Summer 8-2022

Robust Uncertainty Quantification with Analysis of Error in Standard and Non-standard Quantities of Interest

Zachary Stevens
University of New Mexico

Zachary Stevens

Candidate

Mathematics and Statistics

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Dr. Jehanzeb Chaudhary ,Chairperson

Dr. Simon Tavener

Dr. Jacob Schroder

Dr. Stephen Lau

Robust Uncertainty Quantification with Analysis of Error in Standard and Non-standard Quantities of Interest

by

Zachary Stevens

B.S., Applied Mathematics, University of New Mexico, 2016

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Mathematics

The University of New Mexico

Albuquerque, New Mexico

August, 2022

Dedication

To my mother, Deborah Stevens.

Acknowledgments

I would like to thank my advisor Jehanzeb (Zeb) Chaudhry. Zeb has aided in my work and education since my senior year of undergrad. After my first computational math course with Zeb, he reached out and encouraged me to participate in research, give presentations, and meet with other researchers. I have greatly appreciated Zeb's approach to guidance in which he opened up many opportunities for me and fully supported my choices among them.

I also want to thank the other members of my dissertation committee: Simon Tavener, Jacob Schroder, and Stephen Lau. Simon has helped and supported me on two research articles that form a large part of this thesis. In addition to their feedback on my research, Jacob and Stephen have also helped shape the way I approach teaching.

My family and friends have always been my biggest supporters. My mother, Deborah, is an amazing teacher to me and to her actual students. She has shown me the value of education and passed on a desire to continue learning new things. She also taught me the importance of taking care of those around us and passed on her love for cooking and nature. My father, Mark, has always set an example of dedication that I strive to achieve in my academics and other aspects of my life. He has shown me to take pride in anything I do but to also stay modest by letting good work speak for itself. My brother's and sister's, Taylor and Katrina, love and constant support has helped me in ways I cannot express. My partner, Laura, has been by my side the entire time I was at UNM. I am truly grateful for everything she has done for me, from helping me write MatLab code to cooking dinner when I had a looming deadline. Laura inspires me to always improve myself and I am excited to share future accomplishments with her. My mentor and friend Charles has always encouraged me to follow my passions. He always believed I would change the world. My best friends, Daniel and Cody, have brought much needed balance by reminding me to have fun and enjoy the other aspects of my life.

Finally, I would like to thank everyone in the UNM Department of Mathematics and Statistics, including the many professors who have shared their knowledge, advisors who helped shape my path through UNM, and my fellow students who were always willing to share the struggle or grab a bite to eat. This milestone would not have been possible without all of these incredible people and many others.

Robust Uncertainty Quantification with Analysis of Error in Standard and Non-standard Quantities of Interest

by

Zachary Stevens

B.S., Applied Mathematics, University of New Mexico, 2016

Ph.D., Mathematics, University of New Mexico, 2022

Abstract

This thesis derives two Uncertainty Quantification (UQ) methods for differential equations that depend on random parameters: (**i**) error bounds for a computed cumulative distribution function (**ii**) a multi-level Monte Carlo (MLMC) algorithm with adaptively refined meshes and accurately computed stopping-criteria. Both UQ approaches utilize adjoint-based *a posteriori* error analysis in order to accurately estimate the error in samples of numerically approximated quantities of interest. The adaptive MLMC algorithm developed in this thesis relies on the adjoint-based error analysis to adaptively create meshes and accurately monitor a stopping criteria. This is in contrast to classical MLMC algorithms which employ either a hierarchy of uniform meshes or adaptively refined meshes based on Richardson extrapolation. Moreover, they also use a stopping criteria that relies on assumptions on the convergence rate of the MLMC levels. This thesis overcomes these drawbacks of the classical algorithms.

The analysis and UQ methods developed in this these are applied to several types of differential equations and quantities of interest. Classical *a posteriori* error analysis

provides a formulation of the error in a Quantity of Interest (QoI) which is represented as a bounded linear functional of the solution to a differential equation. This thesis derives error estimates for a QoI that describes the *time at which* a linear functional of the solution achieves a threshold value. This QoI is referred to as "non-standard" since it cannot be represented as a linear functional of the solution. The classical analysis does not directly apply to non-standard QoIs.

The adjoint-based error analysis for the different QoIs not only develops accurate error estimates, they also provide a decomposition of the error into contributions from different regions of the domain. The decompositions are utilized to adaptively create meshes when adding new levels in the MLMC algorithm. Two adaptive mesh creation methods are described that can be used to build the MLMC estimator. Many numerical experiments demonstrate the accuracy of the error estimates and the advantages of using adaptive mesh creation in the MLMC algorithm.

Contents

i Notation

Spaces

$\mathbb{R}^d$	Euclidean space of dimension d.
x	Spatial variable in $\mathbb{R}^d$.
t	Temporal variable in $\mathbb{R}$.
Ω	Sub-space of the Euclidean space; $\Omega \subset \mathbb{R}^d$.
$(0, T]$	Interval in $\mathbb{R}$ is temporal domain for differential equations.
$\mathcal{T}_h$	Conforming simplicial decomposition of domain Ω.
$\mathcal{T}_k$	Partition of time domain $[0, T]$
$L^2(\Omega)$	Space of square integrable functions over Ω.
$L^2(\Omega, \mu)$	Lebesgue space of functions that are square-integrable over Ω with respect to measure μ.
$H^1(\Omega)$	Hilbert space where first weak derivatives are $L^2(\Omega)$.
$H^s(\Omega), H_0^s(\Omega)$	Sobolev spaces $W^{k,2}(\Omega)$ and $W_0^{k,2}(\Omega)$, respectively.
$\mathcal{P}^q(\Omega)$	Space of polynomials of degree at most q defined over Ω.
P_h^q	Lagrange finite element space of degree q defined over a simplicial decomposition $\mathcal{T}_h$.

Functions

u, U	True and Computed solutions of a differential equation.
ϕ, ϕ_i	Solutions to adjoint problems.
ψ, ψ_i	Weight functions that define the QoIs.
f	Possibly non-linear RHS of a differential equation

Operations

$\alpha \cdot \beta = \sum_{i=1}^{d} \alpha_i \beta_i$	Euclidean Inner product for $\alpha, \beta \in \mathbb{R}^d$
$(\alpha, \beta)_\Omega = \int_\Omega \alpha \cdot \beta \mathrm{d}\Omega$	L^2 inner-product over domain Ω for $\alpha, \beta \in L^2(\Omega)$.
$\mathcal{D}$	Linear combination of differential operators with respect to spatial variable x.

Differential Equations

$\dot{u} = f$	Equation for Initial-Value Problem.
$\mathcal{D}u = f$	Equation for Boundary-Value Problem.
$\dot{u} + \mathcal{D}u = f$	Equation for Initial-Boundary-Value Problem.
$cG(q)$	Galerkin finite elements method using a degree q continuous element space.
$dG(q)$	Galerkin finite elements method using a degree q discontinuous element space.
ε	Tolerance for uncertainty quantification method.

Quantities of Interest

$Q(u)$	A quantity of interest for the function u.
$Q_S(u), Q_{NS}(u)$	A standard QoI and a nonstandard QoI for the function u.
R	Target value for the NSQoI.
$G(u;t)$	Auxiliary linear functional of $u(x,t)$ implicitly dependent on t. Used to define $Q_{NS}(u)$.
$H(u,\widetilde{t})$	Auxiliary functional of u and a chosen $\widetilde{t}$. Used to define $Q_{NS}(u)$
t_t, t_c	True value and computed value of the nonstandard QoI.

Statistics

$P(\Theta \leq \theta)$	Probability of a sample of Θ taking value less than or equal to θ.
$\mathbb{E}[\Theta]$	Expected value of Θ.
$\mathbb{V}[\Theta]$	Variance of Θ.
$\widehat{\Theta}$	An estimator of a random variable Θ.
$\text{Bias}[\widehat{\Theta}, \Theta]$	Expectation of the difference between the estimator $\widehat{\Theta}$ and the random variable Θ.
$\text{MSE}_{\Theta}[\hat{\Theta}]$	Mean-squared error of the estimator $\hat{\Theta}$ with respect to Θ.
$\widehat{\Theta}_N^{MC}$	Monte Carlo estimator of expected value of Θ using N samples.
$\widehat{\Theta}_{L,\{N_\ell\}}^{ML}$	Multilevel Monte Carlo estimator of Θ using L levels with N_ℓ samples on the ℓ-th level for $\ell = 0, 1, \ldots, L-1$.

Abbreviations

ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
cG	Continuous Galerkin finite element method
dG	Discontinuous Galerkin finite element method
CDF	Cumulative Distribution Function
MSE	Mean Squared Error
MC	Monte Carlo
MLMC	Multilevel Monte Carlo
QoI(s)	Quantity(ies) of Interest
SQoI	Standard Quantity of Interest
NSQoI	Nonstandard Quantity of Interest
UQ	Uncertainty Quantification
C-N	Crank-Nicolson finite difference method
FDM	Finite Difference Method
FEM	Finite Element Method
IVP	Initial-Value Problem involving an ODE
BVP	Boundary-Value Problem involving a PDE
IBVP	Initial-Boundary-Value Problem involving a PDE
RHS	Right-hand side (of an equation)
LHS	Left-hand side (of an equation)

Chapter 1

Introduction

Uncertainty Quantification (UQ) is a rapidly developing subject that plays a key role in scientific computing, predictive science, and numerical analysis. The study of UQ looks to quantify the uncertainty in an output statistic (such as cumulative distribution function, expected value, mean-squared error, etc.) based on the uncertainty of the inputs of the model.

A type of classical, broad approach for UQ in predictive computational sciences are Monte Carlo (MC) methods [43,48,54]. MC methods are simple to implement. As such they are used in a vast array of disciplines including the high-energy physics involved in particle collision, particle transport, finance, statistical mechanics and numerous other engineering applications [49, 55]. MC methods have been modified and expanded to improve cost effectiveness. Some of the methods based on MC methods are the Quasi-Monte Carlo (QMC), Markov chain Monte Carlo (MCMC), multi-fidelity Monte Carlo (MFMC), and multi-level Monte Carlo (MLMC) methods [3, 9, 22, 32, 46, 57]. Further modifications have been made to some of these methods, including adaptive MLMC methods [51], which employ adaptive refinement strategies for pathwise problems.

In particular, UQ for differential equations that depend on some random parameter(s) is of high interest, with several science and engineering applications including

controls, communication, cyber-security, and finance [1, 31]. Often, a particular quantity related to the solution of a differential equation is of interest; for example, the concentration of a chemical over a sub-region of the domain, or the time at which an oscillator reaches a certain position. If a differential equation depends on a random parameter, then the related quantity of interest (QoI) is a random variable and an UQ of this QoI is required.

This thesis develops two forms of UQ for QoIs related to differential equations. In one, we construct an upper-bound for the error in an approximate CDF of the QoI where samples are taken using the Monte Carlo (MC) method. The bound requires the error in samples of approximate QoIs, which are accurately computed using adjoint-based *a posteriori* error analysis. The other form of UQ developed in this thesis focuses on creating an adaptive multi-level Monte Carlo (MLMC) estimator for expected value. Many MLMC algorithms use uniform meshes when creating the levels of the estimator [46]. Past algorithms, including the adaptive methods, rely on assumptions about the convergence of the levels of the estimator and use extrapolation to determine a stopping criteria [47, 48, 51]. In our adaptive MLMC algorithm, we implement adjoint-based error analysis to accurately estimate the bias of the estimator, which is used to monitor a stopping criteria. The error analysis also provides a way to decompose the bias contributions from different regions of the domain. We use this decomposition to adaptively create the meshes that are used when added a new level to the MLMC estimator.

The UQ methods developed in this thesis utilize adjoint-based *a posteriori* error analysis in order to accurately estimate errors in the QoIs. The classical *a posteriori* error analysis provides computable error estimates for QoIs that can be represented as bounded functionals of the solution to a differential equation. Many nonlinear QoIs are handled by linearizing around a computed solution [8, 11, 23]. This adjoint-based analysis has been widely studied and shown to provide accurate estimates in a vast array of settings [2, 5, 7, 8, 10–13, 15–20, 25, 27, 28, 34, 36, 38, 40, 41, 50]. The error estima-

tion utilizes generalized Green's functions solving certain adjoint problems, variational analysis, and computable residuals of the numerical solution [2, 8, 28, 37, 40, 41]. The error estimation method is well-suited to use alongside a finite element or variational numerical method. However this error estimation can be applied to many finite difference and finite volume methods, provided they be recast as an equivalent finite element method [12, 17, 21, 25, 27, 29, 30, 35, 39, 53]. The practice of using adjoint equations to gain information about the solution of a differential equation has been common since the 1970s, with applications to optimal control theory, fluid dynamics, and aeronautical engineering [45, 52].

The types of QoI covered by the classical analysis are referred to as "standard" QoIs. In this thesis, a certain "non-standard" QoI (one that cannot be represented by a linear functional nor can it be trivially linearized) is analyzed. The non-standard QoI represents the *time at which* an event occurs; for example this could denote the time at which a chemical concentration surpasses a threshold, the time at which an oscillator returns to its rest position, or the time at which a wave reaches a certain height. While this non-standard QoI is of great importance to many physical situations, an adjoint-based error analysis of this QoI does not exist in previous literature. In this thesis, we develop two adjoint-based error estimates for this non-standard QoI. The first method utilizes Taylor's theorem to construct a representation of the error in the non-standard QoI. Solutions to certain adjoint problems are then used to create a computable, accurate estimate of the error representation; this analysis was first published in [21] and expanded upon in [14]. The second relies on solutions to adjoint problems and root-finding methods in order to obtain a "corrected" QoI; this analysis was first published in [21].

1 Differential Equation with random parameter(s)

The UQ presented in this thesis is applicable to any quantity of interest (QoI) related to a differential equation that depends on a random parameter, *provided* there is a computable representation or estimate of the error in an approximation of the QoI. Classical *a posteriori* error analysis provides error representations for a wide array of QoI related to several different types of differential equations. We divide these types of differential equations into three broad classes: initial-value problems (IVPs) with ODEs, boundary-value problems (BVPs) with PDEs, and initial-boundary-value problems (IBVPs) with PDEs.

1.1 IVPs

The model initial-value problem with ODEs takes the form

$$\begin{cases} \dot{u} & = f, \quad t \in (0, T], \\ u(0) & = u_0, \end{cases} \tag{1.1}$$

where $\dot{u} = \frac{du}{dt}$, $f = f(u, t; w) \in \mathbb{R}^d$ is a Lipschitz continuous function, and $u(t) \in \mathbb{R}^d$ is the solution to the IVP. The function f or the initial condition $u_0 = u_0(w)$ may depend on a random variable w. Even though the model problem is first order, the analysis of (1.1) applies to higher order IVPs as well. If $u(t) \in \mathbb{R}^d$, any nth order IVP can be rewritten as a system of nd first order IVPs, through a standard change of variables. One example of an IVP of form (1.1) is the harmonic oscillator

$$m\ddot{y} + c\dot{y} + ky = \tilde{f}, \tag{1.2}$$

which, after a reduction to first order, can be expressed as

$$\begin{pmatrix} \dot{u}_1(t) \\ \dot{u}_2(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -k/m & -c/m \end{pmatrix} \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} + \begin{pmatrix} 0 \\ \tilde{f}/m \end{pmatrix}. \tag{1.3}$$

Here $m \neq 0$, $u_1 = y$, $u_2 = \dot{y}$ and any of the parameters $m, c, k, \tilde{f}$ may be random. Other examples include the two-body problem, the Lorentz equations, and the logistic equation.

1.2 BVPs

The boundary-value problems involving PDEs take the form

$$\begin{cases} \mathcal{D}u &= f, \quad x \in \Omega, \\ u &= g, \quad x \in \partial\Omega., \end{cases} \tag{1.4}$$

over a domain $\Omega \subset \mathbb{R}^d$ with differential operator $\mathcal{D} = \mathcal{D}(w)$, and continuous functions $f = f(u, x; w)$ and $g = g(x; w)$, any of which may depend on a random variable w. The BVPs in this thesis all have scalar-valued solutions. An example of a BVP of form (1.4) is the (elliptic) stationary advection-diffusion model

$$\nabla^2 u + b \cdot \nabla u = f, \tag{1.5}$$

where the differential operator is $\mathcal{D}u = \nabla^2 u + b \cdot \nabla u$ and b or f may depend on a random variable.

1.3 IBVPs

Initial-boundary-value problems are PDEs with an explicit time variable paired with initial conditions (in time) and boundary values (in space). We denote these problems as

$$\begin{cases} \dot{u} + \mathcal{D}u &= f, \qquad x \in \Omega, \quad t \in (0, T], \\ u(x, t) &= g(x, t), \quad x \in \partial\Omega, \ t \in (0, T], \\ u(x, 0) &= h(x), \quad x \in \Omega, \end{cases} \tag{1.6}$$

over a spatial domain $\Omega \subset \mathbb{R}^d$ and temporal domain $(0, T]$, with (spatial) differential operator $\mathcal{D} = \mathcal{D}(w)$, and continuous functions $f = f(u, x, t; w)$, $g = g(x, t; w)$, and $h = h(x; w)$, any of which may depend on a random variable w. Similar to the IVP (1.1), the IBVP only contains first order time-derivatives because problems containing higher order time-derivatives can be rewritten as a system of first order (in time) equations. The IBVPs used as examples in this thesis all have scalar-valued solutions. An examples of an IBVP (1.6) is the Vlasov equation

$$\dot{u} + (x_2, 0)^\top \cdot \nabla u = f \tag{1.7}$$

where the spatial differential operator is $\mathcal{D}u = (x_2, 0)^\top \cdot \nabla u = x_2 \frac{\partial u}{\partial x_1}$ and f may depend on a random variable. Other examples of IBVPs (1.6) include the (parabolic) heat equation and the (hyperbolic) wave equation.

2 Layout of Thesis

The remainder of this thesis is organized as follows. Chapter 2 contains all of the relevant background results required for the novel work of this thesis. We begin the review of material in §2.1 by recalling the standard definitions of a few statistical quantities: the cumulative distribution function, expectation, variance, and mean-squared error. We provide details of Monte Carlo estimators of the expectation and cumulative distribution function of a random variable, as well as the multi-level Monte Carlo estimator for expectation. The review continues in §2.2 where we recall the function spaces and dual spaces that are relevant to our analysis. We then discuss adjoint operators/problems and how to obtain them based on the differential equation and QoI. Section 2.3 reviews the variational forms of the model equations and the Galerkin methods used in our numerical experiments. We end the review of material in §2.4 where we provide details of the classical adjoint-based *a posteriori* error analysis for standard QoIs. This includes the error estimations for standard QoIs of our three model equations along with the

adjoint problems necessary to compute them. We also provide numerical experiments to illustrate the accuracy of the error estimations.

Chapters 3 and 4 are dedicated to the novel contributions of this thesis. First, in 3 we provide a rigorous definition of the non-standard QoI followed by *a priori* convergence results. We then derive the two *a posteriori* error analysis methods for this QoI. Numerical experiments showing the accuracy and limitations of both methods are also provided. Finally, in 4 we derive the two novel methods of uncertainty quantification. We derive an upper bound of the error in an estimated cumulative distribution function which relies on *a posteriori* error analysis to be made computable. We also create a novel adaptive multi-level Monte Carlo algorithm which utilizes the *a posteriori* error analysis in order to adaptively refine meshes and accurately compute a stopping criteria. Numerical experiments of the uncertainty quantification methods are provided.

Chapter 2

Review of Material

This chapter provides the background material required to derive the novel work in this thesis. We begin with a review of the relevant statistics material in §2.1. Some basic definitions are provided in §2.1.1. The Monte Carlo estimators for expectation and CDF are provided in §2.1.2. The multi-level Monte Carlo estimator for expectation is reviewed in §2.1.3.

We then turn to a discussion of the classical adjoint-based *a posteriori* error analysis. To begin this discussion we briefly recall the relevant functional analysis in §2.2. This includes a review of certain function spaces, dual spaces, adjoint operators, and adjoint problems. We then provide the variational forms of our three model equations and review Galerkin methods which are used to numerically solve them. Finally, in §2.4 we discuss the details of the classical *a posteriori* error analysis and provide numerical experiments demonstrating the accuracy of these methods.

1 Statistics background

When a differential equation contains a random variable as a parameter, the corresponding QoI can also be viewed as a random variable. We are interested in quantifying the error in computed expected values or cumulative distribution functions of the random QoI. This section discusses the relevant statistics background needed for our approaches to uncertainty quantification. We first present the standard definitions of the cumulative distribution function of a random variable, as well as the expected value and variance. We also briefly discuss estimators of a random variable and the mean squared error of an estimator. Monte Carlo estimators for both the expected value and the cumulative distribution function of a QoI are presented along with the multi-level Monte Carlo estimator for the expected value.

1.1 Basics

This section defines some basic statistics that are often of interest in UQ. In particular, we define the cumulative distribution function, expected value, and variance of an absolutely continuous random variable Θ and present useful properties of each. We also provide the definition for the mean-squared error of an estimator of the random variable.

Cumulative Distribution Functions

For an absolutely continuous random variable Θ belonging to the probability space $(\Omega, \mathcal{F}, \mathbb{P})$, the cumulative distribution function (CDF), evaluated at some θ, is a function that represents the probability that Θ takes a value less than θ. The CDF is denoted as

$$F_\Theta(\theta) = P(\Theta \leq \theta), \tag{2.1}$$

where P denotes the probability function. Some useful properties of the CDF include

$$0 \leq F_\Theta(\theta) \leq 1, \quad \forall\theta, \tag{2.2}$$

$$F_\Theta(\theta_1) \leq F_\Theta(\theta_2) \quad \text{whenever } \theta_1 \leq \theta_2, \tag{2.3}$$

$$\lim_{\theta\to\infty} F_\Theta(\theta) = 1, \tag{2.4}$$

$$\lim_{\theta\to-\infty} F_\Theta(\theta) = 0. \tag{2.5}$$

From the CDF we can define a density function for the absolutely continuous random variable, Θ, as

$$f_\Theta(\theta) = \frac{dF_\Theta}{d\theta}(\theta). \tag{2.6}$$

The density function $f_\Theta(\theta)$ represents the relative likelihood that Θ takes a value near θ.

Expected Values

The expected value, or average value, of an absolutely continuous random variable is defined via the density function:

$$\mathbb{E}[\Theta] = \int_{-\infty}^{\infty} \theta f_\Theta(\theta)\mathrm{d}\theta. \tag{2.7}$$

For two independent random variables Θ_1 and Θ_2, the expected value is linear

$$\mathbb{E}[a\Theta_1 + b\Theta_2] = a\mathbb{E}[\Theta_1] + b\mathbb{E}[\Theta_2], \qquad \text{for real numbers } a, b. \tag{2.8}$$

The expected value of a continuous random variable Θ can be estimated from a finite number of samples. Let $\{\theta_n\}_{n=1}^N$ be N samples of the random variable Θ. Then the sample mean gives an approximation of the expected value and is defined as

$$\mathbb{E}[\Theta] \approx \frac{1}{N}\sum_{n=1}^{N} \theta_n. \tag{2.9}$$

Variance

The variance of a random variable gives information on how much the samples of Θ vary away from the expected value. The variance is defined via the density function as

$$\mathbb{V}[\Theta] = \int_{-\infty}^{\infty} (\theta - \mathbb{E}[\Theta])^2 f_{\Theta}(\theta) \mathrm{d}\theta. \tag{2.10}$$

The variance can also be obtained using the expectation as

$$\mathbb{V}[\Theta] = \mathbb{E}[(\Theta - \mathbb{E}[\Theta])^2] = \mathbb{E}[\Theta^2] - (\mathbb{E}[\Theta])^2 . \tag{2.11}$$

Other useful properties of variance are

$$\mathbb{V}[\Theta] \geq 0 \tag{2.12}$$

$$\mathbb{V}[a\Theta + b] = a^2 \mathbb{V}[\Theta], \quad \text{for real numbers } a, b. \tag{2.13}$$

The standard deviation of the random variable is given as the square-root of the variance

$$\sigma[\Theta] = \sqrt{\mathbb{V}[\Theta]}. \tag{2.14}$$

Mean Squared Error

The mean squared error (MSE) of an *estimator*, $\widehat{\Theta}$ (which is also a random variable), of a random variable Θ is a measure of the average squared-difference between the estimator and the random variable. The MSE is given as

$$\mathrm{MSE}[\widehat{\Theta}] = \mathbb{E}\left[(\widehat{\Theta} - \Theta)^2\right] \tag{2.15}$$

$$= \mathbb{V}[\widehat{\Theta}] + \left(\mathrm{Bias}[\widehat{\Theta}, \Theta]\right)^2, \tag{2.16}$$

where the variance $\mathbb{V}[\widehat{\Theta}]$ is given in (2.11) and the Bias is given as

$$\mathrm{Bias}[\widehat{\Theta}, \Theta] = \mathbb{E}[\widehat{\Theta} - \Theta]. \tag{2.17}$$

The MSE is an *a priori* result, even though it depends on the unknown random data Θ.

Note that the definition of the MSE varies slightly when discussing a *predictor* of a random variable instead of an estimator. For our purposes, we only focus on estimators.

1.2 Monte Carlo

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space with *sample space* Ω, *event space* $\mathcal{F}$, and probability function $\mathbb{P}$. Let $Q = Q(u; w)$ be a QoI related to the solution of a differential equation u dependent on a random variable w which belongs to the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Recall that the solution $u = u(w)$ and thus the QoI $Q(u; w) = Q(u(w); w)$ depend. on the random variable w. As such, the QoI is a random variable and has an associated expected value and CDF.

Let $Q(w^{(n)}) = Q(u; w^{(n)})$, for $n = 1, \ldots, N$, be samples of Q, where $w^{(n)}$ is sampled from the same probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Given a numerical solution U, let $\widehat{Q}(w^{(n)}) = \widehat{Q}(U; w^{(n)})$ be an approximation of $Q(w^{(n)})$. The standard Monte Carlo method estimates the expected value, $\mathbb{E}[Q]$, of Q using the N samples $\widehat{Q}(w^{(n)})$:

$$\mathbb{E}[Q] \approx \frac{1}{N} \sum_{n=1}^{N} \widehat{Q}(w^{(n)}). \tag{2.18}$$

The Monte Carlo method can also be used to estimate the CDF of a random variable. Denote the CDF of the random variable Q as

$$F_Q(\theta) = P(\{w : Q(u; w) \leq \theta\}) = P(Q \leq \theta). \tag{2.19}$$

An approximation to the CDF is computed using the Monte Carlo method with the N samples of numerically computed values $\left\{\widehat{Q}(w^{(n)})\right\}_{n=1}^{N}$,

$$\widehat{F}_N(\theta) = \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}(\widehat{Q}(w^{(n)}) \leq \theta), \tag{2.20}$$

where $\mathbb{1}$ is the indicator function:

$$\mathbb{1}(x \leq y) = \begin{cases} 1 \text{ if } x \leq y, \\ 0 \text{ if } x > y. \end{cases} \tag{2.21}$$

One goal of this thesis is to construct and analyze a computable bound for the error, $\left|F_Q(\theta) - \widehat{F}_N(\theta)\right|$, in the approximation of the CDF (2.20).

1.3 Multilevel Monte Carlo

The Multi-level Monte Carlo (MLMC) method is an extension of the MC method that utilizes several different estimators $\{\widehat{Q}_\ell = \widehat{Q}(U_\ell; w)\}$ in order to obtain a more cost-efficient approximation of the expected value. More rigorously, let $\{\widehat{Q}_\ell = \widehat{Q}(U_\ell; w)\}_{\ell=0}^{L-1}$ be a collection of numerical QoI where the accuracy of U_ℓ increases with ℓ. At each level ℓ, N_ℓ samples $\{w_\ell^{(n)}\}_{n=1}^{N_\ell}$ of the random variable are taken. The MLMC estimator is constructed by expanding the expected value of the most accurate estimator as

$$\mathbb{E}[Q] \approx \mathbb{E}[\widehat{Q}_{L-1}] = \mathbb{E}[\widehat{Q}_0] + \sum_{\ell=1}^{L-1} \mathbb{E}[\widehat{Q}_\ell - \widehat{Q}_{\ell-1}]. \tag{2.22}$$

Using the standard MC method to obtain the expected values on the right side of (2.22) gives the L-level MLMC estimator for the expected value of Q:

$$\mathbb{E}[Q] \approx \widehat{Q}^{ML}_{L,\{N_\ell\}}, \tag{2.23}$$

$$= \frac{1}{N_0} \sum_{n=1}^{N_0} \widehat{Q}_0(w_0^{(n)}) + \sum_{\ell=1}^{L-1} \left\{ \frac{1}{N_\ell} \sum_{n=1}^{N_\ell} \left(\widehat{Q}_\ell(w_\ell^{(n)}) - \widehat{Q}_{\ell-1}(w_\ell^{(n)}) \right) \right\}, \tag{2.24}$$

$$= \sum_{\ell=0}^{L-1} \left\{ \frac{1}{N_\ell} \sum_{n=1}^{N_\ell} \left(\widehat{Q}_\ell(w_\ell^{(n)}) - \widehat{Q}_{\ell-1}(w_\ell^{(n)}) \right) \right\}, \tag{2.25}$$

where $\widehat{Q}_{-1} \equiv 0$. One *sample* on the $\ell - th$ level is

$$Y_\ell(w_\ell^n) = \left(\widehat{Q}_\ell(w_\ell^{(n)}) - \widehat{Q}_{\ell-1}(w_\ell^{(n)}) \right), \tag{2.26}$$

and requires two different estimates of the QoI. Both estimates come with the same sample of the random parameter w, but they are obtained using the different estimators $\widehat{Q}_\ell$ and $\widehat{Q}_{\ell-1}$. The Mean Squared Error (MSE) of the MLMC estimator is given as

$$\text{MSE} = \mathbb{V}\left[\widehat{Q}^{ML}_{\{N_\ell\},L} \right] + \left(\text{Bias}\left[\widehat{Q}^{ML}_{\{N_\ell\},L}, Q \right] \right)^2 \tag{2.27}$$

$$= \sum_{\ell=0}^{L-1} \frac{1}{N_\ell} \mathbb{V}\left[\widehat{Q}_\ell - \widehat{Q}_{\ell-1} \right] + \left(\mathbb{E}\left[\widehat{Q}_{L-1} - Q \right] \right)^2. \tag{2.28}$$

The first term in (2.28) is the variance, which is decomposed into contributions from each level of the multi-level estimator. The second term is the squared bias, which only

depends on the highest level. The variance at level ℓ and the bias can be approximated using a finite number of samples.

$$\mathbb{V}\left[\widehat{Q}_\ell - \widehat{Q}_{\ell-1}\right] \approx$$
$$\frac{1}{N_\ell - 1}\sum_{n=1}^{N_\ell}\left(\widehat{Q}_\ell(w_\ell^{(n)}) - \widehat{Q}_{\ell-1}(w_\ell^{(n)}) - \frac{1}{N_\ell}\sum_{k=1}^{N_\ell}\left(\widehat{Q}_\ell(w_\ell^{(k)}) - \widehat{Q}_{\ell-1}(w_\ell^{(k)})\right)\right)^2, \quad (2.29)$$

$$\mathbb{E}\left[\widehat{Q}_{L-1} - Q\right] \approx \frac{1}{N_{L-1}}\sum_{n=1}^{N_{L-1}}\left(\widehat{Q}_{L-1}(w_{L-1}^{(n)}) - Q(w_{L-1}^{(n)})\right). \quad (2.30)$$

With the approximations (2.29) and (2.30), the MSE can be approximated as

$$\text{MSE} \approx$$
$$\sum_{\ell=0}^{L-1}\frac{1}{N_\ell}\left[\frac{1}{N_\ell - 1}\sum_{n=1}^{N_\ell}\left(\widehat{Q}_\ell(w_\ell^{(n)}) - \widehat{Q}_{\ell-1}(w_\ell^{(n)}) - \frac{1}{N_\ell}\sum_{k=1}^{N_\ell}\left(\widehat{Q}_\ell(w_\ell^{(k)}) - \widehat{Q}_{\ell-1}(w_\ell^{(k)})\right)\right)^2\right]$$
$$+\left(\frac{1}{N_{L-1}}\sum_{n=1}^{N_{L-1}}\left(\widehat{Q}_{L-1}(w_{L-1}^{(n)}) - Q(w_{L-1}^{(n)})\right)\right)^2. \quad (2.31)$$

The squared-bias term in (2.31) requires a true sample of the QoI, $Q(w_{L-1}^{(n)})$, and thus is not directly computable. This thesis utilizes adjoint-based error analysis to accurately estimate the squared bias in (2.31) to use as a stopping criteria in the adaptive MLMC algorithm. The estimate for the MSE shows that the bias depends on the error of the highest level of the MLMC estimator. In order to effectively lower the bias, the estimator $\widehat{Q}_{L-1}$ would need to become more accurate, i.e. a new level would have to be introduced. This new highest level generally does not require a large number of samples [47, 48].

