MONASH University

Multilevel Monte Carlo Samplers in Bayesian Inverse problems

Chuntao Chen

A thesis submitted for the degree of Doctor of Philosophy at

Monash University in 2021

School of Mathematics

Doctoral Adviser
Tiangang Cui

Copyright notice

Abstract

This thesis explores and implements a new multilevel Monte Carlo sampler in Bayesian inverse problems. Inverse problems are important in the mathematical problems that we cannot directly observe the model parameters. Bayesian approach allows us to solve the inverse problems using limited number of data and prior information on the parameters. Monte Carlo method is widely used in calculating approximations of expectation of some functions in Bayesian inference. However, typical Monte Carlo method in Bayesian inference like Markov chain Monte Carlo (MCMC) is hard to apply in large-scale inverse problems since MCMC constructs a sequential Markov chain, which is hard to parallelize, and simulations that evaluating the posterior distribution require solving expensive forward model.

We combine the multilevel Monte Carlo and the optimization-based samplers, including Randomized-and-Then-Optimize (RTO) and Implicit Sampling, to address the challenges that classical MCMC faces, and implements the samplers in computationally costly Bayesian inverse problems including an ODE model and a PDE problem. Simulations using the optimization-based samplers like RTO can be parallelized which allows us to develop efficient MCMC algorithms or self-normalizing estimators to solve the inverse problems. Multilevel Monte Carlo is proven to significantly reduce the computational cost of Monte Carlo simulation, which helps us further improve the RTO method. To adapt the multilevel method on the optimization-based samplers, we develop the complexity theorem for multilevel self-normalizing estimators. The corresponding numerical experiments produce good results on RTO method, showing a high effective sample ratio in the importance sampling scheme, and the variances of the self-normalizing estimators converge when discretization size decreases. The computational results also validate the new complexity theorem for multilevel self-normalizing estimators. In addition, the complexity theorem provides a way to calculate the optimal sample size for the multilevel self-normalizing estimators. The thesis also implements a technique for the RTO method, that applies for parameters that changes dimension at every level.

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Chuntao Chen
08 Apr 2022

Acknowledgements

Thanks for the proof-reading service by the editor, Joan Gladwyn from Proper Words. Her service includes the grammar checking and format suggestions throughout this thesis. I would like to convey my thanks for the administrative support provided by the School of Mathematics at Monash University. I wish to thank John and Linda for helping me with all administrative requirements. I thank for the fellow PhD students and the friends I make in Monash University for making these years enjoyable. I would like to thank for the research and writing suggestions from the panelists, Tianhai Tian, Tim Garoni, Greg Markowsky and Hans De Sterck. Their invaluable feedback on all research milestones pushes and encourages me vastly. I am immensely grateful for my supervisor, Tiangang Cui, who mentors and guides me in researching and writing for the 4 year-long PhD program. I am grateful for his timely feedback all the time, especially in the thesis writing process. I would like to thank for his patience and encouragement that helps me go through several lock-down periods in the past two years.

Contents

Nomenclature

Symbols

α	convergence rate of variance in the complexity theorem
β	unnormalized weight
D	discretization map
F_h	forward model
g	biasing distribution
H	target distribution of RTO method
I_n	identiy matrix with dimension n
$\mathbb{I}_Q$	integral of quantity of interest
J	Jacobian of a function
l	level in multilevel method
L	likelihood function
n	dimension of parameter
N	sample size
p	prior function
P	vector of quantity of prey and predators
Q	quantity of interest
$\widehat{Q}_{\mathbf{1}}$	the first type of multilevel self-normalizing estimator
$\widehat{Q}_{\mathbf{2}}$	the second type of multilevel self-normalizing estimator
R	matrix R in QR factorization
T	transport mapping
U	the upper triangle matrix in QR factorization
w	weight in importance sampling
x	parameter vector in infinite-dimension space
$\boldsymbol{x}$	parameter vector in finite-dimension space
y	data

$\widehat{Y}$	the numerator in multilevel estimator $Q_{\mathbf{2}}$
z	normalizing constant
$\hat{z}$	the denominator in multilevel estimator $Q_{\mathbf{2}}$
ζ	convergence rate of variance in the complexity theorem
η	convergence rate of computation complexity in the complexity theorem
Γ_{obs}	covariance matrix of noise in the forward model
Γ_{prior}	covariance matrix of prior distribution
θ	unknown parameter in delta method
Υ	function of unknown parameter in delta method
$\lambda_{\boldsymbol{\xi}}$	scalar of random map in Implicit Sampling
$\boldsymbol{\xi}$	reference variable in transport mapping

Chapter 1

Introduction

Many mathematical modeling problems need to solve the inverse problems which are essentially to retrieve the unknown parameters in models when the noisy data is known. If the solution of the inverse problem is not unique, or the small perturbations of noise cause great changes of data, it is called an ill-posed inverse problem. Bayesian approach can remove the ill-posedness. Bayesian inference on the inverse problem [30, 28, 50] uses some prior information on parameters and constructs the posterior distribution using Bayes' theorem. In this way, the known data can cooperate with prior information to form the posterior distribution of parameters. Bayesian inverse problem can be formulated in the finite dimensional space and the infinite dimensional space [49, 40, 31]. However, the inverse problem in the infinite dimensional space is limited to the finite dimensional data, and people only want to retrieve a finite number of parameters, which turns the inverse problem into a discrete form [13, 29].

In practical scientific research and application, we usually want to calculate the Quantity of Interest (QoI), which is the expectation of some functions over the posterior in Bayesian inference. For example, in a prey and predator model, given some observations in past years, people might want to know the expected population of the prey at a future time. There are many ways to compute expectations, Monte Carlo method is one of them. The typical ways to simulate the samples in Bayesian inverse problem are Markov chain Monte Carlo (MCMC) methods [46, 17, 9] and sequential Monte Carlo (SMC) methods [8, 6, 32].

In practice, we usually cannot use Monte Carlo to simulate the posterior of Bayesian problem directly. We will sample a proposal density instead and use the corresponding weights, which is the ratio of the proposal density and the posterior. The posterior is obtained after weighting the proposal density. We have a self-normalizing estimator after the weighting process. In this thesis we use the optimization-based samplers to construct the proposal densities. The optimization-based samplers explored in this thesis are

Randomized-and-Then-Optimize (RTO) and Implicit Sampling. RTO was first developed by J. Bardsley [5] at 2014. RTO sampler is first to add the perturbation to the equation constructed from negative log of posterior, and then to solve the resulting optimization equation. Implicit Sampling has been applied in particle filters about a decade ago [10, 38, 39]. Implicit Sampling is quite like the RTO. It also requires solving an optimization equation constructed from negative log of posterior. In both samplers, the proposal densities are unnormalized. The resulting estimators of posteriors are biased self-normalizing estimators.

There are plenty of research on improving Monte Carlo. Multilevel Monte Carlo (MLMC) is a way to further increase the efficiency of Monte Carlo simulation. Multilevel idea was introduced to Monte Carlo simulation by S. Heinrich [23] in 2001. M. Giles first applied MLMC in a stochastic differential equation in 2008 [18]. He further summarizes the findings including generalization and extension in the article [19]. Multilevel estimator decomposes QoI into a telescope sum including the QoI's calculated from different levels, where the coarse levels require less computational cost, and the fine levels asks for more cost. The advantage of MLMC is that we can put most of the cost at the coarse levels and thus reduce the overall cost.

A search of the literature revealed few studies which implement the MLMC with the optimization-based samplers. On the other hand, there are few literatures on the complexity of the multilevel self-normalizing estimators. The importance and originality of this thesis are that it integrates MLMC with optimization-based samplers for solving computationally expensive inverse problems. When using these multilevel optimization-based samplers, we also analyze the complexity of the resulting multilevel ratio estimator, or multilevel self-normalizing estimator by means of the delta method. We demonstrate the new multilevel samplers on an ODE and a PDE problem. The numerical results from these problems also confirm the theoretical results of the complexity analysis. Besides these results, we also present the changing dimension technique in RTO method that applies in parameters which change dimension at each level, with an application in a tomography model.

This thesis is composed of seven themed chapters. Besides this introduction chapter and the conclusion chapter, Chapter 7, the remaining parts of the thesis proceeds as follows: Chapter 2 of this thesis will provide the background knowledge of Bayesian inverse problem, importance sampling, multilevel method and MCMC; Chapter 3 is concerned with the theoretical results for multilevel RTO method, which includes transport mapping, coupling techniques, RTO algorithm and the complexity of the resulting multilevel self-normalizing estimator; Chapter 4 is the numerical results of experiments of multilevel RTO method in prey and predator model and a PDE model, i.e., the direct applications of the third chapter; Chapter 5 introduces another optimization-based sampler, Implicit Sampling and its corresponding numerical results; Chapter 6 establishes a new technique to deal with the parameters which change dimension at every level, and includes the application with

multilevel RTO method in a tomography problem.

Chapter 2

Background and literature

This chapter will introduce the notation for this thesis, provide the literature review and introduce the problem setting. It starts with the inverse problem in a Bayesian framework. After that we review importance sampling, Markov Chain Monte Carlo and multilevel Monte Carlo method.

2.1 Bayesian inverse problem

An inverse problem is the process of retrieving the parameter from some observed data in a model that represents the parameter-to-observation map. Bayesian inference [50, 30, 49] offers a way to solve the inverse problem in the Bayesian framework. We first build a *prior* distribution of parameter, and build a *likelihood* function that can change or update the prior into *posterior* distribution of the parameters. The prior belief or prior distribution is the marginal distribution of parameters. The likelihood is the conditional distribution of data given the parameters, which can be considered as the bridge between the prior belief and the posterior. The prior distribution and the likelihood can vary with assumptions and hence the posterior will follow. We discuss this in the current section.

We will consider two scenarios. One has finite-dimension parameters and the second one has infinite-dimension parameters. In the second scenario, we also consider the hierarchical model.

2.1.1 Finite-dimension parameters

We start with an example. Suppose the dynamics of an ecological system take the form

$$\frac{\partial P}{\partial t} = f(P, t; \theta). \tag{2.1}$$

where $P \in \mathbb{R}^{d_P}$ is a vector of the size of some populations; t is the reference time; $\theta \in \mathbb{R}^{d_\theta}$ is the unknown parameters describing the ecosystem; $f : \mathbb{R}^{d_P} \times [0, T] \times \mathbb{R}^{d_\theta} \to \mathbb{R}^{d_P}$ is some function of P, t and θ. We collect observation of P at times t_i, $i = 1, 2, ..., d_T$:

$$y_i = \Theta\big(P(t_i)\big) + \epsilon_i, \ . \tag{2.2}$$

where $\Theta : \mathbb{R}^{d_T} \to \mathbb{R}^{dy}$ is the observation map, and ϵ_i is a zero mean noise. We denote $\boldsymbol{x} = \{P(t = 0), \theta\}$ including the initial value of P, and θ. Equations (2.1) and (2.2) define a parameter-to-observation map $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}^m$ which takes the form:

$$y = \mathcal{F}(\boldsymbol{x}) + e, \tag{2.3}$$

where $n = d_P + d_\theta$ is the dimension of parameter $\boldsymbol{x}$, and $m = d_T \times dy$ is the number of observation in time interval $[0, T]$ of this ecosystem model; e is assumed to be the Gaussian noise with zero mean and covariance Γ_{prior}. We often refer to the map $\mathcal{F}$ as the forward model.

The posterior $\pi(\boldsymbol{x}|y)$ is proportional to the product of prior $p(\boldsymbol{x})$ and the likelihood $\mathcal{L}(y|\boldsymbol{x})$ up to a constant z:

$$\pi(\boldsymbol{x}|y) = \frac{1}{z}\mathcal{L}(y|\boldsymbol{x})p(\boldsymbol{x}), \text{ where } z = \int \mathcal{L}(y|\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}. \tag{2.4}$$

We can numerically discretize the forward model by time t by Euler's method or the Runge–Kutta method. Suppose we use timestep h, then this defines a discretized model $F_h : \mathbb{R}^n \to \mathbb{R}^m$

$$y = F_h(\boldsymbol{x}) + e. \tag{2.5}$$

F_h converges to $\mathcal{F}$ as $h \to 0$.

The numerical discretization of the forward model also defines the discretized likelihood. Denoting the prior, discretized likelihood and posterior as $p(\boldsymbol{x})$, $L_h(y|\boldsymbol{x})$, and $\pi_h(\boldsymbol{x}|y)$ re-

spectively, the discretized posterior takes the form

$$\pi_h(\boldsymbol{x}|y) = \frac{1}{z_h} L_h(y|\boldsymbol{x})p(\boldsymbol{x}), \tag{2.6}$$

where z_h is a constant. We denote e is Gaussian with zero mean and covariance Γ_{obs}. We have the following likelihood function:

$$L_h(y|\boldsymbol{x}) = (2\pi)^{-\frac{n}{2}} |\det(\Gamma_{\text{obs}})|^{-\frac{1}{2}} \exp\Big(-\frac{1}{2}(F_h - y)^T \Gamma_{\text{obs}}^{-1}(F_h - y)\Big). \tag{2.7}$$

Suppose we want to predict P at some future time T_P, given the observation data. We can define the map from the parameter to the prediction of P at T_P as the Quantity of Interest(QoI), $Q(\boldsymbol{x})$, e.g., $Q(\boldsymbol{x}) \equiv P(T_P)$. The goal is to approximate the posterior expectation:

$$\mathbb{E}_\pi(Q) \approx \mathbb{E}_{\pi_h}(Q_h), \tag{2.8}$$

where Q_h is the QoI integrated over π_h.

Figure (2.1) is the prey and predator model [44], an example of the ecological system. The blue curve is the population of prey in an area while the red curve is the population size of predators in the same range. These two lines are directly generated from the model. Suppose we have the observed data of both populations, which are represented by the small circles or squares at time $t_i = 0, 1, 2, ..., 35$. The quantity of interest here can be the population size of prey or predators at a future time larger than 35.

2.1.2 Infinite-dimension parameters

In some problems, the parameters are functions in space. In this case, we assume infinite-dimension parameter $x(s)$, $s \in \Omega$ in separable Hilbert space $\mathscr{H}(\Omega)$, defined on a bounded domain $\Omega \in \mathbb{R}_{>0}$. Suppose there is a forward model, $\mathcal{F} : \mathscr{H} \to \mathbb{R}^m$ with a noise e,

$$y = \mathcal{F}(x) + e, \tag{2.9}$$

where $y \in \mathbb{R}^m$ is the response or data of this forward model and m is the dimension of the data. Typical problems with this setup include the inverse problem generated by Partial differential equation (PDE).

Suppose we have a prior measure μ_{prior} such that $u_{\text{prior}}(\mathscr{H}) = 1$. By Bayes' law, the

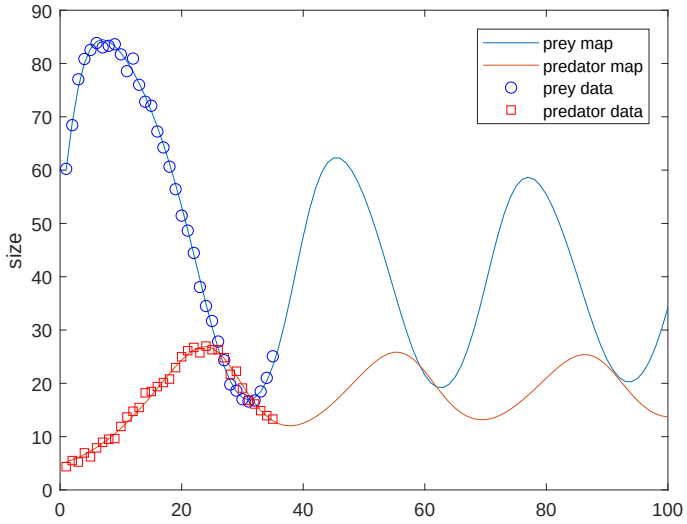

Figure 2.1: Prey and predator model

posterior measure μ_{post} on a volume $dx \in \mathscr{H}$ follows

$$\mu_{\text{post}}(dx) = \frac{1}{z}\mathcal{L}(y|x)\mu_{\text{prior}}(dx), \tag{2.10}$$

where $\mathcal{L}(y|x)$ is the likelihood function [49].

If we assume the noise is Gaussian, i.e., $e \sim \mathcal{N}(0, \Gamma_{\text{obs}})$, then the likelihood function $\mathcal{L}(y|x)$ of y becomes

$$\mathcal{L}(y|x) \propto \exp\Big(-\frac{1}{2}\big(y - \mathcal{F}(x)\big)^T \Gamma_{obs}^{-1}\big(y - \mathcal{F}(x)\big)\Big). \tag{2.11}$$

For some problems, the likelihood function may not be Gaussian, e.g., Poisson likelihood [2, 4] is used in the Positron Emission Tomography (PET) imaging problem in Chapter 6.

For computational purpose, we have to discretize the parameter $x(s)$ from the infinite space $\mathscr{H}$ to a finite vector space $\boldsymbol{x}_h = [x_1, x_2, \ldots, x_n]^T$ and the prior measure. We also need to discretize the forward model. Here is an example of a Gaussian process [25] prior and its discretization.

Example 2.1.1. We specify the Gaussian process prior using Laplace-like stochastic partial

differential equation (SPDE), which is

$$(\gamma - \Delta)^{\eta/2} x(s) = W(s), \text{ for } s \in \Omega \subset \mathbb{R}^d, \tag{2.12}$$

where $W(s)$ is a Gaussian white noise, Δ is the Laplace operator and γ is a scalar variable to model the correlation of the Gaussian process. The finite element method is employed to discretize the prior (2.12). We discretize $x(s)$ as $x(s) = \sum_{j=1}^{n} \phi_j(s) x_j$, where $\{\phi_j(s)\}_{j=1}^{n}$ is a set of locally compact basis functions and x_j is from the discretized vector $\boldsymbol{x}_h = [x_1, x_2, \ldots, x_n]^T$. The support of the basis functions is characterised by h.

In the Galerkin formulation of stochastic partial differential equations, $\boldsymbol{M}$, $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ are some sparse matrices with entries

$$\boldsymbol{M}_{ij} = \langle \phi_i, \phi_j \rangle, \text{ and } \boldsymbol{K} = \langle \nabla \phi_i, \nabla \phi_j \rangle.$$

If $\Omega \subseteq \mathbb{R}$, the we can select $\eta = 1$ for the case $d = 1$, so that the discretized covariance $\boldsymbol{P}_\gamma := \Gamma_{\text{prior}}^{-1}$ takes the form

$$\boldsymbol{P}_\gamma = (\gamma \boldsymbol{M} + \boldsymbol{K}). \tag{2.13}$$

For a given γ, the discretized prior is $p_h(\boldsymbol{x}_h) = \mathcal{N}(0, \boldsymbol{P}_\gamma^{-1})$.

We also need to discretize the forward model $F_{\text{h}} : \mathbb{R}^n \to \mathbb{R}^m$,

$$y = F_{\text{h}}(\boldsymbol{x}_h) + e, \tag{2.14}$$

where $y \in \mathbb{R}^m$ and $\boldsymbol{x}_h \in \mathbb{R}^n$. After the discretization, the prior measure μ_{prior} becomes a probability density in a finite-dimension space, $p_h(\boldsymbol{x}_h)$. The discretized posterior follows:

$$\pi_h(\boldsymbol{x}_h|y) = \frac{1}{z_h} L_h(y|\boldsymbol{x}_h) p_h(\boldsymbol{x}_h) \tag{2.15}$$

$$= \frac{1}{z_h} (2\pi)^{-\frac{m}{2}} |\det(\Gamma_{\text{obs}})|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \big(y - F_h(\boldsymbol{x}_h)\big)^T \Gamma_{\text{obs}}^{-1} \big(y - F_h(\boldsymbol{x}_h)\big) \right) p_h(\boldsymbol{x}_h). \tag{2.16}$$

2.1.3 Hierarchical model

The prior function $p_h(\boldsymbol{x}_h)$ can be a Gaussian random field [34, 47] or Gaussian process with a covariance operation Γ_{prior}, controlled by some hyperparameters. This section will introduce the hierarchical model involving hyperparameters [51, 2, 24]. In example (2.1.1), the hyperparameter controls the correlation of the prior. In the inverse problem, we may

also have other hyperparameters that can control other modeling assumptions, e.g., the variance of prior or the variance of observation noise in the likelihood.

If $p_h(\boldsymbol{x}_h)$ is a Gaussian prior with precision matrix $\boldsymbol{P}_\gamma$, then the discretized prior can take the form

$$p_h(\boldsymbol{x}_h|\delta,\gamma) \propto \delta^{\frac{n}{2}} \det(\boldsymbol{P}_{\boldsymbol{\gamma}})^{\frac{1}{2}} \exp\big(-\frac{\delta}{2}||\boldsymbol{x}_h||^2_{\boldsymbol{P}_\gamma}\big). \tag{2.17}$$

We can also impose some assumptions about the likelihood function. If noise e of the forward model takes the form

$$e \sim \mathcal{N}(0, \lambda^{-1}\Gamma_{\text{obs}}), \tag{2.18}$$

where $\lambda \in \mathbb{R}$ is the variance of the hyperparameter controlling the measurement process, and Γ_{obs} is a variance matrix of the observation, i.e., data.

We can also estimate the hyperparameter in the Bayesian framework,

$$\pi_h(\boldsymbol{x}_h, \lambda, \delta, \gamma|y) \propto \mathcal{L}_h(y|\boldsymbol{x}_h, \lambda) p_0(\boldsymbol{x}_h|\delta,\lambda) p_0(\lambda) p_0(\delta) p_0(\gamma). \tag{2.19}$$

The following densities are used to draw the hyperparameter samples. In some cases, *marginal posterior density* is used to draw hyperparameter samples,

$$p(\lambda, \delta, \gamma|y) \propto \mathcal{L}_h(y|\lambda,\delta,\gamma) p_0(\lambda) p_0(\delta) p_0(\delta). \tag{2.20}$$

The *marginal likelihood* $\mathcal{L}(y|\lambda,\delta,\gamma)$ is

$$\mathcal{L}_h(y|\lambda,\delta,\gamma) = \int \mathcal{L}_h(y|\boldsymbol{x}_h,\lambda) p_0(\boldsymbol{x}_h|\delta,\gamma) d\boldsymbol{x}_h. \tag{2.21}$$

The marginal likelihood above is also a part of the *conditional posterior*, which is used to draw parameter samples, $\boldsymbol{x}_h$:

$$p(\boldsymbol{x}_h|y,\lambda,\delta,\gamma) = \frac{1}{\mathcal{L}(y|\lambda,\delta,\gamma)} \mathcal{L}_h(y|\boldsymbol{x}_h,\lambda) p_0(\boldsymbol{x}_h|\delta,\gamma). \tag{2.22}$$

Since the marginal likelihood is usually unknown, it can be considered as a normalizing constant of the conditional posterior. Therefore, only the *unnormalized conditional posterior* is evaluated in practice:

$$f(\boldsymbol{x}_h|y,\lambda,\delta,\gamma) = \mathcal{L}_h(y|\boldsymbol{x}_h,\lambda) p_0(\boldsymbol{x}_h|\delta,\gamma). \tag{2.23}$$

2.2 Importance sampling

The posterior can be concentrated in some sub-domains of the parameter space instead of the whole support of the parameter space. In addition, the normalizing constant of the posterior is unknown. Thus, we can use importance sampling (see [35], chapter 9 of [43], and [45]) to help shape the unnormalized posterior and estimate the normalizing constant. Here we introduce importance sampling and the corresponding estimators for the expectation of QoI.

2.2.1 Importance Sampling

In general, we consider integrating some function $Q(\boldsymbol{x})$ over the target distribution π where the support of $\boldsymbol{x}$ is $\mathcal{X}$ for all $\boldsymbol{x} \in \mathcal{X}$. We define the integration by

$$\mathbb{I}_Q = \mathbb{E}_\pi\big(Q(\boldsymbol{x})\big) = \int_\pi Q(\boldsymbol{x})\pi(\boldsymbol{x})d\boldsymbol{x}. \tag{2.24}$$

To evaluate the integration, it is natural to discretize the space $\mathcal{X}$ and evaluate $Q(\cdot)$ and $\pi(\cdot)$ on each point of the discretized $\mathcal{X}$, and sum over the products $Q(\cdot)\pi(\cdot)$. Importance sampling calculates this integral by finding another distribution $g(\cdot)$ such that

$$\mathbb{I}_Q = \mathbb{E}_\pi\big(Q(\boldsymbol{x})\big) = \int Q(\boldsymbol{x})\frac{\pi(\boldsymbol{x})}{g(\boldsymbol{x})}g(\boldsymbol{x})d\boldsymbol{x} = \mathbb{E}_g\bigg(Q(\boldsymbol{x})\frac{\pi(\boldsymbol{x})}{g(\boldsymbol{x})}\bigg). \tag{2.25}$$

In other words, instead of integrating over the target density, we want to integrate over $g(\cdot)$. Importance sampling can be summarized in the following procedures:

1. Draw samples $\{\boldsymbol{x}^i\}_{i=1}^N$ from $g(\cdot)$;
2. Denoting $w(\boldsymbol{x}) = \frac{\pi(\boldsymbol{x})}{g(\boldsymbol{x})}$, calculate the ratio of densities:

$$w(\boldsymbol{x}^i) = \frac{\pi(\boldsymbol{x}^i)}{g(\boldsymbol{x}^i)}; \tag{2.26}$$

3. Estimate the integral by

$$\widehat{Q}_{IS} = \frac{1}{N}\sum_{i=1}^{N} Q(\boldsymbol{x}^i)w(\boldsymbol{x}^i). \tag{2.27}$$

To make the importance sampling estimator satisfy the strong law of large numbers (SLLN) and central limit theorem (CLT), we need to impose certain sufficient conditions on g and π.

Strong Law of Large Number. *If a sequence of i.i.d. variables X_i has a common finite expected value $\mathbb{E}(X_i) = \mu < \infty$ for $i = 1, 2 \ldots$, then the sample mean*

$$\overline{X}_N = \frac{\sum_{i=1}^{N} X_i}{N} \xrightarrow{a.s.} \mu \tag{2.28}$$

converges.

We impose that the support of function $Q(\boldsymbol{x})\pi(\boldsymbol{x})$ is a subset of support of $g(\boldsymbol{x})$, i.e., $supp\big(Q(\boldsymbol{x})\pi(\boldsymbol{x})\big) \subseteq supp\big(g(\boldsymbol{x})\big)$, which guarantees that the ratio of $Q(\boldsymbol{x})\pi(\boldsymbol{x})$ and $g(\boldsymbol{x})$ has finite expectation. Then by SLLN, $\widehat{Q}_{IS}$ converges to $\mathbb{I}_Q$.

Central Limit Theorem. *If a sequence of i.i.d. variables X_i has a common finite expected value $\mathbb{E}(X_i) = \mu$ and finite variance $\mathbb{V}(X_i) = \sigma^2$, then the following random variable*

$$\sqrt{N}(\overline{X}_N - \mu) \xrightarrow{d} \mathcal{N}(0, \sigma^2)$$

converges.

The variance of $Q(\boldsymbol{x})w(\boldsymbol{x})$ is finite only when its second moment is finite:

$$\begin{aligned}
\mathbb{E}_g\left(Q^2(\boldsymbol{x})\frac{\big(\pi(\boldsymbol{x})\big)^2}{\big(g(\boldsymbol{x})\big)^2}\right) &= \int Q^2(\boldsymbol{x})\frac{\big(\pi(\boldsymbol{x})\big)^2}{\big(g(\boldsymbol{x})\big)^2} g(\boldsymbol{x}) d\boldsymbol{x} \\
&= \int Q^2(\boldsymbol{x})\frac{\pi(\boldsymbol{x})}{g(\boldsymbol{x})}\pi(\boldsymbol{x}) d\boldsymbol{x} \\
&= \mathbb{E}_\pi\left(Q^2(\boldsymbol{x})\frac{\pi(\boldsymbol{x})}{g(\boldsymbol{x})}\right).
\end{aligned}$$

If the weighted second moment $\mathbb{E}_\pi\left(Q^2(\boldsymbol{x})\frac{\pi(\boldsymbol{x})}{g(\boldsymbol{x})}\right)$ is finite, then by CLT, the mean of $Q(\boldsymbol{x})w(\boldsymbol{x})$ converges to $\mathbb{I}_Q$ (sec 3.3.2 in [45]).

2.2.2 Self-normalizing estimator

In practice, the density ratio w is often known only up to a *normalizing constant*. Suppose we only know the ratio β of the unnormalized posterior $f(\cdot)$,

$$f(\cdot) = z\,\pi(\cdot), \tag{2.29}$$

and the biasing density $g(\cdot)$ up to a normalizing constant z,

$$\beta(\boldsymbol{x}) \equiv \frac{f(\boldsymbol{x})}{g(\boldsymbol{x})} = z\frac{\pi(\boldsymbol{x})}{g(\boldsymbol{x})}. \tag{2.30}$$

Here is a way to estimate z. Rearranging (2.30), we have:

$$w(\boldsymbol{x}) = \frac{\pi(\boldsymbol{x})}{g(\boldsymbol{x})} = \frac{\beta(\boldsymbol{x})}{z}. \tag{2.31}$$

The weight z is unknown in these two equations (2.30) and (2.31) above, and therefore we need to estimate z. First we integrate both sides of (2.31):

$$\int \frac{\pi(\boldsymbol{x})}{g(\boldsymbol{x})} g(\boldsymbol{x}) d\boldsymbol{x} = \int \frac{\beta(\boldsymbol{x})}{z} g(\boldsymbol{x}) d\boldsymbol{x}. \tag{2.32}$$

The left-hand side equals 1, given $supp(\pi) \subseteq supp(g)$, as both π and g are probability densities. Thus, we have

$$z = \int \beta(\boldsymbol{x}) g(\boldsymbol{x}) d\boldsymbol{x}. \tag{2.33}$$

This provides an estimator of z:

$$\hat{z} = \sum_{i=1}^{N} \beta(\boldsymbol{x}^i). \tag{2.34}$$

We can now apply the self-normalizing importance sampling scheme. First we estimate the weight using (2.31) and (2.34):

$$\hat{w}(\boldsymbol{x}^i) = \frac{\beta(\boldsymbol{x}^i)}{\sum_{i=1}^{N} \beta(\boldsymbol{x}^i)}. \tag{2.35}$$

Using the same estimator (2.27), we replace w with $\hat{w}$. Finally, the self-normalized estimator for $\mathbb{I}_Q$ is

$$\widehat{Q}^{SN} = \frac{\sum_{i=1}^{N} \beta(\boldsymbol{x}^i) Q(\boldsymbol{x}^i)}{\sum_{i=1}^{N} \beta(\boldsymbol{x}^i)}. \tag{2.36}$$

It is the ratio between the unbiased estimators. But for finite samples size, $\widehat{Q}^{SN}$ is biased, although we prove that the bias can decay fast with the increasing sample size.

2.2.3 Effective Sample Ratio

Even though any distribution $g(\cdot)$ with $supp\big(\pi(\boldsymbol{x})\big) \subseteq supp\big(g(\boldsymbol{x})\big)$ can be the biasing distribution, there apparently exist some better distributions than others. For instance, if $g(\boldsymbol{x})$ is closer to $\pi(\boldsymbol{x})$, then it is more efficient to obtain samples in the concentrated domain of $\pi(\boldsymbol{x})$. One way to estimate the efficiency of importance sampling scheme is the Effective Sample Ratio(ESR):

$$ESR \equiv \frac{\sum_{i=1}^{N} \big(w(\boldsymbol{x}^i)^2\big)}{\big(\sum_{i=1}^{N} w(\boldsymbol{x}^i)\big)^2}. \tag{2.37}$$

This value $0 < \text{ESR} \leq 1$. If the weight w is same for every $\boldsymbol{x}^i$, i.e., $g(\boldsymbol{x}^i) \propto \pi(\boldsymbol{x}^i)$ for all i, then ESR equals 1. For biasing distributions $g(\cdot)$, the larger ESR, the closer to the target distribution $\pi(\cdot)$.

2.3 Multilevel Monte Carlo

Multilevel method was applied to the Monte Carlo algorithm in a stochastic differential problem by Michael Giles [18]. Multilevel estimator has been proved to be more computationally efficient than the standard Monte Carlo given certain technical requirements [11, 18]. This subsection will introduce the multilevel estimator and its corresponding complexity theorem. This will be used in later chapters to accelerate the computation of posterior expectation.

Suppose we discretize a forward model with a discretization size h_l at the level $l \in \{0, 1, ..., L-1, L\}$. When increasing l, h_l becomes smaller and the model become more accurate. Simultaneously, large l requires a higher computation cost. The discretized forward model together with the potential prior discretization define the discretized posterior

$\pi_l(\boldsymbol{x}_l|y) \equiv \pi_{h_l}(\boldsymbol{x}_l|y)$ where $\boldsymbol{x}_l$ is the parameter at level l. Similarly we can also discretize the QoI using h_l, which leads to $Q_l \equiv Q_{h_l}$ Obviously, the discretized QoI, Q_l, takes more time to compute than $Q_{h_{l-1}}$.

The motivation of multilevel is to compute more samples at the coarser levels, like the zeroth level, and less at the finer levels. In this case a multilevel estimator has the following form,

$$\mathbb{E}_{\pi_L}[Q_L] = \mathbb{E}_{\pi_0}[Q_0] + \sum_{l=1}^{L} \big(\mathbb{E}_{\pi_l}[Q_l] - \mathbb{E}_{\pi_{l-1}}[Q_{l-1}]\big), \tag{2.38}$$

$$\mathbb{I}_Q \equiv \mathbb{E}_{\pi}[Q] \approx \mathbb{E}_{\pi_L}[Q_L] + \underbrace{\mathbb{E}_{\pi}[Q] - \mathbb{E}_{\pi_L}[Q_L]}_{\text{error}}. \tag{2.39}$$

where the first equation follows linearity of expectation. The summation in equation (2.38) is the telescope sum of the QoI. $\mathbb{E}_{\pi_l}[Q_l] - \mathbb{E}_{\pi_{l-1}}[Q_{l-1}]$ is evaluated at consecutive levels. $\mathbb{E}_{\pi}[Q] - \mathbb{E}_{\pi_L}[Q_L]$ is the error between the QoI evaluated at target density π and the discretized density Q_L which is estimated in equation (2.38).

We denote each term above as $\Delta Q_0 \equiv Q_0$ and $\Delta Q_l \equiv Q_l - Q_{l-1}$ for $l \geq 1$. The estimation of each ΔQ_l will be

$$\widehat{\Delta Q_l} \equiv \frac{1}{N_l} \sum_{i=1}^{N_l} \Delta Q_l(\boldsymbol{x}^i),$$

where $\boldsymbol{x}^i$ is the parameter and N_l is the sample size at level l. The multilevel estimator follows as

$$\widehat{Q}^{ML} \equiv \sum_{l=0}^{L} \widehat{\Delta Q_l}. \tag{2.40}$$

The computational complexity of $\widehat{Q}^{ML}$ is given in the following theorem [14, 18, 20]:

Theorem 2.3.1. Suppose the estimators $\widehat{Q}_l$, $\widehat{Q}^{ML}$ and $\widehat{\Delta Q_l}$ are defined as above. If there exist positive constants ζ, α, and η such that:

1. the discretization error is bounded as $\mathbb{E}\left[\widehat{\Delta Q_l} - \mathbb{I}_Q\right] = O(h_l^{\zeta})$;

2. the variance of each $\widehat{\Delta Q_l}$ defined above is bounded as $\mathbb{V}\left[\widehat{\Delta Q_l}\right] = O(\frac{h_l^{\alpha}}{N_l})$;

3. the computational complexity of $\widehat{\Delta Q_l}$ is bounded by $C_l = O(N_l h_l^{-\eta})$,

then the computational cost of multilevel estimator $\widehat{Q}^{ML}$ with a mean square error(MSE) $\mathbb{E}[\big(\widehat{Q}^{ML} - \mathbb{I}_Q\big)^2] < \epsilon^2$, is

$$C^{ML} = \begin{cases} O\big(\epsilon^{-2}\big), & \alpha > \eta, \\ O\Big(\epsilon^{-2}\big(\log(\epsilon)\big)^2\Big), & \alpha = \eta, \\ O\big(\epsilon^{-2-(\eta-\alpha)/\zeta}\big), & 0 < \alpha < \eta. \end{cases}$$

This theorem states that the computation complexity, or the computation cost, when imposing a certain MSE, depends on the converging rate of bias, variance of $\widehat{\Delta Q_l}$, and the computational cost of $\widehat{\Delta Q_l}$ in the telescope sum. The first condition deals with the discretization error [21]. This error is determined by the numerical method [41] used during the computation. For example, if we use the Euler's method, whose error at a given time is proportional to the step size, then the rate ζ equals 1. The other two conditions also depict the relationship with sample size N_l. Ideally, we want the variance of $\widehat{\Delta Q_l}$ to decay at a faster rate than the increasing rate of the computational cost of $\widehat{\Delta Q_l}$, which falls in the case $\alpha > \eta$. In this case, the cost $C^{ML} = O(\epsilon^{-2})$. If the converging rate of variance is less than the increasing rate of cost, then the computation cost of $\widehat{Q}^{ML}$ increases greatly, which is in the case $\alpha < \eta$. In this case, multilevel Monte Carlo might be worse than the standard Monte Carlo. The complexity of standard Monte Carlo has a complexity $C^{MC} = O(\epsilon^{-\iota})$ where $\iota > 2$ [1]. Different converging rates will be discussed in greater depth in later chapters.

2.4 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) (chapter 12 of [35]) is a set of algorithms to construct a Markov chain whose stationary or equilibrium distribution is the target distribution. It is also broadly used as a solution to solve Bayesian inverse problems [16, 26]. The following subsections will introduce the basic properties of Markov chain and bring out the Metropolis–Hastings algorithm [22, 37]. Here we only discuss the discrete-time Markov chain.

