

3.1.1 A Two Point Boundary Value Example

In this section, we give a brief overview of solving a two point boundary value problem using the finite element method. Consider the problem

$$\begin{cases} -(a(x)u(x)')' = f(x), & x \in (0,1), \\ u(1) = g, \\ -u'(0) = h. \end{cases} \tag{3.1.1}$$

The finite element method amounts to approximating the solution u with a function U that solves the differential equation "on average". That

is, we require that

$$\int_0^1 ((-aU')' - f)\, v\, dx = 0 \text{ for all } v \in V_W. \tag{3.1.2}$$

The functions $v \in V_W$ are called test functions, or weighing functions, and the set V_W is called the test space. Thus, we do not require the residual error $-(aU')' - f$ is zero, but rather that the residual error is orthogonal to the set V_W. This condition is known as Galerkin orthogonality.

The finite element method also amounts to solving the weak form of (3.1.1). That is, we integrate by parts, and multiply both sides by a function $v \in V_W$. Specifically, we find $U \in V_T$ satisfying

$$\int_0^1 a\, U'\, v'\, dx = \int_0^1 f\, v\, dx \text{ for all } v \in V. \tag{3.1.3}$$

The space V_T is the set of trial solutions. Using the weak formulation has several consequences. First, to solve (3.1.3), U needs one less derivative than needed to satisfy (3.1.1). This means that the finite element solution U may not even be twice differentiable. Also, we may integrate U' even if it is discontinuous at isolated points.

For the example (3.1.1) posed above, let $V_T = \{u \in H^1 | u(1) = g\}$ be the set of trial solutions, and $V_W = \{w \in H^1 | w(1) = 0\}$ be the set of weighing functions. To derive the weak formulation, we multiply both sides of (3.1.1) by $w \in V_W$ and integrate over $(0, 1)$ to get

$$-\int_0^1 (a\, u')' w\, dx = \int_0^1 f\, w\, dx.$$

Integrating by parts, we get

$$-a(1)u'(1)w(1) - a(0)u'(0)w(0) + \int_0^1 a\, u'\, w'\, dx = \int_0^1 f\, w\, dx.$$

Next, since $w(1) = 0$ and $u'(0) = h$, we have

$$-h\,a(0)w(0) + \int_0^1 a\,u'\,w'\,dx = \int_0^1 f\,w\,dx.$$

The finite element formulation to find $u \in V_T$ such that

$$B(u,w) = (f,w) + h\,A(0)\,w(0) \text{ for all } w \in V_W, \tag{3.1.4}$$

where $B(u,v) \equiv \int_0^1 a\,u'\,v'\,dx$ and $(u,v) \equiv \int_0^1 u\,v\,dx$. Equation (3.1.4) is called the weak, or variational formulation.

We will use the discretization of V_T and V_W over some mesh $\mathcal{T}_h = \{x_0, x_1, ..., x_{N+1}\}$, where $x_0 = 0 < x_1 < ... < x_N < x_{N+1} = 1$. Call these spaces V_T^h and V_W^h. For example, we may discretize the spaces over $\mathcal{T}$ by the space of piecewise polynomials of degree q with specified values on mesh points in $\mathcal{T}_h$. The weak formulation of the discretized problem is to find $u^h \in V_T^h$ such that

$$B(u^h, w^h) = (f, w^h) + ha(0)w^h(0) \text{ for all } w^h \in V_W^h.$$

However, since V_T and V_W only differ at $x = 1$, we have

$$V_T^h = \{u^h | u^h = v^h + g^h, v^h \in V_W^h, g^h(1) = g\}.$$

We express the weak formulation in terms of searching for functions in V_W^h instead of V_T^h. This has the advantage of allowing us to search for solutions in the same space as the space of weighting functions. Then, the weak problem is to find $v^h \in V_W^h$ such that

$$B(v^h, w^h) = (f, w^h) + ha(0)w^h(0) - B(g^h, w^h) \text{ for all } w^h \in V_W^h. \tag{3.1.5}$$

Let $\{\phi_i\}_{i=1}^N$ be basis functions for V_W^h. Note that this implies that $\phi_i(1) = 0$ for all i. Then, $w^h = \sum_{i=1}^N w_i\phi_i$, and $v^h = \sum_{i=1}^N v_i\phi_i$, for some

$\{w_i\}_{i=1}^N, \{v_i\}_{i=1}^N$. We introduce another function ϕ_{N+1} with the property $\phi_{N+1}(1) = 1$. Then $g^h(x) = g\phi_{n+1}(x)$, so that $g^h(1) = g$. Hence, for any $u \in V_T^h$, we have

$$u^h = v^h + g^h = \sum_{i=1}^{N} v_i\phi_i + g\phi_{n+1}.$$

Rewritting (3.1.5), we must find $u^h \in V_T^h$ such that for all $w^h \in V_W^h$,

$$B\left(\sum_{j=1}^{N} v_j\phi_j, \sum_{i=1}^{N} w_i\phi_i\right) = \left(f, \sum_{i=1}^{N} w_i\phi_i\right) + ha(0)\sum_{i=1}^{N} w_i\phi_i(0) - B\left(g\phi_{n+1}, \sum_{i=1}^{N} w_i\phi_i\right).$$

Then,

$$\sum_{i=1}^{N} w_i B\left(\phi_i, \sum_{j=1}^{N} v_j\phi_j\right) = \sum_{i=1}^{N} w_i(f, \phi_i) + \sum_{i=1}^{N} w_i\phi_i(0)ha(0) - \sum_{i=1}^{N} w_i B(\phi_{n+1}, \phi_i)g.$$

Since this holds for all $\{w_i\}_{i=1}^N$,

$$B\left(\phi_i, \sum_{j=1}^{N} v_j\phi_j\right) = (f, \phi_i) + ha(0)\phi_i(0) - B(\phi_i, \phi_{n+1})g,$$

or

$$\sum_{j=1}^{N} v_j B(\phi_i, \phi_j) = (f, \phi_i) + ha(0)\phi_i(0) - B(\phi_i, \phi_{n+1})g \tag{3.1.6}$$

for all $i = 1, 2, ..., N$. Define the $N \times N$ stiffness matrix

$$A_{ij} = B(\phi_i, \phi_j),$$

for $i, j = 1, 2, ...n$ and the N dimensional load vector

$$F_i = (f, \phi_i) + ha(0)\phi_i(0) - B(\phi_i, \phi_{N+1})\, g$$

for $i = 1, 2, ..., N$. Then if the desired weights $V = (v_1, ..., v_n)^T$, we write (3.1.6) as

$$AV = F.$$

For example, let V_T^h be defined as the space of function that are linear on each interval of the mesh $\mathcal{T}$, $[x_i, x_{i+1}]$ for $i = 2, ..., N$. In this case, the linear basis functions $\{\phi_i\}_{i=1}^N$ are defined by

$$\phi_i = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}}, & x \in (x_{i-1}, x_i], \\ \frac{x_{i+1} - x}{x_{i+1} - x_i}, & x \in (x_i, x_{i+1}], \\ 0, & \text{otherwise}, \end{cases}$$

for $i = 2, ..., N$,

$$\phi_1 = \begin{cases} \frac{x_2 - x}{x_2 - x_1}, & x \in [x_1, x_2], \\ 0, & \text{otherwise}, \end{cases}$$

and

$$\phi_{N+1} = \begin{cases} \frac{x - x_N}{x_{N+1} - x_N}, & x \in (x_N, x_{N+1}], \\ 0, & \text{otherwise}. \end{cases}$$

These basis functions are shown in Fig. 3.1. Note that $\phi_i(x_j) = \delta_{ij}$. Consequently the stiffness matrix A has values given by

$$A_{ij} = \begin{cases} B(\phi_i, \phi_i) & j = i \\ B(\phi_i, \phi_{i+1}) & j = i + 1 \\ B(\phi_i, \phi_{i-1}) & j = i - 1 \\ 0 & \text{otherwise}, \end{cases}$$

and the load vector F has values

$$F_i = \begin{cases} (f, \phi_1) + h\, a(0)\phi_1(0) & i = 1 \\ (f, \phi_i) & 2 \le i \le N - 1 \\ (f, \phi_N) - B(\phi_N, \phi_{N+1})g & i = N. \end{cases}$$

We note that A is a sparse matrix due to the fact that the basis functions ϕ_i are zero on most nodes. The matrix is also banded. For this example, since V_T^h and V_W^h consist of functions that are linear on each interval, then all but the main and off-diagonal elements of A are zero. In addition, since

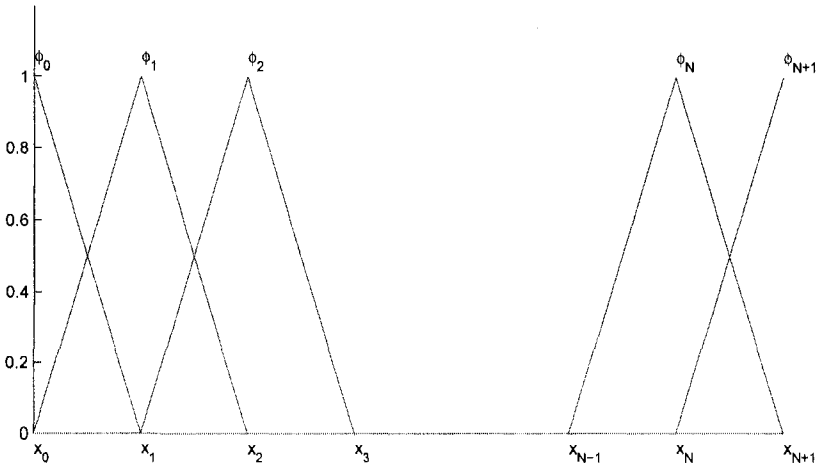

Figure 3.1: Piecewise linear basis functions $\{\phi_i\}_{i=1}^{N+1}$ in 1D.

$a(\phi_i, \phi_j)$ is symmetric, A will be a symmetric matrix. It also turns out that A is positive definite, hence invertible. The nice properties of A add to the appeal of the finite element method. In particular, due to the sparseness of A, we can quickly solve $AV = F$ using an iterative technique.

We solve the problem

$$\begin{cases} -\nabla \cdot (A\nabla u) = f & x \in \Omega \\ u = 0 & x \in \partial\Omega, \end{cases}$$

for a domain $\Omega \subset \mathbb{R}^n$ in higher dimensions by following the same general technique as above. For a problem in two spatial dimensions, we typically assume that Ω is a polygonal domain consisting of triangular elements. We say the set of triangles that partition $\overline{\Omega}$ is a triangulation of Ω if no vertex of any triangle lies in the interior of an edge of another triangle. In other words, the mesh must contain no hanging nodes. In this case, we let $\mathcal{T}_h$ denote a triangulation of Ω, h_k denote the length of the largest edge of

$k \in \mathcal{T}_h$, and set $h(x) = h_k$. Numerous texts such as [11, 28, 17] explain the generalizations to other types of problems or problems of higher dimensions.

3.1.2 A Semilinear Partial Differential Equation

Consider the problem

$$\begin{cases} -\nabla \cdot (A\nabla u) = f(u) & x \in \Omega \\ u = 0 & x \in \partial\Omega. \end{cases} \tag{3.1.7}$$

The weak version of the problem is to find $U \in V_h$ such that

$$\int_\Omega (A\nabla U \cdot \nabla v - f(U)\, v)\, dx = 0 \tag{3.1.8}$$

for all $v \in V_h$. We cannot apply the analysis above for a general nonlinearity f. We can, however, approximate the solution to (3.1.8) by using a fixed point technique applied to the operator $F(u) = -\nabla \cdot (A\nabla u) - f(u) + u$. However, convergence of the method requires that the norm of the derivative is bounded by 1, $\|F'(U)\| \leq C < 1$. The success of this method is therefore fairly limited, since this bound will typically not hold.

A better option is to apply Newton's method to solve (3.1.7). Recall that in order to solve $F(x) = 0$, we start with an initial guess x_0 and iterate:

$$x_i = x_{i-1} - \frac{F(x_{i-1})}{F'(x_{i-1})},$$

until $\|x_i - x_{i-1}\|$ is sufficiently small. Equivalently, we repeatedly solve $F'(x_{i-1})w_i = -F'(x_{i-1})$, and set $u_i = u_{i-1} + w_i$ until $\|w_i\|$ is small.

We apply this technique to solving

$$L(u) = -\nabla \cdot (A\nabla u) - f(u) = 0. \tag{3.1.9}$$

Newton's method is to find $w_i = u_i - u_{i-1}$ such that

$$L'(u_{i-1})w_i = -L(u_{i-1}), \tag{3.1.10}$$

for each iteration i. To linearize L, we find the Gâteaux deriviative $L'(u)$ in the direction v. Following the argument in (2.3.6),

$$L'(u)v = -\nabla \cdot (A\nabla v) - D_u f(u)v.$$

We then write (3.1.10) as

$$-\nabla \cdot (A\nabla w_i) - D_u f(u_{i-1})w_i = \nabla \cdot (A\nabla u_{i-1}) + f(u_{i-1}).$$

Since $w_i = u_i - u_{i-1}$,

$$-\nabla \cdot (A\nabla(u_i - u_{i-1})) - D_u f(u_{i-1})(u_i - u_{i-1}) = \nabla \cdot (a\nabla u_{i-1}) + f(u_{i-1}),$$

or simply

$$-\nabla \cdot (A\nabla u_i) - D_u f(u_{i-1})(u_i) = -D_u f(u_{i-1})u_{i-1} + f(u_{i-1}).$$

Putting this into weak form gives the algorithm:

Choose initial guess u_0.

While $||u_i - u_{i-1}|| >$ TOL or $i >$ MAX ITS

Find $u_{i+1} \in V$ such that $\int_\Omega (A\nabla u_i \cdot \nabla v - D_u f(u_{i-1})u_i\, v)\, dx =$ $\int_\Omega (-D_u f(u_{i-1})u_{i-1} + f(u_i))\, v\, dx$ for all $v \in V_h$.

End while

3.2 *A Posteriori* Error Estimation

In this section, we define a representation for the error in a quantity of interest computed from an elliptic problem, using techniques of *a posteriori* analysis (see [18] or [22] for example). First, we provide an overview of a posteriori error estimation. Consider a simple model problem, where

$A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, and we would like to find $x \in \mathbb{R}^n$ satisfying $Ax = b$. Given an approximation X to x, define the error to be $e = x - X$, and assume that we want to estimate the value of the error functional (e, ψ) for some $\psi \in \mathbb{R}^n$. Then, consider the solution $\phi \in \mathbb{R}^n$ of the associated adjoint problem $A^T\phi = \psi$. This leads us to

$$|(e,\psi)| = |(e, A^T\phi)| = |(Ae,\phi)| = |(b - AX,\phi)| \leq \sum_{i=1}^{n} |b_i - AX_i| \cdot |\phi_i|, \tag{3.2.1}$$

the weighted *a posteriori* estimate. The residuals $\{b_i - AX_i\}_{i=1}^n$ are easy to compute, and the weights $\{\phi_i\}_{i=1}^n$ tell us about the influence of the local residuals on the error of (e, ψ).

3.2.1 A Linear Elliptic Problem

Consider the boundary value problem

$$\begin{cases} -\nabla \cdot (A(x)\nabla u) = f(x) & x \in \Omega \\ u = 0 & x \in \partial\Omega, \end{cases} \tag{3.2.2}$$

where $\Omega \subset \mathbb{R}^n$ is a convex and polygonal domain, f is a smooth function, and $A(x)$ is a symmetric, positive definite matrix with smooth entries such that $y^TAy > a_0\|y\|^2$ for all y. Let $e = u - U$ be the difference between the true solution u and the computed solution U. In general, it is computationally inefficient and impractical to compute the pointwise error at each point in the domain. In many applications, it is useful to compute the average error of the solution against some weighting function ψ. The function ψ reflects the region in Ω in which we would like to gain information about the error. Specifically, we compute (e, ψ), a linear function of the error weighted by $\psi \in L^2(\Omega)$. For example, if $\psi \equiv 1$, then (e, ψ)

gives the average error. If $\psi = \delta_y(x)$, then (e, ψ) gives the error at point y, $u(y) - U(y)$. Denote the linear operator L by

$$Lu = -\nabla \cdot (A\nabla u). \tag{3.2.3}$$

We use the following notation:

$$(u, v) = \int_\Omega u\, v\, dx, \qquad (A\nabla u, \nabla v) = \int_\Omega A\nabla u \cdot \nabla v\, dx.$$

Since L is linear, we define the adjoint operator L^* so that $(Lu, v) = (u, L^*v)$ for all $v \in L^2(\Omega)$. The adjoint equation takes on the form $L^*\phi = \psi$. We use an analogous argument to (3.2.1):

$$|(e, \psi)| = |(e, L^*\phi)| = |(Le, \phi)| = |(f - LU, \phi)| \leq \|f - LU\|\|\phi\|,$$

where we have used the Cauchy-Schwartz inequality. Thus, we are able to obtain an upper bound for the error after the solution U has been computed. To demonstrate, we derive an error estimation formula for (3.2.2).

From (2.3.4), we have $L^*v = -\nabla \cdot (A\nabla v)$. To derive (2.3.4), recall that we assumed that $v = 0$ on the boundary. The adjoint equation for this problem is then

$$\begin{cases} -\nabla \cdot (A\nabla\phi) = \psi & x \in \Omega \\ \phi = 0 & x \in \partial\Omega \end{cases}$$

We also define $\pi_h u$ to be the L^2 projection of u onto the finite element space V. Then by Galerkin orthogonality,

$$(A\nabla u, \nabla \pi_h \phi) = (f, \pi_h \phi).$$

Finally, we obtain the error estimate

$$\begin{aligned} (e, \psi) &= (u - U, -\nabla \cdot (A\nabla\phi)) \\ &= (-\nabla \cdot (A\nabla u), \phi) - (A\nabla U, \nabla\phi) \\ &= (f, \phi) - (A\nabla U, \nabla\phi) \\ &= (f, \phi - \pi_h\phi) - (A\nabla U, \nabla(\phi - \pi_h\phi)). \end{aligned}$$

suppose we would like to include the effects of quadrature error. Let $\bar{g}$ be some approximation of g, so that when we integrate $g(x)$ using a quadrature rule, we obtain $\int_\Omega \bar{g}(x)\,dx$ exactly. For example, if we use a trapezoidal rule, $\bar{g}$ is the projection of g onto the space of piecewise linear polynomials. The actual problem to be solved is: Find $U \in V_h$ such that

$$\int_\Omega \overline{a(x)\nabla U \nabla v}\,dx = \int_\Omega \overline{fv}\,dx$$

for all $v \in V_h$. We use the same error formula as before, then add and subtract the above equation with $\pi_h\phi$ substituted for v:

$$\begin{aligned}
\int_\Omega e\,\psi\,dx &= \int_\Omega f\phi\,dx - \int_\Omega A\nabla U\cdot\nabla\phi\,dx \\
&= \int_\Omega f\phi\,dx - \int_\Omega \overline{f\pi_h\phi}\,dx - \int_\Omega A\nabla U\cdot\nabla\phi\,dx + \int_\Omega \overline{A\nabla\cdot U\nabla\phi}\,dx \\
&= \int_\Omega f(\phi-\pi_h\phi)\,dx - \int_\Omega A\nabla U\cdot\nabla(\phi-\pi_h\phi)\,dx \\
&+ \int_\Omega (f\pi_h\phi - \overline{f\pi_h\phi})\,dx - \int_\Omega (A\nabla U\cdot\nabla\pi_h\phi - \overline{A\nabla U\cdot\nabla\pi_h\phi})\,dx.
\end{aligned}$$

The first two terms measure the discretization error and the last two measure the effects of the quadrature error.

3.2.2 A Semilinear Elliptic Problem

In this section, we provide an error representation formula for the elliptic problem

$$\begin{cases} -\nabla\cdot(A(x)\nabla u(x)) = f(u,x;\lambda), & x\in\Omega, \\ u = 0, & x\in\partial\Omega, \end{cases} \tag{3.2.4}$$

where $\Omega \subset \mathbb{R}^n$ is a convex and polygonal domain, f is a smooth function, $A(x)$ is a symmetric, positive definite matrix with smooth entries, $y^T Ay > a_0\|y\|^2$ for all y, and $\lambda \in \Lambda \subset \mathbb{R}^p$ are parameters.

We let V_h denote the space of continuous piecewise linear functions with respect to $\mathcal{T}_h$ that are zero on $\partial\Omega$. The finite element approximation $U \in V_h$ solves

$$(A\nabla U, \nabla v) = (f(U,\lambda), v). \tag{3.2.5}$$

for all $v \in V_h$. We define the adjoint to the linearized problem about U using (2.3.7),

$$\begin{cases} -\nabla \cdot (A\nabla\phi) - D_u^* f(U;\lambda)\phi = \psi, & x \in \Omega, \\ \phi = 0, & x \in \partial\Omega. \end{cases}$$

Thus,

$$\begin{aligned}(e,\psi) &= (u-U, -\nabla\cdot(a\nabla\phi)) - (D_u^* f(U,\lambda),\phi) \\ &= (u-U, -\nabla\cdot(a\nabla\phi)) - (u-U, D_u^* f(U;\lambda)\phi). \end{aligned} \tag{3.2.6}$$

If we integrate by parts on the first term and use the definition of the adjoint on the second term, we obtain

$$(e,\psi) = (A\nabla(u-U), \nabla\phi) - (D_u f(U,\lambda)(u-U),\phi).$$

Next, we expand the first term, then integrate by parts to obtain

$$\begin{aligned}(e,\psi) &= (A\nabla u, \nabla\phi) - (A\nabla U\cdot\nabla\phi) - (D_u f(U,\lambda)(u-U),\phi) \\ &= (\nabla\cdot(A\nabla u),\phi) - (A\nabla U, \nabla\phi) - (D_u f(U,\lambda)(u-U),\phi). \end{aligned} \tag{3.2.7}$$

Using (3.2.4),

$$(e,\psi) = (f(u,\lambda),\phi) - (A\nabla U, \nabla\phi) - (D_u f(U,\lambda)(u-U),\phi).$$

Next, we make use of the Taylor expansion

$$f(u,\lambda) = f(U,\lambda) + D_u f(U,\lambda)(u-U) + R_1,$$

where $R_1 = \mathbf{o}(|u - U|)$, to obtain the representation:

$$(e, \psi) = (f(u, \lambda), \phi) - (A\nabla U, \nabla\phi) - (f(u, \lambda) - f(U, \lambda) + R_1, \phi).$$

We cancel terms and ignore the higher order term (R_1, ϕ) to obtain the estimate:

$$(e, \psi) \approx (-A\nabla U, \nabla\phi) + (f(U, \lambda), \phi). \tag{3.2.8}$$

Note that terms in (3.2.8) do not cancel since ϕ is not necessarily a member of V. However, we introduce a projection of ϕ onto space V, $\pi_h\phi$. Using the orthogonality condition (3.2.5), we have

$$(-A\nabla U, \nabla\pi_h\phi) + (f(U, \lambda), \pi_h\phi) = 0.$$

Substituting these into (3.2.8) gives

$$(e, \psi) \approx (-A\nabla U, \nabla(\phi - \pi_h\phi)) + (f(U, \lambda), (\phi - \pi_h)). \tag{3.2.9}$$

This analysis also holds for the case in which (3.2.4) is a system of equations.

3.3 Adaptive Error Control

In this section, we use the *a posteriori* error estimate as the basis for adaptivity by employing the following standard "optimization framework" [18, 6, 3].

The overall goal of adaptive error control is to generate a mesh with a small number of elements such that for a given tolerance TOL,

$$\text{error in the quantity of interest} = |(e, \psi)| \leq \text{TOL}. \tag{3.3.1}$$

In general, we cannot use (3.3.1) directly since e is unknown. We thus use an error estimate and construct a mesh that satisfies

a posteriori estimate of the error in the quantity of interest $\leq$ TOL.

We write the estimate as a sum of "element contributions" that indicate the contribution to the total error for each element of the mesh. From (3.2.9) for example, we write

$$|(e,\psi)| \approx |(-A\nabla U, \nabla(\phi - \pi_h\phi)) + (f(U,\lambda), (\phi - \pi_h\phi))|$$
$$= \left| \sum_{K\in\mathcal{T}_h} (-A\nabla U, \nabla(\phi - \pi_h\phi))_K + (f(U,\lambda), (\phi - \pi_h\phi))_K \right| \leq \text{TOL}. \quad (3.3.2)$$

If the approximation U satisfies (3.3.2), the solution is acceptable and the refinement process is stopped.

However, if (3.3.2) is not satisfied, we need to determine which elements of the mesh to refine, or if we should increase the order of the element functions. One problem with using (3.3.2) is that there may be significant cancelation between the element contributions. A large positive contribution from one of the elements could cancel with a large negative contribution from another element. The standard remedy is to introduce norms, to get the acceptance criterion

$$|(e,\psi)| \leq \sum_{K\in\mathcal{T}_h} |-(A\nabla U, \nabla(\phi - \pi_h\phi))_K + (f(U,\lambda), \phi - \pi_h\phi)_K| \leq \text{TOL}. \quad (3.3.3)$$

We may view the problem as a constrained minimization problem. That is, we find a mesh with a minimal number of elements, for which the approximation satisfies (3.3.3). The solution of the problem is achieved by following the Principle of Equidistribution [6], which states that the element contributions should be approximately equal. We therefore accept elements that satisfy

$$|-(A\nabla U, \nabla(\phi - \pi_h\phi))_K + (f(U,\lambda), \phi - \pi_h\phi)_K| \leq \frac{\text{TOL}}{M}, \quad (3.3.4)$$

where M is the number of elements in $\mathcal{T}_h$. The difficulty of finding an optimal mesh selection arises due to the fact that (3.3.3) is typically orders of magnitude larger than the error estimate (3.2.9). To limit the number of elements refined at each step, we could use a number of strategies. For example, we could refine elements whose element contribution to the error bound is greater than a number of standard deviations from the mean error contribution. Alternatively, we could refine a fixed fraction of the elements with the greatest element contributions.

Chapter 4

BACKGROUND: GRADIENT METHODS FOR OPTIMIZATION

In our project, we optimize a quantity of interest with respect to parameters in the model by using a gradient search algorithm. In particular, we use a search method that follows the gradient and avoids the computation of the Hessian. In this chapter, we discuss topics in unconstrained optimization that are relevant to our project. In particular, we discuss steepest descent, conjugate gradient, and Quasi-Newton methods [32, 8, 16, 13]. We also discuss issues that arise when we perform a line search.

The basic descent method $x_{k+1} = x_k + \alpha_k d_k$ is defined to generate the sequence $\{x_k\}$, where $\alpha_k = \underset{\alpha \geq 0}{\operatorname{argmin}} f(x_k + \alpha d_k)$, and d_k is some direction of descent. The value α_k is found using a line search strategy. Under proper conditions, the sequence $\{x_k\}$ will converge to a local minimum of the function $f(x)$. For example, in steepest descent, $d_k = -\nabla f(x_k)$, and for the conjugate gradient method, $d_0 = -\nabla q(\lambda_0)$, and $d_k = -\nabla q(\lambda_k) + \beta_k d_{k-1}$ for $k \geq 1$. Higher-order methods such as Newton's method are commonly used but we will not address them here, since we wish to avoid computation of the Hessian $\nabla^2 f(x)$ in this thesis. We discuss the steepest descent and conjugate gradient methods in detail below.

4.1 Steepest Descent

The descent methods utilize the fact that the gradient points in the direction of the maximum increase of f. This simple fact may be easily verified. Recall that if d is any unit vector, the rate of increase of $f(x)$ in the direction d is given by $\nabla f(x) \cdot d$. Hence, by the Cauchy-Schwartz inequality,

$$\nabla f(x) \cdot d \leq \|\nabla f(x)\|\|d\| = \|\nabla f(x)\| = \nabla f(x) \cdot \frac{\nabla f(x)}{\|\nabla f(x)\|}$$

Thus, $\nabla f(x) \cdot d$ is maximized when $d = \frac{\nabla f(x)}{\|\nabla f(x)\|}$.

To derive the method of steepest descent, let x^0 be an arbitrary point in the domain of $f(x)$. By Taylor's theorem,

$$f(x^0 - \alpha \nabla f(x^0)) = f(x^0) - \alpha \|\nabla f(x)\|^2 + \mathbf{o}(\alpha)$$

So for small $\alpha > 0$, $f(x^0 - \alpha \nabla f(x^0)) < f(x^0)$. At each step, we wish to find the best such $\alpha > 0$. The steepest descent algorithm is as follows:

Choose x^0

For $k = 0, 1, ...$

$\alpha_k = \text{argmin}_{\alpha > 0} f(x^k - \alpha \nabla f(x^k))$

$x^{k+1} = x^k - \alpha_k \nabla f(x^k)$

End

In the case of $f(x)$ being quadratic, we may easily calculate α_k at each iteration. Let $f(x) = \frac{1}{2} x^T Q x - b^T x$, where Q is a symmetric, positive definite matrix. Then the extrema of $f(x)$ is found by solving $0 = \nabla f(x) = Qx - b$. For the sake of readability, let $g_k = \nabla f(x_k)$. We seek a solution to

$\alpha_k = \text{argmin}_{\alpha>0}\phi_k(\alpha)$, where $\phi_k(\alpha) = f(x_k - \alpha g_k)$. Applying the first order necessary conditions, $\phi_k'(\alpha) = 0$. That is,

$$\begin{aligned} 0 &= \frac{d}{d\alpha}(f(x_k - \alpha g_k)) \\ &= -g_k^T \nabla f(x_k - \alpha g_k) \\ &= -g_k^T (Q(x_k - \alpha g_k) - b) \\ &= g_k^T b - g_k^T Q x_k + \alpha g_k^T Q g_k. \end{aligned}$$

Therefore, the optimal α is given by

$$\alpha = \frac{g_k^T(-b + Qx_k)}{g_k^T Q g_k} = \frac{g_k^T g_k}{g_k^T Q g_k}. \tag{4.1.1}$$

4.2 The Conjugate Gradient Method

The steepest descent algorithm works well for simple functions, and converges to the exact minimum in one iteration in the case that f is quadratic. On the other hand, steepest descent works poorly for functions such as the Rosenbrock function,

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2. \tag{4.2.1}$$

The function is well known for having the property that steepest descent performs poorly in finding the local minimum $(1, 1)$. For example, in Fig. 4.1 we run 213 iterations to reach a stopping tolerance,

$$\|\nabla f(x_k) - \nabla f(x_{k-1})\| < 10^{-5} \tag{4.2.2}$$

to find the minimum of (4.2.1).

We want to avoid the oscillating behavior of steepest descent. We avoid this by using a relaxation approach. That is, we do not use only

the current gradient to determine the step direction at each iteration, but rather an average of the current gradient with some previous gradients. It turns out that the conjugate gradient method does exactly this. Instead of stepping in the $-\nabla f(x_k)$ direction, we step in a direction that is a linear combination of the current gradient and all previous directions taken. In particular, the direction taken is one which is called Q-conjugate to all previous search directions.

