Transport maps for accelerated Bayesian computation

by

Matthew David Parno

S.M. Computation for Design and Optimization,
Massachusetts Institute of Technology (2011)

B.S. Applied Mathematics and Statistics,
B.S. Electrical Engineering,
Clarkson University (2009)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computational Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2015

Author .....
Department of Aeronautics and Astronautics
October 29, 2014

Certified by .....
Youssef Marzouk
Associate Professor of Aeronautics and Astronautics, Thesis Advisor

Certified by .....
Karen Willcox
Professor of Aeronautics and Astronautics, Committee Member

Certified by .....
Dennis McLaughlin
Professor of Water Resources, Committee Member

Accepted by .....
Paulo Lozano
Chairman of Department Committee on Graduate Theses

Transport maps for accelerated Bayesian computation

by

Matthew David Parno

Submitted to the Department of Aeronautics and Astronautics
on November 5, 2014, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computational Science and Engineering

Abstract

Bayesian inference provides a probabilistic framework for combining prior knowledge with mathematical models and observational data. Characterizing a Bayesian posterior probability distribution can be a computationally challenging undertaking, however, particularly when evaluations of the posterior density are expensive and when the posterior has complex non-Gaussian structure. This thesis addresses these challenges by developing new approaches for both exact and approximate posterior sampling. In particular, we make use of deterministic couplings between random variables—i.e., *transport maps*—to accelerate posterior exploration.

Transport maps are deterministic transformations between (probability) measures. We introduce new algorithms that exploit these transformations as a fundamental tool for Bayesian inference. At the core of our approach is an efficient method for constructing transport maps using only samples of a target distribution, via the solution of a convex optimization problem. We first demonstrate the computational efficiency and accuracy of this method, exploring various parameterizations of the transport map, on target distributions of low-to-moderate dimension. Then we introduce an approach that composes sparsely parameterized transport maps with rotations of the parameter space, and demonstrate successful scaling to much higher dimensional target distributions. With these building blocks in place, we introduce three new posterior sampling algorithms.

First is an adaptive Markov chain Monte Carlo (MCMC) algorithm that uses a transport map to define an efficient proposal mechanism. We prove that this algorithm is ergodic for the exact target distribution and demonstrate it on a range of parameter inference problems, showing multiple order-of-magnitude speedups over current state-of-the-art MCMC techniques, as measured by the number of effectively independent samples produced per model evaluation and per unit of wall clock time.

Second, we introduce an algorithm for inference in large-scale inverse problems with *multiscale* structure. Multiscale structure is expressed as a conditional independence relationship that is naturally induced by many multiscale methods for the solution of partial differential equations, such as the multiscale finite element method (MsFEM). Our algorithm exploits the offline construction of transport maps that represent the joint distribution of coarse and fine-scale parameters. We evaluate the accuracy of our approach via comparison to single-scale MCMC on a 100-dimensional problem, then demonstrate the algorithm on an inverse problem from flow in porous

media that has over 10^5 spatially distributed parameters.

Our last algorithm uses offline computation to construct a transport map representation of the joint data-parameter distribution that allows for efficient conditioning on data. The resulting algorithm has two key attributes: first, it can be viewed as a "likelihood-free" approximate Bayesian computation (ABC) approach, in that it only requires samples, rather than evaluations, of the likelihood function. Second, it is designed for approximate inference in near-real-time. We evaluate the efficiency and accuracy of the method, with demonstration on a nonlinear parameter inference problem where excellent posterior approximations can be obtained in two orders of magnitude less online time than a standard MCMC sampler.

Thesis Supervisor: Youssef Marzouk
Title: Associate Professor of Aeronautics and Astronautics

Acknowledgments

While I am getting the credit (and the degree), this thesis was only possible because of a massive collaborative effort by many people in my life. While I cannot thank everyone, I would like to at least thank:

My wife, Julie.
Julie has been with me through it all: the good, the bad, and the past few ugly months. She keeps me grounded, makes me run, and keeps me fed with the most amazing baked goods anywhere. We do almost everything together, and this thesis is no different; I could not have done it without her.

My parents.
My parents not only set me on the right track, but they have kept me there as well. Whether I'm lost because I'm driving around downtown San Francisco without a map, or I'm lost because I don't know where I want to go in life, my parents are always there, always helpful, and always stocked with whales, cheese, and juniper.

My advisor, Youssef Marzouk.
Youssef has been an incredibly flexible and supportive influence through the entire Ph.D. process. In more ways than I can list here, he has enabled me to be my best, both in and out of school. I am confident that with any other advisor, I would not have been able to reach this point and would not have been able to hand in this thesis.

The weasels.
After running with these guys, the whole Ph.D. process doesn't seem so bad.

My committee and readers
In addition to Youssef Marzouk, Karen Willcox and Dennis McLaughlin have made up my committee for the the last few years. They have given me great advice and have been a pleasure to work with. I would also like to thank Bart Van Bloemen Waanders and Mark Girolami for reviewing this thesis. Bart has also been a great external advisor of sorts and I am glad to have been able to work (and occasionally run) with him over the past few years.

In addition to the people above, I need to thank the Department of Energy for funding much of my Ph.D. through the DOE Office of Science Graduate Fellowship (SCGF).

And now, "The mountains are calling and I must go." *–John Muir*

Contents

1 Introduction and background **15**
- 1.1 A qualitative Bayesian example . . . 15
- 1.2 Bayesian inference . . . 18
- 1.3 Sampling methods . . . 20
  - 1.3.1 Markov chain Monte Carlo . . . 20
  - 1.3.2 Approximate Bayesian computation . . . 22
- 1.4 A qualitative overview of transport maps . . . 23
- 1.5 Thesis contributions . . . 27

2 Constructing transport maps from samples **29**
- 2.1 Transport map overview . . . 29
- 2.2 Constructing maps from samples . . . 32
  - 2.2.1 Formulation . . . 32
  - 2.2.2 Enforcing monotonicity . . . 34
  - 2.2.3 Simplifications of the KL cost function . . . 36
- 2.3 Transport map parameterization . . . 37
  - 2.3.1 Multivariate polynomials . . . 37
  - 2.3.2 Radial basis functions . . . 38
  - 2.3.3 Choosing map ordering . . . 40
- 2.4 Solving the optimization problem . . . 42
- 2.5 Approximating the inverse map . . . 43
- 2.6 Numerical example . . . 44
- 2.7 Constructing high-dimensional maps . . . 46
  - 2.7.1 Compositional map overview . . . 48
  - 2.7.2 Error of composed maps . . . 49
  - 2.7.3 Monitoring convergence . . . 52
  - 2.7.4 Choosing rotations . . . 52
  - 2.7.5 Relationship to artificial neural networks . . . 57
  - 2.7.6 Numerical examples: layered maps . . . 57
  - 2.7.7 Summary . . . 65

3 Multiscale inference with transport maps **69**
- 3.1 Introduction . . . 69
  - 3.1.1 Overview of multiscale samplers . . . 70
  - 3.1.2 Multiscale definition . . . 70

3.2 Multiscale framework . . . 72
3.2.1 Decoupling the scales with conditional independence . . . 72
3.2.2 Two stages for multiscale inference . . . 73
3.3 Transport maps for multiscale inference . . . 73
3.3.1 Theoretical framework . . . 73
3.3.2 Constructing the maps . . . 75
3.3.3 Choosing the number of fine scale samples . . . 75
3.4 A proof-of-concept example . . . 78
3.5 Application: simple groundwater flow . . . 80
3.5.1 Defining the coarse parameter with the multiscale finite element method (MsFEM) . . . 82
3.5.2 Multiscale framework applied to MsFEM . . . 84
3.6 Numerical results . . . 88
3.6.1 One spatial dimension . . . 88
3.6.2 Two spatial dimensions . . . 93
3.7 Discussion . . . 95

4 Transport map accelerated MCMC 99
4.1 Transport-map accelerated MCMC . . . 100
4.1.1 The Metropolis-Hastings Rule . . . 100
4.1.2 Combining maps and MCMC . . . 101
4.2 Adaptive transport-map MCMC . . . 104
4.2.1 Adaptive algorithm overview . . . 104
4.2.2 Complexity of map update . . . 105
4.2.3 Monitoring map convergence . . . 106
4.2.4 Choice of reference proposal . . . 106
4.3 Relationship to geodesic MCMC . . . 109
4.4 Convergence analysis . . . 110
4.4.1 The need for bounded derivatives . . . 110
4.4.2 Convergence of adaptive algorithm . . . 111
4.5 Numerical examples . . . 114
4.5.1 German credit logistic regression . . . 116
4.5.2 Biochemical oxygen demand model . . . 118
4.5.3 Predator prey system . . . 119
4.6 Conclusions . . . 126

5 Transport maps for fast approximate conditional sampling 129
5.1 A new starting point: the joint distribution . . . 130
5.2 An illustrative example . . . 132
5.3 Layered construction of block lower triangular maps . . . 133
5.3.1 Layered block formulation . . . 133
5.3.2 Defining a compositional inference map . . . 136
5.3.3 Choosing block rotations . . . 136
5.4 Application to biochemical oxygen demand inference . . . 137
5.4.1 Accuracy . . . 138

5.4.2 Timing . . . 140
5.5 Discussion . . . 141

6 MUQ: A software framework for uncertainty quantification 145
6.1 Structured modeling . . . 146
6.1.1 Existing algorithm-model interfaces . . . 146
6.1.2 MUQ's modeling framework . . . 147
6.2 Implementing Markov chain Monte Carlo . . . 148
6.3 Conclusions . . . 149

7 Conclusions and future work 151

A Detailed MCMC convergence analysis 155
A.1 Setting the stage: bounding the target proposal . . . 155
A.2 SSAGE . . . 156
A.3 Minorization . . . 157
A.4 Drift . . . 159

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

1-1 Bill's Utility 16
1-2 Bill's Prior Belief 16
1-3 Bill's Posterior Belief 17
1-4 Illustration of simple rejection ABC algorithm. 23
1-5 Optimal dirt transportation 24
1-6 Suboptimal dirt transportation 25
1-7 Dirt transportation with spreading 26
1-8 Frank's construction choices. 27

2-1 Illustration of exact and inexact transport maps. 31
2-2 Ensuring bounded map derivatives with linear correction. 35
2-3 Visualization of multi-index sets. 39
2-4 Typical objective function in map construction. 43
2-5 Convergence of diagonal maps defined by $\mathcal{J}_d^{NC}$. 46
2-6 Convergence of nonlinear separable maps defined by $\mathcal{J}_d^{NM}$. 47
2-7 Convergence of total order maps maps defined by $\mathcal{J}_d^{TO}$. 47
2-8 The goal of map composition. 49
2-9 Typical convergence of layered map with random rotations. 60
2-10 Typical convergence of layered map with choosy random rotations. . . 61
2-11 Typical convergence of layered map with principal component rotations. 62
2-12 Typical convergence of layered map with alternating SVD with random rotations. 63
2-13 Typical convergence of layered map, alternating SVD with choosy rotations. 64
2-14 Comparision of random field realizations for Besov example. 66
2-15 Map-induced marginal comparison on Besov example. 67

3-1 Optimal number of fine samples per coarse samples. 79
3-2 Convergence of multiscale posterior on a toy example. 81
3-3 Typical MsFEM basis function. 82
3-4 Comparison of multiscale posterior and MCMC gold-standard. 91
3-5 Posterior summary in one dimensional example. 92
3-6 Verification of coarse map accuracy in two dimensional example. . . . 95
3-7 Posterior summary in two dimensional example. 97

4-1 MCMC proposal process using transport maps. 102

4-2 Illustration of map-induced proposals. 107
4-3 Comparison of maps and geodesics. 110
4-4 Comparison of BOD posterior and map-preconditioned density. . . . 120
4-5 MCMC trace plots on BOD problem. 121
4-6 Autocorrelations of MCMC chains on BOD problem. 122
4-7 Predator prey posterior distribution 124
4-8 Comparison of chain autocorrelation on predator prey problem. . . . 125

5-1 Obtaining conditional density from joint density. 131
5-2 Convergence of approximate conditional density. 133
5-3 Illustration of block composed map evaluation. 135
5-4 Approximate BOD posterior densities constructed with 5000 samples. 139
5-5 Approximate BOD posterior densities constructed with 50000 samples. 140
5-6 Approximate BOD joint densities constructed with 5000 samples. . . 143
5-7 Approximate BOD joint densities constructed with 50000 samples. . . 144

6-1 Illustration of graphical modeling approach. MUQ uses graphs like this to define models, compute derivatives with the chain rule, and extract additional problem structure. 147

List of Tables

2.1 Timing comparison of polynomial maps on a simple two-dimensional example. 45
2.2 Accuracy of $\tilde{T}$ on a simple example. 46
2.3 Convergence of compositional map. 59

3.1 Error in posterior reconstruction for one dimensional example. 90
3.2 Posterior sampling efficiency comparison on one dimensional example. 93

4.1 Summary of standard MCMC samplers 115
4.2 Summary of map-accelerated MCMC variants. 115
4.3 MCMC performance comparison of German credit problem. 117
4.4 Comparison of MCMC performance on BOD problem. 118
4.5 Predator prey MCMC performance comparison 123

5.1 Accuracy of offline inference maps on BOD problem. 138
5.2 Efficiency of offline inference maps on the BOD problem. 141

THIS PAGE INTENTIONALLY LEFT BLANK

List of Algorithms

1.1 Simple rejection-based ABC algorithm. 23

2.1 Insertion sort of map dimensions. 42
2.2 Generating random orthonormal matrices. 55
2.3 Tuning random orthonormal matrices to capture non-Gaussianity. . . . 56

3.1 Overview of the entire multiscale inference framework. 76

4.1 MCMC algorithm with fixed map. 103
4.2 MCMC algorithm with adaptive map. 105

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction and background

Scientists and engineers use mathematics as a convenient language for describing our belief of how the world *should* behave. On the other hand, we have reality – how the world *actually* behaves. Ensuring that our mathematical descriptions agree with observations of reality is important when using mathematics as a tool to make predictions and decisions. This fundamental problem of calibrating a mathematical description (a mathematical model) to observations is generally called an *inverse problem*. We will formulate inverse problems with probabilistic representations of what we believe *should* happen and what *actually* happens. This corresponds to formulating the inverse problem as a Bayesian inference problem, where beliefs are represented as probability distributions. Our goal in this work is to develop new and more efficient ways for characterizing probability distributions which arise in the solution of Bayesian inference problems. In particular, we will use deterministic transformations called transport maps to develop efficient posterior sampling algorithms.

1.1 A qualitative Bayesian example

To introduce many of the ideas of Bayesian analysis at a high level, we begin by analyzing a simple question: should our friend Bill decide to go on a run this afternoon? Note that *we* are not trying to decide if we think Bill should run or not. Instead, we are trying to analyze Bill's own decision making process.

We assume Bill's decision to run or not this afternoon is based primarily on how hard it is going to rain this afternoon. If it is not raining, or only sprinkling, Bill is going to enjoy his run. On the other hand, if there is a steady drizzle, Bill is not going to enjoy his run as much. Oddly, for some unexplained reason, our friend Bill also enjoys running in heavy downpours. This means that if the rain becomes harder than a drizzle, Bill will again enjoy his run. These features of Bill's enjoyment are shown graphically in Figure 1-1.

In Bayesian analysis, the function shown in Figure 1-1 is called the utility function and summarizes all the information needed for Bill to make a decision. In more sophisticated situations, the utility function may represent a company's balance of profit and environmental risk, or some other compromise. In any case, utility func-

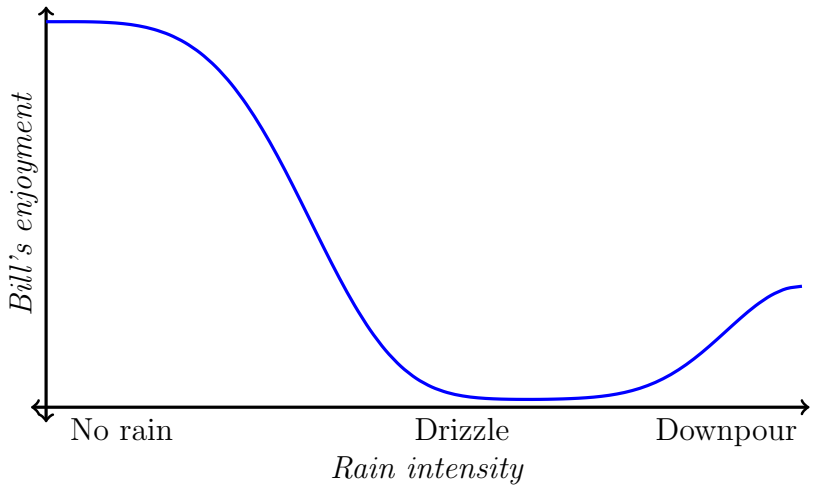


Figure 1-1: Bill utility function. Bill enjoys running when he can stay dry or get completely soaked. However, one of his quirks is that he will not run in a drizzle.

tions are related to specific decisions or questions just like "Will Bill decide to go on a run this afternoon?"

From the utility function in Figure 1-1, we can see that Bill would decide to go on a run if he definitely knew it was going to be sunny or if he knew that it was definitely going to downpour. However, Bill (or anyone for that matter) does not know exactly how hard it is going to rain this afternoon. There might be no rain, or a storm might develop and bring intense rain. Similar to his utility function, Bill's belief about this afternoon's rain is given graphically in Figure 1-2. The spike on the left represents Bill's belief that it will not rain while the bump of high intensity rain indicates Bill's belief that a storm might develop this afternoon.

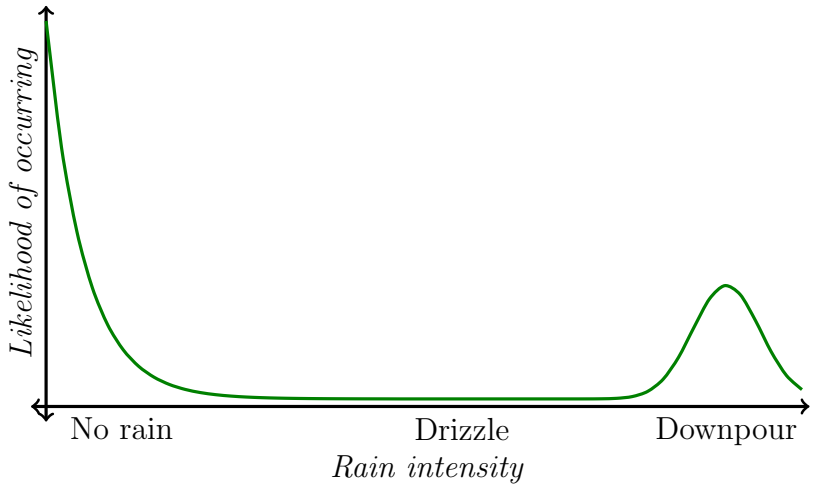


Figure 1-2: Bill's prior beliefs. It looks sunny outside, but rumor has it that there may be a storm developing this afternoon. Therefore, before checking the radar, Bill believes it will either not rain or downpour.

Now that Bill has a representation of his possible enjoyment (via the utility function) and a characterization of his uncertainty in this afternoon's rain, he can make a decision. To make the decision, Bill would implicitly weight the utility function with his belief in the rain intensity and then take an average. The higher the average, the more likely it is that Bill will enjoy his run. Because Bill likes to run when it is either lightly raining or down pouring, and he believes it will either rain lightly or down pour this afternoon, his expected enjoyment would be high and he would decide to head out for a run this afternoon. However, suppose that as Bill is putting on his running shoes, he checks the radar and sees that it is more likely to drizzle! This changes his beliefs about the rain intensity and therefore makes him question his decision to run.

After seeing the radar, Bill's belief might change to something like that shown in Figure 1-3. Comparing Figure 1-3 with his utility function in Figure 1-1 and recomputing his expected enjoyment, Bill decides that he is not likely to enjoy his run. He therefore chooses to unlace his shoes and takes a nap.

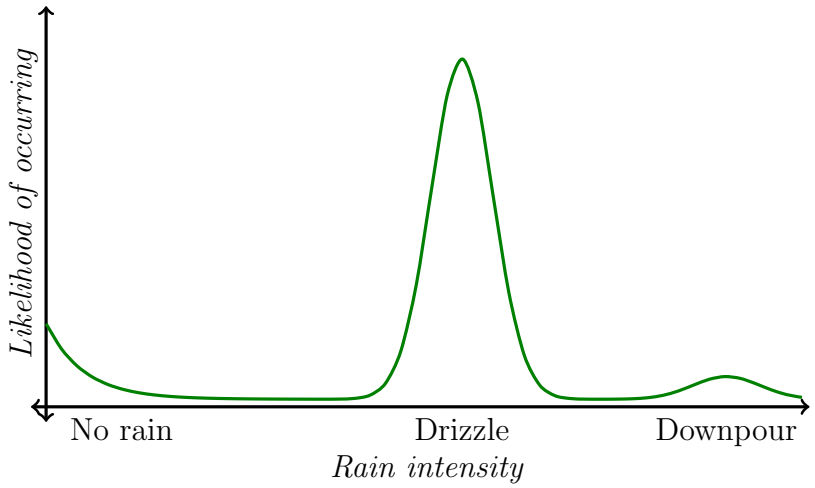


Figure 1-3: Bill's posterior beliefs. After observing the radar, Bill thinks that is more likely for a drizzle to occur.

The process of combining Bill's prior belief in Figure 1-2 with the radar observation to produce his posterior belief in Figure 1-3 is an application of Bayesian inference. Bayesian inference provides a probabilistic mechanism for this type of knowledge integration. The remainder of this thesis develops efficient tools for solving Bayesian inference problems. Our goal is to *efficiently* combine prior beliefs, mathematical forward models, and observations into a complete probabilistic description such as Figure 1-2. Like Bill's combination of belief and utility, the output of our tools provide all the information needed to make well informed decisions that take into account the many sources of uncertainty in our beliefs.

1.2 Bayesian inference

As alluded to in section 1.1, probability distributions provide a way of representing one's degree of belief about some parameter θ. To formalize this concept mathematically, let θ be a random variable defined on the probability space (Ω, σ, P), where Ω is the sample space, σ is a sigma algebra on Ω,[1] and $P : \sigma \to [0, 1]$ is a probability measure that assigns a probability to every set in σ. Recall that a random variable is a function defined over Ω, i.e., $\theta : \Omega \to \mathcal{X}_\theta$ where $\mathcal{X}_\theta$ is the set of possible values of θ. In all of our applications, random variables will be real valued, so we have $\mathcal{X}_\theta \subseteq \mathbb{R}^{D_\theta}$, where D_θ is the dimension of θ. Notice that we can also define a sigma algebra $\mathcal{F}_\theta$ on $\mathcal{X}_\theta$, and a probability measure $\mu_\theta : \mathcal{F}_\theta \to [0, 1]$. The measure μ_θ is called the *distribution* of the random variable θ. Collecting $\mathcal{X}_\theta$, $\mathcal{F}_\theta$, and μ_θ, we will say that the random variable θ corresponds to the probability space $(\mathcal{X}_\theta, \mathcal{F}_\theta, \mu_\theta)$. Notice that for any set $A \in \mathcal{F}_\theta$, the probability density $\pi(\theta)$ is related to the distribution μ_θ by

$$\mu_\theta(A) = \int_A \pi(\theta)\lambda(d\theta), \tag{1.1}$$

where $\lambda(\cdot)$ is a reference measure (often the Lebesgue measure), and $\lambda(d\theta)$ is the measure of the infinitesimal $d\theta$. In this thesis, we will almost always assume that μ_θ admits a density with respect to Lebesgue measure.

In the Bayesian context, the density $\pi(\theta)$ represents an a priori belief about the random variable θ.[2] Observations are represented by another random variable d and the conditional density $\pi(\theta|d)$ represents the a posteriori knowledge of the parameters. Unfortunately, we cannot usually evaluate or sample the density $\pi(\theta|d)$ directly. However, Bayes' rule expands the posterior density into a more useful form. Bayes' rule can be simply derived from the law of total probability[3] to obtain

$$\pi(\theta|d) = \frac{\pi(d, \theta)}{\pi(d)} = \frac{\pi(d|\theta)\pi(\theta)}{\pi(d)} \propto \pi(d|\theta)\pi(\theta), \tag{1.2}$$

[1] Recall that a σ-algebra is a collection of subsets of the sample space Ω that is closed under a countable number of set operations. For more information, chapter 2 of [92] provides a nice introduction to σ-algebras and rigorous probability theory.

[2] In some situations, people will claim that $\pi(\theta)$ represents all prior knowledge. In my view, this claim is incorrect. The model itself and the form of the error model is a form of prior information that is embedded into the likelihood function. Thus, $\pi(\theta)$ represents all prior information given the particular structure of the likelihood. Model selection is beyond the scope of this thesis and we will thus continue to follow the classic presentation of Bayes' rule. However, readers should be aware that a more rigorous presentation of Bayes' rule in our context would include the model parameterization of the posterior: $\pi(\theta|d; M) = \pi(d|\theta; M)\pi(\theta; M)/\pi(d; M)$. In fact, in [56], Jaynes goes a step further, using $\pi(\theta|d; I) = \pi(d|\theta; I)\pi(\theta; I)/\pi(d; I)$ where I represents all prior modeling assumptions, human bias, etc.

[3] Bayes' rule was originally introduced by Reverend Thomas Bayes in the mid 18th century. The original formulation was for the Binomial distribution but was later generalized into its modern form around 1800 by Pierre-Simon Laplace and Carl Friedrich Gauss as a mechanism for combining past information with current observations to learn about parameters of a statistical model. See [44] for a thorough history of Bayesian inference.

where $\pi(\theta)$ is the prior probability density, $\pi(d|\theta)$ is called the likelihood function, and $\pi(d) = \int_{\mathcal{X}_\theta} \pi(d|\theta)\pi(\theta)\lambda(d\theta)$ is a normalization constant called the evidence. Using a predictive model taking θ as an input, the likelihood function compares model predictions to data for a particular value of θ.

Consider again Bill's decision to run or not to run. In this example, $\pi(\theta)$ is representative of Bill's belief in the rain intensity *before* looking at the radar. Assuming Bill is both a runner and a Bayesian statistician, he would use $\pi(\theta)$ and his utility function $h(\theta)$ to express his a priori expected enjoyment as

$$E_{\text{prior}} = \int_{-\infty}^{\infty} h(\theta)\pi(\theta)\lambda(d\theta). \tag{1.3}$$

This is simply the mathematical form of combining Figures 1-1 and 1-2. Now let d denote the radar observations. After observing d, Bill's belief about the rain is represented by a new a posteriori density $\pi(\theta|d)$, which corresponds to Figure 1-3. Bill's expected enjoyment is now given by

$$E_{\text{post}} = \int_{-\infty}^{\infty} h(\theta)\pi(\theta|d)\lambda(d\theta). \tag{1.4}$$

The only thing that has changed between (1.3) and (1.4) is Bill's belief in the rain intensity. Unfortunately, evaluating the posterior integral (1.4) can be quite difficult because no general analytic forms exists for Bayesian posteriors. Moreover, the evidence $\pi(d)$ cannot generally be computed efficiently, so methods for characterizing $\pi(\theta|d)$ and subsequently computing (1.4) need to rely solely on either evaluations of $\pi(d|\theta)\pi(\theta)$ or on prior samples of $\pi(d|\theta)\pi(\theta)$.

Existing methods that only use the numerator of (1.2) to characterize the posterior $\pi(\theta|d)$ can be broadly separated into two groups. The first group contains methods that represent the posterior with samples. Typical examples of this group are Markov chain Monte Carlo methods and importance sampling methods. Importantly, these methods yield *exact* posterior estimates as the number of samples grow to infinity. The second group contains variational Bayesian methods that build variational approximations to the posterior density itself [54]. The accuracy of these method is limited by the class of approximating distributions and how well the approximating distributions match the target distribution. The focus of this work will be on efficient sampling strategies because of their flexibility and robustness. Moreover, properly defined sampling methods can exactly represent any posterior distribution.

After some necessary background in the remainder of this chapter, we will develop three methods for efficiently sampling $\pi(\theta|d)$ in chapters 3, 4, and 5. All three of these approaches utilize a new method for constructing transport maps from samples, which we first introduce in Chapter 2.

1.3 Sampling methods

One approach to approximate (1.3) or (1.4) is with Monte Carlo integration. Monte Carlo strategies rely on numerous realizations of a random variable to approximate the expectations with a finite sum. However, generating the necessary samples from arbitrary densities, such as $\pi(\theta|d)$, is not a trivial task. Variable transformations provide one avenue for this sampling. For example, sampling a one dimensional random variable with analytically defined inverse cumulative distribution functions (CDF) is as easy as evaluating the inverse CDF. This is a simple example of a one dimensional random variable transformation. More general variable transformations, such as the Knothe-Rosenblatt transform, can be constructed in higher dimensions [91, 102], and will serve as a fundamental component of all new methods we develop. However, exact variable transformations are not always available and other indirect methods are required. Indirect methods build upon easily sampled distributions (e.g., Gaussian, uniform, etc...) to generate samples from more difficult densities such as $\pi(\theta|d)$.

An abundance of algorithms have been developed that combine an analytically tractable proposal distribution with a correction step to generate samples (perhaps weighted) of the target random variable. Important examples include importance sampling, rejection sampling, and Markov chain Monte Carlo (MCMC) [68, 82]. Of these strategies, our work will focus on MCMC; however, the underlying concepts introduced in Chapter 4 can easily be passed on to methods with other correction techniques.

1.3.1 Markov chain Monte Carlo

Originally introduced in [76] by a group of Los Alamos and University of Chicago researchers in the context of statistical physics, MCMC methods construct an ergodic Markov chain in the parameter space $\mathcal{X}_\theta$ that has the desired target distribution as a stationary distribution. In the Bayesian setting, the target distribution is the posterior distribution $\mu_{\theta|d}$ defined for any $A \in \mathcal{F}_\theta$ by

$$\mu_{\theta|d}(A) = \int_A \pi(\theta|d)\lambda(d\theta). \tag{1.5}$$

After a sufficient number of steps, the states of the chain can be used as correlated samples of the target distribution. This relies on the fact that the Markov chain is ergodic and has the target distribution as a stationary distribution. The transition kernel of the chain at step t can depend on the current state, θ_t and is often defined by coupling a position dependent proposal distribution, $q(\theta|\theta_t)$, with an accept-reject corrective stage. To ensure convergence to the target distribution, a sample $\theta' \sim q(\theta|\theta_t)$ will be accepted as the next point in the chain, θ_{t+1}, with an acceptance probability given by

$$\alpha = \min\left\{1, \frac{\pi(\theta'|d)q(\theta_t|\theta')}{\pi(\theta_t|d)q(\theta'|\theta_t)}\right\}. \tag{1.6}$$

This expression is called the Metropolis-Hastings rule, and is the basis for almost all modern MCMC approaches. In [45], Hastings proposed this generalization of Metropolis's original acceptance rule in [76].

Unlike direct sampling approaches, MCMC produces *correlated* samples of the posterior[4] because the transition kernel depends on the current state of the chain. Intuitively, this correlation reduces the amount of "information" contained in the samples, and subsequently reduces the Monte Carlo integration accuracy.

Consider in more detail the impact of this correlation on a Monte Carlo estimate. Let $\hat{\bar{\theta}}_n$ be the estimate of $\bar{\theta} = \mathbb{E}_{\theta|d}[\theta]$ using n correlated samples: $\{\theta_0, \theta_1, \ldots, \theta_n\}$. Assuming the target distribution has finite variance and under some additional conditions on the Markov chain (e.g., the chain is geometrically ergodic [55, 84]), the Monte Carlo estimate will satisfy a central limit theorem and converge (see [59] for details) as

$$\sqrt{n}\left(\hat{\bar{\theta}}_n - \bar{\theta}\right) \to N\left(0, \sigma^2\right), \tag{1.7}$$

where

$$\sigma^2 = \underset{\theta|d}{\mathbb{V}\text{ar}}\left[\theta\right] + 2\sum_{i=1}^{\infty} \mathbb{C}\text{ov}\left[\theta_0, \theta_i\right]. \tag{1.8}$$

The inter-sample covariance $\mathbb{C}\text{ov}_{\theta|d}\left[\theta_0, \theta_i\right]$ describes the correlation between states in the chain. It is important to realize that this covariance is a function of the chain itself, and only indirectly depends on the target density. In fact, this covariance qualitatively captures the average difference between subsequent states in the chain.

Notice that the inter-sample covariance $\mathbb{C}\text{ov}_{\theta|d}\left[\theta_0, \theta_i\right]$ increases the variance of the Monte Carlo estimate. Since our goal is to obtain an accurate estimate of $\bar{\theta}$ by reducing the Monte Carlo variance, with larger inter-sample correlations, we will need a longer chain to obtain an accurate estimate $\hat{\bar{\theta}}$. Longer chains mean more computational effort, which may prove intractable when posterior evaluations require expensive model solves. To overcome this, efficient MCMC schemes must reduce the inter-sample correlation. This correlation occurs for two reasons:

- The Metropolis-Hastings correction rejects many proposals because α is near zero.
- The proposed moves are small deviations from the current state, i.e., $\|\theta_{t+1} - \theta_t\|$ is small.

In attempts to alleviate both of these issues, modern MCMC algorithms try to construct proposal mechanisms that (at least locally) mimic the posterior. This ensures that large proposal steps will have a high acceptance probability and subsequently reduce inter-sample correlation. Most strategies to construct such proposal mechanisms use either on-line proposal adaption [41, 5], maintain multiple points in the

[4] In this context, we are discussing the correlation between states in the MCMC chain, not the correlation between components of θ. It may be useful here to think of the entire MCMC chain as a random variable, where one MCMC run produces one sample of the chain random variable.

state space [103, 23], utilize higher order derivative information [39, 73, 20], or combine adaptation with higher order information [72, 9]. There has also been work on problems where θ is high dimensional and represents the discretization of an infinite dimensional random field [43, 25, 65]; however, this work is not applicable to general target distributions and we will not discuss this topic in detail.

1.3.2 Approximate Bayesian computation

For some problems, especially in applications with a large number of latent or hidden variables, it can be *inexpensive* to draw a sample from the likelihood $\pi(d|\theta)$, but *expensive* to evaluate the density. These problems motivated the development of approximate Bayesian computation (ABC) methods, also known as likelihood-free methods, that can approximately solve Bayesian inference problems without evaluating the likelihood. In order to sample the posterior, a very basic ABC algorithm would first generate a sample θ' from $\pi(\theta)$ or some other distribution, then use θ' to generate a sample $d' \sim \pi(d|\theta')$, and finally compare d' to the observed data. The sample will be rejected if there is a large discrepancy between d' and observations. Discrepancy is measured by the difference between some statistics of d and d', e.g., $\rho(d, d') = \|\psi(d) - \psi(d')\|$ where $\psi(d)$ is a sufficient statistic for d. In the simplest case, $\psi(d)$ is simply d itself. Different ABC algorithms will generate the initial sample θ' in difference ways. Basic algorithms simply use the prior while more sophisticated algorithms use techniques such as MCMC [71] or sequential Monte Carlo [95]. Using more sophisticated proposals for θ' reduces the number of rejected samples and increases sampling efficiency. See [27] or [70] for a more complete review of ABC variants.

As Algorithm 1 shows, the basic ABC algorithm can be written in four steps. This simple algorithm provides an intuitive view of Bayesian inference: Bayesian inference creates a "slice" of the joint distribution of (d, θ) by rejecting samples that do not satisfy $d = d'$. To see this more clearly, consider the basic rejection-based algorithm for a problem with $\theta \in \mathbb{R}^2$ and $d \in \mathbb{R}$. Figure 1-4 shows joint samples of (d, θ) and what is left after rejecting samples that disagree with the data. This process becomes exact, although increasingly inefficient, as the tolerance ϵ in Algorithm 1 goes to zero. As mentioned above, more efficient ABC methods use MCMC or importance sampling ideas to steer the proposal towards regions where $d \approx d'$.

ABC methods are often called likelihood-free methods because they do not require evaluations of the likelihood function. In Chapter 5 we will introduce a new likelihood-free inference methodology that, like ABC, uses only samples of $\pi(d, \theta)$ and not likelihood evaluations. However, our approach differs from ABC in two ways: (i) we do not require the observation of d for most of our computation, and (ii) once an initial set of joint samples is generated, we do not even require the model to generate samples of an approximate posterior. In this sense, our new method can be seen as a model-free extension of classic likelihood free algorithms.

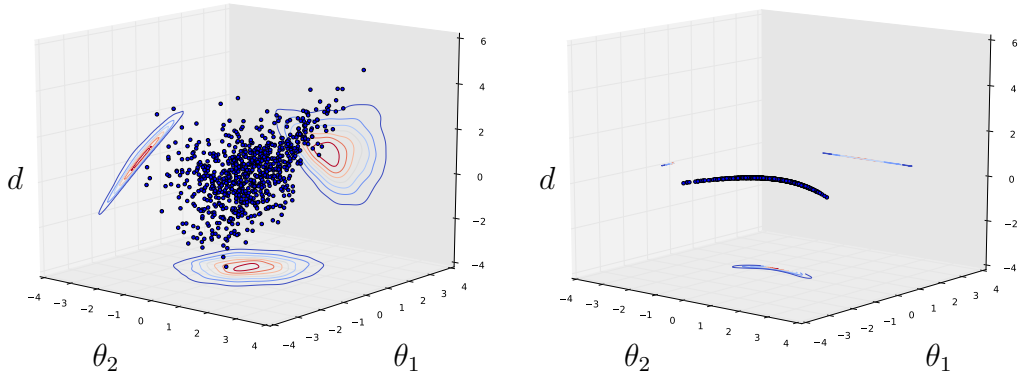
Being model-free also allows us to decompose the posterior sampling into two stages. The first stage is a computational expensive offline stage where we generate joint prior samples of θ and d in order to construct a transport map. This offline stage is followed by an online stage where we use the transport map to rapidly generate

Algorithm 1.1: Simple rejection-based ABC algorithm.

Input: Distance function $\rho(d, d')$, tolerance ϵ, Number of samples N_s
Result: Approximate samples, $\{\theta_1, \theta_2, \ldots, \theta_{N_s}\}$, of the posterior distribution $\pi(\theta|d)$

```
while t < N_s do
    Generate θ' from π(θ)
    Simulation d' from π(d'|θ)
    if ρ(d, d') < ε then
        Accept sample: θ_t = θ'
        t = t + 1
```

samples of $\pi(\theta|d)$ once d has been observed.



(a) Joint samples of $\pi(d, \theta)$ before rejection (b) Posterior samples approximating $\pi(\theta|d = 0)$

Figure 1-4: Illustration of the basic ABC algorithm. We first generate joint samples of θ and d as illustrated in Figure 1-4(a). Then, samples that do not agree with the data are rejected. In this problem, joint samples that do not satisfy $|d| < \epsilon = .0001$ are rejected. This leaves us with samples of the posterior, illustrated in Figure 1-4(b).

1.4 A qualitative overview of transport maps

In order to build upon existing methods such as MCMC and ABC and to make our inference algorithms efficient, we will use a mathematical tool called a transport map. Transport maps are functions that transform between probability distributions such as Bill's beliefs prior beliefs in Figure 1-2 and his posterior beliefs in 1-3. Like Bill's running example, this section uses a scenario to provide a high level description of transport maps.

Here we will discuss the exploits of a man named Tim[5], who has an eyesore of a

[5]We chose the name Tim for a reason. *T*im's actions will be analogous to our use of a transport

dirt pile in his backyard and will transfer the dirt to a pit in his neighbor's backyard. It may seem odd to talk about moving dirt in the context of probability theory, but this dirt transportation problem is intricately linked to transport maps. In fact, the original mathematical discussion of this problem by Gaspard Monge in 1781 also focused on such a dirt transportation problem [78].

Tim wants to move his dirt pile, but needs a place to put it. Fortunately, his neighbor Frank[6] has a large empty pit that is exactly the same size as the dirt pile. After some neighborly discussion, Frank has agreed to let Tim *transport* all the dirt from his pile to the pit. The top row of Figure 1-6 shows the pile of dirt and the nearby pit.

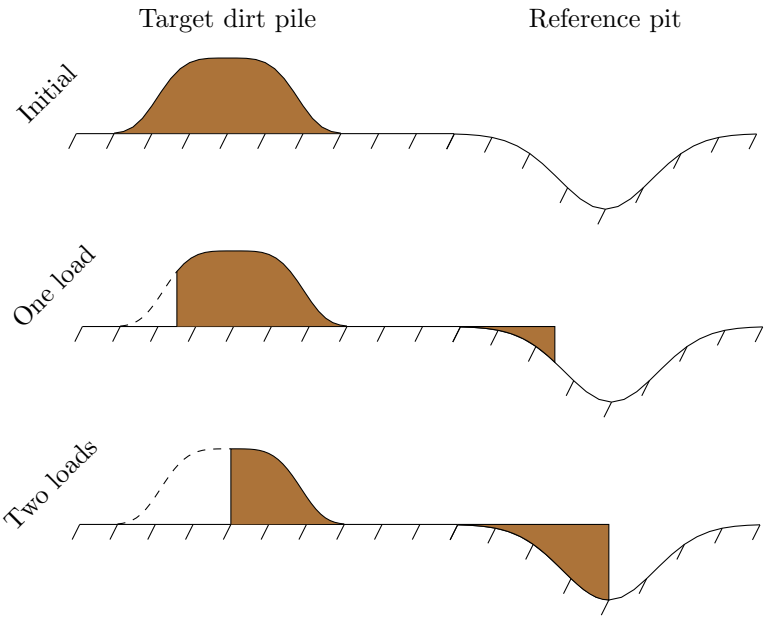


Figure 1-5: Tim's second attempt results in optimal dirt transportation when he cannot spread the dirt throughout the pit.

There are many ways for Tim to move all of the dirt from the pile to the pit. He could move all the dirt on the left of the pile to the right of the pit, he could do the opposite and move all the dirt on the right to the left, etc... Having multiple solutions means this problem is ill-posed. When using transport maps to transform between probability distributions, a similar problem arises: there are multiple ways to exactly transform one distribution into another. In both Tim's situation and the more general transport map context, some form of tie-breaking regularization is needed to find a unique solution. For Tim, the tie-breaker is to minimize the effort it takes to move the dirt pile. In other words, he wants to move the dirt as little as possible on average. This results in the solution shown in Figure 1-5. In this solution, Tim moves the dirt on the right of the pile to the right of the pit.

map denoted by T in later sections.

[6]The name Frank is also chosen for a reason. *F*rank's actions will be analogous to the action of the transport map F in subsequent sections.

Now suppose that Tim instead wants to expend as much effort as possible, perhaps he is trying to lose weight. This results in a different type of regularization and results in the dirt transfer shown in Figure 1-6. When Tim is trying to lose weight, he carries the dirt on the left side of the pile, as far as possible to the right side of the pit.

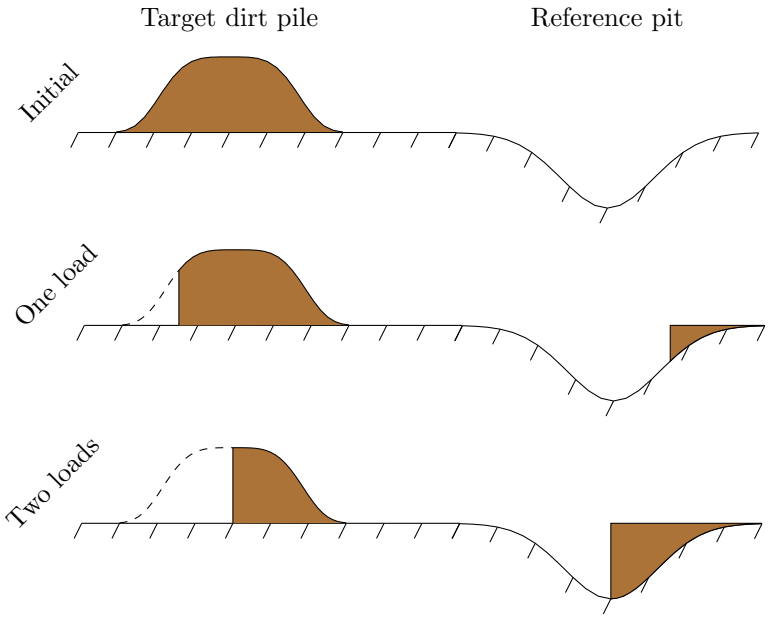


Figure 1-6: Tim's first attempt results in suboptimal dirt transportation.

The minimum effort solution in Figure 1-5 and the maximum effort solution Figure 1-6, illustrate that the best method of moving the dirt is dependent on the Tim's goals. In other words, the type of regularization dictates the form of the transformation. This is the same with transport maps; the form of the optimal transport map is dependent on a regularization cost and different cost functions result in different forms of the map. This fact is given a more mathematical treatment in Section 2.1.

Both the minimum and maximum effort solutions above are deterministic transformations, i.e., Tim took dirt from a small area of the pile and placed it in a small area of the pit. An alternative is for Tim to take a shovel-full of dirt, walk to the edge of the pit, and scatter the dirt randomly into the pit. This would result in the dirt transfer shown by Figure 1-7. In the field of optimal transport, this scattering type of transformation is a generalization of a transport map called a random coupling or random *transport plan*. General couplings are outside the scope of our work but interested readers can find the details of these transformations in [102].

Tim's dirt transportation issues are analogous to many of the issues we face with transport maps: (i) there are many ways that Tim can move the pile of dirt into the pit, (ii) the pile to pit transformation is exact, i.e., all the dirt from the pile makes it into the pit, and the pit does not change shape, and (iii) the optimal way of moving the dirt depends on Tim's ultimate goals. In the transport map context, these three issue remain, except that the piles of dirt correspond to probability densities. Chapter 2 will address these issues mathematically.

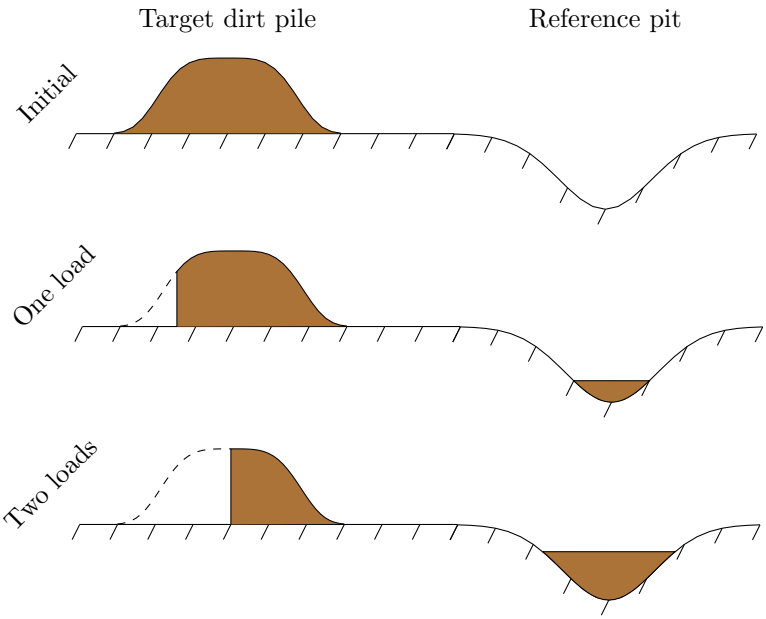
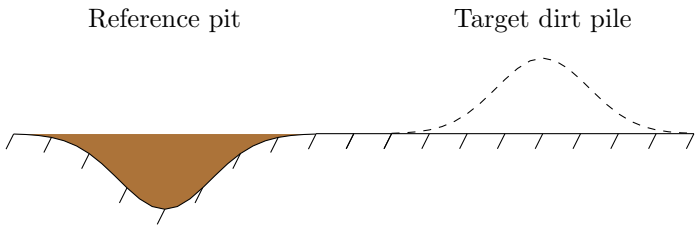


Figure 1-7: Tim can spread dirt through the pit by randomly scattering each load.

