The Dissertation Committee for Vikram Vinod Garg
certifies that this is the approved version of the following dissertation:

Coupled Flow Systems, Adjoint Techniques and Uncertainty Quantification

Committee:

Serge Prudhomme, Supervisor

Graham F. Carey, Supervisor

Clint N. Dawson

Irene Gamba

Omar Ghattas

J. Tinsley Oden

Varis Carey

Coupled Flow Systems, Adjoint Techniques and Uncertainty Quantification

by

Vikram Vinod Garg, B.S. AsE., B.S. Math, M.S.C.A.M.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2012

For my mother Sarita, father Vinod Kumar and sister Neeti.

Dedicated to my late adviser, Dr. Graham Carey.

Acknowledgments

There is a couplet by the Indian poet Kabir that I would often hear from my parents while growing up. Its English translation reads,

Teacher and God both are here
whom should I acknowledge first
All glory be unto the Teacher
path to God who did bestow

It has been a privilege to have learnt from and worked with many outstanding individuals throughout my academic life. I must particularly mention three people, who have had a major impact on me as both a researcher, and as a person.

Dr. Graham Carey, my late adviser who introduced me to this field. His enthusiasm, energy and humility have been among the greatest inspirations in my life. Dr. Carey had a deep committment to his work and his students. Reaching this stage would not have been possible without his guidance.

Dr. Serge Prudhomme, my co-advisor. Dr. Prudhomme's humility, his insistence on understanding every detail of a problem and his discipline are guiding lights for me. His influence on me as a researcher has been profound.

Dr. Roy Stogner, my mentor and hero. Roy's inspirational brilliance, extraordinary humility and his patience in listening to my questions, doubts and ideas has been a key ingredient in my development, both personally and professionally.

I really consider myself lucky to have worked with such sharp minds and simple, humble human beings. This good luck has continued with my other major collaborators, Dr. Kris van der Zee and Dr. Varis Carey. I would also like to thank all the folks that have worked with me in the CFD lab, in particular, Dr. John Peterson, Dr. Ben Kirk, Derek Gaston, Dr. Kemelli Estacio and Dr. David Knezevic.

Thanks also to the CSEM gang, a bunch of smart, funny and talented individuals from whom I have learnt a great deal. I must particularly mention Rhys Ulerich for all his help with C++ and discussions on various aspects of my work. Andrea Hawkins, Chris Mirabito, Thomas Kirschenmann, Omar Hinai and Jeff Zitelli, thank you folks for all the discussions on research, math, economics, politics, romance and everything else. A big thanks to Jeff Hussman for the discussions on some of the proofs in this dissertation.

My friends Akanksha, Avni, Shrawan, Sucheta, Bala, Vinay, Parvathy, I couldnt have made it without you folks.

Finally, dad, mom and didi, thank you for all your love and sacrifices.

Coupled Flow Systems, Adjoint Techniques and Uncertainty Quantification

Publication No. _____

Vikram Vinod Garg, Ph.D.
The University of Texas at Austin, 2012

Supervisors: Serge Prudhomme
Graham F. Carey

Coupled systems are ubiquitous in modern engineering and science. Such systems can encompass fluid dynamics, structural mechanics, chemical species transport and electrostatic effects among other components, all of which can be coupled in many different ways. In addition, such models are usually multiscale, making their numerical simulation challenging, and necessitating the use of adaptive modeling techniques. The multiscale, multiphysics models of electrosomotic flow (EOF) constitute a particularly challenging coupled flow system. A special feature of such models is that the coupling between the electric physics and hydrodynamics is via the boundary.

Numerical simulations of coupled systems are typically targeted towards specific Quantities of Interest (QoIs). Adjoint-based approaches offer the possibility of QoI targeted adaptive mesh refinement and efficient parameter sensitivity analysis. The formulation of appropriate adjoint problems for

EOF models is particularly challenging, due to the coupling of physics via the boundary as opposed to the interior of the domain. The well-posedness of the adjoint problem for such models is also non-trivial. One contribution of this dissertation is the derivation of an appropriate adjoint problem for slip EOF models, and the development of penalty-based, adjoint-consistent variational formulations of these models. We demonstrate the use of these formulations in the simulation of EOF flows in straight and T-shaped microchannels, in conjunction with goal-oriented mesh refinement and adjoint sensitivity analysis.

Complex computational models may exhibit uncertain behavior due to various reasons, ranging from uncertainty in experimentally measured model parameters to imperfections in device geometry. The last decade has seen a growing interest in the field of Uncertainty Quantification (UQ), which seeks to determine the effect of input uncertainties on the system QoIs. Monte Carlo methods remain a popular computational approach for UQ due to their ease of use and "embarassingly parallel" nature. However, a major drawback of such methods is their slow convergence rate.

The second contribution of this work is the introduction of a new Monte Carlo method which utilizes local sensitivity information to build accurate surrogate models. This new method, called the Local Sensitivity Derivative Enhanced Monte Carlo (LSDEMC) method can converge at a faster rate than plain Monte Carlo, especially for problems with a low to moderate number of uncertain parameters. Adjoint-based sensitivity analysis methods enable the computation of sensitivity derivatives at virtually no extra cost after the

forward solve. Thus, the LSDEMC method, in conjuction with adjoint sensitivity derivative techniques can offer a robust and efficient alternative for UQ of complex systems.

The efficiency of Monte Carlo methods can be further enhanced by using stratified sampling schemes such as Latin Hypercube Sampling (LHS). However, the non-incremental nature of Latin Hypercube Sampling has been identified as one of the main obstacles in its application to certain classes of complex physical systems. Current incremental LHS strategies restrict the user to at least doubling the size of an existing LHS set to retain the convergence properties of Latin Hypercube Sampling. The third contribution of this research is the development of a new Hierachical Latin Hypercube Sampling algorithm, that creates designs which can be used to perform LHS studies in a more flexibly incremental setting, taking a step towards adaptive LHS methods.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

The second half of the twentieth century saw enormous strides in the development of high-performance computing and its deployment in the modeling and simulation of a variety of problems in science and engineering. Beginning with the use of standalone codes for structural mechanics, fluid mechanics and other applications, sophisticated computational models are now being used to simulate coupled systems containing multiple scales and physics. As the complexity of computer models and simulations has increased, there has also been a growing interest in the reliability of their predictions [47, 35]. The quantification of uncertainty inherent in complex physical phenomena and numerical models has thus gained increasing importance in computational science and engineering [42].

Alongside the development of the microcomputer and other miniaturized computational devices, there has been a concurrent effort towards developing micro- and nano-scale technologies and devices for various applications [93, 53]. This has led to an increased interest in the physics of small scale phenomena, including fluid flow at micro- and nano-scales [104]. Such flows

and the devices that utilize them are increasingly prevalent in science and commercial enterprises [93, 53]. Examples include bioassays consisting of microfluidic networks designed for patterned drug delivery [60] and microfluidic fuel cells [24]. Such microfluidic devices operate over various length scales and are best described using multiphysics modeling that involves hydrodynamics, electroosmosis, and chemical species transport models.

On account of their coupled, multiscale nature, the development of accurate, efficient and reliable computational simulators of microfluidic devices is challenging and resource intensive. Significant research efforts have been devoted in the past decade towards the development of better numerical methods for simulating microfluidics models and quantifying the uncertainty seen in such flows and related devices. In this work, we make further contributions to the development of efficient and reliable numerical methods for microfluidics models and to the analysis of solution sensitivity to various model and numerical parameters.

1.2 Motivation

Numerical simulations of complex engineering systems are typically targeted towards the calculation of specific Quantities of Interest (QoIs) associated with the systems. Accurate estimation of local QoIs can be achieved using goal-oriented error estimation and adaptive techniques based on the use of adjoint methods [80, 68, 13, 37]. Adjoint methods can also be used to improve the computational performance of parameter sensitivity analyses [50],

especially for systems with a large number of parameters. The application of adjoint methods to coupled flow systems and uncertainty quantification is still an open and active area of study. To our best knowledge, no advances in the application of adjoint-based techniques to microfluidics applications, particularly those involving 'slip' boundary coupling, have yet been published in the literature. For related UQ problems, the low convergence rate of the Monte Carlo method is an important obstacle in its application to complex models such as those encountered in microfluidics.

Therefore, two problems will be addressed in this dissertation:

1. The formulation of an appropriate adjoint problem for coupled electroosmotic flows, which will enable goal-oriented mesh refinement and adjoint sensitivity analysis for such flows.

2. The development of a Monte Carlo method that utilizes adjoint sensitivity derivatives to improve on the convergence properties of plain Monte Carlo.

We propose a modified variational formulation of the slip electroosmotic flow model of microfluidics. Using such a formulation, the adjoint problem can be computed and used in adaptive mesh refinement and parameter sensitivity analysis. Then, building on the work of Cao et al. [19], and the ability to compute sensitivity derivatives efficiently using adjoint techniques, a new Local Sensitivity Derivative Enhanced Monte Carlo method shall be introduced. This method can improve the convergence rate of the Monte Carlo method,

especially for problems with a moderate number of random parameters. These twin contributions can help develop accurate and robust computational models for coupled flow systems and related uncertainty quantification problems.

1.3 Literature Review

The last decade has seen a growth in the modeling and numerical simulation of microfluidic systems. Ren et al. [86] simulated microfluidic injection processes of chemical species. Zhang et al. [108] presented simulations of electroosmotic flow in microchannels of various shapes. Craven et al. [28] explored the implications of the widely used Helmholtz Smoluchowski (HS) slip velocity boundary condition using detailed numerical simulations. Beskok and Hahm studied species entrapment using microfluidic devices through numerical simulation [46]. Zimmerman et al. [110] proposed a new simulation based approach for identifying non-Newtonian fluids, using detailed parameter sensitivity and statistical analyses.

Efforts have also been made towards devising adaptive methods for microfluidics problems. Prachittham et al. [78] presented a space-time adaptive finite element method applied to an electroosmotic flow using large aspect ratio elements. Choi and Paraschivoiu presented a goal-oriented adaptive finite element strategy for microfluidics using the *bound method* [26, 25]. However, their work did not consider adjoint techniques for the coupled electrostatic and hydrodynamic problem and did not use the slip boundary coupling condition. On the other hand, van Brummelen et al. [101] and Estep et al. [38] have

shown the importance of the treatment of boundary flux coupling for the use of adjoint-based techniques.

On the UQ front, Debusschere et al. [29] have analyzed the uncertainties arising in a reacting microchannel flow. Further work on UQ for a broader range of microfluidic systems was presented by Knio et al. [57]. Computational algoritms for UQ cluster around two broad areas: the polynomial chaos and other stochastic expansion based methods that express the stochastic process as a series expansion [106, 7, 2, 3], and the sampling based approaches arising from the Monte Carlo method [47, 51]. Although stochastic expansion methods can deliver fast convergence rates, they have the drawbacks of being intrusive to implement [103] and of requiring a high degree of regularity in the stochastic space [65] and are inefficient for high-dimensional problems [106, 103]. Monte Carlo methods on the other hand, are straightforward to implement, non-intrusive, converge at a dimension independent rate, and are "embarassingly parallel". However, they converge at a slow rate of $N_s^{-\frac{1}{2}}$, where N_s is the number of samples in a Monte Carlo study [71].

Various strategies have been proposed and used for improving convergence properties of the Monte Carlo method. These include modified sampling techniques such as Latin Hypercube Sampling (LHS) [95] and Hammersley sampling [1]. LHS retains the $N_s^{-\frac{1}{2}}$ rate of convergence of Simple Random Sampling (SRS) [95], but can substantially improve the constant of convergence [72]. However, the non-incremental nature of Latin Hypercube Sampling has been identified as one of the main obstacles in its application to

certain classes of complex physical systems [47]. Some approaches have been proposed for overcoming the non-incremental nature of LHS. Robinson [88] proposed an iterative quasi-Monte Carlo method based on the Halton low discrepancy sequence. Pleming and Manteufel presented a "replicated Latin Hypercube Sampling" [77], which increases the number of samples by a user-specified base size but does not retain the Latin hypercube structure for the enlarged design. In current implementations of an Incremental LHS (ILHS) method, one is restricted to at least doubling the size of an existing LHS set to retain the design properties of Latin Hypercube Sampling [90, 1]. Surrogate based approaches such as the Sensitivity Derivative Enhanced Monte Carlo have also been introduced by Cao et al. [18]. Like LHS, these techniques improve the constant of convergence for the Monte Carlo method [19]. The SDEMC method has been used for UQ in fluid mechanics [81] and structural mechanics [51].

1.4 Research Contributions

The new research contributions from this work include:

1. Analysis and application of adjoint-based techniques to widely used microfluidics models.

2. Implementation and verification of adjoint-based refinement and sensitivity analysis in the object oriented C++ Finite Element library `libMesh`.

3. Development of accelerated Monte Carlo techniques to improve the accu-

racy and efficiency of uncertainty quantification using adjoint sensitivity derivatives.

4. Application of above ideas for the numerical simulation of microfluidic devices and uncertainty quantification for a model Poisson problem.

5. Development of an incremental Latin Hypercube Sampling algorithm to allow efficient use of LHS in large-scale UQ problems.

6. Analysis of the error arising in the evaluation of QoIs due to the use of boundary penalty techniques and the development of improved QoI recovery techniques that reduce such errors.

1.5 Outline

Chapter 2 describes the modeling and simulation of microfluidic devices. We then move on to the derivation and variational formulation of the adjoint problem for coupled microfluidics in Chapter 3. Chapter 4 describes the use of adjoints in error estimation and sensitivity analysis. It also includes details on the implementation and verfication of adjoint-based techniques in the software library `libMesh` [56]. In chapter 5, we illustrate the use of these adjoint capabilities in the numerical simulation of microfluidic flow. In Chapter 6, we discuss the evaluation of the boundary flux using a penalty method and the implications for the corresponding adjoint problem. Chapters 7 and 8 describe our UQ work, the new Local Sensitivity Derivative Enhanced Monte

Carlo method and the Hierarchical Incremental Latin Hypercube Sampling method.

Chapter 2

Microfluidics: Multiscale, Multiphysics Flow at the Micron Level

2.1 Introduction

Microfluidics is the branch of fluid mechanics concerned with the understanding, modeling, and control of flows that occur on the micron scale, where the characteristic length (L) is of the order 10^{-6} m. Several changes in flow physics are observed as we approach the micro- and nano-scales. Many of these are driven by the change in the ratio of the surface area of the flow to its volume,

$$\frac{p(A)}{p(V)} \propto \frac{L^2}{L^3} \propto \frac{1}{L} \approx \mathcal{O}(10^6) \tag{2.1}$$

where $p(A)$ are the forces associated with the surface of the flow, e.g. surface tension, near wall electrostatic forces, wall friction, whereas $p(V)$ are the forces associated with the bulk volume of the flow, e.g. inertia, pressure gradients, and gravity. The large surface to volume ratio raises the prospect of precise control over mass and heat transfer, chemical reactions, and separation processes by enabling novel techiques of flow propulsion and control. Prominent among these are microflows driven by electric fields. Such flows may be driven through phenomena known as electroosmosis, electrophoresis, or both. Some

examples of devices that use electric effects to drive flows are cross-channel micro-chips used in various lab-on-a-chip applications [86], micromixers that enhance mixing through the use of electrokinetic instabilities [23] and drug delivery devices [99].

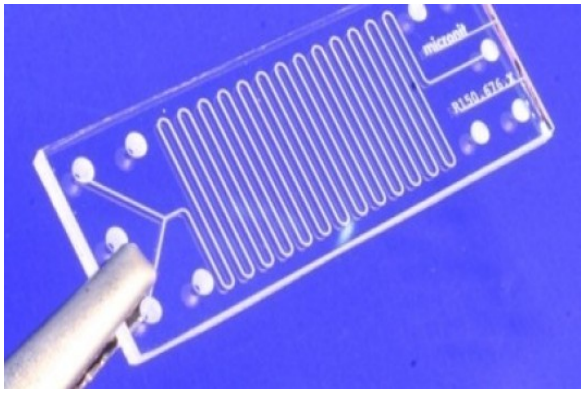


Figure 2.1: A lab on a chip device [45].

Electroosmotic devices utilize the properties of the electric double layer (or the Debye layer, see Figure 2.2) that develops between the fluid and the channel wall. Under the effect of an electric field applied tangential to the channel wall, the charged particles of the double layer experience an electric force, and start motion in the direction of the field. Viscous forces within the fluid then drive the bulk fluid in the direction of the electric field. On the other hand, electrophoresis drives fluid motion through the effect of an electric field on a charged species present in the bulk flow. Electrophoresis is often used to separate certain species from a bulk fluid or other species, for example polymer separation in gels [93]. In this exposition, we will focus on

electroosmotic flows (EOF).

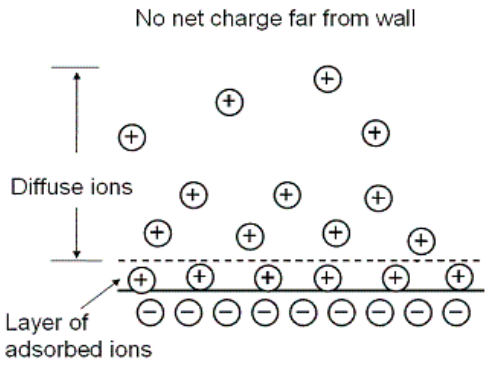


Figure 2.2: Structure of Electric Double Layer (EDL) near the fluid-channel wall interface [58].

2.2 The Physics of an Electroosmotically Driven Flow

As pointed out above, microflows are characterized by the dominance of surface forces over bulk forces and inertia. Interfacial effects near the fluid-channel wall then affect the bulk flow. Since the length scales of microflows are very small, the ratio of the mean free path length (λ) to the characteristic length (L) can be large. However, liquid microflows are still within the continuum regime on account of their small mean free path length, and hence low Knudsen numbers ($\frac{\lambda}{L}$) at ambient pressures. Thus, the incompressible Navier-Stokes equations describe the fluid motion in an Eulerian framework,

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \frac{1}{\mathrm{Re}}\Delta \mathbf{u} + \mathbf{f} \tag{2.2a}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2.2b}$$

where $\mathbf{u}$ and p are the flow velocity and pressure, respectively, $\mathbf{f}$ is the body force and Re is the flow Reynolds number. The dimensionless Reynolds number governs the flow regime,

$$\mathrm{Re} = \frac{\rho U L}{\mu} \tag{2.3}$$

where ρ is the fluid density, μ is the fluid viscosity, U is a characteristic velocity, and L a characteristic length.

The Reynolds number is small, of the order $\mathcal{O}(10^{-6})$ for microflows on account of the small characterstic length. Hence microflows are well described by the Stokes equations, the zero Reynolds number limit of the Navier Stokes equations. If we further assume steady-state flow, the time derivative terms drop out and we get the stationary Stokes equations,

$$-\mu \Delta \mathbf{u} + \nabla p = \mathbf{f} \tag{2.4a}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2.4b}$$

Flows at the microscales are thus dictated by the viscous effects and forces such as electroosmosis and electrophoresis. In addition, the flow may depend on wall shear forces such as surface tension; however those effects will not be considered in this work. Instead, we shall focus our attention on the body force term $\mathbf{f}$ for an electroosmotically driven flow. As mentioned above, in such flows an electric double layer develops near the channel wall with an associated charge distribution ρ_e. It is created by a combination of chemical and thermal effects. Thus, its structure depends on the chemical properties of the fluid and the channel wall surface. This structure is quantatively described

by the Poisson-Boltzmann equation [62],

$$-\Delta\Psi = K^2 \sinh(\Psi) \tag{2.5}$$

where Ψ is the non-dimensional electric potential associated with the channel wall and K is a non-dimensional constant, called the Debye-Huckel parameter.

The Debye-Huckel parameter depends on the dielectric properties of the fluid and some physical constants. It can be calculated as,

$$K = \kappa L \tag{2.6a}$$

$$\kappa = \sqrt{\frac{2 z_v^2 e^2 N_A n_\infty}{\epsilon k_b T}} \tag{2.6b}$$

where the parameter $\frac{1}{\kappa}$, known as the Debye length, is that length scale where the electrostatic interactions between the fluid and the channel wall are significant. The terms in the numerator of Eq. (2.6b) are the ion valence z_v, electron charge e, Avogadro's Number N_A, and bulk concentration of ions n_∞. The terms in the denominator of Eq. (2.6b) are the permittivity (or dielectric constant) ϵ, Boltzmann constant k_b, and temperature T. The Debye-Huckel parameter is the ratio of the bulk fluid and interfacial length scales, and will vary according to the geometry of the channel, the chemical and electric properties of the fluid, and the temperature T. It determines the multiscale nature of the flow. As an example, its value for polystyrene was computed using data from [76] to give, $K \approx 40$. Thus, for polystyrene, the electrostatic interactions take place over a length that is 40 times smaller than the channel height. Consequently, the grid for numerical simulations needs to be much finer near the wall than in the center of the channel [28].

Once Eq. (2.5) has been solved for the electric potential in the electric double layer, the charge distribution can then be recovered from the potential using the expression [62],

$$\rho_e = nz_v e = -2n_\infty \sinh(\Psi) z_v e \tag{2.7}$$

If a tangential electric field $\mathbf{E}$ is now applied, an electric potential ϕ will be generated in the entire channel. The fluid then experiences a net body force,

$$\mathbf{f} = \rho_e \mathbf{E} = -\rho_e \nabla \phi \tag{2.8}$$

This body force can drive an electroosmotic flow even in the absence of any applied pressure gradients. A variety of microfluidic devices use such phenomena for fluid motion [108, 86, 46, 110].

2.3 Modeling

To illustrate and further understand the properties of the electroosmotic flow and the electric double layer, consider a rectangular open domain $\Omega \subset \mathbb{R}^d$, with d = 2, and boundary $\partial\Omega$. The boundary $\partial\Omega$ is composed of the channel wall Γ_w and its inlet/outlet Γ_{io}, such that $\partial\Omega = \overline{\Gamma_w \cup \Gamma_{io}}$. For simplicity, we consider a single species flow through the channel. We consider flows generated purely by electroosmosis, with no pressure driven components. Studying the model in this simple setting will help us understand the essential features of the physical model. Also, many microfluidic devices are simply assembled as a collection of straight channels, for examples cross and T-channels [110, 86].

Based on the discussion in section 2.2, a steady-state EOF in a straight rectangular channel can be modeled with the following set of equations,

$$-\Delta\Psi = K^2 \sinh(\Psi) \qquad \text{in } \Omega \tag{2.9a}$$

$$\rho_e = -2z_v e n_0 \sinh(\Psi) \qquad \text{in } \Omega \tag{2.9b}$$

$$-\nabla \cdot (\sigma_c \nabla\phi) = 0 \qquad \text{in } \Omega \tag{2.9c}$$

$$-\mu\Delta\mathbf{u} + \nabla p = -\rho_e \nabla\phi \qquad \text{in } \Omega \tag{2.9d}$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \Omega \tag{2.9e}$$

The model parameters that the flow field depends on include the Debye-Huckel parameter K, the conductivity of the fluid σ_c, and the fluid viscosity μ. We also have the associated boundary conditions,

$$\Psi = \Psi_0 \quad \text{on } \Gamma_w \tag{2.10a}$$

$$\mathbf{n} \cdot \nabla\Psi = 0 \quad \text{on } \Gamma_{io} \tag{2.10b}$$

$$\mathbf{n} \cdot (\sigma_c \nabla\phi) = 0 \quad \text{on } \Gamma_w \tag{2.10c}$$

$$\phi = \phi_{io} \quad \text{on } \Gamma_{io} \tag{2.10d}$$

$$\mathbf{u} = \mathbf{0} \quad \text{on } \Gamma_w \tag{2.10e}$$

$$\mathbf{u} \cdot \mathbf{t} = 0 \quad \text{on } \Gamma_{io} \tag{2.10f}$$

$$\mathbf{n} \cdot (\boldsymbol{\sigma} \cdot \mathbf{n}) = 0 \quad \text{on } \Gamma_{io} \tag{2.10g}$$

where $\mathbf{n}$ is the unit outward normal vector to the boundary of the domain, $\mathbf{t}$ is a unit tangential vector along $\partial\Omega$, and $\boldsymbol{\sigma}$ is the stress tensor:

$$\boldsymbol{\sigma} = -p\mathbf{I} + \mu(\nabla\mathbf{u} + \nabla^T\mathbf{u}) \tag{2.11}$$

The no-slip condition is applied at the channel walls for the flow velocity. Note that the last two boundary conditions imply that the velocity is normal to Γ_{io} and that the pressure vanishes on Γ_{io} (this is in the case of planar boundaries, see [10]), i.e.

$$p = 0 \quad \text{on } \Gamma_{io} \tag{2.12}$$

Input data Ψ_0 represents what is called the wall zeta potential [62], while ϕ_{io} represents the external potential(s) applied at the inlet and outlet of the channel. Eqs. (2.9) and (2.10) describe the complete or fine-scale EOF model. They form a challenging set of coupled multiscale system of equations. Such systems are computationally expensive to solve, especially in the case of complex geometries [28].

For a straight rectangular channel in a medium of constant conductivity, one can obtain analytic expressions for the flow field variables under certain assumptions [33]. The following expression describes the flow velocity of an electroosmotic flow in a straight channel in the absence of pressure gradients,

$$u_e = -\frac{E_x \Psi_0}{\mu}\left(1 - \frac{\cosh(\kappa y)}{\cosh(\kappa h)}\right) \tag{2.13}$$

where y is the distance from the centerline of the channel and $h = \frac{L}{2}$, the half height of the channel. The velocity u_e is called the Helmholtz-Smoluchowski (HS) velocity. It is the flow velocity tangential to the wall induced by the applied electric field. The Helmholtz-Smoluchowski velocity reaches its free stream value very quickly away from the channel wall for large values of the parameter κh, which occur when the Debye layer length $\left(\frac{1}{\kappa}\right)$ is very small.

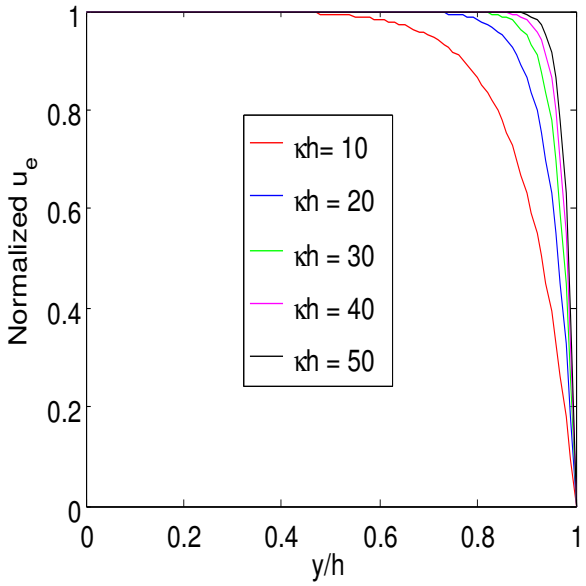


Figure 2.3: EOF velocity profile given by Eq. (2.13) for various values of the parameter κh. The velocity has been normalized as $-u_e/E_x\frac{\Psi_0}{\mu}$. Note the steep rise in u_e for large values of κh.

To reduce the complexity and computational cost associated with the fine-scale model, the Helmholtz-Smoluchowski (HS) velocity approximation is introduced in the model. The approximation states that the body-force term in the Stokes equations (2.9d) can be replaced by an effective 'slip velocity' on the boundary given by,

$$\mathbf{u}_{wall} = \frac{\epsilon\Psi_0}{\mu}\mathbf{E} = \lambda\mathbf{E} \tag{2.14}$$

where we have introduced the new parameter $\lambda = \epsilon\Psi_0/\mu$. The validity of this approximation has been verified through both experiments and numerical simulations [46].

Using the Helmholtz-Smoluchowski approximation, the slip model of

EOF can be stated as,

$$-\nabla \cdot (\sigma_c \nabla \phi) = 0 \quad \text{in } \Omega \tag{2.15a}$$

$$-\mu \Delta \mathbf{u} + \nabla p = \mathbf{0} \quad \text{in } \Omega \tag{2.15b}$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \tag{2.15c}$$

The boundary conditions are,

$$\mathbf{n} \cdot (\sigma_c \nabla \phi) = 0 \quad \text{on } \Gamma_w \tag{2.16a}$$

$$\phi = \phi_{io} \quad \text{on } \Gamma_{io} \tag{2.16b}$$

$$\mathbf{u} + \lambda \nabla \phi = \mathbf{0} \quad \text{on } \Gamma_w \tag{2.16c}$$

$$\mathbf{u} \cdot \mathbf{t} = 0 \quad \text{on } \Gamma_{io} \tag{2.16d}$$

$$\mathbf{n} \cdot (\boldsymbol{\sigma} \cdot \mathbf{n}) = 0 \quad \text{on } \Gamma_{io} \tag{2.16e}$$

We see that the slip model spares one from solving the Poisson-Boltzmann equation, whose solution exhibits a thin layer near the wall. As a remark, the slip boundary approximation model given by Eqs. (2.15) and (2.16) is widely used throughout the microfluidics research and development community for modeling and simulation [53]. The model is even included in the commercial Finite Element software package COMSOL Multiphysics [110].

2.4 The Slip Boundary Condition and its Implications

We now pay close attention to the coupling condition $\mathbf{u} = -\lambda \nabla \phi$. This single constraint actually contains two conditions, one for each component of the velocity. The tangential component of the velocity is proportional to the

tangential gradient of the potential, and the normal component of the velocity is proportional to the normal gradient of the potential.

From the derivation of Eq. (2.13), we recall that physically only the tangential coupling can be justified. In Eq. (2.16c), the no-flux boundary Neumann boundary condition on the potential (see Eq. (2.16a)) automatically enforces a no-penetration boundary condition on the velocity. Thus, expressing the coupling condition as in Eq. (2.16c) is convenient from a notational and implementation standpoint. However, using numerical experiments and some theory, we shall show that coupling the normal component in this manner may lead to ill-posed adjoint problems. Hence, we decouple one of the velocity components from the potential as follows,

$$\begin{cases} \mathbf{u}\cdot\mathbf{t} + \lambda\partial_t\phi = 0 \\ \\ \mathbf{u}\cdot\mathbf{n} + \lambda\partial_n\phi = 0 \end{cases} \quad \Rightarrow \quad \begin{cases} \mathbf{u}\cdot\mathbf{t} + \lambda\partial_t\phi = 0 \\ \\ \mathbf{u}\cdot\mathbf{n} = 0 \end{cases} \tag{2.17}$$

Note that this new coupling is equivalent to the one given by Eq. (2.16c).

In chapter 3, we derive the weak formulation for Eq. (2.15) using the modified coupling given by Eq. (2.17). After developing the appropriate formulation for the forward problem, we obtain the variational statement for the corresponding adjoint problem. Again, we pay close attention to the coupling in the adjoint problem. We then present numerical simulations of EOF in chapter 5.

Chapter 3

The Adjoint Problem for Coupled Electroosmotic Flow

3.1 Introduction

In the previous chapter, we introduced a class of models for microfluidic devices that encompass various physics and length scales. We then discussed a new boundary condition called the slip condition that couples the two physics in an EOF flow problem. We will now develop variational or 'weak' formulations of such slip EOF models, paying particular attention to the boundary coupling condition and the related issues of regularity and well-posedness. The coupling condition will be incorporated into the weak form by using lift and penalty techniques in sections 3.2 and 3.3. This will allow us to develop the weak formulation for the adjoint problem. The consistency of the adjoint problem obtained using these two different techniques will then be shown in section 3.3.3. Finally, in section 3.4 we will discuss the imposition of the coupling boundary condition using penalty techniques and associated well-posedness issues.

3.2 Variational formulation of the slip BC EOF model

3.2.1 Variational formulation of primal problem

We derive here the weak formulation of the equations (2.15) and (2.16). The potential ϕ satisfies,

$$-\nabla \cdot (\sigma_c \nabla \phi) = 0 \quad \text{in } \Omega \tag{3.1a}$$

$$\phi = \phi_{io} \quad \text{on } \Gamma_{io} \tag{3.1b}$$

$$\sigma_c \partial_n \phi = 0 \quad \text{on } \Gamma_w \tag{3.1c}$$

We assume that the data ϕ_{io} are constant on the inlet and outlet. This assumption is usually well justified in applications but can also be easily relaxed. It is a convenient assumption to make in order to simplify some derivations later on. We now introduce the spaces of admissible trial and test functions:

$$Z = H^1(\Omega) \tag{3.2a}$$

$$Z_{\phi_{io}} = \{\varphi \in Z;\ \varphi = \phi_{io} \text{ on } \Gamma_{io}\} \tag{3.2b}$$

$$Z_0 = \{\varphi \in Z;\ \varphi = 0 \text{ on } \Gamma_{io}\} \tag{3.2c}$$

Also, let the conductivity σ_c lie in the space of positive functions that are also bounded below on Ω,

$$\mathrm{C}^+(\Omega) = \{f \in \mathrm{C}(\Omega);\ f \geq f_{\min} > 0,\ f_{\min} \in \mathbb{R}^+\} \tag{3.3}$$

The weak formulation of Eq. (3.1) reads:

$$\text{Given } \sigma_c \in \mathrm{C}^+(\Omega) \text{ and } \phi_{io} \in \mathbb{R}, \text{ find } \phi \in Z_{\phi_{io}} \text{ such that}$$
$$\int_\Omega \sigma_c \nabla \phi \cdot \nabla \psi \, dx = 0, \quad \forall \psi \in Z_0 \tag{3.4}$$

Alternatively, one may introduce a lift function $\phi_{io} \in Z_{\phi_{io}}$ such that $\phi = \varphi + \phi_{io}$ and $\varphi \in Z_0$. In this case, the weak form of the problem can be recast as:

$$\begin{gathered}\text{Given } \sigma_c \in \mathrm{C}^+(\Omega) \text{ and } \phi_{io} \in \mathbb{R}, \text{ find } \varphi \in Z_0 \text{ such that} \\ \int_\Omega \sigma_c \nabla\varphi \cdot \nabla\psi \, dx = -\int_\Omega \sigma_c \nabla\phi_{io} \cdot \nabla\psi \, dx, \quad \forall \psi \in Z_0 \end{gathered} \tag{3.5}$$

We now consider the non-dimensionalized stationary Stokes equation with slip boundaries,

$$-\Delta \mathbf{u} + \nabla p = \mathbf{0} \quad \text{in } \Omega \tag{3.6a}$$

$$-\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \tag{3.6b}$$

$$\mathbf{u} \cdot \mathbf{t} + \lambda \partial_t \phi = 0 \quad \text{on } \Gamma_w \tag{3.6c}$$

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_w \tag{3.6d}$$

$$\mathbf{u} \cdot \mathbf{t} = 0 \quad \text{on } \Gamma_{io} \tag{3.6e}$$

$$\mathbf{n} \cdot (\boldsymbol{\sigma} \cdot \mathbf{n}) = 0 \quad \text{on } \Gamma_{io} \tag{3.6f}$$

We look for the velocity and pressure fields in the function spaces,

$$X = \left[H^1(\Omega)\right]^2 \tag{3.7a}$$

$$X_\phi = \left\{\mathbf{u} \in X;\ \mathbf{u} \cdot \mathbf{t} = -\lambda \partial_t \phi, \mathbf{u} \cdot \mathbf{n} = 0 \text{ on } \Gamma_w,\ \mathbf{u} \cdot \mathbf{t} = 0 \text{ on } \Gamma_{io}\right\} \tag{3.7b}$$

$$X_0 = \left\{\mathbf{u} \in X;\ \mathbf{u} = \mathbf{0} \text{ on } \Gamma_w,\ \mathbf{u} \cdot \mathbf{t} = 0 \text{ on } \Gamma_{io}\right\} \tag{3.7c}$$

$$M = \left\{p \in L^2(\Omega);\ \textstyle\int_\Omega p \, dx = 0\right\} \tag{3.7d}$$

It should be pointed out that the construction of the function space X_ϕ with the specified trace might pose technical difficulties, depending on the regularity

of the boundary $\partial\Omega$ and the subdivisions Γ_w and Γ_{io}. However, to emphasize the main points related to the coupling of the two physics and the resulting adjoint, we will assume that the boundary $\partial\Omega$ has sufficient regularity to enable the construction of X_ϕ. The weak formulation of the flow problem is:

$$\begin{aligned}&\text{Given } \phi \in H^1(\Omega), \text{ find } (\mathbf{u}, p) \in X_\phi \times M \text{ such that}\\ &\int_\Omega \left[\nabla \mathbf{u} \cdot \nabla \mathbf{v} - p\,\nabla \cdot \mathbf{v} - q\,\nabla \cdot \mathbf{u}\right] dx = 0 \quad \forall (\mathbf{v}, q) \in X_0 \times M\end{aligned} \tag{3.8}$$

Introducing the lift function $\mathbf{u}_l(\phi) \in X_\phi$, we may write $\mathbf{u} = \mathbf{w} + \mathbf{u}_l(\phi)$, where $\mathbf{w} \in X_0$, and reformulate Eq. (3.8) as:

$$\begin{aligned}&\text{Given } \phi \in H^1(\Omega), \text{ find } (\mathbf{w}, p) \in X_0 \times M \text{ such that}\\ &\int_\Omega \left[\nabla \mathbf{w} \cdot \nabla \mathbf{v} - p\,\nabla \cdot \mathbf{v} - q\,\nabla \cdot \mathbf{w}\right] dx\\ &\quad = -\int_\Omega \left[\nabla \mathbf{u}_l(\phi) \cdot \nabla \mathbf{v} - q\,\nabla \cdot \mathbf{u}_l(\phi)\right] dx \quad \forall (v, q) \in X_0 \times M\end{aligned} \tag{3.9}$$

Combining Eq. (3.5) and Eq. (3.9) together, we get the coupled variational statement:

$$\begin{aligned}&\text{Given } \sigma_c \in \mathrm{C}^+(\Omega) \text{ and } \phi_{io} \in \mathbb{R}, \text{ find } (\varphi, \mathbf{w}, p) \in Z_0 \times X_0 \times M \text{ such that}\\ &\int_\Omega \sigma_c \nabla\varphi \cdot \nabla\psi\, dx + \int_\Omega \left[\nabla[\mathbf{w} + \mathbf{u}_l(\varphi)] \cdot \nabla \mathbf{v} - p\,\nabla \cdot \mathbf{v} - q\,\nabla \cdot [\mathbf{w} + \mathbf{u}_l(\varphi)]\right] dx\\ &= -\int_\Omega \sigma_c \nabla\phi_{io} \cdot \nabla\psi\, dx - \int_\Omega \left[\nabla \mathbf{u}_l(\phi_{io}) \cdot \nabla \mathbf{v} - q\,\nabla \cdot \mathbf{u}_l(\phi_{io})\right] dx\\ &\forall (\psi, \mathbf{v}, q) \in Z_0 \times X_0 \times M\end{aligned} \tag{3.10}$$

where we emphasize that the lift velocity integrals associated with the Stokes equation depend on the solution φ and should be kept on the left-hand side of the equation. We can recast the bilinear form above in more compact notation

as,

$$\begin{aligned}&\text{Given } \sigma_c \in \mathrm{C}^+(\Omega) \text{ and } \phi_{io} \in \mathbb{R}, \text{ find } \mathbf{U} \in Z_0 \times X_0 \times M \text{ s.t.}\\ &A(\mathbf{U}, \mathbf{V}) = F(\mathbf{V}) \;\; \forall\, \mathbf{V} \in Z_0 \times X_0 \times M \end{aligned} \tag{3.11}$$

where $\mathbf{U} = (\varphi, \mathbf{w}, p)$ and $\mathbf{V} = (\psi, \mathbf{v}, q)$, and $A(\mathbf{U}, \mathbf{V})$ and $F(\mathbf{V})$ are the left- and right-hand sides of Eq. (3.10) respectively.

We have thus incorporated the coupling condition within our bilinear form and can now proceed to derive the adjoint problem.

3.2.2 Adjoint problem

Now, given the primal weak form (3.11), we have the corresponding weak form for the adjoint problem associated with the Quantity of Interest (QoI) $Q : Z_0 \times X_0 \times M \to \mathbb{R}$

$$\begin{aligned}&\text{Given } \sigma_c \in \mathrm{C}^+(\Omega) \text{ find } \mathbf{U}^* \in Z_0 \times X_0 \times M \text{ s.t.}\\ &A(\mathbf{V}, \mathbf{U}^*) = Q(\mathbf{V}) \;\; \forall\, \mathbf{V} \in Z_0 \times X_0 \times M \end{aligned} \tag{3.12}$$

where $\mathbf{U}^* = (\varphi^*, \mathbf{w}^*, p^*)$ is the adjoint solution and $Q(\mathbf{U})$ is a linear functional that prescribes a QoI. The full weak form for the adjoint problem reads,

$$\begin{aligned}&\int_\Omega \sigma \nabla\psi \cdot \nabla\varphi^* \, dx\\ &+ \int_\Omega (\nabla\mathbf{v} \cdot \nabla\mathbf{w}^* - q\, \nabla\cdot\mathbf{w}^* - p^*\, \nabla\cdot\mathbf{v}) \, dx\\ &+ \int_\Omega (\nabla\mathbf{u}_l(\psi) \cdot \nabla\mathbf{w}^* - \nabla\cdot\mathbf{u}_l(\psi)\, p^*) \, dx\\ &= Q(\mathbf{V}) \;\; \forall\, (\psi, \mathbf{v}, q) \in Z_0 \times X_0 \times M \end{aligned} \tag{3.13}$$

Following Eq. (3.7b) the term $\mathbf{u}_l(\psi)$ is defined as,

$$\mathbf{u}_l(\psi) \in \left\{ \mathbf{v} \in [H^1(\Omega)]^2, \mathbf{v} = (-\lambda \partial_t \psi, 0) \text{ on } \Gamma_w, \mathbf{v} \cdot \mathbf{t} = 0 \text{ on } \Gamma_{io} \right\} \tag{3.14}$$

We introduce the adjoint stress tensor for a Newtonian fluid (with unit parameters),

$$\boldsymbol{\sigma}^* = -p^* \mathbf{I} + (\nabla \mathbf{u}^* + \nabla^T \mathbf{u}^*) \tag{3.15}$$

Now integrating by parts 'backwards' we obtain,

$$\begin{aligned}
&\int_\Omega - \nabla \cdot (\sigma_c \nabla \varphi^*)\ \psi\ dx + \int_{\partial\Omega} \sigma_c \partial_n \varphi^*\ \psi\ ds \\
&+ \int_\Omega - \nabla \cdot \boldsymbol{\sigma}^* \cdot \Big(\mathbf{v} + \mathbf{u}_l(\psi) \Big) + \int_{\partial\Omega} \big[\mathbf{v} + \mathbf{u}_l(\psi) \big] \cdot (\boldsymbol{\sigma}^* \cdot \mathbf{n})\ ds \\
&+ \int_\Omega - q\ \nabla \cdot \mathbf{w}^*\ dx = Q(V)
\end{aligned} \tag{3.16}$$

The second boundary term in the formulation of the adjoint problem becomes

$$\begin{aligned}
&\int_{\partial\Omega} \big[\mathbf{v} + \mathbf{u}_l(\psi) \big] \cdot (\boldsymbol{\sigma}^* \cdot \mathbf{n})\ ds \\
&\quad = \int_{\Gamma_w} \big[\mathbf{v} + \mathbf{u}_l(\psi) \big] \cdot (\boldsymbol{\sigma}^* \cdot \mathbf{n})\ ds + \int_{\Gamma_{io}} \big[\mathbf{v} + \mathbf{u}_l(\psi) \big] \cdot (\boldsymbol{\sigma}^* \cdot \mathbf{n})\ ds \\
&\quad = \int_{\Gamma_w} \underbrace{\big(\mathbf{n} \cdot \big[\mathbf{v} + \mathbf{u}_l(\psi) \big] \big)}_{0} \big(\mathbf{n} \cdot (\boldsymbol{\sigma}^* \cdot \mathbf{n}) \big) + \underbrace{\big(\mathbf{t} \cdot \big[\mathbf{v} + \mathbf{u}_l(\psi) \big] \big)}_{-\lambda \partial_t \psi} \big(\mathbf{t} \cdot (\boldsymbol{\sigma}^* \cdot \mathbf{n}) \big)\ ds \\
&\quad + \int_{\Gamma_{io}} \big(\mathbf{n} \cdot \big[\mathbf{v} + \mathbf{u}_l(\psi) \big] \big) \big(\mathbf{n} \cdot (\boldsymbol{\sigma}^* \cdot \mathbf{n}) \big) + \underbrace{\big(\mathbf{t} \cdot \big[\mathbf{v} + \mathbf{u}_l(\psi) \big] \big)}_{0} \big(\mathbf{t} \cdot (\boldsymbol{\sigma}^* \cdot \mathbf{n}) \big)\ ds \\
&\quad = \int_{\Gamma_w} \big[\nabla_{\Gamma_w} \cdot \big(\mathbf{t} \cdot (\boldsymbol{\sigma}^* \cdot \mathbf{n}) \big) \mathbf{t} \big]\ \psi\ ds + \int_{\Gamma_{io}} \big(\mathbf{n} \cdot \big[\mathbf{v} + \mathbf{u}_l(\psi) \big] \big) \big(\mathbf{n} \cdot (\boldsymbol{\sigma}^* \cdot \mathbf{n}) \big)\ ds
\end{aligned} \tag{3.17}$$

where we have used integration by parts for the tangential derivative term along Γ_w [30],

$$\int_{\Gamma_w} - (\lambda \mathbf{t} \cdot \nabla \psi)(\mathbf{t} \cdot \mathbf{z})\ ds = \int_{\Gamma_w} \big[\nabla_{\Gamma_w} \cdot \big(\lambda \mathbf{t} \cdot \mathbf{z} \big) \mathbf{t} \big]\ \psi\ ds \tag{3.18}$$

with $\nabla_{\Gamma_w} \cdot \mathbf{v}$ denoting the surface divergence of vector $\mathbf{v}$. We replace these terms in the adjoint formulation and represent $Q(U)$ as follows,

$$\int_\Omega k(\mathbf{x})\ \mathbf{u} \cdot \boldsymbol{\alpha}\ dx + \int_{\partial\Omega} k_s(s)\ \mathbf{u} \cdot \mathbf{n}\ ds \tag{3.19}$$

where $k(\mathbf{x}) \in L^2(\Omega)$, $k_s(s) \in L^2(\partial\Omega)$, $\boldsymbol{\alpha} \in \mathbb{R}^2$. Eq. (3.17) can now be written as,

$$\begin{aligned}
&\int_\Omega \psi \left[-\nabla \cdot (\sigma_c \nabla \varphi^*) \right] dx + \int_{\Gamma_w} \psi \left[\mathbf{n} \cdot (\sigma_c \nabla \varphi^*) + \nabla_{\Gamma_w} \cdot \big((\lambda \mathbf{t} \cdot (\boldsymbol{\sigma}^* \cdot \mathbf{n})) \mathbf{t} \big) \right] ds \\
&+ \int_\Omega -\big[\mathbf{v} + \mathbf{u}_{\mathrm{w}}(\psi) \big] \cdot \big(\nabla \cdot \boldsymbol{\sigma}^* - k(\mathbf{x}) \alpha \big) \\
&+ \int_{\Gamma_{io}} \big(\mathbf{n} \cdot \big[\mathbf{v} + \mathbf{u}_{\mathrm{w}}(\psi) \big] \big) \big(\mathbf{n} \cdot (\boldsymbol{\sigma}^* \cdot \mathbf{n} - k_s(s)) \big)\ ds \\
&+ \int_\Omega -q \left[\nabla \cdot \mathbf{w}^* \right] dx = 0 \qquad \forall (\psi, \mathbf{v}, q) \in Z_0 \times X_0 \times M
\end{aligned} \tag{3.20}$$

The strong form of the adjoint system then reads:

$$-\nabla \cdot (\sigma_c \nabla \varphi^*) = 0 \qquad \text{in } \Omega \tag{3.21a}$$

$$-\Delta \mathbf{w}^* + \nabla p^* = k\boldsymbol{\alpha} \qquad \text{in } \Omega \tag{3.21b}$$

$$-\nabla \cdot \mathbf{w}^* = 0 \qquad \text{in } \Omega \tag{3.21c}$$

with three boundary conditions defined on Γ_{io}:

$$\varphi^* = 0 \qquad \text{on } \Gamma_{io} \tag{3.22a}$$

$$\mathbf{w}^* \cdot \mathbf{t} = 0 \qquad \text{on } \Gamma_{io} \tag{3.22b}$$

$$\mathbf{n} \cdot (\boldsymbol{\sigma}^* \cdot \mathbf{n}) = k_s \qquad \text{on } \Gamma_{io} \tag{3.22c}$$

and three boundary conditions on Γ_w:

$$\mathbf{n} \cdot (\sigma_c \nabla \varphi^*) + \nabla_{\Gamma_w} \cdot \big(\big(\lambda \mathbf{t} \cdot (\boldsymbol{\sigma}^* \cdot \mathbf{n}) \big) \mathbf{t} \big) = 0 \qquad \text{on } \Gamma_w \tag{3.23a}$$

$$\mathbf{w}^* = \mathbf{0} \qquad \text{on } \Gamma_w \tag{3.23b}$$

We readily observe that the adjoint Stokes problem can be solved first, independently of the adjoint potential problem, but that the latter does depend on the former through the Neumann coupling condition given by Eq. (3.23a). We also note that this coupling condition involves the tangential derivatives of the adjoint stress tensor on the boundary. The imposition of such a boundary condition can be extremely challenging, mainly due to the regularity requirements for the corresponding spaces in the interior and the difficulty of constructing appropriate Finite Element spaces. Therefore, we seek to impose the coupling constraint weakly and reduce the regularity requirements on the spaces containing the solution and the adjoint. As we shall see in the next sections, the penalty method is a natural method for weak enforcement of the coupling. In addition, the penalty formulation gives us an adjoint consistent with the one obtained using the lift technique.