To demonstrate the improved results, in Fig. 4.2 we see that the conjugate gradient method requires 17 iterations to achieve the stopping criterion (4.2.2) for (4.2.1), a big improvement from steepest descent. Another example is shown in Fig. 4.3, where we compare the results of the two search methods for $f(x) = -\sin(5x^2 - \frac{1}{4}y^2 + 3)\cos(2x + 1 - e^y), x_0 = (-.4, .4)^T$. We see that steepest descent requires 34 iterations to achieve (4.2.2) while the conjugate gradient method only requires 6.

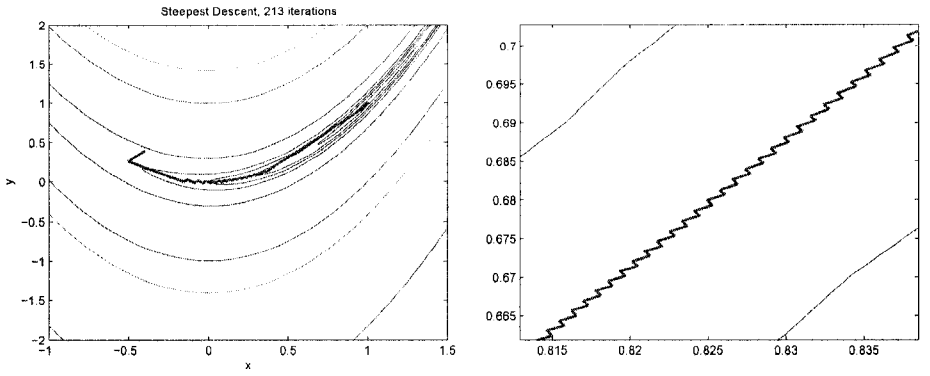

Figure 4.1: Steepest descent applied to the Rosenbrock Function, with a zoomed view on the right. Convergence reached in 213 iterations.

We investigate conjugate direction methods for finding d_k. Again, let $f(x) = \frac{1}{2}x^TQx - x^Tb$, where Q is a symmetric positive definite $n \times n$ matrix, and $x \in R^n$. Let g_k denote the gradient at the k^{th} iteration, $\nabla f(x_k)$.

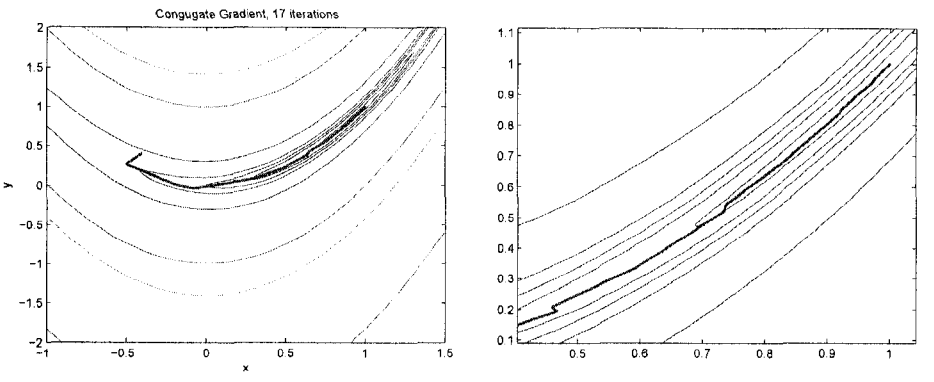

Figure 4.2: Conjugate gradient applied to the Rosenbrock function, with a zoomed view on the right. Convergence reached in 17 iterations.

Definition 4.2.1. *Let Q be a real symmetric matrix. The directions $d_0, d_1, ..., d_m$ are Q-conjugate if $d_i^T Q d_j = 0$ for all $i \neq j$.*

To begin the conjugate gradient algorithm, we choose an initial guess x_0, and set $d_0 = -g_0$. If we have Q-conjugate directions $d_0, d_1, ..., d_n$, then at each iteration we choose $x_{k+1} = x_k + \alpha_k d_k$, where $\alpha_k = \frac{g_k^T d_k}{d_k^T Q d_k}$. We may directly show that α_k is chosen to satisfy $\alpha_k = \text{argmin}_{\alpha \geq 0} \phi_k(\alpha)$, where $\phi_k(\alpha) = f(x_k + \alpha d_k)$. Using the fact that the directions $\{d_i\}_{i=0}^{n-1}$ are linearly independent and Q is positive definite, we obtain the following theorem:

Theorem 4.2.1. *The above algorithm converges in n iterations*

The following lemma gives us that the gradient g_{k+1} is orthogonal to the previous direction d_k.

Lemma 4.2.1. *For each $k = 0, 1, ..., n-1$ in the above algorithm, $g_{k+1}^T d_k = 0$*

The gradient g_k is also orthogonal to all previous directions $d_0, d_1, ...d_{k-1}$, as stated in the following lemma.

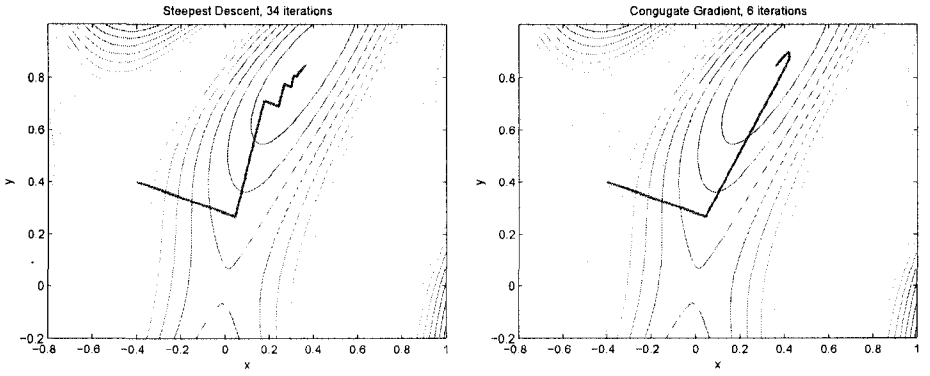

Figure 4.3: Descent methods applied to $f(x) = -\sin(5x^2 - \frac{1}{4}y^2 + 3)\cos(2x + 1 - e^y)$, $x_0 = (-.4, .4)^T$.

Lemma 4.2.2. *For $0 \leq k \leq n-1$, $0 \leq i \leq k$, we have $g_{k+1}^T d_i = 0$*

We are left with the problem of generating Q-conjugate directions $\{d_i\}_{i=0}^n$. This is done by setting d_k to be a linear combination of g_k and d_{k-1}. We use the following proposition to fill in the missing piece of the algorithm.

Proposition 4.2.1. *Let $\beta_k = \frac{g_{k+1}^T Q d_k}{d_k^T Q d_k}$. If $d_0 = -g_0$, and $d_{k+1} = -g_{k+1} + \beta_k d_k$, then the directions $\{d_i\}_{i=0}^{n-1}$ are Q-conjugate.*

Combining (4.1.1) and Proposition 4.2.1, the conjugate gradient algorithm takes on the form:

Choose x^0, set $k = 1$

$g_0 = \nabla f(x_0)$

Set $d_0 = -g_0$

For $k = 0, 1, ...$

$$\alpha_k = \frac{g_k^T d_k}{d_k^T Q d_k}$$

$$x^{k+1} = x^k + \alpha_k d_k$$

$$g_{k+1} = \nabla f(x_{k+1})$$

$$\beta_k = \frac{g_k^T Q d_k}{d_k^T Q d_k}$$

$$d_{k+1} = -g_{k+1} + \beta_k d_k$$

End for

Recall that we do not have the Hessian, Q, which appears in the computation of β_k and α_k. Since $\alpha_k = \text{argmin}_{\alpha \geq 0} f(x_k + \alpha d_k)$, we replace (4.2) with a numerical line search procedure, the details of which will be given later. The other part that requires the Hessian is

$$\beta_k = \frac{g_k^T Q d_k}{d_k^T Q d_k}.$$

We may approximate β_k in several ways, as described in [13]. One method uses the fact that $Qd_k \approx \frac{g_{k+1} - g_k}{\alpha_k}$. To justify this approximation, recall that $x_{k+1} = x_k + \alpha_k d_k$. Pre-multiplying by Q gives $Qx_{k+1} = Qx_k + \alpha_k Q d_k$ and in the quadratic case, $g_k = Qx_k - b$, so that $g_{k+1} = g_k + \alpha_k Q d_k$. But then, $Qd_k = \frac{g_{k+1} - g_k}{\alpha_k}$. Now the formula becomes

$$\beta_k = \frac{g_{k+1}^T (g_{k+1} - g_k)}{d_k^T (g_{k+1} - g_k)} \tag{4.2.3}$$

which is called the Hestenes-Stiefel formula. Other formulas for β_k exist such as the Polak-Ribiere formula,

$$\beta_k = \frac{g_{k+1}^T (g_{k+1} - g_k)}{\|g_k\|}, \tag{4.2.4}$$

and the Fletcher-Reeves formula,

$$\beta_k = \frac{\|g_{k+1}\|}{\|g_k\|}. \tag{4.2.5}$$

4.3 Other Descent Methods

Other methods such as the quasi-Newton method could be used as well. These methods approximate the inverse Hessian without having the Hessian in hand. We outline one type of quasi-Newton method, the BFGS (Broyden, Fletcher, Goldfarb, and Shanno) briefly method below. To implement, suppose we wish to optimize a function $f(x)$, where $x \in \mathbb{R}^n$.

Let H_0 be a real, symmetric, positive matrix.

For $k = 0, 1, 2, ...$

$d_k = -H_k$

$\alpha_k = \min_{\alpha \geq 0} f(x_k + \alpha d_k)$

$x_{k+1} = x_k + \alpha_k d_k$

$\Delta x_k = \alpha_k d_k$

$\Delta g_k = g_k k + 1 - g_k$

$$H_{k+1} = H_k + \left(1 + \frac{\Delta g_k^T H_k \Delta g_k}{\Delta g_k^T \Delta x_k}\right) \frac{\Delta x_k \Delta x_k^T}{\Delta x_k^T \Delta g_k^T} - \frac{H_k \Delta g_k \Delta x_k^T + \left(H_k \Delta g_k \Delta x_k^T\right)^T}{\Delta g_k^T \Delta x_k^T}$$

End for

With an accurate line search, the sequences generated by quasi-Newton methods tends to conjugate directions and the algorithm constructs an approximation to the inverse Hessian matrix. As a result, near a local minimum where the Hessian matrix is positive definite, the method tends to approximate Newton's method, hence it achieves a faster convergence rate.

Another advantage is that quasi-Newton methods are not as sensitive to accuracy in the line search. On the other hand, the conjugate gradient method is faster in large dimensions. To illustrate, let n be the dimension of the variable to optimize, q_T the computational work required to compute the cost function q, and ∇q_T the computational work required to compute the gradient. Then the quasi-Newton method requires roughly $q_T + \nabla q_T + \mathbf{O}(n^2)$ computations per iteration, while the conjugate gradient method requires roughly $q_T + \nabla q_T + \mathbf{O}(n)$ computations per iteration. If $q_T + \nabla q_T << \mathbf{O}(n^2)$, the the conjugate gradient method is preferable, otherwise the quasi-Newton may be preferable due to its previously mentioned advantages.

4.4 Line Search Methods

If we use a descent method without use of the Hessian, we need to apply a line search at some point to minimize $\phi(\alpha) = f(x_k + \alpha d_k)$, where x_k is some iterate and d_k is some direction in the descent method. A simple line search method is the secant method, which is based on Newton's Method. To elaborate further, let $q(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}f''(x_k)(x - x_k)^2$, the second order Taylor series approximation of $f(x)$. Then to optimize $q(x)$, we set $0 = q'(x) = f'(x_k) + f''(x_k)(x - x_k)$, which yields the result

$$x = x_k - \frac{f'(x_k)}{f''(x_k)}.$$

Since we do not wish to compute $f''(x)$, we rely on the secant method and make the substitution

$$f''(x_k) \approx \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}.$$

This yields the line search method:

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f'(x_k) - f'(x_{k-1})} f'(x_k).$$

There are many other commonly used methods, such as polynomial interpolation, described in [16] or [8], for example.

To guarantee that we take a step of decrease, we could naturally impose the condition on α_k that $f(x_k + \alpha_k d_k) < f(x_k)$. In practice, we introduce a stronger requirement on α called the Armijo condition:

$$f(x_k + \alpha d_k) \leq f(x_k) + c_1 \alpha \nabla f(x_k) \cdot d_k. \tag{4.4.1}$$

In other words,

$$\phi(\alpha) \leq \phi(0) + c_1 \alpha \phi'(0).$$

This guarantees a condition on the amount of decrease of f during the line search. Another condition we may impose is a curvature condition,

$$\nabla f(x_k + \alpha_k d_k) \cdot d_k \geq c_2 \nabla f(x_k) \cdot d_k,$$

or

$$\phi'(\alpha_k) \geq c_2 \phi'(0).$$

The strong Wolfe conditions consist of a slight modification of these:

$$\begin{cases} f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k \nabla f(x_k) \cdot d_k, \\ |\nabla f(x_k + \alpha_k d_k) \cdot d_k| \leq c_2 |\nabla f(x_k) \cdot d_k|. \end{cases} \tag{4.4.2}$$

The above conditions guarantee that the descent sequence is decreasing at a sufficient rate. Numerous line search algorithms exist that incorporate these conditions (see [16] for example).

Chapter 5

AN ADAPTIVE ALGORITHM FOR AN ELLIPTIC OPTIMIZATION PROBLEM

5.1 Introduction

In this chapter, we address the problem of optimizing the quantity of interest computed from a solution of an elliptic problem with respect to parameters defining the problem. We let u solve the elliptic boundary value problem,

$$\begin{cases} -\nabla \cdot (A\nabla u) = f(u, x; \lambda), & x \in \Omega, \\ u = 0, & x \in \partial\Omega, \end{cases} \tag{5.1.1}$$

where $\Omega \subset \mathbb{R}^n$ is a convex and polygonal domain, $u : \mathbb{R}^{n+p} \to \mathbb{R}^d$, f is a smooth function, $A(x)$ is a symmetric, positive definite matrix with smooth entries, $y^T A y > a_0 \|y\|^2$ for all y, and $\lambda \in \Lambda \subset \mathbb{R}^p$ are parameters. We assume that the quantity of interest $q(\lambda) = q(u, \lambda) = (u, \psi)_{L^2(\Omega)}$ is a linear functional determined by the data $\psi \in L^2(\Omega)$. For example, if $\psi = 1/\mathrm{vol}(\Omega)$, then $q(\lambda)$ is the average value of the solution u and if $\psi = \delta_x$, then $q(\lambda) = u(x)$. The analysis in this chapter extends to general boundary conditions and to the case $a(x) = a(x, \lambda)$ as well.

In this chapter, we consider the problem of numerically implementing gradient descent methods for searching for an optimal value of $q(\lambda)$. To

define the gradient of the functional $\nabla q(\lambda)$, we use the definition of the Frechét derivative. To obtain a robust search algorithm, we need accurate values for the gradient $\nabla q(\lambda)$ and the quantity of interest $q(\lambda)$. In practice, we use an approximate solution U, thus introducing error in the gradient search. Hence, the accuracy of U is a major concern. In particular, typical searches require several computations of U corresponding to different values of the parameter λ. We may view u as an implicit function of λ and note that as λ varies, then the error in U is also likely to vary. This in turn affects the gradient error. In this chapter we derive an *a posteriori* estimate of the gradient error and devise an efficient adaptive algorithm to control this error.

In [3, 5, 4], a general cost functional $q(u(x), \lambda(x))$ is optimized subject to the constraint $A(u) = f(\lambda)$, where A is a partial differential operator relating a state variable u and a control variable λ. In these papers, stationary points of the associated Lagrangian functional are computed, which correspond with the set of possible local extreme values. The stationary points are found by solving the corresponding Euler-Lagrange system of equations. The mesh adaptation is driven by *a posteriori* error estimates for either the cost functional or an arbitrary quantity of interest. In this chapter, we opt to instead find local extreme values using gradient searches, and we apply *a posteriori* techniques to correct for error in the gradient.

To illustrate, we give an example of a two point boundary value problem in which numerical errors change as the parameter varies,

$$\begin{cases} -u'' = f(u) = -\sin(u) + g(x), & x \in (-1,1), \\ u(-1) = u(1) = 0. \end{cases} \tag{5.1.2}$$

The true solution is $u = \tanh(20e^{\lambda_1(1-\lambda_1)}(x - e^{\lambda_2(1-\lambda_2)-1}))\cos(\pi x/2)(\lambda_1^2 + .1)$, and we take $\psi \equiv 1$, so that $q(\lambda) = \int_{-1}^{1} u\,dx$. In Fig. 5.1, we plot the true

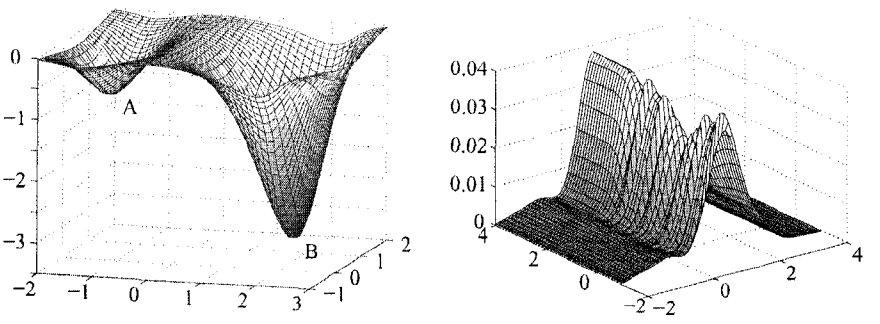

Figure 5.1: On the left is a plot of an approximation of $q(\lambda) = \int_{-1}^{1} u\,dx$ for (5.1.2). There are two extremal values in the parameter domain. On the right is a plot of the L^2 errors of the finite element solutions computed on a uniform mesh versus parameter values. The errors vary significantly.

$q(\lambda)$ as a function of λ, where $\lambda \in \mathbb{R}^2$. We see that there are two local minima, at points A and B. If we compute approximate solutions using one grid for all parameter values, the error varies greatly as the parameter varies, as demonstrated by Fig. 5.1. In addition, the areas in the domain where the error is large vary in parameter space. In Fig. 5.2 we see that the pointwise errors for two different points in the domain vary and are quite distinct from each other.

We also demonstrate that the gradient error can make a significant impact on the progression of a search algorithm. In Fig. 5.3 we plot the errors of the partial derivatives, which vary considerably with respect to the parameters. In Fig. 5.4 we start the search algorithm in the saddle bordering the two local minima. Without error control, the sequence converges to point A, as shown in Fig. 5.4 on the left. On the other hand, if we control the error in the gradient in the way described in this paper, then we obtain a sequence converging to point B, as shown in Fig. 5.4 on the right.

The straightforward "standard" approach is this: We use *a posteriori* analysis employing adjoint operators and computable residuals in order to

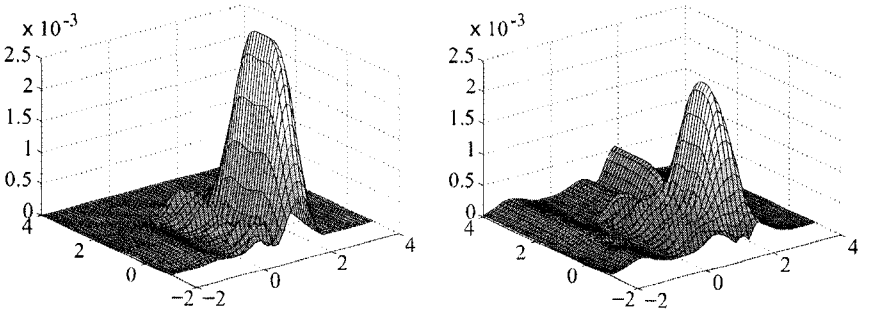

Figure 5.2: Plots of the pointwise error of finite element solutions U computed on uniform meshes as parameters vary. Left: error at $x = 0$. Right: error at $x = 0.5$.

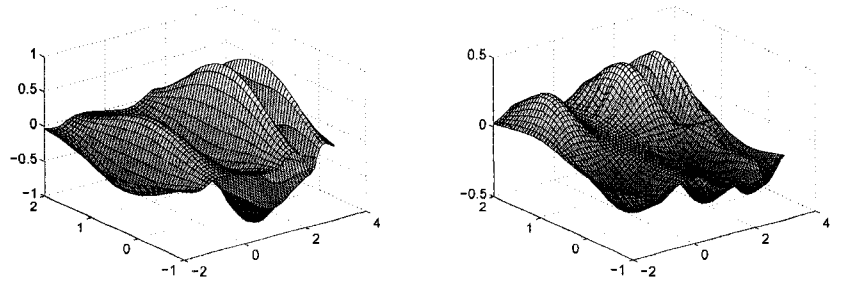

Figure 5.3: Plots of errors in approximated $\partial q/\partial \lambda_1$ (left) and $\partial q/\partial \lambda_2$ (right) computed on uniform meshes as parameters vary.

obtain accurate error estimates in both a quantity of interest and the gradient of the quantity of interest. With an accurate error estimate in hand, it is natural to try to devise an adaptive control algorithm. This is problematic in computational terms. We approximate $U(\lambda_0)$ by computing an adapted mesh that controls error in $q(U(\lambda_0))$ and $\nabla q(U(\lambda_0))$, then compute a new adapted mesh to control the error in $q(U(\lambda_1))$, $\nabla q(U(\lambda_1))$, and so on. Generating new adapted meshes at each step is a very expensive proposition. It is also inefficient given that the parameter λ only changes

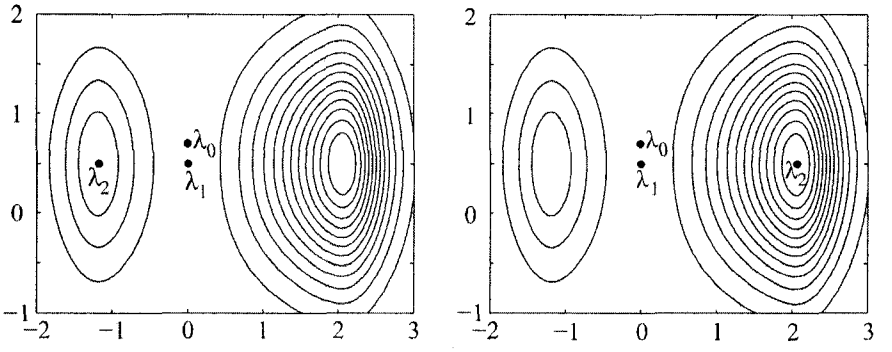

Figure 5.4: Plots of sequences of gradient searches for extrema. Left: computed without error control. Right: computed with error control.

by small increments at each step. Hence, we expect the solution and the numerical error to change by small increments.

We therefore propose a way to make use of an incremental adaptive algorithm in which we refine and unrefine as λ changes. Specifically, we propose a strategy where mesh changes are taken only on those elements in which error contributions take on significant changes. We use the *a posteriori* estimates on the gradient error to control the mesh changes.

5.2 Computing the Gradient

We derive a formula for the gradient $\nabla q(\lambda)$ using duality and adjoint equations. We first consider the change in a quantity of interest corresponding to a change in parameter. Let λ be a parameter value, and u the corresponding solution. We let $\tilde{\lambda}$ denote a perturbation of λ and $\tilde{u}$ the corresponding solution,

$$\begin{cases} -\nabla \cdot (A\nabla\tilde{u}) = f(\tilde{u};\tilde{\lambda}), & x \in \Omega, \\ \tilde{u} = 0, & x \in \partial\Omega. \end{cases} \tag{5.2.1}$$

We define the adjoint by linearizing about $\tilde{u}$. Using (2.3.7), the adjoint operator may be defined as

$$(L'(\tilde{u})\tilde{\phi})^* = -\nabla \cdot (A\nabla\tilde{\phi}) - D_u f(\tilde{u};\tilde{\lambda})\,\tilde{\phi}.$$

Let $\psi \in L^2(\Omega)$ be some smooth function. The adjoint equation is defined as

$$\begin{cases} -\nabla \cdot (A\nabla\tilde{\phi}) - D_u^* f(\tilde{u};\tilde{\lambda})\tilde{\phi} = \psi, & x \in \Omega, \\ \tilde{\phi} = 0, & x \in \partial\Omega. \end{cases} \tag{5.2.2}$$

The relevance of this problem to computing the gradient is given by the following theorem:

Theorem 5.2.1. *Suppose that (5.1.1) satisfies the Lipschitz condition* $\|u - \tilde{u}\|_{L^2(\Omega)} \leq L|\lambda - \tilde{\lambda}|$ *for some* $L > 0$. *Let* $\tilde{\phi}$ *be the solution to the adjoint equation (5.2.2). Then the Frechét derivative of* $q(\lambda)$ *evaluated at* $\tilde{\lambda}$ *may be approximated by*

$$\nabla_\lambda q(\tilde{u};\tilde{\lambda}) \cdot (\lambda - \tilde{\lambda}) \approx (\nabla_\lambda f(\tilde{u};\tilde{\lambda}) \cdot (\lambda - \tilde{\lambda}), \tilde{\phi}), \tag{5.2.3}$$

where $(\cdot,\cdot)$ *denotes the* L^2 *inner product.*

Proof. To compute the Frechét derivative of the quantity of interest with respect to the parameter λ, we need to consider the difference $q(\lambda) - q(\tilde{\lambda})$. Using (5.2.2),

$$\begin{aligned} (u - \tilde{u}, \psi) &= (u - \tilde{u}, -\nabla \cdot (A\nabla\tilde{\phi}) - D_u^* f(\tilde{u};\tilde{\lambda})\phi) \\ &= (u - \tilde{u}, -\nabla \cdot (A\nabla\tilde{\phi})) - (u - \tilde{u}, D_u^* f(\tilde{u};\tilde{\lambda})\tilde{\phi}). \end{aligned}$$

Now integrating by parts twice on the first term and using the definition of the adjoint on the second term gives

$$\begin{aligned} q(\lambda) - q(\tilde{\lambda}) &= (-\nabla \cdot (A\nabla(u - \tilde{u})), \tilde{\phi}) - (D_u f(\tilde{u};\tilde{\lambda})(u - \tilde{u}), \tilde{\phi}) \\ &= (f(u;\lambda) - f(\tilde{u};\tilde{\lambda}), \tilde{\phi}) - (D_u f(\tilde{u};\tilde{\lambda})(u - \tilde{u}), \tilde{\phi}). \end{aligned}$$

Recall the Taylor expansion of $f(u;\lambda)$ centered at $(\tilde{u},\tilde{\lambda})$:

$$f(u;\lambda) = f(\tilde{u};\tilde{\lambda}) + D_u f(\tilde{u};\tilde{\lambda})(u-\tilde{u}) + \nabla_\lambda f(\tilde{u};\tilde{\lambda})\cdot(\lambda-\tilde{\lambda}) + R(u,\tilde{u};\lambda,\tilde{\lambda}),$$

where $R = \mathbf{o}(|\lambda-\tilde{\lambda}|) + \mathbf{o}(\|u-\tilde{u}\|)$. Hence,

$$\begin{aligned} q(\lambda) - q(\tilde{\lambda}) &= (f(u;\lambda) - f(\tilde{u};\tilde{\lambda}),\tilde{\phi}) \\ &\quad - (f(u;\lambda) - f(\tilde{u};\tilde{\lambda}) - \nabla_\lambda f(u;\lambda)\cdot(\lambda-\tilde{\lambda}) - R(u,\tilde{u};\lambda,\tilde{\lambda}),\phi), \end{aligned}$$

or simply

$$q(\lambda) - q(\tilde{\lambda}) = (\nabla_\lambda f(u;\lambda)\cdot(\lambda-\tilde{\lambda}),\phi) + (R(u,\tilde{u};\lambda,\tilde{\lambda}),\phi).$$

This yields (5.2.3) provided that

$$\lim_{|\lambda-\tilde{\lambda}|\to 0} \frac{|(R(u,\tilde{u};\lambda,\tilde{\lambda}),\tilde{\phi})|}{|\lambda-\tilde{\lambda}|} = 0. \tag{5.2.4}$$

To show (5.2.4), we require R to be higher than first order in both $\|u-\tilde{u}\|$ and $|\lambda-\tilde{\lambda}|$. Under reasonable assumptions on f and assuming u and $\tilde{u}$ are contained in a compact region, we can obtain

$$\|R\| \le C\left(\|u-\tilde{u}\|^{1+\epsilon} + |\lambda-\tilde{\lambda}|^{1+\epsilon}\right)$$

for some $\epsilon > 0$, all λ, $\tilde{\lambda} \in \Lambda$. From [10], we know that $\|u-\tilde{u}\| \to 0$ as $|\lambda-\tilde{\lambda}| \to 0$. We assume the problem satisfies the stronger Lipschitz condition $\|u-\tilde{u}\| \le L|\lambda-\tilde{\lambda}|$ for some $L > 0$. With these assumptions, we have

$$\begin{aligned} \frac{\|R\|}{|\lambda-\tilde{\lambda}|} &\le \frac{C\left(\|u-\tilde{u}\|^{1+\epsilon} + |\lambda-\tilde{\lambda}|^{1+\epsilon}\right)}{|\lambda-\tilde{\lambda}|} \\ &\le \frac{CL|\lambda-\tilde{\lambda}|^{1+\epsilon} + C|\lambda-\tilde{\lambda}|^{1+\epsilon}}{|\lambda-\tilde{\lambda}|} \le \tilde{C}|\lambda-\tilde{\lambda}|^{\epsilon} \to 0 \end{aligned}$$

as $|\lambda-\tilde{\lambda}| \to 0$, where $\tilde{C} = C(L^{1+\epsilon}+1)\|\phi\|$. □

5.3 Computing the Gradient Error

We now return to the problem of estimating the error in the gradient $\nabla q(\lambda)$, arising from using a finite element approximate solution. Again, we use the finite element formulation (3.2.5) to solve for the solution of U at parameter value λ. Similarly, we compute the solution to (5.2.1) by finding $\tilde{U} \in V_h$ such that

$$(a\nabla\tilde{U}, \nabla v) = (f(\tilde{U}, \tilde{\lambda}), v) \tag{5.3.1}$$

for all $v \in V_h$. We make use of two dual problems, obtained respectively by linearization about U and $\tilde{U}$. The problems are defined by

$$\begin{cases} -\nabla \cdot (A\nabla\phi) - D_u^* f(U;\lambda)\phi = \psi, & x \in \Omega, \\ \phi = 0, & x \in \partial\Omega, \end{cases} \tag{5.3.2}$$

and

$$\begin{cases} -\nabla \cdot (A\nabla\tilde{\phi}) - D_u^* f(\tilde{U};\tilde{\lambda})\tilde{\phi} = \psi, & x \in \Omega, \\ \tilde{\phi} = 0, & x \in \partial\Omega. \end{cases} \tag{5.3.3}$$