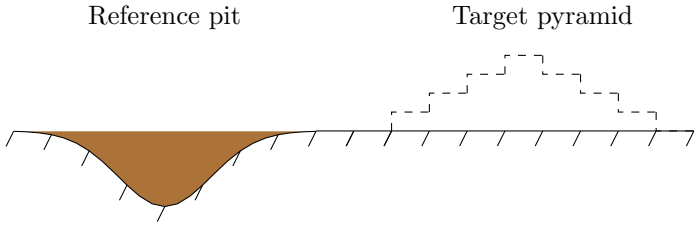
Now that Tim has filled the pit, we turn to his neighbor, and owner of the pit: Frank. Unfortunately, Frank is a very nostalgic guy, who now misses the pit he once thought was an eyesore. Frank therefore decides to reverse Tim's actions and empty the pit. But what should he do with the dirt? One option is to just pile it up without any structure as in Figure 1-8(a). However, an alternative is to build a pyramid using the dirt as in Figure 1-8(b). Frank will obviously need to work more carefully to construct a pyramid than to simply make another pile. More generally, the complexity of Frank's transformation will be proportional to how far his target is from the simple pile in Figure 1-8(a).

A similar problem exists in the transport map setting: it is more difficult to build a map to a complicated probability density than a simple density. More complicated target densities require more degrees of freedom in the map. For very complex densities, there may be too many degrees of freedom and it will not be feasible to construct an exact transformation. In that case, we are forced to live with an approximation. In this dirt transportation example, using an approximate map is like Frank building the approximate pyramid shown in Figure 1-8(c). While not exact, this coarse pyramid still captures much of the form Frank wanted in the original pyramid of Figure 1-8(b). Similarly, approximate transport maps often capture most of the structure we need to develop efficient methods. We will use this fact throughout Chapters 3, 4, and 5 to develop efficient methods for sampling Bayesian posterior distributions.

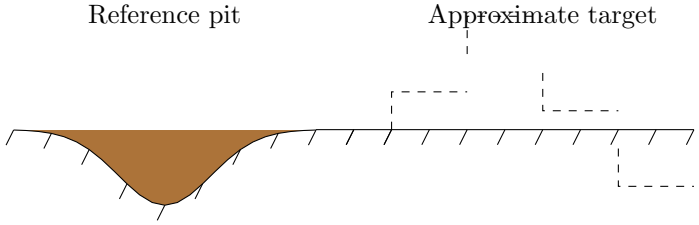
Notice that *T*im and *F*rank have opposing objectives. Tim wants to take the target pile of dirt and transform it to the reference pit. On the other hand, Frank wants to empty the reference pit and create a target pile. In chapter 2, the transport maps T and F will have similar purposes: T will transform a general target distribution to a Gaussian reference distribution, while F will transform the Gaussian reference into



(a) A simple choice for emptying the pit requires little effort.



(b) Building a pyramid with the dirt is more complicated and requires more effort.



(c) If Frank cannot construct the pyramid shown above, he may have to settle with an approximation like the one shown here.

Figure 1-8: Frank's construction choices.

the target distribution.

1.5 Thesis contributions

In Bayesian inference, exploring the posterior distribution $\pi(\theta|d)$ can be computationally intractable due to a few key challenges: (i) computationally expensive posterior evaluations; and (ii) high dimensional, strongly correlated, and non-Gaussian target distributions which are difficult to sample or otherwise characterize. The leading strategies for such problems include the many variants of Markov chain Monte Carlo (MCMC) [17] and, more recently, optimal transport maps [79].

MCMC methods use an approximation to the posterior and a corrective method to generate correlated samples of the posterior distribution. Unfortunately, local correlations make it difficult to obtain adequate approximations even when derivative information is available. This yields MCMC methods that require an unacceptable number of posterior evaluations to estimate posterior expectations. On the other hand, the optimal transport approach of [79] seeks a mapping that can that can push samples of a reference random variable to approximate samples of the posterior. In

many situations, this map is nearly exact and excellent performance can be achieved. However, using the map directly cannot ensure statistically exact sampling in cases where deterministic transformations of reasonable complexity cannot adequately approximate the posterior. In addition to these accuracy concerns, both the existing transport map approach and MCMC also require observations to be known before performing any computation. This precludes the use offline pre-observation computation, which could allow for near real-time post-observation posterior exploration. These inadequacies and fundamental challenges frame the main research objectives of this thesis:

1. To create a framework for approximate Bayesian inference that uses prior samples and extensive offline computation to enable fast Bayesian inference in the context of nonlinear inverse problems with computationally intensive forward models.

2. To rigorously formulate a computationally efficient, broadly applicable, and statistically exact sampling scheme for non-Gaussian Bayesian inference problems: in particular, a scheme that targets posterior distributions with varying local correlation structures while requiring no derivative information.

We will tackle the first objective in two ways, (i) with a framework for posterior sampling when the posterior exhibits multiscale features (Chapter 3), and (ii) by a new use of transport maps for approximate posterior sampling (Chapter 5). Both of these approaches use extensive offline computation; however, the multiscale approach is geared towards problems with large spatially-varying parameters while the direct use of transport maps is more applicable to smaller dimensional problems that require extremely fast post-observation sampling. To tackle the second objective, we will combine transport maps with the Metropolis-Hastings rule to create an adaptive MCMC algorithm (Chapter 4). All of these algorithmic developments require the efficient construction of transport maps. Chapter 2 therefore provides an extensive background on building transport maps from samples. All of the techniques and tests in this thesis rely on the MIT Uncertainty Quantification software library that we have developed. Thus, Chapter 6 provides a high level overview of the library.

Chapter 2

Constructing transport maps from samples

Like an automobile manufacturer that must build a fast engine before constructing a fast sports car, this chapter constructs the engine we need to develop fast algorithms in chapters 3–5. This chapter is devoted to techniques for efficiently constructing nonlinear transformations called transport maps. After an initial discussion of transport map basics in Section 2.1, Section 2.2 will introduce the general framework for constructing transport maps using samples of a target distribution. Sections 2.3 and 2.4 will then discuss more practical aspects of map construction: function parameterizations and efficient computational implementation. Section 2.5 will demonstrate how to approximate the inverse of a transport map, Section 2.6 will provide some initial numerical results, and Section 2.7 will show that compositions of maps can be used to construct transport maps in high dimensional space.

Note that the map construction techniques developed here are different from the original work of [79]. In that work, evaluations of the posterior density were used to construct a transformation from the prior distribution to the posterior distribution. However, in our approach, we use *samples* of an arbitrary distribution to build a transformation to a reference distribution; here the reference distribution is a standard Normal distribution. Our approach is more efficient when samples of a target distribution are available, while the methods in [79] are more useful when only density evaluations of the target are available.

2.1 Transport map overview

Here we will consider two random variables: the target and reference random variables. These random variables will be denoted by θ and r and will correspond to probability spaces $(\mathcal{X}, \mathcal{B}(\mathcal{X}), \mu_\theta)$ and $(\mathcal{Y}, \mathcal{B}(\mathcal{Y}), \mu_r)$, where $\mathcal{X} \subseteq \mathbb{R}^{D_\theta}$, $\mathcal{Y} \subseteq \mathbb{R}^{D_\theta}$, and $\mathcal{B}(\mathcal{A})$ denotes the Borel sets of $\mathcal{A}$. The probability measures μ_θ and μ_r correspond to the distributions of θ and r. A transport map is a nonlinear transformation from $\mathcal{X}$ to $\mathcal{Y}$ that "transports" the probability in μ_θ to μ_r. In fact, an *exact* transport map,

denoted by T, is an exact random variable transformation from θ to r given by,

$$r \overset{i.d.}{=} T(\theta), \tag{2.1}$$

where $\overset{i.d.}{=}$ denotes equality in distribution.

For our use of T as a variable transformation, we need this map to be invertible and have continuous derivatives. We will therefore force T to be a monotone diffeomorphism so that T and T^{-1} are continuously differentiable and monotone almost everywhere. This definition of transport map is equivalent to the deterministic coupling of random variables presented in [102].

Assume that the target distribution μ_θ is a Bayesian posterior or other complicated distribution, but let μ_r to be a well understood distribution, such as Gaussian distribution or uniform distribution. In this case, the complicated structure of μ_θ is captured by the exact map T. This allows sampling and other tasks to be performed with the simple reference distribution instead of the more complicated distribution. In particular, when an exact map is available, sampling the target density π_n is as simple as taking a sample $r' \sim \mu_r$ and pushing it to the target space with $\theta' = T^{-1}(r')$. This concept was exploited by [79] for sampling Bayesian posteriors. Unfortunately, for target distributions that are wildly different than the reference distribution, finding an *exact* map that satisfies (2.1) can become computationally intractable and we are often relegated to working with approximations to T. Figure 2-1 illustrates the important difference between the approximate map $\tilde{T}$ and the exact map T. Even though an approximate map does not capture all of the target distribution's structure, it can still be useful in decoupling multiscale problems (as illustrated in Chapter 3), generating efficient proposals in the MCMC context (as in Chapter 4), or directly sampling an approximate posterior distribution (as in Chapter 5).

As is well known in the optimal transport literature (see [102] for a thorough guide to this area) there can be many monotone transformations that map between any two random variables, making some form of tie-breaking regularization necessary to ensure the map-selection process is unique.

One common approach is to introduce a transport cost. The transport cost, denoted here by $c(\theta, r)$, represents the cost of moving one unit of mass from some point θ to another point r. Using this unit cost, the total cost of pushing μ_θ to μ_r is given by

$$C_{\tilde{T}}(T) = \int_{\mathbb{R}^D} c\left(\theta, T(\theta)\right) \mu_\theta(d\theta). \tag{2.2}$$

The problem of minimizing this cost subject to the constraint $\mu_\theta = T_\sharp \mu_r$ is often called the Monge problem, in reference to Gaspard Monge, who first posed this problem in 1781 [78].[1] The transport map satisfying the distributional equality in (2.1) *and* minimizing the transport cost in (2.2) is the *optimal* transport map. The seminal works of Brenier [16] and McCann [75] show that this map is unique and monotone when

[1] Interestingly, Monge did not work on this problem out of pure mathematical curiosity. Like we described in Section 1.4, he posed this problem to minimize the transport costs of moving dirt excavated during fort construction [100].

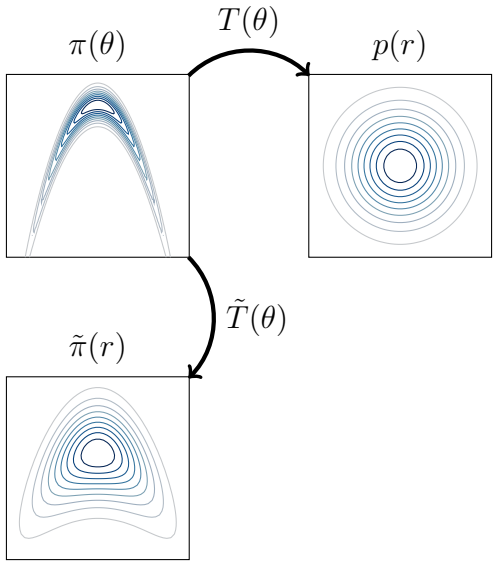


Figure 2-1: Illustration of exact and inexact transformations coming from T and $\tilde{T}$ respectively. The exact map pushes the target measure π to the standard Gaussian reference p while the approximate map only captures some of the structure in π, producing an approximation $\tilde{p}$ to the reference Gaussian.

μ_r does not contain any point masses and the cost function $c(\theta, T(\theta))$ is quadratic. Details of the existence and uniqueness proofs can also be found in [102].

Being a form of regularization, the cost function in (2.2) defines the form and structure of the optimal transport map. For illustration, consider the case when $\theta \sim N(0, I)$ and $r \sim N(0, \Sigma)$ for some covariance matrix Σ. In this Gaussian example, the transport map will be linear: $r \overset{i.d.}{=} \Sigma^{1/2}\theta$, where $\Sigma^{1/2}$ is any one of the many square roots of Σ. Two possible matrix square roots are the Cholesky factor, and the eigenvalue square root. Interestingly, when the cost is given by

$$c_{\text{Eig}}(\theta, T(\theta)) = \|\theta - T(\theta)\|^2, \tag{2.3}$$

the optimal square root, $\Sigma^{1/2}$, will be defined by the eigenvalue decomposition of Σ, but when the cost is given by the limit of a a weighted quadratic defined by

$$c_{\text{Ros}}(\theta, T(\theta)) = \lim_{t \to 0} \sum_{k=1}^{D} t^{k-1} |\theta_k - T_k(\theta)|, \tag{2.4}$$

the optimal square root, $\Sigma^{1/2}$, will be defined by the Cholesky decomposition of Σ. In the more general nonlinear and non-Gaussian setting, this latter cost is shown by [22] and [15] to yield the well-known Rosenblatt transformation from [91].

The Cholesky factor is a special case of the Rosenblatt transformation, which itself is just a multivariate generalization of using cumulative distribution functions to transform between univariate random variables (i.e., the "CDF trick"). Importantly, the lower triangular structure present in the Cholesky factor, which makes inverting

the transformation easy, is also present in the more general Rosenblatt transformation. This structure helps ensure the map is monotone but also has important computational advantages that we will demonstrate in Section 2.2.

While the lower triangular result is advantageous, working with the t^{k-1} term in c_{Ros} quickly results in numerical underflow as the problem dimension, D_θ, increases. The cost function c_{Ros} is meaningful for theoretical analysis, but we have not found it useful in practice. Thus, we will not directly attempt to minimize (2.2). Instead, we directly impose the lower triangular structure and search for an approximate map, $\tilde{T}$, that *approximately* satisfies the measure constraint. i.e., $\mu_r \approx \tilde{T}_\sharp \mu_\theta$. This is a key difference between our approach and classic optimal transport theory. We fix the form of the map and then try to approximately satisfy the distributional equality in (2.1), while classical optimal transport theory begins with exact measure transformation (2.1) and then uses the transport cost to find a unique map whose form comes from the particular choice of transport cost.

2.2 Constructing maps from samples

As mentioned above, the Cholesky factor provides a linear lower triangular transformation. In the more general nonlinear setting, a lower triangular map $T(\theta)$ will have a similar form, given by

$$T(\theta) = T(\theta_1, \theta_2, \ldots, \theta_D) = \begin{bmatrix} T_1(\theta_1) \\ T_2(\theta_1, \theta_2) \\ \vdots \\ T_D(\theta_1, \theta_2, \ldots, \theta_D) \end{bmatrix}, \tag{2.5}$$

where θ_d is component d of θ and $T_d : \mathbb{R}^d \to \mathbb{R}$ is a map for component d of the output. The lower triangular structure in (2.5) helps regularize the choice of maps and we no longer have to explicitly impose a transport cost; however, we do need to search for the lower triangular map that (at least approximately) satisfies the measure constraint $\mu_\theta = T_\sharp \mu_r$. We will do this by first defining a map-induced density $\tilde{\pi}(\theta)$, and then minimizing the difference between this map-induced density and the true density $\pi(\theta)$.

2.2.1 Formulation

To mathematically construct the map-induced density, let $p(r)$ be the reference density over r and consider a μ_θ-differentiable monotone transformation T.[2] This map will induce a density $\tilde{\pi}$ over θ given by

$$\tilde{\pi}(\theta) = p\left(T(\theta)\right) \left|\det(\partial T(\theta))\right|, \tag{2.6}$$

[2]Notice that T takes the target variable θ as an argument, not the reference random variable r. This is in contrast to previous works in [79] or [61] that build a map in the opposite direction: from the reference space to the target space.

where $\partial T(\theta)$ is the Jacobian matrix of the map at the point θ and $|\det(\partial T(\theta))|$ is the absolute value of the Jacobian matrix determinant.

To satisfy the measure constraint $\mu_\theta = T_\sharp \mu_r$, the map induced density $\tilde{\pi}$ must be equivalent to the target density π. Thus, we can find T by minimizing the "difference" between $\tilde{\pi}$ and π. We will use the Kullback-Leibler (KL) divergence to measure this difference. The KL divergence between π and $\tilde{\pi}$ is denoted by $D_{KL}(\pi\|\tilde{\pi})$ and defined by

$$\begin{aligned} D_{KL}(\pi\|\tilde{\pi}) &= \mathbb{E}_\pi\left[\log\left(\frac{\pi(\theta)}{\tilde{\pi}(\theta)}\right)\right] \\ &= \mathbb{E}_\pi\Big[\log \pi(\theta) \\ &\qquad - \log p\left(T(\theta)\right) - \log\left|\det(\partial T(\theta))\right|\Big]. \end{aligned} \tag{2.7}$$

Using this definition, the *exact* transport map T is defined as

$$T = \underset{t\in\mathcal{T}}{\operatorname{argmin}}\, \mathbb{E}_\pi\Big[- \log p\left(T(\theta)\right) - \log\left|\det(\partial T(\theta))\right|\Big], \tag{2.8}$$

where $\mathcal{T}$ is the space of all lower triangular diffeomorphisms on $\mathbb{R}^{D_\theta}$.

Notice that the KL divergence is not symmetric. We chose to take the expectation with respect to $\pi(\theta)$ because in the forthcoming applications we will often only have samples of $\pi(\theta)$ and can therefore only calculate a Monte Carlo approximation to the expectation. Furthermore, as we will show below, this ordering allows us to dramatically simplify (2.7) when p is Gaussian, which leads to a quite manageable optimization problem.

Assume we have K samples of π denoted by $\{\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(K)}\}$. As is done in sample average approximation (SAA) [62], we use these samples to create the following Monte Carlo approximation to (2.7)

$$\begin{aligned} D_{KL}(\pi\|\tilde{\pi}) &\approx \frac{1}{K}\sum_{i=1}^{K}\Big[\log \pi(\theta^{(i)}) \\ &\qquad - \log p\left(T(\theta^{(i)})\right) - \log\left|\det(\partial T(\theta^{(i)}))\right|\Big]. \end{aligned} \tag{2.9}$$

The map minimizing this expression is given by

$$\tilde{T} = \underset{t\in\mathcal{T}}{\operatorname{argmin}}\, \frac{1}{K}\sum_{i=1}^{K}\Big[- \log p\left(T(\theta^{(i)})\right) - \log\left|\det(\partial T(\theta^{(i)}))\right|\Big]. \tag{2.10}$$

Notice that because we have replaced the expectation in (2.7) with the finite sum in (2.9), $\tilde{T}$ is an approximation to T and only *approximately* pushes μ_θ to μ_r. However, when optimizing over the space of all monotone diffeomorphisms, i.e., $\tilde{T} \in \mathcal{T}$, the

approximate map will approach the exact map T as the number of Monte Carlo samples K grows to infinity. However, in practice, we will characterize $\tilde{T}$ with a finite expansion and $\tilde{T}$ will remain in some approximate space $\tilde{\mathcal{T}} \subset \mathcal{T}$, even as $K \to \infty$. As we will describe in section 2.3, the approximate space $\tilde{\mathcal{T}}$ will be the span of a finite number of basis functions. Thus, $\tilde{T}$ is approximate for two reasons: because we use a finite number of samples used in (2.9), and because $\tilde{T}$ lies within $\tilde{\mathcal{T}}$.

2.2.2 Enforcing monotonicity

Recall that we also require the map, $\tilde{T}$ from here on, to be monotone. To enforce monotonicity we need the Jacobian of $\tilde{T}$ to be positive definite μ_θ-almost everywhere. More precisely, we require

$$\partial\tilde{T}(\theta) \succeq 0 \quad \forall\theta \in \left\{x \in \mathbb{R}^D : \pi(x) > 0\right\}. \tag{2.11}$$

Notice that his implies the determinant $\det(\partial\tilde{T}(\theta))$ is also positive μ_θ-almost everywhere. This positive definite constraint is all we need to define the map-induced density $\tilde{\pi}$; however, to show convergence of our MCMC scheme in Chapter 4, we will also need to impose the mildly stricter condition that $\tilde{T}$ be bi-Lipschitz. For this condition, we require $\tilde{T}$ to satisfy the following constraints

$$\|\tilde{T}(\theta') - \tilde{T}(\theta)\| \geq d_{min}\|\theta' - \theta\| \tag{2.12}$$
$$\|\tilde{T}(\theta') - \tilde{T}(\theta)\| \leq d_{max}\|\theta' - \theta\|, \tag{2.13}$$

where $0 < d_{min} \leq d_{max} < \infty$. Notice that the lower bound in (2.12) implies a lower bound on the map derivative given by

$$\frac{\partial\tilde{T}_i(\theta)}{\partial\theta_i} \geq d_{min}. \tag{2.14}$$

Furthermore, because $\tilde{T}$ is lower triangular, the Jacobian $\partial\tilde{T}$ is lower triangular, and (2.14) subsequently ensures the Jacobian is positive definite. Thus, we can impose (2.12) instead of directly requiring (2.11). The reasons for requiring (2.12) and (2.13) instead of (2.11), will become more clear in the MCMC convergence discussion of Section 4.4 and Appendix A.

The derivative lower bound in (2.14) also allows us to remove the absolute values from the determinant term in (2.11). By removing the absolute values, and closing the set of feasible maps (i.e., (2.14) has a $\geq$ while (2.11) uses a $>$), we have made the feasible domain of this optimization problem convex. Because the objective is also convex, we have a convex optimization problem that can be solved efficiently. As pointed out in [61], removing the absolute value, which removes the decreasing solution, does not restrict restrict map performance.

Many representations of $\tilde{T}$ (e.g., a polynomial expansion) will yield a map with finite derivatives over any finite ball, but will have infinite derivatives as $\|\theta\| \to \infty$. Clearly, this class of map would not satisfy the upper bound constraint in (2.13).

Fortunately, a minor alteration of such a map can be used to satisfy (2.13).

Let $P(\theta)$ be a continuously differentiable map with infinite derivatives as $\|\theta\| \to \infty$, but finite derivatives over any ball $B(0,R)$ with $R < \infty$. We can satisfy (2.13) by setting $\tilde{T}(\theta) = P(\theta)$ over $B(0,R)$, but forcing $\tilde{T}(\theta)$ to be linear outside of this ball. To make this concept mathematically concrete, let $w(\theta) = R\frac{\theta}{\|\theta\|}$ be the projection of θ to the closest point in $B(0,R)$ and let $d(\theta) = \frac{\theta}{\|\theta\|} \cdot \nabla P(w(\theta))$ be the directional derivative of $P(\theta)$ at the ball boundary. Using these definitions, we define $\tilde{T}(\theta)$ in terms of $P(\theta)$ as

$$\tilde{T}(\theta) = \begin{cases} P(\theta) & \|\theta\| \leq R \\ P\left(w(\theta)\right) + d(\theta)(\theta - w(\theta)) & \|\theta\| > R \end{cases} . \tag{2.15}$$

Figure 2-2 also illustrates the difference between $P(\theta)$ and $\tilde{T}(\theta)$ in a one dimensional setting.

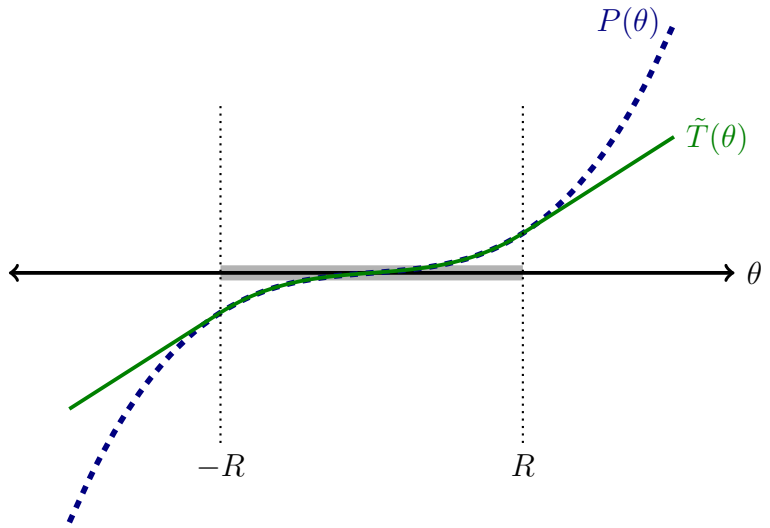


Figure 2-2: Illustration of the difference between $P(\theta)$ and $\tilde{T}(\theta)$ in (2.15). In this one dimensional illustration, $P(\theta)$ and $\tilde{T}(\theta)$ are identical for $\theta \in [-R, R]$; however, when θ is outside this interval, $\tilde{T}$ becomes linear. The slope of $\tilde{T}(\theta)$ for $\theta > R$ is given by $dP/d\theta$ evaluated at R.

Notice that a continuously differentiable $P(\theta)$ will yield a continuously differentiable $\tilde{T}(\theta)$. Moreover, assuming $P(\theta)$ satisfies the lower bound in (2.12), $\tilde{T}(\theta)$ now satisfies both (2.12) and (2.13). We would like to point out that some parameterizations of $\tilde{T}$, such as radial basis perturbations of a linear map (to be be discussed in Section 2.3), do not require the correction in (2.15) to satisfy the bounds in (2.12) and (2.13)

When a finite number of samples are used to approximate (2.9), R can usually be chosen so that all samples lie in $B(0,R)$ and P can be used directly. However, our MCMC convergence theory requires finite derivatives of $\tilde{T}$ as $\|\theta\| \to \infty$, which is provided by the correction in (2.15).

Unfortunately, we cannot generally enforce (2.14) over the entire support of the target measure. This leads us to a weaker, but practically enforceable, alternative – we require the map to be increasing at each sample used to approximate the KL divergence. Mathematically, we have

$$\left.\frac{\partial \tilde{T}_d}{\partial \theta_d}\right|_{\theta^{(i)}} \geq d_{min} \quad \forall d \in \{1, 2, \ldots, D\}, \forall i \in \{1, 2, \ldots, K\}. \tag{2.16}$$

In practice, we have found (2.16) sufficient to ensure the monotonicity of a map represented by a finite polynomial or radial basis expansion.

2.2.3 Simplifications of the KL cost function

Now we consider the task of minimizing the KL divergence in (2.9). From an optimization standpoint, the $\log \pi$ and $1/K$ terms in $D_{KL}(\pi \| \tilde{\pi})$ do not affect the minimization problem and can be removed. Our new goal is then to minimize the cost function given by

$$C_{KL}(\tilde{T}) = \sum_{i=1}^{K} -\log p\left(\tilde{T}(\theta^{(i)})\right) - \log\det\left[\partial \tilde{T}(\theta^{(i)})\right]. \tag{2.17}$$

While we could tackle this minimization problem directly, as in [79], we can further simplify this cost with a strategic choice of the reference density p.

Let $r \sim N(0, I)$. This choice of reference distribution implies

$$\log p(r) = -\frac{D}{2}\log(2\pi) - 0.5\sum_{d=1}^{D} r_d^2. \tag{2.18}$$

Now, notice that the lower triangular form of $\tilde{T}$ yields a lower triangular Jacobian matrix. This allows us to write the determinant term in (2.17) as

$$\log\left(\det\partial \tilde{T}(\theta)\right) = \log\left(\prod_{d=1}^{D} \frac{\partial \tilde{T}_d}{\partial \theta_d}\right) = \sum_{d=1}^{D} \log \frac{\partial \tilde{T}_d}{\partial \theta_d}. \tag{2.19}$$

Using this expression and (2.18) in the KL cost (2.17) yields a new form for the cost given by

$$C_{KL}(\tilde{T}) = \sum_{d=1}^{D}\sum_{i=1}^{K}\left[0.5\tilde{T}_d^2(\theta^{(i)}) - \log \left.\frac{\partial \tilde{T}_d}{\partial \theta_d}\right|_{\theta^{(i)}}\right]. \tag{2.20}$$

As we demonstrate in the next section, an appropriate parameterization of $\tilde{T}$ will allow us to solve the large optimization problem in (2.20) by independently solving D_θ smaller optimizations problems - one for each output of the map.

2.3 Transport map parameterization

Let $\bar{\gamma} = [\gamma_1, \gamma_2, \ldots, \gamma_D]$ be a collection of vector-valued parameters that define the approximate map $\tilde{T}(\theta) = \tilde{T}(\theta; \gamma)$. Assume each vector γ_d has M_d components and takes values in $\mathbb{R}^{M_d}$. The dimension of each parameter vector, M_d, is dependent on the parametric form of $\tilde{T}$, which will be discussed in more detail below. Now, assume that the parameters for each component $\tilde{T}_d$ of the map are independent, that is, $\tilde{T}_d(\theta; \bar{\gamma}) = \tilde{T}_d(\theta; \gamma_d)$. In this case, we can split the minimization of the KL cost in (2.20) into independent minimization problems for each dimension. Each of the D_θ different optimization problems is given by

$$\begin{aligned} \underset{\gamma_d}{\text{minimize}} \quad & \sum_{i=1}^{K} \left[0.5\tilde{T}_d^2(\theta^{(i)}; \gamma_d) - \log \left. \frac{\partial \tilde{T}_d(\theta; \gamma_d)}{\partial \theta_d} \right|_{\theta^{(i)}} \right] \\ \text{subject to} \quad & \left. \frac{\partial \tilde{T}_d(\theta; \gamma_d)}{\partial \theta_d} \right|_{\theta^{(i)}} \geq d_{min} \quad \forall i \in \{1, 2, \ldots, K\} \end{aligned} \tag{2.21}$$

We should also mention that all of these optimization problems could be solved in parallel and no evaluations of the target density $\pi(\theta)$ are required. This is an important feature of our approach that is critical to the multiscale and offline techniques that will be discussed in chapters 3 and 5. Importantly, each of the optimization problems is also convex anytime the map components $\tilde{T}_d$ are linear in the coefficients γ_d. Ensuring $\tilde{T}_d$ is linear in γ_d also has several computational advantages that will be discussed in section 2.4.

2.3.1 Multivariate polynomials

One way to parameterize each component of the map $\tilde{T}_d$, is with an expansion of multivariate polynomials. We define each multivariate polynomial as a tensor product of D_θ scalar polynomials. The particular univariate polynomials used in the multivariate polynomial $\psi_{\mathbf{j}}$ are defined by a multi-index $\mathbf{j} = (j_1, j_2, \ldots, j_D) \in \mathbb{N}^D$ through the expression

$$\psi_{\mathbf{j}}(\theta) = \prod_{p=1}^{D} \varphi_{j_p}(\theta_p). \tag{2.22}$$

The one dimensional polynomials, φ_{j_p}, could be Hermite polynomials, Legendre polynomials, or members of your favorite polynomial family. [3] Using the multivariate polynomials, we can express the map as a finite expansion with the form

$$\tilde{T}_d(\theta) = \sum_{\mathbf{j} \in \mathcal{J}_d} \gamma_{d,\mathbf{j}} \psi_{\mathbf{j}}(\theta), \tag{2.23}$$

[3] In the polynomial chaos community, the polynomials are usually chosen to be orthogonal with respect to the input measure, μ_θ [108, 66]. However, we may only have samples of μ_θ, μ_θ may not be known, or μ_θ may not be one of the canonical distributions found in the Wiener-Askey scheme. Because we cannot follow the Wiener-Askey scheme to choose an orthogonal polynomial family, our choice of scalar polynomial is not as straightforward as in the typical polynomial chaos setting.

where, $\mathcal{J}_d$ is a set of multi-indices defining the polynomial terms in the expansion. Notice that the cardinality of the multi-index set defines the dimension of each parameter vector γ_d, i.e., $M_d = |\mathcal{J}_d|$. Readers should also note that a proper choice of each multi-index set $\mathcal{J}_d$ can force $\tilde{T}$ to be lower triangular. An example of this construction is given in the sets $\mathcal{J}_d^{TO}$, $\mathcal{J}_d^{NM}$ and $\mathcal{J}_d^{NC}$ defined below.

To make the concept of multi-indices and polynomial expansions more concrete, consider the use of monomials in (2.23). In this setting, the multivariate polynomial is defined by the product of monomials with powers given by the multi-index $\mathbf{j} = (j_1, j_2, \ldots, j_D)$. Mathematically, the multivariate polynomial is expressed as $\psi_{\mathbf{j}}(\theta) = \prod_{i=1}^{D} \theta_i^{j_i}$. A typical choice of multi-index set $\mathcal{J}_d$ would then be to limit the total order of the polynomial to some value p such as

$$\mathcal{J}_d^{TO} = \{\mathbf{j} : \|\mathbf{j}\|_1 \leq p,\, j_i = 0\, \forall i > d\}.$$

The first constraint in this set, $\|j_i\|_1 \leq p$, limits the polynomial order while the second constraint $j_i = 0\, \forall i > d$ forces $\tilde{T}(\theta)$ to be lower triangular. A problem with using $\mathcal{J}_d^{TO}$ is that the number of terms in the set grows exponentially. A more feasible multi-index set in moderate dimensions is to remove all mixed terms in the basis,

$$\mathcal{J}_d^{NM} = \{\mathbf{j} : \|\mathbf{j}\|_1 \leq p,\, j_i j_k = 0\, \forall i \neq k,\, j_i = 0\, \forall i > d\}.$$

An even more parsimonious option is to remove all cross terms, yielding the set

$$\mathcal{J}_d^{NC} = \{\mathbf{j} : \|\mathbf{j}\|_1 \leq p,\, j_i = 0\, \forall i \neq d\}.$$

Figure 2-3 illustrates the difference between these three sets for a maximum order of $p = 3$.

Reducing the number of terms in the basis reduces the map complexity, allowing us to tackle higher dimensional problems. However, by reducing the map complexity, we are also limiting how much structure the map can capture. As the problem dimension increases, it becomes more important to strategically choose a basis set. More on constructing high dimensional maps will be discussed in Section 2.7.

2.3.2 Radial basis functions

An alternative to a polynomial parameterization of $\tilde{T}$ is to use an expansion of linear terms and radial basis functions. The general form of the expansion in (2.23) remains the same; however, the nonlinear multivariate polynomials are replaced with radial basis functions to yield the expansion

$$\tilde{T}_d(\theta) = a_{d,0} + \sum_{j=1}^{d} a_{d,j}\theta_j + \sum_{j=1}^{P_d} b_{d,j}\phi_j(\theta_1, \theta_2, \ldots, \theta_d; \bar{\theta}^{d,j}), \tag{2.24}$$

where P_d is the total number of radial basis functions in the expansion, $\phi_j(\theta_1, \theta_2, \ldots, \theta_d; \bar{\theta}^{d,j})$ is a radial basis function centered at some point $\bar{\theta}^{d,j} \in \mathbb{R}^d$ that only depends on the

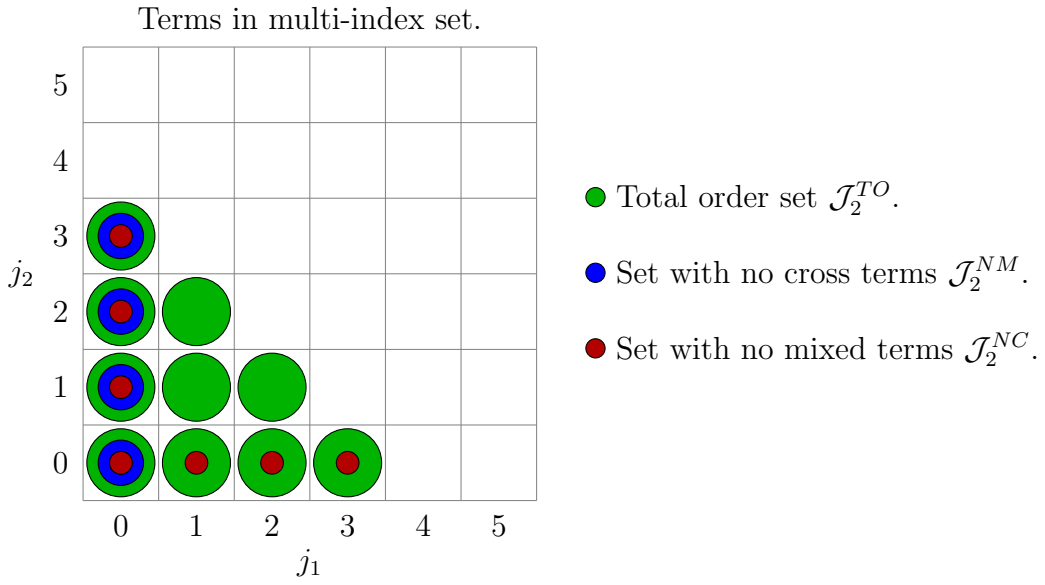


Figure 2-3: Visualization of multi-index sets for the second component of a two dimensional map, $\tilde{T}_2(\theta_1, \theta_2)$. In this case, j_1 is the power on θ_1 and j_2 is the power on θ_2. A filled circle indicates that a term is present in the set of multi-indices.

first d dimensions of the target random variable, and the a and b coefficients constitute the map coefficients $\gamma_d = [a_{d,0}, a_{d,1}, \ldots, a_{d,d}, b_{d,1}, \ldots, b_{d,P_d}]^T$. In (2.24), we have kept the constant term $a_{d,0}$ and linear terms in each direction. We have found that keeping all the linear terms dramatically improves map performance with only slightly more computational costs. However, we should note that monotonicity can also be enforced using only the linear term in dimension d (i.e., θ_d).

Using MATLAB-like notation, where $\theta_{1:d} = [\theta_1, \theta_2, \ldots, \theta_d]^T$, a typical form for the radial basis function ϕ_j is

$$\phi_j(\theta_1, \theta_2, \ldots, \theta_d; \bar{\theta}^{d,j}) = \exp\left[-\left(\theta_{1:d} - \bar{\theta}^{d,j}\right)^T H^j \left(\theta_{1:d} - \bar{\theta}^{d,j}\right)\right]. \tag{2.25}$$

where $H^j \in \mathbb{R}^{d\times d}$ is a symmetric matrix defining the shape and width of the radial basis function. In general, defining a general matrix H^j requires us to choose the $\frac{d(d+1)}{2}$ entries in the matrix. For large dimensions, this becomes even more difficult than working with the total order polynomials shown in Figure 2-3. Fortunately, we can restrict the form of $H^j \in \mathbb{R}^{d\times d}$ to simplify the problem much like we simplified our multi-index set in Figure 2-3 to simplify the polynomial expansion in (2.23).

One simplification is to only use one dimensional radial basis functions in (2.24). In terms of the H^j matrix, this is equivalent to only allowing a single diagonal element of H^j to be nonzero. In this setting (which is similar to $\mathcal{J}_d^{NC}$ above), the expansion

in (2.24) takes the form

$$\tilde{T}_d(\theta) = a_{d,0} + \sum_{j=1}^{d} a_{d,j}\theta_j + \sum_{k=1}^{d}\sum_{j=1}^{P_{d,k}} b_{d,k,j}\phi_{j,k}(\theta_k; \bar{\theta}_k^{d,j}), . \tag{2.26}$$

where $\phi_{j,k}$ is now a one dimensional radial basis function given by

$$\phi_{j,k}(\theta_k; \bar{\theta}_k^{d,j}) = \exp\left[-H_{kk}^{j}\left(\theta_k - \bar{\theta}_k^{d,j}\right)^2\right]. \tag{2.27}$$

Note that we could also require $k = d$ and $a_{d,j} = 0$, $\forall d \neq j$ in (2.26). These restrictions would yield an expansion analogous to the $\mathcal{J}_d^{NM}$ polynomial expansion. Note that this choice leads to a fully diagonal map. For diagonal maps, the dimension of the optimization problem in (2.21) is constant in the dimension. Diagonal maps are therefore useful in very high dimensional problems.

When using radial basis functions, we could try to simultaneously optimize (2.21) over γ_d and the center of each function $\bar{\theta}_k^{d,j}$; however, the nonlinear dependence of the map on $\bar{\theta}_k^{d,j}$ makes the optimization problem significantly more difficult than optimizing only over γ_d. In this work, we instead fix the locations $\bar{\theta}_k^{d,j}$ a priori and only optimize over the expansion coefficients. Choosing the function locations can be a tricky problem-dependent task. However, when using the one dimensional radial basis functions in (2.27), it is helpful to look at the marginal sample distributions of θ before constructing the maps.

In particular, we have found that evenly spacing the centers $\bar{\theta}_k^{d,j}$ of the radial basis functions between the 1% and 99% quantiles of θ_k works well. Let $Q_{k,01}$ denote the 1% quantile of θ_k and $Q_{k,99}$ denote the 99% quantile of θ_k. Evenly spaced node locations are then given by

$$\bar{\theta}_k^{d,j} = Q_{k,01} + (j-1)\frac{Q_{k,99} - Q_{k,01}}{P_{d,k} - 1}. \tag{2.28}$$

We have also found that a reasonable heuristic for the radial basis function scale is given by

$$H_{kk}^{j} = 0.4\frac{P_{d,k}}{Q_{k,99} - Q_{k,01}}. \tag{2.29}$$

Choosing H_{kk}^{j} much larger than this can make enforcing monotonicty difficult while choosing H_{kk}^{j} much smaller than this can make the map too smooth and make the optimization problem in (2.21) more difficult. Of course, better results could be obtained by tuning H_{kk}^{j} on a problem by problem basis, but all the examples in this thesis will use the heuristics in (2.28) and (2.29).

2.3.3 Choosing map ordering

With either polynomials or radial basis functions, we regularize the optimization problem in (2.20) by forcing the map $\tilde{T}$ to be lower triangular. The lower triangular

map also allows us to separate the large optimization problem in (2.20) to the D_θ problems defined by (2.21). However, with a lower triangular map, the ordering of the target random variable θ is important. This section discusses the ordering problem and provides one way of ordering the map.

Assume we have an operator S that reorders θ such that $S(\theta) = [\theta_{s(1)}, \theta_{s(2)}, \ldots, \theta_{s(D_\theta)}]^T$ where s can be any bijection between $\{1, 2, 3, \ldots, D_\theta\}$ and itself. When the target distribution μ_θ contains no atoms, we know by the existence of the Rosenblatt transform, that for any reordering S, an exact map exists in the space of continuous transformations $C(\mathbb{R}^{D_\theta})$. However, when the map is represented within some finite dimensional space $\mathcal{T} \subset C(\mathbb{R}^{D_\theta})$, different reorderings S can result in approximate maps with dramatically different errors.

To illustrate this point, consider a simple two dimensional example where samples of $[\theta_1, \theta_2]^T$ are generated with a simple transformation given by

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_1^2 + r_2 \end{bmatrix}, \tag{2.30}$$

where r_1 and r_2 are standard Normal random variables. In this example, an exact lower triangular map $T_l(\theta_1, \theta_2) \overset{i.d.}{=} r$ is obtained with the simple quadratic polynomial

$$\begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = T_l(\theta_1, \theta_2) = \begin{bmatrix} \theta_1 \\ \theta_2 - \theta_1^2 \end{bmatrix}. \tag{2.31}$$

However, if we swap the order of θ and try to build a lower triangular map $T_u(\theta_2, \theta_1)$, the exact map cannot be expressed in terms of a finite polynomial expansion because constructing the map would require us to use the CDF of $r_1^2 + r_2$, which cannot be represented with analytic functions. Thus, a simple quadratic polynomial can be used to build an exact lower triangular map from (θ_1, θ_2) to (r_1, r_2), but trying to construct a map from (θ_2, θ_1) to (r_1, r_2) requires a much more complicated parameterization of $\tilde{T}$. Clearly, we need a way of detecting this and subsequently finding an appropriate reordering $S(\theta)$.

In the simple two dimensional example given above, building the reverse ordered map (i.e., from (θ_2, θ_1) to (r_1, r_2)) was difficult because the conditional density $\pi(\theta_2|\theta_1)$ was a simple Gaussian density, but the marginal density $\pi(\theta_2)$ was much more complicated. In general, we want to place θ_i ahead of θ_j when the difference between $\pi(\theta_i|\theta_j)$ and $\pi(\theta_i)$ is smaller than the difference between $\pi(\theta_j|\theta_i)$ and $\pi(\theta_j)$.

A useful indication of this ordering dependence is the asymmetric expression given by

$$\Gamma_{i,j} = \mathbb{E}\left[\left(\theta_i - \bar{\theta}_i\right)\left(\theta_j - \bar{\theta}_j\right)^p\right], \tag{2.32}$$

where $\bar{\theta}_i = \mathbb{E}[\theta_i]$ and $p > 1$.[4] This expression represents the nonlinear dependence of

[4]When p is an integer, each value of $\Gamma_{i,j}$ is an entry in a $p+1$ order symmetric tensor. Clearly, when $p = 1$, $\Gamma_{i,j}$ is an entry in a covariance matrix. For $p = 2$, we can write out $\Gamma_{i,j} = \hat{\Gamma}_{i,j,j}$ where $\hat{\Gamma}_{i,j,k} = \mathbb{E}\left[\left(\theta_i - \bar{\theta}_i\right)\left(\theta_j - \bar{\theta}_j\right)\left(\theta_k - \bar{\theta}_k\right)\right]$ defines a symmetric third order tensor. This third order tensor is related to the skewness-tensor, which is often used in combination with the Fisher information metric to define connections on manifolds of statistical models [3].

θ_i on θ_j. When $\Gamma_{i,j} = 0$, this indicates that there is minimal dependence of θ_i on θ_j, thus θ_i should come before θ_j in the reordering. In general, θ_i should come before θ_j when $\Gamma_{i,j} < \Gamma_{j,i}$. This observation allows us to sort the components of θ with Monte Carlo approximations to (2.32). In practice, we typically set $p = 2$ in (2.32). With this value of p, algorithm 2.1 shows one method of computing the reordering using an insertion sort. The insertion sort algorithm was used here for illustration but more efficient sorting algorithms such as merge sort of quicksort should be used in practice. Regardless of the sorting algorithm, the `Gamma` function in algorithm 2.1 would remain unchanged. Also, efficient implementations of algorithm 2.1 should precompute the $(\theta_i^{(k)} - \hat{\bar{\theta}}_i)$ and $(\theta_j^{(k)} - \hat{\bar{\theta}}_j)^2$ terms used by the `Gamma` function.

As a side note, a normalized version of (2.32) is also used to analyze non-Gaussian distributions in remote-sensing [40] and in the quantitative finance community [37], where it is often referred to as co-skewness.

Algorithm 2.1: Example of sorting dimensions of map using an insertion sort and Monte Carlo approximation to (2.32). Recall that K is the number of samples in the Monte Carlo approximation and D_θ is the parameter dimension.

Function `InsertionSort`(*Samples* $\Theta = \{\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(K)}\}$)
  $s_d = d$ for $d \in \{1, 2, \ldots, D_\theta\}$
  for $i \leftarrow 2$ ***to*** D_θ **do**
    $j \leftarrow i$
    while $j > 1$ ***and*** `Gamma`$(s_j, s_{j-1}, \Theta) <$ `Gamma`(s_{j-1}, s_j, Θ) **do**
      Swap s_j and s_{j-1}
      $j \leftarrow j - 1$
    end
  end
  return s

Function `Gamma`(i,j,Θ)
  $\hat{\bar{\theta}}_i = \frac{1}{K} \sum_{k=1}^{K} \theta_i^{(k)}$
  $\hat{\bar{\theta}}_j = \frac{1}{K} \sum_{k=1}^{K} \theta_j^{(k)}$
  $\Gamma_{ij} = \frac{1}{K} \sum_{k=1}^{K} (\theta_i^{(k)} - \hat{\bar{\theta}}_i)(\theta_j^{(k)} - \hat{\bar{\theta}}_j)^2$
  return Γ_{ij}

2.4 Solving the optimization problem

When using either the radial basis expansion with fixed locations, or the polynomial expansion, the map $\tilde{T}_d(\theta)$ is linear in the expansion coefficients. This allows the objective in (2.21) to be evaluated using efficient matrix-matrix and matrix-vector operations. To see this, assume we have two matrices $F_d \in \mathbb{R}^{K \times M_d}$ and $G_d \in \mathbb{R}^{K \times M_d}$ with components defined by $F_{di\mathbf{j}} = \psi_{\mathbf{j}}(\theta^{(i)})$ and $G_{di\mathbf{j}} = \frac{\partial \psi_{\mathbf{j}}}{\partial \theta_d}(\theta^{(i)})$ for all $\mathbf{j} \in \mathcal{J}_d$. Recall that K is the number of samples in our Monte Carlo approximation of the KL diver-

gence. Using these matrices and taking advantage of (2.23) or (2.24), we can rewrite (2.21) as

$$\begin{aligned} \underset{\gamma_d}{\text{minimize}} \quad & \frac{1}{2}\gamma_d^T(F_d^T F_d)\gamma_d - c^T \log(G_d\gamma_d) \\ \text{subject to} \quad & G_d\gamma_d \geq d_{min}, \end{aligned} \tag{2.33}$$

where c is a K dimensional vector of ones and the log is taken componentwise. Clearly, the objective can be evaluated with efficient numerical linear algebra routines.

On top of efficient evaluations, the only difference between (2.33) and a simple quadratic program is the log term. However, as shown in Figure 2-4, the quadratic term often dominates the log term, making a Newton-like optimizer quite efficient. In practice, we usually observe convergence in less than 10 Newton iterations. We should also point out that the constraints are never active at the solution of this problem because the log term in (2.33) acts as a barrier function for the constraints.