A sequence of random variables $\boldsymbol{x}^0$, $\boldsymbol{x}^1$, ... defined on a state space $\mathcal{X}$, is a Markov chain if it satisfies the Markov property:

$$\Pr(\boldsymbol{x}^{t+1} = \boldsymbol{\psi} | \boldsymbol{x}^t = \boldsymbol{\chi}, \ \ldots, \ \boldsymbol{x}^0 = \boldsymbol{\tau}) = \Pr(\boldsymbol{x}^{t+1} = \boldsymbol{\psi} | \boldsymbol{x}^t = \boldsymbol{\chi}). \tag{2.41}$$

That is, the distribution of $\boldsymbol{x}^{t+1}$ only relies on its previous consecutive state $\boldsymbol{x}^t$. If the transition probability $\Pr(\boldsymbol{x}^{t+1} = \boldsymbol{\psi}|\boldsymbol{x}^t = \boldsymbol{\chi}) = \Pr(\boldsymbol{x}^t = \boldsymbol{\psi}|\boldsymbol{x}^{t-1} = \boldsymbol{\chi})$ for all t, i.e., independent of t, then it is expressed as the *transition function*, $A(\boldsymbol{\chi}, \boldsymbol{\psi})$. If the state space is continuous, then transition function is also called the transition kernel and often written as $k(\boldsymbol{\chi}, \boldsymbol{\psi})$.

2.4.1 Discrete case

This subsection considers $\mathcal{X}$ as a finite and discrete space. Based on the nature of probability mass function, $\sum_{\boldsymbol{\psi}} A(\boldsymbol{\chi}, \boldsymbol{\psi}) = 1$ for all $\boldsymbol{\chi}$. This paragraph will briefly discuss two special properties of state $\boldsymbol{\chi}$.

1. A state $\boldsymbol{\chi}$ is *irreducible* if it communicates with any other state, that is, $\boldsymbol{\chi}$ has a nonzero probability of moving from itself to another state and then coming back in finite steps.

2. A state $\boldsymbol{\chi}$ is *aperiodic* if the greatest common divisor of $\{n : A^{(n)}(\chi, \chi) > 0\}$ is 1, where $A^{(n)}(\boldsymbol{\chi}, \boldsymbol{\chi})$ is the n-step transition function from state $\boldsymbol{\chi}$ to itself.

The first property says that no proper "closed" subset of states exists other than the whole Markov chain. The second property demonstrates that the lengths of all transient "path" do not have a nontrivial common divisor. If every state in $\mathcal{X}$ is irreducible and aperiodic, then we say the Markov chain is irreducible and aperiodic. The motivation of stating these two properties is to bring out the theorem of stationary distribution.

Theorem 2.4.1. If the Markov chain in a finite state is irreducible and aperiodic, then, $A^{(n)}(\boldsymbol{\chi}, \boldsymbol{\psi}) = \Pr(\boldsymbol{x}^n = \boldsymbol{\psi}|\boldsymbol{x}^0 = \boldsymbol{\chi})$ as a probability measure on $\boldsymbol{\psi}$ converges to the stationary distribution $\pi(\boldsymbol{\psi})$ at an exponential rate, that is, there exists $0 < r < 1$ and $c > 0$ such that

$$\int |A^{(n)}(\boldsymbol{\chi}, \boldsymbol{\psi}) - \pi(\boldsymbol{\psi})| d\boldsymbol{\psi} \leq cr^n \tag{2.42}$$

The theorem above implies that no matter what the initial state is, given a sufficiently long time, the distribution of states will converge to a stationary distribution controlled by the transition function. That is, if one can design the transition function such that the stationary distribution is the target distribution, one can obtain samples from the target distribution after discarding the first few states.

2.4.2 Continuous case

If the state space $\mathcal{X}$ is continuous, then we can still construct a discrete-time Markov chain on $\mathcal{X}$. The state $\boldsymbol{x}^t$ is from a continuous space and the transition kernel is now a continuous function. We again need to impose some conditions on the Markov chain so that it converges to the stationary distribution. The condition introduced here is the detailed balance.

Theorem 2.4.2. Suppose π is a distribution on $\mathcal{X}$ and $k(\boldsymbol{\chi}, \boldsymbol{\psi})$ is the transition kernel of an ergodic Markov chain. If the following detailed balance condition:

$$\pi(\boldsymbol{\psi})k(\boldsymbol{\psi}, \boldsymbol{\chi}) = \pi(\boldsymbol{\chi})k(\boldsymbol{\chi}, \boldsymbol{\psi}),$$

holds for all $\boldsymbol{\psi}$ and $\boldsymbol{\chi} \in \mathcal{X}$, the π is the stationary distribution of the Markov chain with kernel function k.

Detailed balance condition is sufficient but not necessary for an equilibrium of the Markov chain. We introduce it here since the MCMC sampler depicted later meets the condition.

2.4.3 MCMC sampler

Here is an MCMC sampler that applies the Markov chain theory. The procedures of the Metropolis–Hastings algorithm [37, 22] are: Given a current state $\boldsymbol{x}^t = \boldsymbol{\chi}$:

1. Draw a sample $\boldsymbol{\psi}$ from the proposal distribution $q(\boldsymbol{\psi})$;
2. Accept the sample $\boldsymbol{x}^{t+1} = \boldsymbol{\psi}$ if a random sample $U \sim Uniform[0, 1]$

$$U \leq r\big(\boldsymbol{\chi}, \boldsymbol{\psi}\big) \equiv \min\{1, \frac{\pi(\boldsymbol{\psi})q(\boldsymbol{\psi}, \boldsymbol{\chi})}{\pi(\boldsymbol{\chi})q(\boldsymbol{\chi}, \boldsymbol{\psi})}\} \tag{2.43}$$

   otherwise $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t = \boldsymbol{\chi}$.

Step 2 decides whether to accept a new sample $\boldsymbol{\psi}$ or stay in the current state $\boldsymbol{\chi}$. The notation r in step 2 represents the acceptance probability. The fraction of accepted samples from the proposal distribution over N samples is called the acceptance rate.

The transition kernel $k(\boldsymbol{\chi}, \boldsymbol{\psi})$ for the Metropolis–Hasting sampler takes the form:

$$k(\boldsymbol{\chi}, \boldsymbol{\psi}) = q(\boldsymbol{\chi}, \boldsymbol{\psi})r\left(\boldsymbol{\chi}, \boldsymbol{\psi}\right) + \delta(\boldsymbol{\chi} - \boldsymbol{\psi}) \int q(\boldsymbol{\chi}, \boldsymbol{\psi})\left(1 - r\left(\boldsymbol{\chi}, \boldsymbol{\psi}\right)\right) d\boldsymbol{\psi},$$

where $\delta(\boldsymbol{\chi} - \boldsymbol{\psi})$ is the Dirac delta function. The integral is the probability of rejecting all possible $\boldsymbol{\psi}$ when the current state is $\boldsymbol{\chi}$. The product of Dirac delta function and the integral is the probability of starting from $\boldsymbol{\chi}$ to $\boldsymbol{\psi}$. Detailed balance holds for the Metropolis-Hasting sampler. When $\boldsymbol{\chi} = \boldsymbol{\psi}$, equation (2.4.2) holds trivially. When $\boldsymbol{\chi} \neq \boldsymbol{\psi}$, the right-hand side of equation (2.4.2) is

$$\begin{aligned}
\pi(\boldsymbol{\chi})k(\boldsymbol{\chi}, \boldsymbol{\psi}) &= \pi(\boldsymbol{\chi})q(\boldsymbol{\chi}, \boldsymbol{\psi})r\left(\boldsymbol{\chi}, \boldsymbol{\psi}\right) \\
&= \pi(\boldsymbol{\chi})q(\boldsymbol{\chi}, \boldsymbol{\psi})\text{min}\{1, \frac{\pi(\boldsymbol{\psi})q(\boldsymbol{\psi}, \boldsymbol{\chi})}{\pi(\boldsymbol{\chi})q(\boldsymbol{\chi}, \boldsymbol{\psi})}\} \\
&= \text{min}\{\pi(\boldsymbol{\chi})q(\boldsymbol{\chi}, \boldsymbol{\psi}), \pi(\boldsymbol{\psi})q(\boldsymbol{\psi}, \boldsymbol{\chi})\} \\
&= \pi(\boldsymbol{\psi})q(\boldsymbol{\psi}, \boldsymbol{\chi})\text{min}\{1, \frac{\pi(\boldsymbol{\chi})q(\boldsymbol{\chi}, \boldsymbol{\psi})}{\pi(\boldsymbol{\psi})q(\boldsymbol{\psi}, \boldsymbol{\chi})}\} \\
&= \pi(\boldsymbol{\psi})q(\boldsymbol{\psi}, \boldsymbol{\chi})r\left(\boldsymbol{\psi}, \boldsymbol{\chi}\right) \\
&= \pi(\boldsymbol{\psi})k(\boldsymbol{\psi}, \boldsymbol{\chi}).
\end{aligned}$$

This concludes the detailed balance condition. Therefore Metropolis-Hasting algorithm can produce the stationary distribution of π.

Proposal density

We can see that the proposal density is the key to the efficiency of MCMC. A proposal density that is close to the target distribution can increase the acceptance rate of the Metropolis–Hastings algorithm. Extensive literature exists on the proposal density of MCMC. The stochastic Newton method uses the Hessian information to form a proposal distribution [36]. Optimization-based samplers like the Randomized-and-Then-Optimized (RTO) method can also provide a proposal density that is close to the target distribution.

Chapter 3

Multilevel RTO method

This chapter builds the necessary components of the multilevel RTO method. We first introduce the self-normalizing estimator and transport mapping method in the first two sections. The third section concerns the RTO algorithm. The last section verifies the complexity theorem of the multilevel self-normalizing estimator.

3.1 Multilevel self-normalizing estimator

We mentioned the self-normalizing importance sampling and multilevel Monte Carlo in Chapter 2. We want to combine these two methods to further increase the computational efficiency of problems involving unnormalized posteriors. In many cases, we have the unnormalized posterior $f(\boldsymbol{x})$ with the following identity:

$$f(\boldsymbol{x}) = z\pi(\boldsymbol{x}), \tag{3.1}$$

where $\pi(\boldsymbol{x})$ is the true posterior and z is the normalizing constant. This is the same formula we described in the importance sampling section of Chapter 2. This formula does not include any discretization process yet.

In this chapter, we will use discretized quantities and discretized models. We denote the discretization size as h. If we want to apply the multilevel idea of section 2.3 when evaluating QoI, then we denote the unnormalized posterior, true posterior and the corresponding normalizing constant with discretization size h_l:

$$f_{h_l}(\boldsymbol{x}_{h_l}) = z_{h_l}\pi_{h_l}(\boldsymbol{x}_{h_l}), \tag{3.2}$$

where $l = 0, 1, 2, ..., L$. The value of h_l differs across levels. In numerical experiments of

later chapters, we set $h_l = \frac{1}{2}h_{l-1}$. To simplify the notation, the subscripted quantity h_l is simplified as l hereafter. For example, we write $\boldsymbol{x}_{h_l} = \boldsymbol{x}_l$ for $l = 0, 1, 2, ..., L$.

The following paragraphs establish the importance sampling method in the discretized parameter $\boldsymbol{x}_l$ with discretization size h_l. Suppose there is a biasing density $g_l(\cdot)$ and we have the ratio β_l between $f_l(\cdot)$ and $g_l(\cdot)$

$$\beta_l(\boldsymbol{x}_l) \equiv \frac{f_l(\boldsymbol{x}_l)}{g_l(\boldsymbol{x}_l)} = z_l \frac{\pi_l(\boldsymbol{x}_l)}{g_l(\boldsymbol{x}_l)} \tag{3.3}$$

where the normalizing constant z_l is estimated by the following integration:

$$z_l = \int \beta_l(\boldsymbol{x}_l) g_l(\boldsymbol{x}_l) d\boldsymbol{x}_l. \tag{3.4}$$

Currently, we have not introduced a sample-based estimator, and therefore we assume that we can get the true value of all quantities here. Our goal is to estimate the following expectation under posterior density π, a quantity we discussed in the importance sampling section in chapter 2:

$$\mathbb{I}_Q = \mathbb{E}_\pi\big(Q(\boldsymbol{X})\big) = \int_\pi Q(\boldsymbol{x})\pi(\boldsymbol{x})d\boldsymbol{x}_l, \tag{3.5}$$

where Q is the quantity of interest (QoI). This $\mathbb{I}_Q$ is not estimable in the finite parameter space. We can only estimate the discretized $\mathbb{I}_{Q_l}$:

$$\mathbb{I}_{Q_l} = \mathbb{E}_{\pi_l}\big(Q_l(\boldsymbol{X}_l)\big) = \int_{\pi_l} Q_l(\boldsymbol{x}_l)\pi_l(\boldsymbol{x}_l)d\boldsymbol{x}_l. \tag{3.6}$$

$\boldsymbol{X}_l$ is from a finite-dimension space, and thus we can manage a sample-based estimator for it.

We are now ready for the setup of importance sampling. If we cannot estimate in the true posterior π_l, as is usually the case, then we will find a biasing density g_l such that

$$\mathbb{I}_{Q_l} = \mathbb{E}_{\pi_l}\big(Q_l(\boldsymbol{X}_l)\big) = \int Q_l(\boldsymbol{x}_l)\frac{\pi_l(\boldsymbol{x}_l)}{g_l(\boldsymbol{x}_l)}g_l(\boldsymbol{x}_l)d\boldsymbol{x}_l = \mathbb{E}_{g_l}\left(Q_l(\boldsymbol{X}_l)\frac{\pi_l(\boldsymbol{X}_l)}{g_l(\boldsymbol{X}_l)}\right). \tag{3.7}$$

After replacing $\frac{\pi_l(\boldsymbol{X}_l)}{g_l(\boldsymbol{X}_l)}$ with $w_l(\boldsymbol{X}_l)$, this results in:

$$\mathbb{I}_{Q_l} = \mathbb{E}_{g_l}\big(Q_l(\boldsymbol{X}_l)w_l(\boldsymbol{X}_l)\big); \tag{3.8}$$

and if w_l is only known up to a constant, it results in:

$$\begin{aligned}\mathbb{I}_{Q_l} &= \mathbb{E}_{g_l}\big(Q_l(\boldsymbol{X}_l)w_l(\boldsymbol{X}_l)\big)\\ &= \mathbb{E}_{g_l}\big(Q_l(\boldsymbol{X}_l)\frac{1}{z_l}\beta_l(\boldsymbol{X}_l)\big)\\ &= \frac{1}{z_l}\mathbb{E}_{g_l}\big(Q_l(\boldsymbol{X}_l)\beta_l(\boldsymbol{X}_l)\big)\\ &= \frac{1}{\mathbb{E}_{g_l}\big(\beta_l(\boldsymbol{X}_l)\big)}\mathbb{E}_{g_l}\big(Q_l(\boldsymbol{X}_l)\beta_l(\boldsymbol{X}_l)\big),\end{aligned} \tag{3.9}$$

where equation (3.9) derives from the normalizing constant estimation (3.4). All the equations above follow the logic of the importance sampling section of Chapter 2, except the last line (3.9), which is the ratio estimator for $\mathbb{I}_{Q_l}$. We need to note that this ratio still represents an unbiased estimator of integral for infinite samples. However, when it comes to finite samples, this ratio is biased. More detail is involved when we discuss the estimators of finite samples later in this section. Another highlight is that even though z is an invariant constant, the discretized version of z, z_l changes when the level changes. It becomes more accurate with smaller h_l.

The next step is to apply the multilevel idea using the quantities we re-defined above. We follow the same notation of QoI, Q_l, in the scenario of the multilevel Monte Carlo in section 2.3. We need to consider how to adapt the multilevel Monte Carlo when we only know the unnormalized posterior. This multilevel estimator should include two schemes: the first is the self-normalizing scheme (2.36):

$$\widehat{Q}_l^{SN} = \frac{\sum_{i=1}^{N}\beta(\boldsymbol{X}_l^i)Q(\boldsymbol{X}_l^i)}{\sum_{i=1}^{N}\beta(\boldsymbol{X}_l^i)}, \tag{3.10}$$

which is the estimator of (3.9), and the second scheme is the telescope sum as in (2.39):

$$\mathbb{E}_{\pi_L}[Q_L] = \mathbb{E}_{\pi_0}[Q_0] + \sum_{l=1}^{L}\big(\mathbb{E}_{\pi_l}[Q_l] - \mathbb{E}_{\pi_{l-1}}[Q_{l-1}]\big). \tag{3.11}$$

One way to construct the multilevel estimator is to replace every Q_l in the telescope

sum (3.11) with its corresponding self-normalizing estimator. We name it estimator 1, $\widehat{Q}_{\mathbf{1}}$:

$$\widehat{Q}_{\mathbf{1}} \equiv \widehat{Q}_0^{SN} + \sum_{l=1}^{L} (\widehat{Q}_l^{SN} - \widehat{Q}_{l-1}^{SN}), \tag{3.12}$$

where each $\widehat{Q}_l^{SN}$ has the same structure as (3.10), but the sample size N becomes N_l for each level l:

$$\widehat{Q}_l^{SN} = \frac{\sum_{i=1}^{N_l} \beta(\boldsymbol{X}_l^i) Q(\boldsymbol{X}_l^i)}{\sum_{i=1}^{N_l} \beta(\boldsymbol{X}_l^i)}. \tag{3.13}$$

In this estimator, we need to calculate $1 + 2L$ times this $\widehat{Q}_l^{SN}$. There are ways to make $\boldsymbol{x}_l$ and $\boldsymbol{x}_{l-1}$ similar to each other so that the covariance between each $\widehat{Q}_l^{SN}$ is small. One way is called "coupling", which will be discussed in the next section.

One can consider this estimator as one way to combine (3.10) and (3.11). A paper on sequential Monte Carlo [7] shows a similar estimator to this where each term in the telescope sum is a sequential Monte Carlo estimator. This following inequality from this sequential Monte Carlo paper states that under the scenario of sequential Monte Carlo, the mean square error (MSE) of $\widehat{Q}_{\mathbf{1}}$ is bounded as follows:

$$\begin{aligned}\mathbb{E}(\widehat{Q}_{\mathbf{1}} - \mathbb{E}_{\pi_L}[Q_L]) \leq & c_1\Big(\frac{1}{N_0} + \sum_{l=1}^{L} \frac{1}{N_{l-1}} \Big|\Big|\frac{z_{l-1} f_l}{z_l f_{l-1}} - 1\Big|\Big|_\infty^2 \\ & + \sum_{1 \leq l < q \leq L} \Big|\Big|\frac{z_{l-1} f_l}{z_l f_{l-1}} - 1\Big|\Big|_\infty \Big|\Big|\frac{z_{q-1} f_q}{z_q f_{q-1}} - 1\Big|\Big|_\infty \Big(\frac{c_2^{q-1}}{N_{l-1}} + \frac{1}{N_{l-1}^{1/2} N_{q-1}}\Big)\Big),\end{aligned}$$

where $c_1, c_2 \in (0, 1)$ are some constants. The notation $||\cdot||_\infty$ means the maximum norm and $||\cdot||_\infty^2$ means the square of the maximum norm. The complexity analysis of this estimator is to assume

$$\mathbb{V}_l \equiv \Big|\Big|\frac{z_{l-1} f_l}{z_l f_{l-1}} - 1\Big|\Big|_\infty^2 = O(h_l^s), \tag{3.14}$$

which plays the same role as $\mathbb{V}\left[\widehat{\Delta Q_l}\right]$ in the standard Multilevel Monte Carlo theorem (2.3.1). The complexity result is derived from placing the square of maximum norm with $O(h_l^s)$ in the MSE inequality. The details of complexity result are shown in the papers [7, 27]. More research about the theory of multilevel sequential Monte Carlo is in [33]. This is the result applied to the sequential Monte Carlo [15] but we are now investigating

the performance for standard Monte Carlo. The next paragraph is to introduce another multilevel estimator for standard Monte Carlo.

The other multilevel estimator is to replace both the denominator and numerator in (3.10) with a multilevel estimator. The following paragraphs introduce the notations of the estimator 2, $\widehat{Q}_{\mathbf{2}}$. The multilevel telescope sum of the numerator in (3.9) is

$$\begin{aligned}\mathbb{E}_{g_L}\big(Q_L(\boldsymbol{X}_L)\beta_L(\boldsymbol{X}_L)\big) =&\mathbb{E}_{g_0}\big(Q_0(\boldsymbol{X}_0)\beta_0(\boldsymbol{X}_0)\big)+\\ &\sum_{l=1}^{L}\mathbb{E}_{g_l}\big(Q_l(\boldsymbol{X}_l)\beta_l(\boldsymbol{X}_l)\big)-\mathbb{E}_{g_{l-1}}\big(Q_{l-1}(\boldsymbol{X}_{l-1})\beta_{l-1}(\boldsymbol{X}_{l-1})\big).\end{aligned} \tag{3.15}$$

We approximate every $Q(\cdot)\beta(\cdot)$ term by its corresponding estimator:

$$\mathbb{E}_{g_0}\big(Q_0(\boldsymbol{X}_0)\beta_l(\boldsymbol{X}_0)\big)\approx\frac{1}{N_0}\sum_{i=1}^{N_0}\hat{\beta}_0(\boldsymbol{X}_0^i)\widehat{Q}_0(\boldsymbol{X}_l^i), \tag{3.16}$$

and

$$\begin{aligned}&\mathbb{E}_{g_l}\big(Q_l(\boldsymbol{X}_l)\beta_l(\boldsymbol{X}_l)\big)-\mathbb{E}_{g_{l-1}}\big(Q_{l-1}(\boldsymbol{X}_{l-1})\beta_{l-1}(\boldsymbol{X}_{l-1})\big)\\ \approx&\frac{1}{N_l}\sum_{i=1}^{N_l}\hat{\beta}_l(\boldsymbol{X}_l^i)\widehat{Q}_l(\boldsymbol{X}_l^i)-\hat{\beta}_{l-1}(\boldsymbol{X}_{l-1}^i)\widehat{Q}_{l-1}(\boldsymbol{X}_{l-1}^i).\end{aligned} \tag{3.17}$$

We have listed the necessary components of the multilevel estimator of the numerator. The next step is to summarize and name it. We denote $\widehat{Y}_L$ as a multilevel estimator of $\mathbb{E}_{g_L}\big(Q_L(\boldsymbol{X}_L)\beta_L(\boldsymbol{X}_L)\big)$ in (3.9) where $\widehat{Y}_L$ is

$$\widehat{Y}_L\equiv\sum_{l=0}^{L}\widehat{\Delta Y_l}, \tag{3.18}$$

with

$$\widehat{\Delta Y_l}=\begin{cases}\frac{1}{N_0}\sum\limits_{i=1}^{N_0}\hat{\beta}_0(\boldsymbol{X}_0^i)\widehat{Q}_0(\boldsymbol{X}_l^i), & l=0;\\ \frac{1}{N_l}\sum\limits_{i=1}^{N_l}\hat{\beta}_l(\boldsymbol{X}_l^i)\widehat{Q}_l(\boldsymbol{X}_l^i)-\hat{\beta}_{l-1}(\boldsymbol{X}_{l-1}^i)\widehat{Q}_{l-1}(\boldsymbol{X}_{l-1}^i), & l=1,2,...,L.\end{cases} \tag{3.19}$$

$\boldsymbol{X}_l^i$ is from density g_l at the level l. N_l is the sample size at level l. Note that in the second line of (3.19), $\boldsymbol{X}_l^i$ and $\boldsymbol{X}_{l-1}^i$ should be coupled. This coupling can correlate $\hat{\beta}_l(\boldsymbol{X}_l^i)\widehat{Q}_l(\boldsymbol{X}_l^i)$ and $\hat{\beta}_{l-1}(\boldsymbol{X}_{l-1}^i)\widehat{Q}_{l-1}(\boldsymbol{X}_{l-1}^i)$, to reduce the variance of $\widehat{\Delta Y}_l$. If the reducing of the variance of $\widehat{\Delta Y}_l$ does not compensate the cost of it, and thus the multilevel estimator is less efficient than standard Monte Carlo, i.e., single level Monte Carlo. This was discussed when introducing the complexity theorem in section 2.3. In this thesis, the coupling method is achieved by transport mapping, which will be discussed in the section 3.2.

After we finish the above procedures with the numerator, we can investigate the denominator in (3.9). Expanding it in a telescope sum gives the following:

$$\mathbb{E}_{g_L}\big(\beta_L(\boldsymbol{X}_L)\big) = \mathbb{E}_{g_0}\big(\beta_0(\boldsymbol{X}_0)\big) + \sum_{l=1}^{L} \mathbb{E}_{g_l}\big(\beta_l(\boldsymbol{X}_l)\big) - \mathbb{E}_{g_{l-1}}\big(\beta_{l-1}(\boldsymbol{X}_{l-1})\big). \tag{3.20}$$

The next step that comes naturally is to approximate the denominator $\sum_{i=1}^{N}\beta(\boldsymbol{x}_l^i)$ in (3.9) with a multilevel estimator where

$$\mathbb{E}_{g_0}\big(\beta_0(\boldsymbol{X}_0)\big) \approx \frac{1}{N_0}\hat{\beta}_0(\boldsymbol{X}_0^i),$$

and

$$\mathbb{E}_{g_l}\big(\beta_l(\boldsymbol{X}_l)\big) - \mathbb{E}_{g_{l-1}}\big(\beta_{l-1}(\boldsymbol{X}_{l-1})\big) \approx \frac{1}{N_l}\sum_{i=1}^{N_l}\hat{\beta}_l(\boldsymbol{X}_l^i) - \hat{\beta}_{l-1}(\boldsymbol{X}_{l-1}^i).$$

Likewise, we denote $\hat{z}_L$ as the multilevel estimator of $\mathbb{E}_{g_l}\big(\beta_l(\boldsymbol{X}_l)\big)$ in (3.9), which is estimating the normalizing constant z_l, where

$$\hat{z}_L \equiv \sum_{l=0}^{L}\widehat{\Delta\beta}_l, \tag{3.21}$$

and

$$\widehat{\Delta\beta}_l = \begin{cases} \frac{1}{N_0}\sum_{i=1}^{N_0}\hat{\beta}_0(\boldsymbol{X}_0^i), & l=0; \\ \frac{1}{N_l}\sum_{i=1}^{N_l}\hat{\beta}_l(\boldsymbol{X}_l^i) - \hat{\beta}_{l-1}(\boldsymbol{X}_{l-1}^i), & l=1,2,...,L. \end{cases} \tag{3.22}$$

As when discussing numerator $\widehat{Y}_L$, $\boldsymbol{X}_l^i$ is from density g_l at the level l and N_l is the

sample size at level l. $\boldsymbol{X}_l^i$ and $\boldsymbol{X}_{l-1}^i$ should also be coupled variables to reduce the variance of $\hat{\beta}_l(\boldsymbol{X}_l^i)$ and $\hat{\beta}_{l-1}(\boldsymbol{X}_{l-1}^i)$.

The final step is to take the ratio of these two estimators $\widehat{Y}_L$ and $\hat{z}_L$:

$$\widehat{Q}_{\mathbf{2}} \equiv \frac{\widehat{Y}_L}{\hat{z}_L} = \frac{\sum_{l=0}^{L} \widehat{\Delta Y_l}}{\sum_{l=0}^{L} \widehat{\Delta \beta_l}}. \tag{3.23}$$

The algorithm for constructing $\widehat{Q}_{\mathbf{2}}$ is listed in algorithm (1). $\widehat{Q}_{\mathbf{2}}$ is a multilevel estimator of $\mathbb{E}_{g_l}\big(Q_l(\boldsymbol{x}_l)\beta_l(\boldsymbol{x}_l)\big)/\mathbb{E}_{g_l}\big(\beta_l(\boldsymbol{x}_l)\big)$ in (3.9). Even though both $\widehat{Y}_L$ and $\hat{z}_L$ are unbiased estimators for the numerator and the denominator, the ratio of these two estimators, which are computed from finite samples, are biased. The expectation of the ratio does not equal the ratio of the expectation of two estimators, assuming finite samples.

Algorithm 1 Multilevel self-normalizing estimator

for $l = 0, 1, ..., L$ **do**
  for $i = 1, ..., N_l$ **do**
    Obtain samples $\boldsymbol{X}_l^i$ and $\boldsymbol{X}_{l-1}^i$ evaluated at level l;
    Calculate the unnormalized weight $\beta(\boldsymbol{X}_l^i)$ and QoI $Q(\boldsymbol{X}_l^i)$;
    Construct the $\widehat{\Delta Y_l}$ in (3.19);
    Construct the $\widehat{\Delta \beta_l}$ in (3.22);
  end for
end for
Sum up all $\widehat{\Delta Y_l}$ to get $\widehat{Y} \equiv \sum_{l=0}^{L} \widehat{\Delta Y_l}$;
Sum up all $\widehat{\Delta \beta_l}$ to get $\widehat{z} \equiv \sum_{l=0}^{L} \widehat{\Delta \beta_l}$;
The ratio of $\widehat{Y}$ and $\widehat{z}$ forms $\widehat{Q}_{\mathbf{2}} \equiv \widehat{Y}_L/\hat{z}_L$.

This multilevel estimator is used in previous papers, such as this paper on Quasi Monte Carlo [48]. In this Quasi Monte Carlo paper, the complexity analysis result does not apply in the Gaussian prior case and prior distribution is used as a biasing distribution. The computational performance or the complexity of this ratio estimator in general is not clear yet, i.e., this estimator may have unbounded variance. In this thesis, we will generalize the complexity analysis to the cases including the Gaussian case. Section 3.4 introduces a new way to analyze its complexity. We will prove that this estimator has similar complexity results to the standard multilevel Monte Carlo. In this thesis, transport mapping introduced in the following section, is also used as a technique to minimize the variance of the multilevel estimator. Even though we will use the densities generated from

RTO [5] or Implicit Sampling [38, 39] as the proposal distribution, the results listed in the current section and section 3.2 does not limit the biasing distribution.

3.2 Transport-mapping-based importance sampling

We want to consider the estimator of $\mathbb{I}_Q$ in the narrative of transport mappings. This section combines the importance sampling from Chapter 2 and introduces transport mappings. We will also explore the density of parameters after the transport mappings. The RTO method can be viewed as a transport mapping from prior to posterior and therefore, this section lays the foundation for the RTO densities described in the next section.

3.2.1 Transport mapping

We first start from some definitions in chapter 2. Suppose we have an infinite-dimension parameter x with the posterior measure as in equation (2.10):

$$\mu_{\text{post}}(dx) = \frac{1}{z}\mathcal{L}(y|x)\mu_{\text{prior}}(dx), \tag{3.24}$$

where $\mathcal{L}(y|x)$ is the likelihood function and z is a constant. In practice, we cannot generate the infinite-dimension parameter and its corresponding posterior; we can only compute the finite-dimension parameter. Therefore, the first mapping needed is from the infinite-dimensional space to the finite-dimensional space $D : \mathscr{H} \to \mathbb{R}^n$,

$$D_h(x) = \boldsymbol{x}_h, \tag{3.25}$$

where h is the discretization size. This mapping serves as a discretization step. Based on the different levels in the multilevel method, h varies across levels. Therefore we need to map x into spaces of $\boldsymbol{x}_{h_l}$ with various dimension n_l. For a $l = 0, 1, 2, ..., L$, we denote $D_{h_l}(x) = \boldsymbol{x}_{h_l}$, and simplify it as

$$D_l(x) = \boldsymbol{x}_l. \tag{3.26}$$

Note that x is originally from the prior measure μ_{prior} and after the Bayes' Law, we have the posterior of x, μ_{post}. In the meantime, $\boldsymbol{x}_l$ is also assumed to have prior density p_l. Hence, D_l maps x from the prior measure, μ_{prior}, to $\boldsymbol{x}_l$ whose prior density is p_l.

After applying Bayes' law, and providing the information from data y, the posterior of

this finite-dimension parameter becomes

$$\pi_l(\boldsymbol{x}_l) = \frac{1}{z_l} L_l(y|\boldsymbol{x}_l) p_l(\boldsymbol{x}_l), \tag{3.27}$$

where L_l and p_l are likelihood function and prior density of $\boldsymbol{x}_l$ respectively. The quantity z_l is the normalizing constant in the discretized model. This formula is the same as equation (2.15), but it is in the setting of the multilevel algorithm. The subscript l denotes the level. Quantities including subscript l are used in the multilevel estimators of section 3.1.

As stated in section 3.1, we want to estimate the following integral under posterior density π_l at a single level l:

$$\mathbb{I}_{Q_l} = \mathbb{E}_{\pi_l}\big(Q_l(\boldsymbol{X}_l)\big) = \int_{\pi_l} Q_l(\boldsymbol{x}_l)\pi_l(\boldsymbol{x}_l)d\boldsymbol{x}_l, \tag{3.28}$$

where Q_l is the QoI mapping from the finite parameter space to the real line, i.e., $Q_l : \mathbb{R}^n \to \mathbb{R}$. If we cannot estimate in the true posterior π_l, then we will find a biasing density g_l such that

$$\begin{aligned}\mathbb{I}_{Q_l} = \mathbb{E}_{\pi_l}\big(Q_l(\boldsymbol{X}_l)\big) &= \int Q_l(\boldsymbol{x}_l)\frac{\pi_l(\boldsymbol{x}_l)}{g_l(\boldsymbol{x}_l)} g_l(\boldsymbol{x}_l)d\boldsymbol{x}_l \\ &= \frac{\mathbb{E}_{g_l}\big(Q_l(\boldsymbol{X}_l)\beta_l(\boldsymbol{X}_l)\big)}{\mathbb{E}_{g_l}\big(\beta_l(\boldsymbol{X}_l)\big)}.\end{aligned}$$

This importance sampling estimation is the same as (3.9), so we abbreviate the intermediate steps and only keep the estimator at the last line. All these estimators of $\mathbb{I}_{Q_l}$ were introduced in Chapter 2. Now we are ready to introduce the transport mapping of $\boldsymbol{x}_l$.

Suppose we can generate posterior samples $\boldsymbol{x}_l$ from some other random variable $\boldsymbol{\xi}_l$. We have $T_l : \mathbb{R}^n \to \mathbb{R}^n$

$$T_l(\boldsymbol{\xi}_l) = \boldsymbol{x}_l, \tag{3.29}$$

for different mappings at different levels l. This mapping T_l illustrates the mapping from a variable $\boldsymbol{\Xi}_l$, which is easy to obtain, like Gaussian random variable, to the demanding posterior parameter $\boldsymbol{X}_l$. The quantity $\boldsymbol{\xi}_l$ is a realization of the random variable $\boldsymbol{\Xi}_l$. The

corresponding density of $\boldsymbol{x}_l$ with respect to $\boldsymbol{\xi}_l$ is

$$\begin{aligned}\int g_l(\boldsymbol{x}_l)d\boldsymbol{x}_l &= \int g_l\big(T_l(\boldsymbol{\xi}_l)\big)\frac{\partial T_l(\boldsymbol{\xi}_l)}{\partial \boldsymbol{\xi}_l}d\boldsymbol{\xi}_l \\ &= \int g_l\big(T_l(\boldsymbol{\xi}_l)\big)\nabla T_l(\boldsymbol{\xi})d\boldsymbol{\xi}_l. \end{aligned} \tag{3.30}$$

Here ∇T_l represents the derivative of T_l with respect to $\boldsymbol{\xi}_l$. The mapping T_l from $\boldsymbol{\xi}_l$ to $\boldsymbol{x}_l$ is very general at this point. The density above is only a straight derivation by the rule of calculus. The mapping can be constructed from an algorithm or a theorem. If we consider the process of Bayes' law, which maps the parameter from prior distribution to the posterior distribution, we can also consider this process T_l since it maps the known prior to the target, posterior distribution. We are applying this idea in the whole thesis where we consider $\boldsymbol{\xi}_l$ to be the prior distribution.

3.2.2 Prior distribution

If we set $\boldsymbol{\Xi}_l$ to have the same prior density of $\boldsymbol{x}_l$, i.e., $\boldsymbol{\Xi}_l \sim p_l(\boldsymbol{\xi}_l)$, then the density function of $\boldsymbol{x}_l$ is

$$\begin{aligned}\int g_l(\boldsymbol{x}_l)d\boldsymbol{x}_l &= \int g_l(T_l(\boldsymbol{\xi}_l))\nabla T_l(\boldsymbol{\xi})d\boldsymbol{\xi}_l \\ &= \int p_l(\boldsymbol{\xi}_l)d\boldsymbol{\xi}_l. \end{aligned} \tag{3.31}$$

This transformation between densities is useful when the prior density p_l is known and easy to sample from, which is usually true in Bayesian problems. In this case, the transport mapping T_l is the mapping from prior distribution to the posterior.