3.3 Penalty formulation of the slip BC EOF model

3.3.1 Penalty formulation of the primal problem

The penalty method was introduced as an easy and robust approach for applying Dirichlet boundary conditions. Babuška [5] presented one of the first rigorous analyses of the technique. Further analysis was presented by Utku

and Carey [100] and the method was shown to be an effective alternative for applying boundary conditions. We consider here the penalty method for prescribing boundary conditions in the coupled flow system given by Eq. (2.15). The variational formulation given by Eq. (3.10) with equivalent penalty boundary conditions can be given as:

Given $\sigma_c \in \mathrm{C}^+(\Omega)$ and $\phi_{io} \in \mathbb{R}$, find $(\phi_\epsilon, \mathbf{u}_\epsilon, p_\epsilon) \in Z \times X \times M$ such that

$$\begin{aligned}
&\int_\Omega \sigma_c \nabla\phi_\epsilon \cdot \nabla\psi \; dx + \frac{1}{\epsilon}\int_{\Gamma_{io}} \phi_\epsilon \; \psi \; ds + \int_\Omega \left[\nabla\mathbf{u}_\epsilon \cdot \nabla\mathbf{v} - p_\epsilon \, \nabla\cdot\mathbf{v} - q \, \nabla\cdot\mathbf{u}_\epsilon\right] dx \\
&+ \frac{1}{\epsilon}\int_{\Gamma_w} (\mathbf{u}_\epsilon \cdot \mathbf{n}) \, (\mathbf{v}\cdot\mathbf{n}) \; ds + \frac{1}{\epsilon}\int_{\Gamma_w} (\mathbf{u}_\epsilon \cdot \mathbf{t}) \, (\mathbf{v}\cdot\mathbf{t}) \; ds + \frac{1}{\epsilon}\int_{\Gamma_{io}} (\mathbf{u}_\epsilon \cdot \mathbf{t}) \, (\mathbf{v}\cdot\mathbf{t}) \; ds \\
&+ \frac{1}{\epsilon}\int_{\Gamma_w} (\partial_t\phi_\epsilon) \, (\mathbf{v}\cdot\mathbf{t}) \; ds = \frac{1}{\epsilon}\int_{\Gamma_{io}} \phi_{io} \; \psi \; ds \quad \forall (\psi, \mathbf{v}, q) \in Z \times X \times M
\end{aligned} \tag{3.24}$$

where $\epsilon > 0$. We now verify that the weak form given by Eq. (3.24) is indeed consistent and converges to the BVP given by Eq. (3.1) and Eq. (3.6) in the limit as the penalty parameter ϵ tends to zero. Integrating Eq. (3.24) backwards by parts, we obtain,

$$\begin{aligned}
&\int_\Omega -\nabla\cdot(\sigma_c\nabla\phi_\epsilon)\,\psi \; dx + \int_{\Gamma_w} \sigma_c \partial_n\phi_\epsilon \, \psi \; ds + \int_{\Gamma_{io}} \Big(\sigma_c\partial_n\phi_\epsilon + \frac{1}{\epsilon}(\phi_\epsilon - \phi_{io})\Big)\,\psi \; ds \\
&+ \int_\Omega \Big((-\Delta\mathbf{u}_\epsilon + \nabla p_\epsilon)\cdot\mathbf{v} - q\,\nabla\cdot\mathbf{u}_\epsilon\Big) \; ds \\
&+ \int_{\Gamma_w} \Big(\mathbf{n}\cdot(\boldsymbol{\sigma}_\epsilon\cdot\mathbf{n}) + \frac{1}{\epsilon}(\mathbf{u}_\epsilon\cdot\mathbf{n})\Big)\,(\mathbf{v}\cdot\mathbf{n}) \; ds \\
&+ \int_{\Gamma_w} \Big(\mathbf{t}\cdot(\boldsymbol{\sigma}_\epsilon\cdot\mathbf{n}) + \frac{1}{\epsilon}(\partial_t\phi_\epsilon + \mathbf{u}_\epsilon\cdot\mathbf{t})\Big)\,(\mathbf{v}\cdot\mathbf{t}) \; ds + \int_{\Gamma_{io}} (\mathbf{n}\cdot(\boldsymbol{\sigma}_\epsilon\cdot\mathbf{n}))\,(\mathbf{v}\cdot\mathbf{n}) \; ds \\
&+ \int_{\Gamma_{io}} \Big(\mathbf{t}\cdot(\boldsymbol{\sigma}_\epsilon\cdot\mathbf{n}) + \frac{1}{\epsilon}(\mathbf{u}_\epsilon\cdot\mathbf{t})\Big)\,\mathbf{v}\cdot\mathbf{t} \; ds = 0
\end{aligned} \tag{3.25}$$

The equivalent strong form for finite non-zero ϵ is,

$$-\nabla \cdot (\sigma_c \nabla \phi_\epsilon) = 0 \qquad \text{in } \Omega \tag{3.26a}$$

$$-\Delta \mathbf{u}_\epsilon + \nabla p_\epsilon = \mathbf{0} \qquad \text{in } \Omega \tag{3.26b}$$

$$\nabla \cdot \mathbf{u}_\epsilon = 0 \qquad \text{in } \Omega \tag{3.26c}$$

with three boundary conditions defined on Γ_{io}:

$$\sigma_c \partial_n \phi_\epsilon + \frac{1}{\epsilon}(\phi_\epsilon - \phi_{io}) = 0 \qquad \text{on } \Gamma_{io} \tag{3.27a}$$

$$\mathbf{n} \cdot (\boldsymbol{\sigma}_\epsilon \cdot \mathbf{n}) = 0 \qquad \text{on } \Gamma_{io} \tag{3.27b}$$

$$\mathbf{t} \cdot (\boldsymbol{\sigma}_\epsilon \cdot \mathbf{n}) + \frac{1}{\epsilon}(\mathbf{u}_\epsilon \cdot \mathbf{t}) = 0 \qquad \text{on } \Gamma_{io} \tag{3.27c}$$

and three boundary conditions on Γ_w:

$$\sigma_c \partial_n \phi_\epsilon = 0 \qquad \text{on } \Gamma_w \tag{3.28a}$$

$$\mathbf{n} \cdot (\boldsymbol{\sigma}_\epsilon \cdot \mathbf{n}) + \frac{1}{\epsilon}(\mathbf{u}_\epsilon \cdot \mathbf{n}) = 0 \qquad \text{on } \Gamma_w \tag{3.28b}$$

$$\mathbf{t} \cdot (\boldsymbol{\sigma}_\epsilon \cdot \mathbf{n}) + \frac{1}{\epsilon}(\lambda \partial_t \phi_\epsilon + \mathbf{u}_\epsilon \cdot \mathbf{t}) = 0 \qquad \text{on } \Gamma_w \tag{3.28c}$$

One observes that the penalty method has the effect of replacing the true Dirichlet boundary conditions with a mixed Dirichlet-Neumann condition, which approximates the true Dirichlet conditions. However, upon taking the limit $\epsilon \to 0$ one formally recovers the original problems given by Eq. (3.1) and Eq. (3.6).

3.3.2 Adjoint problem associated with the penalty formulation

The weak form of the adjoint problem associated with problem Eq. (3.24) reads,

$$
\begin{aligned}
&\text{Find } (\phi_\epsilon^*, \mathbf{u}_\epsilon^*, p_\epsilon^*) \in Z \times X \times M \text{ such that} \\
&\int_\Omega \sigma_c \nabla \phi_\epsilon^* \cdot \nabla \psi \, dx + \frac{1}{\epsilon} \int_{\Gamma_{io}} \phi_\epsilon^* \, \psi \, ds \; + \frac{1}{\epsilon} \int_{\Gamma_w} \lambda \partial_t \psi_\epsilon \, (\mathbf{u}_\epsilon^* \cdot \mathbf{t}) \, ds \\
&+ \frac{1}{\epsilon} \int_{\Gamma_w} (\mathbf{u}_\epsilon^* \cdot \mathbf{n}) \, (\mathbf{v} \cdot \mathbf{n}) \, ds + \frac{1}{\epsilon} \int_{\Gamma_w} (\mathbf{u}_\epsilon^* \cdot \mathbf{t}) \, (\mathbf{v} \cdot \mathbf{t}) \, ds + \frac{1}{\epsilon} \int_{\Gamma_{io}} (\mathbf{u}_\epsilon^* \cdot \mathbf{t}) \, (\mathbf{v} \cdot \mathbf{t}) \, ds \\
&+ \int_\Omega (\nabla \mathbf{u}_\epsilon^* \cdot \nabla \mathbf{v} - p_\epsilon^* \, \nabla \cdot \mathbf{v} - q \, \nabla \cdot \mathbf{u}_\epsilon^*) \, dx \\
&= \int_\Omega k(\mathbf{x}) \, \mathbf{v} \cdot \boldsymbol{\alpha} \, dx + \int_{\partial\Omega} k_s(s) \, \mathbf{v} \cdot \mathbf{n} \, ds \quad \forall (\psi, \mathbf{v}, q) \in Z \times X \times M
\end{aligned} \tag{3.29}
$$

Using integration by parts for the term involving the tangential derivative along Γ_w [30], i.e.

$$
\int_{\Gamma_w} \lambda \partial_t \psi_\epsilon \, (\mathbf{u}_\epsilon^* \cdot \mathbf{t}) \, ds = - \int_{\Gamma_w} \nabla_{\Gamma_w} \cdot \big((\lambda \mathbf{u}_\epsilon^* \cdot \mathbf{t}) \mathbf{t} \big) \, \psi_\epsilon \, ds \tag{3.30}
$$

and upon integrating by parts the higher-order terms and combining integrals with same test functions, one obtains:

$$
\begin{aligned}
&\int_\Omega \Big(- \nabla \cdot (\sigma_c \nabla \phi_\epsilon^*) \Big) \, \psi \, dx + \int_{\Gamma_w} \Big(\sigma_c \partial_n \phi_\epsilon^* - \frac{1}{\epsilon} \nabla_{\Gamma_w} \cdot \big((\lambda \mathbf{u}_\epsilon^* \cdot \mathbf{t}) \mathbf{t} \big) \Big) \, \psi \, ds \\
&+ \int_{\Gamma_{io}} \Big(\sigma_c \partial_n \phi_\epsilon^* + \frac{1}{\epsilon} \phi_\epsilon^* \Big) \, \psi \, ds \\
&+ \int_\Omega \Big(- \Delta \mathbf{u}_\epsilon^* + \nabla p_\epsilon^* - k(\mathbf{x}) \boldsymbol{\alpha} \Big) \cdot \mathbf{v} \, dx + \int_\Omega \Big(- \nabla \cdot \mathbf{u}_\epsilon^* \Big) \, q \, dx \\
&+ \int_{\Gamma_w} \Big(\mathbf{n} \cdot (\boldsymbol{\sigma}_\epsilon \cdot \mathbf{n}) + \frac{1}{\epsilon} (\mathbf{u}_\epsilon^* \cdot \mathbf{n}) \Big) \, (\mathbf{v} \cdot \mathbf{n}) \, ds \\
&+ \int_{\Gamma_w} \Big(\mathbf{t} \cdot (\boldsymbol{\sigma}_\epsilon \cdot \mathbf{n}) + \frac{1}{\epsilon} (\mathbf{u}_\epsilon^* \cdot \mathbf{t}) \Big) \, (\mathbf{v} \cdot \mathbf{t}) \, ds
\end{aligned}
$$

$$
\begin{aligned}
&+ \int_{\Gamma_{io}} \Big(\mathbf{n} \cdot (\boldsymbol{\sigma}_\epsilon \cdot \mathbf{n}) - k_s(s) \Big) (\mathbf{v} \cdot \mathbf{n}) \, ds \\
&+ \int_{\Gamma_{io}} \Big(\mathbf{t} \cdot (\boldsymbol{\sigma}_\epsilon \cdot \mathbf{n}) + \frac{1}{\epsilon} (\mathbf{u}_\epsilon^* \cdot \mathbf{t}) \Big) (\mathbf{v} \cdot \mathbf{t}) \, ds = 0 \\
&\forall (\psi, \mathbf{v}, q) \in Z \times X \times M
\end{aligned} \tag{3.31}
$$

The equivalent strong form for finite non-zero ϵ is,

$$
-\Delta \mathbf{u}_\epsilon^* + \nabla p_\epsilon^* = k\boldsymbol{\alpha} \quad \text{in } \Omega \tag{3.32a}
$$

$$
-\nabla \cdot \mathbf{u}_\epsilon^* = 0 \quad \text{in } \Omega \tag{3.32b}
$$

$$
-\nabla \cdot (\sigma_c \nabla \phi_\epsilon^*) = 0 \quad \text{in } \Omega \tag{3.32c}
$$

with the three boundary conditions on Γ_{io}:

$$
\mathbf{n} \cdot (\boldsymbol{\sigma}_\epsilon^* \cdot \mathbf{n}) = k_s \quad \text{on } \Gamma_{io} \tag{3.33a}
$$

$$
\mathbf{t} \cdot (\boldsymbol{\sigma}_\epsilon^* \cdot \mathbf{n}) + \frac{1}{\epsilon} (\mathbf{u}_\epsilon^* \cdot \mathbf{t}) = 0 \quad \text{on } \Gamma_{io} \tag{3.33b}
$$

$$
\sigma_c \partial_n \phi_\epsilon^* + \frac{1}{\epsilon} \phi_\epsilon^* = 0 \quad \text{on } \Gamma_{io} \tag{3.33c}
$$

and the three boundary conditions on Γ_w:

$$
\mathbf{n} \cdot (\boldsymbol{\sigma}_\epsilon^* \cdot \mathbf{n}) + \frac{1}{\epsilon} (\mathbf{u}_\epsilon^* \cdot \mathbf{n}) = 0 \quad \text{on } \Gamma_w \tag{3.34a}
$$

$$
\mathbf{t} \cdot (\boldsymbol{\sigma}_\epsilon^* \cdot \mathbf{n}) + \frac{1}{\epsilon} (\mathbf{u}_\epsilon^* \cdot \mathbf{t}) = 0 \quad \text{on } \Gamma_w \tag{3.34b}
$$

$$
\sigma_c \partial_n \phi_\epsilon^* - \frac{1}{\epsilon} \nabla_{\Gamma_w} \cdot \big((\lambda \mathbf{u}_\epsilon^* \cdot \mathbf{t}) \mathbf{t} \big) = 0 \quad \text{on } \Gamma_w \tag{3.34c}
$$

In the next section, we show that above problem is consistent with the previous formulation of the adjoint problem, in the sense that we recover the adjoint corresponding to the strong problem given by Eq. (3.21), Eq. (3.22), and Eq. (3.23) as ϵ tends to zero.

3.3.3 Consistency of the adjoint penalty problem

The main issue is to ensure that the adjoint solution u_ϵ^* to the adjoint problem obtained from the penalized formulation does in fact converge to the adjoint solution u^* obtained from the primal formulation as the penalty parameter ϵ tends to zero, as illustrated in Figure 3.1. In this case, one has to show that the resulting boundary conditions associated with the penalized and non-penalized formulations of the adjoint problems are consistent. Recall that

$$\begin{array}{ccc} A(u,v) & \xrightarrow{\epsilon} & A_\varepsilon(u_\varepsilon, v) \\ \downarrow & & \downarrow \\ A(v,u^*) & \xleftarrow{?} & A_\varepsilon(v,u_\varepsilon^*) \end{array}$$

Figure 3.1: Consistency of the adjoint problems associated with the original and penalty formulations. The question here is whether the adjoint problem obtained from the penalty formulation converges to the adjoint problem derived from the original formulation in the limit when the penalty parameter ϵ tends to zero.

the non-penalized adjoint solution $(\phi^*, \mathbf{u}^*)$ for the problem of interest satisfies the following boundary conditions

$$\phi^* = 0 \qquad \text{on } \Gamma_{io} \tag{3.35a}$$

$$\mathbf{n} \cdot (\boldsymbol{\sigma}^* \cdot \mathbf{n}) = k \qquad \text{on } \Gamma_{io} \tag{3.35b}$$

$$\mathbf{u}^* \cdot \mathbf{t} = 0 \qquad \text{on } \Gamma_{io} \tag{3.35c}$$

$$\mathbf{u}^* = \mathbf{0} \qquad \text{on } \Gamma_w \tag{3.35d}$$

$$\sigma_c \partial_n \phi_\epsilon^* + \nabla_{\Gamma_w} \cdot \big(\lambda \mathbf{t} \cdot (\boldsymbol{\sigma}^* \cdot \mathbf{n})\mathbf{t}\big) = 0 \qquad \text{on } \Gamma_w \tag{3.35e}$$

while the penalized adjoint solution $(\phi_\epsilon^*, \mathbf{u}_\epsilon^*)$ satisfies,

$$\sigma_c \partial_n \phi_\epsilon^* + \frac{1}{\epsilon}\phi^* = 0 \text{ on } \Gamma_{io} \tag{3.36a}$$

$$\mathbf{n} \cdot (\boldsymbol{\sigma}_\epsilon^* \cdot \mathbf{n}) = k \text{ on } \Gamma_{io} \tag{3.36b}$$

$$\mathbf{t} \cdot (\boldsymbol{\sigma}_\epsilon^* \cdot \mathbf{n}) + \frac{1}{\epsilon}(\mathbf{u}_\epsilon^* \cdot \mathbf{t}) = 0 \text{ on } \Gamma_{io} \tag{3.36c}$$

$$\mathbf{t} \cdot (\boldsymbol{\sigma}_\epsilon^* \cdot \mathbf{n}) + \frac{1}{\epsilon}(\mathbf{u}_\epsilon^* \cdot \mathbf{t}) = 0 \text{ on } \Gamma_{w} \tag{3.36d}$$

$$\mathbf{n} \cdot (\boldsymbol{\sigma}_\epsilon^* \cdot \mathbf{n}) + \frac{1}{\epsilon}(\mathbf{u}_\epsilon^* \cdot \mathbf{n}) = 0 \text{ on } \Gamma_{w} \tag{3.36e}$$

$$\sigma_c \partial_n \phi_\epsilon^* - \frac{1}{\epsilon}\nabla_{\Gamma_w} \cdot \big((\lambda \mathbf{u}_\epsilon^* \cdot \mathbf{t})\mathbf{t}\big) = 0 \text{ on } \Gamma_{w} \tag{3.36f}$$

To formally interpret Eq. (3.36f), we can substitute Eq. (3.36d) into Eq. (3.36f) as follows,

$$\begin{aligned} &\mathbf{t} \cdot (\boldsymbol{\sigma}_\epsilon^* \cdot \mathbf{n}) + \frac{1}{\epsilon}(\mathbf{u}_\epsilon^* \cdot \mathbf{t}) = 0 \;\Rightarrow\; \frac{\lambda}{\epsilon}(\mathbf{u}_\epsilon^* \cdot \mathbf{t}) = -\lambda \mathbf{t} \cdot (\boldsymbol{\sigma}_\epsilon^* \cdot \mathbf{n}) \\ &\sigma_c \partial_n \phi_\epsilon^* = \nabla_{\Gamma_w} \cdot \left(\left(\frac{\lambda \mathbf{u}_\epsilon^* \cdot \mathbf{t}}{\epsilon} \right) \mathbf{t} \right) \;\Rightarrow\; \sigma_c \partial_n \phi_\epsilon^* = -\nabla_{\Gamma_w} \cdot (\lambda \mathbf{t} \cdot (\boldsymbol{\sigma}_\epsilon^* \cdot \mathbf{n})\mathbf{t}) \end{aligned} \tag{3.37}$$

We can derive the following boundary conditions for the adjoint potential,

$$\sigma_c \partial_n \phi_\epsilon^* + \nabla_{\Gamma_w} \cdot \big(\lambda \mathbf{t} \cdot (\boldsymbol{\sigma}_\epsilon^* \cdot \mathbf{n})\mathbf{t}\big) = 0 \quad \text{on } \Gamma_w \tag{3.38a}$$

$$\phi_\epsilon^* + \epsilon\, \sigma_c \partial_n \phi_\epsilon^* = 0 \quad \text{on } \Gamma_{io} \tag{3.38b}$$

These boundary conditions are consistent with the non-penalized forms in the limit $\epsilon \to 0$. Equation (3.36d) corresponds to a penalty representation of the tangential boundary flux. Further discussion of this representation will be presented in chapter 6. We thus see that the penalized formulation of the electroosmotic flow problem is adjoint-consistent.

3.4 The Slip Boundary Condition and Well-Posedness

In the last two sections, we have studied a special slip boundary condition that may not have the regularity one can usually assume for Dirichlet data. We have also seen that the construction of an adjoint problem for such a problem can pose challenges. Adjoint methods usually assume that the boundary condition is given data and as such they do not enter the picture for the adjoint problem. For the models under consideration in this work however, estimating the error due to the coupling is critical, since such errors can be large [37, 22]. Also, without incorporating the coupling condition in the adjoint, adjoint sensitivity analysis cannot include the sensitivity of quantities in the fluid physics to parameters in the electrostatic physics.

These considerations necessitate the inclusion of the boundary coupling condition in the variational form, where it can naturally enter the adjoint problem and the regularity requirements on it can be reduced. In Finite Element analysis, many strategies exist for the weak imposition of boundary conditions: Lagrange multiplier methods, the Nitsche method, and penalty methods. A survey of such methods, their benefits, and limitations is given in a review article by Babuška et al. [6]. As we have seen in sections 3.3 and 3.3.3, penalty methods can offer a natural methodology to achieve our goals. We now seek to put that choice on a firmer theoretical ground. In the next section, we will discuss a natural projection operation that offers the possibility of smoothing irregular data. We will point out why the natural operation is not well-posed and cannot be a projection. Then, in section 3.4.2 we will discuss how the op-

eration can be made well-posed by introducing a penalty term. We will then extend these ideas to operations on the boundary and show how the penalty method can be seen as a projection operation that regularizes ill-posed boundary data.

Finally, in section 3.4.3 we will be able to show the well-posedness of a modified version of the slip model, and show why modifying the original slip boundary condition in section 2.4 is essential if we want to obtain a well-posed adjoint problem.

3.4.1 Smoothing Interior Data

Consider a function $g \in L^2(\Omega)$. Suppose now that we need to smooth this data and use its projection in $H^1(\Omega) \subset L^2(\Omega)$ as data for another problem. A natural projection operator $\pi_\Omega : L^2(\Omega) \to H^1(\Omega)$, $\pi_\Omega(g) = u$ may be defined as,

$$\text{Find } u \in H^1(\Omega) \text{ s.t. } \int_\Omega u\, v\, dx = \int_\Omega g\, v\, dx \quad \forall v \in H^1(\Omega) \tag{3.39}$$

Unfortunately, this mapping is not well-posed. Recall that projections can only be made to complete subspaces of $L^2(\Omega)$ and that the $H^1(\Omega)$ subspace is not complete with respect to the L^2 norm. However, one can make this operation well-posed by introducing a regularization,

$$\begin{aligned} &\text{Given } \epsilon > 0, \text{ find } u \in H^1(\Omega) \text{ s.t.} \\ &\epsilon \int_\Omega \nabla u_\epsilon \cdot \nabla v\, dx + \int_\Omega u_\epsilon\, v\, dx = \int_\Omega g\, v\, dx \quad \forall v \in H^1(\Omega) \end{aligned} \tag{3.40}$$

This operation is well-posed, it is in fact the weak form for an elliptic PDE when complemented with boundary conditions. Note that this operation is not a true projection: u_ϵ will not be the best approximation to g in $H^1(\Omega)$, but for small values of ϵ, we anticipate u_ϵ to be close to the true projection. The stability of this operation depends on the parameter ϵ and reflects a choice between stability and projection accuracy. Eq. (3.40) thus offers us a method of smoothing data that lacks regularity. We now discuss a similar operation on the boundary and show that it is equivalent to the penalty method for imposing boundary conditions.

3.4.2 Smoothing Boundary Data

Consider a function $g \in H^{-\frac{1}{2}}(\partial\Omega)$. Suppose now that one needs to smooth this data and use its projection in $H^{\frac{1}{2}}(\partial\Omega) \subset H^{-\frac{1}{2}}(\Omega)$ as data for another problem. For reasons similar to those discussed in the previous section, the natural mapping, $\pi_{\partial\Omega} : H^{-\frac{1}{2}}(\Omega) \to H^{\frac{1}{2}}(\partial\Omega)$, $\pi_{\partial\Omega}(g) = u$ defined as,

$$\text{Find } u \in H^{\frac{1}{2}}(\partial\Omega) \text{ s.t. } \int_{\partial\Omega} u\, v\, ds = \int_{\partial\Omega} g\, v\, ds \quad \forall\, v \in H^{\frac{1}{2}}(\partial\Omega) \tag{3.41}$$

is ill-posed. However, consider the modified mapping: Given $\epsilon > 0$, find $u_\epsilon \in H^{\frac{1}{2}}(\partial\Omega)$ such that,

$$\epsilon \int_{\partial\Omega} \partial_n u_\epsilon\, v\, ds + \int_{\partial\Omega} u_\epsilon\, v\, ds = \int_{\partial\Omega} g\, v\, ds \quad \forall\, v \in H^{\frac{1}{2}}(\partial\Omega)$$

The solution u_ϵ would approximate an $H^{\frac{1}{2}}(\partial\Omega)$ projection of the Dirichlet data g. The question now is whether this operation is well-posed. If we

denote the extension of u into Ω as $\widehat{u}$, we can write Eq. (3.42) as,

$$\begin{aligned}&\int_{\Omega} -\epsilon\ \Delta\widehat{u}\ v\ ds + \int_{\Omega} \epsilon\ \nabla\widehat{u}\cdot\nabla v\ dx + \int_{\partial\Omega} u\ v\ ds \\ &= \int_{\partial\Omega} g\ v\ ds = F_2(v) \quad \forall\ v \in H^{\frac{1}{2}}(\partial\Omega)\end{aligned} \tag{3.42}$$

Note that $\widehat{u}$ is not unique. However, if we require that it satisfy the constraint $-\Delta\widehat{u} = f$ in the interior, where $f \in L^2(\Omega)$, then $\widehat{u}$ is indeed unique. We can thus write $\widehat{u}$ as u. The weak form given by Eq. (3.42) thus becomes,

$$\begin{aligned}&\text{Find } u \in H^{\frac{1}{2}}(\partial\Omega) \text{ s.t.} \int_{\Omega} \epsilon\ \nabla u\cdot\nabla v\ dx + \int_{\partial\Omega} u\ v\ ds \\ &= \int_{\partial\Omega} g\ v\ ds + \int_{\Omega} \epsilon\ f\ v\ ds \quad \forall\ v \in H^{\frac{1}{2}}(\partial\Omega)\end{aligned} \tag{3.43}$$

The question now is whether this operation is well-posed, and Theorem 3.4.1 confirms that this is indeed the case.

Theorem 3.4.1. *The bilinear form, $B : H^{\frac{1}{2}}(\partial\Omega) \times H^{\frac{1}{2}}(\partial\Omega) \to \mathbb{R}$ given by*

$$B(u,v) = \int_{\Omega} \epsilon\ \nabla u_\epsilon\cdot\nabla v\ dx + \int_{\partial\Omega} u_\epsilon\ v\ ds \tag{3.44}$$

where $\Omega \subset \mathbb{R}^2$ has a Lipschitz boundary, and $\epsilon > 0$, is bounded and coercive.

Proof. First we show boundedness,

$$\begin{aligned}B(u,v) &= \epsilon \int_{\Omega} \nabla u_\epsilon\cdot\nabla v\ dx + \int_{\partial\Omega} u_\epsilon\ v\ ds \\ &\leq \epsilon |u_\epsilon|_{H^1(\Omega)}\ |v|_{H^1(\Omega)} + \|u_\epsilon\|_{L^2(\partial\Omega)}\ \|v\|_{L^2(\partial\Omega)} \\ &\leq \epsilon |u_\epsilon|_{H^1(\Omega)}\ |v|_{H^1(\Omega)} + \|u_\epsilon\|_{H^1(\Omega)}\ \|v\|_{H^1(\Omega)} \\ &\leq (1+\epsilon)\|u_\epsilon\|_{H^1(\Omega)}\ \|v\|_{H^1(\Omega)}\end{aligned}$$

$$\leq C_1(1+\epsilon)\|u_\epsilon\|_{H^{\frac{1}{2}}(\partial\Omega)} \|v\|_{H^{\frac{1}{2}}(\partial\Omega)} \qquad \text{(Corollary B.53, Pg. 488 [36])} \tag{3.45}$$

Now for the coercivity,

$$B(u,u) = \epsilon|u_\epsilon|^2_{H^1(\Omega)} + \|u_\epsilon\|^2_{L^2(\partial\Omega)} \tag{3.46}$$

Again, corollary B.53, page 488 from [36] states that if $1 \leq p < \infty$ and $\frac{1}{p} + \frac{1}{p'} = 1$, there exists a constant c such that, $\forall\ u \in W^{\frac{1}{p'},p}(\partial\Omega)$, one can find an extension $u \in W^{1,p}(\Omega)$ which satisfies,

$$\|u_\epsilon\|_{W^{1,p}(\Omega)} \leq c \|u_\epsilon\|_{W^{\frac{1}{p'},p}(\partial\Omega)} \tag{3.47}$$

If we denote the measure of $\partial\Omega$ by $\mu(\partial\Omega)$ and choose $p = 1$, $p^{'} = \infty$, we have,

$$\|u_\epsilon\|_{W^{1,1}(\Omega)} \leq c \|u_\epsilon\|_{W^{0,1}(\partial\Omega)} = c \|u_\epsilon\|_{L^1(\partial\Omega)} \leq c \sqrt{\mu(\partial\Omega)} \|u_\epsilon\|_{L^2(\partial\Omega)} \tag{3.48}$$

By Corollary B.43, page 486 in [36], $W^{1,1}(\Omega) \subset L^2(\Omega)$ continuously,

$$\|u_\epsilon\|_{L^2(\Omega)} \leq C_2 \|u_\epsilon\|_{W^{1,1}(\Omega)} \tag{3.49}$$

Therefore we have,

$$\|u_\epsilon\|_{L^2(\Omega)} \leq c\, C_2 \sqrt{\mu(\partial\Omega)} \|u_\epsilon\|_{L^2(\partial\Omega)} \tag{3.50}$$

And Eq. (3.46) can now be written as,

$$\begin{aligned} B(u,u) &\geq \epsilon|u_\epsilon|^2_{H^1(\Omega)} + \frac{1}{(c\sqrt{\mu(\partial\Omega)})}\|u_\epsilon\|^2_{L^2(\Omega)} \\ &\geq \min\left(\epsilon, \frac{1}{(C_2\sqrt{\mu(\partial\Omega)})}\right)\|u_\epsilon\|^2_{H^1(\Omega)} \\ &\geq \frac{1}{C_3}\ \min\left(\epsilon, \frac{1}{(C_2\sqrt{\mu(\partial\Omega)})}\right)\|u_\epsilon\|^2_{H^{\frac{1}{2}}(\partial\Omega)} \text{ (Theorem B.52, Pg. 488 [36])} \end{aligned}$$

□

We identify Eq. (3.43) with the weak formulation of a Poisson problem with boundary conditions g enforced using the penalty method. Therefore, the penalty 'regularizes' the forward problem with irregular data via this smoothing operation. The solution we obtain by solving Eq. (3.43) will thus satisfy a regularized constraint.

3.4.3 Well-Posedness of the Penalty Formulation

In the previous section, we discussed how a weak enforcement of an irregular boundary condition corresponds to a smoothing operation. Based on that discussion, we now derive some theoretical results for the penalty formulation of the microfluidics problem. In all the statements below we assume that the domain Ω is Lipschitz and the conductivity $\sigma_c \in \mathrm{C}^+(\Omega)$.

Analysis for a decoupled version of the microfludics model

Theorem 3.4.2. *The variational problems*

$$\begin{aligned} &\textit{Given } \phi_{io} \in \mathbb{R}, \textit{ find } \phi_\epsilon \in Z \textit{ such that} \\ &\mathcal{B}_\phi(\phi_\epsilon, \psi) = \int_\Omega \sigma_c \nabla\phi_\epsilon \cdot \nabla\psi \, dx + \frac{1}{\epsilon}\int_{\Gamma_{io}} \phi_\epsilon \, \psi \, ds = \frac{1}{\epsilon}\int_{\Gamma_{io}} \phi_{io} \, \psi \, ds \\ &= \mathcal{F}_\phi(\psi) \quad \forall \psi \in Z \end{aligned} \tag{3.51}$$

and

$$\begin{aligned} &\textit{Given } \phi_\epsilon, \textit{ find } (\mathbf{u}_\epsilon, p_\epsilon) \in X \times M \textit{ such that} \\ &\mathcal{B}_{\mathbf{u}}(\mathbf{u}_\epsilon, \mathbf{v}) - \mathcal{B}_p(\mathbf{v}, p_\epsilon) - \mathcal{B}_p(\mathbf{u}, q) = \int_\Omega \nabla\mathbf{u}_\epsilon \cdot \nabla\mathbf{v} \, dx + \frac{1}{\epsilon}\int_{\Gamma_w} (\mathbf{u}_\epsilon \cdot \mathbf{n})\,(\mathbf{v} \cdot \mathbf{n}) \, ds \end{aligned}$$

$$
\begin{aligned}
&+\frac{1}{\epsilon}\int_{\Gamma_w}(\mathbf{u}_\epsilon\cdot\mathbf{t})\,(\mathbf{v}\cdot\mathbf{t})\,ds+\frac{1}{\epsilon}\int_{\Gamma_{io}}(\mathbf{u}_\epsilon\cdot\mathbf{t})\,(\mathbf{v}\cdot\mathbf{t})\,ds\\
&-\int_\Omega p_\epsilon\,\nabla\cdot\mathbf{v}\,dx-\int_\Omega q\,\nabla\cdot\mathbf{u}_\epsilon\,dx=\frac{1}{\epsilon}\int_{\Gamma_w}\lambda(s)\,(-\partial_t\phi_\epsilon)\,(\mathbf{v}\cdot\mathbf{t})\,ds\\
&=\mathcal{F}_{\mathbf{u}}((\mathbf{v},q))\quad\forall(\mathbf{v},q)\in X\times M
\end{aligned}
\tag{3.52}
$$

where $\lambda(s)$ is a smooth function that is zero on Γ_{io}, $\epsilon>0$, and the function spaces Z, X and M are as specified in Eq. (3.2a) and Eq. (3.7), are well-posed and the solution $\mathbf{U}_\epsilon=(\phi_\epsilon,\mathbf{u}_\epsilon,p_\epsilon)$ is unique and bounded.

Proof. The bilinear form $\mathcal{B}_\phi(\phi_\epsilon,\psi)$ is clearly bounded and coercive. The right-hand side $\mathcal{F}_\phi(\psi)$ is also clearly bounded. Thus, by the Lax-Milgram theorem the variational problem Eq. (3.51) is well-posed and has a unique and bounded solution.

The bilinear forms $\mathcal{B}_{\mathbf{u}}(\mathbf{u}_\epsilon,\mathbf{v})$ and $\mathcal{B}_p(\mathbf{v},p_\epsilon)$ are bounded and satisfy an inf-sup property [36]. The right-hand side $\mathcal{F}_{\mathbf{u}}((\mathbf{v},q))$ is also bounded since,

$$
\begin{aligned}
\mathcal{F}_{\mathbf{u}}((\mathbf{v},q))&=\frac{1}{\epsilon}\int_{\Gamma_w}\lambda(s)\,(-\partial_t\phi_\epsilon)\,(\mathbf{v}\cdot\mathbf{t})\,ds\\
&=\frac{1}{\epsilon}\int_{\partial\Omega}\lambda(s)\,(-\partial_t\phi_\epsilon)\,(\mathbf{v}\cdot\mathbf{t})\,ds\\
&\leq\frac{1}{\epsilon}\|\lambda(s)\|_\infty\,\|\phi_\epsilon\|_{H^{-\frac{1}{2}}(\partial\Omega)}\,\|\mathbf{v}\|_{H^{\frac{1}{2}}(\partial\Omega)}
\end{aligned}
$$

The weak tangential derivative on the boundary is well defined for all functions in $H^1(\Omega)$. This can be seen by an application of the curl theorem [91]. Thus $\mathcal{F}_{\mathbf{u}}\in X^*$ and the problem Eq. (3.52) is well-posed. □

We have a similar theorem and proof for the corresponding adjoint problem.

Theorem 3.4.3. *The variational problems,*

Given k, k_s and $\boldsymbol{\alpha}$ as in Eq. (3.19) find $(\mathbf{u}_\epsilon^*, p_\epsilon^*) \in X \times M$ such that

$$\mathcal{B}_{\mathbf{u}}^*(\mathbf{v}, \mathbf{u}_\epsilon^*) - \mathcal{B}_p^*(q_\epsilon, \mathbf{u}_\epsilon^*) - \mathcal{B}_p^*(p_\epsilon^*, \mathbf{v}) = \int_\Omega \nabla \mathbf{u}_\epsilon^* \cdot \nabla \mathbf{v} + \frac{1}{\epsilon}\int_{\Gamma_w} (\mathbf{u}_\epsilon^* \cdot \mathbf{n})\,(\mathbf{v} \cdot \mathbf{n})\; ds$$
$$+ \frac{1}{\epsilon}\int_{\Gamma_w} (\mathbf{u}_\epsilon^* \cdot \mathbf{t})\,(\mathbf{v} \cdot \mathbf{t})\; ds + \frac{1}{\epsilon}\int_{\Gamma_{io}} (\mathbf{u}_\epsilon^* \cdot \mathbf{t})\,(\mathbf{v} \cdot \mathbf{t})\; ds$$
$$- \int_\Omega p_\epsilon^*\, \nabla \cdot \mathbf{v}\; dx - \int_\Omega q\, \nabla \cdot \mathbf{u}_\epsilon^*\; dx = \int_\Omega k(x)\, \boldsymbol{\alpha} \cdot \mathbf{v}\; dx + \int_{\Gamma_{io}} k_s(s)\; \mathbf{v} \cdot \mathbf{n}\; ds$$
$$= \mathcal{F}_{\mathbf{u}}^*((\mathbf{v}, q)) \quad \forall (\mathbf{v}, q) \in X \times M \tag{3.53}$$

and

Given $\mathbf{u}_\epsilon^*$, find $\phi_\epsilon \in Z$ such that

$$\mathcal{B}_\phi^*(\psi, \phi_\epsilon^*) = \int_\Omega \sigma_c \nabla \phi_\epsilon^* \cdot \nabla \psi\; dx + \frac{1}{\epsilon}\int_{\Gamma_{io}} \phi_\epsilon^*\; \psi\; ds = -\frac{1}{\epsilon}\int_{\Gamma_w} \lambda(s)\, \partial_t \psi_\epsilon\, (\mathbf{u}_\epsilon^* \cdot \mathbf{t})\; ds$$
$$= \mathcal{F}_\phi^*(\psi) \quad \forall \psi \in Z \tag{3.54}$$

where $\lambda(s)$ is a smooth function that is zero on Γ_{io}, $\epsilon > 0$ and the function spaces Z, X and M are as specified in Eq. (3.2a) and Eq. (3.7), are well-posed and the solution $\mathbf{U}_\epsilon^* = (\phi_\epsilon^*, \mathbf{u}_\epsilon^*, p_\epsilon^*)$ is unique and bounded.

Remark 3.4.1. As mentioned before an application of the curl theorem shows that the weak tangential derivative on the boundary is well defined for functions in $H^1(\Omega)$ [91]. However, the weak normal derivative is only well defined for [91],

$$H^1_\Delta(\Omega) = \{u \in H^1(\Omega) : \Delta u \in L^2(\Omega)\} \tag{3.55}$$

For the microfluidics model, the primal potential does indeed lie in $H^1_\Delta(\Omega)$ and hence coupling the normal component weakly in the forward problem can

be justified. However, we only require that the potential lie in $H^1(\Omega)$ and correspondingly choose $H^1(\Omega)$ as the test space. If we were to couple both the tangential and normal components of the potential gradient to the velocity, the weak form of the adjoint problem for the potential ϕ_ϵ^* would then read,

$$\begin{aligned}
&\text{Given } \mathbf{u}_\epsilon^* \in H^1(\Omega), \text{ find } \ \phi_\epsilon^* \in Z \text{ such that} \\
&\mathcal{B}_\phi^*(\psi, \phi_\epsilon^*) = \int_\Omega \sigma_c \nabla \phi_\epsilon^* \cdot \nabla \psi \, dx + \frac{1}{\epsilon} \int_{\Gamma_{io}} \phi_\epsilon^* \, \psi \, ds \\
&= -\frac{1}{\epsilon} \int_{\Gamma_w} \lambda(s) \, \partial_t \psi \, (\mathbf{u}_\epsilon^* \cdot \mathbf{t}) \, ds - \frac{1}{\epsilon} \int_{\Gamma_w} \lambda(s) \, \partial_n \psi \, (\mathbf{u}_\epsilon^* \cdot \mathbf{n}) \, ds \\
&= \mathcal{F}_\phi^*(\psi) \quad \forall \psi \in H^1(\Omega)
\end{aligned} \tag{3.56}$$

Since the weak normal derivative is well defined only for $\psi \in H^1_\Delta(\Omega)$, the right-hand side of the adjoint problem is not bounded. Hence, once cannot couple the normal component of the adjoint potential to the adjoint velocity without requiring that the primal solution lie in the space $H^1_\Delta(\Omega)$ and testing with the corresponding test functions. See also remark 6.3.3.