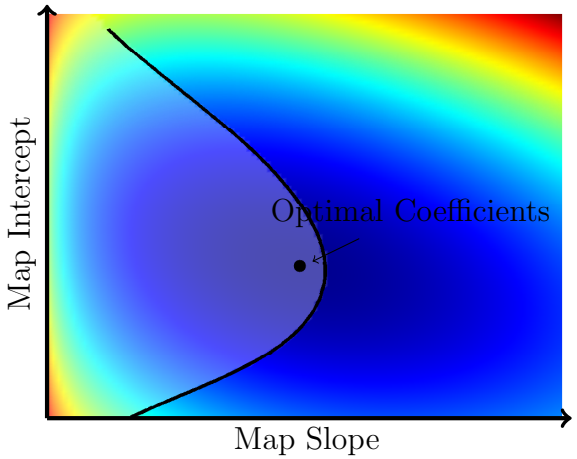


Figure 2-4: Illustration of the objective function in (2.21) for a Gaussian π and linear map. The colored surface are the values of the objective in (2.33). The lightly shaded region to the left of the contour line is the region of the parameter space where the log term is larger than the quadratic term in the objective. Clearly the problem is convex and most of the solution space is dominated by the quadratic term. These features make Newton methods particular suitable for efficient for solving this problem.

All the of exploitable structure in (2.33) means that repeatedly solving this problem in an adaptive MCMC framework is tractable. More details on such an adaptive MCMC sampler are given in Section 4.2.

2.5 Approximating the inverse map

Until now, we have focused entirely on constructing a map from the target random variable θ to the reference random variable r. However, in many cases, an inverse map from r to θ is needed. Because $\tilde{T}(\theta)$ is lower triangular, we could simply perform back-substitution through a sequence of D_θ one-dimensional nonlinear solves, but this

becomes computationally cumbersome when many evaluations of $\tilde{T}^{-1}(r)$ are required. This section shows that once $\tilde{T}(\theta)$ has been computed, an approximation $\tilde{F}(r) \approx \tilde{T}^{-1}(r)$ can be computed with standard regression techniques.

Assume we have already constructed $\tilde{T}(\theta)$ using K samples $\{\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(K)}\}$. Pushing each of these samples through $\tilde{T}$, gives a set of samples on the reference space $\{r^{(1)}, r^{(2)}, \ldots, r^{(K)}\}$, where $r^{(k)} = \tilde{T}(\theta^{(k)})$. Notice that there is a one to one correspondence between target samples and reference samples. Thus, we can use these pairs to define a simple least squares problem for an approximate inverse $\tilde{F}$. The problem we now need to solve is

$$\tilde{F}^* = \underset{\tilde{F}}{\operatorname{argmin}} \sum_{k=1}^{K} \sum_{d=1}^{D_\theta} \left(\tilde{F}_d(r^{(k)}) - \theta_d^{(k)} \right)^2. \tag{2.34}$$

When each dimension of $\tilde{F}$ is parameterized independently (like $\tilde{T}$ in Section 2.3), then (2.34) can be separated into individual problems for each dimension. This is similar to our separation of (2.20) into the D_θ problems defined by (2.21).

Each of these D_θ least squares problems will take the form

$$\tilde{F}_d^* = \underset{\tilde{F}_d}{\operatorname{argmin}} \sum_{k=1}^{K} \left(\tilde{F}_d(r^{(k)}) - \theta_d^{(k)} \right)^2. \tag{2.35}$$

This is a classic least squares problem that can easily be solved by representing $\tilde{F}_d$ with a finite basis, constructing a Vandermonde matrix, and solving the system with a QR decomposition. When polynomials are used to represent $\tilde{T}_d$, we use polynomials of the same order to represent $\tilde{F}_d$. However, when radial basis functions are used to represent $\tilde{T}_d$, we also use radial basis functions to describe $\tilde{F}_d$, but we choose node locations and widths using quantiles of the reference random variable r; the samples $\{r^{(1)}, r^{(2)}, \ldots, r^{(K)}\}$ are used to estimate the quantiles in (2.28) and (2.29).

2.6 Numerical example

This section provides an initial illustration of sample-based map construction on a simple two dimensional problem. We will use samples of the target random variable θ to construct $\tilde{T}$ by solving (2.21) and then find $\tilde{F}$ by solving the regression problem in (2.35). Each target sample $\theta^{(k)}$ is generated first sampling $z_1^{(k)}$ and $z_2^{(k)}$ from a scalar standard normal distribution and then evaluating the transformation

$$\theta^{(k)} = \begin{bmatrix} \theta_1^{(k)} \\ \theta_2^{(k)} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} z_1^{(k)} \\ \cos\left(z_1^{(k)}\right) + \frac{1}{2} z_2^{(k)} \end{bmatrix}. \tag{2.36}$$

Notice that this is a 45° degree rotation of a "banana" looking density. The transformation gives the target density shown in Figure 2-5(a). We will construct linear, cubic, and fifth order maps using Hermite polynomials and the three types of multi-

index sets from Section 2.3.1: $\mathcal{J}_d^{NC}$, $\mathcal{J}_d^{NM}$, $\mathcal{J}_d^{TO}$. Recall that $\mathcal{J}_d^{NC}$ yields a diagonal map, $\mathcal{J}_d^{NM}$ yields a lower triangular map without mixed terms, and $\mathcal{J}_d^{TO}$ yields a total order map.

In this example, we generated 10000 samples of the target distribution and then solved (2.21) to find $\tilde{T}$. After construction $\tilde{T}$, we evaluated $\tilde{T}$ at the 10000 target samples to get target-reference sample pairs; these samples were used in the regression framework of the previous section to construct the approximate inverse map $\tilde{F}(r)$. Table 2.1 shows the wall-clock times for these three steps. The times were obtained running 8 OpenMP threads on a fourth generation Intel core i7 running at 3.5GHz. In all cases, we were able to construct both $\tilde{T}$ and $\tilde{F}$ in under four seconds.

Interestingly, the time it took to evaluate $\tilde{T}$ at 10000 samples took much longer than the optimization time. This is a result of our code structure. The optimization code utilizes the efficient matrix-vector operations mentioned in Section 2.4 as well as multiple threads for each optimization problem; however, the evaluation function is optimized for a single fast evaluation (not 10000) and therefore does not exploit the same structure. We will address this issue in future implementations.

Table 2.1: Timing comparison of polynomial maps. Wall-clock times are shown for constructing $\tilde{T}$, evaluating $\tilde{T}$ on each of the 10000 samples, and constructing $\tilde{F}$ using regression. All times are shown in milliseconds.

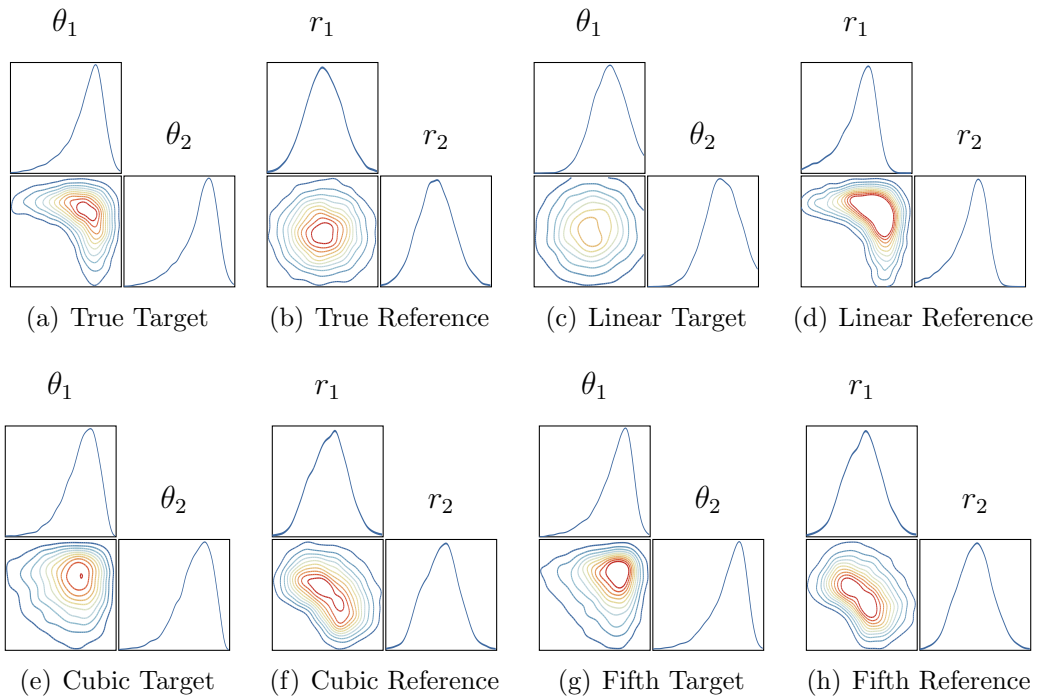
Basis	Order	$\tilde{T}$ Const. (ms)	$\tilde{T}$ Eval. (ms)	$\tilde{F}$ Regr. (ms)
$\mathcal{J}_d^{NM}$	1	80	750	30
	3	170	870	50
	5	300	1000	90
$\mathcal{J}_d^{NC}$	1	100	730	40
	3	220	1000	70
	5	370	1250	120
$\mathcal{J}_d^{TO}$	1	90	720	40
	3	280	1360	100
	5	660	2730	190

The accuracy of $\tilde{T}(\theta)$ is also demonstrated in Table 2.2. We compare the marginal moments of the map output with the known values of a standard normal distribution. Because standard Gaussian densities are by definition marginally Gaussian along any direction, we also compare the moments of $\tilde{r}_m = \frac{1}{\sqrt{2}}(\tilde{T}_1(\theta) + \tilde{T}_2(\theta))$ to a standard normal distribution. From the errors, we can see that all of the fifth order maps correctly capture the marginal moments of r_1 and r_2. However, the additional mixed terms in the total order map are needed to adequately capture the skewness and kurtosis of r_m. This can also be seen by comparing Figures 2-5–2-7.

The diagonal maps and separable lower triangular maps defined by $\mathcal{J}_d^{NM}$ and $\mathcal{J}_d^{NC}$ can easily capture the marginal distributions of the target distribution $\pi(\theta)$, but cannot adequately capture the target correlation structure. However, as we see from the timings in Table 2.1, these types of maps can be slightly more efficient to evaluate because they involve fewer basis functions. Using fewer basis functions becomes more important for high dimensional problems, which will become clear in the next section.

Table 2.2: Comparison of $\tilde{T}(\theta)$ and r using various polynomial maps. Note that $\tilde{r}_1 = \tilde{T}_1(\theta)$ is the map induced approximation to r and $\tilde{r}_2 = \tilde{T}_2(\theta)$ is the map-induced approximation to r_2. We also show the marginal moments of $\tilde{r}_m$, which would also be identical to a standard normal distribution with an exact map $\tilde{T}$.

Basis	Order	Mean			Variance			Skewness			Kurtosis		
		$\tilde{r}_1$	$\tilde{r}_2$	$\tilde{r}_m$	$\tilde{r}_1$	$\tilde{r}_2$	$\tilde{r}_m$	$\tilde{r}_1$	$\tilde{r}_2$	$\tilde{r}_m$	$\tilde{r}_1$	$\tilde{r}_2$	$\tilde{r}_m$
Truth		**0.00**	**0.00**	**0.00**	**1.00**	**1.00**	**1.00**	**0.00**	**0.00**	**0.00**	**3.00**	**3.00**	**3.00**
$\mathcal{J}_d^{NM}$	1	0.00	0.00	0.00	1.00	1.00	0.61	-1.10	-1.10	-0.41	4.27	4.30	3.23
	3	0.00	0.00	0.00	1.00	1.00	0.63	-0.02	-0.04	0.48	3.16	3.13	3.89
	5	**0.00**	**0.00**	**0.00**	**1.00**	**1.00**	**0.63**	**0.00**	**0.00**	**0.49**	**3.08**	**3.07**	**3.63**
$\mathcal{J}_d^{NC}$	1	0.00	0.00	0.00	1.00	1.00	1.00	-1.08	-0.92	-0.49	4.23	4.39	3.44
	3	0.00	0.00	0.00	1.00	1.00	1.00	0.00	0.22	0.35	3.30	3.45	3.31
	5	**0.00**	**0.00**	**0.00**	**1.00**	**1.00**	**1.00**	**-0.01**	**0.18**	**0.26**	**3.16**	**3.39**	**3.04**
$\mathcal{J}_d^{TO}$	1	0.00	0.00	0.00	1.00	1.00	1.00	-1.13	-0.96	-0.50	4.44	4.54	3.49
	3	0.00	0.00	0.00	1.00	1.00	1.00	-0.03	0.11	0.07	3.07	3.70	3.10
	5	**0.00**	**0.00**	**0.00**	**1.00**	**1.00**	**1.00**	**0.00**	**0.05**	**0.01**	**3.11**	**3.12**	**2.98**



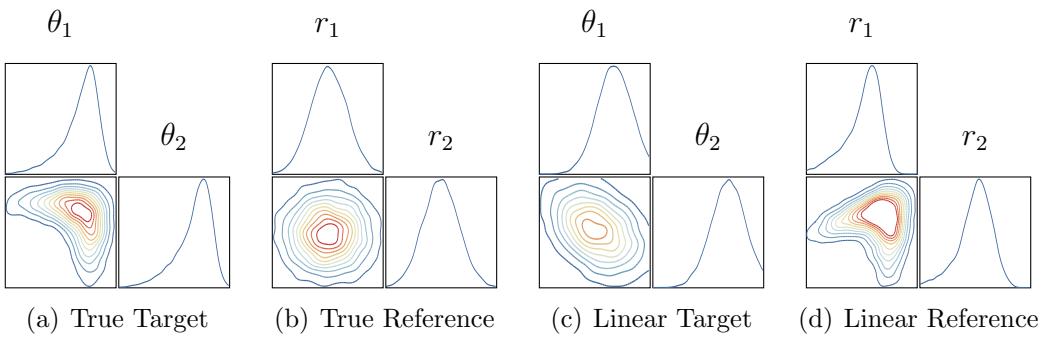
(a) True Target (b) True Reference (c) Linear Target (d) Linear Reference

(e) Cubic Target (f) Cubic Reference (g) Fifth Target (h) Fifth Reference

Figure 2-5: Convergence of diagonal maps defined by $\mathcal{J}_d^{NC}$.

2.7 Constructing high-dimensional maps

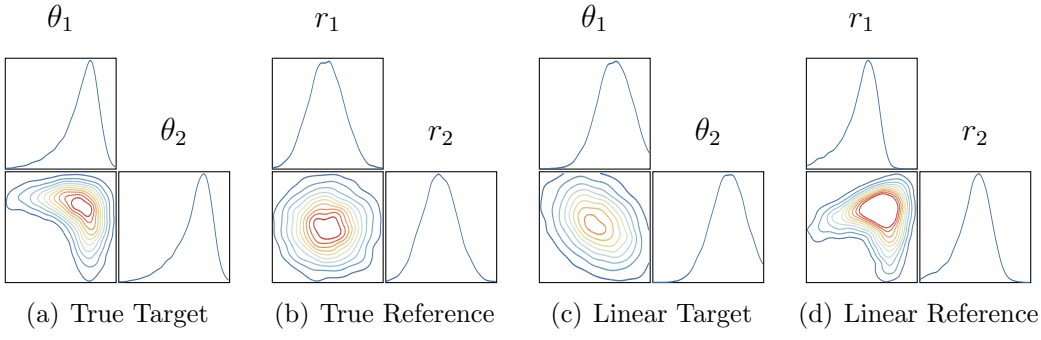
When the target random variable θ is high dimensional, parameterizing the map with total-order limited polynomials or arbitrary radial basis functions becomes intractable because the number of unknowns in the expansions (2.23) or (2.24) becomes too large to practically solve the optimization problem in (2.21). The obvious solution is not to use total order polynomials or arbitrary radial basis functions, but to only use a



(a) True Target (b) True Reference (c) Linear Target (d) Linear Reference

(e) Cubic Target (f) Cubic Reference (g) Fifth Target (h) Fifth Reference

Figure 2-6: Convergence of nonlinear separable maps defined by $\mathcal{J}_d^{NM}$.



(a) True Target (b) True Reference (c) Linear Target (d) Linear Reference

(e) Cubic Target (f) Cubic Reference (g) Fifth Target (h) Fifth Reference

Figure 2-7: Convergence of total order maps maps defined by $\mathcal{J}_d^{TO}$.

relatively small number of basis functions. In terms of multivariate polynomials, the

obvious choice is to remove any mixed terms, such as $\theta_1\theta_2$ or $\theta_1^2\theta_2$, from the expansion. A sparser alternative would be to remove cross terms altogether. In this case, each output of the map $\tilde{T}_i(\theta)$ would depend only on θ_i. Similarly, one dimensional radial basis functions can be used to ensure $\tilde{T}_i(\theta)$ depends only on θ_i.

When mixed terms are removed, the maximum number of unknowns in $\tilde{T}_i(\theta)$ will grow linearly in the dimension. Moreover, when all cross terms are removed or one dimensional radial basis functions are used, the number of unknowns in $\tilde{T}_i(\theta)$ will be constant with the dimension. In this latter case, the map will also be diagonal. Unfortunately, by reducing the number of terms in the expansion, we simultaneously reduce the expressive power of the map, which will cause $\tilde{T}(\theta)$ to be an inadequate approximation to the exact map $T(\theta)$. Composing multiple simple maps is one way to overcome this inadequacy. With the ultimate goal of constructing maps in high dimensions, the remainder of this section will introduce and analyze several ways of constructing complicated maps through composition.

2.7.1 Compositional map overview

Even in high dimensions, we can use the tools of Section 2.2, a small set of basis functions, and samples $\{\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(K)}\}$ from the target density $\pi(\theta)$ to construct an initial approximate map $\tilde{T}^1(\theta)$. However, being an approximation, the output of this map will only approximately satisfy the equality constraint

$$\tilde{T}^1(\theta) \overset{i.d.}{\approx} r. \tag{2.37}$$

If this approximation is insufficient, we can again use the standard tools of Section 2.2 to construct another map $\tilde{T}^2$, from the output of $\tilde{T}^1$ to the reference Gaussian. This second map is constructed using samples $\{\tilde{r}^{1,(1)}, \tilde{r}^{1,(2)}, \ldots, \tilde{r}^{1,(K)}\}$, which are outputs of layer 1 defined by $\tilde{r}^{1,(k)} = \tilde{T}(\theta^{(k)})$. Figure 2-8 illustrates this concept. The first layer of the map does not adequately capture all the target structure. However, when the output of the first layer is again transformed with the second layer, the composition produces a transformed measure that is much closer to the desired Gaussian reference distribution. In essence, the second layer corrected for the inadequacies of the first layer.

Once $\tilde{T}^2$ has been constructed, the composed map is expressed by

$$(\tilde{T}^2 \circ \tilde{T}^1)(\theta) = \tilde{T}^2(\tilde{T}^1(\theta)) = \tilde{r}^2 \overset{i.d.}{\approx} r. \tag{2.38}$$

Notice that $\tilde{T}^1$ and $\tilde{T}^2$ do not necessarily need to be lower triangular as long as they are invertible transformations. The basic idea here is that $\tilde{T}^2$ will correct for the error remaining after an application of $\tilde{T}^1$. We can even repeat this layering process N times to obtain a compositional map of the form

$$(\tilde{T}^N \circ \tilde{T}^{N-1} \circ \cdots \circ \tilde{T}^2 \circ \tilde{T}^1)(\theta) = \tilde{r}^N \overset{i.d.}{\approx} r. \tag{2.39}$$

Layer i of this composition will be constructed using the tools of Section 2.2 and

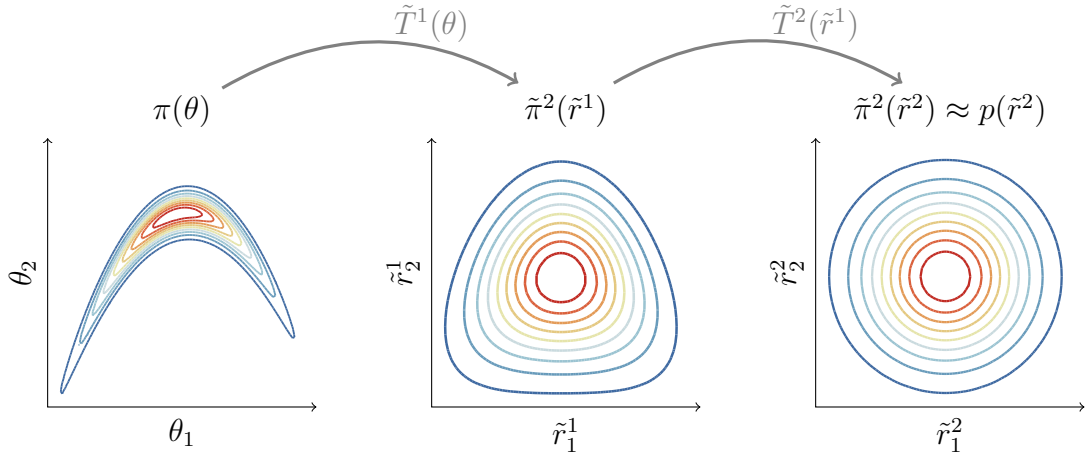


Figure 2-8: The goal of map composition is to introduce new layers that correct for errors that stem from using simple map parameterizations. The density on the left is the target distribution, the density in the middle is the density after transforming θ with $\tilde{T}^1$, and the right density is the effect of transforming the target distribution with $(\tilde{T}^2 \circ \tilde{T}^1)(\theta)$.

samples $\{\tilde{r}^{i-1,(1)}, \tilde{r}^{i-1,(2)}, \ldots, \tilde{r}^{i-1,(K)}\}$ defined by

$$\tilde{r}^{i,(k)} = (\tilde{T}^i \circ \tilde{T}^{i-1} \circ \cdots \circ \tilde{T}^2 \circ \tilde{T}^1)(\theta^{(k)}). \tag{2.40}$$

Importantly, because of the composition, each map $\tilde{T}^i$ in (2.39) can be a rough approximation constructed from a small number of basis functions. Moreover, under some mild conditions on the choice of basis functions and the map construction, each layer in (2.39) will decrease the overall error of the composed map. In next section will further discuss this error analysis.

2.7.2 Error of composed maps

Here we theoretically study the error and convergence of the compositional map. An important quantity in this study will be the map-induced density $\tilde{\pi}^i$, that approximates the target density π using the first i layers of the map. Using the compositional form of the map, this approximate density is given by

$$\begin{aligned}
\tilde{\pi}^i(\theta) &= p\left((\tilde{T}^i \circ \tilde{T}^{i-1} \circ \cdots \circ \tilde{T}^1)(\theta)\right) \left|\det\left[\partial\left(\tilde{T}^j \circ \tilde{T}^{j-1} \circ \cdots \circ \tilde{T}^1\right)(\theta)\right]\right| \\
&= p\left((\tilde{T}^i \circ \tilde{T}^{i-1} \circ \cdots \circ \tilde{T}^1)(\theta)\right) \prod_{j=1}^{i} \left|\det\left[\partial\tilde{T}^j\left((\tilde{T}^{j-1} \circ \tilde{T}^{j-2} \circ \cdots \circ \tilde{T}^1)(\theta)\right)\right]\right|,
\end{aligned} \tag{2.41}$$

where $|\det(\cdot)|$ is the absolute determinant and $\partial\tilde{T}^j(\cdot)$ is the Jacobian matrix of $\tilde{T}^j$. Notice that (2.41) is simply a multilayer version of the standard change of variable

formula in (2.6). Indeed, when only one layer is present, the two expressions in (2.41) and (2.6) are identical.

Our algorithm constructs one map at a time - first $\tilde{T}^1(\theta)$, then $\tilde{T}^2(\tilde{r}^1)$, etc. During this process, samples of $\tilde{r}^{i-1}$ are used in the optimization problem (2.21) to find $\tilde{T}^i$. Thus, each incrementally constructed map attempts to minimize $D_{KL}(\tilde{\pi}^{i-1}\|\tilde{\tilde{\pi}}^{i-1})$ where $\tilde{\tilde{\pi}}^{i-1}$ is the incremental pull-back measure defined by

$$\tilde{\tilde{\pi}}^{i-1}(\tilde{r}^{i-1}) = p(\tilde{T}^i(\tilde{r}^{i-1}))\left|\det(\partial\tilde{T}^i(\tilde{r}^{i-1}))\right|. \tag{2.42}$$

However, we ultimately want to find a map $\tilde{T}^i$ that minimizes $D_{KL}(\pi\|\tilde{\pi}^i)$. Fortunately, we can show the minimizer of $D_{KL}(\tilde{\pi}^{i-1}\|\tilde{\tilde{\pi}}^{i-1})$ is equivalent to the minimizer of $D_{KL}(\pi\|\tilde{\pi}^i)$ when previous maps $\{\tilde{T}^1, \tilde{T}^2, \ldots, \tilde{T}^{i-1}\}$ are fixed. Mathematically, we show this by simplifying the optimizations to remove terms that do not depend on $\tilde{T}^i$. First, consider the minimizer of $D_{KL}(\tilde{\pi}^{i-1}\|\tilde{\tilde{\pi}}^{i-1})$, which is also the minimizer of the following simplifications of $D_{KL}(\tilde{\pi}^{i-1}\|\tilde{\tilde{\pi}}^{i-1})$

$$\begin{aligned}
\operatorname*{argmin}_{\tilde{T}^i} D_{KL}(\tilde{\pi}^{i-1}\|\tilde{\tilde{\pi}}^{i-1}) &= \operatorname*{argmin}_{\tilde{T}^i} \mathbb{E}_{\tilde{\pi}^{i-1}}\left[\log\tilde{\pi}^{i-1}(\tilde{r}^{i-1}) - \log\tilde{\tilde{\pi}}^{i-1}(\tilde{r}^{i-1})\right] \\
&= \operatorname*{argmin}_{\tilde{T}^i} \mathbb{E}_{\tilde{\pi}^{i-1}}\left[-\log\tilde{\tilde{\pi}}^{i-1}(\tilde{r}^{i-1})\right] \\
&= \operatorname*{argmin}_{\tilde{T}^i} \mathbb{E}_{\tilde{\pi}^{i-1}}\left[-\log p(\tilde{T}^i(\tilde{r}^{i-1})) - \log\left|\det(\partial\tilde{T}^i(\tilde{r}^{i-1}))\right|\right] \\
&= \operatorname*{argmin}_{\tilde{T}^i} \mathbb{E}_{\pi}\Big[-\log p\left((\tilde{T}^i\circ\tilde{T}^{i-1}\circ\cdots\circ\tilde{T}^1)(\theta)\right) \\
&\qquad -\log\left|\det\left[\partial\tilde{T}^i\left((\tilde{T}^{i-1}\circ\tilde{T}^{i-2}\circ\cdots\circ\tilde{T}^1)(\theta)\right)\right]\right|\Big].
\end{aligned} \tag{2.43}$$

Similarly, we can look at the minimization of $D_{KL}(\pi\|\tilde{\pi}^i)$ and remove any terms that do not depend no $\tilde{T}^i$. These simplifications yield the minimizer

$$\begin{aligned}
\operatorname*{argmin}_{\tilde{T}^i} D_{KL}(\pi\|\tilde{\pi}^i) &= \operatorname*{argmin}_{\tilde{T}^i} \mathbb{E}_{\pi}\left[\log\pi(\theta) - \log\tilde{\pi}^i(\theta)\right] \\
&= \operatorname*{argmin}_{\tilde{T}^i} \mathbb{E}_{\pi}\Bigg[-\log p\left((\tilde{T}^i\circ\tilde{T}^{i-1}\circ\cdots\circ\tilde{T}^1)(\theta)\right) \\
&\qquad -\sum_{j=1}^{i}\log\left|\det\left[\partial\tilde{T}^j((\tilde{T}^{j-1}\circ\tilde{T}^{j-2}\circ\cdots\circ\tilde{T}^1)(\theta))\right|\right]\Bigg] \\
&= \operatorname*{argmin}_{\tilde{T}^i} \mathbb{E}_{\pi}\Big[-\log p\left((\tilde{T}^i\circ\tilde{T}^{i-1}\circ\cdots\circ\tilde{T}^1)(\theta)\right) \\
&\qquad -\log\left|\det\left[\partial\tilde{T}^i\left((\tilde{T}^{i-1}\circ\tilde{T}^{i-2}\circ\cdots\circ\tilde{T}^1)(\theta)\right)\right|\right]\Big]. \\
&= \operatorname*{argmin}_{\tilde{T}^i} D_{KL}(\tilde{\pi}^{i-1}\|\tilde{\tilde{\pi}}^{i-1}).
\end{aligned} \tag{2.44}$$

From (2.43) and (2.44), we can see that constructing $\tilde{T}^i$ incrementally by minimizing

(2.43) is equivalent to a greedy procedure applied to the overall KL divergence in (2.44) – we are minimizing the right thing. However, does this greedy approach ensure $D_{KL}(\pi\|\tilde{\pi}^i)$ decreases with each new layer? As we show below, the answer is "yes," when the optimization is performed in a properly defined function space.

We now derive sufficient conditions on $\tilde{T}^i$ to ensure the KL divergence $D_{KL}(\pi\|\tilde{\pi}^i)$ decreases or remains the same with each new layer. Mathematically, these sufficient conditions will place constraints on $\tilde{T}^i$ to ensure that for all i,

$$D_{KL}(\pi\|\tilde{\pi}^{i+1}) \leq D_{KL}(\pi\|\tilde{\pi}^i). \tag{2.45}$$

By writing out the KL divergences explicitly and simplifying, we obtain the condition

$$\begin{aligned}&\mathbb{E}_\pi\left[-\log\left(p\left((\tilde{T}^{i+1}\circ\tilde{T}^i\circ\cdots\circ\tilde{T}^1)(\theta)\right)\right)-\log\left|\det\left(\partial\tilde{T}^{i+1}\left((\tilde{T}^i\circ\tilde{T}^{i-1}\circ\cdots\circ\tilde{T}^1)(\theta)\right)\right)\right|\right]\\&\leq\mathbb{E}_\pi\left[-\log\left(p\left((\tilde{T}^i\circ\tilde{T}^{i-1}\circ\cdots\circ\tilde{T}^1)(\theta)\right)\right)\right].\end{aligned} \tag{2.46}$$

Notice that we no longer have the determinant sum on the left hand side of this inequality and there are no determinant terms on the right hand side. The extra determinant terms appeared on both sides of the inequality and were removed during simplifications. The left hand side of this expression is also equivalent to the objective function we minimize to find the map $\tilde{T}^{i+1}$ in (2.21). We should also point out that right hand side of (2.46) is equivalent to left hand side of (2.46) when $\tilde{T}^{i+1}(\tilde{r}^i)$ is the identity map, so $\tilde{T}^{i+1}(\tilde{r}^i)=\tilde{r}^i$. This observation is important for us to show that each layer of the map will yield a better approximation to the target distribution.

Let $\mathcal{T}^{i+1}$ be a function space containing all possible maps $\tilde{T}^{i+1}(\tilde{r}^i)$. In practice, $\mathcal{T}^{i+1}$ is the span of the basis functions used in either (2.23) or (2.24) to describe the map. When the space of maps $\mathcal{T}^{i+1}$ includes the identity, it is possible for the left hand side of the inequality in (2.46) to be equal to the right hand side. Thus, because $\tilde{T}^{i+1}$ minimizes the KL cost over $\mathcal{T}^{i+1}$ by construction (see (2.7)), the left hand side of (2.46) will always be less than or equal to the right hand side. Equality will only occur when the identity map is optimal.

We construct the map using Monte Carlo approximations to (2.46), not the exact expectation used in (2.46). In the Monte Carlo case, the decreasing error condition in (2.46) becomes

$$\begin{aligned}&\sum_{k=1}^{K}-\log\left(p\left((\tilde{T}^{i+1}\circ\tilde{T}^i\circ\cdots\circ\tilde{T}^1)(\theta^{(k)})\right)\right)-\log\left|\det\left(\partial\tilde{T}^{i+1}\left((\tilde{T}^i\circ\tilde{T}^{i-1}\circ\cdots\circ\tilde{T}^1)(\theta^{(k)})\right)\right)\right|\\&\leq\sum_{k=1}^{K}-\log\left(p\left((\tilde{T}^i\circ\tilde{T}^{i-1}\circ\cdots\circ\tilde{T}^1)(\theta^{(k)})\right)\right),\end{aligned} \tag{2.47}$$

where K is the number of samples used in the Monte Carlo integration. As in (2.46), this condition will hold assuming $\mathcal{T}^{i+1}$ contains the identity. Notice however, that for the Monte Carlo condition in (2.47) to hold, the same samples must be used to construct each layer of the map.

2.7.3 Monitoring convergence

Ideally, we would like to continue to add layers until we observe the KL divergence $D_{KL}(\pi\|\tilde{\pi}^i)$ drop below some tolerance. Unfortunately, we cannot use the KL divergence as a stopping criteria directly because computing $D_{KL}(\pi\|\tilde{\pi}^i)$ requires density evaluations of the target density $\pi(\theta)$ and we assume that only samples of $\pi(\theta)$ are available. The sample mean, covariance, and other moments could be used; however, simply matching a few moments does not necessarily indicate that $\tilde{\pi}^i$ is close to π, or analogously, that the output of the map is close to Gaussian, i.e., that $\tilde{r}^i \overset{i.d.}{\approx} r$. Nonlinear dependencies may exist that are not captured by the moments. Marginal quantiles are an easily computed alternative that provide a more sensitive estimate of the map error.

Our ultimate goal is to construct a compositional map whose output is jointly Gaussian. By definition, r is jointly Gaussian if and only if $a^T r$ has a normal distribution for any vector $a \in \mathbb{R}^{D_\theta}$ with unit norm. Thus, for an inexact map with output $\tilde{r}^i$, an indication of the error along the a-direction is the difference between quantiles of $a^T\tilde{r}^i$ and the known standard normal quantiles of $a^T r$. Moreover, quantiles can be compared in many directions to get a more accurate measure of the "non-Gaussianity" in $\tilde{r}^i$.

Assume we have N_q probabilities $\{p_1, p_2, \ldots, p_{N_q}\}$ evenly spaced over $(0, 1)$. The quantiles of a standard normal distribution at these levels are given by

$$Q_j = \sqrt{2}\,\mathrm{erf}^{-1}(2p_j - 1). \tag{2.48}$$

Now, assume we have N_a directions defined by $\{a_1, a_2, .., a_{N_a}\}$, where $a_i \in \mathbb{R}^{D_\theta}$ and $\|a_i\| = 1$. Using the N_q quantiles and N_a directions, we can define the error e_q of the samples $\{\tilde{r}^{i,(1)}, \tilde{r}^{i,(2)}, \ldots, \tilde{r}^{i,(K)}\}$ as

$$e_q(\tilde{r}^{i,(1)}, \tilde{r}^{i,(2)}, \ldots, \tilde{r}^{i,(K)}) = \sum_{d=1}^{N_a}\sum_{j=1}^{N_q}\left[\hat{Q}_j\left(a_d^T\tilde{r}^{i,(1)}, a_d^T\tilde{r}^{i,(2)}, \ldots, a_d^T\tilde{r}^{i,(K)}\right) - Q_j\right]^2, \tag{2.49}$$

where $\hat{Q}_j(\cdot)$ is a sample estimate of the quantile with probability p_j and $a_d^T\tilde{r}^{i,(k)}$ is the projection of the k^{th} sample of $\tilde{r}^i$ onto the d^{th} direction used in the error estimate. We will use this error estimate as both a stopping criteria and an indication of how to rotate the coordinates when constructing each layer of the map.

2.7.4 Choosing rotations

In Section 2.2, our formulation makes extensive use of lower triangular maps to help reduce the computational complexity of map construction. While we know an exact lower triangular matrix exists, it is often advantageous to rotate the coordinates of the parameter space (i.e., apply a linear transformation to $\tilde{r}^i$), before constructing the lower triangular map. Rotating the coordinates before building the nonlinear map is especially important when a small number of basis functions parameterize $\tilde{T}^i$ and the map alone is incapable of capturing nonlinear dependencies between all components

of θ. Introducing a new rotation at each stage of the map can allow each layer to capture new nonlinear structure. However, without paying special attention to the transformation, the linear transformation can worsen the map performance and even prevent the KL divergence from decreasing as we would expect from (2.47).

To see this, let $\tilde{P}^i : \mathbb{R}^{D_\theta} \to \mathbb{R}^{D_\theta}$ be a lower triangular map in a function space $\mathcal{P}^i$. Here again, $\mathcal{P}^i$ is the span of the finite number of basis functions used to parameterize $\tilde{P}^i$. Moreover, let $\mathcal{T}_{\mathcal{A}}{}^i$ represent possible transformations obtained by composing any map in $\mathcal{P}^i$ with an invertible linear transformation $A^i \in \mathbb{R}^{D_\theta \times D_\theta}$. Thus, elements of $\mathcal{T}_{\mathcal{A}}{}^i$ will take the form of $\tilde{T_A}{}^i(\tilde{r}^{i-1}) = \tilde{P}^i(A^i\tilde{r}^{i-1})$. Unfortunately, we can no longer guarantee that $\mathcal{T}_{\mathcal{A}}{}^i$ contains the identity – just consider the case where A^i is completely dense and $\tilde{P}^i$ is lower triangular. We make sure the map includes the identity by introducing an orthonormal matrix $B^{(i)} \in \mathbb{R}^{D_x \times D_x}$ so that the map takes the form

$$\tilde{T}^i(\tilde{r}^{i-1}) = B^i\tilde{P}^i(A^i\tilde{r}^{i-1}). \tag{2.50}$$

Note that we require B^i to be orthonormal so that we can build $\tilde{P}^i$ by solving (2.21) and driving the output of $\tilde{P}^i$ to a standard normal distribution. A general choice of B^i would require us to build $\tilde{P}^i$ with something other than a standard normal reference distribution and we would lose all the computational advantages pertaining to (2.21).

Notice that the completed one-layer map $\tilde{T}^i$ lies in the function space $\mathcal{T}^i$ containing maps of the form in (2.50). This space is dependent on the choices of A^i and B^i. By strategically choosing these matrices, we can ensure that $\mathcal{T}^i$ contains the identity, implying that each additional layer will decrease the overall map error as in (2.50).

When A^i is also orthonormal, the inverse of A^i is a straightforward choice for B^i. Qualitatively, this choice will rotate the coordinate system of $\tilde{r}^{i-1}$, apply the nonlinear map, and then rotate the result back to the original coordinate system. Moreover, with this choice of A^i and B^i, the nonlinear map $\tilde{P}^i$ can be either diagonal or lower triangular. In either of these cases, $\mathcal{T}^i$ will contain the identity and each additional layer in the composed map will decrease the overall error.

In some situations, it may be advantageous to use an arbitrary invertible matrix for A^i whose columns are not orthogonal. Because B^i needs to be orthonormal, we can not use the inverse of A^i in this case. However, when $\tilde{P}^i$ is lower triangular, we can still ensure $\mathcal{T}^i$ contains the identity by using a QL decomposition of A^i to define B^i. A QL decomposition is similar to the typical QR factorization of an invertible matrix; however, the QL decomposition yields an orthonormal matrix Q and a *lower* triangular matrix L instead of the usual upper triangular R. Assume we have such a decomposition, given by

$$A^i = Q^iL^i, \tag{2.51}$$

where Q^i is orthonormal, and L^i is lower triangular with positive diagonal coeffi-

cients.[5] We then have

$$(A^i)^{-1} = (L^i)^{-1}(Q^i)^T, \tag{2.52}$$

where $(L^i)^{-1}$ is again a lower triangular matrix with positive diagonal entries. This identity provides a way to choose B^i in (2.50). When $(L^i)^{-1}\tilde{r}^{i-1}$ is in the function space $\mathcal{P}^i$, we can find a map such that $\tilde{P}^i(\tilde{r}^{i-1}) = (L^i)^{-1}\tilde{r}^{i-1}$. With this choice of nonlinear component, and setting $B^i = (Q^I)^T$, the map at layer i becomes $\tilde{T}^i(\tilde{r}^{i-1}) = (Q^i)^T(L^i)^{-1}A^i\tilde{r}^{i-1} = \tilde{r}^{i-1}$. Thus, for any invertible A^i, choosing $B^i = (Q^i)^T$ ensures the identity map exists in $\mathcal{T}^i$. This subsequently implies that the decreasing KL divergence from (2.46) will hold.

The optimal[6] choice of A^i and B^i is problem dependent and may not be known a priori. However, we have found that reasonable instances of A^i can be computed by using either random rotations, principal components, or alternating between these two approaches. Furthermore, regardless of how an initial rotation is constructed, we will reorder the rows of A^i (and possibly columns of B^i) based on the sorting technique in 2.3.3 with $p = 2$. This helps the map[7] capture the correlation structure in the rotated coordinate system defined by A^i.

Principal component rotations

The principal components of a set of samples are the eigenvectors of the empirical covariance matrix or the left singular vectors of a matrix of samples. Let $\hat{r}^i$ be the sample average of $\tilde{r}^i$ and define the matrix X as

$$X = \left[\tilde{r}^{i-1,(1)} - \hat{r}^i, \tilde{r}^{i-1,(2)} - \hat{r}^i, \ldots, \tilde{r}^{i-1,(K)} - \hat{r}^i\right] \tag{2.53}$$

Notice that each column of X contains a single shifted sample. The left singular vectors of X, defined by U in the singular value decomposition (SVD) $X = U\Sigma V^T$, define the principal components of the samples. The matrix U is orthonormal and defines a rotation of X into linearly uncorrelated samples $Y = U^T X$, i.e., $YY^T = I$. In this way, $A^i = U^T$ and $B^i = U$ seem like natural choices. In fact, with this choice of rotations and a Gaussian $\pi(\theta)$, a linear diagonal $\tilde{P}$ would yield an *exact* map. However, as shown in Figure 2-11 and Table 2.3, principal component rotations can hinder the convergence speed of the map. This is because the principal components capture linear correlations but do not consider nonlinear correlations and therefore, once most of the linear correlation in $\pi(\theta)$ has been captured, result in nearly constant rotations, i.e., $A^i \approx A^{i+1}$. When $\pi(\theta)$ is non-Gaussian, the remaining nonlinear dependencies can be difficult or impossible to capture with the nearly constant principal components. One way to overcome this is to simply use random rotations. With a

[5] While classic methods for computing a QL decomposition, such as modified Gram-Schmidt, may not produce an L^i with positive diagonal coefficients, we can always transform such a solution to one with positive diagonal entries by multiplying appropriate columns of Q and rows of L with -1.

[6] The optimal choice of each A^i and B^i in a composed map with N layers will minimize the KL divergence in (2.46).

[7] Reordering the components is helpful except when the map is diagonal. For a diagonal map, reordering has no effect on the map performance.

random A^i, the rotations will continue to change with additional layers and we should therefore be able to capture more of the nonlinear correlations.

Purely random rotations

A random orthonormal matrix R in $\mathbb{R}^{D_\theta \times D_\theta}$ dimensions can be constructed by first generating a matrix of independent standard normal samples and then orthogonalizing the columns of this Gaussian matrix with a modified Gram-Schmidt procedure. This approach for generating R is shown in algorithm 2.2. The algorithm produces matrices whose columns are uniformly distributed on the unit ball in D_θ dimensions.

At each layer in our compositional map, we generate a new R and set $A^i = R$ and $B^i = R^T$. While the rotations using this random scheme will never get "stuck" like the principal components, these random rotations do not take advantage of any information about the samples in X, or the current approximation $\tilde{\pi}^i(\theta)$. From our numerical experiments, we have seen that this leads to slow, but eventual convergence. One possibility for accelerating the convergence is to be more "choosy" about the random directions by favoring directions with non-Gaussian marginals.

Algorithm 2.2: Procedure for generating random orthonormal matrices uniformly spread over a D_θ-dimensional unit ball. This procedure uses a modified Gram-Schmidt procedure for orthogonalizing the columns of R. See §3.4.1 of [63] for further discussion of generating random points on a hypersphere.

Input: The dimension D_θ
Output: A random orthonormal matrix R.

Generate $R \in \mathbb{R}^{D_\theta \times D_\theta}$ such that $R_{ij} \sim N(0,1)$
for $i \leftarrow 1$ ***to*** D_θ **do**
  for $j \leftarrow 1$ ***to*** $i-1$ **do**
    $R_{:,i} \leftarrow R_{:,i} - (R_{:,i}^T R_{:,j}) R_{:,j}$
  $R_{:,i} = \frac{R_{:,i}}{\|R_{:,i}\|}$
return R

Choosy random rotations

Our ultimate goal is to create a transformation from the target density $\pi(\theta)$ to an iid standard normal density. Therefore, an intuitive heuristic for choosing the rotations A^i and B^i is to focus the map on directions whose marginal distribution is non-Gaussian. We want to choose A^i such that our samples of $y = A^i \tilde{r}^{i-1}$ maximize the quantile error in (2.49) for some pre-specified quantile levels $\{p_1, p_2, \ldots, p_{N_q}\}$.

Finding the globally optimal rotation is computationally infeasible. As an alternative, we randomly generate a set of M orthonormal matrices using algorithm 2.2 and out of these M options, choose the rotation R^* that has the maximum non-Gaussianity, as defined by (2.49). We then set $A^i = (R^*)^T$. Algorithm 2.3 illustrates this procedure. Notice that as the number of random orthonormal matrices $M \to \infty$,

this method becomes a random search global optimizer. Also notice that the purely random rotations discussed above a special case of these choosy random rotations when $M = 1$.

In practice we have found that choosing M between 100 and 1000 seems to work well. In our tests, values of M below this did not adequately find non-Gaussian directions. Moreover, values of M larger than this range did not retain enough randomness and suffered from the same issue as the principal components – the rotations would become limited to a few directions, limiting the maps ability to capture nonlinear dependencies.

Algorithm 2.3: Procedure for generating a random orthonormal matrices that will cluster on "non-Gaussian" directions.

Input: The samples X, and the number of trials M
Output: A random orthonormal matrix R with preference for non-Gaussian directions.

```
Set e_q^* = -1
Set m^* = 0
Initialize R^* = I
for m ← 1 to M do
    Generate R using algorithm 2.2
    Compute Y' = R^T X
    Compute quantile error, e_q, of Y' using (2.49).
    if e_q > e_q^* then
        e_q^* = e_q
        m^* = m
        R^* = R
return R^*
```

Hybrid rotation schemes

The principal component method of choosing A^i and the purely random method converge slowly for very different reasons. The principal component approach can get stuck on only a few rotations and will not be able to capture more nonlinear structure. On the other hand, the purely random approach will eventually expose all nonlinear correlations, but will converge slowly because it does not take into account any information about the sample distribution. This disparity indicates that combining these methods could create a strong hybrid approach. In the upcoming numerical examples, we demonstrate such a hybrid approach by alternating between a principal components rotation and a random rotation. The hope is that the random rotation will prevent the principal components from getting "stuck," but still better tune the rotation A^i for the specific correlation found in the samples.

2.7.5 Relationship to artificial neural networks

Artificial neural networks (also called multi-layer perceptrons) are a class of methods for regression and classification. Just like our approach, these methods use multiple layers of simple functions to approximate complex functions. Moreover, artificial neural networks have been well studied in the machine learning community, with theoretical studies, e.g., [50], [49], [7] showing that artificial neural networks can approximate very complicated functions to arbitrary accuracy. Artificial neural networks have also seen more recent use in "deep learning," where many layers are used in a network for unsupervised or semi-supervised learning [6].

Our use of rotations, then one dimensional functions, followed by weighting and re-rotation, is a simple multilayered network called a feed forward neural network. In future work, we will exploit this relationship to develop more efficient algorithms for constructing our compositional maps and to formalize a convergence theory. The greatest algorithmic improvement will likely come from jointly optimizing over the rotations and nonlinear functions.

In addition to regression and classification, researchers in the neural network community have also studied independent component analysis (ICA). The goal of ICA is similar to our use of transport maps: ICA aims to find a transformation that splits a random signal into independent components. A classic example of this is the "cocktail problem," where the goal is to separate out individual speakers from the garbled sound of a party. Many algorithms from this community try to find the most non-Gaussian independent components of a signal by finding a transformation that minimizes a Kullback-Lebler divergence [24, 14, 21, 53]. While similar to our approach, our use of transport maps also require Gaussianity of the independent components. Nevertheless, ICA provides a wealth of previous work that we may be able to adapt to our use of transport maps.