Theorem 5.3.1. *Under the assumptions of Theorem 5.2.1 above,*

$$\nabla_\lambda q(\tilde{\lambda}) \cdot (\lambda - \tilde{\lambda}) = \mathcal{R}(U, \phi) - \mathcal{R}(U, \tilde{\phi}) + (\nabla_\lambda f(\tilde{U}, \tilde{\lambda}) \cdot (\lambda - \tilde{\lambda}), \tilde{\phi}). \tag{5.3.4}$$

Proof. We estimate $q(\lambda) - q(\tilde{\lambda})$ using the decomposition

$$q(\lambda) - q(\tilde{\lambda}) = (u - \tilde{u}, \psi) = (u - U, \psi) + (U - \tilde{U}, \psi) + (\tilde{U} - \tilde{u}, \psi) = T_1 + T_2 + T_3.$$

First, by (3.2.9) we have

$$T_1 = -(A\nabla U, \nabla\phi) + (f(U, \lambda), \phi) + (R_1(u, U), \phi), \tag{5.3.5}$$

for a remainder term R_1. We estimate T_3 analogously, since

$$(\tilde{U} - \tilde{u}, \psi) = -(\tilde{u} - \tilde{U}, \psi).$$

Hence,

$$T_3 = (a\nabla\tilde{U}, \nabla\tilde{\phi}) - (f(\tilde{U}, \tilde{\lambda}), \tilde{\phi}) - (R_3(\tilde{u}, \tilde{U}), \phi), \tag{5.3.6}$$

for remainder term R_3. To compute T_2, we use either (5.3.2) or (5.3.3). Using (5.3.3), we get

$$\begin{aligned} T_2 &= (U - \tilde{U}, -\nabla \cdot (a\nabla\tilde{\phi}) - D_u^* f(\tilde{U}, \tilde{\lambda})\tilde{\phi}) \\ &= (U - \tilde{U}, -\nabla \cdot (a\nabla\tilde{\phi})) - (U - \tilde{U}, D_u^* f(\tilde{U}, \tilde{\lambda})\tilde{\phi}). \end{aligned}$$

Integration by parts on the first term and using the definition of the adjoint on the second term gives

$$T_2 = (a\nabla U, \nabla\tilde{\phi}) - (a\nabla\tilde{U}, \nabla\tilde{\phi}) - (D_u f(\tilde{U}, \tilde{\lambda})(U - \tilde{U}), \tilde{\phi}).$$

Using Taylor's theorem,

$$f(U, \lambda) = f(\tilde{U}, \tilde{\lambda}) + D_u f(\tilde{U}, \tilde{\lambda})(U - \tilde{U}) + D_\lambda f(\tilde{U}, \tilde{\lambda}) \cdot (\lambda - \tilde{\lambda}) + R(U, \tilde{U}, \lambda, \tilde{\lambda}).$$

Hence, T_2 becomes

$$\begin{aligned} T_2 &= (A\nabla U, \nabla\tilde{\phi}) - (A\nabla\tilde{U}, \nabla\tilde{\phi}) \\ &\quad - (f(U, \lambda) - f(\tilde{U}, \tilde{\lambda}) - D_\lambda f(\tilde{U}, \tilde{\lambda}) \cdot (\lambda - \tilde{\lambda}) - R_2(U, \tilde{U}; \lambda, \tilde{\lambda}), \tilde{\phi}), \end{aligned} \tag{5.3.7}$$

for a remainder R_2. Combining (5.3.5), (5.3.6), and (5.3.7) and rearranging gives

$$\begin{aligned} q(\lambda) - q(\tilde{\lambda}) \;=\; & -(A\nabla U, \nabla\phi) + (f(U, \lambda), \phi) + (R_1, \phi) \\ & + (a\nabla\tilde{U}, \nabla\tilde{\phi}) - (f(\tilde{U}, \tilde{\lambda}), \tilde{\phi}) - (R_3, \phi) \\ & + (A\nabla U, \nabla\tilde{\phi}) - (A\nabla\tilde{U}, \nabla\tilde{\phi}) - (f(U, \lambda), \tilde{\phi}) + (f(\tilde{U}, \tilde{\lambda}), \tilde{\phi}) \\ & + (D_\lambda f(\tilde{U}, \tilde{\lambda}) \cdot (\lambda - \tilde{\lambda}), \tilde{\phi}) - (R_2, \tilde{\phi}), \end{aligned}$$

or

$$q(\lambda) - q(\tilde{\lambda}) = -(A\nabla U, \nabla\phi) + (f(U,\lambda),\phi) + (A\nabla U, \nabla\tilde{\phi}) - (f(U,\lambda),\tilde{\phi}) + (D_\lambda f(\tilde{U},\tilde{\lambda})\cdot(\lambda-\tilde{\lambda}),\tilde{\phi}) + R, \quad (5.3.8)$$

where $R = R(U,\tilde{U},\lambda,\tilde{\lambda},\phi,\tilde{\phi})$ is defined by $R \equiv R_1(u,U),\phi)+(R_2(U,\tilde{U};\lambda,\tilde{\lambda}),\tilde{\phi})-(R_3(\tilde{u},\tilde{U}),\tilde{\phi})$. Again, we consider the projection of ϕ onto space V, $\pi_h\phi$. Using Galerkin orthogonality condition (3.2.5),

$$(a\nabla U, \nabla\pi_h\phi) - (f(U,\lambda),\pi_h\phi) = 0,$$

and

$$(a\nabla U, \nabla\pi_h\tilde{\phi}) - (f(U,\lambda),\pi_h\tilde{\phi}) = 0.$$

Substituting these into (5.3.8) gives

$$q(\lambda) - q(\tilde{\lambda}) = -(A\nabla U, \nabla(\phi-\pi_h\phi)) + (f(U,\lambda),\phi-\pi_h\phi) + (A\nabla U, \nabla(\tilde{\phi}-\pi_h\tilde{\phi})) - (f(U,\lambda),\tilde{\phi}-\pi_h\tilde{\phi}) + (D_\lambda f(\tilde{U},\tilde{\lambda})\cdot(\lambda-\tilde{\lambda}),\tilde{\phi}) + R.$$

For simplicity of notation, we denote $\mathcal{R}(u,v) = -(A\nabla v, \nabla(v-\pi_h v)) + (f(u,\lambda), v-\pi_h v)$. Thus,

$$q(\lambda) - q(\tilde{\lambda}) = \mathcal{R}(U,\phi) - \mathcal{R}(U,\tilde{\phi}) + (D_\lambda f(\tilde{U},\tilde{\lambda})\cdot(\lambda-\tilde{\lambda}),\tilde{\phi}) + R.$$

In order to show that

$$\lim_{|\lambda-\tilde{\lambda}|\to 0} \frac{R}{|\lambda-\tilde{\lambda}|} = 0,$$

we assume that R is higher than first order in $\|u-\tilde{u}\|$ and $|\lambda-\tilde{\lambda}|$. We thus prove (5.3.4) by the same argument as in Theorem 5.2.1. □

5.4 Convergence Properties

Our approach applies to a wide variety of gradient-based optimization algorithms. We focus on a particular choice for clarity. Typical optimization methods include steepest descent, conjugate gradient, and Quasi-Newton methods, as discussed in Chapter 4. The basic descent method $\lambda_{k+1} = \lambda_k + \alpha_k d_k$ is defined to generate the sequence $\{\lambda_k\}$, where $\alpha_k = \underset{\alpha \geq 0}{\operatorname{argmin}} \varphi(\alpha) \equiv q(\lambda_k + \alpha d_k)$, and d_k is some direction of descent. For the conjugate gradient method, we have $d_0 = -\nabla q(\lambda_0)$, and $d_k = -\nabla q(\lambda_k) + \beta_k d_{k-1}$ for $k \geq 1$. To compute β_k, we used the Fletcher-Reeves formula (4.2.5), in which $\beta_k = \frac{\|\nabla q(x_k)\|^2}{\|\nabla q(x_{k-1})\|^2}$, although others such as (4.2.3) or (4.2.4) may be used. To compute α_k, we optimize $\varphi(\alpha) \equiv q(\lambda_k + \alpha d_k)$ by performing a line search which uses $\varphi(\alpha)$ along with the first derivative information $\varphi'(\alpha) = \nabla_\lambda q(\lambda_k + \alpha d_k) \cdot d_k$.

The global convergence properties of gradient descent methods with errors are well known. In [9], a result for general descent methods is given, and in [50], a result for the Fletcher-Reeves Conjugate Gradient method is given. For the following, assume that $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function in which ∇f is Lipschitz continuous.

Theorem 5.4.1. *Let the sequence $\{x_k\}$ be generated by the descent method*

$$x_{k+1} = x_k + \alpha_k(d_k + e_k),$$

and suppose that the following estimates hold:

$$c_1\|\nabla f(x_k)\|^2 \leq -\nabla f(x_k)^T d_k, \tag{5.4.1}$$

$$\|d_k\| \leq c_2(1 + \|\nabla f(x_k)\|), \tag{5.4.2}$$

$$\|e_k\| \leq \gamma_k(q + p\|\nabla f(x_k)\|), \tag{5.4.3}$$

$$\sum_{k=0}^{\infty} \gamma_k = \infty, \tag{5.4.4}$$

$$\sum_{k=1}^{\infty} \gamma_k^2 < \infty, \tag{5.4.5}$$

where c_1, c_2, p and q are scalars. Then either $f(x_k) \to -\infty$ or $f(x_k) \to L < \infty$ and $\nabla f(x_k) \to 0$. Also, whenever $x_k \to x^*$, $\nabla f(x^*) = 0$.

Note that in the case of steepest descent, $d_k = \nabla f(x_k)$ and so (5.4.1) and (5.4.2) are trivially satisfied. However, this suggests that it is practical to check that the gradient error $R(U, \phi) - R(U, \tilde{\phi})$ does not exceed

$$C \cdot \frac{1}{k}\left(1 + \|\nabla f(x_k)\|\right), \tag{5.4.6}$$

for some constant C, in order to satisfy (5.4.3), (5.4.4), and (5.4.5).

Theorem 5.4.2. *(The Fletcher-Reeves Conjugate Gradient Method) Let $x_{k+1} = x_k + \alpha_k(s_k + e_k)$, where α_k is the step-size determined from an inexact Wolfe or Armijo line search, and the direction s_k is determined by*

$$s_k = \begin{cases} -\nabla f(x_k), & k = 1, \\ -\nabla f(x_k) + \beta_k d_{k-1}, & k \geq 2, \end{cases}$$

and $d_k = s_k + w_k$, where $\beta_k = \frac{\|\nabla f(x_k)\|^2}{\|\nabla f(x_{k-1})\|^2}$. We assume that

$$c_1\|\nabla f(x_k)\|^2 \leq -\nabla f(x_k)^T s_k \tag{5.4.7}$$

$$\|e_k\| \leq \gamma_k(q + p\|\nabla f(x_k)\|) \tag{5.4.8}$$

$$\gamma_k = O\left(\frac{1}{k}\right), \tag{5.4.9}$$

where $c_1, p, q, \gamma_k > 0$. *If* ∇f *is Lipschitz continuous, then either* $f(x_k) \to -\infty$ *or* $\liminf_{k\to\infty} \|\nabla f(x_k)\| = 0$.

In this theorem, (5.4.8) and (5.4.9) are equivalent to conditions (5.4.3), (5.4.4), and (5.4.5), so we may still use (5.4.6) to define the tolerance.

5.5 An Adaptive Algorithm

We now devise an efficient adaptive algorithm that insures an accurate approximate gradient during the optimization search. A straightforward approach would be to use the standard adaptive error approach, described in section 3.3, at each iteration in the gradient search. This requires the generation of a new mesh at each step in the search algorithm, which is needlessly expensive. Assuming the solution depends continuously on the parameter, small changes in the parameter lead to small changes in the error indicators. The alternative is to seek changes in the error indicators as the parameters change, and refine and unrefine the mesh as needed.

On the first step, we create an adapted mesh that produces an accurate value of $q(U)$. Specifically, we refine the mesh using (3.2.9), $(u - U, \psi) \approx -(A\nabla U, \nabla(\phi - \pi_h\phi)) + (f(U,\lambda), \phi - \pi_h\phi) + (R_1, \phi)$. To control the error, we write

$$|(u - U, \psi)| \leq \sum_k |-(A\nabla U, \nabla(\phi - \pi_h\phi))_k + (f(U,\lambda), \phi - \pi_h\phi)_k| \,. \quad (5.5.1)$$

Denoting $\mathcal{R}(U,\phi)_k = (A\nabla U, \nabla(\phi - \pi_h\phi))_k - (f(U,\lambda), \phi - \pi_h\phi)_k$, the equation (5.5.1) becomes $|(u - U, \psi)| \leq \sum_k |\mathcal{R}(U,\phi)_k|$. We use the standard optimization based adaptive mesh control as described in section 3.2. For example, if $|\mathcal{R}(U,\phi)_k| > \text{TOL}$, we could refine element k.

Next, we control the error in $\nabla_\lambda q(\lambda)$. By (5.3.4),

$$\nabla_\lambda q(\tilde{u}, \tilde{\lambda}) \cdot (\lambda - \tilde{\lambda}) = \mathcal{R}(U, \phi) - \mathcal{R}(U, \tilde{\phi}) + (\nabla_\lambda f(\tilde{U}, \tilde{\lambda}) \cdot (\lambda - \tilde{\lambda}), \tilde{\phi}).$$

In practice, we compute the gradient at $\tilde{U}$, in the direction $\lambda - \tilde{\lambda}$,

$$\nabla_\lambda q(\tilde{U}, \tilde{\lambda}) \cdot (\lambda - \tilde{\lambda}) \approx (\nabla_\lambda f(\tilde{U}, \tilde{\lambda}) \cdot (\lambda - \tilde{\lambda}), \tilde{\phi}). \tag{5.5.2}$$

The terms $\mathcal{R}(U, \phi) - \mathcal{R}(U, \tilde{\phi})$ indicate the difference in the approximated and true gradients, and may be found after computing the approximated gradient. To summarize, we first approximate $\tilde{U}$ at $\tilde{\lambda}$, the adjoint $\tilde{\phi}$ at $\tilde{\lambda}$, then use this information to compute the approximated gradient $(\nabla_\lambda f(\tilde{U}, \tilde{\lambda}) \cdot (\lambda - \tilde{\lambda}), \tilde{\phi})$. Next we compute the solution U and the adjoint ϕ at λ, then compute the gradient error $\mathcal{R}(U, \phi) - \mathcal{R}(U, \tilde{\phi})$. If the gradient error is high, we can refine the mesh to reduce this error. We can decompose further by noting that

$$\begin{aligned} |\mathcal{R}(U, \phi) - \mathcal{R}(U, \tilde{\phi})| &= |\mathcal{R}(U, \phi) - \mathcal{R}(\tilde{U}, \tilde{\phi}) + \mathcal{R}(\tilde{U}, \tilde{\phi}) - \mathcal{R}(U, \tilde{\phi})| \\ &\leq |\mathcal{R}(U, \phi) - \mathcal{R}(\tilde{U}, \tilde{\phi})| + |\mathcal{R}(\tilde{U}, \tilde{\phi}) - \mathcal{R}(U, \tilde{\phi})|. \end{aligned} \tag{5.5.3}$$

Note that the first term $|\mathcal{R}(U, \phi) - \mathcal{R}(\tilde{U}, \tilde{\phi})|$ gives the change in error from one iteration to the next. So we do not need a completely new mesh each time we solve the PDE, but may use (5.5.3) to refine and unrefine the mesh. That is, we start with a good mesh by using (5.5.1) then as λ changes, we find elements in which the error estimate has changed. We refine element K if either $|\mathcal{R}(U, \phi)_K - \mathcal{R}(\tilde{U}, \tilde{\phi})_K|$ or $|\mathcal{R}(\tilde{U}, \tilde{\phi})_K - \mathcal{R}(U, \tilde{\phi})_K|$ is large. If refinement is to take place, we then recompute the solution $\tilde{U}$, adjoint $\tilde{\phi}$ at $\tilde{\lambda}$, and the new approximated gradient $(\nabla_\lambda f(\tilde{U}, \tilde{\lambda}) \cdot (\lambda - \tilde{\lambda}), \tilde{\phi})$ on this new mesh, and repeat this process.

We incorporate a tolerance which takes into account Theorems 5.4.1 and 5.4.2, so set TOL $= \frac{C}{k}(1 + \|\nabla q(\lambda_k)\|)$ at each step k in the descent method. Also, we avoid over-refining or under-refining, by selecting a certain percentage of elements which have the largest or smallest error contributions. For our test problems, we refined elements in which the error contribution is greater than either TOL$/N$ or $\mu + c\sigma^2$, where N is the number of elements, μ and σ^2 are the mean and standard deviation of the contributions respectively, and c is some constant.

Recall that the line search step in the descent method requires solving $\nabla q(\lambda + \alpha d) \cdot d$ for various values of α. Here, we take $\lambda = \tilde{\lambda} + d$ in (5.3.4) to obtain

$$\nabla_\lambda q(\tilde{\lambda}) \cdot d = \mathcal{R}(U, \phi) - \mathcal{R}(U, \tilde{\phi}) + (D_\lambda f(\tilde{U}, \tilde{\lambda}) \cdot d, \tilde{\phi}), \tag{5.5.4}$$

where U and ϕ are computed at $\lambda = \tilde{\lambda} + d$. Note that from the reference value $\tilde{\lambda}$, we could also take a smaller step in the direction d, so that $\lambda = \tilde{\lambda} + \epsilon d$ for some $\epsilon > 0$. Then $\lambda - \tilde{\lambda} = d\epsilon$, so that (5.3.4) becomes

$$\nabla_\lambda q(\tilde{\lambda}) \cdot (d\epsilon) = \mathcal{R}(U, \phi) - \mathcal{R}(U, \tilde{\phi}) + (D_\lambda f(\tilde{U}, \tilde{\lambda}) \cdot (d\epsilon), \tilde{\phi}),$$

or

$$\nabla_\lambda q(\tilde{\lambda}) \cdot d = \frac{\mathcal{R}(U, \phi) - \mathcal{R}(U, \tilde{\phi})}{\epsilon} + (D_\lambda f(\tilde{U}, \tilde{\lambda}) \cdot d, \tilde{\phi}),$$

where U and ϕ are computed at $\lambda = \tilde{\lambda} + \epsilon d$.

Unlike the line search, the descent method requires the full gradient $\nabla q(\tilde{\lambda})$. We simply apply (5.5.2) using $\lambda - \tilde{\lambda} = e_i$ for $i = 1, ..., p$. In this case, the formula becomes

$$\nabla_\lambda q(\tilde{\lambda}) \cdot e_i = (\nabla_\lambda f(\tilde{U}, \tilde{\lambda}) \cdot e_i, \phi).$$

This allows us to reconstruct the gradient, since $\nabla_\lambda q(\tilde{\lambda}) \cdot e_i = \frac{\partial}{\partial \lambda_i} q(\tilde{\lambda})$.

5.6 Pseudo-code

To solve the problem (5.1.1), we propose the following adaptive algorithm:

Choose initial guess $\lambda_1 \in \mathbb{R}^p$

$g_1 = [\text{EstimateGradient}(\lambda_1, e_1), ...\text{EstimateGradient}(\lambda_1, e_p)]$

$d_1 = -g_1$

For $k = 1$ to n $(n \geq p)$

$\alpha_k = \text{LineSearch}(\lambda_k, d_k)$

$\lambda_{k+1} = \lambda_k + \alpha_k d_k$

$g_{k+1} = [\text{EstimateGradient}(\lambda_{k+1}, e_1), ...\text{EstimateGradient}(\lambda_{k+1}, e_p)]$

$\beta_k = \frac{g_{k+1}^T g_{k+1}}{g_i^T g_i}$ (Fletcher-Reeves formula, others may be used)

$d_{k+1} = -g_{k+1} + \beta_k d_k$

End for

Function LineSearch(λ, d) (This function performs an inexact line search that optimizes $\varphi(\alpha) = q(\lambda + \alpha d)$ using the derivative $\varphi'(\alpha) = \nabla q(\lambda + \alpha d) \cdot d$).

Set constants $c_1 = 10^{-4}, c_2 = 0.1, \text{LINETOL} = 10^{-4}$.

Choose initial guesses $\alpha_0 = 0$, $\alpha_1 = 1$.

Call EstimateGradient(λ, d) to get $\varphi(\alpha_0)$ and $\varphi'(\alpha_0)$.

Set $j = 1$.

Loop

Call EstimateGradient($\lambda + \alpha_j d, d$) to get $\varphi(\alpha_j)$ and $\varphi'(\alpha_j)$

If $\varphi(\alpha_j) > \varphi(\alpha_0) + c_1\alpha_j\varphi'(\alpha_0)$ (check for sufficient decrease), set $\alpha_{j+1} < \alpha_j$.

Else, if $\varphi'(\alpha_j) < c_2\varphi'(\alpha_0)$ (curvature condition), set $\alpha_{j+1} > \alpha_j$.

Else, exit loop and increment j.

If $\varphi(\alpha_j) \approx \varphi(\alpha_0)$ then d is not a direction of descent, return 0.

End loop

Loop until $|\alpha_j - \alpha_{j-1}| <$ LINETOL

Call EstimateGradient($\lambda + \alpha_j d, d$) to obtain $\varphi'(\alpha_j)$.

Set $\alpha_{j+1} = \alpha_j - \varphi'(\alpha_j) \cdot \frac{\alpha_j - \alpha_{j-1}}{\varphi'(\alpha_j) - \varphi'(\alpha_{j-1})}$ (secant method, see [13])

Increment j

End loop

Return α_j

End function

Function EstimateGradient($\tilde{\lambda}, d$) (This function estimates $\nabla q(\tilde{\lambda}) \cdot d$).

Solve for $\tilde{U}$ and $\tilde{\phi}$ at $\tilde{\lambda}$.

Set $\lambda = \tilde{\lambda} + \epsilon d$.

Loop until error $\mathcal{R}(U, \phi) - \mathcal{R}(U, \tilde{\phi})$ is controlled.

Solve for U and ϕ at λ

Refine and/or un-refine using criteria $|\mathcal{R}(U,\phi) - \mathcal{R}(\tilde{U},\tilde{\phi})|$ and $|\mathcal{R}(\tilde{U},\tilde{\phi}) - \mathcal{R}(U,\tilde{\phi})|$.

End loop

Return $\nabla q(\tilde{\lambda}) \cdot d = \int_\Omega \nabla_\lambda f(\tilde{U},\tilde{\lambda}) \cdot d\,\tilde{\phi}\,dx$

End function

5.7 Numerical Results

We write code for performing the gradient search in MATLAB. We also make use of the finite element package ACES (Adaptive Coupled Equation Solver) developed by Tim Wildey. We use the package for solving the forward and the adjoint problems, computing the error indicators, and refining and unrefining the mesh.

5.7.1 A Semilinear Example in One Dimension

Consider

$$\begin{cases} -u'' = f(u) = -u^2 + g(x), & x \in (-1,1), \\ u(-1) = u(1) = 0, \end{cases}$$

where $g(x)$ is chosen so that the solution is

$$u = \tanh\left(20e^{\lambda_1(1-\lambda_1)}\left(x - e^{\lambda_2(1-\lambda_2)-1}\right)\right)\cos\left(\frac{\pi x}{2}\right).$$

The quantity of interest used is $\psi(x) = 1$, so $q(\lambda) = \int_{-1}^{1} u\,dx$. This example has the property that the solution changes as the parameter λ makes small changes. For example, in Fig. 5.5 on the left, we see that the solution is very steep at $x \approx 0.5$. The gradient error terms $\mathcal{R}(U,\phi) - \mathcal{R}(U,\tilde{\phi})$ are naturally higher here, so these elements are marked for refinement. The Fig.

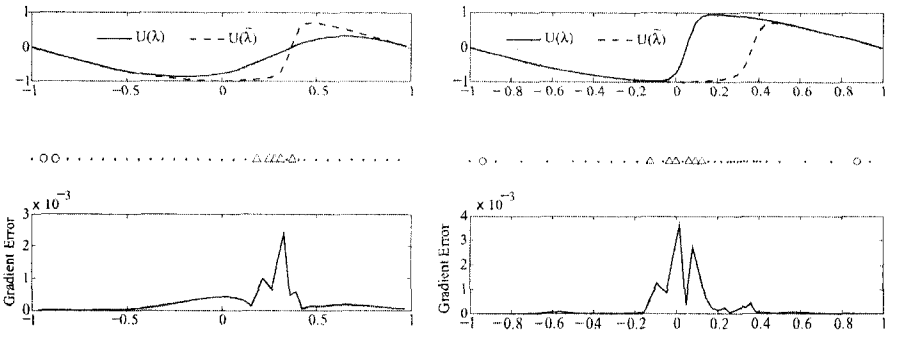

Figure 5.5: Left: Top panel shows $\tilde{U}(\tilde{\lambda})$ and $U(\lambda)$ at $\tilde{\lambda} = (1,1)^T$, $d = (1,0)^T$. Middle panel shows mesh, where triangles denote elements marked for refinement and circles denote elements marked for unrefinement. Bottom panel shows error in the gradient at $\tilde{\lambda}$. Right: Plots at $\tilde{\lambda} = (1,1)^T$, $d = (0,1)^T$.

shows the initial mesh, and elements marked for refinement are shown with triangles. As the parameter changes, we obtain a very different solution, which is shown in Fig. 5.5 on the right. Here the solution is steep at $x \approx 0$, and very smooth at $x \approx 0.5$. We see that the elements near 0 are marked for refinement and elements near 0.5 are marked for un-refinement, which is indicated by circles. The mesh up to convergence is shown in Fig. 5.6, where again triangles indicate that the element is marked for refinement and circles indicate that the element is marked for un-refinement. In Table 9.3.1 we provide an example in which controlling the error improves the approximation of the gradient.

	$\nabla q(\tilde{\lambda})$
No error control	$(-0.00198, 0.603)^T$
After error control	$(0.00275, 0.613)^T$
True Gradient	$(0.00352, 0.615)^T$

Table 5.1: Computed gradient at $\tilde{\lambda} = (1,1)^T$, before and after error control.

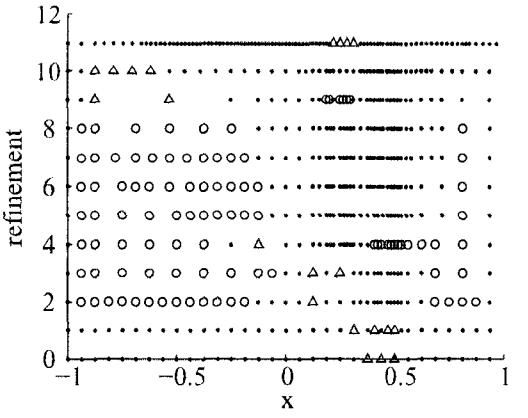

Figure 5.6: Mesh gridpoints at each level of refinement. Triangles indicate elements to refine, circles indicate elements to be unrefined.

5.7.2 A Wound Healing Model in Two Dimensions

To give an example in two dimensions, consider the stationary state of the wound healing model as described in [37],

$$\begin{cases} q_1 \Delta u_1 = f_1(u_1, u_2; \lambda), \\ q_2 \Delta u_2 = f_2(u_1, u_2; \lambda), \end{cases} \tag{5.7.1}$$

in the wound domain $\Omega \subset \mathbb{R}^2$, where $u_1 = u_2 = 1$ on the wound boundary. In this model, we have two conservative equations, one for the epithelial cell density per unit area, u_1, and one for the concentration, u_2, of the mitosis-regulating chemical. Here we assume that the mitosis-regulating chemical controls the rate of proliferation of new cells in the wound domain. The nonlinearity f_1 describes the difference between the mitotic generation and the natural cell loss rate, and f_2 represents the difference in the production of the chemical by cells and the decay of the active chemical . The parameter $\lambda \in \mathbb{R}^3$ describes the time decay rate of the chemical, the maximum rate of chemical production, and the maximum level of chemical activation of mitosis. We optimize the quantity of interest $q(\lambda) = (u(\lambda), \psi)$.

We use the q_1, q_2, f_1, f_2 described in [37] to obtain

$$\begin{cases} -0.035\Delta u_1 = \frac{2\mu_2(\mu_3-\beta(\mu))u_2}{\mu_2^2+u_2^2}u_1(2-u_1) - u_1, \\ -0.31\Delta u_2 = \alpha(\mu)\left(\frac{u_1(1+\mu_1^2)}{u_1^2+\mu_1^2} - u_2\right), & x \in \Omega, \\ \beta(\mu) = \frac{1+\mu_2^2-2\mu_2\mu_3}{(1-\mu_2)^2}, \\ \alpha(\mu) = \frac{1}{12}\log(2\mu_3). \end{cases} \tag{5.7.2}$$

Here, $\mu = (\mu_1, \mu_2, \mu_3)^T$ are parameters. To obtain useful numerical results, we artificially set

$$\begin{cases} \mu_1 = 0.99e^{1-4\lambda_1(1-\lambda_1)}, \\ \mu_2 = 3e^{1-4\lambda_2(1-\lambda_2)}, \\ \mu_3 = 1.01e^{4\lambda_3(1-\lambda_3)}. \end{cases}$$

This results in analyzing parameter values of μ near $(0.99, 3, 1.01)^T$ and small changes near these values as a result of the optimization search. The new parameter λ has the property that the quantity of interest $q(\lambda) = \int_\Omega (u_1 + u_2)\, dx$ has a local extrema at $\lambda = (0.5, 0.5, 0.5)^T$. Fig. 5.7 depicts the change in mesh during the stepping algorithm, as λ varies from $(1, 1, 1)$ to $(0.5, 0.5, 0.5)$.

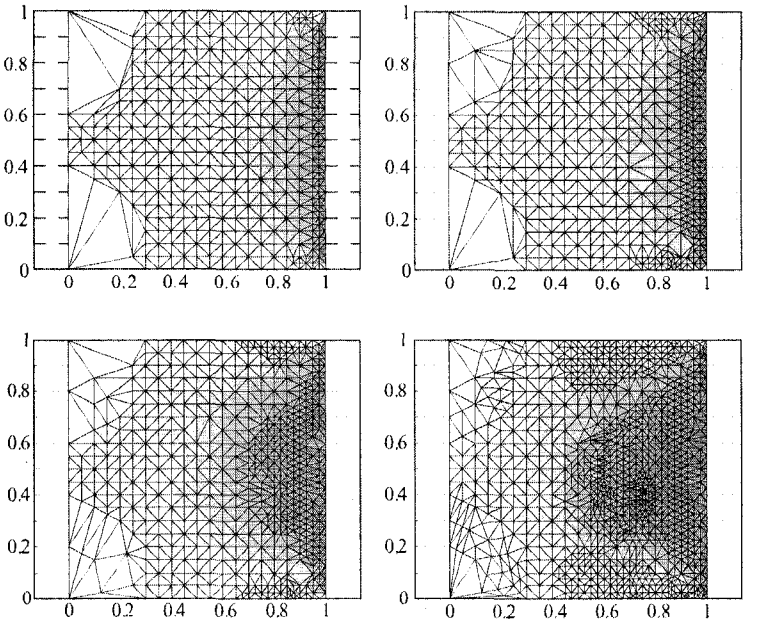

Figure 5.7: Plots of $U_1 + U_2$ and the evolving mesh, at four values of λ in the optimization algorithm.