Chapter 4

Implementation of Adjoint Techniques in libMesh

4.1 Introduction

In this chapter, we will discuss the theory underlying the use of adjoint methods in adaptive mesh refinement and sensitivity analysis. An inexpensive and easy method to compute adjoint-based error indicators will be derived. Adjoint sensitivity analysis will also be discussed and the relevant theory will be developed. We shall then discuss the implementation of these adjoint-based methods in the C++ Finite Element library `libMesh`. The software architecture and class structure used to enable such adjoint functionality will be described. We will conclude the chapter with verification studies for the new adjoint methods in `libMesh`.

4.2 Adjoint Residual based Error Indicators

Adjoint-based methods for adaptive refinement and error estimation are being increasingly used in Finite Element analysis [44, 79, 13]. Adjoint-based methods for goal-oriented error estimation can be viewed as an extension of the concepts of Green's functions [39]. Such methods can be generalized to many

types of applications, including coupled systems analysis [22, 59], multiscale systems analysis [11, 37], boundary QoIs [105], and non-linear problems [79, 39]. The Adjoint Residual Error Indicator implemented in `libMesh` provides a robust, inexpensive, and easily computable error indicator to guide adaptive mesh refinement. We begin with the statement and proof of a theorem that underlies the Adjoint Residual Error Indicator.

Theorem 4.2.1. *Consider the variational problem,*

$$\text{Given } \mathcal{R} : U \times V \to \mathbb{R}, \text{ find } u \in U \text{ s.t. } \mathcal{R}(u; v) = 0 \quad \forall\, v \in V \tag{4.1}$$

where U and V are Hilbert spaces. The adjoint solution z satisfies,

$$\text{Given } Q : U \to \mathbb{R}, \text{ find } z \in V \text{ s.t. } \frac{\partial \mathcal{R}}{\partial u}(u; v, z) = \frac{\partial Q}{\partial u}(u; v) \quad \forall\, v \in V \tag{4.2}$$

where $\frac{\partial}{\partial u}$ is the Fréchet derivative w.r.t. u. Let u_h and z_h denote the discrete approximations to u and z in the subspaces U_h and V_h, obtained by solving,

$$\mathcal{R}(u_h, v_h) = 0 \quad \forall\, v_h \in V_h \tag{4.3}$$

$$\frac{\partial \mathcal{R}}{\partial u}(u_h; v_h, z_h) = \frac{\partial Q}{\partial u}(u_h; v_h) \quad \forall\, v_h \in V_h \tag{4.4}$$

Further, let U^B be a neighbourhood of u s.t. both Q and $\mathcal{R}$ are bounded there with bounded derivatives Q_u and $\mathcal{R}_u$. Then, we have the following estimate,

$$Q(u) - Q(u_h) = \mathcal{R}_u(u_h; z - z_h)(u - u_h) + R_Q - R_{\mathcal{R}} \tag{4.5}$$

where,

$$\lim_{\|u-u_h\| \to 0} \frac{\|R_Q\|}{\|u - u_h\|} = 0 \text{ and } \lim_{\|u-u_h\| \to 0} \frac{\|R_{\mathcal{R}}\|}{\|u - u_h\|} = 0. \tag{4.6}$$

Proof. Since $u_h \to u$, we have,

$$\exists\ \bar{h} > 0,\ \text{s.t.}\ \forall\, h < \bar{h},\ u_h \in U^B \tag{4.7}$$

Now, by the definition of the Fréchet derivative,

$$Q(u) - Q(u_h) = Q_u(u_h)(u - u_h) + R_Q \tag{4.8}$$

By definition of the adjoint we have,

$$Q_u(u_h)(u - u_h) = \mathcal{R}_u(u_h; z)(u - u_h) \tag{4.9}$$

Substituting in Eq. (4.8), we obtain,

$$Q(u) - Q(u_h) = \mathcal{R}_u(u_h; z)(u - u_h) + R_Q \tag{4.10}$$

Again by definition of the Fréchet derivative, we have,

$$\mathcal{R}(u, z) - \mathcal{R}(u_h, z) = \mathcal{R}_u(u_h; z)(u - u_h) + R_{\mathcal{R}} \tag{4.11}$$

Substituting Eq. (4.11) in Eq. (4.10) and using Galerkin orthogonality we obtain,

$$\begin{aligned} Q(u) - Q(u_h) &= \mathcal{R}(u, z) - \mathcal{R}(u_h, z) + R_Q - R_{\mathcal{R}} && \text{(4.12a)} \\ &= \mathcal{R}(u, z - z_h) - \mathcal{R}(u_h, z - z_h) + R_Q - R_{\mathcal{R}} \\ &= \mathcal{R}_u(u_h; z - z_h)(u - u_h) + R_Q - R_{\mathcal{R}} && \text{(4.12b)} \end{aligned}$$

which completes the proof. □

If one neglects the higher order terms, a computable error indicator can be obtained using either Eq. (4.12a) or Eq. (4.12b). If we were to use Eq. (4.12a), then we would need to compute the adjoint on an enriched space, and use $\mathcal{R}(u_h, z - z_h)$ to form our error estimate. Eq. (4.12b) can also be used to obtain a computable error indicator for adaptive mesh refinement as follows,

$$\begin{aligned} |Q(u) - Q(u_h)| &\leq |\mathcal{R}_u(u_h, z - z_h)(u - u_h)| + \text{H.O.T.} \\ &\leq \sum_{i=1}^{N_{el}} \|\mathcal{R}_u\|_i \|z - z_h\|_i \|u - u_h\|_i + \text{H.O.T.} \end{aligned} \tag{4.13}$$

Thus, we can compute an element-wise error indicator of the form,

$$\tilde{e}_K = (\text{Error bound for primal}) \times (\text{Error bound for adjoint}) \tag{4.14}$$

In `libMesh`, this method is called the Adjoint Residual Error Indicator. Existing error indicators such as the uniform error estimator [31] or the flux-jump error estimator [54] can give us estimates for the errors in both the primal and adjoint solutions. Patch-recovery methods [109], which bound the interpolation error, can also be used as error indicators. The existing error indicator infrastructure in `libMesh` already contains implementations for the uniform, flux-jump and patch-recovery indicators. In section 4.4.3, we will show how this infrastructure was leveraged to implement the Adjoint Residual Error Indicator. Section 4.4 discusses the verification of this new indicator in `libMesh`.

4.2.1 Multiphysics Problems

Eq. (4.13) needs to be generalized for use in multivariable and multi-physics problems. For example, consider the residual for a Stokes flow problem in two-dimensions, with homogenous boundary conditions,

$$\mathcal{R}((\mathbf{u},p),(\mathbf{v},q)) = \int_\Omega \nabla\mathbf{u}\cdot\nabla\mathbf{v}\,dx - \int_\Omega \nabla\cdot\mathbf{u}\,q\,dx - \int_\Omega \nabla\cdot\mathbf{v}\,p\,dx \tag{4.15}$$

Given $Q(\mathbf{u})$, using Eq. (4.12b) we have the following representation for the error in $Q(\mathbf{u}_h)$,

$$\begin{aligned} Q(\mathbf{u}) - Q(\mathbf{u}_h) &= \int_\Omega \nabla(\mathbf{u}-\mathbf{u}_h)\cdot\nabla(\mathbf{u}^*-\mathbf{u}_h^*)\,dx \\ &\quad - \int_\Omega \nabla\cdot(\mathbf{u}-\mathbf{u}_h)\,(p^*-p_h^*)\,dx - \int_\Omega \nabla\cdot(\mathbf{u}^*-\mathbf{u}_h^*)\,(p-p_h)\,dx \\ &\le |e(u_1)|_{H^1(\Omega)}|e(u_1^*)|_{H^1(\Omega)} + |e(u_2)|_{H^1(\Omega)}|e(u_2^*)|_{H^1(\Omega)} \\ &\quad + \|e(u_{1,1})\|_{L^2(\Omega)}\|e(p^*)\|_{L^2(\Omega)} + \|e(u_{2,2})\|_{L^2(\Omega)}\|e(p^*)\|_{L^2(\Omega)} \\ &\quad + \|e(u_{1,1}^*)\|_{L^2(\Omega)}\|e(p)\|_{L^2(\Omega)} + \|e(u_{2,2}^*)\|_{L^2(\Omega)}\|e(p)\|_{L^2(\Omega)} \end{aligned} \tag{4.16}$$

where $e(u_1)$ denotes the error in the first component of $\mathbf{u}$, $e(u_{1,1})$ denotes the error in the first derivative of the first component of $\mathbf{u}$ and so on. To compute an estimate of this form while maintaining the physics-independent nature of `libMesh`, we express the estimate in Eq. (4.16) as a matrix-weighted inner-product,

$$\begin{aligned} Q(\mathbf{u}) - Q(\mathbf{u}_h) \le &\begin{bmatrix} e(u_1) \\ e(u_2) \end{bmatrix}^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} e(u_1^*) \\ e(u_2^*) \end{bmatrix} \\ &+ \begin{bmatrix} e(u_{1,1}) \\ e(u_{2,2}) \\ e(p) \end{bmatrix}^T \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} e(u_{1,1}^*) \\ e(u_{2,2}^*) \\ e(p^*) \end{bmatrix} \end{aligned}$$

$$= \begin{bmatrix} e(u_1) \\ e(u_2) \\ e(u_{1,1}) \\ e(u_{2,2}) \\ e(p) \end{bmatrix}^T \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} e(u_1^*) \\ e(u_2^*) \\ e(u_{1,1}^*) \\ e(u_{2,2}^*) \\ e(p^*) \end{bmatrix} \tag{4.17}$$

Thus, for the multivariable case, if the residual $\mathcal{R}(u, v)$ can be split into the contributions from each variable as $\sum_j \mathcal{R}^j(u, v)$, we can generalize Eq. (4.13) in the following manner,

$$\begin{aligned} |Q(u) - Q(u_h)| &\leq |\mathcal{R}_u(u_h, z - z_h)(u - u_h)| \\ &= |\sum_j \mathcal{R}_u^j(u_h, z - z_h)(u - u_h)| \\ &= |\sum_{n=1}^{N_{el}} \sum_j \mathcal{R}_u^j(u_h, z - z_h)(u - u_h)| \\ &\leq \sum_{n=1}^{N_{el}} \|z^j - z_h^j\| \; M_{ij} \; \|u^i - u_h^i\| \end{aligned} \tag{4.18}$$

where M_{ij} is a matrix of weights. In the `libMesh` implementation of the Adjoint Residual Error Indicator, the user supplies the weight matrices and an estimate of the form given by Eq. (4.18) is computed for every element by the library.

4.2.2 Nonlinear Problems

One can further consider the case of nonlinear problems, where the weight matrix described in the previous section will no longer contain constants but functions. As an example, consider the residual for a combined Stokes and

convection-diffusion problem,

$$\mathcal{R}((\mathbf{u},p,C),(\mathbf{v},q,H)) = \int_\Omega \nabla\mathbf{u}\cdot\nabla\mathbf{v}\;dx - \int_\Omega \nabla\cdot\mathbf{u}\;q\;dx - \int_\Omega \nabla\cdot\mathbf{v}\;p\;dx + \int_\Omega \mathbf{u}\cdot\nabla C\;H\;dx + \frac{1}{\mathrm{Pe}}\int_\Omega \nabla C\cdot\nabla H\;dx \tag{4.19}$$

where C is the species concentration and Pe is the Péclet number. Upon linearizing about an approximate solution $(\mathbf{u}_h, p_h, C_h)$, and computing the adjoint solution $(\mathbf{u}_h^*, p_h^*, C_h^*)$ one can obtain an expression for the QoI error along the lines of Eq. (4.12b). We focus our attention on the nonlinear term arising due to the product of the velocity and concentration gradient. The first part of the error contribution due to this term is given by,

$$\begin{aligned}
&\int_\Omega \mathbf{u}_h\cdot\nabla(C-C_h)\,(C^*-C_h^*)\,dx = \sum_{i=1}^{N_{el}}\int_{K_i}\mathbf{u}_h\cdot\nabla(C-C_h)\,(C^*-C_h^*)\,dx\\
&= \sum_{i=1}^{N_{el}}\int_{K_i}\left((u_1)_h\,(C_1-(C_1)_h)\,(C^*-C_h^*)\;+(u_2)_h\,(C_2-(C_2)_h)\,(C^*-C_h^*)\right)\,dx\\
&= \sum_{i=1}^{N_{el}}\int_{K_i}\left(e((u_1)_hC_1)\,e(C^*)+e((u_2)_hC_2)\,e(C^*)\right)\,dx\\
&\leq \sum_{i=1}^{N_{el}}\|e((u_1)_hC_1)\|_{L^2(K_i)}\|e(C^*)\|_{L^2(K_i)} + \|(u_2)_hC_2\|_{L^2(K_i)}\|e(C^*)\|_{L^2(K_i)}
\end{aligned} \tag{4.20}$$

where C_i is the derivative of the scalar variable C in the ith direction. One can think about the error term $\|e((u_1)_hC_1)\|_{L^2(K_i)}$ as the error in the variable C_1 in a weighted L^2 norm, with weight given by $(u_1)_h$. It is not difficult to incorporate this interpretation in the Patch-recovery Error Estimator. We simply scale the computed error variable by the appropriate weight at all the

relevant quadrature points. A new weighted patch recovery error estimator function was added to the library to allow the computation of such a weighted error estimate.

To implement this functionality in `libMesh` without changing the physics independent nature of the library, an abstract base class `FEMFunctionBase` was added to the library. This class has a single virtual function. The user simply needs to derive a new class from `FEMFunctionBase` and overload the virtual function with a local function. The local function has to return the appropriate functionals of the finite element solution. The user can then obtain pointers to objects that access these local functions and pass them to the new weighted patch recovery error estimator function. This function will then compute the weighted error indicators.

4.3 Adjoint-based Parameter Sensitivity Analysis

Adjoint-based methods can be used for parameter sensitivity analysis. In comparison to other techniques such as finite difference analysis or forward QoI sensitivity analysis, adjoint methods are especially efficient when the number of parameters exceeds the number of QoIs [50]. We will now derive an expression for the sensitivity of a QoI, $Q : U \to \mathbb{R}$ w.r.t. a parameter p, in terms of the adjoint problem corresponding to Q. The sensitivity to p at the point p_0 is given by,

$$Q' = \left.\frac{dQ}{dp}\right|_{p=p_0} \tag{4.21}$$

Taking the derivative of the residual $\mathcal{R}$ w.r.t. to p, we obtain,

$$\mathcal{R}(u(p), v; p) \equiv 0 \quad \forall\, v \in V \tag{4.22a}$$

$$\frac{d\mathcal{R}}{dp} = 0 \tag{4.22b}$$

$$\frac{\partial \mathcal{R}}{\partial p} + \frac{\partial \mathcal{R}}{\partial u}\frac{\partial u}{\partial p} = 0 \tag{4.22c}$$

The adjoint problem satisfies,

$$\frac{\partial \mathcal{R}}{\partial u}(u, z(u, p); p_0) = \frac{\partial Q}{\partial u}(u; p_0) \tag{4.23}$$

We can now derive an expression for Q' in terms of z,

$$Q' = \frac{\partial Q}{\partial p} + \frac{\partial Q}{\partial u}\frac{\partial u}{\partial p} \tag{4.24a}$$

$$= \frac{\partial Q}{\partial p} + \frac{\partial \mathcal{R}}{\partial u}\frac{\partial u}{\partial p} \tag{4.24b}$$

$$= \frac{\partial Q}{\partial p} - \frac{\partial \mathcal{R}}{\partial p}(u, z(u, p_0)) \tag{4.24c}$$

Thus only one adjoint problem has to be solved to get sensitivities of a particular QoI to many parameters. Since the adjoint problem is linear, the computational expense of this method mainly involves the cost of a linear solve (per QoI) and a matrix-vector inner product (per parameter). We will now prove a theorem that establishes sufficient conditions for the discrete approximation to the senstivity to converge to the true sensitivity.

Theorem 4.3.1. *Consider the variational problem,*

Given $p \in \mathbb{R}$, $\mathcal{R} : U \times V \times \mathbb{R} \to \mathbb{R}$, *find* $u \in U$ *s.t.* $\mathcal{R}(u(p), v; p) = 0 \quad \forall\, v \in V$

where U and V are Hilbert spaces. The adjoint solution z then satisfies,

$$\textit{Given } Q : U \to \mathbb{R}, \textit{ find } z \in V \textit{ s.t. } \frac{\partial \mathcal{R}}{\partial u}(u(p); v, z(p)) = \frac{\partial Q}{\partial u}(u(p); v) \quad \forall\, v \in V$$

Let Q' denote the sensitivity of Q to p. Further, let u_h and z_h denote the discrete approximations to u and z in the subspaces U_h and V_h, obtained by solving,

$$\mathcal{R}(u_h(p), v_h; p) = 0 \quad \forall\, v_h \in V_h \tag{4.25}$$

$$\frac{\partial \mathcal{R}}{\partial u}(u_h(p); v_h, z_h) = \frac{\partial Q}{\partial u}(u_h(p); v_h) \quad \forall\, v_h \in V_h \tag{4.26}$$

Denote by Q_h and $\mathcal{R}_h$ the evaluation of Q and $\mathcal{R}$ using u_h. Let Q'_h correspond to the approximation of Q' using the discrete solution u_h. Further, let U^B be a neighborhood of u s.t. both Q_h and $\mathcal{R}_h$ are bounded with bounded derivatives w.r.t. u. Then, if the discrete solution u_h and the adjoint z_h converge to the continuous solutions u and z, the discrete sensitivity Q'_h converges to Q'.

Proof. From Eq. (4.24), we have,

$$Q' = Q_p - \mathcal{R}_p(u, z(u, p)) \tag{4.27}$$

On subtracting the discrete sensitivity, we obtain,

$$\begin{aligned} Q' - Q'_h &= Q_p(u; p) - Q_p(u_h; p) + \mathcal{R}_p(u_h, z(u, p)) - \mathcal{R}_p(u, z(u, p)) \\ &\quad + \mathcal{R}_p(u_h, z_h(u_h, p)) - \mathcal{R}_p(u_h, z(u, p)) \\ &= Q_p(u; p) - Q_p(u_h; p) + \mathcal{R}_p(u_h, z(u, p)) \\ &\quad - \mathcal{R}_p(u, z(u, p)) - \mathcal{R}_p(u_h, z - z_h; p) \end{aligned} \tag{4.28}$$

Since $u_h \to u$, we have,

$$\exists\, \bar{h} > 0,\ \text{s.t.}\ \forall\, h < \bar{h}, u_h \in U^B \tag{4.29}$$

Now by the generalized mean-value theorem we have,

$$|Q_p(u;p) - Q_p(u_h;p)| \leq \sup_{u \in U^B} \|Q_{pu}(u;p)\| \|u - u_h\|_U \tag{4.30}$$

$$|\mathcal{R}_p(u_h, z(u,p)) - \mathcal{R}_p(u, z(u,p))| \leq \sup_{u \in U^B} \|\mathcal{R}_{pu}(u,z)\| \|u - u_h\|_U \tag{4.31}$$

Thus we have,

$$\begin{aligned} Q' - Q'_h \leq & \left(\sup_{u \in U^B} \|Q_{pu}(u;p)\| + \sup_{u \in U^B} \|\mathcal{R}_{pu}(u,v)\| \right) \|u - u_h\|_U \\ & + |\mathcal{R}_p(u_h, z - z_h; p)| \end{aligned} \tag{4.32}$$

We see that the discrete sensitivity approaches the true sensitivity as u_h and z_h approach u and z, respectively. □

4.4 Implementation of adjoint-based methods in libMesh

4.4.1 libMesh: a Parallel C++ Finite Element software library

The `libMesh` open-source software library has been developed to facilitate the parallel simulation of multiscale, multiphysics applications using adaptive mesh refinement and coarsening strategies [56]. An array of linear solvers is available through linear solver packages like `PETSc` [8] and Trilinos [48]. Various continuous and discontinuous Finite Element families can be used, such as Lagrange, Clough-Tocher [96], or Discontinuous Galerkin elements. The

library supports unstructured, structured, and hybrid grids in two or three dimensions. Adaptive mesh refinement and coarsening strategies can be implemented using a variety of error indicators. These include the uniform [31], flux-jump (or Kelly) [54], patch-recovery [109] and Laplacian jump [96] error indicators. Adaptive mesh redistribution techniques are also included [15, 41]. Adaptive time-stepping schemes can be utilized for time-dependent problems [74]. Support for visualization using `TecPlot`, `GMV`, and `Paraview` is available. Support for subdomain restricted variables has also been added, allowing the library to be used conveniently for multiphysics problems. In the course of the present work, adjoint methods for adaptive mesh refinement and sensitivity analysis methods [97] were added to `libMesh`.

4.4.2 Software Requirements and Design

Inexpensive, physics independent implementation of the adjoint techniques discussed in sections 4.2 and 4.3 requires that our implementation have the ability to,

1. Compute an approximation to the adjoint solution given the user-specified variational form for the primal problem, and the right-hand sides associated with the QoIs. Ideally, the user should not have to specify the weak form for the adjoint problem.

2. Use existing error indicators like flux-jump and patch-recovery to compute error bounds for both the primal and adjoint problems. See Eq. (4.18).

3. Flag and refine/coarsen elements.

4. Compute Finite Difference perturbations in the QoI $Q(u)$, and the residual $\mathcal{R}(u, z(u,p))$, by automatically varying the parameter values.

The new `AdjointResidualErrorEstimator` class and `adjoint_solve`, `adjoint_qoi_parameter_sensitivity` functions in `libMesh` accomplish all of the above goals, while maintaining the object-oriented, physics-independent philosophy of `libMesh` [97].

For adjoint-consistent formulations of the forward problem, the discrete adjoint can be computed simply by solving the transpose of the already assembled stiffness matrix. The user simply has to specify the right-hand side corresponding to the derivative of the QoI, in the `element_qoi_derivative` and `side_qoi_derivative` functions. Then on calling the function `adjoint_solve` library will then compute the corresponding adjoint solution. After this solution has been computed, adjoint-based error indicators can be computed by calling the `adjoint_residual_error_estimator` function. For sensitivity analysis, the user has to specify the QoI functionals in `element_qoi` and `side_qoi`. Once the adjoint has been computed, the sensitivities can then be computed by calling `adjoint_qoi_parameter_sensitivity`. Figure 4.1 shows a schematic of the implementation, with the new functions added to `libMesh` highlighted in bold. These functions are members of the library and independent of the user's own application code, thus enabling their easy use on a wide range of problems. The development of Application Programming Interfaces

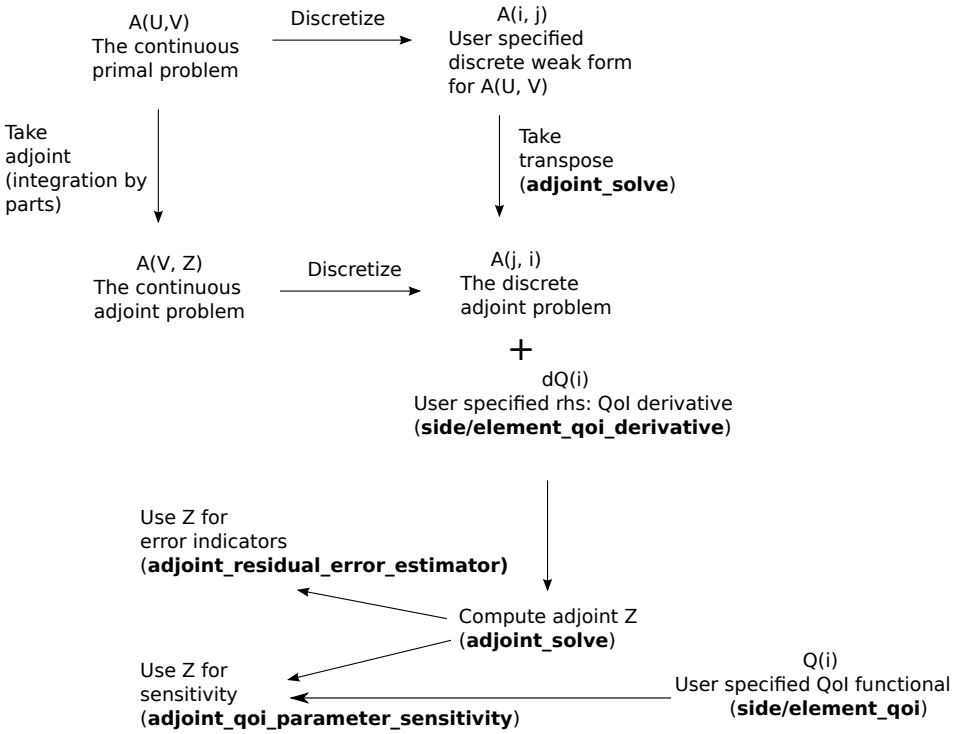


Figure 4.1: A schematic showing the use of adjoint-based methods in `libMesh`. Note that the user only needs to specify the discrete weak form for the primal, the right-hand side for the adjoint problem and the QoI functional, all other functionality is accessed within the library.

(APIs) for these new methods and their verification was carried out during the course of this research.

4.4.2.1 Preconditioner Reuse

Solving the discrete forward problem involves the solution of the linear system,

$$Ku = f$$

Due to the design of our software, for an adjoint-consistent formulation, the discrete adjoint problem involves solving,

$$K^T z = q$$

The forward problem is typically solved using a preconditioned Krylov subspace method. This creates an opportunity for more efficient solution of the discrete adjoint problem. One can reuse the preconditioner and the Krylov subspace used for solving the discrete forward problem. As an example, consider solving the forward problem using LU preconditioning, we then have

$$u = K^{-1} f = U^{-1} L^{-1} f$$

and

$$z = (K^T)^{-1} q = (U^T L^T)^{-1} q = (L^T)^{-1} (U^T)^{-1} q = (L^{-1})^T (U^{-1})^T q$$

Thus, once we solve the forward problem, the solution of the adjoint problem can be obtained with a matrix tranposition operation (which is free if the right data structures are used) and two matrix-vector multiplications. The adjoint problem can thus be solved at virtually no extra cost. In general, preconditioner reuse can lead to significant savings in computational costs. The reuse option has been provided in `libMesh` and users can avail of it by passing a preconditioner reuse flag in the `adjoint_solve` function.

4.4.2.2 Patch Reuse

Eq. (4.14) requires that we compute error estimates for the forward and adjoint solutions. In `libMesh`, the uniform, flux-jump or patch-recovery

error estimators can be used to compute these estimates. While the uniform error estimator is quite accurate, it is rather expensive. On the other hand, while the flux-jump error estimator is quite inexpensive, it is not very reliable, especially for systems in which the dominant mode of transport is not diffusion. In our experiments, the patch-recovery error estimator emerged as a relatively inexpensive and reliable error estimator.

The patch-recovery error estimator recovers a higher-order approximation of the numerical solution (or its derivatives) in a given element; this is done by using data from a patch of elements in the neighborhood of the element. An approximation of the error is then obtained by computing the difference between the original and recovered solutions (or derivatives). Details are given in a paper by Zienkiewicz and Zhu [109].

It was observed that in some cases, the construction of patches around elements was relatively expensive. For elliptic problems, the same patch can be used for multiple elements without affecting the accuracy of the estimator significantly [21]. Therefore, the library provides a patch reuse flag that can be set by the user as per his or her requirements. Reusing patches reduces costs but results in less accurate error estimates, while not reusing them increases costs but also provides better error estimates. The default option is for patches to be reused. This can lead to mesh refinement patterns that are not very regular in appearance. Despite this, the convergence performance of the error indicators with patch reuse enabled was indistinguishable with reuse disabled in all our test problems.

The next section will present verification studies for the new adjoint methods.

4.4.3 Verification of adjoint-based Error Indicators

The goal of code verification is to ensure that the computer implementation of an algorithm matches its true operations exactly [87, 66]. Of course, complete verification of complex algorithms such as the ones discussed in the earlier sections is extremely difficult. Therefore, we verified the methods and the software on representative test cases. Using such model cases, we were able to excercise most of the code added to the `libMesh` library and ensure that the software worked as expected for a range of inputs. Both QoI-based adaptive mesh refinement and adjoint-based parameter sensitivity software were verified. Several important technical issues were revealed in the numerical studies conducted during the course of this research. They were subsequently resolved through modifications to the code.

The model problem used for verification was a Poisson equation,

$$-\nabla \cdot (\alpha \nabla u) = f \ \text{ in } \ \Omega \tag{4.33a}$$

$$u = g \ \text{ on } \ \partial\Omega \tag{4.33b}$$

on the domain $\Omega = (0,1) \times (0,1)$, where we chose u to be the manufactured solution, $u(x, y; \alpha) = 4(1 - e^{-\alpha x_1} - (1 - e^{-\alpha})x_1)(x_2)(1 - x_2)$ and α to be 100. We calculated the corresponding f through differentiation. This manufactured solution exhibits multiscale spatial behavior due to a boundary layer near the

left vertical boundary, see Figure 4.2. It provided an exact analytic solution for comparison purposes and has been used extensively for verification purposes [75].

In the numerical simulations, the manufactured solution values on the boundary were specified as Dirichlet data for the associated computational problem. The boundary conditions evaluate to zero. Figure 4.2 shows a contour plot of the manufactured solution. Note the strong gradients in the boundary layer near the left boundary. The QoI chosen for the verification of the adjoint-based refinement methods was an integral of the solution over a subdomain,

$$Q(u(x_1, x_2; \alpha)) = \int_{0.5}^{0.75} \int_{0.5}^{0.75} u(x_1, x_2)\, dx_1\, dx_2 \tag{4.34}$$

The exact value of this QoI was 2.1484375×10^{-2}, and this was obtained using the symbolic toolkit in `MATLAB`.

To test the adjoint capability, the approximate problem was solved using the h-adaptive mesh refinement capability of `libMesh`, beginning from an initial coarse uniform mesh. First-order quad Lagrange elements were used for the simulations. Boundary conditions were set using the penalty method, and a value of 10^{-8} was given to the penalty parameter ϵ. The variational form used to solve Eq. (4.33) in `libMesh` was,

Given $\alpha \in \mathbb{R}$, find $u_\epsilon \in H^1(\Omega)$ such that

$$\int_\Omega \alpha \nabla u_\epsilon \cdot \nabla v \, dx + \frac{1}{\epsilon} \int_{\partial\Omega} u_\epsilon \, v \, ds = \int_{\partial\Omega} f \, v \, dx \quad \forall\, v \in H^1(\Omega)$$

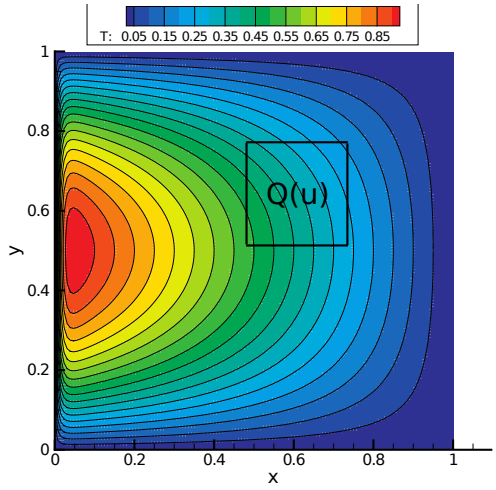


Figure 4.2: The manufactured solution $u(x, y; \alpha)$ of the model problem with $\alpha = 100$. Note the boundary layer on the left-hand side. The QoI region corresponding to Eq. (4.34) is also shown.

The problem was solved recursively with adaptive mesh refinement using algorithm 1. The solution process was terminated on either reaching a target number of elements in the FE grid or a maximum number of adaptive steps. These parameters were chosen to be 100,000 and 20 for the experiments done in this study.

Figure 4.3 shows an adaptively refined mesh, after 12 refinement steps. The mesh has been superimposed on a color plot of the numerically computed solution of the adjoint problem. This mesh was obtained by using the Adjoint Residual method, where the primal and dual estimates were obtained using the patch-recovery estimators. We see extensive mesh refinement in the boundary layer where most of the error in the primal solution originates. We also see refinement near the subdomain associated with the QoI. This is driven by the

Algorithm 1 Compute a Finite Element Solution to Eq. (4.33) using an adaptive meshing strategy targeted at the QoI given by Eq. (4.34). Stop on either reaching a mesh size limit $n_{elemtarget}$ or maximum number of steps (n_{steps_max}).

1: Start step counter n_{step}
2: Compute the Finite Element Solution (u_ϵ^h) to the problem using a uniform mesh (M_{start}) of resolution h_{start}
3: Compute an a-posteriori error indicator ($\widetilde{e_h}$) for the QoI based on the adjoint residual error indicator and flag elements to be refined
4: **if** $n_{elem} \geq n_{elemtarget}$ OR $n_{step} > n_{steps\_max}$ **then**
5: Go to step 11
6: **else**
7: Refine the top 10 % of the flagged elements to obtain an adaptive mesh $M_{adaptive}$
8: Increment n_{step} by 1
9: Repeat steps 2, 3 and 4 using the adapted mesh $M_{adaptive}$
10: **end if**
11: Output the results

error in the adjoint solution, which is the highest in the region near the QoI subdomain.

Log-log convergence plots for uniform and adaptive mesh refinement strategies are shown in Figure 4.4. The absolute error in the QoI is plotted against the number of degrees of freedom (dofs). Initially, the flux-jump error indicator performs comparably to the adjoint residual error indicator. This is because the dominant error in the computation is due to the sharp boundary layer. However, the flux-jump indicator is unable to detect the error contribution from the QoI location. Hence, once the error due to the QoI location becomes dominant, the flux-jump based refinement curve stagnates until the elements in the QoI location get refined (around step number 16). However,

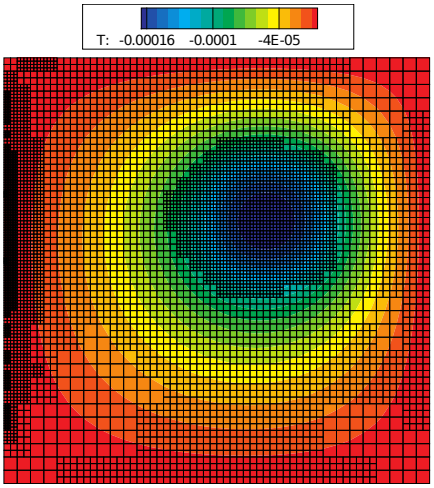


Figure 4.3: The adaptive mesh and numerically computed solution to the adjoint of the problem Eq. (4.33) with the right-hand side corresponding to the QoI given by Eq. (4.34).

as seen in Figure 4.3, the adjoint residual indicator identifies the error arising from the QoI location and refines in that region. Therefore, the adjoint-based adaptive refinement curve shows a consistent reduction in error till we hit the region where the error levels off. This leveling of the error occurs in the $\mathcal{O}\left(10^{-6}\right)$ region, which is to be expected since the penalty parameter used to set the boundary conditions was 10^{-8}. The error due to the penalty method is at least $\mathcal{O}\left(\epsilon|\alpha\int_{\partial\Omega}\partial_n u_\epsilon|\right)$, see Chapter 6 for more details on the penalty error.

Since first order elements have been used, the expected convergence rate for the QoI is 2 [36]. The rate obtained for the uniform refinement curve was ≈ 3. It is not clear why this higher rate was obtained. In general, one expects the convergence plot for an adaptive method to show very rapid decrease in the preasymptotic region and then become parallel to the plot for the uniform

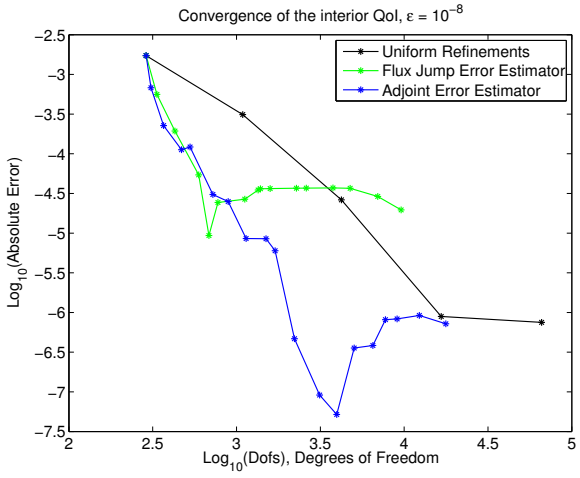


Figure 4.4: Convergence plot for the QoI given by Eq. (4.34), obtained by solving Eq. (4.35) using uniform refinements, flux-jump and adjoint-based adaptive refinement strategies. Stagnation in error reduction is seen for the flux-jump curve from the 8th to 12th steps. In contrast, consistent error reduction is achieved by the adjoint-based refinement strategy.

method in the asymptotic region. The adjoint residual based adaptive curve converges at a rate of nearly 4 initially; it then enters a region of slower convergence, before converging at a rate of ≈ 6 till it hits the leveling off region.

4.4.4 Verification of adjoint-based Sensitivity Analysis

Next, we verified the adjoint parameter sensitivity analysis software. The model problem was still given by Eq. (4.33). However, a different QoI was used,

$$Q(u) = -\int_{\Omega} \alpha \, \nabla w \cdot \nabla u \, dx \tag{4.35}$$

where the weight function $w(x_1, x_2)$ was given by,

$$w(x_1, x_2) = x_1 \times (1 - x_1) \times (1 - x_2) \tag{4.36}$$

This QoI was sensitive to the parameter α and excercised both components of the adjoint sensitivity analysis; the partial derivative of the QoI functional, and the partial derivative of the adjoint-weighted residual. The exact sensitivity of the QoI was obtained using the symbolic toolkit in Matlab. It was computed to be,

$$Q' = \left.\frac{dQ}{d\alpha}\right|_{\alpha=100} = -\frac{1}{3} \tag{4.37}$$

Numerical experiments were conducted to determine sensitivity of the QoI to the parameter α. A setup identical to the one used for the verification of the adjoint residual error indicator was used. However, second-order Lagrange elements were used instead of first-order elements.

Convergence plots of the absolute error in the sensitivity versus the number of dofs are shown in Figure 4.5. The results illustrate two points,

1. The ability of adjoint sensitivity analysis method in `libMesh` to accurately compute sensitivities.

2. The improved convergence of the parameter sensitivity due to the use of QoI targeted mesh refinement.

There was no extra cost associated with the computation of the adjoint for mesh refinement, since the adjoint was already computed for the sensitivity

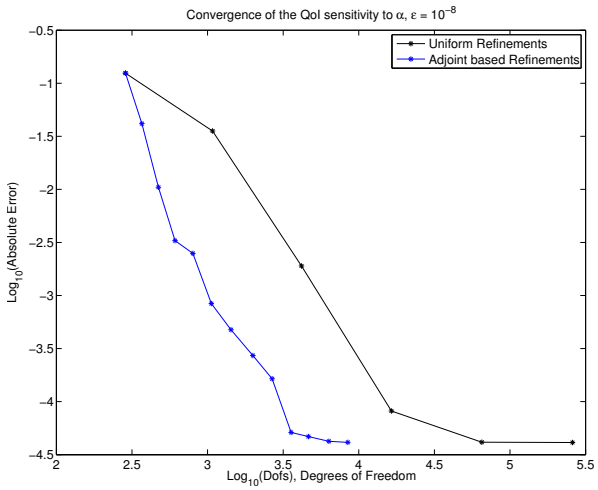


Figure 4.5: Convergence plot for the sensitivity of the QoI given by Eq. (4.35), to the parameter α. The weights in the Adjoint Residual Error Indicator are computed using the patch-recovery estimator.

analysis. We were easily able to combine adjoint-based sensitivity calculation with adjoint-based refinement for the corresponding QoI. This provides both the higher efficiency of the adjoint sensitivity method and the improved accuracy due to the adaptive mesh refinement.

We observe that the leveling of the error occurs at around $\mathcal{O}\left(10^{-4.5}\right)$, which we suspect is due to the use of the penalty method to set boundary conditions. This is still rather surprising, considering that the penalty parameter was 10^{-8} and shall be explored more in chapter 6.

4.5 Conclusions

Adjoint-based methods offer significant savings in computational cost for both estimating target QoIs and their parameter sensitivities. In this chapter, we presented the theory that underlies the use of adjoints for mesh refinement and sensitivity analysis. The adjoint methods were implemented and verified in the C++ Finite Element library `libMesh`, where they are now available for general use by the scientific computing community. In the next chapter, we shall apply these methods to the numerical analysis of microfluidic systems such as those discussed in chapter 3. Then, in chapter 7 we shall introduce a Monte Carlo method that uses the adjoint sensitivity derivatives to expedite UQ.

Chapter 5

Numerical Simulation of Electroosmotic Flow using libMesh

5.1 Introduction

We now consider the application of the new EOF formulation on specific microfluidic examples. First, we simulate a flow in a straight microchannel driven purely by electrosmosis. The objective here is to highlight the convergence and stability properties of the adjoint solution. We then showcase an adjoint-based adaptive strategy for mesh refinement on a T-shaped microchannel flow and adjoint-based parameter sensitivity analyses. We discuss the improvement of the convergence rates with respect to quantities of interest and their sensitivities when using adjoint-based techniques. Simulations are performed using the adjoint capabilities added to the `libMesh` Finite Element library [56]. For both applications, second-order Lagrange elements are employed for the potential and velocity approximations. Linear Lagrange elements are selected to approximate the pressure field in order to satisfy the inf-sup condition. Initial meshes in all the experiments dealing with the straight and T-channel domains consist of structured meshes of bi-quadratic quadrilateral elements. Numerical errors to generate the convergence plots are estimated in this work using so-called overkilled reference solutions of the two

problems. These are obtained on a uniform mesh of 428,676 degrees of freedom for the straight channel problem, and a combined adaptive-uniform mesh with 288,160 degrees of freedom for the T-channel problems. Numerical solutions are calculated using an ILU preconditioned GMRES iterative method for both problems. The linear algebra library PETSc [8] is accessed through `libMesh` to obtain these solutions.

5.2 Electroosmotic flow in a straight channel

Numerical experiments are performed here in the case of an electroosmotic flow in a straight channel. The channel has unit width and the length is five times the width. Since the objective of these simulations is to illustrate the numerical properties of the adjoint solution obtained by using the formulation given by Eq. (3.24), we set arbitrary values of the model parameters rather than choosing values representative of an actual flow. The fluid viscosity μ, electroosmotic slip parameter κ, and fluid density ρ are all taken to be unity. Constant potentials $\phi_i = 8$ and $\phi_o = 0$ are prescribed at the inlet Γ_{in} and outlet Γ_{out} boundaries, respectively. The electric conductivity of the fluid is chosen as $\sigma_c = 1 + x$ (note that this particular form of the conductivity is chosen for no other reason than better illustrate the properties of the computed adjoint).

The quantity of interest is defined here in terms of the bounded linear functional:

$$Q(U) = \int_{\Omega} \mathbf{u} \cdot \boldsymbol{\alpha} \, dx \tag{5.1}$$

where $\boldsymbol{\alpha} = (1, 1)$, $k(\mathbf{x}) = 1$, $\forall \, \mathbf{x} \in \Omega$. Such a bounded functional ensures that

any oscillations observed in the numerical results solely arise from the definition of the bilinear form in the adjoint problem. We consider the formulation of the adjoint problem as given in Eq. (3.29). After computing the forward solution using the numerical set-up as above, we obtain the adjoint potential ϕ_ϵ^* as seen in Figure 5.1. It was numerically verified that the adjoint potential and velocities were all in $H^1(\Omega)$. We also studied the convergence rates for the approximate primal and adjoint potentials and x-component of the velocity. Recall that the potential and velocity fields are both approximated using second-order Lagrange elements so that one would expect first-order convergence rates with respect to the number of degrees of freedom in the H^1-norm. However, one observes from Figure 5.2 that the primal velocity and the adjoint potential converge at a slower than optimal rate while the primal potential and adjoint velocity converge at the optimal rate. We speculate that this is due to the tangential 'slip' coupling given by Eq. (2.17) for the forward problem and the Neumann conditions given by Eq. (3.38a) for the adjoint problem. Essentially, we can say that the forward Stokes problem and the adjoint potential problem have non-smooth and non-accurate data, leading to higher errors in the computation of their solutions.

Remark 5.2.1. **Consequences of coupling both normal and tangential components:** Coupling both the normal and tangential components of the velocity to the potential may lead to an ill-posed adjoint problem. On directly enforcing the constraint given by Eq. (2.16c) on the wall boundary rather than splitting the two velocity components as in Eq. (2.17), one observes spurious

oscillations in the numerical adjoint potential field ϕ_ϵ^*, as shown in Figure 5.3. Note that these are visible only when adapted meshes are considered. However, this does not exclude the possibility that oscillations may appear on uniform grids in the case of other QoIs. One clearly observes in Figure 5.3(a) the presence of closed contour lines along the top and bottom wall boundaries. This result is confirmed in Figure 5.3(b), which shows the solution ϕ_ϵ^* along the top boundary. For further analysis and discussion see section 3.4.3 in chapter 3.