2.7.6 Numerical examples: layered maps

To illustrate the compositional map idea, we will use two examples. The first is a small two dimensional example using samples of a "banana" distribution, while our second example involves a much larger random field with 64 dimensions.

Banana distribution

In this example, we will build a map for a target distribution $\pi(\theta)$ defined by the transformation

$$\begin{aligned} \theta_1 &= r_1 \\ \theta_2 &= r_1^2 + r_2, \end{aligned}$$

where r_1 and r_2 are from a standard normal distribution $N(0,1)$. The corresponding target density is given by

$$\pi(\theta) = \frac{1}{2\pi} \exp\left[-0.5\left(\theta_1^2 + (\theta_2 - \theta_1^2)^2\right)\right]. \tag{2.54}$$

For this example, we use $20,000$ samples of $\pi(\theta)$ during map construction and we construct compositional maps with $N = 6$ layers. The goal of this example is to study the effect of the basis type and rotation type on the map convergence. With this in mind, we study three types of bases using one dimensional radial basis functions. First, we will use a completely diagonal map, where each nonlinear map $\tilde{P}^i$ in (2.50) is defined in a similar way to (2.24) by

$$\tilde{P}_d^i(\tilde{r}^{i-1}) = a_{d,0}^i + a_{d,1}^i \tilde{r}_d^{i-1} + \sum_{k=1}^{P_d} b_{d,k}\phi_{d,k}(\tilde{r}_d^{i-1}), \tag{2.55}$$

where $\phi_{d,i}(\tilde{r}_d^{i-1})$ is a univariate radial basis function. In Table 2.3, this type of map is denoted with a D.

The second basis type is linear in its lower triangular components, but still nonlinear along the diagonal. The form of this map is given by

$$\tilde{P}_d^i(\tilde{r}^{i-1}) = a_{d,0}^i + \sum_{k=1}^{d} a_{d,k}^i \tilde{r}_k^{i-1} + \sum_{j=1}^{P_d} b_{d,j}\phi_{d,j}(\tilde{r}_d^{i-1}). \tag{2.56}$$

The only difference between this expression and the diagonal map in (2.55) are the additional linear terms in the first sum. This linear lower triangular map is labeled LL in Table 2.3.

The last type of basis we use in this example also introduces nonlinear terms in other dimensions. However, one dimensional radial basis functions are still employed, so the expansion is still separable by dimension. The form of this nonlinear separable map is

$$\tilde{P}_d^i(\tilde{r}^{i-1}) = a_{d,0}^i + \sum_{k=1}^{d} a_{d,k}^i \tilde{r}_k^{i-1} + \sum_{k=1}^{d}\sum_{j=1}^{P_d} b_{d,j,k}\phi_{d,j,k}(\tilde{r}_k^{i-1}). \tag{2.57}$$

This type of map is denoted by LN in Table 2.3. In all these expansions (2.55), (2.56), and (2.57), the number of radial basis functions is $P_d = 31$. We use Gaussian radial basis functions evenly spaced between the 1% and 99% quantiles of the rotated samples $A^i\tilde{r}^i$.

In addition to the three map forms described above, we also investigate five different rotations types in this example. We will use the completely random rotations, choosy random rotations, and principal components (labeled SVD in Table 2.3), as well hybrid rotations based on alternating the principal components with completely random rotations (labeled by R-SVD) or choosy random rotations (labeled C-SVD). For the choosy rotations, we set the number of trial rotations to $M = 1000$ in algorithm 2.3.

Table 2.3: Convergence of compositional map for different map forms. The quantile based error from Section 2.7.3 is used with 6 random (but fixed between runs) directions. Each method was run 20 times. The mean and standard deviation of the errors computed from the 20 runs are provided here.

		Mean Error						Error Std. Dev.					
	Layer	1	2	3	4	5	6	1	2	3	4	5	6
Rand.	D	0.619	0.467	0.373	0.308	0.274	0.221	0.198	0.192	0.185	0.167	0.164	0.121
	LL	0.575	0.470	0.376	0.314	0.243	0.215	0.228	0.219	0.187	0.160	0.109	0.108
	LN	0.384	0.186	0.111	0.080	0.058	0.045	0.100	0.087	0.086	0.070	0.065	0.051
Choosy	D	0.431	0.152	0.125	0.072	0.058	**0.048**	0.102	0.037	0.039	0.012	0.014	0.010
	LL	0.393	0.176	0.129	0.067	0.063	**0.047**	0.076	0.027	0.048	0.014	0.013	0.009
	LN	0.314	0.219	0.112	0.067	0.049	**0.037**	0.070	0.053	0.025	0.026	0.014	0.011
SVD	D	0.405	0.170	0.142	0.133	0.128	0.126	0.087	0.042	0.025	0.030	0.029	0.030
	LL	0.404	0.154	0.135	0.128	0.124	0.115	0.090	0.031	0.024	0.027	0.032	0.030
	LN	0.307	0.276	0.275	0.234	0.233	0.233	0.065	0.086	0.082	0.101	0.101	0.101
R-SVD	D	0.774	0.533	0.258	0.259	0.205	0.197	0.156	0.232	0.064	0.074	0.046	0.072
	LL	0.404	0.316	0.160	0.139	0.114	0.113	0.103	0.109	0.023	0.043	0.034	0.043
	LN	0.322	0.264	0.228	0.155	0.138	0.116	0.048	0.060	0.083	0.055	0.064	0.063
C-SVD	D	0.393	0.184	0.155	0.115	0.071	0.064	0.093	0.029	0.027	0.042	0.012	0.012
	LL	0.398	0.178	0.152	0.119	0.072	0.062	0.101	0.028	0.031	0.052	0.018	0.018
	LN	0.313	0.211	0.137	0.087	0.078	0.053	0.071	0.048	0.050	0.032	0.034	0.018

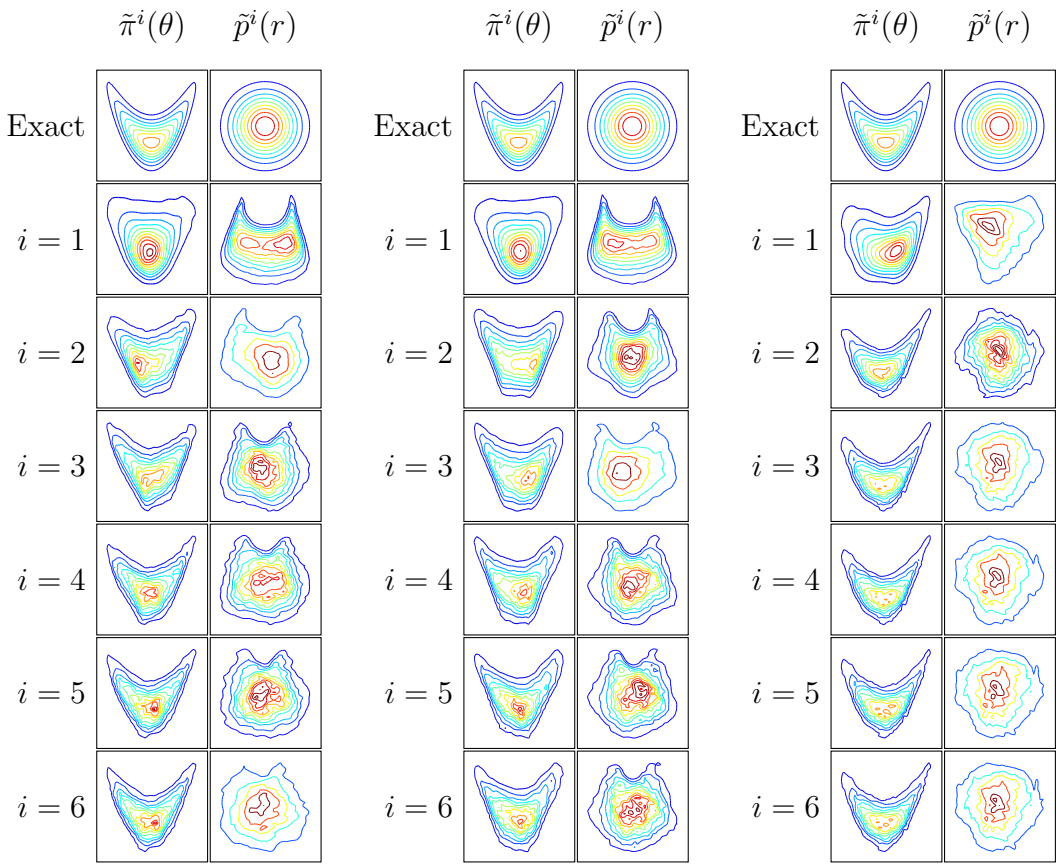
Convergence of the composed map can be studied on either the target θ side of the map, or on the reference r side of the map. More precisely, we can either compare the map-induced approximation $\tilde{\pi}^i$ to the true target density π or we can compare a map-induced reference density $\tilde{p}^i$ to the true Gaussian reference density, where $\tilde{p}^i$ is defined by the inverse map composition

$$\tilde{p}^i(r) = \pi(\tilde{F}^1 \circ \tilde{F}^2 \circ \cdots \circ \tilde{F}^i(r)) \prod_{j=1}^{i} \left| \det D\tilde{F}^j((\tilde{F}^{j+1} \circ \tilde{F}^{j+2} \circ \cdots \circ \tilde{F}^i)(r)) \right|, \quad (2.58)$$

where $\tilde{F}^i(r) \approx \tilde{T}^{i,-1}(r)$ is an approximation to the inverse of $\tilde{T}^i$ constructed using the regression approach from Section 2.5. Both the convergence of $\pi^i(\theta)$ to the true target density and $\tilde{p}^i(r)$ to the true reference density are shown in Figures 2-9 through 2-13.

For each map form, Figure 2-9 shows the compositional map convergence using up to 6 layers. Moreover, Figure 2-10 shows the convergence with choosy rotations, Figure 2-11 shows convergence with principal components, Figure 2-12 shows results using R-SVD rotations, and Figure 2-13 shows convergence with C-SVD rotations. The convergence plots come from one run of each approach and should be treated as typical results. A more thorough error analysis is provided in Table 2.3, where the average quantile errors over 20 runs are reported.

Looking at Table 2.3, we see slow but steady convergence when using completely random rotations and stalled convergence when using principal components. On the other hand, focusing directions on non-Gaussian directions with the choosy random



(a) Diagonal Map (b) Linear Lower (c) Nonlinear Separable

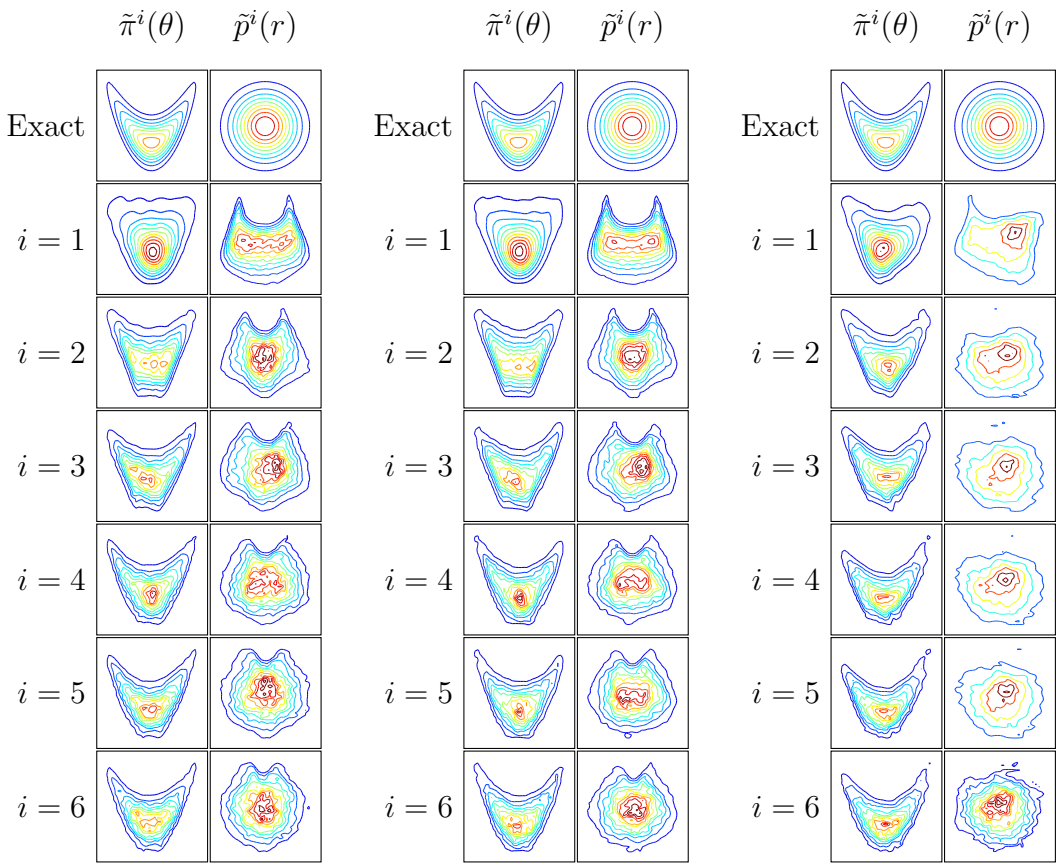
Figure 2-9: Typical convergence of layered map using purely random rotations.

approach, or alternating between a random approach and principal components, can converge quite quickly. This fast convergence can also be seen in Figures 2-10, 2-12, and 2-13. Remember that the algorithm is random, so the plots in these figures are just one realization of a possible outcome.

Importantly, the diagonal map has comparable performance to the more sophisticated lower triangular maps when using either choosy or C-SVD rotations. This is important as we move beyond this simple two dimensional problem to larger dimensional problems – the number of coefficients in one output of the map $\tilde{T}_d$, is constant with the parameter dimensions D_θ with a diagonal map.

Besov random field

In small dimensional problems such as the banana example above, the map composition method is not necessary. In these low dimensional problems, we can make a single layered transport map complicated enough to capture all the nonlinear correlations and structure in $\pi(\theta)$. It is only in high dimensional problems that we are really required to use the multi-layered map. This section gives an example of constructing a map with a 64 dimensional Besov space prior density.



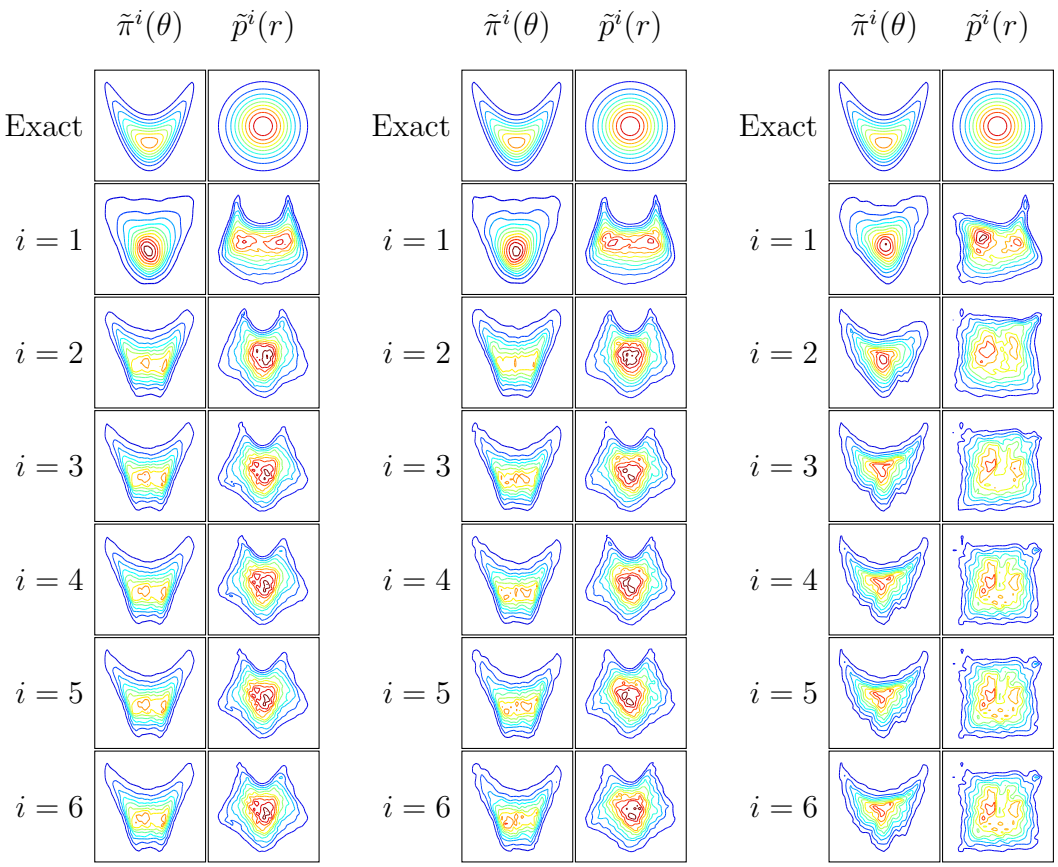
(a) Diagonal Map (b) Linear Lower (c) Nonlinear Separable

Figure 2-10: Typical convergence of layered map using choosy random rotations.

Assume our target random variable θ is actually a discretization of an infinite dimensional random field. Besov prior distributions are used by [64] and [28] to define both discretization-invariant and sparsity inducing priors for Bayesian inference. These prior distributions are more applicable than classic Gaussian smoothness priors when the target random field contain a few sharp edges and but is otherwise piecewise regular. Mathematically, a Besov prior is obtained by representing the random field with a wavelet expansion that has independent Laplace-distributed weights. After truncating the expansion and discretizing the field, we are left with the following definition of θ

$$\theta_i = \sum_{k=1}^{L} \sum_{j=1}^{N_k} c_{j,k} \psi_{j,k}(x_i), \tag{2.59}$$

where k is the level of the wavelet expansion, L is the number of levels used in the expansion, j is the index of the wavelet function at level k, N_k is the number of wavelets at level k, and $\psi_{j,k}$ is the j^{th} wavelet at level k of the multi resolution wavelet expansion. Notice that x_i is the i^{th} node in the spatial discretization, implying that θ_i is an evaluation of the random field at x_i. The coefficients $c_{j,j}$ in the expansion are



(a) Diagonal Map (b) Linear Lower (c) Nonlinear Separable

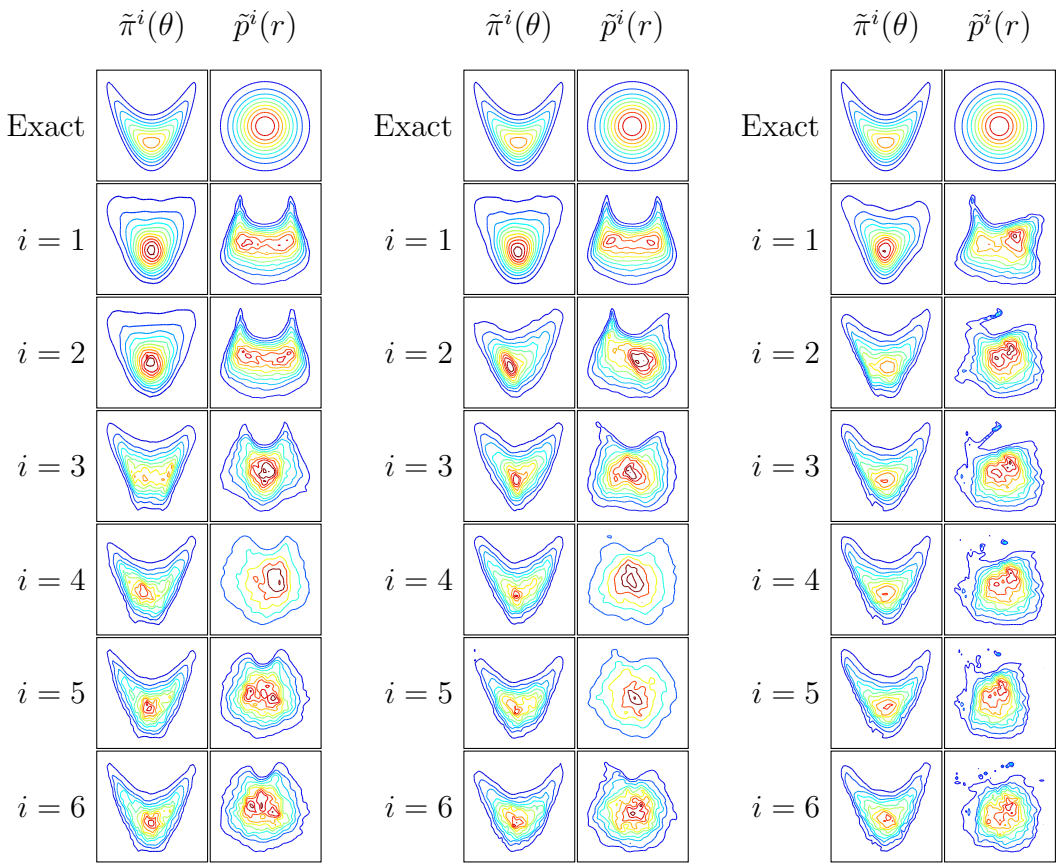
Figure 2-11: Typical convergence of layered map using only principal components.

distributed according to a Laplace distribution defined by the density

$$q(c_{i,j}) = \frac{1}{2b} \exp\left(-\frac{|c_{i,j}|}{b_k}\right), \tag{2.60}$$

where b_k controls the width of the density around $c_{i,j} = 0$. In this example, we use $L = 3$ levels in the wavelet expansion and choose $b_k = 2^{-2k+4}$. This choice of wavelet coefficient will result in a wider distribution on the "coarse" wavelets, and a tighter distribution on the "fine" wavelets. Combined with our use of Haar wavelets, this yields the step behavior shown in Figure 2-14(a). Notice that in the true field realizations there are large steps corresponding to the coarse wavelets while only smaller fine scale variations are present from the fine wavelets. The locations of the large jumps in the field are also fixed. With $L = 3$ layers, we can expect significant jumps at $i \in \{8, 16, 24, 32, 40, 48, 56\}$ because these are the edges of the most significant wavelets. This feature is also shown in Figure 2-14(a).

Using 50000 samples of the Besov random field defined by (2.59), we composed maps using the choosy random rotation algorithm from Section 2.7.4. Figure 2-15 summarizes the map performance. The diagonal (D), lower linear (LL), and lower

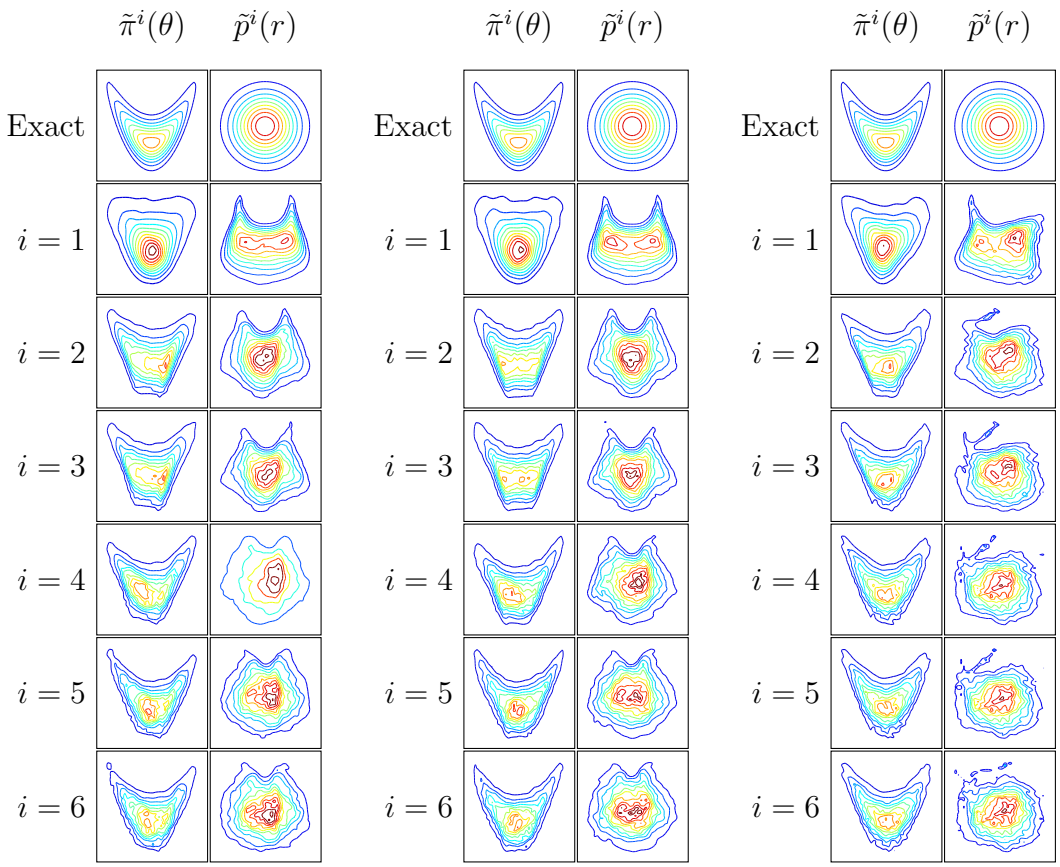


(a) Diagonal Map (b) Linear Lower (c) Nonlinear Separable

Figure 2-12: Typical convergence of layered map alternating SVD with random rotations.

nonlinear (LN), maps were constructed using 35 layers in the composition and 51 radial basis functions along the map diagonal at each layer. The diagonal map captures the mean and standard deviation, of the field, but clearly has trouble capturing the correlation structure and the coarse and fine behavior exhibited by the true field. Introducing the lower triangular linear terms in the LL map allows the map to capture more of this structure, but the distinct jumps shown in the realizations of the true field do not seem to be as present in the map-based samples. Looking more closely at the correlation structure shown in Figure 2-15, we can see why.

Figure 2-15 shows univariate and bivariate marginal distributions between 6 components of θ in the middle of the domain for $i \in \{30, 31, 32, 33, 34, 35\}$. Notice that in the middle of this segment, we have a boundary between coarse wavelets. This boundary causes the independence between the sets $\{\theta_{30}, \theta_{31}, \theta_{32}\}$ and $\{\theta_{33}, \theta_{34}, \theta_{35}\}$ as well as the strong correlation (caused by smaller values of the Laplace scale b_k) within these regions. Notice that the diagonal map captures the one dimensional marginal quite well, but does not seem to capture the strong correlation – this weaker correlation manifests in the more "random" looking realizations in Figure 2-14. On the other hand, the lower linear map captures more of the strong correlation, but does



(a) Diagonal Map (b) Linear Lower (c) Nonlinear Separable

Figure 2-13: Typical convergence of layered map alternating SVD with choosy rotations.

not capture the one dimensional marginals as well. This explains why the lower linear realizations in Figure 2-14 have more of the large-step behavior we desire.

The diagonal map captures the non-Gaussian marginal behavior but cannot fully capture the correlation. Introducing the linear lower triangular structure helps capture the correlation, but mangles the one dimensional marginals. Looking more closely at the first layer of the inverse map, $\tilde{F}^{35}(r)$, this mangling makes intuitive sense. Each component of the map $\tilde{F}_d^{35}(r)$ is linear in the first $d-1$ components of r. This means that the output of $\tilde{F}_d^{35}(r)$ is the sum of a Gaussian random variable and a non-Gaussian random variable defined by radial basis functions in the r_d direction. The additive Gaussian results in the more Gaussian looking marginals in Figure 2-15.

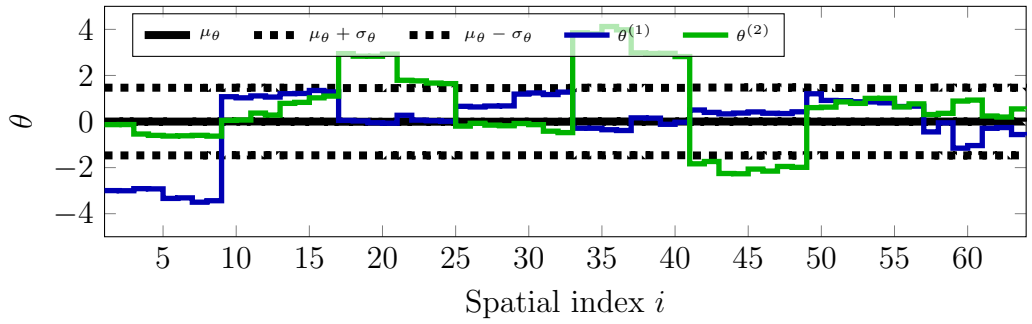
The impact of an additive Gaussian is likely to be an issue anytime a lower linear type map is used in highly non-Gaussian situations. To overcome this issue, nonlinear terms in all $d-1$ components can be used. This results in a map defined like the LN map in the banana example. The marginal distributions using such a map are shown in Figure 2-15(d). Using off-diagonal nonlinear linear terms allow the map to adequately capture both the non-Gaussian marginals and the correlation. Alternatively, a diagonal map could be used with a more strategic choice of rotation than

the random choosy rotations used in this example. The next section indicates one avenue for future work on this topic.

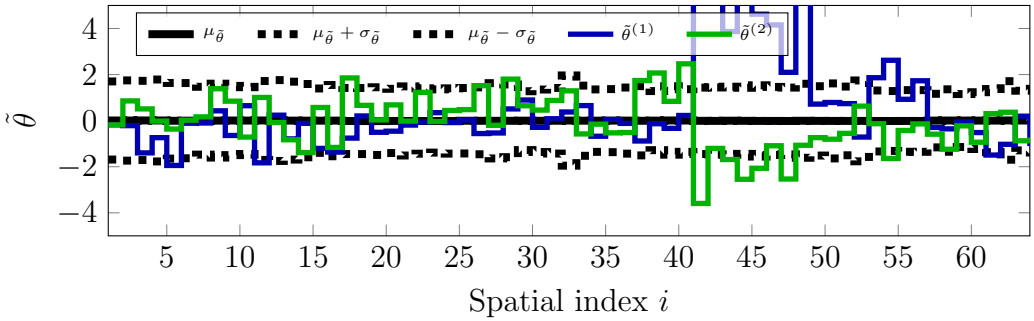
2.7.7 Summary

Constructing maps in high dimensional problems is difficult because obtaining sufficient accuracy with one-layer maps requires representing the map with an exorbitantly large number basis functions, which makes solving the optimization problem in (2.21) computationally intractable. In this section we have overcome the dimensionality issue by composing many maps into a sophisticated multilayer function. By introducing rotations we were able to tackle a large 64 dimensional problem. While the high dimensional problem required many layers to adequately describe the target density, future work investigating the relationship between layered maps and artificial neural networks will likely be able to reduce the number of necessary layers – especially in high dimensions.

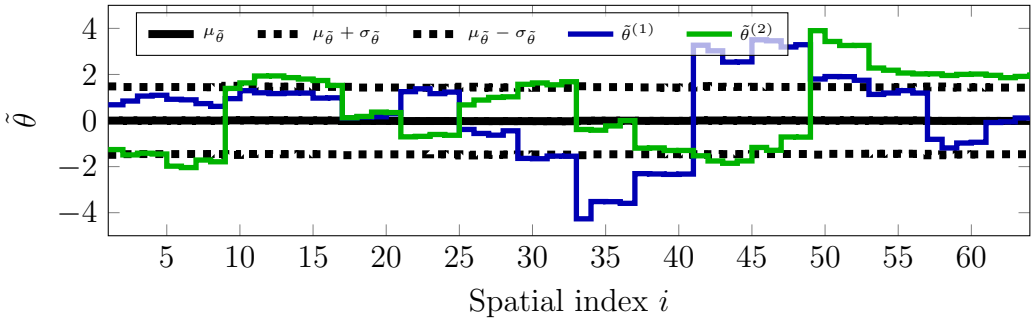
Efficient map construction (both in low and high dimensional settings) is a fundamental tool that we will exploit in every algorithm described in this thesis. Thus, any improvement in map construction will immediately allow us to tackle larger and more difficult problems.



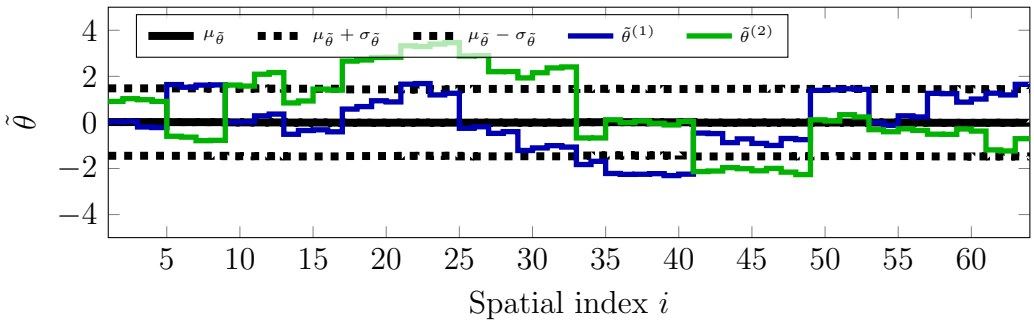
(a) Illustration of the target Besov random field. The mean and standard deviation are shown as well as two realizations.



(b) Summary of a diagonal composed map approximation to the Besov field.

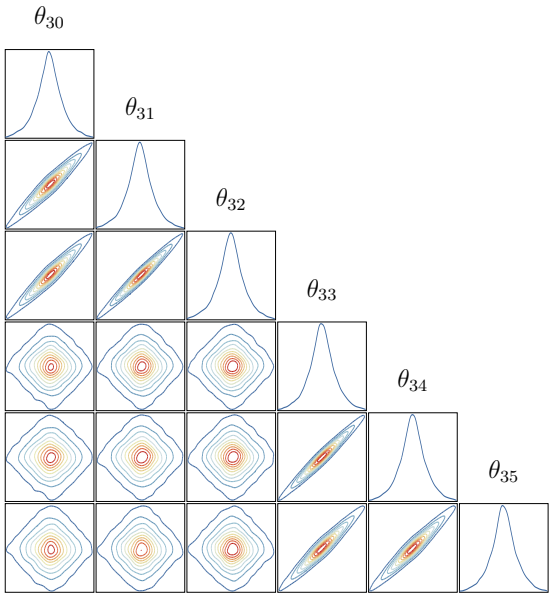


(c) Summary of a linear lower composed map approximation to the Besov field.

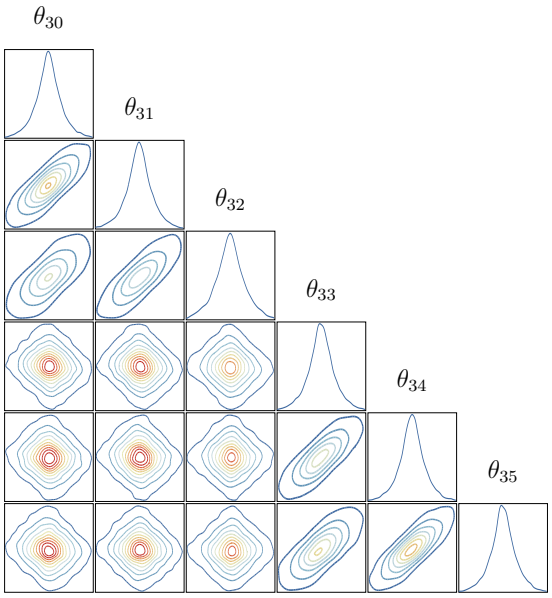


(d) Summary of a nonlinear lower composed map approximation to the Besov field.

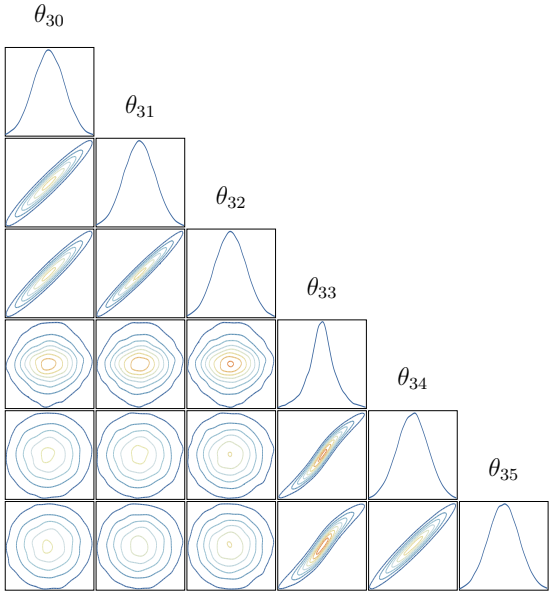
Figure 2-14: Comparison of true Besov random field with map approximation.



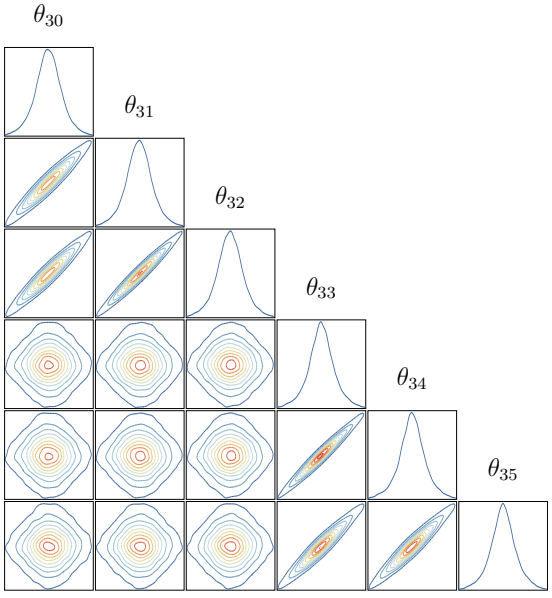
(a) True field



(b) Composed map with diagonal layers



(c) Composed map with linear lower triangular layers



(d) Composed map with nonlinear lower triangular layers

Figure 2-15: Comparison of map-induced joint marginal densities with true joint marginal densities. Note that the 6 marginal distributions shown here are just a subset of the full 64 dimensional target random variable θ.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Multiscale inference with transport maps

3.1 Introduction

While deterministic approaches are routinely used for large-scale inverse problems (e.g., [33, 18, 19]), these approaches cannot capture the impact of ill-posedness, which can stem from incomplete observations, observational error, or nonlinear physics, on parameter uncertainty. However, capturing this uncertainty can be critical when model predictions are to be used in future decision making and design [104, 36, 32, 101]. To capture these uncertainties, a statistical solution to the inverse problem is required. This work uses the Bayesian approach. In this chapter, the novelty of our approach stems from our use of a special multiscale structure exhibited in many Bayesian inference problems.

As described in Chapter 1, there are many approaches for generating posterior samples. However, most of these approaches suffer in high-dimensional parameter spaces, especially with expensive forward models. It is therefore not surprising that much of the current research in this area (including the work in this chapter) focuses on shrinking the parameter space exposed to the sampler and/or reducing the number of expensive model evaluations required by the sampler.

To reduce the parameter dimension for spatially distributed parameters, Karhunen-Loève (KL) expansions have proven quite useful [67, 31, 74]. Unfortunately, to effectively reduce the parameter dimension, the correlation structure of the field must be known a priori and the parameter field must be sufficiently smooth. Moreover, the parameter dimension is only part of the problem. Even with a smaller parameter space, an intractable number of expensive model evaluations can still be required to adequately sample the posterior. The direction taken in this work is thus to reduce both the parameter dimension exposed to the sampler *and* the computational cost of the forward model.

3.1.1 Overview of multiscale samplers

Multiscale inference methods explicitly take advantage of scale separation in the inference problem to reduce the number of expensive model evaluations required to generate posterior samples. Existing sampling strategies that exploit this scale separation do so in two ways: (i) by using a sequence of coarse and fine descriptions of the parameter field itself, or (ii) by using an efficient forward solver that exploits the scale separation. As examples of first strategy, [105] combines a sequential Monte Carlo method with a sequence of progressively finer grids to represent the parameter field. On the other hand, [46] uses parallel chains on both coarse and fine parameter fields in a reversible-jump MCMC setting. Occasional coarse to fine "swap" proposals allow information to be shared between scales. Alternatively, [31] and [29] pursue the second strategy. These works only consider a single-level discretization of the parameter field, but use a multiscale forward solver to create a more efficient MCMC proposal mechanism.

All of the existing multiscale sampling approaches above use the multiscale structure to accelerate *exact* sampling of the original fine scale posterior. Regardless of the approach, this still involves many online evaluations of the computationally expensive forward model. Our use of multiscale structure is fundamentally different. Instead of directly solving the full fine-scale problem, we create and solve an approximate low-dimensional inference problem that captures the intrinsic dimensionality of the original problem. The solution to this low dimensional coarse problem can then be "projected"[1] to the original high dimensional fine scale parameter space. While existing multiscale approaches can be likened to multigrid methods in linear algebra, our approach is more akin to reduced order modeling, where a lower dimensional problem is solved and a projection is used to create an *approximate* high dimensional solution. In our approach, transport maps are used extensively to describe the coarse inference problem as well as the coarse to fine "projection."

3.1.2 Multiscale definition

Multiscale is an inherently vague term because many systems have multiple time or spatial scales of one form or another. Our framework however, has a very specific requirement that we will use as the definition of a "multiscale" system. Consider two real valued random variables d and θ corresponding to probability spaces $(\mathcal{X}_d, \mathcal{F}_d, \mu_d)$ and $(\mathcal{X}_\theta, \mathcal{F}_\theta, \mu_\theta)$, where $\mathcal{X}_d \subseteq \mathbb{R}^{D_d}$, and $\mathcal{X}_\theta \subseteq \mathbb{R}^{D_\theta}$. We will say that a system mapping θ to d is a multiscale system if there is a *naturally* motivated sufficient statistic ζ such that d and θ are independent given ζ. Mathematically, this definition can be expressed as

$$\pi(d|\zeta, \theta) = \pi(d|\zeta). \tag{3.1}$$

[1] In the multigrid community, people would usually call this operation a prolongation and not a "projection".

To help ensure that working with the coarse likelihood $\pi(d|\zeta)$ is easier than working with the fine scale likelihood $\pi(d|\theta)$, ζ should have a smaller dimension than θ

$$D_\zeta \leq D_\theta, \tag{3.2}$$

where D_ζ is the dimension of the coarse parameter. Even though this definition of a multiscale system might seem abstract, many real systems exhibit behavior that approximately satisfies 3.1. For example, observations d of the pressure in an aquifer can often be well modeled by an "upscaled" permeability field ζ, instead of a very high dimensional field described by θ. Given the "upscaled" field ζ, the fine scale field θ does not introduce additional information that is important to describe the pressure observations d, which yields the conditional independence in (3.1). Additional examples can be found in ecology, finance, and other area with simulations that depend largely on the aggregate behavior of the parameter θ.

In most real systems, such as the subsurface flow example, the multiscale definition (3.1) is only approximately satisfied. All of the tools developed in this chapter can still be applied to these approximately multiscale systems, but the resulting posterior samples will also be approximate. However, this approximation is usually small and our use of the coarse parameter ζ allows us to tackle very large problems where directly sampling $\pi(\theta|d)$ is otherwise intractable. An example of such a large problem is given in Section 3.6.2.

As an example multiscale system, consider a deterministic fine scale model $f : \mathbb{R}^{D_\theta} \to \mathbb{R}^{D_d}$, a coarse model $\tilde{f} : \mathbb{R}^{D_\zeta} \to \mathbb{R}^{D_d}$, and an upscaling function $g : \mathbb{R}^{D_\theta} \to \mathbb{R}^{D_\zeta}$. The fine model relates the parameter θ to the data d with an additive Gaussian error model

$$d = f(\theta) + \epsilon \tag{3.3}$$

where $\epsilon \sim N(0, \Sigma_{\epsilon\epsilon})$. This additive Gaussian error defines the fine likelihood $\pi(d|\theta)$. Now, the coarse model and upscaling operator can be combined to approximate the fine scale model as follows

$$f(\theta) \approx \tilde{f}(g(\theta)) \quad = \quad \tilde{f}(\zeta). \tag{3.4}$$

Using these expressions with the additive error model allows us to define the joint likelihood $\pi(d|\zeta, \theta)$. Notice however, that only ζ appears in the coarse model (3.4). This observation allows us to write

$$\pi(d|\zeta, \theta) \approx \pi(d|\zeta), \tag{3.5}$$

where the approximation comes from the approximate coarse model in (3.4). When equality holds in (3.4), the coarse parameter ζ is a sufficient statistic and this system will satisfy our multiscale definition in (3.1) exactly, i.e., $\pi(d|\zeta, \theta) = \pi(d|\zeta)$ in (3.5). In practice however, the upscaling operation and coarse model only provide an approximation to the fine model $f(\theta)$. In this setting, ζ is only approximately a sufficient statistic, which means the approximate equality in (3.4) will remain. In this setting, our method can only generate approximate posterior samples; however,

when $\tilde{f}(g(\theta))$ is a good approximation to $f(\theta)$, the posterior can also be a good approximation. We will also have to make additional approximations in Section 3.3 to construct transport maps in high dimensions. With these additional approximations, the approximation in (3.5) will only be a mild concern when the coarse model $\tilde{f}(g(\theta))$ is an adequate approximation to the fine model $f(\theta)$.

Section 3.2 will show how the multiscale definition given in (3.1) allows us to decompose the usual Bayesian inference into two parts: characterizing a coarse posterior on ζ, and "projecting" the coarse posterior to the original fine posterior. After introducing the general framework, Section 3.3 will show that an appropriate transport map can be used to tackle both the coarse characterization and the coarse to fine "projection." Finally, Section 3.5 will demonstrate the efficiency of this multiscale approach on two large applications in porous media, where a Multiscale Finite Element method is used to simultaneously define the coarse parameter ζ and to facilitate efficient simulation. Before proceeding however, we would like to point out that our framework is not inextricably tied to MsFEM or any physical model, researchers could apply this framework to any problem where a coarse parameter ζ can be defined such that the conditional independence in (3.1) is a reasonable assumption.

3.2 Multiscale framework

3.2.1 Decoupling the scales with conditional independence

In the usual single scale setting, we combine the prior $\pi(\theta)$ and likelihood $\pi(d|\theta)$ with Bayes' rule

$$\pi(\theta|d) \propto \pi(d|\theta)\pi(\theta). \tag{3.6}$$

In the multiscale problem however, the likelihood has additional structure that we will use to develop an alternative expression of Bayes' rule involving the coarse parameter ζ. Without introducing any new ideas, we can use the coarse parameter ζ to rewrite Bayes' rule for the joint posterior over (θ, ζ) as

$$\pi(\theta, \zeta|d) \propto \pi(d|\theta, \zeta)\pi(\zeta, \theta). \tag{3.7}$$

Now, we assume the forward model is a multiscale system in the sense of (3.1). Because of the conditional independence in the multiscale definition, we can completely describe the forward model output with a set of coarse parameters ζ. With this in mind, we can simplify the right hand side of (3.7) and obtain

$$\pi(\theta, \zeta|d) \propto \pi(d|\zeta)\pi(\zeta, \theta). \tag{3.8}$$

We can further expand the joint prior, $\pi(\zeta, \theta)$, in this expression to create the following multiscale form of Bayes' rule

$$\pi(\theta, \zeta|d) \propto \pi(d|\zeta)\pi(\zeta)\pi(\theta|\zeta). \tag{3.9}$$

This simple expression, which stems directly from the multiscale definition, is the foundation of our multiscale framework. The three densities on the right hand side, $\pi(d|\zeta)$, $\pi(\zeta)$, and $\pi(\theta|\zeta)$, correspond to a coarse likelihood, a coarse prior, and a downscaling density. Notice that only downscaling density depends on high dimensional fine scale parameters θ.

3.2.2 Two stages for multiscale inference

We now take a deeper look at (3.9). Ignoring the downscaling density $\pi(\theta|\zeta)$ in (3.9), we have two remaining terms $\pi(d|\zeta)$ and $\pi(\zeta)$. These two densities form a coarse posterior density $\pi(\zeta|d) \propto \pi(d|\zeta)\pi(\zeta)$. Using this coarse posterior, we can break the sampling of the fine posterior $\pi(\theta|d)$ into two parts, (1) a coarse scale component that infers ζ directly by sampling $\pi(\zeta|d)$ without any concern for the fine scale parameters, and (2) a fine-scale component that for each coarse sample of $\pi(\zeta|d)$ will generate one or more samples of the fine scale posterior, $\pi(\theta|d,\zeta) = \pi(\theta|\zeta)$. The combination of these two steps will generate samples of the joint posterior $\pi(\theta,\zeta|d)$. Marginalizing out the coarse parameter (e.g., ignoring the coarse samples), will produce samples of the fine-scale posterior – our ultimate goal. While this two-step process is conceptually simple, there are two important issues that need to be considered:

1. Coarse scale sampling requires us to have a prior on the coarse parameter ζ, but the original inference problem defines a prior on the fine parameter θ.