Part II

Stochastic-Deterministic Coupling: A Mathematical Framework

Chapter 6

BACKGROUND: PROBABILITY

6.1 Limit Theorems

In this section, we introduce some of the statistical tools used in Monte Carlo techniques. Monte Carlo techniques involve repetitively simulating a random process to obtain information about the process. Each time we simulate the process, we obtain what we call a realization of the process. Using the realizations, we may compute sample statistics such as mean and variance. In this section, we examine the tools used to understand the relationship between the number of realizations and the accuracy of the statistical information obtained.

If X is a random variable with probability density function f, denote $E[X] = \int x f(x)\, dx$ to be the mean, or expectation of X.

Proposition 6.1.1. *(Markov Inequality) Let X be a non-negative random variable. If $E[X]$ exists,*

$$P(|X| > a) \leq a^{-1} E(|X|) \text{ for any } a > 0.$$

Applying the Markov inequality to $(X - E[X])^2$, we obtain the following characterization of variance.

Proposition 6.1.2. *If a random variable X has finite variance, then for any $a > 0$,*

$$P((X - E[X])^2 \geq a) \leq a^{-2} var(X).$$

The Central Limit Theorem states that the sum of a large number of independent random variables taken from the same distribution behaves like a single normal random variable. To analyze large sets of random variables, we need the Central Limit Theorem, as well as the Law of Large Numbers. We first review some basic modes of convergence.

Definition 6.1.1. *Let X_n be a sequence of random variables, and let X be a random variable.*

- *X_n converges to X almost surely (a.s.), if*

$$P\left(\lim_{n\to\infty} X_n = X\right) = 1.$$

- *X_n converges to X in probability if*

$$\lim_{n\to\infty} P(|X_n - X| \geq \epsilon) = 0,$$

  for every $\epsilon > 0$.

- *X_n converges to X in L^2, or in mean square, if*

$$\lim_{n\to\infty} E[|X_n - X|^2] = 0.$$

If X_n converges to X almost surely, this means that the set of values ω, such that $X_n(\omega)$ does not converge to $X(\omega)$, is a set of measure zero. Convergence in probability is a weaker mode of convergence, in fact, both almost sure and L^2 convergence imply convergence in probability.

Theorem 6.1.1. *(Central Limit Theorem) Let* $\{X_i\}_{i=1}^{\infty}$ *be independent, identically distributed random variables with mean* μ *and variance* $\sigma^2, 0 < \sigma^2 < \infty$. *Then for any* $\alpha < \beta$,

$$\lim_{M\to\infty} P\left(\alpha \leq \frac{\sum_{i=1}^{M} X_i - M\mu}{\sqrt{M}\sigma} \leq \beta\right) = \frac{1}{\sqrt{2\pi}} \int_{\alpha}^{\beta} e^{-\frac{1}{2}x^2}\, dx.$$

In experiments, we would like to understand how well the average $\frac{1}{M}\sum_{i=1}^{M} X_i$ approximates the mean of X_i. After rearranging terms and dividing by $\sqrt{M}$, the Central Limit Theorem tells us that for large M,

$$P\left(\frac{\alpha\sigma}{\sqrt{M}} \leq \frac{1}{M}\sum_{i=1}^{M} X_i - \mu \leq \frac{\beta\sigma}{\sqrt{M}}\right) \approx \frac{1}{\sqrt{2\pi}} \int_{\alpha}^{\beta} e^{-\frac{1}{2}x^2}\, dx. \qquad (6.1.1)$$

For example, for large M, if we choose $\alpha = -M^{1/4}, \beta = M^{1/4}$, we see that $M^{-1}\sum_{i=1}^{M} X_i$ is close to μ with high probability. We get an error bound by noticing that (6.1.1) becomes

$$P\left(-\frac{\sigma}{M^{1/4}} \leq \frac{1}{M}\sum_{i=1}^{M} X_i - \mu \leq \frac{\sigma}{M^{1/4}}\right) \approx \frac{1}{\sqrt{2\pi}} \int_{-M^{1/4}}^{M^{1/4}} e^{-\frac{1}{2}x^2}\, dx.$$

Theorem 6.1.2. *(Strong Law of Large Numbers) Let* $\{X_i\}_{i=1}^{\infty}$ *be independent, identically distributed random variables,* $\mu = E[X_i], \sigma^2 = var(X_i) < \infty$. *Then,*

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} X_i = \mu \text{ a.s. and in } L^2.$$

This also implies the convergence is in probability, which is called the Weak Law of Large Numbers.

6.2 Sampling Techniques

The goal in this section is to generate random variables with a known distribution F. Assume we have a pseudo-random number generator that draws numbers from the uniform distribution on $[0, 1]$. We denote $U([a, b])$

to be the uniform distribution on $[a, b]$. The corresponding density function is $f_U = 1/(b-a)$. We note that samples from $U([a, b])$ are easily generated by taking $U_1(b-a)+a$, where U_1 is a random number from the interval $[0, 1]$.

We discuss some of the basic methods of generating random variables from an arbitrary distribution. We denote $X \sim f$ to mean that a random variable X is taken from a distribution with density function f. One way to generate random variables from a distribution with density function f is to find a deterministic function $h : [0, 1] \to \mathbb{R}$ such that $h(U) \sim f$, where U is drawn from a uniform distribution on $[0, 1]$. For example, suppose we wish to draw random variables from the exponential distribution. The exponential distribution has density function

$$f(x) = \begin{cases} 7e^{-7x}, & x \geq 0, \\ 0, & x < 0, \end{cases}$$

and cumulative density function $F(x) = 1 - e^{-7x}, x \geq 0$. We define $h(u) = -\frac{1}{7}\log u, 0 < u \leq 1$. Then it can be shown that if U is taken from the uniform distribution on $[0, 1]$, then $h(U)$ has the exponential distribution with density function f.

We computed F^{-1} to devise the h, which uses the fact that F is monotone. In general, if F is right-continuous and non-decreasing, we define

$$F^{-1} = \inf_{z \in \mathbb{R}} F(z) \geq u \text{ for } u \in [0, 1].$$

Theorem 6.2.1. *Let F be the cumulative density function of a random variable and U be a random variable taken from the uniform distribution on $[0, 1]$. Then $F^{-1}(U) \sim F$ and $F^{-1}(1-U) \sim F$.*

Example 6.2.1. *Suppose we want to generate an exponential random variable X with rate λ. The density function in this case is $f(x) = \lambda e^{-\lambda x}$, and the cumulative density function is $F(x) = 1 - e^{-\lambda x}$. Thus, we can generate a random variable by inverting F, and have $X = -\frac{1}{\lambda}\log(U)$, where $U \sim U(0,1)$.*

The second approach is a Monte Carlo approach called the rejection method, or accept-rejection sampling. This method avoids the need for inverting the function F. The goal is to generate a random variable X taken from a distribution with density function $f(x)$. Suppose that f is bounded and zero outside of $[a,b]$. Set $c = \sup_{x\in[a,b]} f(x)$. To perform the basic rejection method, we run the following to generate $X \sim f$:

1. Generate $Q \sim U[a,b]$.
2. Generate $Y \sim U[0,c]$.
3. If $Y \leq f(Q)$, then set $X := Q$. Otherwise, return to step 1.

To summarize, we generate a random vector $(Q,Y) \in [a,b] \times [0,c]$. We check that the random vector is in the region below the graph of f. This method generalizes to random vectors in higher dimensions as well.

We describe a more general approach which may be used to decrease the number of rejected samples, and therefore speed up the algorithm. Suppose there is a function $g(x)$, called the proposal density function, and scalar $M > 1$ such that $f(x) < Mg(x)$. In addition, suppose we can easily sample $g(x)$. Then instead of sampling from f, we sample from $Mg(x)$, where sampling is easier. For example there are many advanced algorithms, often included in software packages, that efficiently generate random numbers from

common distributions such as the normal, the gamma, or the exponential distributions. To generate $X \sim f$, we follow the following algorithm.

1. Generate $Q \sim g(x)$.

2. Generate $Y \sim U[0,1]$.

3. If $Y \leq f/(Mg)$, then set $X := Q$. Otherwise return to step 1.

6.3 Kernel Density Estimation

A kernel density estimator $\hat{f}$ is used to approximate a density function f, given a set of samples $\{x_1, ..., x_n\}$ from $f(x)$. The density histogram is the most basic of kernel density estimators. To obtain a histogram, we need a sample $\{x_1, ..., x_n\}$ from $f(x)$, and a mesh $\{t_k, k \in \mathbb{R}\}$. If, in addition, we require that $t_{k+1} - t_k = h$ for all k, then the histogram is said to have bin width h. The common frequency histogram is built using blocks of height 1 and width h and integrates to nh. The density histogram uses blocks of height $1/(nh)$, so that the histogram integrates to 1, and is defined as

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} I_{[t_k, t_{k+1})}(x_i), x \in [t_k, t_{k+1}),$$

where

$$I_{[t_k, t_{k+1})}(x_i) = \begin{cases} 0, & x_i \notin [t_k, t_{k+1}), \\ 1, & x_i \in [t_k, t_{k+1}). \end{cases}$$

Definition 6.3.1. *We say a kernel is a function $K : \mathbb{R} \to \mathbb{R}$ for which*

$$\int_{\mathbb{R}} K(x)dx = 1.$$

An example of such a kernel is $K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$, the density function for a Gaussian random variable with zero mean and unit variance. Another example is the delta function, $K(x) = \delta(x)$.

Let K be some kernel function, and $x_1, x_2, ..., x_n \in \mathbb{R}$ be data samples. Then the basic kernel estimator is

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right).$$

Here h is the smoothing parameter, also called the window width, or bandwidth. The estimate $\hat{f}$ approximates the density function f by using the samples $x_1, ..., x_n$. This approximate density function inherits the smoothness properties of the kernel K. Note that $\hat{f}(x)$ integrates to 1, since

$$\int_{\mathbb{R}} \hat{f}(x)dx = \frac{1}{nh} \sum_{i=1}^{n} \int_{\mathbb{R}} K\left(\frac{x - x_i}{h}\right) dx = \frac{1}{nh} h \sum_{i=1}^{n} \int_{\mathbb{R}} K(u)du = \frac{1}{n} \sum_{i=1}^{n} 1 = 1.$$

For example if we use the Gaussian kernel $K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$, the corresponding density estimate becomes

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) = \frac{1}{nh\sqrt{2\pi}} \sum_{i=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{x - x_i}{h}\right)^2\right).$$

This density estimate is a sum of Gaussians centered around each data point, each with standard deviation h. Note that if h is too small we will under-smooth the data and if it is too large we will over-smooth the data. In many cases, correcting for over-smoothing results in noise in the tail areas. Thus, it may be necessary to use a larger h near the tails. In Fig. 6.1, we take 222 samples of the waiting times for a particular geyser to strike. We apply a kernel density estimate for varying values of h. We see that if h is too small, we get unnecessary peaks around data points. By using a favorable value of h, we see that the data is clearly bimodal. If h is too large, the density estimate does not reflect the fact that the data is bimodal, and we have over-smoothed the data.

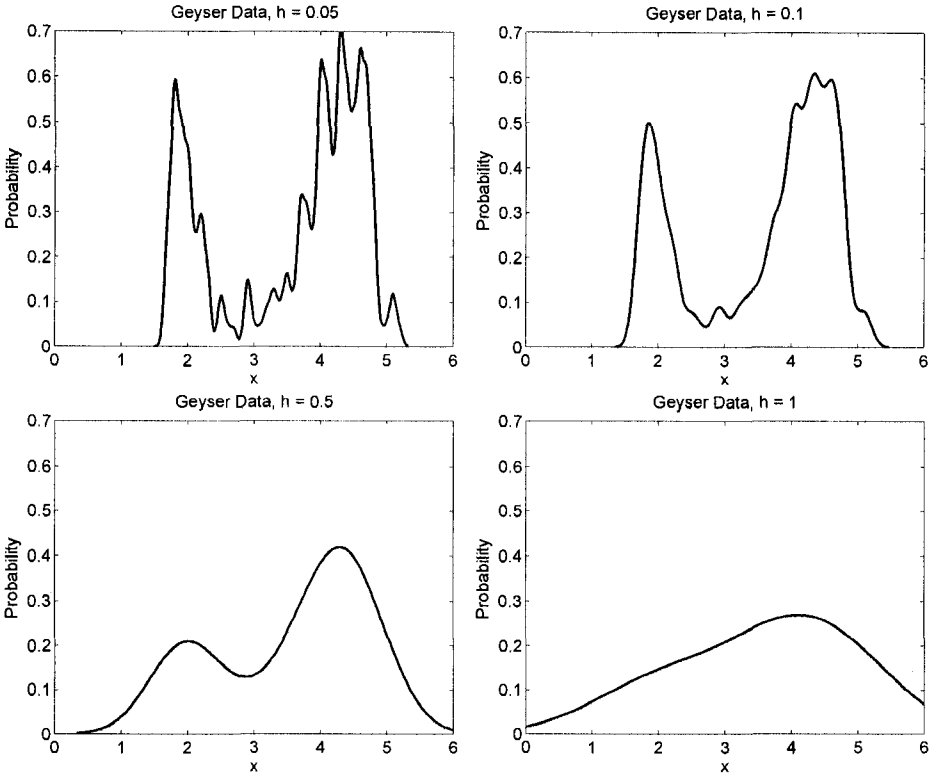

Figure 6.1: Kernel Density Estimates for varying smoothing parameters, $h = 0.05, 0.1, 0.5, 1$.

The empirical cumulative density function is given by

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} I_{(-\infty,x]}(x_i) = \frac{\#\{x_i \leq x\}}{n}.$$

To approximate the corresponding density function, one could use divided differences to approximate the derivative of the empirical cumulative density function,

$$\hat{f}(x) = \frac{F_n(x) - F_n(x-h)}{h} = \frac{1}{nh}\left[\sum_{i=1}^{n} I_{(-\infty,x]}(x_i) - \sum_{i=1}^{n} I_{(-\infty,x+h]}(x_i)\right]$$
$$= \frac{1}{nh}\sum_{i=1}^{n} I_{(x-h,x]}(x_i) = \frac{1}{nh}\sum_{i=1}^{n} I_{(0,1]}\left(\frac{x - x_i}{h}\right). \quad (6.3.1)$$

This is indeed a kernel estimator with kernel $K(x) = I_{(0,1]}$. The function is a step function that gives the number of points x_i in the interval $(x-h, x]$

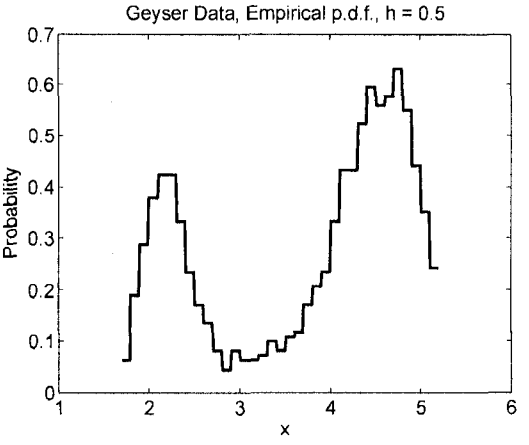

Figure 6.2: Approximate empirical density function.

for any given value of x. An example is shown in Fig. 6.2, using the geyser data with $h = 0.5$.

Note that if we take the limit $h \to 0$ in (6.3.1), we obtain the derivative of the empirical cumulative density function, which we call the empirical probability density function,

$$\frac{dF_n}{dx}(x) = \frac{1}{n}\sum_{i=1}^{n}\delta(x - x_i).$$

6.4 Matrix Algebra Background

Definition 6.4.1. *A stochastic matrix is a square matrix with non-negative entries in which each row sums to 1. A stochastic vector is a row vector with entries summing to 1.*

A stochastic vector v has the property that $ve = 1$, where e is defined by $e = (1, 1, ..., 1)^T$.

Definition 6.4.2. *If $vA = \lambda v$ for some scalar λ and row vector $v \neq 0$, we say that v is a left-eigenvector of A.*

The entire theory of eigenvectors could be done for left-eigenvectors. The eigenvalues would be the same. To see this, suppose that $Ax = \lambda x$ for $x \neq 0$, so that $\det(\lambda I - A) = 0$. Then,

$$\det(\lambda I - A) = \det[(\lambda I - A)^T] = \det(\lambda I - A^T).$$

Thus, A and A^T share eigenvalues and multiplicities. In addition, $A^T y = \lambda y$, or $y^T A = \lambda y^T$. In general, the eigenvectors will not be the same, that is $x \neq y$.

Theorem 6.4.1. *(Perron-Frobenius theorem for non-negative matrices). Let A be a real square matrix with non-negative entries. Then:*

1. *The spectral radius of A is an eigenvalue. That is, there is a real eigenvalue $r > 0$ such that $|\lambda| \leq r$ for all eigenvalues λ.*

2. *There is an eigenvector associated with r having non-negative entries.*

3. $$\min_i \sum_j a_{ij} \leq r \leq \max_i \sum_j a_{ij}. \tag{6.4.1}$$

Note that for a right stochastic matrix, $\sum_j a_{ij} = 1$ for each row i, therefore (6.4.1) implies that the dominant eigenvalue is $r = 1$. Therefore, the dominant left eigenvector π satisfies $\pi A = \pi$. In fact, since the rows of A sum to one, we have $A\,e = e$, where $e = (1, 1, ..., 1)^T$, the dominant right eigenvector. The Perron-Frobenius theorem also asserts that $|\lambda| \leq 1$ for all eigenvalues λ.

Definition 6.4.3. *An $m \times m$ matrix A is a permutation matrix if it is row equivalent to I_m. That is, if there is a 1 in every row and column and zeros elsewhere.*

Definition 6.4.4. *An $m \times m$ matrix A is reducible if there exists a permutation matrix P such that*

$$PAP^T = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix},$$

where A_{11}, A_{22} are square matrices. A is irreducible if no such matrix exists.

It can be shown that positive matrices (i.e., $a_{ij} > 0$ for each i, j) are irreducible.

Definition 6.4.5. *A non-negative, irreducible matrix is primitive if the number of eigenvalues having modulus equal to the spectral radius is 1.*

The proof of the following proposition is outlined in Appendix A

Proposition 6.4.1. *Let A be a primitive, stochastic matrix, where $xA = x, Ae = e$, and $L = ex$. Then $\lim_{m\to\infty} = L$.*

In the case that A is not primitive, the sequence xA is generally cyclic, and does not approach a limit. Suppose, for example, that A has eigenvalues of 1 and -1, and the other eigenvalues are distinct and satisfy $|\lambda| < 1$. In this case, we show that xA^k is a 2 cycle. Let $\{v_i\}_{i=1}^n$ be the left eigenvectors of A. As these form a basis for $\mathbb{R}^n$, for arbitrary x we can write

$$x = \sum_1^n \alpha_i v_i = \alpha_1 v_1 - \alpha_2 v_2 + \sum_{i=3}^n \alpha_i v_i.$$

If v_1, v_2 correspond to eigenvalues 1 and -1, respectively, then

$$xA = \alpha_1 v_1 A - \alpha_2 v_2 A + \sum_{i=3}^n \alpha_i v_i A = \alpha_1 v_1 - \alpha_2 v_2 + \sum_{i=3}^n \alpha_i \lambda_i v_i.$$

Inductively,

$$xA^k = \alpha_1 v_1 + \alpha_2 (-1)^k v_2 + \sum_{i=3}^n \alpha_i \lambda_i^k v_i.$$

Since $|\lambda_i| < 1$ for $i = 3, 4, ..., n$, we have

$$\lim_{k\to\infty} xA^k = \alpha_1 v_1 + \alpha_2(-1)^k v_2. \tag{6.4.2}$$

We see that for large values of k, the sequence $x\,A^k$ approximately alternates between the sequences $\alpha_1 v_1 + \alpha_2 v_2$ and $\alpha_1 v_1 - \alpha_2 v_2$.

6.5 Random Variables and Stochastic Processes

In this section, we review some facts about random variables and stochastic processes.

6.5.1 Joint Probability Density Functions

Let $f_n(X_1, ..., X_n)$ be the joint probability density function of random variables $X_1, ..., X_n$, so that for any measurable set $D \subset \mathbb{R}^n$,

$$P(X_1, ..., X_n \in D) = \int_D f_n(x_1, ..., x_n)\, dx_1 ... dx_n.$$

Then, for $s < n$,

$$f_s(x_1, ..., x_s) = \int f_n(x_1, ..., x_n) dx_{s+1} \cdots dx_n, \tag{6.5.1}$$

the marginal distribution for the subset. The equation (6.5.1) is sometimes called the Chapman-Kolmogorov equation. Note that in particular,

$$f_{n-1}(x_1, ..., x_{n-1}) = \int f_n(x_1, ..., x_n) dx_n,$$

and

$$f_1(x_i) = \int f_n(x_1, ..., x_n)\, dx_1 ... dx_{i-1}\, dx_{i+1} ... dx_n.$$

The conditional probability density function is denoted as $f_{s|r-s}(x_1, ...x_s|x_{s+1}, ...x_r)$, and has the property that

$$P(X_1, ...X_s \in D|X_{s+1} = x_{s+1}, ..., X_r = x_r) = \int_D f_{s|r-s}(x_1, ...x_s|x_{s+1}, ...x_r) dx_1 ... dx_s.$$

By Bayes rule,

$$f_r(x_1, ...x_r) = f_{r-s}(x_{s+1}, ...x_r) f_{s|r-s}(x_1, ..., x_s | x_{s+1}, ...x_r).$$

Definition 6.5.1. *If* $f_r(x_1, ..., x_r) = f_s(x_1, ..., x_s) f_{r-s}(x_{s+1}, ..., x_r)$, *we say the sets* $\{X_1, ..., X_s\}$ *and* $\{X_{s+1}, ..., X_r\}$ *are independent of each other. In this case,*

$$f_{s|r-s}(x_1, ..., x_s | x_{s+1}, ..., x_r) = f_s(x_1, ..., x_s).$$

6.5.2 Transformation of Random Variables

Let X be a random variable with density function $f_X(x)$. Let $Y = g(X)$ be a new random variable with density function $f_Y(x)$. We would like to express $f_Y(y)$ in terms of $f_X(x)$. If $g(x) = x$, we know that

$$f_X(x) = f_Y(y) = \int \delta(x - y) f_X(x)\, dx.$$

For arbitrary g, we write

$$F_Y(y + \Delta y) - F_Y(y) = P(y < Y \leq y + \Delta y) = \int_{[y<g(x)\leq y+\Delta y]} f_X(x)\, dx.$$

Dividing by Δy,

$$\begin{aligned} f_Y(y) \approx \frac{F(y + \Delta y) - F(y)}{\Delta y} &= \frac{1}{\Delta y} \int_{[y<g(x)\leq y+\Delta y]} f_X(x)\, dx \\ &= \int \frac{H(g(x) - y) - H(g(x) - (y + \Delta y))}{\Delta y} f_X(x)\, dx, \end{aligned}$$

where H is the Heaviside function. Taking $\Delta y \to 0$ yields the relationship

$$f_Y(y) = \int \delta(g(x) - y) f_X(x)\, dx.$$

6.5.3 Stochastic Processes

Definition 6.5.2. *A stochastic process is a family of random variables X_t for t in some indexed set T.*

For a continuous time stochastic process, we typically have $T = [0, \infty)$. A discrete time process corresponds to $T = \{0, 1, 2, ...\}$. The state space is the range of the random variables. If the state space S is countable, we say that X_t is a discrete valued stochastic process. We sometimes denote the stochastic process with $Y_X(t) = f(X, t)$. A realization of the process is $Y_x(t) = f(x, t)$. The probability density function for Y_t is given by

$$f(y, t) = \int \delta(y - Y_t(\omega)) f(\omega)\, d\omega.$$

Definition 6.5.3. *A Markov process is a stochastic process X_t with the property that for any set of n successive times ($t_1 < t_2 < ... < t_n$),*

$$f_{1|n-1}(x_{t_n}|x_{t_1}, ..., x_{t_{n-1}}) = f_{1|1}(x_{t_n}|x_{t_{n-1}}).$$

That is, given $X(s)$, the values of $X(t)$ for $t > s$ do not depend on the values of $X(u)$ for $u < s$. This can be stated as, if $t_1 < t_2 < ... < t_n < t$, then

$$P(a \leq X_t \leq b|X_{t_1} = x_1, ..., X_{t_n} = x_n) = P(a \leq X_t \leq b|X_{t_n} = x_n).$$

For a Markov process, we may generate an identity on transition densities. By applying Bayes law, we know that for $t_1 < t_2 < t_3$,

$$f_3(x_{t_1}, x_{t_2}, x_{t_3}) = f_1(x_{t_1}) f_{1|1}(x_{t_2}|x_{t_1}) f_{1|2}(x_{t_3}|x_{t_1}, x_{t_2}).$$

By the Markov property,

$$f_3(x_{t_1}, x_{t_2}, x_{t_3}) = f_1(x_{t_1}) f_{1|1}(x_{t_2}|x_{t_1}) f_{1|1}(x_{t_3}|x_{t_2}).$$

Integrating the above with respect to x_{t_2}, we obtain

$$\int f_3(x_{t_1}, x_{t_2}, x_{t_3})\, dx_{t_2} = \int f_1(x_{t_1}) f_{1|1}(x_{t_2}|x_{t_1}) f_{1|1}(x_{t_3}|x_{t_2})\, dx_{t_2}.$$

Using (6.5.1),

$$f_2(x_{t_1}, x_{t_3}) = f_1(x_{t_1}) \int f_{1|1}(x_{t_2}|x_{t_1}) f_{1|1}(x_{t_3}|x_{t_2}) dx_{t_2},$$

and therefore

$$f_1(x_{t_1}) f_{1|1}(x_{t_3}|x_{t_1}) = f_1(x_{t_1}) \int f_{1|1}(x_{t_2}|x_{t_1}) f_{1|1}(x_{t_3}|x_{t_2}) dx_{t_2},$$

and dividing yields the forward Kolmogorov equation,

$$f_{1|1}(x_{t_3}|x_{t_1}) = \int f_{1|1}(x_{t_2}|x_{t_1}) f_{1|1}(x_{t_3}|x_{t_2}) dx_{t_2}. \tag{6.5.2}$$

This equation gives a relationship between the values of the stochastic process at three times $t_1 < t_2 < t_3$. We can compute the transition probabilities from t_1 to t_3, if we know the transition probabilities from t_1 to t_2, and from t_2 to t_3.

6.6 Markov Chains

A simple class of Markov processes are Markov chains. A Markov chain is a Markov process in which the both the time variable and the state space are discrete.

Definition 6.6.1. *Let $\{X_n\}_{n=0}^{\infty}$ be a stochastic process that takes values from a countable set S, called the state space. We say that X_n is a Markov chain if*

$$P(X_{n+1} = j | X_0 = i_0, X_1 = i_1, ..., X_{n-1} = i_{n-1}, X_n = i) = P(X_{n+1} = j | X_n = i), \tag{6.6.1}$$

for $n = 0, 1, 2, ...$ and states $i_0, i_1, ..., i_{n-1}, i, j$.

The equation (6.6.1) is called the Markov condition. It is easy to show that (6.6.1) is equivalent to

$$P(X_{n+1} = j | X_{n_1} = i_1, ..., X_n = i_k) = P(X_{n+1} = j | X_n = i_k),$$

for all $0 \leq n_1 < n_2 < ... \leq n, j, i_1, ..., i_k$ in the state space and

$$P(X_{m+n} = j | X_0 = i_0, ..., X_m = i_m) = P(X_{m+n} = j | X_m = i_m),$$

for all $m, n \geq 0, i_0, ..., i_m, j$ in the state space. We assume Markov chains satisfy a homogeneity condition $P(X_{n+1} = j | X_n = i) = P(X_1 = j | X_0 = i)$ for all n. That is, the probability transitions do not depend on n, but upon the time difference alone. We then denote

$$p_{ij} = P(X_{n+1} = j | X_n = i),$$

the probability of X_{n+1} being in state j, given that X_n is in state i. Using the law of total probability, we can show that $\sum_{j=0}^{\infty} p_{ij} = 1$ for all i. Each vector $\{p_{ij}\}_{j=1}^{\infty}$ may be thought of as a probability mass function for the value of the chain, assuming the chain is in state i at the previous time step.

The simplest type of Markov chain is a finite Markov chain, in which the state space S is finite with N possible states. In this case, the probability mass function is an N-component vector and the transition probabilities $p_{ij} = P(X_2 = i | X_1 = j)_{i,j}$ may be thought of as a $N \times N$ matrix. We call this the probability transition matrix of the Markov chain. Since the row sums are one, the probability transition matrix is a stochastic matrix.

We develop an analogous equation to (6.5.2). By the law of total probability,

$$P(X_1 = i, X_3 = j) = \sum_k P(X_1 = i, X_2 = k, X_3 = j). \tag{6.6.2}$$

By the Markov property,

$$P(X_1 = i, X_2 = k, X_3 = j)$$
$$= P(X_1 = i)P(X_2 = k|X_1 = i)P(X_3 = j|X_2 = k),$$

and summing each side over k,

$$\sum_k P(X_1 = i, X_2 = k, X_3 = j)$$
$$= \sum_k P(X_1 = i)P(X_2 = k|X_1 = i)P(X_3 = j|X_2 = k).$$

Now using (6.6.2),

$$P(X_1 = i, X_3 = j) = P(X_1 = i) \sum_k P(X_2 = k|X_1 = i)P(X_3 = j|X_2 = k).$$

After applying Bayes law on the left hand side, we obtain

$$P(X_1 = i)P(X_3 = j|X_1 = i)$$
$$= P(X_1 = i) \sum_k P(X_2 = k|X_1 = i)P(X_3 = j|X_2 = k).$$

Dividing gives the Kolmogorov equation for Markov chains,

$$P(X_3 = j|X_1 = i) = \sum_k P(X_2 = k|X_1 = i)P(X_3 = j|X_2 = k).$$

We may easily generalize to the case of arbitrary time steps between the random variables. Indeed,

$$P(X_{m+n+r} = j|X_m = i)$$
$$= \sum_k P(X_{m+n} = k|X_m = i)P(X_{m+n+r} = j|X_{m+n} = k), \quad (6.6.3)$$

for $m, n, r \geq 0$.

6.6.1 Recurrence and Periodicity

The following definitions characterize the long time behavior of states in a Markov chain. In particular, assuming the Markov chain starts in a particular state, we want to know the probability that the chain returns to the state.