5.3 Electrosmotic flow in a T-channel

Crossing T- and H-channels are commonly utilized in microfluidics. Applications typically involve mixing of two chemical species [86], purification [93], or fluid identification [110]. However, numerical modeling of electrosmotic flows with slip boundary conditions in such geometrical configurations poses distinctive challenges due to the presence of corner singularities [28]. One immediate consequence is the observation of reduced convergence rates in the approximation of the global solution. A possible remedy is to use adaptive finite element methods to help restore the optimal convergence properties of such singular problems [31]. Likewise, adaptive methods can also improve the convergence behavior of the adjoint solution and potentially restore the optimal rates that one may expect when estimating linear QoIs.

We consider below a T-channel geometry. The two upper ends of the T-channel, $\Gamma_{i,l}$ and $\Gamma_{i,r}$, correspond to the left and right inlets, respectively, at

Table 5.1: Values of the input parameters for the T-channel flow.

Parameter	Symbol	Value
Conductivity	σ_c	1.0
Inlet potentials	ϕ_i	8.0
Outlet potential	ϕ_o	0.0
Fluid viscosity	μ	1.0
Slip parameter	λ	1.0

which a high potential ϕ_i is prescribed, while the bottom end of the channel Γ_o, the flow outlet, is set to the ground potential $\phi_o = 0$. The flow is assumed here to be purely electrically driven, in which case Dirichlet pressure boundary conditions $p = 0$ are considered at the inlet and outlet boundaries. The flow parameters used for the numerical experiments are provided in Table 5.1. In the numerical experiments below, we consider the following quantity of interest:

$$Q(U) = \int_{\Gamma_o} \mathbf{u} \cdot \mathbf{n} \, ds \tag{5.2}$$

i.e. $k_s(\mathbf{x}) = 1, \forall \mathbf{x} \in \Gamma_o$. We also estimate the sensitivity of the QoI with respect to the parameters ϕ_i, ϕ_o, and λ, evaluated in terms of the first derivatives $dQ/d\phi_i$, $dQ/d\phi_o$ and $dQ/d\lambda$. We used ten adaptive refinement steps followed by two uniform refinements (a total of 288,160 dofs) to calculate the reference values of these quantities. These values are reported in Table 5.2 and were used as exact values to compute numerical errors. The adaptive strategy for mesh refinement with respect to the QoI is described in algorithm 2, which has been implemented in `libMesh`. We show in Figure 5.4 the horizontal

Table 5.2: Estimated reference values of QoI and of its sensitivity to ϕ_i, ϕ_o, and λ.

$Q(U)$	$dQ/d\phi_i$	$dQ/d\phi_o$	$dQ/d\lambda$
1.0205649	0.1275705	-0.0637853	1.0203276

and vertical components of the primal velocity $\mathbf{u}$. We note that the vertical component of the velocity, shown in Figure 5.4(b), is close to zero near the inlets, but then undergoes a stiff acceleration around the corners. Likewise, we observe in Figure 5.4(a) the rapid deceleration of the horizontal velocity near the corners. This clearly induces a singular behavior of the solution at the two corners. Note also that the solution is symmetric about the centerline of the vertical channel, as expected, given that the inlet potentials at stations $\Gamma_{i,r}$ and $\Gamma_{i,l}$ are equal.

Next, we show the adjoint solutions computed using the adaptive procedure described in algorithm 2. The vertical velocity, displayed in Figure 5.5(a), exhibits a parabolic profile that reaches the maximum value along the centerline of the vertical channel and vanishes on its boundaries. Therefore, the presence of corners is solely responsible for the singular behavior in the velocity field. The adjoint potential solution is shown in Figure 5.5(b). The potential is of course singular at the corners due to the geometrical discontinuity and to the fact that the coupling boundary condition, although almost zero everywhere along the boundaries, becomes non-zero near the corners since $\nabla_{\Gamma_w} \cdot \big(\lambda \mathbf{t} \cdot (\boldsymbol{\sigma}_\epsilon^* \cdot \mathbf{n})\mathbf{t}\big)$ may not be zero there. This should imply extensive refinement near the corners, as confirmed by the adapted mesh shown in Figure 5.6.

Algorithm 2 Compute the finite element solution to Eq. (3.24) that either reaches a prescribed mesh size $h_{\min}$ or is obtained after a given number of adaptive steps $n_{\max}$ using an adaptive meshing strategy based on the dual approach with respect to the QoI Eq. (5.2).

1: Start step counter n_{step}
2: Compute the finite element solution u_h to the problem using a uniform mesh M_{start} of resolution $h_{\text{elem}} = h_{\text{start}}$
3: Compute an a posteriori error indicator $\tilde{e}_h$ for the QoI based on an adjoint residual based error indicator and flag elements to be refined
4: **if** $h_{\text{elem}} \leq h_{\min}$ OR $n_{\text{step}} > n_{\max}$ **then**
5: Go to step 11
6: **else**
7: Refine the top 30 percent of the flagged elements to obtain an adaptive mesh M_{adaptive}
8: Increment n_{step} by 1
9: Repeat steps 2, 3, and 4 using the adapted mesh M_{adaptive}
10: **end if**
11: Postprocess results.

We also used an adjoint method to compute parameter sensitivities for the given QoI to the parameters ϕ_i, ϕ_o and λ. The advantage of using an adjoint method for sensitivity analysis is that the sensitivity to all three parameters could be found with a single adjoint solve. This is considerably more efficient than using a finite difference or a forward sensitivity method. In addition, we can also combine the adjoint-based mesh refinement and sensitivity analysis for further improvements in the convergence of the sensitivities.

Convergence plots are shown in Figure 5.7. In particular, the relative error in the quantity of interest estimated using uniform refinement and adjoint-based adaptive refinement is shown in Figure 5.7(a) against the total number of degrees of freedom (dofs). Relative errors in the estimated sensi-

tivities of the QoI with respect to parameters are displayed in Figure 5.7(b). We note that the adaptive refinement strategy offers much improved error reduction than uniform refinement for both the estimation of the quantity of interest and its sensitivity derivatives.

In fact, on account of the geometric corner singularities present in the problem, we obtain an inferior convergence rate on using uniform refinement. However, with the adaptive method we obtain a rate of 1.5 (vs DoFs) for the QoI, which can be said to be semi-optimal. We had observed earlier that there is a loss of one convergence order for the forward velocity and adjoint potential for the straight channel problem where there are no corner singularities. We recall that with second-order Lagrange Finite Elements this would result in a convergence rate of 1.5 ($N^1 \times N^{\frac{1}{2}}$) for a linear QoI.

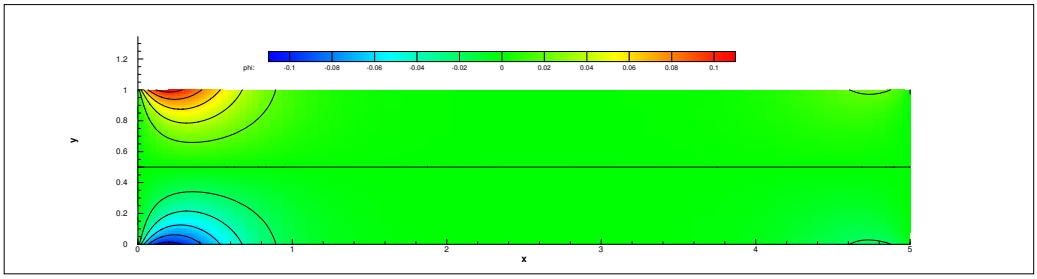
5.4 Conclusions

In chapter 3, we had presented an analysis of an electroosmotic flow model with slip boundary conditions and its adjoint. The slip boundary conditions require the evaluation of potential derivatives on the boundary, which increases the regularity requirements on the potential. We emphasize that a naive enforcement of the standard slip boundary condition leads to an ill-posed adjoint problem (see section 3.4.3 in chapter 3). A well-posed adjoint problem can be obtained by modifying the slip boundary condition ($\mathbf{u} + \lambda \nabla \phi = \mathbf{0}$), i.e. specifying the normal velocity at the wall independently of the potential ($\mathbf{u} \cdot \mathbf{n} = 0, \mathbf{u} \cdot \mathbf{t} + \lambda \nabla \phi \cdot \mathbf{t} = 0$). We further proposed a penalty formulation of

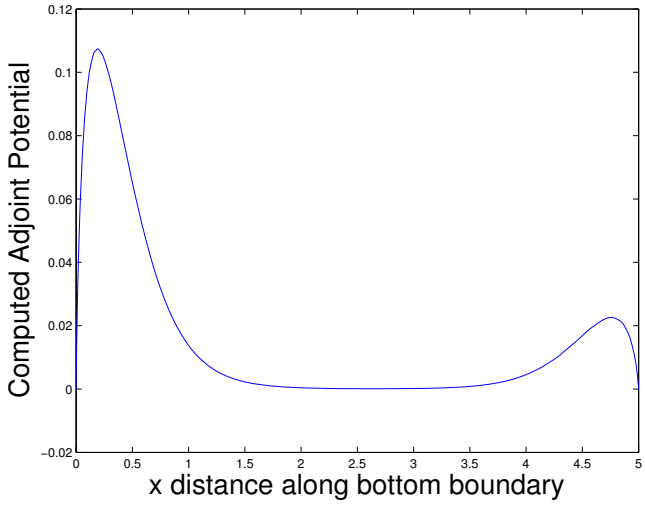
the forward problem that does not require any extra regularity for the potential, and leads to a well-posed, consistent adjoint formulation as well.

Then in chapter 4, we discussed the theory behind the use of adjoints in mesh refinement and sensitivity analysis and their implementation in the C++ Finite Element library `libMesh`. The penalty boundary conditions lead to a weak enforcement of the boundary coupling, allowing us to easily compute the adjoint problem using the adjoint capabilities of `libMesh`. The adjoint residual error estimator was derived, which was successfully used in this chapter for goal-oriented adaptive mesh refinement. The adjoint sensitivity method discussed in chapter 4 was also used to compute the sensitivities of the QoI.

We presented numerical experiments for a simple straight channel microflow and a more challenging T-channel flow. The convergence results for the straight channel problem indicate that the primal velocity and the adjoint potential converge at sub-optimal rates due to the nature of the coupling between the potential and the velocity. For the T-channel, we presented QoI computation and QoI adjoint sensitivity results for a practical engineering QoI. We observed a loss of convergence order due to the singularities in the T-channel geometry, and substantial improvements in the rate on using an adjoint-based adaptive method. However, the fully optimal convergence rate for the QoI could not be achieved, possibly due to the convergence properties of the adjoint potential.

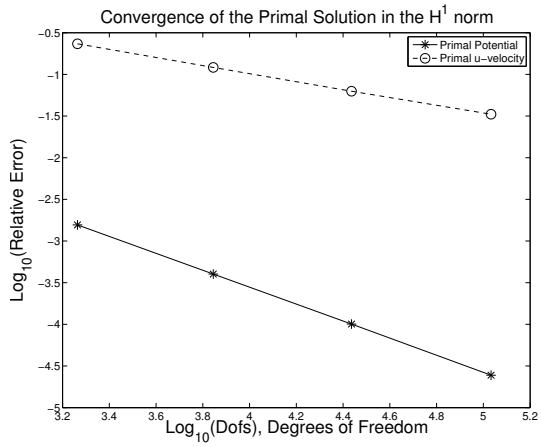


(a) Dual potential ϕ_ϵ^* computed with the penalty formulation.

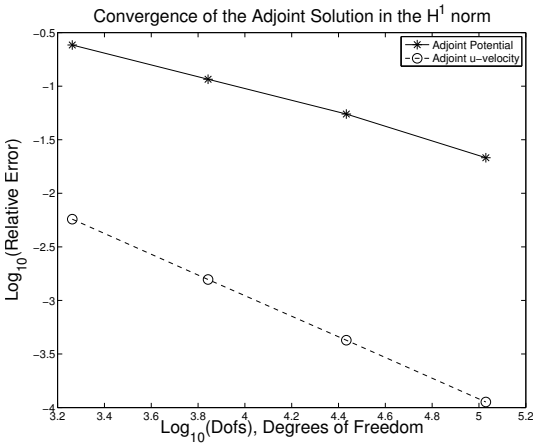


(b) Cutline of computed dual potential ϕ_ϵ^* along the bottom boundary.

Figure 5.1: Solutions to the adjoint problem obtained using the penalty formulation given by Eq. (3.29).

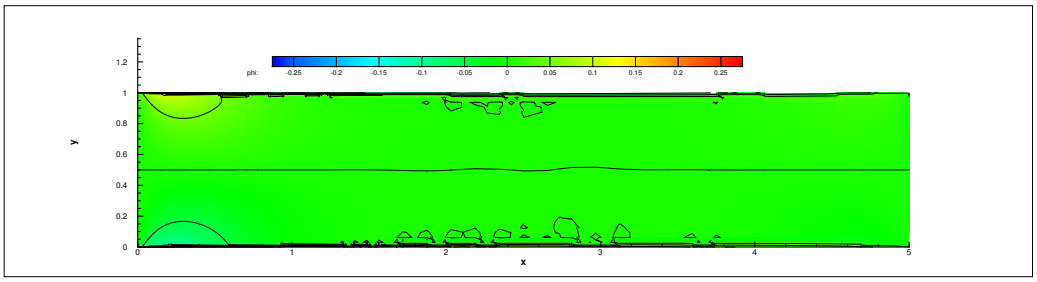


(a)

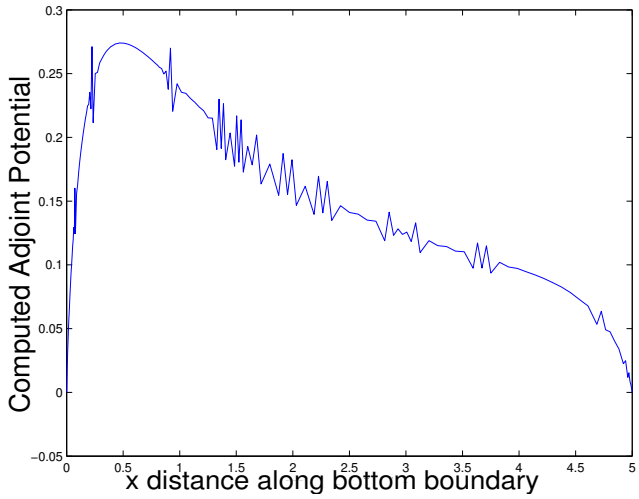


(b)

Figure 5.2: Convergence plot for the relative errors in the numerical primal and adjoint potentials and x-component of the primal and adjoint velocity with respect to the H^1-norm. Note the slower rate of convergence for the velocity in the forward problem and the potential in the dual problem.

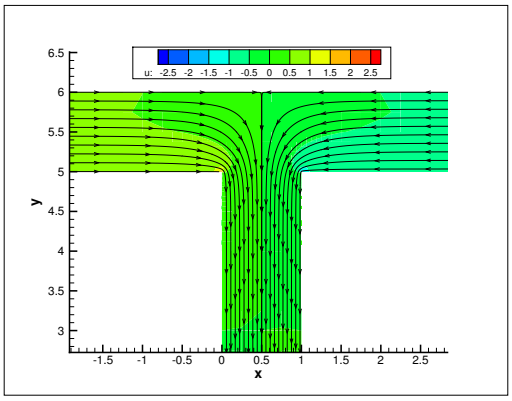


(a) Dual potential ϕ_ϵ^* computed with the ill-posed formulation.

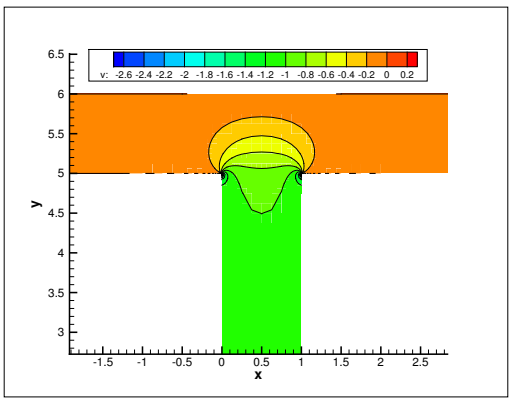


(b) Cutline of computed dual potential ϕ_ϵ^* along bottom boundary.

Figure 5.3: The solutions to the adjoint problems obtained using a naive penalty formulation.

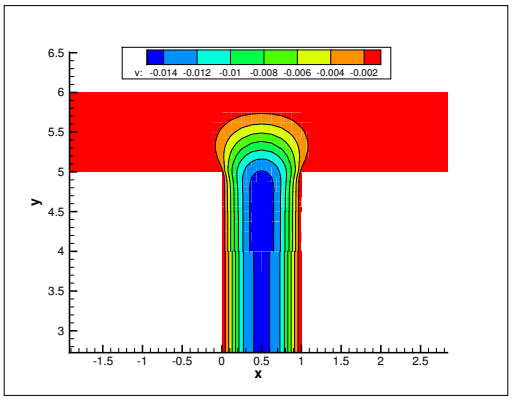


(a) x-component u_1 of velocity $\mathbf{u}$.

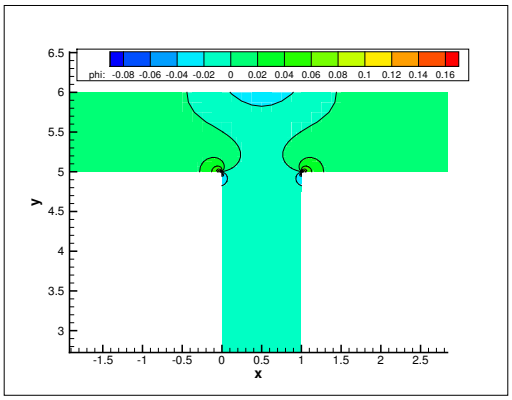


(b) y-component u_2 of velocity $\mathbf{u}$.

Figure 5.4: Contour plot of the primal solution obtained using the penalty formulation. The corner singularities are clearly visible due to the clustering of countour lines near them. The solution appears smooth away from the corners.



(a) y-component u_2^* of adjoint velocity $\mathbf{u}^*$. It is mainly different from zero inside the vertical channel indicating that the primal solution needs to be accurate in that region.



(b) Adjoint potential ϕ^*. Note that ϕ^* almost vanishes everywhere except at the corners and along the middle section of the top wall.

Figure 5.5: Contour plot of the y-component of the adjoint velocity $\mathbf{u}^*$ and of the adjoint potential ϕ^*.

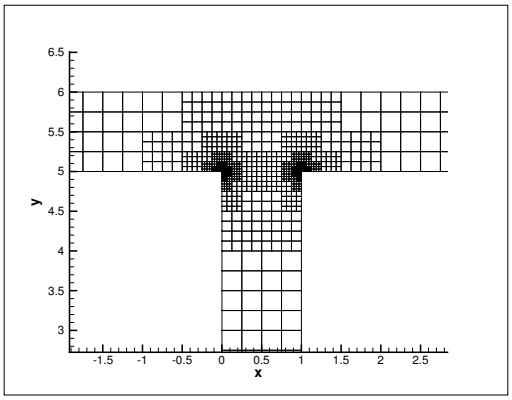
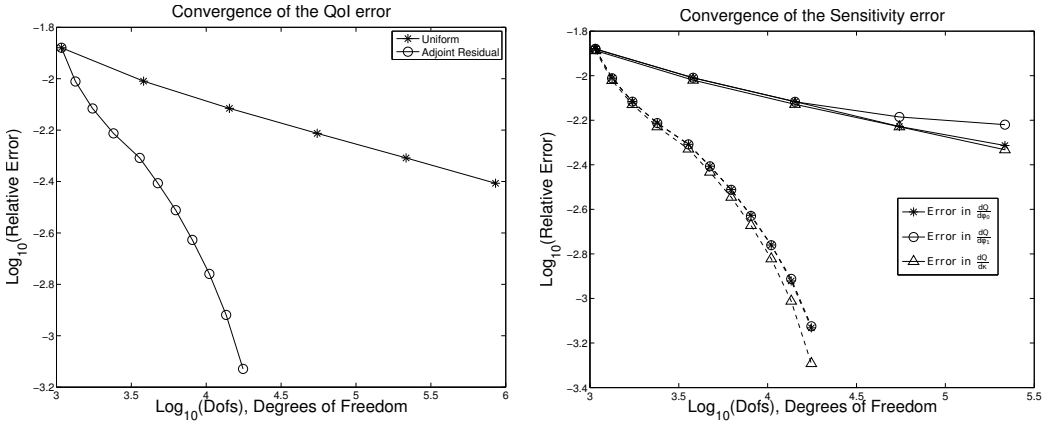


Figure 5.6: Adaptive mesh obtained using adjoint-based error estimates. Note that the elements get refined almost exclusively near the corners due to the singularities in the primal velocity and adjoint potential.



(a) Convergence plots for the relative error in QoI Eq. (5.2) using uniform and adjoint-based refinements.

(b) Convergence plots for the relative errors in the sensitivities of the QoI Eq. (5.2) using uniform and adjoint-based refinements.

Figure 5.7: Convergence plots for the approximation of the quantity of interest and its sensitivity to the parameters ϕ_i, ϕ_o, and λ.

Chapter 6

Penalty Recovery of the Normal Boundary Flux

6.1 Introduction

In chapter 3 we used the penalty method for applying a coupling boundary condition, and obtaining a well-posed adjoint problem. We saw that the penalty method implicitly defines a flux which is related to the actual boundary flux. In this chapter, we develop those ideas further and focus our attention on the implication of using a penalty method for enforcing Dirichlet boundary conditions while computing quantities of interest evaluated on the boundary. We are specifically interested in QoIs defined in terms of the normal flux. In this context, we first define an affine functional of the penalized solution called the 'penalty flux', and relate it to the actual flux for the non-penalized problem. We also analyze the effect of introducing the penalty by expressing the penalty flux in a power series about the actual flux. This series expansion leads naturally to an improved normal flux estimator, that takes into account the error introduced due to the penalty.

Although Lagrange multiplier techniques and Nitsche's method can offer better accuracy [6], the penalty method still finds widespread application

due to its simplicity and ease of implementation [67, 69, 5, 100, 61]. Some formulations of the Discontinuous Galerkin methods also employ penalty methods [4]. The penalty method has also been applied lately to solve elliptic problems in complicated domains [63]. There has also been recent work on asymptotic expansions for penalty methods used for PDE constrained optimization [14]. Babuška [5], Kikuchi and Oden [55, 70], Utku and Carey [100], have previously analyzed penalty methods. However, their work was mainly concerned with the behavior of the penalized solution in global norms such as the H^1- or L^2-norms, not in terms of specific QoIs such as the normal flux.

As mentioned earlier, QoI error estimation and sensitivity analysis naturally lend themselves to adjoint-based techniques [68, 13]. Since normal fluxes are often important QoIs from an engineering perspective, they have received particular interest in this context [105]. Adjoint problems associated with normal flux QoIs are therefore analyzed here. In particular, we emphasize that using a naive representation of the normal flux leads to an ill-posed adjoint problem. We also show that the penalty flux leads to an adjoint problem consistent with the modified adjoint problem derived by Giles et al. for weighted normal fluxes [43].

This chapter is organized as follows. In Section 6.2, we introduce our model problem and give some background on the penalty method. We also introduce the so-called 'penalty' flux and relate it to the true normal flux. In Section 6.3, we present the analysis for the error in the normal flux, focusing on the errors due to the use of the penalty method. We also discuss the

adjoint problem associated with the penalty flux, and use it to analyze the discretization error in the computation of the normal flux. In Section 6.4, we present numerical experiments that illustrate findings from our theoretical results. Finally, we give some concluding remarks from this work in Section 6.5.

6.2 The Penalty Method and the Normal Boundary Flux

6.2.1 Model Problem

Consider a model Poisson problem defined on an bounded, open subset Ω of $\mathbb{R}^d$, $d = 2, 3$, with a Lipschitz boundary $\partial\Omega$,

$$-\Delta u_0 = f \quad \text{in } \Omega \tag{6.1a}$$

$$u_0 = g \quad \text{on } \partial\Omega \tag{6.1b}$$

where the forcing function $f \in L^2(\Omega)$, and boundary conditions $g \in H^{\frac{1}{2}}(\partial\Omega)$. We look for a variational solution of Eq. (6.1) in the function space,

$$H^1_g(\Omega) = \{u \in H^1(\Omega),\ u = g \text{ on } \partial\Omega\} \tag{6.2}$$

By the principle of Dirichlet [27], the solution of the BVP given by Eq. (6.1) is the unique minimizer, $\min_{u \in H^1_g(\Omega)} J(u)$, of the convex functional,

$$J(u) = \frac{1}{2}\int_\Omega \nabla u \cdot \nabla u \, dx - \int_\Omega f u \, dx \tag{6.3}$$

The choice of function space means that the boundary condition $u = g$ is automatically defined, and the weak formulation of Eq. (6.1) then reads,

Given $f \in L^2(\Omega)$, find $u_0 \in H^1_g(\Omega)$ such that

$$\int_\Omega \nabla u_0 \cdot \nabla v \, dx = \int_\Omega f \, v \, dx \quad \forall v \in H_0^1(\Omega) \tag{6.4}$$

By the trace theorem we have that $u_0 \in H^{\frac{1}{2}}(\partial\Omega)$ and $\partial_n u_0 \in H^{-\frac{1}{2}}(\partial\Omega)$. We choose a weight function w such that $w \in H^{\frac{1}{2}}(\partial\Omega)$, and define the solution flux as the QoI. The weight function makes the QoI well defined, and restricts the evaluation of the flux to the relevant portion on the boundary $\partial\Omega$.

Definition 6.2.1. Let u_0 be the solution of the variational problem given by Eq. (6.4), and let $w \in H^1(\Omega)$. The solution flux $Q(u_0)$ is a linear functional $H^1(\Omega) \to \mathbb{R}$ given by,

$$Q(u_0) = \int_{\partial\Omega} \partial_n u_0 \, w \, ds \tag{6.5}$$

This QoI is important from an engineering perspective. For example, in heat conduction problems, the flux represents the heat transfer through a wall. In electrostatics, the normal flux is the electric flux across the wall. In flow simulations, the normal components of the lift and drag acting on an immersed body, are boundary momentum fluxes. Thus, solution accuracy near and on the boundary of the computational domain is particularly important.

6.2.2 The Penalty Method

Various strategies have been used to impose the Dirichlet boundary conditions $u_0 = g$ in Finite Element simulations. When these conditions are imposed using a penalty method, we add a penalty term to the convex functional given by Eq. (6.3) and seek its minimizer in $H^1(\Omega)$. The model problem

given by Eq. (6.1) is thus solved approximately by finding $\min_{u_\epsilon \in H^1(\Omega)} J_\epsilon(u_\epsilon)$, where

$$J_\epsilon(u_\epsilon) = \frac{1}{2}\int_\Omega \nabla u_\epsilon \cdot \nabla u_\epsilon \, dx - \int_\Omega f u_\epsilon \, dx + \frac{\epsilon^{-1}}{2}\int_{\partial\Omega} (u_\epsilon - g)^2 \, ds \tag{6.6}$$

with parameter $\epsilon \ll 1$. The corresponding weak formulation is given by,

$$\begin{gathered}\text{Given } f \in L^2(\Omega), \text{ find } u_\epsilon \in H^1(\Omega) \text{ such that} \\ \int_\Omega \nabla u_\epsilon \cdot \nabla v \, dx - \int_\Omega f\, v \, dx + \frac{1}{\epsilon}\int_{\partial\Omega} (u_\epsilon - g)\, v \, ds = 0 \quad \forall\, v \in H^1(\Omega)\end{gathered} \tag{6.7}$$

The penalty method replaces the true Dirichlet boundary condition with an approximate Robin boundary condition that is easier to enforce,

$$u_\epsilon + \epsilon(\partial_n u_\epsilon) = g \quad \text{on } \partial\Omega \tag{6.8}$$

Therefore the penalty method affects solution accuracy, especially on and near the boundary. It is critical that we understand the error introduced in the simulation due to the use of a penalty method, especially for QoIs like those given by Eq. (6.5), which are evaluated on the boundary. If the penalty method is used, a naive approximation to the solution flux defined by Eq. (6.5) is given by,

$$Q(u_\epsilon) = \int_{\partial\Omega} \partial_n u_\epsilon \, w \, ds \tag{6.9}$$

Using such a naive approximation can lead to inferior convergence behaviour in comparison with more sophisticated techniques of computing the flux [20]. Writing the flux in this form also leads to an ill-posed adjoint problem (see Remark 6.3.3).

Earlier analyses by Babuška [5], Utku and Carey [100] have analyzed the penalty method results in global norms such as the H^1- or the L^2-norms. They pay special attention to the relationship between the mesh size h and the penalty parameter ϵ. However, very often in practice, and especially in adaptive mesh simulations, the penalty parameters is set to be a small, fixed value. This results in an error of order $\mathcal{O}\left(\epsilon|\int_{\partial\Omega}\partial_n u\, ds|\right)$ in the $H^1(\Omega)$ norm due to the presence of the penalty term, i.e. if we were decreasing h to reduce discretization error, we would see the $H^1(\Omega)$ error plateau at this level, regardless of how small h is. As illustrated through both theory and numerical experiments in the following sections, the penalty error for the normal flux QoI can be much larger than $\mathcal{O}\left(\epsilon|\int_{\partial\Omega}\partial_n u\, ds|\right)$.

6.2.3 Equivalence of Penalty and Solution Flux

We now give the definition of the penalty flux and relate it to the solution flux. This relationship can be seen simply as a special case of a well known property of the penalty method [16, 63], however we give a short, complete proof here.

Definition 6.2.2. Let u_ϵ be obtained by finding the minimizer of penalized functional Eq. (6.6). Also, let $w \in H^1(\Omega)$. The penalty flux is defined as the affine functional $H^1(\Omega) \to \mathbb{R}$,

$$Q_\epsilon(u_\epsilon) = \int_{\partial\Omega} \frac{g - u_\epsilon}{\epsilon}\, w\, ds \tag{6.10}$$

Note that the penalty flux functional is linear if the boundary data g

are zero on the entire boundary. We now give the lemma that relates the solution flux and the penalty flux. We only consider a Poisson boundary value problem here. The extension to more general linear BVPs is intuitive.

Lemma 6.2.1. *Let u_0 be the solution to (6.4) and let u_ϵ be the solution to (6.7), with $\epsilon > 0$. Given $w \in H^1(\Omega)$, the following relationship holds:*

$$\lim_{\epsilon \to 0} Q_\epsilon(u_\epsilon) = Q(u_0)$$

Proof. We subtract the solution flux (6.5) from both sides of Eq. (6.7), and use the divergence theorem,

$$\begin{aligned}
Q_\epsilon(u_\epsilon) - Q(u_0) &= \frac{1}{\epsilon}\int_{\partial\Omega} (g - u_\epsilon)\, w\, ds - \int_{\partial\Omega} \partial_n u_0\, w\, ds \\
&= \int_\Omega f\, w\, dx - \int_\Omega \nabla u_\epsilon \cdot \nabla w\, dx - \int_{\partial\Omega} \partial_n u_0\, w\, ds \\
&= \int_\Omega f\, w\, dx - \int_\Omega \nabla u_\epsilon \cdot \nabla w\, dx - \int_\Omega f\, w\, dx + \int_\Omega \nabla u_0 \cdot \nabla w\, dx \\
&= \int_\Omega \nabla(u_0 - u_\epsilon) \cdot \nabla w\, dx
\end{aligned}$$

By the Cauchy-Schwarz inequality we then have,

$$|Q_\epsilon(u_\epsilon) - Q(u_0)| \leq |u_0 - u_\epsilon|_{H^1(\Omega)}\, |w|_{H^1(\Omega)} \tag{6.11}$$

Babuška [5] has shown that $\|u_0 - u_\epsilon\|_{H^1(\Omega)} \to 0$, as $\epsilon \to 0$. Hence proved. □

6.3 Error Analysis

We will now develop the error analysis for the approximation of the solution flux given by Eq. (6.5) by a Finite Element method that used the

penalty method to apply boundary conditions. This entails solving the discrete variational formulation of Eq. (6.7),

$$\text{Find } u_\epsilon^h \in U^h, \text{ s.t.}$$
$$\int_\Omega \nabla u_\epsilon^h \cdot \nabla v \, dx + \frac{1}{\epsilon}\int_{\partial\Omega} u_\epsilon^h \, v \, ds = \int_\Omega f \, v \, ds + \frac{1}{\epsilon}\int_{\partial\Omega} g \, v \, ds \quad \forall v \in U^h \tag{6.12}$$

where $U^h \subset H^1(\Omega)$ is the approximation space. There are various sources of error in such a computation, however, we shall focus on the discretization error and the penalty error. Either of these errors might be dominant, depending on the mesh size parameter h and the penalty parameter ϵ. We shall first focus solely on the error due to the penalty method. Our analysis will lead to an improved estimator for the solution flux QoI, which will reduce the penalty error by an order of magnitude. Then, in section 6.3.2 we will analyze the combined discretization and penalty error for this improved estimator.

6.3.1 Error Analysis for the Penalty Flux

Lemma 1 gives an asymptotic relationship between the penalty flux and the solution flux. We can gain further insight into the behavior of the penalty flux by expanding it in a series around the solution flux in terms of ϵ. Bonnans and Silva have considered asymptotic expansions for penalty methods used in PDE constrained optimization [14], however they did not consider expansions for QoIs error control or reduction. The following theorem provides the necessary result for the normal flux QoI and will allow us to reduce the penalty error in the calculation of the QoI.

Theorem 6.3.1. *Let Ω, u_0, g, u_ϵ, and ϵ be as defined in Lemma 6.2.1. Further assume that for any $w \in H^1(\Omega)$, the functional $\int_{\partial\Omega} u_\epsilon \, w \, ds$ is thrice differentiable with respect to ϵ in the open interval $(0, \epsilon)$ and continuous on the closed interval $[0, \epsilon]$. Then we have,*

$$Q_\epsilon(u_\epsilon) - Q(u_0) = \left(\int_\Omega \nabla \left(\left. \frac{du_\epsilon}{d\epsilon} \right|_{\epsilon=0} \right) \cdot \nabla w \right) \epsilon + \mathcal{O}\left(\epsilon^2\right) \tag{6.13}$$

Proof. We first derive a series expansion for the functional $G(u_\epsilon) = \int_{\partial\Omega} (u_\epsilon - g) \, w \, ds$ about $\epsilon = 0$;

$$G(u_\epsilon) = G(u_\epsilon)|_{\epsilon=0} + \left. \frac{dG}{d\epsilon} \right|_{\epsilon=0} \epsilon + \frac{1}{2} \left. \frac{d^2G}{d\epsilon^2} \right|_{\epsilon=0} \epsilon^2 + \mathcal{O}\left(\epsilon^3\right) \tag{6.14}$$

We will now evaluate $G(u_\epsilon)|_{\epsilon=0}$, $\left. \frac{dG}{d\epsilon} \right|_{\epsilon=0}$ and $\left. \frac{d^2G}{d\epsilon^2} \right|_{\epsilon=0}$. By the trace theorem,

$$\int_{\partial\Omega} (u_\epsilon - g) \, w \, ds \leq \|u_\epsilon - g\|_{L^2(\partial\Omega)} \|w\|_{L^2(\partial\Omega)} \leq C \|u_\epsilon - u_0\|_{H^1(\Omega)} \|w\|_{H^1(\Omega)} \tag{6.15}$$

where C is some constant that depends on the domain Ω. It follows that $G(u_\epsilon)|_{\epsilon=0} = 0$. By definition of $G(u_\epsilon)$, we have

$$\left. \frac{dG}{d\epsilon} \right|_{\epsilon=0} = \int_{\partial\Omega} \left. \frac{du_\epsilon}{d\epsilon} \right|_{\epsilon=0} w \, ds$$

Now, recall from (6.7), taking $v = w$, that

$$\int_\Omega \nabla u_\epsilon \cdot \nabla w \, dx + \frac{1}{\epsilon} \int_{\partial\Omega} u_\epsilon \, w \, ds = \int_\Omega f \, w \, ds + \frac{1}{\epsilon} \int_{\partial\Omega} g \, w \, ds$$

Differentiating this expression w.r.t. ϵ we obtain,

$$\int_\Omega \nabla \left(\frac{du_\epsilon}{d\epsilon} \right) \cdot \nabla w \, dx + \frac{1}{\epsilon} \int_{\partial\Omega} \frac{du_\epsilon}{d\epsilon} \, w \, ds - \frac{1}{\epsilon^2} \int_{\partial\Omega} u_\epsilon \, w \, ds = -\frac{1}{\epsilon^2} \int_{\partial\Omega} g \, w \, ds \tag{6.16}$$

Rearranging and multiplying throughout by ϵ,

$$\epsilon \int_\Omega \nabla \left(\frac{du_\epsilon}{d\epsilon} \right) \cdot \nabla w \, dx = \int_{\partial\Omega} \frac{u_\epsilon - g}{\epsilon} \, w \, ds - \int_{\partial\Omega} \frac{du_\epsilon}{d\epsilon} \, w \, ds \tag{6.17}$$

Now taking the limit as $\epsilon \to 0$ and using Lemma 6.2.1, we have,

$$\left. \frac{dG(u_\epsilon)}{d\epsilon} \right|_{\epsilon=0} = \lim_{\epsilon \to 0} \int_{\partial\Omega} \frac{u_\epsilon - g}{\epsilon} \, w \, ds = - \int_{\partial\Omega} \partial_n u_0 \, w \, ds$$

We now find the second derivative of $B(u_\epsilon)$,

$$\left. \frac{d^2 G(u_\epsilon)}{d\epsilon^2} \right|_{\epsilon=0} = \int_{\partial\Omega} \left. \frac{d^2 u_\epsilon}{d\epsilon^2} \right|_{\epsilon=0} w \, ds$$

Differentiating Eq. (6.17) and then using it in the resulting expression, we get,

$$\begin{aligned}
&\int_\Omega \nabla \left(\frac{du_\epsilon}{d\epsilon} \right) \cdot \nabla w \, dx + \epsilon \int_\Omega \nabla \left(\frac{d^2 u_\epsilon}{d\epsilon^2} \right) \cdot \nabla w \, dx \\
&= -\frac{1}{\epsilon^2} \int_{\partial\Omega} (u_\epsilon - g) \, w \, ds + \frac{1}{\epsilon} \int_{\partial\Omega} \frac{du_\epsilon}{d\epsilon} \, w \, ds - \int_{\partial\Omega} \frac{d^2 u_\epsilon}{d\epsilon^2} \, w \, ds \\
&= \frac{1}{\epsilon} \left[\int_{\partial\Omega} \frac{du_\epsilon}{d\epsilon} \, w \, ds - \int_{\partial\Omega} \frac{u_\epsilon - g}{\epsilon} \, w \, ds \right] - \int_{\partial\Omega} \frac{d^2 u_\epsilon}{d\epsilon^2} \, w \, ds \\
&= - \int_\Omega \nabla \left(\frac{du_\epsilon}{d\epsilon} \right) \cdot \nabla w \, dx - \int_{\partial\Omega} \frac{d^2 u_\epsilon}{d\epsilon^2} \, w \, ds
\end{aligned}$$

Thus we have,

$$\int_{\partial\Omega} \frac{d^2 u_\epsilon}{d\epsilon^2} \, w \, ds = -2 \int_\Omega \nabla \left(\frac{du_\epsilon}{d\epsilon} \right) \cdot \nabla w \, dx - \epsilon \int_\Omega \nabla \left(\frac{d^2 u_\epsilon}{d\epsilon^2} \right) \cdot \nabla w \, dx$$

At $\epsilon = 0$,

$$\left. \frac{d^2 G(u_\epsilon)}{d\epsilon^2} \right|_{\epsilon=0} = -2 \int_\Omega \nabla \left(\left. \frac{du_\epsilon}{d\epsilon} \right|_{\epsilon=0} \right) \cdot \nabla w \, dx$$

Since $\frac{d^3 G(u_\epsilon)}{d\epsilon^3} = \int_{\partial\Omega} \frac{d^3 u_\epsilon}{d\epsilon^3} w\, ds$ is finite due to our hypotheses, we can apply Taylor's theorem and obtain a series expansion for $G(u_\epsilon)$,

$$G(u_\epsilon) = \frac{1}{1!}\left(-\int_{\partial\Omega} \partial_n u_0\, w\, ds\right)\epsilon - \frac{2}{2!}\left(\int_\Omega \nabla\left(\frac{du_\epsilon}{d\epsilon}\bigg|_{\epsilon=0}\right)\cdot\nabla w\right)\epsilon^2 + \frac{1}{3!}\left(\int_{\partial\Omega} \frac{d^3 u_\epsilon}{d\epsilon^3}\bigg|_{\epsilon=\epsilon_L} w\, ds\right)\epsilon^3$$

Finally, on dividing through by ϵ, we get a series for the penalty flux,

$$-\int_{\partial\Omega} \frac{u_\epsilon - g}{\epsilon}\, w\, ds = \int_{\partial\Omega} \partial_n u_0\, w\, ds + \left(\int_\Omega \nabla\left(\frac{du_\epsilon}{d\epsilon}\bigg|_{\epsilon=0}\right)\cdot\nabla w\, dx\right)\epsilon + \mathcal{O}\left(\epsilon^2\right)$$

Hence proved. □

Definition 6.3.1. Let the hypotheses of Theorem 6.3.1 hold. We define the penalty sensitivity improved flux estimator as,

$$\widehat{Q}_\epsilon(u_\epsilon) = \int_\Omega f\, w\ - \nabla w\cdot\nabla u_\epsilon\, dx - \left(\int_\Omega \nabla\left(\frac{du_\epsilon}{d\epsilon}\bigg|_{\epsilon=0}\right)\cdot\nabla w\, dx\right)\epsilon \qquad (6.19)$$

Note that we have used the penalized weak form (6.7) to replace the boundary integral in the result of Theorem 6.3.1 with an interior integral. This improved estimator has $\mathcal{O}\left(\epsilon^2\right)$ error due to the penalty, in comparison to the usual estimator which will have $\mathcal{O}\left(\epsilon\right)$ error. Note that if the derivative $\left(\int_\Omega \nabla\left(\frac{du_\epsilon}{d\epsilon}\big|_{\epsilon=0}\right)\cdot\nabla w\, dx\right)$ is large, then $\mathcal{O}\left(\epsilon\right)$ can be significant, even for small values of ϵ. Thus the improvement offered by the more accurate $\mathcal{O}\left(\epsilon^2\right)$ estimator can be substantial. We will illustrate these points via our numerical experiments in Section 6.4.

Remark 6.3.1. We have assumed a substantial degree of regularity for the penalty approximation with respect to the penalty parameter in the hypothesis

for Theorem 6.3.1. We believe that this Theorem can be shown to hold with weaker hypotheses.

Remark 6.3.2. We would like to mention here that Dwight [34] has presented an analysis that uses senstivity derivatives with respect to artifical dissipation parameters. He used these derivatives to derive error indicators for the error introduced by some stabilization schemes. We believe that similar expansion based error analysis for the stabilized methods can lead to estimators that remove some of the error introduced by the stabilization.