2. Generating fine scale posterior samples requires us to sample from the conditional density $\pi(\theta|\zeta)$, a potentially nontrivial task.

Both of these problems will be addressed with a unique form of optimal transport map. Section 3.3 will show how this map can be constructed offline, using only samples of the *joint prior* density $\pi(\theta,\zeta) = \pi(\zeta|\theta)\pi(\theta)$.

3.3 Transport maps for multiscale inference

Here we will apply the transport map concepts from Chapter 2 to both describe the coarse prior distribution $\pi(\theta)$ and the downscaling density $\pi(\theta|\zeta)$.

3.3.1 Theoretical framework

For the moment, assume we have used the optimization and regression approaches from Chapter 2 to construct a transport map $F : \mathbb{R}^{(D_\theta+D_\zeta)} \to \mathbb{R}^{(D_\theta+D_\zeta)}$, from some joint reference Gaussian random variables (r_1, r_2) to (ζ,θ), where r_1 is D_ζ-dimensional and r_2 is D_θ-dimensional. Notice that F maps (r_1, r_2) to (ζ,θ), while the map T in Section 2 would map (ζ,θ) to (r_1, r_2). Enforcing the same type of lower triangular structure as before, the joint transport map F will take the form

$$\begin{bmatrix} \zeta \\ \theta \end{bmatrix} = F(r_1, r_2) = \begin{bmatrix} F_1(r_1) \\ F_2(r_1, r_2) \end{bmatrix}, \tag{3.10}$$

where F_1 is lower triangular and the r_2 portion of F_2 is also lower triangular.[2] With respect to our framework, this block triangular structure is like caffeine to a typical academic: without it, no progress can be made. On the one hand, F_1 can be used to infer r_1 instead of ζ, which solves the problem of describing $\pi(\zeta)$. On the other hand, $F_2(r_1, r_2)$ will enable sampling of $\pi(\theta|r_1)$, which serves as a one-to-one proxy for $\pi(\theta|\zeta)$.

Characterizing the coarse prior

Notice that with the coarse map F_1, we can easily replace the original coarse likelihood $\pi(d|\zeta)$ with a likelihood $\pi(d|r_1)$ based on the reference random variable r_1. To see this in the deterministic setting, consider the deterministic coarse model $\tilde{f}(\zeta)$ from Section 3.1.2. Composing the coarse model $\tilde{f}$ with coarse map F_1 allows us to define an error model in terms of the reference random variable r_1, given by

$$d = \tilde{f}\left(F_1(r_1)\right) + \epsilon. \tag{3.11}$$

This error model defines a coarse reference likelihood $\pi(d|r_1)$ and coarse reference prior $\pi(r_1) = N(0, I)$ instead of their ζ equivalents. Notice that even with a probabilistic coarse model, the coarse reference likelihood $\pi(d|r_1)$ can easily be defined with F_1 and $\pi(d|\zeta)$ using a simple change of variables.

By combining the coarse reference likelihood and prior with the multiscale form of Bayes' rule in (3.9), we obtain a form of Bayes' rule on the coarse reference variable

$$\pi(\theta, r_1|d) \propto \pi(d|r_1)\pi(r_1)\pi(\theta|r_1). \tag{3.12}$$

Crucially, this expression does not have the prior term, $\pi(\zeta)$, that plagued (3.9) – we are one step closer to a complete multiscale framework. As before, this coarse reference posterior can be broken into two parts: (1) the coarse posterior $\pi(d|r_1)\pi(r_1)$, and (2) the coarse to fine downscaling density $\pi(\theta|r_1)$. Sampling the downscaling density is the topic of the next section.

Sampling the fine scale parameter

We can easily sample the coarse posterior because only the low dimensional coarse parameters are needed and we can apply standard approaches such as MCMC without extensive craftiness. Fortunately, we can easily sample $\pi(\theta|r_1)$ as well using the second part of our map F_2. To generate a sample of $\pi(\theta|r_1')$, we start by generating a sample $r_2^* \sim \pi(r_2)$ and evaluate $F_2(r_1', r_2^*)$. This fine scale sampling is made possible by the block lower triangle structure in (5.3).

We have now used the transport map F to characterize $\pi(\zeta)$ and $\pi(\theta|\zeta)$; however, we have not discussed the construction of F. This is the topic of the following section.

[2]To be pedantic, F_1 and F_2 do not necessarily need to be lower triangular for our method to work. Only the block triangular structure, where F_1 only depends on r_1, is important. We use lower triangular F_1 and F_2 here to help ensure that the joint map F is monotone and so that we can use the efficient optimization techniques from Chapter 2.

3.3.2 Constructing the maps

For the moment, forget about the high dimensionality of θ and possibly of ζ – those issues will be addressed in the examples below. Ignoring the dimensionality, we can construct an approximation to F in the same way as Chapter 2: (1) build a map $\tilde{T}$ from (ζ, θ) to (r_1, r_2) by solving the optimization problem in (2.21), and (2) use regression to construct an approximation $\tilde{F}$ of F. Solving the optimization problem in (2.21) requires samples of the joint prior $\pi(\zeta, \theta) = \pi(\zeta|\theta)\pi(\theta)$, which can easily be generated by sampling the fine scale prior $\pi(\theta)$ and then sampling the upscaling density $\pi(\zeta|\theta)$. Moreover, when the upscaling is deterministic, generating a sample of $\pi(\zeta|\theta^{(k)})$ only requires running the fine to coarse model at $\theta^{(k)}$.

Once the joint prior samples are generated, we use the optimization and regression approaches introduced in Section 2 to define $\tilde{F}$. We then split $\tilde{F}$ into the two components, $\tilde{F}_1$ and $\tilde{F}_2$, we need in the multiscale framework to define the coarse prior and enable fine scale sampling. Importantly, it is always possible to split a lower triangular map $\tilde{F}$ into $\tilde{F}_1$ and $\tilde{F}_2$. Algorithm 3.1 shows a high level outline of the entire inference framework.

3.3.3 Choosing the number of fine scale samples

After sampling the coarse posterior, we have samples $\{r_1^{(1)}, r_1^{(2)}, ..., r_1^{(N)}\}$ of $\pi(r_1|d)$. Our next step is then to fill in the fine scale parameters θ by sampling $\pi(\theta|r_1^{(i)})$ for each coarse sample. But how many fine scale samples should we generate for each coarse sample? We could simply generate one fine sample, or we could generate 100 fine samples. With more samples, we lower the variance of our Monte Carlo estimates, but at the same time increase the computational cost of the sampling. The remainder of this section analyses this tradeoff in an attempt to find the "optimal" number of fine samples for each coarse sample.

Assume we have N correlated samples of the the coarse posterior $\pi(r_1|d)$ that are collected in the set $\mathbf{r_1}$ as

$$\mathbf{r_1} = \left\{ r_1^{(1)}, r_1^{(2)}, ..., r_1^{(N)} \right\}. \tag{3.13}$$

Now, for each coarse sample, assume we generate M samples of $\pi(\theta|r_1)$. Thus, we will produce NM fine scale samples. We collect these samples in the set $\mathbf{r_2}$ given by

$$\mathbf{r_2} = \left\{ r_2^{(1,1)}, r_2^{(1,2)}, ..., r_2^{(1,M)}, r_2^{(2,1)}, ..., r_2^{(2,M)}, r_2^{(3,1)}, ..., r_1^{(N,M)} \right\}. \tag{3.14}$$

Now, let $\hat{\theta}$ be a Monte Carlo estimator of the posterior mean. This estimator is given by

$$\hat{\theta}(\mathbf{r_1}, \mathbf{r_2}) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} T_2 \left(r_1^{(i)}, r_2^{(i,j)} \right). \tag{3.15}$$

Notice that $\hat{\theta}(\mathbf{r_1}, \mathbf{r_2})$ is an unbiased estimator of the posterior mean – the posterior

Algorithm 3.1: Overview of the entire multiscale inference framework.

Input: A prior density $\pi(\theta)$, an upscaling distribution $\pi(\zeta|\theta)$, and a way to sample the coarse posterior $\pi(\zeta|d)$.

Output: Samples of the fine scale posterior $\pi(\theta|d)$

`// Generate prior samples`

for $k \leftarrow 1$ ***to*** K **do**

  Sample $\theta^{(k)}$ from $\pi(\theta)$ Sample $\zeta^{(k)}$ from $\pi\left(\zeta|\theta^{(k)}\right)$

`// Construct` $\tilde{T}$`, the map from` (ζ, θ) `to` (r_1, r_2)`.`

for $i \leftarrow 1$ ***to*** $D_\zeta + D_\theta$ **do**

  Solve (2.21) to get $\tilde{T}_i$

`// Build` $\tilde{F}$`, the map from` (r_1, r_2) `to` (ζ, θ)`, using regression.`

for $k \leftarrow 1$ ***to*** K **do**

  $\left(r_1^{(k)}, r_2^{(k)}\right) = \tilde{T}\left(\zeta^{(k)}, \theta^{(k)}\right)$

Solve (2.34) to get $\tilde{F}_1$ and $\tilde{F}_2$

`// Coarse scale posterior sampling.`

Generate samples $\{r_1^{(1)}, r_1^{(2)}, ..., r_1^{(N)}\}$ of $\pi(r_1|d)$ using MCMC.

`// Fine scale posterior sampling.`

for $i \leftarrow 1$ ***to*** N **do**

  for $j \leftarrow 1$ ***to*** M **do**

    Sample $r_2^{(i,j)}$ from iid Gaussian

    $\theta^{(iM+j)} \leftarrow \tilde{F}_2\left(r_1^{(i)}, r_2^{(i,j)}\right)$

return *Posterior samples* $\left\{\theta^{(1)}, \theta^{(2)}, ..., \theta^{(NM)}\right\}$

induced by the maps F_1 and F_2. Thus, we are not considering the error in the maps here, simply the best way to explore the map-induced approximation to $\pi(\theta|d)$. Mathematically, our goal is to find the value of M that balances minimizing the variance of the Monte Carlo estimator $\hat{\theta}$ and the computational expense of sampling. The basic idea is that beyond some critical M, there will be a diminishing reward for generating more fine scale samples because most of the estimator variance stems from the lack of coarse samples. To see this, we can use the law of total variance.

We start with the identity

$$\underset{\mathbf{r_1},\mathbf{r_2}}{\mathbb{V}\text{ar}}\left[\hat{\theta}(\mathbf{r_1},\mathbf{r_2})\right] = \underset{\mathbf{r_1}}{\mathbb{V}\text{ar}}\left[\underset{\mathbf{r_2}}{\mathbb{E}}\left\{\hat{\theta}(\mathbf{r_1},\mathbf{r_2})|\mathbf{r_1}\right\}\right] + \underset{\mathbf{r_1}}{\mathbb{E}}\left[\underset{\mathbf{r_2}}{\mathbb{V}\text{ar}}\left\{\hat{\theta}(\mathbf{r_1},\mathbf{r_2})|\mathbf{r_1}\right\}\right]. \tag{3.16}$$

Now, define $f_1(r_1) = \underset{\mathbf{r_2}}{\mathbb{E}}\left\{T_2\left(r_1^{(i)}, r_2^{(i,j)}\right)\right\}$ and $f_2(r_1) = \underset{\mathbf{r_2}}{\mathbb{V}\text{ar}}\left\{T_2\left(r_1^{(i)}, r_2^{(i,j)}\right)\right\}$. Notice that these quantities involve expectations over the collection of samples, not over the random variables r_1 and r_2. Also, $f_1(r_1)$ and $f_2(r_1)$ are functions of the random variable r_1 that do not depend on the samples $\mathbf{r_2}$. Using these quantities and some algebra, we can rewrite (3.16) as

$$\begin{aligned}\underset{\mathbf{r_1},\mathbf{r_2}}{\mathbb{V}\text{ar}}\left[\hat{\theta}(\mathbf{r_1},\mathbf{r_2})\right] &= \underset{\mathbf{r_1}}{\mathbb{V}\text{ar}}\left[\frac{1}{N}\sum_{i=1}^{N} f_1(r_1)\right] + \underset{\mathbf{r_1}}{\mathbb{E}}\left[\frac{1}{N^2 M}\sum_{i=1}^{N} f_2(r_1)\right] \\ &= \frac{1}{N^2}\sum_{i=1}^{N}\underset{\mathbf{r_1}}{\mathbb{V}\text{ar}}\left[f_1(r_1)\right] + \frac{1}{N^2 M}\sum_{i=1}^{N}\underset{\mathbf{r_1}}{\mathbb{E}}\left[f_2(r_1)\right]. \end{aligned} \tag{3.17}$$

Observe that $\underset{\mathbf{r_1}}{\mathbb{V}\text{ar}}\left[f_1(r_1)\right]$ and $\underset{\mathbf{r_1}}{\mathbb{E}}\left[f_2(r_1)\right]$ are constants, which allow us to express the estimator variance with the simple expression

$$\underset{\mathbf{r_1},\mathbf{r_2}}{\mathbb{V}\text{ar}}\left[\hat{\theta}(\mathbf{r_1},\mathbf{r_2})\right] = \frac{C_1}{N} + \frac{C_2}{NM}, \tag{3.18}$$

where $C_1 = \underset{\mathbf{r_1}}{\mathbb{V}\text{ar}}\left[f_1(r_1)\right]$ and $C_2 = \underset{\mathbf{r_1}}{\mathbb{E}}\left[f_2(r_1)\right]$ are constants depending on the form of $\pi(\theta|r_1)$ and the samples of $\pi(r_1|d)$. Importantly, C_1 captures the inter-sample correlation of the MCMC chain used to generate the coarse posterior samples $\mathbf{r_1}$.

The expression in (3.18) is quite intuitive – some of the estimator variance is dependent only on the number of coarse samples and some of the variance is determined by the number of fine samples generated for each coarse sample. Interestingly, no matter how large M becomes, the estimator variance will never go to zero because the first term in the variance only depends on the number of coarse samples N. Still, coarse and fine samples have different computational costs, and there is clearly a tradeoff between obtaining more coarse samples and obtaining more fine samples. To help choose an "optimal" tradeoff, we need to incorporate the computational time it takes to sample $\pi(r_1|d)$ and the time it takes to generate a fine scale sample from $\pi(\theta|r_1)$.

Let t_1 be the average run time for one coarse MCMC step on $\pi(r_1|d)$ and let t_2

be the average time required to generate one sample of $\pi(\theta|r_1)$ using $F_2(r_1, r_2)$. The total sampling time is then given by

$$t_{tot} = t_1 N + t_2 N M. \tag{3.19}$$

Using a fixed total time, our goal now is to minimize the estimator variance from (3.18) by choosing N and M. To do this, we first solve for the optimal value of N by solving (3.19) for M, substituting the result into the variance from (3.18), and finding the N that minimizes the result. After some simplification, we find the optimal number of coarse samples to be

$$N = \frac{t_{tot}\left(C_1 t_1 - \sqrt{C_1 C_2 t_1 t_2}\right)}{C_1 t_1^2 - C_2 t_1 t_2}. \tag{3.20}$$

Using this expression in the total time (3.19) constraint, we obtain

$$M = \frac{t_{tot} - t_1 N}{t_2 N}, \tag{3.21}$$

which is the optimal number of fine samples for each coarse sample. While useful, these expressions require knowledge of the total run time, a quantity that will change depending on how many MCMC steps are taken. However, by combining (3.20) and (3.21), we can obtain an expression for M that is independent of t_{tot} and N. This expression is given by

$$M = \frac{t_1}{t_2}\left[\frac{C_1 t_1 - C_2 t_2}{\left(C_1 t_1 - \sqrt{C_1 C_2 t_1 t_2}\right)} - 1\right]. \tag{3.22}$$

While C_1 and C_2 are often not known exactly, this expression provides a guideline for choosing M. Moreover, as we show in Section3.6.1, multiple runs of the multiscale algorithm can be used to estimate the values of C_1 and C_2 using regression.

Figure 3-1 shows the qualitative behavior of the optimal M in (3.22) for a fixed C_1 and C_2 but varying t_1 and t_2. As we would expect, when fine samples are less expensive than coarse samples, i.e., $t_2 < t_1$, it is usually advantageous to produce more than one fine sample per coarse sample. Moreover, the advantage diminishes as coarse and fine sampling times get close, $t_2 \to t_1$. It is not always advantageous to sample more than one fine sample when $t_2 < t_1$ because N is in the denominator of both terms in (3.18), implying that we get more "bang for our buck" when generating coarse samples.

3.4 A proof-of-concept example

Here we give an initial example of our multiscale framework on a simple example with two fine scale parameters. The upscaling is deterministic and represents the harmonic mean of a lognormal random variable – a common upscaling operation in porous media.

Consider a two dimensional fine scale parameter, θ, and a scalar coarse parameter.

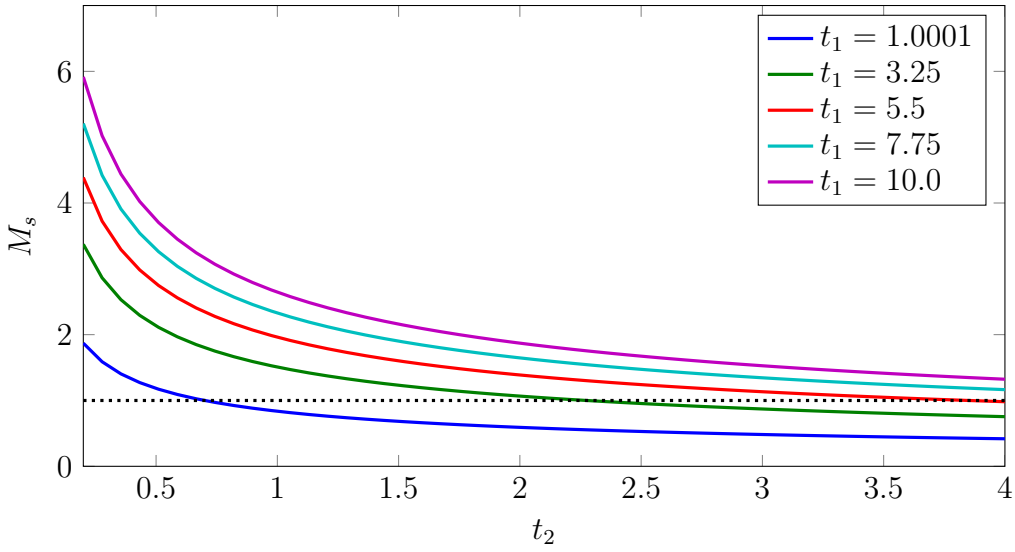


Figure 3-1: Illustration of optimal M for various t_1 and t_2. Note that $C_1 = 1$ and $C_2 = 0.7$ are fixed in this illustration.

The fine to coarse model in this example is defined by

$$\zeta = \frac{1}{\exp(-\theta_1) + \exp(-\theta_2)}. \tag{3.23}$$

Also, the fine scale prior is a zero mean correlated Gaussian with density given by

$$\pi(\theta) = N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 & 0.6 \\ 0.6 & 2 \end{bmatrix} \right). \tag{3.24}$$

The final component of the inference problem is the coarse error model. For this problem, the scalar data is related to the coarse parameter through the nonlinear model

$$d = \zeta^3 - 2 + \epsilon, \tag{3.25}$$

where $\epsilon \sim N(0, \sigma_\epsilon^2)$.

In the solution of this problem, a map constructed from multivariate Hermite polynomials was used. A Hermite expansion was used here because this type of expansion has optimal convergence properties when the map input is an uncorrelated Gaussian random variable [108]. While the map $\tilde{T}(d, \theta)$, does not have a Gaussian input, the inverse map $\tilde{F}(r_1, r_2)$, does have Gaussian input. In addition to choosing the type of polynomial, we also need to decide on which polynomial terms should be included in the expansion. This boils down to choosing an appropriate set of multi indices in (2.23). In this two dimensional example, we will limit the total order of the multi index to be less than or equal to P. This choice allows the maps to reflect strong nonlinearities in any variable or combination of variables; however, with more flexibility comes more degrees of freedom, making both the optimization problem in (2.21) and the regression problem in (2.34) more difficult. In this small dimensional

problem, the increased difficulty is manageable, but in larger applications, like those discussed below, a more strategic choice of map truncation will be required.

Figure 3-2 compares the true posterior density with posteriors obtained using third, fifth, and seventh order maps. We first use MCMC to sample the coarse posterior $\pi(r_1|d = -1.8)$ and then use the coarse to fine map $F_2(r_1, r_2)$ to generate approximate conditional samples of $\pi(\theta|d)$. As the map order is increased, the map-based approximate posterior density converges to the true posterior. Note that a small amount of the posterior differences may be relics of the kernel density estimation process used for plotting. However, in this example, the coarse parameter ζ is a sufficient statistic, which means that (3.1) is satisfied exactly and any significant error in Figure 3-2 is caused by inadequate parameterizations of the maps.

For this problem $50,000$ samples were used to construct the maps and a value of $d = -1.8$ was used to define the posterior.

3.5 Application: simple groundwater flow

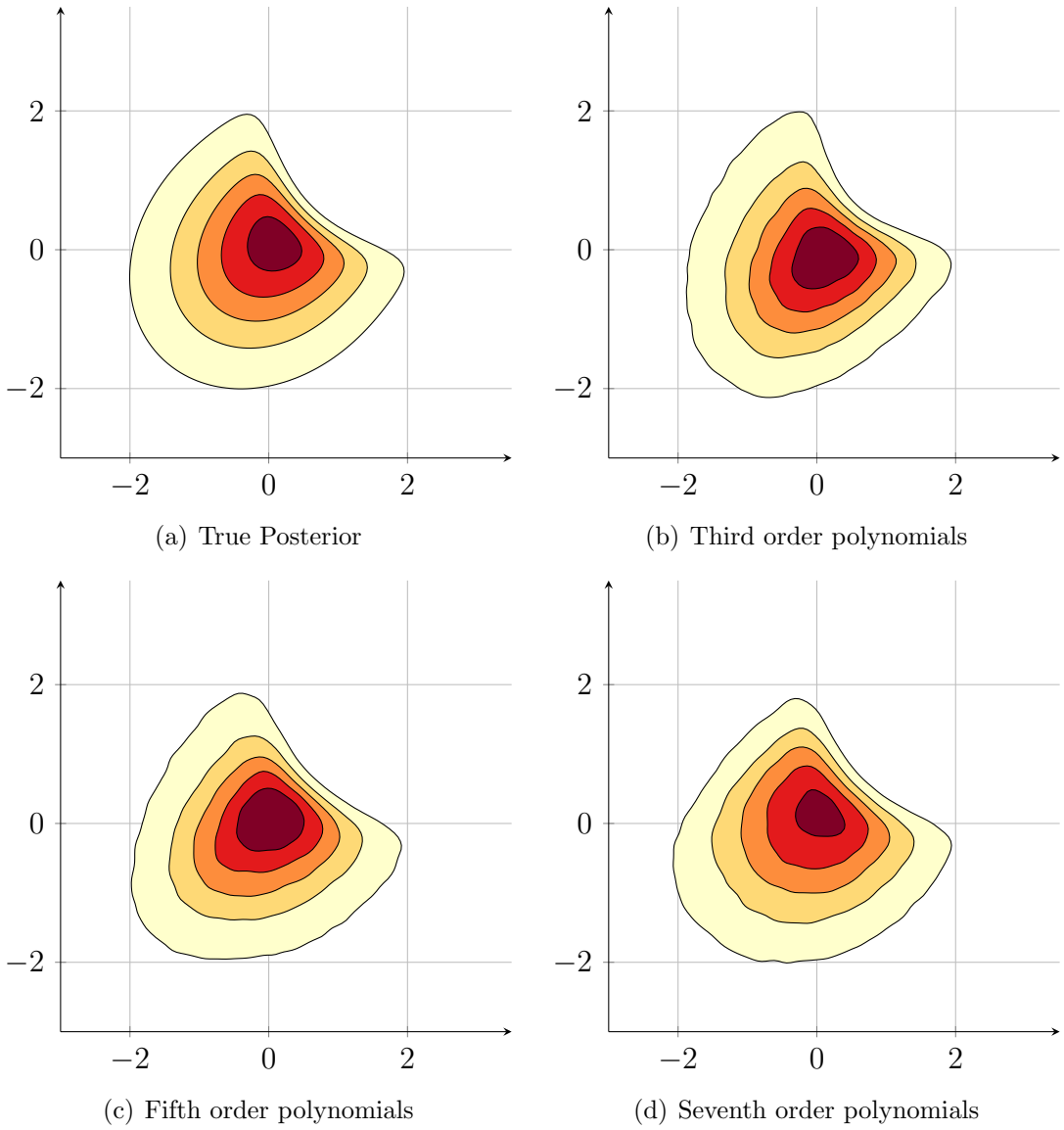
To illustrate the accuracy and performance of our multiscale approach we will consider an example inverse problem from subsurface hydrology. The goal is to describe subsurface structure by characterizing a spatially distributed conductivity field using limited observations of hydraulic head. An elliptic equation, commonly called the pressure equation, will serve as a simple steady state model of groundwater flow in a confined aquifer. The model is given by

$$-\nabla \cdot (\kappa(x)\nabla h(x)) = f(x), \tag{3.26}$$

where x is a spatial location in one or two spatial dimensions (depending on the test case below), $\kappa(x)$ is the permeability field[3] field we want to characterize, $f(x)$ contains well or recharge terms, and $h(x)$ is the hydraulic head solution that we can measure at several locations throughout the domain. See [13] for a thorough derivation of this model and a comprehensive discussion of flow in porous media. From a hydrology standpoint, this steady state model is quite simple; however, in an inference context, elliptic problems provide many interesting challenges: the parameter space is a high dimensional random field, elliptic operators are smoothing operators that yield ill-posed inverse problems, and solving the system in (3.26) on very fine meshes can become computationally expensive.

The elliptic model in (3.26) acts as a nonlinear lowpass filter, removing high frequency features of $\kappa(x)$ from $h(x)$. This means that some features of $\kappa(x)$ can not be estimated even if $h(x)$ was known exactly. Variational methods such as Multiscale Finite Element Methods (MsFEM) [51, 1], Multiscale Finite Volume Methods [57], Variational Multiscale Methods [52, 60], Heterogeneous Multiscale Methods [30], and Subgrid Upscaling [8] take advantage of the smoothing to create a small, easy to

[3]While we will talk of $\kappa(x)$ as the permeability, we are working in two dimensions and our use of $\kappa(x)$ is more in line with the usual definition of transmissivity. At any rate, $\kappa(x)$ is a strictly positive spatially varying field.



(a) True Posterior

(b) Third order polynomials

(c) Fifth order polynomials

(d) Seventh order polynomials

Figure 3-2: Convergence of the multiscale posterior to the true posterior as the total polynomial order is increased. In all cases, $50,000$ samples of $\pi(\zeta, \theta)$ were used to build the maps.

solve, coarse problem. The common idea behind all of these strategies is to implicitly coarsen the elliptic operator to allow for more efficient solution while simultaneously maintaing the accuracy of the solution. In this application, MsFEM will be used to solve (3.26) and simultaneously define the coarse parameter ζ used in our multiscale framework.

3.5.1 Defining the coarse parameter with the multiscale finite element method (MsFEM)

The formulation here closely follows the introduction given by [51] and we only provide enough details to understand our use of MsFEM in the multiscale inference setting. Readers may consult [51] or other MsFEM-specific texts for further details of MsFEM implementation and theory. Alternatively, more general information on finite element methods can be found in [97].

Let Ω be the spatial domain of interest, where the pressure equation is to be solved and consider a coarse triangulation $\mathcal{T}_h$ of Ω into finite elements. While classic finite elements may use simple basis functions defined by polynomial nodal basis functions, the multiscale finite element method (MsFEM) uses an additional fine mesh to construct special basis functions for the coarse solve. These special basis functions come from local fine-scale solves of the pressure equation. Figure 3-3 shows one quarter of a typical nodal MsFEM basis function. The bulges in the basis function come from variations in the fine-scale permeability field $\kappa(x)$.

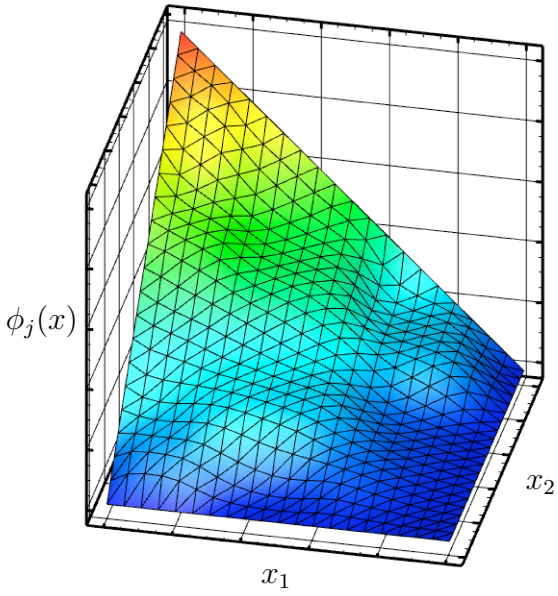


Figure 3-3: Example of one quarter of a nodal MsFEM basis function. The function has a maximum height of 1 and comes from the solution to a homogeneous elliptic problem over the coarse element. In this figure, a triangular fine-scale mesh was used inside a quadrilateral coarse mesh.

Let C be one coarse quadrilateral element in the coarse triangulation $\mathcal{T}_h$. Such an element is one shown in Figure 3-3. Furthermore, assume that node i of the triangulation is at the bottom left corner of element C. Over the element C, the MsFEM basis function $\phi_i(x)$ satisfies a homogeneous version of the pressure equation given by

$$-\nabla \cdot (\kappa(x)\nabla\phi_i) = 0 \qquad x \in C \tag{3.27}$$

$$\phi_i = \phi_i^0 \qquad x \in \partial C, \tag{3.28}$$

where ϕ_i^0 are boundary conditions chosen so that $\phi_i = 1$ at the bottom left corner of C, zero on the right and top edges of C, and decreasing from 1 to 0 along the bottom and left boundaries. While imposing linear boundary conditions with sometime like $\phi_i^0 = 1 - x_1$ are possible. As [51] and [57] are clear to point out, this choice does not accurately reflect the impact of the spatially varying $\kappa(x)$ on can lead to significant errors in the MsFEM solution. With this in mind, we follow [57] and solve a 1D analog to (3.27) along the boundaries to compute ϕ_i^0. Readers can consult [57] for more details on this approach, or [51] for an alternative approach to defining ϕ_i^0 that is based on enlarging the coarse element C.

Regardless of how ϕ_i^0 is computed, similar choices of boundary conditions are used to define MsFEM basis function centered at other corners of C. Usually solutions of (3.27) are found with standard Galerkin finite element methods.

Once local solves of (3.27) have been performed on each coarse element to construct all MsFEM basis functions, the MsFEM basis function are coupled into a global coarse solve to approximate the solution to the pressure equation in (3.26). Let p_h be the approximate hydraulic head solution. MsFEM methods try to find coefficients $\tilde{p}_i$ on each of the MsFEM basis function ϕ_i such that

$$p_h = \sum_i \tilde{p}_i \phi_i. \tag{3.29}$$

The weights are the solution to the linear system

$$A_c \tilde{p} = b \tag{3.30}$$

where A_c is the coarse stiffness matrix and b is a vector of discretized source terms. These quantities come from a Galerkin projection of the solution onto the MsFEM basis, see [51] for a more rigorous treatment of this topic. The form for each entry a_{ij} in A_c is ultimately given by

$$a_{ij} = \sum_{C \in \mathcal{T}_h} e_{ij,C} \tag{3.31}$$

where each $e_{ij,C}$ will be referred to as an *elemental integral* and is defined by

$$e_{ij,C} = \int_C \kappa(x)\nabla\phi_j(x) \cdot \nabla\phi_i(x) dx \tag{3.32}$$

These elemental integrals completely describe the coarse stiffness matrix, which in turn completely describes the solution coefficients $\tilde{p}_i$. *The elemental integrals can therefore be used to define the coarse parameter ζ in the multiscale inference framework*. The next section will show this in detail. Importantly, because each MsFEM basis function ϕ_j was computed using fine-scale solves of the local elliptic equation in (3.27), nearly all of the fine-scale physics are embedded in each $e_{ij,C}$ and the solution of the coarse system will be a good approximation to a global fine-scale solution.

Note that the coarse system A_c is the same size of a standard Galerkin finite element discretization on the coarse scale, but has comparable accuracy to a global fine-scale solution because the MsFEM basis functions were used. Solving the small coarse system for $\tilde{p}$ is inexpensive in itself, but many local solves of (3.27) are still required to find the multiscale basis functions ϕ_i and subsequently build the coarse stiffness matrix A_c. Fortunately, the local solves are easily parallelized, and in the inference setting can be performed offline – before any data is observed.

3.5.2 Multiscale framework applied to MsFEM

Recall that our interest in MsFEM is to accelerate inference for the fine-scale conductivity field, $\kappa(x)$. MsFEM methods assume the elemental integrals are sufficient to describe the pressure, $h(x)$, which is equivalent to the conditional independence assumption in (3.1). The elemental integrals can therefore be used to define the coarse parameters, ζ, in the multiscale inference framework. The exact relationship between the elemental integrals, e_{ijk}, and the coarse parameters will depend on the spatial dimension (1D or 2D in our case) and will be discussed below. However, in all cases, the fine-scale parameter will be the log-permeability

$$\theta = \log \kappa. \tag{3.33}$$

The large dimension of θ makes this problem interesting, but at the same time makes construction of the transport map, F, much more difficult. For example, a cubic total order map in 110 dimensions will have 234136 polynomial coefficients! Clearly, such a general form for the map is infeasible and a more strategic choice of basis is required. The set of polynomial terms, defined through a set of multi indices, will be tuned to a specific definition of ζ. Thus, different multi-index sets will be used for the 1D and 2D problems. The particular forms for the 1D and 2D multi indices are described independently below.

While the multi-layered maps in Chapter 2 are one approach for constructing maps in general high-dimensional problems, we focus here on techniques that exploit specific spatial structure from the PDE model.

Strategies for building map in one spatial dimension

In one spatial dimension, MsFEM produces one elemental integral per coarse element. Thus, when 10 coarse elements are used, the coarse dimension is one tenth of the fine dimension, i.e., $D_\zeta = D_\theta/10$. In our example, $D_\theta = 100$, so the dimension of the

coarse parameter is $D_\zeta = 10$. Fortunately, a third order total-order limited map is feasible in 10 dimensions and no tricks need to be used to define $\tilde{F}_1$. However, the coarse to fine map, $\tilde{F}_2$, is from $\mathbb{R}^{110}$ to $\mathbb{R}^{100}$, which cannot be directly attacked with a general total order map and a more problem-specific form of $\tilde{F}_2$ is necessary.

In this application, the prior on θ is Gaussian. This means that θ, r_1, and r_2 are all *marginally* Gaussian random variables. While this does not mean that θ, r_1 and r_2 are *jointly* Gaussian, it does indicate that a linear map may characterize much of the joint structure between these random variables and localized nonlinear terms in the map may be able to adequately characterize the non-Gaussian features of the joint distribution. To mathematically define such a map, consider the vector of reference random variables

$$r = [r_1, r_2]^T = [r_{11}, r_{12}, ..., r_{1K}, r_{21}, r_{22}, ..., r_{2N}]^T . \tag{3.34}$$

The definition of $\tilde{F}_2$ will start with linear components and then enrich the linear multi-index with choice nonlinear terms. Consider the expansion used to define $\tilde{F}_{2,d}$ – the output of $\tilde{F}_2$ corresponding to θ_d. The set of linear multi indices used in $\tilde{F}_{2,d}$ is given by

$$\mathcal{J}_d^L = \{\mathbf{j} : \|\mathbf{j}\|_1 \leq 1,\, j_i = 0\, \forall i > d + D_\zeta\}, \tag{3.35}$$

where $\mathbf{j} \in \mathbb{N}^{D_\zeta + D_\theta}$ is a multi-index over both r_1 and r_2. The condition that $j_i = 0\, \forall i > d + D_\zeta$ is needed to ensure the map is lower triangular. Furthermore, the $d + D_\zeta$ term takes into account of the D_ζ dimensional r_1 input.

To introduce some nonlinear structure on top of this linear set without using total order limited polynomials everywhere, we will take advantage of the spatial locality in the MsFEM upscaling. Each component of θ represents the log-permeability over one cell in the finite element discretization. Thus, each component of θ lies within one coarse element, and directly impacts only one component of ζ. Specifically, component d of the fine-scale field, θ_d, is related to $\zeta_{\rho(d)}$, where $\rho(d)$ is the coarse element containing the d^{th} fine element, i.e.,

$$\rho(d) = \left\lfloor \frac{d}{D_\zeta} \right\rfloor + 1. \tag{3.36}$$

In the expansion for $\tilde{F}_{2,d}$, we will strategically introduce nonlinear terms in $r_{1,\rho(d)}$ and $r_{2,d}$. These nonlinear terms are introduced to help capture spatially localized nonlinear dependencies. Combining these terms with the linear multi-index set, yields

$$\mathcal{J}_d^N = \mathcal{J}_d^L \cup \left\{ \mathbf{j} : \left\{ \begin{array}{ll} j_d \leq P & d \in \{\rho(d), d + D_\zeta\} \\ j_d = 0 & d \notin \{\rho(d), d + D_\zeta\} \end{array} \right\} \right. . \tag{3.37}$$

This multi-index set has taken advantage of locality in the MsFEM upscaling to dramatically reduce the degrees of freedom in the transport map. The set $\mathcal{J}_d^N$ allows $\tilde{F}_2$ to capture more nonlinear correlation structure; however, in some instances it may be feasible to simply use the linear map $\mathcal{J}_d^L$. In this linear setting, a more efficient method for constructing $\tilde{F}_2$ exists.

Special case: linear F_2

We now consider the special case when all the local terms in (3.37) are linear, i.e., $P = 1$. In this case, the map from (r_1, r_2) to θ, $\tilde{F}_2$, is completely linear. While we could still use the optimization and regression approach from Chapter 2, when the prior $\pi(\theta)$ is Gaussian, we can construct $\tilde{F}_2$ much more efficiently using an approach based on cross covariances.

To see this, first assume that we construct $\tilde{F}_1$ using the optimization and regression approach. During the regression part of that procedure, we push samples $\{\zeta^{(1)}, \zeta^{(2)}, ..., \zeta^{(K)}\}$ through the inverse of $\tilde{F}_1^{-1}(\zeta) = T_1(\zeta)$, to obtain corresponding samples of r_1 defined by

$$r_1^{(k)} = T_1\left(\zeta^{(k)}\right). \tag{3.38}$$

Furthermore, because $\zeta^{(k)} \sim \pi(\zeta|\theta^{(k)})$, each reference sample $r_1^{(k)}$, is matched with a prior sample, $\theta^{(k)}$, from $\pi(\theta)$. We know r_1 and θ are marginally Gaussian, so if we make the further assumption that they are jointly Gaussian (or can be well-approximated by a jointly distributed Gaussian random variable), then we can use the sample cross covariance of r_1 and θ, denoted by $\Sigma_{r\theta}$, to construct $\tilde{F}_2$. Under this jointly Gaussian assumption, we can write the the conditional distribution of θ given r_1 as

$$\pi(\theta|r_1) = N\left(\bar{\theta} + \Sigma_{r\theta}^T r_1, \Sigma_{\theta\theta} - \Sigma_{r\theta}^T \Sigma_{r\theta}\right), \tag{3.39}$$

where $\bar{\theta}$ is the prior mean of θ and $\Sigma_{\theta\theta}$ is the prior covariance. We should point out that this use of the sample covariance to define a Gaussian conditional distribution is very similar to use of sample covariances in the ensemble Kalman filter (EnKF) [34]. Moreover, this expression for $\pi(\theta|r_1)$ implies that $\tilde{F}_2$ can be defined as

$$\theta = F_2(r_1, r_2) = \bar{\theta} + \Sigma_{r\theta}^T r_1 + \left(\Sigma_{\theta\theta} - \Sigma_{r\theta}^T \Sigma_{r\theta}\right)^{1/2} r_2, \tag{3.40}$$

where $(\cdot)^{1/2}$ is a matrix square root (the Cholesky square root was used in our implementation). From numerical experiments, we have observed that this method of constructing a linear $\tilde{F}_2$ is much more efficient than performing the optimization and regression from Chapter 2. Just as importantly, in our applications, this linear map seems to give the same posterior accuracy as a linear map produced via optimization. The accuracy and efficiency tables in the next section will illustrate the performance of this cross covariance approach.

Strategies for building the maps in two spatial dimensions

In two spatial dimensions, there are 10 elemental integrals on each coarse element. While this may be significantly smaller than the number of fine-scale elements in each coarse element, the number of coarse quantities D_ζ still becomes too large for us to tackle it with a total order limited map. Just as we used problem structure to define $\tilde{F}_2$ in the one dimensional setting, we will again use structure in the two dimensional problem to define a more tractable form for the coarse map $\tilde{F}_1$. In particular, we will restrict our focus to problems with stationary prior distributions and combine this

stationarity with the inherent locality of MsFEM to derive an expressive nonlinear coarse map $\tilde{F}_1$. For convenience of notation below, let $V = D_\zeta/10$ be the number of coarse elements in our 2D discretization.

As in the one-dimensional derivation above, we will again combine r_1 and r_2 into a single vector, but now the expression

$$r = [r_1, r_2]^T = [r_{11}, r_{12}, ..., r_{1V}, r_{21}, r_{22}, ..., r_{2D_\theta}]^T \tag{3.41}$$

will be defined blockwise. That is, r_{1d} contains the 10 components of r_1 corresponding to coarse element d. Similarly, we use a block definition of ζ given by $\zeta = [\zeta_1, \zeta_2, ..., \zeta_V]$. Now *assume we have a spatially stationary prior on* θ. In this case, we will also obtain a stationary prior on ζ, which implies that each marginal distribution of ζ_d is the same, i.e.,

$$\zeta_i \overset{i.d.}{=} \zeta_j \quad \forall i, j \in \{1, 2, ..., V\}. \tag{3.42}$$

We will exploit this structure to build the coarse map $\tilde{F}_1$. First, consider a marginal map, $\tilde{F}_{1,m} : \mathbb{R}^{10} \to \mathbb{R}^{10}$, that pushes a 10 dimensional standard normal random variable to any one of the marginal densities, $\pi(\zeta_d)$. Notice that the m subscript refers to the fact that $\tilde{F}_{1,m}$ is to the **m**arginal distribution of ζ_d. Given a 10 dimensional standard normal random variable r_m, we have $\zeta_d \overset{i.d.}{=} \tilde{F}_{1,m}(r_m)$ for any coarse element $d \in \{1, 2, ..., V\}$. Notice that this map is 10 dimensional regardless of how many coarse elements are used.

The marginal map $\tilde{F}_{1,m}$ captures nonlinear correlations *within* each coarse element. Now, assume we have constructed $\tilde{F}_{1,m}$ as well as its inverse $\tilde{T}_{1,m}(\zeta_i) = \tilde{F}^{-1}_{1,m}(\zeta_i)$ using the optimization and regression approach from Chapter 2. Using V repetitions of $\tilde{T}_{1,m}$, one for each coarse element, we can define an intermediate D_ζ dimensional random variable $r_{1,m}$ as follows

$$r_{1,m} = \left[\tilde{T}_{1,m}(\zeta_1), \tilde{T}_{1,m}(\zeta_2), ..., \tilde{T}_{1,m}(\zeta_V)\right]^T \tag{3.43}$$

Notice that each block of $r_{1,m}$ is marginally IID Gaussian, but the entire variable, $r_{1,m}$ may have nonlinear correlations and will not necessarily be jointly Gaussian. Since our goal is to build a map $\tilde{F}_1$ from the IID Gaussian random variable r_1 to ζ, we need to remove the inter-block correlations present in $r_{1,m}$. For computational efficiency, we will only consider linear correlations and we will use a lower triangular Cholesky decomposition L, of the $r_{1,m}$ covariance

$$\mathbb{C}\text{ov}\left[r_{1,m}\right] = LL^T. \tag{3.44}$$

By dividing the lower triangular Cholesky factor, L, into blocks corresponding to the

coarse elements, we obtain

$$L = \begin{bmatrix} L_{11} & 0 & 0 & \cdots & 0 \\ L_{21} & L_{22} & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ L_{(V-1)1} & L_{(V-1)2} & \cdots & L_{(V-1)(V-1)} & 0 \\ L_{V1} & L_{V2} & \cdots & L_{V(V-1)} & L_{VV} \end{bmatrix}, \tag{3.45}$$

where each diagonal entry is a 10×10 lower triangular matrix. Notice that applying L^{-1} to $r_{1,m}$ will remove linear correlations from $r_{1,m}$, leading to

$$\begin{aligned} r_1 &\approx L^{-1} r_{1,m} \\ \Rightarrow L r_1 &\approx r_{1,m} \end{aligned} \tag{3.46}$$

Combining L with the local nonlinear map, $\tilde{F}_{1,m}$, we obtain a complete coarse map

$$\zeta \overset{i.d.}{=} \begin{bmatrix} T_1 \left(L_{11} r_{11} \right) \\ \tilde{F}_{1,m} \left(L_{21} r_{11} + L_{22} r_{12} \right) \\ \vdots \\ \tilde{F}_{1,m} \left(L_{D_\zeta 1} r_{11} + L_{D_\zeta 2} r_{12} + ... + L_{D_\zeta D_\zeta} r_{1 D_\zeta} \right) \end{bmatrix}. \tag{3.47}$$

Crucially, constructing this map only requires building a single 10 dimensional nonlinear map. As mentioned in our 1D map strategy, total order limited polynomial expansions can be applied to maps of this size. In the 2D example below, we construct $\tilde{F}_1$ with (3.47), but construct $\tilde{F}_2$ using the same cross covariance approach described above for the 1D problem. The samples of r_1 used in the covariance are computed using the nonlinear inverse map $\tilde{T}_{1,m}$ and the inverse Cholesky factor L^{-1}.

3.6 Numerical results

3.6.1 One spatial dimension

Here we apply our multiscale framework and the previous section's problem-specific map structure to our first "large-scale" inference problem. The goal of this section is to analyze the efficiency and accuracy of our approach by comparing the multiscale method with a standard MCMC approach. The inverse problem is to infer a spatially distributed one dimensional log conductivity field using MsFEM as the forward model. While our multiscale approach can handle much larger problems (as demonstrated in the 2D section below), the problem size is restricted in this section to enable comparison with fine-scale MCMC.

This example aims to sample the posterior distribution $\pi(\theta|d)$ where $\theta = \log \kappa(x)$ is the log conductivity field depending on the spatial position $x \in [0, 1]$, and the data d is a set of pressure observations. The prior on θ uses an exponential covariance

kernel with the form

$$\mathbb{C}\text{ov}\left(\theta(x_1), \theta(x_2)\right) = \sigma_\theta^2 \exp\left[-\frac{|x_1 - x_2|}{L}\right]. \tag{3.48}$$

We set the correlation length be $L = 0.1$ and the prior variance be $\sigma_\theta^2 = 1.0$. An exponential kernel was chosen for two reasons, (i) this class of covariance kernel yields rough fields that are often found in practice but difficult to handle with typical dimension reduction techniques such as Karhunen-Loève decompositions, and (ii) MsFEM is most accurate for problems with strong scale separation, which is the case for problems with these rough fields. We use 10 coarse elements and 10 fine elements per coarse element. This means that θ is a 100 dimensional random variable. Moreover, the data is 11 dimensional, coming from observation at all 11 nodes in the coarse mesh (10 coarse cells implies 11 coarse nodes). Dirichlet boundary conditions are used with $p(0) = 1$ and $p(1) = 0$. To generate the data, a realization of the prior log-conductivity (shown in Figure 3-5) was used with a full fine-scale FEM forward solver to produce a representative pressure field. The pressure field was then down sampled and combined with additive IID Gaussian noise to obtain the data. The noise variance is 1×10^{-4}.