If we are able to integrate $\mathbb{I}_{Q_l}$ with respect to $\boldsymbol{\xi}_l$, then the ratio estimator (3.9) can be expressed in terms of $\boldsymbol{\xi}_l$. We first expand the ratio estimator into an integration:

$$\mathbb{I}_{Q_l} = \frac{\mathbb{E}_{g_l}\big(Q_l(\boldsymbol{X}_l)\beta_l(\boldsymbol{X}_l)\big)}{\mathbb{E}_{g_l}\big(\beta_l(\boldsymbol{X}_l)\big)} \tag{3.32}$$

$$= \frac{\int Q_l(\boldsymbol{x}_l)\beta_l(\boldsymbol{x}_l)g_l(\boldsymbol{x}_l)d\boldsymbol{x}_l}{\int \beta_l(\boldsymbol{x}_l)g_l(\boldsymbol{x}_l)d\boldsymbol{x}_l}. \tag{3.33}$$

If we want to introduce $\boldsymbol{\xi}_l$ to these equations (3.31, 3.32 and 3.33), we need the transport mapping T_l between $\boldsymbol{\xi}_l$ and $\boldsymbol{x}_l$. After the function compositions, the numerator of this ratio

estimator is

$$\int \big(Q_l \circ T\big)(\boldsymbol{\xi}_l)\big(\beta_l \circ T\big)(\boldsymbol{\xi}_l) g_l(T_l(\boldsymbol{\xi}_l))\nabla T_l(\boldsymbol{\xi}_l) d\boldsymbol{\xi}_l, \tag{3.34}$$

$$= \int \big(Q_l \circ T\big)(\boldsymbol{\xi}_l)\big(\beta_l \circ T\big)(\boldsymbol{\xi}_l) p_l(\boldsymbol{\xi}_l) d\boldsymbol{\xi}_l. \tag{3.35}$$

while the denominator becomes

$$\int \big(\beta_l \circ T\big)(\boldsymbol{\xi}_l) g_l(T_l(\boldsymbol{\xi}_l))\nabla T_l(\boldsymbol{\xi}_l) d\boldsymbol{\xi}_l, \tag{3.36}$$

$$= \int \big(\beta_l \circ T\big)(\boldsymbol{\xi}_l) p_l(\boldsymbol{\xi}_l) d\boldsymbol{\xi}_l. \tag{3.37}$$

Note here that both equations use the transformation between densities in equation (3.31). After we transform this integral to the expectation, we have the following ratio estimator with respect to $\boldsymbol{\Xi}_l$:

$$\mathbb{I}_{Q_l} = \frac{\mathbb{E}_{\boldsymbol{\Xi}_l}\big(\big(Q_l \circ T_l\big)(\boldsymbol{\Xi}_l)\big(\beta_l \circ T_l\big)(\boldsymbol{\Xi}_l)\big)}{\mathbb{E}_{\boldsymbol{\Xi}_l}\big(\big(\beta_l \circ T_l\big)(\boldsymbol{\Xi}_l)\big)}. \tag{3.38}$$

If we consider the discretization map D_l, then the ratio estimator becomes

$$\mathbb{I}_{Q_l} = \frac{\int \big(Q_l \circ T_l \circ D_l\big)(x)\big(\beta_l \circ T_l \circ D_l\big)(x) dx}{\int \big(\beta_l \circ T_l \circ D_l\big)(x) dx}, \tag{3.39}$$

for some volume dx in the prior measure. D_l is mapping from the prior measure μ_{prior} to the prior density p_l of $\boldsymbol{x}_l$. In other words, $D_l(x)$ is the prior density p_l. We have assumed that $\boldsymbol{\Xi}_l \sim p_l(\boldsymbol{\xi}_l)$, thus $\boldsymbol{\Xi}_l$ has the same density of $D_l(x)$. Thus, the first composition $T_l \circ D_l$ represents a map from the infinite-dimension parameter x to the discretized finite-dimension parameter $\boldsymbol{x}_l$. The second composition $Q_l \circ T_l$ is again using the transport mapping $T_l(\boldsymbol{\xi}) = \boldsymbol{x}_l$. Note that $\boldsymbol{x}_l$ here is from the posterior instead of prior p_l. At this point, we have introduced and connected all the transport mappings needed in this thesis.

Setting $\boldsymbol{\xi}_l$ to have the same distribution as the prior distribution p_l actually links two different processes: one is the bridging from prior x to posterior $\boldsymbol{x}_l$, the other is the RTO transformation that will be discussed in section 3.3. Notice that both processes usually start from a Gaussian random variable. We unite both processes by setting the same starting point, i.e., the same known random variable, $\boldsymbol{\xi}_l$, usually Gaussian. One of the consequences of the linkage between the two processes will be see in Chapter 6.

3.2.3 Coupling variables

When introducing the multilevel estimator $\widehat{Q}_{\mathbf{2}}$ in equation (3.23), we suggest that $\boldsymbol{X}_l^i$ and $\boldsymbol{X}_{l-1}^i$ should be coupled to reduce the variance of $\widehat{\Delta Y}$ and $\widehat{\Delta\beta}$. The coupling can be realized using the transport mapping T_l introduced in this section. There are other coupling techniques, but we only discuss the coupling using T_l in this thesis.

The main idea is to use T_l as an intermediate step to bond $\boldsymbol{X}_l^i$ and $\boldsymbol{X}_{l-1}^i$. As stated before, if we set $\boldsymbol{\Xi}_l$ to have the Gaussian prior, i.e., $\boldsymbol{\Xi}_l \sim p_l(\boldsymbol{\xi}_l)$, we are mapping the Gaussian variables $\boldsymbol{\Xi}_l$ to the sample $\boldsymbol{X}_l$. If we can make $\boldsymbol{\Xi}_l$ and $\boldsymbol{\Xi}_{l-1}$ linked, then we also link or couple $\boldsymbol{X}_l^i$ and $\boldsymbol{X}_{l-1}^i$.

If the dimension of $\boldsymbol{x}_l$ remains the same for every level l, then we can actually map the same $\boldsymbol{\xi}$ to both $\boldsymbol{x}_l$ and $\boldsymbol{x}_{l-1}$. If the dimension of $\boldsymbol{x}_l$ is different across level l, we can also make $\boldsymbol{X}_l^i$ and $\boldsymbol{X}_{l-1}^i$ coupled. In this case, $\dim(\boldsymbol{x}_{l-1}) < \dim(\boldsymbol{x}_l)$, the corresponding $\boldsymbol{\xi}_{l-1}$ also has a smaller dimension than $\boldsymbol{\xi}_l$. Thus some parts of $\boldsymbol{\xi}_{l-1}$ and $\boldsymbol{\xi}_l$ can have the same values. The other parts are also linked. More about this coupling method of changing dimension across levels will be addressed in Chapter 6.

This subsection mainly discusses the case when $\dim(\boldsymbol{x}_l)$ remains the same for different levels. The variable $\boldsymbol{\xi}$ with the subscript is denoted as $\boldsymbol{\xi}_l$ but it is actually mapped to two different parameter vectors $\boldsymbol{X}_l$ and $\boldsymbol{X}_{l-1}$. We notice that in this scenario $\boldsymbol{x}_l$ and $\boldsymbol{x}_{l-1}$ are close to each other since they are mapping from the same reference variable $\boldsymbol{\xi}_l$. We take the estimation of $Q(\cdot)\beta(\cdot)$ as an example. After replacing sample $\boldsymbol{X}_l$ with $T_l(\boldsymbol{\Xi}_l)$, we have

$$\begin{aligned}\mathbb{E}_{g_l}\big(Q_l(\boldsymbol{X}_l)\beta_l(\boldsymbol{X}_l)\big) &= \mathbb{E}_{p_l}\big(Q_l \circ T_l(\boldsymbol{\Xi}_l)\beta_l \circ T_l(\boldsymbol{\Xi}_l)\big)\\ \mathbb{E}_{g_{l-1}}\big(Q_{l-1}(\boldsymbol{X}_{l-1})\beta_{l-1}(\boldsymbol{X}_{l-1})\big) &= \mathbb{E}_{p_l}\big(Q_{l-1} \circ T_{l-1}(\boldsymbol{\Xi}_l)\beta_{l-1} \circ T_{l-1}(\boldsymbol{\Xi}_l)\big).\end{aligned}$$

We can see that the equations on the right-hand side are taking expectation over the same distribution p_l since $\boldsymbol{\Xi}_l \sim p_l(\boldsymbol{\xi}_l)$. We only take one random sample $\boldsymbol{\Xi}_L$ in this scenario. Therefore in the telescope sum, any difference between two consecutive levels can be united as

$$\begin{aligned}&\mathbb{E}_{g_l}\Big(Q_l(\boldsymbol{X}_l)\beta_l(\boldsymbol{X}_l)\Big) - \mathbb{E}_{g_{l-1}}\Big(Q_{l-1}(\boldsymbol{X}_{l-1})\beta_{l-1}(\boldsymbol{X}_{l-1})\Big)\\ =&\mathbb{E}_{p_l}\Big(Q_l \circ T_l(\boldsymbol{\Xi}_l)\beta_l \circ T_l(\boldsymbol{\Xi}_l) - Q_{l-1} \circ T_{l-1}(\boldsymbol{\Xi}_l)\beta_{l-1} \circ T_{l-1}(\boldsymbol{\Xi}_l)\Big).\end{aligned} \tag{3.40}$$

The telescope sum is re-interpreted as

$$\begin{aligned}\mathbb{E}_{g_L}\big(Q_l(\boldsymbol{X}_L)\beta_l(\boldsymbol{X}_L)\big) =&\mathbb{E}_{p_0}\Big(Q_0\circ T_0(\boldsymbol{\Xi}_0)\beta_0\circ T_0(\boldsymbol{\Xi}_0)\Big)+ \\ &\sum_{l=1}^{L}\mathbb{E}_{p_l}\Big(Q_l\circ T_l(\boldsymbol{\Xi}_l)\beta_l\circ T_l(\boldsymbol{\Xi}_l)-Q_{l-1}\circ T_{l-1}(\boldsymbol{\Xi}_l)\beta_{l-1}\circ T_{l-1}(\boldsymbol{\Xi}_l)\Big).\end{aligned} \tag{3.41}$$

In the multilevel theorem in (2.3.1), there are two convergence rates that decide the efficiency of the multilevel method. One is the variance of each difference in the telescope sum. In this case, it is $\mathbb{V}\big(Q_l(\cdot)\beta_l(\cdot)-Q_{l-1}(\cdot)\beta_{l-1}(\cdot)\big) = O(\frac{h_l^\alpha}{N_l})$. The other rate is the computational cost of this difference, i.e., $C_l = O(N_l h_l^{-\eta})$. We can see that the complexity of the multilevel estimator is categorized in three categories: $\alpha > \eta$, $\alpha = \eta$ and $\alpha < \eta$, where α is the rate of variance and η is the rate of computational cost. We want the case that the variance is decreasing faster than the computational cost when the level increases. If the cost of the difference in the telescope sum increases faster than its variance, then adding this difference does not add any efficiency to the estimation of $Q(\cdot)\beta(\cdot)$. Therefore, the multilevel method is not a good candidate here. We break down the variance as follows:

$$\begin{aligned}&\mathbb{V}\big(Q_l(\cdot)\beta_l(\cdot)-Q_{l-1}(\cdot)\beta_{l-1}(\cdot)\big)\\ =&\mathbb{V}\big(Q_l(\cdot)\beta_l(\cdot)\big)+\mathbb{V}\big(Q_{l-1}(\cdot)\beta_{l-1}(\cdot)\big)-2\mathrm{Cov}\big(Q_l(\cdot)\beta_l(\cdot),Q_{l-1}(\cdot)\beta_{l-1}(\cdot)\big).\end{aligned} \tag{3.42}$$

Consequently, we need to maximize the covariance $\mathrm{Cov}\big(Q_l(\cdot)\beta_l(\cdot)),Q_{l-1}(\cdot)\beta_{l-1}(\cdot))\big)$, so that $\mathbb{V}\big(Q_l(\cdot)\beta_l(\cdot)-Q_{l-1}(\cdot)\beta_{l-1}(\cdot)\big)$ is small. If we use uncorrelated $\boldsymbol{X}_l$ and $\boldsymbol{X}_{l-1}$ in this estimation, $\mathrm{Cov}\big(Q_l(\boldsymbol{X}_l)\beta_l(\boldsymbol{X}_l),Q_{l-1}(\boldsymbol{X}_{l-1})\beta_{l-1}(\boldsymbol{X}_{l-1}))\big)$ will diminish. However, if we coupled them under one reference variable $\boldsymbol{\Xi}$ as in (3.40), the covariance, $\mathrm{Cov}\big(Q_l\circ T_l(\boldsymbol{\Xi}_l)\beta_l\circ T_l(\boldsymbol{\Xi}_l),Q_{l-1}\circ T_{l-1}(\boldsymbol{\Xi}_l)\beta_{l-1}\circ T_{l-1}(\boldsymbol{\Xi}_l)\big)$, is large since these two terms are strongly correlated.

We also list the corresponding sample-based estimators here. We use the same multilevel estimator $\widehat{Q}_{\mathbf{2}} = \widehat{Y}_L/\hat{z}_L$ introduced in section 3.1. We define $\widehat{Y}_L$ as a multilevel estimator of $\mathbb{E}_{g_L}\big(Q_L(\boldsymbol{X}_L)\beta_L(\boldsymbol{X}_L)\big)$ where

$$\widehat{Y}_L \equiv \sum_{l=0}^{L}\widehat{\Delta Y_l}.$$

$$\widehat{\Delta Y}_l = \begin{cases} \frac{1}{N_0}\sum_{i=1}^{N_0} \hat{\beta}_l\big(T_l(\boldsymbol{\Xi}_l^i)\big)\widehat{Q}_l\big(T_l(\boldsymbol{\Xi}_l^i)\big), & l=0; \\ \frac{1}{N_l}\sum_{i=1}^{N_l} \hat{\beta}_l\big(T_l(\boldsymbol{\Xi}_l^i)\big)\widehat{Q}_l\big(T_l(\boldsymbol{\Xi}_l^i)\big) - \hat{\beta}_{l-1}\big(T_{l-1}(\boldsymbol{\Xi}_l^i)\big)\widehat{Q}_{l-1}\big(T_{l-1}(\boldsymbol{\Xi}_l^i)\big). & l=1,2,...,L. \end{cases} \tag{3.43}$$

Similarly, $\hat{z}_L$ is the multilevel estimator of $\mathbb{E}_{g_l}\big(\beta_l(\boldsymbol{X}_l)\big)$ where

$$\hat{z}_L \equiv \sum_{l=0}^{L} \widehat{\Delta\beta}_l.$$

Each $\widehat{\Delta\beta}_l$ is also represented in terms of $\boldsymbol{\Xi}_l$ here:

$$\widehat{\Delta\beta}_l = \begin{cases} \frac{1}{N_0}\sum_{i=1}^{N_0} \hat{\beta}_0\big(T_0(\boldsymbol{\Xi}_0^i)\big), & l=0; \\ \frac{1}{N_l}\sum_{i=1}^{N_l} \hat{\beta}_l\big(T_l(\boldsymbol{\Xi}_l^i)\big) - \hat{\beta}_{l-1}\big(T_{l-1}(\boldsymbol{\Xi}_{l-1}^i)\big), & l=1,2,...,L. \end{cases} \tag{3.44}$$

The algorithm is the same as that in algorithm (1), except adding one more step before obtaining samples $\boldsymbol{X}_l^i$ and $\boldsymbol{X}_{l-1}^i$, that is, to sample $\boldsymbol{\Xi}_l^i$ from prior p_l and then use the transport mapping T_l and T_{l-1} to obtain sample $\boldsymbol{X}_l^i$ and $\boldsymbol{X}_{l-1}^i$.

3.3 RTO method

This section introduces an optimization-based sampling method we use in different models and problems. We first introduce the target distribution that is optimized in the method. The following subsection is the detailed algorithm and corresponding densities. The correlation of the RTO method and the transport mapping is also depicted. This section finishes up with the further research results in the RTO application in high dimensions.

3.3.1 Target distribution

We construct the target distribution from retrospecting the posterior density. Starting from a Bayesian inverse problem, we have the discretized forward model:

$$y = F_l(\boldsymbol{x}_l) + e, \tag{3.45}$$

where $y \in \mathbb{R}^m$, $\boldsymbol{x}_l \in \mathbb{R}^{n_l}$, and e is assumed herein to have a Gaussian distribution with zero mean and covariance Γ_{obs}; y is the data, and $F_l : \mathbb{R}^n \to \mathbb{R}^m$ is the discretized forward model with discretization size h_l. We also assume that the prior distribution of $\boldsymbol{x}_l$ is a zero mean Gaussian with covariance Γ_{prior}. With all these assumptions, the posterior takes the form

$$\begin{aligned}
\pi_l(\boldsymbol{x}_l|y) =& \frac{1}{z_l} L_l(\boldsymbol{x}_l|y) p(\boldsymbol{x}_l) \\
=& \frac{1}{z_l}(2\pi)^{-\frac{m}{2}} |\det(\Gamma_{\text{obs}})|^{-\frac{1}{2}} \exp\Big(-\frac{1}{2}\big(y - F_l(\boldsymbol{x}_l)\big)^T \Gamma_{\text{obs}}^{-1}\big(y - F_l(\boldsymbol{x}_l)\big)\Big) p_l(\boldsymbol{x}_l) \\
=& \frac{1}{z_l}(2\pi)^{-\frac{m}{2}} |\det(\Gamma_{\text{obs}})|^{-\frac{1}{2}} \exp\Big(-\frac{1}{2}\big(y - F_l(\boldsymbol{x}_l)\big)^T \Gamma_{\text{obs}}^{-1}\big(y - F_l(\boldsymbol{x}_l)\big)\Big) \\
& (2\pi)^{-\frac{n}{2}} |\det(\Gamma_{\text{prior}})|^{-\frac{1}{2}} \exp\big(-\frac{1}{2}\boldsymbol{x}_l^T \Gamma_{\text{prior}}^{-1} \boldsymbol{x}_l\big),
\end{aligned}$$

where z_l is again the normalizing constant. We can see from the posterior above that only the exponential part has the variable that we can optimize. Therefore we make some change to the exponential part so that it is easy to optimize. We suppress the posterior as:

$$\pi_l(\boldsymbol{x}_l|y) = \frac{1}{z}(2\pi)^{-\frac{m+n}{2}} |\det(\Gamma_{\text{obs}})|^{-\frac{1}{2}} |\det(\Gamma_{\text{prior}})|^{-\frac{1}{2}} \exp\Big(-\frac{1}{2} H_l(\boldsymbol{x}_l)^T H_l(\boldsymbol{x}_l)\Big), \tag{3.46}$$

where $H : \mathbb{R}^{n_l} \to \mathbb{R}^{m+n_l}$ is parameterized as

$$H_l(\boldsymbol{x}_l) = \begin{pmatrix} \Gamma_{\text{prior}}^{-\frac{1}{2}}(\boldsymbol{x}_l) \\ \Gamma_{\text{obs}}^{-\frac{1}{2}}[F_l(\boldsymbol{x}_l) - y] \end{pmatrix}. \tag{3.47}$$

The aim is to make the posterior proportional to a simply formed exponential function:

$$\pi_l(\boldsymbol{x}_l|y) \propto \exp(-\frac{1}{2}\|H_l(\boldsymbol{x}_l)\|^2).$$

The notation $||\cdot||$ means 2-norm. $H_l(\boldsymbol{x}_l)$ is the target distribution used in the RTO algorithm. In Chapter 5, the implicit sampling chapter, we also have the target distribution of implicit sampling method, which is very similar to this form. This kind of reinterpretation does not change the posterior but only adds convenience to the following optimization process.

3.3.2 RTO algorithm

Linear forward model

To understand the motivation of this relatively new algorithm, we first introduce the RTO method in the linear case before its general application in the nonlinear case. Following the same assumptions for parameter $\boldsymbol{x}$ of the inverse problem in Chapter 2, we assume the noise e is standard Gaussian here to simplify the formula. For this linear case only, the notations of all corresponding quantities do not have the level l subscript since we do not apply the multilevel method in this case. Hence,

$$\pi(\boldsymbol{x}|y) \propto \exp\bigg(-\frac{1}{2}\|F(\boldsymbol{x}) - y\|^2 \bigg), \tag{3.48}$$

where F is linear with respect to $\boldsymbol{x}$, i.e., $F(\boldsymbol{x}) = J\boldsymbol{x}$, $J \in \mathbb{R}^{m\times n}$. We need to point out that this linear forward model is only for intuition. In practice, we are interested in the nonlinear inverse problem. The posterior of $\boldsymbol{x}$ is

$$\pi_{\text{post}}(\boldsymbol{x}|y) \propto (J^T J)^{\frac{1}{2}} \exp\bigg(-\frac{1}{2}\|J\boldsymbol{x} - y\|^2 \bigg). \tag{3.49}$$

Since J is a known matrix, we are interested in minimizing the equation inside the 2-norm. We obtain the MAP point at the point where $||J\boldsymbol{x} - y||^2$ is minimized. In this setting, $J\boldsymbol{x} - y$ is the target function. Adding a random perturbation to this equation can produce new samples. Optimizing this stochastic equation,

$$\arg\min_{\boldsymbol{x}} \|J\boldsymbol{x} - y - \boldsymbol{\xi}\|^2, \tag{3.50}$$

where $\boldsymbol{\xi}$ is a standard Gaussian vector, we obtain the samples from $\pi_{\text{post}}(\boldsymbol{x}|y)$. This optimization is solving the following equation with respect to $\boldsymbol{x}$:

$$J\boldsymbol{x} = y + \boldsymbol{\xi}. \tag{3.51}$$

We can compute the thin QR factorization of J, that is, decompose J into $UR = J$ so that where $U \in \mathbb{R}^{m\times n}$ is a matrix with orthogonal columns and $R \in \mathbb{R}^{n\times n}$ is an upper triangular matrix. We multiply U^T on both sides of (3.51):

$$U^T J\boldsymbol{x} = U^T(y + \boldsymbol{\xi}). \tag{3.52}$$

$U^T J$ is upper triangular since $U^T J = R$. Therefore the linear system (3.52) can be solved using backward substitution, a computational technique of solving the linear equation $U'\boldsymbol{x} = y'$ where U' is an upper triangular matrix. This technique increases the computation efficiency. Hence, the optimization equation becomes

$$\arg\min_{\boldsymbol{x}} \|U^T J\boldsymbol{x} - U^T(y + \boldsymbol{\xi})\|^2. \tag{3.53}$$

This optimization is merely solving the linear system in equation (3.52). It is listed in the form of optimization, and we can compare it with its counterpart in the nonlinear case later. The main idea of this method is to replace solving the linear system in (3.51) with solving the other equation in (3.52) that is easier to compute. The price for this method is to do one QR factorization before solving the linear system, which is worthwhile in high dimension problems.

Nonlinear forward model

We carry this transformation in (3.52) to the nonlinear case. The target distribution is $H_l(\boldsymbol{x}_l)$ in subsection 3.3.1. First we need to find the suitable $U_l \in \mathbb{R}^{(m+n_l)\times n_l}$ in the nonlinear case. One choice is to find the gradient or Jacobian $J_l(\boldsymbol{x}_l) \in \mathbb{R}^{(m+n_l)\times n_l}$ of the discrete forward mapping $H_l(\boldsymbol{x}_l)$ at the maximum a posterior (MAP) point, i.e.,

$$J_l(\boldsymbol{x}_l^{MAP}) = \left.\frac{\partial H_l(\boldsymbol{x}_l)}{\partial \boldsymbol{x}_l}\right|_{\boldsymbol{x}_l^{MAP}}. \tag{3.54}$$

Following the procedures in the linear case, the next step is to do the thin QR factorization of $J(\boldsymbol{x}_l^{MAP})$:

$$U_l R_l = J_l(\boldsymbol{x}_l^{MAP}). \tag{3.55}$$

We add the disturbance to $U_l^T H_l(\boldsymbol{x}_l)$, similar to the step in equation (3.51). The optimization equation now becomes

$$\arg\min_{\boldsymbol{x}_l} \|U_l^T H_l(\boldsymbol{x}_l) - \boldsymbol{\xi}_l\|^2. \tag{3.56}$$

This $\boldsymbol{\xi}_l$ is a standard Gaussian random vector as the linear case. Adding $\boldsymbol{\xi}_l$ is called the randomization step. A new sample $\boldsymbol{x}_l$ comes from solving this equation, which is called the optimization step. Every new sample vector $\boldsymbol{x}_l$ is obtained from adding different $\boldsymbol{\xi}_l$ each time. These two steps give the Randomize-and-then-Optimize (RTO) method. The cost for this method is again the QR factorization before optimization, plus the cost of optimization iterations whose number is not deterministic. If $H_l(\cdot)$ is a linear model, we can figure out

the actual cost of each optimization iteration. However $H_l(\cdot)$ is a nonlinear function, thus we cannot determine the optimization iterations as we can do in the linear model.

The optimization equation (3.56) is basically finding $\boldsymbol{x}_l$ that satisfies

$$U_l^T H_l(\boldsymbol{x}_l) = \boldsymbol{\xi}_l. \tag{3.57}$$

It looks a bit different than its analogue in the linear case (3.51) since we embed the data y in $H(\boldsymbol{x}_l)$, and part of the right-hand side $U_l^T \boldsymbol{\xi}_l$ is also a standard Gaussian like $\boldsymbol{\xi}_l$. The probability density function of sample $\boldsymbol{x}_l$ from optimizing (3.56) becomes

$$g_{\text{RTO}}(\boldsymbol{x}_l) = |\det(U^T J_l(\boldsymbol{x}_l))| p_l(\boldsymbol{\xi}_l), \tag{3.58}$$

where $p_l(\boldsymbol{\xi}_l)$ is the pdf of a standard Gaussian random vector. It turns out that

$$\begin{aligned} g_{\text{RTO}}(\boldsymbol{x}_l) &= |\det(U_l^T J_l(\boldsymbol{x}_l))| (2\pi)^{-\frac{n_l}{2}} \exp(-\frac{1}{2} \boldsymbol{\xi}_l^T \boldsymbol{\xi}_l) \\ &= |\det(U_l^T J(\boldsymbol{x}_l))| (2\pi)^{-\frac{n_l}{2}} \exp(-\frac{1}{2} \| U_l^T H_l(\boldsymbol{x}_l) \|^2). \end{aligned} \tag{3.59}$$

The derivation above follows the rule of calculus and transform rule of densities. The Euclidean norm in (3.59) is derived from (3.57). The pdf of the RTO sample $\boldsymbol{x}_l$ is not Gaussian. It is strongly correlated to Gaussian as shown in the pdf. It is a known pdf, which is a good candidate for a biasing distribution in the importance sampling technique. If we adapt importance sampling in the RTO method, the unnormalized weight is

$$\beta_l(\boldsymbol{x}_l) = |\det(U_l^T J_l(\boldsymbol{x}_l))|^{-1} \exp(-\frac{1}{2} \| H_l(\boldsymbol{x}_l) \|^2 + \frac{1}{2} \| U_l^T H_l(\boldsymbol{x}_l) \|^2). \tag{3.60}$$

The normalized weight is

$$\hat{w}_l(\boldsymbol{x}_l^i) = \frac{\beta_l(\boldsymbol{x}_l^i)}{\sum_{i=1}^{N} \beta_l(\boldsymbol{x}_l^i)}. \tag{3.61}$$

The QoI evaluated using the normalized weight is

$$\int Q_l(\boldsymbol{x}_l) g_{\text{RTO}}(\boldsymbol{x}_l) d\boldsymbol{x}_l \approx \sum_{i=1}^{N_l} Q_l(\boldsymbol{X}_l^i) \hat{w}_l(\boldsymbol{X}_l^i) = \frac{\sum_{i=1}^{N_l} Q_l(\boldsymbol{X}_l^i) \beta_l(\boldsymbol{X}_l^i)}{\sum_{i=1}^{N_l} \beta_l(\boldsymbol{X}_l^i)}. \tag{3.62}$$

To generate one sample $\boldsymbol{x}_l$ from this RTO algorithm, one first needs to run and solve the optimization equation several times. Each iteration of optimization needs to multiply $U_l^T \in \mathbb{R}^{n_l \times (m+n_l)}$, which costs $O\big((m+n_l) \times n_l\big)$ floating point operations. After the optimization, we need to calculate the weight $\beta_l(\boldsymbol{x}_l)$. $U_l^T J_l(\boldsymbol{x}_l)$ requires $m+n_l$ matrix-vector multiplications and its determinant needs $O(n_l^3)$ floating point operations. Overall, the computational cost of RTO is not cheap.

The following assumptions ensure equation (3.57) has a unique solution for each different $\boldsymbol{\xi}_l$ and the pdf of RTO density (3.59) exists for any sample.

Assumption 3.3.1. (Assumptions for RTO method)

- The target function H_l is continuously differentiable with Jacobian $\frac{\partial H_l(\boldsymbol{x}_l)}{\partial \boldsymbol{x}_l}$.
- The Jacobian $J_l(\boldsymbol{x}_l)$ is full column for every $\boldsymbol{x}_l$.
- The mapping $\boldsymbol{x}_l \to U_l^T H_l(\boldsymbol{x}_l)$ is invertible.

The first and second assumption guarantee that we can do QR decomposition at any $\boldsymbol{x}_l$ of Jacobian $J_l(\boldsymbol{x}_l)$. The third assumption ensures that we can project the reference variable $\boldsymbol{\xi}$ to the target sample $\boldsymbol{x}_l$.

3.3.3 Transport mapping

We frame the RTO method in terms of transport mapping as in section 3.2. Based on the RTO equilibrium (3.57), we acquire sample $\boldsymbol{x}_l$ from a Gaussian random vector $\boldsymbol{\xi}_l$. We construct a mapping $T_l : \mathbb{R}^{n_l} \to \mathbb{R}^{n_l}$, i.e.,

$$T_l(\boldsymbol{\xi}_l) = \boldsymbol{x}_l. \tag{3.63}$$

This mapping is the transport mapping we mention in equation (3.29). We have assumed Gaussian prior in this method. We can also interpret T_l as the transformation from n_l-dimension Gaussian random samples, or the prior, to n_l-dimension RTO samples. Applying the same pdf of the transporting mapping in equation (3.30) of section 3.2 and the RTO

pdf in equation (3.59), we have:

$$\begin{aligned}\int g_{\text{RTO}}(\boldsymbol{x}_l)d\boldsymbol{x}_l &= \int g_{\text{RTO}}\big(T_l(\boldsymbol{\xi}_l)\big)\nabla T_l d\boldsymbol{\xi}_l \\ &= \int |\det\left(U_l{}^T J_l(\boldsymbol{x}_l)\right)|(2\pi)^{-\frac{n_l}{2}}\exp(-\frac{1}{2}\boldsymbol{\xi}_l^T\boldsymbol{\xi}_l)\nabla T_l d\boldsymbol{\xi}_l \\ &= \int |\det(U_l{}^T J_l(\boldsymbol{x}_l))|(2\pi)^{-\frac{n_l}{2}}\exp(-\frac{1}{2}\boldsymbol{\xi}_l^T\boldsymbol{\xi}_l)\det(\frac{\partial(H_l{}^{-1}(U_l\boldsymbol{\xi}_l))}{\partial\boldsymbol{\xi}_l})d\boldsymbol{\xi}_l \\ &= \int (2\pi)^{-\frac{n_l}{2}}\exp(-\frac{1}{2}\boldsymbol{\xi}_l^T\boldsymbol{\xi}_l)d\boldsymbol{\xi}_l = 1.\end{aligned}$$

This verifies that the density function of RTO is a well-defined probability density function.

There are other algorithms using the optimization to determine a new sample like RTO, for example, Randomize Maximum likelihood [42] and the random-map Implicit Sampling [38, 39], which is discussed in Chapter 5. The common properties of these three algorithms is that they push a reference distribution towards the demanded target densities. In other words, they share the same transport mapping (3.63). The coupling strategy in section 3.2.3 can be applied in this RTO method. The discussion of coupling RTO method will be described in the multilevel RTO section (3.3.5).

3.3.4 RTO in high-dimensional problems

We notice that in the optimization equation (3.56), the product of U_l^T and $H_l(\cdot)$ requires $O(n_l^2)$ floating point operations. When calculating the weight (3.60), we need to compute the determinant that needs $O(n_l^3)$ floating point operations. When n_l is very large, the computation increases cubically, which is not manageable. We need an alternative calculation to reduce the cost. The following proposition and results are from [3]. We first reform the parameter and forward model as:

$$\boldsymbol{v}_l = \Gamma_{\text{prior}}^{-\frac{1}{2}}(\boldsymbol{x}_l),\ G_l(\boldsymbol{v}_l) = \Gamma_{\text{obs}}^{-\frac{1}{2}}[F_l(\Gamma_{\text{prior}}^{\frac{1}{2}}(\boldsymbol{v}_l)) - y]. \tag{3.64}$$

We define

$$\tilde{H}_l(\boldsymbol{v}_l) = \begin{pmatrix}\boldsymbol{v}_l \\ G_l(\boldsymbol{v}_l)\end{pmatrix}. \tag{3.65}$$

This is the notation used in the paper [3], and is called the scalable implementation of RTO. It is easier to follow the idea with same notation. This expression is essentially the same as in previous sections after the transformation. The matrix size of $\tilde{H}_l(\boldsymbol{v}_l)$ is identical

to $H_l(\boldsymbol{x}_l)$. In fact, $\tilde{H}_l(\boldsymbol{v}_l)$ is just a re-parameterization of $H_l(\boldsymbol{x}_l)$. We still use the same optimization equation (3.56). Now we need to do the QR factorization of $J_l(\boldsymbol{v}_l^{MAP})$ where $J_l(\boldsymbol{v}_l) \equiv \frac{\partial \tilde{H}(\boldsymbol{v}_l)}{\partial \boldsymbol{v}_l}$ at MAP point $\boldsymbol{v}_l^{MAP}$. We keep in mind that $\boldsymbol{v}_l^{MAP} = \Gamma_{\text{prior}}^{-\frac{1}{2}}(\boldsymbol{x}_l^{MAP})$.

In order to reduce the computational cost of the optimization, we need to change the optimization equation from the QR factorization. Instead of computing the QR factorization of $J_l(\boldsymbol{v}_l^{MAP})$, we do the polar decomposition:

$$J_l(\boldsymbol{v}_l^{MAP}) = \tilde{U}_l\big(J_l(\boldsymbol{v}_l^{MAP})^T J_l(\boldsymbol{v}_l^{MAP})\big)^{\frac{1}{2}}, \tag{3.66}$$

where $\tilde{U}_l \in \mathbb{R}^{(m+n_l)\times n_l}$ has orthogonal columns and $J_l(\boldsymbol{v}_l^{MAP})^T J_l(\boldsymbol{v}_l^{MAP})$ is positive definite, and therefore, it is a proper polar decomposition. We construct $\tilde{U}_l$ as

$$\tilde{U}_l = J_l(\boldsymbol{v}_l^{MAP})\big(J_l(\boldsymbol{v}_l^{MAP})^T J_l(\boldsymbol{v}_l^{MAP})\big)^{-\frac{1}{2}}. \tag{3.67}$$

This reinterpretation of U_l is essential to the following results. We introduce a proposition from the paper [3] without proof. The main purpose is to explain how the alternative optimization equation, determinant equation and weight equation reduce the computation, so we aim to avoid technical details. All the I symbol with subscripts denotes the identity matrices.

Proposition 3.3.1. Let $\nabla G_l(\boldsymbol{v}_l^{MAP}))$ denote the derivative of $G_l(\boldsymbol{v})$ at point $\boldsymbol{v}_l^{MAP}$. We consider its reduced singular value decomposition (SVD),

$$\nabla G_l(\boldsymbol{v}_l^{MAP}) = \Psi_l \Lambda_l \Phi_l^T. \tag{3.68}$$

The nonlinear system $\tilde{U}_l \tilde{H}_l(\boldsymbol{v}_l) = \boldsymbol{\xi}_l$ defining the RTO mappling can be rewritten as

$$\begin{cases} (I_{n_l} - \Phi_l \Phi_l^T)\boldsymbol{\xi}_l = (I_{n_l} - \Phi_l \Phi_l^T)\boldsymbol{v}_l \\ \Phi_l \Phi_l^T \boldsymbol{\xi}_l = \Phi_l[(\Lambda_l^2 + I_{r_l})^{-\frac{1}{2}}\big(\Phi_l^T \boldsymbol{v}_l + \Lambda_l \Psi_l^T G_l(\boldsymbol{v}_l)\big)] \end{cases}. \tag{3.69}$$

The weighting function in (3.60) can be expressed as

$$\beta_l(\boldsymbol{v}_l) = |\det(\tilde{U_l}^T J_l(\boldsymbol{v}_l))|^{-1} \exp(-\frac{1}{2}||G_l(\boldsymbol{v}_l)||^2 - \frac{1}{2}||\Phi_l^T \boldsymbol{v}_l||^2 + \frac{1}{2}||\Phi_l^T \boldsymbol{v}_l + \Lambda_l \Psi_l^T G_l(\boldsymbol{v}_l)||^2), \tag{3.70}$$

where the determinant takes the simplified form

$$|\det(\tilde{U_l}^T J_l(\boldsymbol{v}_l))| = |\det(\Lambda_l^2 + I_{r_l})^{-\frac{1}{2}}||\det(I_{r_l} + \Lambda_l \Psi_l^T \nabla G_l(\boldsymbol{v}_l))\Phi|. \tag{3.71}$$

This result provides the alternative optimization equation (3.69), a new weight (3.70) and a new determinant (3.71) on low-dimension matrices. However, after this formation, instead of optimizing the equation (3.56) with $O((m + n_l) \times n_l)$ floating point operations, we solve the r-dimension equation of $\boldsymbol{v}_r$ from equation (3.69):

$$\arg\min_{\boldsymbol{v}_{r_l}} \|(\Lambda^2 + I_{r_l})^{-\frac{1}{2}}(\boldsymbol{v}_r + \Lambda_l \Psi_l^T G_l(\boldsymbol{v}_{l_\perp} + \Phi_l \boldsymbol{v}_{r_l}) - \Phi_l^T \boldsymbol{\xi}_l)\|^2, \tag{3.72}$$

where $\boldsymbol{v}_{r_l} \equiv \Phi_l^T \boldsymbol{v}_l$, and $\boldsymbol{v}_{l_\perp} \equiv \boldsymbol{v}_l - \Phi \boldsymbol{v}_{r_l}$ is an element in the orthogonal complement of range(Φ_l). We can first solve $\boldsymbol{v}_{l_\perp}$ by this linear multiplication:

$$\boldsymbol{v}_{l_\perp} = (I_{n_l} - \Phi_l \Phi_l^T)\boldsymbol{\xi}_l. \tag{3.73}$$

When optimizing the r_l-dimension equation above, each iteration is expected to evaluate the derivative of the vector-value function inside the norm. Hence, we need to evaluate the following linearization as a derivative of the optimization equation:

$$(\Lambda^2 + I_{r_l})^{-\frac{1}{2}}(I_r + \Lambda_l \Psi_l^T \nabla G_l(\boldsymbol{v}_{l_\perp} + \Phi \boldsymbol{v}_{r_l})\Phi_l). \tag{3.74}$$

After computing the linear multiplication (3.73) and the derivative (3.74) of the optimization equation (3.72), the demanding RTO sample $\boldsymbol{v}_l$ equals

$$\boldsymbol{v}_l = \Phi \boldsymbol{v}_{r_l} + \boldsymbol{v}_{l_\perp}. \tag{3.75}$$

The algorithm of getting $\boldsymbol{v}_l$ is shown in algorithm (2).