The MSE estimate (2.31) also shows that taking more samples on a given level (increasing some N_ℓ) decreases the overall variance of the MLMC estimator. However, taking more samples increases the cost of the estimator so a balance must be met in order to decrease the variance without increasing the cost significantly. Let the variance on level ℓ be denoted as $V_\ell = \mathbb{V}\left[\widehat{Q}_\ell - \widehat{Q}_{\ell-1}\right]$ and let the cost of taking one sample on

the ℓ-th level be C_ℓ. Using the method of Lagrange multipliers to minimize overall cost, $\sum N_\ell C_\ell$, under the constraint for the total variance $\sum \frac{V_\ell}{N_\ell} < \frac{1}{2}\epsilon$ gives the optimal number of samples to take on level ℓ to be [47]

$$N_{\ell,opt} = \left\lceil \frac{2}{\epsilon}\sqrt{\frac{V_\ell}{C_\ell}} \sum_{k=0}^{L-1} \sqrt{\frac{V_k}{C_k}} \right\rceil . \tag{2.32}$$

Note that in practice only an approximation of the variances V_ℓ are available and thus only an approximate $N_{\ell,opt}$ can be computed. If this yields an under-approximation of $N_{\ell,opt}$, the MSE of the MLMC estimator will be larger than the desired tolerance. In practice, this can be remedied by multiplying $N_{\ell,opt}$ by a factor $\delta > 1$.

Convergence of the MLMC estimator is discussed in [47] and [24]. The first Theorem in [47] guarantees convergence in MSE under certain assumptions of the convergence of expected values and variances of the levels in the MLMC estimator. In [24] the authors use the Central Limit Theorem to obtain a confidence interval which bounds the true expected value. The use of the Central Limit Theorem relies on the fact that each level is asymptotically normal and thus the overall MLMC estimator is as well. Since levels of the MLMC estimator are constructed as the *difference* of two approximations (recall (2.26)), high levels have relatively small expected values and variances. Because of this, the asymptotic approximation still applies even though few samples are taken on the higher levels of the MLMC estimator.

2 Functional Analysis Background

This section provides the background necessary for the adjoint-based analysis. We start with the definitions of useful vector and function spaces. We introduce the idea of weak derivatives and use them to define the Sobolev spaces. Finally, we define dual spaces and adjoint operators and describe how adjoint problems can be used to provide information about the solution to a differential equation.

2.1 Function Spaces

We begin the discussion on functional analysis by introducing some standard, but very important, definitions and properties. First, let X be a real vector space with norm $||a||_X$ for $a \in X$. The vector spaces that are required for the analysis of differential equations have a property that relies on Cauchy sequences.

Definition 1 (Cauchy Sequence)**.** *A sequence $\{a_n\}$ in X is a Cauchy sequence if for any ϵ there is an N such that for any $n, m > N$,*

$$||a_n - a_m||_X < \epsilon. \tag{2.33}$$

An immediate consequence from the definition is that all Cauchy sequences are bounded. A less obvious, but very important, property of Cauchy sequences is that any Cauchy sequence $\{a_n\}$ in X is convergent. Depending on the structure of the vector space X, the limit of the Cauchy sequence may or may not also belong to X. This leads to our next definition.

Definition 2 (Banach Space)**.** *A Banach (or complete) space is a vector space X such that all Cauchy sequences $\{a_n\}$ in X converge to a limit that is also in X:*

$$\lim_{n\to\infty} a_n = A \in X \quad \forall \text{ Cauchy sequence } \{a_n\} \text{ in } X. \tag{2.34}$$

A set of spaces that are of particular importance for functional analysis are the Lebesgue function spaces over a domain $\Omega \subset \mathbb{R}^d$.

Definition 3 (Lebesgue Function Spaces)**.** *The Lebesgue function spaces over $\Omega \subset \mathbb{R}^d$ are $L^p(\Omega) = \{f : f \text{ is measurable and } ||f||_p < \infty\}$, for $1 \leq p < \infty$, with norms*

$$||f||_p = \left(\int_\Omega |f|^p \mathrm{d}\Omega \right)^{1/p}, \quad 1 \leq p < \infty \tag{2.35}$$

$$||f||_\infty = esssup_\Omega |f|. \tag{2.36}$$

The Riesz–Fischer theorem shows that the $L^p(\Omega)$, for $1 \leq p \leq \infty$ are complete vector spaces. The function spaces that we will work with are not only Banach. For (weak) solutions to differential equations, we require the function space to be complete with respect to the norm induced by space's inner-product. These types of inner-product spaces are called **Hilbert spaces**. Examples of Hilbert spaces include $\mathbb{R}^d$ with the Euclidean inner-product, the Lebesgue space $L^2(\Omega)$, and Sobolev spaces $H^s(\Omega)$. Sobolev spaces are especially important in the study of solutions to differential equations. Before defining the Sobolev spaces, we introduce the concept of a weak derivative.

Definition 4 (Weak derivative)**.** *Given a function $u \in L^2(\Omega)$ and multi-index α, the α-th weak derivative of u is the function $v \in L^2(\Omega)$ such that*

$$\int_\Omega u \cdot D^\alpha \phi \mathrm{d}\Omega = (-1)^{|\alpha|} \int_\Omega v \cdot \phi \mathrm{d}\Omega, \tag{2.37}$$

for all infinitely differentiable functions ϕ with compact support in Ω, where the multi-indexed derivative is

$$D^\alpha \phi = \frac{\partial^{|\alpha|}\phi}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}. \tag{2.38}$$

For non-negative integers s and a domain $\Omega \subset \mathbb{R}^d$, the Sobolev space $H^s(\Omega)$ is the space of $L^2(\Omega)$ functions whose weak derivatives up to order s are also in $L^2(\Omega)$.

Definition 5 (L^2-based Sobolev Spaces)**.** *The Sobolev space $H^s(\Omega)$, for $s \in \mathbb{N}$, is $H^s(\Omega) = \{f \in L^2(\Omega) : D^\alpha f \in L^2(\Omega) \ \forall \ |\alpha| < s\}$.*

The inner-product for the Sobolev space $H^s(\Omega)$ is

$$(f, g)_{H^s} = \int_\Omega f \cdot g \mathrm{d}\Omega + \int_\Omega Df \cdot Dg \mathrm{d}\Omega + \cdots + \int_\Omega D^s f \cdot D^s g \mathrm{d}\Omega. \tag{2.39}$$

We often work with a specific sub-space of a Sobolev space that has the additional condition that functions and their normal derivatives vanish on the boundary of Ω. These sub-spaces are denoted as $H_0^s = \{f \in H^s(\Omega) : f = \frac{\partial f}{\partial n} = \cdots = \frac{\partial^{s-1} f}{\partial n^{s-1}} = 0 \text{ on } \partial\Omega\}$, where $\frac{\partial f}{\partial n}$ is the normal derivative on the boundary of Ω; see [37].

2.2 Dual Spaces, Adjoint Operators, and Adjoint Problems

Definition 6 (Dual Space X^*)**.** *The dual space of a vector space X is the space of all linear functionals on X. The dual space, denoted by X^* is a normed vector space with the dual (or operator) norm*

$$||\phi||_{X^*} = \sup_{\substack{x \in X \\ ||x||_X = 1}} |\phi(x)|. \tag{2.40}$$

The duality pairing $\phi(x)$ is often written in bracket notation: $\phi(x) = \langle x, \phi \rangle$. For example, consider the vector space $X = L^2(\Omega)$. Using Hölder's inequality, the dual space $(L^2(\Omega))^*$ can be identified with $L^2(\Omega)$ meaning that each function $g \in L^2(\Omega)$ is associated with a linear functional $\phi \in (L^2(\Omega))^*$ by

$$\phi(f) = \langle f, \phi \rangle = \int_\Omega g(x) \cdot f(x) \mathrm{d}x. \tag{2.41}$$

Another example of a vector space that is isometric with its dual space is the Euclidean space $\mathbb{R}^d$ with the dot-product. Moreover, by the Riesz Representation theorem, *all* Hilbert spaces are isometric with their dual space.

For two normed vector spaces X and Y, let $\mathcal{L}(X, Y)$ denote the space of linear transformations from X to Y. Each $L \in \mathcal{L}(X, Y)$ is associated with another linear transformation between the dual spaces Y^* and X^*. This other linear map, which we denote $L^* \in \mathcal{L}(Y^*, X^*)$, is called the **adjoint operator** of L. More rigorously, with $L \in \mathcal{L}(X, Y)$, for each $y^* \in Y^*$ we define a unique bounded, linear functional

$$x^*(x) := y^*(L(x)) = \langle Lx, y^* \rangle. \tag{2.42}$$

Since this x^* is a bounded linear functional on X, it is an element of the dual space $x^* \in X^*$. In this way, for every $y^* \in Y^*$ we have associated a unique $x^* \in X^*$, thus creating a linear transformation $L^* \in \mathcal{L}(Y^*, X^*)$.

Definition 7 (Adjoint operator)**.** *For a linear transformation $L \in \mathcal{L}(X, Y)$, the adjoint operator L^* is defined as the operator which satisfies the bi-linear identity*

$$\langle Lx, y^* \rangle = \langle x, L^* y^* \rangle, \tag{2.43}$$

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

For example, let $X = Y = \mathbb{R}^d$ with the usual dot-product. Any linear transformation $L \in \mathcal{L}(\mathbb{R}^d, \mathbb{R}^d)$ can be represented by multiplication with a unique $d \times d$ matrix:

$$L(x) = A_{d\times d}x. \tag{2.44}$$

Recall that $\mathbb{R}^d$ is isomorphic to its dual space. With this isomorphism, the left side of bi-linear identity (2.43) is

$$\langle Lx, y\rangle = A_{d\times d}x \cdot y. \tag{2.45}$$

Basic linear algebra allows us to rearrange the matrix multiplication by taking the transpose of $A_{d\times d}$:

$$A_{d\times d}x \cdot y = x \cdot A^{\top}_{d\times d}y. \tag{2.46}$$

Identifying the adjoint operator L^* with the transpose of the matrix $A_{d\times d}$ satisfies the bi-linear identity:

$$\langle Lx, y\rangle = A_{d\times d}x \cdot y = x \cdot A^{\top}_{d\times d}y = \langle x, L^*y\rangle. \tag{2.47}$$

Some useful properties of the adjoint operator are presented below.

- The norm of the adjoint operator is $||L^*|| = ||L||$.
- The adjoint of the null operator is $0^* = 0$.
- For $L_1, L_2 \in \mathcal{L}(X, Y)$ the adjoint of the sum is $(L_1 + L_2)^* = L_1^* + L_2^*$.
- For a scalar $a \in \mathbb{R}$, $(aL)^* = a(L^*)$.
- For $L_1 \in \mathcal{L}(X, Y)$ and $L_2 \in \mathcal{L}(Y, Z)$, the adjoint of the composition is $(L_1L_2)^* = L_2^*L_1^*$ and $(L_1L_2)^* \in \mathcal{L}(Z^*, X^*)$.
- The adjoint of the adjoint is the original operator: $(L^*)^* = L$

The error analysis used in this thesis mostly deals with Hilbert spaces like $L^2(\Omega)$, $H_0^1(\Omega)$, and $H^1(\Omega)$ where the duality pairing can be identified with the $L^2(\Omega)$ inner-product. The operators are usually linear differential operators $L = \mathcal{D}$ and the bi-linear identity can be written as

$$\int_\Omega \mathcal{D}u \cdot v \mathrm{d}x = \int_\Omega u \cdot \mathcal{D}^* v \mathrm{d}x. \tag{2.48}$$

The adjoint operators $\mathcal{D}^*$ are combined with initial or boundary values in order to yield specific adjoint problems that are required for our error analysis.

The adjoint operator $\mathcal{D}^*$ of a differential operator $\mathcal{D}$ also called the *formal adjoint operator*. The identity (2.48) is called the *bi-linear identity* with smooth functions, u and v, that have compact support inside Ω. The Hilbert space W is chosen so that $\mathcal{D}u$ and $\mathcal{D}^*v$ are well-defined. The formal adjoint is found by repeatedly using integration by parts and linear algebra (and the divergence theorem in higher dimensions) to move all derivatives and multipliers from u onto v. Since the functions are compactly supported, any boundary terms are zero.

For example, let $u, v \in H_0^2(\Omega)$ and $Du = -\nabla^2 u$. Starting with the left side of the bi-linear identity and integrating by parts gives

$$\langle \mathcal{D}u, v \rangle = \int_\Omega \left(-\nabla^2 u\right) v \mathrm{d}x = \int_\Omega \left(\nabla u\right) \left(\nabla v\right) \mathrm{d}x = \int_\Omega u \left(-\nabla^2 v\right) \mathrm{d}x = \langle u, \mathcal{D}^* v \rangle, \tag{2.49}$$

where the boundary integrals vanish due to the compact support of u and v. Thus the formal adjoint is $\mathcal{D}^* = -\nabla^2$, where we have abused the asterisk notation.

In the context of analyzing QoI that stem from problems involving differential equations, the formal adjoint will be paired with initial or boundary conditions to create an *adjoint problem*. The initial or boundary conditions for the adjoint problem depend on the initial or boundary conditions for the original problem and are chosen to ensure the boundary terms vanish.

For example, consider the IVP

$$\begin{cases} \dot{u} & = f, \quad t \in (0, T] \\ u(0) & = 0. \end{cases} \tag{2.50}$$

When computing the formal adjoint via integration by parts we get

$$\int_0^T \dot{u}v \mathrm{d}t = \int_0^T u(-\dot{v})\mathrm{d}t + u(0)v(0) - u(T)v(T) = \int_0^T u(-\dot{v})\mathrm{d}t - u(T)v(T). \tag{2.51}$$

To create the adjoint problem, we impose a condition to force $v(T) = 0$. The adjoint problem takes the form

$$\begin{cases} -\dot{v} & = \overline{f}, \quad t \in [0, T) \\ v(T) & = 0. \end{cases} \tag{2.52}$$

The function $\overline{f}$ is chosen based off the original function f and the QoI that is being analyzed. In practice, the condition on v does not have to be homogeneous and again is chosen to help analyze the QoI at hand. The QoI and adjoint problems associated with our three differential equations are presented later in §2.4 and in Chapter 3.

The next section gives a brief discussion on the weak form of a differential equation and presents the weak forms of (1.1), (1.4), and (1.6). Galerkin methods for numerically solving variational differential equations are also presented.

3 Variational Forms and Suitable Numerical Methods

Galerkin finite element methods (FEMs) utilize the variational form of a differential equation to obtain numerical solutions that are piece-wise polynomials. The error analysis used in this thesis is also based on the variational form of a differential equation and thus it is well-suited for solutions obtained via a Galerkin FEM. However, the error

analysis is not limited to these FEMs since many other numerical methods can be identified with a Galerkin FEM using a certain integration scheme. This section first details the variational forms of our three model equations (1.1), (1.4), and (1.6). We then present the continuous and discontinuous Galerkin FEMs for the equations. We also present a few other numerical methods and show how to identify them with certain Galerkin methods.

3.1 Variational forms of differential equations

Variational, or weak, forms of differential equations give a generalized view of the equations and often allow for relaxed requirements on the differentiability of solutions. A weak form is created so that the solution behaves similarly to solution of the original differential equation under the inner-product with certain test functions. As an illustrative example, consider the differential equation

$$-\nabla^2 u = f, \qquad x \in \Omega. \tag{2.53}$$

In order to satisfy (2.53), the function u must be twice differentiable. Take (2.53), multiply by a continuously differentiable function v with compact support in Ω and use integration by parts on the left. This yields

$$\int_\Omega \nabla u \cdot \nabla v \mathrm{d}x = \int_\Omega f v \mathrm{d}x. \tag{2.54}$$

In order to satisfy (2.54), the function u only requires a single (weak) derivative. To be a *weak solution* of the differential equation (2.53), we require a weakly differentiable function $u \in H^1(\Omega)$ that satisfies (2.54) for all test functions $v \in H_0^1(\Omega)$. Note that for the variational form, we do not remove all of the derivatives from u. The weak form is considered a generalization of the strong form because if a solution to (2.54) is sufficiently differentiable, it will also satisfy (2.53).

In this section, we present the Lax-Milgram theorem for existence and uniqueness

of solutions to weak-forms of differential equations. We then give the weak forms of our three model equations (1.1), (1.4), and (1.6).

Existence and Uniqueness of Weak Solution

The existence of a unique solution to the weak form of a differential equation comes from a form of the Lax-Milgram Theorem. To present this theorem, first let $a(u, v)$ be the bi-linear form that represents the left side of any of the below weak forms; i.e. $a(u, v) = \int_0^T \dot{u} \cdot v \mathrm{d}t$ or $a(u, v) = \int_\Omega \mathcal{D}_1 u \cdot \mathcal{D}_2 v \mathrm{d}x$ or $a(u, v) = \int_\Omega \dot{u} \cdot v \mathrm{d}x + \int_\Omega \mathcal{D}_1 u \cdot \mathcal{D}_2 v \mathrm{d}x$. We also present two definitions related to a bi-linear form.

Definition 8 (Bounded bi-linear form)**.** *A bi-linear form $a(u, v)$ over a Hilbert space W is bounded if there exists a $C \in \mathbb{R}$ such that $|a(u, v)| \leq C||u||_W||v||_W$ for all $u, v \in W$.*

Definition 9 (Coercive bi-linear form)**.** *A bi-linear form $a(u, v)$ over a Hilbert space W is coercive if there exists a $c \in \mathbb{R}$ such that $|a(u, u)| \leq c||u||_W^2$ for all $u \in W$.*

With these two definitions we now present the Lax-Milgram Theorem as applied to the variational forms of our differential equations.

Theorem 1 (Lax-Milgram)**.** *If the bi-linear form $a(u, v)$ over a Hilbert space W is both bounded and coercive, then the variational equation*

$$a(u, v) = \int_\Omega f \cdot v \mathrm{d}x \quad \forall v \in W, \tag{2.55}$$

has a unique solution $u \in W$, for all $f \in W^*$.

Variational form: IVPs

Recall the differential equation used in the model IVP (1.1) is

$$\dot{u} = f, \quad t \in (0, T], \tag{2.56}$$

where $u = u(t)$ and $f = f(u, t; w)$ are vector functions. Since this is a first order differential equation, we do not perform any integration by parts. The variational form of (2.56) is: Find $u \in (H^1([0,T]))^d$ such that

$$\int_0^T \dot{u} \cdot v \mathrm{d}t = \int_0^T f \cdot v \mathrm{d}t, \quad \forall v \in \left(L^2[0,T]\right)^d. \tag{2.57}$$

Variational form: BVPs

The differential equation from the model BVP (1.4) is

$$\mathcal{D}u = f, \quad x \in \Omega, \tag{2.58}$$

where $\mathcal{D}$ is a differential operator of arbitrary order p. Let W^q be a Hilbert space that allows derivatives of up to order q. The weak form of (2.58) is: Find $u \in W^q$ such that

$$\int_\Omega \mathcal{D}_1 u \cdot \mathcal{D}_2 v \mathrm{d}x = \int_\Omega f \cdot v \mathrm{d}x, \qquad \forall v \in W^{p-q}. \tag{2.59}$$

The operators $\mathcal{D}_1, \mathcal{D}_2$ are linear differential operators, such that $\mathcal{D}_2^* \mathcal{D}_1 u = \mathcal{D}u$. The operator $\mathcal{D}_2^*$ is the formal adjoint of $\mathcal{D}_2$ which satisfies the property (2.48). The differential operator $\mathcal{D}_1$ has derivatives up to order q which may be of lower order than $\mathcal{D}$.

Variational form: IBVPs

The differential equation from the model IBVP (1.6) is

$$\dot{u} + \mathcal{D}u = f, \quad x \in \Omega, \quad t \in (0,T]. \tag{2.60}$$

Before presenting the weak form of (2.60), we must define an appropriate solution space over the space-time domain $\Omega \times (0,T]$. Given a Hilbert space W over Ω, let $L^2(0,T;W)$ denote the space of functions $u(x,t)$ such that for every fixed value of $t = \mathbf{t}$, we have

$u(x) = u(x, \mathbf{t}) \in W$. Now, the variational form of (2.60) is: Find $u \in L^2(0, T; W^q)$ such that

$$\int_\Omega \dot{u} \cdot v \mathrm{d}x + \int_\Omega \mathcal{D}_1 u \cdot \mathcal{D}_2 v \mathrm{d}x = \int_\Omega f \cdot v \mathrm{d}x, \qquad \forall v \in W^{p-q} \qquad \forall t \in (0, T]. \quad (2.61)$$

The operators $\mathcal{D}_1, \mathcal{D}_2$ are similar to those in (2.59).

3.2 Continuous Galerkin Methods

Continuous Galerkin finite element methods are constructed using the standard continuous Lagrangian finite element spaces. In this section we describe the continuous Galerkin methods for the three model equations (1.1), (1.4), and (1.6). To define the methods, we first define the Lagrange spaces which act as our solution spaces. The spaces for IVPs and BVPs are similar while the spaces for IBVPs require a different formulation to deal with the separated time and space domains.

Galerkin Methods: IVPs

We first define the Lagrange space over the time domain $[0, T]$. Let $\mathcal{T}_k$ be a partition of $[0, T]$ into N_t sub-intervals with endpoints $\{t_0, t_1, \ldots, t_{N_t}\}$ such that $t_{n+1} - t_n \leq k$ for $n = 0, 1, \ldots, N_t - 1$. The degree q continuous Lagrange finite element space is defined as

$$\mathcal{P}_k^q = \left\{ v \in C([0, T]) : \forall (t_n, t_{n+1}) \in \mathcal{T}_k,\, v|_{(t_n, t_{n+1})} \in \mathcal{P}^q(t_n, t_{n+1}) \right\}, \quad (2.62)$$

where $\mathcal{P}^q(t_n, t_{n+1})$ is the space of polynomials of degree at most q defined on the element (t_n, t_{n+1}). The continuous Galerkin finite element method of degree q, denoted cG(q), for solving (1.1) is defined interval-wise by: Find $U \in \mathcal{P}_k^q$ such that the restriction of U to any sub-interval $(t_n, t_{n+1}) \in \mathcal{T}_k$ satisfies

$$\int_{t_n}^{t_{n+1}} \dot{U}(t) \cdot v(t)\, \mathrm{d}t = \int_{t_n}^{t_{n+1}} f(U, t) \cdot v(t) dt, \quad \forall v \in \mathcal{P}^{q-1}(t_n, t_{n+1}), \quad (2.63)$$

for $n = 0, 1, 2, ..., N_t - 1$.

Galerkin Methods: BVPs

The Lagrange space over the spatial domain Ω is similar to $\mathcal{P}_k^q$. Let $\mathcal{T}_h$ be a simplicial decomposition of the spatial domain Ω, where h denotes the maximum diameter of the elements of $\mathcal{T}_h$. Specifically, $\bigcup_{\tau\in\mathcal{T}_h} \tau = \Omega$ and the intersection of any two elements is either a common edge, a node, or is empty. The degree q continuous Lagrange finite element space is then defined as

$$\mathcal{P}_h^q = \left\{v \in C(\Omega) : \forall \tau \in \mathcal{T}_h,\, v|_\tau \in \mathcal{P}^q(\tau)\right\}, \tag{2.64}$$

The degree q continuous Galerkin finite element method, with respect to $\mathcal{T}_h$, for the PDE (1.4) is: Find $U \in \mathcal{P}_h^q$ such that

$$\int_\Omega \mathcal{D}_1 U \cdot \mathcal{D}_2 v \mathrm{d}x = \int_\Omega f \cdot v \mathrm{d}x, \qquad \forall v \in \mathcal{P}_h^q. \tag{2.65}$$

Galerkin Methods: IBVPs

Finally, we define the Lagrange space of the space-time domain $\Omega \times [0,T]$. We use the spatial decomposition $\mathcal{T}_h$ and the temporal partition $\mathcal{T}_k$ to define space-time slabs over which we form the Galerkin method. The *space-time slabs* with respect to $\mathcal{T}_h$ and $\mathcal{T}_k$ are $\{s_n = \mathcal{T}_h \times (t_n, t_{n+1})\}_{n=0}^{N_t-1}$, where (t_n, t_{n+1}) are the sub-intervals of $\mathcal{T}_k$. For $n = 0, \ldots, N_t - 1$, let $\{l_{n,j}\}_{j=0}^{q_2}$ be the q_2+1 Lagrange basis polynomials of degree q_2 over the time interval (t_n, t_{n+1}). The Lagrange space over the space-time domain $\Omega \times [0,T]$ is defined over each space-time slab s_n as

$$P_n^{q_1,q_2} = \left\{ v : v(x,t) = \sum_{j=0}^{q_1} l_{n,j}(t) w_j(x), \quad \text{where } w_j(x) \in \mathcal{P}_h^{q_1}, \quad (x,t) \in s_n \right\}. \tag{2.66}$$

For any fixed time t, functions in $P_n^{q_1,q_2}$ are piece-wise polynomials of degree q_1 in space, with respect to $\mathcal{T}_h$. Similarly, for fixed spatial coordinate x, functions in $P_n^{q_1,q_2}$ are piece-wise polynomials of degree q_2 in time, with respect to $\mathcal{T}_k$.

The continuous Galerkin finite element method of degree q_1 in space and q_2 in time, with respect to the spatial decomposition $\mathcal{T}_h$ and the temporal decomposition $\mathcal{T}_k$ is: Find $U \in C(\Omega \times [0,T])$ such that its restriction to any space-time slab s_n is $U|_{s_n} \in P_n^{q_1,q_2}$ and satisfies

$$\int_{t_n}^{t_n+1} \left[\int_\Omega \dot{U} \cdot v \mathrm{d}x + \int_\Omega \mathcal{D}_1 U \cdot \mathcal{D}_2 v \mathrm{d}x \right] \mathrm{d}t = \int_{t_n}^{t_n+1} \int_\Omega f \cdot v \mathrm{d}x \mathrm{d}t, \quad \forall v \in P_n^{q_1,q_2-1}, \tag{2.67}$$

for $n = 0, 1, \ldots, N_t - 1$.

3.3 Other methods nodally equivalent to Galerkin methods

We show how two finite difference methods are identified with certain Galerkin methods. In particular, the Backward Euler method can be identified with the discontinuous Galerkin method of order zero and the Crank-Nicolson method can be identified with the continuous Galerkin method of order one. We provide the work in the one-dimensional case, but this identification still holds in higher-dimensions.

Backward Euler

Given an interval $[t_n, t_{n+1}]$ the Backward Euler finite difference method is defined by the equation

$$u(t_{n+1}) - u(t_n) = (t_{n+1} - t_n) f(u(t_{n+1}), t_{n+1}). \tag{2.68}$$

Theorem 2. *Numerical solutions obtained via the Backward Euler scheme are nodally equivalent to solutions obtained using a dG(0) finite element method in which the integrals are evaluated with the right-hand rectangle rule.*

Proof. The dG(0) formulation over a sub-interval (t_n, t_{n+1}), with the constant test function $v(t) = 1$ is

$$\int_{t_n}^{t_{n+1}} \dot{u} \, \mathrm{d}t = \int_{t_n}^{t_{n+1}} f(u,t) \, \mathrm{d}t. \tag{2.69}$$

Where, by the fundamental theorem of calculus,

$$\int_{t_n}^{t_{n+1}} \dot{u} \, \mathrm{d}t = u(t_{n+1}) - u(t_n). \tag{2.70}$$

Using the right-hand rectangle rule, we obtain

$$\int_{t_n}^{t_{n+1}} f(u,t) \, \mathrm{d}t \approx (t_{n+1} - t_n) f(u(t_{n+1}), t_{n+1}). \tag{2.71}$$

Substituting (2.70) and (2.71) into (2.69) results in the Backward Euler scheme. □

Crank-Nicolson

Given an interval $[t_n, t_{n+1}]$ the Crank-Nicolson finite difference method is defined by the equation

$$u(t_{n+1}) - u(t_n) = \frac{t_{n+1} - t_n}{2}(f(u(t_{n+1}), t_{n+1}) + f(u(t_n), t_n)). \tag{2.72}$$

Theorem 3. *Numerical solutions obtained via the Crank-Nicolson finite difference scheme are nodally equivalent to solutions obtained using a cG(1) finite element method in which the integrals are evaluated with the trapezoidal rule.*

Proof. The cG(1) formulation over a sub-interval (t_n, t_{n+1}), with the constant test function $v(t) = 1$ is

$$\int_{t_n}^{t_{n+1}} \dot{u} \, \mathrm{d}t = \int_{t_n}^{t_{n+1}} f(u,t) \, \mathrm{d}t. \tag{2.73}$$

Where, by the fundamental theorem of calculus,

$$\int_{t_n}^{t_{n+1}} \dot{u} \, \mathrm{d}t = u(t_{n+1}) - u(t_n). \tag{2.74}$$

Using the trapezoidal quadrature rule, we obtain

$$\int_{t_n}^{t_{n+1}} f(u,t) \, \mathrm{d}t \approx \frac{t_{n+1} - t_n}{2}(f(u(t_{n+1}), t_{n+1}) + f(u(t_n), t_n)). \tag{2.75}$$

Substituting (2.74) and (2.75) into (2.73) results in the Crank-Nicolson scheme. □

Many other numerical schemes have been shown to be identifiable with a Galerkin method. This includes the Lax-Wendroff finite difference method, Picard Iteration, and many implicit-explicit (IMEX) schemes [27, 29, 30, 35].

4 Review of Classical a Posteriori Analysis

We present classical adjoint-based error analysis for standard QoIs. Recall that a "standard QoI" is one that can be represented, or nicely approximated by, a linear functional of the solution to the differential equation. Classical adjoint-based analysis provides representations of the error in a standard QoI and also details the necessary adjoint problems in order to accurately estimate the error representations.

Throughout this section, let u be a solution to a differential equation and U be a numerically computed solution. The particular differential equation will be obvious from context. Since a standard QoI can be represented as a bounded linear functional of the solution to the differential equation, by the Riesz Representation Theorem, these QoI can be generically written as

$$Q(u) = (\psi, u)\,, \tag{2.76}$$

where ψ is some weight function and (u, v) is an inner-product. Specific forms of the standard QoI, with explicitly chosen inner-product, are detailed in this section along with error representations and necessary adjoint problems. The error representations and adjoint problems are different for the three differential equations (1.1), (1.4), and (1.6). As such, this section is divided into discussions detailing the analysis for each differential equation.

While the main goal of this thesis is to develop UQ for differential equations (1.1), (1.4), and (1.6) (and their QoIs) that depend on a random parameter w, the error analysis is applied only to solutions involving *an individual sample* of the random parameter. As such, the error analysis need not be modified to be useful for our UQ. For clarity of

notation, dependence on the random parameter w is omitted in this section. Numerical experiments demonstrating the accuracy of the error estimates are presented throughout the section. In these experiments, the accuracy of an error estimate, $\eta \approx Q(u) - Q(U)$, is measured by the *effectivity ratio*

$$\rho_{eff} = \frac{\eta}{Q(u) - Q(U)}. \tag{2.77}$$

Ideally, η is close to the true error $Q(u) - Q(U)$ and thus the effectivity ratio ρ_{eff} is close to one.

4.1 Classical Analysis: IVPs

For IVPs involving ODEs of form (1.1) we discuss two types of standard QoI. In this section we first define the standard QoIs for IVPs involving ODEs. Then we derive the representations of the error in a computed QoI and the corresponding adjoint problems. The error representations also provide a way to decompose the error into contributions from sub-intervals of the domain.

IVPs: Standard QoI Type 1

The first type of QoI for our IVPs represents a linear-combination of the components of the solution u evaluated at a particular time value $t = t^\star$:

$$Q(u) = \psi \cdot u(t^\star), \tag{2.78}$$

for some $\psi \in \mathbb{R}^d$, with the Euclidean inner-product (a.k.a dot-product). For example, the QoI (2.78) could represent the value of the first component of u evaluated at the final time $t^\star = T$. In this example, $\psi \in \mathbb{R}^d$ is $\psi = (1, 0, \ldots, 0)^\top$ and the QoI is $Q(u) = \psi \cdot u(t^\star) = u_1(T)$. The error in a computed QoI, $Q(U)$, of form (2.78) is given in Theorem 4.