Benchmark results were obtained using MCMC on the full 100 dimensional fine-scale space with MsFEM as the forward model. We used two variants of MCMC in our tests: the delayed rejection Adaptive-Metropolis (DRAM) MCMC algorithm [41] and a preconditioned Metropolis-adjusted Langevin Algorithm (PreMALA) [86]. DRAM uses a Gaussian random walk proposal with a covariance that is adapted based on the sample covariance. The DRAM algorithm was tuned to have an acceptance rate of 35%. Two stages were used for DR part of DRAM, but the second stage was turned off after 7e4 MCMC steps. While the PreMALA proposal is not adaptive, we set the proposal covariance to the inverse Hessian at the posterior MAP point. The PreMALA algorithm also uses Gradient information to shift the proposal towards high density regions of the posterior. For the single-scale posterior here, finite differences were used to compute the Hessian, which may have hindered PreMALA performance in Table 3.2. Also, for both PreMALA and DRAM, 5e6 steps were used in the chain, 1e5 of which were used as a burn in period after starting the chain from the MAP point. The DRAM samples are used for the accuracy comparison in Table 3.1.

Sampling the coarse posterior $\pi(r_1|d)$ was also performed with PreMALA. Again the Hessian at the MAP point was used; however, PreMALA was found to be more efficient than DRAM for exploring this coarse posterior. The algorithm was tuned to have an acceptance rate of around 55%.

When the multiscale definition in (3.1) is completely satisfied and exact transport maps are used, posterior samples produced by our multiscale framework will be samples from $\pi(\theta|d)$. However, as described in the preceding sections, various assumptions and approximations are necessary to efficiently compute the transport maps. This means that, *in this application*, our multiscale method only approximately samples $\pi(\theta|d)$. Table 3.1 and Figure 3-4 show that this approximation is reasonable and that no significant errors were made in the construction of $\tilde{F}_1$ and $\tilde{F}_2$.

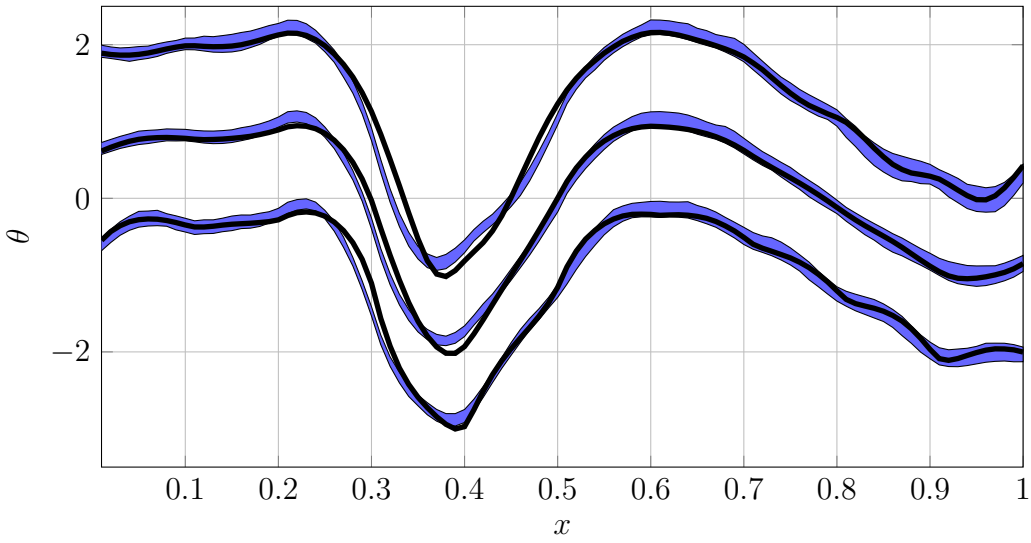
However, as shown in Figure 3-4, there is an obvious negative bias in the results near $x = 0.3$. This is likely caused by the approximation of F_2 on parameters near that point. A coarse element boundary exists at $x = 0.3$ and there is large dip in θ over the element boundary. Such large dips are not in the high density regions of the prior and restricting F_2 to only have local nonlinearities may be preventing the map from adequately capturing the tail behavior necessary to exactly characterize the posterior. With a more expressive coarse to fine map F_2, this bias would decrease. However, in all other locations, the true MCMC posterior and the multiscale posterior are in good agreement.

Table 3.1: Estimate of bias in quantile estimates for the multiscale inference framework. Index refers to a particular fine element where the quantile was computed. E_a is the average error (i.e., bias) between the MCMC estimate and multiscale estimate of the a-percent quantile.

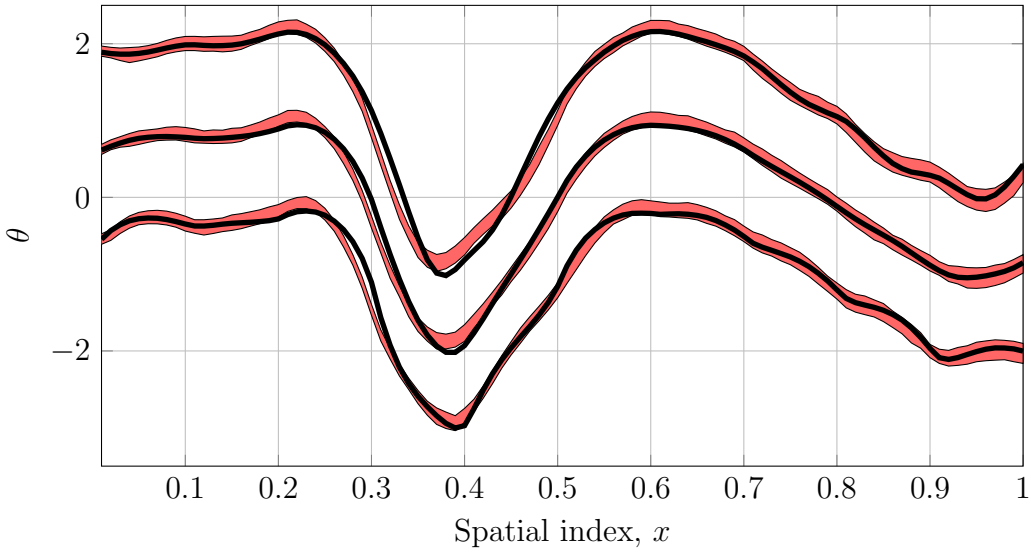
$\tilde{F}_1$ order	$\tilde{F}_2$ order	x	E_{05}	E_{25}	E_{50}	E_{75}	E_{95}
1	1	0.1	1.11e-02	4.72e-02	5.92e-02	5.67e-02	3.35e-02
		0.3	-3.58e-01	-3.05e-01	-2.83e-01	-2.75e-01	-2.83e-01
		0.5	-1.28e-01	-1.40e-01	-1.62e-01	-1.95e-01	-2.60e-01
		0.9	1.86e-02	6.46e-02	7.85e-02	7.92e-02	5.02e-02
	3	0.1	2.16e-01	1.26e-01	5.42e-02	-2.58e-02	-1.53e-01
		0.3	-1.49e-01	-2.21e-01	-2.80e-01	-3.44e-01	-4.42e-01
		0.5	7.50e-02	-5.98e-02	-1.61e-01	-2.67e-01	-4.23e-01
		0.9	2.45e-01	1.54e-01	7.91e-02	-4.53e-03	-1.35e-01
3	1	0.1	-9.24e-04	3.60e-02	4.77e-02	4.76e-02	2.28e-02
		0.3	-3.60e-01	-3.05e-01	-2.84e-01	-2.73e-01	-2.79e-01
		0.5	-1.00e-01	-1.12e-01	-1.34e-01	-1.69e-01	-2.34e-01
		0.9	2.17e-03	4.84e-02	6.26e-02	6.31e-02	3.53e-02
	3	0.1	2.15e-01	1.22e-01	4.17e-02	-4.46e-02	-1.90e-01
		0.3	-4.11e-02	-9.15e-02	-1.32e-01	-1.78e-01	-2.54e-01
		0.5	1.06e-01	-3.52e-02	-1.39e-01	-2.49e-01	-4.15e-01
		0.9	2.42e-01	1.30e-01	4.74e-02	-3.97e-02	-1.79e-01

Posterior expectations such as quantiles only tell part of the story. Another important feature of the posterior is the correlation structure of the posterior realizations. As shown in Figure 3-5, our multiscale approach correctly produces posterior samples with the same rough structure as the prior.

Now consider the efficiency of the multiscale method. The effective sample size (ESS) is one measure of the information contained in a set of posterior samples. The ESS represents the number of effectively independent samples contained in the set. In an MCMC context, we can easily compute this quantity for a chain at equilibrium [107]. However, a more fundamental definition of ESS using the variance of a Monte Carlo estimator will be used here. Assume we have a Monte Carlo estimator $\hat{\theta}_i$ for the mean of θ_i. The ESS for such an estimator is given by the ratio of the random



(a) 95% Region of multiscale quantile estimator (shaded blue) using cross-covariance map and "Gold Standard" MCMC quantile.



(b) 95% Region of multiscale quantile estimator (shaded red) using local polynomial map and "Gold Standard" MCMC quantile.

Figure 3-4: Comparison of multiscale estimate of posterior quantiles and a fine-scale MCMC approach. The MCMC chain was run 4.9 million steps and the quantiles here are taken as the "true" quantiles in our analysis. Note that the vertical grid lines correspond to coarse element boundaries. A quantitative summary of these plots is given in Table 3.1.

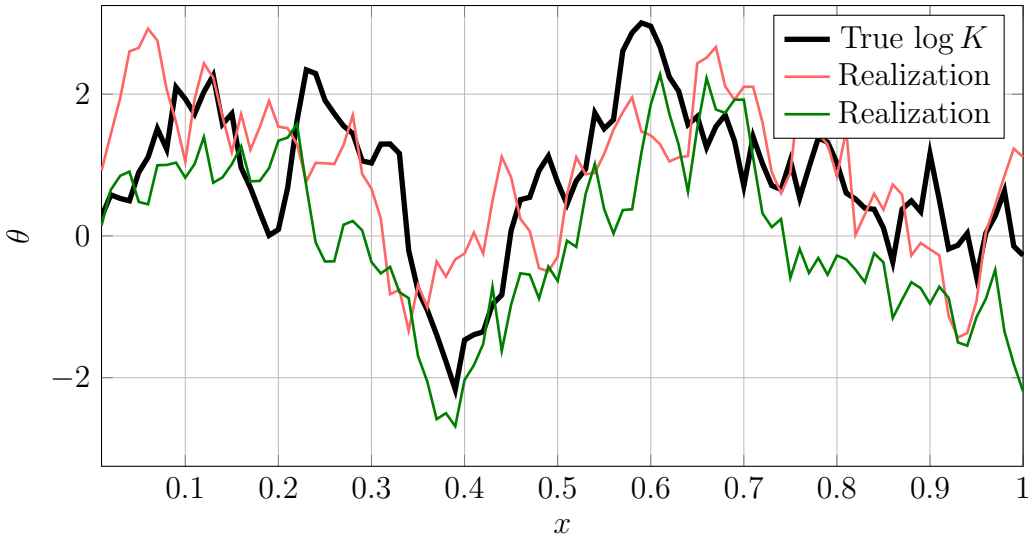


Figure 3-5: Comparison of posterior realizations with true log conductivity. Clearly, the posterior samples maintain the same rough correlation structure as the true log conductivity. As in Figure 3-4, the vertical grid lines correspond to coarse element boundaries.

variable variance and the estimator variance

$$\text{ESS}_i = \frac{\mathbb{V}\text{ar}(\theta_i)}{\mathbb{V}\text{ar}(\hat{\theta}_i)}, \tag{3.49}$$

where ESS_i is the effective sample size for dimension i. This expression for ESS is more difficult to compute than the methods in [107] because evaluating $\mathbb{V}\text{ar}(\hat{\theta}_i)$ requires many samples of the Monte Carlo estimator (i.e., running the inference procedure many times); however, this approach is less susceptible to errors stemming from autocorrelation integration and does not require us to use samples from an MCMC scheme; we can instead use samples from our multiscale scheme. The effective sample size can be computed for each dimension of the chain so a range of effective sample sizes is obtained. Unless otherwise noted, the ESS values reported here will refer to the minimum effective sample size over all dimensions.

Table 3.2 shows the efficiency of our multiscale approach compared to two hand tuned fine-scale MCMC samplers. Comparing the DRAM MCMC with the multiscale results, we can see that even when a nonlinear coarse to fine map $\tilde{F}_2$ is used, proper tuning of the method can speed up the number of effectively independent samples generated per second by a factor of 2 with one fine sample per coarse sample, $M = 1$. Moreover, when a linear $\tilde{F}_2$ is employed, we can see a speed up of 4.5 times when 5 fine samples are generated for each coarse sample $M = 5$. These results indicate that as long as minor approximations to the posterior are acceptable, even when direct single scale approaches could be applied, there is a clear advantage of using our multiscale approach.

Table 3.2: Comparison of posterior sampling efficiency between DRAM MCMC and variants of our multiscale framework. The key column is ESS/t_{on}, where the high numbers indicate our method generates more effectively independent samples per second. Note that even with a more efficient MCMC sampler, it is unlikely that the MCMC sampler will be able to outperform the efficiency of our multiscale approach with linear $\tilde{F}_2$.

Method	N	M	t_{on} (sec)	Min ESS	Max ESS	Min ESS/t_{on}
MCMC-DRAM	4900000	NA	2252.63	6340	11379	**2.8**
MCMC-PreMALA	4900000	NA	2773.47	274	729	**0.1**
Cross Covariance	500000	1	287.31	2987	20244	**10.4**
Cross Covariance	500000	5	314.17	3971	14597	**12.6**
Local Cubic	450000	1	937.41	5408	23679	**5.8**
Local Cubic	450000	5	3555.05	5294	18759	**1.5**

Using the timing and ESS data from Table 3.2 for $M = 1$ and $M = 5$, we can also compute the optimal number of fine samples to generate for each coarse sample. To use the optimal expression in (3.22), we first use a simple least squares approach to compute the unknown coefficients C_1 and C_2. For the linear case, we obtain $C_1 = 22.7867$ and $C_2 = 10.2019$, which yields an optimal value of $M = 4$. For the local cubic case, we obtain $C_1 = 11.6076$ and $C_2 = 3.3135$ which yields an optimal value of $M = 1$. Clearly, it is worth generating additional fine-scale samples for the inexpensive linear map, but for the slightly more expensive cubic map it is not worthwhile (in terms of time) to generate more fine-scale samples. The time would be better spent generating coarse samples. This can also be seen in Table 3.2 directly. For the linear cross covariance map, the ESS/t_{on} is larger for $M = 5$ than $M = 1$, while the opposite is true for the local cubic map.

These values for M are dependent on the cost of each coarse model evaluation. In this 1d problem, the coarse model is incredibly cheap to evaluate, on par with the cost of evaluating the cubic map. However, for problems with expensive model evaluations or with poor coarse MCMC mixing, this will not be the case, and larger values of M will be optimal.

3.6.2 Two spatial dimensions

The relatively small dimension of θ in the 1D problem above allowed us to compare our multiscale approach with an MCMC gold-standard. However, we expect our multiscale inference approach to yield even larger performance increases on large-scale problems where direct use of MCMC may not be feasible at all. Here we will again infer a log conductivity field using MsFEM as a forward solver; however, this example will have two spatial dimensions. The 2D grid is defined by an 8×8 mesh of coarse elements over $[0, 1] \times [0, 1]$, with 13×13 fine elements in each coarse element. The log-conductivity is defined as piecewise constant on each fine element, resulting in a 10816 dimensional inference problem! The zero mean prior is again defined by an exponential kernel with correlation length 0.1. In two dimensions, this kernel takes

the form

$$\mathbb{C}\text{ov}\left(\theta(x_1), \theta(x_2)\right) = \sigma_\theta^2 \exp\left[-\frac{\|x_1 - x_2\|_2}{L}\right], \tag{3.50}$$

where $\|\cdot\|_2$ is the usual Euclidean norm, $\sigma_\theta^2 = 1.0$, and $L = 0.1$. Notice that is an isotropic kernel, but is not separable.

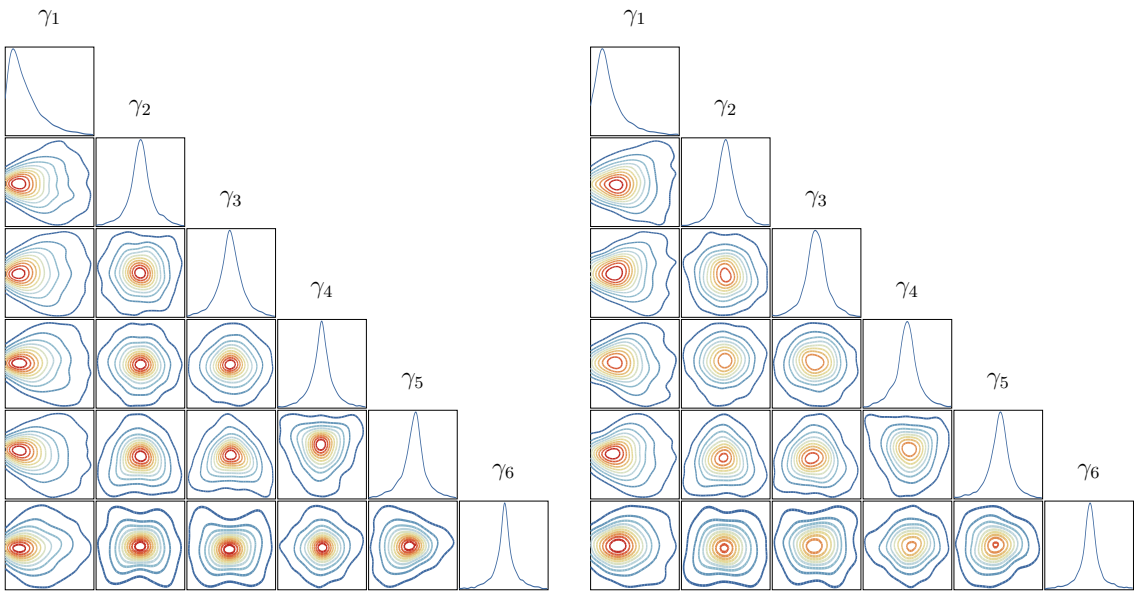
Synthetic data are generated by a full fine-scale simulation using a standard Galerkin FEM and iid Gaussian noise is added to the pressure at each of the coarse nodes. The noise variance is $1\text{e}-6$. For boundary conditions, an increasing Dirichlet condition is used at $y = 0$ while a decreasing Dirichlet condition is used at $y = 1$. Homogeneous (i.e., no flow) boundaries are used for $x = 0$ and $x = 1$. Specifically the top and bottom Dirichlet conditions are given by:

$$p(x, y = 0) = x \tag{3.51}$$

$$p(x, y = 1) = 1 - x \tag{3.52}$$

Since we cannot apply any other sampling method to this large problem directly, our confidence in the posterior accuracy stems directly from the accuracy of transport maps $\tilde{F}_1$ and $\tilde{F}_2$. From the one dimensions results, we know that linear $\tilde{F}_2$ derived from the cross covariance of r_1 and θ performs quite well and it is reasonable to assume the same performance in the 2d case. In addition to the accuracy of $\tilde{F}_2$, a qualitative verification of the coarse map is given in Figure 3-6. The figure shows the true prior density of the coarse parameters over one coarse element as well as the density defined by the coarse map, $\tilde{F}_1$ in (3.47). From this figure, we see that the coarse map well represents the coarse prior. In terms of computational effort, the map was constructed using 85000 prior samples and multivariate Hermite polynomials limited to a total order of 7. Taking advantage of 16 compute nodes, each employing 4 threads on a cluster with 3.6GHz intel Xeon E5-1620 processors, the prior sampling, $\tilde{F}_1$ construction, and $\tilde{F}_2$ construction took under an hour for this problem.

With confidence in the transport maps, we can move on to posterior sampling. The PreMALA MCMC algorithm with a scaled inverse MAP Hessian as the proposal covariance was again used to sample the coarse posterior. The coarse MCMC chain was run for 2e5 steps to generate the coarse samples. It is relatively simple to compute gradients of the coarse posterior using adjoint methods, which allows us to use the Langevin approach. Moreover, incorporating derivative information helps us tackle the still relatively large dimension of the coarse posterior. The Hessian of the coarse posterior at the maximum a posteriori (MAP) point was used as the preconditioner. Even though the coarse sampling problem still has 384 parameters, the coarse map $\tilde{F}_1$ captures much of the problem structure and the coarse MCMC chain mixes remarkably well, achieving a near optimal acceptance rate of 60%. Ten independent parallel chains were run and completed the coarse sampling in 49 minutes. After coarse sampling with MCMC, the coarse samples were combined with independent samples of r_2 through $\tilde{F}_2$ to generate posterior samples of the fine-scale variable θ. This coarse to fine sampling took 61 minutes. Figure 3-7 shows the posterior sample mean and variance as well as two posterior samples. A single fine sample was generated for each



(a) True coarse density.

(b) Map-induced coarse density.

Figure 3-6: Comparison of the true coarse prior density and the coarse prior density induced by $\tilde{F}_1$. In this case, a seventh order Hermite polynomial expansion was used to parameterize $\tilde{F}_1$. The first coarse parameter on each coarse cell, corresponding to ζ_1, is the most difficult for the map to capture because of the lognormal shape. The color scales, contour levels, and axis bounds are the same for both plots.

coarse sample. Notice that the fine-scale realizations have the same rough structure as the true $\log(\kappa)$ field. This is an important feature of our work that is not present in many methods based on a priori dimension reduction, such as the use of truncated KL expansions.

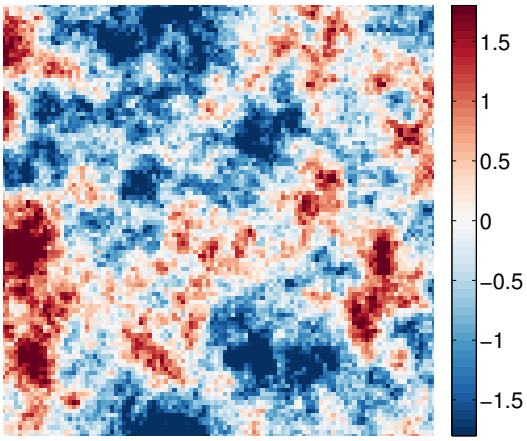
3.7 Discussion

We have developed a method for utilizing optimal transport maps for efficiently solving Bayesian inference problems exhibiting the multiscale structure defined in (3.1). Our use of optimal transport maps enabled us to decouple the original inference problem into a manageable coarse sampling problem (sampling $\pi(r_1|d)$) and a coarse to fine "projection" of the posterior (evaluating T_2 with posterior r_1 samples). By utilizing locality and stationary, we were able to build these transport maps despite the large dimension of spatially distributed parameters in our examples. While not exact, our method does produce samples that well-approximate the true posterior. Moreover, as illustrated in the 2d example, this approach can be applied to problems that are intractable when using standard sampling methods.

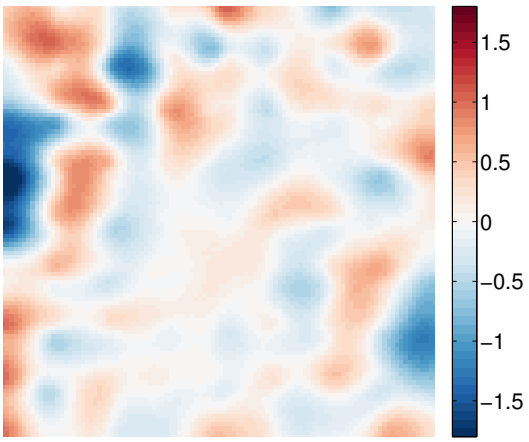
Tackling a Bayesian inference problem as large as the 2d example above would take several days if not weeks with any typical sampler. However, our use of transport maps to decouple the problem has allowed us to efficiently solve the problem in only

a few hours. Part of this dramatic time reduction stems from the inherit parallelism in our approach. All the prior sampling, much of the optimization used to build the transport maps, and all of the post-MCMC $\tilde{F}_2$ evaluations can be parallelized. This is contrast to nearly all samplers based on MCMC, as MCMC is an inherently serial process. While we use some algorithm level parallelism utilizing MPI over multiple CPU's, more sophisticated parallel architectures could also be exploited in future applications of this work. One possibility is to use general purpose GPU computing to accelerate the fine to coarse model evaluations, and the coarse to fine evaluations of $\tilde{F}_2$. This has the potential to dramatically accelerate the sampling.

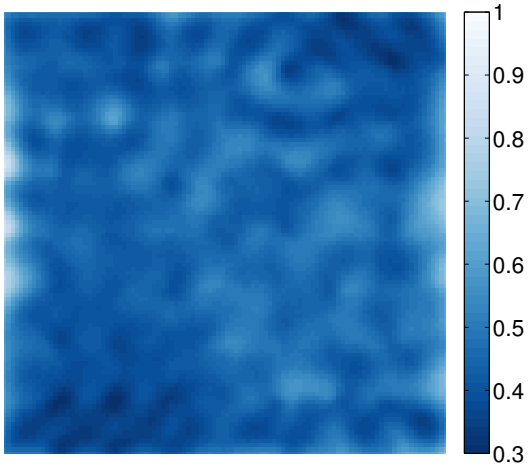
In the two porous media examples described above, the posterior samples generated by our multiscale method are approximate. However, the accuracy of this approximation can be controlled by the representation of the maps $\tilde{F}_1$ and $\tilde{F}_2$. In example 2, we restricted ourselves to a linear $\tilde{F}_2$ because one of our major concerns was computational run time. However, in applications where more exact posterior sampling is required, a higher polynomial order or alternative functional representation could be employed. In the same vein, future development of adaptive map-construction techniques could automatically find the problem structure (locality and stationarity) that we exploited in our examples. Simultaneous use of problem structure and the layered maps from Chapter 2 is one potentially exciting area. However, our multiscale framework is not tied to a specific approach for constructing transport maps and will benefit from any future research in that area.



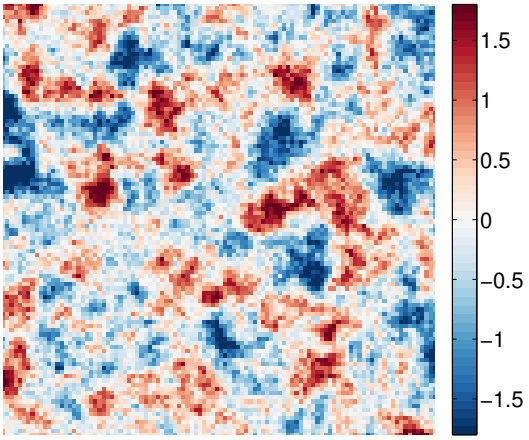
(a) True $\theta = \log(\kappa)$ field.



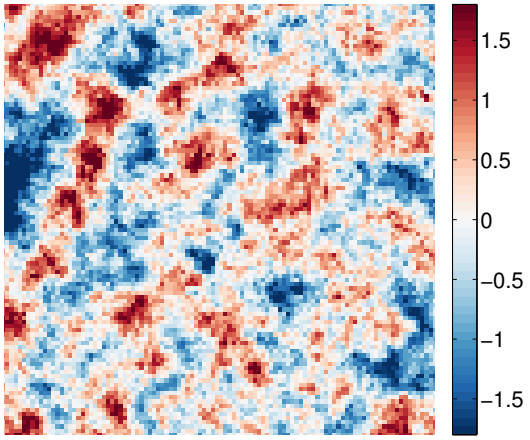
(b) Posterior mean using multiscale approach.



(c) Posterior variance using multiscale approach.



(d) Posterior Realization



(e) Posterior Realization

Figure 3-7: Application of multiscale inference framework to 10404 dimensional problem using MsFEM and 360 dimensional coarse problem.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Transport map accelerated MCMC

The ultimate goal of Bayesian inference and uncertainty quantification in general, is to answer questions like: *what is the expected global temperature rise over the next ten years?* or *how much will energy production vary in Wyoming?* As mathematicians and engineers, we express these questions in terms of a quantity of interest, Q, which is defined by an integral over the D_θ-dimensional random variable θ. Recall that $\mathcal{X} \subseteq \mathbb{R}^D$ is the sample space containing all possible values of θ and μ_θ is the distribution of θ. We assume that μ_θ admits a continuous density, which allows us to write the quantity of interest as

$$Q = \int h(\theta)\pi_n(\theta)d\theta. \tag{4.1}$$

Here, $\pi_n(\theta)$ is the probability density related to μ_θ and $h(\theta)$ is an application-specific function.

Unfortunately, computing expectations such as (4.1) is analytically intractable in realistic situations. A general and robust solution is then to use Monte Carlo integration to approximate (4.1) with a finite set of samples $\{\theta^{(1)}, \theta^{(2)}, ..., \theta^{(N)}\}$ taken from $\pi_n(\theta)$. The Monte-Carlo approximation is given by

$$Q \approx \frac{1}{N}\sum_{i=1}^{N} h(\theta^{(i)}). \tag{4.2}$$

The Monte Carlo approach is easy to apply when independent samples of $\pi_n(\theta)$ are efficiently obtained; however, our primary interest is in Bayesian inference problems, where the target density $\pi_n(\theta)$ can be constructed from sophisticated physical models, which makes direct sampling impossible. We therefore need more advanced methods for generating samples of $\pi_n(\theta)$, such as Markov chain Monte Carlo (MCMC).

Using the thorough discussion of transport maps from Chapter 2, the remaining sections of this chapter will first expand on Chapter 1 with a more thorough overview of MCMC (Section 4.1.1), then formulate our new class of adaptive map-based MCMC algorithms (Sections 4.1 and 4.2), describe the relationship between our approach and differential geometric MCMC methods (Section 4.3), briefly discuss convergence (Section 4.4), and finally, compare the performance of map-based MCMC algorithms

against existing methods on a range of test problems (Section 4.5).

4.1 Transport-map accelerated MCMC

Unlike the direct use of an approximate map $\tilde{T}(\theta)$, MCMC methods provide a mechanism for generating *exact* samples of the target distribution π [83]. These methods work by constructing a Markov a chain with correlated states $\{\theta^{(1)}, \theta^{(2)}, ..., \theta^{(K)}\}$ such that the stationary distribution of the chain corresponds to the density $\pi_n(\theta)$, and that the chain states themselves can be used in a Monte Carlo approximation similar to (4.2). One of the most general ways of defining such a chain is the Metropolis-Hastings rule.

4.1.1 The Metropolis-Hastings Rule

In the Metropolis-Hastings setting, generating a new state $\theta^{(n+1)}$ from the current state $\theta^{(n)}$ in a Markov chain is achieved with a two step process. First, a sample θ' is drawn from some proposal distribution $q_\theta(\theta'|\theta^{(n)})$. Then, an accept-reject stage is performed: $\theta^{(n+1)}$ is set to θ' with probability $\alpha(\theta', \theta^{(n)})$ and is set to $\theta^{(n)}$ with probability $1 - \alpha(\theta', \theta^{(n)})$, where the acceptance probability $\alpha(\theta', \theta^{(n)})$ is given by the well-known Metropolis-Hastings rule [45, 76]

$$\alpha\left(\theta', \theta^{(n)}\right) = \min\left\{1, \frac{\pi(\theta')q_\theta(\theta^{(n)}|\theta')}{\pi(\theta^{(n)})q_\theta(\theta'|\theta^{(n)})}\right\}. \tag{4.3}$$

Notice that the choice of proposal distribution q_θ controls the correlation between states in the MCMC chain through both the acceptance rate and the step size. Since the correlation in the Markov chain can dramatically alter the accuracy of a Monte Carlo approximate based on the samples, it is important to to use a proposal distribution that proposes large steps with a high probability of acceptance.

Classic examples of proposals include a Gaussian density with mean θ and fixed covariance (i.e., "Random-Walk Metropolis" (RWM)), a Gaussian density with mean $\theta + d$ for a drift d based on the gradient of $\log \pi$ (i.e., "Metropolis-Adjusted Langevin Algorithm" (MALA)), and a mechanism based on Hamiltonian Dynamics (i.e., "Hamiltonian Monte Carlo" (HMC)). The mixing and convergence properties of these proposals have been extensively studied, see [84] and [55] for RWM, [88] and [9] for MALA, and [80] for HMC. A nice discussion of optimal scaling results for both RWM and MALA proposals can also be found in [89].

Notice that all of these proposals have parameters that need to be tuned, e.g., the Gaussian proposal covariance or the number of integration steps to use in HMC. While MCMC theory provides a guideline for optimally tuning these parameters, effectively sampling a particular target density π_n usually involves hand tuning the parameters, which can be a painfully tricky task. Adaptive strategies try to overcome this issue by learning the proposal parameters as the MCMC chain progresses.

In the RWM and MALA proposals, the previous states can be used to obtain sample estimates of the target covariance and the proposal covariance can then be

set to a scaled version of sample covariance. In the RWM case, this idea leads to the "Adaptive Metropolis" (AM) algorithm first proposed in [42]. For the MALA case, this type of covariance adaptation is discussed in [9]. By matching the proposal covariance to the target covariance, the proposals can take much larger steps and still stay within high density regions of the parameter space. This helps these methods automatically find an adequate proposal.

A chain constructed with an adaptive proposal is not strictly Markov, but as shown in [4] and reviewed in [5], the adaptive chain can still be ergodic under some surprisingly mild conditions on the sequence of proposal distributions $q_\theta^{(n)}$. More on the convergence properties of adaptive MCMC will be discussed in Section 4.4, where we show the ergodicity of our own adaptive approach.

4.1.2 Combining maps and MCMC

Assume for this section that we are given a previously constructed approximate map $\tilde{T}(\theta)$. Unfortunately, using an approximate map means that pushing samples of the reference density $p(r)$ through $\tilde{T}^{-1}$ will yield *inexact* samples of the target density $\pi(\theta)$. However, as shown in Figure 2-1, even an approximate map can still capture much of the target distribution's structure. This feature of $\tilde{T}$ can be exploited by combining the map with either a local or independent Metropolis-Hastrings proposal distribution on the reference space, denoted by $q_r(r'|r)$, to create an efficient proposal density on the target space, denoted by $q_\theta(\theta'|\theta)$. When the map is fixed, this process can be viewed in two ways: (1) the canonical reference proposal $q_r(r'|r)$ is applied to a map-induced approximate reference density $\tilde{p}(r)$, or (2) the map-induced target proposal $q_\theta(\theta'|\theta)$ is applied to the original target density $\pi(\theta)$. The details below will generally follow this second map-induced proposal view, but readers may find the first idea of transforming the target density intuitively useful.

Algorithm details

Let $q_r(r'|r)$ be a standard MCMC proposal on the reference space. Note that this proposal could be any valid MCMC proposal such as the typical random walk proposal, Langevin proposal, Gaussian independence proposal, or even a delayed rejection proposal. Using the transport map, this reference-space proposal induces a target-space proposal density defined by

$$q_\theta(\theta'|\theta) = q_r\left(\tilde{T}(\theta')|\tilde{T}(\theta)\right)\left|\det\left(\partial\tilde{T}(\theta')\right)\right| \tag{4.4}$$

This expression provides an easy way of evaluating the target proposal. However, in an MCMC context, we need to be able to both evaluate the proposal density and sample from the proposal. Fortunately, sampling the target proposal $q_\theta(\theta'|\theta)$ only involves the three steps outlined below,

1. Use the current target state θ_c to compute the current reference state, $r_c = \tilde{T}(\theta_c)$
2. Draw a sample r' from the reference proposal, $r' \sim q_r(r'|r_c)$.

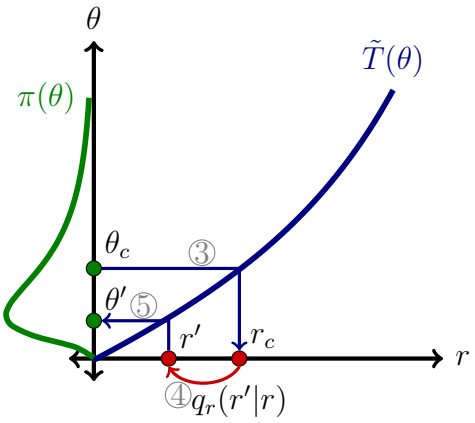


Figure 4-1: Illustration of Metropolis-Hastings proposal process in transport-map accelerated MCMC. Here, the reference proposal r' is accepted. The gray circled numbers on each arrow correspond the the line number in algorithm 4.1.

3. Evaluate the inverse map at r' to obtain a sample of the target proposal q_θ: $\theta' = \tilde{T}^{-1}(r')$.

These steps are also outlined as steps 3-5 in Algorithm 4.1 and Figure 4-1. Using this proposal process in a standard Metropolis-Hastings MCMC algorithm yields the method in Algorithm 4.1. Notice that Algorithm 4.1 is fundamentally equivalent to the standard Metropolis-Hastings algorithm.

We also need to point out that evaluating the inverse map $\tilde{T}^{-1}(r)$ only involves D one dimensional nonlinear solves. This is a result of the map's lower triangular structure. These one dimensional problems can be solved efficiently with a simple Newton method or, in the presence of a polynomial map, with a bisection solver based on Sturm sequences [106].

Handling derivative-based proposals

An important feature of our approach is that the map-induced proposal $q_\theta(\theta'|\theta)$ only requires derivative information from the target density $\pi(\theta)$ when the reference proposal $q_r(r'|r)$ requires derivative information. For example, if $q_r(r|r_c)$ requires gradient information, then our approach will require gradient information from $\pi(\theta)$. However, when $q_r(r|r_c)$ only requires density evaluations, our method will only require density evaluations. Another important feature of Algorithm 4.1 is that we do not require $\pi(\theta)$ to take any particular form (e.g., $\pi(\theta)$ does not need to be a Bayesian posterior, have a Gaussian prior, etc..). The ability to work on arbitrary densities where derivative information may or may not be present distinguishes us from some other recent MCMC approaches such as the Geodesic MCMC introduced in [39], the no u-turn sampler in [47], the discretization invariant approaches of [26], or the randomize-then-optimize algorithm of [12]. While our approach can perform quite well without derivative information, we can still accommodate proposals that require such higher order information.

Algorithm 4.1: MCMC algorithm with fixed map.

Input: A starting point θ_0, A preconstructed transport map $\tilde{T}(\theta)$ and a valid reference proposal $q_r(r|r_c)$

Output: Samples of the target distribution, $\left\{\theta^{(1)}, \theta^{(2)}, ..., \theta^{(N)}\right\}$

Set the current state to the starting point $\theta_c = \theta_0$

for $k \leftarrow 1$ ***to*** N **do**

  Compute the reference state, $r_c = \tilde{T}(\theta_c)$

  Sample the reference proposal, $r' \sim q_r(r|r_c)$

  Compute the target proposal, $\theta' = \tilde{T}^{-1}(r')$

  Calculate the acceptance probability given by

$$\alpha = \min\left\{1, \frac{\pi(\tilde{T}^{-1}(r'))}{\pi(\tilde{T}^{-1}(r))} \frac{q_r\left(r|r'\right)}{q_r\left(r'|r\right)} \frac{\det[\partial_r \tilde{T}^{-1}(r')]}{\det[\partial_r \tilde{T}^{-1}(r)]}\right\} \tag{4.5}$$

  Set θ_c to θ' with probability α

  Store the sample, $\theta^{(k)} = \theta_c$

return *Target samples* $\left\{\theta^{(1)}, \theta^{(2)}, ..., \theta^{(N)}\right\}$

Here we show that a simple application of the chain rule allows us propagate target density derivatives through the transport map, which then enables us to use reference proposals that exploit derivative information. From the acceptance ratio in Algorithm 4.1, we see that the reference proposal q_r is targeting the map-induced density $\tilde{p}$ defined by

$$\tilde{p}(r) = \pi(\tilde{T}^{-1}(r)) \left|\det(\partial \tilde{T}^{-1}(r))\right|. \tag{4.6}$$

By taking advantage of the map's lower triangular structure, we can write the log of this density as

$$\log \tilde{p}(r) = \log \pi(\tilde{T}^{-1}(r)) + \sum_{d=1}^{D} \log \frac{\partial \tilde{T}_d^{-1}}{\partial r_d}. \tag{4.7}$$

We will use the chain rule to get the gradient of this expression. First, we make the substitution $\theta = \tilde{T}^{-1}(r)$ and then take the gradient with respect to θ. This leads to

$$\nabla_\theta \log \tilde{p}(\tilde{T}(\theta)) = \nabla_\theta \log \pi(\theta) - \sum_{d=1}^{D} \left(\frac{\partial \tilde{T}_d}{\partial \theta_d}\right)^{-1} H_d(\theta), \tag{4.8}$$

where H_i is a row vector containing second derivative information coming from the determinant term $H_d(\theta) = \left[\begin{array}{cccc} \frac{\partial^2 \tilde{T}_d}{\partial \theta_1 \theta_d} & \frac{\partial^2 \tilde{T}_d}{\partial \theta_2 \theta_d} & ... & \frac{\partial^2 \tilde{T}_d}{\partial \theta_D \theta_d} \end{array}\right]$. Now, accounting for our change of variables, we have an expression for the reference gradient given by

$$\nabla_r \log \tilde{p}(r) = \left(\nabla_\theta \log \pi(\theta) - \sum_{d=1}^{D} \left(\frac{\partial \tilde{T}_d}{\partial \theta_d}\right)^{-1} H_d(\theta)\right)(D_\theta \tilde{T}(\theta))^{-1}. \tag{4.9}$$

Note that this expression is only valid when $\theta = \tilde{T}^{-1}(r)$. This expression can now be used by any gradient-based reference proposal.

The lower triangular structure not only allows us to expand the determinant and derive (4.9), but also allows us to easily apply the inverse Jacobian $(D_\theta T(\theta))^{-1}$ through forward substitution. Furthermore, computing the Jacobian $D_\theta \tilde{T}(\theta)$ or the second derivatives in $H_d(\theta)$ is trivial when polynomials or other well-studied basis functions are used to parameterize the map.

4.2 Adaptive transport-map MCMC

With a sufficiently accurate map, Algorithm 4.1 provides a way to efficiently generate samples of the target density $\pi(\theta)$. However, to construct the transport map used in the algorithm, we need to already have samples of $\pi(\theta)$. This is a classic chicken and the egg problem – we need the map to generate the samples, but we need the samples to construct the map. We will overcome this dilemma with an adaptive MCMC approach that builds $\tilde{T}$ as the MCMC iteration progresses.

4.2.1 Adaptive algorithm overview

In our adaptive MCMC scheme, we initialize the sampler with an existing map $\tilde{T}_0$ and update the map every N_U steps using *all* of the previous states in the MCMC chain. This is conceptually similar to the usual adaptive Metropolis algorithm from [42]. However, in [42], the previous states are only used to update the covariance matrix of a Gaussian proposal but in our case, the previous states are used to construct a nonlinear transport map that yields a more sophisticated non-Gaussian proposal.

The simplest version of our adaptive algorithm would find the coefficients for each dimension of the map γ_d by solving (2.21) directly. However, when the number of existing samples k is small, or there is a lot of correlation in the chain, the Monte Carlo sum in (2.21) will poorly approximate the true integral and lead to transport maps that do not capture the structure of $\pi(\theta)$. To overcome this issue, we use a regularization term on the map coefficients γ_d. Our goal for introducing this term, call it $g(\gamma_d)$, is to help ensure the map does not collapse onto one region of the target space. Such a collapse would make it difficult for the chain to efficiently explore the entire support of π. To build a map with the regulation, we use the following modified objective:

$$\begin{aligned} \underset{\gamma_d}{\text{minimize}} \quad & g(\gamma_d) + \sum_{i=1}^{k} \left[0.5\tilde{T}_d^2(\theta^{(i)};\gamma_d) - \log \left. \frac{\partial \tilde{T}_d(\theta;\gamma_d)}{\partial \theta_d} \right|_{\theta^{(i)}} \right] \\ \text{subject to} \quad & \left. \frac{\partial \tilde{T}_d(\theta;\gamma_d)}{\partial \theta_d} \right|_{\theta^{(i)}} \geq d_{min} \quad \forall i \in \{1,2,...,k\} \end{aligned} \tag{4.10}$$

In practice, we choose $g(\gamma_d)$ to prevent the map from "getting too far" from the identity map. However, if additional problem structure is known, such as a the

covariance of θ, this could also be incorporated into the regularization. In the identity regularization case, we use a simple quadratic regularization function centered at the coefficients of the identity map. Let γ_I be coefficients of the identity map, then our choice of $g(\gamma_d)$ takes the form $g(\gamma_d) = k_R\|\gamma_d - \gamma_I\|^2$ where k_R is a user-defined regularization parameter that needs to be tuned for each target density. In practice, we have found most small values of k_R to yield similar performance and we usually set $k_R = 10^{-4}$.

Algorithm 4.2 shows how we use the regularized objective in (4.10) for our adaptive MCMC framework. Notice that the only difference between the adaptive approach in Algorithm 4.2 and the fixed-map approach in Algorithm 4.1 is the map update on lines 10-14 of Algorithm 4.2.

Algorithm 4.2: MCMC algorithm with adaptive map.

Input: A starting point θ_0, An initial vector of transport map parameters $\gamma^{(0)}$ and a valid reference proposal $q_r(r'|r)$

Output: Samples of the target distribution, $\left\{\theta^{(1)}, \theta^{(2)}, ..., \theta^{(N)}\right\}$

1 Set the current state to the starting point $\theta_c = \theta_0$

2 Set the current map parameters $\gamma^{(1)} = \gamma^{(0)}$

3 **for** $k \leftarrow 1$ ***to*** N **do**

4 Compute the reference state, $r_c = \tilde{T}(\theta_c; \gamma^{(k)})$

5 Sample the reference proposal, $r' \sim q_r(r|r_c)$

6 Compute the target proposal, $\theta' = \tilde{T}^{-1}(r'; \gamma^{(k)})$

7 Calculate the acceptance probability given by

$$\alpha = \min\left\{1, \frac{\pi(\tilde{T}^{-1}(r';\gamma^{(k)}))}{\pi(\tilde{T}^{-1}(r;\gamma^{(k)}))}\frac{q_r\left(r|r'\right)}{q_r\left(r'|r\right)}\frac{\det[\partial_r\tilde{T}^{-1}(r';\gamma^{(k)})]}{\det[\partial_r\tilde{T}^{-1}(r;\gamma^{(k)})]}\right\}$$

8 Set θ_c to θ' with probability α

9 Store the sample, $\theta^{(k)} = \theta_c$

10 **if** $(k \bmod N_U) \equiv 0$ **then**

11 **for** $d \leftarrow 1$ ***to*** D **do**

12 Update $\gamma_d^{(k+1)}$ by solving (4.10)

13 **else**

14 $\gamma(k+1) = \gamma^{(k)}$

15 **return** *Target samples* $\left\{\theta^{(1)}, \theta^{(2)}, ..., \theta^{(N)}\right\}$

4.2.2 Complexity of map update

At first glance, updating the map every N_U steps may seem computationally intractable. Fortunately, the form of the map optimization problem in (4.10) allows

for efficient updates. When N_U is small relative to the current number of steps k, the map objective function in (4.10) changes little between updates and the previous map coefficients provide a good initial guess for the new optimization problem. This means that the new optimal coefficients can be found in only a few Newton iterations, sometimes only one or two. Moreover, when a polynomial expansion like (2.23) is used to parameterize the map, we can simply add a new row to the matrices in (2.33) at each iteration, which helps prevent duplicate polynomial evaluations. Combining this caching with efficient matrix-vector products makes evaluating (4.10) very efficient. As the timing results show in Section 4.5, by using these computational innovations, the advantage of using the map to define q_θ greatly outweighs the additional cost of updating the map.

4.2.3 Monitoring map convergence

As the map in Algorithm 4.2 is adapted, the best choice of reference proposal $q_r(r|r')$ changes as well. As more samples of $\pi(\theta)$ are generated by Algorithm 4.2, $\tilde{p}(r)$ becomes closer to the reference Gaussian density $p(r)$. Thus, a small random walk proposal may be the most appropriate at early iterations, but a larger, perhaps position-independent, proposal may be advantageous after the map has captured more of the target density structure. By monitoring how well the map characterizes π (measured by the difference between $\tilde{p}$ and an IID Gaussian density), we can tune the reference proposal q_r, e.g., we can shift from a random walk proposal to an independent Gaussian.