We need to notice that r_l is much smaller than $\min(m, n_l)$ because r_l is the dimension of the diagonal matrix from the reduced SVD. Therefore, we decompose a big nonlinear problem with complexity $O((m + n_l) \times n_l)$ into two components: one is an r_l-dimensional nonlinear system problem (3.72) and the other is a linear problem (3.73). The matrix-vector multiplication $U_l^T J_l(\boldsymbol{x}_l)$ is reduced. The determinant of $U_l^T J(\boldsymbol{x}_l)$ has been reduced to computing an $r_l \times r_l$ matrix as shown in the proposition.

Counting the floating point operations needed to evaluate the optimization function (3.72) and the linear multiplication of the derivative (3.74), the number of floating point

Algorithm 2 Scalable implementation of RTO samples

1: Compute the MAP point $\boldsymbol{v}_l^{MAP}$;
2: Form the Jacobian matrix of $\nabla G_l(\boldsymbol{v}_l^{MAP})$);
3: Construct the SVD of $\nabla G_l(\boldsymbol{v}_l^{MAP}))$ in (3.68);
4: **for** i=1,...,N_l **do**
5: Sample $\boldsymbol{\xi}_l$ from a standard Gaussian;
6: Solve $\boldsymbol{v}_{l_\perp}$ from (3.73);
7: Solve $\boldsymbol{v}_{r_l}$ from (3.72) and then obtain $\boldsymbol{v}_l = \Phi \boldsymbol{v}_{r_l} + \boldsymbol{v}_{l_\perp}$;
8: Obtain the weight (3.70) using the new determinant (3.71);
9: **end for**

operations needed for each optimization iteration is

$$O\big((k_{\text{obj}} + k_{\text{adj}})(mr + n_l r)\big) + k_{\text{obj}} C_1(n_l) + k_{\text{adj}} C_2(n_l), \tag{3.76}$$

where k_{obj} is the average number of optimization iterations; k_{adj} is the adjoint matrix-vector multiplication in optimization for each iteration; $C_1(n_l)$ is the cost of evaluating $G(\boldsymbol{v})$, and $C_2(n_l)$ is the matrix-vector multiplication of $\nabla G(\boldsymbol{v})$ and its ajoint vectors. The number of floating point operations needed to evaluate the determinant (3.71) is $O(mr^2+r^3)+rC_2(n_l)$. Thus, a total of

$$O\big(k_{\text{opt}}(k_{\text{obj}} + k_{\text{adj}})(mr + n_l r) + mr^2 + r^3\big) + k_{\text{opt}} k_{\text{obj}} C_1(n_l) + (k_{\text{opt}} k_{\text{adj}} + r) C_2(n_l) \tag{3.77}$$

floating point operations are needed to compute one RTO sample, where the big-O term above refers to the total linear algebra cost, and the other terms refer to the total cost of evaluating $G(\boldsymbol{v}_l)$ and the matrix-vector multiplications of $\nabla G(\boldsymbol{v}_l)$.

We can see that the complexity of this scalable implementation is much smaller than the quadratic $O(n_l^2)$ and even cubic $O(n_l^3)$ complexity of the original version of RTO method. More details on the complexity of this reduced dimension optimization problem are discussed in the paper [3]. There is another technique to further reduce complexity, rank truncation of $J_l(\boldsymbol{v}_l^{MAP})$ in the paper [3].

3.3.5 Multilevel RTO estimator

This subsection combines all techniques depicted in this chapter so far to form the multilevel RTO estimator. We will have a multilevel estimator as $Q_{\mathbf{2}}$ with samples from the RTO density. We will describe the algorithm with the coupling and the transport mapping will

play an important role. We have the ultimate multilevel-RTO estimator as

$$\widehat{Q}_{\mathbf{2}} \equiv \frac{\widehat{Y}_L}{\hat{z}_L} = \frac{\sum_{l=0}^{L} \widehat{\Delta Y_l}}{\sum_{l=0}^{L} \widehat{\Delta \beta_l}} \tag{3.78}$$

where

$$\widehat{\Delta Y}_l = \begin{cases} \frac{1}{N_0} \sum_{i=1}^{N_0} \hat{\beta}_l\big(T_l(\boldsymbol{\Xi}_l^i)\big) \widehat{Q}_l\big(T_l(\boldsymbol{\Xi}_l^i)\big), & l = 0; \\ \frac{1}{N_l} \sum_{i=1}^{N_l} \hat{\beta}_l\big(T_l(\boldsymbol{\Xi}_l^i)\big) \widehat{Q}_l\big(T_l(\boldsymbol{\Xi}_l^i)\big) - \hat{\beta}_{l-1}\big(T_{l-1}(\boldsymbol{\Xi}_l^i)\big) \widehat{Q}_{l-1}\big(T_{l-1}(\boldsymbol{\Xi}_l^i)\big). & l = 1, 2, ..., L. \end{cases}$$

and

$$\widehat{\Delta \beta}_l = \begin{cases} \frac{1}{N_0} \sum_{i=1}^{N_0} \hat{\beta}_0\big(T_0(\boldsymbol{\Xi}_0^i)\big), & l = 0; \\ \frac{1}{N_l} \sum_{i=1}^{N_l} \hat{\beta}_l\big(T_l(\boldsymbol{\Xi}_l^i)\big) - \hat{\beta}_{l-1}\big(T_{l-1}(\boldsymbol{\Xi}_{l-1}^i)\big), & l = 1, 2, ..., L. \end{cases}$$

All the T_l's here demonstrate the transport mapping in RTO as in equation (3.63). $\boldsymbol{X}_l^i$ is the ith RTO samples at level l. We use $\boldsymbol{\Xi}$ to couple the samples from consecutive levels. The multilevel RTO algorithm with transport mapping is the shown in algorithm (3):

3.4 Analysis of complexity

This section will discuss the computational complexity of $\widehat{Q}_{\mathbf{2}}$. We will first discuss the decomposition of the mean square error of $\widehat{Q}_{\mathbf{2}}$. The delta method will work as a critical tool in the approximation of the MSE. We will combine all the derivations and provide a complexity theorem in the last section.

3.4.1 Decomposition of MSE

There are two possible biases or errors here: the discretization error and the bias between the expected value and the true value, which comes from the self-normalizing process. In the estimator $Q_{\mathbf{2}}$ we discussed here, both error terms exist. Later in this section we will prove that the bias is negligible compared to the sample variance of $\widehat{Q}_{\mathbf{2}}$ itself. The decomposition

Algorithm 3 Multilevel RTO with transport mapping

for $l = 0, 1, ..., L$ **do**
  Compute the MAP point $\boldsymbol{x}_l^{MAP}$;
  Form the Jacobian matrix $J(\boldsymbol{x}_l^{MAP})$;
  Construct the thin QR factorization of $J(\boldsymbol{x}_l^{MAP})$;
  for $i = 1, ..., N_l$ **do**
    Obtain one Gaussian random sample $\boldsymbol{\Xi}_l^i$;
    Obtain RTO samples $\boldsymbol{X}_l^i$ and $\boldsymbol{X}_{l-1}^i$ from solving (3.56) with the same $\boldsymbol{\Xi}_l^i$;
    Calculate the unnormalized weight: $\beta(\boldsymbol{X}_l^i)$ and $\beta(\boldsymbol{X}_{l-1}^i)$; and QoI $Q(\boldsymbol{X}_l^i)$ and $Q(\boldsymbol{X}_{l-1}^i)$;
    Construct the $\widehat{\Delta Y_l}$ in (3.19);
    Construct the $\widehat{\Delta \beta_l}$ in (3.22);
  end for
end for
Sum up all $\widehat{\Delta Y_l}$ to get $\widehat{Y}_L \equiv \sum_{l=0}^{L} \widehat{\Delta Y_l}$;
Sum up all $\widehat{\Delta \beta_l}$ to get $\widehat{z} \equiv \sum_{l=0}^{L} \widehat{\Delta \beta_l}$;
The ratio of $\widehat{Y}_L$ and $\widehat{z}$ forms $\widehat{Q}_{\mathbf{2}} \equiv \widehat{Y}_L / \hat{z}_L$.

of MSE starts from:

$$\begin{aligned}
\mathbb{E}\left(\widehat{Q}_{\mathbf{2}} - \mathbb{I}_Q\right)^2 &= \mathbb{E}\left(\widehat{Q}_{\mathbf{2}} - \mathbb{I}_{Q_L} + \mathbb{I}_{Q_L} - \mathbb{I}_Q\right)^2 \\
&= \mathbb{E}\left(\widehat{Q}_{\mathbf{2}} - \mathbb{I}_{Q_L}\right)^2 + \left(\mathbb{E}\left(\mathbb{I}_{Q_L} - \mathbb{I}_Q\right)\right)^2 + 2\mathbb{E}\left(\widehat{Q}_{\mathbf{2}} - \mathbb{I}_{Q_L}\right)\left(\mathbb{I}_{Q_L} - \mathbb{I}_Q\right) \\
&\leq 2\mathbb{E}\left(\widehat{Q}_{\mathbf{2}} - \mathbb{I}_{Q_L}\right)^2 + 2\left(\mathbb{I}_{Q_L} - \mathbb{I}_Q\right)^2, \qquad (3.79)
\end{aligned}$$

where the last line uses Cauchy–Schwarz inequality. The cross term $2\mathbb{E}\left(\widehat{Q}_{\mathbf{2}} - \mathbb{I}_{Q_L}\right)\left(\mathbb{I}_{Q_L} - \mathbb{I}_Q\right)$ cannot be removed since $\widehat{Q}_{\mathbf{2}}$ is biased, i.e., $\mathbb{E}\left(\widehat{Q}_{\mathbf{2}} - \mathbb{I}_{Q_L}\right) \neq 0$. The second term in the last inequality is the discretization error. We first rule out the discretization error $2\left(\mathbb{E}\left(\mathbb{I}_{Q_L} - \mathbb{I}_Q\right)\right)^2$ which is unavoidable and listed as a condition in the complexity theorem.

The other term can be further decomposed as

$$\begin{aligned}\mathbb{E}(\widehat{Q}_{\mathbf{2}} - \mathbb{I}_{Q_L})^2 &= \mathbb{E}\left(\widehat{Q}_{\mathbf{2}} - \mathbb{E}\left(\widehat{Q}_{\mathbf{2}}\right) + \mathbb{E}\left(\widehat{Q}_{\mathbf{2}}\right) - \mathbb{I}_{Q_L}\right)^2 \\ &= \mathbb{E}\left(\widehat{Q}_{\mathbf{2}} - \mathbb{E}\left(\widehat{Q}_{\mathbf{2}}\right)\right)^2 + \mathbb{E}\left(\mathbb{E}\left(\widehat{Q}_{\mathbf{2}}\right) - \mathbb{I}_{Q_L}\right)^2 + 2\mathbb{E}\left(\widehat{Q}_{\mathbf{2}} - \mathbb{E}\left(\widehat{Q}_{\mathbf{2}}\right)\right)\left(\mathbb{E}\left(\widehat{Q}_{\mathbf{2}}\right) - \mathbb{I}_{Q_L}\right) \\ &= \mathbb{V}(\widehat{Q}_{\mathbf{2}}) + \left(\mathbb{E}(\widehat{Q}_{\mathbf{2}}) - \mathbb{I}_{Q_L}\right)^2. \end{aligned} \tag{3.80}$$

The cross term is eliminated since $\mathbb{E}\left(\widehat{Q}_{\mathbf{2}} - \mathbb{E}\left(\widehat{Q}_{\mathbf{2}}\right)\right)$ is zero. $\mathbb{V}(\widehat{Q}_{\mathbf{2}})$ is the variance of the estimator $\widehat{Q}_{\mathbf{2}}$. The square $\left(\mathbb{E}(\widehat{Q}_{\mathbf{2}}) - \mathbb{I}_{Q_L}\right)^2$ is the square of bias from this ratio estimator. We are interested in which term dominates this sum. We expect the bias is far smaller than the variance so that we can omit it. The following section uses the delta method to verify the cost of both terms.

3.4.2 Delta method

The delta method is the approximation of a probability distribution derived from the first order Taylor polynomial. In this section, we will be specifically circumspect about the sample size of the estimators. We start with a univariate delta method to introduce the basic idea.

Univariate delta method. *Suppose there is a sequence of random variable X_i, i=1, 2, ..., N, satisfying*

$$\sqrt{N}(\overline{X}_N - \mu) \xrightarrow{d} \mathcal{N}(0, \sigma^2), \tag{3.81}$$

where μ and σ are mean and variance of X_i respectively, while μ and σ are both finite. Then for any first order differentiable function Υ, if Υ' is nonzero, we have

$$\sqrt{N}(\Upsilon(\overline{X}_N) - \Upsilon(\mu)) \xrightarrow{d} \mathcal{N}(0, \sigma^2\left(\Upsilon'(\mu)\right)^2). \tag{3.82}$$

A condition of the delta method is that the derivative Υ' exists and it is nonzero. The delta method gives an approximation of variance of a new random variable, a function Υ of a known random variable $\overline{X}_N$. This is a good strategy to deal with new random variables that are functions of known random variable. We are not using this univariate version in this thesis. We need the multivariate version of the delta method since we are dealing with a parameter vector.

Multivariate Delta method. *Suppose a vector $\boldsymbol{\Theta}$ is a consistent estimator whose true*

value is $\boldsymbol{\theta}$*. Asymptotically, the following holds:*

$$\sqrt{N}(\boldsymbol{\Theta} - \boldsymbol{\theta}) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}) \tag{3.83}$$

where $\boldsymbol{\Sigma}$ *is a positive semi-definite covariance matrix. Then, a scalar value function* $\Upsilon(\boldsymbol{\Theta})$ *converges in distribution as the following:*

$$\sqrt{N}\big(\Upsilon(\boldsymbol{\Theta}) - \Upsilon(\boldsymbol{\theta})\big) \xrightarrow{d} \mathcal{N}\bigg(0, \big(\nabla\Upsilon(\boldsymbol{\theta})\big)^T \boldsymbol{\Sigma} \big(\nabla\Upsilon(\boldsymbol{\theta})\big)\bigg). \tag{3.84}$$

A consistent estimator or asymptotically consistent estimator means that if the sample size of this estimator increases indefinitely, it converges in probability to its true value. The theorem can be proved by either the first-order Taylor theorem or the mean value theorem.

Ratio estimator by delta method

In our case, we want to estimate the performance of the ratio estimator $\widehat{Q}_{\mathbf{2}} = \widehat{Y}_L / \hat{z}_L$ where $\widehat{Y}_L$ and $\hat{z}_L$ are two multilevel estimators. In the framework of delta method, we have the vector $(\widehat{Y}_L, \hat{z}_L)^T$ as the consistent estimator $\boldsymbol{\Theta}$ (see chapter 6 of [12]). $\boldsymbol{\Theta}$ converges to

$$\boldsymbol{\theta} = \big(\mathbb{E}(\widehat{Y}_L), \mathbb{E}(\hat{z}_L)\big)^T = \bigg(\mathbb{E}_{g_L}\big(Q_L(\boldsymbol{X}_L)\beta_L(\boldsymbol{X}_L)\big), \mathbb{E}_{g_L}\big(\beta_L(\boldsymbol{X}_L)\big)\bigg)^T. \tag{3.85}$$

We denote $\boldsymbol{\theta} \equiv (\mu_{Y_L}, \mu_{z_L})^T$. $\boldsymbol{\Sigma}$ is the covariance matrix of $(Y_L, z_L)^T$. The subscripts here are the random variable Y_L and z_L instead of the sample-based multilevel estimators $\widehat{Y}_L$ and $\hat{z}_L$. Therefore function Υ is

$$\Upsilon(\boldsymbol{\Theta}) = \Upsilon\big((\widehat{Y}_L, \hat{z}_L)^T\big) \tag{3.86}$$

$$= \frac{\widehat{Y}_L}{\hat{z}_L} \equiv \widehat{Q}_{\mathbf{2}}. \tag{3.87}$$

The sample size N is primitively defined for an ordinary estimator or a single level estimator who takes N samples at a fixed level. In our setting, since it is a multilevel estimator, the definition of sample size is different. We will discuss the choice of sample size in detail later, while at the current stage we assume that the ratios between different sample sizes N_0, N_1, ..., N_L at the corresponding levels are fixed. Consequently, we only need to know one of the sample sizes. We can say that N_1, ..., N_L are determined by N_0. Assuming we

know N_0, then by delta method we have

$$\sqrt{N_0}\left(\frac{\widehat{Y}_L}{\hat{z}_L}-\frac{\mathbb{E}_{g_L}\left(Q_L\left(\boldsymbol{X}_L\right)\beta_L\left(\boldsymbol{X}_L\right)\right)}{\mathbb{E}_{g_L}\left(\beta_L(\boldsymbol{X}_L)\right)}\right)\xrightarrow{d}\mathcal{N}\left(0,\left(\nabla\Upsilon(\boldsymbol{\theta})\right)^T\boldsymbol{\Sigma}\left(\nabla\Upsilon(\boldsymbol{\theta})\right)\right). \tag{3.88}$$

The specific value of the asymptotic variance $\big(\nabla\Upsilon(\boldsymbol{\theta})\big)^T\boldsymbol{\Sigma}\big(\nabla\Upsilon(\boldsymbol{\theta})\big)$ is not listed here, where the key information is that it is a scalar irrelevant to the sample size N_0. Hence, $\widehat{Y}_L/\hat{z}_L$ has the variance proportional to $\frac{1}{N_0}$:

$$\mathbb{V}\left(\frac{\widehat{Y}_L}{\hat{z}_L}\right)\approx\frac{1}{N_0}\left(\nabla\Upsilon(\boldsymbol{\theta})\right)^T\boldsymbol{\Sigma}\left(\nabla\Upsilon(\boldsymbol{\theta})\right) \tag{3.89}$$

We have approximated the variance of $\widehat{Q}_{\mathbf{2}}$ in (3.79). The next step is to explore the bias term $\big(\mathbb{E}(\widehat{Q}_2)-\mathbb{I}_{Q_L}\big)^2$ by means of the delta method. The following paragraphs show that this bias term is $O(\frac{1}{N_0^2})$.

We again approximate the ratio $\widehat{Y}_L/\hat{z}_L$ in a second order Taylor series as in the following equation:

$$\Upsilon(\boldsymbol{\Theta})\approx\Upsilon(\boldsymbol{\theta})+\left(\nabla\Upsilon\left(\boldsymbol{\theta}\right)\right)^T\left(\boldsymbol{\Theta}-\boldsymbol{\theta}\right)+\frac{1}{2}(\boldsymbol{\Theta}-\boldsymbol{\theta})^T\nabla^2\Upsilon(\boldsymbol{\theta})(\boldsymbol{\Theta}-\boldsymbol{\theta}), \tag{3.90}$$

where $\nabla^2\Upsilon(\boldsymbol{\theta})$ is the Hessian matrix at the fixed point $\boldsymbol{\theta}$, and thus it is a fixed 2×2 matrix in the case where $\Upsilon(\boldsymbol{\Theta})=\widehat{Y}_L/\hat{z}_L$. In the scenario of the ratio estimator, $\mathbb{E}\big(\Upsilon(\boldsymbol{\Theta})\big)-\Upsilon(\boldsymbol{\theta})=\mathbb{E}(\widehat{Q}_{\mathbf{2}})-\mathbb{I}_{Q_L}$. Therefore, we can take the expectation of $\Upsilon(\boldsymbol{\Theta})-\Upsilon(\boldsymbol{\theta})$ from the second order Taylor series:

$$\mathbb{E}(\widehat{Q}_2)-\mathbb{I}_{Q_L}\equiv\mathbb{E}\left(\Upsilon(\boldsymbol{\Theta})-\Upsilon(\boldsymbol{\theta})\right)\approx\frac{1}{2}\mathbb{E}\left(\left(\boldsymbol{\Theta}-\boldsymbol{\theta}\right)^T\nabla^2\Upsilon(\boldsymbol{\theta})\left(\boldsymbol{\Theta}-\boldsymbol{\theta}\right)\right).$$

The first order term $\big(\nabla\Upsilon(\boldsymbol{\theta})\big)^T(\boldsymbol{\Theta}-\boldsymbol{\theta})$ vanishes since $\mathbb{E}(\boldsymbol{\Theta})=\boldsymbol{\theta}$.

The following equations deal with the expectation of the second order remainder:

$$\begin{aligned}
\mathbb{E}\big((\boldsymbol{\Theta}-\boldsymbol{\theta})^T\nabla^2\Upsilon(\boldsymbol{\theta})(\boldsymbol{\Theta}-\boldsymbol{\theta})\big) &\approx \mathbb{E}\bigg(\operatorname{tr}\big((\boldsymbol{\Theta}-\boldsymbol{\theta})^T\nabla^2\Upsilon(\boldsymbol{\theta})(\boldsymbol{\Theta}-\boldsymbol{\theta})\big)\bigg) \\
&= \mathbb{E}\bigg(\operatorname{tr}\big(\nabla^2\Upsilon(\boldsymbol{\theta})(\boldsymbol{\Theta}-\boldsymbol{\theta})(\boldsymbol{\Theta}-\boldsymbol{\theta})^T\big)\bigg) \\
&= \operatorname{tr}\bigg(\mathbb{E}\big(\nabla^2\Upsilon(\boldsymbol{\theta})(\boldsymbol{\Theta}-\boldsymbol{\theta})(\boldsymbol{\Theta}-\boldsymbol{\theta})^T\big)\bigg) \\
&= \operatorname{tr}\bigg(\nabla^2\Upsilon(\boldsymbol{\theta})\mathbb{E}\big((\boldsymbol{\Theta}-\boldsymbol{\theta})(\boldsymbol{\Theta}-\boldsymbol{\theta})^T\big)\bigg) \\
&= \operatorname{tr}\left(\nabla^2\Upsilon(\boldsymbol{\theta})\mathbb{V}\left(\boldsymbol{\Theta}\right)\right) \\
&\approx \frac{1}{N_0}\operatorname{tr}\left(\nabla^2\Upsilon\left(\boldsymbol{\theta}\right)\boldsymbol{\Sigma}\right). \qquad (3.91)
\end{aligned}$$

The first equation holds because the quadratic term is simply a scalar. The second equation uses the property of trace. The third applies linearity of expectation. From an assumption of the delta method, the asymptotic variance of $\boldsymbol{\Theta}$ is $1/N_0\boldsymbol{\Sigma}$, which gives the last approximation. We can see that the bias is proportional to $1/N_0$. In the MSE decomposition, the bias term $\left(\mathbb{E}(\widehat{Q}_2)-\mathbb{I}_{Q_L}\right)^2$ equals $\left(\mathbb{E}\left(\Upsilon\left(\boldsymbol{\Theta}\right)-\Upsilon\left(\boldsymbol{\theta}\right)\right)\right)^2$. Hence, the bias term is proportional to $1/N_0^2$, which is negligible compared to the variance term, that is $O(\frac{1}{N_0})$, for large N_0.

3.4.3 Complexity analysis

From the previous subsection we have proved that the variance of this ratio estimator dominates the quantity in (3.80). We can now further break down the variance. We first review the variance of the first order approximation of the $\widehat{Y}_L/\hat{z}_L$ as shown in the delta method:

$$\mathbb{V}\left(\frac{\widehat{Y}_L}{\hat{z}_L}\right) \approx \frac{1}{N_0}\left(\nabla\Upsilon(\boldsymbol{\theta})\right)^T\boldsymbol{\Sigma}\big(\nabla\Upsilon(\boldsymbol{\theta})\big).$$

We need an explicit form of this variance. The first part is the derivative $\nabla\Upsilon(\boldsymbol{\theta})$ which can be expressed as follows:

$$\nabla\Upsilon(\boldsymbol{\theta}) = (\frac{1}{\mathbb{E}(z_L)}, -\frac{\mathbb{E}(Y_L)}{\mathbb{E}^2(z_L)})^T. \qquad (3.92)$$

This vector contains the partial derivative with respect to $\mathbb{E}(Y_L)$ and $\mathbb{E}(z_L)$. The Y_L and z_L here are random variables, not sample-based estimators. Y_L and z_L are $Q(\cdot)\beta(\cdot)$ and $\beta(\cdot)$

at level L. We can also denote $\mathbf{\Sigma}$ explicitly as

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma^2_{Y_L} & \sigma_{Y_L,z_L} \\ \sigma_{Y_L,z_L} & \sigma^2_{z_L} \end{pmatrix}, \tag{3.93}$$

since $\mathbf{\Sigma}$ is the covariance matrix of $(Y_L, z_L)^T$. The three equations above imply the following expression of variance:

$$\begin{aligned}
\mathbb{V}(\frac{\widehat{Y}_L}{\hat{z}_L}) &\approx \frac{1}{N_0}\big(\nabla\Upsilon(\boldsymbol{\theta})\big)^T\mathbf{\Sigma}\big(\nabla\Upsilon(\boldsymbol{\theta})\big) \\
&= \frac{1}{N_0}\left(\sigma^2_{Y_L}\left(\frac{1}{\mathbb{E}(z_L)}\right)^2 + \sigma^2_{z_L}\left(\frac{\mathbb{E}(Y_L)}{\mathbb{E}^2(z_L)}\right)^2 - 2\sigma_{Y_L,z_L}\frac{\mathbb{E}(Y_L)}{\mathbb{E}^3(z_L)}\right) \\
&= \frac{1}{N_0}\frac{1}{\mathbb{E}^2(z_L)}\left(\sigma^2_{Y_L} + \sigma^2_{z_L}\left(\frac{\mathbb{E}(Y_L)}{\mathbb{E}(z_L)}\right)^2 - 2\sigma_{Y_L,z_L}\frac{\mathbb{E}(Y_L)}{\mathbb{E}(z_L)}\right) \\
&= \frac{1}{N_0}\frac{1}{\mathbb{E}^2(z_L)}\mathbb{V}\left(Y_L - z_L\frac{\mathbb{E}(Y_L)}{\mathbb{E}(z_L)}\right) \\
&= \frac{1}{N_0}\frac{1}{\mathbb{E}^2(z_L)}\mathbb{E}\left(Y_L - z_L\frac{\mathbb{E}(Y_L)}{\mathbb{E}(z_L)} - \mathbb{E}(Y_L - z_L\frac{\mathbb{E}(Y_L)}{\mathbb{E}(z_L)})\right)^2 \\
&= \frac{1}{N_0}\frac{1}{\mathbb{E}^2(z_L)}\mathbb{E}\left(Y_L - z_L\frac{\mathbb{E}(Y_L)}{\mathbb{E}(z_L)} - 0\right)^2 \\
&= \frac{1}{N_0}\frac{1}{\mathbb{E}^2(z_L)}\mathbb{E}\left(Y_L - z_L\Upsilon\left(\boldsymbol{\theta}\right)\right)^2.
\end{aligned} \tag{3.94}$$

All the intermediate steps are simply playing with definitions and properties of expectation and variance. The second line is simply substituting $\nabla\Upsilon(\boldsymbol{\theta})$ and $\mathbf{\Sigma}$ into the formula. The third line pulls out the common factor $1/\mathbb{E}^2(z_L)$. The fourth line is to form a variance or covariance based on the definition. The fifth line is expanding the variance in another form. Note that on the last line of the equations above, the ratio $\mathbb{E}(Y_L)/\mathbb{E}(z_L)$ is equivalent to the demanding true value, $\Upsilon(\boldsymbol{\theta})$. Equation (3.94) provides us with something estimable. It is a ratio, and the denominator $\mathbb{E}^2(z_L)$ is a constant. If we replace z_L with its estimator $\hat{z}_L$, $\mathbb{E}^2(\hat{z}_L)$ is bounded. It is reassuring to know that the denominator is bounded so that the ratio is less likely to tend to infinity.

The formation of variance in (3.94) is not related to any estimator we can yet compute. However if we can split out Y_L and z_L out, we can replace these two with corresponding

estimators. We can further decompose the variance (3.94) above:

$$\begin{aligned}
\mathbb{V}(\frac{\widehat{Y}_L}{\hat{z}_L}) &\approx \frac{1}{N_0}\frac{1}{\mathbb{E}^2(z_L)}\mathbb{E}\big(Y_L - z_L\Upsilon(\boldsymbol{\theta})\big)^2 \\
&= \frac{1}{N_0}\frac{1}{\mathbb{E}^2(z_L)}\mathbb{E}\big(Y_L - \mathbb{E}(Y_L) + \mathbb{E}(Y_L) - z_L\Upsilon(\boldsymbol{\theta})\big)^2 \\
&= \frac{1}{N_0}\frac{1}{\mathbb{E}^2(z_L)}\mathbb{E}\left(\big(Y_L - \mathbb{E}(Y_L)\big)^2 + \big(\mathbb{E}(Y_L) - z_L\Upsilon(\boldsymbol{\theta})\big)^2 + 2\big(Y_L - \mathbb{E}(Y_L)\big)\big(\mathbb{E}(Y_L) - z_L\Upsilon(\boldsymbol{\theta})\big)\right) \\
&\leq 2\frac{1}{N_0}\frac{1}{\mathbb{E}^2(z_L)}\left(\mathbb{E}\big(Y_L - \mathbb{E}(Y_L)\big)^2 + \mathbb{E}\big(\mathbb{E}(Y_L) - z_L\Upsilon(\boldsymbol{\theta})\big)^2\right) \\
&= 2\frac{1}{N_0}\frac{1}{\mathbb{E}^2(z_L)}\left(\mathbb{V}(Y_L) + \mathbb{E}\big(\Upsilon(\boldsymbol{\theta})\mathbb{E}(z_L) - z_L\Upsilon(\boldsymbol{\theta})\big)^2\right) \\
&= 2\frac{1}{N_0}\frac{1}{\mathbb{E}^2(z_L)}\left(\mathbb{V}(Y_L) + \Upsilon^2(\boldsymbol{\theta})\mathbb{V}(z_L)\right). \qquad (3.95)
\end{aligned}$$

We now have a good approximation of the variance, which is a linear combination of two variances. Note that $\mathbb{V}(Y_L)$ and $\mathbb{V}(z_L)$ are not sample-based estimators. A sample-based estimate is

$$\begin{aligned}
\mathbb{V}(\widehat{Q}_{\mathbf{2}}) \equiv \mathbb{V}(\frac{\widehat{Y}_L}{\hat{z}_L}) &\approx 2\frac{1}{N_0}\frac{1}{\mathbb{E}^2(z_L)}\big(N_0\mathbb{V}(\widehat{Y}_L) + \Upsilon^2(\boldsymbol{\theta})N_0\mathbb{V}(\hat{z}_L)\big) \\
&= 2\frac{1}{\mathbb{E}^2(\hat{z})}\big(\mathbb{V}(\widehat{Y}_L) + \Upsilon^2(\boldsymbol{\theta})\mathbb{V}(\hat{z}_L)\big). \qquad (3.96)
\end{aligned}$$

We then have the variance of this self-normalizing multilevel estimator in terms of variances of two standard multilevel estimators. This builds a solid bridge with the complexity theorem we have in 2.3. This variance is very important when deciding the proportional sample sizes across levels. It is also critical when proving the complexity theorem.

Therefore, this ratio estimator has the MSE in equation (3.79) approximated as

$$\begin{aligned}
\mathbb{E}(\widehat{Q}_{\mathbf{2}} - \mathbb{I}_Q)^2 &\leq 2\mathbb{E}(\widehat{Q}_{\mathbf{2}} - \mathbb{I}_{Q_L})^2 + 2\left(\mathbb{I}_{Q_L} - \mathbb{I}_Q\right)^2 \\
&= 2\left(\mathbb{V}(\widehat{Q}_{\mathbf{2}}) + \big(\mathbb{E}(\widehat{Q}_{\mathbf{2}}) - \mathbb{I}_{Q_L}\big)^2\right) + 2\left(\mathbb{I}_{Q_L} - \mathbb{I}_Q\right)^2 \\
&\approx 2\left(2\frac{1}{\mathbb{E}^2(\hat{z})}\left(\mathbb{V}(\widehat{Y}_L) + \Upsilon^2(\boldsymbol{\theta})\mathbb{V}(\hat{z}_L)\right)\right) + 2\left(\mathbb{I}_{Q_L} - \mathbb{I}_Q\right)^2. \qquad (3.97)
\end{aligned}$$

The second equation uses the results in equation (3.80), and the last equation uses the

results in equations (3.91) and (3.96), that is, the square of bias is $O(\frac{1}{N_0^2})$. We can see from the approximation in equation (3.97) that the MSE is decomposed into a sum of variance and the discretization error, which is very similar to the MSE decomposition of a standard multilevel estimator. Hence, we can start with the same conditions of standard multilevel complexity theorem.

We are ready to connect the MSE result and the complexity theorem. We first restate the conditions of the standard multilevel complexity theorem (2.3.1) here:

Assumption 3.4.1. For a multilevel estimator $\widehat{Q} \equiv \sum_{l=0}^{L} \widehat{\Delta Q_l}$, there exists positive constants ζ, α, and η such that:

1. the discretization error is bounded as $\mathbb{E}\left[\widehat{\Delta Q_l} - \mathbb{I}_Q\right] = O(h_l^{\zeta})$;
2. the variance of each $\widehat{\Delta Q_l}$ defined above is bounded as $\mathbb{V}\left[\widehat{\Delta Q_l}\right] = O(\frac{h_l^{\alpha}}{N_l})$;
3. the computational complexity of $\widehat{\Delta Q_l}$ is bounded by $C_l = O(N_l h_l^{-\eta})$.

The first condition is about the discretization error, while the second one is about the variance. In the case that $\widehat{Q}_{\mathbf{2}} \equiv \frac{\widehat{Y}_L}{\widehat{z}_L}$, $\widehat{\Delta Q_l}$ is either $\widehat{\Delta Y}_l$ in $\widehat{Y}_L \equiv \sum_{l=0}^{L} \widehat{\Delta Y_l}$, or $\widehat{\Delta \beta}_l$ in $\hat{z}_L \equiv \sum_{l=0}^{L} \widehat{\Delta \beta}_l$. If we could assume these three conditions hold for $\widehat{\Delta Y}_l$, then

$$\mathbb{V}(\widehat{Y}_L) \equiv \mathbb{V}(\sum_{l=0}^{L} \widehat{\Delta Y_l}) \tag{3.98}$$

$$\leq \sum_{l=0}^{L} \mathbb{V}(\widehat{\Delta Y_l}) + 2 \sum_{l \neq k} \mathrm{Cov}(\widehat{\Delta Y}_l, \widehat{\Delta Y}_k). \tag{3.99}$$

Since we use the coupling variate in the multilevel RTO algorithm, two different $\widehat{\Delta Y_l}$'s are not correlated since they are mapped from different reference variables $\boldsymbol{\Xi}$, therefore their covariance is close to zero. We can also use other coupling techniques to make the covariance small, but we do not discuss other coupling techniques here. We do not dismiss the possibility of zero covariance here, but the possibility is very small. We can have

$$\mathbb{V}(\widehat{Y}_L) \approx \sum_{l=0}^{L} \mathbb{V}(\widehat{\Delta Y_l}) = \sum_{l=0}^{L} O(\frac{h_l^{\alpha}}{N_l}). \tag{3.100}$$

The same derivation also applies to $\mathbb{V}(\hat{z}_L)$:

$$\mathbb{V}(\hat{z}_L) \approx \sum_{l=0}^{L} \mathbb{V}(\widehat{\Delta\beta_l}) = \sum_{l=0}^{L} O(\frac{h_l^\alpha}{N_l}). \tag{3.101}$$

In numerical experiments we have conducted, $\widehat{\Delta Y}_l$ and $\widehat{\Delta\beta}_l$ have the same convergent rate α. The other two convergent rates are also the same for $\widehat{\Delta Y}_l$ and $\widehat{\Delta\beta}_l$ respectively in numerical results. We will have more discussion on these convergent rates in Chapter 4. We return the discussion to the MSE. The following is a summary of the MSE result derived from equation (3.97):

$$\mathbb{E}(\widehat{Q}_{\mathbf{2}} - \mathbb{I}_Q)^2 \leq 2\left(2\frac{1}{\mathbb{E}^2(\hat{z})}\left(\mathbb{V}(\widehat{Y}_L) + \Upsilon^2(\boldsymbol{\theta})\mathbb{V}(\hat{z}_L)\right)\right) + 2\left(\mathbb{E}\left(\mathbb{I}_{Q_L} - \mathbb{I}_Q\right)\right)^2 \tag{3.102}$$

$$= \sum_{l=0}^{L} O(\frac{h_l^\alpha}{N_l}) + O(h_l^\zeta). \tag{3.103}$$

These $big - O$ terms are using conditions 1 and 2 in assumption (3.4.1).