6.3.1.1 Illustrative one-dimensional example

We can use a simple 1-D problem to illustrate the fact that the magnitude of the first order term in Eq. (6.19) can be large relative to the usual error encountered due to the penalty method. Consider a 1-D Poisson problem on the domain $\Gamma = (0, 1)$ given by,

$$-u'' = f \quad \text{on } \Gamma \tag{6.20a}$$

$$u(0) = 0 \tag{6.20b}$$

$$u(1) = 1 \tag{6.20c}$$

where $f \in L^2(\Gamma)$. Let $F(x)$ be the anti-derivative of $f(x)$,

$$F(x) = \int -f(x)\, dx$$

and let $G(x)$ be the anti-derivative of $F(x)$,

$$G(x) = \int \left(\int -f(x)\, dx \right) dx$$

The exact solution to Eq. (6.20) is given by,

$$u(x) = G(x) + Ax + B \tag{6.21}$$

where A and B are constants. Let our QoI be the solution flux at $x = 1$,

$$Q(u) = u'(1) \tag{6.22}$$

If we enforce the boundary conditions using a penalty method, we obtain the following weak formulation,

Given $f \in L^2(\Gamma)$, find $u_\epsilon \in H^1(\Omega)$ such that

$$\int_0^1 u'_\epsilon \, v' \, dx + \frac{1}{\epsilon} u_\epsilon(0) \, v(0) + \frac{1}{\epsilon} u_\epsilon(1) \, v(1) = \frac{1}{\epsilon} v(1) \quad \forall \, v \in H^1(\Omega) \tag{6.23}$$

The penalty method replaces the Dirichlet boundary conditions in Eq. (6.20) with Robin boundary conditions. Thus the penalized version of the 1-D system reads as,

$$-u''_\epsilon = f \tag{6.24a}$$

$$u_\epsilon(0) + \epsilon u'_\epsilon(0) = 0 \tag{6.24b}$$

$$u_\epsilon(1) + \epsilon u'_\epsilon(1) = 1 \tag{6.24c}$$

The exact solution to this penalized problem is given by,

$$u_\epsilon(x) = G(x) + A_\epsilon x + B_\epsilon \tag{6.25}$$

The constants A, B, A_ϵ and B_ϵ can be related as follows,

$$A_\epsilon = A + \epsilon \left(F(0) - F(1) \right) \tag{6.26a}$$

$$B_\epsilon = B - \epsilon\left(A + F(0)\right) + \epsilon^2\left(F(1) - F(0)\right) \tag{6.26b}$$

Since we have exact solutions for both the true and penalized problems, we can easily obtain the penalty error $Q_\epsilon(u_\epsilon) - Q(u)$,

$$\begin{aligned} Q_\epsilon(u_\epsilon) - Q(u) &= -\frac{u_\epsilon(1) - 1}{\epsilon} - u'(1) \\ &= \frac{1 - B_\epsilon - A_\epsilon - F(1)}{\epsilon} - u'(1) \\ &= \frac{1 - B_\epsilon - A_\epsilon - F(1)}{\epsilon} - \left(A + F(1)\right) \\ &= A + F(1) - \epsilon\left(F(1) - F(0)\right) - \left(A + F(1)\right) \\ &= \epsilon\left(F(0) - F(1)\right) \end{aligned} \tag{6.27}$$

We can better understand the penalty error if we choose a specific f and evaluate $F(0)$ and $F(1)$. We chose f as,

$$f(x) = \beta^2 \frac{e^{\beta(1-x)}}{1 - e^\beta}$$

This function has a very large value near the boundary $x = 0$, but then quickly decays away from this boundary. On evaluating $F(0)$ and $F(1)$ for this choice of f, we obtain,

$$\epsilon\left(F(0) - F(1)\right) = \epsilon\beta$$

In contrast, the usual error due to the penalty method is given by,

$$\epsilon u'(1) = \epsilon(A + F(1)) = \epsilon A - \epsilon\frac{\beta}{1 - e^\beta}$$

For this choice of f, the constant A evaluates to zero. Therefore, for large β, $\epsilon u'(1)$ approaches zero, however the penalty error is $\epsilon\beta$ due to the higher

order terms. Thus we see that the penalty error can be much larger than just $\epsilon u'(1)$. It is also seen that the source of the first order penalty error arises from the behaviour of the forcing function in the interior of the domain. If the forcing function has a sharp layer even away from the boundary of interest, the penalty error can be very large. We will further illustrate this point using numerical experiments for a 2-D Poisson in Section 6.4.

6.3.2 Total Error and the Adjoint Problem for Boundary Flux QoIs

In the previous section, we analyzed the error introduced in the computation of the normal flux due to the use of the penalty method. In actual simulations, we also have other sources of error, such as discretization errors. Adjoint-based methods and adaptive mesh refinement are established techniques of estimating such errors and reducing them. However, challenges remain in the application of adjoint methods to certain classes of problems. It has been shown that boundary coupling necessitates careful analysis of the adjoint problem [101, 40]. It is also well known that if the QoI is defined on the boundary, and especially if it is the flux of a solution variable on the boundary, there are special considerations involved in formulating the adjoint problem [43, 105]. Giles et al. [43] derived and analyzed the adjoint problem for the model Poisson system given by Eq. (6.1). They derived the following adjoint problem for the flux QoI corresponding to an interior representation

of Eq. (6.5),

$$-\Delta z = 0 \quad \text{in } \Omega \tag{6.28a}$$

$$z = w \quad \text{on } \partial\Omega \tag{6.28b}$$

where z denotes the adjoint solution. Computing this adjoint allows the calculation of accurate error estimates for the flux. In our analysis of the penalty flux in the previous section, we saw that it represented a convergent approximation to the true flux. Therefore, we propose another adjoint problem for the flux which uses the penalty flux (6.10) to specify the right hand side,

$$\begin{aligned} &\text{Find } z_\epsilon \in H^1(\Omega), \text{ s.t.} \\ &\int_\Omega \nabla v \cdot \nabla z_\epsilon \, dx + \frac{1}{\epsilon}\int_{\partial\Omega} v \, z_\epsilon \, ds = \frac{1}{\epsilon}\int_{\partial\Omega} v \, w \, ds \quad \forall v \in H^1(\Omega) \end{aligned} \tag{6.29}$$

The strong form corresponding to this variational formulation is,

$$-\Delta z_\epsilon = 0 \quad \text{in } \Omega \tag{6.30a}$$

$$z_\epsilon + \epsilon(\partial_n z_\epsilon) = w \quad \text{on } \partial\Omega \tag{6.30b}$$

We see that in the limit $\epsilon \to 0$ we obtain an adjoint problem consistent with Eq. (6.28a). We will now use the adjoint problem given by Eq. (6.29) to analyze the discretization error of the improved estimator given by Eq. (6.19), in combination the result of Theorem 6.3.1 to analyze the penalty error.

Remark 6.3.3. If we simply consider the adjoint variational formulation with the naive boundary flux given by Eq. (6.9) forming the right hand side, we obtain,

$$\text{Find } z_\epsilon \in H^1(\Omega), \text{ s.t.}$$

$$\int_\Omega \nabla v \cdot \nabla z_\epsilon \, dx + \frac{1}{\epsilon}\int_{\partial\Omega} v \, z_\epsilon \, ds = \int_{\partial\Omega} \partial_n v \, w(x_1, x_2) \, ds \quad \forall v \in H^1(\Omega) \tag{6.31}$$

The variational problem given by Eq. (6.31) is ill-posed, because the test function $v \notin H^1_\Delta(\Omega)$, where,

$$H^1_\Delta(\Omega) = \{v \in H^1(\Omega) : \Delta v \in L^2(\Omega)\} \tag{6.32}$$

making the right-hand side unbounded [91]. See also Remark 3.4.1.

Theorem 6.3.2. *Let the hypotheses of Theorem 6.3.1 hold, and let u^h_ϵ be the solution of Eq. (6.12). We then have the following error estimate,*

$$\begin{aligned} &\widehat{Q}_\epsilon(u^h_\epsilon) - Q(u_0) \\ &= -\mathcal{R}_\epsilon(u^h_\epsilon; z_\epsilon) + \left(\int_\Omega \nabla \left(\frac{de^h_\epsilon}{d\epsilon}\bigg|_{\epsilon=0} \right) \cdot \nabla w \, dx \right) \epsilon + \mathcal{O}\left(\epsilon^2\right) \end{aligned} \tag{6.33}$$

where,

$$\begin{aligned} \mathcal{R}_\epsilon(u^h_\epsilon; z_\epsilon) &= \int_\Omega f \, z_\epsilon \, dx + \frac{1}{\epsilon}\int_{\partial\Omega} g \, z_\epsilon \, ds - \int_\Omega \nabla z_\epsilon \cdot \nabla u^h_\epsilon \, dx - \frac{1}{\epsilon}\int_{\partial\Omega} u^h_\epsilon \, z_\epsilon \, ds \\ e^h_\epsilon &= u_\epsilon - u^h_\epsilon \end{aligned}$$

Proof. By definition,

$$\begin{aligned} &\widehat{Q}_\epsilon(u^h_\epsilon) - Q(u_0) \\ &= \int_\Omega f \, w \; - \nabla w \cdot \nabla u^h_\epsilon \, dx - \left(\int_\Omega \nabla \left(\frac{du^h_\epsilon}{d\epsilon}\bigg|_{\epsilon=0} \right) \cdot \nabla w \, dx \right) \epsilon - \int_{\partial\Omega} \partial_n u_0 \, w \, ds \end{aligned}$$

Adding and subtracting $\int_\Omega \nabla w \cdot \nabla u_\epsilon \, dx$ and $\left(\int_\Omega \nabla \left(\frac{du_\epsilon}{d\epsilon}\Big|_{\epsilon=0} \right) \cdot \nabla w \, dx \right) \epsilon$, we get,

$$\widehat{Q}_\epsilon(u^h_\epsilon) - Q(u_0)$$

$$
\begin{aligned}
&= \int_\Omega f\, w \; - \nabla w \cdot \nabla (u_\epsilon^h - u_\epsilon)\, dx - \int_\Omega \nabla w \cdot \nabla u_\epsilon \, dx \\
&- \left(\int_\Omega \nabla \left(\frac{du_\epsilon^h}{d\epsilon}\bigg|_{\epsilon=0} \right) \cdot \nabla w \, dx \right) \epsilon + \left(\int_\Omega \nabla \left(\frac{du_\epsilon}{d\epsilon}\bigg|_{\epsilon=0} \right) \cdot \nabla w \, dx \right) \epsilon \\
&- \left(\int_\Omega \nabla \left(\frac{du_\epsilon}{d\epsilon}\bigg|_{\epsilon=0} \right) \cdot \nabla w \, dx \right) \epsilon - \int_{\partial\Omega} \partial_n u_0 \, w \, ds
\end{aligned}
$$

Rearranging terms to obtain the non-discretized, improved flux estimator given by Eq. (6.19),

$$
\begin{aligned}
&\widehat{Q}_\epsilon(u_\epsilon^h) - Q(u_0) \\
&= -\int_\Omega \nabla w \cdot \nabla (u_\epsilon^h - u_\epsilon)\, dx + \left(\int_\Omega \nabla \left(\frac{de_\epsilon^h}{d\epsilon}\bigg|_{\epsilon=0} \right) \cdot \nabla w \, dx \right) \epsilon \\
&+ \int_\Omega f\, w \; - \nabla w \cdot \nabla u_\epsilon \, dx - \left(\int_\Omega \nabla \left(\frac{du_\epsilon}{d\epsilon}\bigg|_{\epsilon=0} \right) \cdot \nabla w \, dx \right) \epsilon - \int_{\partial\Omega} \partial_n u_0 \, w \, ds
\end{aligned}
$$

The terms on the last line are $\mathcal{O}\left(\epsilon^2\right)$ due to the result of Theorem 6.3.1. Now adding and subtracting $\int_\Omega f\, w \, dx$, we get,

$$
\begin{aligned}
&\widehat{Q}_\epsilon(u_\epsilon^h) - Q(u_0) = \int_\Omega f\, w \; - \nabla w \cdot \nabla u_\epsilon^h \, dx - \left(\int_\Omega f\, w \; - \nabla w \cdot \nabla u_\epsilon \, dx \right) \\
&+ \left(\int_\Omega \nabla \left(\frac{de_\epsilon^h}{d\epsilon}\bigg|_{\epsilon=0} \right) \cdot \nabla w \, dx \right) \epsilon + \mathcal{O}\left(\epsilon^2\right) \\
&= \int_{\partial\Omega} \frac{u_\epsilon^h - g}{\epsilon}\, w \, ds - \int_{\partial\Omega} \frac{u_\epsilon - g}{\epsilon}\, w \, ds + \left(\int_\Omega \nabla \left(\frac{de_\epsilon^h}{d\epsilon}\bigg|_{\epsilon=0} \right) \cdot \nabla w \, dx \right) \epsilon + \mathcal{O}\left(\epsilon^2\right) \\
&= \frac{1}{\epsilon} \int_{\partial\Omega} u_\epsilon^h \, w \, ds - \frac{1}{\epsilon} \int_{\partial\Omega} u_\epsilon \, w \, ds + \left(\int_\Omega \nabla \left(\frac{de_\epsilon^h}{d\epsilon}\bigg|_{\epsilon=0} \right) \cdot \nabla w \, dx \right) \epsilon + \mathcal{O}\left(\epsilon^2\right)
\end{aligned}
$$

Finally, using the discrete version of the penalized adjoint problem given by Eq. (6.29) and the primal problem Eq. (6.12) we have,

$$
\widehat{Q}_\epsilon(u_\epsilon^h) - Q(u_0) = \int_\Omega \nabla u_\epsilon^h \cdot \nabla z_\epsilon \, dx + \frac{1}{\epsilon} \int_{\partial\Omega} u_\epsilon^h \, z_\epsilon \, ds
$$

$$
\begin{aligned}
&- \int_\Omega f\, z_\epsilon \, dx - \frac{1}{\epsilon}\int_{\partial\Omega} g\, z_\epsilon \, ds + \left(\int_\Omega \nabla \left(\left.\frac{de_\epsilon^h}{d\epsilon}\right|_{\epsilon=0}\right) \cdot \nabla w \, dx\right)\epsilon + \mathcal{O}\left(\epsilon^2\right) \\
&= -\mathcal{R}_\epsilon(u_\epsilon^h; z_\epsilon) + \left(\int_\Omega \nabla \left(\left.\frac{de_\epsilon^h}{d\epsilon}\right|_{\epsilon=0}\right) \cdot \nabla w \, dx\right)\epsilon + \mathcal{O}\left(\epsilon^2\right)
\end{aligned}
$$

Hence proved.

Remark 6.3.4. The non-improved solution flux estimator is given by,

$$
\widetilde{Q}(u_\epsilon) = \int_\Omega f\, w \; - \nabla w \cdot \nabla u_\epsilon^h \, dx \tag{6.34}
$$

If a similar analysis is carried out for this estimator we obtain the following expression for the error,

$$
\widetilde{Q}(u_\epsilon^h) - Q(u_0) = -\mathcal{R}_\epsilon(u_\epsilon^h; z_\epsilon) + \left(\int_\Omega \nabla \left(\left.\frac{du_\epsilon}{d\epsilon}\right|_{\epsilon=0}\right) \cdot \nabla w \, dx\right)\epsilon + \mathcal{O}\left(\epsilon^2\right)
$$

Note that the error for this estimator depends on the derivative of the penalty solution u_ϵ itself rather than the discretization error in the solution $u_\epsilon - u_\epsilon^h$. Thus, in the non-improved estimator it is asymptotically dominated by the $\left(\int_\Omega \nabla \left(\left.\frac{du_\epsilon}{d\epsilon}\right|_{\epsilon=0}\right) \cdot \nabla w \, dx\right)\epsilon$ term, whereas the error in the improved estimator is asymptotically dominated by the $\mathcal{R}_\epsilon(u_\epsilon^h; z_\epsilon)$ term.

Remark 6.3.5. In practical use of the improved estimator, the penalty derivative is evaluated at the value of the penalty parameter used in the simulation, rather than at $\epsilon = 0$. This introduces another source of error, however it can be shown that this error is $\mathcal{O}\left(\epsilon^2\right)$.

□

6.4 Numerical Experiments

In this section, we present numerical experiments to illustrate the use of the improved flux estimator and adjoint based mesh adaptation in the computation of the solution flux. We consider here the same two dimensional Poisson PDE that we used as a benchmark problem for code verification in Section 4.4,

$$-\nabla \cdot (\alpha \nabla u) = f \ \text{ in } \ \Omega \tag{6.35a}$$

$$u = g \ \text{ on } \ \partial\Omega \tag{6.35b}$$

where $\Omega = (0,1) \times (0,1)$ and $\alpha \in \mathbb{R}$. In the following experiments, the value of α was set to be 100. The same manufactured solution, $u(x, y; \alpha) = 4(1 - e^{-\alpha x_1} - (1 - e^{-\alpha})x_1)(x_2)(1 - x_2)$ was used to derive f. The value of the boundary data evaluated to $g = 0$. This solution has a sharp layer near the left horizontal boundary, see Figure 6.5(a). The QoI to be estimated was given by,

$$Q(u(x_1, x_2; \alpha)) = -\int_{\partial\Omega} \alpha \, \partial_n u \, w(x_1, x_2) \, ds \tag{6.36}$$

where the weight function $w(x_1, x_2)$ was given by,

$$w(x_1, x_2) = x_1 \times (1 - x_1) \times (1 - x_2) \tag{6.37}$$

This weight function has a parabolic profile on the bottom horizontal boundary and vanishes on the remaining boundaries. The exact value for the QoI was -33.2941333333, which was obtained using the symbolic toolkit in Matlab. The numerical simulations were done using `libMesh`, and we utilized the adjoint

capabilities that were described in chapter 4. Second-order Lagrange basis functions defined on quad elements were used for the approximation. The penalty parameter was given various values between 10^{-5} and 10^{-10}.

6.4.1 Comparison of the improved and naive flux estimators

We solved the model problem given by Eq. (6.35) using a penalty method. Our objective was to compare the non-improved flux estimator with the improved estimator. Our results illustrate the importance of reducing the penalty error by using derivative information. In these experiments, only uniform mesh refinement was used. For each value of the penalty parameter ϵ, we started with a coarse grid of 289 dofs and refined uniformly till we reach about a million dofs. At each refinement step, the QoI was computed and the error was obtained. We first show the results for the non-improved flux estimator. These results are plotted on a log-log regression plot showing the absolute error vs the number of dofs, see Figure 6.1 for three values of the penalty parameter: 10^{-6}, 10^{-8}, and 10^{-10}. We see that all three plots initially show a decrease in the error but later plateau at about 10^{-1}, 10^{-3}, and 10^{-5}. These numbers are all about 10^5 away from the penalty parameter values themselves, suggesting that the first derivative term in Eq. (6.19) is $\mathcal{O}\left(10^5\right)$.

To confirm this, another set of experiments were performed. The objective of these experiments was to obtain a log-log convergence plot of only the penalty error in computing the QoI versus the penalty parameter ϵ. Eq. (6.19) suggested that this plot would be linear with unit slope, and that the intercept

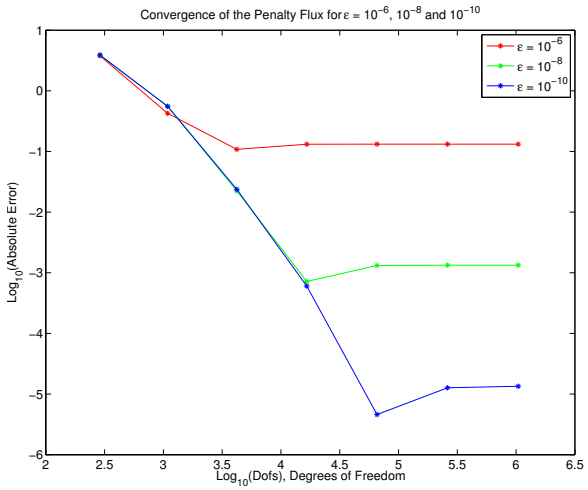


Figure 6.1: Log-log convergence plots for the boundary flux QoI computed using Eq. (6.34). Note that all the curves plateau about 5 orders of magnitude above the value of the penalty parameter they correspond to.

of such a plot would give us an estimate of the first derivative term. To ensure that the error from the Finite Element approximation was very small and that we were left mainly with the penalty error, a very fine mesh with about a million dofs was used for the computations. The penalty parameter was then progressively decreased from 10^{-5} to 10^{-10}, and the error between the computed flux and true flux was calculated for each parameter value. Figure 6.2 shows the log-log plot of the penalty error versus penalty parameter. We observe a linear relationship between the log of the penalty error and the log of the penalty parameter. The slope and intercept of the linear curve were also obtained by regression. The slope was 1.0006 and the intercept was 5.1204. Thus, we see that it is indeed the linear term in Eq. (6.19) that dominates the penalty error and confirm that the derivative term $\left(\int_\Omega \nabla\left(\frac{du_\epsilon}{d\epsilon}\right)\cdot\nabla w\big|_{\epsilon=0}\right)$

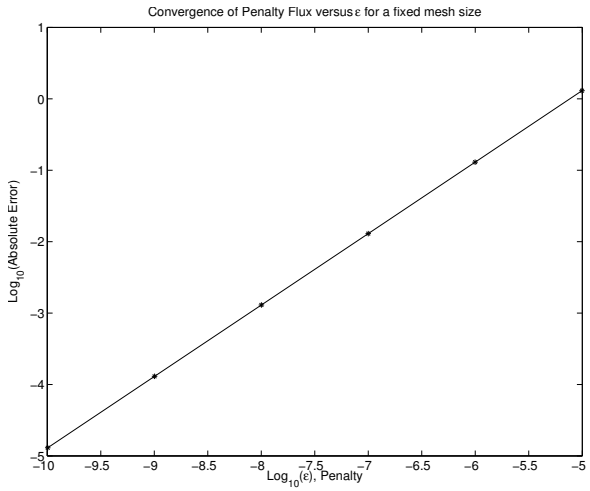


Figure 6.2: Log-log plot of the approximate error due to the use of the penalty versus the value of the penalty parameter. The intercept of this curve gives us an estimate of the magnitude of the first derivative term in Eq. (6.19).

is indeed of the order $\mathcal{O}\left(10^5\right)$.

We now move on to the results using the improved estimator. Since we cannot calculate derivatives at $\epsilon = 0$, we simply compute the first derivative improvement at $\epsilon = \epsilon_0$, where ϵ_0 is the value of the penalty parameter used for the simulation. We thus use a slightly modified improved flux estimator given by,

$$Q_\epsilon(u_\epsilon^h) = \int_\Omega f\, w \;\; - \alpha\, \nabla w \cdot \nabla u_\epsilon^h \; dx + \left(\int_\Omega \nabla \left(\left. \frac{du_\epsilon^h}{d\epsilon} \right|_{\epsilon_0 = \epsilon} \right) \cdot \nabla w \, dx \right) \epsilon_0 \quad (6.38)$$

The approximate derivative $\left(\int_\Omega \nabla \left(\left. \frac{du_\epsilon^h}{d\epsilon} \right|_{\epsilon = \epsilon_0} \right) \cdot \nabla w \, dx \right)$ was computed using the adjoint sensitivity derivative method described in Section 4.3. Figure 6.3 shows the convergence plots one obtains on using this gradient enhanced estimator. We observed improved error reduction in comparison to Figure 6.1 and

see that the convergence plots plateau near the expected regions ($\epsilon \left|\alpha \int_{\partial\Omega} \partial_n u_0 \, ds\right|$) for the $\epsilon = 10^{-6}$ and $\epsilon = 10^{-8}$ curves. However, we see no improvement in the plateau region for the $\epsilon = 10^{-10}$ curve. The reason for this is that the sensitivity derivative calculations for this parameter value were affected by roundoff error. For numerical stability, we actually obtained the sensitivity derivative to the QoI in our code using the following expression,

$$\frac{dQ}{d\epsilon} = \underbrace{\frac{dQ}{d\frac{1}{\epsilon}}}_{\text{Computed by sensitivity derivative function}} \times -\frac{1}{\epsilon^2} \tag{6.39}$$

If the actual derivative is $\mathcal{O}\left(10^5\right)$, then the code had to compute a quantity of $\mathcal{O}\left(\epsilon^2 \times 10^5\right)$. If $\epsilon = 10^{-10}$, then the quantity to be computed was $\mathcal{O}\left(10^{-15}\right)$ which is near the limit of machine precision. Thus, the derivative calculations for this value of the penalty were inaccurate. However, if $\epsilon = 10^{-8}$, then the quantity to be computed was $\mathcal{O}\left(10^{-11}\right)$, which was within machine precision.

6.4.2 Adaptive mesh refinement using adjoint techniques

In this section, we present numerical experiments that illustrate the use of adjoint-based goal-oriented mesh adaptation in the calculation of the solution flux. Theorem 6.3.2 states that the error in computing the solution flux using the improved flux is dominated by $\mathcal{R}(u_\epsilon^h; z_\epsilon)$. This residual cannot be computed since we do not have the exact penalized adjoint z_ϵ. Computing z_ϵ on a finer subspace to enable approximation of $\mathcal{R}(u_\epsilon^h; z_\epsilon)$ can be prohibitively expensive for use in adaptive mesh refinement. Therefore in our adaptive mesh refinement studies, adjoint error indicators were computed using an equal order

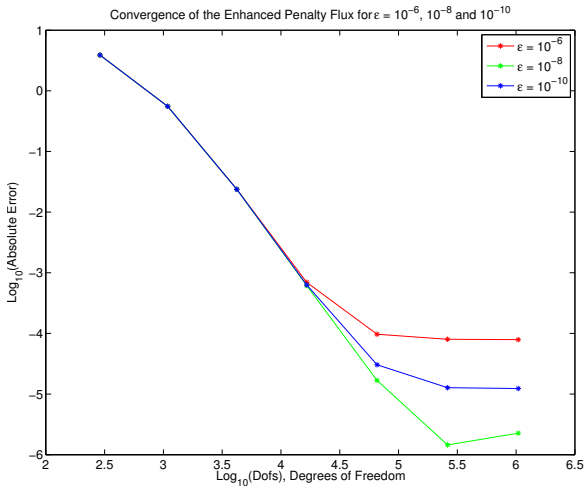


Figure 6.3: Log-log convergence plots for the boundary flux QoI computed using Eq. (6.19). The curves for $\epsilon = 10^{-6}$ and 10^{-8} plateau around $\mathcal{O}\left(\epsilon|\int_{\partial\Omega}\alpha\partial_n u_\epsilon\, ds|\right)$. However, the curve for the $\epsilon = 10^{-10}$ still plateaus around 10^{-5} due to round-off error issues in computing the sensitivity derivative at that value of ϵ.

Adjoint Residual method that we introduced and described in Section 4.2. The same adjoint solution was also used to compute the sensitivity derivative needed for the improved flux estimator.

The approximate adjoint solution was computed using the consistent adjoint formulation given by Eq. (6.29). For comparison purposes, a flux-jump error estimator [54] was also used for guiding adaptive refinement in a different experiment for the same problem. For all the simulations, the improved flux estimator was used, and the penalty parameter ϵ was fixed to be 10^{-8}, so that the errors are expected to plateau around 10^{-6}. Convergence plots are shown in Figure 6.4.

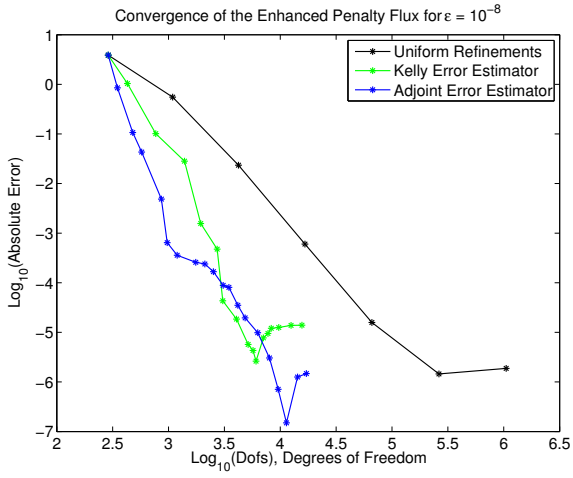
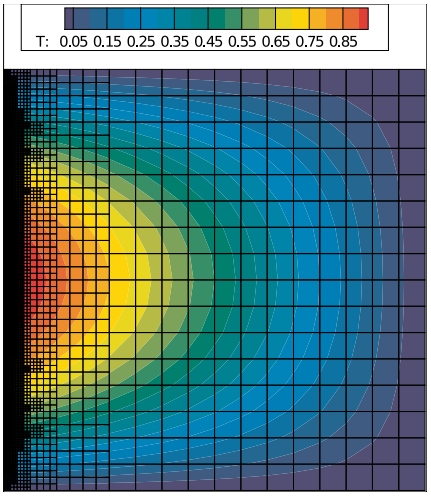
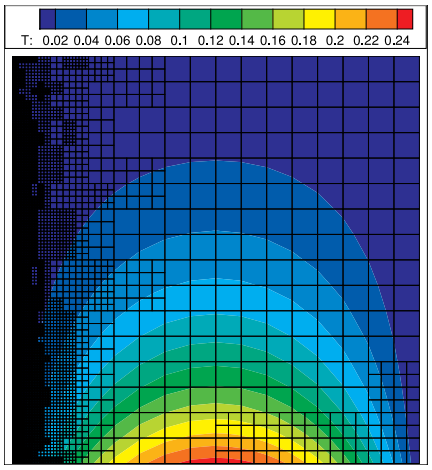


Figure 6.4: Convergence plots for the evaluation of Eq. (6.19) using various refinement strategies. The penalty was set to be 10^{-8}. Both the Kelly and Adjoint Error Indicators outperform uniform refinement. However, the Kelly estimator does not refine in the region near the QoI and hence plateaus around 10^{-5}. On the other hand, the Adjoint Error Indicator curve plateaus around the region where the uniform curve does.

As expected, the curves obtained from adaptive refinement converge faster than the one from uniform refinement. The uniform and adjoint-based adaptive curves plateau around the 10^{-6}, but the flux-jump adaptive curve plateaus around 10^{-5}. This can be understood by observing Figure 6.5 which shows the adaptive meshes obtained from the flux-jump indicator and the adjoint residual indicator. We see that both the flux-jump and adjoint-based indicators refine extensively near the boundary layer present in the primal problem. This is why the two indicators perform at about the same level till the 10^{-5} level is reached in Figure 6.4. However, the adjoint-based error indicator detects the additional error for this specific QoI arising from the



(a) The adaptive mesh obtained on using the Kelly Error Indicator superimposed on a plot of the primal solution. The mesh refinement is concentrated entirely in the boundary layer on the left.



(b) The adaptive mesh obtained on using the Adjoint Error Indicator superimposed on a plot of the dual solution. There is mesh refinement around the boundary layer as well as the QoI region near the bottom.

Figure 6.5: A comparison of the adaptive meshes obtained using the Kelly and Adjoint Residual Error Estimators. The refinement near the QoI region in the mesh for the Adjoint Residual Error Estimator allows it to eliminate an extra order of error as compared to the Kelly Error Estimator. See Figure 6.4.

bottom boundary and other parts of the domain, and accordingly refines there as seen in Figure 6.5(b). This allows the error to decrease further to the 10^{-6} level. On the other hand, the flux-jump indicator keeps refining the boundary layer and is unable to reduce the larger error contributions from other parts of the domain. Therefore, the error plateaus at the 10^{-5} mark for the flux-jump indicator.

6.5 Conclusions

We have presented an analysis of the boundary flux QoI in the context of the penalty method. We have shown that an appropriate flux for analysis is a 'penalty flux' defined by Eq. (6.10). Further, we derived a series expansion for the penalty flux in terms of the penalty parameter and showed that the penalty method can contribute a large error to the calculation of a boundary flux QoI. We then presented numerical experiments for a model Poisson problem that confirmed our theoretical results. We also showed how the accuracy of the penalty flux can be improved by the use of a higher order term based on the derivative of the QoI with respect to the penalty parameter.

An analysis of the adjoint problem for the flux QoI was also presented. We showed that the penalty flux leads to an adjoint problem that is well-posed, provides the correct error representation for the QoI, and is consistent with the corresponding adjoint problem derived in the existing literature. Finally, we illustrated the use of this adjoint problem in calculating error indicators for mesh refinement, and that such adjoint-based mesh refinement is superior to the 'flux-jump' error indicator for the boundary flux QoI.

In conclusion, one can say that if computing the flux through the boundary is an important part of the calculation, then care must be taken whenever the penalty method is used. Derivatives of the QoI to the penalty parameter should be computed, monitored and, if possible, be used to improve the accuracy of the QoI calculation.

Chapter 7

Local Sensitivity Derivative Enhanced Monte Carlo Methods

7.1 Introduction

The rapid development of high performance computing over the past five decades, along with the supporting progress in algorithmic research and software engineering has resulted in the extensive use of mathematical modeling and numerical simulation in science and engineering. The last decade has seen an increased interest in the reliability and quantification of the uncertainty inherent in the use of such complex mathematical models. The broad area of 'uncertainty quantification' (UQ) has come to be identified with development of the appropriate models that address such uncertainties, and the computational algorithms needed to address such problems.

A variety of numerical techniques have been used for UQ problems. Stochastic expansion techniques such as polynomial chaos methods [107, 106] and stochastic collocation [106, 7, 2, 3] have gained popularity due to their fast convergence rates for certain classes of problems. However, expansion based methods suffer from the "curse of dimensionality", and can be inefficient for high dimensional problems [107, 106, 103]. Also, such tecnhiques can

be intrusive to implement, especially for multiphysics codes and legacy codes [107, 103]. Expansion based techniques also require a high level of continuity in the stochastic space, otherwise they can suffer from degraded convergence rates [65]. Sampling based techniques such as the Monte Carlo method have also been applied to UQ problems. Such techniques are attractive for high dimensional problems, since their convergence rate is independent of dimension. They are also easily parallelizable given their "embarrasingly parallel" nature.

The major drawback of Monte Carlo methods is their slow convergence rate. Such methods converge with an asymptotic rate of $N_s^{-\frac{1}{2}}$, where N_s is the number of samples in a Monte Carlo study [71]. In this chapter, we will introduce a new Monte Carlo technique called Local Sensitivity Derivative Enhanced Monte Carlo (LSDEMC), which can offer improvements in the convergence rate in comparison to the plain Monte Carlo method. This new technique relies on the ability to construct more accurate surrogates using local sensitivity derivatives. Such sensitivity derivatives can be expensive to compute, especially for complex applications such as those described in chapter 2. However, the adjoint sensitivity derivative methods discussed in chapter 4 provide an inexpensive method to obtain such derivatives, even for large dimensional systems.

This chapter is organized as follows, the first section briefly discusses the Sensitivity Derivative Enhanced Monte Carlo (SDEMC) method introduced by Cao, Hussaini, and Zang [18]. The next section introduces the new LSDEMC method. We then analyze the LSDEMC method in section 7.4 and provide

some theoretical results. In section 7.5, we present numerical experiments that illustrate the performance of the LSDEMC method for a model Poisson problem and a microchannel problem. We then provide some conclusions and directions for future work.

7.2 Sensitivity Derivative Enhanced Monte Carlo

The senstivity derivative enhanced Monte Carlo (SDEMC) method was introduced by Cao, Hussaini, and Zang [18] as an improvement over the traditional Simple Random Sampling method. Consider a probability space (Ω, $\mathcal{F}$, P), where Ω is the sample space, $\mathcal{F}$ is the sigma algebra of events and P is a probability measure. Define a random variable $\boldsymbol{\xi}$: $\Omega \rightarrow \mathbb{R}$ with finite expectation $\boldsymbol{\mu_\xi}$. Let $\mathcal{S}$ denote the set $\mathbb{R}^d \times \Omega$. Let $u \in V(\mathcal{S})$ where $V(\mathcal{S}) = \{v : \mathcal{S} \rightarrow \mathbb{R}\}$, be a function over both the physical and stochastic spaces. Also, let $Q(u(\mathbf{x};\boldsymbol{\xi});\boldsymbol{\xi})$ be a functional Q: $V \times \Omega \rightarrow \mathbb{R}$, and $Q^{'}(u)$ be the derivative of Q w.r.t. $\boldsymbol{\xi}$. We have the following Taylor series approximation for $Q(u)$ around the mean $\boldsymbol{\mu_\xi}$.

$$Q_1(u;\boldsymbol{\xi}) = Q(u;\boldsymbol{\mu_\xi}) + Q^{'}(u;\boldsymbol{\mu_\xi})(\boldsymbol{\xi} - \boldsymbol{\mu_\xi}) \tag{7.1}$$

Let $\{\boldsymbol{\xi}_l\}_{l=1}^{N_{ss}}$ and $\{\boldsymbol{\xi}_i\}_{i=1}^{N_s}$ be i.i.d. samples from Ω, where N_s is the number of true samples and N_{ss} is the number of surrogate samples. Using the Taylor expansion as a surrogate one can use the following sensitivity derivative enhanced estimator to find the mean of the random variable $Q(u;\boldsymbol{\xi})$,

$$\mu_{SDEMC} = \frac{1}{N_{ss}} \sum_{l=1}^{N_{ss}} Q_1(u;\boldsymbol{\xi}_l) + \frac{1}{N_s} \sum_{i=1}^{N_s} (Q(u;\boldsymbol{\xi}_i) - Q_1(u;\boldsymbol{\xi}_i)) \tag{7.2}$$

There are N_{ss} inexpensive evaluations of the surrogate given by Eq. (7.1) and N_s expensive evaluations of the actual functional Q. Cao, Hussaini, and others show in [19] that the variance of such an estimator is lower than that of a simple sample average, if the functional Q is at least twice differentiable. Thus, SDEMC offers a more accurate estimate of the mean with very little additional cost. The SDEMC method has already been used for UQ in fluid mechanics [81] and structural mechanics [51].

7.3 Local Sensitivity Derivative Enhanced Monte Carlo

SDEMC offers more accuracy than plain Monte Carlo by allowing us to replace the expensive response function evaluations with inexpensive surrogate evaluations without adding bias. A natural question is whether an even better surrogate can be constructed using sensitivity derivative information from more points than just the mean. Indeed, the cheap derivative information that adjoint-based techniques can provide raise the prospect of using derivative information at every sample point to build surrogates. However, before we describe the details of such an estimator, we will need some preliminaries on Voronoi diagrams and Taylor series, which play a critical role in the LSDEMC method.

Voronoi Diagrams Voronoi diagrams are a decomposition of any n-dimensional space into subsets. These subsets have the property that there exists a point within each of them (which we call the Voronoi center), such

that the distance between any point in the subset and the Voronoi center is less than that between that point and any other Voronoi center. These subsets are then called Voronoi cells. Figure 7.1 shows such a nearest point Voronoi diagram in two dimensions. Such Voronoi decompositions have wide ranging

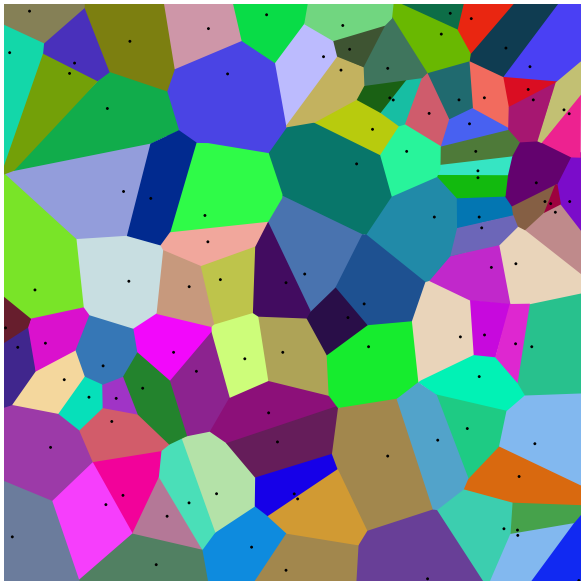


Figure 7.1: A Voronoi diagram in 2 dimensions.

applications. We now give the formal definition of a Voronoi diagram,

Definition 1. Let X be a nonempty set endowed with a distance metric *dist*. Let I be a set of indices and $(P_i)_{i\in I}$ be a collection of points in X. Consider the subsets $(R_i)_{i\in I}$, where each R_i is associated with a unique P_i and $\bigcup_{i\in I} R_i = X$. The R_i's constitute a Voronoi decomposition of X if,

$$R_i = \{x \in X \,|\, dist(x, P_i) \leq dist(x, P_j)\ \forall\, j \neq i\} \tag{7.3}$$

One method of constructing Voronoi diagrams in $\mathbb{R}^n$ is via projections of a related convex hull in $\mathbb{R}^{n+1}$ [17]. This relationship between convex hulls and Voronoi diagrams can potentially play an important part in the theoretical analysis of LSDEMC methods (see section 7.4.2). Voronoi diagrams play an important role in the construction of LSDEMC surrogates, as we shall see next.

Surrogate Construction In definition 1 let X be the domain Ω of the probability space $(\Omega, \mathcal{F}, P)$ and *dist* be the standard Euclidean distance. Also let $(P_i)_{i\in I}$ be an ensemble of randomly drawn samples (called true samples) $\{\boldsymbol{\xi}_i\}_{i=1}^{N_s}$ from Ω. Then each Voronoi cell R_i in $\{R_i\}_{i=1}^{N_s}$ is associated with a sample point from a Monte Carlo process. Now consider another set of randomly drawn samples from Ω, $\{\boldsymbol{\xi}_l\}_{l=1}^{N_{ss}}$ where $N_{ss} \geq N_s$. Suppose some $\boldsymbol{\xi}_l \in R_i$. We construct an approximation to the Quantity of Interest Q at the surrogate points $\boldsymbol{\xi}_l$ using a Taylor series about the Voronoi centre of R_i (i.e. the nearest true sample point $\boldsymbol{\xi}_i$),

$$Q_1(u;\boldsymbol{\xi}_l) = Q(u;\boldsymbol{\xi}_i) + Q^{'}(u;\boldsymbol{\xi}_i)(\boldsymbol{\xi}_l - \boldsymbol{\xi}_i) \tag{7.4}$$

Thus for local surrogate points within a Voronoi cell we construct a local surrogate using solution and derivative information from the associated Voronoi center. Such a local surrogate approximation can be constructed for every surrogate point by locating it in a particular Voronoi cell.

LSDEMC estimator for the mean In general, directly using the surrogate mean,

$$\hat{\mu} = \frac{1}{N_{ss}} \sum_{l=1}^{N_{ss}} Q_1(\boldsymbol{\xi}_l) \tag{7.5}$$

as an estimator for the expectation of Q gives us a biased estimator. The bias error becomes larger as the number of dimensions increases. To remove this bias, we split our original ensemble of samples $\{\boldsymbol{\xi}_i\}_{i=1}^{N_s}$ into N_R disjoint subsets, each of size of $\frac{N_s}{N_R}$.

$$\{\boldsymbol{\xi}_i\}_{i=1}^{N_s} = \bigcup_{r=1}^{N_R} \{\boldsymbol{\xi}_{r,j}\}_{j=1}^{N_s/N_R} \tag{7.6}$$

Each subset $\{\boldsymbol{\xi}_{r,j}\}_{j=1}^{N_s/N_R}$ has the same distribution as the original set of samples and is called a representation. Now, for each surrogate point $\boldsymbol{\xi}_l$ we will have N_R surrogates, each associated with one representation. We can form a surrogate model with each representation,

$$Q_{r,1}(u;\boldsymbol{\xi}_l) = Q(u;\boldsymbol{\xi}_{r,j(r,l)}) + Q^{'}(u;\boldsymbol{\xi}_{r,j(r,l)})(\boldsymbol{\xi}_l - \boldsymbol{\xi}_{r,j(r,l)}) \tag{7.7}$$

Here, $\boldsymbol{\xi}_{r,j(r,l)}$ is the Voronoi center nearest to $\boldsymbol{\xi}_l$ in the Voronoi diagram generated by the rth representation. Also, for any r let,

$$\{\boldsymbol{\xi}_{c,r}\}_{c=1}^{N_s - \frac{N_s}{N_R}} = \{\boldsymbol{\xi}_i\}_{i=1}^{N_s} \setminus \{\boldsymbol{\xi}_{r,j}\}_{j=1}^{N_s/N_R} \tag{7.8}$$

This is the complement set of the samples in the rth representation. We propose the following estimator for the expected value of Q,

$$\mu_{LSDEMC} = \frac{1}{N_R} \sum_{r=1}^{N_R} \left(\frac{1}{N_{ss}} \sum_{l=1}^{N_{ss}} Q_{r,1}(\boldsymbol{\xi}_l) + \frac{1}{N_s - \frac{N_s}{N_R}} \sum_{c=1}^{N_s - \frac{N_s}{N_R}} (Q(\boldsymbol{\xi}_c) - Q_{r,1}(\boldsymbol{\xi}_c)) \right) \tag{7.9}$$

In other words, we construct a surrogate function with each representation and get an approximate average using the surrogate points. We then use the points in the complement set to correct the bias. Then we average over all the representations. In the next section, we analyze the estimator given by Eq. (7.9). We will show that this estimator is unbiased and that under certain conditions, it will always converge faster than plain Monte Carlo.

7.3.1 Computational Complexity

Unlike plain Monte Carlo, the construction of the LSDEMC surrogate involves operations other than evaluations of the response function. A brute-force distance test of all Voronoi centers to find the Voronoi cell containing each surrogate sample is an $\mathcal{O}(N_s)$ operation, doing so for all N_{ss} surrogate samples would make the total complexity $\mathcal{O}(N_s N_{ss})$. Despite the superior convergence properties of LSDEMC, this cost could make it an inefficient method for statistical analysis of algebraic response functions.

However, numerical modeling of large-scale engineering systems, typically involves the inversion of very large matrices. Thus each response function evaluation has a cost of $\mathcal{O}\left(N_{dofs}^{\gamma}\right)$. Here, N_{dofs} is the so called degree of freedom (dof) count, i.e. the rank of the matrix to be inverted to numerically solve the system, while γ is the rate at which the cost of inverting the matrix scales with the dof count. This rate depends on the matrix conditioning and on the type of linear solver used to invert the matrix. Values of γ typically range from one, in cases amenable to the multigrid method [32], to approximately

two, with the more robust and widely used Generalized Minimum Residual Method (GMRES) [89].

Even with brute-force Voronoi sorting, LSDEMC can be a efficient method for uncertainty quantification of large-scale systems where

$$\mathcal{O}\left(N_s N_{ss}\right) \leq \mathcal{O}\left(N_s N_{dofs}^{\gamma}\right) \Rightarrow N_{ss} \leq N_{dofs}^{\gamma} \tag{7.10}$$

The degree of freedom count even for a simple two dimensional partial differential equation (PDE) model ranges in the tens of thousands. For large scale systems such as those used to model high speed aerodynamics, climate modeling, nuclear reactor modeling and other multiphysics systems, the degree of freedom count can be in the billions. When LSDEMC is used for such systems, the response function evaluations and the the surrogate associations can be done in parallel. Recall that although the surrogate evaluations need response function evaluation and response function derivative information, the association of surrogate points with sample points only requires the location of these points in stochastic space.