Definition 6.6.2. *For a Markov Chain with state space S, we say that state i is recurrent if $P(X_n = i \text{ for some } n \geq 1 | X_0 = i) = 1$, and is transient if $P(X_n = i \text{ for some } n \geq 1 | X_0 = i) < 1$.*

That is, when a chain exits a recurrent state, it is sure to return, and when a chain exits a transient state there is a nonzero probability of the chain not returning.

Definition 6.6.3. *The mean recurrence time of state i of a Markov chain is the average number of steps required to return to state i given that it started from state i. That is, $\mu_i = E[T_i]$, where $T_i = \min_n\{X_n = i | X_0 = i\}$.*

It can be shown that

$$\mu_i = \begin{cases} \sum_n n f_i(n), & i \text{ is recurrent,} \\ \infty, & j \text{ is transient,} \end{cases}$$

where $f_{ij}(n) = P(\{X_1 \neq i, ..., X_{n-1} \neq i, X_n = i | X_0 = i\})$. Note that μ_j may be infinite if i is recurrent.

Definition 6.6.4. *For a recurrent state i, i is null if $\mu_i = \infty$ and non-null or positive if $\mu_i < \infty$.*

Remark 6.6.1. *We denote p_{ij}^m to mean the (i, j) entry of the matrix P^m.*

Definition 6.6.5. *The period of state i is $d(i) = \gcd\{n : P_{ii}^n > 0\}$. If $d(i) = 1$, we say that state i is aperiodic and is periodic otherwise.*

Theorem 6.6.1. *If state j is positive recurrent and has period t, then*

$$\lim_{n\to\infty} p_{jj}^{nt} = t/\mu_j.$$

Proposition 6.6.1. *If state j is null recurrent or transient, then as $n \to \infty$,*

$$p_{jj}^{n} \to 0, \text{ and } p_{ij}^{n} \to 0 \text{ for all } i.$$

Definition 6.6.6. *A state is ergodic if it is positive recurrent and aperiodic.*

6.6.2 Irreducibility

Definition 6.6.7. *A Markov chain is irreducible if it has a single communicating class. That is, if for every i, j in the state space, there is some m such that $p_{ij}^{m} > 0$.*

It can be shown that if a Markov chain is irreducible, then either all states are transient, or all states are recurrent. Also, states of a Markov chain share the same period.

A Markov chain is irreducible if and only if its transition matrix is irreducible. Another equivalent condition for irreducibility is for its digraph to be strongly connected, where we think of the matrix as the adjacency matrix of the graph. These conditions are difficult to check. However, we may use the following to test for irreducibility, the proof of which can be found in Appendix A.

Proposition 6.6.2. *A finite Markov chain with n states is irreducible if and only if its probability transition matrix P satisfies $(I + P)^{n-1} > 0$.*

We may also use Proposition 6.6.2 to determine if a finite chain has recurrent states by use of the following.

Proposition 6.6.3. *For a finite irreducible Markov chain, all of the states are positive recurrent.*

6.6.3 Limit Behavior

Definition 6.6.8. *Let S be the state space of a Markov chain. The row vector $\pi = (\pi_j)_{j \in S}$ is a stationary distribution of the chain if $\pi_j \geq 0$ for all j, $\sum_j \pi_j = 1$, and $\pi_j = \sum_{i \in S} \pi_i p_{ij}$, (i.e. $\pi = \pi P$.)*

Remark 6.6.2. *For a finite Markov chain we can use Theorem 6.4.1 to deduce that P will always have a stationary distribution, which is the left eigenvector corresponding the the eigenvalue 1.*

Definition 6.6.9. *If there exists a probability distribution q on a state space S such that $\lim_{n \to \infty} p_{ij}^n = q_j$ for all i, j, then q is a limiting distribution of the chain.*

This convergence means that, in the long run, as $n \to \infty$, the probability of finding the Markov chain in state j is approximately q_j, independent of the initial condition.

We list some basic limit theorems below.

Theorem 6.6.2.

1. *An irreducible chain has a stationary distribution π if and only if all states are positive recurrent. In this case, π is unique, and moreover, $\pi_i = 1/\mu_i$ for all i, where μ_i is the mean recurrence time of state i.*

2. *If P is the transition matrix for an irreducible aperiodic Markov Chain, then*

$$\lim_{n \to \infty} p_{ij}^n = \frac{1}{\mu_j} \text{ for all } i, j.$$

3. *If a chain is positive recurrent and aperiodic (i.e., ergodic) then there is a limiting distribution which is also the unique stationary distribution.*

Corollary 6.6.1. *1. If the chain is irreducible, aperiodic, and in addition either transient or null-recurrent, then* $p_{ij}^n \to 0$ *for all* i, j *as* $n \to \infty$.

2. If the chain is irreducible, aperiodic, and positive recurrent, then $p_{ij}^n \to \pi_j = 1/\mu_j$, *which is also the unique stationary distribution.*

Remark 6.6.3. *If a chain has a limiting distribution* π, *then* $vP^n \to \pi$ *for any stochastic row vector* v, *and* $P^n \to e\pi$. *To see one direction, if* $P^n \to e\pi$, *then*

$$ve\pi = (v_1, ..., v_n) \cdot (1, ..., 1)\pi = \pi,$$

since $\sum_i v_i = 1$. *Thus,* $vP^n \to \pi$. *Conversely, if* $vP^n \to \pi$, *then* $e\,vP^n \to e\pi$. *Since* v *is a stochastic vector,* $e\,v = 1$, *and so* $P^n \to e\pi$.

One of the ways to verify that a finite Markov chain has a limiting distribution is to examine its spectrum. Recall that by Proposition 6.4.1, $\lim_{m\to\infty} P^m \to e\,x$, where x is the dominant left eigenvector. This shows that P has limiting distribution x in view of Remark 6.6.3. Since $x > 0$, it follows that $e\,x > 0$, so that $P^m > 0$ for some $m \geq 1$.

Proposition 6.6.4. *If a finite Markov chain has a primitive transition matrix* P, *then it has a limiting distribution, and* $P^m > 0$ *for some* $m \geq 1$.

Assuming that $P^m > 0$ for some $m > 0$, it follows that $P^k > 0$ for all $k \geq m$. Hence, for any state i, $P_{ii}^n > 0$ for $n \in \{m, m+1, ..., m+n\}$, which only has one as a common divisor. The Markov chain in this case is aperiodic. This leads to the following Corollary.

Corollary 6.6.2. *If a finite Markov chain has a primitive transition matrix, then it is aperiodic.*

The proof of the following converse is given in [27].

Proposition 6.6.5. *If a Markov chain is aperiodic and irreducible, then it has a primitive transition matrix.*

Given that a finite Markov chain has a limiting distribution, the following proposition gives a rate of convergence.

Proposition 6.6.6. *Suppose that P is a primitive stochastic matrix. If λ_2 is an eigenvalue such that $|\lambda_2| < r < 1$, and $|\lambda| \leq |\lambda_2|$ for all $\lambda \neq 1$, then there is a constant C depending on P and r such that*

$$\|P^m - e\,x\|_\infty \leq C\,r^m, \text{ for all } m \geq 1.$$

This shows that the convergence rate is dependent on the second largest eigenvalue. The rate slows as $\lambda_2 \to 1$ and speeds up as $\lambda_2 \to 0$. We thus view $|\lambda_1/\lambda_2|$, or in our case,

$$1/|\lambda_2| \tag{6.6.4}$$

as a convergence rate for the sequence P^m.

If P is periodic with period d, the sequence vP^n fails to converge in general. However, the following proposition gives a characteristic of the long term behavior of the sequence. The proof is given in Appendix A.

Proposition 6.6.7. *Suppose P is periodic with period 2. Then for large n, the sequence vP^n is a 2 cycle, whose average approaches the stationary distribution π. That is,*

$$\pi_j = \lim_{n\to\infty} \frac{1}{2}\left(p_{ij}^n + p_{ij}^{n+1}\right).$$

In general, if P is irreducible with period d, then P will have d complex eigenvalues. For each of these d eigenvalues, z, we have $|z| = 1$, and $z^d = 1$. In addition, each of the d eigenvalues is simple. In this case, vP^n will cycle through d different distributions, but they will average to the stationary distribution π. That is,

$$\lim_{n\to\infty} \frac{1}{d}(vP^{n+1} + ... + vP^{n+d}) = \pi.$$

6.6.4 Reducible Chains

In probability transition matrices encountered in our research project, we often find a column of zeros, meaning that some state cannot be reached from the other states. We may still obtain a limiting distribution, provided that the nonzero states are aperiodic and irreducible. The zero columns will correspond with zero entries in the limiting distribution. We use the following theorem which states that we can essentially remove the zero columns and corresponding rows from the matrix, before applying Theorem 6.6.2 on the remaining states.

Theorem 6.6.3. *Suppose a Markov chain contains finitely many states, denoted by S. Furthermore, suppose that the chain is reducible and contains a single closed class $C \subset S$ of aperiodic states. Let P be the transition matrix for states belonging to C. Since this closed class is in itself an irreducible, aperiodic Markov chain, we have $P^n \to e\pi$. Then, if we denote the transition matrix for S by P_S, we have*

$$P_S^n \to e(\pi, 0).$$

Example 6.6.1. *Consider the matrix* $\begin{bmatrix} 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$. *Using Propositions 6.6.2 and 6.6.3, we examine the matrix* $(I+P)^{5-1}$, *and see that 1,3,4 are recurrent states, as they correspond to the positive columns. States 2 and 5 are transient states. The subset* $\{1,3,4\}$ *has period 2 and hence is not ergodic. In fact, P has an eigenvalue of -1.*

Example 6.6.2. *Consider the matrix* $\begin{bmatrix} 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$. *By examining the matrix* $(I+P)^{5-1}$, *we see that states 2 and 3 are recurrent. The set* $\{2,3\}$ *is also aperiodic, hence ergodic. So there is a limiting distribution* $\pi = (0, 0.5, 0.5, 0, 0)$ *where the nonzero entries correspond with the transient states.*

Now suppose P has multiple recurrent classes $R_1, ..., R_r$, and transient classes $T_1, ..., T_s$. For R_k, we have a stationary distribution π^k such that $\pi_i^k = 0$ if $i \notin R_k$. Assume the submatrix P_k corresponding to R_k is aperiodic. Then, $p_{ij}^n \to \pi_j^k$ for $j \in R_k$, and $p_{ij}^n \to 0$ if $j \notin R_k$. As before, if j is transient, then $p_{ij}^n \to 0$ for all i.

6.7 Continuous State Markov Chains

For a discrete-time Markov process with a continuous state space, $X_0, X_1, ...$ are random variables such that

$$P(X_{m+1} \leq x | X_0 = y_0, ..., X_{m-1} = y_{m-1}, X_m = y) = P(X_{m+1} \leq x | X_m = y). \tag{6.7.1}$$

If the cumulative density function of the transitional probability (6.7.1) is independent of m, we say the chain is homogeneous. The n-step density function is defined by

$$P_n(x;y) \equiv P(X_{m+n} \leq x | X_0 = y_0, ..., X_{m-1} = y_{m-1}, X_m = y)$$
$$= P(X_{m+n} \leq x | X_m = y),$$

the probability of the process jumping from y to a value less than or equal to x in n steps.

Note that $P_n(x;y)$ and $P_0(x) \equiv P(X_0 \leq x)$ determines $P_n(x)$ uniquely. The Chapman-Kolmogorov equation is

$$P_{n+m}(x;y) = \int_{\mathbb{R}} P_n(z;y) P_m(x;z) dz.$$

Note that this corresponds to the Chapman-Kolmogorov equation for the discrete state space, (6.6.3).

Suppose $P_n(x;y) \to P(x)$ as $n \to \infty$ (regardless of $P_0(x)$.) Then the limiting distribution $P(x)$ of the Markov chain satisfies the stationary condition

$$P(x) = \int P(z) P(x;z) dz = \int P(x;z) dP(z).$$

This corresponds to the case of a discrete valued Markov chain in that if $p_{ij}^n \to v_j$, and the chain is finite, irreducible, and aperiodic, then v coincides with the stationary distribution, so that $v_j = \sum_i v_i p_{ij}$, $\sum_j v_j = 1$.

Analyzing a chain with a continuous state space poses challenges in that the theory is not as well developed as in the discrete time case. However, special methods are developed for certain types of chains, such as random walks.

Chapter 7

BACKGROUND: ATOMISTIC TECHNIQUES

7.1 Molecular Dynamics

Molecular dynamics (MD) is a computer simulation technique where the time evolution of a set of interacting atoms is followed by integrating their equations of motion. This is done by following Newton's law: $F_i = m_i a_i$ for each atom i in a system of N atoms, where m_i is the atom mass, a_i its acceleration, and F_i the force acting on it due to interactions with other atoms. Molecular dynamics may be viewed as a simulation of the system as it develops over time.

Molecular dynamics is a statistical mechanics method. The goal is to obtain a set of configurations distributed according to some statistical distribution function, or statistical ensemble. The microcanonical, or NVE ensemble, is the natural statistical ensemble to use with MD. Here the number of molecules, N, the volume, V, and the total energy, E remain constant. We assume that there is no heat exchange, and the total energy is conserved. In three dimensions, we keep track of the position $r_i(t) \in \mathbb{R}^3$ and the velocity $v_i(t) \in \mathbb{R}^3$ of each particle i at time t. If we consider phase space to be the $6N$ dimensional space of all possible velocities and positions of the

N particles, we wish to describe the distribution of some statistical quantity of interest in phase space. This process requires ergodicity in order to be valid. The ergodic hypothesis of statistical physics states that observing a process for a long time is equivalent to sampling many realizations of the same process. That is, if we run the simulation forward in time using a single initial condition, then for long time the trajectory will densely fill in the manifold and we use this to compute statistical quantities. Typically, however, engineers will perform several MD simulations for reasonably long times and average the results.

7.1.1 Quantities to Estimate

In general, consider a property A at time t:

$$A(t) = f(r_1(t), ..., r_N(t), v_1(t), ..., v_N(t)).$$

After an MD simulation we may be interested in its time average over the system trajectory: $E[A] = \frac{1}{N_T} \sum_{t=1}^{N_T} A(t)$, where N_T is the total number of time steps. Examples of quantities A include:

- Average Potential Energy: $V(r(t)) = \sum_i \sum_{j>i} \phi(|r_i(t) - r_j(t)|)$, where $\phi(r)$ is the potential energy between atoms separated by distance r.
- Kinetic Energy: $K(t) = \frac{1}{2} \sum m_i v_i(t)^2$.
- Total energy: $H(r(t), v(t)) = K(t) + V(r(t))$.
- Temperature: $T = \frac{2}{3} \frac{K}{N k_B}$, where K is the average kinetic energy and k_B is the Boltzmann constant (Equipartition formula with 3 degrees of freedom).

- Mean Square Displacement: MSD $= E[(r(t) - r(0))^2]$, where $E[\cdot]$ indicates the average over all N particles. This contains information on the atomic diffusivity. The diffusion coefficient, D, is given by

$$D = \lim_{t \to \infty} \frac{1}{6t} E[(r(t) - r(0))^2],$$

where 6 is replaced by 4 in two-dimensional systems.

7.1.2 A Molecular Dynamics Program

The program consists of the following steps.

- Read in parameters that specify the conditions of the simulation (e.g., initial temperature, number of particles, density, time step).
- Select initial positions and velocities.
- Compute forces on all particles.
- Integrate Newton's equations of motion. This step and the previous step are repeated for the desired length of time.
- Compute the averages of measured quantities.

One way to set up the initial positions is to arrange the particles in a lattice. Then we could give random velocities according to some distribution. If we are using the micro-canonical ensemble for example, we would shift velocities so that the velocity of the center of mass is zero and scale the resulting velocities to adjust the mean kinetic energy to the desired value.

We must choose an appropriate potential for the model. Potential energy is typically a sum of pairwise interactions, and is given by $V(r) = \sum_i \sum_{j>i} \phi(|r_i - r_j|)$. For example, the Lennard-Jones potential is frequently

used, which is given by $\phi(r) = 4\epsilon\left(\left(\frac{\sigma}{r}\right)^{12} - 2\left(\frac{\sigma}{r}\right)^{6}\right)$, where r is the atomic separation. In this potential, the term $1/r^{12}$ dominates at a short distance, and models the repulsion between atoms which are close to each other. The term $1/r^6$ dominates at a large distance, and models the attraction between atoms. Next, we compute the forces using the relationship $F_i = -\nabla_{r_i} V(r)$. We then obtain the new positions $r(t)$ and velocities $v(t)$ by applying an integrating scheme to solve $F = m\,a$. After applying the potentials to each particle we use Newton's equations to get new positions and velocities of the particles at each new time step $t + \Delta t$. In this section, we discuss the Verlet, Leap Frog, and Velocity-Verlet algorithms.

7.1.3 Integration Algorithms

After applying forces to each particle, we use Newton's equations to get new positions and velocities of the particles at the new time step $t + \Delta t$. Assuming we use the microcanonical ensemble, we would like to hold the total energy constant. The integrators that we discuss have the property that the total energy drift is minimized, provided that the time steps are sufficiently small. To derive the Verlet algorithm we use a third order Taylor expansion:

$$r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{1}{2}\frac{F(t)}{m}\Delta t^2 + \frac{1}{6}r'''(t)\Delta t^3 + \mathbf{O}(\Delta t^4) \quad (7.1.1)$$

where we have used $r'(t) = v(t)$ and Newton's law $F(t) = mr''(t)$. Similarly,

$$r(t - \Delta t) = r(t) - v(t)\Delta t + \frac{1}{2}\frac{F(t)}{m}\Delta t^2 - \frac{1}{6}r'''(t)\Delta t^3 + \mathbf{O}(\Delta t^4) \quad (7.1.2)$$

Adding these equations we obtain the Verlet algorithm:

$$r(t + \Delta t) = 2r(t) - r(t - \Delta t) + \frac{F(t)}{m}\Delta t^2 + \mathbf{O}(\Delta t^4) \quad (7.1.3)$$

Given the position of a particle at times t and $t - \Delta t$, we find the position of the particle at time $t + \Delta t$. This allows us to compute the new potential and forces.

To acquire the new kinetic energy and temperature we need to compute the velocity. A simple way to compute the velocity is to subtract (7.1.2) from (7.1.1) to obtain

$$r(t + \Delta t) - r(t - \Delta t) = 2v(t)\Delta t + \mathbf{O}(\Delta t^3).$$

Then,

$$v(t) = \frac{r(t + \Delta t) - r(t - \Delta t)}{2\Delta t} + \mathbf{O}(\Delta t^2).$$

To derive the Leap-Frog algorithm, we compute velocity at half time steps,

$$v\left(t + \frac{\Delta t}{2}\right) = v\left(t - \frac{\Delta t}{2}\right) + \frac{f(t)}{m}\Delta t,$$

and at full time steps we compute the new positions,

$$r(t + \Delta t) = r(t) + v\left(t + \frac{\Delta t}{2}\right)\Delta t.$$

It turns out that this method yields the same trajectories as the Verlet scheme. However, since r and v are computed at different time steps, we cannot compute the total energy.

Another algorithm that will turn out to be equivalent to the original Verlet scheme is the Velocity-Verlet algorithm. To obtain the new positions, we use a second order expansion

$$r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{f(t)}{2m}\Delta t^2.$$

To obtain the new velocities, use a second order expansion, averaging the forces at t and $t + \Delta t$.

$$v(t + \Delta t) = v(t) + \frac{f(t + \Delta t) + f(t)}{2m}\Delta t.$$

To reduce the number of flops, we store $v(t) + \frac{f(t)}{2m}\Delta t$, which are computed in both steps. This algorithm generates the same trajectories as the original Verlet scheme, in a more efficient manner.

7.2 Other Stochastic Techniques

It is possible to perform MD in other ensembles. Instead of keeping the total energy constant, we could instead keep temperature or pressure constant[25, 46]. MD has several shortcomings. We note that quantum laws hold rather than Newton's law, and perhaps the Schrodinger equation should be used. However, for most particles, the classical approximation is valid. Quantum effects become important if temperature is low, and if the particles are very lightweight (H_2, He, Ne for example.) There are methods such as quantum molecular dynamics and quantum Monte Carlo, which take these quantum effects into account. However, these methods are much more demanding in terms of computational power.

The main disadvantage of MD is that many applications require short time steps, on the order of $10^{-15}s$ [49]. Realistically, we can run the simulation on the order of microseconds, but to obtain the desired quantities of interest we need to run the simulation for much longer time.

Monte Carlo methods are different in that they approximate the system by following a directed random walk. These methods are very popular for studying static properties of a system. Kinetic Monte Carlo methods [49, 12, 34] approximate the evolution of a system in time. These methods overcome the time scale problem by using the fact that the long-time dynamics for certain types of systems consists of diffusive jumps from state to state. In the kinetic Monte Carlo method, we assume that all processes occur with

known transition rates. We can expect a considerable savings in cost over MD since there are typically long periods of inactivity between transitions among states.

7.3 Random Walks and the Diffusion Equation

In this section, we examine the unrestricted one-dimensional random walk problem, and explain its connection to the diffusion equation (2.4.1). The following derivations can be found in detail in [52]. Assume that a particle starts at the origin of the x-axis and takes steps of length δ to the left or to the right. The probabilities for the increment x_i are $P(x_i = \delta) = p$, and $P(x_i = -\delta) = q$. We view $\{x_i\}$ as independent identically distributed random variables which take on the value δ if the particle moves to the right at the ith step and $-\delta$ if the particle moves to the left. The position of the particle after n steps is $X_n = \sum_{i=1}^{n} x_i$. We compute the expectation of a single x_i,

$$E(x_i) = \delta P(x_i = \delta) + (-\delta)P(x_i = -\delta) = \delta p - \delta q = (p - q)\delta.$$

We use linearity to compute the expected value of X_n,

$$E(X_n) = E\left(\sum_{i=1}^{n} x_i\right) = \sum_{i=1}^{n} E(x_i) = (p - q)\delta n.$$

To compute the variance, we note that

$$E[x_i^2] = \delta^2 P(x_i = \delta) + (-\delta)^2 P(x_i = -\delta) = \delta^2(p + q) = \delta^2,$$

and so

$$\sigma^2(x_i) = E[x_i^2] - E[x_i]^2 = \delta^2 - (p - q)^2\delta^2.$$

Since the $\{x_i\}$ are independent,

$$\begin{aligned}\sigma^2(X_n) &= \sigma^2(\sum_{i=1}^{n} x_i) = \sum_{i=1}^{n} \sigma^2(x_i) = n\delta^2(1-(p-q)^2) \\ &= n\delta^2(1-(p^2-2pq+q^2)) = n\delta^2(1-(p^2+2pq+q^2)+4pq) \\ &= n\delta^2(1-(p+q)^2+4pq) = 4pq\delta^2 n,\end{aligned}$$

where we have used the fact that $(p+q)^2 = 1$.

To derive the Brownian motion model we assume that we have experimentally found the average displacement of the particle per unit time c, and the variance of the displacement to be D. If we assume that there are r collisions per unit time, then after r steps,

$$(p-q)\delta r \approx c,$$

and

$$4pq\delta^2 r \approx D. \tag{7.3.1}$$

If we assume that $p = q = \frac{1}{2}$, then (7.3.1) becomes $\delta^2 r \approx D$.

To make the problem continuous, we take the step length $\delta \to 0$ and $r \to \infty$. If $p = q = \frac{1}{2}$, then it is clear that $\delta^2 r \to D$ in the limit. If $p \neq q$, then

$$\delta r \to \frac{c}{p-q} \text{ as } \delta \to 0 \text{ and } r \to \infty. \tag{7.3.2}$$

This, in turn, implies

$$4pq\delta^2 r = 4pq\delta(\delta r) \to \frac{4pqc}{p-q}\delta \to 0, \text{ as } \delta \to 0, r \to \infty.$$

But we know that the variance should tend to $D \neq 0$ in this limit, so we require $p - q \to 0$ as $\delta \to 0, r \to \infty$. Hence (7.3.1), along with the fact that $p \to \frac{1}{2}, q \to \frac{1}{2}$, imply that

$$\delta^2 r \to D, \text{ as } \delta \to 0, r \to \infty. \tag{7.3.3}$$

Since r is the number of steps per unit time, then one step occurs in $1/r = \tau$ units of time, and n steps occur in $n/r = n\tau$ time units. In our random walk model, we would like to obtain the probability that the particle is at position x at time t, given that it started at $x = 0$ at time $t = 0$. After, n steps, we require that $X_n = x$. Since n steps occur in $n\tau$ time units, we also require that $n\tau = t$.

We define $v(x,t) = P(X_n = x)$ at time $t = n\tau$ to be the probability that at time t, the particle is located at point x. We could determine $v(x,t)$ explicitly using the binomial distribution, but since we are interested in the limit as the position and time steps $\delta, \tau \to 0$, we instead construct a difference equation satisfied by v. We have:

$$\begin{aligned} v(x,t+\tau) &= P(X_n = x \text{ at time } t+\tau) \\ &= P(X_n = x-\delta \text{ at time } t)\, p + P(X_n = x+\delta \text{ at time } t)\, q \\ &= p\, v(x-\delta,t) + q\, v(x+\delta,t). \end{aligned} \tag{7.3.4}$$

Using Taylor's formula,

$$\begin{aligned} v(x,t+\tau) &= v(x,t) + \tau v_t(x,t) + \mathbf{O}(\tau^2), \text{ and} \\ v(x \pm \delta, t) &= v(x,t) \pm v_x(x,t) + \frac{1}{2}\delta^2 v_{xx}(x,t) + \mathbf{O}(\delta^3). \end{aligned}$$

Substituting these into (7.3.4),

$$\begin{aligned} v(x,t) + \tau v_t(x,t) + \mathbf{O}(\tau^2) = v(x,t+\tau) &= p\, v(x-\delta,t) + q\, v(x+\delta,t) \\ &= p(v(x,t) - \delta v_x(x,t) + \frac{1}{2}\delta^2 v_{xx}(x,t) + \mathbf{O}(\delta^3)) \\ &\qquad + q(v(x,t) + \delta v_x(x,t) + \frac{1}{2}\delta^2 v_{xx}(x,t) + \mathbf{O}(\delta^3)) \\ &= (p+q)v + \delta(p-q)v_x + \frac{1}{2}(p+q)\delta^2 v_{xx} + (p+q)\mathbf{O}(\delta^3) \\ &= v + \delta(p-q)v_x + \frac{1}{2}\delta^2 v_{xx} + \mathbf{O}(\delta^3), \end{aligned}$$

so that

$$\tau\, v_t + \mathbf{O}(\tau^2) = \delta(p-q)v_x + \frac{1}{2}\delta^2 v_{xx} + \mathbf{O}(\delta^3),$$

or

$$v_t = \frac{\delta}{\tau}(p-q)v_x + \frac{1}{2}\frac{\delta^2}{\tau}v_{xx} + \mathbf{O}(\delta^3/\tau) + \mathbf{O}(\tau).$$

Taking $\delta \to 0, \tau \to 0$ yields the limiting partial differential equation

$$\frac{\partial v}{\partial t} = -c\frac{\partial v}{\partial x} + \frac{1}{2}D\frac{\partial^2 v}{\partial x^2}, \tag{7.3.5}$$

where we have used (7.3.2) and (7.3.3). The equation (7.3.5) is the diffusion equation in one dimension with diffusion coefficient D.

Example 7.3.1. *Consider the diffusion problem*

$$\begin{cases} u_t(t,x) = \frac{1}{2}Du_{xx}(t,x), & t > 0, x \in \mathbb{R}, \\ u(0,x) = \delta_0(x). \end{cases} \tag{7.3.6}$$

This corresponds to a random walk in which every particle lies at $x = 0$ at time $t = 0$. We simulate the random walk up to a final time of $t = 1$ several times and compare the results with the solution to (7.3.6),

$$u(t,x) = \frac{1}{\sqrt{2\pi Dt}} \exp\left(-\frac{x^2}{2Dt}\right).$$

Results are shown in Fig. 7.1.

7.3.1 Absorbing Boundary Conditions

In the case of an absorbing boundary at $x = l$, the particle exits at $x = l$ and cannot move back into the region $x < l$. Again, let p denote the probability of a jump to the right. Then,

$$v(l, t+\tau) = p\, v(l-\delta, t).$$

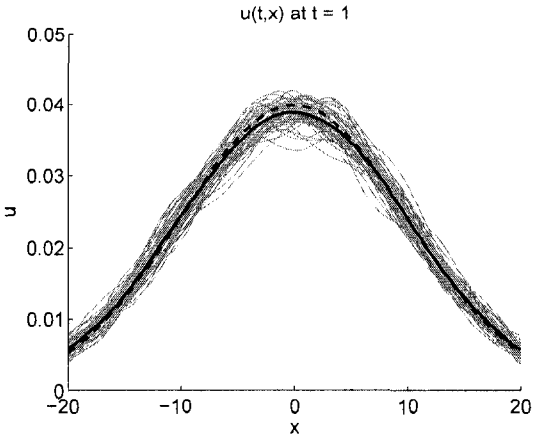

Figure 7.1: Kernel smoothed trajectories of 50 random walks at $t = 1$. The mean value of these random walks is shown in bold, and the true solution is the dashed line.

We have essentially given the condition that there is a zero probability of moving left from the right at the point $x = l$. By Taylor's theorem,

$$v(l, t + \tau) = v(l, t) + v_t(l, t)\tau + \mathbf{O}(\tau^2), \text{ and}$$
$$v(l - \delta, t) = v(l, t) - v_x(l, t)\delta + \mathbf{O}(\delta^2).$$

Thus,

$$\begin{aligned} v(l, t) + v_t(l, t)\tau + \mathbf{O}(\tau^2) = v(l, t + \tau) &= p\, v(l - \delta, t) \\ &= p\, v(l, t) - p\, v_x(l, t)\delta + p\, \mathbf{O}(\delta^2). \end{aligned}$$

Combining terms,

$$(1 - p)v(l, t) = -v_t(l, t)\tau - p\, \delta v_x(l, t) + \mathbf{O}(\tau^2) + p\mathbf{O}(\delta^2).$$

Now taking $\tau, \delta \to 0$, we have

$$(1 - p)v(l, t) = \mathbf{O}(\tau) + \mathbf{O}(\delta).$$

Hence, $v(l, t) = 0$ is the corresponding boundary condition for the continuous problem.