Before diving into the proof of complexity theorem, we explain how to determine the sample size across levels. We want the sample size to minimize the cost $\sum_{l=0}^{L} N_l C_l$, constrained to make the variance $\sum_{l=0}^{L} \mathbb{V}\left(\widehat{\Delta Y}_l\right)/N_l$ fixed. Using a Lagrange multiplier gives:

$$N_l \propto \sqrt{\frac{\mathbb{V}\left(\widehat{\Delta Y}_l\right)}{C_l}}. \tag{3.104}$$

The same minimisation applies to $\widehat{\Delta\beta}_l$ as well.

The chapter finishes with the following complexity theorem and its according proof:

Theorem 3.4.1. Suppose the decretization size $h_l = M^{-l}T$. If

$$\widehat{\Delta Y}_l \equiv \begin{cases} \hat{\beta}_0\widehat{Q}_0 & l = 0; \\ \hat{\beta}_l\widehat{Q}_l - \hat{\beta}_{l-1}\widehat{Q}_{l-1}, & l = 1,\ 2,\ ...,\ L; \end{cases}$$

satisfies

(i) the discretization error is bounded as $\mathbb{E}|\hat{\beta}_l\widehat{Q}_l - Q\beta| \leq c_1 h_l^\zeta$,

(ii) the variance of each $\widehat{\Delta Y}_l$ defined above is bounded as $\mathbb{V}\left(\widehat{\Delta Y}_l\right) \leq c_2 \frac{h_l^\alpha}{N_l}$,

(iii) the computational complexity of $\widehat{\Delta Y}_l$ is bounded by $C_l \leq c_3 N_l h_l^{-\eta}$,

for positive constants $\zeta \geq \frac{1}{2}, \alpha,\ \eta,\ c_1,\ c_2$ and c_3; and

$$\widehat{\Delta\beta}_l \equiv \begin{cases} \hat{\beta}_0 & l = 0; \\ \hat{\beta}_l - \hat{\beta}_{l-1}, & l = 1,\ 2,\ ...,\ L; \end{cases}$$

satisfies

(a) the discretization error is bounded as $\mathbb{E}|\hat{\beta}_l - \beta| \leq d_1 h_l^\zeta$,

(b) the variance of each $\widehat{\Delta\beta}_l$ defined above is bounded as $\mathbb{V}\left(\widehat{\Delta\beta}_l\right) \leq d_2 \frac{h_l^\alpha}{N_l}$,

(c) the computational complexity of $\widehat{\Delta\beta}_l$ is bounded by $C_l \leq d_3 N_l h_l^{-\eta}$,

for positive constants $\zeta \geq \frac{1}{2},\ d_1,\ d_2$ and d_3, then the computational cost of the multilevel ratio estimator $\widehat{Q}_{\mathbf{2}}$ with a mean square error(MSE) $\mathbb{E}\big(\widehat{Q}_{\mathbf{2}} - \mathbb{I}_Q\big)^2 < \epsilon^2$, where $\epsilon < e^{-1}$, is

$$C^{ML} = \begin{cases} O\big(\epsilon^{-2}\big), & \alpha > \eta, \\ O\Big(\epsilon^{-2}\big(\log(\epsilon)\big)^2\Big), & \alpha = \eta, \\ O\big(\epsilon^{-2-(\eta-\alpha)/\zeta}\big), & 0 < \alpha < \eta. \end{cases}$$

Proof. This proof follows the paper [18] with some adjustments. The MSE decomposition is different from the original MSE in [18]:

$$\mathbb{E}(\widehat{Q}_{\mathbf{2}} - \mathbb{I}_Q)^2 \approx 2\left(\frac{1}{\mathbb{E}^2(\hat{z})}\big(\mathbb{V}(\widehat{Y}_L) + \Upsilon^2(\boldsymbol{\theta})\mathbb{V}(\hat{z}_L)\big)\right) + 2\left(\mathbb{I}_{Q_L} - \mathbb{I}_Q\right)^2. \tag{3.105}$$

Following the idea of the proof in [18], we first want to make the discretization error bounded. However, we can't directly address this error since we don't have an unbiased estimator for $\mathbb{I}_{Q_L}$. We need to link the conditions (i) and (a) to the discretization error.

We call $\mathbb{I}_{Q_L}$ as follows:

$$\mathbb{I}_{Q_L} \equiv \frac{Y_L}{z_L} = \frac{\mathbb{E}\left(\widehat{Y}_L\right)}{\mathbb{E}\left(\hat{z}_L\right)} = \frac{\mathbb{E}\left(\sum_{l=0}^{L} \widehat{\Delta Y}_l\right)}{\mathbb{E}\left(\sum_{l=0}^{L} \widehat{\Delta \beta}_l\right)} = \frac{\sum_{l=0}^{L}\left(\mathbb{E}(\widehat{\Delta Y}_l)\right)}{\sum_{l=0}^{L}\left(\mathbb{E}(\widehat{\Delta \beta}_l)\right)}. \tag{3.106}$$

Thus, the discretization error can be written as

$$|\mathbb{I}_{Q_L} - \mathbb{I}_Q| = \left|\frac{Y_L}{z_L} - \frac{Y_\infty}{z_\infty}\right|$$

since when the level L approaches infinity, the ratio estimator is $\mathbb{I}_Q$. We can bound this error as follows:

$$\begin{aligned}
\left|\frac{Y_L}{z_l} - \frac{Y_\infty}{z_\infty}\right| &= \left|\frac{Y_L}{z_L} - \frac{Y_\infty}{z_L} + \frac{Y_\infty}{z_L} - \frac{Y_\infty}{z_\infty}\right| \\
&= \left|\frac{Y_L - Y_\infty}{z_L} + Y_L\left(\frac{1}{z_L} - \frac{1}{z_\infty}\right)\right| \\
&\leq \left|\frac{Y_L - Y_\infty}{z_L}\right| + \left|Y_L\left(\frac{1}{z_L} - \frac{1}{z_\infty}\right)\right| \\
&= \left|\frac{\sum_{l=0}^{L}\left(\mathbb{E}(\widehat{\Delta Y}_l)\right) - Y_\infty}{z_L}\right| + \left|Y_L \frac{1}{z_L z_\infty}\left(z_\infty - \sum_{l=0}^{L}\left(\mathbb{E}(\widehat{\Delta \beta}_l)\right)\right)\right| \\
&\leq \left|\frac{c_1 h_L^\zeta}{z_L}\right| + \left|Y_L \frac{1}{z_L z_\infty}\left(d_1 h_L^\zeta\right)\right| \\
&\leq d_4 h_L^\zeta,
\end{aligned}$$

where $d_4 = \left|\frac{c_1}{z_L}\right| + \left|\frac{Y_L d_1}{z_L z_\infty}\right|$ and the last line uses condition (i) and (a). In summary,

$$\mathbb{E}|\mathbb{I}_{Q_L} - \mathbb{I}_Q| \leq d_4 h_L^\zeta.$$

We use the notation $\lceil \cdot \rceil$ to denote the ceiling of a real number. We choose L to be

$$L = \lceil \frac{\log\left(\sqrt{2} d_4 T^\zeta \epsilon^{-1}\right)}{\zeta \log M} \rceil \tag{3.107}$$

so that

$$\frac{1}{\sqrt{2}}M^{-\zeta}\epsilon \leq d_4 h_L^{\zeta} \leq \frac{1}{\sqrt{2}}\epsilon, \tag{3.108}$$

and hence

$$\left(\mathbb{I}_{Q_L} - \mathbb{I}_Q\right)^2 \leq \frac{1}{2}\epsilon^2. \tag{3.109}$$

We have distributed $\epsilon^2/2$ to the discretization error, and the other $\epsilon^2/2$ is the variance of $\widehat{Q}_{\mathbf{2}}$.

There is another useful inequality. We obtain the following formula from a geometric series, which will be essential in a later proof:

$$\sum_{l=0}^{L} h_l^{-\eta} = h_L^{-\eta} \sum_{l=0}^{L} M^{-\eta l} < \frac{M^{\eta}}{M^{\eta}-1} h_L^{-\eta}. \tag{3.110}$$

On the other hand, $h_L^{-\eta}$ is also bounded as

$$h_L^{-\eta} < M^{\eta} \left(\frac{\epsilon}{\sqrt{2}d_4}\right)^{-\frac{1}{\zeta}}. \tag{3.111}$$

Combining the first inequality of (3.108) and the inequality above, we have the following useful inequality:

$$\sum_{l=0}^{L} h_l^{\eta} \leq \frac{M^{2\eta}}{M^{\eta}-1} \left(\sqrt{2}d_4\right)^{1/\zeta} \epsilon^{-2}. \tag{3.112}$$

(**I**) If $\alpha = \eta$, we set $N_l = \lceil 2\epsilon^{-2}(L+1)c_4 h_l^{\eta} \rceil$ so that

$$\begin{aligned}
\mathbb{V}(\widehat{Q}_{\mathbf{2}}) &\approx 2\left(2\frac{1}{\mathbb{E}^2(\hat{z})}\left(\mathbb{V}(\widehat{Y}_L) + \Upsilon^2(\boldsymbol{\theta})\mathbb{V}(\hat{z}_L)\right)\right) \\
&\equiv \widetilde{\mathbb{V}}(\widehat{Q}_{\mathbf{2}}) && (3.113) \\
&\leq 4\left(\frac{1}{\mathbb{E}^2(\hat{z})}\left(c_2 \sum_{l=0}^{L} \frac{h_l^{\alpha}}{N_l} + \Upsilon^2(\boldsymbol{\theta}) d_2 \sum_{l=0}^{L} \frac{h_l^{\alpha}}{N_l}\right)\right) && (3.114) \\
&= c_4 \sum_{l=0}^{L} \frac{h_l^{\alpha}}{N_l} = c_4 \sum_{l=0}^{L} \frac{h_l^{\eta}}{N_l} \leq \frac{1}{2}\epsilon^2, && (3.115)
\end{aligned}$$

where c_4 is given by

$$c_4 = 4\frac{1}{\mathbb{E}^2(\hat{z})}\left(c_2 + \Upsilon^2(\boldsymbol{\theta})d_2\right). \tag{3.116}$$

This sets the upper bound of the variance to $\epsilon^2/2$. We also need to bound the complexity:

$$C^{ML} \leq \sum_{l=0}^{L}(c_3+d_3)N_l h_l^{-\eta} \leq \sum_{l=0}^{L}(c_3+d_3)\left(2\epsilon^{-2}\left(L+1\right)^2 c_4 h_l^{\eta} + h_l^{-\eta}\right). \tag{3.117}$$

The only quantity we can control here is L. We can set:

$$L \leq \frac{\log\epsilon^{-1}}{\zeta\log M} + \frac{\log\left(\sqrt{2}d_4 T^{\zeta}\right)}{\zeta\log M} + 1. \tag{3.118}$$

Since $\epsilon < e^{-1}$, we have $1 < \log\epsilon^{-1}$. It follows that

$$L + 1 \leq c_5 \log\epsilon^{-1}, \tag{3.119}$$

where c_5

$$c_5 = \frac{1}{\zeta\log M} + \max\left(0, \frac{\log\left(\sqrt{2}d_4 T^{\zeta}\right)}{\zeta\log M}\right) + 2.$$

If we use the upper bound for $L+1$, inequality (3.112), and the fact that $1 < \log\epsilon^{-1}$, we have

$$C^{ML} \leq c_6\epsilon^{-2}\left(\log\epsilon\right)^2, \tag{3.120}$$

where

$$c_6 = 2(c_3+d_3)c_5^2 c_4 + (c_3+d_3)\frac{M^{1\eta}}{M^{\eta}-1}(\sqrt{2}c_4)^{1/\zeta} \tag{3.121}$$

We have finished proving the second case when $\alpha = \eta$.

(**II**) For $\alpha > \eta$, if we set

$$N_l = \lceil 2\epsilon^2 c_4 T^{(\alpha-\eta)/2}(1 - M^{-(\alpha-\eta)/2})^{-1} h_l^{(\alpha+\eta)/2}\rceil, \tag{3.122}$$

then the variance

$$\begin{aligned}\widetilde{\mathbb{V}}(\widehat{Q}_{\mathbf{2}}) &\leq c_4 \sum_{l=0}^{L} \frac{h_l^{\eta}}{N_l} \\ &\leq \frac{1}{2}\epsilon^2 c_4 T^{-(\alpha-\eta)/2}(1 - M^{-(\alpha-\eta)/2}) \sum_{l=0}^{L} h_l^{(\alpha-\eta)/2}.\end{aligned}$$

Using the geometric series result again gives:

$$\begin{aligned}\sum_{l=0}^{L} h_l^{(\alpha-\eta)/2} &= T^{(\alpha-\eta)/2} \sum_{l=0}^{L} (M^{-(\alpha-\eta)/2})^l \\ &< T^{(\alpha-\eta)/2}(1 - M^{-(\alpha-\eta)/2})^{-1}. \end{aligned} \tag{3.123}$$

We state the upper bound of N_l again:

$$N_l < 2\epsilon^2 T^{(\alpha-\eta)/2}(1 - M^{-(\alpha-\eta)/2})^{-1} h_l^{(\alpha+\eta)/2} + 1.$$

Therefore the computational complexity is bounded by

$$C^{ML} \leq (c_3 + d_3)\left(N_l h_l^{-\eta}\right) \leq (c_3 + d_3)\left(2\epsilon^{-2} c_4 T^{(\alpha-\eta)/2}(1 - M^{-(\alpha-\eta)/2})^{-1} \sum_{l=0}^{L} h_l^{(\alpha-\eta)/2} + \sum_{l=0}^{L} h_l^{-\eta}\right).$$

From inequalities (3.112) and (3.123), the cost is bounded as

$$C^{ML} \leq c_7 \epsilon^{-2},$$

where

$$c_7 = 2(c_3 + d_3) c_4 T^{\alpha-\eta}\left(1 - M^{-(\alpha-\eta)/2}\right)^{-2} + (c_3 + d_3)\frac{M^{2\eta}}{M^{\eta} - 1}(\sqrt{2} d_4)^{1/\zeta}.$$

(**III**) This case is similar to case (b). For $\alpha < \eta$, we set the sample size N_l as

$$N_l = \lceil 2\epsilon^2 c_4 h_L^{-(\eta-\alpha)/2}(1 - M^{-(\eta-\alpha)/2})^{-1} h_l^{(\alpha+\eta)/2} \rceil. \tag{3.124}$$

The variance is bounded as

$$\begin{aligned}\widetilde{\mathbb{V}}(\widehat{Q}_{\mathbf{2}}) &\leq c_4 \sum_{l=0}^{L} \frac{h_l^{\eta}}{N_l} \\ &\leq \frac{1}{2}\epsilon^2 c_4 h_L^{(\eta-\alpha)/2}(1 - M^{-(\eta-\alpha)/2}) \sum_{l=0}^{L} h_l^{-(\eta-\alpha)/2}. \end{aligned} \tag{3.125}$$

We again bound the geometric series as

$$\begin{aligned}\sum_{l=0}^{L} h_l^{-(\eta-\alpha)/2} &= h_L^{(\alpha-\eta)/2} \sum_{l=0}^{L} (M^{-(\alpha-\eta)/2})^l \\ &< h_L^{(\alpha-\eta)/2}(1 - M^{-(\alpha-\eta)/2})^{-1}. \end{aligned} \tag{3.126}$$

Combining this result of the geometric series and (3.125), the variance is bounded by $\epsilon^2/2$.

We use the N_l upper bound again to bound the computational complexity as

$$C^{ML} \leq (c_3 + d_3)\left(N_l h_l^{-\eta}\right) \leq (c_3 + d_3)\left(2\epsilon^{-2} c_4 h_L^{-(\eta-\alpha)/2}(1 - M^{-(\eta-\alpha)/2})^{-1} \sum_{l=0}^{L} h_l^{-(\eta-\alpha)/2} + \sum_{l=0}^{L} h_l^{-\eta}\right).$$

The inequality in (3.126) implies

$$h_L^{-(\eta-\alpha)/2}(1 - M^{-(\eta-\alpha)/2})^{-1} \sum_{l=0}^{L} h_l^{-(\eta-\alpha)/2} \leq h_L^{-(\eta-\alpha)}(1 - M^{-(\eta-\alpha)/2})^{-2}. \tag{3.127}$$

The first inequality in (3.108) also provides an upper bound for $h_L^{-(\eta-\alpha)}$:

$$h_L^{-(\eta-\alpha)} < (\sqrt{2}c_4)^{(\eta-\alpha)/\zeta} M^{1-\alpha} \epsilon^{-(\eta-\alpha)/\zeta}. \tag{3.128}$$

Combining these two inequalities, and the inequality in (3.112), as well as the fact that $\epsilon^{-2} < \epsilon^{-2-(\eta-\alpha)/\zeta}$ for $\epsilon < e^{-1}$, gives

$$C^{ML} \leq c_8 \epsilon^{-2},$$

where

$$c_8 = 2(c_3 + d_3)c_4(\sqrt{2}d_4)^{(\eta-\alpha)/\zeta}M^{\zeta-\alpha}\left(1 - M^{-(\alpha-\eta)/2}\right)^{-2} + (c_3 + d_3)\frac{M^{2\eta}}{M^{\eta} - 1}(\sqrt{2}d_4)^{1/\zeta}.$$

This concludes the case $\alpha > \eta$ as well as the whole proof.

□

Chapter 4

Numerical experiments of the Multilevel RTO method

This chapter mainly discusses two direct applications of the multilevel RTO method in Chapter 3. The first is an ODE, prey and predator model, whose parameter has a fixed dimension. The other is a PDE model where the parameter dimension increases when level increases.

4.1 Prey and predator model

The prey and predator equations, or Lotka—Volterra equations are a pair of first-order nonlinear differential equations, which build a classical model in dynamics of biological systems. This section will first briefly explain the problem set-up of the prey and predator model, the numerical solver that has been used, and then show the convergence results of the multilevel RTO method. Finally, we verify the multilevel complexity theorem by showing the MSE plot.

4.1.1 Model equations

These equations follow the form of section (2.1.1). We consider the following prey (P_1) and predator (P_2) model:

$$\begin{aligned} \text{Prey:} \quad & \frac{dP_1}{dt} = rP_1(1 - \frac{P_1}{K}) - s\frac{P_1P_2}{a + P_1}, \\ \text{Predator:} \quad & \frac{dP_2}{dt} = u\frac{P_1P_2}{a + P_1} - vP_2, \end{aligned}$$

where P_1 is the number of prey (like rabbits) and P_2 is the number of predators (like foxes); t stands for time, and therefore $\frac{dP_1}{dt}$ and $\frac{dP_2}{dt}$ represent the growth rate of populations of prey and predator; $\theta = (r, K, s, a, u, v, P_{1_0}, P_{2_0})^T$ are parameters describing the process. The parameter vector of this model has a fixed dimension. This model belongs to the same categories of model we discussed in section (2.1.1):

$$\frac{\partial P}{\partial t} = f(P, t; \theta),$$

where $P = (P_1,\ P_2)$ and

$$f(P, t; \theta) = \begin{pmatrix} rP_1(1 - \frac{P_1}{K}) - s\frac{P_1P_2}{a+P_1} \\ u\frac{P_1P_2}{a+P_1} - vP_2 \end{pmatrix}.$$

Suppose we are taking observations at some discrete times t_i. The observation $y = (y_1,\ y_2,\ ...,\ y_m)^T$ comes from synthetic data

$$y_i = P(t_i) + \epsilon_i$$

where ϵ_i are from standard Gaussian. We need to notice that y_i here is a 2 by 1 vector since $P(t_i)$ is a vector capturing the numbers of prey and predators.

We are going to derive the derivative of the ODE. We rewrite the prey and predator as

$$\frac{\partial P}{\partial t} = f(y; \theta), \quad \text{with} \quad y(t = 0) \equiv y_0 = (P_{1_0}, P_{2_0})^T.$$

The Jacobians of the right hand side, i.e., function f are

$$J_y(y, \theta) = \begin{pmatrix} r - 2rP_1/K - saP_2/(a + P_1)^2 & -sP_1/(a + P_1) \\ auP_2/(a + P_1)^2 & uP_1/(a + P_1) - v \end{pmatrix},$$

and $J_\theta(y, \theta) =$

$$\begin{pmatrix} 0 & 0 & P_1(1 - P_1/K) & r(P_1/K)^2 & -P_1P_2/(a + P_1) & sP_1P_2/(a + P_1)^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -uP_1P_2/(a + P_1)^2 & P_1P_2/(a + P_1) & -P_2 \end{pmatrix}.$$

The ODE problem we want to solve here is to find y and $A(t_i, \theta)$ at every t_i, $i \neq 0$, such

that

$$\frac{\partial}{\partial t}\left[y \mid A\right] = \left[f(y;\theta) \mid J_y\left(y,\theta\right)A + J_\theta\left(y,\theta\right)\right]$$

with the initial condition

$$\left[y(t=0) \mid A(t=0)\right] = \left[y_0 \mid 0 \mid I\right]. \tag{4.1}$$

Assuming we know the true parameters

$$\theta_{\text{true}} = (49.8, 5, 0.56, 100, 1.16, 25, 0.51, 0.3)^T,$$

we can generate the measurement of P_1 and P_2 at time 10, 15, 20, 25, 30 and 35. All the parameters here should be positive, and it is even better if they are bounded in some ranges. We use the error function to achieve that. We assume that the parameter θ has a prior such that

$$\theta \equiv \theta(\lambda) = \frac{1}{2}(b_{\text{upper}} - b_{\text{lower}})\left(1 + \text{erf}\left(\lambda/\sqrt{2}\right)\right) + b_{\text{lower}}. \tag{4.2}$$

where erf stands for the error function, λ is just any real value, and b_{upper}, b_{lower} stand for upper bound and lower bound of θ. We can control the values of these bounds. In this formation, θ is positive no matter what the value of λ is, and also always falls between the bounds. When solving this ODE, we are actually searching for the answer for λ instead of θ since it is easier to control the value of θ. In this case, we need to use the chain rule to update the ODE problem with respect to λ. We only need to change A here:

$$\begin{aligned} A(t_i, \theta(\lambda)) &= A(t_i, \theta)\text{diag}\left(\frac{\partial\theta}{\partial\lambda}\right) \\ &= A(t_i, \theta)\text{diag}\left(e^{-\lambda^2/2}\left(b_{\text{upper}} - b_{\text{lower}}\right)\right)\frac{1}{\sqrt{2\pi}}, \end{aligned}$$

The synthetic data comes from adding zero-mean i.i.d. standard Gaussian noise. Therefore the discretized likelihood function is:

$$L_h(y|\theta) = (2\pi)^{-\frac{m}{2}}|\det(\Gamma_{\text{obs}})|^{-\frac{1}{2}}\exp\big(-\frac{1}{2}(F_h(\theta) - y)^T\Gamma_{\text{obs}}^{-1}(\ F_h(\theta) - y)\big), \tag{4.3}$$

where in the current scenario, $n = 8$, $\Gamma_{\text{obs}} = I_8$. Since we have observations at 6 times (5, 10, 15, 20, 25, 30, 35), the data y has the total length of $2 \times 6 = 12$. $F_h(\theta)$ is the number

$\mathbb{V}(\widehat{\Delta Y}_l)$	23.59	1.73	0.19	0.015	0.001
C_l(cost)	21.48	26.28	35.41	53.19	86.44

Table 4.1: Variance and cost in equation (3.104) for 3000 samples of the prey model under the RTO method.

Level	N_0	N_1	N_2	N_3	N_4
Sample size	10,000	2451	708	161	33

Table 4.2: Proportional sample sizes for different levels of the prey and predator model given 10, 000 samples at zeroth level.)

of P_1 and P_2 at these observation times, which come from the solution of ODE.

The posterior under standard Gaussian prior is

$$\begin{aligned}\pi_h &= L_h(y|\theta)p(\theta)\\ &= L_h(y|\theta)(2\pi)^{-\frac{n}{2}}\exp\left(-\frac{1}{2}\theta^T\theta\right).\end{aligned}$$

The following contour Figure (4.1) is the density of 3000 RTO samples of all eight parameters. The other contour Figure (4.2) is the weighted RTO samples, which is the posterior. The effective sample ratio (ESR) is about 70%.

Multilevel RTO

The quantity of interest here is the number of prey at a future time, 50. We will follow the steps in (3) to estimate the number of prey at time 50. We are not sure about the largest level L here, nor the sample size at each level, and therefore we first run some tests on a small sample size at each level. Based on (3.104), we should obtain the variance of each $\widehat{\Delta Y}_l$ and the corresponding cost (computational time here), C_l. At this test stage, we assign 3000 samples to $L = 4$ levels. In summary, we run the algorithm in (3) with $N_l = 3000$ for all l, and $L = 4$. We record the variance of each $\widehat{\Delta Y}_l$ and corresponding C_l as Table (4.1). Table (4.1) shows the computational cost and variance in equation (3.104), where the fixed ratio of N_l can be calculated as shown in Table (4.2). We can see that the majority of samples are on the first level, and only very few samples are required in the finest level.

We need to notice that these in Table (4.2) are the ratios of sample size given the 10, 000 samples at zeroth level, not the exact sample sizes we use. The question rises as to

the exact sample size we should use. There is a trade-off in this selection: in order to minimize the MSE, we need to use more samples; in order to save computational cost, we need to use fewer samples. The trade-off reaches a balance at the point, when using more samples does not minimize the MSE as much as before. This point appears when the variance in the MSE equals the square of the discretization error. This is because that the discretization error is not affected by sample size, but only by the discretization size h_l. When increasing the sample size, only the variance part of the MSE is decreasing. When the variance is decreased to a number that is smaller than the square of the discretization error, the discretization error term dominates the MSE. After passing this point, we cannot improve the MSE much as before by increasing the sample size. Therefore, we want the sample size to reach a point so that in the approximation of MSE:

$$\mathbb{E}(\widehat{Q}_{\mathbf{2}} - \mathbb{I}_Q)^2 \approx 2\left(\frac{1}{\mathbb{E}^2(\hat{z})}\big(\mathbb{V}(\widehat{Y}_L) + \Upsilon^2(\boldsymbol{\theta})\mathbb{V}(\hat{z}_L)\big)\right) + 2\left(\mathbb{I}_{Q_L} - \mathbb{I}_Q\right)^2, \tag{4.4}$$

we have

$$\frac{1}{\mathbb{E}^2(\hat{z})}\left(\mathbb{V}(\widehat{Y}_L) + \Upsilon^2(\boldsymbol{\theta})\mathbb{V}(\hat{z}_L)\right) = \left(\mathbb{I}_{Q_L} - \mathbb{I}_Q\right)^2. \tag{4.5}$$

Note that this right-hand side is determined by the largest level L, which we also call the stopping level. Hence, in every stopping level, there is one optimal sample size for it.

We have been able to obtain the convergence plots based on this small sample size test. Recall that there are three conditions for the complexity theorem: the discretization error, the variance of differences and the cost of differences. All three conditions should apply to both $\widehat{\Delta Y}_l$ and $\widehat{\Delta\beta}_l$. We expect these three terms to be bounded to the power of h_l, e.g., h_l^{ζ} for the discretization error. In other words, if we plot the logarithm of the discretization error, it is expected to decrease in a straight line when the level goes up. If that were this case, it guarantees that the discretization error is diminishing, which should be the case if we discretize the model properly. In practice, we are very concerned about the second condition and the third condition: that if the variance is decreasing at a higher rate than the increasing rate of the cost. In total we will show two sets of convergence plots: discretization error and variance convergence plots on $\widehat{\Delta Y}_l$ and $\widehat{\Delta\beta}_l$.

4.1.2 Numerical solver

Depending on the different numerical solvers, the rate of convergence changes. In this prey and predator problem, we need to control the discretization size of the solver, and therefore we need to program a new solver. We first started with Euler's method, which uses the

following equation to update the solution to this ODE:

$$y_{j+1}(t_j + h_l) = y_j(t_j) + h_l \frac{\partial}{\partial t} [y \mid A] \mid_{y=y_j(t_j)}. \tag{4.6}$$

The step size h_l is the discretization size. Data used in this problem is the numbers of prey and predators at times 5, 10, 15, 20, 25, 30 and 35. But the step size, or the discretization size, is not the same as 5, the gap between data. The step size is smaller than the gap, and decreases when level increases. Euler's method is a first order method, which means that the difference between $y_{j+1}(t_j + h_l)$ and the exact solution at time $t_{j+1} = t_j + h_l$, the local truncation error, is approximately proportional to h_l^2.

Euler's method is the simplest Runge–Kutta method. We use the RK2 method which updates

$$y_{j+1}(t_j + h_l) = y_j(t_j) + h_l k_2, \tag{4.7}$$

where

$$k_1 = \frac{\partial}{\partial t} [y \mid A] \mid_{y=y_j(t_j)}$$
$$k_2 = \frac{\partial}{\partial t} [y \mid A] \mid_{y=y_j(t_j)+0.5h_l k_1}.$$

The method has a local truncation error approximated proportional to h_l^3. The convergence rates, ζ and α, are different for Euler's method and the RK2 method. The ζ of RK2 is twice that of Euler's method, and the α of RK2 is also twice that of Euler's method.

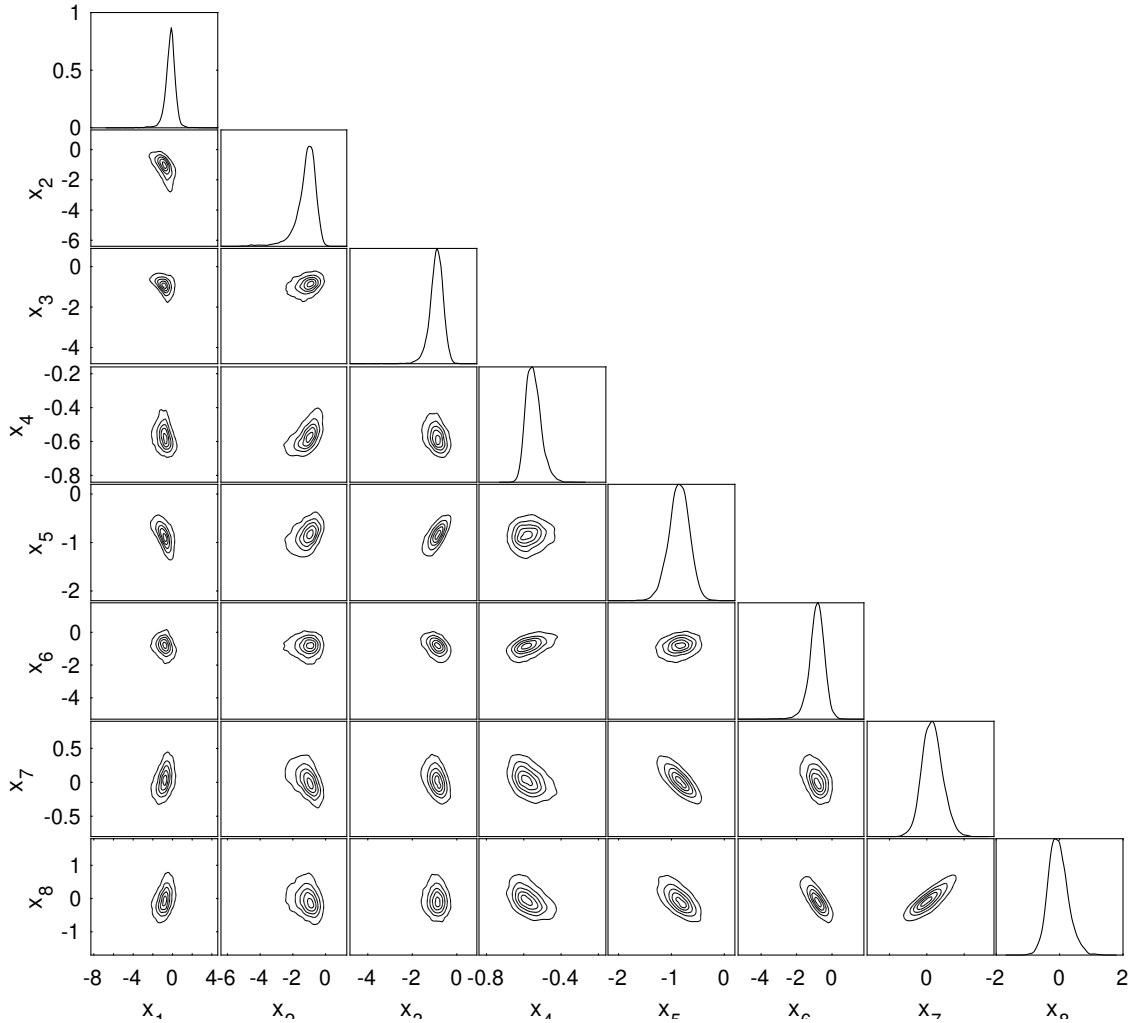

Figure 4.1: Unnormalized posterior for 8 parameters of prey and predator model. The figure is simply the RTO density of an 8-dimension parameter vector.

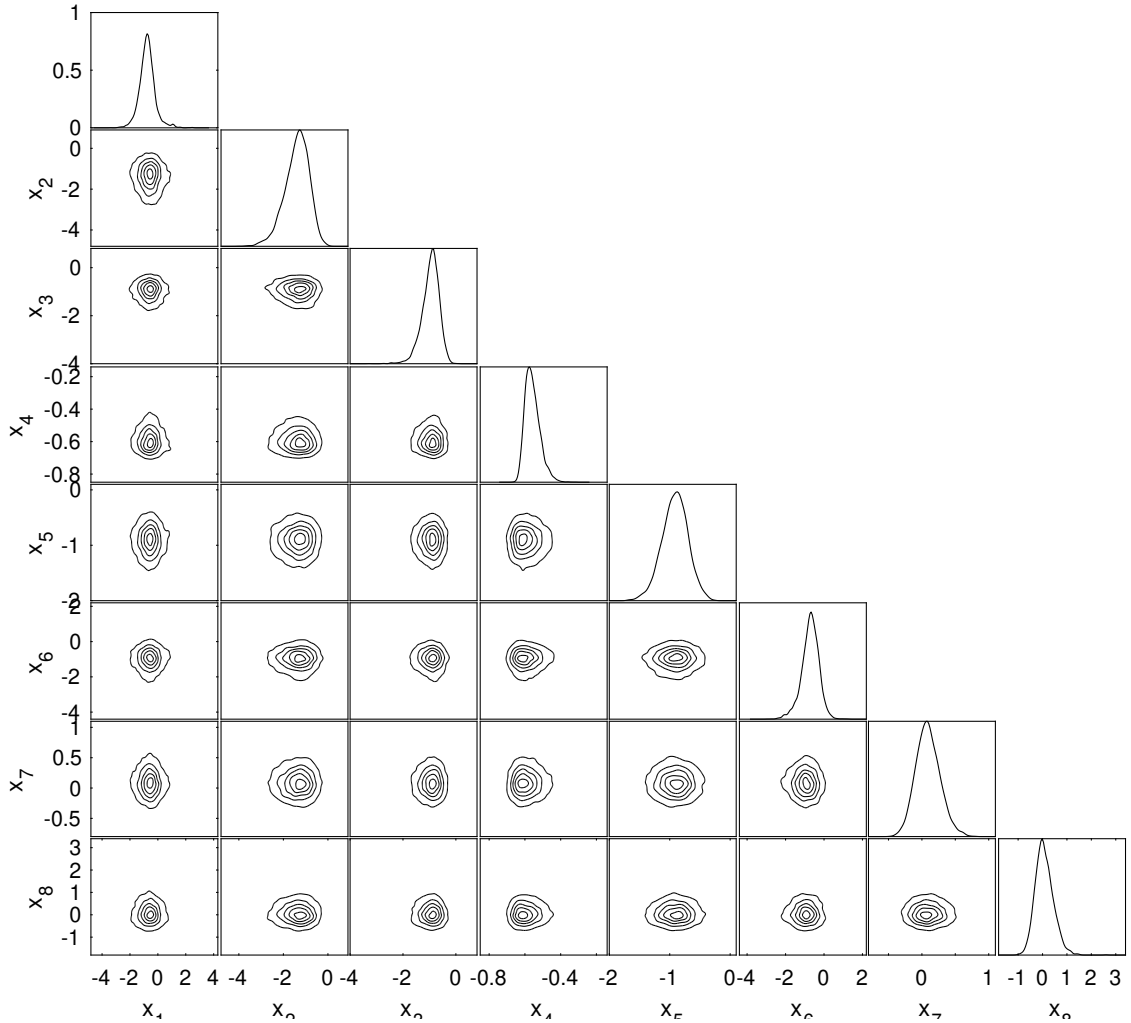

Figure 4.2: Posterior of 8 parameters of prey and predator model. The samples here have been weighted by their corresponding weights.

4.1.3 Convergence rates

We are going to discuss the convergence plots we obtained from the small dataset, with 3000 samples at every level. These are not the samples that we actually use to calculate the quantity of interest. We conduct this to examine the convergence rates in the complexity theorem and evaluate the performance of the multilevel estimator.

Figure (4.3) is the convergence plot reflecting the discretization error. The blue flat line is the mean of $\mathbb{E}|\hat{\beta}_l\widehat{Q}_l|$ in the numerator of the ratio estimator $\widehat{Y} = \sum_{l=0}^{L}\widehat{\Delta Y}_l$. The red oblique line is the mean of the difference, $\mathbb{E}|\hat{\beta}_l\widehat{Q}_l - \hat{\beta}_{l-1}\widehat{Q}_{l-1}|$ for the numerator. Condition 1 of complexity theorem (3.4.1) is:

$$\mathbb{E}|\hat{\beta}_l\widehat{Q}_l - \hat{\beta}_{l-1}\widehat{Q}_{l-1}| \leq c_1 h_l.$$

The y-axis here is in log2 scale and $h_l \propto 2^{-l}$. Therefore, the slope of the red line is the negative of the rate ζ in complexity theorem (3.4.1). The slope of the red line is about -2 here, i.e., $\zeta = 2$. If the gap between the flat line and oblique line is small, it is a sign that the current multilevel setting is not a good option. If the red line crosses the blue line, in other words, the difference $\mathbb{E}|\hat{\beta}_l\widehat{Q}_l - \hat{\beta}_{l-1}\widehat{Q}_{l-1}|$ is even larger than $\mathbb{E}|\hat{\beta}_l\widehat{Q}_l|$ itself, then discretization size at the coarse level is too large that the quantity of interest is inaccurate. It indicates that the current multilevel setting is not good. This happens in the numerical results in Chapter 5.