A useful indicator of map accuracy is the variance σ_M defined by

$$\sigma_M = \mathop{\mathbb{V}\mathrm{ar}}_{\theta}\left[\log \pi(\theta) - \log p\left(\tilde{T}(\theta)\right) - \log\left|\det(\partial\tilde{T}(\theta))\right|\right]. \tag{4.11}$$

This value was extensively used in [79] to monitor map convergence. Notice that $\sigma_M = 0$ implies

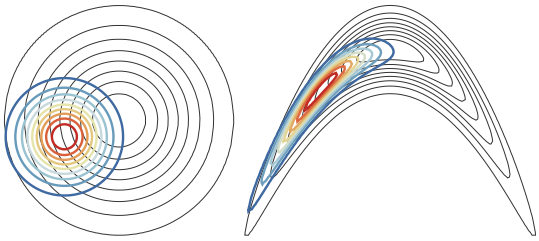
$$\frac{\pi(\theta)}{p\left(\tilde{T}(\theta)\right)\left|\det(\partial\tilde{T}(\theta))\right|} = 1$$

When this occurs, the map has captured all of the structure in $\pi(\theta)$ and $\tilde{p}(r) = p(r)$ is an IID Gaussian. We will use σ_M to define an adaptive mixture proposal discussed in the next section. While not discussed in this paper, we want to acknowledge that σ_M could be be used to tune other types of proposal as well, e.g., to adapt proposal variances or set weights in a Crank-Nicholson proposal [25].

4.2.4 Choice of reference proposal

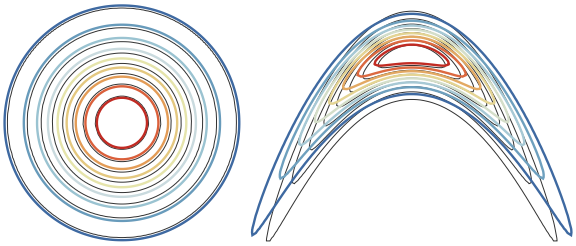
Until now, we have left the choice of reference proposal $q_r(r'|r)$ to the reader's imagination. Indeed, any non-adaptive choice of this proposal, including both independent proposals and random walk proposals, could be used within our framework. Figure 4-2 shows some typical proposals on both the reference space and the target space. In this section, we provide brief descriptions of a few reference proposals that we use

in our results. We acknowledge that this list is not exhaustive but feel that these algorithms shed sufficient light on the strengths and weaknesses of our algorithm.



(a) Local proposal $q_r(r'|r)$ on reference space.

(b) Local proposal mapped to target space.



(c) Independent proposal $q_r(r')$ on reference space.

(d) Independent proposal mapped to target space.

Figure 4-2: Proposal types in reference space and target space. The black lines show the target distributions (Gaussian on the r space and "banana" shaped on the θ space) while the colored contours illustrate the proposal densities. When the map is adequate, the local and independence proposals will have the same behavior as the target density.

Metropolis-Adjusted Langevin: The Langevin diffusion is a continuous time stochastic differential equation of the form

$$d\theta(t) = \nabla_\theta \log \pi(\theta(t))dt/2 + d\mathbf{b}(t),$$

where $d\mathbf{b}(t)$ is a D-dimensional Brownian motion term. If solved exactly, the solution to the Langevin diffusion is the target density $\pi(\theta)$ [87]. However, an exact solution is infeasible and a first order Euler discretization is often used as a proposal in the Metropolis-Hastings rule. With this discretization, the proposal density is given by

$$q_{MALA}(r'|r) = N\left(r + \frac{1}{2}\Sigma\nabla \log \tilde{p}(r), \Sigma\right), \tag{4.12}$$

for some covariance matrix Σ. In this work, we set the covariance to be a scaled identity: $\Sigma = \sigma^2_{MALA} I$. For general MALA use, better choices of Σ exist to capture the structure of π. However, in our case, $\tilde{T}$ will capture the problem structure. Note that to evaluate the drift term in this proposal, we need to evaluate $\nabla \log \tilde{p}(r)$,

which involves computing the gradient of the target density and following the steps in Section 8.

Delayed rejection: In [77], a proposal mechanism called delayed rejection (DR) was developed that allows several proposals to be tried during each MCMC step. This multiple-stage proposal allows us to try a large proposal at the first stage, followed by smaller proposals that are more likely to produce accepted moves. We use this feature to define $q_r(r'|r)$ in two ways.

Our first use of DR uses an IID Gaussian distribution with zero mean and unit variance as the first stage proposal. If a proposed point from this first stage is rejected, the second stage uses a small Gaussian random walk proposal. Our motivation for these global-then-local stages is based on the evolving nature of $\tilde{p}(r)$. After many MCMC steps, this density may become almost Gaussian, in which case the independent proposal in the first stage could generate nearly independent samples. On the other hand, we need many samples to build a good map and an independent Gaussian proposal will be inefficient during the early steps of the MCMC chain. This is where the random walk second stage comes into play. Even when the first stage of the proposal is rejected, the second stage will ensure our sampler continues to explore the target space. As $\tilde{T}(\theta)$ begins to capture the structure in $\pi(\theta)$, the first stage of the proposal will be accepted more often and the chain will mix better.

Our second use of DR involves two stages of symmetric random walk proposals. In the first stage, we use a larger proposal while in the second stage we again use a small proposal. Our motivation is the same as the before, the larger proposal should target a more "Gaussian looking" $\tilde{p}(r)$, while the smaller proposal will ensure the chain explores the target space even when $\tilde{T}$ does not capture any of the structure in $\pi(\theta)$.

Gaussian mixtures: An alternative to using the independent proposal within the delayed rejection framework is to use the same independent Gaussian proposal as one component in a Gaussian-mixture proposal. The other component, like delayed rejection, is a small random walk. A sample from the independent Gaussian is used with a probability w, and a sample from the random walk proposal is used with probability $(1 - w)$. Our key to making this method efficient is that we choose the weight w based on the map's performance, i.e., how well the the map captures the target density π. Let $\hat{\sigma}_M^{(k)}$ be a Monte Carlo approximation of σ_M at step k of the MCMC iteration. We choose the mixture weight based on the following simple function

$$w = \frac{w_{max}}{1 + w_{scale}\hat{\sigma}_M^{(k)}}, \tag{4.13}$$

where $w_{max} \in [0, 1]$ and $w_{scale} \in [0, \infty)$ are tunable parameters. Notice that $w \to w_{max}$ as $\hat{\sigma}_M^{(k)} \to 0$. When the chain is mixing well and $\hat{\sigma}_M^{(k)}$ is a good estimate of $\sigma_M^{(k)}$, this means that $w \to w_{max}$ as the map captures more of the structure in π. However, the Monte Carlo estimate of $\hat{\sigma}_M^{(k)}$ can also give an erroneous impression of map convergence when the chain is not mixing properly. We have practically overcome this issue by always choosing $w_{max} < 1$ (e.g., 0.9) and by setting N_U so that at least a few steps are likely to be accepted between map updates.

4.3 Relationship to geodesic MCMC

A class of differential geometric MCMC approaches was introduced by [39]. These differential geometric approaches use a position specific metric to define a Riemannian manifold on which to perform MCMC. At each point, the manifold captures some of the local correlation structure of the target distribution and allows the proposal to locally adapt to the target. In the original work, [39] used the expected Fisher information metric to define the manifold. However, as the authors are quick to point out, alternative metrics can also be used. In fact, the map in our approach can be used to define such a metric. Interestly, our numerical experiments also suggest that moving along geodesics on this map-induced manifold is equivalent to moving linearly in the reference space. Recall that geodesics are the shortest paths on a manifold.

To derive the map-induced metric, consider the approximate map $\tilde{T}(\theta)$. Now, let $J^{-1}(r) = \partial \tilde{T}^{-1}(r)$ be the Jacobian matrix of the inverse map evaluated at r and $J(\theta) = \partial \tilde{T}(\theta)$ be the Jacobian matrix of the forward map $\tilde{T}(\theta)$ evaluated at θ. A small change $\delta\theta$ from θ in the target space corresponds to a small change $\delta r = J(\theta)\delta\theta$ in the reference space. The inner product $\delta r^T \delta r$ can then be written in term of θ as

$$\delta r^T \delta r = \delta\theta^T \left(J(\theta)^T J(\theta) \right) \delta\theta. \tag{4.14}$$

This inner product defines a position-dependent metric defined by the matrix

$$G^M(\theta) = J(\theta)^T J(\theta). \tag{4.15}$$

Since $|\det J| \geq d_{min}$ by our monotonicty constraint, the metric G^M is guaranteed to be symmetric and positive definite. This metric defines a manifold on the target space much like the expected Fisher information metric used in [39].

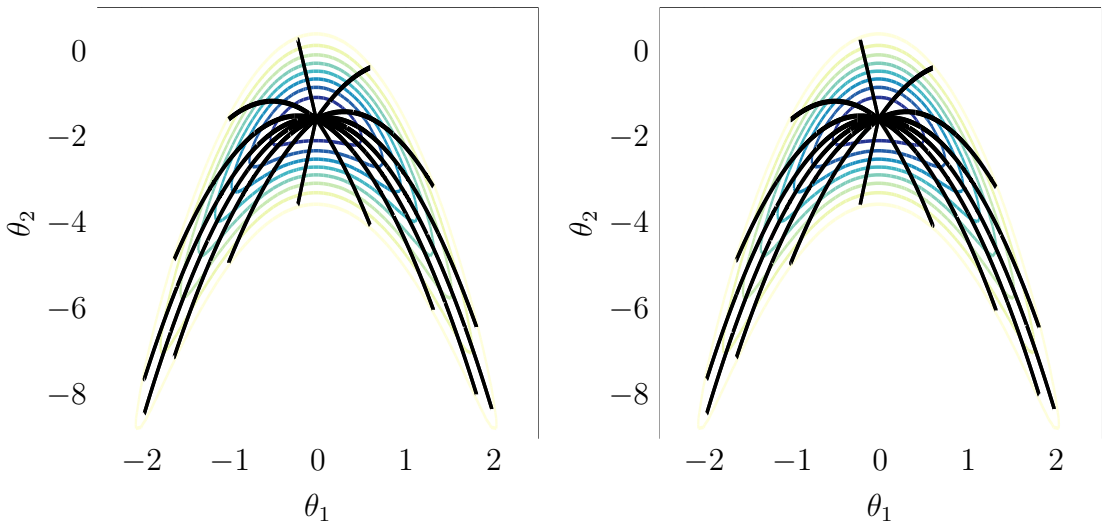
Not only does the map define a metric, but from our observations it seems that evaluating the map is equivalent to moving along a geodesic on the manifold. For illustration, consider a simple "banana" shaped density defined by the following quadratic transformation

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = T^{-1}(r) = \begin{bmatrix} r_1 \\ r_2 - 2(r_1^2 + 1) \end{bmatrix},$$

where r_1 and r_2 are (as usual) IID Gaussian random variables. Figure 4-3 shows a comparison of Geodesic paths computed using (4.15) and linear paths on the reference space pushed through the map. Let $w_g(t)$ be the geodesic path, parameterized by $t \in [0, 1]$, between $w(0)$ and $w(1)$ and $w_m(t)$ be the another path between the same two points defined by

$$w_m(t) = \tilde{T}^{-1}\left(\tilde{T}(w(0)) + t\left[\tilde{T}(w(1)) - \tilde{T}(w(0)) \right] \right). \tag{4.16}$$

Figure 4-3 compares $w_m(t)$ and $w_g(t)$ for 15 different values of $w(1)$ and $w(0) = [0, 0]^T$. In this simple example, the mapped paths from (4.16) and the geodesic paths look identical, which seems to imply that our transport maps are also time one geodesic flows on a Riemannian manifold defined by $G^M(\theta)$. A more theoretical analysis of



(a) Geodesic paths, $w_g(t)$, constructed on the manifold defined by $G^M(\theta)$ in (4.15).

(b) Map paths, $w_m(t)$, constructed using (4.16).

Figure 4-3: Comparison of 15 different geodesic paths and mapped paths. Paths are shown in black. The Geodesic paths were constructed by integrating a Hamiltonian system describing the geodesic. An initial momentum p was used and a leap frog integrator was used until some finishing time t_f. On the other hand, the map paths are evaluations of $\tilde{T}^{-1}(tp) \quad \forall t \in (0, t_f)$ for the same 15 initial momentums p used for the Geodesic plot. Colored contours of the target density $\pi(\theta)$ are shown in the background.

this observation could provide a rigorous connection between our map-based MCMC, geodesic MCMC, and Riemannian geometry.

4.4 Convergence analysis

Without some care, general adaptive proposal mechanisms can break the ergodicity of an MCMC chain. Thus, this section investigates conditions on our adaptive approach that ensure our algorithm yields an appropriate ergodic chain. We first build some intuition by analyzing the fixed-map algorithm and then proceed with a a high level description of our adaptive algorithm's convergence. A more detailed step-by-step convergence proof for the adaptive algorithm can be found in appendix A.

4.4.1 The need for bounded derivatives

Using the map-induced proposal in Algorithm 4.1 seems like a perfectly reasonable combination of transport maps and MCMC. But how does the transport map effect the convergence properties of the MCMC chain? To illustrate the importance of this question, consider a random walk proposal on the reference space $q_r(r'|r) = N(r, \sigma^2 I)$ with some pre-defined variance σ^2. Now, assume the target density is a

standard normal distribution: $\pi(\theta) = N(0, I)$. The RWM algorithm is known to be geometrically ergodic for any density satisfying the following two conditions (see theorem 4.3 of [55] for details)

$$\limsup_{\|\theta\|\to\infty} \frac{\theta}{\|\theta\|} \cdot \nabla \log \pi(\theta) = -\infty, \tag{4.17}$$

and

$$\lim_{\|\theta\|\to\infty} \frac{\theta}{\|\theta\|} \cdot \frac{\nabla \log \pi(\theta)}{\|\nabla \log \pi(\theta)\|} < 0. \tag{4.18}$$

The first condition implies the target density is super exponentially light. A little algebra easily shows that our example Gaussian density satisfies these conditions. However, in Algorithm 4.1, instead of applying the RWM proposal to π directly, we instead apply the RWM proposal to a map-induced density $\tilde{p}(r) = \pi(\tilde{T}^{-1}(r)) \left|\det(\partial \tilde{T}^{-1}(r))\right|$.

Without applying the derivative upper bound correction in (2.15), we can show that even when π is Gaussian, any nonlinear monotone *polynomial* map results in a density $\tilde{p}(r)$ that is no longer super exponentially light. For example, assume $\tilde{T}$ has a maximum odd polynomial order of $M > 1$. Then the following expression holds:

$$\begin{aligned} \limsup_{\|r\|\to\infty} \frac{r}{\|r\|} \cdot \nabla \log \tilde{p}(r) &= \limsup_{\|r\|\to\infty} \frac{1}{\|r\|} \sum_{d=1}^{D} r_d \left(\frac{\partial \tilde{T}^{-1}}{\partial r_d}\right)^{-1} \frac{\partial^2 \tilde{T}^{-1}}{\partial r_d^2} & (4.19) \\ &= \limsup_{\|r\|\to\infty} \frac{D}{\|r\|} \left(\frac{1}{M} - 1\right) = 0 & (4.20) \end{aligned}$$

Clearly, the map-induced density is not super-exponentially light. We have therefore jeopardized the geometric ergodicity of our sampler on a simple Gaussian target! We obviously need additional restrictions on the map to ensure we retain convergence properties. The fact that polynomial maps have unbounded derivatives and thus do not satisfy (2.13) is the problem that led to our loss of geometric ergodicity in (4.20). The unbounded derivatives of $\tilde{T}$ imply that $\tilde{T}^{-1}$ has zero derivatives as $\|r\| \to \infty$, which leads to (4.20). More intuitively, without the upper bound, polynomial maps move too much weight to the tails of $\tilde{p}$. In Section 4.4 we will show that even when the map is adapted as the MCMC chain progresses, the upper and lower derivative bounds ensure the ergodicity of both Algorithm 4.1 and the upcoming adaptive approach.

4.4.2 Convergence of adaptive algorithm

Our goal in this section is to show that the adaptive Algorithm 4.2 will produce samples that can be used in the Monte Carlo approximation of our quantity of interest as in (4.2). To have this property, we need to show that Algorithm 4.2 is ergodic for the target density $\pi(\theta)$.

For the upcoming analysis, we assume that the target density is finite, continuous, and is super exponentially light. Note that some densities which are not super-

exponentially light can be transformed to super-exponentially light densities using the techniques from [58]. We also assume the reference proposal $q_r(r'|r)$ is a Gaussian density with bounded mean (note that RWM proposals and truncated MALA proposals fall into this category). Furthermore, we define Γ as the space of the map parameters γ such that $\tilde{T}_\gamma$ satisfies the bi-Lipschitz condition given by (2.12) and (2.13).

The map at iteration k of the MCMC chain is defined by the coefficients $\gamma^{(k)}$. Let $P_{\gamma^{(k)}}$ be the transition kernel of the chain at iteration k which is constructed from the map $\tilde{T}(\theta;\gamma^{(k)})$, the target proposal in (4.4), and the Metropolis-Hastings kernel given by

$$P_{MH}(\theta,\mathcal{A}) = \int_{\mathcal{A}} \alpha(\theta',\theta)q(\theta'|\theta) + (1-r(\theta))\,\delta_\theta(\theta')d\theta', \tag{4.21}$$

where $r(\theta) = \int \alpha(\theta',\theta)q(\theta'|\theta)d\theta'$. Now, following [85] and [11] we can show the ergodicity of our adaptive algorithm by showing diminishing adaptation and containment. Diminishing adaptation is defined by

Definition 1 (Diminishing adaptation)**.** *Diminishing adaptation is the property that for any starting point $x^{(0)}$, and initial set of map parameters $\gamma^{(0)}$, the following holds:*

$$\lim_{k\to\infty} \sup_{x\in\mathbb{R}^D} \left\| P_{\gamma^{(k)}}(x,\cdot) - P_{\gamma^{(k+1)}}(x,\cdot) \right\|_{TV} = 0 \tag{4.22}$$

where $\|\cdot\|_{TV}$ is the total variation norm.

Furthermore, by theorem 3 of [85], our adaptive MCMC algorithm will satisfy containment if it satisfies the simultaneous strongly aperiodic geometric ergodicity (SSAGE) condition given below by

Definition 2 (SSAGE)**.** *Simultaneous strongly aperiodic geometric ergodicity (SSAGE) is the condition that there is a measurable set $C \in \mathcal{B}(\mathbb{R}^D)$, a drift function $V : \mathbb{R}^D \to [1,\infty)$, and three scalars $\delta > 0$, $\lambda < 1$, and $b < \infty$ such that $\sup_{x\in C} V(x) < \infty$ and the following two conditions hold:*

1. *(Minorization) For each vector of map parameters $\gamma \in \Gamma$, there is a probability measure $\nu_\gamma(\cdot)$ defined on $C \subset \mathbb{R}^D$ with $P_\gamma(x,\cdot) \geq \delta\nu_\gamma(\cdot)$ for all $x \in C$.*
2. *(Simultaneous drift) $\int_{\mathbb{R}^D} V(x)P_\gamma(x,dx) \leq \lambda V(x) + bI_C(x)$ for all $\gamma \in \Gamma$ and $x \in \mathbb{R}^D$.*

The following three lemmas will establish diminishing adaptation and SSAGE. For the following, define $C = B(0,R_C)$ as a ball with some radius $R_C > 0$ and let $V(x) = k_v\pi^{-\alpha}(x)$ for some $\alpha \in (0,1)$ and $k_v = \sup_x \pi^\alpha(x)$. Also, assume $\pi(x) > 0$ for all $x \in C$. For this choice of $V(x)$ and our assumption that $\pi(x) > 0$ for $x \in C$, we have that $\sup_{x\in C} V(x) < \infty$.

Because the reference proposal is a Gaussian with bounded mean, we can find two scalars k_1 and k_2 as well as two zero mean Gaussian densities g_1 and g_2, such that the reference proposal is bounded like

$$k_1 g_1(r'-r) \leq q_r(r'|r) \leq k_2 g_2(r'-r). \tag{4.23}$$

Furthermore, the bounds on the norms in (2.12) and (2.13) imply that the target density can also bounded as

$$k_L g_L(\theta' - \theta) \leq q_\theta(\theta'|\theta) \leq k_U g_U(\theta' - \theta), \tag{4.24}$$

where $k_L = k_1 d_{min}^D$, $k_U = k_2 d_{max}^D$, $g_L(x) = g_1(d_{max}x)$, and $g_U(x) = g_2(d_{min}x)$. These upper and lower bounds on the proposal density are key to the proofs below. In fact, with these bounds, the proofs of lemma 2 and lemma 3 given below, are nearly identical to the proofs of proposition 2.1 in [9].

Lemma 1 (Diminishing adaptation)**.** *Assume the map parameters γ are restricted to a compact space Γ. Then, the sequence of transition kernels defined by the update step in lines 10-14 of Algorithm 4.2 satisfies the diminishing adaptation condition.*

Lemma 1. When the MCMC chain is not at an adapt step, $\gamma^{(k+1)} = \gamma^{(k)}$. Thus, to show diminishing adaptation, we need to show that

$$\lim_{k\to\infty} \sup_{x\in\mathbb{R}^D} \left\| P_{\gamma^{(k)}}(x,\cdot) - P_{\gamma^{(k+N_U)}}(x,\cdot) \right\|_{TV} = 0$$

Because the map is continuous in γ (consider (2.23)), diminishing adaptation is equivalent to

$$\lim_{k\to\infty} \left\| \gamma^{(k)} - \gamma^{(k+N_U)} \right\| = 0$$

Recall that $\gamma^{(k)}$ is the minimizer of (4.10), which is based on a Monte Carlo approximation to KL divergence. As the number of samples grows, this Monte Carlo expectation will converge to the KL divergence. Moreover, by proposition 2.2 of [48], the minimizer of the KL divergence, γ, will also converge. Thus, as the number of samples goes to infinity, $\left\| \gamma^{(k)} - \gamma^{(k+N_U)} \right\|$ goes to 0 and the diminishing adaptation condition is satisfied. □

Lemma 2 (Minorization)**.** *There is a scalar δ and a set of probability measures ν_γ defined on C such that $P_\gamma(x,\cdot) \geq \delta\nu_\gamma(\cdot)$ for all $x \in C$ and $\gamma \in \Gamma$.*

Lemma 2. Let τ be the minimum acceptance rate over all $x, y \in C$ and over all map-induced proposals defined by

$$\tau = \inf_\gamma \inf_{x,y\in C} \min\left\{1, \frac{\pi(y)q_{\theta,\gamma}(x|y)}{\pi(x)q_{\theta,\gamma}(y|x)}\right\}.$$

Notice that $\tau > 0$ because

$$\pi(y)q_{\theta,\gamma}(x|y) \geq \pi(y)k_L g_L(x-y) > 0 \quad \forall x, y \in C.$$

Now, using the Metropolis-Hastings kernel in (4.21) and the lower bound in (4.24), we have

$$P_\gamma(x, dy) \geq \tau k_L g_L(x-y)dy.$$

Define the new probability density $g_{L2}(y) = \frac{\inf_{x\in C} g_L(x-y)}{\int_{\mathbb{R}^D} \inf_{x\in C} g_L(x-y)dy}$. Because g_L is a Gaussian density and thus nonzero over C, $g_{L2}(y)$ is a valid probability density.

Define the scalar $k_{L2} = k_L \int_{\mathbb{R}^D} \inf_{x \in C} g_L(x - y)dy$ and set $\delta = \tau k_{L2}$. Notice that $P_\gamma(x, dy) \geq \tau k_{L2} g_L(y)dy$. Now define the measure

$$\nu_\gamma(A) = \nu(A) = \frac{\int_{A \cap C} g_{L2}(z)dz}{\int_C g_{L2}(z)dz}$$

This is a nontrivial probability measure defined over C. Furthermore, we have $P_\gamma(x, \cdot) \geq \delta\nu(\cdot)$ and the minorization condition is satisfied. □

Lemma 3 (Drift)**.** *For all points $x \in \mathbb{R}^D$ and all feasible map parameters $\gamma \in \Gamma$, there are scalars λ and b such that $\int_{\mathbb{R}^D} V(x)P_\gamma(x, dx) \leq \lambda V(x) + bI_C(x)$*

Lemma 3. Using the bounds in (4.24), we can follow the proof of Lemma 6.2 in [9] to show that the following two conditions hold:

$$\sup_{x \in \mathbb{R}^D} \sup_{\gamma \in \Gamma} \frac{\int_{\mathbb{R}^D} V(y)P_\gamma(x, dy)}{V(x)} < \infty$$

and

$$\limsup_{\|x\| \to \infty} \sup_{\gamma \in \Gamma} \frac{\int_{\mathbb{R}^D} V(y)P_\gamma(x, dy)}{V(x)} < 1.$$

Appendix A verifies these conditions in more detail. With these two conditions in hand, lemma 3.5 of [55] gives the existence of λ and b needed to satisfy the drift condition. □

Theorem 1 (Ergodicity of adaptive map MCMC)**.** *A theoretical version of Algorithm 4.2, where γ lies in a compact space and $\tilde{T}_\gamma$ is guaranteed to satisfy (2.12) and (2.13) everywhere, is ergodic for the target distribution $\pi(\theta)$.*

Theorem 1. Lemmas 2 and 3 ensure that SSAGE is satisfied, which subsequently ensures containment. The diminishing adaptation from lemma 1 combined with SSAGE implies ergodicity by theorem 3 of [85]. □

4.5 Numerical examples

To illustrate the effectiveness of our adaptive Algorithm 4.2, we will compare the algorithm against several existing MCMC methods including DRAM [41] , simplified Manifold MALA [39], adaptive truncated MALA [9], and the No-U-Turn Sampler [47]. Table 4.1 provides a summary of these algorithms and the acronyms we will use in the results. Notice also that Algorithm 4.2 defines an adaptive framework that is not restricted to a particular reference proposal q_r. Thus, for a full comparison, we will include several reference proposal mechanisms including a basic random walk, delayed rejection, a Gaussian mixture proposal, and a MALA proposal. These proposals are summarized in Table 4.2. As all algorithms have their strengths and weaknesses, we will illustrate our approach on three examples with varying characteristics.

Table 4.1: Summary of standard MCMC samplers used in the results. The table shows the acronym used for the method in this paper, the name of the algorithm, if the method requires derivative information, and if the method is adaptive.

Acronym	Method	$\frac{\partial \pi}{\partial \theta}$?	Adapts?
DRAM	Delayed rejection adaptive Metropolis [41]	No	Yes
sMMALA	Simplified Manifold MALA [39]	Yes	No
AMALA	Adaptive MALA [9]	Yes	Yes
NUTS	No-U-Turn Sampler [47]	Yes	Yes

Table 4.2: Summary of map-accelerated MCMC samplers used in the results. The table shows the acronym used for the method in this paper, a brief description of the reference proposal $q_r(r'|r)$, and if the method requires derivative information. See Section 4.2.4 for more details on each proposal mechanism.

Acronym	Reference proposal	$\frac{\partial \pi}{\partial \theta}$?
TM+RW	Isotropic Gaussian random walk.	No
TM+DRG	Delayed rejection. First stage is independence proposal and second stage is random walk..	No
TM+DRL	Delayed rejection. First stage is random walk with large variance and second stage is random walk with small variance.	No
TM+MIX	Mixture of independence proposal and random walk. Weights are controlled by (4.13).	No
TM+LA	Metropolis-Adjusted Langevin (MALA)	Yes

The first test problem is a 25 parameter logistic regression using the German credit dataset. This problem is also used as an example in [39] and [47]. The second test is sampling a Bayesian posterior for a simple two-parameter Biochemical Oxygen Demand model. Finally, the third example is sampling a posterior for an 8 dimensional predator-prey dynamical system. The following sub-sections give detailed descriptions of these problems and performance comparisons on each problem.

An effective MCMC algorithm minimizes the correlation between states in the MCMC chain. Thus, a standard measure for MCMC performance is the integrated autocorrelation time of the chain, see [107] for details on accurately computing this quantity. A shorter autocorrelation time implies the states in the chain are less correlated and thus that the MCMC sampler is more effective. Let τ_d be the integrated autocorrelation time for dimension d. Using the autocorrelation, another commonly reported statistic of the MCMC chain is the effective sample size, which is given by

$$\text{ESS}_d = \frac{N}{1 + 2\tau_d}, \tag{4.25}$$

where N is the number of steps in the chain after a burn-in period. The ESS represents the number of effectively independent samples produced by the chain. Our results report the minimum ESS_d, which corresponds to the maximum τ_d over the dimensions, and is an indication of the "worst case" performance.

4.5.1 German credit logistic regression

Consider a binary response variable $t \in \{0, 1\}$ that depends on $D - 1$ predictor variables $\{x_1, x_2, ..., x_{D-1}\}$. A simple model of the binary response is provided by the logistic function. In this model, the probability that $t = 1$ is given by the following parameterized model

$$\mathbb{P}(t = 1|\theta) = \frac{1}{1 + \exp\left[-\left(\theta_1 + \sum_{i=2}^{D} \theta_i x_{i-1}\right)\right]}, \tag{4.26}$$

where $\theta = \{\theta_1, \theta_2, ..., \theta_D\}$ are model parameters. As always, $\mathbb{P}(t = 0|\theta) = 1 - \mathbb{P}(t = 1|\theta)$. The goal of this logistic regression problem is to infer the parameters given N joint observations of the predictor variable x and the response variable t. Following [39] and [47], a simple Gaussian prior is employed here, $\pi(\theta) = N(0, aI)$ with $a = 100$. The observations of x and t come from the German credit dataset, which is available from the UCI database [10]. In this dataset, there are 24 predictor variables and 1000 observations. Thus, the inference problem has 25 dimensions. All 24 predictor variables were normalized to have zero mean and unit variance.

For this problem, each sampler was run for 75,000 steps, of which 5,000 were treated as burn-in. All algorithms were extensively tuned to minimize τ_{max}. Moreover, each MCMC algorithm was independently run 30 times starting from the MAP point. The 30 independent replicates were used in the MATLAB code distributed by [107] to estimate the integration autocorrelation time, as well as the variance in

Table 4.3: Performance of MCMC samplers on German credit logistic regression problem. For this 25 dimensional problem, the maximum correlation time, $\tau_{\max}$, and corresponding minimum effective sample size, ESS, is displayed in this table. The number of gradient evaluations and density evaluations were combined to evaluated ESS/eval.

Method	$\tau_{\max}$	σ_τ	ESS	ESS/sec	ESS/eval
DRAM	43.6	5.005	803	21.6	0.0065
sMMALA	4.9	0.222	7121	4.0	0.0474
AMALA	8.2	0.468	4244	153.6	0.0282
NUTS	1.0	0.023	34008	181.6	0.0311
TM+DRG	1.7	0.047	21108	76.3	0.2058
TM+DRL	47.9	5.718	731	2.4	0.0053
TM+RW	56.2	7.160	623	2.3	0.0083
TM+MIX	5.3	0.247	6634	17.7	0.0882
TM+LA	3.3	0.125	10602	8.9	0.0706

the estimate of this time. The maximum integrated autocorrelation times and corresponding minimum effective sample sizes are shown in Table 4.3. In the table, σ_τ is the variance of the $\tau_{\max}$ estimator. The NUTS and sMMALA results for this dataset match those given by [39] and [47] respectively, indicating that we have properly tuned those algorithms. We would like to point out that all of the comparisons were performed using efficient implementations from the MUQ c++ library[81]. In the case of NUTS, MUQ links to the STAN library [96].

In this problem, the posterior is very close to Gaussian, and we found it sufficient to use a *linear* map in the transport map proposal process, meaning that all proposals are Gaussian. However, the results in Table 4.3 still show the top transport map proposal (TM+DRG) to have a nearly identical autocorrelation time to the top standard proposal (NUTS), even though TM+DRG does not use any derivative information.

Recall that the autocorrelation (labeled $\tau_{\max}$ in Table 4.3 measures the correlation between states in the MCMC chain. A lower autocorrelation indicates there is more information in the chain, which then implies that any Monte-Carlo approximations based on the MCMC result will be more accurate. The effective sample size (ESS) is inversely proportional to the autocorrelation and measures the number of "effectively independent" samples in the MCMC chain. Larger ESS values imply there is more information in the chain.

From Table 4.3, we see that NUTS yields a higher raw ESS, but when the ESS is normalized by the number of density evaluations, the ESS/eval of TM+DRG is much higher. This is because TM+DRG requires at most 2 density evaluations per MCMC step, while NUTS requires many more evaluations. In fact, we have an ESS per evaluation that is almost an order of magnitude larger than NUTS. This is in large part due to the independent first stage of our TM+DRG proposal.

Also, notice that the performance of TM+DRL is essentially the same as DRAM, and the performance of TM+LA is very similar to AMALA. In this case, learning

Table 4.4: Performance of MCMC samplers on the BOD problem. Like the other examples, the maximum correlation time, $\tau_{\max}$, and corresponding minimum effective sample size, ESS, is displayed in this table.

Method	$\tau_{\max}$	σ_τ	ESS	ESS/sec	ESS/eval
DRAM	46.1	5.413	759	127.3	0.0058
sMMALA	83.5	12.514	419	1.1	0.0028
AMALA	35.1	3.682	997	209.0	0.0066
NUTS	13.9	0.984	2517	57.0	0.0014
TM+DRG	2.3	0.073	15397	1467.7	0.1614
TM+DRL	5.0	0.230	6957	487.3	0.0570
TM+RW	4.9	0.221	7157	882.4	0.0953
TM+MIX	2.6	0.090	13422	1495.4	0.1786
TM+LA	793.1	271.538	44	3.3	0.0003

the fully linear map is equivalent to using the sample covariance in the proposal. Hence, TM+DRL boils down to DRAM, and TM+LA is equivalent to AMALA. Fundamentally though, by learning the map instead of the sample covariance, we can apply the same global independent proposals to more difficult problems exhibiting more non-Gaussian behavior.

4.5.2 Biochemical oxygen demand model

In water quality monitoring, the biochemical oxygen demand (BOD) test is often used to investigate the consumption of dissolved oxygen in a water column. See [99] for an example. To learn about the asymptotic behavior of the biochemical oxygen demand, the simple exponential model $B(t) = \theta_0(1-\exp(-\theta_1 t))$ is often fit to observations of $B(t)$ at early times ($t \leq 5$). Assume we have N observations available, $\{B(t_1), B(t_2), ..., B(t_N)\}$. From these observations, we form an inference problem for the model coefficients θ_1 and θ_2. In our example, we use 20 observations evenly spread over $[1, 5]$ with the following additive error model $B(t_i) = \theta_0(1-\exp(-\theta_1 t_i))+e$ where $e \sim N(0, \sigma_B^2)$ with $\sigma_B^2 = 2e - 4$.

Our synthetic "data," denoted by $\{B_d(t_1), B_d(t_2), ..., B_d(t_{20})\}$, comes from sampling e and evaluating $B(t_i)$ with $\theta_0 = 1$ and $\theta_1 = 0.1$. Using a uniform prior (over $\mathbb{R}^2$), we have the target density given by

$$\log \pi(\theta_0, \theta_1) = -2\pi\sigma_B^2 - 0.5 \sum [\theta_0(1 - \exp(-\theta_1 t_i)) - B_d(t_i)]^2 \tag{4.27}$$

Like the logistic regression posterior, it is easy to obtain any derivative information about this density, which allows us to again compare many different sampling approaches – both derivative free and derivative based.

Each chain was run for 75,000 steps with 5,000 burn in steps. The methods were also independently run 30 times starting at the MAP point to generate the results in Table 4.4. We also show typical trace plots and autocorrelation plots in Figures 4-5 and 4-6.

While the logistic regression posterior was nearly Gaussian and a linear map sufficed, a third order polynomial map was used in this example. The additional nonlinear terms help the map capture the changing posterior correlation structure shown in Figure 4-4(a). The narrow curving posterior shown in 4-4(a) is incredibly difficult for standard samplers to explore. Methods like DRAM and AMALA may capture the global covariance structure, but that is not enough to efficiently sample the posterior. Other methods like sMMALA and NUTS use derivative information to capture more local structure, but do not have a representation of the global correlation structure. This prevents those methods from efficiently taking very large jumps through the parameter space. Our transport map proposals on the other hand, use a polynomial map to characterize the global correlation structure and are capable of capturing the non-Gaussian structure shown in Figure 4-4(a).

The map is actually capable of transforming the very narrow BOD posterior into the easily sampled density shown in Figure 4-4(b). This allows methods with global independence proposals (TM+DRG and TM+MIX) to efficiently "jump" across the entire parameter space, yielding the much shorter integrated autocorrelation times shown in 4.4. Specifically, in terms of ESS per evaluation, the best transport map method (TM+DRG) is about 30 times more efficient than the best standard approach (DRAM).

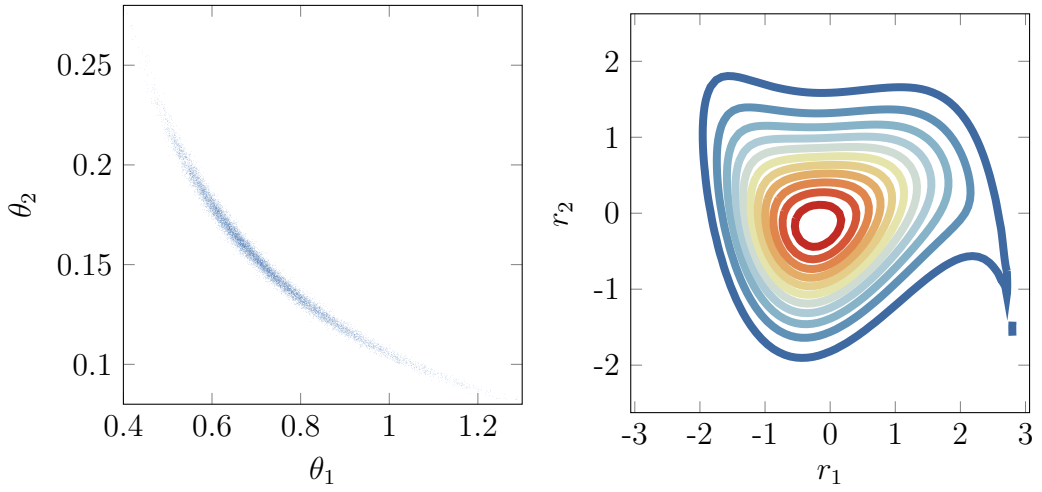
Another interesting result in Table 4.4 is the incorrigible performance of TM+LA. In this example, with its tight changing correlation, the basic MALA algorithm was not able to sufficiently explore the space on its own. This prevented the adaptive scheme from constructing an adequate map and resulted in the poor performance shown here. Consider again to the chicken-and-egg problem – where we need to samples to construct the map, but need the map to efficiently generate samples. From the TM+LA results, it should be clear that we need to have a pretty good representation of an egg before we can produce a chicken.

4.5.3 Predator prey system

The previous two examples have posteriors with analytic derivatives that are easy to evaluate. However, many realistic inference problems may be based on complicated forward models where derivative information is expensive to compute or even unavailable. This example takes a step in that direction with an ODE model of a predator prey system. The parameters in this system are given by the initial populations and 6 other scalar parameters governing the system dynamics, $\theta = \{P(0), Q(0), r, K, s, a, u, v\}$. These coefficients are part of the following ODE model

$$\begin{aligned} \frac{dP}{dt} &= rP\left(1 - \frac{P}{K}\right) - s\frac{PQ}{a+P} \\ \frac{dQ}{dt} &= u\frac{PQ}{a+P} - vQ \end{aligned}$$

In this model, r represents the prey growth rate, K is the prey carrying capacity, s is the predation rate, a represents how much food the predator can pro-



(a) Scatter plot of BOD posterior samples.

(b) Contour of map induced density and reference space $\tilde{p}(r)$. The tear drop feature near $r_1 = 3$ is a result of the third order polynomial map.

Figure 4-4: The narrow high density region and changing correlation structure on the left is difficult for existing adaptive strategies to capture. However, at the end of the adaptive MCMC run, the reference proposal is effective sampling the much less correlated density on the right.

cess (i.e., eat and digest), u is the predator growth rate, and v is the predator death rate. See [90] for details. The inference problem is to infer the 8 model parameters given 5 noisy observations of the prey and predator populations, $d = \{P_d(t_1), P_d(t_2), ..., P_d(t_5), Q_d(t_1), Q_d(t_2), ..., Q_d(t_5)\}$, where $\{t_1, t_2, ..., t_5\}$ are regularly space on $[0, 1]$ and $P_d(t)$, $Q_d(t)$ are solutions of (4.28) along with some additive Gaussian noise defined by

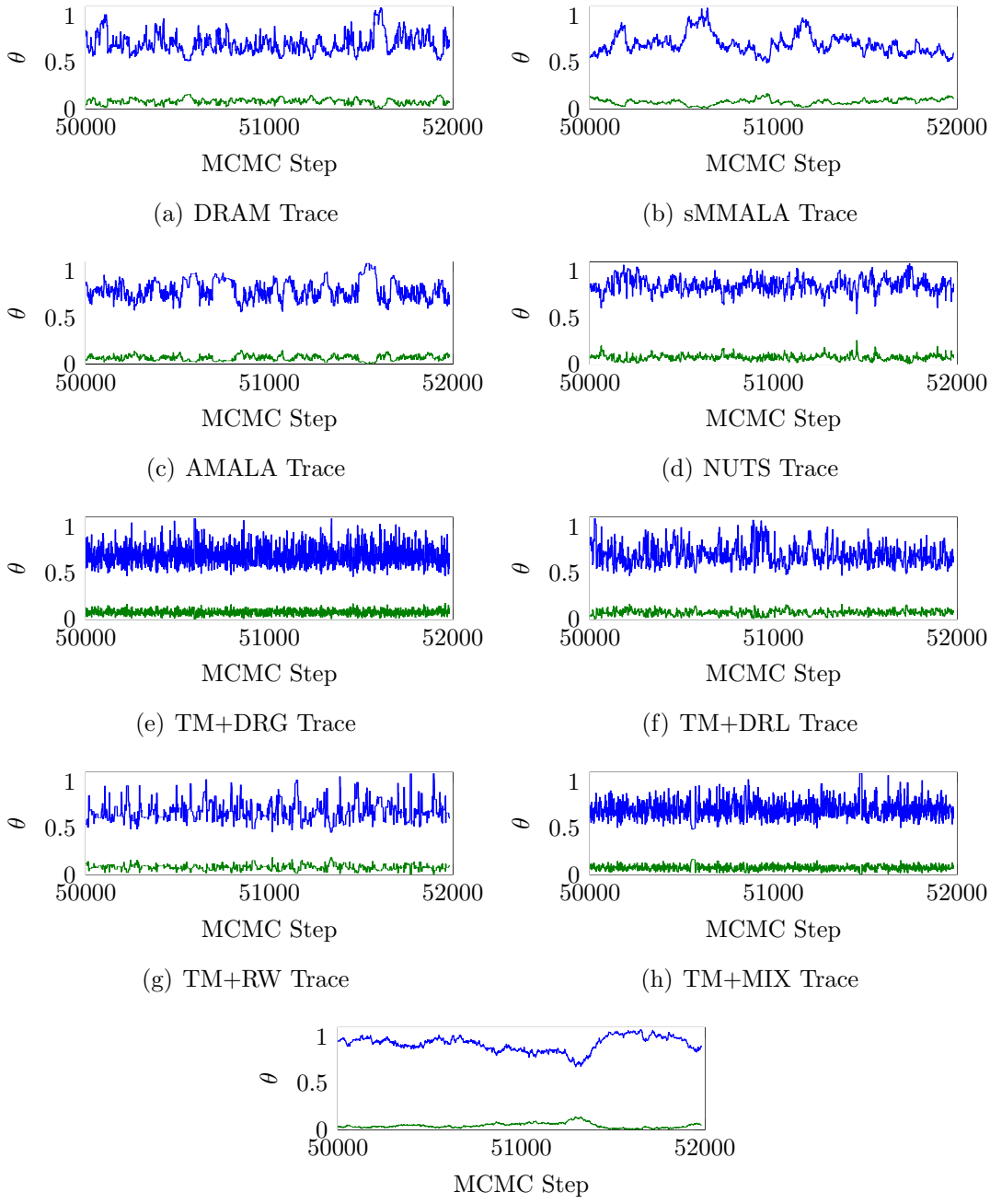
$$P_d(t_i) = Q(t_i) + e_i, \tag{4.28}$$

where $e_i \sim N(0, 10)$. This error model yields a likelihood function $\pi(d|\theta)$. We generated the data using the "true" parameters, given by

$$\begin{aligned} \theta_{true} &= [P(0), Q(0), r, K, s, a, u, v]^T \\ &= [50, 5, 0.6, 100, 1.2, 25, 0.5, 0.3]^T . \end{aligned} \tag{4.29}$$

The prior for this problem is uniform over parameter combinations that yield stable cyclic solutions. This feature can be recast to conditions on the parameters by looking at the fixed points of (4.28) and the Jacobian of (4.28). The fixed point, denoted by $[P_f, Q_f]$, must satisfy $P_f > 0$ and $Q_f > 0$. Also, to ensure the solution is cyclic, the Jacobian of the right hand side of (4.28) must have eigenvalues with positive real components when evaluated at $[P_f, Q_f]$ [98].

Let $\lambda_1(\theta)$ and $\lambda_2(\theta)$ be the eigenvalues of Jacobian matrix at the fixed point. The



(a) DRAM Trace

(b) sMMALA Trace

(c) AMALA Trace

(d) NUTS Trace

(e) TM+DRG Trace

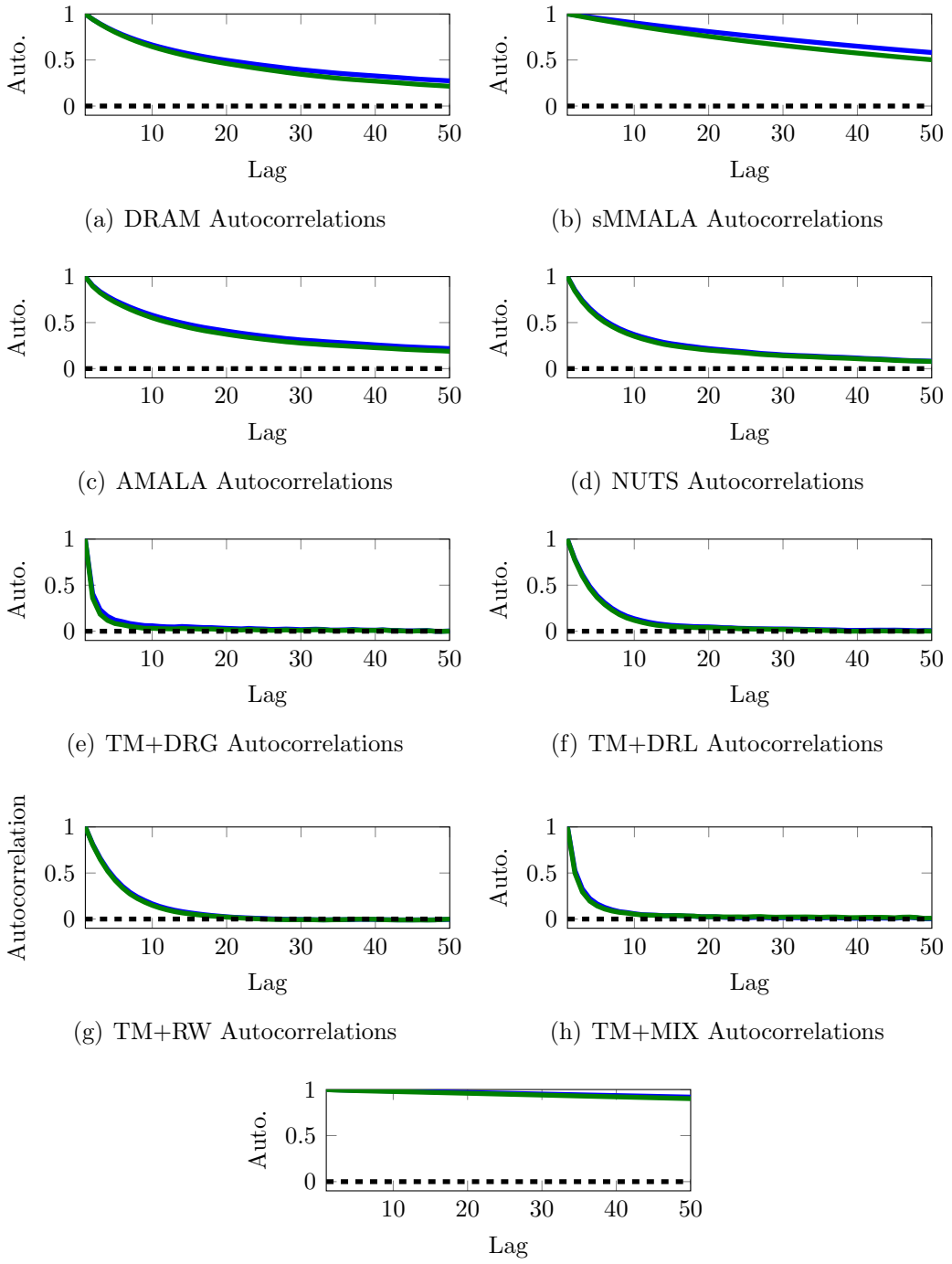
(f) TM+DRL Trace

(g) TM+RW Trace

(h) TM+MIX Trace

(i) TM+LA Trace

Figure 4-5: Typical traces of the MCMC chains on the two dimensional BOD problem. The blue lines show the values of θ_1 and the green lines show the values of θ_2. Trace plots that look more like white noise indicate less correlation in the MCMC chain and that the chain is mixing better. Notice that transport map accelerated methods, especially TM+DRG and TM+MIX, have better mixing than the best standard method (NUTS).