7.3.2 Reflecting Boundary Conditions

In the case of a reflecting boundary at $x = l$, the particle may move to the right with probability p, and in this case, the particle ends back at $x = l$ at time $t + \tau$. Then, the number of particles at $x = l$ at time t is given by sum of the number of particles that came from $x = l - \delta$ and the number of particles that jumped to the left from $x = l$. Therefore,

$$v(l, t + \tau) = p\, v(l - \delta, t) + p\, v(l, t).$$

Using Taylor's formulas,

$$\begin{aligned} v(l,t) + v_t(l,t)\tau + \mathbf{O}(\tau^2) = v(l, t+\tau) &= p\, v(l-\delta, t) + p\, v(l,t) \\ &= p\, v(l,t) - p\, v_x(l,t)\delta + p\, \mathbf{O}(\delta^2) + p\, v(l,t). \end{aligned}$$

Now, collecting terms,

$$v(l,t)(1-2p) + v_t(l,t)\tau + \mathbf{O}(\tau^2) = -p\, v_x(l,t)\delta + p\, \mathbf{O}(\delta^2).$$

Noting that $2p - 1 = 2p - (p + q) = p - q$, we have

$$v(l,t)(p-q) + v_t(l,t)\tau + \mathbf{O}(\tau^2) = -p\, v_x(l,t)\delta + p\, \mathbf{O}(\delta^2).$$

Multiplying by δ/τ,

$$v(l,t)(p-q)\delta/\tau + v_t(l,t)\delta + \mathbf{O}(\tau\delta) = -p\, v_x(l,t)\delta^2/\tau + p\, \mathbf{O}(\delta^3/\tau).$$

Next, taking $\tau, \delta \to 0$, we obtain

$$v(l,t)c = -\frac{D}{4q} v_x(l,t).$$

If $p = q = 1/2$, then we get $0 = -\frac{1}{2} D v_x(l,t)$, or simply $v_x(l,t) = 0$.

The probability transition matrices for the case of reflecting or absorbing boundary conditions takes on the following form, for the case of five grid points, $\begin{bmatrix} \alpha & 1-\alpha & 0 & 0 & 0 \\ q & 0 & p & 0 & 0 \\ 0 & q & 0 & p & 0 \\ 0 & 0 & q & 0 & p \\ 0 & 0 & 0 & 1-\alpha & \alpha \end{bmatrix}$. The parameter $\alpha = 0$ corresponds to a reflecting boundary, and $\alpha = 1$ corresponds to an absorbing boundary.

7.3.3 The Stationary Problem

To obtain the stationary elliptic problem, we simulate the random walk for a sufficiently long time. This results in approximating the equation $u''(x) = 0$. We pose this as a two-point boundary value problem. A nonzero Dirichlet boundary condition may be thought of as a source or sink of particles on the boundary. Implementing this boundary condition is discussed in detail in Chapter 9.

Consider the stationary diffusion problem:

$$\begin{cases} u'' = 0, & x \in \Omega = (0,1), \\ u(0) = 100, \\ u(1) = 200. \end{cases}$$

We simulate the random walk and approximate the gradient by computing the difference in the number of particles at $x = 1$ and $x = 1 - \Delta x$, at some final time t_f. We repeat this process N times and compute the gradient by taking an ensemble average. We expect the error to decrease as N increases. In fact, if we take a number of realizations of the process to give a distribution of ensemble averages, we expect the standard deviation of the M realizations to follow the asymptotic relationship

$$\sigma \approx c_1/\sqrt{N}. \tag{7.3.7}$$

In Fig. 7.3, we see that increasing N reduces the variance as expected.

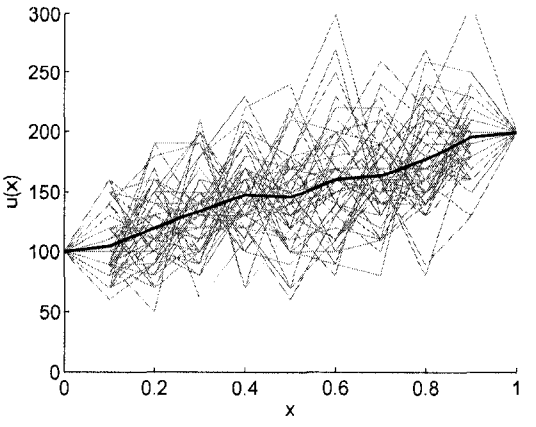

Figure 7.2: 50 random walks, with the mean shown in bold.

Alternatively, we compute the gradient at each time step for $t \in (t_b, t_f)$, where $t_b > 0$ is the burn time for the simulation, and compute the gradient as the mean of these values. If the process is ergodic, we expect that this yields similar results to computing the ensemble average as described above, and in particular we expect to see the asymptotic relationship

$$\sigma \approx c_2/\sqrt{t_f - t_b}. \tag{7.3.8}$$

In Fig. 7.4, we see that increasing the final time reduces the variance. Evidence of (7.3.7) and (7.3.8) is shown in Fig. 7.5.

We could also model diffusion with forcing, $a\,u'' = f$, in a region by adding particles at points in the domain as dictated by f.

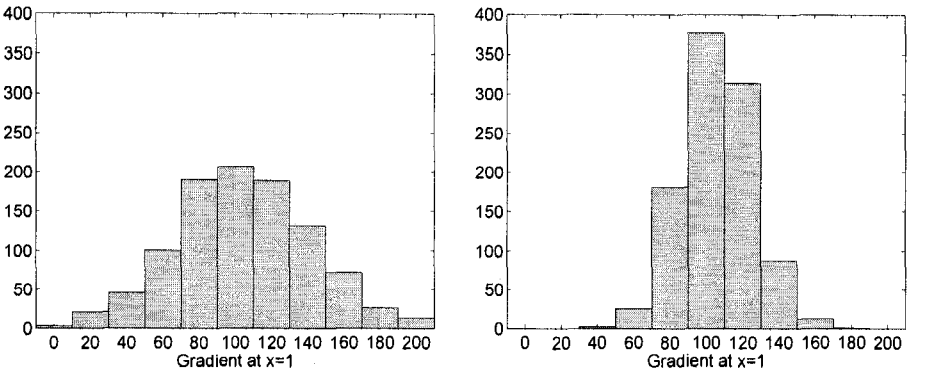

Figure 7.3: Computation of the gradient at $x = 1$ for 1000 realizations. We compute each gradient by taking the ensemble average of N random walks. Left: $N = 100$. Right: $N = 400$.

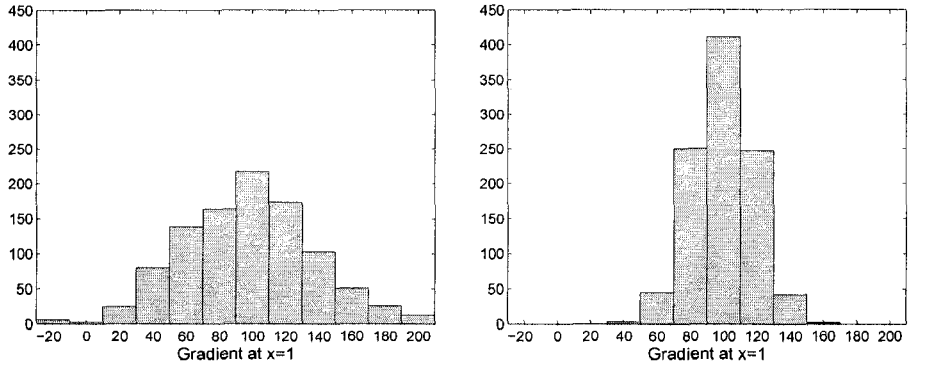

Figure 7.4: Computation of the gradient at $x = 1$ for 1000 realizations. We compute each gradient by taking the average gradient for each time step, for each $t \in (t_b, t_f)$. Left: $t_f = 10$. Right: $t_f = 50$.

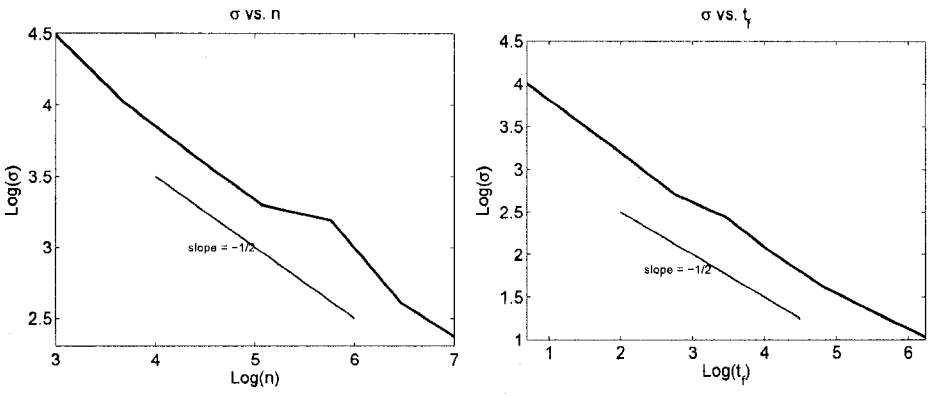

Figure 7.5: Figure shows results of increasing n and increasing the final time t_f. A line of slope -1/2 is shown in each case. Left: Relationship between n and σ. Right: Relationship between t_f and σ.

Chapter 8

FIRST RESULTS: CONTINUUM COUPLING

8.1 The Deterministic Problem

In this section, we analyze the problem in which we have continuous diffusion in two adjacent regions, with differing diffusion coefficients. We solve the problem by constructing a fixed point iteration that leads to the solution on the boundary. We first solve the Dirchlet to Neumann problem in one region, pass the Neumann boundary condition to the second region, then solve the Neumann to Dirichlet problem in the second region. We repeat this process, which creates a fixed point map. We want to find conditions on the problem so that this fixed point iteration converges.

8.1.1 The Fixed Point Iteration

Consider the deterministic problem in one dimension:

$$\begin{cases} u_1'' = 0, & x \in \Omega_1 = (0,1), \\ u_1(0), = \alpha, & \\ \begin{cases} a_1\, u_1'(1) = a_2\, u_2'(1), \\ u_1(1) = u_2(1), \end{cases} & \\ u_2'' = 0, & x \in \Omega_2 = (1,2), \\ u_2(2) = \beta. & \end{cases} \tag{8.1.1}$$

The analytic solution is two straight lines that meet at the interface $x = 1$, and the value at the interface is

$$u_1(1) = u_2(1) = \frac{r\alpha + \beta}{1 + r}, \tag{8.1.2}$$

where $r = \frac{a_1}{a_2}$ represents the ratio of diffusivity coefficients of regions Ω_1 and Ω_2.

Consider solving the system iteratively. That is, we find iterates $\{u_1^{(i)}\}_{i=0}^{\infty}$, $\{u_2^{(i)}\}_{i=0}^{\infty}$. Under desirable conditions, the solution to (8.1.1) is given by $\lim_{k\to\infty} u_1^{(k)}(x)$ and $\lim_{k\to\infty} u_2^{(k)}(x)$. We may also view this as a fixed point problem, in which we find a fixed point of the map $g : u_1(1) \to u_1(1)$. The iterates $u_1(1), u_2(1)$ are given by

$u_1^{(0)}(1) = u_2^{(0)}(1) = c_0$ (initial guess for solution at the interface),

For $i = 1, 2, ...$

$$\begin{aligned} (u_1^{(i)})'(1) &= u_1^{(i-1)}(1) - \alpha, \\ (u_2^{(i)})'(1) &= r(u_1^{(i)})'(1), \\ u_2^{(i)}(1) &= \beta - r\,(u_1^{(i)})'(1), \\ u_1^{(i)}(1) &= u_2^{(i)}(1). \end{aligned} \tag{8.1.3}$$

The first few iterates are then given as:

$$\begin{aligned} (u_1^{(0)})'(1) &= c_0 - \alpha, & u_2^{(0)}(1) &= \beta - r\,c_0 + r\alpha, \\ (u_1^{(1)})'(1) &= \beta - r\,c_0 + (r-1)\alpha, & u_2^{(1)}(1) &= \beta(1-r) + r^2 c_0 + (-r^2 + r)\alpha, \end{aligned}$$

and inductively,

$$u_2^{(k)}(1) = \beta \sum_{i=0}^{k-1} (-r)^i + (-r)^k c_0 - \alpha \sum_{i=1}^{k} (-r)^i. \tag{8.1.4}$$

Provided that $|r| < 1$, the limit of (8.1.4) as $k \to \infty$ agrees with the analytic solution (8.1.2).

8.1.2 Convergence Analysis for the Fixed Point Map

Consider the more general diffusion coupling problem with forcing,

$$\begin{cases} u_1'' = f_1, & x \in \Omega_1 = (0,1), \\ u(0) = \alpha, & \\ \begin{cases} r u_1'(1) = u_2'(1), \\ u_1(1) = u_2(1), \end{cases} & \\ u_2'' = f_2, & x \in \Omega_2 = (1,2), \\ u_2(2) = \beta, & \end{cases}$$

where $r = \frac{a_1}{a_2}$ represents the ratio of diffusivity coefficients of regions Ω_1 and Ω_2. We summarize the iterative technique:

Given initial guess $u_2(1) = c_0$, loop until convergence.

Given $u_1(1) = u_2(1)$, solve for u_1 and compute $a_1 u_1'(1)$. (8.1.5)

Given $a_2 u_2'(1) = a_1 u_1'(1)$, solve for u_2 and compute $u_2(1)$. (8.1.6)

Each iteration consists of solving two boundary value problems. Let g_2 denote the Neumann-to-Dirichlet problem, (8.1.6). Specifically, $g_2 : \mathbb{R} \to \mathbb{R}$, it is the map that takes values of the flux $a_2 u_2'(1)$ at the boundary and returns values of the concentration $u_2(1)$ at the boundary, for the problem

$$\begin{cases} u_2'' = f_2, & x \in \Omega_2 = (1,2), \\ u_2(2) = \beta, & \\ -a_2 u_2'(1) = h. & \end{cases}$$

We would like to compute the Frechét derivative $\frac{dg_2}{d(a_2 u_2'(1))}$. To do this, we consider the perturbed problem,

$$\begin{cases} w_2'' = f_2, & x \in \Omega_2 = (1,2), \\ w_2(2) = \beta, & \\ -a_2 w_2'(1) = h + \delta. & \end{cases}$$

We consider the change in $u_2(2)$, given the change $h \to h + \delta$. Let $\epsilon_2 = w_2 - u_2$, so that ϵ_2 solves the problem

$$\begin{cases} \epsilon_2'' = 0, & x \in \Omega_2 = (1,2), \\ \epsilon_2(2) = 0, \\ -a_2 w_2'(1) = \delta. \end{cases}$$

The Frechét derivative is expressed as

$$\frac{dg_2}{dh} = \lim_{\delta \to 0} \frac{w_2(1) - u_2(1)}{\delta} = \lim_{\delta \to 0} \frac{e_2(1)}{\delta}.$$

Then ϵ_2 is linear, so that the analytic solution is $\epsilon_2 = -\frac{\delta}{a_2}x + \frac{2\delta}{a_2}$. Hence, $\epsilon_2(1) = \frac{\delta}{a_2}$. Next, we compute the Frechét derivative,

$$\frac{dg_2}{dh} = \lim_{\delta \to 0} \frac{\epsilon_2(1)}{\delta} = \lim_{\delta \to 0} \frac{\delta}{a_2 \delta} = \frac{1}{a_2}. \tag{8.1.7}$$

We perform a similar computation for the Dirichlet-to-Neumann problem, (8.1.5). Call this map $g_1 : \mathbb{R} \to \mathbb{R}$, the map that takes values of $u_1(1)$ at the boundary and returns the flux $-a_1 u_1'(1)$, for the problem

$$\begin{cases} u_1'' = f_1, & x \in \Omega_1 = (0,1), \\ u_1(0) = \alpha, \\ u_1(1) = h. \end{cases}$$

Again, we consider the perturbed problem

$$\begin{cases} w_1'' = f_1, & x \in \Omega_1 = (0,1), \\ w_1(0) = \alpha, \\ w_1(1) = h + \delta. \end{cases}$$

Let $\epsilon_1 = w_1 - u_1$, so that ϵ_1 solves

$$\begin{cases} \epsilon_1'' = 0, & x \in \Omega_1 = (0,1), \\ \epsilon_1(0) = 0, \\ \epsilon_1(1) = \delta. \end{cases}$$

The Frechét derivative is

$$\frac{dg_1}{dh} = \lim_{\delta \to 0} \frac{-a_1 w_1'(1) + a_1 u_1'(1)}{\delta} = \lim_{\delta \to 0} \frac{-a_1 \epsilon_1'(1)}{\delta}.$$

Provided that a_1 is constant, the analytic solution is $\epsilon_1(x) = \delta x$, so that $\epsilon'(1) = \delta$. Then the Frechét derivative becomes

$$\lim_{\delta \to 0} \frac{-a_1 \epsilon_1'(1)}{\delta} = \lim_{\delta \to 0} \frac{-a_1 \delta}{\delta} = -a_1. \tag{8.1.8}$$

Next, define the map g to be the composition $g = g_2 \circ g_1$, the map that takes values $u_1^{(i)}(1)$ and returns the next value in the iteration $u_1^{(i+1)}(1)$. Then, the Frechét derivative of g is

$$\frac{dg}{du_1(1)} = \frac{dg_2}{d(a_2 u_2'(1))} \frac{dg_1}{du_1(1)} = -\frac{a_1}{a_2}.$$

By Theorem 2.2.2, the fixed point iteration converges provided that

$$\left| \frac{a_1}{a_2} \right| < 1. \tag{8.1.9}$$

In Fig. 8.1, we see evidence of (8.1.9).

8.1.3 Applying Relaxation to the Fixed Point Problem

Suppose that instead of computing $u_1^{(i)}(1) = u_2^{(i)}(1)$ at each iteration, we take a weighted average of the Dirichlet value given from the continuum problem with the Dirichlet value at the previous iteration, $u_1^{(i-1)}(1)$. That is, during each iteration we limit the amount that the boundary condition can change. Upon obtaining $u_2(1)$, we compute $u_1(1)$ using the map g_3 : $(u^{(i)}(1), u^{(i-1)}(1)) \to u^{(i)}(1)$, defined by

$$g_3(u_1^{(i)}(1), u_1^{(i-1)}(1)) = \lambda u_1^{(i-1)}(1) + (1 - \lambda) u_1^{(i)}(1), \tag{8.1.10}$$

where the relaxation parameter $\lambda \in [0, 1)$. The map corresponding to one fixed point iteration is now given by $g = g_3 \circ g_2 \circ g_1$. To compute the spectrum, we need the Frechét derivatives,

$$\frac{\partial g_3}{\partial u_1^{(i-1)}(1)} = \lambda, \qquad \frac{\partial g_3}{\partial u_1^{(i)}(1)} = (1 - \lambda).$$

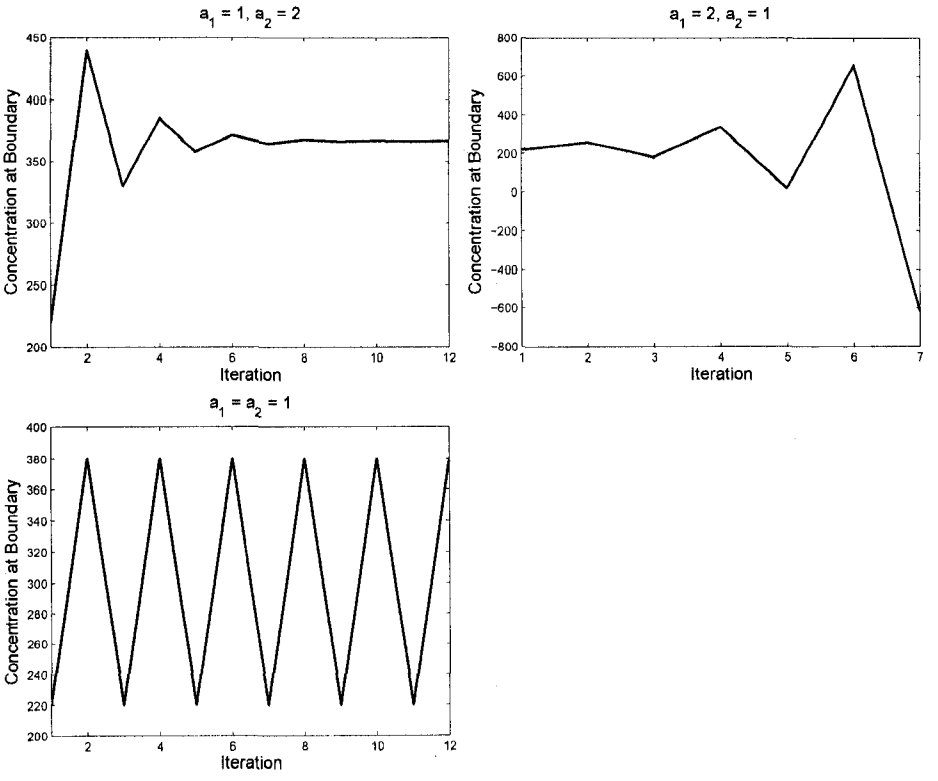

Figure 8.1: Concentration at the interface as a function of iteration, $r = 1/2, 2, 1$.

To determine the effect that changing $u_1^{(i-1)}(1)$ has on $u_1^{(i)}(1)$, we apply the chain rule to get

$$\frac{\partial g}{\partial u_1(1)} = \frac{\partial g_3}{\partial u_1(1)} + \frac{\partial g_1}{\partial u_1(1)} \frac{\partial g_2}{\partial (a_2 u_2'(1))} \frac{\partial g_3}{\partial u_2(1)} = \lambda - r(1-\lambda).$$

The fixed point iteration will converge provided that $|\lambda - r(1-\lambda)| < 1$, or

$$\frac{r-1}{r+1} < \lambda < 1. \tag{8.1.11}$$

If we consider the optimal parameter λ to be the value that minimizes $|\lambda - r(1-\lambda)|$, we see that the optimal value is given by

$$\lambda = r/(1+r). \tag{8.1.12}$$

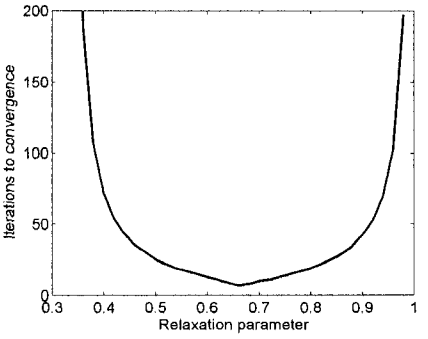

Figure 8.2: Number of iterations to convergence for varying relaxation parameters.

Example 8.1.1. *We solve (8.1.1) using* $\alpha = 100, \beta = 500, u_1^{(0)}(1) = 220, r = 2$, *for varying parameters* λ. *We iterate until the stopping criterion of* $|u_1^{(i)}(1) - u_1^{(i-1)}(1)| < 10^{-5}$ *has been reached. Following (8.1.11) and (8.1.12), the iteration converges provided that* $\frac{1}{3} < \lambda < 1$, *with optimal parameter* $\frac{2}{3}$. *Evidence of this is shown in Fig. 8.2.*

8.1.4 Reversing the Coupling

In the event that (8.1.9) is not satisfied, we reverse the direction of the coupling to guarantee convergence. We find iterates $\{u_1^{(i)}\}_{i=1}^{\infty}$, $\{u_2^{(i)}\}_{i=1}^{\infty}$ given by the following algorithm.

$(u_1^{(0)})'(1) = d_0$, (initial guess for gradient at the interface)

For $i = 1, 2, ...$

$$\begin{aligned} u_1^{(i)}(1) &= (u_1^{(i-1)})'(1) + \alpha, \\ u_2^{(i)}(1) &= u_1^{(i)}(1), \\ (u_2^{(i)})'(1) &= \beta - u_2^{(i)}(1), \\ (u_1^{(i)})'(1) &= \frac{a_2}{a_1}(u_2^{(i)})'(1). \end{aligned} \tag{8.1.13}$$

Following the argument as in section 8.1.2, the fixed point iteration (8.1.13) converges provided that $|a_2/a_1| < 1$.

8.2 A Green's Function Approach to Convergence Analysis

For a general n-dimensional problem, we use the generalized Green's function ϕ to find a formula for the derivative of the Dirichlet-to-Neumann map g_1. We similarly find the derivative of the Neumann-to-Dirichlet map g_2 in terms of ϕ. The problem remains as to how to obtain the derivative of $g = g_2 \circ g_1$, since this technique only gives a linear functional of the Neumann or Dirichlet data.

8.2.1 The Dirichlet-to-Neumann Derivative

The Dirichlet-to-Neumann problem is

$$\begin{cases} -\nabla \cdot (a\nabla u) = f, & x \in \Omega, \\ u = \lambda(x), & x \in \Gamma \subset \partial\Omega, \\ u = \alpha(x), & x \in \partial\Omega \setminus \Gamma. \end{cases}$$

The variational formulation is to find $u \in H^1(\Omega)$ such that $u = \lambda$ for $x \in \Gamma$, $u = \alpha$ for $x \in \partial\Omega \setminus \Gamma$, and $(a\nabla u, \nabla v) = (f, v)$ for all $v \in H_0^1(\Omega)$. The quantity of interest is

$$q(\lambda) = -(a\partial_\eta u, \psi)_\Gamma,$$

a functional of the flux on the common boundary Γ. Following the method described in [51], the adjoint problem is

$$\begin{cases} -\nabla \cdot (a\nabla \phi) = 0, & x \in \Omega, \\ \phi = \psi, & x \in \Gamma, \\ \phi = 0, & x \in \partial\Omega \setminus \Gamma. \end{cases}$$

Note that this problem satisfies

$$\begin{aligned}(f,\phi) &= (-\nabla\cdot(a\nabla u),\phi)\\ &= (a\nabla u,\nabla\phi)-(a\partial_\eta u,\phi)_\Gamma-(a\partial_\eta u,\phi)_{\partial\Omega\setminus\Gamma}\\ &= (a\nabla u,\nabla\phi)-(a\partial_\eta u,\psi)_\Gamma\\ &= (-\nabla\cdot(a\nabla\phi),u)+(a\partial_\eta\phi,u)_\Gamma+(a\partial_\eta\phi,u)_{\partial\Omega\setminus\Gamma}-(a\partial_\eta u,\psi)_\Gamma\\ &= (\partial_\eta\phi,\lambda)_\Gamma+(a\partial_\eta\phi,\alpha)_{a\partial\Omega\setminus\Gamma}-(a\partial_\eta u,\psi)_\Gamma.\end{aligned}$$

Hence, the quantity of interest is written as

$$q(\lambda)=-(a\partial_\eta u,\psi)_\Gamma=(f,\phi)-(a\partial_\eta\phi,\lambda)_\Gamma-(a\partial_\eta\phi,\alpha)_{\partial\Omega\setminus\Gamma}.$$

To compute the Frechét derivative, consider the perturbed problem,

$$\begin{cases}-\nabla\cdot(a\nabla w),=f & x\in\Omega,\\ w=\lambda(x)+\delta(x), & x\in\Gamma\subset\partial\Omega,\\ w=\alpha(x), & x\in\partial\Omega\setminus\Gamma.\end{cases}$$

Then, we compute the difference $q(\lambda+\delta)-q(\lambda)$,

$$\begin{aligned}q(\lambda+\delta)-q(\lambda) &= -(a\partial_\eta w,\psi)_\Gamma+(a\partial_\eta u,\psi)_\Gamma\\ &= (f,\phi)-(a\partial_\eta\phi,\lambda+\delta)_\Gamma-(a\partial_\eta\phi,\alpha)_{\partial\Omega\setminus\Gamma}\\ &\quad -(f,\phi)+(a\partial_\eta\phi,\lambda)_\Gamma+(a\partial_\eta\phi,\alpha)_{\partial\Omega\setminus\Gamma}\\ &= -(a\partial_\eta\phi,\delta)_\Gamma.\end{aligned}$$

The Frechét derivative in the direction $\delta(x)$ is

$$\nabla_\lambda q(\lambda)\cdot\delta=-(a\partial_\eta\phi,\delta)_\Gamma. \tag{8.2.1}$$

To verify this,

$$\frac{\|q(\lambda+\delta)-q(\lambda)-\nabla_\lambda q(\lambda)\cdot\delta\|}{\|\delta\|}=\frac{\|-(a\partial_\eta\phi,\delta)_\Gamma+(a\partial_\eta\phi,\delta)_\Gamma\|}{\|\delta\|}=0.$$

8.2.2 The Neumann-to-Dirichlet Derivative

The Neumann-to-Dirichlet problem is stated as

$$\begin{cases} -\nabla \cdot (a\nabla u) = f, & x \in \Omega, \\ -a\partial_\eta u = \lambda(x), & x \in \Gamma \subset \partial\Omega, \\ u = \beta(x), & x \in \partial\Omega \setminus \Gamma. \end{cases}$$

The quantity of interest is

$$q(\lambda) = (u, \psi)_\Gamma,$$

a functional of the solution on the boundary Γ. The adjoint problem is

$$\begin{cases} -\nabla \cdot (a\nabla \phi) = 0, & x \in \Omega, \\ -a\partial_\eta \phi = \psi(x), & x \in \Gamma \subset \partial\Omega, \\ \phi = 0, & x \in \partial\Omega \setminus \Gamma. \end{cases} \tag{8.2.2}$$

We note that

$$\begin{aligned} (f, \phi) &= (-\nabla \cdot (a\nabla u), \phi) \\ &= (a\nabla u, \nabla\phi) - (a\partial_\eta u, \phi)_\Gamma - (a\partial_\eta u, \phi)_{\partial\Omega\setminus\Gamma} \\ &= (a\nabla u, \nabla\phi) + (\lambda, \phi)_\Gamma \\ &= (-\nabla \cdot (a\nabla\phi), u) + (a\partial_\eta\phi, u)_\Gamma + (a\partial_\eta\phi, u)_{\partial\Omega\setminus\Gamma} + (\lambda, \phi)_\Gamma \\ &= -(u, \psi)_\Gamma + (a\partial_\eta\phi, \beta)_{\partial\Omega\setminus\Gamma} + (\lambda, \phi)_\Gamma. \end{aligned}$$

After rearranging terms,

$$q(\lambda) = (u, \psi)_\Gamma = -(f, \phi) + (a\partial_\eta\phi, \beta)_{\partial\Omega\setminus\Gamma} + (\lambda, \phi)_\Gamma. \tag{8.2.3}$$

In order to compute the Frechét Derivative, we perturb the flux at the common boundary Γ to obtain the problem

$$\begin{cases} -\nabla \cdot (a\nabla w), = f & x \in \Omega, \\ -a\partial_\eta w = \lambda(x) + \delta(x), & x \in \Gamma \subset \partial\Omega, \\ w = \beta(x), & x \in \partial\Omega \setminus \Gamma. \end{cases}$$

Then,

$$\begin{aligned} q(\lambda+\delta)-q(\lambda) &= -(f,\phi)+(a\partial_\eta\phi,\beta)_{\partial\Omega\setminus\Gamma}+(\lambda+\delta,\phi)_\Gamma \\ &\quad +(f,\phi)-(a\partial_\eta,\beta)_{\partial\Omega\setminus\Gamma}-(\lambda,\phi)_\Gamma \\ &= (\delta,\phi)_\Gamma. \end{aligned}$$

The Frechét Derivative in the direction $\delta(x)$ is

$$\nabla_\lambda q(\lambda)\cdot\delta=(\delta,\phi)_\Gamma. \tag{8.2.4}$$

To give an example, we verify that (8.2.1) and (8.2.4) agree with the formulas (8.1.8) and (8.1.7) in one dimension. For the Dirichlet-to-Neumann case, we have the problem

$$\begin{cases} -a_1\,u''=f, & x\in(0,1),\\ u(0)=\alpha,\\ u(1)=\lambda, \end{cases}$$

and the adjoint problem,

$$\begin{cases} -a_1\,\phi''=0, & x\in(0,1),\\ \phi(0)=0,\\ \phi(1)=\psi. \end{cases} \tag{8.2.5}$$

If $\psi=1$, the quantity of interest is $q(\lambda)=u(1)$. According to (8.2.1), the derivative is given by $q'(\lambda)=-a_1\phi'(1)$. If we solve the adjoint equation (8.2.5), we get $\phi'(1)=1$. The derivative is then $q'(\lambda)=-a_1$, in agreement with (8.1.8).