7.4 Analysis of the LSDEMC method

7.4.1 Unbiasedness

Lemma 7.4.1. *Consider a probability space $(\Omega, \mathcal{F}, P)$ and define a random variable $\boldsymbol{\xi}$: $\Omega \rightarrow \mathbb{R}$. Let $\mathcal{S}$ denote the set $\mathbb{R}^d \times \Omega$ and let $u \in V(\mathcal{S})$ where $V(\mathcal{S}) = \{v : \mathcal{S} \rightarrow \mathbb{R}\}$, be a function over physical and stochastic space. Also, let $Q(u(\mathbf{x}; \boldsymbol{\xi}); \boldsymbol{\xi})$ be a functional Q: $V \times \Omega \rightarrow \mathbb{R}$, such that Q is differentiable,*

and $E(Q) = \int_{\boldsymbol{\xi}} Q(u(\mathbf{x};\boldsymbol{\xi});\boldsymbol{\xi})dP(\boldsymbol{\xi})$ *is finite. Let* $\{\boldsymbol{\xi}_i\}_{i=1}^{N_s}$ *and* $\{\boldsymbol{\xi}_l\}_{l=1}^{N_{ss}}$ *be i.i.d. samples drawn from* Ω*. Let* $\{\boldsymbol{\xi}_{r,j}\}_{j=1}^{\frac{N_s}{N_R}}$, $\{\boldsymbol{\xi}_{c,r}\}_{c=1}^{N_s}$, *and* $Q_{r,1}(u;\boldsymbol{\xi}_l)$ *be as defined as in Eqs.* (7.6), (7.8), *and* (7.7), *respectively. Then the estimator for* $E(Q)$ *given by* (7.9) *is unbiased.*

Proof. The proof is straightforward. We have to find,

$$E\left(\frac{1}{N_R}\sum_{r=1}^{N_R}\left(\frac{1}{N_{ss}}\sum_{l=1}^{N_{ss}}Q_{r,1}(\boldsymbol{\xi}_l)+\frac{1}{N_s-\frac{N_s}{N_R}}\sum_{c=1}^{N_s-\frac{N_s}{N_R}}(Q(\boldsymbol{\xi}_c)-Q_{r,1}(\boldsymbol{\xi}_c))\right)\right)$$

Using the linearity of the expectation operator, we obtain,

$$\begin{aligned}
&= \frac{1}{N_R}\sum_{r=1}^{N_R}E\left(\frac{1}{N_{ss}}\sum_{l=1}^{N_{ss}}Q_{r,1}(\boldsymbol{\xi}_l)+\frac{1}{N_s-\frac{N_s}{N_R}}\sum_{c=1}^{N_s-\frac{N_s}{N_R}}(Q(\boldsymbol{\xi}_c)-Q_{r,1}(\boldsymbol{\xi}_c))\right)\\
&= \frac{1}{N_R}\sum_{r=1}^{N_R}\Bigg(\frac{1}{N_{ss}}\sum_{l=1}^{N_{ss}}E(Q_{r,1}(\boldsymbol{\xi}_l))\ \ldots\\
&\ldots\ +\frac{1}{N_s-\frac{N_s}{N_R}}\sum_{c=1}^{N_s-\frac{N_s}{N_R}}(E(Q(\boldsymbol{\xi}_c))-E(Q_{r,1}(\boldsymbol{\xi}_c)))\Bigg)\\
&= \frac{1}{N_R}\sum_{r=1}^{N_R}\left(\frac{1}{N_{ss}}(N_{ss}E(Q_{r,1}))+\frac{1}{N_s-\frac{N_s}{N_R}}\left(N_s-\frac{N_s}{N_R}(E(Q)-E(Q_{r,1}))\right)\right)\\
&= \frac{1}{N_R}\sum_{r=1}^{N_R}\cancel{E(Q_{r,1})}+E(Q)-\cancel{E(Q_{r,1})}\\
&= E(Q) \qquad (7.11)
\end{aligned}$$

Hence proved. □

7.4.2 The Asymptotic Distribution for LSDEMC

General Taylor Series Expansions Before we derive the asymptotic distribution for the LSDEMC estimator we state the following generalization of Taylor's theorem [12] for the multi-variable case.

Theorem 7.4.2. *Let f: $\mathbb{R}^n \to \mathbb{R}$ be a k times continuously differentiable function in the closed ball B. Introduce the multi-index notation, $|\alpha| = \alpha_1 + \alpha_2 + ... + \alpha_n$, $\alpha! = \alpha_1!\alpha_2!...\alpha_n!$, $\mathbf{x}^\alpha = x_1^{\alpha_1}x_2^{\alpha_2}...x_n^{\alpha_n}$, $D^\alpha f(\mathbf{x}) = \frac{\partial^{|\alpha|}f}{\partial x_1^{\alpha_1}\partial x_2^{\alpha_2}...\partial x_n^{\alpha_n}}$, where α, $\alpha_i \in \mathbb{N}$ and $\mathbf{x} \in B$. Let $\mathbf{a} \in B$. We then have the following Taylor series of order k for f around $\mathbf{a}$,*

$$f(\mathbf{x}) = \sum_{|\alpha|=0}^{k} \frac{D^\alpha f(\mathbf{a})}{\alpha!}(\mathbf{x}-\mathbf{a})^\alpha + \sum_{|\alpha|=k} h_\alpha(\mathbf{x})(\mathbf{x}-\mathbf{a})^\alpha$$
$$\text{where } \lim_{\mathbf{x}\to\mathbf{a}} h_\alpha(\mathbf{x}) = 0 \tag{7.12}$$

We will derive the asymptotic distribution for the LSDEMC estimator by comparing it with the unbiased estimator given by,

$$\tilde{\mu} = \sum_{l=1}^{N_{ss}} Q(\boldsymbol{\xi}_l) \tag{7.13}$$

where $\boldsymbol{\xi}_l$ are as in Lemma 7.4.1. First we consider the error in any evalution of the surrogate function,

$$\begin{aligned} &Q_{r,1}(\boldsymbol{\xi}_l) - Q(\boldsymbol{\xi}_l) = -(Q(\boldsymbol{\xi}_l) - Q_{r,1}(\boldsymbol{\xi}_l)) \\ &= -\sum_{|\alpha|=1} h_\alpha(\boldsymbol{\xi}_l)(\boldsymbol{\xi}_l - \boldsymbol{\xi}_{r,j(r,l)})^\alpha \text{ Using Theorem 7.4.2, Eq. (7.7)} \end{aligned} \tag{7.14}$$

where $h_\alpha(\boldsymbol{\xi}_l)$ is as defined in Theorem 7.4.2. Therefore the difference between the biased estimator given by Eq. (7.5) and the unbiased estimator given by Eq. (7.13) is,

$$\frac{1}{N_{ss}}\sum_{l=1}^{N_{ss}} Q_{r,1}(\boldsymbol{\xi}_l) - \frac{1}{N_{ss}}\sum_{j=1}^{N_{ss}} Q(\boldsymbol{\xi}_l) = \frac{1}{N_{ss}}\sum_{j=1}^{N_{ss}} - \sum_{|\alpha|=1} h_\alpha(\boldsymbol{\xi}_l)(\boldsymbol{\xi}_l - \boldsymbol{\xi}_{r,j(r,l)})^\alpha \quad (7.15)$$

We can now get an expression for the error if the LSDEMC estimator given by Eq. (7.9) is used. Subtracting the estimator given by Eq. (7.13) from the LSDEMC estimator we get,

$$\begin{aligned}
&\frac{1}{N_R}\sum_{r=1}^{N_R}\left(\frac{1}{N_{ss}}\sum_{l=1}^{N_{ss}} Q_{r,1}(\boldsymbol{\xi}_l) + \frac{1}{N_s - \frac{N_s}{N_R}}\sum_{c=1}^{N_s-\frac{N_s}{N_R}} (Q(\boldsymbol{\xi}_c) - Q_{r,1}(\boldsymbol{\xi}_c))\right) \\
&- \frac{1}{N_{ss}}\sum_{j=1}^{N_{ss}} Q(\boldsymbol{\xi}_l) \\
&= \frac{1}{N_R}\sum_{r=1}^{N_R}\left(\frac{1}{N_{ss}}\sum_{l=1}^{N_{ss}} Q_{r,1}(\boldsymbol{\xi}_l) - \frac{1}{N_{ss}}\sum_{j=1}^{N_{ss}} Q(\boldsymbol{\xi}_l) \ldots\right. \\
&\left. \ldots + \frac{1}{N_s - \frac{N_s}{N_R}}\sum_{c=1}^{N_s-\frac{N_s}{N_R}} (Q(\boldsymbol{\xi}_c) - Q_{r,1}(\boldsymbol{\xi}_c))\right)
\end{aligned} \quad (7.16)$$

Using Eq. (7.15), we see that the above expression is equal to,

$$\begin{aligned}
&\frac{1}{N_R}\sum_{r=1}^{N_R}\left(\frac{1}{N_{ss}}\sum_{l=1}^{N_{ss}}\left(-\sum_{|\alpha|=1} h_\alpha(\boldsymbol{\xi}_l)(\boldsymbol{\xi}_l - \boldsymbol{\xi}_{r,l(r)})^\alpha\right) \ldots.\right. \\
&\left. \ldots + \frac{1}{N_s - \frac{N_s}{N_R}}\sum_{c=1}^{N_s-\frac{N_s}{N_R}}\left(\sum_{|\alpha|=1} h_\alpha(\boldsymbol{\xi}_c)(\boldsymbol{\xi}_c - \boldsymbol{\xi}_{r,c(r)})^\alpha\right)\right]
\end{aligned} \quad (7.17)$$

The error in the LSDEMC estimator is given by,

$$\frac{1}{N_R}\sum_{r=1}^{N_R}\left(\frac{1}{N_{ss}}\sum_{l=1}^{N_{ss}}Q_{r,1}(\boldsymbol{\xi}_l)+\frac{1}{N_s-\frac{N_s}{N_R}}\sum_{c=1}^{N_s-\frac{N_s}{N_R}}\left(Q(\boldsymbol{\xi}_c)-Q_{r,1}(\boldsymbol{\xi}_c)\right)\right)-E(Q) \tag{7.18}$$

Using Eqs. (7.16) and (7.17), we see that Eq. (7.18) can be written as the sum of two error components,

$$\begin{aligned}&\frac{1}{N_{ss}}\sum_{j=1}^{N_{ss}}Q(\boldsymbol{\xi}_l)-E(Q)\\&+\frac{1}{N_R}\sum_{i=1}^{N_R}\left[\frac{1}{N_{ss}}\sum_{l=1}^{N_{ss}}-\sum_{|\alpha|=1}\left(h_\alpha(\boldsymbol{\xi}_l)(\boldsymbol{\xi}_l-\boldsymbol{\xi}_{r,l(r)})^\alpha\right)\ \ldots\right.\\&\left.\ldots+\frac{1}{N_s-\frac{N_s}{N_R}}\sum_{c=1}^{N_s-\frac{N_s}{N_R}}\sum_{|\alpha|=1}\left(h_\alpha(\boldsymbol{\xi}_c)(\boldsymbol{\xi}_c-\boldsymbol{\xi}_{r,c(r)})^\alpha\right)\right]\end{aligned} \tag{7.19}$$

If Q is integrable, the first error component in Eq. (7.18) goes to zero as we increase the number of surrogate samples N_{ss}, in accordance with the Strong Law of Large Numbers. If Q also has finite variance $V(Q)$, then by the Central Limit Theorem,

$$\frac{1}{N_{ss}}\sum_{j=1}^{N_{ss}}Q(\boldsymbol{\xi}_l)-E(Q)\xrightarrow{d}N\left(0,\frac{V(Q)}{N_{ss}}\right) \tag{7.20}$$

The second component of the error is harder to analyze. Complete analysis of this part of the error will require some new results in stochastic geometry. These results are related to the moments of the asymptotic width of Voronoi cells in random point processes. Such an analysis is beyond the scope of this dissertation. Instead, we will prove a lemma which will show that, under

certain conditions, the asymptotic distribution of the second error component has a variance that is lower than that for the plain Monte Carlo method.

Lemma 7.4.3. *Let the hypotheses of Lemma 7.4.1 hold. Also, let the random variables* $\sum_{|\alpha|=1} h_\alpha(\boldsymbol{\xi})(\boldsymbol{\xi}-\boldsymbol{\xi}_{r,i})^\alpha$ *have finite expectation* ($\widehat{E}_{r,i}$) *and variance* ($\widehat{V}_{r,i}$). *Assume that* N_s *and* N_R *are increased such that* $\frac{N_s}{N_R} = C$, *where* $C \in \mathbb{N}$ *is a finite constant. Then we have,*

$$\begin{aligned}
&\frac{1}{N_R}\sum_{r=1}^{N_R}\Bigg[\frac{1}{N_{ss}}\sum_{l=1}^{N_{ss}}\left(-\sum_{|\alpha|=1} h_\alpha(\boldsymbol{\xi}_l)(\boldsymbol{\xi}_l-\boldsymbol{\xi}_{r,l(r)})^\alpha\right) \ \dots \\
&\dots + \frac{1}{N_s-\frac{N_s}{N_R}}\sum_{c=1}^{N_s-\frac{N_s}{N_R}}\left(\sum_{|\alpha|=1} h_\alpha(\boldsymbol{\xi}_c)(\boldsymbol{\xi}_c-\boldsymbol{\xi}_{r,c(r)})^\alpha\right)\Bigg] \\
&\xrightarrow{d} N\Bigg(0, \frac{1}{N_{ss}N_R}\frac{1}{N_R}\sum_{r=1}^{N_R}\left[\frac{1}{N_{ss}}\sum_{l=1}^{N_{ss}}\widehat{V}_{r,l}\right] \ \dots \\
&\dots + \frac{1}{N_s(N_R-1)}\frac{1}{N_R}\sum_{r=1}^{N_R}\left(\frac{1}{N_s-\frac{N_s}{N_R}}\sum_{c=1}^{N_s-\frac{N_s}{N_R}}\widehat{V}_{r,c}\right)\Bigg)
\end{aligned} \tag{7.21}$$

Proof. We can rewrite Eq. (7.18) using summation of the surrogate and complement samples over the Voronoi cells $(R_i)_{i\in I}$, weighted by the fraction of surrogate and complement samples in each cell:

$$\begin{aligned}
&\frac{1}{N_R}\sum_{r=1}^{N_R}\Bigg(-\sum_{i=1}^{\frac{N_s}{N_R}}\sum_{j=1}^{N_{ss}^i}\frac{\sum_{|\alpha|=1} h_\alpha(\boldsymbol{\xi}_j)(\boldsymbol{\xi}_j-\boldsymbol{\xi}_{r,i})^\alpha}{N_{ss}^i}\frac{N_{ss}^i}{N_{ss}} \ \dots \\
&\dots + \sum_{i=1}^{\frac{N_s}{N_R}}\sum_{m=1}^{N_c^i}\frac{\sum_{|\alpha|=1} h_\alpha(\boldsymbol{\xi}_m)(\boldsymbol{\xi}_m-\boldsymbol{\xi}_{r,i})^\alpha}{N_c^i}\frac{N_c^i}{N_s-\frac{N_s}{N_R}}\Bigg)
\end{aligned} \tag{7.22}$$

We can now consider the asymptotic behaviour of the local sums in each Voronoi cell, $\sum_{j=1}^{N_{ss}^i} \frac{\sum_{|\alpha|=1} h_\alpha(\boldsymbol{\xi}_j)(\boldsymbol{\xi}_j-\boldsymbol{\xi}_{r,i})^\alpha}{N_{ss}^i}$ and $\sum_{m=1}^{N_c^i} \frac{\sum_{|\alpha|=1} h_\alpha(\boldsymbol{\xi}_m)(\boldsymbol{\xi}_m-\boldsymbol{\xi}_{r,i})^\alpha}{N_c^i}$. The local surrogate samples N_{ss}^i and complement set samples N_c^i are i.i.d. Since the random variables $\sum_{|\alpha|=1} h_\alpha(\boldsymbol{\xi})(\boldsymbol{\xi}-\boldsymbol{\xi}_{r,i})^\alpha$ have finite expectation ($\widehat{E}_{r,i}$) and variance ($\widehat{V}_{r,i}$), by the Central Limit Theorem we have,

$$\sum_{j=1}^{N_{ss}^i} \frac{\sum_{|\alpha|=1} h_\alpha(\boldsymbol{\xi}_j)(\boldsymbol{\xi}_j-\boldsymbol{\xi}_{r,i})^\alpha}{N_{ss}^i} \xrightarrow{d} N\left(\widehat{E}_{r,i}, \frac{\widehat{V}_{r,i}}{N_{ss}^i}\right) \tag{7.23a}$$

$$\sum_{m=1}^{N_c^i} \frac{\sum_{|\alpha|=1} h_\alpha(\boldsymbol{\xi}_m)(\boldsymbol{\xi}_m-\boldsymbol{\xi}_{r,i})^\alpha}{N_c^i} \xrightarrow{d} N\left(\widehat{E}_{r,i}, \frac{\widehat{V}_{r,i}}{N_c^i}\right) \tag{7.23b}$$

Using Eqs. (7.23a) and (7.23b) in Eq. (7.22), we obtain,

$$\begin{aligned}
&\frac{1}{N_R}\sum_{r=1}^{N_R}\left[-\sum_{i=1}^{\frac{N_s}{N_R}} N\left(\widehat{E}_{r,i}, \frac{\widehat{V}_{r,i}}{N_{ss}^i}\right)\frac{N_{ss}^i}{N_{ss}} + \sum_{i=1}^{\frac{N_s}{N_R}} N\left(\widehat{E}_{r,i}, \frac{\widehat{V}_{r,i}}{N_c^i}\right)\frac{N_m^i}{N_s-\frac{N_s}{N_R}}\right] \\
&= \frac{1}{N_R}\sum_{r=1}^{N_R}\left[\sum_{i=1}^{\frac{N_s}{N_R}} N\left(-\widehat{E}_{r,i}\frac{N_{ss}^i}{N_{ss}}, \frac{\widehat{V}_{r,i}}{N_{ss}^i}\left(\frac{N_{ss}^i}{N_{ss}}\right)^2\right) \ \ldots\right. \\
&\ldots \left. + \sum_{i=1}^{\frac{N_s}{N_R}} N\left(\widehat{E}_{r,i}\frac{N_m^i}{N_s-\frac{N_s}{N_R}}, \frac{\widehat{V}_{r,i}}{N_c^i}\left(\frac{N_m^i}{N_s-\frac{N_s}{N_R}}\right)^2\right)\right] \\
&= \frac{1}{N_R}\sum_{r=1}^{N_R}\left[N\left(\sum_{i=1}^{\frac{N_s}{N_R}} -\widehat{E}_{r,i}\frac{N_{ss}^i}{N_{ss}}, \sum_{i=1}^{\frac{N_s}{N_R}}\frac{\widehat{V}_{r,i}}{N_{ss}^2}N_{ss}^i\right) \ \ldots\right. \\
&\ldots \left. + N\left(\sum_{i=1}^{\frac{N_s}{N_R}}\widehat{E}_{r,i}\frac{N_m^i}{N_s-\frac{N_s}{N_R}}, \sum_{i=1}^{\frac{N_s}{N_R}}\frac{\widehat{V}_{r,i}}{\left(N_s-\frac{N_s}{N_R}\right)^2}N_m^i\right)\right] \\
&= \frac{1}{N_R}\sum_{r=1}^{N_R}\left[N\left(-\sum_{l=1}^{N_{ss}}\frac{\widehat{E}_{r,l}}{N_{ss}}, \frac{1}{N_{ss}}\sum_{l=1}^{N_{ss}}\frac{\widehat{V}_{r,l}}{N_{ss}}\right) \ \ldots\right.
\end{aligned}$$

$$\ldots + N\left(\sum_{c=1}^{N_s-\frac{N_s}{N_R}} \frac{\widehat{E}_{r,c}}{N_s-\frac{N_s}{N_R}}, \frac{1}{N_s-\frac{N_s}{N_R}} \sum_{c=1}^{N_s-\frac{N_s}{N_R}} \frac{\widehat{V}_{r,c}}{N_s-\frac{N_s}{N_R}}\right)\Bigg]$$

$$= \frac{1}{N_R}\sum_{r=1}^{N_R} N\Bigg(-\sum_{l=1}^{N_{ss}} \frac{\widehat{E}_{r,l}}{N_{ss}} + \sum_{c=1}^{N_s-\frac{N_s}{N_R}} \frac{\widehat{E}_{r,c}}{N_s-\frac{N_s}{N_R}}, \ldots$$

$$\ldots \frac{1}{N_s-\frac{N_s}{N_R}} \sum_{c=1}^{N_s-\frac{N_s}{N_R}} \frac{\widehat{V}_{r,c}}{N_s-\frac{N_s}{N_R}} + \frac{1}{N_{ss}}\sum_{l=1}^{N_{ss}} \frac{\widehat{V}_{r,l}}{N_{ss}}\Bigg) \tag{7.24}$$

Since the LSDEMC estimator is unbiased (Lemma 7.4.1), the mean of the normal distribution in Eq. (7.24) converges to zero. Therefore, Eq. (7.24) converges to,

$$\xrightarrow{d} \frac{1}{N_R}\sum_{i=1}^{N_R} N\left(0, \frac{1}{N_s-\frac{N_s}{N_R}} \sum_{c=1}^{N_s-\frac{N_s}{N_R}} \frac{\widehat{V}_{r,c}}{\left(N_s-\frac{N_s}{N_R}\right)} + \frac{1}{N_{ss}}\sum_{l=1}^{N_{ss}} \frac{\widehat{V}_{r,l}}{N_{ss}}\right)$$

$$= N\Bigg(0, \frac{1}{N_s(N_R-1)}\left(\frac{1}{N_R}\sum_{i=1}^{N_R}\sum_{c=1}^{N_s-\frac{N_s}{N_R}} \frac{\widehat{V}_{r,c}}{\left(N_s-\frac{N_s}{N_R}\right)}\right) \ldots$$

$$\ldots + \frac{1}{N_{ss}N_R}\left(\frac{1}{N_R}\sum_{i=1}^{N_R}\sum_{l=1}^{N_{ss}} \frac{\widehat{V}_{r,l}}{N_{ss}}\right)\Bigg) \tag{7.25}$$

Hence proved. □

We see that in the case of holding the ratio of true samples to representations constant $\left(\frac{N_S}{N_R} = C\right)$, the asymptotic variance of the LSDEMC estimator is given by,

$$\frac{V(Q)}{N_{ss}} + \frac{1}{N_s(N_R-1)}\left(\frac{1}{N_R}\sum_{i=1}^{N_R}\sum_{c=1}^{N_s-\frac{N_s}{N_R}} \frac{\widehat{V}_{r,c}}{\left(N_s-\frac{N_s}{N_R}\right)}\right)$$

$$+\frac{1}{N_{ss}N_R}\left(\frac{1}{N_R}\sum_{i=1}^{N_R}\sum_{l=1}^{N_{ss}}\frac{\widehat{V}_{r,l}}{N_{ss}}\right) \tag{7.26}$$

Since we can make N_{ss} very large, we can neglect the contribution of the terms having N_{ss} in the denominator. The error is therefore dominated by $\frac{1}{N_s(N_R-1)}\left(\frac{1}{N_R}\sum_{i=1}^{N_R}\sum_{c=1}^{N_s-\frac{N_s}{N_R}}\frac{\widehat{V}_{r,c}}{\left(N_s-\frac{N_s}{N_R}\right)}\right)$. The error thus depends on the true number of samples N_s, the number of representations N_R, and an averaged variance of the random variable $\sum_{|\alpha|=1} h_\alpha(\boldsymbol{\xi})(\boldsymbol{\xi}-\boldsymbol{\xi}_{r,i})^\alpha$.

Although the restriction $\left(\frac{N_S}{N_R}=C\right)$ was necessary to prove Lemma 7.4.3, we anticipate that a similar result will hold even if we let N_R be fixed. Indeed, this restriction has not been seen as necessary during our numerical experiments, where we have obtained good results with the number of representations being fixed. Therefore, if we assume that the LSDEMC error is still dominated by a term of the form $\frac{1}{N_s(N_R-1)}\left(\frac{1}{N_R}\sum_{i=1}^{N_R}\sum_{c=1}^{N_s-\frac{N_s}{N_R}}\frac{\widehat{V}_{r,c}}{\left(N_s-\frac{N_s}{N_R}\right)}\right)$, we can attempt further analysis of the error. In this case, since N_R is fixed, the asymptotic structure of the Voronoi diagrams corresponding to each representation will be the same in distribution. We can thus drop the dependance of $\widehat{V}_{r,c}$ on r and the averaging over all representations, leading us to the expression $\frac{1}{N_s(N_R-1)}\sum_{c=1}^{N_s-\frac{N_s}{N_R}}\frac{\widehat{V_c}}{\left(N_s-\frac{N_s}{N_R}\right)}$.

The term $\sum_{c=1}^{N_s-\frac{N_s}{N_R}}\frac{\widehat{V_c}}{\left(N_s-\frac{N_s}{N_R}\right)}$ is the average variance of $\sum_{|\alpha|=1} h_\alpha(\boldsymbol{\xi})(\boldsymbol{\xi}-\boldsymbol{\xi}_i)^\alpha$ over all Voronoi cells for a Voronoi diagram with $N_s-\frac{N_s}{N_R}$ generating points. We anticipate that a Central Limit Theorem for such a weighted moment of

Voronoi cell diameters generated by a random point process should hold. Central Limit Theorems for the diameter of convex hulls of points generated via normal [102], uniform [102], and Poisson [84] point processes in $\mathbb{R}^d$ have been recently derived and presented in the stochastic geometry literature. The deep connection between convex hulls and Voronoi diagrams [17] that we mentioned earlier indicates that similar Central Limit Theorems may apply to the diameters of Voronoi cells. Personal communication with experts in the field of stochastic geometry [85] indicate that such results should be within reach using the Efron-Stein jackknife inequality [9] and Stein's method [94]. However, they are beyond the scope of this dissertation.

It is conjectured that for a uniform point process the Voronoi diameter will converge to zero as $\frac{1}{(N_s)^{\frac{2}{d-1}}}$ [85]. Similar results should hold for other point processes. Therefore, it can be conjectured that the asymptotic error for the LSDEMC estimator approaches zero as $\frac{1}{N_s^{\frac{1}{2}+\frac{f(d)}{d}}}$ where d is the dimension of the stochastic space and $\lim_{d\to\infty} \frac{f(d)}{d} = 0$. Thus, the LSDEMC estimator will always converge faster than plain Monte Carlo, however the magnitude of improvement over plain Monte Carlo will diminish as the number of stochastic dimensions increases. We now move on to numerical experiments that illustrate the improvements made possible by the use of LSDEMC.

7.5 Numerical Experiments

Two sets of numerical experiments were performed to compare the performance of LSDEMC with plain MC. The two experiments were as follows:

1. The calculation of the mean for an exponential response function having 1, 8, and 32 random arguments, all of which were distributed normally. The exact mean for such a problem can be evaluated analytically, and was used to prepare convergence plots.

2. The calculation of the mean for a QoI in a model Poisson problem with two random parameters, both distributed normally. For this problem, we were able to compute a 'true' mean with the help of analytic results and the `dblquad` function of Matlab.

In all experiments, samples were generated using Simple Random Sampling and one hundred trials were performed at each sample size. Since we had the exact answer for the first two sets of experiments, we could obtain convergence rates using plots and linear regression. The last experiment also illustrates the importance of adjoint-based techniques in Finite Element analysis, both for error control (so the FE error does not pollute the calculation of the mean) and adjoint sensitivity analysis (to obtain cheap sensitivity derivatives for LS-DEMC to be computationally feasible).

7.5.1 Multiparameter Exponential Response Function

For the first test case, we consider the exponential response function,

$$Q(\boldsymbol{\xi}) = e^{\sum_{i=1}^{d} \xi_i} \tag{7.27}$$

and normally-distributed input parameters

$$\xi_i \equiv \mathcal{N}\left(\frac{\mu_{input}}{d}, \frac{\sigma^2_{input}}{d}\right) \tag{7.28}$$

$Q(\boldsymbol{\xi})$ is then distributed lognormally; error convergence plots which follow are based on the analytic expressions for its mean and standard deviation. We chose values of 1 and 0.1 for the parameters μ_{input} and σ_{input}. To evaluate the performance of the algorithm with increasing number of dimensions, we did experiments with dimension d as 1, 8, and 32. For each dimensional size, the number of true samples were increased from 32 to 512, with a factor of 2 increment at each step, while the number of representations N_R was chosen to be 2. The number of surrogate samples N_{ss} at each step was the square of the number of true samples N_s^2.

Figure 7.2 shows the log-log convergence plots for the one, eight and thirty-two dimensional cases. We observe that the LSDEMC method comfortably outperforms plain MC, converging at a faster rate. Regression analysis gave a convergence rate around 0.5 for the plain MC method for all three experiments, while the rate for LSDEMC decreased from 1.04 in the one-dimensional case to 0.82 in the thirty-two dimensional case. Table 7.1 below summarizes the results from the three experiments. We see that LSDEMC maintains a superior rate and convergence performance in comparison to plain MC even for moderately high dimensional problems. However, as we conjectured earlier, the improvement in the convergence rate decreases as we increase the number of dimensions.

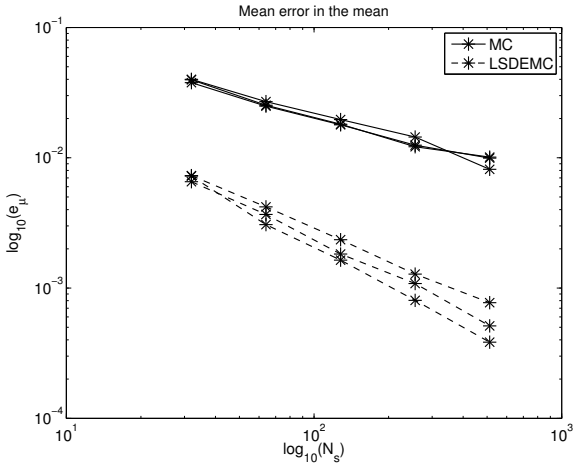


Figure 7.2: Comparison of MC and LSDEMC methods for computing the mean of the response function given by Eq. (7.27) with 1, 8, and 32 dimensional versions of the distribution given by Eq. (7.28). The input mean was 1 and standard deviation 0.1.

Table 7.1: Rates of convergence for MC and LSDEMC simulations for the calculation of the mean of a response function given by Eq. (7.27) with distribution given by Eq. (7.28)

d	MC	LSDEMC
1	0.55	1.04
8	0.5	0.91
32	0.5	0.82

7.5.2 Model Poisson Problem

We now turn our attention to a model PDE test case. We use the Poisson problem given by Eq. (4.33) as our model PDE. However, we change the definition of the PDE slightly to make the problem dependent on two

parameters. Accordingly, we choose our manufactured solution to be,

$$u(x_1, x_2; \alpha_1, \alpha_2) = \alpha_2(4(1 - e^{-\alpha_1 x_1} - (1 - e^{-\alpha_1})x_1)(x_2)(1 - x_2)) \tag{7.29}$$

As before, our model Poisson problem is given by,

$$-\alpha_1 \Delta u = f \tag{7.30}$$

where the forcing function is now obtained by differentiating the manufactured solution given by Eq. (7.29). The QoI was given by Eq. (4.35). The parameters α_1 and α_2 were both distributed as truncated normal random variables, with parameters given by,

$$\alpha_1 \in [50, 150],\ \mu_{\alpha_1} = 100,\ \sigma_{\alpha_1} = 10 \tag{7.31}$$

$$\alpha_2 \in [0.5, 1.5],\ \mu_{\alpha_2} = 1,\ \sigma_{\alpha_2} = 0.1 \tag{7.32}$$

For this two-dimensional (in stochastic space) problem, the mean of the QoI functional could be computed to a high precision using numerical quadrature. Using the `dblquad` function of Matlab, the 'true' mean was computed to be -32.66663.

Approximations to the mean were calculated using the plain MC and LSDEMC method in combination with an adjoint-based adaptive FE strategy. Second-order Lagrange basis functions on quadrilateral elements were used for the FE solution. An adjoint residual based adaptive FE strategy was used to obtain adaptive grids tailored to reduce the error in the QoI. Adjoint sensitivity analysis was used to obtain the sensitivities of the QoI to each

parameter at all sample points efficiently. More details about these strategies can be found in sections 4.4.3 and 4.4.4 of chapter 4. The MC samples were generated using the Sandia National Labs stochastic toolkit Dakota [1], which was coupled to `libMesh`. The parallel processing abilities of Dakota were used to perform the independent MC evaluations in parallel. The LSDEMC analysis was performed in Matlab by processing the output data containing the QoI values and derivatives obtained from Dakota.

Starting with 40 true samples, the number of samples was increased to 320 samples by doubling the number of samples at each step. One hundred trials were performed at each step, and the average absolute error was computed for both the MC and LSDEMC strategies. These errors were plotted against the number of samples in a log-log plot as shown in Figure 7.3. We see that LSDEMC delivers better convergence performance than plain MC. The rate of convergence for the plain MC method was 0.52, while LSDEMC converged at a rate of 1.005. This seems better than the rates observed with the algebraic response function. We speculate that this superior rate has been obtained because the random parameters in this experiment have truncated normal distributions, which have compact support. The standard normal distributions used for the exponential response function do not have compact support, which could lead to Voronoi cells with larger diameters.

Since we had analytic expressions for the QoI and its sensitivity derivatives, we could monitor the FE error at each sample point. The maximum FE errors for Q, $\frac{dQ}{d\alpha_1}$ and $\frac{dQ}{d\alpha_2}$ over the entire simulation process were 6.67e-5, 3.0e-6

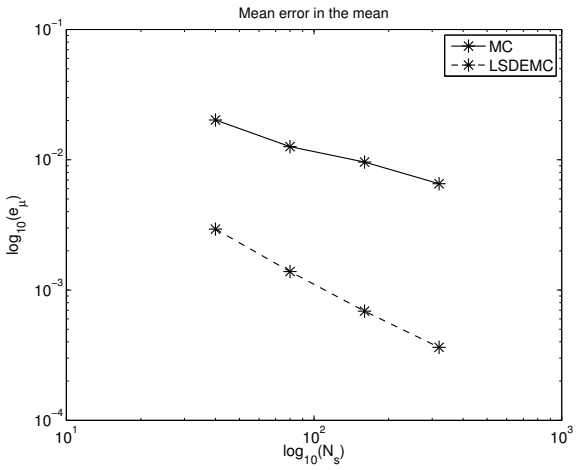


Figure 7.3: Comparison of MC and LSDEMC methods for computing the mean of the QoI given by Eq. (4.35). There were two random parameters whose distributions were given by Eq. (7.31).

and 6.19e-5, indicating that the adjoint-based methods gave us good meshes and accurate sensitivity derivatives. Thus, in conjunction with adjoint-based mesh refinement and sensitivity analysis, the Local Sensitivity Derivative Enhanced Monte Carlo method can be a fast and accurate method for UQ.

7.6 Conclusion

A new Monte Carlo algorithm for numerical integration has been developed and presented. The new method, called Local Sensitivity Derivative Enhanced Monte Carlo (LSDEMC) utilizes local senstivity information to build surrogates for inexpensive integration. Voronoi decompositions of the stochastic space centred on true sample points are used for surrogate evaluation, ensuring asymptotic accuracy. It has also been shown that the LSDEMC

estimator is unbiased. The superior convergence of the LSDEMC method in comparison to plain Monte Carlo methods has also been shown through theoretical analysis, under the assumption of a constant sample number to representation number ratio. Numerical experiments using an exponential response function and a model Poisson problem were conducted to compare the LSDEMC and plain MC method. The results of these experiments indicate that the LSDEMC method converges at a faster rate than plain MC, and that the improvement in the rate diminishes with increasing dimension.

An important consideration for further analysis of the LSDEMC method is the general case where the ratio of the number of true samples to the number of representations is not held constant. The numerical results demonstrate that results similar to those derived for the constant ratio case should hold. Knowing the asymptotic distribution for LSDEMC can give us better variance estimates for the LSDEMC estimator and quantify the dependence of the rate of convergence on dimension size. Another critical component needed for LSDEMC analysis is a Central Limit Theorem for weighted moments of Voronoi cells generated by random point processes. The derivation of such results is beyond the scope of this dissertation and has been left for future work.

Chapter 8

Hierarchical Incremental Latin Hypercube Sampling

8.1 Introduction

The Monte Carlo method is a widely used statistical integration technique [52]. The simplicity and robustness of Monte Carlo algorithms together with the dimension independent nature of error reduction for such algorithms make them very suitable for the analysis of complex systems with a large number of input parameters. The advent of parallel computing has further increased the attractiveness of such algorithms [73], given their "embarassingly parallel" nature. Plain Monte Carlo methods converge with an asymptotic rate of $N_s^{-\frac{1}{2}}$, where N_s is the number of samples in a Monte Carlo study [71]. This slow rate of convergence is one of the main limitations of the Monte Carlo method, inhibiting its use in some computationally intensive applications.

Various strategies have been proposed and used for improving convergence properties of the Monte Carlo method. These include modified sampling techniques such as Latin Hypercube Sampling (LHS) [95] and Hammersley sampling [1]. Latin Hypercube Sampling has found widespread acceptance and application. For general response functions, LHS retains the $N_s^{-\frac{1}{2}}$ rate of

convergence of Simple Random Sampling (SRS) [95], but LHS can substantially improve the constant of convergence [72]. Latin Hypercube Sampling implementations are used extensively in Uncertainty Quantification [47].

Due to the large computational cost of simulating modern engineering systems, it is desirable to obtain an accurate statistical quantity of interest using a minimal number of Monte Carlo samples. The ability to add Monte Carlo samples incrementally is an important tool here: by allowing the user to add samples to an existing Monte Carlo sample set, error tolerances can be satisfied using tight *a posteriori* bounds with lower overall computational work. With Simple Random Sampling adding samples is natural, because each is independent of all preceding samples. However, the standard Latin Hypercube Sampling sample set construction begins with a fixed set size as input.

The non-incremental nature of Latin Hypercube Sampling has been identified as one of the main obstacles in its application to certain classes of complex physical systems [47]. In current implementations of an Incremental LHS (ILHS) method, one is restricted to at least doubling the size of an existing LHS set to retain the convergence properties of Latin Hypercube Sampling [90, 1]. This restriction can result in expensively over-solving a problem if the existing sampling results in answers that are close to meeting the desired error tolerance. An alternative Latin Hypercube Sampling algorithm is developed here, which allows the user to perform LHS studies in a more flexibly incremental setting.

The following sections detail the proposed algorithm, associated theory and numerical results. Section 8.2 describes an algorithm to construct a "Hierarchical Latin Hypercube Sampling" (HLHS) set, which is an LHS set that can be partitioned into HLHS subsets. In Section 8.3 we describe the use of HLHS sets in "Hierarchical Incremental Latin Hypercube Sampling" (HILHS) based Monte Carlo integration. Theoretical results concerning the variance of response function means computed using such algorithms are also stated in this section. Representative numerical experiments are then described in section 8.4 and their results are discussed in detail. Conclusions from the experiments, further work and future applications associated with the HLHS algorithms are discussed in Section 8.5. Section 8.6 contains proofs for the theoretical results stated in section 8.3.

8.2 Hierarchical Latin Hypercube Sample Generation

In this section, we first describe the basic ideas of Latin Hypercube Sampling and give a brief overview of techniques used to generate LHS sets. We then describe a method for generating a special kind of LHS set, a "Hierarchical Latin Hypercube Sample" sequence. In addition to being an LHS set, a HLHS sequence has a property of self-similarity, i.e. each HLHS sequence of even cardinality can be subdivided into two contiguous HLHS subsequences, which themselves may be subdivided and so on. We also discuss a correlation-reduction technique that can construct such a set while reducing correlations between independent parameters and provide better coverage of the sample

space.

8.2.1 Latin Hypercube Sampling

The Latin Hypercube Sampling method seeks to improve coverage of a sample space by stratifying such a space in each parameter into subsets having equal probability. Latin Hypercube Sampling stratifies all input dimensions simultaneously and reduces sampling error by providing a more representative sample ensemble. Latin Hypercube Sampling was introduced by McKay et al. [64], and relevant theory and error estimates were given by Stein and Owen [95, 71]. Like standard Monte Carlo, the output of an LHS method is also a random variable, whose variance typically converges with a rate dictated by a Central Limit Theorem of the form

$$V_{\hat{\mu}_Q} \leq \frac{C}{N_s^p} \tag{8.1}$$

where $V_{\hat{\mu}_Q}$ is the variance of the approximate statistical Quantity of Interest (SQoI), N_s is the number of samples in the set, and p is the associated rate of convergence. For purely additive response functions, i.e. response functions of the form

$$Q(\boldsymbol{\xi}) = \sum_i Q_i(\xi_i) \tag{8.2}$$

we have $p = 2$, while in general $p = 1$. The constant of convergence C is dependent on the SQoI being evaluated, but is typically smaller for Latin Hypercube Sampling than Simple Random Sampling, leading to much faster convergence [95, 71].

Latin Hypercube Sample Set Generation Following Owen [71], consider a set of N_s samples in a space of N_ξ parameters. Then the LHS set D_{ij} is of the form,

$$D_{ij} = F_j^{-1} \frac{(\pi_{ij} - U_{ij})}{N_s} \tag{8.3}$$

where each sequence $\{\pi_{1j}, \pi_{2j}, ...\pi_{N_s j}\}$ denotes a random permutation of $\{1, 2, ..., N_s\}$, each U_{ij} is an independent variable with a uniform distribution $U[0, 1]$, each F_j is the probability distribution function of input parameter j, and $F_j^{-1}(u) = \inf(y|F_j(y) \geq u)$. Equation (8.3) can be used to generate LHS sets and perform computational experiments. Such a capability is present in statistical software like `MATLAB` and `DAKOTA`. Note that the generation of such LHS sets is typically done in serial, as all stratification occurs simultaneously on the same parent set.

8.2.2 Hierarchical Latin Hypercube Sampling

A Hierarchical Latin Hypercube Sample (HLHS) set is a union of self-similar Latin Hypercube sets. Figure 8.1 illustrates the design and construction of a small HLHS set. We observe that the top box contains an LHS set with 8 samples coming from equally stratified subspaces. However, instead of constructing the HLHS sequence by simultaneously stratifying the entire space into 8^2 subspaces as suggested by Eq. (8.3), it is combined from the two smaller 4 sample LHS sets in the second row. These sets are recursively combined from pairs of HLHS sets with 2 samples each. Constructed in such a manner, the top LHS set can be viewed as the concatenation of two subsets

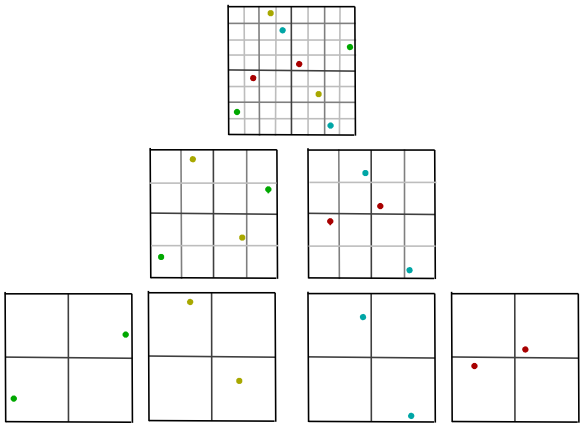


Figure 8.1: An illustration of HLHS set construction. Each box in the figure contains an HLHS set. The top box contains an eight sample HLHS set, the two mid level boxes each contain a four sample HLHS set, and the four bottom row boxes each have a two sample HLHS set.

which themselves are LHS sets, and who each are also concatenations of LHS sets. Such nested LHS sets have wide utility [82, 83].

Definition 2. A binary HLHS sequence is either a singleton sequence $\{\boldsymbol{\xi}_1\}$, or a sequence $\{\boldsymbol{\xi}_i\}_{i=1}^{2N}$ of $\boldsymbol{\xi}_i \in \mathbb{R}^K$, such that the two subsequences $\{\boldsymbol{\xi}_i\}_{i=1}^{N}$ and $\{\boldsymbol{\xi}_i\}_{i=N+1}^{2N}$ are themselves binary HLHS sequences.

We can construct a full HLHS sequence recursively, using Algorithm 3 to generate the permutation matrix π and then using these permutations in Eq. (8.3) to generate the sample values.

The utility of an HLHS sequence in practice will depend on how many levels exist in the hierarchy. N_l levels of binary HLHS subsequences will exist if and only if N_s is divisible by 2^{N_l}. In the extreme case where N_s is prime, Algorithm 3 reduces to generation of a plain LHS set. This algorithm can

Algorithm 3 Generate a simple HLHS permutation π_{ij} with N_s samples in N_ξ parameters

1: **if** $\frac{N_s}{2} \in \mathbb{Z}$ **then**
2: Generate two HLHS permutations, $\pi^{(1)}$ and $\pi^{(2)}$, each of length $N_s/2$ in N_ξ parameters
3: Generate a discrete random matrix C of size $(N_s/2 \times N_\xi)$, such that each independent identically distributed $C_{ij} = 0$ with probability .5 and $C_{ij} = 1$ with probability 0.5. Note that the random matrix C is indexed by the permutation $\pi^{(1)}$.
4: $\pi^{(1)} \leftarrow 2\pi^{(1)} - \mathbf{1} + C$
5: $\pi^{(2)} \leftarrow 2\pi^{(2)} - C$
6: $\pi \leftarrow \left[\pi^{(1)}\pi^{(2)}\right]$ (matrix concatenation)
7: **else**
8: Generate an LHS permutation π of length N_s in N_ξ parameters
9: **end if**
10: **return** π

obviously be generalized to generate hierarchic subsequences for any divisor of N_s, not merely $\frac{N_s}{2_l^N}$; however the remainder of the discussion will be restricted to a binary hierarchy.