Theorem 4 (Error representation of QoI type 1 for IVPs)**.** *Given a finite element solution U(t) of* (1.1) *and* $\psi \in \mathbb{R}^d$*, let* $e(t) = u(t) - U(t)$*. The error* $\psi \cdot e(t^\star)$ *in the QoI* (2.78) *at time* $t^\star \in (0, T]$ *is represented as*

$$\psi \cdot e(t^\star) = \psi \cdot u(t^\star) - \psi \cdot U(t^\star) = \int_0^{t^\star} \phi \cdot [f(U,t) - \dot{U}]\mathrm{d}t, \tag{2.79}$$

where ϕ *is the solution to the adjoint equation*

$$\begin{cases} -\dot{\phi} = \overline{f_{u,U}(t)}^\top \phi, \ t \in [0, t^\star), \\ \phi(t^\star) = \psi, \end{cases} \tag{2.80}$$

with

$$\overline{f_{u,U}(t)} = \int_0^1 \nabla_z f(z,t)\mathrm{d}s \tag{2.81}$$

and $z = su + (1-s)U$*.*

The function $\overline{f_{u,U}(t)}$ has the property that $\overline{f_{u,U}(t)}e(t) = f(u,t) - f(U,t)$. Since $\overline{f_{u,U}(t)}$ requires the true solution u, in practice it is approximated by

$$\overline{f_{u,U}(t)} \approx \nabla_u f(U,t). \tag{2.82}$$

Proof. Take the adjoint equation (2.80), multiply by the error function $e(t)$, and integrate over the interval $(0, t^\star)$,

$$\int_0^{t^\star} -\dot{\phi}(t) \cdot e(t)\mathrm{d}t = \int_0^{t^\star} \overline{f_{u,U}(t)}^\top \phi \cdot e(t)\mathrm{d}t. \tag{2.83}$$

Using basic linear algebra and the definition of $\overline{f_{u,U}(t)}$, the right side of (2.83) becomes

$$\int_0^{t^\star} \overline{f_{u,U}(t)}^\top \phi \cdot e(t)\mathrm{d}t = \int_0^{t^\star} \phi \cdot \overline{f_{u,U}(t)}e(t)\mathrm{d}t = \int_0^{t^\star} \phi \cdot [f(u,t) - f(U,t)]\,\mathrm{d}t. \tag{2.84}$$

With integration by parts, the left side of (2.83) is

$$\int_0^{t^\star} -\dot{\phi}(t) \cdot e(t)\mathrm{d}t = \int_0^{t^\star} \phi(t) \cdot \dot{e}(t)\mathrm{d}t - \phi(t^\star) \cdot e(t^\star) + \phi(0) \cdot e(0). \tag{2.85}$$

From the initial condition of the IVP (1.1), $u(0) = u_0 = U(0)$, so $e(0) = 0$. From the condition for the adjoint IVP (2.80), $\phi(t^\star) = \psi$. Putting these into (2.83) and rearranging gives

$$\psi \cdot e(t^\star) = \int_0^{t^\star} \phi(t) \cdot \dot{e}(t) \mathrm{d}t - \int_0^{t^\star} \phi \cdot [f(u,t) - f(U,t)] \, \mathrm{d}t, \tag{2.86}$$

$$= \int_0^{t^\star} \phi \cdot [\dot{e}(t) - f(u,t) + f(U,t)] \, \mathrm{d}t, \tag{2.87}$$

$$= \int_0^{t^\star} \phi \cdot \left[-\dot{U}(t) + f(U,t) \right] \mathrm{d}t, \tag{2.88}$$

where the final line comes from the fact that $e(t) = u(t) - U(t)$ and $\dot{u}(t) = f(u,t)$. □

Taking advantage of the linearity of the integral in (2.79) gives a decomposition of the error. If the domain of integration $(0, t^\star)$ is partitioned into $N^\star$ sub-intervals with endpoints $\{t_0, t_1, \ldots, t_{N^\star}\}$, a decomposition of the error (2.79) is

$$e(t^\star) \cdot \psi = \sum_{i=0}^{N^\star - 1} \int_{t_i}^{t_{i+1}} \left[f \cdot \phi - \dot{U} \cdot \phi \right] \mathrm{d}t = \sum_{i=0}^{N^\star - 1} e_{(t_i, t_{i+1})}, \tag{2.89}$$

with error contributions, $e_{(t_i,t_{i+1})}$, given by the integrals

$$e_{(t_i,t_{i+1})} = \int_{t_i}^{t_{i+1}} \left[f \cdot \phi - \dot{U} \cdot \phi \right] \mathrm{d}t, \quad \text{for } i = 0, 1, \ldots, N^\star - 1. \tag{2.90}$$

The error contributions allow us to see how the error behaves over the time-interval $(0, t^\star)$ and is used in the adaptive refinement methods presented later in §4.2.1.

In practice, the adjoint solution ϕ must be approximated. The adjoint problem is always linear and can be accurately approximated with an appropriate Galerkin method. Often, the adjoint problem will either be solved over the same grid as the IVP using a higher order Galerkin method, or the adjoint will be solved with the same Galerkin method over a finer mesh. This is done so that any error associated to the adjoint solution is negligible compared to other sources of error. In the case of a non-linear IVP, we also use the approximation $\overline{f_{u,U}} \approx \nabla_u f(U,t)$. The accuracy of this approximation correlates to the accuracy of the numerical solution U. We provide two

numerical experiments (one linear and one non-linear) to demonstrate the accuracy of the error representation (2.79).

Numerical Experiment 1: IVP with standard QoI Type 1

Consider the damped harmonic oscillator

$$\ddot{\omega} = -\frac{k}{m}\omega - \frac{c}{m}\dot{\omega} + \frac{F_0}{m}\cos(\gamma t + \theta_d),\ t \in (0,3], \quad \omega(0) = 5,\ \dot{\omega}(0) = 0. \tag{2.91}$$

with

$$k = 50,\ m = 0.25,\ c = 1,\ F_0 = 50,\ \theta_d = 0,\ \gamma = 10. \tag{2.92}$$

Rewriting as a system of first-order ODEs, $\dot{u} + Au = h(t)$, gives

$$\begin{pmatrix} \dot{u_1}(t) \\ \dot{u_2}(t) \end{pmatrix} + \begin{pmatrix} 0 & -1 \\ 200 & 4 \end{pmatrix} \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = \begin{pmatrix} 0 \\ 200\cos(10t) \end{pmatrix}. \tag{2.93}$$

The type 1 QoI (2.78) represents the position of the oscillator at the final time $t = 3$:

$$Q(u) = \psi \cdot u(3) = (1,0)^\top \cdot u(3) = u_1(3). \tag{2.94}$$

We obtain a numerical solution $U(t)$ using the cG(1) method for IVPs over a uniform partition with $N_t = 30$ sub-intervals. We also obtain a highly-accurate reference solution, using the cG(3) method for IVPs over a uniform partition with 600 sub-intervals, in order to check the accuracy of the error estimate.

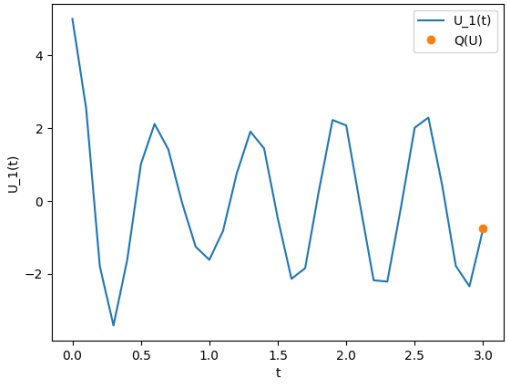
The adjoint-based error representation given by (2.79), requiring an adjoint problem in the form (2.80). Since the equation of the harmonic oscillator is linear, the function on the RHS of the adjoint problem is the constant matrix $\overline{f}(u,U) = A$. Thus the adjoint problem we need is

$$\begin{cases} -\dot{\phi} = -A^\top \phi,\ t \in [0, t^\star), \\ \phi(t^\star) = (1,0)^\top. \end{cases} \tag{2.95}$$

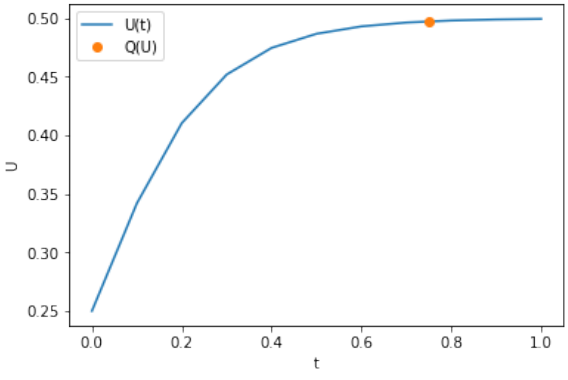
The adjoint problem is solved, backwards in time starting at $t = t^\star$, using the cG(3) method for IVPs over a uniform partition with 30 sub-intervals. Note that this is the same partition that we used to obtain the numerical solution U, but the degree is higher. The computed QoI $Q(U)$, reference QoI $Q(u)$, error estimate η from (2.79), reference error $Q(u) - Q(U)$, and effectivity ratio ρ_{eff} are all shown in Table 2.1. We see that the effectivity ratio is very close to one, indicating an accurate error estimate. The first component of the numerical solution $U_1(t)$ and the computed QoI $Q(U)$ are shown in figure 2.1a.

$Q(U)$	$Q(u)$	η	$Q(u) - Q(U)$	ρ_{eff}
-0.758799	-0.418642	0.340109	0.340157	0.99985

Table 2.1: Results for Numerical example 1 of an IVP with QoI type 1.



(a) Numerical solution and standard QoI for Harmonic oscillator in §2.4.1.



(b) Numerical solution and standard QoI for Harmonic oscillator in §2.4.1.

Figure 2.1

Numerical Experiment 2: IVP with standard QoI Type 1

Next we consider the nonlinear IVP

$$\dot{u}(t) = \sin(2\pi u(t)),\ t \in (0, 1], \quad u(0) = \frac{1}{4}.$$

The analytic solution to this problem is

$$u(t) = \frac{1}{\pi}\arctan(e^{2\pi t}).$$

The QoI chosen here represents the value of the solution at a time, $t^\star = 0.75$, part-way through the domain: $Q(u) = \psi u(0.75) = (1)u(0.75) = u(0.75)$. A numerical solution is obtained using the Crank-Nicolson method (which is nodally equivalent to the cG(1) method with trapezoid rule) over a partition with $N_t = 10$ sub-intervals. To create the adjoint problem, we approximate the function $\overline{f}$ in (2.80) by the derivative $\overline{f}(u, U) \approx \frac{\partial f}{\partial u}(U) = 2\pi\cos(2\pi U(t))$. The condition for the adjoint is posed at $t = 0.75$, and the problem is solved backwards in time using the cG(3) method over a partition with 10 sub-intervals. Results are presented in Table 2.2. The effectivity ratio is again close to one, meaning the estimate is accurate. For this nonlinear problem, the estimate is *slightly* less accurate than the previous linear example due to the approximation of $\overline{f}(u, U)$. The first component of the numerical solution $U_1(t)$ and the computed QoI $Q(U)$ are shown in figure 2.1b.

$Q(U)$	$Q(u)$	η	$Q(u) - Q(U)$	ρ_{eff}
0.497259	0.497140	-0.000118	-0.000119	0.9893

Table 2.2: Results for Numerical example 2 of an IVP with QoI type 1.

IVPs: Standard QoI Type 2

The second type of QoI represents a weighted integral of the solution u, often over a sub-interval inside the domain:

$$Q(u) = \int_0^T \psi(t) \cdot u(t)\mathrm{d}t, \tag{2.96}$$

for some $\psi \in \mathbb{R}^d$ with the $L^2([0, T])$ inner-product. For example, if $d = 1$ the QoI (2.96) could represent the average value of the solution over an internal sub-interval (t_α, t_β) and the weight function $\psi(t)$ would equal 1 over the sub-interval and 0 everywhere else. The error in an approximated QoI, $Q(U)$, of form (2.96) is given in Theorem 5.

Theorem 5 (Error representation of QoI type 2 for IVPs)**.** *Given a finite element solution U(t) of* (1.1) *and* $\psi \in L^2([0,T])^d$*, let* $e(t) = u(t) - U(t)$*. The error* $\int_0^T \psi(t) \cdot e(t)\mathrm{d}t$ *in the QoI* (2.96) *is represented as*

$$\int_0^T \psi(t)\cdot e(t)\mathrm{d}t = \int_0^T \psi(t)\cdot u(t)\mathrm{d}t - \int_0^T \psi(t)\cdot U(t)\mathrm{d}t = \int_0^T \phi\cdot[f(U,t)-\dot{U}(t)]\mathrm{d}t, \quad (2.97)$$

where ϕ *is the solution to the adjoint equation*

$$\begin{cases} -\dot{\phi} = \overline{f_{u,U}(t)}^\top \phi + \psi(t), \ t \in [0,T), \\ \phi(T) = 0. \end{cases} \quad (2.98)$$

The function $\overline{f_{u,U}(t)}$ is the same as in Theorem 4. The error representation (2.97) is nearly identical to the error representation for QoI type 1 (2.79) with the only changes being the domain of integration and the associated adjoint problem. The proof of Theorem 4 follows the same steps as the previous Theorem.

Proof. Take the adjoint equation (2.98), multiply by the error function $e(t)$, and integrate over the domain $(0,T)$:

$$\int_0^T -\dot{\phi}(t)\cdot e(t)\mathrm{d}t = \int_0^T \overline{f_{u,U}(t)}^\top \phi \cdot e(t)\mathrm{d}t + \int_0^T \psi(t)\cdot e(t)\mathrm{d}t. \quad (2.99)$$

Rearrange to isolate the term we are interested in and use integration by parts to get

$$\int_0^T \psi(t)\cdot e(t)\mathrm{d}t = \int_0^T \phi(t)\cdot\dot{e}(t)\mathrm{d}t - \phi(T)\cdot e(T) + \phi(0)\cdot e(0) - \int_0^T \phi(t)\cdot\overline{f_{u,U}(t)}e(t)\mathrm{d}t. \quad (2.100)$$

From the initial condition of the IVP (1.1), $u(0) = u_0 = U(0)$, so $e(0) = 0$. From the condition for the adjoint IVP (2.98), $\phi(T) = 0$. Putting these into (2.100) and expanding $e(t) = u(t) - U(t)$ yields

$$\begin{aligned} \int_0^T \psi(t)\cdot e(t)\mathrm{d}t &= \int_0^T \phi(t)\cdot(\dot{u}(t)-\dot{U}(t))\mathrm{d}t - \int_0^T \phi(t)\cdot(f(u,t)-f(U,t))\mathrm{d}t \\ &= \int_0^T \phi(t)\cdot(f(U,t)-\dot{U}(t))\mathrm{d}t. \end{aligned} \quad (2.101)$$

where the cancellation to get the final line comes from the IVP. □

The error representation (2.97) can be decomposed in the exact same way as the type 1 QoI for IVP. Again, let the domain of integration $(0, T)$ be partitioned into N_t sub-intervals with endpoints $\{t_0, t_1, \ldots, t_{N_t}\}$, a decomposition of the error (2.97) is

$$\int_0^T \psi(t) \cdot e(t) \mathrm{d}t = \sum_{i=0}^{N_t-1} \int_0^T \phi(t) \cdot (f(U,t) - \dot{U}(t)) \mathrm{d}t = \sum_{i=0}^{N_t-1} e_{(t_i,t_{i+1})}, \tag{2.102}$$

with error contributions, $e_{(t_i,t_{i+1})}$, given by the integrals

$$e_{(t_i,t_{i+1})} = \int_{t_i}^{t_{i+1}} \left[f \cdot \phi - \dot{U} \cdot \phi \right] \mathrm{d}t, \quad \text{for } i = 0, 1, \ldots, N_t - 1. \tag{2.103}$$

4.2 Classical Analysis: BVPs

The standard QoI for BVPs also represents a weighted integral of the solution u:

$$Q(u) = \int_\Omega \psi(x) \cdot u(x) \mathrm{d}x, \tag{2.104}$$

for some $\psi(x) \in C(\Omega)$ with compact support in Ω. This QoI often represents the weighted average of the solution of a sub-domain inside of Ω. The error representation for a computed QoI of form (2.104) is given in Theorem 6 and requires a numerical solution to the weak form of the BVP (2.59).

Theorem 6 (Error representation of QoI for BVPs)**.** *Given a finite element solution $U(t)$ of (2.59) and $\psi \in C(\Omega)$ such that $\psi(x) = 0$ for $x \in \partial\Omega$, let $e(x) = u(x) - U(x)$. The error $\int_\Omega \psi(x) \cdot e(x) \mathrm{d}x$ in the QoI (2.104) is represented as*

$$\int_\Omega \psi(x) \cdot e(x) \mathrm{d}x = \int_\Omega \left([\mathcal{D}_1 U(x) \cdot \mathcal{D}_2 \phi(x)] - f(U,x) \cdot \phi(x) \right) \mathrm{d}x, \tag{2.105}$$

where $\phi(x)$ is the solution to the adjoint equation

$$\begin{cases} \mathcal{D}^* \phi &= \overline{f_{u,U}(x)}^\top \phi + \psi(x), \quad x \in \Omega, \\ \phi(x) &= 0, \qquad\qquad\qquad\quad x \in \partial\Omega. \end{cases} \tag{2.106}$$

with

$$\overline{f_{u,U}(x)} = \int_0^1 \nabla_z f(z,x) \mathrm{d}s \tag{2.107}$$

and $z = su + (1-s)U$.

The function $\overline{f_{u,U}(x)}$ is the analog of (2.81) used in previous theorems. This function has the property that $\overline{f_{u,U}(x)}e(x) = f(u,x) - f(U,x)$. Since $\overline{f_{u,U}(x)}$ requires the true solution u, it is often approximated by

$$\overline{f_{u,U}(x)} \approx \nabla_u f(U,x). \tag{2.108}$$

The operator $\mathcal{D}^*$ in (2.106) is the formal adjoint of $\mathcal{D}$ from the BVP (1.4). The two operators $\mathcal{D}_1$ and $\mathcal{D}_2$ in (2.105) are such that

$$\int_\Omega \mathcal{D}_1 u \cdot \mathcal{D}_2 v \mathrm{d}x = \int_\Omega \mathcal{D}_2^* \mathcal{D}_1 u \cdot v \mathrm{d}x = \int_\Omega \mathcal{D}u \cdot v \mathrm{d}x, \tag{2.109}$$

for any appropriate u and v. In this way we can write $\mathcal{D}u = \mathcal{D}_2^*\mathcal{D}_1 u$ where equality is meant in the weak sense, i.e. in the sense of (2.109). Using the property of the adjoint of a composition, we also have $\mathcal{D}^* = \mathcal{D}_1^*\mathcal{D}_2$.

The proof of Theorem 6 is again similar to those of the previous error representations, this time utilizing the formal adjoint (which comes from integration by parts and the Divergence Theorem). We also invoke the weak form (2.59) of the BVP rather than the strong form.

Proof. Multiply the adjoint equation (2.106) by the error $e(x)$ and integrate over the domain Ω:

$$\int_\Omega \mathcal{D}^*\phi(x) \cdot e(x) \mathrm{d}x = \int_\Omega \overline{f_{u,U}(x)}^\top \phi(x) \cdot e(x) \mathrm{d}x + \int_\Omega \psi(x) \cdot e(x) \mathrm{d}x. \tag{2.110}$$

Rearrange to isolate the desired term and using properties of formal adjoints we get

$$\int_\Omega \psi(x) \cdot e(x) \mathrm{d}x = -\int_\Omega \mathcal{D}^*\phi(x) \cdot e(x) \mathrm{d}x + \int_\Omega \overline{f_{u,U}(x)}^\top \phi(x) \cdot e(x) \mathrm{d}x, \tag{2.111}$$

$$= -\int_\Omega \mathcal{D}_2\phi(x) \cdot \mathcal{D}_1 e(x) \mathrm{d}x + \int_\Omega \phi(x) \cdot \overline{f_{u,U}(x)} e(x) \mathrm{d}x. \tag{2.112}$$

Expand $e(x)$ on the right, use the property (2.107), and apply the weak form (2.59).

$$\int_\Omega \psi(x) \cdot e(x) \mathrm{d}x = -\int_\Omega \mathcal{D}_2\phi(x) \cdot \mathcal{D}_1(u(x) - U(x)) \mathrm{d}x + \int_\Omega \phi(x) \cdot (f(u,x) - f(U,x)) \mathrm{d}x,$$

$$= \int_\Omega \left(\mathcal{D}_2\phi(x) \cdot \mathcal{D}_1 U(x) - \phi(x) \cdot f(U,x)\right) \mathrm{d}x.$$

□

We take advantage of the linearity of the integral with respect to domain to decompose the error. Let $\mathcal{T}_h$ be a simplicial decomposition of the domain Ω. Then

$$\begin{aligned}\int_\Omega \psi(x)\cdot e(x)\mathrm{d}x &= \int_\Omega \left(\mathcal{D}_2\phi(x)\cdot\mathcal{D}_1U(x) - \phi(x)\cdot f(U,x)\right)\mathrm{d}x,\\ &= \sum_{\tau\in\mathcal{T}}\int_\tau \left(\mathcal{D}_2\phi(x)\cdot\mathcal{D}_1U(x) - \phi(x)\cdot f(U,x)\right)\mathrm{d}x = \sum_{\tau\in\mathcal{T}} e_\tau, \quad (2.113)\end{aligned}$$

where the error contribution, e_τ, from a particular simplex $\tau \in \mathcal{T}_h$ is given by the integral

$$e_\tau = \int_\tau \left(\mathcal{D}_2\phi(x)\cdot\mathcal{D}_1U(x) - \phi(x)\cdot f(U,x)\right)\mathrm{d}x. \quad (2.114)$$

The error decomposition will be used to guide the adaptive refinement methods discussed later in §4.2.1. Next we provide a numerical experiment to show the accuracy of the error representation (2.105) when the adjoint solution is also numerically approximated.

Numerical Experiment: BVP with standard QoI

Consider the stationary advection-diffusion equation

$$\begin{cases}\nabla^2 u(x) + b\cdot\nabla u(x) = f(x), & x\in\Omega,\\ u(x) = 0, & x\in\partial\Omega,\end{cases} \quad (2.115)$$

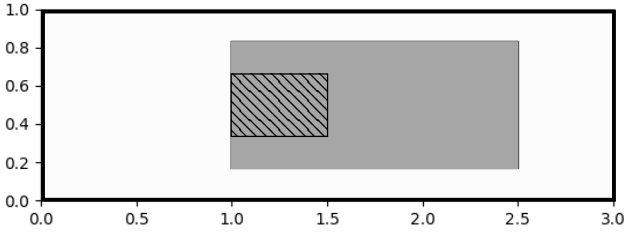
where $b = (300, 0)^\top$ and $\Omega = (0,3)\times(0,1)$. The source f, which is non-zero only over an interior region of the domain, is

$$f = \begin{cases}10000(x_1-1)(x_1-2.5)(x_2-\frac{1}{6})(x_2-\frac{5}{6}) & 1\le x_1\le 2.5,\ \frac{1}{6}\le x_2\le\frac{5}{6},\\ 0 & \text{else.}\end{cases} \quad (2.116)$$

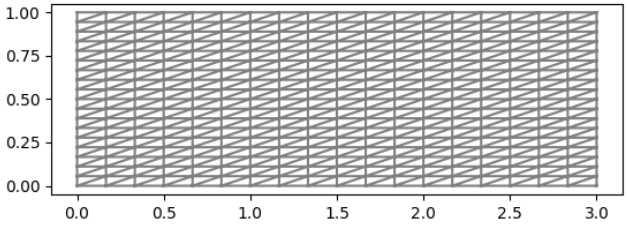
The standard QoI of form (2.104) is the integral of the solution over the rectangle $(1, 1.5)\times(1/3, 2/3)$:

$$Q(u) = \int_\Omega \psi(x)\cdot u(x)\mathrm{d}x, \quad \text{where} \quad \psi(x) = \begin{cases}1, & x\in(1,1.5)\times(\frac{1}{3},\frac{2}{3})\\ 0, & else.\end{cases} \quad (2.117)$$

Figure 2.2a provides a visualization of the supports of f and ψ relative to the domain Ω. For the numerical experiment, we solve the BVP (2.115) using the continuous



(a) Illustration of overlapping supports of functions from example in §2.4.2. The domain is $\Omega = (0,1) \times (0,3)$ outlined in black, $supp(f) = (1, 2.5) \times (1/6, 5/6)$ in grey, and $supp(\psi) = (1, 1.5) \times (1/3, 2/3)$ marked with diagonal lines.



(b) Mesh used for numerical solution in §2.4.2.

Figure 2.2

Galerkin method with piece-wise linear polynomials. The mesh used for the numerical solution is created by first making a uniform 18-by-18 rectangular grid over the domain $\Omega = (0,3) \times (0,1)$. Then each sub-region is bisected diagonally from top right corner to bottom left corner to create a mesh with 648 elements; see Figure 2.2b. A reference solution is obtained using the same numerical method but over a much finer mesh with 2880000 elements, created in the same fashion starting with a 1200-by-1200 grid. The adjoint problem is solved over the same mesh as the numerical solution but using fourth-degree piece-wise polynomials. The computed QoI $Q(U)$, reference QoI $Q(u)$, error estimate η from (2.105), reference error $Q(u) - Q(U)$, and effectivity ratio ρ_{eff} are all shown in Table 2.3. We see that the effectivity ratio is close to one, indicating

an accurate error estimate.

$Q(U)$	$Q(u)$	η	$Q(u) - Q(U)$	ρ_{eff}
-0.27790	-0.27863	-0.00072	-0.00073	0.988

Table 2.3: Results for numerical experiment of a BVP with standard QoI.

4.3 Classical Analysis: IBVPs

For IBVPs of form (1.6) we analyze one type of standard QoI that represents a weighted average (over space) of the solution u evaluated at a specific time $t^\star$. To simplify notation, let $(u(t^\star), v(t^\star)) = \int_\Omega u(x,t^\star) \cdot v(x,t^\star)\mathrm{d}x$ denote the $L^2(\Omega)$ inner-product of the two functions u, v evaluated at a specific time-value $t^\star$, where we have suppressed the dependence on x. The standard QoI is then written as

$$Q(u) = \int_\Omega \psi(x) \cdot u(x,t^\star)\mathrm{d}x = (\psi, u(t^\star))\,, \tag{2.118}$$

for some $\psi \in H_0^1(\Omega)$ with the $L^2(\Omega)$ inner-product. The error representation for an approximate QoI of form (2.118) is given in Theorem 7. Again, for this Theorem, we only require a numerical solution to the weak form of the IBVP (2.61).

Theorem 7. *Given a numerical solution $U(x,t)$ to (2.61) and data $\psi(x)$, for any $t^\star \in (0,T]$ the error $(\psi, e(t^\star))$ is given by*

$$(\psi, e(t^\star)) = (\phi(0), e(0)) + \int_0^{t^\star} (\phi(t), f(U,t) - U_t(t)) - (\mathcal{D}_2\phi(t), \mathcal{D}_1 U(t))\,\mathrm{d}t \tag{2.119}$$

where $\phi(x,t)$ is the solution of the adjoint problem

$$\begin{cases} -\dot{\phi}(x,t) = -\mathcal{D}^*\phi(x,t) + \overline{f_{u,U}(x,t)}^\top \phi(x,t), & x \in \Omega, \quad t \in [0,t^\star), \\ \phi(x,t) = 0, & x \in \partial\Omega, \quad t \in [0,t^\star), \\ \phi(x,t^\star) = \psi(x), & x \in \Omega. \end{cases} \tag{2.120}$$

The operator $\overline{f_{u,U}}^{\top}(x,t)$ *is the adjoint of the linear operator*

$$\overline{f_{u,U}}(x,t) = \int_0^1 \nabla_z f(z,x,t) ds, \tag{2.121}$$

and $z = su + (1-s)U$.

The linear differential operator $\mathcal{D}^*$ is the formal adjoint operator of a linear operator $\mathcal{D}$ in (1.6). Again, the function $\overline{f_{u,U}(x,t)}$ is chosen because it has the property

$$\overline{f_{u,U}(x,t)}e(x,t) = f(u,x,t) - f(U,x,t). \tag{2.122}$$

Note again that the adjoint problem is solved backwards in time with the initial condition given at $t = t^\star$.

Proof. Multiply the adjoint equation (2.120) by the error $e(x,t) = u(x,t) - U(x,t)$, and integrate over the space-time domain $\Omega \times [0, t^\star]$ to give

$$0 = \int_0^{t^\star} \left(\dot{\phi}(t), e(t)\right) \, \mathrm{d}t - \int_0^{t^\star} (\mathcal{D}^*\phi(t), e(t)) \, \mathrm{d}t + \int_0^{t^\star} \left(\overline{f_{u,U}(t)}^{\top}\phi(t), e(t)\right) \, \mathrm{d}t. \tag{2.123}$$

Looking at each term in (2.123) individually, integrate the first term by parts in time and enforce the initial condition $\phi(x,t^\star) = \psi(x)$,

$$\int_0^{t^\star} \left(\dot{\phi}(t), e(t)\right) \, \mathrm{d}t = (\psi, e(t^\star)) - (\phi(0), e(0)) - \int_0^{t^\star} (\phi(t), \dot{e}(t)) \, \mathrm{d}t. \tag{2.124}$$

From (2.109), The second term of (2.123) becomes

$$\int_0^{t^\star} (\mathcal{D}^*\phi(t), e(t)) \, \mathrm{d}t = \int_0^{t^\star} (\mathcal{D}_2\phi(t), \mathcal{D}_1 e(t)) \, \mathrm{d}t. \tag{2.125}$$

Similarly, using the property of the adjoint (2.48) and the property (2.122)

$$\begin{aligned} \int_0^{t^\star} \left(\overline{f_{u,U}(t)}^{\top}\phi(t), e(t)\right) \, \mathrm{d}t &= \int_0^{t^\star} \left(\phi(t), \overline{f_{u,U}(t)}e(t)\right) \, \mathrm{d}t, \\ &= \int_0^{t^\star} (\phi(t), f(u,t) - f(U,t)) \, \mathrm{d}t. \end{aligned} \tag{2.126}$$

Combining (2.123), (2.125), and (2.126) yields,

$$(\psi, e(t^\star)) = (\phi(0), e(0)) + \int_0^{t^\star} (\phi(t), \dot{e}(t)) \ \mathrm{d}t, \\ + \int_0^{t^\star} (\mathcal{D}_2\phi(t), \mathcal{D}_1 e(t)) \ \mathrm{d}t - \int_0^{t^\star} (\phi(t), f(u,t) - f(U,t)) \ \mathrm{d}t.$$

Recalling that $e(x,t) = u(x,t) - U(x,t)$ and applying the weak form of the IBVP (2.61) gives the error representation

$$(\psi, e(t^\star)) = (\phi(0), e(0)) + \int_0^{t^\star} (\phi(t), -U_t(t)) \ \mathrm{d}t, \\ + \int_0^{t^\star} (\mathcal{D}_2\phi(t), -\mathcal{D}_1 U(t)) \ \mathrm{d}t - \int_0^{t^\star} (\phi(t), -f(U,t)) \ \mathrm{d}t.$$

□

Once again, in practice, since operator $\overline{f_{u,U}(x,t)}$ requires the true solution to (1.6), it is approximated by

$$\overline{f_{u,U}(x,t)} \approx \nabla_u f(U,x,t). \tag{2.127}$$

With this, the right side of the adjoint equation (2.120) is approximated by

$$\overline{f_{u,U}(x,t)}^\top \phi(x,t) \approx (\nabla_u f(U,x,t))^\top \phi(x,t). \tag{2.128}$$

Note that for the IBVP, we **do not** say that $e(x,0) = u(x,0) - U(x,0)$ is zero. This is due to the fact that the initial condition of the numerical solution $U(x,0)$ is the projection of the true initial condition $u(x,0)$ onto the solution space. This projection introduces a non-zero error that must be accounted for.