In the Neumann-to-Dirichlet case, the problem is

$$\begin{cases} -a_2\,u''=f, & x\in(1,2),\\ -a_2u'(1)=\lambda,\\ u(2)=\beta, \end{cases}$$

and the adjoint problem,

$$\begin{cases} -a_2\,\phi'' = 0, & x \in (0,1), \\ -a_2\phi'(1) = \psi, \\ \phi(2) = 0. \end{cases} \tag{8.2.6}$$

Assuming $\psi = 1$, the quantity of interest is $q(\lambda) = u(1)$. By (8.2.4), we have $q'(\lambda) = \phi(1)$. The solution to the adjoint equation (8.2.6) gives us $\phi(1) = 1/a_2$. The derivative becomes $q'(\lambda) = 1/a_2$, in agreement with (8.1.7).

8.3 Adding Randomness

We discuss ways of adding stochastic behavior to (8.1.1), and explain the effect that a random boundary condition, or a random forcing function has on the fixed point iteration. We also give an application of the Green's function approach to the Neumann-to-Dirichlet map discussed above.

8.3.1 Randomness at the Interface

Consider the problem (8.1.1), where at each iteration we add a random variable with density function $N(\mu, \sigma^2)$ to the interface. This models the situation where the values at the interface are obtained from a stochastic technique. This is certainly the case if the value at the interface comes from results from an atomistic simulation in an adjacent region, as we explore in Chapter 9. In particular, suppose that $(u_1^{(i)})'(1) = u_1^{(i)}(1) - \alpha + \lambda_i$, where $\lambda_i \sim N(\mu, \sigma^2)$. We define the sequence $\{u_2^{(k)}(1)\}_{k=0}^{\infty}$ analogously to (8.1.3), and it turns out that

$$u_2^{(k)}(1) = \beta \sum_{i=0}^{k-1} (-r)^i + (-r)^k c_0 - \alpha \sum_{i=1}^{k} (-r)^i + \sum_{i=1}^{k} (-r)^i \lambda_i.$$

As expectation is linear,

$$E\left[\sum_{i=1}^{k}(-r)^i\lambda_i\right] = \sum_{i=1}^{k}(-r)^i\mu.$$

Since λ_i's are independent,

$$\text{var}\left(\sum_{i=1}^{k}(-r)^i\lambda_i\right) = \sum_{i=1}^{k}(-r)^{2i}\text{var}(\lambda_i),$$

so that $\sum_{i=1}^{k}(-r)^i\lambda_i \sim N\left(\sum_{i=1}^{k}(-r)^i\mu, \sum_{i=1}^{k+1} r^{2i}\sigma^2\right)$. Provided that $|r| < 1$, we let $k \to \infty$ to obtain

$$u_1(1) = u_2(1) = \frac{r\alpha+\beta}{1+r} + \lambda^*, \text{ where } \lambda^* \sim N\left(\frac{-r}{1+r}\mu, \frac{r^2}{1-r^2}\sigma^2\right).$$

We note that the variance increases without bound as r approaches 1.

8.3.2 Random Forcing

The preceding argument includes adding randomness to the boundary condition used in the coupling. A different problem involves adding a random forcing function on one of the sub-domains, i.e.,

$$\begin{cases} -u_1'' = f(\lambda), & x \in \Omega_1 = (0,1), \\ u_1(0) = \alpha, & \\ \begin{cases} r\,u_1'(1) = u_2'(1), \\ u_1(1) = u_2(1), \end{cases} & \\ u_2'' = 0, & x \in \Omega_2 = (1,2), \\ u_2(2) = \beta, & \end{cases} \tag{8.3.1}$$

where λ is some random variable.

For example, in the case where $f(\lambda) = \lambda$, the analytic solution to the problem in Ω_1 is $u_1(x) = -\frac{1}{2}\lambda x^2 + (u_1(1) + \frac{1}{2}\lambda - \alpha)x + \alpha$, and thus $u_1'(1) = u_1(1) - \frac{1}{2}\lambda - \alpha$. To iterate, we use the analytic solutions at each

iteration,

$u_1^{(0)}(1) = u_2^{(0)}(1) = c_0$ (initial guess for solution at the interface),

for $i = 1, 2, ...$

$$(u_1^{(i)})'(1) = u_1^{(i-1)}(1) - \frac{1}{2}\lambda - \alpha,$$

$$(u_2^{(i)})'(1) = r(u_1^{(i)})'(1),$$

$$u_2^{(i)}(1) = \beta - r\,(u_2^{(i)})'(1),$$

$$u_1^{(i)}(1) = u_2^{(i)}(1).$$

The result, considering that λ is constant, is

$$u_2^{(k)}(1) = \beta\sum_{i=0}^{k-1}(-r)^i + (-r)^k c_0 - \alpha\sum_{i=1}^{k}(-r)^i - \frac{\lambda}{2}\sum_{i=1}^{k}(-r)^i.$$

Taking the limit $k \to \infty$, we obtain $u_2(1) = \frac{\beta + r\alpha + \frac{1}{2}r\lambda}{1+r}$, provided that $|r| < 1$.

On the other hand, if λ is a random variable with density $N(\mu, \sigma^2)$, then

$$u_2^{(k)}(1) = \beta\sum_{i=0}^{k-1}(-r)^i + (-r)^k c_0 - \alpha\sum_{i=1}^{k}(-r)^i - \frac{1}{2}\sum_{i=1}^{k}(-r)^i\lambda_i,$$

where $\lambda_i \sim N(\mu, \sigma^2)$. Then, using a similar argument as described in section 8.3.1,

$$-\frac{1}{2}\sum_{i=1}^{k}(-r)^i\lambda_i \sim N\left(-\frac{1}{2}\sum_{i=1}^{k}(-r)^i\mu, \frac{1}{4}\sum_{i=1}^{k}r^{2i}\sigma^2\right).$$

Assuming $|r| < 1$, we take the limit $k \to \infty$, to get

$$u_2(1) = \frac{\beta + r\alpha}{1+r} + \lambda^*, \text{ where } \lambda^* \sim N\left(\frac{1}{2}\frac{r}{1+r}\mu, \frac{1}{4}\frac{r^2}{1-r^2}\sigma^2\right). \qquad (8.3.2)$$

We experimentally find that the sample mean and variance of $u_2^{(k)}(1)$ approach that of (8.3.2) as $k \to \infty$ for a variety of $r, \mu, \sigma, \alpha, \beta$.

8.3.3 The Green's Function Representation

Consider the continuum problem,

$$\begin{cases} -\nabla \cdot (a_2 \nabla u_2) = f, & x \in \Omega, \\ a_2 \delta_\eta u_2 = \lambda(x), & x \in \Gamma, \\ u_2 = \beta(x), & x \in \partial\Omega \setminus \Gamma, \end{cases} \tag{8.3.3}$$

where $\lambda(x)$ is a random vector representing the flux on Γ. This may model the case in which diffusion is coupled with a stochastic model through the boundary Γ. Assume that we have a distribution for λ and wish to sample from this distribution to compute a quantity of interest, $q(\lambda) = (u, \psi)_\Gamma$. The quantity of interest is a linear functional describing the concentration on Γ. Obtaining several samples of $q(\lambda)$ might involve solving (8.3.3) repeatedly for each realization of λ. Alternatively, we may use the Green's function representation of the Neumann to Dirichlet map to speed up the sampling process. Assuming we have obtained the solution ϕ to the dual problem (8.2.2), we recall (8.2.3) to get

$$q(\lambda) = (u_2, \psi)_\Gamma = -(f, \phi) + (a_2 \partial_\eta \phi, \beta)_{\partial\Omega\setminus\Gamma} + (\lambda, \phi)_\Gamma.$$

To summarize, if we have a distribution of λ, we easily compute several values of $q(\lambda)$ by solving only one adjoint problem. We may also use the methods described in [19] to increase the efficiency of the sampling process.

Chapter 9

FIRST RESULTS: ATOMISTIC TO CONTINUUM COUPLING

9.1 Overview

The coupling of atomistic and continuum models has emerged as a critical component in computational materials science. For a multiscale process, the finite element method may fail to describe processes occurring on a small time or spatial scale. For example, a material could have a crack or deformation that occurs on an extremely small scale, yet still affects the macroscopic behavior of the problem. In a crack tip region, the laws of linear elasticity fail to hold and we need an atomistic simulation to model the problem. However, fully atomistic simulations of many domains are computationally infeasible. The atomistic-to-continuum methods address this problem by performing a continuum calculation on a majority of the domain, and an expensive atomistic simulation on the subset of the domain where the atomistic detail is needed. For example, Rudd and Broughton [42] discuss multiscale computations for a rotary nano-engine with a diameter of 30nm. The engine drives Salmonella and E. coli bacteria by rotating close to 20,000 rpm while using very little energy. The engine is too small to obey continuum principles, as a continuum model fails to describe surface effects

[24]. However, a fully atomistic model results in billions of unknowns. Some of the popular atomistic-to-continuum methods include quasi-continuum [44], Finite Element-Atomistic (FEAt) [29], and Coarse Grained Molecular Dynamics [42].

The primary issue in this chapter is to give a mathematically sound framework for coupling stochastic and deterministic descriptions of nature. These two kinds of descriptions apply at different scales and describe different phenomena that nevertheless interact closely. Stochastic descriptions often apply at very fine scales, describing the detailed motion and interaction of the smallest building blocks, e.g., molecules or bacteria. Deterministic models usually apply at larger scales and describe the aggregate behavior, e.g., fluid flow.

We note that error is also a difficult issue for stochastic-deterministic coupling. For a stochastic simulation, there is a statistical or probabilistic description of error and uncertainty at each step. On the other hand, a deterministic simulation is affected by deterministic error, such as discretization error. Hence, the description of uncertainty and error between the two types of simulation is fundamentally different.

In this chapter, we formulate coupling between stochastic and deterministic models as an iterative process, and begin to study the convergence analysis for the iterative process. In a stochastic simulation, the quantities of interest are random variables computed as statistical averages, generated from a finite number of realizations of the simulation. On the other hand, a deterministic problem produces one solution for each data set, and when the data input is random, the output is another random variable or field. Hence, in a coupled system the information that is passed back and forth

are random variables. In the case of feedback between the components we have to describe a framework of iteration that converges to the solution of the complete system.

In [1], the authors formulate a hybrid particle/continuum algorithm to solve a diffusion problem, where the particles follow a random walk on a lattice. The domains for the two models overlap, and the flux is matched in the common region. Motivated by [1], we pose a problem that couples a Brownian motion model and a continuum model, with non-overlapping domains. This idealized model has Brownian motion on a lattice next to a material treated as a continuum. We make an initial guess for the concentration at the boundary, then solve the Brownian motion model. As this is a stochastic model, we simulate many realizations, then obtain an approximate distribution for the solution value on the interface. We sample from this distribution and pass these values into the continuum region as boundary conditions. This, in turn, provides another distribution which is passed back into the Brownian motion region. This process is repeated until some convergence criteria is met.

The Brownian motion region yields random variables that are associated with probability densities as the computed quantities. The accuracy is improved by taking more realizations. In this case, the continuum model describes the expected value of particle behavior, but does not model individual particles.

As described, the process gives a sequence of distributions. We therefore should discuss convergence of this sequence. We view this problem as a fixed point problem for the probability distribution on the interface. We addressed convergence for the analogous fixed point iteration for coupled

continuum models in Chapter 8. For the continuum-continuum problem, we may obtain the derivative of the iterative map G, and show under certain conditions that we have $|G'| \leq L < 1$. Let G_δ denote the map of the atomistic to continuum model, where we conjecture that $G_\delta \rightarrow G$ as $\delta \rightarrow 0$, and so under appropriate conditions, $|G'_\delta| \leq L + \epsilon < 1$. In other words, since the map G_δ approximates the continuum map G, we conjecture that it inherits the convergence properties of the continuum map.

Another important question that arises is whether we can interchange the limits of the number of samples approaching infinity, with the number of fixed point iterations approaching infinity. This is a practical issue. If the continuous region can be solved relatively cheaply, then it may be more cost effective to run several fixed point iterations before increasing the number of samples. However, if the continuous region involves an expensive calculation then we may be better off with a large number of samples for each fixed point iteration.

Another interesting subject is numerical error. The accuracy for the continuum model is controlled by numerical error, and by sampling from the distribution at the interface. We would like to control the numerical errors so it does not significantly bias the densities computed.

9.2 Coupling a Random Walk with a Continuum Model

Consider diffusion on two adjacent regions Ω_1 and Ω_2, with diffusivity a_1 and a_2, respectively:

$$\begin{cases} a_1 u_1'' = 0, & x \in \Omega_1 = [a, b], \\ u_1(a) = \alpha, & \\ \begin{cases} a_1 u_1'(b) = a_2 u_2'(b), \\ u_1(b) = u_2(b), \end{cases} & \\ a_2 u_2'' = 0, & x \in \Omega_2 = [b, c], \\ u_2(c) = \beta. & \end{cases} \tag{9.2.1}$$

We simulate the Brownian motion simulation in Ω_1 and pass the results to the continuum problem in Ω_2. Since we are considering a stationary state problem, we run the atomistic simulation for a sufficiently long time in order to eliminate the initial transient behavior.

Assume the Brownian motion region $[a, b]$ is computed on a lattice with grid values $a = x_0 < x_1 < ... < x_m = b$, where $x_i - x_{i-1} = \Delta x$. We need a way to convert between concentration and the number of particles at a point. Let $c(x)$ be the concentration per unit length at point x, and $n(x)$ be the number of particles at grid point x. Given the concentration $c(x_i)$ at a grid point x_i, we need to approximate $n(x_i)$, the number of particles at x_i. We assume the particles are evenly distributed in a region of width Δx about each x_i. We denote this region as $I_i = [x_i - \Delta x, x_i + \Delta x)$ for $i = 0, 1, ..., m$. Since the concentration is defined at all points in $[a, b]$ including the endpoints, we effectively simulate Brownian motion in the region $[a - \Delta x, b + \Delta x)$. We write this interval as

$$[a - \Delta x, b + \Delta x) = \bigcup_{i=0}^{m} I_i.$$

The particles in each interval I_i have concentration $c(x_i)$. The number of particles at the grid point is therefore approximated by $c(x_i)$ multiplied by

the interval length Δx. Hence, for $i = 0, 1, ..., m$,

$$\begin{aligned} n(x_i) &= \text{round}(c(x_i)\Delta x) \\ c(x_i) &= n(x_i)/\Delta x. \end{aligned}$$

We simulate the atomistic simulation in Ω_1 first, using $u_1(b) = c_0$ as an initial guess for the concentration at the midpoint. We then pass the flux $a_1 u'(b)$ as a boundary condition into Ω_2. Suppose we run the simulation up to a final time of t_f, so that $0 \leq t \leq t_f$. Define the rate r to be the number of steps per time unit. The total number of random walk steps to take is then

$$T = rt_f.$$

When we set the initial configuration of particles, we enforce the boundary conditions by requiring that the number of particles at x_0 and x_m be

$$n(x_0) = \text{round}(c(x_0)\Delta x), \quad n(x_m) = \text{round}(c(x_m)\Delta x). \tag{9.2.2}$$

Now for each of the T time steps, we move each particle to the right or left with probabilities p and $1 - p$, respectively. If a particle falls outside of the range $[a, b]$, we remove it from the simulation. After we have moved every particle, we again enforce the boundary conditions (9.2.2).

One way to approximate the gradient at the interface b is to compute the difference in particles between the grid points x_m and $x_m - \Delta x$ at the final time. That is,

$$u_1'(b) \approx \frac{n(x_m) - n(x_{m-1})}{\Delta x} \cdot \frac{1}{\Delta x}. \tag{9.2.3}$$

We may then follow the approach described in section 7.3.3, where we view one "realization" of a gradient as the ensemble average of the N gradients,

$$u_1'(b) \approx \frac{1}{N} \sum_{i=1}^{N} \frac{n^{(i)}(x_m) - n^{(i)}(x_{m-1})}{\Delta x^2}. \tag{9.2.4}$$

Increasing N has the effect of decreasing the error in the gradient. We repeat this process M times, to get a distribution of the sample means. We note that this means the random walk is simulated a total of NM times. To clarify, increasing N reduces the variance of the distribution, while increasing M has a smoothing effect on the histogram of the sample means.

Alternatively, we may opt to compute (9.2.3) at each time step $t \in (t_b, t_f)$, where t_b is some appropriate "burn time", at which the initial transient behavior has stabilized. We then compute the realization of the gradient as the average over the T time steps,

$$u_1'(b) \approx \frac{1}{T} \sum_{i=1}^{T} \frac{n^{(i)}(x_m) - n^{(i)}(x_{m-1})}{\Delta x^2}.$$

If ergodicity holds, this time average is equivalent to taking the statistical ensemble as described above.

We describe a strategy for passing the flux into the continuum region Ω_2. We run the simulations to produce several realizations of (9.2.4), and obtain a distribution of gradients. After converting the gradients to obtain a distribution of fluxes, we use a kernel density estimator to approximate an approximate density function for the flux, $\hat{f}$. Using rejection sampling, we then take several samples $z^{(k)} \sim \hat{f}, k = 1, ..., S$, to obtain a distribution of boundary conditions for the continuum region Ω_2. The continuum problem is solved using a finite element method. Since the flux on the interface is a random variable, we have introduced data error into the continuum problem. We solve the continuum problem for each $z^{(k)}$ to compute a distribution for $u_2(b)$. We pass a distribution of Dirichlet boundary conditions into the Brownian motion region in an analogous manner. We repeat the process of passing distributions between the regions.

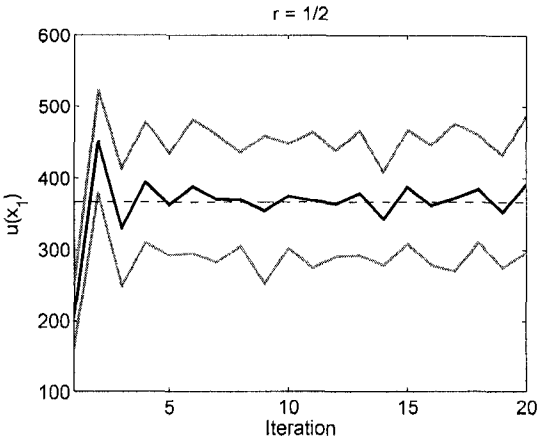

Figure 9.1: Concentration at the boundary at each iteration for $a_1/a_2 = 1/2$. The black line indicates the mean concentration and the grey lines indicate three standard deviations above and below the mean. The dashed line indicates the analytic solution.

In Fig. 9.1, we see that during the first few iterations, the mean value of the iterates approaches the analytic solution. Subsequently, however, the mean value does not improve but oscillates about the analytic solution due to the presence of random noise from the atomistic simulation.

To reduce the variance of the distribution of concentrations, we increase the number of samples N. In Fig. 9.2, we see the effect that doubling N at each fixed point iteration has on the variance. Here, we assume that it is valid to interchange increasing the number of samples with increasing the number of fixed point iterations.

9.2.1 An Example with Nonzero Advection

Suppose the probability, p, of a particle moving to the right, is not equal to the probability, q, of the particle moving to the left. Then by (7.3.5), the corresponding coupling problem has a nonzero advection term

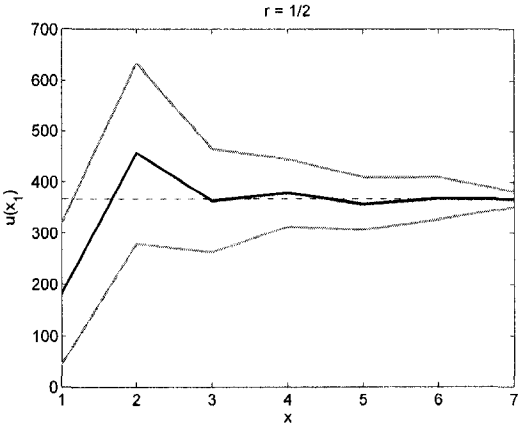

Figure 9.2: The black line indicates the mean concentration and the grey lines indicate three standard deviations above and below the mean. The dashed line indicates the analytic solution. At each iteration, the number of samples is doubled.

with coefficient c,

$$\begin{cases} \frac{1}{2}a_1\, u_1'' - c\, u_1' = 0, & x \in \Omega_1 = (0,1), \\ u_1(0) = \alpha, \\ \begin{cases} a_1\, u_1'(1) = a_2\, u_2'(1), \\ u_1(1) = u_2(1), \end{cases} \\ \frac{1}{2}a_2\, u_2'' - c\, u_2' = 0, & x \in \Omega_2 = (1,2), \\ u_2(2) = \beta, \end{cases} \tag{9.2.5}$$

where $c = (p-q)\frac{\delta}{\tau}$. The solution to the problem

$$\begin{cases} \frac{1}{2}a_2\, u_2'' - c\, u_2 = 0, & x \in (1,2), \\ u_2'(1) = m, \\ u_2(2) = \beta, \end{cases}$$

is

$$u_2(x) = \frac{m}{k}e^{k(x-1)} + \beta - \frac{m}{k}e^{k},$$

where $k = \frac{2c}{a_2}$. Next, we give the analytic solution to the problem (9.2.5). Let $k_1 = \frac{2c}{a_1}$, $k_2 = \frac{2c}{a_2}$, and $r = \frac{a_1}{a_2}$. Then,

$$u_1(x) = Ae^{k_1 x} + \alpha - A, \quad \text{and} \quad u_2(x) = Ce^{k_2 t} + \beta - Ce^{2k_2},$$

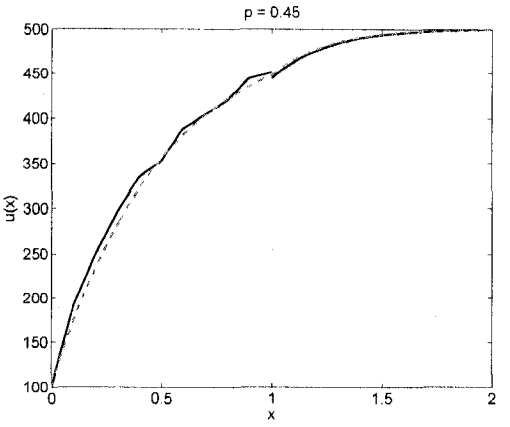

Figure 9.3: Solution to problem (9.2.5), for $\alpha = 100, \beta = 500, p = 0.45, r = 0.5$. The dashed line shows the analytic solution, and the solid line shows the approximate solution for the iterative scheme with Brownian motion in $(0, 1)$, and continuous diffusion in $(1, 2)$.

where

$$A = \frac{C(e^{k_2} - e^{2k_2}) + \beta - \alpha}{e^{k_1} - 1}, \quad \text{and} \quad C = \frac{\alpha - \beta}{e^{k_2} - e^{2k_2} - \frac{k_2 e^{k_2}(e^{k_1} - 1)}{r k_1 e^{k_1}}}.$$

A plot of a solution to (9.2.5) is shown in Fig. 9.3.

9.2.2 Applying Relaxation to the Iterative Map

In section 8.1.3, we describe a method to apply relaxation to the fixed point iteration for the deterministic case. Since the Brownian motion model approaches the deterministic model, we hope that relaxation may provide similar results. In Fig. 9.4, we see that by using the optimal parameter given by (8.1.12), the oscillation in concentrations is dampened. In Fig. 9.5, we attempt to solve the problem with $a_1/a_2 = 2$. We see that the distribution diverges without relaxation, but converges using relaxation.

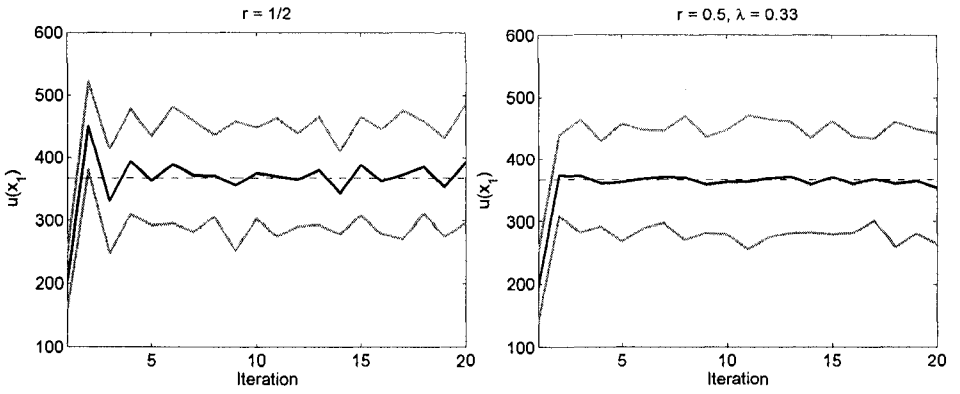

Figure 9.4: Concentration at the boundary at each iteration for $a_1/a_2 = 2$. The black line indicates the mean concentration and the grey lines indicate three standard deviations above and below the mean. The dashed line indicates the analytic solution. Left: no relaxation. Right: $\lambda = 0.33$.

9.2.3 Reversing the Direction of the Coupling

We investigate the effects and benefits of reversing the coupling, that is, we pass the Dirichlet condition to the continuum region and enforce a Neumann condition at the interface. The result of a particle simulation is shown in Fig. 9.6. In the continuous case discussed in section 8.1.4, we address the case of the unstable ratio of diffusivities $|a_1/a_2| > 1$ by reversing the direction of the coupling. We find this works well for the case of stochastic-deterministic coupling as well. We solve the Neumann to Dirichlet problem in the Brownian motion region and solve the Dirichlet to Neumann problem in the continuum region. An example is shown in Fig. 9.7.

9.3 A Probability Transition Matrix Approach

In this section, we use an alternative to the fixed point formulation described in section 9.2. We examine a single step of the stochastic process, which is assumed to be a Markov process. If we can obtain a probability

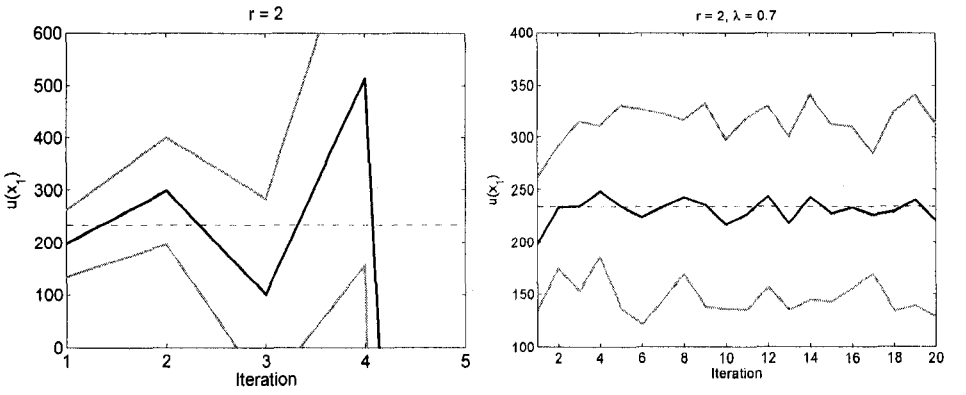

Figure 9.5: Concentration at the boundary at each iteration for $a_1/a_2 = 2$. The black line indicates the mean concentration and the grey lines indicate three standard deviations above and below the mean. The dashed line indicates the analytic solution. Left: no relaxation. Right: $\lambda = 0.7$.

transition matrix for the process and verify that the Markov properties hold, then we may be able to find a limiting distribution for the Markov chain. Under desirable conditions, the limiting distribution obtained from this approach will coincide with the distribution obtained using the fixed point iteration. Consider the following algorithm.

1. Given a number of particles at b, $n(b)^{(k)}$, run the atomic simulation in Ω_1 and compute a single realization of the flux $z = a_1 u'(b)$.

2. Solve the continuum equations in Ω_2 using the Newmann boundary condition $u_2'(b) = z/a_2$.

3. Compute the new number of particles $n(b)^{(k+1)}$ on the boundary by converting the concentration $u_2(b)$ to a number of particles as described in section 9.2.

We consider (1)-(3) as a single step of a stochastic process that provides a mapping from the number of particles $n(b)^{(k)}$ to $n(b)^{(k+1)}$. Since the new

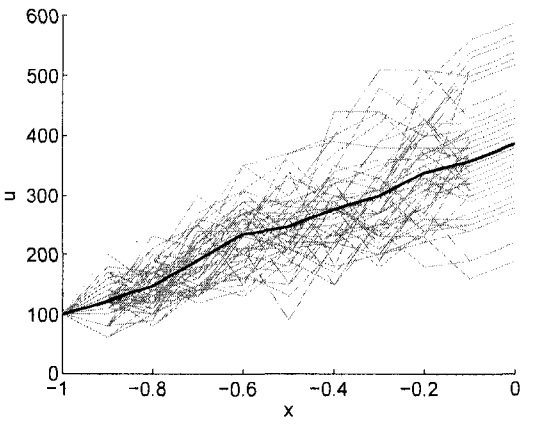

Figure 9.6: Particle concentration for 50 random walks with a fixed Dirichlet condition at $x = -1$ and a Neumann condition at $x = 0$. The bold line indicates the mean concentration of the 50 random walks.