Algorithm 3 will recurse up to $\log_2(N_s)$ levels deep; each level involves $\mathcal{O}(N_\xi N_s)$ operations. Like the generation of a plain LHS set, the complexity is $\mathcal{O}(N_\xi N_s \log_2(N_s))$. Because of its recursive nature and its dependence on element-wise matrix operations, Algorithm 3 is trivial to parallelize.

8.2.3 Correlation-Reduced HLHS Set Generation

Even in the construction of standard LHS sets in multi-parameter spaces, improved convergence constants can often be found by using heuristics to reduce any spurious correlations between different sampled parame-

ters [72, 49]. These spurious correlations may also be avoided using a modified algorithm for HLHS set generation. First define a composite inter-parameter covariance function on a permutation matrix π as:

$$R(\pi) \equiv \sum_{\xi_i=1}^{N_\xi} \sum_{\xi_j=\xi_i+1}^{N_\xi} (\mathrm{Cov}(\pi_{\cdot\xi_i}, \pi_{\cdot\xi_j}))^2 \tag{8.4}$$

where Cov denotes the covariance of two vectors. Reducing this function will also reduce the root-mean-squared correlation of the parameter matrix. HLHS set construction with control of inter-parameter correlations is then accomplished using Algorithm 4. The additional correlation-reduction step here is a descent walk in the space of all possible orientations of the permutation matrix. Before merging the two sub-permutations, we attempt "flipping" one of them in each parameter direction, and continue to flip until we find a local minimum of the covariance of the merged permutation. Because this optimization is applied at each merge step, in a many-level hierarchy it can be quite effective.

8.3 Hierarchical Incremental Latin Hypercube Sampling

The HLHS sequences described in the previous section enable a new, finer-grained variety of incremental sampling. Because HLHS sequences are concatenations of HLHS subsequences, one can add samples from a larger HLHS sequence to an initial HLHS subsequence and eventually obtain the larger HLHS sequence. This enables the development of an automatic incre-

Algorithm 4 Generate a reduced-correlation HLHS permutation π_{ij} with N_s samples in N_ξ parameters

if $N_s/2 \in \mathbb{Z}$ **then**
  Generate two HLHS permutations, $\pi^{(1)}$ and $\pi^{(2)}$, each of length $N_s/2$ in N_ξ parameters
  Generate a discrete random matrix C of size $(N_s/2 \times N_\xi)$, such that each independent identically distributed $C_{ij} = 0$ with probability .5 and $C_{ij} = 1$ with probability 0.5. Note that the random matrix C is indexed by the permutation $\pi^{(1)}$.
  $f \leftarrow 1$
  while $f <= N_\xi$ **do**
    $\pi \leftarrow \left[\pi^{(1)}\pi^{(2)}\right]$ (matrix concatenation)
    $\pi^{(2b)} \leftarrow (N_s/2 + 1)\mathbf{1} - \pi^{(2)}$
    $\pi^{(b)} \leftarrow \left[\pi^{(1)}\pi^{(2b)}\right]$
    if $R(\pi^{(b)}) < R(\pi)$ **then**
      $\pi^{(2)} \leftarrow \pi^{(2b)}$
      $f \leftarrow 1$
    else
      $f \leftarrow f + 1$
    end if
  end while
  $\pi^{(1)} \leftarrow 2\pi^{(1)} - \mathbf{1} + C$
  $\pi^{(2)} \leftarrow 2\pi^{(2)} - C$
  $\pi \leftarrow \left[\pi^{(1)}\pi^{(2)}\right]$ (matrix concatenation)
else
  Generate an LHS permutation π of length N_s in N_ξ parameters
end if
return π

mental Monte Carlo method that has LHS-like performance at specific sample sizes on a convergence plot yet also provides performance close to an LHS method when adding moderate increments to those sample sizes.

To illustrate the benefits of such a technique, consider a specific LHS simulation to estimate the mean of a response function S to an error tolerance

of ϵ. If, after N_s samples the error is 1.1 ϵ, then current ILHS techniques would require doubling the number of samples and thus doubling the computational work to obtain lower error. However, with the incremental technique discussed in this section the user can add $\frac{N_s}{4}$ samples to the existing N_s samples and double check that the error tolerance is met. It likely will be, in which case the simulation can be terminated, saving three eighths of the work as compared to the original ILHS method. If it is not, the user can keep adding incremental sample sets, but still be guaranteed LHS like performance in between N_s and 2 N_s samples. We now describe this technique in a generalized manner and develop the associated theory.

8.3.1 HILHS Basics

Figure 8.2 shows a Hierarchical Latin Hypercube Sampling sequence $\mathbf{X}_{N_l,0}$ of size N_s with N_l levels. At any level i, $\mathbf{X}_{N_l,0}$ can be constructed by concatenating HLHS subsequences $\mathbf{X}_{i,j}$, each of size $\frac{N_s}{2^{N_l-i}}$, where j ranges from 0 to $2^{N_l-i}-1$. Denote the Monte Carlo estimated SQoI of Q using $\mathbf{X}_{i,j}$ as $\mu(S(\mathbf{X}_{i,j}))$. The following properties hold:

1. $\forall\, j, \quad \{\mathbf{X}_{i,2j}, \mathbf{X}_{i,2j+1}\} = \mathbf{X}_{i+1,j}$

2. $\forall\, i, j, \quad \mathbf{X}_{i,j}$ is a Latin Hypercube Sample set

3. $\mathrm{Cov}(\mu(S(\mathbf{X}_{i,j_1})), \mu(S(\mathbf{X}_{k,l_1}))) = \mathrm{Cov}(\mu(S(\mathbf{X}_{i,j_2})), \mu(S(\mathbf{X}_{k,l_2})))$
   $\forall\, k, j_n, l_n$ such that $i \leq k < N_l, \quad 2^{i-k} j_n \leq l_n < 2^{i-k}(j_n+1)$

The third property is the result of symmetry, due to construction algorithms in which there is no special ordering of HLHS subsequences within the same HLHS supersequence. For an HLHS sequence constructed this way, the covariance with any subsequence depends only on the levels of each.

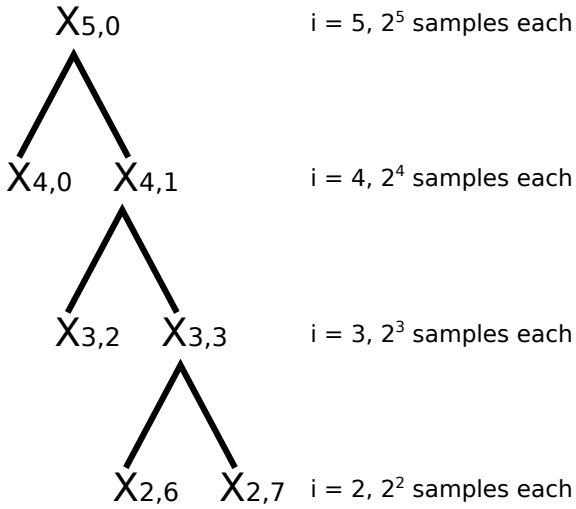


Figure 8.2: A tree diagram of nested HLHS sequences. Each sequence is denoted by $\mathbf{X}_{ij}$, where i is the level in the tree to which the set belongs and $j \in 0...2^{5-i} - 1$ is an index. Shown are parts of the top 4 levels of a 5 level 32 sample design.

A Monte Carlo sample set of any size between $\frac{N_s}{2}$ and N_s can be obtained by combining sequences of appropriate sizes from Figure 8.2. As an illustrative example, consider an HLHS set $\mathbf{X}_{0,0}$ of size $N_s = 2^{10} = 1024$ samples. To identify the largest possible HLHS subsequences from which a 704-element set for Monte Carlo sampling can be constructed, write the decimal 704 in binary,

$$(704)_{10} = (1011000000)_2 \tag{8.5}$$

and note that the 9th, 7th, and 6th bits are 1 - in this case we want to combine latin hypercubes of sizes 2^9 (level 9), 2^7 (level 7) and 2^6 (level 6). One such combination is $\{\mathbf{X}_{9,0}, \mathbf{X}_{7,4}, \mathbf{X}_{6,10}\}$, the first 704 elements of $\mathbf{X}_{10,0}$. In fact, given the element ordering in our HLHS construction algorithms, any maximal combination of subsequences can be obtained by simply taking the desired number of elements from the head of the supersequence. This maximality property remains true even when the number of elements is extended. For example, if the above subset is extended from 704 to 768 elements, merely taking the next 54 elements from $\mathbf{X}_{10,0}$ results in the combination $\{\mathbf{X}_{9,0}, \mathbf{X}_{8,2}\}$. Hierarchical Incremental LHS simulation (HILHS) then becomes straightforward, as outlined in Algorithm 5. Choice of initial sample set size N depends on *a priori*

Algorithm 5 Estimate the SQoI S of a QoI Q to an error tolerance (ϵ) using HILHS simulation

Choose an initial sample set size N
Initial error estimate $e^{(i,N)} \leftarrow \infty$
Generate an HLHS sequence $\mathbf{X^{(i)}}$ with $N_i \geq N$ samples
while $e_{i,N} > \epsilon$ **do**
  Evaluate Q for the first N samples
  Use the first N samples from $\mathbf{X^{(i)}}$ to estimate the SQoI $S^{(i,N)}$ and associated error $e^{(i,N)}$
  if $e^{(i,N)} > \epsilon$ **then**
    Choose sample set growth rate α
    $N \leftarrow \alpha N$ (rounding up)
    if $N > 2\ N_i$ **then**
      $N_{i+1} \leftarrow 2\ N_i$
      Extend $\mathbf{X^{(i)}}$ to an HLHS set $\mathbf{X^{(i+1)}}$ with N_{i+1} samples
      $i \leftarrow i + 1$
    end if
  end if
end while

error estimates. Choice of growth rate α depends on the relative error excess $e^{(i,N)}/\epsilon$ and on the expected convergence rate. Choosing N or α too large risks over-solving the problem; choosing too small limits parallelizability by prescribing steps where too few samples are added simultaneously.

8.3.2 The Asymptotic Distribution for HILHS

We now present theoretical results for HILHS. In particular, we give results for the variance in HILHS simulations. The proofs of these results follow the techniques used by Stein [95] and Qian [82], and are given in the appendix.

Covariance Estimates We first introduce the notion of sister samples.

Definition 3. Consider an HLHS set $\mathbf{X}_{n,0}$ with 2^n total samples. Consider its constituent 2-sample LHS sets at the base level, $\mathbf{X}_{1,j}$, where $0 \leq j < 2^{n-1}$. Two different samples both in the same $\mathbf{X}_{1,j}$ are called level 1 sisters; i.e. the level of sisterhood $s(2j-1, 2j) = 1$. Next consider the 4-sample sets $\mathbf{X}_{2,k}$, where $0 \leq k < 2^{n-2}$, each formed by the union of two base LHS sets. Any pair of samples in $\mathbf{X}_{2,k}$ which are not level 1 sisters are called level 2 sisters; e.g. $s(4k-3, 4k-1) = 2$. This proceeds analogously for higher levels, up to the highest level, wherein every sample in $X_{n,0}$ is at least a level n sister with every other sample.

As a concrete example, in an HLHS with 2^3 total samples, we will have pairs of level three, level two, and level one sisters. The covariances between sisters (and their response function evaluations) of any level are symmetric.

One possible such permutation of HLHS bins is,

$$[[[8\ 3]_1\ [6\ 1]_1]_2\ [[2\ 7]_1\ [5\ 4]_1]_2]_3$$

The numbers in the brackets represent the level of sisterhood of samples contained in those bins. Note that samples in bins 8 and 3 are sisters only of level 1, bins 8 and 6 only of level 2 and bins 8 and 2 only of level 3. In the case above, we will have 3 kinds of covariances.

We can derive the following estimate of the covariance of response function evaluations at two sister samples,

Lemma 8.3.1. *Consider K independent uniformly distributed random variables $\boldsymbol{\Xi} = \{\Xi^k\}_{k=1}^K$. Consider an HLHS sample set with 2^n samples, $\{\{\xi_i^k\}_{k=1}^K\}_{i=1}^{2^n}$. Let $\boldsymbol{\xi}_i$ be the ith complete sample with K components and ξ_i^k be the kth component of that sample. Let S be the response function of interest, $S : \mathbb{R}^K \to \mathbb{R}$. If $E(S^2) < \infty$, then the following estimate for the covariance between response function evaluations for sisters of level $m \in \mathbb{Z}_n$ holds.*

$$Cov(S(\boldsymbol{\xi}_1), S(\boldsymbol{\xi}_{2^{n-l}+1})) = -\sum_{k=1}^{K}\sum_{j=1}^{2^{m-1}}\left(\int_{\frac{2j-2}{2^m}}^{\frac{2j-1}{2^m}} g_k(\xi_1)\, d\xi_1 - \int_{\frac{2j-1}{2^m}}^{\frac{2j}{2^m}} g_k(\xi_1)\, d\xi_1\right)^2 + \mathcal{O}\left(\frac{K(K-1)}{2^{2m-2}}\right) \tag{8.6}$$

where,

$$g_k = \int S(\boldsymbol{\Xi})\, dF_{-k} \tag{8.7}$$

is the 'effect' of the kth random variable.

The proof is included in section 8.6.

The Distribution for HILHS We seek to show that the variance of the SQoI,

$$\mu_{\text{HILHS}} = \frac{\sum_{i=1}^{2^n} S(\boldsymbol{\xi}_i)}{2^n} \tag{8.8}$$

is less than the corresponding variance for μ_{SRS}. First, we have the following theorem, which governs the behavior of the covariance terms arising in the variance expansion for HILHS.

Theorem 8.3.2. *Consider K uniformly distributed random variables $\boldsymbol{\Xi} = \{\Xi^k\}_{k=1}^K$. Consider an HLHS sample set with 2^n samples, $\{\{\xi_i^k\}_{k=1}^K\}_{i=1}^{2^n}$. Let $\boldsymbol{\xi}_i$ be the ith complete sample with K components where ξ_i^k is the kth component of that sample. Let S be the response function of interest, $S : \mathbb{R}^K \to \mathbb{R}$. Assume further that S is bounded from above and/or from below in its domain $[0,1]^K$. $m \in \mathbb{Z}_n$. If $E(S^2) < \infty$, then $\forall\ K \in \mathbb{N}$, the following property for the covariance between response function evaluations for sisters of level holds,*

$$Cov(S(\boldsymbol{\xi}_1), S(\boldsymbol{\xi}_{2^{n-l}+1})) \leq 0 \tag{8.9}$$

The proof is included in section 8.6. With this theorem, we immediately have the following result,

Corollary 8.3.3. *With notations as in Lemma 8.3.1, if $E(S^2) < \infty$, then,*

$$V(\mu_{HILHS}) \leq \frac{V(S(\boldsymbol{\Xi}))}{2^n} \tag{8.10}$$

Note that this result is true regardless of whether one is in the asymptotic region or not. The extension to general distributions is straightforward, one simply replaces $S(\boldsymbol{\xi})$ with $S(F^{-1}(\mathbf{Y}))$ where the $\mathbf{Y}$ are uniformly distributed [95].

8.4 Numerical Results

Numerical experiments were performed to assess the performance of the proposed HILHS strategy and compare it with other modern Latin Hypercube sampling algorithms. Two representative response functions were used for the tests shown here, each with sixteen I.I.D. parameters. An exponential response function represents the continuous response function case, and a rounded sum function tests the discrete response function case. The standard deviations of the input parameters were varied to assess the impact of increasing data spread on the performance of the proposed algorithm.

Because a MC sample solution is a random variable, so is the error in such a solution. For unbiased statistics where a Central Limit Theorem applies [71], the error will asymptotically resemble a normal distribution with mean zero and converging variance. Scalar interpretations of this error, such as confidence limit widths, expected absolute value of the error, etc. will typically be proportional to the standard deviation of the random error; e.g. $\mathcal{O}\left(N^{-1/2}\right)$ in most cases and $\mathcal{O}\left(N^{-1}\right)$ in the additive LHS case. Therefore, from here on, we consider only the expected value of the absolute error. All the figures included in this section show convergence plots for the expected absolute error

in the output means and standard deviations.

The first set of plots show the convergence of the output mean for the following Monte Carlo methods,

1. The HILHS simulation method described in algorithm 5 without any correlation reduction.
2. The standard LHS algorithm included in `MATLAB`
3. The standard SRS algorithm included in `MATLAB`

The second set of plots show the convergence of the output mean and standard deviation for the following Monte Carlo methods,

1. The correlation-reduced HILHS simulation method described in algorithm 5 with the correlation reduction applied as in algorithm 4.
2. The standard LHS algorithm
3. The correlation-reduced LHS algorithm used in the `DAKOTA` [1] statistical analysis package. The correlation reduction there is based on the approach of Iman and Conover [49].

In each HILHS simulation, a special case of algorithm 5 was used, with a fixed number of samples being added on each incremental step for each simulation.

With all six algorithms, to apply the Central Limit Theorem and deduce convergence rates from the numerical studies, several hundred Monte Carlo

trials were conducted for each numerical experiment. The expected absolute value of the estimator error was calculated by averaging over all trials, and the error convergence graphs in this section plot this error versus sample set size on a log-log scale.

8.4.1 Multiparameter Exponential Response Function

For the first test case, we considered the exponential response function

$$Q(\boldsymbol{\xi}) = e^{\sum_{i=1}^{16} \xi_i} \tag{8.11}$$

and normally-distributed input parameters

$$\xi_i \equiv \mathcal{N}\left(\mu_{input}, \sigma_{input}\right) \tag{8.12}$$

$Q(\boldsymbol{\xi})$ is then distributed lognormally; error convergence plots which follow are based on the analytic expressions for its mean and standard deviation.

We first show comparisons of the non-correlation reduced HILHS algorithm with the standard LHS generator in `MATLAB` and SRS. Figure 8.3 shows convergence plots obtained for the calculations of the mean of the response function given by Eq. (8.11) when the input distributions are given by

$$\xi_i \equiv \mathcal{N}\left(1, 1\right) \tag{8.13}$$

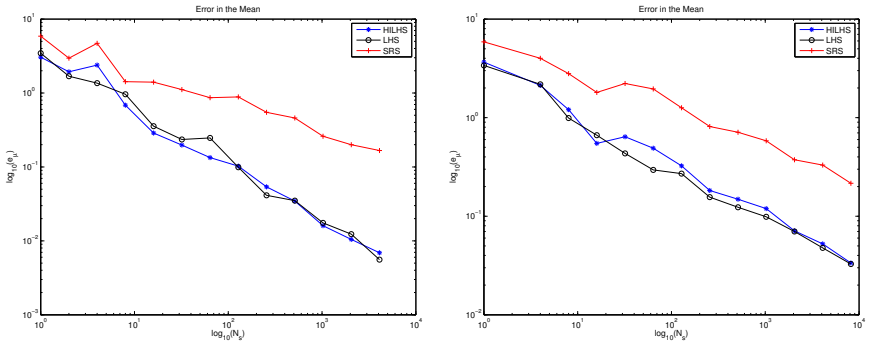
Five hundred Monte Carlo trials were done with each strategy to obtain error plots. We first discuss the plot in Figure 8.3(a), where only a single dimensional response function was used, i.e. $i = 1$. For this case, we expect that the

convergence rate of the two LHS methods will be higher than that for SRS [72]. We observe that this is indeed the case and both HILHS and LHS converge at a higher rate than SRS. We also observe that HILHS and SRS have very similar convergence plots, i.e. for this single dimensional benchmark case the performance of HILHS is similar to that of LHS. We next observe the plot in Figure 8.3(b), which is for a 16-dimensional case, i.e. $i = 16$. We see that the rate of convergence for HILHS, LHS, and SRS are about the same, with both HILHS and LHS converging faster due to better constants. We also observe that HILHS and LHS perform almost identically as they did for the single dimensional case. Figure 8.3(c) then shows the same convergence plots for the 64 dimensional case. Again, we see that HILHS and LHS decrease error at about the same rate and with better constants than SRS. We thus observe that HILHS and LHS offer the same improved error reduction over SRS, with HILHS having the added benefit of the user being able to add points incrementally.

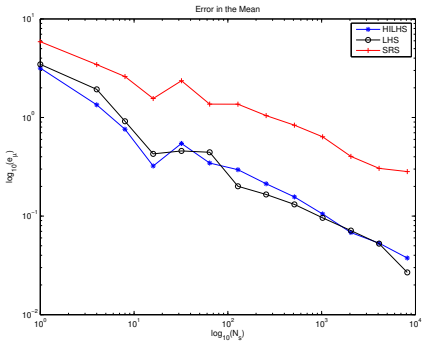
We now move on to comparisons of a correlation reduced version of HILHS with the correlation reduced ILHS method present in `DAKOTA` and regular LHS. Figure 8.4 shows convergence plots obtained for the mean and standard deviation for the response function given by Eq. (8.11) when the input distributions are given by

$$\xi_i \equiv \mathcal{N}(1, 0.5) \tag{8.14}$$

Five hundred Monte Carlo trials were done with each strategy to obtain error plots. We see that all three strategies perform virtually identically in the initial



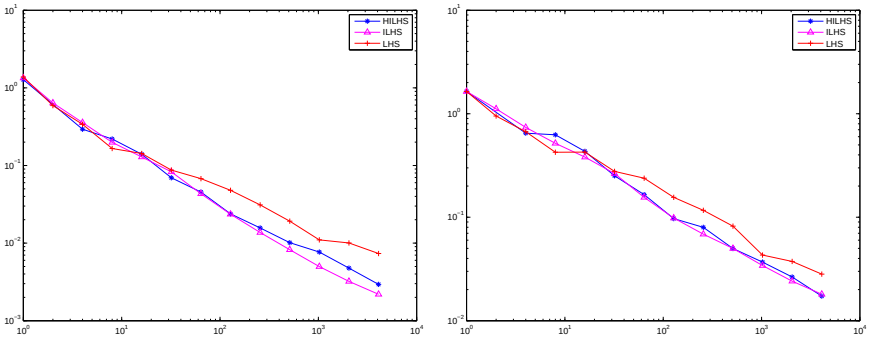
(a) 1 dimensional case (b) 16 dimensional case



(c) 64 dimesional case

Figure 8.3: Comparison of HILHS, standard LHS and SRS methods for computing the mean of the response function given by Eq. (8.11) with 1, 16 and 64 dimensional versions of the distribution given by Eq. (8.13). The input mean was 1 and standard deviation 1.

preasymptotic region of the plots, but the standard LHS and HILHS curves branch off and enter the asymptotic regime with inferior constants than that for DAKOTA's ILHS method. For HILHS, the constant is only slightly inferior to that for the ILHS method. In the standard deviation convergence plots, the HILHS and ILHS methods perform virtually identically, both doing better than the standard LHS method.



(a) Convergence of the mean (b) Convergence of the standard deviation

Figure 8.4: Comparison of HILHS, standard LHS and ILHS methods for computing the mean and standard deviation of the 16 parameter response function in Eq. 8.11. The input mean was 1 and standard deviation 0.5

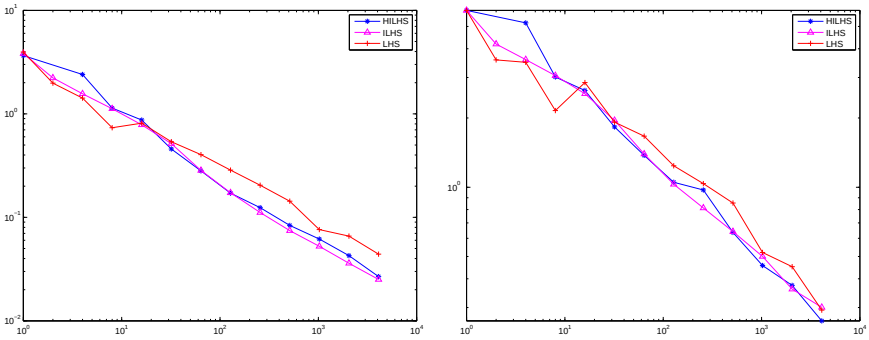
Next, the input standard deviation is increased, correspondingly increasing the difficulty of the problem,

$$\xi_i \equiv \mathcal{N}(1, 1) \tag{8.15}$$

In Figure 8.5, the HILHS and ILHS strategies perform virtually identically for these input parameters, both outperforming standard LHS. For the standard deviation convergence, all three strategies perform nearly identically. Convergence rates and constants are collected in Table 8.1, obtained from log-log linear least-squares fits for all three strategies for both numerical experiments.

The following observations follow from Figures 8.4(a), 8.5(a) and Table 8.1.

1. For the $\sigma_{input} = 0.5$ case, ILHS narrowly outperforms HILHS. Both ILHS



(a) Convergence of the mean

(b) Convergence of the standard deviation

Figure 8.5: Comparison of HILHS, standard LHS and ILHS methods for computing the mean and standard deviation of the 16 parameter response function in Eq. 8.11. The input mean was 1 and standard deviation 1.

Table 8.1: Rates and Constants of convergence for LHS simulations for response function Eq. (8.11) using input parameters given by Eq. (8.14) and Eq. (8.15)

	LHS		HILHS		ILHS	
σ_{input}	Rate	Constant	Rate	Constant	Rate	Constant
0.5	0.5577	0.6674	0.5918	0.3720	0.5937	0.3027
1.0	0.5321	0.5613	0.5395	0.3969	0.5344	0.3278

and HILHS outperform plain LHS sampling by a significant margin. The correlation reduction improves the constant of convergence for HILHS and ILHS results.

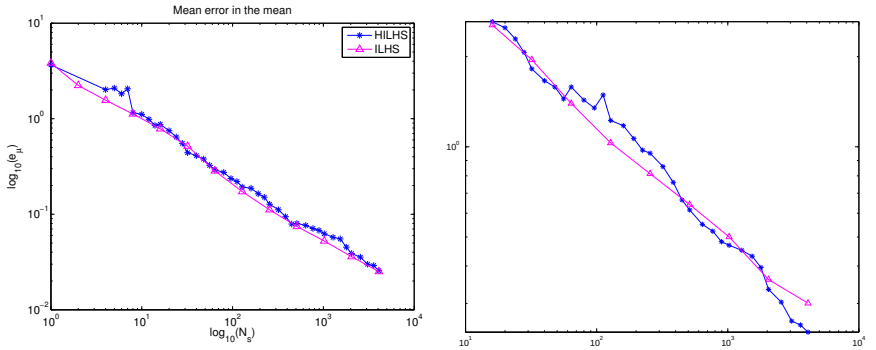
2. For the $\sigma_{input} = 1$ case, all three methods give similar rates of convergence. The ILHS and HILHS methods give a very similar constant, whereas plain LHS results in a substantially larger constant.

These results indicate that the proposed HILHS correlation-reduction strategy and the correlation-reduced ILHS strategy used in DAKOTA performed virtually identically for a truly non-additive nonlinear function. It should be noted that most engineering applications for uncertainty quantification and response functions used in other areas of science tend to be non-additive and highly nonlinear. The results for the exponential response function indicate that the HILHS method can perform at least as well as the existing correlation reduced LHS methods in the literature. However, the advantage of HILHS over previous ILHS strategies is its truly incremental nature, which is emphasized in the next set of experiments.

Figure 8.6 shows convergence plots for the same experimental setup as Figure 8.5, but this time with the incremental HILHS points shown, i.e. points in between the true LHS steps for the HILHS strategy. We observe that the incremental steps consistently decrease the error for both the mean and the standard deviation in the asymptotic range. Also, the incremental steps allow the termination of an incremental Monte Carlo algorithm before the ILHS and standard LHS strategies would allow.

For example, consider the last five points on the blue HILHS convergence curve and last two points on the purple ILHS curve. If the desired error tolerance was in between the errors at the last two points on the ILHS curve, then a simulation using the ILHS strategy could only be terminated at 4096 samples. But, with the HILHS strategy, the error tolerance could be reached with 3574, 3072 or 2560 samples, a 10%-40% savings in computational time

and expense.



(a) Convergence of the mean

(b) Convergence of the standard deviation

Figure 8.6: Comparison of HILHS and ILHS methods for computing the mean and standard deviation of the 16 parameter response function in Eq. 8.11. The input mean was 1 and standard deviation 1.

Numerical experiments were also conducted to compare the correlation reduced HILHS and ILHS methods with Quasi Monte Carlo methods, in particular sample generation using Sobol sequences [92]. In our experiments, the Sobol sequences we generated using `MATLAB`'s `sobolset` function. Figure 8.7 shows the expected error for computing the mean of the response function given by Eq. (8.11), where the input means and standard deviations were both 1. We see that the correlation reduced HILHS and ILHS methods outperform the Sobol method. At lower sample counts, the Sobol method is bettered by standard LHS, however it approaches LHS accuracy as the sample size is increased.

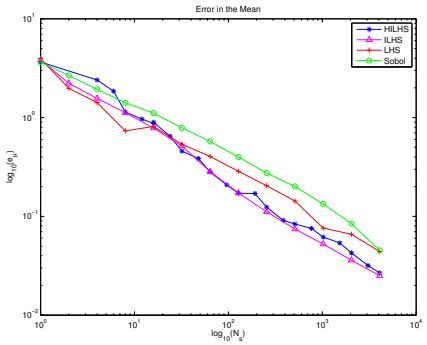


Figure 8.7: Comparison of HILHS, ILHS and Sobol methods for computing the mean of the 16 parameter response function in Eq. 8.11. The input mean was 1 and standard deviation 1.

8.4.2 Multiparameter Rounded Sum Response Function

For the second test case, we used the rounded sum response function

$$Q(\boldsymbol{\xi}) = \text{round}(\sum_{i=1}^{16} \xi_i) \tag{8.16}$$

and normally-distributed input parameters

$$\xi_i \equiv \mathcal{N}\left(\mu_{input}, \sigma_{input}\right) \tag{8.17}$$

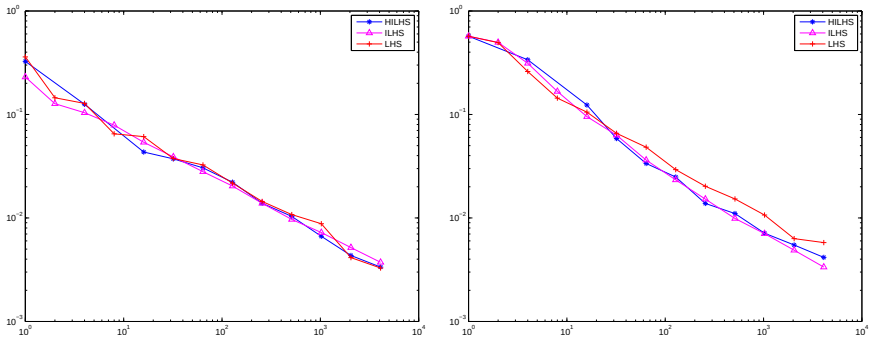
The normcdf function in `MATLAB` was used to obtain the means and standard deviations of $Q(\boldsymbol{\xi})$ to high accuracy through direct numerical integration of an equivalent one-parameter benchmark problem. These values were then used as "truth values" to construct the mean and standard deviation convergence plots.

Just as with the exponential response function, we use this benchmark to compare the new HILHS sample generation strategy with the standard

LHS method and the correlation-reduced ILHS method. Figure 8.8 shows convergence plots obtained by using the three strategies for computing the mean and standard deviation for the response function in Eq. 8.16 when the input distributions are given by,

$$\xi_i \equiv \mathcal{N}(1, 0.5) \tag{8.18}$$

Five hundred Monte Carlo trials were done with each strategy to obtain the plots. All three strategies result in nearly identical convergence plots reflecting the nonlinear and nonadditive nature of the response function. The standard LHS method converges with a slightly inferior constant as compared to the HILHS and ILHS methods.



(a) Convergence of the mean

(b) Convergence of the standard deviation

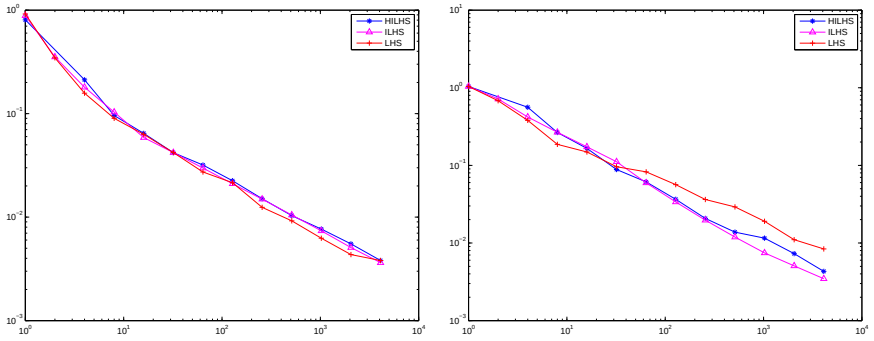
Figure 8.8: Comparison of HILHS, standard LHS and ILHS methods for computing the mean and standard deviation of the 16 parameter response function in Eq. (8.16). The input mean was 1 and standard deviation 0.5.

Next, we repeat the above procedure but use a different input standard

deviation, as we did with the exponential response function.

$$\xi_i \equiv \mathcal{N}(1,1) \tag{8.19}$$

The results are shown in Figure 8.9. Again, we observe virtually identical plots for the three strategies. The standard deviation convergence plots show that the HILHS and ILHS methods perform about the same while the LHS strategy converges slower. Recovered convergence rates and constants for the



(a) Convergence of the mean (b) Convergence of the standard deviation

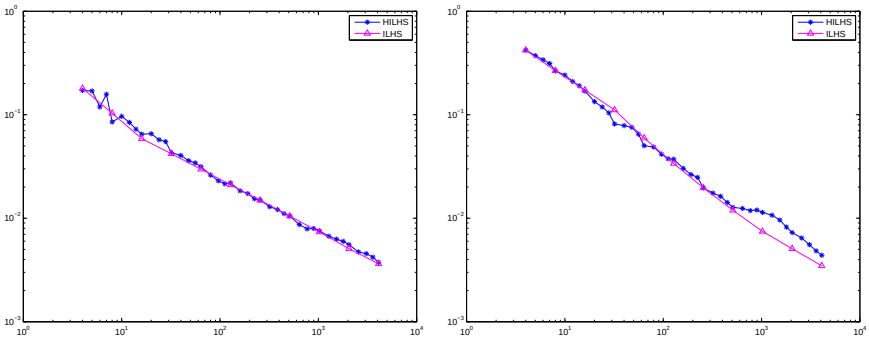
Figure 8.9: Comparison of HILHS, standard LHS and ILHS methods for computing the mean and standard deviation of the 16 parameter response function in Eq. (8.16). The input mean was 1 and standard deviation 1.

discrete response function are summarized in Table 8.2. These results indicate that the performance of the correlation reduced ILHS strategy in `DAKOTA`, the HILHS strategy and the plain LHS method is roughly the same for the rounded sum function. The correlation reduction appears to have no substantial effect on the convergence results for this response function. However, the HILHS strategy is again shown to be a competitive one for this response function.

Table 8.2: Rates and Constants of convergence for LHS simulations for response function Eq. (8.16) using input parameters given by Eq. (8.18) and Eq. (8.19)

	LHS		HILHS		ILHS	
σ_{input}	Rate	Constant	Rate	Constant	Rate	Constant
0.5	0.5216	0.2637	0.5471	0.3009	0.4863	0.2096
1.0	0.5250	0.2542	0.5005	0.2458	0.5042	0.2414

Moving on, Figure 8.10 shows convergence plots for the same experimental setup as Figure 8.9, but this time with the incremental points, i.e. points between true LHS steps shown for the HILHS strategy. Observe that the incremental steps consistently decrease the error for both the mean and the standard deviation in the asymptotic range. Again, the incremental steps allow the termination of an incremental Monte Carlo algorithm before the ILHS and standard LHS strategies would allow. For this response function, the HILHS strategy essentially interpolates the ILHS points. The behavior of the standard deviation convergence plot Figure 8.10(b) merits some discussion. We see a change in the nature of the HILHS plot around the 512 sample mark, with the error reduction per sample addition decreasing substantially and then returning close to the pre 512 sample mark further along. It is unclear what caused this behavior at this point. But we note that throughout this part of the curve, the error is consistently reduced by the HILHS strategy and offers similar benefits in computational costs as those seen for the mean, but with a smaller magnitude.



(a) Convergence of the mean

(b) Convergence of the standard deviation

Figure 8.10: Comparison of HILHS and ILHS methods for computing the mean and standard deviation of the 16 parameter response function in Eq. (8.11). The input mean and standard deviation were both 1.

8.5 Conclusions

We have presented a new Hierarchical Incremental Latin Hypercube Sampling Monte Carlo method that allows the addition of an arbitrary number of sample points to an existing LHS sample set during a simulation. By generating the Latin Hypercube Samples in a hierarchical, self-similar manner, the addition of sample sets to an existing Latin Hypercube can be performed with reliable expected error reduction. We have stated and proven relevant theorems to show that the HILHS technique guarantees consistent error reduction and has a lower variance than random sampling. Relevant numerical experiments and their results have been shown. The tests confirm that HILHS provides performance on par with existing LHS schemes while maintaining consistent error reduction on each incremental step. Thus, it appears that the HILHS method is well-suited for application to various estimation prob-

lems. Our continuing work is turning now to the application of HILHS in new Monte Carlo integration techniques such as Local Sensitivity Derivative Enhanced Monte Carlo (LSDEMC) [98] and Finite Element/Monte Carlo error redistribution.

Stochastic analysis toolkits like `DAKOTA` can benefit from the addition of such incremental Latin Hypercube Sampling methods, since they serve the high performance, large scale computing community where this method can result in substantial functionality and performance gains. The application of such methods to large scale computational problems that result from Uncertainty Quantification studies from hypersonic flows, coupled electroosmotic flows and other engineering problems will be another focus of our continuing work.

8.6 Proofs

Consider K independently distributed random variables $\boldsymbol{\Xi} = \{\Xi^k\}_{k=1}^K$. Assume that their distributions are uniform and scaled to be between $(0, 1)$. Relaxing these assumptions to obtain results for the general case is straightforward [95]. Consider an HLHS sample set with 2^n samples, $\{\{\xi_i^k\}_{k=1}^K\}_{i=1}^{2^n}$. From here on, we understand the notation $\boldsymbol{\xi}_i$ to mean the ith complete sample with K components and ξ_i^k to mean the kth component of that sample. We seek to show that the variance of the SQoI,

$$\mu_{\text{HILHS}} = \frac{\sum_{i=1}^{2^n} S(\boldsymbol{\xi}_i)}{2^n} \tag{8.20}$$

is less than the corresponding variance for μ_{SRS}. Following Stein [95] and Qian [82], the proof consists of three major steps:

1. Decompose the overall variance into variance and covariance terms
2. Compute the covariance terms:
   (a) Obtain the discrete pdfs for choosing pairs of HILHS bins at each level
   (b) Use these to obtain continuous joint density functions of sister samples at each level
   (c) Estimate the covariance for pairs of sisters of at each level
3. Using results from 2c in 1, estimate the overall variance

8.6.1 Variance Decomposition

Consider the variance of Eq. (8.20),

$$V(\mu_{\text{HILHS}}) = V\left(\frac{\sum_{i=1}^{2^n} S(\boldsymbol{\xi}_i)}{2^n}\right)$$
$$= \frac{V(S(\boldsymbol{\Xi}))}{2^n} + \frac{1}{2^{2n}} \sum_{i=1}^{2^n} \sum_{j=1, j\neq i}^{2^n} \text{Cov}(S(\boldsymbol{\xi}_i), S(\boldsymbol{\xi}_j)) \tag{8.21}$$

There are a total of $2^n \cdot (2^n - 1)$ different covariance terms. Each $\boldsymbol{\xi}_i$ sample generates $2^n - 1$ covariance terms total, one for each other $\boldsymbol{\xi}_j$. Because of the symmetry of HLHS construction, the terms for each $\boldsymbol{\xi}_i$ can be rearranged by level, into groups of 2^{m-1} equivalent terms for the members $\boldsymbol{\xi}_j$ of each set of

its level m sisters.

$$\sum_{j=1,j\neq i}^{2^n} \mathrm{Cov}(S(\boldsymbol{\xi}_i), S(\boldsymbol{\xi}_j)) \tag{8.22}$$

$$= \sum_{m=1}^{n} 2^{m-1} \mathrm{Cov}(S(\boldsymbol{\xi}_1), S(\boldsymbol{\xi}_{2^m})) \tag{8.23}$$

8.6.2 Proof of Lemma 8.3.1: Covariance Estimates for each level

By definition the covariance can be computed as,

$$\mathrm{Cov}(X, Y) \equiv E[XY] - E[X]E[Y] \tag{8.24}$$

where X and Y are random variables. The covariance for a pair of sisters of level m is given by,

$$\mathrm{Cov}(S(\boldsymbol{\xi}_1), S(\boldsymbol{\xi}_{2^{n-l}+1})) = E[S(\boldsymbol{\xi}_1)S(\boldsymbol{\xi}_{2^{n-l}+1})] - E[S(\boldsymbol{\xi}_1)]E[S(\boldsymbol{\xi}_{2^{n-l}+1})] \tag{8.25}$$

To compute this covariance, we need the joint density function of $(\boldsymbol{\xi}_1, \boldsymbol{\xi}_{2^{n-l}+1})$. We consider the single variable case and obtain the joint density for $(\xi_1, \xi_{2^{n-l}+1})$.

$$pr(\xi_1 = z_1, \xi_{2^{n-l}+1} = z_2) = pr(\xi_1, \xi_{2^{n-l}+1}) \qquad 0 \leq z_1, z_2 \leq 1 \tag{8.26}$$

The generalization to arbitrary dimensions is straightforward.

We begin with definitions of convenient notation, for the "parents" of a sample index or a bin index:

Definition 4. Given $i \in \mathbb{Z}_{2^n}$, a sample index in an HLHS set $\mathbf{X}_{n,k}$ of size 2^n, the parent of i is $P(i) \in \mathbb{Z}_{2^{n-1}}$, a corresponding sample index with respect to one of the HLHS subsets $\mathbf{X}_{n-1,2k}, \mathbf{X}_{n-1,2k+1}$, which is given by

$$P(i) = ((i-1) \mod n) + 1 \tag{8.27}$$

Parents of parents of sample indices, e.g. $P^2(i) \equiv ((P(i)-1) \mod (n/2))+1$, are defined naturally up to $P^n(i) = 1$.

Definition 5. Given a bin index $a \in \mathbb{Z}_{2^n}$, the parent of a, $p(a) \in \mathbb{Z}_{2^{n-1}}$, is defined as

$$p(a) = \left\lceil \frac{a}{2} \right\rceil = \begin{cases} \frac{a}{2} & \text{if } a \text{ is even;} \\ \frac{a+1}{2} & \text{if } a \text{ is odd.} \end{cases}$$

We can further consider a parent $p(p(a))$ of $p(a)$, and so on for any level $l \leq n-1$,

$$p^l(a) = \left\lceil \frac{a}{2^l} \right\rceil \tag{8.28}$$

A lower level of sisterhood corresponds to stricter conditions on common parentage. The requirement that HLHS subsets also be valid LHS sets implies that samples from the same HLHS subset cannot fall within the same bin at the level of that subset.

Lemma 8.6.1. *Consider two samples, from an HLHS permutation of size 2^n, with distinct indices i and j, whose values are located in HLHS bins $\pi(i) = a$ and $\pi(j) = b$. If the samples are sisters of level $s(i,j) = m$, then $p^l(a) \neq p^l(b)\ \forall\ l \leq n-m$.*

Discrete probability distribution functions for bin pairs

Theorem 8.6.2. *Let π_n denote an HLHS permutation of $\mathbb{Z}_{2^n}$. Let i and j be*

distinct sisters of level $m \leq n$*;* $s(i,j) = m$*. Given bins* $a, b \in \mathbb{Z}_{2^n}$*, we have*

$$pr(\pi_n(i) = a, \pi_n(j) = b) = \begin{cases} 0 & \text{if } p^{n-m}(a) = p^{n-m}(b) \\ 2^{-2n+1} & \text{if } p^{n-m}(a) \neq p^{n-m}(b), \\ & p^{n-m+1}(a) = p^{n-m+1}(b) \\ 2^{-2n} & \text{if } p^{n-m+1}(a) \neq p^{n-m+1}(b) \end{cases} \tag{8.29}$$

Proof. In the $m = n$ case, i and j come from different HILHS subsets of size 2^{n-1}. In the non-correlation-reduced algorithm, their parent bin positions $\pi_{n-1}(P(i)) = p(a)$ and $\pi_{n-1}(P(j)) = p(b)$ in these subsets are independent. The child bin probabilities $pr(\pi_n(i) = a)$ and $pr(\pi_n(j) = b)$ are determined by parent bin positions and by either one or two entries (depending on whether $p(a) == p(b)$) in the associated "coin flip" matrix.

$$pr(\pi_n(i) = a, \pi_n(j) = b) = \tag{8.30}$$

$$pr(\pi_{n-1}(P(i)) = p(a)) \cdot pr(\pi_{n-1}(P(j)) = p(b)) \cdot \begin{cases} 1/4 & \text{if } p(a) \neq p(b), \\ 1/2 & \text{if } p(a) = p(b), \\ 0 & \text{if } a = b \end{cases} \quad a \neq b, \tag{8.31}$$

Each probability of the form $pr(\pi_{n-1}(P(i)) = p(a))$ is simply 2^{1-n}, because each of the 2^{n-1} possibilities for each parent bin is equally likely.