We decompose the error representation (2.119) over the time domain in a similar fashion as (2.89). Let the domain of integration $(0, t^\star)$ be partitioned into $N^\star$ sub-

intervals with endpoints $\{t_0, t_1, \ldots, t_{N^\star}\}$. Then we decompose the error (2.119) as

$$(\psi, e(t^\star)) = (\phi(0), e(0)) + \sum_{i=0}^{N^\star-1} \int_{t_i}^{t_{i+1}} (\phi(t), f(U,t) - U_t(t)) - (\mathcal{D}_2\phi(t), \mathcal{D}_1 U(t))\,\mathrm{d}t, \tag{2.129}$$

$$= (\phi(0), e(0)) + \sum_{i=0}^{N^\star-1} e_{(t_i,t_{i+1})}. \tag{2.130}$$

Where the error contributions, $e_{(t_i,t_{i+1})}$, are given by the integrals

$$e_{(t_i,t_{i+1})} = \int_{t_i}^{t_{i+1}} (\phi(t), f(U,t) - U_t(t)) - (\mathcal{D}_2\phi(t), \mathcal{D}_1 U(t))\,\mathrm{d}t, \quad \text{for } i = 0, 1, \ldots, N^\star - 1. \tag{2.131}$$

Numerical Experiment: IBVP with standard QoI

Consider the collisionless Vlasov equation with forcing term

$$\dot{u}(x,p,t) + p\nabla_x \cdot u(x,p,t) = f(x,p) \quad \Omega_x \times \Omega_p \times (0,1], \tag{2.132}$$

$$u(x,p,0) = u_0(x,p) \quad \Omega_x \times \Omega_p \tag{2.133}$$

where x is the position of the particle, p is the momentum , and the solution $u(x,p,t)$ is the density. We use the domains $\Omega_x = (0,8) = \Omega_p$. The initial condition is given by the bump-function

$$u_0(x,p) = \begin{cases} 200(x-1.5)^2(x-3)^2(p-1.5)^2(p-3)^2, & 1.5 < x, p < 3, \\ 0, & else. \end{cases} \tag{2.134}$$

When solving (2.132) we treat the variables x and p as the "spatial" variable in the order (x,p) and the "spatial domain" is $\Omega = \Omega_x \times \Omega_p$. Let the right-hand side function be

$$f = \frac{3}{\sqrt{2\pi}} e^{\frac{-9(p-3.5)^2}{2}} + \frac{3}{\sqrt{2\pi}} e^{\frac{-9(p-4.5)^2}{2}}. \tag{2.135}$$

The standard QoI of form (2.118) is the weighted integral

$$Q(u) = \int_{\Omega} \psi(x,p) \cdot u(x,p,0.8) \mathrm{d}\Omega, \tag{2.136}$$

with weight function

$$\psi(x,p) = \begin{cases} 200(x-1.5)^2(x-3)^2(p-1.5)^2(p-3)^2, & 1.5 < x,p < 3, \\ 0, & else. \end{cases} \tag{2.137}$$

A numerical solution is obtained using the continuous Galerkin method with linear polynomials over Ω and the Crank-Nicolson method to take steps in time. The spatial mesh is created by making a uniform 32-by-32 grid over Ω and bisecting each sub-region diagonally from top right to bottom left, giving a mesh with 2048 elements. The temporal domain $(0, 1]$ is partitioned into 10 sub-intervals of equal length. A reference solution is obtained using quadratic polynomials over Ω using a mesh that is created from a 128-by-128 grid. The reference solution also uses Crank-Nicolson for the time steps using 100 sub-intervals. The adjoint problem is solved using quadratic polynomials in space with a mesh created from a 128-by-128 grid and Crank-Nicolson steps in time with 40 sub-intervals. The computed QoI $Q(U)$, reference QoI $Q(u)$, error estimate η from (2.119), reference error $Q(u) - Q(U)$, and effectivity ratio ρ_{eff} are all shown in Table 2.4. We see that the effectivity ratio is close to one, indicating an accurate error estimate.

$Q(U)$	$Q(u)$	η	$Q(u) - Q(U)$	ρ_{eff}
0.16095	-0.13248	-0.29636	-0.29343	1.009

Table 2.4: Results for numerical experiment of a IBVP with standard QoI.

Chapter 3

Adjoint-Based a Posteriori Error Analysis for NSQoI

This chapter presents two novel methods of adjoint-based *a posteriori* error analysis for a QoI that is not covered in previous work. We begin by introducing a rigorous definition of a QoI that represents the *time at which* a specified event occurs. This QoI is non-standard as it cannot be represented by a linear functional of the solution, nor can it be trivially linearized. *A priori* convergence results for the non-standard QoI are derived with a numerical experiment showcasing the convergence rate. We then derive two adjoint-based *a posteriori* methods to accurately estimate the error in our non-standard QoI. The first method relies on multiply applications of Taylor's Theorem with remainder and the weak-form of the differential equation. The second method applies root-finding methods to a highly accurate "corrected" solution of the differential equation which allows us to compute the error in the computed QoI. Both methods take advantage of the classical analysis presented in §2.4. Numerical examples showing the accuracy and limitations of the error estimates of this non-standard QoI are provided at the end of this chapter.

1 Analysis of the Non-standard QoI

This section provides details and analysis of our non-standard QoI. We begin by rigorously defining the QoI that represents that time at which a specified event occurs. We then give an *a priori* convergence result for a numerically computed non-standard QoI based on the convergence rate of the numerical method used. Finally, we derive the adjoint-based *a posteriori* error analysis of this non-standard QoI. We take two approaches to the *a posteriori* analysis. The first method holds the same spirit as the classical analysis, providing an error representation that depends on solutions to adjoint problems. The second method uses the classical results along with root-finding methods to obtain a corrected QoI which allows us to compute the error. Since this non-standard QoI requires an explicit dependence on time, it is only applicable to the two time-dependent differential equations: IVPs (1.1) and IBVPs (1.6).

1.1 Defining the Non-standard QoI

Let $G(u;t)$ be a linear functional of u, which is implicitly dependent on t through u, but not implicitly dependent on x (i.e. if $u = u(x,t)$ the x will be integrated out in G). Let R be a chosen threshold value and assume that there is at least one time-value $t^{\star\star}$ during the interval $(0,T]$ for which we have $G(u;(t^{\star\star})) = R$. Define the time $H(u,\hat{t})$ for fixed G and R as

$$H(u,\hat{t}) = \min \underset{t\in(\hat{t},T]}{\arg} (G(u;t) = R). \tag{3.1}$$

The input $\hat{t}$ is chosen to obtain different occurrences of the event $G(u;t) = R$. Notice that we are finding the minimum t such that $G(u;t) = R$ over the sub-interval $(\hat{t},T]$. Thus, the $\hat{t}$ we choose must be before the event we are interested in but no earlier than the previous occurrence of $G(u;t) = R$. To illustrate, if $\hat{t} = 0$, then $H(u,0)$ is the very first time on the interval (0,T] that the functional $G(u;t)$ achieves the threshold value R. Similarly, to obtain the second time that the functional achieves the threshold value,

we can choose any $\hat{t}$ between the first and second occurrence. More precisely, $H(u,\hat{t})$ will give the second occurrence of $G(u;t) = R$ over $(0,T]$ if $H(u,0) \leq \hat{t} < H(u,H(u,0))$. Since the left inequality allows equality, a sequence of nested functions can be used to obtain any occurrence of the event.

Lemma 1. *Assume there are at least J occurrences of the event $G(u;t) = R$ over the time interval $(0,T]$. Let $\left\{\hat{t}_i\right\}_{i=1}^{J}$ be a sequence defined as*

$$\hat{t}_1 = 0, \tag{3.2}$$

$$\hat{t}_i = H(u,t_{i-1}), \quad \textit{for } i = 2,3,\ldots,J. \tag{3.3}$$

Then setting $\hat{t} = \hat{t}_i$ in (3.1) will obtain the i-th occurrence of the event $G(u;t) = R$, for $i = 1,2,\ldots,J$.

Finally, the non-standard quantity of interest $Q(u)$ for fixed $\hat{t}$ is defined as

$$Q(u) = H(u,\hat{t}). \tag{3.4}$$

The only difference between applying this QoI to the differential equations is the form of the functional $G(u;t)$. This is described in the following sections.

NSQoI for IVPs

For IVPs (1.1) the functional $G(u;t)$ in (3.1) takes the form

$$G(u;t) = \psi \cdot u(t). \tag{3.5}$$

In terms of our analysis, this can be thought of as a standard QoI for IVPs of form (2.78) without plugging in a specific time value. The non-standard QoI (3.4) then represents the time at which a linear combination of the components of the solution achieves a threshold value.

NSQoI for IBVPs

For IBVPs (1.6) the functional $G(u;t)$ in (3.1) takes the form

$$G(u;t) = \int_\Omega \psi(x) \cdot u(x,t)\mathrm{d}x = (\psi, u(t))\,. \tag{3.6}$$

This can be thought of as a standard QoI for IBVPs of form (2.118) without plugging in a specific time value. The non-standard QoI (3.4) then represents the time at which a weighted integral of the solution achieves a threshold value.

1.2 A Priori Analysis of NSQoI

This section derives the *a priori* convergence results for a numerically computed non-standard QoI (3.4), assuming a certain convergence rate of the numerical solution itself. The results apply to both of our time-dependent differential equations (1.1) and (1.6). As such, we keep the notation general enough to cover both cases and point out any subtleties.

Let U be a numerical solution to one of our time-dependent differential equations and assume that

$$\|u(t) - U(t)\| \leq Ch^p, \tag{3.7}$$

for all $t \in [0,T]$, for some constant $C > 0$, and where h denotes the temporal step-size used to compute the numerical solution. The norm used in (3.7) depends on the context of the differential equation. For IVPs (1.1) this will be the standard Euclidean norm over $\mathbb{R}^d$. For IBVPs (1.6) the $L^2(\Omega)$ norm is used and the dependence on x is omitted in (3.7).

For a given value of the threshold R define the functional $G(u;t)$ either as (3.5) or (3.6), depending on the type of differential equation. For ease of notation, let $t_t = Q(u)$ be the true non-standard QoI (3.4) and let $t_c = Q(U)$ be the computed non-standard

QoI. Here, we assume that G satisfies the Lipschitz condition in u,

$$|G(u_1;t) - G(u_2;t)| \leq K\|u_1(t) - u_2(t)\|, \tag{3.8}$$

for some constant $K > 0$. Define the true error in the QoI, e_Q, to be

$$e_Q = t_t - t_c. \tag{3.9}$$

Theorem 8 (Convergence of the non-standard QoI)**.** *Assume there is a numerical approximation to the solution, U, of either* (1.1) (1.6) *satisfying* (3.7)*, and the functional $G(u;t)$ is continuously differentiable with respect to t in a neighborhood, B, which contains both the true QoI, t_t, as well as its numerical approximation, t_c. Further assume there exists an $M > 0$ such that*

$$\left|\frac{dG}{dt}(u;t)\right| > M, \tag{3.10}$$

for all $t \in B$. Then the error e_Q in the computed QoI, defined by (3.9)*, satisfies the bound,*

$$e_Q \leq \widehat{C}h^p,$$

for some constant $\widehat{C}$ which depends on M, C and K.

Proof. Given the true solution $u(t)$ to the differential equation, we consider the functional G as an explicit function of t, i.e.,

$$G(u;t) = G(u(t)) = G(t). \tag{3.11}$$

Since $G(u;t)$ is continuously differentiable in t, for $t \in B$, by the Inverse Function Theorem (see [56]) we have $t = t(G)$ for G in the image of B, and

$$\frac{dt}{dG}(G(t)) = \frac{1}{\frac{dG}{dt}(t(G))}. \tag{3.12}$$

Applying the Mean-value Theorem (see [4]) we have, for some ξ between $G(u;t_t)$ and $G(u;t_c)$,

$$t_t - t_c = \frac{dt}{dG}(\xi)\left[G(u;t_t) - G(u;t_c)\right] = \frac{1}{\frac{dG}{dt}(t(\xi))}\left[G(u;t_t) - G(u;t_c)\right]. \tag{3.13}$$

Adding and subtracting the term $G(U;t_c)$ and recalling that $G(u;t_t) = R = G(u;t_c)$,

$$\begin{aligned} t_t - t_c &= \frac{1}{\frac{dG}{dt}(t(\xi))}\left[G(u;t_t) - G(u;t_c) + G(U;t_c) - G(U;t_c)\right], \\ &= \frac{1}{\frac{dG}{dt}(t(\xi))}\left[G(U;t_c) - G(u;t_c)\right]. \end{aligned} \tag{3.14}$$

Taking norms (absolute values for scalars), and using (3.10) and (3.8),

$$|t_t - t_c| = \left|\frac{1}{\frac{dG}{dt}(t(\xi))}\right| |G(U;t_c) - G(u;t_c)| \leq \frac{1}{M} K \left\|u(t_c) - U(t_c)\right\| \leq \frac{1}{M} KCh^p. \tag{3.15}$$

Defining $\widehat{C} := \frac{K\,C}{M}$ gives the desired result. □

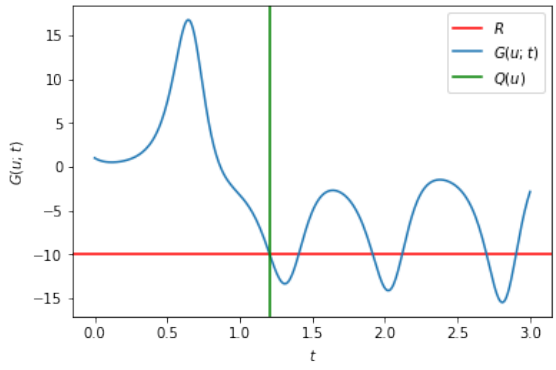
We provide an example to illustrate the *a priori* results.

Numerical Experiment: NSQoI Convergence Rate

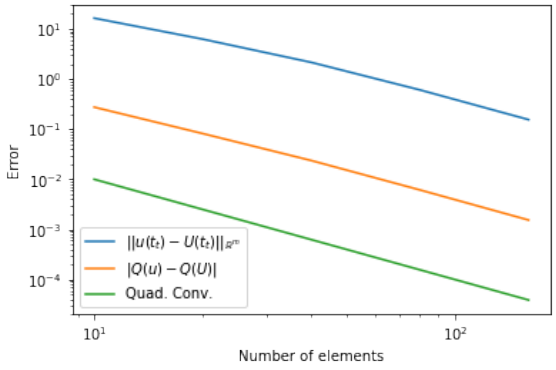
To illustrate the convergence results in Theorem 8, we consider the Lorenz system,

$$\left.\begin{aligned} \dot{u}_1 &= \sigma(u_2 - u_1), \\ \dot{u}_2 &= ru_1 - u_2 - u_1u_3, \\ \dot{u}_3 &= u_1u_2 - bu_3, \end{aligned}\right\} \; t \in (0,3] \qquad \text{with} \qquad \begin{cases} u_1(0) = 1, \\ u_2(0) = 0, \\ u_3(0) = 24, \end{cases} \tag{3.16}$$

and set $\sigma = 10, r = 28,$ and $b = \frac{8}{3}$ (see §4.1.1 for more details of this example). We define the functional $G(u;t) = u_1(t)$ set the threshold value $R = -10$, and choose $\hat{t} = 0$ to obtain the first occurrence of $u_1(t) = -10$. Figure 3.1a illustrates an accurate reference solution as well as the threshold value and the QoI. Figure 3.1b shows the convergence rates for the error in the solution and the error in the non-standard QoI when using the cG(1) method for computing the numerical solution. The cG(1) method has second order accuracy and this convergence rate is observed both for the solution and the non-standard QoI.



(a) Reference solution and QoI for the Lorenz system (3.16).



(b) Converge rates of the error in the solution and the error in the QoI. The numerical solution U and QoI t_c, are computed using the cG(1) method.

Figure 3.1

1.3 A Posteriori Analysis of NSQoI

This section derives the adjoint-based *a posteriori* error analysis of the non-standard QoI (3.4). We take two approaches to the error analysis. In the first approach, we utilize Taylor's Theorem with remainder and the weak form of the differential equation to create a representation of the error. This error representation contains terms that can be viewed as the error in a standard QoI and thus is computable by applying classical analysis. The second approach uses classical results to make accurate pointwise corrections to the numerical solution. An iterative root-finding process is used with these corrected points to find an accurate reference value of the non-standard QoI which is then used to compute the error. Both approaches are discussed in detail followed by several numerical experiments to show the accuracy and limitations of the methods.

Adjoint-based Error Representation

We derive an *a posteriori* representation of the error in the non-standard QoI (3.4) using Taylor's Theorem with remainder. The error representation is then approximated and finally made computable by utilizing the classical analysis presented in §2.4. Theorem 9 and its proof are given in the context of the IBVP (1.6). The error representation for the NSQoI for IVPs (1.1) is obtained by making the identification $\mathcal{D} = 0$ and replacing the $L^2(\Omega)$ inner-product by the Euclidean inner-product. The result for IVPs is provided in Corollary 1.

Theorem 9 (Error Representation for NSQoI for IBVPs)**.** *Let u be the solution to* (1.6) *and U be an approximation of u obtained via a Galerkin method. Let the functional G be defined by* (3.6)*, and denote $t_t = Q(u)$ and $t_c = Q(U)$ for the true and computed non-standard QoI* (3.4)*, respectively. The error in the computed non-standard QoI is given as*

$$e_Q = t_t - t_c = \frac{(\psi, e(t_c)) + \widetilde{\mathcal{R}}_1}{(\mathcal{D}_2\psi, \mathcal{D}_1 U(t_c)) - (\psi, f(U,t_c)) - (\nabla_u f(U,t_c)^\top \psi, e(t_c)) + (\mathcal{D}_2\psi, \mathcal{D}_1 e(t_c)) - \widetilde{\mathcal{R}}_2} \tag{3.17}$$

Where $e(x,t) = u(x,t) - U(x,t)$, and the two remainders are $\widetilde{\mathcal{R}}_1 = (\psi, \mathcal{R}_1(t_c,t_t))$ where $\mathcal{R}_1(x,t_c,t_t) = \mathcal{O}\left(|t_t - t_c|^2\right)$ and $\widetilde{\mathcal{R}}_2 = (\psi, \mathcal{R}_2(u,U,t_c))$ where $\mathcal{R}_2(u,U,x,t_c) = \mathcal{O}\left(||u-U||^2_{L^2(\Omega)}\right)$.

Proof. From the definitions of t_t and t_c we have

$$G(u;t_t)) - G(U;t_c)) = (w, u(t_t) - U(t_c)) = R - R = 0. \tag{3.18}$$

Linearizing $u(x,t_t)$ around t_c using Taylor's Theorem with remainder and defining $e(x,t) = u(x,t) - U(x,t)$ gives

$$\begin{aligned} 0 &= (w, u(t_c) + (t_t - t_c)u_t(t_c) + \mathcal{R}_1(t_c,t_t) - U(t_c)) \\ &= (w, e(t_c)) + (t_t - t_c)\,(w, u_t(t_c)) + (w, \mathcal{R}_1(t_c,t_t)) \end{aligned} \tag{3.19}$$

where the remainder $\mathcal{R}_1(x, t_c, t_t)$ is of order $\mathcal{O}\left((t_t - t_c)^2\right)$. Rearranging (3.19) to isolate the error we are interested in yields

$$t_t - t_c = -\frac{(w, e(t_c)) + (w, \mathcal{R}_1(t_c, t_t))}{(w, u_t(t_c))}. \tag{3.20}$$

From the weak formulation of the IBVP (2.61), the denominator of (3.20) becomes

$$\begin{aligned}(w, u_t(t_c)) &= -\left(\mathcal{D}_2 w, \mathcal{D}_1 u(t_c)\right) + \left(w, f(u, t_c)\right) \\ &= -\left(\mathcal{D}_2 w, \mathcal{D}_1 u(t_c)\right) + \left(w, f(u, t_c)\right) - \left(\mathcal{D}_2 w, \mathcal{D}_1 U(t_c)\right) + \left(\mathcal{D}_2 w, \mathcal{D}_1 U(t_c)\right) \\ &= -\left(\mathcal{D}_2 w, \mathcal{D}_1 U(t_c)\right) - \left(\mathcal{D}_2 w, \mathcal{D}_1 e(t_c)\right) + \left(w, f(u, t_c)\right).\end{aligned} \tag{3.21}$$

We use Taylor's Theorem one more time to linearize $f(u, x, t_c)$ around U giving

$$\left(w, f(u, t_c)\right) = \left(w, f(U, t_c)\right) + \left(w, \nabla_u f(U, t_c) e(t_c)\right) + \left(w, \mathcal{R}_2(u, U, t_c)\right), \tag{3.22}$$

$$= \left(w, f(U, t_c)\right) + \left(\nabla_u f(U, t_c)^\top w, e(t_c)\right) + \left(w, \mathcal{R}_2(u, U, t_c)\right), \tag{3.23}$$

where the remainder $\mathcal{R}_2(u, U, x, t_c)$ is of order $\mathcal{O}\left(||u - U||^2_{L^2(\Omega)}\right)$. Substituting (3.23) into (3.21) and combining that back into (3.20) yields the final result.

□

Remark 1. *From Taylor's Theorem, the remainder $\mathcal{R}_1(t_t, t_c)$ is*

$$\mathcal{R}_1(t_t, t_c) = \frac{1}{2}\frac{d^2 G}{dt^2}(y(\xi))(t_t - t_c)^2, \tag{3.24}$$

for some ξ between t_t and t_c.

The error representation for NSQoI (3.4) related to an IVP (1.1) is easily obtained from Theorem 9 by identifying the differential operator $\mathcal{D}$ with the null operator $\mathcal{D}u = 0$ for all u. Then $\mathcal{D}_1$ and $\mathcal{D}_2$ are also null. In the context of IVPs, we also use the Euclidean inner-product instead of the $L^2(\Omega)$ inner-product.

Corollary 1. *Let u be the solution to* (1.1) *and U be an approximation of u obtained via a Galerkin method. Let the functional G be defined by* (3.5)*, and denote $t_t = Q(u)$*

and $t_c = Q(U)$ *for the true and computed non-standard QoI* (3.4)*, respectively. The error in the computed non-standard QoI is given as*

$$e_Q = t_t - t_c = -\frac{\psi \cdot e(t_c) + \psi \cdot \mathcal{R}_1(t_c, t_t)}{\psi \cdot f(U, t_c) + \nabla_u f(U, t_c)^\top \psi \cdot e(t_c) + \psi \cdot \mathcal{R}_2(u, U, t_c)}. \tag{3.25}$$

Where $e(x,t) = u(x,t) - U(x,t)$*, and the remainders are such that* $\mathcal{R}_1(x, t_c, t_t) = \mathcal{O}\left(|t_t - t_c|^2\right)$ *and* $\mathcal{R}_2(u, U, x, t_c) = \mathcal{O}\left(||u - U|\,|_{\mathbb{R}^d}^2\right)$.

The error representations (3.17) and (3.25) contain remainders that decay faster than other terms as the numerical solution U becomes more accurate. So, setting $\mathcal{R}_1 \approx 0$ and $\mathcal{R}_2 \approx 0$ gives approximations to the error representations. The approximations contain terms that contain the error $e = u - U$ which can be viewed as standard QoIs. The classical analysis from §2.4 allows us to compute the approximations to the error in the non-standard QoI. The error approximations and the adjoint problems required to make them computable are presented below.

NSQoI Error Approximation: IVPs

$$t_t - t_c \approx -\frac{\psi \cdot e(t_c)}{\psi \cdot f(U, t_c) + \nabla_u f(U, t_c)^\top \psi \cdot e(t_c)}. \tag{3.26}$$

The approximation (3.26) contains two terms that are computed using Theorem 4, thus we have two adjoint problems associated with the error.

First adjoint problem To obtain $\psi \cdot e(t_c)$, we solve the adjoint problem

$$\begin{cases} -\dot{\phi}_1 = \overline{f_{u,U}(t)}^\top \phi_1, & t \in [0, t_c), \\ \phi_1(t_c) = \psi. \end{cases} \tag{3.27}$$

Second adjoint problem To obtain $\nabla_u f(U, t_c)^\top \psi \cdot e(t_c)$, we solve the adjoint problem

$$\begin{cases} -\dot{\phi}_2 = \overline{f_{u,U}(t)}^\top \phi_2, & t \in [0, t_c), \\ \phi_2(t_c) = \nabla_u f(U, t_c)^\top \psi. \end{cases} \tag{3.28}$$

NSQoI Error Approximation: IBVPs

$$t_t - t_c \approx \frac{(\psi, e(t_c))}{(\mathcal{D}_2\psi, \mathcal{D}_1 U(t_c)) - (\psi, f(U,t_c)) - (\nabla_u f(U,t_c)^\top \psi, e(t_c)) + (\mathcal{D}_2\psi, \mathcal{D}_1 e(t_c))}. \tag{3.29}$$

The approximation (3.29) contains three terms that are computed using Theorem 7, thus we have three adjoint problems associated with the error.

First adjoint problem To obtain $(\psi, e(t_c))$, we solve the adjoint problem

$$\begin{cases} -\dot{\phi}(x,t) = -\mathcal{D}^*\phi(x,t) + \overline{f_{u,U}(x,t)}^\top \phi(x,t), & x \in \Omega, \quad t \in [0, t_{t_c}), \\ \phi(x,t) = 0, & x \in \partial\Omega, \quad t \in [0, t_{t_c}), \\ \phi(x, t_{t_c}) = \psi(x), & x \in \Omega. \end{cases} \tag{3.30}$$

Second adjoint problem To obtain $\left(\nabla_u f(U,t_c)^\top \psi, e(t_c)\right)$, we solve the adjoint problem

$$\begin{cases} -\dot{\phi}(x,t) = -\mathcal{D}^*\phi(x,t) + \overline{f_{u,U}(x,t)}^\top \phi(x,t), & x \in \Omega, \quad t \in [0, t_{t_c}), \\ \phi(x,t) = 0, & x \in \partial\Omega, \quad t \in [0, t_{t_c}), \\ \phi(x, t_{t_c}) = \nabla_u f(U,t_c)^\top \psi(x), & x \in \Omega. \end{cases} \tag{3.31}$$

Third adjoint problem To obtain $(\mathcal{D}_2\psi, \mathcal{D}_1 e(t_c))$, we solve the adjoint problem

$$\begin{cases} -\dot{\phi}(x,t) = -\mathcal{D}^*\phi(x,t) + \overline{f_{u,U}(x,t)}^\top \phi(x,t), & x \in \Omega, \quad t \in [0, t_{t_c}), \\ \phi(x,t) = 0, & x \in \partial\Omega, \quad t \in [0, t_{t_c}), \\ \phi(x, t_{t_c}) = \mathcal{D}^*\psi(x), & x \in \Omega. \end{cases} \tag{3.32}$$

Remark 2. *Note that the functional G may achieve the value R at multiple times. Assume there exists a time $\tilde{t} > t_t$ such that $G(u;\tilde{t}) = R$. Equation* (3.18) *is then valid at time $\tilde{t}$, i.e., $G(U;t_c) = R = G(u;\tilde{t})$ and the error approximation (either* (3.26) *or* (3.29)*) follows with t_t replaced by $\tilde{t}$. In the approximations we have replaced the remainder $\mathcal{R}_1$ by zero. If the numerical solution is sufficiently accurate, then $|t_t - t_c| < |\tilde{t} - t_c|$ and $0 \approx \mathcal{R}_1(t_c, t_t) \ll \mathcal{R}_1(t_c, \tilde{t})$. However, if the numerical solution is inaccurate,*

we may have the reverse situation, where $|t_t - t_c| > |\tilde{t} - t_c|$*, in which case the error estimate will be inaccurate or worse,* $\mathcal{R}_1(t_c, \tilde{t}) \approx 0$ *and the estimate may indicate the value of* $\tilde{t} - t_c$ *rather than* $t_t - t_c$*. We observe this phenomenon in §2.1.4 which is illustrated by Table 3.10 and Figure 3.5b.*

Many numerical experiments showcasing the accuracy of the approximations (3.26) and (3.29) are provided later in §3.1.4.

Decomposition of Error Approximations Both of the approximations (3.26) and (3.29) can be written in the form

$$t_t - t_c \approx \frac{\mathbb{E}}{C}, \tag{3.33}$$

where $\mathbb{E} = \psi \cdot e(t_c)$ in (3.26) and $\mathbb{E} = (\psi, e(t_c))$ in (3.29). We obtain a decomposition of the errors by decomposing $\mathbb{E}$ in the same fashion as (2.89) and (2.129) while leaving the denominator C intact.

The error (3.26) in the NSQoI associated with IVPs is decomposed as

$$t_t - t_c \approx \frac{1}{\mathcal{C}} \sum_{i=0}^{N^*-1} e_{(t_i, t_{i+1})} \tag{3.34}$$

with error contributions $e_{(t_i,t_{i+1})}$ given in (2.90).

The error (3.29) in the NSQoI associated with IBVPs is decomposed as

$$t_t - t_c \approx \frac{1}{\mathcal{C}} \left((\phi(0), e(0)) + \sum_{i=0}^{N^\star-1} e_{(t_i, t_{i+1})} \right) \tag{3.35}$$

with error contributions $e_{(t_i,t_{i+1})}$ given in (2.131).

Error from Iterative Root-finding Method

This section describes our second approach to creating an approximation of the error of a computed non-standard QoI (3.4). The work in this section applies to both IVPs (1.1)

and IBVPs (1.6). As such, we use a generic functional $G(u;t)$ that could be of either form (3.5) or (3.6). Given an approximate solution $U(t)$ to the differential equation with numerical QoI t_c, define $g(t)$ as

$$\begin{aligned} g(t) &= G(u;t) - R, \\ &= G(U;t) + \big(G(u;t) - G(U;t)\big) - R, \\ &= G(U;t) + G(e;t) - R. \end{aligned} \tag{3.36}$$

With this definition, $g(t)$ has the property $g(t_t) = 0$. At any given $t = t^\star$ we estimate $G(e;t^\star)$ using the error representation from the classical analysis (either Theorem 4 for IVPs or Theorem 7 for IBVPs). The initial conditions for the adjoint problems are $\phi(t = t^\star) = \psi$, where ψ comes from the definition of $G(u;t)$.

With this point-wise method of computing $g(t)$, we find t^* such that $g(t^*) \approx 0$ via a standard root finding procedure. Then

$$\eta(U) = t^* - t_c, \tag{3.37}$$

is an approximation of the error e_Q in the non-standard QoI (3.4). There are many options for root finding methods for computing η. In this thesis, we use two of the basic root finding methods: the secant method and the inverse quadratic method, which are briefly presented below.

Secant method

Given initial values x_0, x_1, the method is defined by the recurrence

$$x_n = \frac{x_{n-2} * g(x_{n-1}) - x_{n-1} * g(x_{n-2})}{g(x_{n-1}) - g(x_{n-2})} \quad n = 2, 3, \ldots \tag{3.38}$$

(See [44]). For the initial guesses the examples presented choose $x_0 < t_c < x_1$. These choices are made precise in the numerical examples in §3.1.4.

Inverse quadratic interpolation

Given initial values x_0, x_1, x_2, the method is defined by the recurrence

$$\begin{aligned} x_n = \quad & \frac{x_{n-3}\ g_{n-2}\ g_{n-1}}{(g_{n-3}-g_{n-2})(g_{n-3}-g_{n-1})} + \frac{x_{n-2}\ g_{n-3}\ g_{n-1}}{(g_{n-2}-g_{n-3})(g_{n-2}-g_{n-1})}, \\ & + \frac{x_{n-1}\ g_{n-2}\ g_{n-3}}{(g_{n-1}-g_{n-2})(g_{n-1}-g_{n-3})}, \qquad n = 3, 4, \ldots. \end{aligned} \tag{3.39}$$

(See [33]). The choice of the initial guesses is made precise in the numerical examples in §3.1.4.

1.4 Numerical Experiments: Error in NSQoI

This section provides several numerical experiments for both methods of our analysis of the error in the non-standard QoI (3.4). These experiments include many types of IVPs (linear, nonlinear, systems) and IBVPs. First we provide specifics for how we compute the numerical non-standard QoI and how we implement root-finding for the method presented in §3.1.3.