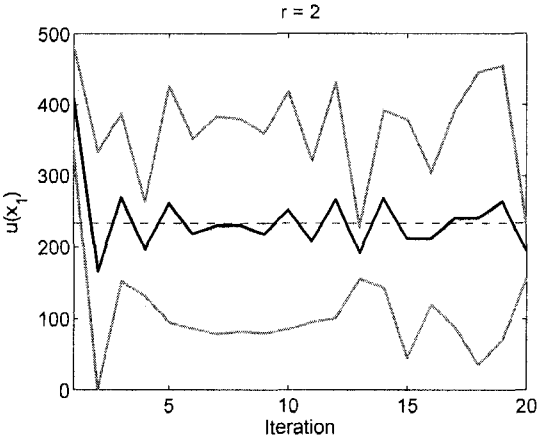

Figure 9.7: Concentration at the boundary at each iteration for $a_1/a_2 = 2$, with the direction of the coupling reversed. The black line indicates the mean concentration and the grey lines indicate three standard deviations above and below the mean. The dashed line indicates the analytic solution.

number of particles $n(b)^{(k)}$ only depends on the previous number of particles $n(b)^{(k-1)}$, we make the Markov assumption

$$P(n(b)^{(k+1)} = j | n(b)^{(1)} = i_1, n(b)^{(2)} = i_2, ..., n(b)^{(k)} = i_k = i)$$
$$= P(n(b)^{(k+1)} = j | n(b)^{(k)} = i).$$

We construct a transition matrix P, where P_{ij} is the probability that given $n(b)^{(k)} = i$, applying steps (1)-(3) results in $n(b)^{(k+1)} = j$. Obtaining P analytically is difficult, since we cannot necessarily obtain the exact transition probabilities for the Brownian motion model. However, rows of P may be estimated numerically by taking many realizations of the process and observing the results.

As P is a stochastic matrix, it has an eigenvalue of 1, with all eigenvalues satisfying $|\lambda| \leq 1$. The stationary distribution π is defined to be the left eigenvector corresponding to the eigenvalue of one, so that $\pi = \pi P$. If P is a primitive matrix, then by Proposition 6.6.4, it has a limiting distribution which is equal to π.

We illustrate this approach for a range of diffusivity ratios $r = a_1/a_2$. The mean and standard deviation for the equilibrium solution π is defined to be

$$\mu = \sum_{k=0}^{K} k\,\pi_k, \qquad \text{and} \qquad \sigma^2 = \sum_{k=0}^{K} (k-\mu)^2\,\pi_k,$$

where K is an upper bound for the number of particles on the boundary.

Although the spectral radius is fixed at 1, the convergence rate $1/|\lambda_2|$, as given in (6.6.4), is affected by r. In addition, the mean and variance of π also varies with $r = a_1/a_2$. From numerical tests, we find that P is generally a primitive matrix whenever $r < 1$. In Table 9.1, we see that the second largest eigenvalue λ_2 approaches -1 as r approaches 1. In addition,

the variance of the limiting distribution is increasing as $r \to 1$, as shown in Table 9.1 and Fig. 9.8.

a_1/a_2	True Solution	μ	σ^2	λ_2
0.5	366.7	366.4	18.8	-0.51
0.6	350	348.9	36.5	-0.69
0.7	335.3	335.4	43.9	-0.77
0.8	322.2	322.2	113	-0.86
0.9	310.5	309.2	240	-0.91

Table 9.1: Mean, variance, and second largest eigenvalue for different diffusivity ratios.

In the case that $r \geq 1$, the matrix fails to be primitive and does not model the original problem (9.2.1). From numerical experiments we typically find that there are two eigenvalues of modulus one in this case, 1 and -1. Although P does have a stationary distribution π in this case which we may find using (6.6.7), the distribution π does not represent the solution to (9.2.1). In general, π will be highly bimodal, as we see in Fig. 9.8 for $r = 1.1$. However, we can address the situation in which $r > 1$ be reversing the directions in which the Dirichlet and Neumann boundary conditions are passed.

The accuracy of the limiting distribution π depends on the accuracy of the transition matrix. The accuracy of the matrix, in turn, depends on the Brownian motion simulation, and the accuracy of the continuum solution, which is addressed using standard *a posteriori* techniques.

9.3.1 A Relaxed Probability Transition Matrix

In section 9.2.2, we discussed a relaxation approach for the fixed point iteration. Here we discuss ways to apply relaxation to the transition probability matrix approach. The basic probability transition matrix approach

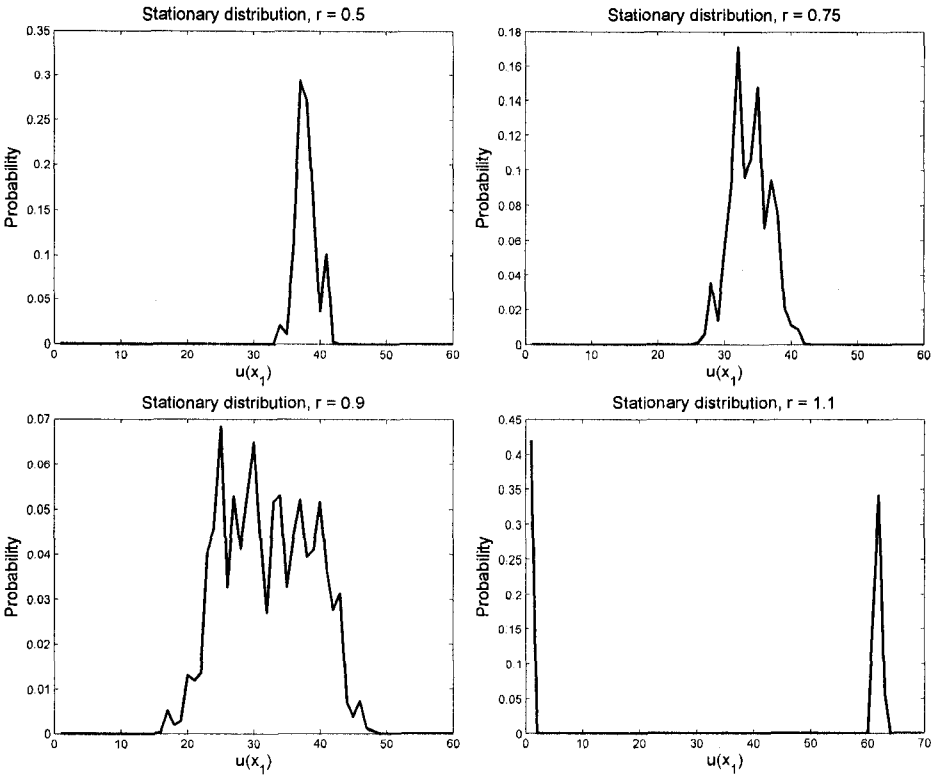

Figure 9.8: Stationary Distributions, $r = 0.5, 0.75, 0.9, 1.1$

involves approximating P, and solving the problem

$$\begin{cases} Y_0 = X_0, \\ Y_n = Y_{n-1}P \quad n \geq 1. \end{cases} \tag{9.3.1}$$

The sequence $x\,Y_n$ converges to the limiting distribution of the Markov chain, for any stochastic row vector x. A natural way to define relaxation in this case is with the fixed point iteration defined by

$$\begin{cases} Y_0 = X_0, \\ Y_n = \alpha Y_{n-1} + (1-\alpha)Y_{n-1}P \quad n \geq 1, \end{cases} \tag{9.3.2}$$

where α is the relaxation parameter, with $\alpha = 0$ corresponding to no relaxation. We define the relaxation matrix, $P^R = \alpha I + (1-\alpha)P$, so that (9.3.2) can be written

$$\begin{cases} Y_0 = X_0, \\ Y_n = Y_{n-1}P^R \quad n \geq 1. \end{cases}$$

We remark that P^R is a stochastic matrix. To see this, we have $\sum_j P_{ij} = 1$ since P is a stochastic matrix. Then,

$$\sum_j P^R_{ij} = \sum_j (\alpha I_{ij} + (1-\alpha)P_{ij}) = \sum_j \alpha P_{ij} + (1-\alpha) = \alpha + 1 - \alpha = 1.$$

The stochastic process (9.3.2) is an entirely different stochastic process than (9.3.1); however, we may show that P and P^R share stationary distributions. Suppose that $\pi P = \pi$. Then,

$$\pi P^R = \pi(\alpha I + (1-\alpha)P) = \alpha\pi + \pi P - \alpha\pi P = \pi P = \pi.$$

There is a clear relationship between the eigenvalues of P and P^R. By linearity, if P has eigenvalue λ, then P^R has eigenvalue $\alpha + (1-\alpha)\lambda$. Define the map $g : \mathbb{R} \to \mathbb{R}$ to be the map from eigenvalues of P, to eigenvalues of P^R, so that

$$g(\lambda) = \alpha + (1-\alpha)\lambda.$$

We note that $g(1) = 1$, and by the Perron-Frobenius Theorem, we have $|g(\lambda)| \leq 1$. In addition, $g(-1) = 2\alpha - 1$, which implies that an eigenvalue of -1 is mapped strictly inside the unit circle, provided that $0 < \alpha < 1$. In fact, the map g generally shrinks the range of eigenvalues with respect to the unit circle. The new spectrum lies inside the circle with radius $1 - \alpha$ and center $(\alpha, 0)$. An example with $r = \alpha = 1/2$ is shown in Fig. 9.9.

From (6.6.4), the convergence rate of (9.3.1) is $1/|\lambda_2|$, where λ_2 is the second largest eigenvalue of P. Since $g(\lambda_2) = \alpha + (1-\alpha)\lambda_2$, the convergence rate is at most

$$\frac{1}{|\alpha + (1-\alpha)\lambda_2|}. \tag{9.3.3}$$

If α is close to 1, the upper bound (9.3.3) of the convergence rate approaches 1, and this relaxation approach fails. In general, relaxation may or may not

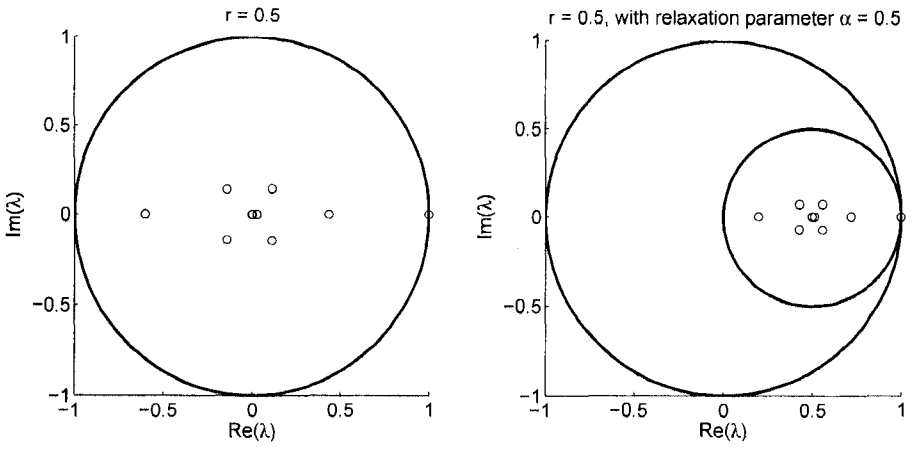

Figure 9.9: Plots of spectrum, $r = 1/2$. Left: Spectrum of P, with the unit circle. Right: Spectrum of P^R, $\alpha = 1/2$, with plots of unit circle and circle of radius $1/2$, with center $(1/2, 0)$.

improve the convergence rate. If λ_2 is close to -1, the map will generally move λ_2 closer to the origin. However, if λ is positive and real, then g moves λ further from the origin. For example, in Table 9.2, we see that $\lambda_2 = -0.603$, and $g(\lambda_2) = 0.199$. However, $|g(0.435)| = 0.718$, and we see that although g has mapped λ_2 closer to the origin, the convergence rate has decreased. For larger ratios r, relaxation may yield a slight improve-

λ	$\lvert\lambda\rvert$	λ^R	$\lvert\lambda^R\rvert$
0	0	0.500	0.500
-0.002	0.002	0.499	0.499
0.027	0.027	0.514	0.514
$0.114 \pm 0.144i$	0.183	$0.557 \pm 0.072i$	0.562
$-0.143 \pm 0.140i$	0.200	$0.429 \pm 0.070i$	0.434
0.435	0.435	0.718	**0.718**
-0.603	**0.603**	0.199	0.199
1	1	1	1

Table 9.2: Spectrum of P and P^R, before and after relaxation ($r = \alpha = 1/2$).

ment in the convergence rate. For example, using $r = 0.9$ and $\alpha = 1/2$,

we numerically obtain convergence rates of 1.07 and 1.12 before and after relaxation, respectively.

Chapter 10

CONCLUSIONS

In the first part of this thesis, we study optimization of a quantity of interest of a solution to an elliptic problem, with respect to parameters in the data. We use the generalized Green's function as an efficient way to compute the gradient. We used gradient search techniques as described in Chapter 4 to generate a minimizing sequence. We analyze the effect of numerical error on a gradient search. Using *a posteriori* error control based on adjoints and residuals as introduced in Chapter 3, we devise an efficient adaptive algorithm to control the error in the gradient. We apply this technique to an example in one dimension with dramatic results, and to a wound healing model in two dimensions.

In the second part of this thesis, we create a mathematical framework for coupling atomistic with continuum models. We first look at the case of coupled diffusion, and examine the criteria for the fixed point iteration to converge. In case the condition is not satisfied, we look to use relaxation or to reverse the direction of the coupling. We use the Green's function approach to get an expression for the values at the interface, as well as the derivatives of the Dirichlet-to-Neumann and Neumann-to-Dirichlet maps.

In the last chapter, we look at the simple one dimensional case in which the particles undergo a random walk on a lattice, next to a continuum region. We generate samples and use kernel density estimation to

approximate the density functions of the uncertainties. We then sample from these density functions to generate the boundary conditions for the adjacent region. This defines a fixed point iteration of distributions. From numerical tests, the convergence criteria mimics that of the criteria for the case of coupled continuous diffusion. Just as in the case of continuous coupling, we overcome an unstable ratio of diffusivities by using relaxation or by reversing the direction of the coupling. It is not clear how to define convergence of the distributions. We see that after a certain number of fixed point iterations, the values oscillate due to the random noise from the atomistic region. If needed, we dampen the oscillation by increasing the number of samples. We also discussed a probability transition matrix approach, where we assume the boundary conditions at each iteration follow a Markov chain. We discussed criteria for the existence of a limiting distribution. We finished the chapter with the formulation of a relaxed probability transition matrix.

The future holds many possibilities for further investigations. We would like to develop a rigorous notion of convergence for a sequence of distributions. We would like to account and correct for numerical errors in the coupling in an efficient way. The mathematical theory for achieving these goals is lacking.

Bibliography

[1] Francis J. Alexander, Alejandro L. Garcia, and Daniel M. Tartakovsky. Algorithm refinement for stochastic partial differential equations: I. linear diffusion. *J. Comput. Phys.*, 182(1):47–66, 2002.

[2] Antonio Ambrosetti and Giovanni Prodi. *A primer of nonlinear analysis*, volume 34 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1993.

[3] Wolfgang Bangerth and Rolf Rannacher. *Adaptive finite element methods for differential equations*. Lectures in Mathematics ETH Zürich. Birkhäuser Verlag, Basel, 2003.

[4] R. Becker, H. Kapp, and R. Rannacher. Adaptive finite element methods for optimization problems. In *Numerical analysis 1999 (Dundee)*, volume 420 of *Chapman & Hall/CRC Res. Notes Math.*, pages 21–42. Chapman & Hall/CRC, Boca Raton, FL, 2000.

[5] Roland Becker, Hartmut Kapp, and Rolf Rannacher. Adaptive finite element methods for optimal control of partial differential equations: basic concept. *SIAM J. Control Optim.*, 39(1):113–132 (electronic), 2000.

[6] Roland Becker and Rolf Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numer.*, 10:1–102, 2001.

[7] T. Belytschko, S. P. Xiao, G. C. Schatz, and R. S. Ruoff. Atomistic simulations of nanotube fracture. *Phys. Rev. B*, 65, 2002.

[8] Dimitri P. Bertsekas. *Nonlinear programming*. Athena Scientific, second edition, 2003.

[9] Dimitri P. Bertsekas and John N. Tsitsiklis. Gradient convergence in gradient methods with errors. *SIAM J. Optim.*, 10(3):627–642 (electronic), 2000.

[10] D. Bors and S. Walczak. Stability of nonlinear elliptic systems with distributed parameters and variable boundary data. In *Proceedings of the 10th International Congress on Computational and Applied Mathematics (ICCAM-2002)*, volume 164/165, pages 117–130, 2004.

[11] Susanne C. Brenner and L. Ridgway Scott. *The mathematical theory of finite element methods*, volume 15 of *Texts in Applied Mathematics*. Springer-Verlag, New York, second edition, 2002.

[12] Abhijit Chatterjee and Dionisios G. Vlachos. An overview of spatial microscopic and accelerated kinetic monte carlo methods. *J Computer-Aided Mater Des*, 14:253–308, 2007.

[13] Edwin K.P. Chong and Stanislaw H. Zak. *An Introduction to Optimization*. Wiley, second edition, 2001.

[14] W. Curtin and R. Miller. Atomistic/ continuum coupling methods in multi-scale materials modeling. *Modeling and Simulation in Materials Science and Engineering*, 11:R33–R68, 2003.

[15] James W. Demmel. *Applied numerical linear algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[16] J.E. Dennis and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, 1996.

[17] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. *Computational differential equations*. Cambridge University Press, Cambridge, 1996.

[18] Kenneth Eriksson, Don Estep, Peter Hansbo, and Claes Johnson. Introduction to adaptive methods for differential equations. In *Acta numerica, 1995*, Acta Numer., pages 105–158. Cambridge Univ. Press, Cambridge, 1995.

[19] D. Estep and D. Neckels. Fast methods for determining the evolution of uncertain parameters in reaction-diffusion equations. *Comput. Methods Appl. Mech. Engrg.*, 196(37-40):3967–3979, 2007.

[20] Donald Estep, Michael Holst, and Mats Larson. Generalized Green's functions and the effective domain of influence. *SIAM J. Sci. Comput.*, 26(4):1314–1339 (electronic), 2005.

[21] Donald Estep, Axel Malqvist, and Simon Tavener. Error estimation and adaptive computation for elliptic problems with randomly perturbed data. *(draft)*.

[22] Donald J. Estep, Mats G. Larson, and Roy D. Williams. Estimating the error of numerical solutions of systems of reaction-diffusion equations. *Mem. Amer. Math. Soc.*, 146(696):viii+109, 2000.

[23] Lawrence C. Evans. *Partial Differential Equations.* American Mathematical Society, 1998.

[24] Jacob Fish. Bridging the scales in nano engineering and science. *Journal of Nanoparticle Research*, 8:577–594, 2006.

[25] Daan Frenkel and Berend Smit. *Understanding Molecular Simulation.* Computational Science Series, Volume 1. Academic Press, second edition, 2002. From Algorithms to Applications.

[26] Desmond J. Higham. Modeling and simulating chemical reactions. *SIAM Rev.*, 50(2):347–368, 2008.

[27] Roger A. Horn and Charles R. Johnson. *Matrix analysis.* Cambridge University Press, Cambridge, 1990. Corrected reprint of the 1985 original.

[28] Thomas J. R. Hughes. *The finite element method.* Prentice Hall Inc., Englewood Cliffs, NJ, 1987. Linear static and dynamic finite element analysis, With the collaboration of Robert M. Ferencz and Arthur M. Raefsky.

[29] S. Kohlhoff and S. Schmauder. A new method for coupled elastic-atomistic modelling. In V. Vitek and D. Srolovitz, editors, *Atomistic Simulation of Materials: Beyond Pair Potentials.* Plenum Press, 1989.

[30] Cornelius Lanczos. *Linear differential operators.* Dover Publications Inc., Mineola, NY, 1997. Reprint of the 1961 original.

[31] Gregory F. Lawler. *Introduction to stochastic processes.* Chapman & Hall/CRC, Boca Raton, FL, second edition, 2006.

[32] David G. Luenberger. *Linear and nonlinear programming.* Kluwer Academic Publishers, Boston, MA, second edition, 2003.

[33] Guri I. Marchuk, Valeri I. Agoshkov, and Victor P. Shutyaev. *Adjoint equations and perturbation algorithms in nonlinear problems.* CRC Press, Boca Raton, FL, 1996.

[34] E. Martinez, J. Marian, M.H. Kalos, and J.M. Perlado. Synchronous parallel kinetic monte carlo for continuum diffusion-reaction systems. *Journal of Computational Physics*, 227:3804–3823, 2008.

[35] Robert C. McOwen. *Partial Differential Equations.* Prentice-Hall, 1996.

[36] J. Medhi. *Stochastic processes.* John Wiley & Sons Inc., New York, second edition, 1994.

[37] J. D. Murray. *Mathematical biology. II*, volume 18 of *Interdisciplinary Applied Mathematics.* Springer-Verlag, New York, third edition, 2003. Spatial models and biomedical applications.

[38] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*, volume 30 of *Classics in Applied Mathematics.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000. Reprint of the 1970 original.

[39] Michael Renardy and Robert C. Rogers. *An introduction to partial differential equations*, volume 13 of *Texts in Applied Mathematics.* Springer-Verlag, New York, second edition, 2004.

[40] Frigyes Riesz and Béla Sz.-Nagy. *Functional analysis.* Dover Books on Advanced Mathematics. Dover Publications Inc., New York, 1990. Translated from the second French edition by Leo F. Boron, Reprint of the 1955 original.

[41] Sheldon M. Ross. *Simulation.* Academic Press, fourth edition, 2006.

[42] R. Rudd and J. Broughton. Coarse-grained molecular dynamics and atomic limit of finite elements. *Phys Rev B*, 58:R5893–R5896, 1998.

[43] David W. Scott. *Multivariate density estimation.* Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons Inc., New York, 1992. Theory, practice, and visualization, A Wiley-Interscience Publication.

[44] E. Tadmor, M. Ortiz, and R. Phillips. Quasicontinuum analysis of defects in solids. *Phil Mag A*, 73(6).

[45] J. Tello and W.A. Curtin. A coupled discrete/continuum model for multiscale diffusion. *International Journal for Multiscale Computational Engineering*, 3:257–265, 2005.

[46] Jos Thijssen. *Computational Physics.* Cambridge University Press, second edition, 2007.

[47] Mark E. Tuckerman and Glenn J. Martyna. Understanding modern molecular dynamics: Techniques and applications. *J. Phys. Chem. B*, 104:159–178, 2000.

[48] N. G. van Kampen. *Stochastic processes in physics and chemistry.* North-Holland Publishing Co., Amsterdam, 1981. Lecture Notes in Mathematics, 888.

[49] A.F. Voter. Introduction to the kinetic monte carlo method. In Kurt E. Sickafus, Eugene A. Kotomin, and Blas P. Uberuaga, editors, *Radiation Effects in Solids.* Springer.

[50] C. Y. Wang and M. X. Li. Convergence property of the Fletcher-Reeves conjugate gradient method with errors. *J. Ind. Manag. Optim.*, 1(2):193–200, 2005.

[51] T. Wildey, S. Tavener, and D. Estep. A posteriori error estimation of approximate boundary fluxes. *Communications in Numerical Methods in Engineering*, 24:421–434, 2008.

[52] Erich Zauderer. *Partial differential equations of applied mathematics.* Pure and Applied Mathematics (New York). John Wiley & Sons Inc., New York, second edition, 1989. A Wiley-Interscience Publication.

[53] J.A. Zimmerman, P.A. Klein, and E.B. Webb III. Coupling and communicating between atomistic and continuum simulation methodologies. In G.C. Sih, editor, *Multiscaling in Molecular and Continuum Mechanics: Interaction of Time and Size from Macro to Nano.* Springer, 2006.

Appendix A

SELECTED PROOFS

Theorem A.0.1. *Let B de defined as in (2.5.6). If $-\frac{1}{2}\nabla \cdot \mathbf{b} + c \geq 0$, then B is coercive.*

Proof. First, we claim that

$$\int_\Omega w\,\mathbf{b}\cdot\nabla w\,dx = -\frac{1}{2}\int_\Omega (\nabla\cdot\mathbf{b})w^2\,dx + \frac{1}{2}\int_{\partial\Omega} w^2\mathbf{b}\cdot\mathbf{n}\,dS. \tag{A.0.1}$$

Using the identity $\nabla\cdot(u\mathbf{b}) = \nabla u\cdot\mathbf{b} + u\nabla\cdot\mathbf{b}$, we have

$$-\frac{1}{2}\int_\Omega (\nabla\cdot\mathbf{b})w^2\,dx = \frac{1}{2}\int_\Omega \mathbf{b}\cdot\nabla(w^2)\,dx - \frac{1}{2}\int_\Omega \nabla\cdot(w\mathbf{b})\,dx.$$

Using the identity $\nabla(w^2) = 2w\nabla w$ on the first term, and the Divergence theorem on the last term,

$$-\frac{1}{2}\int_\Omega (\nabla\cdot\mathbf{b})w^2\,dx = \int_\Omega w\,(\mathbf{b}\cdot\nabla w)\,dx - \frac{1}{2}\int_{\partial\Omega} w^2\,\mathbf{b}\cdot\mathbf{n}\,dS,$$

which proves (A.0.1). Next,

$$\begin{aligned} B(u,u) &= \int_\Omega a|\nabla u|^2\,dx + \int_\Omega u\,\mathbf{b}\cdot u\,dx + \int_\Omega c\,u^2\,dx \\ &= \int_\Omega a|\nabla u|^2\,dx - \frac{1}{2}(\nabla\cdot\mathbf{b})\,u^2\,dx + \frac{1}{2}\int_{\partial\Omega} w^2\,\mathbf{b}\cdot\mathbf{n}\,dS + \int c\,u^2\,dx \\ &= \int_\Omega a|\nabla u|^2\,dx + \int_\Omega (c - \frac{1}{2}\nabla\cdot\mathbf{b})\,u^2\,dx \\ &\geq a_0\|u\|_{H_0^1(\Omega)}. \end{aligned}$$

□

Proposition A.0.1. *Let A be a primitive, stochastic matrix, where $xA = x, Ae = e$, and $L = ex$. Then $\lim_{m\to\infty} A^m = L$.*

Proof. First, note that $L^m = (ex)(ex)...(ex)$, but since $\sum x_i = 1$, we have $xe = 1$. It follows that

$$L^m = e\,x = L. \tag{A.0.2}$$

Next, $A^m\,L = A^{m-1}AL = A^{m-1}Aex = A^{m-1}ex = A^{m-1}L$, and inductively,

$$A^m L = L, \text{ and similarly, } LA^m = L. \tag{A.0.3}$$

Next, we claim that

$$(A - L)^m = A^m - L. \tag{A.0.4}$$

This is trivially true if $m = 1$. If we suppose it is true for $m = k$, then $(A - L)^k = A^k - L$. Then,

$$\begin{aligned}(A - L)^{k+1} &= (A - L)^k(A - L) = (A^k - L)(A - L) \\ &= A^{k+1} - LA - A^kL + L^2 = A^{k+1} - L - L + L = A^{k+1},\end{aligned}$$

where we have used (A.0.2) and (A.0.3).

Next, we claim that if $\mu \neq 0$ is an eigenvalue of $A - L$, then μ is also an eigenvalue of A. Suppose that $(A - L)w = \mu w$ for $w \neq 0$. Then premultiplying by L gives

$$\mu Lw = L(A - L)w = (LA - L^2)w = (L - L)w = 0,$$

hence $Lw = 0$. Then, $\mu w = (A - L)w = Aw - Lw = Aw$ which shows that μ is an eigenvalue of A.

We can then show that 1 is not an eigenvalue of $A-L$. If $(A-L)w = w$, then $Aw = w$ as shown above with $\mu = 1$. But since A is primitive, the eigenvalue of 1 is simple, so that $w = \alpha e$ for some $\alpha \neq 0$. Hence,

$$w = (A - L)w = A\alpha e - L\alpha e = \alpha e - e\,x\alpha e = \alpha e - \alpha e = 0,$$

where we have used the fact that $x\,e = 1$. This contradicts the fact that $w \neq 0$. Hence, 1 is not an eigenvalue of $A - L$.

Next, we show that $\rho(A - L) \leq |\lambda_2|$, where $1 = \lambda_1 \geq |\lambda_2| \geq ... \geq |\lambda_N|$. If $\mu \neq 0$ is an eigenvalue of $A - L$, then μ is an eigenvalue of A as shown above. Thus, $\rho(A - L) = |\lambda_k|$, where λ_k is some eigenvalue of A. By the previous fact, $\lambda_k \neq 1$, and since A is primitive, we then have that $\rho(A - L) = |\lambda_k| < 1$.

Finally,

$$\begin{aligned} A^m &= L + A^m - L \\ &= L + (A - L)^m, \text{ by } (A.0.4). \end{aligned}$$

Then, since $\rho(A - L) < 1$,

$$\lim_{m\to\infty} A^m = L = e\,x.$$

□

Proposition A.0.2. *A finite Markov chain with n states is irreducible if and only if its probability transition matrix P satisfies* $(I + P)^{n-1} > 0$.

Proof. We first claim that $p_{ij}^m > 0$ if and only if the chain can go from state i to state j in m steps. If $m = 1$, this is trivial. If $m = 2$, we have $p_{ij}^2 = \sum_{k=1}^n p_{ik}p_{kj} > 0$ if and only if $p_{ik} > 0$ and $p_{kj} > 0$ for some k. Hence,

a chain starting in state i can go to state j in 2 steps. Now, suppose the property is true for $m = q$. Then,

$$p_{ij}^{q+1} = \sum_{k=1}^{n} p_{ij}^{q} p_{kj} > 0,$$

if and only if $p_{ik}^{q} > 0$ and $p_{kj} > 0$ for some k. That is, there is a k such that we can get from i to k in q steps and k to j in one step. This proves the claim.

Next,

$$(I+P)^{n-1} = I + (n-1)P + \binom{n-1}{2} P^2 + ... + \binom{n-1}{n-2} P^{n-1}.$$

This is a positive matrix if and only if for all entries (i,j), at least one of $I, P, P^2, ..., , P^{n-1}$ has a positive (i,j) entry. By the claim, this holds if and only if we can get from state i to state j in less than n steps. □

Proposition A.0.3. *Suppose P is periodic with period 2. One can check that for large n, the sequence vP^n is a 2 cycle, whose average approaches the stationary distribution π. That is,*

$$\pi_j = \lim_{n\to\infty} \frac{1}{2}\left(p_{ij}^{n} + p_{ij}^{n+1}\right).$$

Proof. Since P has period 2, it has eigenvalues 1 and -1, with all other eigenvalues satisfying $|\lambda| < 1$. By (6.4.2), we have

$$xP^k = \alpha_1 v_1 + \alpha_2(-1)^k v_2 + \sum_{i=3}^{n} \alpha_i \lambda_i^k v_i \to \alpha_1 v_1 + \alpha_2(-1)^k v_2,$$

for arbitrary initial distribution x. Hence,

$$\begin{aligned}\frac{1}{2}(xP^k + xP^{k+1}) &= \frac{1}{2}[\alpha_1 v_1 + \alpha_2(-1)^k v_2 + \alpha_1 v_1 + \alpha_2(-1)^{k+1} v_2 + \xi(k)] \\ &= \alpha_1 v_1 + \frac{1}{2}\xi(k),\end{aligned}$$

where $\xi(k) \to 0$ as $k \to \infty$. Since v_1 is the left eigenvector corresponding the the eigenvalue of 1, it is also the stationary distribution. Taking the limit $k \to \infty$ completes the proof. □