The Figure (4.4) on variance is what we are mostly concerned about. One can see that this figure is quite similar to the Figure (4.3): one flat line indicating the variance of $\hat{\beta}_l\widehat{Q}_l$ and the oblique line showing the variance of differences. The slope here is the negative of rate α in the complexity theorem (3.4.1). The slope is about -4 here, i.e., $\alpha = 4$. We can see the decreasing rate of this variance figure the red line is about twice that of the red line in Figure (4.3), which means that α is approximately twice of ζ.

We observe that these two lines do not cross. If they crossed, which happened in other samplers we use later (Chapter 5), it means that the variance of difference of consecutive terms is even larger than the variance of the terms themselves, which indicates that the variance of the telescope sum is larger than the sum of the variance of individual terms. If we calculated the sample size using (3.104), N_1 is even larger than N_0. In this case, the multilevel setting is not ideal. We can increase the discretization size of the zero level to try again.

Figures (4.6) and (4.5) are convergence figures for the mean and variance for $\hat{\beta}_l$, which constitutes the multilevel estimator of β, i.e., the ratio between the unnormalized posterior and the density of RTO from levels 0 to 4. These plots are quite similar to the convergence plots on $\hat{\beta}_l\widehat{Q}_l$. The slopes of the oblique lines of mean and variance are the same respectively.

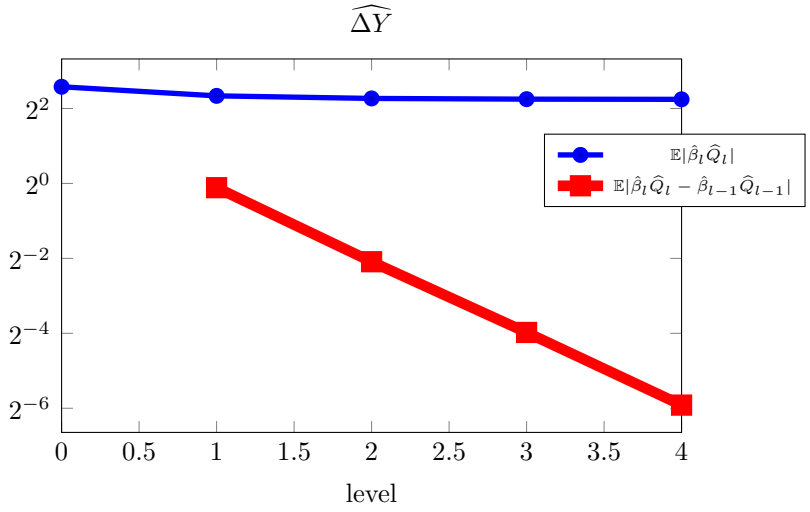

Figure 4.3: Convergence plot of mean for prey model with RTO method. The blue line is $\widehat{\Delta Y}$, the mean of the numerator in the multilevel self-normalizing estimator (3.23). The red line is the mean of the difference, corresponding to condition 1 of the complexity theorem (3.4.1). The slope of the red line, -2, is the negative of the rate ζ.

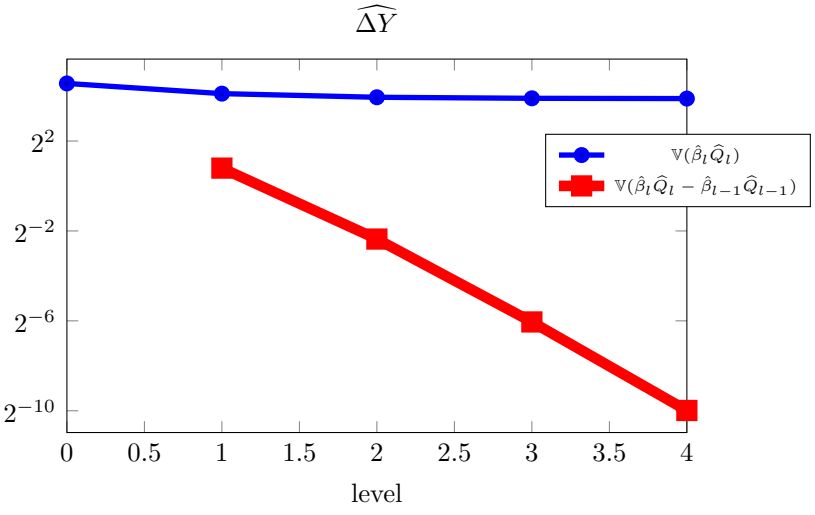

Figure 4.4: Convergence plot of variance for prey model with RTO method. The plot is checking condition 2 of the complexity theorem (3.4.1), and the negative of slope of the red line is denoted as α in the theorem. We can see that the slope of the red line here, -4, is twice that of the slope of the red line in the mean plot. This vaguely shows that $2\zeta \approx \alpha$.

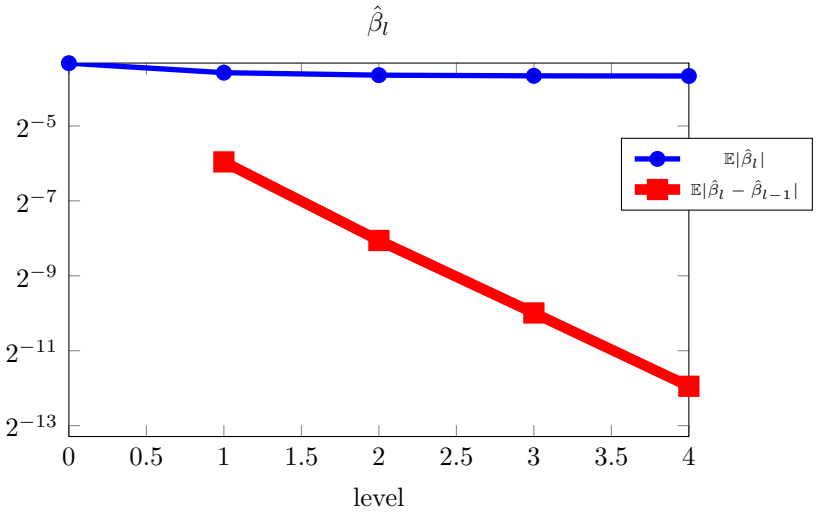

Figure 4.5: The mean plot of unnormalized weight. The slope of red line is approximately the same as Figure (4.3).

For the other samplers we have used, the slopes of the oblique lines of mean and variance are also the same respectively. Therefore, we assume that the convergence rates of $\hat{\beta}_l$ and $\hat{\beta}_l\widehat{Q}_l$ are identical in the complexity theorem. The analysis of convergence plots on $\hat{\beta}_l\widehat{Q}_l$ applies on $\hat{\beta}_l$.

The blue curve in Figure (4.7) is the CPU time across levels with MAP points as initial values to start the optimization. This figure demonstrates condition 3 of the complexity theorem (3.4.1). The computational cost C_l increases as discretization size h_l increases. The slope of this blue line is $-\eta$. We can tell that η is about 1 here. In this scenario, $\alpha > \eta$, which is the first case of the conclusion of the complexity theorem (3.4.1). We expect that the cost of the multilevel estimator is proportional to reciprocal MSE, i.e., $C^{ML} = O(\epsilon^{-2})$. We will verify the relationship between cost and MSE in the following subsection.

In the ratio estimator, we have two multilevel estimators: the numerator $\widehat{\Delta Y}$ and the denominator $\hat{z}$. At the moment, we have verified that $\widehat{\Delta Y}$, $\hat{z}$ and C_l converge as the conditions stated in the complexity theorem (3.4.1). We are going to verify if the cost of multilevel estimator was as shown in complexity theorem (3.4.1) given all these three conditions. It is verified by the MSE plots in subsection 4.1.4.

4.1.4 Optimal sample size and MSE

Before diving into details of the CPU time versus MSE figure, there is a trick to reduce the CPU time. For each level, we need to do the optimization with an initial value. For level

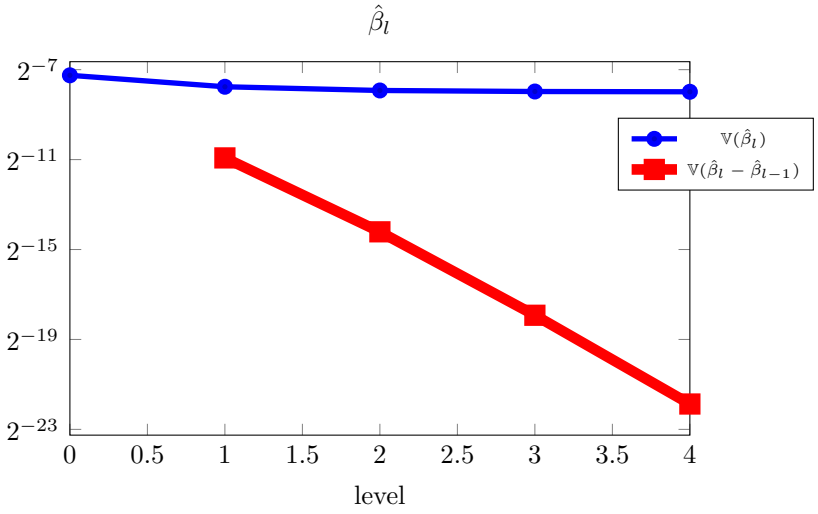

Figure 4.6: Convergence plots for variance of unnormalized weight. The weight here is actually the denominator of multilevel estimator (3.23). Even though the exact values of $\widehat{\Delta Y}$ and $\widehat{\beta}$ are different, the rates of convergence of the mean and variance are the same respectively. The slope of the red line is identical to those in Figure (4.4).

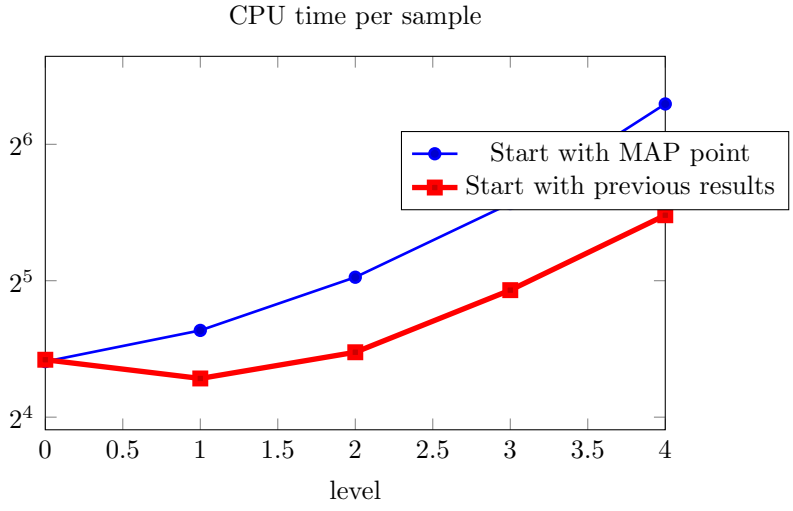

Figure 4.7: CPU time comparison. The plot is to compare the different CPU time of opimizations starting with MAP point and starting with the previous samples obtained at the last level. The increasing rate of the blue curve here corresponds to the rate η in the complexity theorem (3.4.1).

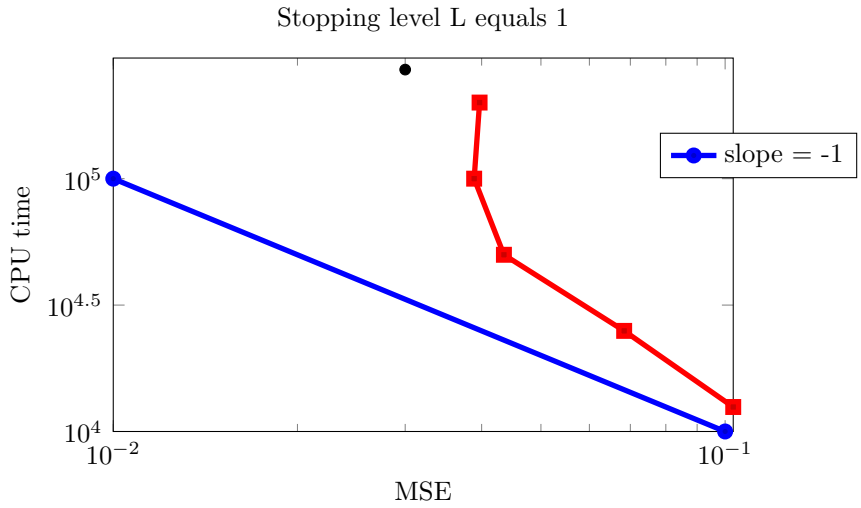

Figure 4.8: CPU time versus MSE for stopping level 1, i.e., including level 0, 1. The blue line is just a straight line with slope of -1. The red line is the actual computational cost versus the mean square error. We can see the red line goes up vertically after one point. This indicates that after this point, greater cost fails to reduce the MSE as quickly as before. Therefore, this point corresponds to the optimal sample size. The black dot is the optimal sample size calculated from equation (4.5) derived from complexity theorem.

zero, we start with the MAP point to optimize. For the level one, we can choose the samples from level zero as the initial guess to optimize, or just use the MAP point again. Figure (4.7) shows the cost of both methods. Figure (4.7) shows a comparison of the cost when optimization starts with a result from the previous level, and when optimization starts with the MAP point. We can see that the cost of the first method is cheaper than the second. If the dimension of the parameters changes at every levels, we might not be able to use the result from previous level as an initial guess.

The following CPU time versus MSE figures (4.8 and 4.9) verify the complexity theorem in Chapter 3. The first figure (4.8) is for stopping level $L = 1$, which means that there are only two levels of samples: level zero and level one. The second figure (4.9) is for stopping level $L = 2$, including three levels of samples: level zero, level one and level two.

The red lines in figures (4.8) and (4.9) are from numerical experiments. The blue lines serve as a reference to check if some segments of the red lines has the same slope as the reference line. We can see that when the MSE is large (the bottom right corner of the plots), the cost versus MSE has a slope close to -1. This is because the MSE is dominated by the variance instead of discretization error at this time. The variance decreases linearly as sample size goes up, which is the property of Monte Carlo simulation. Figure (4.10)

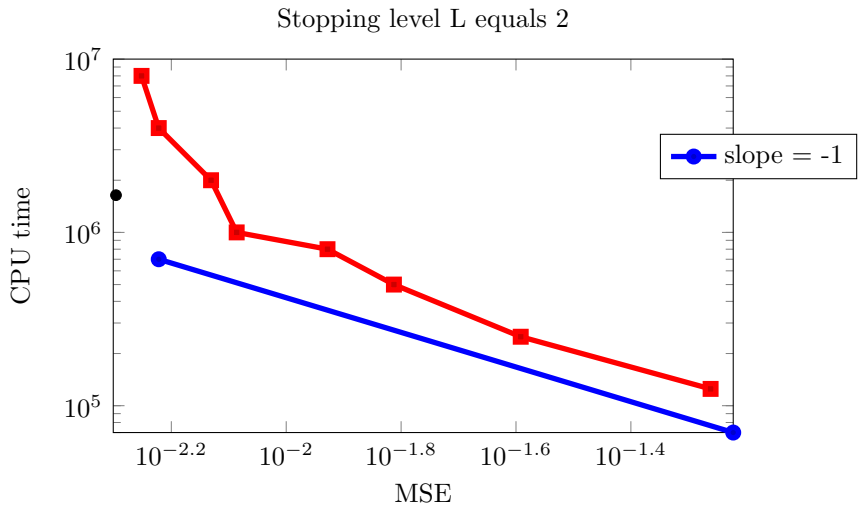

Figure 4.9: CPU time versus MSE for stopping level 2. This plot is similar to Figure 4.8, but for stopping level 2. Stopping level 2 includes level 0, 1, 2. In this figure, the actual cost (red line) does go up as it did in Figure 4.9. The black dot is the predicted "changing point" corresponding to the optimal sample size. We can also observe that the line of the five points starting from the right-hand side, has slope of -1. This indicates that it is in the best scenario $O(\epsilon^{-2})$ of complexity theorem (3.4.1).

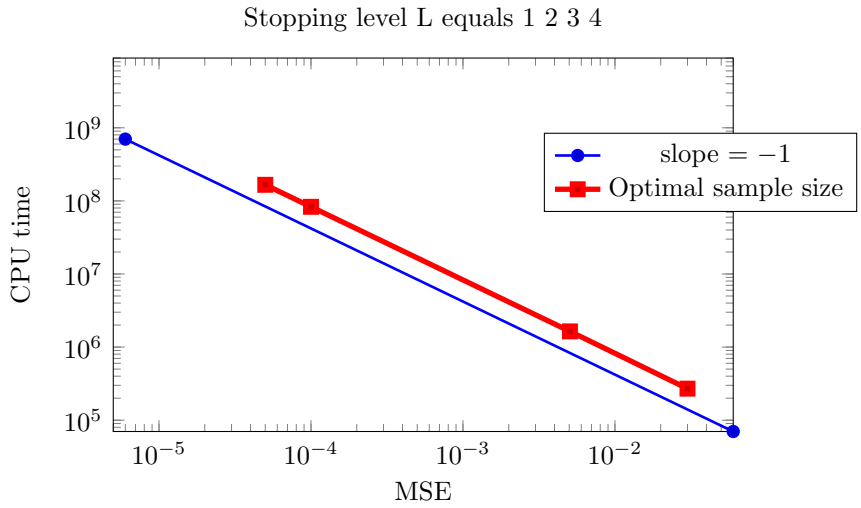

Figure 4.10: Optimal sample sizes of difference stopping level for prey model. This figure shows all the predicted "changing points" corresponding to optimal sample size, for stopping level 1, 2, 3, and 4. We can see that the line of these four dots has slope -1 again. The points for stopping level 1 and 2 are actually shown as black dots in Figure (4.8) and Figure (4.9) respectively.

shows the optimal sample size of four stopping levels versus MSE. These optimal sample sizes also line up and is parallel to the line with slope -1.

The sample size of the multilevel level is not determined yet even though we have the scales of sample sizes at each level given a fixed N_0. We need to decide what is the best sample size, or equivalently best N_0 since the scales of sample sizes are fixed. The optimal sample size occurs when the variance part of MSE equals the discretization error. However, we cannot exactly calculate the discretization error. The way to find this optimal sample size is to construct the CPU time versus MSE figure and observe its curve. When it reaches a certain point, the CPU time increases, while the MSE decreases more slowly than before. This point corresponds to the optimal sample size we are looking for, and it is the "changing point". However, there is a high cost to find this point. A cheaper solution we provide here is to calculate the sample size to reach the variance bound in equation (4.5), while the optimal sample size should occur around this calculated sample size. In the case of stopping level 1 in Figure (4.8), the changing point is about the middle point; for stopping level 2 in Figure (4.9), the changing point is the forth highest point.

Estimation of MSE

There are two black points in these two figures (4.8 and 4.9). These are the two approximate "changing points".

We first approximate the discretization error by using the finest level, 4, we have obtained. We approximate the left-hand side in equation (4.5) by these small size samples $N_l = 3000$ for all levels. For stopping level $L = 1$, the discretization error is 0.1737. Square $0.1737^2 = 0.03017$. The left-hand side of equation (4.5) is 8226. In order to make equation (4.5) balanced, $N_0 = 8226/0.03017 = 2.7e5$. We can see that this black point (predicted "changing point") in Figure (4.8 is smaller than the MSE at the actual "changing point" (the third red dot in the plot), which requires more samples than the actual "changing point". For stopping level $L = 2$, the square of the discretization error is $0.0711^2 = 0.00500$. The left-hand side of equation (4.5) is 8288. Therefore the N_0 required is $8288/0.005 = 1.6e6$. The black dot (predicted "changing point") in Figure 4.9) is again on the left-hand side of the actual "changing point", which demands more samples than the changing point. Hence, these two black dots, $2.7e5$ and $1.6e6$, serve as the upper bounds of the actual optimal sample sizes. We can select $2.7e5$ samples for N_0 for stopping level 1, and $1.6e6$ samples for stopping level 2. If we need to find the optimal sample size, or simply want to find the sample size that is close to the optimal sample size, equation (4.5) can provide a route as shown in this paragraph.

The second line of the following equation which including the covariance is a more accurate approximation of the variance of $\widehat{Q}_{\boldsymbol{2}}$ than that in equation (4.5):

$$\begin{aligned}
\mathbb{V}(\frac{\widehat{Y}_L}{\hat{z}_L}) &\approx \frac{1}{N_0}\frac{1}{\mathbb{E}^2(z_L)}\mathbb{V}\left(Y_L - z_L\frac{\mathbb{E}(Y_L)}{\mathbb{E}(z_L)}\right) \\
&= \frac{1}{N_0}\frac{1}{\mathbb{E}^2(z_L)}\left(\mathbb{V}\left(Y_L\right) + \Upsilon\left(\boldsymbol{\theta}\right)^2\mathbb{V}\left(z_L\right) - 2\Upsilon\left(\boldsymbol{\theta}\right)\mathrm{Cov}\left(Y_L,\ z_L\right)\right); \\
\left(\mathbb{I}_{Q_L} - \mathbb{I}_Q\right)^2 &= \frac{1}{N_0}\frac{1}{\mathbb{E}^2(z_L)}\left(\mathbb{V}\left(Y_L\right) + \Upsilon\left(\boldsymbol{\theta}\right)^2\mathbb{V}\left(z_L\right) - 2\Upsilon\left(\boldsymbol{\theta}\right)\mathrm{Cov}\left(Y_L,\ z_L\right)\right).
\end{aligned}$$

The last line above is the same logic as in equation (4.5), making the variance equal the discretization error.

4.2 PDE model

The following subsections will discuss a more complicated model. The application of multilevel RTO will be a bit different from the prey and predator model.

4.2.1 Problem background

We have an elliptic PDE modelling the pressure distribution $r(\boldsymbol{s})$ of a stationary fluid, in a domain $\Omega = [0,1]^2$ with boundary $\partial\Omega$. The fluid flows in a permeability field $k(\boldsymbol{s})$, where $\boldsymbol{s} \in \Omega$ represents the spatial coordinates. We also denote $\mathbf{n}(\boldsymbol{s})$ as the outward normal vector along the boundary. The PDE has the form

$$-\nabla \cdot (k(\boldsymbol{s})\nabla r(\boldsymbol{s})) = 0,$$

where $\boldsymbol{s} \in \Omega \equiv [0,1]^2$.

In the setting, we want to fetch the permeability field $k(\boldsymbol{s})$ when we obtain some observations of pressure $r(\boldsymbol{s})$. We make some assumptions about the prior before generating the posterior. We assume that $k(\boldsymbol{s})$ has a log-normal prior, i.e., $k(\boldsymbol{s}) = \exp(x(\boldsymbol{s}))$ where $x(\boldsymbol{s})$ is a Gaussian process defined by the exponential kernel $k(\boldsymbol{s}, \boldsymbol{s}') = \exp(-5|\boldsymbol{s} - \boldsymbol{s}'|)$. x is the parameter we are trying to retrieve, while it is in infinite-dimension space. Therefore, for a realization $x(\boldsymbol{s})$, the pressure has the PDE

$$-\nabla \cdot \left(e^{x(\boldsymbol{s})}\nabla r(\boldsymbol{s})\right) = 0, \tag{4.8}$$

with boundary conditions:

$$\begin{cases} r(\boldsymbol{s}) = 0, & \boldsymbol{s} \in \partial\Omega_{\text{left}}; \\ r(\boldsymbol{s}) = 1, & \boldsymbol{s} \in \partial\Omega_{\text{right}}; \\ e^{x(\boldsymbol{s})}\nabla r(\boldsymbol{s}) \cdot \ \mathbf{n}(\boldsymbol{s}) = 0, & \boldsymbol{s} \in \{\partial\Omega_{\text{top}}, \partial\Omega_{\text{bottom}}\}. \end{cases} \tag{4.9}$$

The equations above (4.9) suggests that in the square domain $\Omega = [0,1]^2$, the left and right boundaries have Dirichlet boundary conditions, while the top and bottom have Neumann boundary conditions.

The quantity of interest Q here is the outflow through the left boundary:

$$Q(x) = -\int_0^1 e^{x(\boldsymbol{s})} \frac{\partial r(\boldsymbol{s})}{\partial s_1}|_{s_1=0} ds_2 = -\int e^{x(\boldsymbol{s})}\nabla r(\boldsymbol{s}) \cdot \nabla \phi'(\boldsymbol{s}) d\boldsymbol{s} \tag{4.10}$$

where $\boldsymbol{s} = (s_1, s_2)^T$, and $\phi'(\boldsymbol{s})$ is a linear function on the left and right boundaries, $\partial\Omega_{\text{left}}$ and $\partial\Omega_{\text{right}}$.

This is a more complicated model than the ODE model. The setting above is in the infinite-dimension space, and we need to discretize the model. The observations, y, of the pressure field are collected from 71 sensors in the permeability field. Therefore $y \in \mathbb{R}^{71}$. For

a discrete model, we denote $\boldsymbol{x}$ as the discretized version of x. Thence, the forward model is

$$y = F_h(\boldsymbol{x}_h) + \epsilon, \tag{4.11}$$

where y represents the pressure measurements collected from 71 sensors and F_h is the discretized forward model where h is the discretization size. $\boldsymbol{x}_h$ is the discretized version of x in (4.8). We have the same posterior as in (2.16). We need to arrange the prior distribution of the infinite-dimension parameter x before discretizing it.

Convolution process and finite element method

The prior of the parameter x is a Gaussian process convolving a continuous white noise process $X(\boldsymbol{s})$, $\boldsymbol{s} \in \Omega$, with a Gaussian kernel $k(\boldsymbol{s})$ so that for $\boldsymbol{s} \in \Omega$,

$$x(\boldsymbol{s}) = \int_\Omega k(\boldsymbol{u} - \boldsymbol{s}) X(\boldsymbol{s}) d\boldsymbol{u}. \tag{4.12}$$

One reason for this convolution is to make the covariance function of the Gaussian process depend only on the distance, or displacement, of two points regardless of their location. We suppose the covariance of the displacement for $\boldsymbol{d} = \boldsymbol{s} - \boldsymbol{s}'$ is

$$\mathbb{V}(\boldsymbol{d}) = \mathrm{Cov}\left(x(\boldsymbol{s}), x(\boldsymbol{s}')\right) = \int_\Omega k(\boldsymbol{u} - \boldsymbol{s}) k(\boldsymbol{u} - \boldsymbol{s}') d\boldsymbol{u} \tag{4.13}$$

$$= \int_\Omega k(\boldsymbol{u} - \boldsymbol{d}) k(\boldsymbol{u}) d\boldsymbol{u}. \tag{4.14}$$

In this specific discretized model, we pick five sites, $\boldsymbol{\omega} = (0.1, 0.3, 0.5, 0.7, 0.9)$ at each dimension of $\Omega = [0,1]^2$, and then we have $5 \times 5 = 25$ grids in total. We form the latent process $X(\boldsymbol{s})$ as $X = (X_1, X_2, X_3, X_4, X_5)^T$ at these 25 grids. The resulting continuous Gaussian process is then

$$x(\boldsymbol{s}) = \sum_{j=1}^{m} X_j k(\boldsymbol{s} - \boldsymbol{\omega}_j). \tag{4.15}$$

Apart from the fixed grids above, we need to discuss how to discretize infinite-dimension parameter x. This is the mesh we can change at every level. The permeability field $k(\boldsymbol{s}) \equiv \exp(x(\boldsymbol{s}))$ spreads around the 2D domain $\Omega = [0,1]^2$, and therefore x also takes value in the square domain Ω. Naturally we attempt to discretize the Ω into identical small square grids. For each of the square grids, we assign one parameter. The mesh size is $h_l = \frac{1}{20} 2^{-l}$ for $l = 0, 1, 2, 3$. Therefore, the number of grids for $l = 0, 1, 2, 3$, are 400, 800,

1600 and 25, 600 respectively. Hence, there are $\left(\frac{1}{h_l}+1\right)^2$ nodes for each level, i.e., 441, 1681, 6561 and 25, 921 nodes for $l = 0, 1, 2, 3$ respectively. These are the dimensions of parameter space at each level. We can see that the parameter dimension is very large at the finest level.

We use the finite element method to solve equations (4.8), (4.9) and (4.10). The "element" here is these small square grids of Ω in the discretization. For the basis function in the finite element method, we use a piece-wise bilinear function. Accordingly, the number of "element" increases exponentially with the mesh size. In this situation, it is extremely costly to compute the parameters at fine levels. It takes several hours to generate 100 samples at the fourth level with 25, 600 grids, and therefore we expect the multilevel method to reduce the cost.

4.2.2 Numerical results

This dimension of parameter in this PDE model increases exponentially as the level increases, whereas in the prey and predator model the parameter dimension is fixed. We can now verify the multi-RTO method in a practical and high-dimension setting. Since it is expensive to compute samples at fine levels, we only compute a small sample test with 100 samples at each level, which are on mesh size $\frac{1}{20}$, $\frac{1}{40}$, $\frac{1}{80}$ and $\frac{1}{160}$. The following two figures (4.11 and 4.12) are convergence plots of mean and variance respectively for the quantity $\hat{\beta}_l\widehat{Q}_l$ in this PDE model, where Q_l is the outflow through the left boundary and β_l is the ratio of unnormalized posterior and RTO density.

We can see that the mean plot in Figure(4.11) looks similar to that of prey model (4.3). The difference of mean has a clear tendency to decrease with the increasing level. Figure (4.12) is the variance plot. When observing that the flat line departs from the red line, it is clear that we can apply multilevel method in this problem. If these two lines crossed, we might need to adapt another scenario to construct the telescope sum of the quantity of interest.

Comparing the mean plot in Figure (4.11) and the variance plot in Figure (4.12), the slope of the variance plot is twice that of the mean plot. This indicates that α is approximately twice of ζ , where ζ is the decreasing rate of discretization error when level increases and α is the decreasing rate of variance of difference $|\hat{\beta}_l\widehat{Q}_l - \hat{\beta}_{l-1}\widehat{Q}_{l-1}|$. This result is similar to that of the prey model.

The last two convergence figures (4.13) and (4.14) are the mean and variance plots of the unnormalized weight. They show a similar pattern to the convergence plot of $\hat{\beta}_l\widehat{Q}_l$. The slopes of these two mean plots are the same, and slope of two variance plots are identical. Therefore we can use the complexity theorem, since it is assumed that the convergence

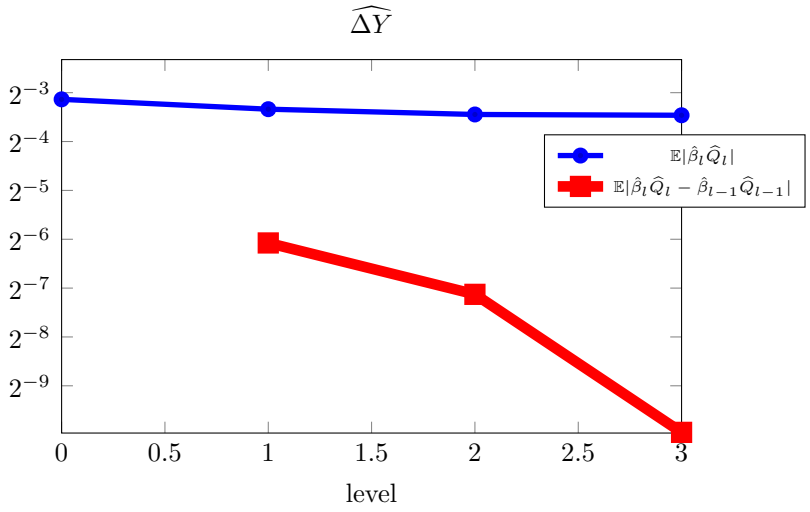

Figure 4.11: Convergence plot for mean in PDE. The slope of the red line responds to the rate ζ in complexity theorem (3.4.1). This plot has a slightly "bent" red line, and a higher variance at the zeroth level, which means the difference between level 0 and 1 is smaller. This is different from the straight red line in the prey and predator model.

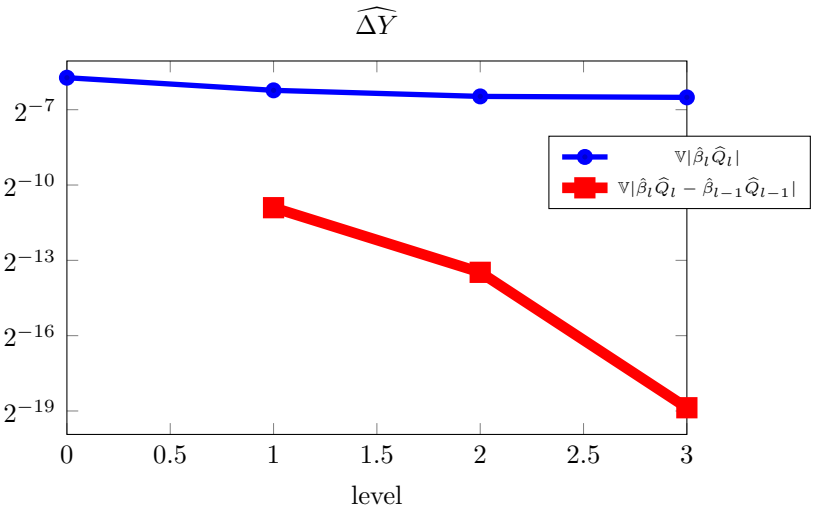

Figure 4.12: Convergence plot for variance in PDE. The slope of red line (variance of difference) declines at a rate about twice that of mean plot. The slope here is proportional to the rate α in the complexity theorem (3.4.1).

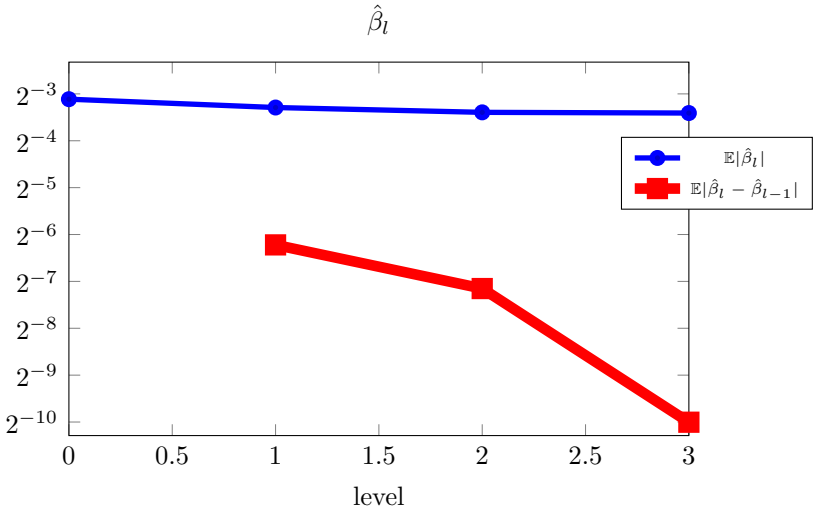

Figure 4.13: The mean plot of weight for PDE model.

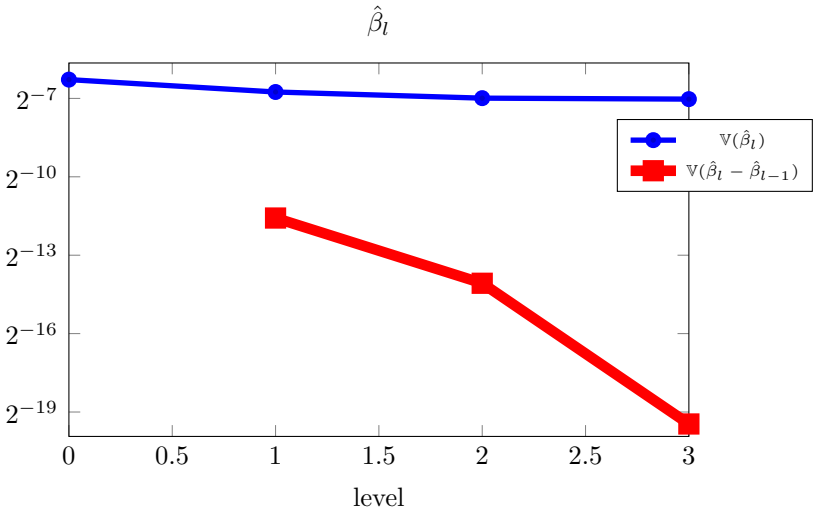

Figure 4.14: Convergence plots for variance of weight. The weight plots have a similar pattern to the $\widehat{\Delta Y}$.

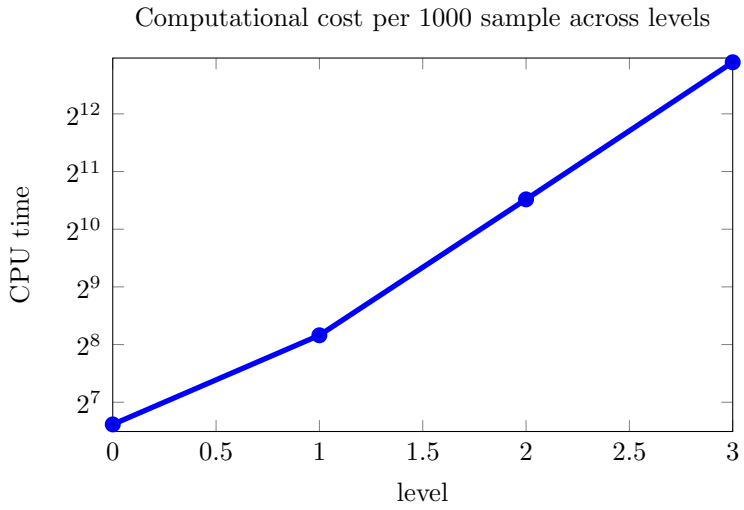

Figure 4.15: This figure plots the cost for the PDE model across levels. The increasing rate here corresponds to the rate η in the complexity theorem (3.4.1).

rates of mean plots, (4.11) and (4.13), are both ζ, and convergence rates of variance plots, (4.12) and (4.14), are both α.