Numerical Methods:
All numerical solutions are obtained either using the continuous Galerkin method with piece-wise linear polynomials or using the Crank-Nicolson method. Since the Crank-Nicolson finite difference scheme is nodally equivalent to the cG(1) finite element method with a trapezoidal rule quadrature, given $t_i < t_c < t_{i+1}$, the numerical QoI $Q(U) = t_c$ may be computed by using linear interpolation as,

$$t_c = \frac{R(t_i - t_{i+1})}{G(U;t_i) - G(U;t_{i+1})} - \frac{t_i G(U;t_i) - t_{i+1} G(U;t_{i+1})}{G(U;t_i) - G(U;t_{i+1})}.$$

Root-finding Methods:
When implementing the secant method (3.38), the two grid-points closest to the QoI are used as initial guesses:

$$x_0 = t_L \text{ and } x_1 = t_R, \tag{3.40}$$

where $t_L < t_c < t_R$, with no other grid-points in between.

For the inverse quadratic interpolation scheme (3.39), the initial guesses are the two closest grid-points to the left of the QoI and one to the right:

$$x_0 = t_{LL},\ x_1 = t_L \text{ and } x_2 = t_R, \tag{3.41}$$

where $t_{LL} < t_L < t_c < t_R$, with no other grid-points in between. For most examples the adjoint solutions are computed using the cG(3) method with 100 finite elements, with the exceptions of §3.1.4 where cG(3) is used with 40 elements and §4.1.1 where cG(2) with 100 elements is used. For all methods define n_{adj} to be the number of adjoint solutions required to compute the error in the QoI. This number can be seen as the relative cost of implementing the different methods.

Linear problem

We consider the initial value problem

$$\dot{u} = \sin(2\pi t)u, \quad t \in (0,1], \quad u(0) = 1,$$

with analytic solution

$$u(t) = \exp\left(\frac{1}{2\pi}(1 - \cos(2\pi t))\right).$$

Let $R = 1.3$ and $G(u;t) = u(t)$. The true QoI is given by

$$t_t = Q(u) = \min_{t\in(0,1]} \arg(u(t) = 1.3) = \frac{1}{2\pi}(\arccos(-2\pi \ln(1.3) + 1)).$$

For this problem, the terms in (3.26) are

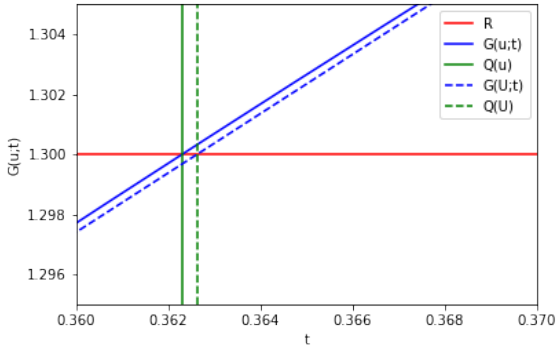
$$\psi = 1,\ f(u,t) = \sin(2\pi t)u,\ \nabla_u f(u,t) = \sin(2\pi t),$$

hence, for (3.27), (3.28), and (3.36) the values needed are

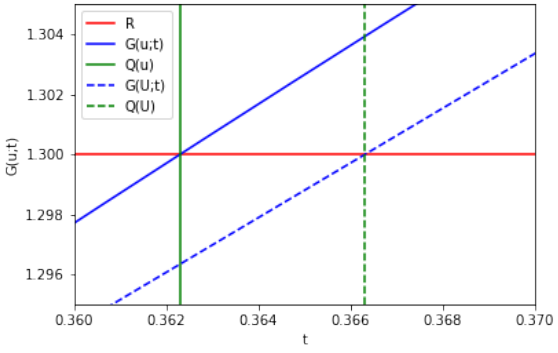
$$\psi_1 = -1,\ \psi_2 = \sin(2\pi t_c),\ \psi_3 = 1.$$

The true solution and QoI are shown in Figure 3.2. This graph includes a horizontal line at $G(u;t) = R$, to indicate the threshold value of interest, as well as a vertical line

denoting the true value of the QoI, i.e. the first time the threshold is crossed. Figure 3.2 compares the numerical QoI to the true value for both the numerical schemes. True errors, error estimates and effectivity ratios are provided in Tables 3.1 and 3.2. All methods provide excellent effectivity ratios, but the iterative methods require many more applications of Theorem 4 and hence require solving more adjoint problems of the form (2.80), as shown by the values of η_{adj}.



(a) Comparing cG(1) solution and computed QoI(3.4) to the true values for linear example in §3.1.4.



(b) Comparing Crank-Nicolson solution and computed QoI (3.4) to the true values for linear example in §3.1.4.

Figure 3.2

Method	t_c	t_{LL}	t_L	t_R	e_Q	η	ρ_{eff}	n_{adj}
Taylor series	0.3626	–	–	–	-3.267e-4	-3.269e-4	1.000	2
Secant	0.3626	–	0.35	0.375	-3.267e-4	-3.267e-4	1.000	6
Inverse quad.	0.3626	0.325	0.35	0.375	-3.267e-4	-3.267e-4	1.000	7

Table 3.1: Results of the different methods on the linear example in §3.1.4 using cG(1) with 40 elements.

Nonlinear problem

Next we consider the nonlinear initial value problem

$$\dot{u}(t) = \sin(2\pi u(t)),\ t \in (0,1], \quad u(0) = \frac{1}{4}.$$

Method	t_c	t_{LL}	t_L	t_R	e_Q	η	ρ_{eff}	n_{adj}
Taylor series	0.3663	–	–	–	-4.017e-3	-4.056e-3	1.010	2
Secant	0.3663	–	0.35	0.4	-4.017e-3	-4.017e-3	1.000	7
Inverse quad.	0.3663	0.3	0.35	0.4	-4.017e-3	-4.017e-3	1.000	7

Table 3.2: Results of the different methods on the linear example in §3.1.4 using Crank-Nicolson with 21 nodes.

The analytic solution to this problem is

$$u(t) = \frac{1}{\pi}\arctan(e^{2\pi t}).$$

Let $R = 0.4$ and $G(u;t) = u(t)$. The true QoI is

$$t_t = Q(u) = \min_{t\in[0,1]} \arg(u(t) = 0.4) = \frac{\ln(\tan(0.4\pi))}{2\pi}.$$

Here, the terms in (3.26) are

$$\psi = 1,\ f(u,t) = \sin(2\pi u),\ \nabla_u f(u,t) = 2\pi\cos(2\pi u),$$

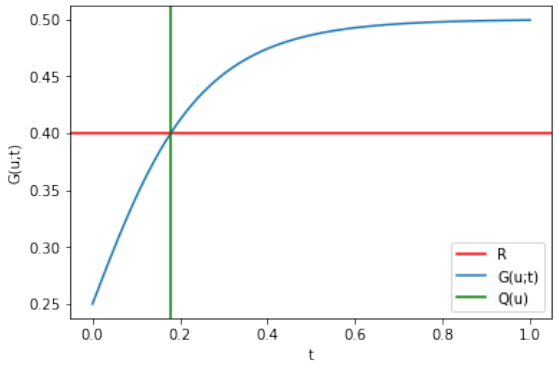
so the data needed for (3.27), (3.28), and (3.36) are

$$\psi_1 = -1,\ \psi_2 = 2\pi\cos(2\pi R),\ \psi_3 = 1.$$

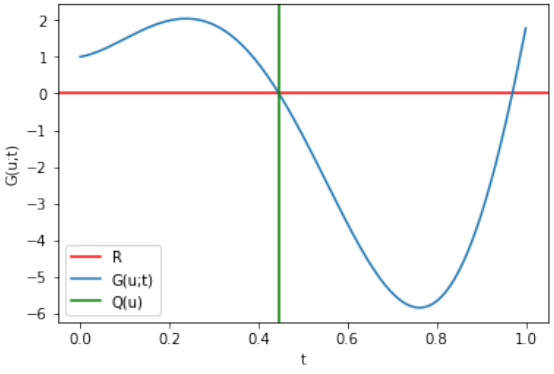
Figure 3.3a shows the true values of the linear functional $G(u;t)$ as well as the event in question and the true QoI. The values in Tables 3.3 and 3.4 indicate that all three methods are fairly accurate. The two iterative methods again require more adjoint equations to be solved.

Method	t_c	t_{LL}	t_L	t_R	e_Q	η	ρ_{eff}	n_{adj}
Taylor series	0.1790	–	–	–	-1.087e-4	-1.086e-4	1.000	2
Secant	0.1790	–	0.175	0.2	-1.087e-4	-1.087e-4	1.000	6
Inverse quad.	0.1790	0.15	0.175	0.2	-1.087e-4	-1.087e-4	1.000	6

Table 3.3: Results for nonlinear example in §3.1.4 using the different methods on cG(1) solution with 40 elements.



(a) Chosen value of R, true data $S(u(t))$, and true QoI for nonlinear example in §3.1.4.



(b) Chosen value of R, true data $S(u(t))$, and true QoI for linear system example in §3.1.4

Figure 3.3

Method	t_c	t_{LL}	t_L	t_R	e_Q	η	ρ_{eff}	n_{adj}
Taylor series	0.1810	–	–	–	-2.156e-3	-2.141e-3	1.007	2
Secant	0.1810	–	0.15	0.2	-2.156e-3	-2.144e-3	1.001	7
Inverse quad.	0.1810	0.1	0.15	0.2	-2.156e-3	-2.144e-3	1.001	7

Table 3.4: Results for nonlinear example in §3.1.4 using the different methods on Crank-Nicolson solution with 21 nodes.

Linear system

We consider the two dimensional system $\dot{u} + A(t)u = 0$, with matrix

$$A(t) = \begin{pmatrix} 1 + 9\cos^2(6t) - 6\sin(12t) & -12\cos^2(6t) - 9/2\sin(12t) \\ 12\sin^2(6t) - 9/2\sin(12t) & 1 + 9\sin^2(6t) + 6\sin(12t) \end{pmatrix}$$

over time domain $t \in (0, 1]$ with initial conditions $u_1(0) = u_2(0) = 1$. The analytic solution to this problem is

$$\begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = \begin{pmatrix} 3/5\exp(2t)(\cos(6t) + 2\sin(6t)) - 1/5\exp(-13t)(\sin(6t) - 2\cos(6t)) \\ 3/5\exp(2t)(2\cos(6t) - \sin(6t)) - 1/5\exp(-13t)(\cos(6t) + 2\sin(6t)) \end{pmatrix}.$$

Set $R = 0$ and $G(u;t) = u_1(t)$ in order to analyze the first component. The true quantity of interest is

$$t_t := Q(u) = 0.446255366908554$$

The parameters needed for (3.26) are

$$\psi = (1,0)^\top,\ f(u,t) = -A(t)u,\ \nabla_u f(u,t) = -A(t).$$

For (3.27), (3.28), and (3.36) the values needed are

$$\psi_1 = -(1,0)^\top, \psi_2 = (1 + 9\cos^2(6t_c) - 6\sin(12t_c),\ -12\cos^2(6t_c) - \frac{9}{2}\sin(12t_c))^\top,$$
$$\psi_3 = (1,0)^\top.$$

The true solution and QoI are shown in Figure 3.3b. Tables 3.5 and 3.6 show the results for cG(1) and Crank-Nicolson respectively. Again, all methods are accurate using either numerical method. The two iterative methods require many more adjoint problems to be solved than the Taylor series method without any increase in accuracy.

Method	t_c	t_{LL}	t_L	t_R	e_Q	η	ρ_{eff}	n_{adj}
Taylor series	0.4463	–	–	–	-1.323e-4	-1.322e-4	0.999	2
Secant method	0.4463	–	0.425	0.45	-1.323e-4	-1.323e-4	1.000	6
Inverse quad.	0.4463	0.4	0.425	0.45	-1.323e-4	-1.323e-4	1.000	8

Table 3.5: Results of the different methods on linear system example in §3.1.4 using cG(1) with 40 elements.

Table 3.6: Results of the different methods on linear system example in §3.1.4 using Crank-Nicolson with 21 nodes.

Method	t_c	t_{LL}	t_L	t_R	e_Q	η	ρ_{eff}	n_{adj}
Taylor series	0.4462	–	–	–	2.675e-5	2.675e-5	1.000	2
Secant	0.4462	–	0.4	0.45	2.675e-5	2.675e-5	1.000	6
Inverse quad.	0.4462	0.35	0.4	0.45	2.675e-5	2.675e-5	1.000	8

Harmonic oscillator

Consider the harmonic oscillator

$$\ddot{\omega} = -\frac{k}{m}\omega - \frac{c}{m}\dot{\omega} + \frac{F_0}{m}\cos(\gamma t + \theta_d),\ t \in (0, 2], \quad \omega(0) = 5,\ \dot{\omega}(0) = 0.$$

with

$$k = 50,\ m = 0.25,\ c = 1,\ F_0 = 50,\ \theta_d = 0,\ \gamma = 10.$$

Rewriting as a system of first-order ODEs, $\dot{u} + Au = h(t)$, gives

$$\begin{pmatrix} \dot{u_1}(t) \\ \dot{u_2}(t) \end{pmatrix} + \begin{pmatrix} 0 & -1 \\ 200 & 4 \end{pmatrix} \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = \begin{pmatrix} 0 \\ 200\cos(10t) \end{pmatrix}.$$

Set $R = 0$ and $G(u; t) = u_1(t)$ in order to observe when the oscillator first reaches the origin. The true solution in [6] is used to determine

$$t_t := Q(\omega) = 0.14034864129073557.$$

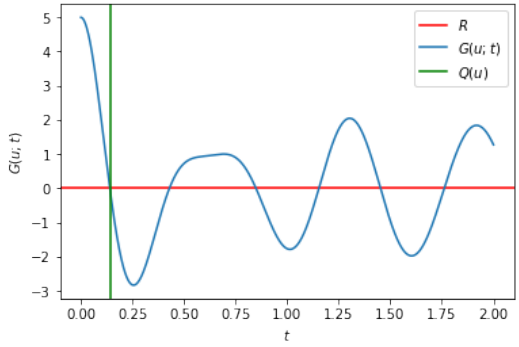
Here for (3.26), the values needed are

$$\psi = (1, 0)^\top,\ f(u, t) = -Au + h(t),\ \nabla_u f(u, t) = -A.$$

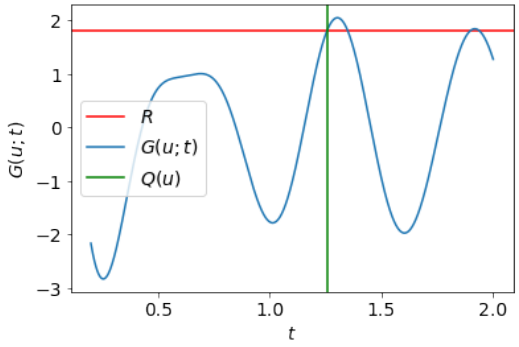
To compute (3.27), (3.28), and (3.36), let

$$\psi_1 = -(1, 0)^\top,\ \psi_2 = (0, 1)^\top,\ \psi_3 = (1, 0)^\top.$$

The true data $G(u; t)$ and QoI are given in Figure 3.4a and the results using cG(1) and Crank-Nicolson method are provided in Tables 3.7 and 3.8 respectively. All methods using either numerical method give effectivity ratios close to one. The two iterative methods require more adjoint problems to be solved than the Taylor series estimate, but they do lead to a slightly more accurate error estimate.



(a) Chosen value of R, true data $S(u(t))$, and true QoI $Q(u)$ for oscillator example 3.1.4.



(b) Chosen value of R, true data $G(u;t)$, and true QoI $Q(u)$ for oscillator example in §3.1.4.

Figure 3.4

Method	t_c	t_{LL}	t_L	t_R	e_Q	η	ρ_{eff}	n_{adj}
Taylor series	0.1447	–	–	–	-4.440e-3	-4.449e-3	1.011	2
Secant method	0.1447	–	0.1	0.15	-4.440e-3	-4.440e-3	1.000	7
Inverse quad.	0.1447	0.05	0.1	0.15	-4.440e-3	-4.440e-3	1.000	8

Table 3.7: Results of the different methods on the oscillator example in §3.1.4 using cG(1) with 40 elements.

Method	t_c	t_{LL}	t_L	t_R	e_Q	η	ρ_{eff}	n_{adj}
Taylor series	0.1575	–	–	–	-1.715-02	-1.816e-2	1.059	2
Secant method	0.1575	–	0.1	0.2	-1.715-02	-1.715e-2	0.999	8
Inverse quad.	0.1575	0.0	0.1	0.2	-1.715-02	-1.715e-2	0.999	10

Table 3.8: Results of the different methods on the oscillator example in §3.1.4 using Crank-Nicolson with 21 nodes.

Harmonic oscillator: Effect of the choice of interval

We consider the same equation and function as in §3.1.4, except over the time interval $t \in (0.2, 2]$ and with $R = 1.8$. In effect, we are choosing $\hat{t} = 0.2$ in (3.1) to obtain the *second* occurrence of $G(u;t) = 1.8$.

Applying the secant method to the true solution results in the true QoI,

$$t_t = 1.2558594599461572.$$

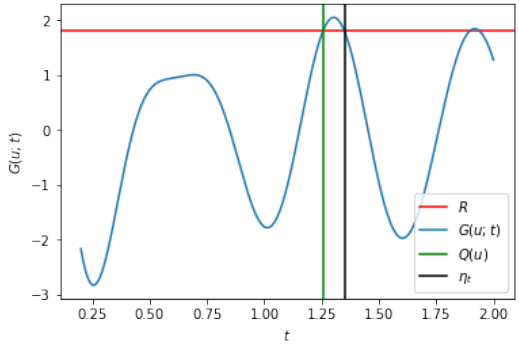
Since this problem has the same ODE and functional S as in §3.1.4, the parameters and steps laid out in that section can be used to obtain the error estimates.

The true functional and QoI are shown in Figure 3.4b and the results when using the different methods in Tables 3.9 and 3.10. The Taylor series method is slightly less accurate compared to the iterative methods when using the cG(1) method. This is due to the size of the second derivative of the functional near the event, leading to a larger absolute value of the remainder in (3.22). Since the error estimate (3.26) neglects this remainder, if its absolute value is too large the estimate will not be accurate. Examples in §3.1.4 take a further look into this effect.

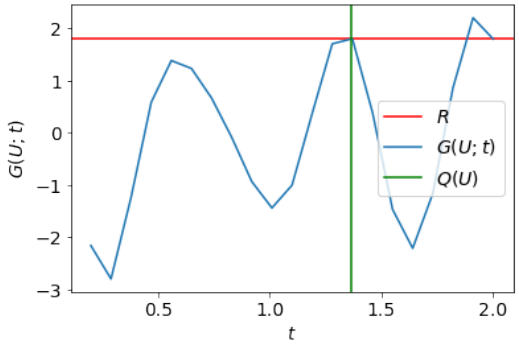
In this example, both the Taylor series and iterative methods are poor for the Crank-Nicolson method. This is due to the low accuracy of the numerical solution as illustrated in Figure 3.5b. The potential inaccuracy of the Taylor series estimate under these circumstances is discussed in Remark 2. The root-finding methods are converging to the *second* time the event occurs (which is 1.3237), rather than the first. Because of the small difference in time between the locations of the two roots (see Figure 3.5a), the proximity of the second root to the numerical QoI, and the size of the numerical time step, both roots are contained within the initial interval over which the iterative methods are applied. It is therefore possible for the iterative methods to converge to the larger of the two roots.

Harmonic oscillator: Effect of the choice of R

Again consider the harmonic oscillator of §3.1.4, and estimate the error of the QoI (3.4) with several different values of R, increasing R until it is very close to the maximum of the true data. The maximum value of the true data is approximately 2.05015. Results



(a) Figure detailing issue with iterative methods for oscillator examples in §3.1.4 and §3.1.4 when the numerical solution is not accurate near the event. The iterative methods result in $t^* = \eta_{it}$, which is the second occurrence of the event rather than the first. This figure specifically details the case when $R = 2$.



(b) Numerical values for oscillator example in §3.1.4 when using Crank-Nicolson method with 21 nodes.

Figure 3.5

Method	t_c	t_{LL}	t_L	t_R	e_Q	η	ρ_{eff}	n_{adj}
Taylor series	1.2637	–	–	–	-7.887e-3	-8.623e-3	1.093	2
Secant	1.2637	–	1.235	1.37	-7.887e-3	-7.887e-3	0.999	8
Inverse quad.	1.2637	1.19	1.235	1.37	-7.887e-3	-7.887e-3	0.999	9

Table 3.9: Results of the different methods on oscillator example in §3.1.4 using cG(1) with 40 elements.

Method	t_c	t_{LL}	t_L	t_R	e_Q	η	ρ_{eff}	n_{adj}
Taylor series	1.3674	–	–	–	-1.116e-1	-1.542e-2	0.138	2
Secant method	1.3674	–	1.28	1.37	-1.116e-1	-1.746e-2	0.156	8
Inverse quad.	1.3674	1.19	1.28	1.37	-1.116e-1	-1.746e-2	0.156	10

Table 3.10: Results of the different methods on oscillator example in §3.1.4 using Crank-Nicolson with 21 nodes.

are provided in Tables 3.11, 3.12 and 3.13 for increasingly fine finite element meshes. The tables contain the effectivity ratios, ρ_{eff}, for each method and each value of R.

Notice that the iterative methods appear to be more sensitive to the accuracy of the numerical solution than the Taylor series method. In extreme cases, the iterative methods fail to converge. This occurs when a root-finding iteration falls outside of the domain of the IVP (1.6), i.e., if x_n the approximation to the root at the nth iteration, $x_n < 0$ or $x_n > T$. As the number of finite elements used to solve the ODE increases, the two iterative methods eventually recover their accuracy even when the threshold value is very close to an extremum. For the cases where the iterative methods are inaccurate, note that the root-finding schemes do *not* converge to the true QoI. Instead, the convergence is to the *second* occurrence of the event rather than the first (see Figure 3.5a).

The estimate derived from Taylor's theorem is generally more accurate for the less accurate numerical solutions, However, even when using an accurate numerical solution, the Taylor series approach becomes inaccurate when the curvature of G as a function of t is large near the threshold value. The remainder $\mathcal{R}_1(t_t, t_c)$ is one half of the second derivative of G with respect to t at some point between t_t and t_c. As the threshold value R moves closer to the local maximum, this derivative grows and the assumption that $\mathcal{R}_1(t_t, t_c)$ is small is no longer valid, resulting in an inaccurate estimate. The iterative methods do not depend on the values of the second derivative of the solution and those methods are able to produce accurate error estimates provided the numerical solution is sufficiently accurate near the event.

Method	R=1.95	R=2.0	R=2.01	R=2.02	R=2.03	R=2.04	R=2.05
Taylor series	1.061	1.095	1.251	1.603	3.470	-1.137	0.427
Secant	0.999	-11.305	-4.952	-2.650	-1.405	1.000	fail
Inverse quad.	0.999	-11.305	-4.952	-2.650	-1.405	fail	fail

Table 3.11: Effectivity ratio for the different methods for varying values of R on oscillator example in §3.1.4 using cG(1) with 40 elements.

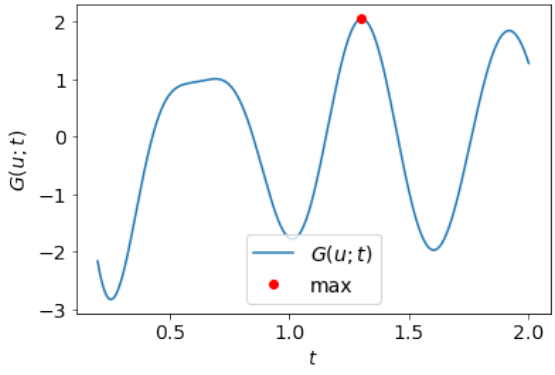


Figure 3.6: True data for oscillator example in §3.1.4, showing max value of ≈ 2.05015.

Method	R=1.95	R=2.0	R=2.01	R=2.02	R=2.03	R=2.04	R=2.05
Taylor series	1.033	0.999	1.043	1.100	1.179	1.283	0.758
Secant	1.000	0.999	0.999	0.999	-6.545	-4.520	3.133
Inverse quad.	1.000	0.999	0.999	0.999	-6.545	-4.520	3.133

Table 3.12: Effectivity ratio for the different methods for varying values of R on oscillator example in §3.1.4 using cG(1) with 60 elements.

Method	R=1.95	R=2.0	R=2.01	R=2.02	R=2.03	R=2.04	R=2.05
Taylor series	1.017	1.001	1.019	1.100	1.039	0.998	0.588
Secant	0.999	0.999	0.999	1.000	0.999	0.999	0.999
Inverse quad.	0.999	0.999	0.999	1.000	0.999	0.999	0.999

Table 3.13: Effectivity ratio for the different methods for varying values of R on example in §3.1.4 using cG(1) with 100 elements.

One dimensional heat equation

We consider the one dimensional heat equation with boundary and initial conditions

$$\begin{aligned} &u_t(x,t) = u_{xx}(x,t) + 3e^t \sin(\pi x), \quad (x,t) \in (0,1) \times (0,1], \\ &u(x,0) = 0, \quad x \in (0,1), \\ &u(0,t) = 0,\ u(1,t) = 0, \quad t \in (0,1]. \end{aligned} \tag{3.42}$$

This section analyzes the system of ordinary differential equations that arises from a spatial discretization of (3.42) using a central-difference method. In particular using a

uniform partition of the spatial interval $[0, 1]$ with 22 nodes:

$$\{0 = x_0 < x_1 < \cdots < x_{21} = 1\}.$$

Since boundary values are specified, this semi-discretization leads to a system of 20 first-order ODEs of the form $\dot{u}(t) = Au(t) + k(t)$, where $h = \frac{1}{21}$ and

$$A = \frac{1}{h^2}\begin{pmatrix} -2 & 1 & 0 & \cdots & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & -2 & 1 \\ 0 & \cdots & \cdots & 0 & 1 & -2 \end{pmatrix}, \quad k(t) = \begin{pmatrix} 3e^t \sin(\pi x_1) \\ 3e^t \sin(\pi x_2) \\ 3e^t \sin(\pi x_3) \\ \vdots \\ 3e^t \sin(\pi x_{19}) \\ 3e^t \sin(\pi x_{20}) \end{pmatrix}$$

Since this problem will only analyze the semi-discrete system and not the full PDE, a reference solution is obtained using an accurate time-integrator (SciPy's solve_ivp) using an absolute tolerance of 10^{-15}. Let $R = 0.33$ and $G(u; t) = \frac{1}{20}\sum_{i=1}^{20} u_i(t)$ in order to analyze the discrete average of the solution over the spatial domain at a time t. This library function also has the capability of tracking when specified events occur, which is used to obtain a reference for the true QoI,

$$t_t = 0.5834435609935992.$$

For this problem, the parameters in (3.26) are

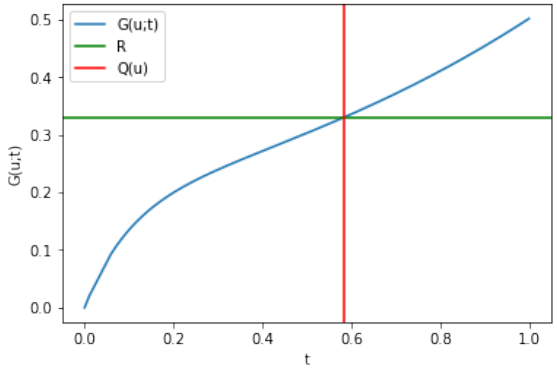
$$\psi = \frac{1}{20}(1, 1, \ldots, 1)^\top,\ f(u, t) = Au + k(t),\ \nabla_u f(u, t) = A.$$

For (3.27), (3.28), and (3.36), set

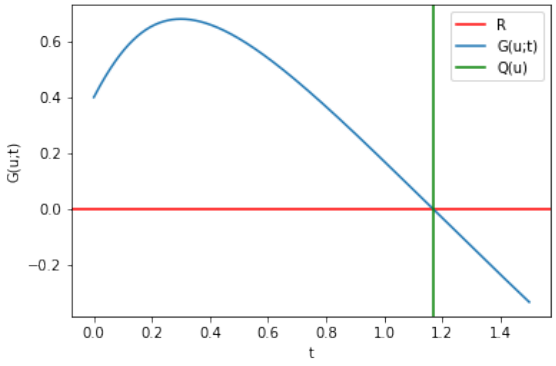
$$\psi_1 = -\frac{1}{20}(1, 1, \ldots, 1)^\top,\ \psi_2 = \frac{1}{20h^2}(-1, 0, \ldots, 0, -1)^\top,\ \psi_3 = \frac{1}{20}(1, 1, \ldots, 1)^\top.$$

The true solution and QoI are shown in Figure 3.7a and the results when using cG(1) or Crank-Nicolson methods are shown in Tables 3.14 and 3.15 respectively. All methods

are accurate using either numerical method. The two iterative methods require more adjoint problems to be solved than the Taylor series estimate without any noticeable increase in accuracy.



(a) Chosen value of R, true data $G(u;t)$, and true QoI for example of the heat equation in §3.1.4.



(b) Chosen value of R, true data $G(u;t)$, and true QoI for example of the two-body problem in §3.1.4.

Figure 3.7

Method	t_c	t_{LL}	t_L	t_R	e_Q	η	ρ_{eff}	n_{adj}
Taylor series	0.5834	–	–	–	6.157e-5	6.151e-5	0.999	2
Secant	0.5834	–	0.575	0.6	6.157e-5	6.150e-5	0.999	6
Inverse quad.	0.5834	0.55	0.575	0.6	6.157e-5	6.150e-5	0.999	7

Table 3.14: Results of the different methods on the heat equation example in §3.1.4 using cG(1) with 40 elements.

Method	t_c	t_{LL}	t_L	t_R	e_Q	η	ρ_{eff}	n_{adj}
Taylor series	0.5830	–	–	–	4.457e-4	4.457e-4	1.000	2
Secant	0.5830	–	0.55	0.6	4.457e-4	4.456e-4	0.999	6
Inverse quad.	0.5830	0.5	0.55	0.6	4.457e-4	4.456e-4	0.999	7

Table 3.15: Results of the different methods on the heat equation example in §3.1.4 using Crank-Nicolson with 21 nodes.

Two body problem

We consider the two body problem

$$\left.\begin{aligned} \dot{u}_1 &= u_3, \\ \dot{u}_2 &= u_4, \\ \dot{u}_3 &= \frac{-u_1}{(u_1^2+u_2^2)^{3/2}}, \\ \dot{u}_4 &= \frac{-u_2}{(u_1^2+u_2^2)^{3/2}}, \end{aligned}\right\} \quad t \in (0, 1.5], \quad u(0) = (0.4, 0, 0, 2.0)^\top, \tag{3.43}$$

which models a small body orbiting a much larger body. Here u_1, u_2 are the planar spatial coordinates of the orbiting body relative to the larger body, and u_3, u_4 are the respective velocities. The initial conditions are chosen so that the analytic solution is [26]

$$u = \left(\cos(\tau) - 0.6, 0.8\sin(\tau), \frac{-\sin(\tau)}{1-0.6\cos(\tau)}, \frac{0.8\cos(\tau)}{1-0.6\cos(\tau)}\right)^\top,$$

where τ solves $\tau - 0.6\sin(\tau) = t$. Let $R = 0$ and $G(u;t) = u_1(t) + u_2(t)$. The true QoI can be found exactly:

$$t_t = Q(u) = \cos^{-1}((15-16\sqrt{2})/41) - 0.6\sin\left(\cos^{-1}((15-16\sqrt{2})/41)\right).$$

The values needed to compute (3.26) are

$$\psi = (1,1,0,0)^\top, \; f(u,t) = \left(u_3, u_4, \frac{-u_1}{(u_1^2+u_2^2)^{3/2}}, \frac{-u_2}{(u_1^2+u_2^2)^{3/2}}\right)^\top,$$

and

$$\nabla_u f(u,t) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{2u_1^2-u_2}{(u_1^2+u_2^2)^{5/2}} & \frac{3u_1u_2}{(u_1^2+u_2^2)^{5/2}} & 0 & 0 \\ \frac{3u_1u_2}{(u_1^2+u_2^2)^{5/2}} & \frac{2u_1^2-u_2}{(u_1^2+u_2^2)^{5/2}} & 0 & 0 \end{pmatrix}.$$

For (3.27), (3.28), and (3.36), the data needed are

$$\psi_1 = (-1,-1,0,0)^\top, \; \psi_2 = (0,0,1,1)^\top \; \psi_3 = (1,1,0,0)^\top.$$

The true data $G(u;t)$ and QoI are shown in Figure 3.7b and the results using the cG(1) and Crank-Nicolson method appear in Tables 3.16 and 3.17 respectively. All methods have larger error than in other examples so far due to the non-linear nature of (3.43). However the error estimates are accurate using either numerical method; each with an effectivity ratio close to one.