For the $m < n$ cases, i and j belong to cardinality 2^m HLHS subsets of the 2^n HLHS parent set, and coin flips from the subsequent $n - m$ recursive HLHS construction steps are independent.

$$\begin{aligned} &pr(\pi_n(i) = a, \pi_n(j) = b : s(i,j) = m) \\ &= pr(\pi_{n-m}(P^m(i)) = p^m(a), \pi_{n-m}(P^m(j)) \end{aligned}$$

$$= p^m(b) : s(P^m(i), P^m(j)) = 0) \cdot \frac{1}{2^{2(n-m)}} \tag{8.32}$$

Combining these results gives the final formula. □

Continuous Joint Density functions For convenience of notation, indicator function $\delta_n(x_1, x_2)$ will express whether sample values x_1 and x_2 fall in the same LHS bin of size 2^{-n}, and $\gamma_l(a, b)$ will express whether bin indices a and b share the same level-l parent $P^l(a) \stackrel{?}{=} P^l(b)$:

$$\delta_n(x_1, x_2) = \begin{cases} 1 & \lceil 2^n x_1 \rceil = \lceil 2^n x_2 \rceil \\ 0 & \text{otherwise} \end{cases} \tag{8.33}$$

$$\gamma_l(a, b) = \begin{cases} 1 & \left\lceil \frac{a}{2^l} \right\rceil = \left\lceil \frac{b}{2^l} \right\rceil \\ 0 & \text{otherwise} \end{cases} \tag{8.34}$$

Now we can write the continuous pdfs for level m sisters as

$$\begin{aligned} pr(\xi_1,&\xi_{2^{n-l}+1}) = pr(\pi(1) = \lceil 2^n \xi_1 \rceil, \pi(2^m) = \lceil 2^n \xi_{2^{n-l}+1} \rceil) \cdot (2^n)^2 \\ =&\left(\frac{1 - \gamma_{n-m+1}(\lceil 2^n \xi_1 \rceil, \lceil 2^n \xi_{2^{n-l}+1} \rceil)}{2^{2n}} + \frac{\gamma_{n-m+1}(\lceil 2^n \xi_1 \rceil, \lceil 2^n \xi_{2^{n-l}+1} \rceil)}{2^{2n-1}} \right) \\ &\cdot (1 - \gamma_{n-m}(\lceil 2^n \xi_1 \rceil, \lceil 2^n \xi_{2^{n-l}+1} \rceil)) \cdot (2^n)^2 \\ =&(1 - \gamma_{n-m+1}(\lceil 2^n \xi_1 \rceil, \lceil 2^n \xi_{2^{n-l}+1} \rceil) + 2\gamma_{n-m+1}(\lceil 2^n \xi_1 \rceil, \lceil 2^n \xi_{2^{n-l}+1} \rceil)) \\ &\cdot (1 - \gamma_{n-m}(\lceil 2^n \xi_1 \rceil, \lceil 2^n \xi_{2^{n-l}+1} \rceil)) \\ =&(1 + \gamma_{n-m+1}(\lceil 2^n \xi_1 \rceil, \lceil 2^n \xi_{2^{n-l}+1} \rceil))(1 - \gamma_{n-m}(\lceil 2^n \xi_1 \rceil, \lceil 2^n \xi_{2^{n-l}+1} \rceil)) \end{aligned} \tag{8.35}$$

We can further compute,

$$\begin{aligned} &pr(\xi_1, \xi_{2^{n-l}+1}) \\ &= (1 + \gamma_{n-m+1}(\lceil 2^n \xi_1 \rceil, \lceil 2^n \xi_{2^{n-l}+1} \rceil))(1 - \gamma_{n-m}(\lceil 2^n \xi_1 \rceil, \lceil 2^n \xi_{2^{n-l}+1} \rceil)) \end{aligned}$$

$$= 1 - \gamma_{n-m}(\lceil 2^n \xi_1 \rceil, \lceil 2^n \xi_{2^{n-l}+1} \rceil) + \gamma_{n-m+1}(\lceil 2^n \xi_1 \rceil, \lceil 2^n \xi_{2^{n-l}+1} \rceil) \ldots$$

$$\ldots - \gamma_{n-m+1}(\lceil 2^n \xi_1 \rceil, \lceil 2^n \xi_{2^{n-l}+1} \rceil) \gamma_{n-m}(\lceil 2^n \xi_1 \rceil, \lceil 2^n \xi_{2^{n-l}+1} \rceil)$$

$$= 1 - \delta_m(\xi_1, \xi_{2^{n-l}+1}) + \delta_{m-1}(\xi_1, \xi_{2^{n-l}+1}) - \delta_{m-1}(\xi_1, \xi_{2^{n-l}+1}) \delta_m(\xi_1, \xi_{2^{n-l}+1})$$

$$= 1 - \delta_m(\xi_1, \xi_{2^{n-l}+1}) + \delta_{m-1}(\xi_1, \xi_{2^{n-l}+1}) - \delta_m(\xi_1, \xi_{2^{n-l}+1})$$

$$= 1 + \delta_{m-1}(\xi_1, \xi_{2^{n-l}+1}) - 2\delta_m(\xi_1, \xi_{2^{n-l}+1}) \tag{8.36}$$

For the K variable case with no HLHS correlation reduction applied, we have,

$$\begin{aligned} pr(\boldsymbol{\xi}_1, \boldsymbol{\xi}_{2^{n-l}+1}) &= \prod_{k=1}^{K} (1 + \delta_{m-1}(\xi_1^k, \xi_{2^m}^k) - 2\,\delta_m(\xi_1^k, \xi_{2^m}^k)) \\ &= \prod_{k=1}^{K} (1 + \delta_{m-1}^k - 2\,\delta_m^k) \end{aligned} \tag{8.37}$$

In the $m = 1$ case, we have just one HLHS set with two samples, hence we simply obtain a regular LHS sample set. Also, in this case $n = l = 1$. Therefore, Eq. (8.37) becomes,

$$pr(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2) = \prod_{k=1}^{K} (1 + \delta_0^k - 2\,\delta_1^k) = \prod_{k=1}^{K} (2 - 2\,\delta_1^k) = 2^K \prod_{k=1}^{K} (1 - \delta_1^k)$$

This matches exactly the expression given by Stein for regular LHS design with two samples [95].

Covariance Estimate for a single level

$$\mathrm{Cov}(S(\boldsymbol{\xi}_1), S(\boldsymbol{\xi}_{2^{n-l}+1})) = \int S(\boldsymbol{\xi}_1)\, S(\boldsymbol{\xi}_{2^{n-l}+1}) \prod_{k=1}^{K} (1 + \delta_{m-1}^k - 2\,\delta_m^k)$$

$$= \int S(\boldsymbol{\xi}_1)\, S(\boldsymbol{\xi}_{2^{n-l}+1}) \left(1 + \sum_{k=1}^{K} (\delta_{m-1}^k - 2\,\delta_m^k)\right) + \mathcal{O}\left(\frac{K(K-1)}{2^{2m-2}}\right)$$

$$= (E(S))^2 + \sum_{k=1}^{K} \int S(\boldsymbol{\xi}_1)\ S(\boldsymbol{\xi}_{2^{n-l}+1})\ \delta_{m-1}^k - 2\sum_{k=1}^{K} \int S(\boldsymbol{\xi}_1)\ S(\boldsymbol{\xi}_{2^{n-l}+1})\ \delta_m^k$$
$$+ \mathcal{O}\left(\frac{K(K-1)}{2^{2m-2}}\right) \tag{8.38}$$

Now following [95] and using the notation in Eq. (8.6), we have

$$\begin{aligned}\int S(\boldsymbol{\xi}_1)\ S(\boldsymbol{\xi}_{2^{n-l}+1})\ \delta_m^k\ d\boldsymbol{\xi}_1\ d\boldsymbol{\xi}_{2^{n-l}+1} \\ = \int g_k(\xi_1)\ g_k(\xi_2)\ \delta_m^k\ d\xi_1\ d\xi_2 \\ = \sum_{j=1}^{2^m} \left(\int_{\frac{j-1}{2^m}}^{\frac{j}{2^m}} g_k(\xi_1)\ d\xi_1 \right)^2 \end{aligned} \tag{8.39}$$

Substitution and simplification then leads to an estimate for the covariance,

$$\begin{aligned}&\mathrm{Cov}(S(\boldsymbol{\xi}_1), S(\boldsymbol{\xi}_{2^{n-l}+1})) = (E(S))^2 + \sum_{k=1}^{K} \int S(\boldsymbol{\xi}_1)\ S(\boldsymbol{\xi}_{2^{n-l}+1})\ \delta_{m-1}^k \\ &- 2\sum_{k=1}^{K} \int S(\boldsymbol{\xi}_1)\ S(\boldsymbol{\xi}_{2^{n-l}+1})\ \delta_m^k + \mathcal{O}\left(\frac{K(K-1)}{2^{2m-2}}\right) - (E(S))^2 \\ &= \sum_{k=1}^{K} \int S(\boldsymbol{\xi}_1)\ S(\boldsymbol{\xi}_{2^{n-l}+1})\ \delta_{m-1}^k - 2\sum_{k=1}^{K} \int S(\boldsymbol{\xi}_1)\ S(\boldsymbol{\xi}_{2^{n-l}+1})\ \delta_m^k \\ &+ \mathcal{O}\left(\frac{K(K-1)}{2^{2m-2}}\right) \\ &= \sum_{k=1}^{K} \sum_{j=1}^{2^{m-1}} \left(\int_{\frac{j-1}{2^{m-1}}}^{\frac{j}{2^{m-1}}} g_k(\xi_1)\ d\xi_1 \right)^2 - 2\sum_{k=1}^{K} \sum_{j=1}^{2^m} \left(\int_{\frac{j-1}{2^m}}^{\frac{j}{2^m}} g_k(\xi_1) \right)^2 \ldots \\ &\ldots + \mathcal{O}\left(\frac{K(K-1)}{2^{2m-2}}\right) \\ &= \sum_{k=1}^{K} \sum_{j=1}^{2^{m-1}} \left(\int_{\frac{j-1}{2^{m-1}}}^{\frac{j}{2^{m-1}}} g_k(\xi_1)\ d\xi_1 \right)^2 - 2\sum_{j=1}^{2^m} \left(\int_{\frac{j-1}{2^m}}^{\frac{j}{2^m}} g_k(\xi_1)\ d\xi_1 \right)^2 \\ &+ \mathcal{O}\left(\frac{K(K-1)}{2^{2m-2}}\right)\end{aligned} \tag{8.40}$$

$$
\begin{aligned}
&= \sum_{k=1}^{K} \sum_{j=1}^{2^{m-1}} \left(\int_{\frac{2j-2}{2^m}}^{\frac{2j-1}{2^m}} g_k(\xi_1)\, d\xi_1 + \int_{\frac{2j-1}{2^m}}^{\frac{2j}{2^m}} g_k(\xi_1)\, d\xi_1 \right)^2 \\
&- 2\left[\left(\int_{\frac{2j-2}{2^m}}^{\frac{2j-1}{2^m}} g_k(\xi_1)\, d\xi_1 \right)^2 + \left(\int_{\frac{2j-1}{2^m}}^{\frac{2j}{2^m}} g_k(\xi_1)\, d\xi_1)^2 \right)^2 \right] + \mathcal{O}\left(\frac{K(K-1)}{2^{2m-2}} \right) \\
&= -\sum_{k=1}^{K} \sum_{j=1}^{2^{m-1}} \left(\int_{\frac{2j-2}{2^m}}^{\frac{2j-1}{2^m}} g_k(\xi_1)\, d\xi_1 - \int_{\frac{2j-1}{2^m}}^{\frac{2j}{2^m}} g_k(\xi_1)\, d\xi_1 \right)^2 + \mathcal{O}\left(\frac{K(K-1)}{2^{2m-2}} \right)
\end{aligned}
\tag{8.41}
$$

8.6.3 Proof of Theorem 8.3.2: Behavior of Covariance Terms

We now proceed to prove Theorem 8.3.2 using mathematical induction. We start with the single variable case, i.e. $K = 1$. In this case the pdf for for sisters of level m is given by,

$$
pr(\xi_1, \xi_{2^{n-l}+1}) = 1 + \delta^1_{m-1}(\xi_1, \xi_{2^{n-l}+1}) - 2\delta^1_m(\xi_1, \xi_{2^{n-l}+1}) \tag{8.42}
$$

The covariance is then given by,

$$
\begin{aligned}
&\mathrm{Cov}(S(\xi_1), S(\xi_{2^{n-l}+1})) \\
&= \int S(\xi_1)\, S(\xi_{2^{n-l}+1})\, (1 + \delta^1_{m-1}(\xi_1, \xi_{2^{n-l}+1}) - 2\delta^1_m(\xi_1, \xi_{2^{n-l}+1})) - (E(S))^2 \\
&= (E(S))^2 + \int S(\xi_1)\, S(\xi_{2^{n-l}+1}) \delta^1_{m-1} - 2 \int S(\xi_1)\, S(\xi_{2^{n-l}+1}) \delta^1_m - (E(S))^2 \\
&= -\sum_{j=1}^{2^{m-1}} \left(\int_{\frac{2j-2}{2^m}}^{\frac{2j-1}{2^m}} S(\xi)\, d\xi - \int_{\frac{2j-1}{2^m}}^{\frac{2j}{2^m}} S(\xi)\, d\xi \right)^2 \leq 0 \quad \text{(Using Eq. (8.40))}
\end{aligned}
\tag{8.43}
$$

Thus the result holds for $K = 1$.

Begin with the inductive assumption that the result is true for $K = L$.

The pdf for sisters of level m is then given by,

$$pr(\boldsymbol{\xi}_1, \boldsymbol{\xi}_{2^{n-l}+1}) = \prod_{k=1}^{L} (1 + \delta_{m-1}^k(\xi_1^k, \xi_{2^m}^k) - 2\delta_m^k(\xi_1^k, \xi_{2^m}^k)) \tag{8.44}$$

This pdf can also be written as,

$$pr(\boldsymbol{\xi}_1, \boldsymbol{\xi}_{2^{n-l}+1}) = 1 + R_L(\boldsymbol{\xi}_1, \boldsymbol{\xi}_{2^{n-l}+1}) \tag{8.45}$$

where $R_L(\boldsymbol{\xi}_1, \boldsymbol{\xi}_{2^{n-l}+1})$ are the remaining terms obtained on expanding the product given by Eq. 8.44. With this notation the covariance for the $K = L$ case can be given as,

$$\begin{aligned}
&\mathrm{Cov}(S(\boldsymbol{\xi}_1), S(\boldsymbol{\xi}_{2^{n-l}+1})) \\
&= \int S(\boldsymbol{\xi}_1)\, S(\boldsymbol{\xi}_{2^{n-l}+1})\, (1 + R_L(\boldsymbol{\xi}_1, \boldsymbol{\xi}_{2^{n-l}+1})) - (E(S))^2 \\
&= (E(S))^2 + \int S(\boldsymbol{\xi}_1)\, S(\boldsymbol{\xi}_{2^{n-l}+1}) R_L(\boldsymbol{\xi}_1, \boldsymbol{\xi}_{2^{n-l}+1}) - (E(S))^2 \\
&= \int S(\boldsymbol{\xi}_1)\, S(\boldsymbol{\xi}_{2^{n-l}+1}) R_L(\boldsymbol{\xi}_1, \boldsymbol{\xi}_{2^{n-l}+1}) \leq 0 \quad \text{by the inductive hypothesis}
\end{aligned} \tag{8.46}$$

Then completing the induction requires proving the result for the case $K = L + 1$. The pdf for sisters of level m is,

$$\begin{aligned}
pr(\boldsymbol{\xi}_1, \boldsymbol{\xi}_{2^{n-l}+1}) &= \prod_{k=1}^{L+1} (1 + \delta_{m-1}^k(\xi_1^k, \xi_{2^m}^k) - 2\delta_m^k(\xi_1^k, \xi_{2^m}^k)) \\
&= (1 + \delta_{m-1}^{L+1}(\xi_1^{L+1}, \xi_{2^m}^{L+1}) - 2\delta_m^{L+1}(\xi_1^{L+1}, \xi_{2^m}^{L+1})) \prod_{k=1}^{L} (1 + \delta_{m-1}^k - 2\delta_m^k) \\
&= (1 + R_L)(1 + \delta_{m-1}^{L+1} - 2\delta_m^{L+1})
\end{aligned} \tag{8.47}$$

And so the covariance for the $K = L + 1$ case can be expressed as,

$$\mathrm{Cov}(S(\boldsymbol{\xi}_1), S(\boldsymbol{\xi}_{2^{n-l}+1}))$$

$$= \int S(\boldsymbol{\xi}_1)\, S(\boldsymbol{\xi}_{2^{n-l}+1})\, (1+R_L)(1+\delta_{m-1}^{L+1}-2\delta_m^{L+1}) - (E(S))^2$$

$$= \underbrace{\int S(\boldsymbol{\xi}_1)\, S(\boldsymbol{\xi}_{2^{n-l}+1})\, (1+R_L) - (E(S))^2}_{I_1} +$$

$$\underbrace{\int S(\boldsymbol{\xi}_1)\, S(\boldsymbol{\xi}_{2^{n-l}+1})\, (1+R_L)\, (\delta_{m-1}^{L+1}-2\delta_m^{L+1})}_{I_2} \tag{8.48}$$

Integrating out the effect of the $L+1$st random variable in the term I_1 and leaving it with $g_{-(L+1)}(\{\xi_1^k\}_{k=1}^L)$ and $g_{-(L+1)}(\{\xi_{2^m}^k\}_{k=1}^L)$ casts it in the form of the inductive hypothesis. Therefore $I_1 < 0$.

These covariances are unaffected by adding or subtracting a constant function to S. Thus if S is not uniformly positive or uniformly negative, we can consider without loss of generality an equivalent one-signed response function created by either subtracting the supremum from an S which is bounded from above or adding the infimum to an S which is bounded from below. In these cases $S(\boldsymbol{\xi}_1)S(\boldsymbol{\xi}_{2^{n-l}+1})$ will be uniformly non-negative.

Considering I_2, note that the function $1+R_L$ is also non-negative. Also, $1+R_L \leq 2^L$. Using the one-signedness of S and the boundedness of $1+R_L$, the complete integral I_2 can be written as,

$$\int_0^1 \int_0^1 \left[\prod_{k=1}^{L} \int_0^1 \int_0^1 S(\xi_1^{L+1}, \{\xi_1^k\}_{k=1}^L\}) S(\xi_{2^m}^{L+1}, \{\xi_{2^m}^k\}_{k=1}^L)(1+R_L)\, d\xi_1^k d\xi_{2^m}^k \right] \ldots$$
$$\ldots. \ (\delta_{m-1}^{L+1} - 2\delta_m^{L+1})\, d\xi_{2^m}^{L+1} d\xi_1^{L+1} \tag{8.49}$$

Using the mean value theorem we have a non negative constant $\gamma \leq 2^L$ such

that,

$$I_2 = \int_0^1 \int_0^1 \gamma \left[\prod_{k=1}^{L} \int_0^1 \int_0^1 S(\xi_1^{L+1}, \{\xi_1^k\}_{k=1}^L) S(\xi_{2^m}^{L+1}, \{\xi_{2^m}^k\}_{k=1}^L) \, d\xi_1^k d\xi_{2^m}^k \right] \ldots$$
$$\ldots. \, (\delta_{m-1}^{L+1} - 2\delta_m^{L+1}) \, d\xi_{2^m}^{L+1} d\xi_1^{L+1}$$
$$= \gamma \underbrace{\int_0^1 \int_0^1 g_{L+1}(\xi_1^{L+1}) \, g_{L+1}(\xi_{2^m}^{L+1})(\delta_{m-1}^{L+1} - 2\delta_m^{L+1}) \, d\xi_{2^m}^{L+1} d\xi_1^{L+1}}_{\leq 0 \text{ from the single variable } (K=1) \text{ case}} \tag{8.50}$$

Here, $g_{L+1}(\xi_1^{L+1})$ and $g_{L+1}(\xi_{2^m}^{L+1})$ are the remainders after integrating out the effect of the first L random variables. Thus $I_1 + I_2 \leq 0$ and the inductive hypothesis is true for the case $K = L + 1$. By mathematical induction, the theorem holds for all K.

8.6.4 Corollary 8.3.3 and Notes on the overall Variance

We can use the result of Theorem 8.3.2 in Eq. (8.21) and Eq. (8.22) to obtain,

$$V(\mu_{\text{HILHS}}) \leq \frac{V(S(\boldsymbol{\Xi}))}{2^n} \tag{8.51}$$

Further, on substituting Eq. (8.40) in Eq. (8.21) and Eq. (8.22) we get,

$$V(\mu_{\text{HILHS}}) = \frac{V(S(\boldsymbol{\Xi}))}{2^n} \ldots$$
$$\ldots + \frac{1}{2^{2n}} \sum_{i=1}^{2^n} \sum_{m=1}^{n} 2^{m-1} \left[\sum_{k=1}^{K} \sum_{j=1}^{2^{m-1}} - \left(\int_{\frac{2j-2}{2^m}}^{\frac{2j-1}{2^m}} g_k(\xi_1) \, d\xi_1 - \int_{\frac{2j-1}{2^m}}^{\frac{2j}{2^m}} g_k(\xi_1) \, d\xi_1 \right)^2 \ldots \right.$$
$$\left. \ldots + \mathcal{O}\left(\frac{K(K-1)}{2^{2m-2}} \right) \right]$$
$$= \frac{V(S(\boldsymbol{\Xi}))}{2^n} + \frac{1}{2^n} \sum_{m=1}^{n} 2^{m-1} \left[\sum_{k=1}^{K} \sum_{j=1}^{2^{m-1}} - \left(\int_{\frac{2j-2}{2^m}}^{\frac{2j-1}{2^m}} g_k(\xi_1) \, d\xi_1 - \int_{\frac{2j-1}{2^m}}^{\frac{2j}{2^m}} g_k(\xi_1) \, d\xi_1 \right)^2 \right.$$

$$+ \mathcal{O}\left(\frac{K(K-1)}{2^{2m-2}}\right)\Bigg] \tag{8.52}$$

By Theorem 8.3.2 we know that all the covariance terms are non-positive. We see that the covariance terms, if not zero, are atleast of order $o(\frac{1}{2^n})$. If the terms in the square brackets are of $\mathcal{O}\left(\frac{1}{2^{2m-2}}\right)$, then the covariance terms will be of order $\mathcal{O}\left(\frac{1}{2^n}\right)$.

Chapter 9

Conclusions and Future Work

This dissertation has addressed two major problems. The first was the analysis of electroosmotic flow (EOF) models that utlize the Helmholtz slip boundary condition, especially in the context of the associated adjoint problem. A penalty based variational formulation of such models was developed. It was shown that the adjoint problem for such a formulation is well-posed and that this formulation is adjoint-consistent. Adjoint-based goal-oriented mesh refinement and sensitivity analysis methods were added to the C++ Finite Element software library `libMesh`. These methods were then used for goal oriented mesh refinement for relevant model problems and EOF in straight and T-channel geometries. It was demonstrated that the adjoint methods can substantially improve QoI convergence and accelerate sensitivity analysis. It was also shown that the sensitivity derivative can be useful in devising more accurate estimators for the calculation of normal fluxes, when a penalty method is used for enforcing boundary constraints.

The second problem addressed was the development of an accelerated Monte Carlo method, called Local Sensitivity Derivative Enhanced Monte Carlo (LSDEMC). The new method uses local derivative information to build

surrogates for inexpensive Monte Carlo integration. In conjuction with adjoint sensitivity analysis, LSDEMC can offer superior Monte Carlo convergence with virtually no extra cost for the evaluation of sensitivity derivatives, especially for Finite Element models with a large number of parameters. This new method was used for UQ in a model Poisson problem, in conjuction with goal oriented mesh refinement and adjoint sensitivity derivative evaluation. Finally, a new Latin Hypercube Sampling method was introduced. The new method constructs Latin Hypercube designs with a hierarchical tree based structure. Such a construction enables the application of LHS integration in a less restrictive incremental setting than previous incremental LHS implementations. This is especially important for the use of LHS based UQ in large scale engineering systems, where its use can lead to substantial savings in overall computational costs.

Various directions for future work present themselves. An important question is the asymptotic distribution of LSDEMC based statistical estimators, in the general case where the ratio of the number of true samples to the number of representations is not held constant. Such an analysis can provide us with better variance estimates for the LSDEMC estimator and quantify the dependance of LSDEMC's rate of convergence on dimension size. To complete such an analysis, some theoretical developments will be needed in the field of stochastic geometry, namely, a central limit theorem for weighted moments of Voronoi cells generated by random point processes. For the HLHS designs, additional correlation reduction techniques can be investigated for further im-

provement in the method's convergence properties.

While adjoint-based error analysis can help us improve convergence and quantify the reliability of numerical simulations, more work is needed to develop techniques for actual model validation, with the eventual goal that the reliability of the model in representing reality itself be quantifiable. The slip EOF models considered in this dissertation offer a rich set of possibilities in this regard. In particular, the notion of model adaptivity, with the complete, more expensive EOF model used in corner regions of a device geometry and the inexpensive slip model used in the straight sections, presents a challenge for adaptive modeling methods. Finally, it is hoped that the addition of extensive adjoint analysis support to the `libMesh` library will enable the application of such methods to a broad spectrum of problems in Finite Element analysis.

Bibliography

[1] Adams, Dalbey, Eldred, Gay, Swiler, Bonhoff, Eddy, Haskell, and Hough. DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis Version 5.0 User's Manual. Technical report, SAND2001-3514, Sandia National Labs., Albuquerque, NM (US) Sandia National Labs., Livermore, CA (US), 2009.

[2] N. Agarwal and N. R. Aluru. A domain adaptive stochastic collocation approach for analysis of MEMS under uncertainties. *Journal of Computational Physics*, 228(20):7662–7688, 2009.

[3] N. Agarwal and N. R. Aluru. Stochastic Analysis of Electrostatic MEMS Subjected to Parameter Variations. *Journal of Microelectromechanical Systems*, 18(6), 2009.

[4] D.N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM Journal on Numerical Analysis*, pages 742–760, 1982.

[5] I. Babuska. The finite element method with penalty. *Math. Comp*, 27(122):221–228, 1973.

[6] I. Babuška, U. Banerjee, and J.E. Osborn. Survey of meshless and generalized finite element methods: a unified approach. *Acta Numerica*, 12(1):1–125, 2003.

[7] I. Babuška, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Numer. Anal*, 45(3):1005–1034, 2007.

[8] S. Balay, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. *PETSc Users Manual*, 2003.

[9] I. Bárány, F. Fodor, and V. Vígh. Intrinsic volumes of inscribed random polytopes in smooth convex bodies. *Advances in Applied Probability*, 42(3):605–619, 2010.

[10] W.L. Barth and G.F. Carey. On a boundary condition for pressure-driven laminar flow of incompressible fluids. *International Journal for Numerical Methods in Fluids*, 54(11):1313–1325, 2007.

[11] P.T. Bauman, J.T. Oden, and S. Prudhomme. Adaptive multiscale modeling of polymeric materials with Arlequin coupling and goals algorithms. *Computer Methods in Applied Mechanics and Engineering*, 198(5-8):799–818, 2009.

[12] P.R. Baxandall and H. Liebeck. *Vector Calculus*. Clarendon Press, 1986.

[13] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*, 10:1–102, 2003.

[14] J.F. Bonnans and F. J. Silva. Asymptotic expansions for interior solutions of semilinear elliptic problems. *SIAM Journal on Control and Optimization*, 49(6), 2011.

[15] L. V. Branets. *A variational grid optimization method based on a local cell quality metric*. PhD thesis, The University of Texas at Austin, 2005.

[16] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods*. Springer-Verlag, 1991.

[17] K.Q. Brown. Voronoi diagrams from convex hulls. *Information Processing Letters*, 9(5):223–228, 1979.

[18] Y. Cao, M. Y. Hussaini, and T. A. Zang. Exploitation of sensitivity derivatives for improving sampling methods. *AIAA Journal*, 42(4):815–822, 2004.

[19] Y. Cao, M.Y. Hussaini, T. Zang, and A. Zatezalo. A variance reduction method based on sensitivity derivatives. *Applied Numerical Mathematics*, 56(6):800–813, 2006.

[20] GF Carey. Derivative calculation from finite element solutions. *Computer Methods in Applied Mechanics and Engineering*, 35(1):1–14, 1982.

[21] V. Carey and G.F. Carey. Flexible patch post-processing recovery strategies for solution enhancement and adaptive mesh refinement. *International Journal for Numerical Methods in Engineering*, 2011.

[22] V. Carey, D. Estep, and S. Tavener. A posteriori analysis and adaptive error control for multiscale operator decomposition solution of elliptic systems I: Triangular systems. *SIAM J. Numer. Anal*, 47(1):740–761, 2009.

[23] C. H. Chen, H. Lin, S. K. Lele, and J. G. Santiago. Convective and absolute electrokinetic instability with conductivity gradients. *Journal of Fluid Mechanics*, 524:263–303, 2005.

[24] E. R. Choban, L. J. Markoski, A. Wieckowski, and P. J. A. Kenis. Microfluidic fuel cell based on laminar flow: 1. *Journal of Power Sources*, 128(1):54–60, 2004.

[25] H.W. Choi. *A-posteriori finite element output bounds for the electro-osmotic flow in microchannels*. PhD thesis, University of Toronto, 2006.

[26] H.W. Choi and M. Paraschivoiu. Advanced hybrid-flux approach for output bounds of electro-osmotic flows: adaptive refinement and direct equilibrating strategies. *Microfluidics and Nanofluidics*, 2(2):154–170, 2006.

[27] R. Courant. Calculus of variations and supplementary notes and exercises. *New York University*, 1956.

[28] T. J. Craven, J. M. Rees, and W. B. Zimmerman. On slip velocity boundary conditions for electroosmotic flow near sharp corners. *Physics of Fluids*, 20, 2008.

[29] B. Debusschere, H. Najm, A. Matta, T. Shu, O. Knio, R. Ghanem, and O. Le Maitre. Uncertainty quantification in a reacting electrochemical microchannel flow model. In *Proc. 5th Int. Conf. on Modeling and Simulation of Microsystems*, pages 384–387, 2002.

[30] M. Delfour and J. Zolesio. *Shapes and Geometries: Analysis, Differential Calculus, and Optimization*. SIAM, 2001. Vol. 4 of SIAM Series on Advances in Design and Control.

[31] L. Demkowicz. *Computing with hp-adaptive Finite Elements: One and two dimensional elliptic and Maxwell problems*. CRC Press, 2006.

[32] J. A. Demmel. *Applied Numerical Linear Algebra*. SIAM, 1997.

[33] P. Dutta and A. Beskok. Analytical solution of combined electroosmotic/pressure driven flows in two-dimensional straight channels: finite Debye layer effects. *Anal. Chem*, 73(9):1979–1986, 2001.

[34] R.P. Dwight. Heuristic a-posteriori estimation of error due to dissipation in finite volume schemes and application to mesh adaptation. *Journal of Computational Physics*, 227(5):2845–2863, 2008.

[35] M. S. Eldred, S. R. Subia, D. Neckels, M. M. Hopkins, P. K. Notz, B. M. Adams, B. Carnes, J. W. Wittwer, B. J. Bichon, and K. D. Copps.

Solution-verified reliability analysis and design of bistable MEMS using error estimation and adaptivity. Technical report, SAND2006-6286, Sandia National Laboratories, 2006.

[36] A. Ern and J. L. Guermond. *Theory and practice of finite elements*. Springer Verlag, 2004.

[37] D. Estep, V. Carey, V. Ginting, S. Tavener, and T. Wildey. A posteriori error analysis of multiscale operator decomposition methods for multiphysics models. *J. Phys.: Conf. Ser.*, 125(1), 2008.

[38] D. Estep, S. Tavener, and T. Wildey. A posteriori error estimation and adaptive mesh refinement for a multi-discretization operator decomposition approach to fluid-solid heat transfer. *J. Comput. Phys.*, 229:4143–4158, 2010.

[39] D. J. Estep. A short course on duality, adjoint operators, Green's functions, and a posteriori error analysis. *Lecture Notes*, 2004.

[40] V.V. Garg, S. Prudhomme, K.G. van der Zee, and G.F. Carey. Adjoint consistent formulations of slip models for coupled electroosmotic flow systems. *Journal of Computational Physics*, Submitted.

[41] D. R. Gaston, G. Carey, R. Stogner, J. Peterson, B. Kirk, and L. Branets. On combining mesh redistribution with h-adaptivity, 2007.

[42] R.G. Ghanem. V&V or Investigation on the Multiple Personalities of Predictions. Talk given at The University of Texas at Austin on March 30, 2010.

[43] M. Giles, M. Larson, M. Levenstam, and E. Suli. Adaptive error control for finite element approximations of the lift and drag coefficients in viscous flow. *Technical Report NA-97/06, Comlab, Oxford University*, 1997.

[44] M.B. Giles and E. Süli. Adjoint methods for pdes: a posteriori error analysis and postprocessing by duality. *Acta Numerica*, 11(1):145–236, 2002.

[45] Gizmag. http://c0378172.cdn.cloudfiles.rackspacecloud.com/lab_on_a_chip.jpg.

[46] J. Hahm, A. Balasubramanian, and A. Beskok. Flow and species transport control in grooved microchannels using local electrokinetic forces. *Physics of Fluids*, 19, 2007.

[47] J. C. Helton and F. J. Davis. Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliability Engineering & System Safety*, 81(1), 2003.

[48] M.A. Heroux, R.A. Bartlett, V.E. Howle, R.J. Hoekstra, J.J. Hu, T.G. Kolda, R.B. Lehoucq, K.R. Long, R.P. Pawlowski, E.T. Phipps, et al.

An overview of the Trilinos project. *ACM Transactions on Mathematical Software (TOMS)*, 31(3):423, 2005.

[49] R.L. Iman and WJ Conover. A distribution-free approach to inducing rank correlation among input variables. *Communications in Statistics-Simulation and Computation*, 11(3):311–334, 1982.

[50] M. Ionescu-Bujor and D.G. Cacuci. A comparative review of sensitivity and uncertainty analysis of large-scale systems. i: Deterministic methods. *Nuclear Science and Engineering*, 147(3):189–203, 2004.

[51] E. Jimenez, N. Lay, and M. Y. Hussaini. A systematic study of efficient sampling methods to quantify uncertainty in crack propagation and the Burgers equation. *Monte Carlo Methods and Applications*, 16(1):69–93, 2010.

[52] M. H. Kalos and P. A. Whitlock. *Monte Carlo Methods*. Wiley-VCH, New York, 2008.

[53] G. Karniadakis, A. Beskok, and N. R. Aluru. *Microflows and Nanoflows: Fundamentals and Simulation*. Springer Verlag, 2005.

[54] D. W. Kelly, Gago, O. C. Zienkiewicz, and I. Babuska. A posteriori error analysis and adaptive processes in the finite element method: Part I-Error analysis. *International Journal for Numerical Methods in Engineering*, 19(11):1593–1619, 1983.

[55] N. Kikuchi and J.T. Oden. *Contact problems in elasticity: a study of variational inequalities and finite element methods*, volume 8. Society for Industrial Mathematics, 1988.

[56] Benjamin Kirk, John Peterson, Roy Stogner, and Graham Carey. libMesh: a C++ library for parallel adaptive mesh refinement/coarsening simulations. *Engineering with Computers*, 22(3):237–254, December 2006.

[57] O.M. Knio, R.G. Ghanem, A. Matta, H.N. Najm, B. Debusschere, and O.P. LeMaitre. Quantitative Uncertainty Assessment and Numerical Simulation of Micro-Fluid Systems. Technical report, Johns Hopkins University, Baltimore, MD, 2005.

[58] Stanford Microfluidics Lab. http://microfluidics.stanford.edu/Projects/Archive/bioanal/edl.gif.

[59] M. G. Larson, R. Soderlund, and F. Bengzon. Adaptive finite element approximation of coupled flow and transport problems with applications in heat transfer. *International Journal for Numerical Methods in Fluids*, 57(9):1397–1420, 2008.

[60] A. P. Lee. Digital microfluidics for bioassays and drug delivery. In *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, volume 2, 2004.

[61] Z.C. Li. Boundary penalty finite element methods for blending surfaces ii: biharmonic equations. *Journal of Computational and Applied*

Mathematics, 110(1):155–176, 1999.

[62] J. H. Masliyah and S. Bhattacharjee. *Electrokinetic and Colloid Transport Phenomena*. Wiley-Interscience, 2006.

[63] B. Maury. Numerical analysis of a finite element/volume penalty method. *SIAM J. Numer. Anal*, 47(2):1126–1148, 2009.

[64] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.

[65] M. Motamed, F. Nobile, and R. Tempone. A stochastic collocation method for the second order wave equation with a discontinuous random speed. Presentation at USNCCM 2011, Minneapolis, USA, 2011-07-28.

[66] W.L. Oberkampf and C.J. Roy. *Verification and validation in scientific computing*. Cambridge University Press, 2010.

[67] J.T. Oden, N. Kikuchi, and Y.J. Song. Penalty-finite element methods for the analysis of Stokesian flows. *Computer Methods in Applied Mechanics and Engineering*, 31(3):297–329, 1982.

[68] J.T. Oden and S. Prudhomme. Goal-oriented error estimation and adaptivity for the finite element method. *Computers & Mathematics with Applications*, 41(5-6):735–756, 2001.

[69] K. Ohtake, J.T. Oden, and N. Kikuchi. Analysis of certain unilateral problems in von Karman plate theory by a penalty method. i- a variational principle with penalty. *Computer Methods in Applied Mechanics and Engineering*, 24:187–213, 1980.

[70] K. Ohtake, J.T. Oden, and N. Kikuchi. Analysis of certain unilateral problems in von Karman plate theory by a penalty method. ii- approximation and numerical analysis. *Computer Methods in Applied Mechanics and Engineering*, 24:317–337, 1980.

[71] A. B. Owen. A central limit theorem for Latin hypercube sampling. *Journal of the Royal Statistical Society. Series B (Methodological)*, 54(2), 1992.

[72] A. B. Owen. Controlling Correlations in Latin Hypercube Samples. *Journal of the American Statistical Association*, 89(428), 1994.

[73] M. F. Pellissetti and G. I. Schuëller. Scalable uncertainty and reliability analysis by integration of advanced monte carlo simulation and generic finite element solvers. *Computers & Structures*, 87(13-14), 2009.

[74] J. W. Peterson. *Parallel adaptive finite element methods for problems in natural convection.* PhD thesis, The University of Texas at Austin, 2008.

[75] M. Picasso. Adaptive finite elements with large aspect ratio based on an anisotropic error estimator involving first order derivatives. *Comp.*

Meth. Appl. Mech. Eng., 196(14-23), 2006.

[76] Boedeker Plastics. Polystyrene datasheet. http://www.boedeker.com/polyst_p.htm.

[77] J. B. Pleming and R. D. Manteufel. Replicated latin hypercube sampling. In *Proceedings of the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, pages 1–18, 2005.

[78] V. Prachittham, M. Picasso, and M.A.M. Gijs. Adaptive finite elements with large aspect ratio for mass transport in electroosmosis and pressure-driven microflows. *International Journal for Numerical Methods in Fluids*, 63(9):1005–1030, 2010.

[79] S. Prudhomme and J. T. Oden. Computable error estimators and adaptive techniques for fluid flow problems. *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, 25:207–268, 2003.

[80] S. Prudhomme and J.T. Oden. On goal-oriented error estimation for elliptic problems: application to the control of pointwise errors. *Computer Methods in Applied Mechanics and Engineering*, 176(1):313–331, 1999.

[81] M. M. Putko, P. A. Newman, and A. C. Taylor. Employing sensitivity derivatives to estimate uncertainty propagation in CFD. *9th ASCE Spe-*

cialty Conference on Probabilistic Mechanics and Structural Reliability, 2004.

[82] Peter Z. G. Qian. Nested latin hypercube designs. *Biometrika*, September 2009.

[83] Peter Z. G. Qian. Sliced Latin Hypercube Designs. *Journal of the American Statistical Association*, To Appear 2011.

[84] M. Reitzner. Central limit theorems for random polytopes. *Probability Theory and Related Fields*, 133(4):483–507, 2005.

[85] M. Reitzner and V.V. Garg. Personal communication, 2012. Email Exchange.

[86] L. Ren, D. Sinton, and D. Li. Numerical simulation of microfluidic injection processes in crossing microchannels. *Journal of Micromechanics and Microengineering*, 13:739, 2003.

[87] P.J. Roache. Verification and validation in computational science and engineering. *Computing in Science Engineering*, pages 8–9, 1998.

[88] D. G. Robinson. An iterative Monte Carlo method for efficient structural reliability and uncertainty analysis. *Structural Safety and Reliability: ICOSSAR'01*, page 2001, 2001.

[89] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

[90] C.J. Sallaberry, J.C. Helton, and S.C. Hora. Extension of latin hypercube samples with correlated variables. *Reliability Engineering & System Safety*, 93(7):1047–1059, 2008.

[91] F. Sayas. Weak normal derivatives, normal and tangential traces, and tangential differential operators on Lipschitz boundaries. 2009. Unpublished notes, available on request.

[92] I.M. Sobol. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 7(4):784–802, 1967.

[93] T. M. Squires and S. R. Quake. Microfluidics: Fluid physics at the nanoliter scale. *Reviews of Modern Physics*, 77(3):977–1026, 2005.

[94] C. Stein. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proc. Sixth Berkeley Symp. Math. Stat. Prob.*, pages 583–602, 1972.

[95] M. Stein. Large sample properties of simulations using Latin hypercube sampling. *Technometrics*, 29(2):143–151, 1987.

[96] R. H. Stogner and G. F. Carey. C1 macroelements in adaptive finite element methods. *Int. J. Numer. Meth. Eng*, 70(9):1076–1095, 2007.

[97] R. H. Stogner and V.V. Garg. Physics-Independent Adjoint Capabilities with libMesh. In Preparation, 2012.

[98] R.H. Stogner and V.V. Garg. Local Sensitivity Derivative Enhanced Monte Carlo Methods. *Journal of the American Statistical Association*, Submitted.

[99] M. M. Teymoori and E. Abbaspour-Sani. Design and simulation of a novel electrostatic peristaltic micromachined pump for drug delivery applications. *Sensors and Actuators A: Physical*, 117(2):222–229, 2005.

[100] M. Utku and G.F. Carey. Boundary penalty techniques. *Computer Methods in Applied Mechanics and Engineering*, 30(1):103–118, 1982.

[101] E.H. van Brummelen, K.G. van der Zee, V.V. Garg, and S. Prudhomme. Flux evaluation in primal and dual boundary-coupled problems. *Journal of Applied Mechanics*, 2011. Accepted.

[102] V. Vu. Central limit theorems for random polytopes in a smooth convex set. *Advances in Mathematics*, 207(1):221–243, 2006.

[103] Q. Wang. *Uncertainty quantification for unsteady fluid flow using adjoint-based approaches*. PhD thesis, Stanford University, 2008.

[104] G. M. Whitesides. The origins and the future of microfluidics. *Nature*, 442(7101):368–373, 2006.

[105] T. Wildey, S. Tavener, and D. Estep. A posteriori error estimation of approximate boundary fluxes. *Communications in Numerical Methods in Engineering*, 24(6):421–434, 2008.

[106] D. Xiu. Fast numerical methods for stochastic computations: a review. *Communications in Computational Physics*, 5(2-4):242–272, 2009.

[107] D. Xiu and G.E. Karniadakis. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, 187(1):137–167, 2003.

[108] Y. Zhang, T. N. Wong, C. Yang, and K. T. Ooi. Electroosmotic flow in irregular shape microchannels. *International Journal of Engineering Science*, 43(19-20):1450–1463, 2005.

[109] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity. *International Journal for Numerical Methods in Engineering*, 33(7):1365–1382, 1992.

[110] W.J. Zimmerman, J. Rees, and T. Craven. Rheometry of non-newtonian electrokinetic flow in a microchannel T-junction. *Microfluidics and Nanofluidics*, 2(6):481–492, November 2006.

Vita

Vikram Vinod Garg was born in Mumbai on December 1, 1985 to Vinod Kumar and Sarita Garg. He completed his schooling at St. Xavier's High School, Mumbai and Thakur Junior College, Mumbai.

He entered the University of Texas at Austin as a freshman in 2003. At UT, Vikram learnt, played and made friends from every continent. He graduated with bachelors degrees in both Engineering and Mathematics in 2007.

Vikram then joined the Computational Science, Engineering and Mathematics (CSEM) PhD. program at UT after his bachelors. He completed his dissertation under the guidance of late Dr. Graham Carey and Dr. Serge Prudhomme, and defended this thesis on May 1, 2012. He will start a postdoc with Dr. Karen Wilcox at the Massachusetts Institute of Technology in September 2012.

Permanent address: vikram.v.garg@gmail.com

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.