Figure (4.15) shows the CPU time for 1000 samples at each level. The slope here corresponds to the rate η in the complexity theorem. Comparing Figure (4.12) and Figure (4.15), $\alpha > \eta$. Therefore we predict that the cost is proportional to reciprocal MSE, i.e., $C^{ML} = O(\epsilon^{-2})$.

4.2.3 Optimal sample size and MSE

We want to find the optimal sample size as was done in the prey model. We first calculate the ratio of sample sizes as in Table (4.3). We can see from Table (4.3) that the difference between the zeroth level and first level is not as large as it was in the prey model. In this section, we will use the following equation again to find the optimal sample size of each stopping level, i.e., $L = 1$ and 2. We again make the discretization error equal the

Level	N_0	N_1	N_2	N_3
Sample size	10,000	3487	1500	280

Table 4.3: This table includes the proportional sample size across all level for PDE model given 10, 000 samples at zeroth level. It is calculated using (3.104).

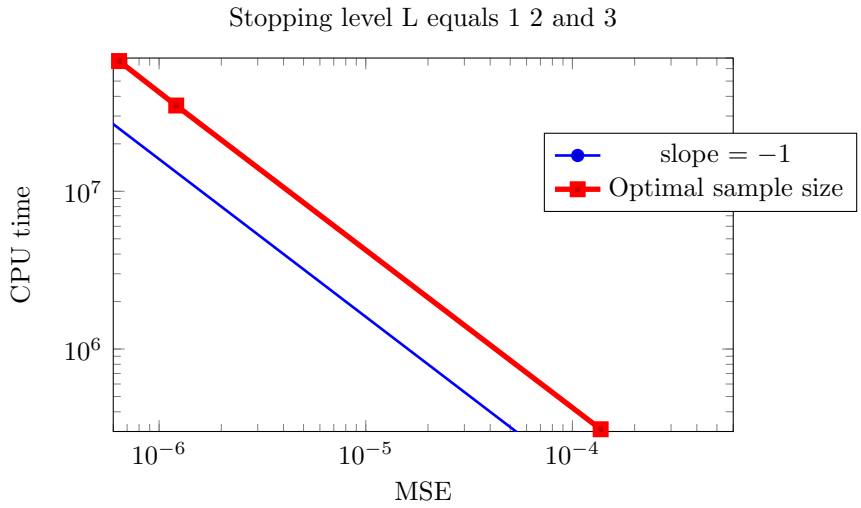

Figure 4.16: PDE model: optimal sample size for different stopping levels. These are the predicted points that corresponds to the predicted optimal sample size and are similar to figure (4.10).

approximation of MSE as follows:

$$(\mathbb{I}_{Q_L} - \mathbb{I}_Q)^2 = \frac{1}{N_0}\frac{1}{\mathbb{E}^2(z_L)}\left(\mathbb{V}(Y_L) + \Upsilon(\boldsymbol{\theta})^2\,\mathbb{V}(z_L) - 2\Upsilon(\boldsymbol{\theta})\,\mathrm{Cov}(Y_L,\ z_L)\right).$$

The left-hand side equals $(0.0117)^2$=0.000137, $(0.0011)^2 = 1.21e-6$, and $(0.0008)^2 = 6.4e-7$. The right-hand side excluding the factor $\frac{1}{N_0}$ equals 42. Therefore the resulting optimal sample size is 3e5, 3e7, and 6e7. We can see that in Figure (4.16), the line of predicted optimal sample size of different stopping levels is parallel to a straight line with slope -1 again. This shows that $C^{ML} = O(\epsilon^{-2})$. This again verifies the case $\alpha > \eta$ in the complexity theorem.

Chapter 5

Implicit Sampling

This chapter introduces another optimization-based sampler similar to RTO, Implicit Sampling. We will depict the target distribution specifically for Implicit Sampling, and describe its algorithm. We apply this new method in the same prey and predator model in Chapter 4.

5.1 Implicit Sampling

5.1.1 Implicit Sampling

We restate the discretized forward model in a Bayesian inverse problem:

$$y = F_h(\boldsymbol{x}_h) + e,$$

where $y \in \mathbb{R}^m$, $\boldsymbol{x}_h \in \mathbb{R}^n$, and e is a Gaussian distribution with zero mean and covariance Γ_{obs}, y is the data, and $F_h : \mathbb{R}^n \to \mathbb{R}^m$ is the discretized forward model with discretization size h. We also assume that the prior distribution $p(\boldsymbol{x}_h)$ is a zero mean Gaussian with covariance Γ_{prior}. The posterior of $\boldsymbol{x}_h$ is

$$\pi_h(\boldsymbol{x}_h|y) \propto \frac{1}{z_h}(2\pi)^{-\frac{m}{2}}|\det(\Gamma_{\text{obs}})|^{-\frac{1}{2}}\exp\Big(-\frac{1}{2}\big(y - F_h(\boldsymbol{x}_h)\big)^T\Gamma_{\text{obs}}^{-1}\big(y - F_h(\boldsymbol{x}_h)\big)\Big) \tag{5.1}$$

$$(2\pi)^{-\frac{n}{2}}|\det(\Gamma_{\text{prior}})|^{-\frac{1}{2}}\exp\big(-\frac{1}{2}\boldsymbol{x}_h^T\Gamma_{\text{prior}}^{-1}\boldsymbol{x}_h\big). \tag{5.2}$$

We need to reform the posterior so that it is easier to analyze. The target distribution of Implicit Sampling takes the form

$$\pi_h(\boldsymbol{x}_h|y) \propto \exp(-\phi(\boldsymbol{x}_h)), \tag{5.3}$$

where $\phi(\boldsymbol{x}_h)$ is the negative log of the posterior:

$$\phi(\boldsymbol{x}_h) = \frac{1}{2}\big(y - F_h(\boldsymbol{x}_h)\big)^T \Gamma_{\text{obs}}^{-1}\big(y - F_h(\boldsymbol{x}_h)\big) + \frac{1}{2}\boldsymbol{x}_h^T \Gamma_{\text{prior}}^{-1}\boldsymbol{x}_h. \tag{5.4}$$

This is different from the target distribution $\exp(-\frac{1}{2}||H(\boldsymbol{x}_h)||^2)$ used in RTO.

The following paragraphs introduce the basic idea of Implicit Sampling. Before finding sample $\boldsymbol{x}_h$, we want to find a reference variable $\boldsymbol{\xi}$ with known pdf $p(\boldsymbol{\xi})$. We also define $G(\boldsymbol{\xi}) \equiv -\log(p(\boldsymbol{\xi}))$, the negative log of $p(\boldsymbol{\xi})$. We want to obtain sample $\boldsymbol{x}_h$ satisfying

$$\phi(\boldsymbol{x}_h) - \phi_{\min} = G(\boldsymbol{\xi}) - G_{\min}, \tag{5.5}$$

where $\phi_{\min}$ and $G_{\min}$ are minimum of function $\phi(\boldsymbol{x}_h)$ and $G(\boldsymbol{\xi})$ respectively. By drawing a random sample $\boldsymbol{\xi}^i$ each time, we obtain a new sample $\boldsymbol{x}_h^i$ from solving (5.5). This the the idea of Implicit Sampling method. This again defines a transport mapping from $\boldsymbol{\xi}$ to $\boldsymbol{x}_h$, and therefore we can also use the coupling method for Implicit Sampling.

In this case, we choose the Gaussian variable as $\boldsymbol{\xi}$. We can specify $\boldsymbol{\xi}$ with zero mean and covariance matrix H_ϕ^{-1}, where H_ϕ is the Hessian of function ϕ at the minimum, i.e., the MAP point $\boldsymbol{x}_h^{MAP}$. Therefore the equation above becomes

$$\phi(\boldsymbol{x}_h) - \phi_{\min} = \frac{1}{2}\boldsymbol{\xi}^T H_\phi \boldsymbol{\xi}. \tag{5.6}$$

We will use the random map to solve equation (5.6). The basic idea of the random map is to find a scalar $\lambda_{\boldsymbol{\xi}}$ such that the following equation holds

$$\boldsymbol{x}_h = \boldsymbol{x}_h^{MAP} + \lambda_{\boldsymbol{\xi}}\boldsymbol{\xi}. \tag{5.7}$$

We want to solve equation (5.6) constraint to equation (5.7). The equation above is looking for a solution $\boldsymbol{x}_h$ at the random direction of $\boldsymbol{\xi}$. Therefore, there is a special assumption of the implicit sampling method: the domain of $\boldsymbol{x}_h$ should be star-convex so that $\lambda_{\boldsymbol{\xi}}$ is unique.

The weight of this sampler is the ratio of the posterior and the sample density $\pi_{I.S.}(\boldsymbol{x}_h)$

in equation (5.5):

$$w(\boldsymbol{x}_h) = \frac{\pi_h(\boldsymbol{x}_h|y)}{\pi_{I.S.}(\boldsymbol{x}_h)}. \tag{5.8}$$

The weight is proportional to the Jacobian of this transport mapping $T : \boldsymbol{\xi} \to \boldsymbol{x}_h$:

$$w(\boldsymbol{x}_h) \propto J(\boldsymbol{x}_h) \equiv \beta(\boldsymbol{x}_h).$$

We use β as the notation of unnormalized weight.

The corresponding weight for this random map method is

$$\begin{aligned} w \propto J(\boldsymbol{x}_h) &= J(\boldsymbol{x}_h^{MAP} + \lambda_{\boldsymbol{\xi}}\boldsymbol{\xi}) \equiv \beta(\boldsymbol{x}_h) \\ &= |\lambda_{\boldsymbol{\xi}}^{m_\lambda - 1} \frac{\boldsymbol{\xi}^T H_\phi \boldsymbol{\xi}}{\nabla\phi\ \boldsymbol{\xi}}|, \end{aligned} \tag{5.9}$$

where m_λ is the number of nonzero eigenvalues in Hessian matrix H_ϕ.

Implicit sampling is quite similar to RTO. They both obtain samples from optimising the exponential parts of the posterior. We bring back the optimisation equation of RTO:

$$\arg\min_{\boldsymbol{x}_h} \|U^T H(\boldsymbol{x}_h) - \boldsymbol{\xi}\|^2, \tag{5.10}$$

where $H(\boldsymbol{x}_h)$ is the square root of negative log posterior. In contrast, Implicit Sampling focuses on the negative log posterior ϕ. We use the Newton method to solve (5.6). We use both RTO and Implicit Sampling to generate fixed-dimension samples of the prey and predator model. It transpires that the Implicit Sampling method uses about twice the time that RTO uses, to compute the same number amount of samples.

5.1.2 Multilevel Implicit Sampling algorithm

Similar to the multilevel RTO algorithm (3), the multilevel Implicit Sampling method is used here. In this algorithm (4), we need to do a Cholesky decomposition of the Hessian matrix at the MAP point. In fact, for high dimension problems, it is also difficult to construct the Hessian matrix. We can approximate the Hessian matrix as

$$H_\phi(\boldsymbol{x}_l^{MAP}) \approx J_l(\boldsymbol{x}_l^{MAP})^T J_l(\boldsymbol{x}_l^{MAP}) + I, \tag{5.11}$$

where $J_l(\boldsymbol{x}_l^{MAP})$ is the same Jacobian (3.54) we use in the RTO method. We use the Newton method to look for the value of $\lambda_{\boldsymbol{\xi}}$ in (5.7). The performance of this numeric method is

slightly worse than the trust region method we use in the RTO method.

Algorithm 4 Multilevel Implicit Sampling with transport mapping

for $l = 0, 1, ..., L$ **do**
  Compute the MAP point $\boldsymbol{x}_l^{MAP}$, and construct the Hessian matrix $H_\phi(\boldsymbol{x}_l^{MAP})$ in equation (5.11);
  for $i = 1, ..., N_l$ **do**
    Compute a Cholesky decomposition: $LL^T = H_\phi^{-1}$;
    Obtain one standard Gaussian random sample ϵ, $\boldsymbol{\Xi}_l^i = L\epsilon$;
    Obtain samples $\boldsymbol{X}_l^i$ and $\boldsymbol{X}_{l-1}^i$ from solving (5.6) with the same $\boldsymbol{\Xi}_l^i$ using Newton method;
    Calculate the unnormalized weight $\beta(\boldsymbol{X}_l^i)$ in (5.9) and QoI $Q(\boldsymbol{X}_l^i)$;
    Construct the $\widehat{\Delta Y_l}$ in (3.19);
    Construct the $\widehat{\Delta \beta_l}$ in (3.22);
  end for
end for
Sum up all $\widehat{\Delta Y_l}$ to get $\widehat{Y}_L \equiv \sum_{l=0}^{L} \widehat{\Delta Y_l}$;
Sum up all $\widehat{\Delta \beta_l}$ to get $\widehat{z} \equiv \sum_{l=0}^{L} \widehat{\Delta \beta_l}$;
The ratio of $\widehat{Y}_L$ and $\widehat{z}$ forms $\widehat{Q}_{\mathbf{2}} \equiv \widehat{Y}_L / \hat{z}_L$.

5.2 Numerical experiments

We are using the Implicit Sampling method to investigate the same prey and predator model in (4.1.1). The QoI remains the same, that is, the number of prey at time 50. Figure (5.1) shows the density of 3000 Implicit Sampling samples. We can see that this density from Implicit Sampling is quite different from the RTO density in figure (4.1).

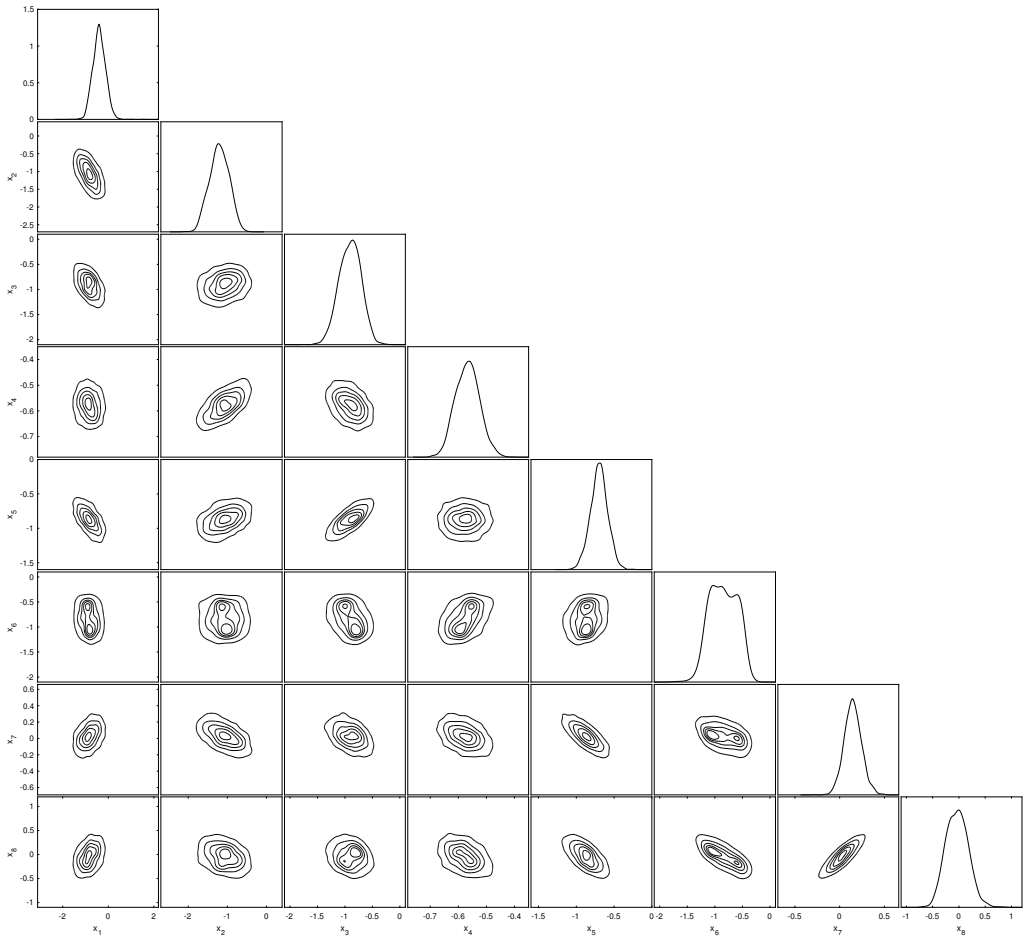

Figure 5.1: Samples of 8 parameters of the prey and predator model from Implicit Sampling. This is showing the density of samples generated from the Implicit Sampling method.

5.2.1 Convergence plots

The following plots evaluate the performance of multilevel estimators. We can see that mean and variance decay at certain rates when level increases. This again confirms the conditions in the complexity theorem (3.4.1). Comparing the mean and variance plots in figures (5.2, 5.3) shows that, the slope of the red line from the variance plot is twice that from the mean plot. However, we can see from the mean and variance plots in figures (5.2 and 5.3) of $\widehat{\Delta Y}$, that the blue and red line are very close to each other. This indicates that the variance reduction is small when applying multilevel estimators. If we calculate the proportional sample sizes across level based on this setting, the difference of sample

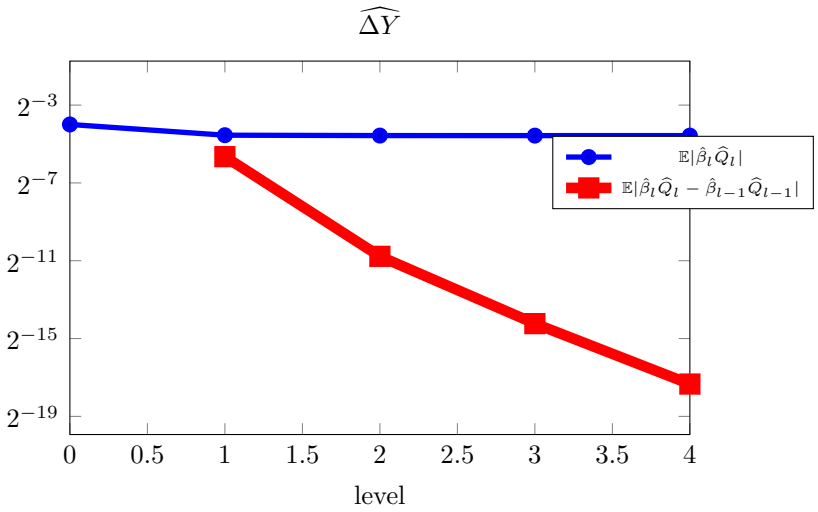

Figure 5.2: Convergence plot of mean for prey and predator model using Implicit Sampling method. We can see the blue line (mean) and red line (difference of mean) are so close at one point. If we check on the mean plots of RTO method (4.3) again, it should be some distance between blue line and red lines.

sizes between zeroth level and first level is not large. Therefore, we want to use another scenario: eliminating the zeroth level and setting the first level as the new zeroth level. We can see the mean and variance plots of this new setting in figures (5.4 and 5.5). The distance between blue lines and red lines is a lot greater now.

We decide to use the second scenario. Comparing the slopes these convergence plots, we observe that the variance plot in Figure (5.5) has a slope roughly twice that of the slope in the mean plot of Figure (5.4). The same slope pattern occurs in the weight plots in Figure (5.7).

Figure (5.8) demonstrates the rate η in condition 3 of complexity theorem (3.4.1). Here η is approximately 1. Comparing Figure (5.8) and Figure (5.5) shows that $\alpha > \eta$. Complexity theorem (3.4.1) predicts that $C^{ML} = O(\epsilon^{-2})$.

5.2.2 Optimal sample size and MSE figure

We can have the proportional sample sizes for 5 different levels in Table (5.1). The mesh size, or discretization size of zero's level is 2.5. The sample size of level 1 is about half that of level 0. Compared to previous sample sizes in Table (4.2) with the same mesh sizes across levels, where the sample sizes of level 1 are about 20% of the sample sizes of level 0. We expect the sample sizes in all level $l \geq 1$ are much less than level 0, since we want to

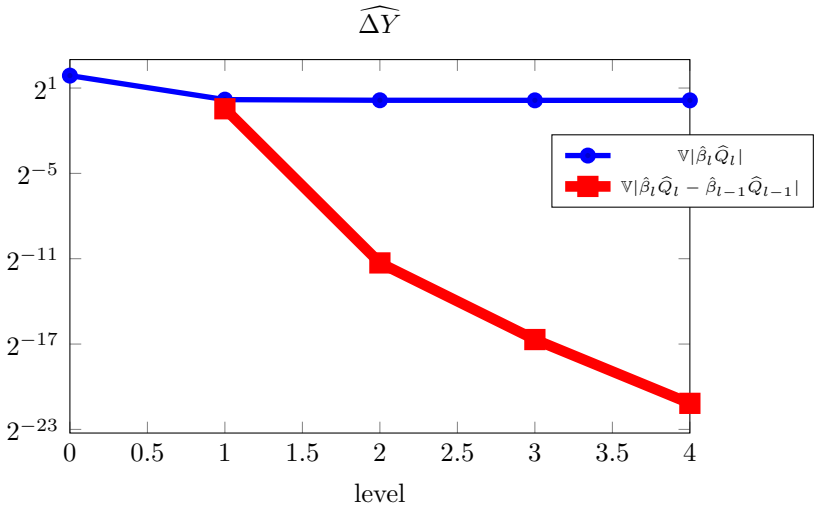

Figure 5.3: Convergence plot of variance for prey model using the Implicit sampling. The blue and red line intersect at one point indicates that there is not much variance improvement from level 0 to level 1 when we calculate the scale of sample size in equation (3.104). Therefore, N_0 and N_1 will be very close, as shown in Table (5.1).

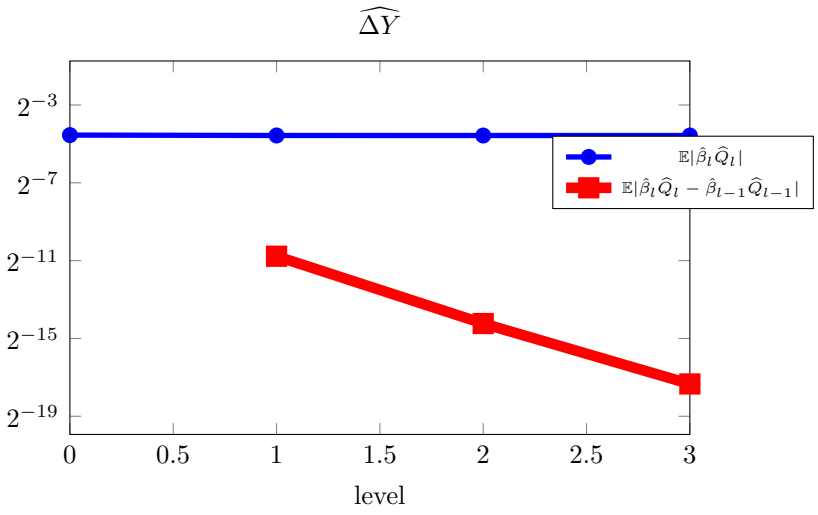

Figure 5.4: Convergence plot of the mean for Implicit sampling (4 levels). We can see that this plot has distant blue line (mean) and red line (difference of mean). They no longer cross at any point.

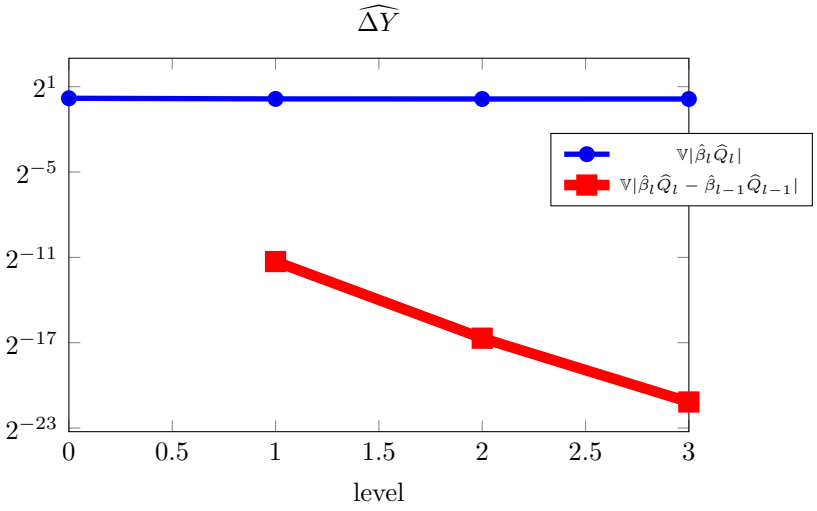

Figure 5.5: Convergence plot of the variance for Implicit Sampling (4 levels). The red line is far away from blue line now, which results in a better sample sizes scale in Table (5.2) than the last scenario shown in Table (5.1). We again observe that the slope α of this red line is approximately twice that slope ζ of Figure (5.4).

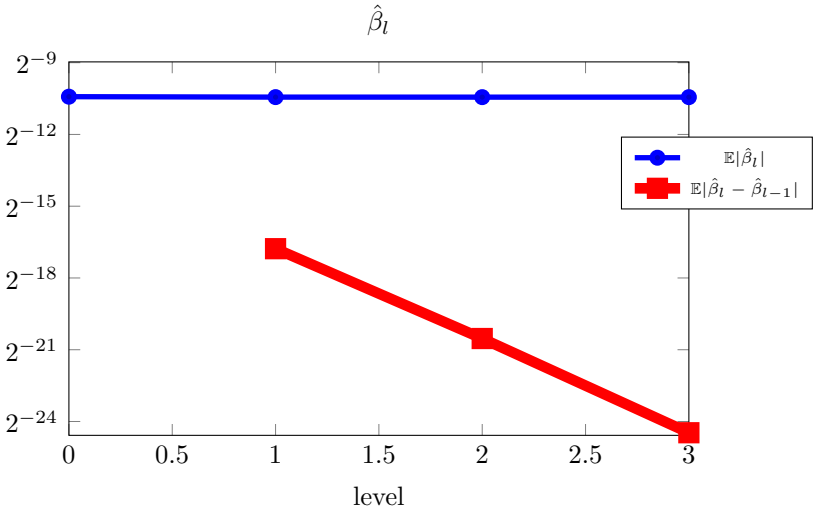

Figure 5.6: The mean plot of weight for Implicit Sampling. The slope of red line here is approximately the same as Figure (5.2).

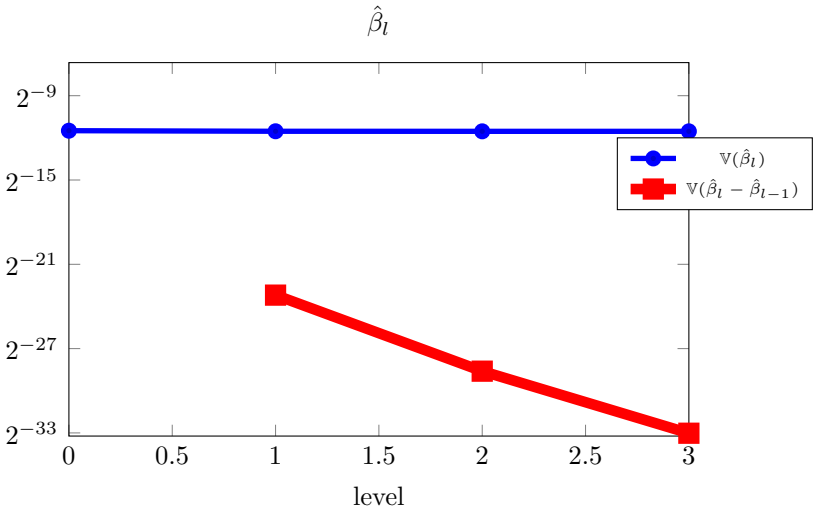

Figure 5.7: Convergence plots for variance of unnormalized weight for Implicit Sampling. The rates of converging (slope of red lines) is the same as the rates in figures (5.4) and (5.5).

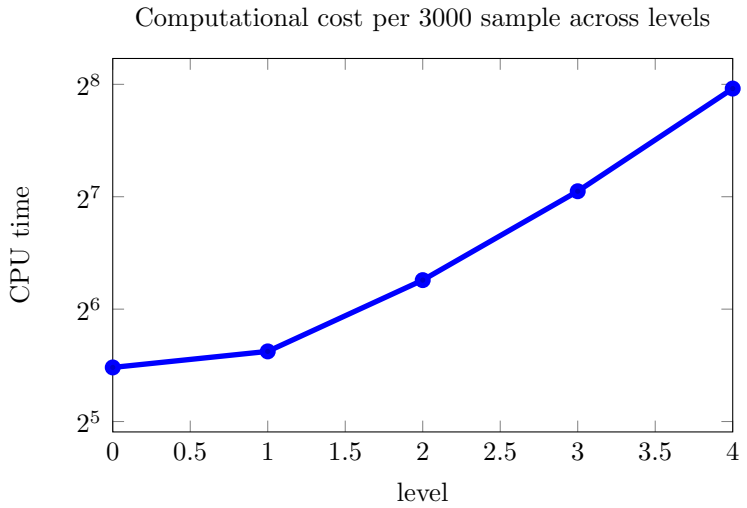

Figure 5.8: This figure plots the cost across level. The increasing rate here corresponds to the rate η in complexity theorem (3.4.1).

Level	N_0	N_1	N_2	N_3	N_4
Sample	10,000	5428	738	169	43

Table 5.1: Proportional sample sizes across five levels for Implicit Sampling method applied in prey model given 10, 000 samples at zeroth level, calculated from equation (3.104). This is the scale of sample size assuming the zeroth level has 10,000 samples. The mesh size of the zeroth level is 2.5.

Level	N_0	N_1	N_2	N_3
Sample	10,000	150	18	3

Table 5.2: This is the table of proportional four levels sample sizes for Implicit Sampling given 10, 000 samples at zeroth level. This is the scale of sample size assuming the zeroth level has 10,000 samples. The mesh size of zeroth level is 1.25. Compared to table (5.1), level 1 requires fewer samples.

put lots of computation on the zeroth level instead of finer levels. But this scale in Table (5.1) requires a lot computation on fine levels $l \geq 1$ as well, which is why we need another scenario.

When we eliminate the zeroth level in Table (5.1) and set the first level as the zeroth level, we can obtain a new sample scale in Table (5.2). Compared to Table (5.1), the fine levels $l \geq 1$ requires fewer samples, given the same N_0. We can see that the second scenario in in Table (5.2) has a greater sample size reduction in all levels $l \geq 1$. We will adopt this setting to construct the multilevel estimator. However, sample sizes in Table (5.2) starts with mesh size 1.25, which is actually costlier computation than that in Table (4.2) with mesh size 2.5 at zeroth level. Comparing to the scenario of RTO method in Chapter 4, the zeroth level of the RTO method starts with mesh size 2.5. This might implies that the Implicit Sampling is less efficient than RTO method.

The Figure (5.9) shows the calculated optimal sample sizes for stopping level 1, 2, and 3. The red line is the predicted optimal sample size for three stopping levels. $L = 1$ indicates that it has level 0 and 1. The red line has slope of -1 verifies that we are in the best case $C^{ML} = O(\epsilon^{-2})$ in the complexity theorem (3.4.1).

5.2.3 Comparison with RTO

Comparing the results with the RTO method, the ESR (effective sample ratio) in equation (2.37) is 0.03, 3% in Implicit Sampling, which is significantly smaller than the RTO, 70% given the same data y. We can weight the proposal density, 8 dimension parameter space

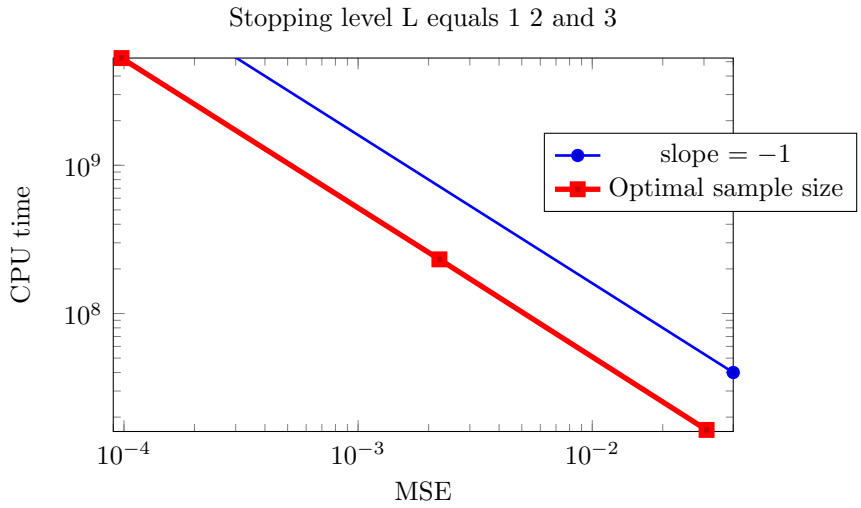

Figure 5.9: Optimal sample size for different stopping levels for Implicit Sampling. The red line is the predicted optimal sample size for three stopping levels. $L = 1$ indicates that it has level 0 and 1. Since red line has slope of -1, it verifies that we are in the best case $C^{ML} = O(\epsilon^{-2})$ of the complexity theorem (3.4.1).

in Figure (5.1) with related weight. When the ESR is high, we can accept more proposal samples into the posterior. But when the ESR is low, the weighted posterior does not represent the actual posterior like the weighted posterior (4.2) in the RTO method. Ideally, we might obtain accurate weighted posteriors when endorsing large samples.

Furthermore, the computational time of Implicit Sampling is about double the time for that of the RTO method. Implicit Sampling uses the Newton method to search a scalar in equation (5.7). This line searching might not be very efficient in the prey model, which results in a higher cost than the RTO method. Overall, the RTO method performs better in the prey and predator model than the Implicit Sampling method.

Chapter 6

Change of dimension

When we are dealing with infinite-dimension inverse problems, we discretize the parameter space to a finite-dimension space. In this discretization process, the dimension of parameters varies when we use different increments. In the multilevel method, one can have different dimensions of parameters across levels. It is very expensive to compute the parameters in the fine levels, like level 4, so we want to use the parameters in the coarse levels to construct ones in the fine levels. This chapter introduces a way to use the parameters at the previous level to generate the new parameters. The first section introduces the derivation of this method and the second section includes the implementation in a tomography model.

6.1 Theoretical result

This section describes how we use parameters from a previous level to form the new parameters. Suppose we have parameter vectors $\boldsymbol{x}_{l-1}$ at level l and $\boldsymbol{x}_l$ at level $l-1$. $\boldsymbol{x}_l$ and $\boldsymbol{x}_{l-1}$ have different dimensions, and dimension of $\boldsymbol{x}_l$ is larger than $\boldsymbol{x}_{l-1}$. We have discussed the transport mapping in section 3.2. There is a transport mapping T, from $\boldsymbol{\xi}$ to $\boldsymbol{x}$. We are using the reverse map $T^{-1} \equiv S_{RTO}$ here. We define S_{RTO} as the the mapping from $\boldsymbol{x}$ to $\boldsymbol{\xi}$. After the RTO transformations at two levels, l and $l-1$, we have

$$\begin{aligned} S_{RTO}(\boldsymbol{x}_l) &= \boldsymbol{\xi}_l \sim \mathcal{N}(0, I_l); \\ S_{RTO}(\boldsymbol{x}_{l-1}) &= \boldsymbol{\xi}_{l-1} \sim \mathcal{N}(0, I_{l-1}). \end{aligned}$$

The dimensions of $\boldsymbol{x}_l$ and $\boldsymbol{x}_{l-1}$ are not identical. We denote the Laplace precision matrices P_l, P_{l-1} with the same dimensions as the identity matrices I_l, I_{l-1} respectively. We can

generate the two new random vector as:

$$\begin{aligned} P_l^{-\frac{1}{2}} S_{RTO}(\boldsymbol{x}_l) = P_l^{-\frac{1}{2}} \boldsymbol{\xi}_l \equiv \boldsymbol{r}_l \sim \mathcal{N}(\mathbf{0}, P_l^{-1}); \\ P_{l-1}^{-\frac{1}{2}} S_{RTO}(\boldsymbol{x}_{l-1}) = P_{l-1}^{-\frac{1}{2}} \boldsymbol{\xi}_{l-1} \equiv \boldsymbol{r}_{l-1} \sim \mathcal{N}(\mathbf{0}, P_{l-1}^{-1}). \end{aligned} \tag{6.1}$$

This step above is simply reforming the transport mapping processes. The next step is to partition $\boldsymbol{r}_l$ as $(\tilde{\boldsymbol{r}}_l, \tilde{\boldsymbol{r}}_{l-1})^T$ where the dimension of $\tilde{\boldsymbol{r}}_{l-1}$ is the same as for $\boldsymbol{r}_{l-1}$. For example, if $\boldsymbol{r}_l = (r_1, r_2, r_3, r_4, r_5)^T$ where r_j's are drawn from $\mathcal{N}(\mathbf{0}, P_l^{-1})$, and $\boldsymbol{r}_{l-1} = (r_1, r_3, r_5)^T$, then $\boldsymbol{r}_l = (\tilde{\boldsymbol{r}}_l, \tilde{\boldsymbol{r}}_{l-1})^T = (\, r_2, r_4, \;\; r_1, r_3, r_5\,)^T$. We denote the partitioned mean and precision matrix of $\tilde{\boldsymbol{r}}_l$ and $\tilde{\boldsymbol{r}}_{l-1}$ as $(\boldsymbol{\mu}_A, \boldsymbol{\mu}_B)^T = (\vec{\mathbf{0}}, \vec{\mathbf{0}}_{l-1})^T$ and

$$P_l = \begin{pmatrix} P_{AA} & P_{AB} \\ P_{BA} & P_{BB} \end{pmatrix}. \tag{6.2}$$

The inverse of this precision matrix is denoted as

$$P_l^{-1} = \begin{pmatrix} C_{AA} & C_{AB} \\ C_{BA} & C_{BB} \end{pmatrix}. \tag{6.3}$$

Therefore the marginal distribution of $\tilde{\boldsymbol{r}}_{l-1}$ is $\mathcal{N}(\mathbf{0}, C_{BB})$. The marginal distribution of $\tilde{\boldsymbol{r}}_{l-1}$ is not identical to $\boldsymbol{r}_{l-1}$, but has the connection:

$$\begin{aligned} \tilde{\boldsymbol{r}}_{l-1} &= C_{BB}^{\frac{1}{2}} P_{l-1}^{-\frac{1}{2}} \left(\boldsymbol{r}_{l-1} - \vec{\mathbf{0}}_{l-1} \right) + \boldsymbol{\mu}_B, \\ &= C_{BB}^{\frac{1}{2}} P_{l-1}^{-\frac{1}{2}} \boldsymbol{r}_{l-1}, \end{aligned} \tag{6.4}$$

where C_{BB} is the inverse of Schur complement of block A of the matrix P_l:

$$C_{BB} = \left(P_{BB} - P_{BA} P_{AA}^{-1} P_{AB} \right)^{-1}. \tag{6.5}$$

At the current stage, we have rearranged the order of vector $\boldsymbol{r}_l$ so that we can separate the known $\tilde{\boldsymbol{r}}_{l-1}$ and the unknown vector $\tilde{\boldsymbol{r}}_l$. We have partitioned the mean and precision matrix accordingly so that it is ready to generate the unknown $\tilde{\boldsymbol{r}}_l$.