Method	t_c	t_{LL}	t_L	t_R	e_Q	η	ρ_{eff}	n_{adj}
Taylor series	1.1601	–	–	–	8.262e-3	8.287e-3	1.003	2
Secant	1.1601	–	1.125	1.1625	8.262e-3	8.287e-3	1.003	5
Inverse quad.	1.1601	1.0875	1.125	1.1625	8.262e-3	8.287e-3	1.003	6

Table 3.16: Results of the different methods on the two-body example in §3.1.4 using cG(1) with 40 elements.

Method	t_c	t_{LL}	t_L	t_R	e_Q	η	ρ_{eff}	n_{adj}
Taylor series	1.2091	–	–	–	-4.068e-2	-4.078e-2	1.002	2
Secant	1.2091	–	1.2	1.275	-4.068e-2	-4.077e-2	1.002	5
Inverse quad.	1.2091	1.125	1.2	1.275	-4.068e-2	-4.077e-2	1.002	6

Table 3.17: Results of the different methods on the two-body example in §3.1.4 using Crank-Nicolson with 21 nodes.

Logistic Equation

Consider the Logistic equation

$$\dot{u} = ku\left(1 - \frac{u}{K}\right),\ t \in (0, 20], \quad u(0) = \frac{1}{2}, \tag{3.44}$$

where $k = 0.25$ and $K = 1$. The analytic solution is,

$$u(t) = \frac{K\,u(0)}{u(0) + (K - u(0))e^{-kt}} = \frac{1}{1 + 3e^{-0.25t}}. \tag{3.45}$$

Let $G(u;t) = u(t)$ and consider several threshold values,
$R \in \{0.55, 0.8, 0.9, 0.94, 0.98, 0.99, 0.995\}$. The values needed for (3.26) are

$$\psi = 1,\ f(u,t) = ku(1 - \frac{u}{k}),\ \nabla_u f(u,t) = k - \frac{2k}{K}u,$$

so the data needed for (3.27), (3.28), and (3.36) are

$$\psi_1 = -1,\ \psi_2 = k - \frac{2k}{K}R,\ \psi_3 = 1.$$

The numerical solution is computed using the cG(1) method with five elements. Figure 3.8 shows the true functional and QoI for a chosen threshold value. Table 3.18 shows the true error in the QoI and the effectivity ratio for each method as the threshold value increases. As the error in the QoI increases, the Taylor series method loses accuracy, presumably since the remainder terms are no longer negligible, despite the fact the second derivatives with respect to t are small. However, the iterative methods are accurate even when the true error is large.

Collisionless Vlasov Equation

We consider the homogeneous Vlasov problem

$$\begin{cases} \dot{u}(x,p,t) + p\nabla_x \cdot u(x,p,t) &= 0, & \Omega_x \times \Omega_p \times (0, 0.25], \\ u(x,p,0) &= u_0(x,p), & \Omega_x \times \Omega_p, \\ u(x,p,t) &= 0, & \partial\Omega_x \times \partial\Omega_p \times (0, 0.25]. \end{cases} \tag{3.46}$$

	R=0.55	R=0.8	R=0.9	R=0.94	R=0.98	R=0.99	R=0.995
e_Q	-0.090	-0.117	-0.166	0.194	0.829	0.610	1.513
Taylor series	1.001	1.021	1.041	0.957	0.902	0.919	0.830
Secant	0.999	0.987	0.977	1.023	1.007	1.011	1.005
Inverse quad.	0.999	0.987	0.977	1.023	1.007	1.011	1.005

Table 3.18: Error in QoI and effectivity ratio of the different methods for varying values of R on the logistic example in §3.1.4 using cG(1) with 5 elements.

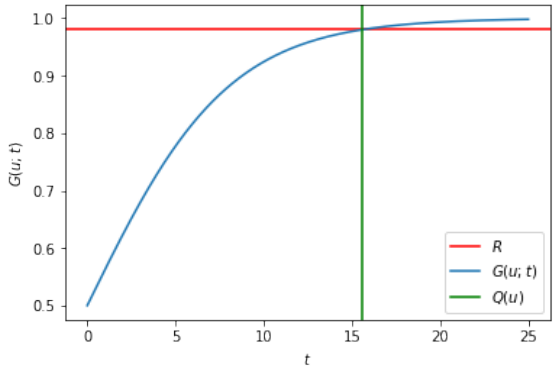


Figure 3.8: True values of functional and QoI for example of the Logistic Equation in §3.1.4, when $R = 0.94$

We use the domains $\Omega_x = (0,1) = \Omega_p$. The initial condition is given by the bump-function

$$u_0(x,p) = \begin{cases} 200000(x-0.3)^2(x-0.7)^2(p-0.3)^2(p-0.7)^2, & \text{for} \quad 0.3 \leq x,p \leq 0.7 \\ 0 & else \end{cases} \tag{3.47}$$

Recall that when solving (3.46) we view the pair of variables (x,p) as the "spatial" variable over the "spatial domain" $\Omega = \Omega_x \times \Omega_p$. The problem is well-posed over the given time interval because the data never reaches the space-momentum boundaries.

We define the non-standard QoI (3.4) by setting $\psi(x)$ from (3.6) to be

$$\psi(x,p) = \begin{cases} 200000(x-0.3)^2(x-0.7)^2(p-0.3)^2(p-0.7)^2, & \text{for} \quad 0.3 \leq x,p \leq 0.7 \\ 0 & else. \end{cases} \tag{3.48}$$

For (3.1) choose the threshold value $R = 0.005$ and $\hat{t} = 0$ to obtain the first occurrence of the event $(\psi, u(t)) = R$. The error approximation (3.29) for this homogeneous problem is

$$t_t - t_c \approx \frac{(-\psi, e(t_c))}{(p\psi_x, U(t_c)) + (p\psi_x, e(t_c))}. \tag{3.49}$$

The terms containing the function $e = u - U$ are computed using Theorem 7 which requires the solutions to associated adjoint problems. The adjoint equation for (3.46) is

$$\phi_t + p \cdot \nabla_x \phi = 0 \quad \text{in} \quad \Omega_x \times \Omega_p \times [0, t_c), \tag{3.50}$$

with zero Dirichlet boundary conditions on $\Omega_x \times \Omega_p$. The initial condition for the adjoint problem depends the term on (3.49). To compute $(-\psi, e(t_c))$ we use the initial condition $\phi(x,p,0) = -\psi$. To obtain the term $(p\psi_x, e(t_c))$, we solve the adjoint problem with initial condition $\phi(x,p,0) = -p\psi_x$. With the solutions to these adjoint problems, the terms are computed via (2.119).

A reference solution is obtained using a continuous Galerkin method with quadratic polynomials over a 200×200 spatial mesh and time-steps are taken using the Crank-Nicolson method with 100 time sub-intervals. Figure 3.9 shows the functional $G(u;t) = (\psi, u(t))$ over time with the reference solution u. The figure also depicts the threshold value R and reference value of the QoI (3.4) which is $t_t := Q(u) = 0.18586362344$.

With this example, we explore how different discretizations for the numerical and adjoint solutions effect the accuracy of the error approximation (3.49). We also look at the effectivity ratios for the individual computed error terms inside the approximation

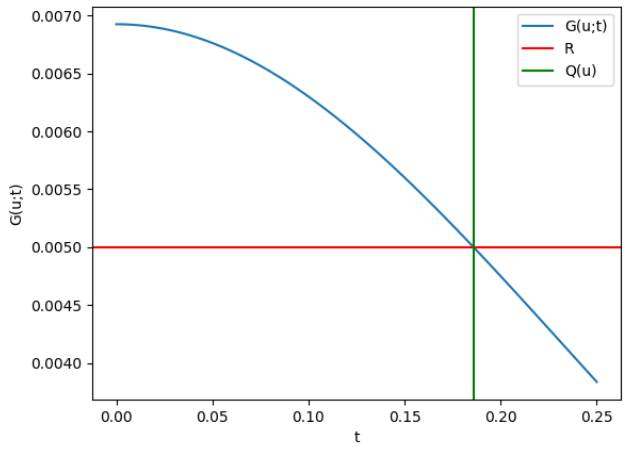


Figure 3.9: Functional from Vlasov example in §3.1.4 with true solution.

in particular showing the size of the error in initial value of the numerical solution. For ease of notation, let $E_1 \approx (-\psi, e(t_c))$ and $E_2 \approx (p\psi_x, e(t_c))$ be the values computed via the solution of the associated adjoint problems. In the Tables below, "E_1 initial" denotes the value of $(\phi(0), e(0))$ in (2.119) when computing E_1 and "ρ_{eff} of E_1" denotes the effectivity ratio for the estimation E_1.

In all cases numerical solutions are computed using Crank-Nicolson for the time steps and a linear space discretization, while adjoint solutions are computed using Crank-Nicolson in time and quadratic space discretizations. For Tables 3.19, 3.20, and 3.21, the numerical solution uses a mesh created from a 10-by-10 grid over $\Omega = \Omega_x \times \Omega_p$ and a time grid with 10 sub-intervals. In Tables 3.22, 3.23, and 3.24, the numerical solution uses a mesh created from a 20-by-20 grid over $\Omega = \Omega_x \times \Omega_p$ and a time grid with 10 sub-intervals. In both cases, the adjoint problem is solved over a time grid with 40 sub-intervals and different spatial meshes in order to determine any effect.

The estimates are accurate in all experiments with effectivity ratios close to 1. Tables 3.21 and 3.24 show that we accuracy does not change much as we refine the adjoint solution's spatial mesh, but accuracy of the error estimate increases as U becomes more

adj_N_x	30	40	60
E_1	-3.40E-4	-3.40E-4	-3.40E-5
E_1 initial	-4.52E-4	-4.52E-4	-4.52E-5
ρ_{eff} of E_1	0.9999	1.0000	1.0000

Table 3.19: Values and effectivity ratios for E_1 from Vlasov example using $N_x = 10 = N_t$ and $adj\_N_t = 40$ with different spatial discretizations for the adjoint.

adj_N_x	30	40	60
E_2	-3.32E-3	-3.32E-3	-3.32E-3
E_2 initial	-2.98E-3	-2.98E-3	-2.98E-3
ρ_{eff} of E_2	1.0000	1.0001	1.0002

Table 3.20: Values and effectivity ratios for E_2 from Vlasov example using $N_x = 10 = N_t$ and $adj\_N_t = 40$ with different spatial discretizations for the adjoint.

$t_t - t_c$	0.01989	0.01989	0.01989
η	0.020417	0.020418	0.020419
ρ_{eff}	1.0263	1.0263	1.0263

Table 3.21: True errors, computed errors, and effectivity ratios for NSQoI in Vlasov example using $N_x = 10 = N_t$ and $adj\_N_t = 40$ with different spatial discretizations for the adjoint. The numerical QoI is $t_c = 0.16597052$.

adj_N_x	30	40	60
E_1	-1.01E-4	-1.01E-4	-1.01E-4
E_1 initial	-1.14E-4	-1.14E-4	-1.14E-4
E_1 effect.	0.997	0.999	0.999

Table 3.22: Values and effectivity ratios for E_1 from Vlasov example using $N_x = 20, N_t = 10$ and $adj\_N_t = 40$ with different spatial discretizations for the adjoint.

accurate.

adj_N_x	30	40	60
E_2	-8.50E-4	-8.46E-4	-8.46E-4
E_2 initial	-8.57E-4	-8.57E-4	-8.57E-4
E_2 effect.	1.0045	0.999	1.000

Table 3.23: Values and effectivity ratios for E_2 from Vlasov example using $N_x = 20, N_t = 10$ and $adj\_N_t = 40$ with different spatial discretizations for the adjoint.

adj_N_x	30	40	60
$t_t - t_c$	0.005824	0.005824	0.005824
η	0.005844	0.005858	0.005859
ρ_{eff}	1.0033	1.0057	1.0059

Table 3.24: True errors, computed errors, and effectivity ratios for NSQoI in Vlasov example using $N_x = 20, N_t = 10$ and $adj\_N_t = 40$ with different spatial discretizations for the adjoint. The numerical QoI is $t_c = 0.180039009533$.

Chapter 4

Uncertainty Quantification: CDF Bound and MLMC Algorithm

This chapter utilizes the adjoint-based error analysis from §2.4 and Chapter 3 to derive two methods of uncertainty quantification for QoIs related to differential equations with random parameters. In §4.1 we derive an upper-bound on the error in a computed MC estimator of the CDF. To create our adaptive MLMC algorithm, we first describe two adaptive refinement methods and modify them to work in the context of the MLMC method in §4.2.1. We then provide a description and pseudo-code for our adaptive MLMC algorithm in §4.2.2. For both UQ methods we detail the need for the *a posteriori* error estimates and provide numerical experiments.

1 A Posteriori Error Analysis of the Cumulative Density Function

If the differential equation (either (1.1), (1.4), or (1.6)) depends on a random parameter w, then the solution u and the QoI, $Q(u;w)$, are random variables. As a random

variable, $Q(u;w)$ has a corresponding cumulative distribution function (CDF),

$$F(t) = P(\{w : Q(u;w) \leq t\}) = P(Q \leq t).$$

An approximation to the CDF is computed using the Monte Carlo method with a finite number of numerically computed sample values $\{\hat{Q}(U^{(n)}, w^{(n)}) = \hat{Q}^{(n)}\}_{n=1}^{N}$,

$$\hat{F}_N(t) = \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}(\hat{Q}^{(n)} \leq t), \tag{4.1}$$

where $\mathbb{1}$ is the indicator function. A nominal sample distribution is computed using exact values of the QoI,

$$F_N(t) = \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}(Q^{(n)} \leq t). \tag{4.2}$$

An estimate of the error in an approximate distribution of the non-standard QoI (3.4) is computed for two examples in §4.1.1. The estimate takes into account error contributions due to finite sampling and errors arising from the discretization of the ODE. The expressions (2.20) and (4.2) decompose the error in to sampling and discretization contributions,

$$F(t) - \hat{F}_N(t) = (F(t) - F_N(t)) + (F_N(t) - \hat{F}_N(t)).$$

This decomposition is used to derive the following error bound.

Theorem 10. *For* $0 < \varepsilon < 1$,

$$\begin{aligned}\left|F(t) - \hat{F}_N(t)\right| \leq & \left(\frac{\hat{F}_N(t)\left(1 - \hat{F}_N(t)\right)}{N\varepsilon}\right)^{1/2}, \\ & + \left(\frac{1}{N} + \frac{1}{N\varepsilon^{1/2}}\right)\left|\sum_{n=1}^{N}\left(\mathbb{1}(\hat{Q}^{(n)} - \left|e_Q^{(n)}\right| \leq t \leq \hat{Q}^{(n)} + \left|e_Q^{(n)}\right|)\right)\right|, \\ & + \frac{2}{(2N\varepsilon)^{3/4}}\end{aligned} \tag{4.3}$$

with probability greater than or equal to $1 - 2\varepsilon + \varepsilon^2$, *where* $e_Q^{(n)} = Q^{(n)} - \hat{Q}^{(n)}$ *is the error in a numerically computed sample of the QoI.*

Proof. We decompose the error as

$$\left|F(t)-\hat{F}_N(t)\right| \leq |F(t)-F_N(t)| + \left|F_N(t)-\hat{F}_N(t)\right| = I + II. \tag{4.4}$$

Focusing on the term $II = \left|F_N(t)-\hat{F}_N(t)\right| = \left|\hat{F}_N(t)-F_N(t)\right|$,

$$\begin{aligned}
II &= \left|\frac{1}{N}\sum_{n=1}^{N}\left(\mathbb{1}(\hat{Q}^{(n)}\leq t)-\mathbb{1}(Q^{(n)}\leq t)\right)\right|,\\
&= \left|\frac{1}{N}\sum_{n=1}^{N}\left(\mathbb{1}(\hat{Q}^{(n)}\leq t)-\mathbb{1}(\hat{Q}^{(n)}+e_Q^{(n)}\leq t)\right)\right|,\\
&= \left|\frac{1}{N}\sum_{\substack{n=1\\ e_Q^{(n)}\leq 0}}^{N}\left(\mathbb{1}(\hat{Q}^{(n)}-\left|e_Q^{(n)}\right|\leq t\leq \hat{Q}^{(n)})\right),\right.\\
&\qquad\left.+\frac{1}{N}\sum_{\substack{n=1\\ e_Q^{(n)}>0}}^{N}\left(\mathbb{1}(\hat{Q}^{(n)}\leq t\leq \hat{Q}^{(n)}+\left|e_Q^{(n)}\right|)\right)\right|,\\
&\leq \left|\frac{1}{N}\sum_{n=1}^{N}\left(\mathbb{1}(\hat{Q}^{(n)}-\left|e_Q^{(n)}\right|\leq t\leq \hat{Q}^{(n)})\right)+\frac{1}{N}\sum_{n=1}^{N}\left(\mathbb{1}(\hat{Q}^{(n)}\leq t\leq \hat{Q}^{(n)}+\left|e_Q^{(n)}\right|)\right)\right|,\\
&= \left|\frac{1}{N}\sum_{n=1}^{N}\left(\mathbb{1}(\hat{Q}^{(n)}-\left|e_Q^{(n)}\right|\leq t\leq \hat{Q}^{(n)}+\left|e_Q^{(n)}\right|)\right)\right|,
\end{aligned} \tag{4.5}$$

Now consider the term $I = |F(t)-F_N(t)|$. We start with the Chebyshev Inequality:

$$P\left(|F(t)-F_N(t)|\geq ks\right)\leq \frac{1}{k^2}$$

for any real number k, where s^2 is the variance of F_N given by [42, 58],

$$s^2 = \frac{F(t)\left(1-F(t)\right)}{N}.$$

Choosing $\varepsilon = \frac{1}{k^2}$ leads to

$$I = |F(t)-F_N(t)| \leq \left(\frac{F(t)\left(1-F(t)\right)}{N\varepsilon}\right)^{1/2}, \tag{4.6}$$

with a probability greater than $1-\varepsilon$. Now,

$$F(t)\left(1-F(t)\right) = F_N(t)\left(1-F_N(t)\right) + \left(F(t)-F_N(t)\right)\left(1-F(t)-F_N(t)\right). \quad (4.7)$$

Taking absolute values in (4.7), dividing by $N\varepsilon$, taking the square root, and using $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for any $a, b \geq 0$,

$$\left|\frac{F(t)\left(1-F(t)\right)}{N\varepsilon}\right|^{1/2} \leq \left|\frac{F_N(t)\left(1-F_N(t)\right)}{N\varepsilon}\right|^{1/2} + \left|\frac{\left(F(t)-F_N(t)\right)\left(1-F(t)-F_N(t)\right)}{N\varepsilon}\right|^{1/2} \quad (4.8)$$

Multiplying and dividing the second term on the right-hand side of (4.8) by $\sqrt{2}\delta$ and using the fact that $ab \leq \frac{a^2}{2} + \frac{b^2}{2}$,

$$\begin{aligned}\left|\frac{\left(F(t)-F_N(t)\right)\left(1-F(t)-F_N(t)\right)}{N\varepsilon}\right|^{1/2} &\leq \left|\delta^2\left(F(t)-F_N(t)\right)^2 + \frac{\left(1-F(t)-F_N(t)\right)^2}{4\delta^2 N^2\varepsilon^2}\right|^{1/2} \\ &\leq \delta\left|F(t)-F_N(t)\right| + \frac{1}{2\delta N\varepsilon},\end{aligned}$$

where we obtain the final line by observing that $\left(1-F(t)-F_N(t)\right)^2 \leq 1$. Substituting back into (4.8) and combining with (4.6),

$$I \leq \left(\frac{F_N(t)\left(1-F_N(t)\right)}{N\varepsilon}\right)^{1/2} + \delta\left|F(t)-F_N(t)\right| + \frac{1}{2\delta N\varepsilon}. \quad (4.9)$$

From [58], for any $\varepsilon > 0$ we have with a probability greater than $1-\varepsilon$,

$$I \leq \left(\frac{\log(\varepsilon^{-1})}{2N}\right)^{1/2} \leq \left(\frac{1}{2N\varepsilon}\right)^{1/2}, \quad (4.10)$$

where we also used that $\log(x) \leq x$ for all $x > 0$. Substituting this into the right-hand side of (4.9),

$$I \leq \left(\frac{F_N(t)\left(1-F_N(t)\right)}{N\varepsilon}\right)^{1/2} + \delta\left(\frac{1}{2N\varepsilon}\right)^{1/2} + \frac{1}{2\delta N\varepsilon}. \quad (4.11)$$

Consider the function

$$D(\delta) = \frac{\delta}{a} + \frac{1}{\delta a^2}, D(\delta) = \frac{\delta}{(2N\varepsilon)^{1/2}} + \frac{1}{\delta(2N\varepsilon)}.$$

Taking the derivative of $D(\delta)$ and setting it equal to zero leads to

$$D'(\delta) = \frac{1}{a} - \frac{1}{\delta^2 a^2} = 0 \quad \Rightarrow \quad \delta = \pm\sqrt{\frac{1}{a}} = \pm\left(\frac{1}{2N\varepsilon}\right)^{1/4}. \tag{4.12}$$

A local minimum of $D(\delta)$ occurs at $\delta_{min} = +\left(\frac{1}{2N\varepsilon}\right)^{1/4}$, because $G''(\delta_{min}) > 0$. With this choice of δ, (4.11) becomes

$$I \leq \left(\frac{F_N(t)\left(1 - F_N(t)\right)}{N\varepsilon}\right)^{1/2} + \frac{2}{(2N\varepsilon)^{3/4}}. \tag{4.13}$$

The numerator of the first term in (4.13) is expanded as

$$\begin{aligned}
&\left|F_N(t)\left(1 - F_N(t)\right)\right| \\
&\quad = \left|\hat{F}_N(t)\left(1 - \hat{F}_N(t)\right) + \left(F_N(t) - \hat{F}_N(t)\right)\left(1 - F_N(t) - \hat{F}_N(t)\right)\right|, \\
&\quad \leq \left|\hat{F}_N(t)\left(1 - \hat{F}_N(t)\right)\right| + \left|\left(F_N(t) - \hat{F}_N(t)\right)\left(1 - F_N(t) - \hat{F}_N(t)\right)\right|.
\end{aligned} \tag{4.14}$$

Using $\left|1 - F_N(t) - \hat{F}_N(t)\right| \leq 1$ in (4.14) together with (4.5) and (4.13),

$$\begin{aligned}
I &\leq \left(\frac{\hat{F}_N(t)\left(1 - \hat{F}_N(t)\right)}{N\varepsilon}\right)^{1/2} + \frac{2}{(2N\varepsilon)^{3/4}}, \\
&\quad + \frac{1}{N\varepsilon^{1/2}}\left(\left|\sum_{n=1}^{N}\left(\mathbb{1}(\hat{Q}^{(n)} - \left|e_Q^{(n)}\right| \leq t \leq \hat{Q}^{(n)} + \left|e_Q^{(n)}\right|)\right)\right|\right)^{1/2}, \\
&\leq \left(\frac{\hat{F}_N(t)\left(1 - \hat{F}_N(t)\right)}{N\varepsilon}\right)^{1/2} + \frac{2}{(2N\varepsilon)^{3/4}}, \\
&\quad + \frac{1}{N\varepsilon^{1/2}}\left(\left|\sum_{n=1}^{N}\left(\mathbb{1}(\hat{Q}^{(n)} - \left|e_Q^{(n)}\right| \leq t \leq \hat{Q}^{(n)} + \left|e_Q^{(n)}\right|)\right)\right|\right),
\end{aligned} \tag{4.15}$$

where we also used $\sqrt{x} \leq x$ if $x = 0$ or $x \geq 1$. Since (4.15) relies on both (4.6) and (4.10), this bound occurs with a probability of at least $(1-\varepsilon)^2 = 1 - 2\varepsilon + \varepsilon^2$. Combining (4.5) and (4.15) with (4.4) completes the proof.

□

The decomposition of the error into sampling and discretization contributions can be used to adaptively improve a given CDF estimator. If sampling error is the larger contributor to the overall error, more samples should be taken to decrease the sampling contribution. On the other hand, if the error majorly comes from discretization, a more accurate numerical solution should be used.

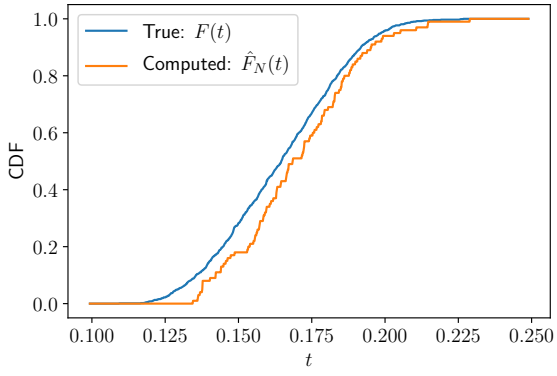
1.1 Numerical Experiments: Error in CDF

Harmonic oscillator

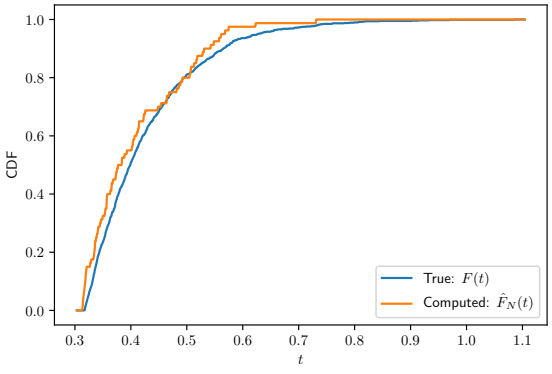
Reconsider the harmonic oscillator from §2.1.4 this time with parameters k and m as random variables:

$$\begin{pmatrix} \dot{u}_1(t) \\ \dot{u}_2(t) \end{pmatrix} + \begin{pmatrix} 0 & -1 \\ k/m & 1/m \end{pmatrix} \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = \begin{pmatrix} 0 \\ 50/m * \cos(10t) \end{pmatrix}, \ t \in (0, 2],$$

with initial conditions $(u_1(0), u_2(0)) = (5, 0)$. Let k have a normal distribution with mean 50 and a standard deviation of 5 and m be uniformly distributed over $[.125, .325]$. For the QoI, choose $R = -1$ and $G(u; t) = u_1(t)$. With $\varepsilon = 0.05$ in (4.3), the nominal CDF (4.2) is computed using the true solution given in [6] with 1000 samples. The numerical solution is obtained using cG(1) with 40 elements and the approximate CDF (2.20) is computed with $N = 100$ samples. The nominal and computed CDF are shown in Figure 4.1a. The computed error bound along with the sampling and discretization contributions are shown in Figure 4.2. Both sources contribute to the error, with the sampling error being slightly more dominant. The computed bound is indeed larger than the actual error in the distribution. Both the bound and the error peak near the inflection point of the CDF, with the error bound being about six times larger than the true error.

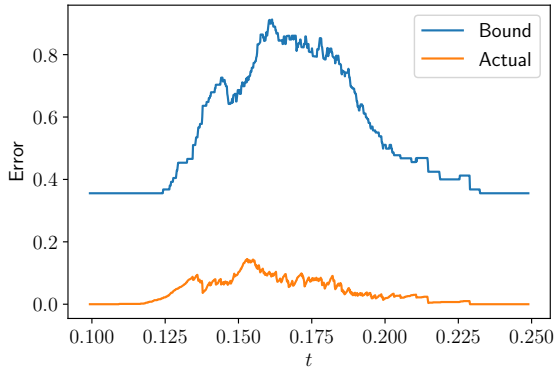


(a) Nominal CDF using 1000 samples and computed CDF using 100 samples for oscillator example in §4.1.1.

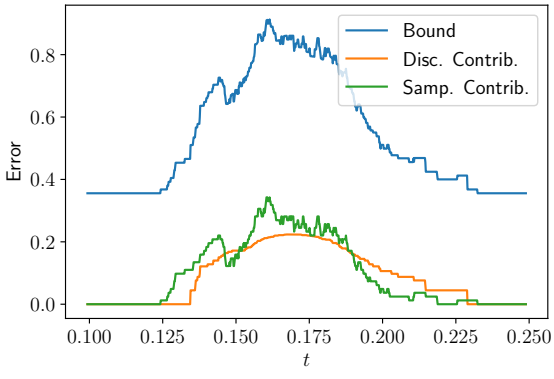


(b) Comparing nominal CDF using 1000 samples to computed CDF using 80 samples for Lorenz example in §4.1.1.

Figure 4.1



(a) Comparing computed error bound (4.3) to true error for the oscillator problem in §4.1.1 when using 1000 samples for the nominal CDF and 100 samples for the numerical CDF.



(b) Breaking the error bound into sampling and discretization contributions for the oscillator problem in §4.1.1 when using 100 samples. The sampling and discretization contributions are computed as the first and second terms of (4.3), respectively.

Figure 4.2: Error bound for oscillator example in §4.1.1.

Lorenz System

Consider the Lorenz system (3.16), where we let one of the initial conditions be a random variable. More precisely, $u_1(0) = \theta$ is uniformly distributed over the interval

$(0, 2]$. Again let $\sigma = 10$, $r = 28$, and $b = \frac{8}{3}$. For the QoI (3.4), set $R = 3$ and $G(u; t) = u_1(t)$. A reference solution and QoI are obtained using an accurate time-integrator (SciPy's solve_ivp with event tracker) with an absolute tolerance of 10^{-15} and a relative tolerance of 10^{-8}. This time, the numerical solution is computed using the cG(1) method with 30 elements.

The values needed for equation (3.26) are

$$\psi = (1, 0, 0)^\top,\ f(u, t) = (\sigma(u_2 - u_1),\ ru_1 - u_2 - u_1u_3,\ u_1u_2 - bu_3)^\top,$$

and

$$\nabla_u f(u, t) = \begin{pmatrix} \sigma & -\sigma & 0 \\ r - u_3 & -1 & -u_1 \\ -u_2 & u_1 & -b \end{pmatrix}.$$

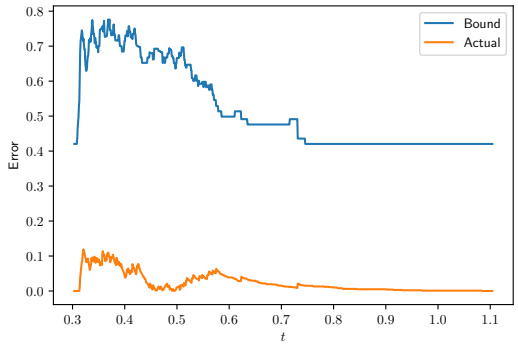
hence, for (3.27), (3.28), and (3.36) the data are

$$\psi_1 = (-1, 0, 0)^\top, \psi_2 = (-\sigma, \sigma, 0)^\top, \psi_3 = (1, 0, 0)^\top.$$

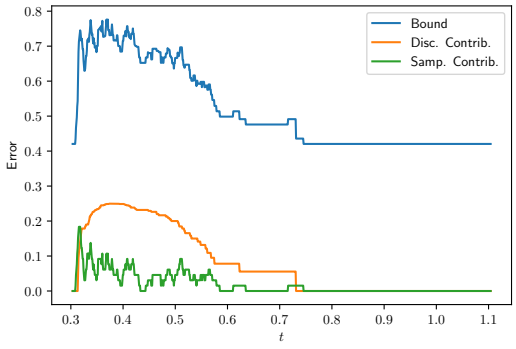
The bound (4.3) is computed with $\varepsilon = 0.05$. The Figure 4.1b compares the numerical CDF computed using 80 samples to the nominal CDF using 1000 samples. Figure 4.3 shows the discretization and sampling contributions to the calculated error bound. For this example, the discretization is the larger contributor to the error in the CDF, which is likely due to the chaotic nature of the system. As in §4.1.1 the error bound is roughly six times the true error at its peak.

2 Adjoint-based Adaptive MLMC Algorithm

This section presents the novel adaptive MLMC algorithm. Recall that $\widehat{Q}^{ML}_{\{N_\ell\},L}$ denotes the L-level MLMC estimator with N_ℓ samples taken on level ℓ. The algorithm utilizes *a posteriori* error analysis for different QoIs in two ways: i) to accurately compute a



(a) Comparing error bound (4.3) to true error for Lorenz example in §4.1.1 when using 1000 samples for the nominal CDF and 80 samples for the numerical CDF.



(b) Showing the sampling and discretization contributions to the error bound for the Lorenz example in §4.1.1 when using 100 samples. The sampling contribution is computed as the first term of (4.3), while the second term gives the discretization contribution.

Figure 4.3

stopping criteria and ii) to adaptively create meshes as new levels are added. For the stopping criteria, note that if a tolerance ϵ is desired for the MSE, it is enough to have

$$\mathbb{V}\left[\widehat{Q}^{ML}_{\{N_\ell\},L}\right] < \epsilon/2, \quad \text{and} \quad \left(\mathbb{E}\left[\widehat{Q}^{ML}_{\{N_\ell\},L} - Q\right]\right)^2 < \epsilon/2. \tag{4.16}$$

The variance $\mathbb{V}\left[\widehat{Q}^{ML}_{\{N_\ell\},L}\right]$ is controlled by the number of samples taken on each level and was discussed in §4.1.3. The bias $\mathbb{E}\left[\widehat{Q}^{ML}_{\{N_\ell\},L} - Q\right]$ is controlled by the accuracy of the highest level. Recall that the bias can be approximated with a finite number of samples as

$$\mathbb{E}\left[\widehat{Q}_{L-1} - Q\right] \approx \frac{1}{N_{L-1}} \sum_{n=1}^{N_{L-1}} \left(\widehat{Q}_{L-1}(w^{(n)}_{L-1}) - Q(w^{(n)}_{L-1})\right). \tag{4.17}$$

The term $\widehat{Q}_{L-1}(w^{(n)}_{L-1}) - Q(w^{(n)}_{L-1})$ is an error in the QoI and as such can be accurately estimated via the adjoint-based *a posteriori* analysis discussed early in this thesis.

The adaptive creation of meshes also relies on *a posteriori* error estimates. More precisely, these methods utilize decompositions of the error estimates to determine

how to effectively refine the mesh. The two adaptive methods used in this thesis are presented in §4.2.1 along with slight modifications to work in the context of MLMC. A description and pseudo-code of the adaptive MLMC algorithm are provided in §4.2.2. Finally, we present numerical experiments to compare the efficiency of using adaptively created meshes versus uniform meshes.

2.1 Adaptive Creation of New Levels for MLMC

There are many refinement methods to choose from when creating the grid for a new level. MLMC algorithms usually employ standard uniform refinement [47]. We take advantage of the form of the error decompositions presented throughout Chapter §2.4 and 3 in order to adaptively create new grids and in turn create a more efficient MLMC algorithm. The rest of this section describes the different adaptive refinement methods used in this article. Since these refinement methods are applied to individual samples, we also discuss how each refinement method is adapted to deal with multiple samples.

Dual Weighted Residual Refinement

For the dual-weighted-residual (DWR) refinement method, the regions, τ (either time-intervals or regions in Ω), corresponding to the largest contributions of error are refined by a given factor [5, 8]. This type of grid refinement relies on the error decompositions and some criteria to determine whether a region should be refined or not. Possible criteria include refining any region that corresponds to an error larger than some tolerance, or refining a certain number of regions that contribute the most error.

When using this refinement method in an MLMC algorithm, we want to combine data from multiple samples to determine which regions to refine. We do this by determining which regions should be refined for each individual sample, and then refining *all* of these regions.

More precisely, let $\mathcal{T}_h = \{\tau_1, \tau_2, ..., \tau_M\}$ be a simplicial decomposition of a domain Ω, where h denotes the maximum diameter of the elements of $\mathcal{T}_h$. Let $U(w^{(1)}) = U(x; w^{(1)})$ and $U(w^{(2)}) = U(x; w^{(2)})$ be two numerical solutions corresponding to different samples of the random parameter w. Let $\mathcal{J} = \{\tau_{J_1}, \tau_{J_2}, ..., \tau_{J_{\widehat{M}}}\}$ be the set of $\widehat{M} < M$ regions to be refined based on the error decomposition corresponding to the numerical solution $U(w^{(1)})$. Also let $\mathcal{K} = \{\tau_{K_1}, \tau_{K_2}, ..., \tau_{K_{\widetilde{M}}}\}$ be the similar set corresponding to $U(w^{(2)})$. We then refine all of the regions in the union of both sets, $\mathcal{J} \cup \mathcal{K}$.

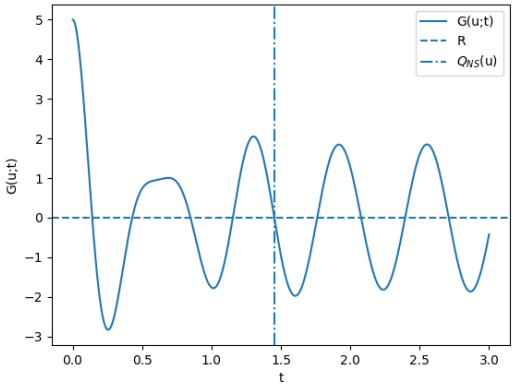
Meso-scale Refinement

In time-dependent problems, a meso-scale refinement of the time domain may be used in order to preserve cancellations of the error over large sections of the domain [20]. This refinement method considers the accumulation of the error over sub-intervals and forms "meso-scale regions", i.e. regions of maximal cancellation in the error. The meso-scale regions are each uniformly refined, using different scaling factors on different regions. Since uniform refinement usually preserves cancellations, the locations of these minima are preserved but the error decreases. Examples presented in this article follow the "Allocation of fixed resources" algorithm from [20], which is described below.

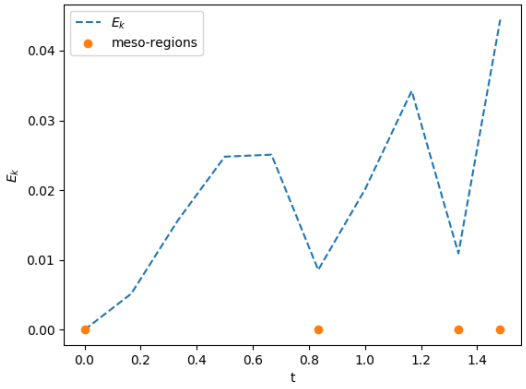
Given a numerical solution U of an IVP (1.1) (or IBVP (1.6)), computed over a temporal grid $\mathcal{T} = \{I_1, I_2, \ldots, I_{\widetilde{N}}\}$, define the accumulated contributions to the error as

$$E_k = \left| \sum_{i=1}^{k} e_{I_i} \right|, \tag{4.18}$$

where the e_{I_i} are define in (2.90) (or (2.131)). Meso-scale regions, or meso-regions, are determined by finding the global minimum of the accumulated error after the initial increase of accumulated error. This is then repeated, starting at the minimum, until the end of the temporal domain is reached. See Figure 4.4b for an example of an E_k and the corresponding meso-regions.



(a) Nonstandard QoI for the harmonic oscillator in §3.2.3, yielding the fifth occurrence of $u_1 = 0$.



(b) Accumulated error E_k and corresponding meso-regions for level $\ell = 0$ of oscillator example in §3.2.3.

Figure 4.4

Let P be the number of meso-scale regions and $\widetilde{N}_i$ be the number of time-steps in the i-th meso-scale region. If the i-th meso-scale region starts at interval I_p and ends at interval I_r, then $\widetilde{N}_i = r - p + 1$. Also let $\mathcal{E}_i = (E_r - E_p)$ denote the error accumulated over the i-th meso-scale region. Assume that this error accumulated over the i-th meso-scale region satisfies

$$\mathcal{E}_i = \frac{c_i}{\widetilde{N}_i^q}, \tag{4.19}$$

where c_i is some positive constant and q is determined by the order of the numerical method. The goal of the meso-scale refinement is to create a new grid with $\widehat{N}$ sub-intervals that minimizes the total error. If the total number of intervals $\widehat{N}$ is fixed, the total error is minimized if

$$\frac{c_i}{\widehat{N}_i^{q+1}} = K, \quad \forall i, \tag{4.20}$$

for some constant K. The $\widehat{N}_i s$ are obtained as follows: First obtain all c_i from (4.19).

Then, rearranging (4.20) to get an expression for $\widehat{N}_i$ and using the fact that

$$\sum_{i=1}^{P} \widehat{N}_i = \widehat{N}, \tag{4.21}$$

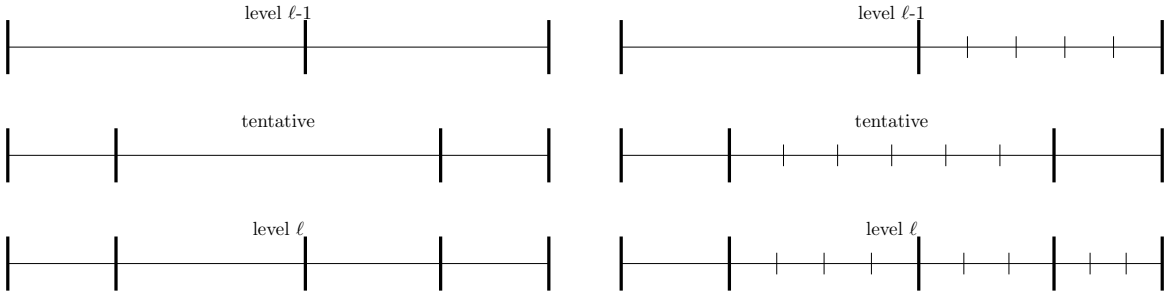
compute the value of K as

$$K = \left[\frac{1}{\widehat{N}} \sum_{i=1}^{P} \left(c_i^{1/(q+1)} \right) \right]^{q+1}. \tag{4.22}$$

Finally, obtain the $\widehat{N}_i$s from (4.20).

It is not clear how to modify the meso-scale refinement to combine information from multiple samples. Instead of combining the samples, we use the refinement that corresponds to the single sample that had the largest computed absolute error in the QoI, with the added condition that regions are never unrefined.

This is accomplished as follows. First, a tentative level ℓ grid is created with the meso-regions and the $\widehat{N}_i$s as described above. Then, a common refinement of the meso-regions from this tentative and the level $\ell-1$ is taken, see Figure 4.5a for an illustration. For each of the regions in this common refinement, determine whichever grid ($\ell - 1$ or tentative) has more nodes in that region. Finally, using that number of nodes, make a uniform sub-grid over that region. This process is explained in Figure 4.5b. Here the grid at level $\ell - 1$ has two meso-scale regions, with the first meso-region having a single interval while the second meso-region has five sub-intervals. The tentative grid (shown in the middle) has three meso-regions having one, six and one intervals respectively. The refinement, which forms the level ℓ grid is shown at the bottom. This grid has four meso-regions having one, four, three and three intervals, respectively. We observe that all regions of level ℓ grid have a finer discretization than the corresponding regions in the level $\ell - 1$ grid.



(a) Meso-regions (b) Meso-regions and sub-intervals

Figure 4.5: From top to bottom: level $\ell-1$, tentative, and level ℓ grid. The meso-regions are marked by thicker and longer lines, while the intervals within each meso-region are marked by lighter and shorter lines.

2.2 MLMC Algorithm

The adaptive MLMC routine is given in Algorithm 1. It requires a user supplied initial mesh, an initial number of samples N, and a tolerance ϵ for which the goal is to achieve MSE$< \epsilon$. First N samples (2.26) involving the QoI (2.104) are obtained. Each of these N samples requires sampling the random parameter w, solving (1.4) numerically over the initial provided mesh, and using that solution to obtain the QoI. Along with each sample, the contributions (2.114) to the error in the computed QoI are obtained and the sample variance (2.29) and sample bias (2.30) are computed. With the sample variances, the optimal number of samples (2.32) is computed. If the optimal number of samples is larger than N, more samples (2.26) involving the QoI (2.104) are taken and the bias and variance are updated to include the new data.

Next, the bias is compared to the desired tolerance. If the tolerance has not been achieved, a new level is added. The mesh for the new level is constructed using an adaptive refinement method as discussed in §4.2.1, based off the error contributions (2.114). N samples (2.26) involving the QoI (2.104) are obtained on this new level, along with the error contributions (2.114),variance (2.29), and bias (2.30). The optimal number of samples (2.32) is computed for *all* levels. Extra samples are taken on each

level as needed and the variances are updated. The bias is also updated to include all samples taken on the highest level. This process is repeated, adding as many levels as needed, until the bias has reached the desired tolerance. The MSE is then computed via (2.31).

Algorithm 1: Adaptive MLMC driver routine

Data: N,init_mesh,ϵ

Set L=1 and $\ell = 0$

Compute N samples (2.26) involving the QoI (2.104), error contributions (2.114), variance (2.29), and bias (2.30).

Find optimal number of samples, $N_{0,opt}$, using (2.32).

Compute $N_{0,opt} - N$ new samples (2.26) involving the QoI (2.104), update variance (2.29) and bias (2.30) to include all $N_{0,opt}$ samples.

while $bias^2 > \epsilon/2$ **do**

- Add new level, L=L+1 and $\ell = \ell + 1$
- Create new mesh.
- Compute N samples (2.26) involving the QoI (2.104), error contributions (2.114), variance (2.29), and bias (2.30).
- **for** $\widehat{\ell}$ *in 0,...,L-1* **do**
  - Find optimal number of samples (2.32) for level $\widehat{\ell}$.
  - Take extra samples (2.26) as needed. Update variance (2.29) to include new samples.
- **end**
- Compute bias (2.30).

end

Compute MSE using (2.31)

Compute MLMC estimator, $\widehat{Q}^{ML}_{L,\{N_\ell\}}$, using (2.25) and the MSE using (2.31)

Result: $\widehat{Q}^{ML}_{L,\{N_\ell\}}$, MSE

Remark 3. *When creating new levels, the initial number of samples taken does not have*

to be the same number N as for the lowest level. If the same large N is used for every level, starting new levels will be much more expensive than starting previous levels and it becomes more likely that unnecessary samples are taken on high levels. In our examples below, we keep the cost of starting a level relatively fixed for the first three levels, after which we keep the number of initial samples constant. This maintains a reasonable computational cost while also taking enough samples for the variance estimates to be meaningful.

Remark 4. *Algorithm 1 details how to obtain an estimated expected value in a standard QoI* (2.104) *corresponding to the solution of the differential equation* (1.4)*, although the algorithm is applicable to any differential equation and QoI, provided an error decomposition is available.*

2.3 Numerical Experiments: Adaptive MLMC

Harmonic Oscillator

Consider the harmonic oscillator

$$\ddot{\omega} = -\frac{k}{m}\omega - \frac{c}{m}\dot{\omega} + \frac{F_0}{m}\cos(\nu t + w_d),\ t \in (0,3], \quad \omega(0) = 5,\ \dot{\omega}(0) = 0.$$

with deterministic parameters $c = 1,\ F_0 = 50,\ \theta_d = 0,\ \nu = 10$, and where k and m are random parameters. Rewriting as a system of first-order ODEs, $\dot{u} + Au = \tilde{f}(t)$, gives

$$\begin{pmatrix} \dot{u_1}(t) \\ \dot{u_2}(t) \end{pmatrix} + \begin{pmatrix} 0 & -1 \\ k/m & 1/m \end{pmatrix} \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = \begin{pmatrix} 0 \\ 50/m * \cos(10t) \end{pmatrix},\ t \in (0,3]. \tag{4.23}$$

With initial conditions $(u_1(0), u_2(0)) = (5, 0)$. We look at examples of both a standard QoI and the aforementioned non-standard QoI.

Oscillator: Standard QoI

Consider the oscillator (4.23) where $k \sim N(50, 2)$, a normal distribution with mean 50 and standard deviation 2, and $m \sim Unif[0.225, 0.275]$, a uniform distribution between 0.225 and 0.275. The standard QoI is $Q_S(u) = [(1,0)^\top \cdot u(T)] = u_1(T)$, the position of the oscillator at the final time $T = 3$. For this QoI, the error representation (2.79) is

$$e(T) \cdot (1,0)^\top = \int_0^T \phi \cdot \left(\tilde{f} - \dot{U} - AU \right) \mathrm{d}t, \tag{4.24}$$

where ϕ is the solution to the adjoint problem

$$\begin{cases} -\dot{\phi} + A^\top \phi &= 0 \quad t \in [0, T), \\ \phi(T) &= 1. \end{cases} \tag{4.25}$$

The grid for the lowest level has 27 elements, with the number of elements roughly doubling for each further level. The algorithm is run using three different grid creation methods (uniform grid creation, DWR refinement, and meso-scale refinement) to compare cost efficiencies. In each, we start by taking 100 samples on the lowest level. When creating further levels, the second level starts with 50 samples and all further levels start with 20 samples. The tolerance for the MSE is set to $\epsilon = 0.001$.

Results when using uniform grids are shown in Tables 4.1 and 4.2. Table 4.1 provides details for each level of the estimator, including the number of elements used to create the grid, the relative cost per sample, the number of samples, and variance contribution (2.29). Table 4.2 gives results for the MLMC estimator, including the variance, squared bias, MSE (2.31), the estimated expected value (2.25), and the total relative cost. The cost is computed by summing across all levels, the number of samples at each level times the cost per sample at that level. The cost of the sample at level $\ell = 0$ (i.e. the cost of computing a single value of $\widehat{Q}_0(w_\ell^{(n)})$, for any ℓ) is normalized to 1. Samples for levels $\ell > 0$ require computing two values, $\widehat{Q}_\ell(w_\ell^{(n)})$ and $\widehat{Q}_{\ell-1}(w_\ell^{(n)})$. Hence, the cost of taking a sample on level ℓ is the sum of the costs of computing $\widehat{Q}_\ell(w_\ell^{(n)})$ and $\widehat{Q}_{\ell-1}(w_\ell^{(n)})$ relative to the cost of computing $\widehat{Q}_0(w_\ell^{(n)})$. For example, if

$\ell = 1$ and computing a single value of $\widehat{Q}_1(w_1^{(n)})$ is twice the computational cost of computing a single $\widehat{Q}_0(w_1^{(n)})$, then the relative cost of a sample on the $\ell = 1$ level is $Cost(\widehat{Q}_1(w_1^{(n)})) + Cost(\widehat{Q}_0(w_1^{(n)})) = 2 + 1 = 3$.

Tables 4.3 and 4.4 show similar details when the grids are created using DWR refinement. Results for the MLMC estimator using meso-scale refinement are provided in Tables 4.5 and 4.6. Since levels of the MLMC estimator are made from the *difference* between two approximate QoIs, the variance per level decrease for higher levels (i.e. as the approximations converge). When using uniform grids, the MLMC estimator requires four levels in order to achieve tolerance in bias. The estimator using DWR refinement requires three levels. With meso-scale refinement, only two levels are required to have a bias less than tolerance. More samples are required on level $\ell = 1$ when using DWR or meso-scale refinement but the overall costs are still lower compared to using uniform refinement. Grids for each refinement method are shown in Figure 4.6.

Level	# Elems	Cost Per Sample	# Samples	Var. Per Level
0	27	1	227	4.32968E-4
1	54	3	52	7.36927E-4
2	108	6	20	1.90041E-6
3	216	12	20	3.28090E-7

Table 4.1: Results for each level of estimator in oscillator example from §4.2.3 with $\epsilon = .001$. New grids are obtained via uniform refinement.

Tot. Var.	Squared Bias	MSE	Est. Exp. Val	Tot. Cost
0.00117	2.82413E-5	0.00120	-0.38276	743

Table 4.2: Results of MLMC estimator in oscillator example from §4.2.3 with $\epsilon = .001$. New grids are obtained via uniform refinement.

Level	# Elems	Cost Per Sample	# Samples	Var. Per Level
0	27	1	227	4.32968E-4
1	63	3.333	75	4387715E-4
2	153	8	20	1.36574E-6

Table 4.3: Results for each level of estimator in oscillator example from §4.2.3 with $\epsilon = .001$. New grids are obtained via DWR refinement where the 50% largest contributions to the error are refined by a factor of 3.

Tot. Var.	Squared Bias	MSE	Est. Exp. Val	Tot. Cost
9.22049E-4	8.02556E-5	0.001002	-0.40173	637

Table 4.4: Results of MLMC estimator in oscillator example from §4.2.3 with $\epsilon = .001$. New grids are obtained via DWR refinement where the 50% largest contributions to the error are refined by a factor of 3.

Level	# Elems	Cost Per Sample	# Samples	Var Per Level
0	27	1	227	4.32968E-4
1	59	3.185	65	8.27785E-4

Table 4.5: Results for each level of estimator in oscillator example from §4.2.3 with $\epsilon = .001$. New grids are obtained via meso-scale refinement.

Tot. Var	Squared Bias	MSE	Est. Exp. Val	Tot. Cost
0.00126	1.80659E-5	0.00127	-0.3816	434.03

Table 4.6: Results of MLMC estimator in oscillator example from §4.2.3 with $\epsilon = .001$. New grids are obtained via meso-scale refinement.

Oscillator: Nonstandard QoI

Using the same equation and deterministic parameters as in §4.2.3, now let $k \sim N(50, 1)$ and $m \sim Unif(0.235, 0.265)$. Let the nonstandard QoI to be the time of the 5th occurrence of $u_1 = 0$. More precisely, set $\psi = (1, 0)^\top$ in (3.5). Also, for the function H in (3.1), set $R = 0$ and choose a $\hat{t}$ between the fourth and fifth occurrence of $u_1 = 0$;

see Figure 4.4a. The error representation (3.26) becomes

$$t_t - t_c \approx \frac{\psi \cdot e(t_c))}{A^\top \psi \cdot U(t_c) - \psi \cdot \tilde{f}(t_c) + A^\top \psi \cdot e(t_c)}. \tag{4.26}$$

Using Theorem 4, the two error terms are given as

$$\psi \cdot e(t_c) = \int_0^{t_c} \left[\tilde{f} \cdot \phi_1 - \frac{dU}{dt} \cdot \phi_1 - AU \cdot \phi_1 \right] \mathrm{d}t \tag{4.27}$$

$$A^\top \psi \cdot e(t_c) = \int_0^{t_c} \left[\tilde{f} \cdot \phi_2 - \frac{dU}{dt} \cdot \phi_2 - AU \cdot \phi_2 \right] \mathrm{d}t, \tag{4.28}$$

where ϕ_1 and ϕ_2 are the solutions to the adjoint problems:

$$\begin{cases} -\dot{\phi_1} + A^\top \phi_1 &= 0 \quad t \in [0, t_c), \\ \phi_1(t_c) &= \psi, \end{cases}, \qquad \begin{cases} -\dot{\phi_2} + A^\top \phi_2 &= 0 \quad t \in [0, t_c), \\ \phi_2(t_c) &= A^\top \psi, \end{cases}. \tag{4.29}$$

All examples in this section start with 100 samples of a numerical solution obtained over a uniform grid with 18 sub-intervals. The second level starts with 50 samples and all further levels with 20 samples. The grids used for levels beyond the first are obtained from the different creation methods as discussed in §4.2.1.

Tables 4.7 and 4.8 provide results for the levels and overall MLMC estimator when using uniform grids. Results for the MLMC estimator using DWR refinement are shown in Tables 4.9 and 4.10. Tables 4.11 and 4.12 give results for the estimator when grids are created using meso-scale refinement. The grids for different levels of each method can be seen in Figure 4.7. In all three cases, the initial number of samples is enough to achieve tolerance for the variance of the estimators. The different grid creation methods perform similarly, all requiring four levels and the same number of samples.

Lorenz Equations

Consider the (nonlinear) Lorenz system,

$$\left.\begin{aligned} \dot{u}_1 &= \sigma(u_2 - u_1), \\ \dot{u}_2 &= r u_1 - u_2 - u_1 u_3, \\ \dot{u}_3 &= u_1 u_2 - b u_3, \end{aligned}\right\} \; t \in (0, 2] \qquad \text{with} \qquad \begin{cases} u_1(0) = w, \\ u_2(0) = 0, \\ u_3(0) = 24, \end{cases} \tag{4.30}$$

Level	# Elems	Cost Per Sample	# Samples	Var. Per Level
0	18	1	100	1.04977E-6
1	36	3	50	1.44317E-6
2	72	6	20	5.22219E-6
3	144	12	20	2.11021E-9

Table 4.7: Results for each level of estimator in oscillator example with NSQoI from §4.2.3 with $\epsilon = 1E-5$. New grids are obtained via uniform refinement.

Tot. Var.	Squared Bias	MSE	Est. Exp. Val	Tot. Cost
3.01727E-6	2.02532E-6	5.04259E-6	1.45701	610

Table 4.8: Results of MLMC estimator in oscillator example with NSQoI from §4.2.3 with $\epsilon = 1E-5$. New grids are obtained via uniform refinement.

Level	# Elems	Cost Per Sample	# Samples	Var. Per Level
0	18	1	100	1.04977E-6
1	34	2.888	50	2.17999E-6
2	72	5.888	20	1.54181E-6
3	140	11.777	20	2.25459E-9

Table 4.9: Results for each level of estimator in oscillator example with NSQoI from §4.2.3 with $\epsilon = 1E-5$. New grids are obtained via DWR refinement where the 50% largest contributions to the error are refined by a factor of 3.

Tot. Var.	Squared Bias	MSE	Est. Exp. Val	Tot. Cost
4.77384E-6	1.73603E-7	4.94744E-6	1.45620	597.77

Table 4.10: Results of MLMC estimator in oscillator example with NSQoI from §4.2.3 with $\epsilon = 1E-5$. New grids are obtained via DWR refinement where the 50% largest contributions to the error are refined by a factor of 3.

and set $\sigma = 10, r = 28$, and $b = \frac{8}{3}$. The initial condition w is a random variable $w \sim Unif(0, 2]$.

The nonstandard QoI is the time of the 2nd occurrence of $u_1 = 3$. That is, set $\psi = (1, 0, 0)^\top$, in (3.5). In (3.1) set $R = 3$ and choose a $\hat{t}$ between the first and second occurrence of $u_1 = 3$.

Level	# Elems	Cost Per Sample	# Samples	Var. Per Level
0	18	1	100	1.0497E-6
1	39	3.166	50	1.06958E-6
2	70	6.055	20	1.23273E-7
3	121	10.611	20	3.83435E-9

Table 4.11: Results for each level of estimator in oscillator example with NSQoI from §4.2.3 with $\epsilon = 1E-5$. New grids are obtained via meso-scale refinement.

Tot. Var.	Squared Bias	MSE	Est. Exp. Val	Tot. Cost
2.24646E-6	2.11483E-7	2.45794E-6	1.45597	591.66

Table 4.12: Results of MLMC estimator in oscillator example with NSQoI from §4.2.3 with $\epsilon = 1E-5$. New grids are obtained via meso-scale refinement.

The error approximation (3.26) becomes

$$t_t - t_c \approx \frac{\psi \cdot e(t_c)}{\left(\nabla_u f(t_c)\right)^\top \psi \cdot U(t_c) - \psi \cdot f(t_c) + \left(\nabla_u f(t_c)\right)^\top \psi \cdot e(t_c)}. \tag{4.31}$$

Using Theorem 4, the two error terms are given as

$$\psi \cdot e(t_c) = \int_0^{t_c} \left[f \cdot \phi_1 - \frac{dU}{dt} \cdot \phi_1 \right] \mathrm{d}t \tag{4.32}$$

$$\left(\nabla_u f(t_c)\right)^\top \psi \cdot e(t_c)) = \int_0^{t_c} \left[f \cdot \phi_2 - \frac{dU}{dt} \cdot \phi_2 \right] \mathrm{d}t, \tag{4.33}$$

where ϕ_1 and ϕ_2 are the solutions to the adjoint problems

$$\begin{cases} -\dot{\phi_1} & = \left(\nabla_u f\right)^\top \phi \quad t \in [0, t_c), \\ \phi_1(t_c) & = \psi, \end{cases}, \qquad \begin{cases} -\dot{\phi_2} & = \left(\nabla_u f\right)^\top \phi \quad t \in [0, t_c), \\ \phi_2(t_c) & = \left(\nabla_u f\right)^\top \psi, \end{cases}. \tag{4.34}$$

The algorithm starts with 100 samples of a numerical solution obtained over a uniform grid with 15 sub-intervals. The second level starts with 50 samples and all further levels with 20 samples. The grids used for levels beyond the first are obtained from the different creation methods as discussed in §4.2.1. Tables 4.13, 4.15, and 4.17 show results for each level of the estimators when using uniform grids, DWR refinement and

meso-scale refinement, respectively. Results for the MLMC estimators, using uniform grids, DWR refinement, and meso-scale refinement are shown in Tables 4.14, 4.16, and 4.18, respectively. The grids used in each estimator are shown in Figure 4.8. For this nonlinear problem, all three grid creation methods perform similarly. The different methods require two levels and use the same number of samples and hence the comparable approximate costs.

Level	# Elems	Cost Per Sample	# Samples	Var. Per Level
0	24	1	101	4.59302E-5
1	48	3	50	5.66121E-7

Table 4.13: Results for each level of the estimator in Lorenz example from §4.2.3 with $\epsilon = 1E-4$. New grids are obtained via uniform refinement.

Tot. Var.	Squared Bias	MSE	Est. Exp. Val	Tot. Cost
4.64963E-5	3.68484E-5	8.33448E-5	0.84945	251

Table 4.14: Results of the MLMC estimator in Lorenz example from §4.2.3 with $\epsilon = 1E-4$. New grids are obtained via uniform refinement.

Level	# Elems	Cost Per Sample	# Samples	Var. Per Level
0	24	1	100	4.59302E-5
1	50	3.083	50	8.13257E-7

Table 4.15: Results for each level of the estimator in Lorenz example from §4.2.3 with $\epsilon = 1E-4$. New grids are obtained via DWR refinement where the 50% largest contributions to the error are refined by a factor of 3.

Tot. Var.	Squared Bias	MSE	Est. Exp. Val	Tot. Cost
4.67435E-5	6.52347E-6	5.32670E-5	0.84501	254.16

Table 4.16: Results of the MLMC estimator in Lorenz example from §4.2.3 with $\epsilon = 1E-4$. New grids are obtained via DWR refinement where the 50% largest contributions to the error are refined by a factor of 3.

Level	# Elems	Cost Per Sample	# Samples	Var. Per Level
0	24	1	100	4.59302E-5
1	53	3.208	50	4.71161E-6

Table 4.17: Results for each level of the estimator in Lorenz example from §4.2.3 with $\epsilon = 1E-4$. New grids are obtained via meso-scale refinement.

Tot. Var.	Squared Bias	MSE	Est. Exp. Val	Tot. Cost
5.06418E-5	3.03186E-5	8.09605E-5	0.84931	260.4

Table 4.18: Results of the MLMC estimator in Lorenz example from §4.2.3 with $\epsilon = 1E-4$. New grids are obtained via meso-scale refinement.

Two-Body Problem

Consider the two body problem

$$\left.\begin{aligned} \dot{u}_1 &= u_3, \\ \dot{u}_2 &= u_4, \\ \dot{u}_3 &= \frac{-u_1}{(u_1^2+u_2^2)^{3/2}}, \\ \dot{u}_4 &= \frac{-u_2}{(u_1^2+u_2^2)^{3/2}}, \end{aligned}\right\} \quad t \in (0,10], \quad u(0) = (0.4,0,0,w)^\top, \tag{4.35}$$

which models a small body orbiting a much larger body. Here u_1, u_2 are the spatial coordinates of the orbiting body relative to the larger body, and u_3, u_4 are the respective velocities. The last component of the initial condition, w, is a random variable $w \sim Unif[1.97, 2]$.

For the nonstandard QoI, set $\psi = (1,0,0,0)^\top$ in (3.5). Also set $R = 0$ in (3.1) and choose a $\hat{t}$ to obtain the 3rd occurrence of $u \cdot \psi = u_1 = 0$. The error representation (3.26) for this problem is the same as (3.4). From Theorem 4, the two error terms are

again given by (4.32) where ϕ_1 and ϕ_2 are the solutions to the adjoint problems

$$\begin{cases} -\dot{\phi_1} & = (\nabla_u f)^\top \phi \quad t \in [0, t_c), \\ \phi_1(t_c) & = \psi, \end{cases}, \qquad \begin{cases} -\dot{\phi_2} & = (\nabla_u f)^\top \phi \quad t \in [0, t_c), \\ \phi_2(t_c) & = (\nabla_u f)^\top \psi, \end{cases}. \tag{4.36}$$

The MLMC algorithm begins with 100 samples of a numerical solution obtained over a uniform grid with 40 sub-intervals. The second level starts with 50 samples and all further levels with 20 samples. The grids used for levels beyond the first are obtained from the different creation methods as discussed in §4.2.1.