We suppose at the $l-1$ level, $\boldsymbol{r}_{l-1}$ is transformed by known samples $\boldsymbol{x}_{l-1}$ as in equation

(6.1). The unknown $\tilde{\boldsymbol{r}}_l$ is thus a conditional distribution in the following form:

$$\tilde{\boldsymbol{r}}_l|\tilde{\boldsymbol{r}}_{l-1} \sim \mathcal{N}\left(\boldsymbol{\mu}_A - P_{AA}^{-1}P_{AB}\left(\tilde{\boldsymbol{r}}_{l-1} - \boldsymbol{\mu}_B\right), P_{AA}^{-1}\right),$$

With the assumptions $(\boldsymbol{\mu}_A, \boldsymbol{\mu}_B)^T = (\vec{\mathbf{0}}, \vec{\mathbf{0}}_{l-1})^T$, the conditional distribution has the following form:

$$\tilde{\boldsymbol{r}}_l|\tilde{\boldsymbol{r}}_{l-1} \sim \mathcal{N}\left(-P_{AA}^{-1}P_{AB}\tilde{\boldsymbol{r}}_{l-1}, P_{AA}^{-1}\right). \tag{6.6}$$

The result above uses the Schur complement of block P_{BB} of the matrix P_l in (6.2). This requires the matrix P_{BB} to be invertible. The new parameters $\tilde{\boldsymbol{r}}_l$ in the next level follow a conditional distribution of the known mean and variance.

Combining equations (6.4) and (6.6), the realisation of $\tilde{\boldsymbol{r}}_l$ comes from a linear transformation:

$$\begin{aligned}\tilde{\boldsymbol{r}}_l &= -P_{AA}^{-1}P_{AB}\tilde{\boldsymbol{r}}_{l-1} \\ &= -P_{AA}^{-1}P_{AB}C_{BB}^{\frac{1}{2}}P_{l-1}^{-\frac{1}{2}}\boldsymbol{r}_{l-1}. \\ &= -P_{AA}^{-1}P_{AB}C_{BB}^{\frac{1}{2}}P_{l-1}^{-1}\boldsymbol{\xi}_{l-1}\end{aligned} \tag{6.7}$$

where the second line is derived from (6.4) and the last line uses equation (6.1). After obtaining $\tilde{\boldsymbol{r}}_l$, we can use this combined $\boldsymbol{r}_l = (\tilde{\boldsymbol{r}}_{l-1}, \tilde{\boldsymbol{r}}_l)^T$ to generate $\boldsymbol{x}_l$:

$$\begin{aligned}P_l^{-\frac{1}{2}}S_{RTO}(\boldsymbol{x}_l) &= \boldsymbol{r}_l \\ S_{RTO}(\boldsymbol{x}_l) &= P_l^{\frac{1}{2}}\boldsymbol{r}_l \\ T(P_l^{\frac{1}{2}}\boldsymbol{r}_l) &= \boldsymbol{x}_l.\end{aligned} \tag{6.8}$$

$P_l^{\frac{1}{2}}\boldsymbol{r}_l$ is served as the reference variable $\boldsymbol{\xi}_l$ in the transport mapping. Instead of using a new random vector from Gaussian as the reference variable, we use a fixed vector $P_l^{\frac{1}{2}}\boldsymbol{r}_l$, so that it is possible to reduce the iterations of optimisation needed, given the information of the previous level embedded in $P_l^{\frac{1}{2}}\boldsymbol{r}_l$. This is similar to the technique that uses the samples from the previous level as the initial values to optimize the current level in the case which the dimension of parameters does not change. This technique in changing dimension setting can be summarized by the algorithm (5) below. Algorithm (5) only depicts the procedures of solving one sample $\boldsymbol{x}_l^i$ at level l, which should be embedded in multilevel algorithm (3). The full algorithm is shown in (6).

Algorithm 5 Compute parameter in the finer level given samples in a coarse level

1: Obtain $\boldsymbol{r}_{l-1}^i$ at $l-1$ level after the matrix multiplication in (6.1);
2: Compute $\tilde{\boldsymbol{r}}_l^i$ using the linear transformation in (6.7);
3: Combine $\boldsymbol{r}_l^i = (\tilde{\boldsymbol{r}}_l^i, \tilde{\boldsymbol{r}}_{l-1}^i)^T$ from (6.4) and (6.7);
4: Solve $T(P_l^{\frac{1}{2}}\boldsymbol{r}_l^i) = \boldsymbol{x}_l^i$.

6.2 Numerical experiments

6.2.1 Tomography problem

This is a model in two dimensional positron emission tomography (PET) imaging. PET imaging scans an object of interest through the emission of gamma rays to reveal the densities of the object. When gamma rays pass out of the object, photon detectors count the occurrence of gamma rays from sources and we use the count from detector to reconstruct the density of the object. Suppose we are interested in the image in a 2D domain Ω. The density function is represented as $\exp(x(s))$, where $x(s)$ follows a Gaussian prior. The intensity of gamma rays after scanning the object is modeled as

$$I_{d,i} = I_{s,i}\exp\left(-\exp\left(\int_{l_i(s)} \exp(x(s))\right)\right), \tag{6.9}$$

where $I_{d,i}$ and $I_{s,i}$ are the intensities along the path $l_i(s)$ at the photon detectors and sources. $I_{s,i}$ is parameterised as λ for all i, which is an unknown hyperparameter in this inverse problem. This is a hierarchical model, with a posterior in equation (2.19). If we split the 2D domain Ω into grids with n even cells, then the parameter $x(s)$ can be discretized as $\boldsymbol{x}$ with

$$\int_{l_i(s)} \exp(x(s)) \approx \Sigma_{j=1}^n \mathbf{B}_{ij}\exp(\boldsymbol{x}_j), \tag{6.10}$$

where $\mathbf{B}_{ij}$ is the intersection of path $l_i(s)$ and cell j, and $\exp(\boldsymbol{x}_j)$ is the density in cell j. Here $\boldsymbol{x}_j$ is the jth entry in vector $\boldsymbol{x}$. It is a line integration along different paths $l_i(s)$. As we split Ω into different numbers of grids, the dimension, n, of parameter space changes. In the multilevel method, the discretization size is the cell size in Ω.

The forward model $F : \mathbb{R}^n \to \mathbb{R}^m$ is defined as

$$F(\boldsymbol{x}) = \exp\left(-\mathbf{B}\exp\left(\boldsymbol{x}\right)\right), \tag{6.11}$$

where $\mathbf{B} \in \mathbb{R}^{m\times n}$ has non-negative entries, since its entry is the length of intersection.

Algorithm 6 Multilevel RTO with transport mapping for changing dimension of parameters

Follow algorithm (3) when $l = 0$;
for $l = 1, ..., L$ **do**
    Compute MAP point $\boldsymbol{x}_l^{MAP}$;
    Form the Jacobian matrix $J(\boldsymbol{x}_l^{MAP})$;
    Construct the thin QR factorization of $J(\boldsymbol{x}_l^{MAP})$;
    for $i = 1, ..., N_l$ **do**
        Draw a sample $\boldsymbol{\xi}_{l-1}^i$ from $\mathcal{N}(0, I_{l-1})$;
        Obtain RTO samples $\boldsymbol{X}_{l-1}^i$ from solving (3.56) with the $\boldsymbol{\xi}_{l-1}^i$ as the reference variable;
        Use algorithm (5) to get $\boldsymbol{X}_l^i$;
        Calculate the unnormalized weight: $\beta(\boldsymbol{X}_l^i)$ and $\beta(\boldsymbol{X}_{l-1}^i)$; and QoI $Q(\boldsymbol{X}_l^i)$ and $Q(\boldsymbol{X}_{l-1}^i)$;
        Construct the $\widehat{\Delta Y_l}$ in (3.19);
        Construct the $\widehat{\Delta \beta_l}$ in (3.22);
    end for
end for
Sum up all $\widehat{\Delta Y_l}$ to get $\widehat{Y}_L \equiv \sum_{l=0}^{L} \widehat{\Delta Y_l}$;
Sum up all $\widehat{\Delta \beta_l}$ to get $\widehat{z} \equiv \sum_{l=0}^{L} \widehat{\Delta \beta_l}$;
The ratio of $\widehat{Y}_L$ and $\widehat{z}$ forms $\widehat{Q}_{\mathbf{2}} \equiv \widehat{Y}_L / \hat{z}_L$.

The data are integer numbers collected by the detectors. We model the data in Poisson distributions. Unlike the previous models, including the prey model and PDE, the likelihood function is not Gaussian. We have the integer data $y \in \mathbb{R}^m$, thus the Poisson likelihood is

$$\mathcal{L}(y|\boldsymbol{x},\lambda) = \frac{\lambda \sum_{i=1}^{m} y_i}{\Pi_{i=1}^{m} y_i!} \exp\left(\sum_{i=1}^{m} \left(y_i \log F_i(\boldsymbol{x}) - \lambda F_i(\boldsymbol{x})\right)\right), \tag{6.12}$$

where y_i is the ith element in vector y and F_i is the ith element in vector F, the output of forward model. We need to approximate this Poisson likelihood so that we can put it in the posterior function of the RTO distribution. The equations (6.13) and (6.14) below provides a Gaussian approximation of this Poisson likelihood [2]. We first take the log of the likelihood:

$$\log\mathcal{L}(y|\boldsymbol{x},\lambda) = -\sum_{i=1}^{m} \log y_i! - \lambda \sum_{i=1}^{m} \exp\left(\log F(\boldsymbol{x})\right) + \sum_{i=1}^{m} y_i \left(\log\lambda + \log F(\boldsymbol{x})\right). \tag{6.13}$$

If we fixed the parameter $\boldsymbol{x}$, data y is also fixed. We denote the fixed vectors as $\boldsymbol{x}^*$ and y^*. The resulting $\log\mathcal{L}(y|\boldsymbol{x},\lambda)$ can be expanded in a second-order Taylor series at the fixed point:

$$\begin{aligned}
\log\mathcal{L}(y^*|\boldsymbol{x},\lambda) =& \log\mathcal{L}(y^*|\boldsymbol{x}^*,\lambda) + \big(F(\boldsymbol{x}) - F(\boldsymbol{x}^*)\big)^T \nabla_{F(\boldsymbol{x})} \log\mathcal{L}(y^*|\boldsymbol{x}^*,\lambda) \\
&+ \frac{1}{2}\big(F(\boldsymbol{x}) - F(\boldsymbol{x}^*)\big)^T \nabla^2_{F(\boldsymbol{x})} \log\mathcal{L}(y^*|\boldsymbol{x}^*,\lambda)\big(F(\boldsymbol{x}) - F(\boldsymbol{x}^*)\big) \\
&+ O\left(\left\|F(\boldsymbol{x}) - F(\boldsymbol{x}^*)\right\|^3\right) \\
=& \log\mathcal{L}(y^*|\boldsymbol{x}^*,\lambda) - \frac{\lambda}{2}\big(F(\boldsymbol{x}) - \log(y/\lambda)\big)^T \mathrm{diag}\big(F(\boldsymbol{x}^*)\big)^{-1}\big(F(\boldsymbol{x}) - \log(y/\lambda)\big) \\
&+ O\left(\left\|F(\boldsymbol{x}) - F(\boldsymbol{x}^*)\right\|^3\right) \\
\approx& \text{constant} - \frac{\lambda}{2}\big(F(\boldsymbol{x}) - \log(y/\lambda)\big)^T \mathrm{diag}\left(F(\boldsymbol{x}^*)\right)^{-1}\big(F(\boldsymbol{x}) - \log(y/\lambda)\big).
\end{aligned} \tag{6.14}$$

The last equation (6.14) is the Gaussian approximation of the Poisson likelihood. The prior function for this model is the same as in example (2.1.1). The posterior of the hierarchical model is shown in section 2.1.3. In this chapter, we fix the hyper-parameter and only

evaluate parameter $\boldsymbol{x}$. The posterior takes the following form:

$$p(\boldsymbol{x}|y,\lambda) \propto (\frac{2\pi}{\lambda})^{-\frac{m}{2}} |\det\big(\operatorname{diag}\left(F(\boldsymbol{x}^*)\right)\big)|^{-\frac{1}{2}} \exp\left(-\frac{\lambda}{2}\big(F(\boldsymbol{x}) - \log\left(y/\lambda\right)\big)^T \operatorname{diag}\left(F(\boldsymbol{x}^*)\right)^{-1} \big(F(\boldsymbol{x}) - \log\left(y/\lambda\right)\big)\right) p_0\left(\boldsymbol{x}\right).$$

The dimension of the parameter starts from a large number. The zero level has $(32+1)^2 = 1089$ parameters, which means that the 2D domain Ω is split into 16×16 grids. The other three levels have $(64+1)^2 = 4225$, $(128+1)^2 = 16641$ and $(256+1)^2 = 66049$ parameters.

6.2.2 Numerical results

The following table (6.1) shows the effective sample ratio at all four levels. The zero level has the slightly larger effective sample ratio since we use the MAP point as the initial value to perform the optimisations at the beginning. The remaining effective sample ratios increase along with discretization levels.

Level	$l=0$	$l=1$	$l=2$	$l=3$
Effective Sample Ratio (ESR)	0.23275	0.20430	0.21044	0.21324

Table 6.1: This table is about the effective sample ratios in tomography problem across four different level with degree of freedom 32, 64, 128, 256.

The following two figures (6.1) and (6.2) are the convergence plots of $\widehat{\Delta Y}$ in this tomography problem. We can see that the variance (blue lines) become flat after level 1. The plots of weights in figures (6.3) and (6.4), are also showing a similar pattern to the figures (6.1) and (6.2). The slopes of reds lines in figures (6.1) and (6.2) are actually the same as those in figures (6.3) and (6.4) respectively. The mean and variance of weights (blue lines) are converging to a value, and their difference (red lines) are also reducing. The means and variances of difference (red lines) are decreasing at two certain rates, denoted as ζ and α respectively, which match the assumptions of the complexity theorem(2.3.1). Therefore, we can apply the complexity theorem to approximate the MSE and corresponding optimal sample size.

We first calculate the sample size again with the formula (3.104), which requires the variance of difference at each level as well as the CPU time across levels. The CPU time across levels are shown in Figure (6.5). The rate in this figure corresponds to the rate η in the complexity theorem (3.4.1). Comparing this Figure (6.5) and variance Figure (6.2),

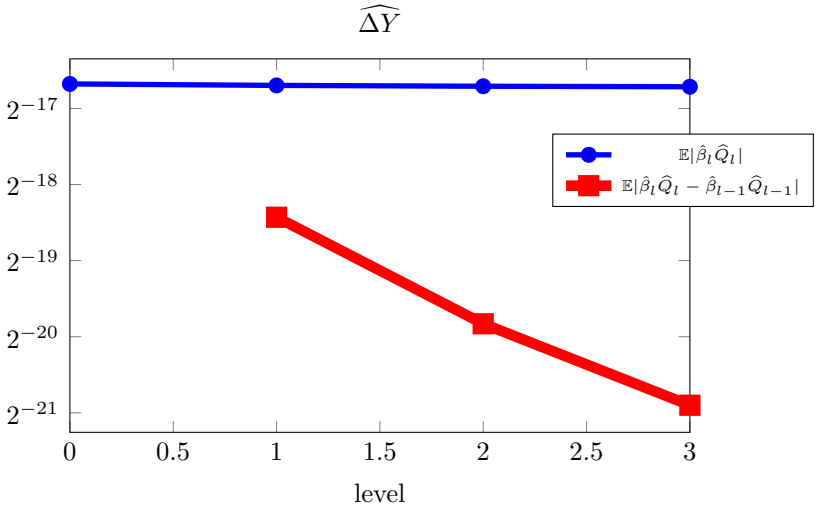

Figure 6.1: Convergence plot for mean in tomography problem. The slope of the red line correlates to the converging rate ζ in the complexity theorem (3.4.1).

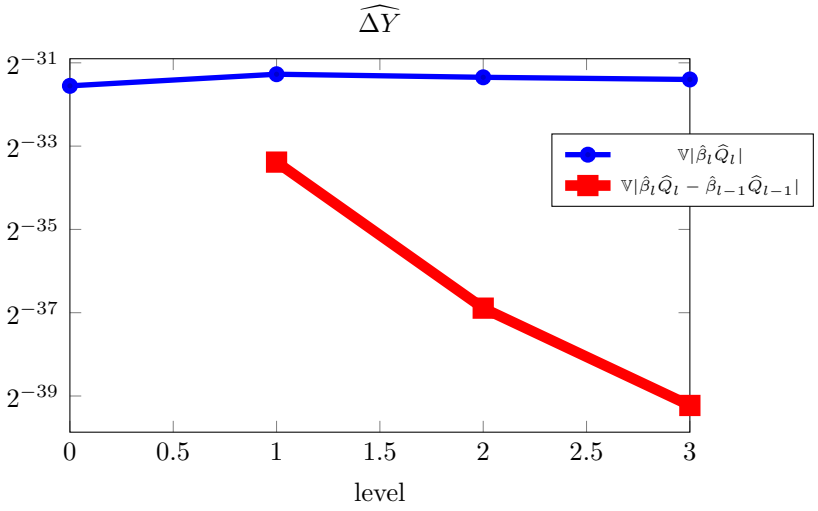

Figure 6.2: Convergence plot for variance in tomography problem. The slope of the red line here corresponds to α in the complexity theorem (3.4.1). The rate here is approximately twice that in Figure (6.1).

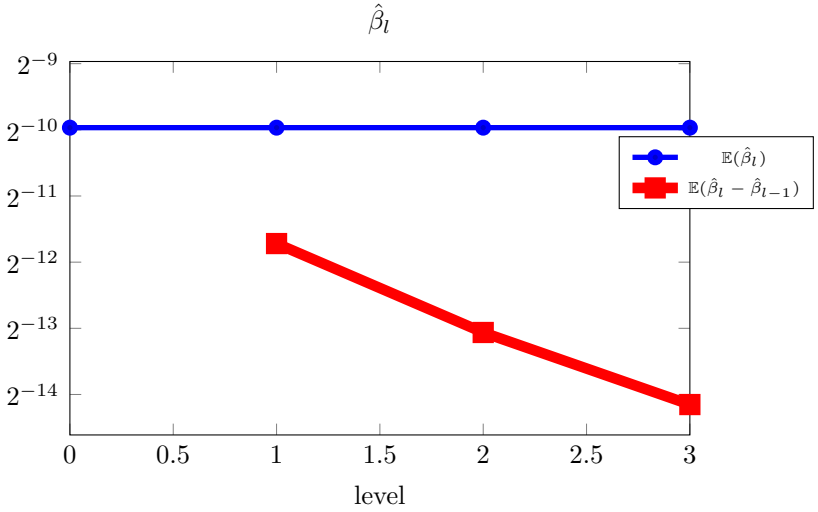

Figure 6.3: The mean plot of weight in tomography model. The slope of red line approximately equals in that in Figure (6.1).

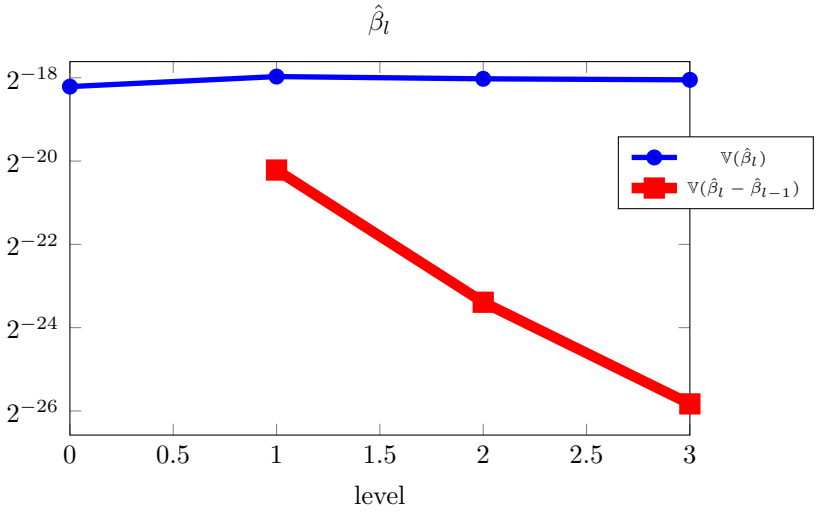

Figure 6.4: Convergence plots for the variance of unnormalized weight in tomography problem. The slope of red line is approximately equal to that in Figure (6.2).

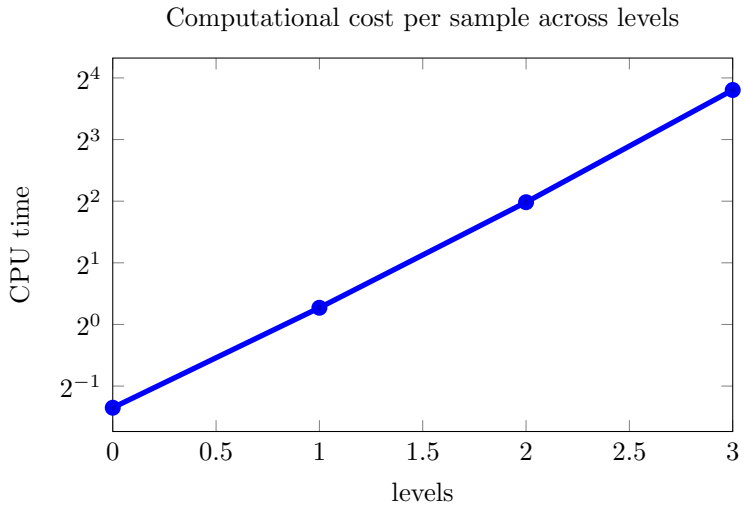

Figure 6.5: CPU time per sample across levels in tomography problem. The computational cost increases exponentially with respect to level. This slope here corresponds to η in the complexity theorem (3.4.1).

Level	N_0	N_1	N_2	N_3
Sample	10,000	3022	419	117

Table 6.2: Proportional sample sizes across levels for Tomography model given 10, 000 samples at the zeroth level. This is calculated from equation (3.104).

where the y-axis is in the log scale, the slope of Figure (6.5) is about 0.5 units per level whilst the slope of red line in Figure (6.2) is about 1 unit per level. Therefore, the rate α is greater then rate η. Based on the complexity theorem (3.4.1), we have the best case, i.e., the computation cost C^{ML} is proportional to reciprocal MSE, i.e., $C^{ML} = O(\epsilon^{-2})$. We will test this in the CPU time versus MSE figure.

We can have the following scale of sample size across levels in Table (6.2). After obtaining the proportional samples sizes, we can have the MSE versus optimal sample sizes as Figure (6.6). This figure verifies that $C^{ML} = O(\epsilon^{-2})$. When comparing Figure (6.5) and variance Figure (6.2), we can find that $\alpha > \eta$. The complexity theorem succeeds to predict that the cost of the multilevel self-normalizing estimator is proportional to reciprocal MSE.

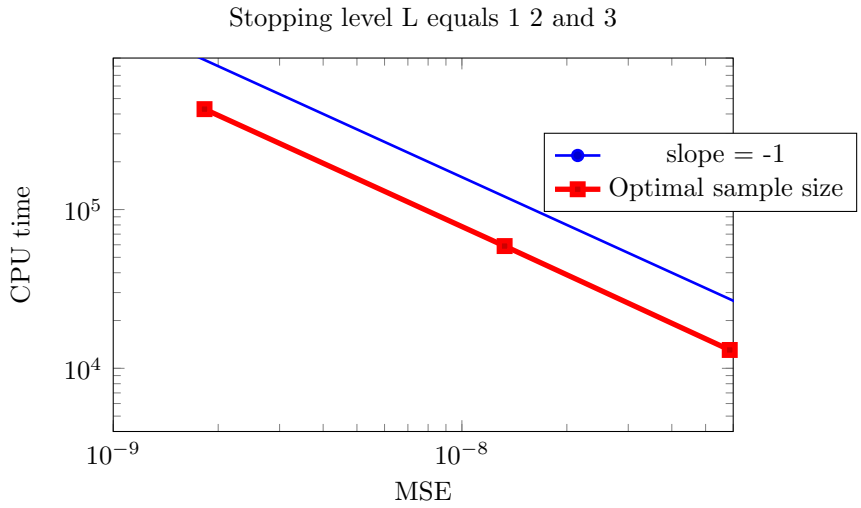

Figure 6.6: Optimal sample size for different stopping levels. This plot shows the optimal sample size for different stopping levels $L = 1, 2$ and 3. $L = 1$ means that the multilevel structure includes two levels, level 0 and 1. The parallel lines indicate that increasing CPU time is proportional to decreasing MSE. This is the best case, $C^{ML} = O(\epsilon^{-2})$ in the complexity theorem (3.4.1).

Performance of changing dimension technique

This numerical experiment uses the changing dimension method adapted in the RTO algorithm. We also compute the single level results with the MAP point as the initial guess in the optimisation, i.e., without the changing dimension method. We were expecting that the changing dimension method would accelerate the computation of the fine levels. In fact, the single level computation with the MAP point is even faster than the changing dimension results. The changing dimension method does not play an important role in reducing cost here.

Chapter 7

Conclusion

The thesis sets out to investigate the Monte Carlo samplers, especially for optimisation-based samplers, in Bayesian inverse problem. This study has also identified and develop the complexity theorem for the multilevel self-normalizing estimator.

The numerical experiments are RTO sampler and Implicit Sampling, in prey model, the PDE model and a tomography model. The experiment in prey models first uses the multilevel RTO method. There are 8 fixed parameters in prey model and the RTO method performs quite well. The Effective Sample Ratio (ESR) for RTO in prey model is about 70%. The multilevel estimator in prey model falls at the best scenario in complexity theorem, i.e., the computational cost is proportional to the reciprocal MSE. The elliptic PDE model is on a square domain with a Gaussian process prior. The RTO method also performs quite good with an Effective Sample Ratio of about 30% in the PDE model. Considering the PDE model is quite complicate and expensive to evaluate, the ESR of RTO is high. The multilevel estimator in PDE model also falls at the best scenario in complexity theorem. The multilevel estimator for both experiments in Chapter 5 and 6 are in the best scenario in complexity theorem as well. However, the Implicit Sampling implemented on the same prey model has a very low ESR, which is smaller than 1%. Implicit Sampling is much less efficient than RTO in the prey model. Chapter 6 is working on a tomography model with changing dimension of parameters when level increases. We implements the technique shown in Chapter 6, and unfortunately the technique doesn't accelerate the computational speed a lot in this model.

In general, it seems that even though the self-normalizing estimator is biased, the bias occurred in self-normalizing process is negligible compared to the MSE. Taken together, these numerical results suggest that if the conditions are all met as suggested in the complexity theorem, the MSE will converge at the rate as suggested in the theorem. The findings of this investigation complement those of earlier studies on complexity theorem

of multilevel estimators. The findings reported here shed new light on self-weighting estimators in multilevel setting. It turns out that the complexity theorem of self-weighting estimators and its corresponding proof are quite similar to standard complexity theorem. This work contributes to existing knowledge of multilevel idea by using delta method to prove the complexity theorem of the self-normalizing estimator. This new understanding should help to improve predictions of optimal sample size for a multilevel self-normalizing estimator.

Several questions still remain to be answered. Possible research are implementing RTO in the hierarchical model, and using Quasi Monte Carlo (QMC) to further accelerate the multilevel sampler. The hierarchical model involves hyper-parameters as shown in section 2.1.3 where RTO will be used to sample the unnormalized conditional posterior in equation (2.23). QMC will be a useful technique to deal with high dimension problems.

Bibliography

[1] Anderson, D. F., Higham, D. J., and Sun, Y. (2018). Computational complexity analysis for monte carlo approximations of classically scaled population processes. *Multiscale Modeling and Simulation*, 16:1206–1226.

[2] Bardsley, J. M. and Cui, T. (2020). Optimization-based mcmc methods for nonlinear hierarchical statistical inverse problems. *SIAM/ASA J. Uncertainty Quantification*, 9:29–64.

[3] Bardsley, J. M., Cui, T., Marzouk, Y. M., and Wang, Z. (2020). Scalable optimization-based sampling on function space. *SIAM Journal on Scientific Computing*, 42(2):A1317–A1317.

[4] Bardsley, J. M. and Luttman, A. (2016). A metropolis-hastings method for linear inverse problems with poisson likelihood and gaussian prior. *International Journal for Uncertainty Quantification*.

[5] Bardsley, J. M., Solonen, A., Haario, H., and Laine, M. (2014). Randomize-then-optimize: A method for sampling from posterior distributions in nonlinear inverse problems. *SIAM Journal on Scientific Computing*.

[6] Beskos, A., Jasra, A., Law, K., Tempone, R., and Zhou, Y. (2017a). Multilevel sequential monte carlo samplers. *Stochastic Processes and their Applications*.

[7] Beskos, A., Jasra, A., Law, K., Tempone, R., and Zhou, Y. (2017b). Multilevel sequential monte carlo samplers. *Stochastic Processes and their Applications*, 127(5):1417–1440.

[8] Beskos, A., J. A. M. E. e. a. (2015). Sequential monte carlo methods for bayesian elliptic inverse problems. *Stat Comput*, 25:727–737.

[9] Calvetti, D. and Somersalo, E. (2006). Large-scale statistical parameter estimation in complex systems with an application to metabolic models. *Multiscale Modeling & Simulation*, 5(4):1333–1366.

[10] Chorin, A. J. and Tu, X. (2009). Implicit sampling for particle filters. *Proceedings of the National Academy of Sciences*, 106(41):17249–17254.

[11] Cliffe, K. A., Giles, M. B., Scheichl, R., Teckentrup, A. L., Oosterlee, C. W., and Cliffe, A. K. B. A. (2011). Multilevel monte carlo methods and applications to elliptic pdes with random coefficients. *Comput Visual Sci*, 14:3–15.

[12] Cochran, W. G. (1977). *Sampling Techniques, 3rd Edition*. John Wiley and Sons.

[13] Cotter, S. L., Dashti, M., Robinson, J. C., and Stuart, A. M. (2009). Bayesian inverse problems for functions and applications to fluid mechanics. *Inverse Problems*, 25(11):115008.

[14] Cui, T., Detommaso, G., and Scheichl, R. (2019). Multilevel dimension-independent likelihood-informed mcmc for large-scale inverse problems. *ArXiv*, abs/1910.12431.

[15] Del, P., Doucet, A., and Jasra, A. (2006). Sequential monte carlo samplers. *J. R. Statist. Soc. B*, 68:411–436.

[16] Dodwell, T. J., Ketelsen, C., Scheichl, R., and Teckentrup, A. L. (2013). A hierarchical multilevel markov chain monte carlo algorithm with applications to uncertainty quantification in subsurface flow. very complex notation, one of the early papers.

[17] Dostert, P., Efendiev, Y., Hou, T., and Luo, W. (2006). Coarse-gradient langevin algorithms for dynamic data integration and uncertainty quantification. *Journal of Computational Physics*, 217(1):123–142. Uncertainty Quantification in Simulation Science.

[18] Giles, M. B. (2008). Multilevel Monte Carlo Path Simulation. *Operations Research*, 56(3):607–617.

[19] Giles, M. B. (2018). Multilevel monte carlo methods. *Acta Numerica*.

[20] Giles, M. B. and Waterhouse, B. J. (2009). Multilevel quasi-monte carlo path simulation. *Radon Series Comp. Appl. Math*, 8:1–18.

[21] Goodman, J. and Sokal, A. D. (1989). Multigrid mc method. conceptual foundation. *PHYSICAL REVIEW D*, 40.

[22] Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97.

[23] Heinrich, S. (2001). Multilevel monte carlo methods. pages 58–67.

[24] Helin, T. and Lassas, M. (2009). Hierarchical models in statistical inverse problems and the mumford-shah functional *.

[25] Higdon, D. (2006). A primer on space-time modeling from a bayesian perspective.

[26] Hoang, V. H., Schwab, C., and Stuart, A. M. (2013). Complexity analysis of accelerated MCMC methods for bayesian inversion. *Inverse Problems*, 29(8):085010.

[27] Jasra, A., Kamatani, K., Osei, P. P., and Zhou, Y. (2018). Multilevel particle filters: normalizing constant estimation. *Stat Comput*, 28:47–60.

[28] Kaipio, J. P. (2005). *Statistical Inversion Theory*.

[29] Kaipio, J. P., Kolehmainen, V., Somersalo, E., and Vauhkonen, M. (2000). Statistical inversion and monte carlo sampling methods in electrical impedance tomography. *Inverse Problems*, 16(5):1487–1522.

[30] Kaipio, J. P. and Somersalo, E. (2004). *Statistical and Computational Inverse Problems*, volume 160.

[31] Kaltenbacher, B., Schöpfer, F., and Schuster, T. (2009). Iterative methods for nonlinear ill-posed problems in banach spaces: convergence and applications to parameter identification problems. *Inverse Problems*, 25(6):065003.

[32] Kantas, N., Beskos, A., and Jasra, A. (2014). Sequential monte carlo methods for high-dimensional inverse problems: A case study for the navier–stokes equations. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):464–489.

[33] Latz, J., Papaioannou, I., and Ullmann, E. (2018). Multilevel sequential monte carlo for bayesian inverse problems. *Journal of Computational Physics*, 368:154–178. another ml sequential paper¡br/¿¡br/¿.

[34] Lindgren, F., Rue, H., and Lindström, J. (2011). An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498.

[35] Liu, J. S. (2001). *Monte Carlo Strategies In Scientific Computing*.

[36] Martin, J., Wilcox, L. C., Burstedde, C., and Ghattas, O. (2012). A stochastic newton mcmc method for large-scale statistical inverse problems with application to seismic inversion. *SIAM Journal on Scientific Computing*.

[37] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *Citation: J. Chem. Phys*, 21:1087. metro-hasting ref¡br/¿¡br/¿.

[38] Morzfeld, M., Tu, X., Atkins, E., and Chorin, A. J. (2012). A random map implementation of implicit filters. *Journal of Computational Physics*, 231:2049–2066. implicit filters, complex formation¡br¿.

[39] Morzfeld, M., Tu, X., Wilkening, J., and Chorin, A. J. (2015). Parameter estimation by implicit sampling. *Commun. Appl. Math. Comput. Sci.*, 10:205–225.

[40] Neubauer, A. (2009). On enhanced convergence rates for tikhonov regularization of nonlinear ill-posed problems in banach spaces. *Inverse Problems*, 25(6):065009.

[41] Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, New York, NY.

[42] Oliver, D. S. (2016). Metropolized randomized maximum likelihood for sampling from multimodal distributions.

[43] Owen, A. B. (2013). *Monte Carlo theory, methods and examples*.

[44] Paul, S., Mondal, P., and Bhattacharya, P. (2016). Numerical solution of lotka volterra prey predator model by using rungeâ€“kuttaâ€“fehlberg method and laplace adomian decomposition method.

[45] Robert, C. P. and Casella, G. (2004). *Controling Monte Carlo Variance*. Springer New York, New York, NY.

[46] Roberts, G. O. and Rosenthal, J. S. (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16(4):351 – 367.

[47] Rue, H. and Held, L. (2005). *Gaussian Markov Random Fields: Theory And Applications (Monographs on Statistics and Applied Probability)*. Chapman and Hall/CRC.

[48] Scheichl, R., Stuart, A. M., and Teckentrup, A. L. (2017). Quasi-monte carlo and multilevel monte carlo methods for computing posterior expectations in elliptic inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):493–518.

[49] Stuart, A. M. (2010). Inverse problems: A bayesian perspective. *Acta Numerica*, 19:451–559.

[50] Tarantola, A. (2005). *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics.

[51] Wang, J. and Zabaras, N. (2004). Hierarchical bayesian models for inverse problems in heat conduction. *Inverse Problems*, 21(1):183–206.