Results for the MLMC estimator using uniform grids are shown in Tables 4.19 and 4.20. Tables 4.21 and 4.22 give results when using DWR refinement. Results when using meso-scale refinement are provided in Tables 4.23 and 4.24. The grids used in each estimator are provided in Figure 4.9.

Here, the MLMC estimator using DWR refinement is the most cost efficient, requiring three levels to achieve the desired tolerance in bias. Using meso-scale refinement, the estimator requires four levels, and the estimator using uniform grids requires five levels. Notice that the two estimators using adaptive grid creation methods do not achieve the desired tolerance for variance, due to under-sampling on some level(s). The estimator using DWR refinement is close to meeting tolerance and would only require some more samples on the lowest, cheapest level. The estimator using meso-scale refinement is further from tolerance and would require more samples on all levels, including the higher, more expensive levels. The large variance and slow convergence of the levels when using meso-scale refinement is due to the large variance of the approximated QoI and the fact that our grid creation method only uses information from a single sample of the QoI. The random parameter, $u_4(0) = w$, is the initial velocity of the of the body in the y-direction. The trajectory of the body is sensitive to initial velocity, thus small changes in $u_4(0)$ lead to large changes in the sample QoI. This causes the grid creation method to be less effective because we base the refinement off of a single sample, which

can lead to a poor refinement when taking a different sample.

Level	# Elems	Cost Per Sample	# Samples	Var. Per Level
0	40	1	389	5.1388E-4
1	80	3	50	1.86982E-6
2	160	6	20	1.35918E-6
3	320	12	20	1.20867E-7
4	640	24	20	1.49859E-8

Table 4.19: Results for each level of the estimator in two-body example from §4.2.3 with $\epsilon = 1E-3$. New grids are obtained via uniform refinement.

Tot. Var.	Squared Bias	MSE	Est. Exp. Val	Tot. Cost
5.17251E-4	3.38445E-5	5.55109E-4	7.24045	1379

Table 4.20: Results of the MLMC estimator in two-body example from §4.2.3 with $\epsilon = 1E-3$. New grids are obtained via uniform refinement.

Level	# Elems	Cost Per Sample	# Samples	Var. Per Level
0	40	1	255	6.99600E-4
1	74	2.85	50	4.15496E-6
2	160	5.85	20	8.88944E-7

Table 4.21: Results for each level of the estimator in two-body example from §4.2.3 with $\epsilon = 1E-3$. New grids are obtained via DWR refinement where the 50% largest contributions to the error are refined by a factor of 3.

Tot. Var.	Squared Bias	MSE	Est. Exp. Val	Tot. Cost
7.04644E-4	3.72135E-4	0.00107	7.06503	514.5

Table 4.22: Results of the MLMC estimator in two-body example from §4.2.3 with $\epsilon = 1E-3$. New grids are obtained via DWR refinement where the 50% largest contributions to the error are refined by a factor of 3.

Level	# Elems	Cost Per Sample	# Samples	Var. Per Level
0	40	1	204	4.46312E-4
1	81	3.025	64	8.85888E-4
2	176	6.425	55	1.49708E-3
3	394	14.25	20	1.10254E-3

Table 4.23: Results for each level of the estimator in two-body example from §4.2.3 with $\epsilon = 1E-3$. New grids are obtained via meso-scale refinement.

Tot. Var.	Squared Bias	MSE	Est. Exp. Val	Tot. Cost
2.93953E-3	2.18287E-4	0.00315	6.88846	1035.975

Table 4.24: Results of the MLMC estimator in two-body example from §4.2.3 with $\epsilon = 1E-3$. New grids are obtained via meso-scale refinement.

Stationary Advection-Diffusion Equation

Consider the equation

$$\begin{cases} \nabla^2 u(x) + b \cdot \nabla u(x) = f(x), & x \in (0,3) \times (0,1), \\ u(x) = 0, & x \in \partial\Omega. \end{cases} \tag{4.37}$$

The vector $b = (w, 0)^\top$ has, as its first component, the random parameter $w = unif(1200, 1600)$. The source f is non-zero only over an interior region of the domain, and is given as

$$f = \begin{cases} 10000(x-1)(x-2.5)(y-\frac{1}{6})(y-\frac{5}{6}) & 1 \le x \le 2.5,\ \frac{1}{6} \le y \le \frac{5}{6}, \\ 0 & \text{else.} \end{cases} \tag{4.38}$$

The weak form of (4.37) is: Find $u \in H^1(\Omega)$ such that

$$-\Big(\nabla u, \nabla v\Big) + \Big(b \cdot \nabla u, v\Big) = \Big(f, v\Big), \quad \forall v \in H_0^1(\Omega). \tag{4.39}$$

The QoI is the integral of the solution over the rectangle $(1, 1.5) \times (1/3, 2/3)$:

$$Q_S(u) = \int_\Omega \psi \cdot u \mathrm{d}\Omega, \quad \text{where} \quad \psi = \begin{cases} 1, & (x,y) \in (1, 1.5) \times (\frac{1}{3}, \frac{2}{3}) \\ 0, & else. \end{cases} \tag{4.40}$$

See Figure 2.2a for a visualization of the supports of f and ψ. For the standard QoI (2.118) the error representation (2.119) becomes

$$\Big(e, \psi\Big) = -\Big(\nabla U, \nabla \phi\Big) + \Big(\nabla \cdot (bU), \phi\Big) - \Big(f, \phi\Big), \tag{4.41}$$

where ϕ is the solution to the adjoint problem

$$\begin{cases} \nabla^2 \phi(x) - b \cdot \nabla \phi(x) + \psi(x) = 0, & x \in \Omega, \\ \phi(x) = 0, & x \in \partial\Omega. \end{cases} \tag{4.42}$$

The MLMC algorithm is applied to this problem using two different mesh creation methods; uniform and DWR refinement. In both cases we aim to roughly double the number of elements when creating new levels. This is done so that each level of the different methods can be more easily compared. When using uniform meshes, we multiply the number of nodes in each coordinate by $\sqrt{2}$ and round up. The uniform meshes are shown in Figure 4.10 and results in Table 4.25 and 4.26. For the DWR refinement, the regions corresponding to the 25% largest contributions to the error are refined (using DOLFIN Python's in-built refine function). The adaptively refined meshes are shown in Figure 4.11 and results in Tables 4.27 and 4.28. For this higher dimensional, linear problem, the MLMC estimator using DWR refinement is much more cost efficient than using uniform meshes. With uniform meshes, the estimator requires five levels to achieve the desired tolerance in bias, while the DWR method requires three levels. Also, when using uniform meshes, more samples on each level are required than in the DWR method in order to reduce the variance.

Level	# Elems	Cost Per Sample	# Samples	Var. Per Level
0	72	1	1530	3.19378E-6
1	162	3.25	29	7.93078E-7
2	288	6.25	34	1.96356E-6
3	578	12.027	10	3.15403E-7
4	1152	24.027	10	2.41456E-6

Table 4.25: Results for each level of the estimator in the stationary advection-diffusion example from §4.2.3 using uniform meshes with tolerance $\epsilon = 10^{-5}$.

Tot. Var.	Squared Bias	MSE	Est. Exp. Val	Tot. Cost
8.68040E-6	5.87876E-7	9.26828E-6	-0.23215	2197.2

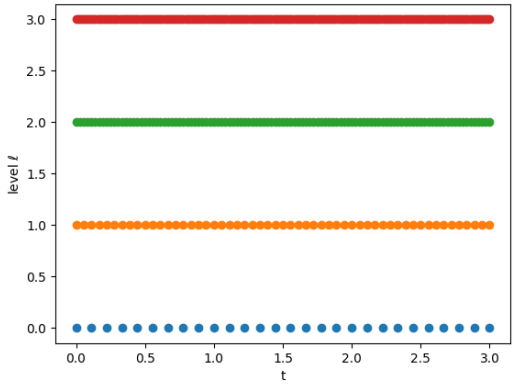
Table 4.26: Results of the MLMC estimator in the stationary advection-diffusion example from §4.2.3 using uniform meshes with tolerance $\epsilon = 10^{-5}$.

Level	# Elems	Cost Per Sample	# Samples	Var. Per Level
0	72	1	979	4.98828E-6
1	135	2.680	25	8.14877E-8
2	291	4.972	10	1.68624E-7

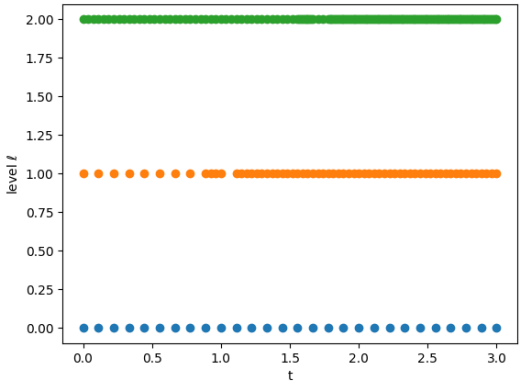
Table 4.27: Results for each level of the estimator in the stationary advection-diffusion example from §4.2.3 with tolerance $\epsilon = 10^{-5}$. New meshes are obtained using DWR refinement where the 25% largest contributions to the error are refined.

Tot. Var.	Squared Bias	MSE	Est. Exp. Val	Tot. Cost
5.23839E-6	4.10668E-7	5.64906E-6	-0.22919	1120.5

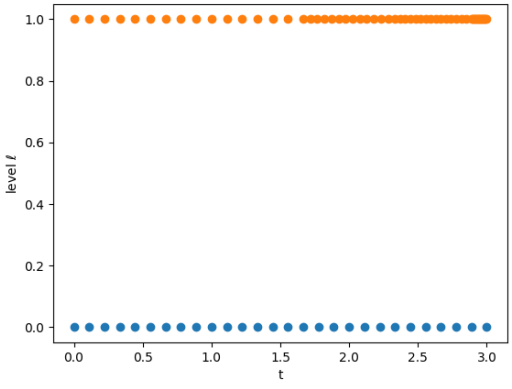
Table 4.28: Results of the MLMC estimator in the stationary advection-diffusion example from §4.2.3 with tolerance $\epsilon = 10^{-5}$. New meshes are obtained using DWR refinement where the 25% largest contributions to the error are refined.



(a) Uniform refinement grids

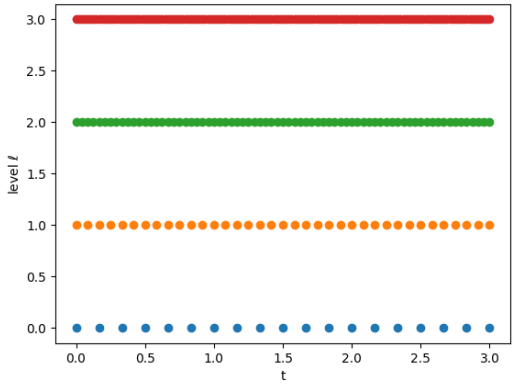


(b) DWR refinement grids

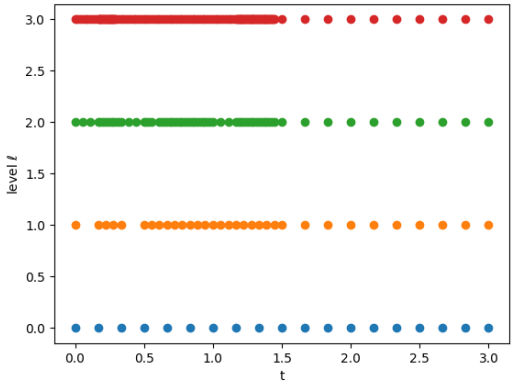


(c) Meso-scale refinement grids

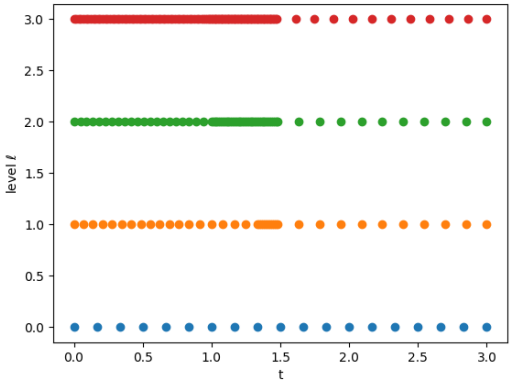
Figure 4.6: Grids from the different refinement methods for the oscillator example in §4.2.3.



(a) Uniform refinement grids

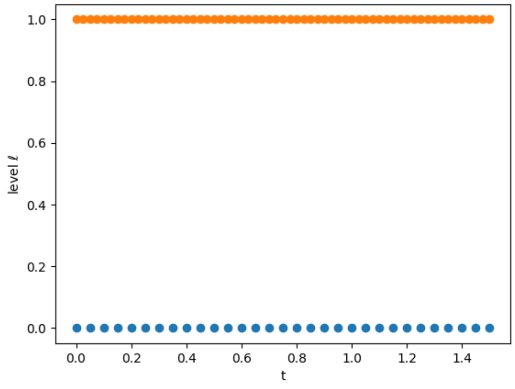


(b) DWR refinement grids

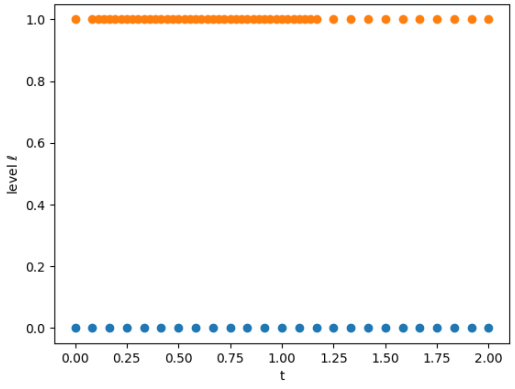


(c) Meso-scale refinement grids

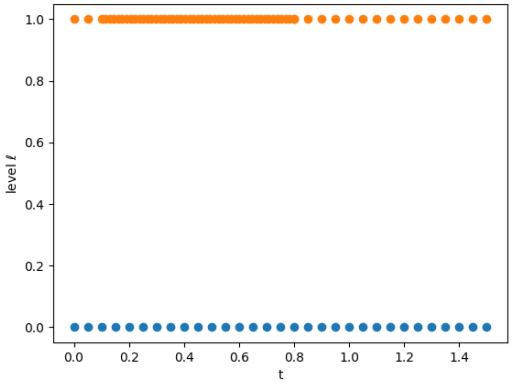
Figure 4.7: Grids from the different refinement methods for the oscillator example with NSQoI in §4.2.3.



(a) Uniform refinement grids

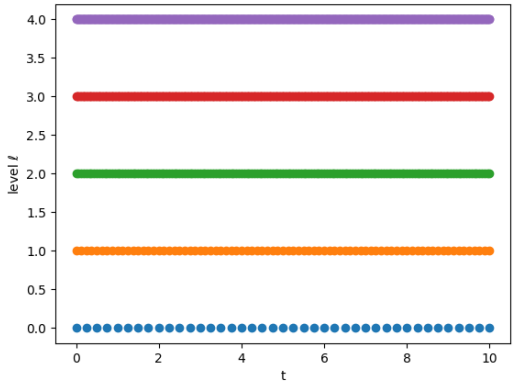


(b) DWR refinement grids

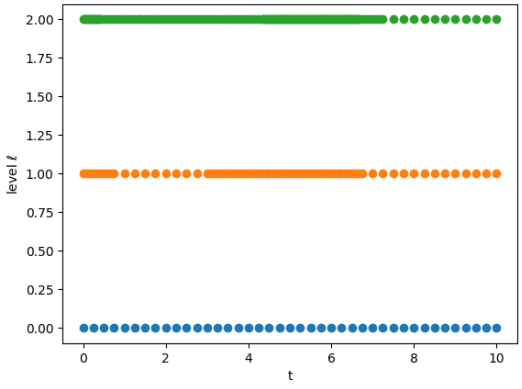


(c) Meso-scale refinement grids

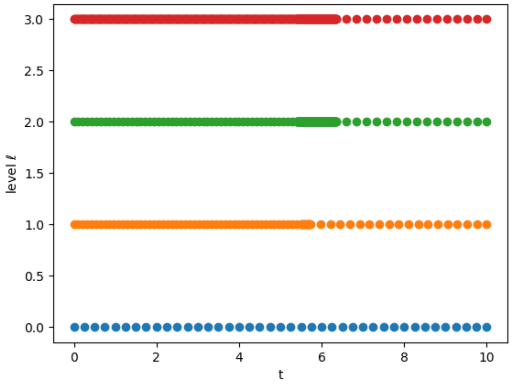
Figure 4.8: Grids from the different refinement methods for the Lorenz example in §4.2.3



(a) Uniform refinement grids

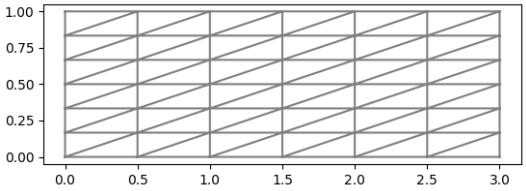


(b) DWR refinement grids

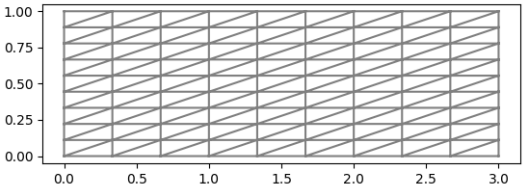


(c) Meso-scale refinement grids

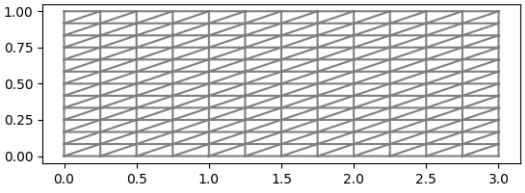
Figure 4.9: Grids from the different refinement methods for the two-body example in 4.§2.3



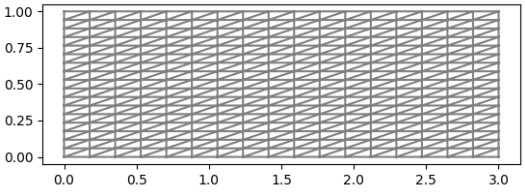
(a) Initial mesh, $\ell = 0$



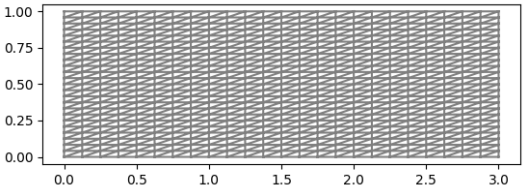
(b) Level $\ell = 1$



(c) Level $\ell = 2$

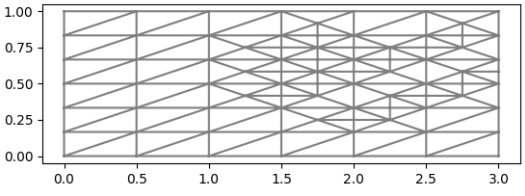


(d) Level $\ell = 3$

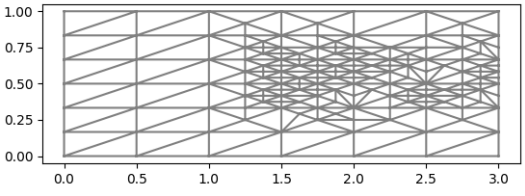


(e) Level $\ell = 4$

Figure 4.10: Uniform meshes for the stationary advection-diffusion example in §4.2.3



(a) Level $\ell = 1$



(b) Level $\ell = 2$

Figure 4.11: Adaptively refined meshes for the stationary advection-diffusion example in §4.2.3

Chapter 5

Conclusions

We have developed two methods of uncertainty quantification for several quantities of interest related to differential equations that depend on random parameters. A bound in the error of a computed cumulative distribution function has been derived. Also, an adaptive MLMC algorithm with accurately computed stopping criteria has been created.

The UQ methods have been applied to a wide array of problems and quantities of interest and require adjoint-based *a posteriori* error analysis. We have analyzed the error in a QoI that is not covered by classical *a posteriori* error analysis, namely the first time when a given functional G of the solution achieves a specific value. To fully analyze this QoI, we have derived *a priori* convergence results in Theorem 8 and have developed two different classes of accurate *a posteriori* error estimates for a QoI that cannot be expressed as a bounded functional of the solution. The first method, Theorem 9, is based on Taylor's Theorem and is accurate whenever the numerical solution is sufficiently accurate and the curvature of the functional G is not too large. Moreover this method is cost effective, requiring the solution of only two adjoint problems. The second class of methods, detailed in §3.1.3, are based on standard root-finding techniques and are accurate provided the numerical solution is sufficiently accurate near the event of

interest. These estimates however are more costly, requiring one adjoint solution per iteration of the root-finding algorithm. Both the Taylor series and the root-finding approaches provide accurate error estimates in most cases. Some limitations of these methods have been revealed in §3.1.4 on pages 66 and 67. The poor results in the example on page 66 are caused by the use of a low accuracy solution and the fact that computed QoI was closer to the second time the threshold value was crossed than the first. In the example on page 67, specifically Tables 3.11, 3.12 and 3.13, we have observed that the issue which arose on page 66 can be remedied by using a numerical solution that is more accurate near the QoI. Although another issue is revealed in the final column of Table 3.13, where the Taylor series approach gives poor results even though the numerical solution is quite accurate. There, the poor result is caused by assuming that the second derivative of $G(u;t)$ with respect to t can be neglected. The example in §3.1.4 on page 75 shows that the Taylor series approach may not be accurate if the error in the QoI is large, but the iterative methods are accurate provided the root finding technique locates the correct root.

Both methods can be used as a basis for determining the discretization contribution to an error bound on a CDF, Theorem 10, of the functional when one or more of the parameters governing the system of differential equations are random variables. The iterative methods are not suitable for the adaptive MLMC algorithm as they do not provide a decomposition of the error to use in mesh creation. The MLMC algorithm 1 relies on the error decomposition to accurately approximate the bias and adaptively refine meshes when creating new levels. It is shown that using an adaptive refinement method, either meso-scale or refining regions of largest error, leads to a more cost-effective method than uniform refinement. The advantages of adaptive refinement become more prominent in higher-dimensional problems and in problems where error accumulation is localized, as is illustrated in §4.2.3.

Bibliography

[1] Chapter 8 differential equations with random coefficients. In T.T. Soong, editor, *Random Differential Equations in Science and Engineering*, volume 103 of *Mathematics in Science and Engineering*, pages 217–254. Elsevier, 1973.

[2] M. Ainsworth and T. Oden. A posteriori *error estimation in finite element analysis*. John Wiley-Teubner, 2000.

[3] David F. Anderson and Desmond J. Higham. Multilevel monte carlo for continuous time markov chains, with applications in biochemical kinetics. *Multiscale Modeling & Simulation*, 10, 2012.

[4] Tom M. Apostol. *Calculus*, volume 1. Wiley, 2nd edition, 1967.

[5] W. Bangerth and R. Rannacher. *Adaptive Finite Element Methods for Differential Equations*. Birkhauser Verlag, 2003.

[6] V. Barger and M. Olsson. *Classical Mechanics, A Modern Perspective*. McGraw-Hill, New York, 1973.

[7] T. J. Barth. A posteriori *Error Estimation and Mesh Adaptivity for Finite Volume and Finite Element Methods*, volume 41 of *Lecture Notes in Computational Science and Engineering*. Springer, New York, 2004.

[8] R. Becker and R. Rannacher. An optimal control approach to *a posteriori* error estimation in finite element methods. *Acta Numerica*, pages 1–102, 2001.

[9] Russel E. Caflisch. Monte carlo and quasi-monte carlo methods. *Acta Numerica*, 7:1–49, 1998.

[10] Yang Cao and Linda Petzold. *A posteriori* error estimation and global error control for ordinary differential equations by the adjoint method. *SIAM Journal on Scientific Computing*, 26(2):359–374, 2004.

[11] 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 Journal on Numerical Analysis*, 47(1):740–761, 2008.

[12] J. H. Chaudhry, D. Estep, V. Ginting, and S. Tavener. *A posteriori* analysis for iterative solvers for non-autonomous evolution problems. *SIAM Journal on Uncertainty Quantification*, 3, 2015.

[13] J. H. Chaudhry, J.N. Shadid, and T. Wildey. *A posteriori* analysis of an IMEX entropy-viscosity formulation for hyperbolic conservation laws with dissipation. *Applied Numerical Mathematics*, 135, 2019.

[14] Jehanzeb Chaudhry, Don Estep, Trevor Giannini, Zachary Stevens, and Simon Tavener. Error estimation for the time to a threshold value in evolutionary partial differential equations, 2021.

[15] J.H. Chaudhry. *A posteriori* analysis and efficient refinement strategies for the Poisson–Boltzmann equation. *SIAM Journal on Scientific Computing*, 40(4):A2519—-A2542, 2018.

[16] J.H. Chaudhry, J.B. Collins, and J.N. Shadid. *A posteriori* error estimation for multi-stage Runge-Kutta IMEX schemes. *Applied Numerical Mathematics*, 117:36–49, Jul 2017.

[17] J.H. Chaudhry, D. Estep, V. Ginting, J.N. Shadid, and S. Tavener. *A posteriori* error analysis of IMEX multi-step time integration methods for advection–diffusion–reaction equations. *Computer Methods in Applied Mechanics and Engineering*, 285:730–751, 2015.

[18] J.H. Chaudhry, D. Estep, V. Ginting, and S.J. Tavener. *A posteriori* analysis of an iterative multi-discretization method for reaction-diffusion systems. *Computer Methods in Applied Mechanics and Engineering*, 267:1–22, 2013.

[19] J.H. Chaudhry, D. Estep, and S. Tavener. *A posteriori* error analysis for Schwarz overlapping domain decomposition methods. *arXiv e-prints*, page arXiv:1907.01139, Jul 2019.

[20] J.H. Chaudhry, D. Estep, S. Tavener, V. Carey, and J. Sandelin. *A posteriori* error analysis of two-stage computation methods with application to efficient discretization and the Parareal algorithm. *SIAM Journal on Numerical Analysis*, 54(5):2974–3002, 2016.

[21] J.H. Chaudry, D. Estep, Z. Stevens, and S. Tavener. Error estimation and uncertainty quantification for first time to a threshold value. *BIT Numerical Mathematics*, 61:275–307, 2021.

[22] K. Andrew Cliffe, Michael B. Giles, Robert Scheichl, and Aretha L. Teckentrup. Multilevel monte carlo methods and applications to elliptic pdes with random coefficients. *Computing and Visualization in Science*, 14:3–15, 2011.

[23] K.A. Cliffe, J. Collis, and P. Houston. Goal-oriented *a posteriori* error estimation for the travel time functional in porous media flows. *SIAM Journal of Scientific Computing*, 37(2):B127–B152, 2015.

[24] Nathan Collier, Abdul-Lateef Haji-Ali, Fabio Nobile, Erik von Schwerin, and Raul Tempone. A continuation multilevel monte carlo algorithm. *BIT Numerical Mathematics*, 55:399–432, 02 2014.

[25] J. B. Collins, D. Estep, and S. Tavener. *A posteriori* error analysis for finite element methods with projection operators as applied to explicit time integration techniques. *BIT Numerical Mathematics*, 55(4):1017–1042, 2015.

[26] J. B. Collins, D. Estep, and S. Tavener. *A posteriori* error analysis for finite element methods with projection operators as applied to explicit time integration techniques. *BIT Numerical Mathematics*, 55(4):1017–1042, 2015.

[27] James B. Collins, Don Estep, and Simon Tavener. *A posteriori* error estimation for the Lax–Wendroff finite difference scheme. *Journal of Computational and Applied Mathematics*, 263:299–311, 2014.

[28] B.N. Davis and R.J. LeVeque. Adjoint methods for guiding adaptive mesh refinement in tsunami modeling. *Pure and Applied Geophysics*, 173:4055–4074, 2016.

[29] M. Delfour, W. Hager, and F. Trochu. Discontinuous Galerkin methods for ordinary differential equations. *Math. Comp.*, 36(154):455–473, 1981.

[30] M. C. Delfour and F. Dubeau. Discontinuous polynomial approximations in the theory of one-step, hybrid and multistep methods for nonlinear ordinary differential equations. *Math. Comp.*, 47(175):169–189, S1–S8, 1986.

[31] Josef Diblík, Irada Dzhalladova, and Miroslava Růžičková. A dynamical system with random parameters as a mathematical model of real phenomena. *Symmetry*, 11(11), 2019.

[32] Josef Dick, Frances Kuo, Gareth Peters, and Ian Sloan. *Monte Carlo and Quasi-Monte Carlo Methods 2012*, volume 65. Springer, Berlin, Heidelberg, 2013.

[33] James F. Epperson. *An Introduction to Numerical Methods and Analysis*. Wiley-Interscience, 2007.

[34] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. Introduction to adaptive methods for differential equations. In *Acta Numerica, 1995*, Acta Numerica, pages 105–158. Cambridge Univ. Press, Cambridge, 1995.

[35] K. Eriksson, C. Johnson, and A. Logg. Explicit time-stepping for stiff ODEs. *SIAM Journal on Scientific Computing*, 25(4):1142–1157, 2004.

[36] D. Estep. *A posteriori* error bounds and global error control for approximation of ordinary differential equations. *SIAM J. Numer. Anal.*, 32(1):1–48, 1995.

[37] D. Estep. A short course on duality, adjoint operators, Green's functions, and *A Posteriori* error analysis. Unpublished, 2004.

[38] D. Estep. Error estimates for multiscale operator decomposition for multiphysics models. In J. Fish, editor, *Multiscale methods: bridging the scales in science and engineering*. Oxford University Press, USA, 2009.

[39] D. Estep, V. Ginting, and S. Tavener. *A posteriori* analysis of a multirate numerical method for ordinary differential equations. *Computer Methods in Applied Mechanics and Engineering*, 223:10–27, 2012.

[40] D. Estep, M. Holst, and D. Mikulencak. Accounting for stability: *a posteriori* error estimates based on residuals and variational analysis. *Comm. Numer. Methods Engrg.*, 18:15–30, 2002.

[41] D. Estep, M. Larson, and R. Williams. Estimating the error of numerical solutions of systems of reaction-diffusion equations. *Memoirs of the American Mathematical Society*, 696, 07 2000.

[42] D. Estep, A. Målqvist, and S. Tavener. Nonparametric density estimation for randomly perturbed elliptic problems I: Computational methods, *a posteriori* analysis, and adaptive error control. *SIAM Journal on Scientific Computing*, 31(4):2935–2959, 2009.

[43] George Fishman. *Monte Carlo: concepts, algorithms, and applications*. Springer-Verlag New York, 2013.

[44] Walter Gautschi. *Numerical Analysis*. Birkhäuser, 2011.

[45] 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.

[46] Michael B. Giles. Multilevel Monte Carlo Path Simulation. *Operations Research*, 56(3):607–617, 2008.

[47] Michael B. Giles. Multilevel monte carlo methods. *Acta Numerica*, 24, 2015.

[48] Stefan Heinrich. Multilevel monte carlo methods. In *Large-Scale Scientific Computing*, pages 58–67, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[49] A.M. Johansen. Monte carlo methods. In *International Encyclopedia of Education (Third Edition)*, pages 296–303. Elsevier, Oxford, third edition edition, 2010.

[50] A. Johansson, J. H. Chaudhry, V. Carey, D. Estep, V. Ginting, M. Larson, and S.J. Tavener. Adaptive finite element solution of multiscale PDE–ODE systems. *Computer Methods in Applied Mechanics and Engineering*, 287:150–171, 2015.

[51] Ralf Kornhuber and Evgenia Youett. Adaptive multilevel monte carlo methods for stochastic variational inequalities. *SIAM Journal on Numerical Analysis*, 56(4):1987–2007, 2018.

[52] Jacques Louis Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer Berlin, Heidelberg, 1 edition, 1971.

[53] Anders Logg. Multi-adaptive time integration. *Appl. Numer. Math.*, 48(3-4):339–354, mar 2004.

[54] Ryan G. McClarren. *Uncertainty Quantification and Predictive Computational Science*. Springer Cham, 2018.

[55] Nicholas Metropolis and S. Ulam. The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

[56] Walter Rudin. *Principles of mathematical analysis*. McGraw-Hill New York, 3d ed. edition, 1976.

[57] R. Scheichl, A. M. Stuart, and A. L. Teckentrup. Quasi-monte carlo and multilevel monte carlo methods for computing posterior expectations in elliptic inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*, 2017.

[58] Robert J. Serfling. *Approximation Theorems of Mathematical Statistics*. John Wiley and Sons, 1980.