(a) DRAM Autocorrelations

(b) sMMALA Autocorrelations

(c) AMALA Autocorrelations

(d) NUTS Autocorrelations

(e) TM+DRG Autocorrelations

(f) TM+DRL Autocorrelations

(g) TM+RW Autocorrelations

(h) TM+MIX Autocorrelations

(i) TM+LA Autocorrelations

Figure 4-6: Autocorrelation plots for each MCMC method. The blue line shows the autocorrelation of the MCMC chain in the θ_1 component while the green line is the autocorrelation in the θ_2 component of the chain. Faster decay indicates that the chain "forgets" its previous states more quickly and has better mixing. Like Figure 4-5, better mixing can be found in the transport map accelerated methods, especially TM+DRG and TM+MIX, where the autocorrelation decays much faster than the best standard method (NUTS).

Table 4.5: Performance of MCMC samplers on predator-prey parameter inference problem. For this 8 dimensional problem, the maximum correlation time and corresponding minimum effective sample size is displayed provided.

Method	$\tau_{\max}$	σ_τ	ess	ess/sec	ess/eval
DRAM	1057.1	394.4	33	4.3e-01	1.8e-04
TM+DRG	12.4	0.8	2815	5.2e+00	2.7e-02
TM+DRL	145.2	27.1	241	4.7e-01	1.6e-03
TM+RWM	54.4	6.8	644	1.3e+00	7.3e-03
TM+MIX	17.6	1.4	1992	3.8e+00	2.3e-02

prior over θ for this problem is then given by:

$$\pi(\theta) \propto \begin{cases} 1 & P_f(\theta) > 0 \wedge Q_f(\theta) > 0 \wedge \mathrm{Re}(\lambda_1) > 0 \wedge \mathrm{Re}(\lambda_2) > 0 \\ 0 & \text{otherwise} \end{cases}$$

Combining the prior and likelihood yields the usual form of the posterior density $\pi(\theta|d) \propto \pi(d|\theta)\pi(\theta)$.

Our goal in this example is to sample the posterior, which is plotted in Figure 4-7. Notice that the posterior, while not as narrow as the BOD posterior, is non-Gaussian and various marginal distributions have the changing correlation structure that is difficult for standard methods to capture. Also, posterior evaluations require integrating the ODE in (4.28), making the posterior evaluations more expensive to evaluate than the German credit and BOD examples. Moreover, we treat the ODE integration here as a black box and do not have *efficient* access to the derivative information needed by NUTS, AMALA, and sMMALA. A performance comparison of the derivative-free samplers can be found in Table 4.5. Figure 4-8 also contains typical trace plots and autocorrelation plots.

The results are based on MCMC chains with 120,000 steps including 50,000 burn in steps. The transport map algorithms used multivariate polynomials with total order 3. The algorithms were started at the MAP point and 30 repetitions were used to generate the summary in Table 4.5.

As in the previous examples, the map-accelerated approaches that utilize some form of independence proposal have a dramatically shorter integrated autocorrelation time. In terms of raw ESS, TM+DRG is about 85 times more efficient than DRAM and in terms of ESS/eval, TM+DRG is about 150 times more efficient than DRAM. This means that more independent first stage proposals are accepted in TM+DRG than local first stage proposals in DRAM. Even when normalized by run time, the ESS/sec of TM+DRG is still over 10 times the level of DRAM. For readers with sampling problems even more computationally expensive than this one, ESS/eval is the result to consider. In this comparison, DRAM would require 150 times more evaluations to produce the same ESS as TM+DRG. For a computationally expensive sampling problem, especially one that exhibits a changing correlation structure, this result shows that our approach could reduce required MCMC runtime by days.

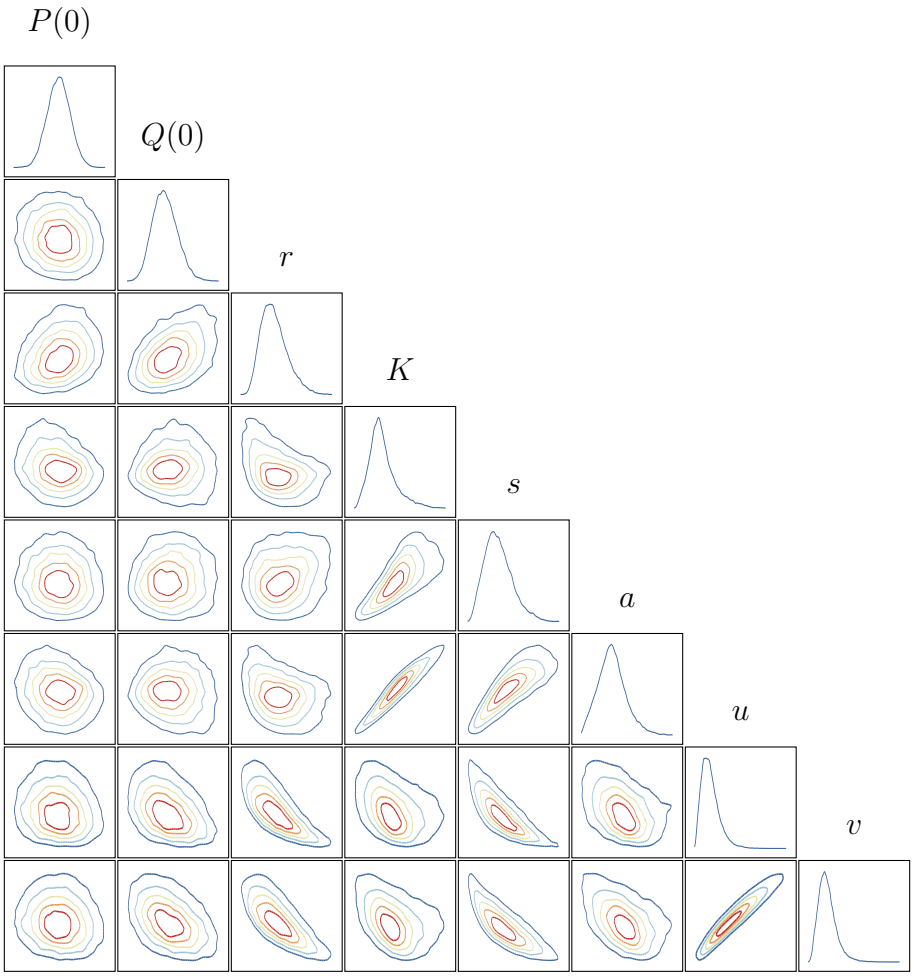
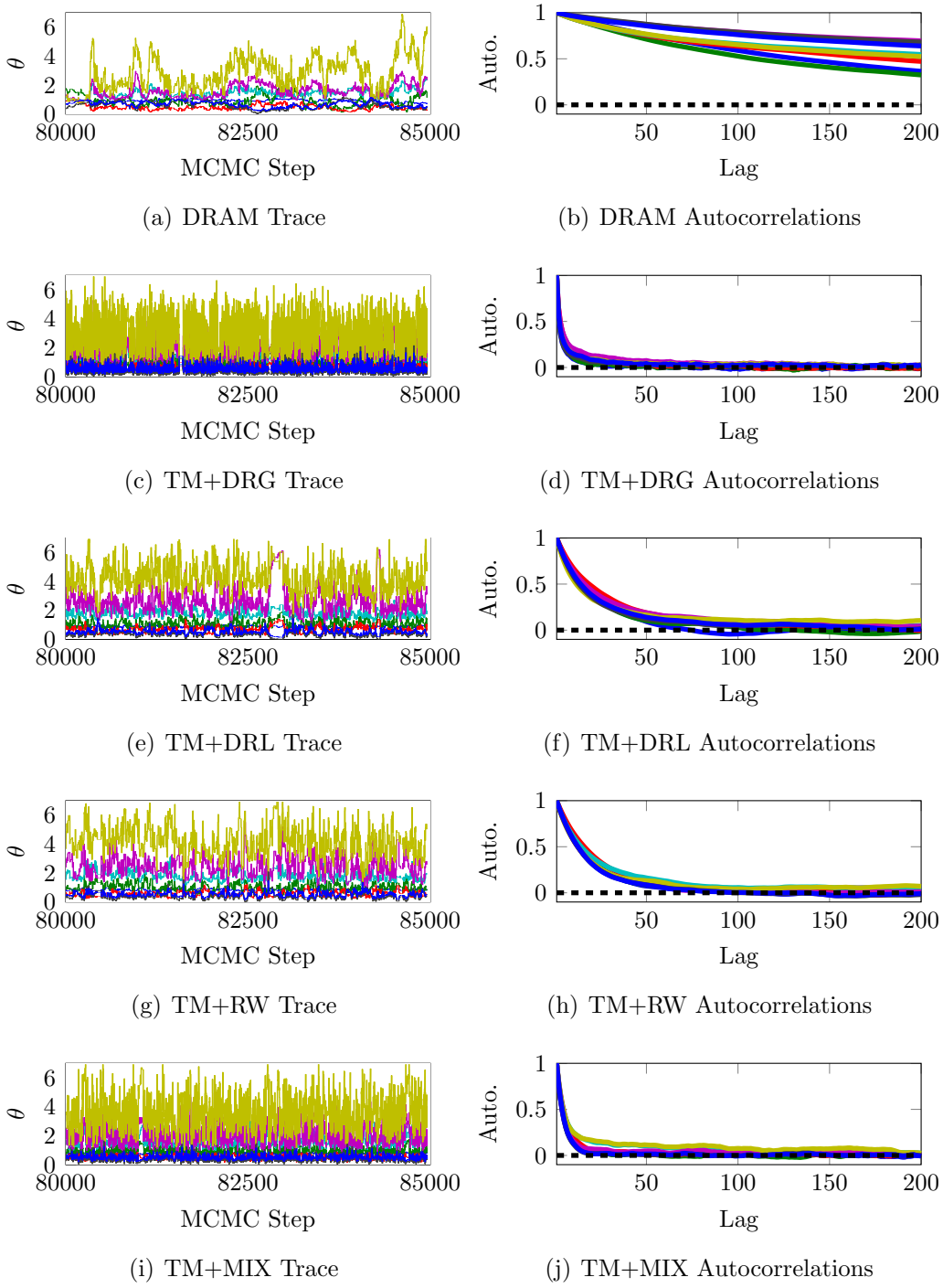


Figure 4-7: Posterior distribution for the predator prey inference example.



(a) DRAM Trace (b) DRAM Autocorrelations

(c) TM+DRG Trace (d) TM+DRG Autocorrelations

(e) TM+DRL Trace (f) TM+DRL Autocorrelations

(g) TM+RW Trace (h) TM+RW Autocorrelations

(i) TM+MIX Trace (j) TM+MIX Autocorrelations

Figure 4-8: Illustration of MCMC performance on the 8 parameter predator prey example. The left columns shows a segment of the MCMC chain. Each color corresponds to a different dimension of the chain. The right column shows typical autocorrelations for each dimension. Both the trace plot and the autocorrelation plot are taken from one run of the algorithm. Notice that the transport map chains look more like white noise, especially the TM+DRG and TM+MIX methods that utilize an independence proposal. This indicates the transport map methods are mixing better than DRAM.

4.6 Conclusions

We have introduced a new framework for adaptive MCMC that uses approximate transport maps to capture non stationary correlation structure. This method stands apart from many recently proposed approaches in that we do not require derivative information. In fact, the generally most efficient of our tested approaches (TM+DRG) requires nothing other than evaluations of the target density. While our examples focused on inference problems, we feel it is important to point out that no particular form of the the target density π is required. The efficiency of our framework is a result of two things: (1) capturing nonlinear global problem structure and (2) when possible, exploiting this global knowledge with independence proposals. The first component comes from our use of transport maps and the second component comes from our choice of reference proposal distributions (primarily TM+DRG and TM+MIX).

There is an additional cost to our framework from updating the transport map, which may become a factor for simple problems (like our logistic regression example). For these types of problems, existing methods such as NUTS or AMALA may provide the most efficient sampling strategy. However, as shown in our BOD example, our methods can be more efficient on strongly correlated problem, even when the target density is inexpensive to evaluate. It is also important to point out that our implementation[1] does not take advantage of any parallelism even though there are many levels of parallelism that could be taken advantage of when updating the map.

In this work, we used a polynomial basis to represent the transport map. However, future users are not restricted to this choice. In fact, the optimization problem for the map coefficients in (4.10) will be unchanged for any representation of the map that is linear in the coefficients. This includes other choices such as radial basis functions. Polynomials were used here primarily for their global smoothness properties, their properties are well understood, and they have long been used in the uncertainty quantification community (e.g., [74] and [66]). Even if using polynomials, choosing the basis for a specific problem (instead of the total order limited polynomials used here) is likely to decrease the computational cost of constructing the map or increase the descriptive power of the map. In either case, users are likely to see increased performance.

We should also note that extending this work to higher dimensional problems will require a more frugal choice of polynomial basis. Here we used total order limited polynomials. In high dimensional applications, more approximate maps, such as those discussed in Section 2.7 should be employed. On such high dimensional problems, combining these more approximate maps with proposal mechanisms that exploit derivative information (such as HMC or MMALA) is also likely to improve convergence. When the target density is expensive to evaluate, any parameterization of the transport map will help capture some of the target structure and yield improved sampling.

In conclusion, combining inexact transport maps with standard proposal mecha-

[1] Our implementation is freely available in MUQ, which can be downloaded at `https://bitbucket.org/mituq/muq`

nism provides a novel new framework for adaptive (or standard) MCMC that can lead to significant performance improvements. We believe this new framework is valuable tool to consider as the statistical community continues to invent and tackle ever more challenging sampling problems.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Transport maps for fast approximate conditional sampling

Algorithms for Bayesian inference typically start with Bayes' rule written as

$$\pi(\theta|d) = \frac{\pi(d|\theta)\pi(\theta)}{\pi(d)}, \tag{5.1}$$

where d and θ are real-valued random variables with corresponding joint probability space $(\mathcal{X}, \mathcal{B}(\mathcal{X}), \mu_{d,\theta})$ with $\mathcal{X} \subseteq \mathbb{R}^{(D_d+D_\theta)}$. Let D_d and D_θ be the dimensions of d and θ respectively. We assume that the distribution $\mu_{d,\theta}$ admits a *continuous* joint density $\pi(d, \theta)$ with respect to Lebesgue measure. Notice that the continuity of $\pi(d, \theta)$ implies that the posterior changes smoothly with the data d; this is a key requirement for the method discussed in this chapter.

This chapter is motivated by two challenges of working with Bayes' rule in the typical form given by (5.1). First, algorithms based solely on evaluations of the posterior in (5.1) cannot be applied until a particular value of the data d has been observed. This precludes full exploration of Bayesian posteriors in time-critical applications. We on the other hand, would like to develop algorithms for near real-time Bayesian inference. Our desired algorithms should also allow for practical tradeoffs between accuracy and computational cost. Second, with the exception of approximate Bayesian computation (ABC) approaches, most algorithms for sampling the posterior distribution require evaluations of the likelihood $\pi(d|\theta)$. In many other situations, only joint samples of (d, θ) are available; we would still like to perform inference in this settting, i.e., in the absence of a prescribed likelihood function.

In this chapter, we use transport maps to develop an approximate approach to inference that does not require likelihood evaluations and can utilize offline computation to ensure a posterior can be obtained quickly after observing the data d. We use samples of the joint distribution $\pi(d, \theta)$ to construct a transport map that allows for future approximate sampling of $\pi(\theta|d)$. The map can be constructed before any particular value of d is known, making our method well suited for time-critical applications, or for preconditioning exact inference strategies such as the map-accelerated MCMC in Chapter 4.

First, Section 5.1 will introduce our approach and then Section 5.2 will provide a small illustrative example. After that, Section 5.3 will outline how the layered maps of Section 2.7 can be used in this setting. Finally, we will apply our approach to an inference problem based on the biochemical oxygen demand model in Section 5.4. Some conclusions and future research directions will be presented in Section 5.5.

5.1 A new starting point: the joint distribution

Representing the relationship between parameters θ and observations d with the conditional density in Bayes' rule (5.1) is intuitive because it clearly distinguishes prior information in $\pi(\theta)$ from the forward models (e.g., ODES, PDES, etc...) and observations embedded in $\pi(d|\theta)$. However, in this section, we will find it more useful to work with the joint density $\pi(d, \theta)$. This joint density collects the information contained in all possible posterior distributions, which can be seen from the law of total probability

$$\pi(d, \theta) = \pi(d|\theta)\pi(\theta). \tag{5.2}$$

Moreover, in many cases, samples from the joint distribution can be easily generated. For well studied canonical prior distributions (a Gaussian distribution for instance), a sample of the joint prior $\pi(d, \theta)$ can easily be generated by first generating a sample θ' from the parameter prior $\pi(\theta)$ and then generating a sample d' from the likelihood $\pi(d|\theta')$. Notice that to sample $\pi(d, \theta)$, we do not need to have observed any particular value of d; we simply sample the prior distribution. It is only in evaluating the likelihood function that a particular value of the data d is required. This idea of *sampling* the likelihood instead of *evaluating* the likelihood is critical to our approach; this idea is also used throughout the likelihood-free methods of approximate Bayesian computation [27, 70].

Notice that the joint density $\pi(d, \theta)$ contains all the information in $\pi(\theta|d)$ for any value of d. Indeed, the conditional density $\pi(\theta|d)$ is simply a "slice" of the joint density. This concept is illustrated in Figure 5-1. The left of the figure shows the joint density as well as the value of d that we condition on to obtain the conditional density shown on the right. The values of $\pi(d, \theta)$ along the line $d = 0$ are unnormalized values of $\pi(\theta|d)$. Our goal in this work is to minimize the time it takes to characterize $\pi(\theta|d)$ *after* a particular value of d has been observed. We call the post-observation runtime of our algorithm the online time. Our idea is to use extensive offline computation (before d is observed) to characterize $\pi(d, \theta)$, so that the online time required to sample $\pi(\theta|d)$ is minimized. The key challenge is characterizing the joint distribution in a way that makes sampling the conditional distribution trivial for any d. As we show below, transport maps provide a way of describing $\pi(d, \theta)$ that suits our needs.

As in previous chapters, consider a reference random variable $r \sim N(0, I)$ that is decomposed into two components, r_d and r_θ. Naturally, r_d is a D_d dimensional random variable, and r_θ is a D_θ dimensional random variable. Also, assume we have

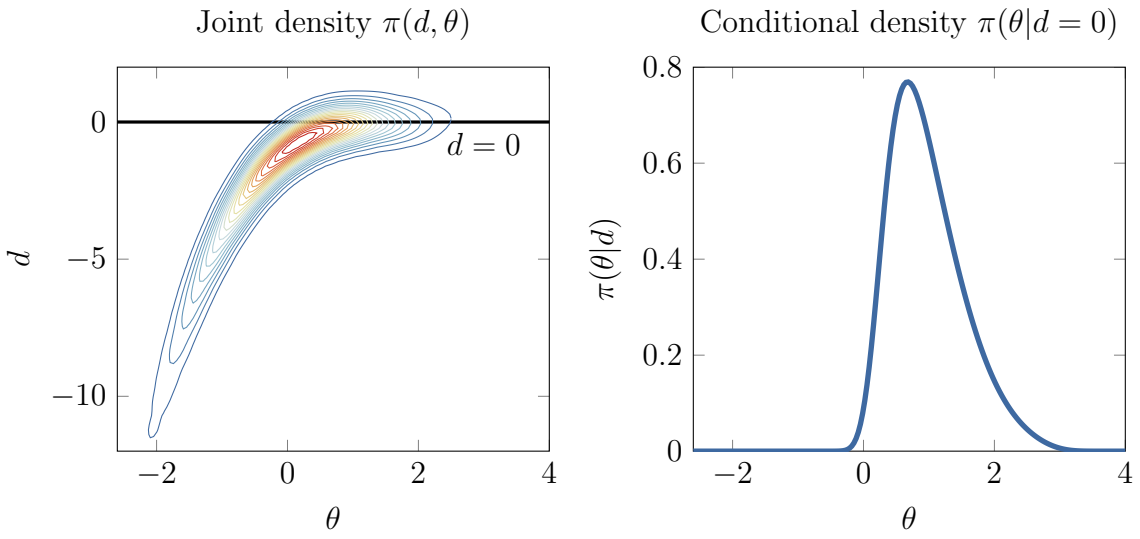


Figure 5-1: Illustration of extracting a conditional density from the joint density.

a block lower triangular map of the form

$$\begin{bmatrix} d \\ \theta \end{bmatrix} \overset{i.d.}{=} \begin{bmatrix} F_d(r_d) \\ F_\theta(r_d, r_\theta) \end{bmatrix}. \tag{5.3}$$

Even though F is a map to the joint distribution of d and θ, a map to $\theta|d$ can easily be constructed by splitting the reference random variables and taking advantage of the block triangular structure in (5.3). First, the inverse of $F_d(r_d)$, expressed as $T_d(d) = F_d^{-1}(d)$, allows us to find the reference variable r_d from the data d through

$$r_d = T_d(d). \tag{5.4}$$

We can now generate samples of the posterior random variable $\theta|d$ by fixing $r_d = T_d(d)$, sampling r_θ, and evaluating F_θ. Mathematically, we have constructed the following transport map for the posterior random variable

$$\theta|d \overset{i.d.}{=} F_\theta\left(T_d(d), r_\theta\right). \tag{5.5}$$

It is intuitive to view this composition of F_θ and T_d as a single map parameterized by the data. With this intuition, setting d to a particular value in (5.5) is like selecting a particular transport map from a parameterized family of maps. When T_d and F_θ are exact, the composed map (5.5) is also an exact map for the posterior. In practice however, only approximations $\tilde{T}_d$ and $\tilde{F}_\theta$ are available. Fortunately, these approximations can still allow us to very quickly generate approximate samples of the posterior. This can be extremely valuable when extensive offline computing is possible but extensive online computational restrictions exist. Application areas that could benefit from this include mobile and embedded systems as well as dynamic data assimilation and robust control. Imagine performing near real-time inference for a

nonlinear and non-Gaussian problem on your cell phone!

5.2 An illustrative example

As a simple illustration of using (5.5) to sample a posterior distribution, consider a simple one dimensional inference problem with the prior

$$\pi(\theta) = N(0, 1), \tag{5.6}$$

and a forward model given by

$$d = (0.6\theta - 1.0)^3 + \epsilon, \tag{5.7}$$

where the additive error is Gaussian with variance 0.2, i.e., $\epsilon \sim N(0, 0.2)$. Sampling the joint density $\pi(d, \theta)$ simply involves generating samples of ϵ and θ, and then evaluating the forward model in (5.7) to obtain d. We can trivially generate prior samples of ϵ and θ because they are both normally distributed. The left part of Figure 5-1 shows the joint distribution for this small example.

Once we have a large number of samples, 60000 in this case, we choose an appropriate basis to represent $\tilde{T}$ and $\tilde{F}$, and then solve the optimization problem in (2.21) to obtain the approximate maps $\tilde{T}_d$ and $\tilde{T}_\theta$. As in Section 2.5, we use regression to obtain the inverse map $\tilde{F}_\theta$.

Now, assume that we observe $d = 0$. Our approximate map to the posterior random variable is given by

$$(\theta|d = 0) \overset{i.d.}{\approx} \tilde{F}_\theta \left(\tilde{T}_d(0), r_\theta \right). \tag{5.8}$$

Figure 5-2 shows approximate posterior densities based on third and seventh order Hermite polynomial maps. Because this problem only contains two dimensions, we were able to use a set of total order limited multi-indices to parameterize both $\tilde{F}_\theta$ and $\tilde{T}_d$.

This example is unique in that we can analytically compute the exact posterior density. This exact density is shown in black in Figure 5-2. Clearly, the seventh order polynomial map does a superb job of capturing the posterior structure and the third order polynomial captures most of the posterior structure. Unfortunately though, our use a total order polynomial truncation in this example cannot be sustained when either the data d or parameters θ are high dimensional. In the total order setting, the number of parameters grows exponentially and in high dimensions there are simply too many coefficients in (2.23) to optimize due to both memory and time restrictions. To overcome overfitting issues, we need a large number of samples; however, with a large number of samples and many coefficients, the matrices used in Section 2.4 become too large to fit in memory. While future implementations could overcome this point, even after construction, the large expansions needed for high dimensional total order maps will be computationally expensive to evaluate.

However, a blocked version of the compositional maps in Section 2.7 is a possible

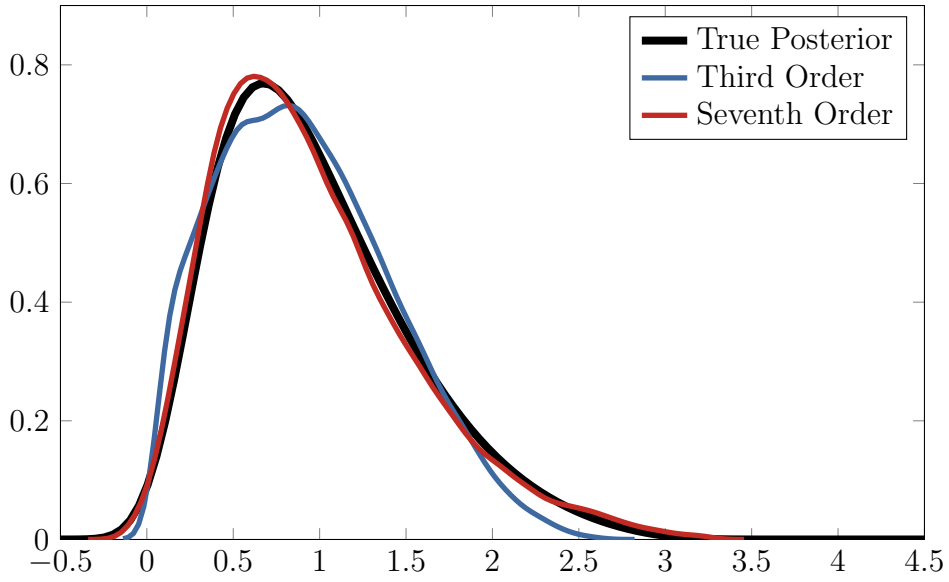


Figure 5-2: Convergence of approximate conditional density.

alternative for tackling such high dimensional problems. The next section describes how the compositional map can be adapted to maintain the block lower triangular form in (5.3). After discussing block layered maps, Section 5.4 will apply both total order polynomial maps and layered radial basis function maps to an inference problem based on the biochemical oxygen demand model.

5.3 Layered construction of block lower triangular maps

To build an approximate map to the posterior distribution, as in (5.5), we need the block lower triangular form of the map in (5.3). Unfortunately, in the layered map context, general choices of the rotations A^i and B^i in (2.50) could possibly destroy the block triangular form. The composed map will only exhibit a block lower triangular form when the linear operations A^i and B^i, as well as the map $\tilde{P}^i$, are themselves block lower triangular.

5.3.1 Layered block formulation

Let $A^i \in \mathbb{R}^{(D_d+D_\theta)\times(D_d+D_\theta)}$ and $B^i \in \mathbb{R}^{(D_d+D_\theta)\times(D_d+D_\theta)}$ be rotations of the joint (d,θ) random variable. To ensure the composed map in (2.39) is block lower triangular, A^i and B^i must take the form

$$A^i = \begin{bmatrix} A^i_{dd} & 0 \\ A^i_{\theta d} & A^i_{\theta\theta} \end{bmatrix} \tag{5.9}$$

$$B^i = \begin{bmatrix} B^i_{dd} & 0 \\ 0 & B^i_{\theta\theta} \end{bmatrix}, \tag{5.10}$$

where $A^i_{dd}, B^i_{dd} \in \mathbb{R}^{D_d \times D_d}$, $A^i_{\theta d} \in \mathbb{R}^{D_\theta \times D_d}$, and $A^i_{\theta\theta}, B^i_{\theta\theta} \in \mathbb{R}^{D_\theta \times D_\theta}$. Notice that the lower left block of B^i must be 0 for B^i to be orthonormal; $B^i_{\theta\theta}$ and B^i_{dd} must also be orthonormal. To ensure A^i is invertible, A^i_{dd} and $A^i_{\theta,\theta}$ must both be invertible.

Now, following the form for one layer in the composed map (see (2.50)), let $\tilde{P}^i : \mathbb{R}^{(D_d+D_\theta)} \to \mathbb{R}^{(D_d+D_\theta)}$ be a lower triangular map consisting of the following two components

$$\tilde{P}^i(d,\theta) = \begin{bmatrix} \tilde{P}^i_d(d) \\ \tilde{P}^i_\theta(d,\theta) \end{bmatrix}. \tag{5.11}$$

Combining this nonlinear transformation with the linear transformations A^i and B^i yields the following form for one layer of the composed block lower triangular map,

$$\tilde{T}^i(d,\theta) = \begin{bmatrix} B^i_{dd}\tilde{P}^i_d(A^i_{dd}d) \\ B^i_{\theta\theta}\tilde{P}^i_\theta(A^i_{dd}d, A^i_{\theta d}d + A^i_{\theta\theta}\theta) \end{bmatrix}. \tag{5.12}$$

The approximate inverse of $\tilde{T}^i$, denoted by $\tilde{F}^i$, can also be broken into two components as follows

$$\tilde{F}^i(r_d, r_\theta) = \begin{bmatrix} \tilde{F}^i_d(r_d) \\ \tilde{F}^i_\theta(r_d, r_\theta) \end{bmatrix}, \tag{5.13}$$

where

$$\tilde{F}^i_d(r_d) = [A^i_{dd}]^{-1}(\tilde{P}^i_d)^{-1}\left(B^{iT}_{dd} r_d\right), \tag{5.14}$$

and $\tilde{F}^i_\theta$ is the "inverse" of $\tilde{T}^i_\theta$ for a fixed value of d. More precisely, for any $d \in \mathcal{X}_d$ $\tilde{F}^i_\theta$ satisfies

$$\tilde{T}^i_\theta\left(d, \tilde{F}^i_\theta\left(\tilde{T}^i_d(d), r_\theta\right)\right) = r_\theta. \tag{5.15}$$

This definition of $\tilde{F}^i_d$ and $\tilde{F}^i_\theta$ may seem convoluted, but constructing the joint map $\tilde{F}^i$ in this block-layered setting is no different than the standard regression approach introduced in Section 2.5.

To see this connection, we need to define intermediate random variables $\tilde{r}^i_d$ and $\tilde{r}^i_\theta$ that are approximations to r_d and r_θ after the first i layers of the composed map. In this block composed map, $\tilde{r}^i_d$ is given by

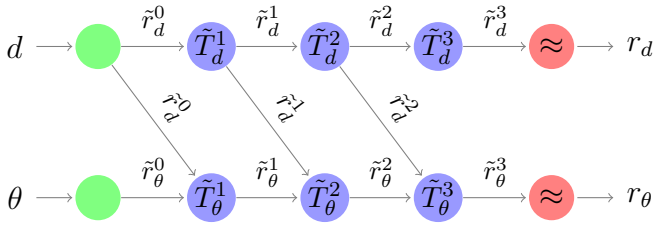
$$\tilde{r}^i_d = \begin{cases} d & i = 0 \\ B^i_{dd}\tilde{P}^i_d(A^i_{dd}\tilde{r}^{i-1}_d) & i > 0 \end{cases}. \tag{5.16}$$

Similarly, the θ component $\tilde{r}^i_\theta$ is defined by

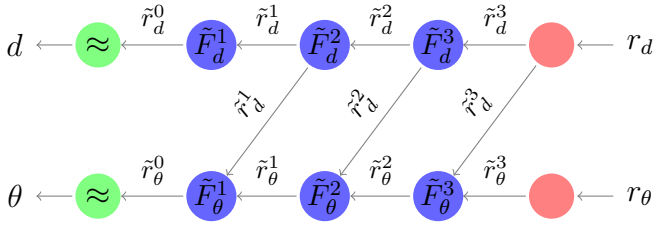
$$\tilde{r}^i_\theta = \begin{cases} \theta & i = 0 \\ B^i_{\theta\theta}\tilde{P}^i_\theta(A^i_{dd}\tilde{r}^{i-1}_d, A^i_{\theta d}\tilde{r}^{i-1}_d + [A^i_{\theta\theta}]^{-1}\tilde{r}^{i-1}_\theta) & i > 0 \end{cases}. \tag{5.17}$$

Figure 5-3 shows a graphical interpretation of these quantities.

Now, assume we have already constructed $\tilde{T}^i$ and thus have A^i, B^i, and $\tilde{P}^i(\tilde{r}^{i-1}) = \left[\tilde{P}^i_d(\tilde{r}^{i-1}_d),\ \tilde{P}^i_\theta(\tilde{r}^{i-1}_d, \tilde{r}^{i-1}_\theta)\right]^T$ from (5.12). Just like the standard layered map in Section



(a) Illustration of $\tilde{T}(d, \theta)$ evaluation.



(b) Illustration of $\tilde{F}(r_d, r_\theta)$ evaluation.

Figure 5-3: Illustration of block composed map evaluation.

2.7, the inverse map $\tilde{F}^i(\tilde{r}^i)$ will take the form

$$\begin{aligned}\tilde{F}^i(\tilde{r}^i) = \tilde{r}^{i-1} &= [A^i]^{-1}(\tilde{P}^i)^{-1}\left(B^{iT}\tilde{r}^i\right) \\ &= \begin{bmatrix} A^i_{dd} & 0 \\ A^i_{\theta d} & A^i_{\theta\theta} \end{bmatrix}^{-1} (\tilde{P}^i)^{-1}\left(\begin{bmatrix} B^i_{dd} & 0 \\ 0 & B^i_{\theta\theta} \end{bmatrix}^T \begin{bmatrix} \tilde{r}^i_d \\ \tilde{r}^i_\theta \end{bmatrix}\right). \end{aligned} \tag{5.18}$$

Now, to construct the nonlinear inverse $(\tilde{P}^i)^{-1}$ using the regression techniques of Section 2.5, we need samples of the input and output of $(\tilde{P}^i)^{-1}$. Fortunately, these samples can easily be obtained using A^i and $\tilde{P}^i$ from the already computed forward map $\tilde{T}^i$. More precisely, the inputs to $(\tilde{P}^i)^{-1}$ are given by $\tilde{P}^i(A^i\tilde{r}^{i-1})$ and the outputs are $A^i\tilde{r}^{i-1}$.

Using these input-output samples, the intermediate inverse map $(\tilde{P}^i)^{-1}$ can easily be constructed using regression. Critically, because the resulting map $(\tilde{P}^i)^{-1}$ will be lower triangular, the tools from Section 2.5 can be used directly without consideration of the block structure needed here. The blocked choices of A^i and B^i defined above allow us to simply use the first D_d outputs of $(\tilde{P}^i)^{-1}$ as $(\tilde{P}^i_d)^{-1}$, and the last D_θ outputs as $(\tilde{P}^i_\theta)^{-1}$. With these definitions, one complete layer of the inverse map is given by

$$\begin{aligned}\tilde{F}^i(\tilde{r}^i) &= \begin{bmatrix} \tilde{F}^i_d(\tilde{r}^i_d) \\ \tilde{F}^i_\theta(\tilde{r}^i_d, \tilde{r}^i_\theta) \end{bmatrix} \\ &= \begin{bmatrix} [A^i_{dd}]^{-1}(\tilde{P}^i_d)^{-1}\left(B^{iT}_{dd}\tilde{r}^i_d\right) \\ [A^i_{\theta\theta}]^{-1}\left[(\tilde{P}^i_\theta)^{-1}\left(B^{iT}_{dd}\tilde{r}^i_d, B^{iT}_{\theta\theta}\tilde{r}^i_\theta\right) - A^i_{\theta d}\tilde{F}^i_d(\tilde{r}^i_d)\right] \end{bmatrix}, \end{aligned} \tag{5.19}$$

where the appearance of $\tilde{F}^i_d(\tilde{r}^i_d)$ in the definition of $\tilde{F}^i_\theta(\tilde{r}^i_d, \tilde{r}^i_\theta)$ stems from using block-

wise forward substitution to evaluate $[A^i]^{-1}$.

Notice that in (5.19) and in Figure 5-3, information never flows from the θ components of the map to the d components. This means that given d, we can compute each $\tilde{r}_d^i$ without knowing the target random variable θ. As discussed in the next section, this separability is critical to constructing a layered equivalent to the conditional map $\tilde{F}_\theta\left(\tilde{T}_d(d), r_\theta\right)$ defined in (5.8).

5.3.2 Defining a compositional inference map

Our goal is to sample $\theta|d$ using a combination of the maps $\tilde{F}_d^i$, $\tilde{F}_\theta^i$, and $\tilde{T}_d^i$. We will not directly require $\tilde{T}_\theta^i$ to sample the posterior random variable $\theta|d$.[1] Recall from section 5.1 that we can view the map $\tilde{F}_\theta\left(\tilde{T}_d(d), r_\theta\right)$, as a map from $\mathbb{R}^{D_\theta} \to \mathbb{R}^{D_\theta}$ that is parameterized by the data d. A similar view can be taken with the layered map. Each layer $\tilde{F}_\theta^i$ can also be viewed as a map from $\mathbb{R}^{D_\theta} \to \mathbb{R}^{D_\theta}$, but one that is parameterized by the intermediate variable $\tilde{r}_d^i$. This can be seen graphically in Figure 5-3(b).

For a fixed d, we construct the layered map to $\theta|d$ by first evaluating each layer of $\tilde{T}_d^i$ to obtain the values of $\{\tilde{r}_d^1, \tilde{r}_d^2, ..., \tilde{r}_d^N\}$, where N is the total number of layers in the compositional map. Graphically, this process is simply moving from left to right across the top nodes in Figure 5-3(a). Once these values have been obtained, we can generate approximate samples of $\theta|d$ by sampling $r_\theta \sim N(0, I)$ and then moving from right to left along the bottom nodes in Figure 5-3(b). Mathematically, this simply involves evaluating the composition

$$\theta|d \overset{i.d.}{\approx} \tilde{F}_\theta^1(\tilde{r}_d^1,\ \tilde{F}_\theta^2(\tilde{r}_d^2,\ ...\ \tilde{F}_\theta^{N-1}(\tilde{r}_d^{N-1},\ \tilde{F}_\theta^N(\tilde{r}_d^N, r_\theta))\ ...\ )), \tag{5.20}$$

where each intermediate variable $\tilde{r}_d^i$ is fixed.

5.3.3 Choosing block rotations

In Section 2.7.4 we introduced five different methods for computing the rotations A^i and and B^i. These methods were based on completely random rotations, random rotations targeting non-Gaussian directions, principal components, and alternating random rotations with the principal components. Unfortunately, these rotations cannot be used directly in the block-lower triangular case because A^i and B^i must also respect the block lower triangular form in (5.10).

While the construction of block triangular layered maps in Section 5.3.1 is completely general, it did not specify how to select the matrices A^i and B^i. In the present work, we take a basic approach for constructing A^i and B^i. We first set $A_{\theta d}^i = 0$ and then use any one of the techniques from Section 2.7.4 to independently construct A_{dd}^i and $A_{\theta\theta}^i$. The post-transformation rotations are then simply defined by $B_{dd}^i = A_{dd}^{iT}$, and $B_{\theta\theta}^i = A_{\theta\theta}^{iT}$. These choices for A^i and B^i ensure that both the block structure

[1] Note that $\tilde{T}_\theta^i$ still needs to be constructed. This map is not required to sample the posterior, but it is required to find the pairs of $\tilde{r}^i$ and $\tilde{r}^{i-1}$ that we need to construct $\tilde{F}_\theta^i$.

in (5.10) is satisfied and that the space of all possible maps at layer i contains the identity map. This is needed to show that the layered map has a non-increasing error.

While this choice of rotations is simple, using block diagonal matrices for A^i and B^i can hinder the ability of the layered map to adequately characterize the posterior $\pi(\theta|d)$.

The correlation structure between θ and d needs to be adequately captured for any pair of forward and inverse maps $\tilde{T}(d,\theta)$ and $\tilde{F}(r_d, r_\theta)$, to adequately characterize the posterior $\pi(\theta|d)$. However, by setting $A^i_{\theta d} = 0$ in the layered map, we have prevented A^i from rotating the coordinates in many directions. Unfortunately, this restriction does not allow the map to fully explore all of the joint dependencies between the parameters and data. The results below suggest that this is a significant limiter to the layered map's accuracy. Future work will need to explore alternative choices where $A^i_{\theta d} \neq 0$.

5.4 Application to biochemical oxygen demand inference

Here we will again consider the biochemical oxygen demand (BOD) model introduced in Section 4.5.2. Recall the simple exponential BOD model defined by $B(t) = a(1 - \exp(-bt)) + e$, where $e \sim N(0, 10^{-3})$ is an additive error. In this example we use 5 observations of $B(t)$ at $t = \{1, 2, 3, 4, 5\}$. Our goal is to infer the scalars a and b. For the illustrative purposes of this example, we also assume that a and b have uniform priors:

$$a \sim U(0.4, 1.2) \tag{5.21}$$
$$b \sim U(0.01, 0.31). \tag{5.22}$$

The target random variables $\theta_1 \sim N(0,1)$ and $\theta_2 \sim N(0,1)$ are chosen to be independent standard normal random variables that are related to the parameters a and b through the CDF of a standard normal distribution. Mathematically, this relationship is expressed as

$$a \overset{i.d.}{=} \left[0.4 + 0.4\left(1 + \text{erf}\left(\frac{\theta_1}{\sqrt{2}}\right)\right)\right] \tag{5.23}$$
$$b \overset{i.d.}{=} \left[0.01 + 0.15\left(1 + \text{erf}\left(\frac{\theta_2}{\sqrt{2}}\right)\right)\right]. \tag{5.24}$$

We chose to use these transformations instead of inferring a and b directly for two reasons: (i) this choice ensures that $a > 0$ and $b > 0$ for any map, and (ii) using a polynomial map to approximate the error function erf adds an unnecessary level of complexity to this example.

In this example, d is a vector-valued random variable defined by

$$d = [B(t=1;\theta_1,\theta_2), B(t=2;\theta_1,\theta_2), B(t=3;\theta_1,\theta_2), B(t=4;\theta_1,\theta_2), B(t=5;\theta_1,\theta_2)]^T.$$

We use joint samples of d and θ to construct the transport maps $\tilde{T}(d, \theta)$, and $\tilde{F}(r_d, r_\theta)$. Below, we compare the use of either 5000 or 50000 samples to construct the map.

After generating the samples, a fixed noisy realization of the data is generated using "true" parameter values $\theta_1 = 0.7$ and $\theta_2 = 0.3$ to define the inference problem. We test both the accuracy of our approach and the efficiency of our approach against an adaptive Metropolis MCMC scheme [42]. The MCMC result is used as a gold-standard "truth" in our accuracy comparisons below. While more efficient MCMC algorithms exist and were applied to a similar BOD problem in Chapter 4, the adaptive Metropolis algorithm is well known and provides a good qualitative feel for the speed of our approach relative to a well-used standard method.

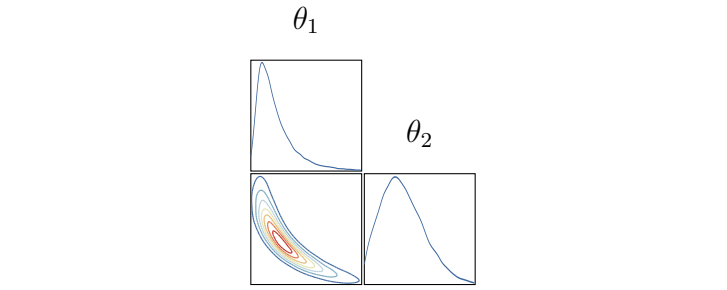
Table 5.1: Accuracy of offline inference maps on BOD problem. All polynomial maps used a Hermite basis.

Map Type	Training Samples	Mean		Variance		Skewness		Kurtosis	
		θ_1	θ_2	θ_1	θ_2	θ_1	θ_2	θ_1	θ_2
MCMC "Truth"		0.075	0.875	0.190	0.397	1.935	0.681	8.537	3.437
Linear	5000	0.199	0.717	0.692	0.365	-0.005	0.010	2.992	3.050
	50000	0.204	0.718	0.669	0.348	0.016	-0.006	3.019	3.001
Cubic	5000	0.066	0.865	0.304	0.537	0.909	0.718	4.042	3.282
	50000	0.040	0.870	0.293	0.471	0.830	0.574	3.813	3.069
Fifth Order	5000	0.027	0.888	0.200	0.447	1.428	0.840	5.662	3.584
	50000	0.018	0.907	0.213	0.478	1.461	0.843	6.390	3.606
Seventh Order	5000	0.090	0.908	0.180	0.490	2.968	0.707	29.589	16.303
	50000	0.034	0.902	0.206	0.457	1.628	0.872	7.568	3.876
Layered R-SVD	5000	0.001	0.813	0.534	0.256	0.353	1.201	4.296	5.951
	50000	0.167	0.690	0.580	0.235	0.497	1.116	4.317	5.279
Layered Choosy	5000	0.226	0.640	0.537	0.191	0.660	1.002	4.653	4.982
	50000	0.081	0.851	0.558	0.260	0.648	0.795	4.351	4.320

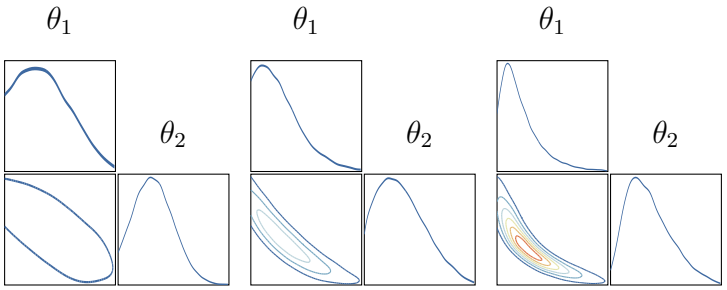
5.4.1 Accuracy

Table 5.1 compares the accuracy of our map-based approach with the MCMC sampler. The adaptive Metropolis sampler was tuned to have an acceptance rate of 26%. Moreover, the chain was run for 6e6 steps, 2e4 of which were used as a burn-in. After constructing $\tilde{T}$ and $\tilde{F}$, the maps were used to generate 3e4 approximate samples of $\pi(\theta|d)$. The moments calculated from those samples are compared to the MCMC gold-standard in Table 5.1. Kernel density estimates of the approximate posterior densities are also illustrated in Figures 5-4 and 5-5. For the layered maps, 5 layers were constructed and each layer was defined by the nonlinear lower triangular expansion in (2.26) using 15 radial basis functions in each direction. The comparison considers layered maps with choosy random rotations, or with principal components interleaved with random rotations (R-SVD).

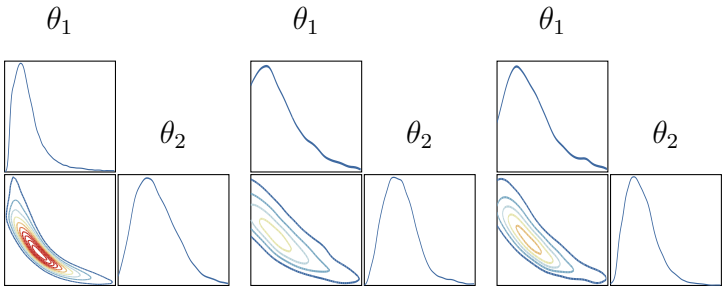
From the accuracy table, we can see that with a cubic parameterization of $\tilde{T}$ and $\tilde{F}$, the posterior map captures the posterior mean and variance, but cannot capture higher moments. However, the seventh order maps are quite accurate. From Table 5.1, we see that when constructed from 50,000 samples, the seventh order map even



(a) MCMC Truth



(b) Linear (c) Cubic (d) Fifth Order

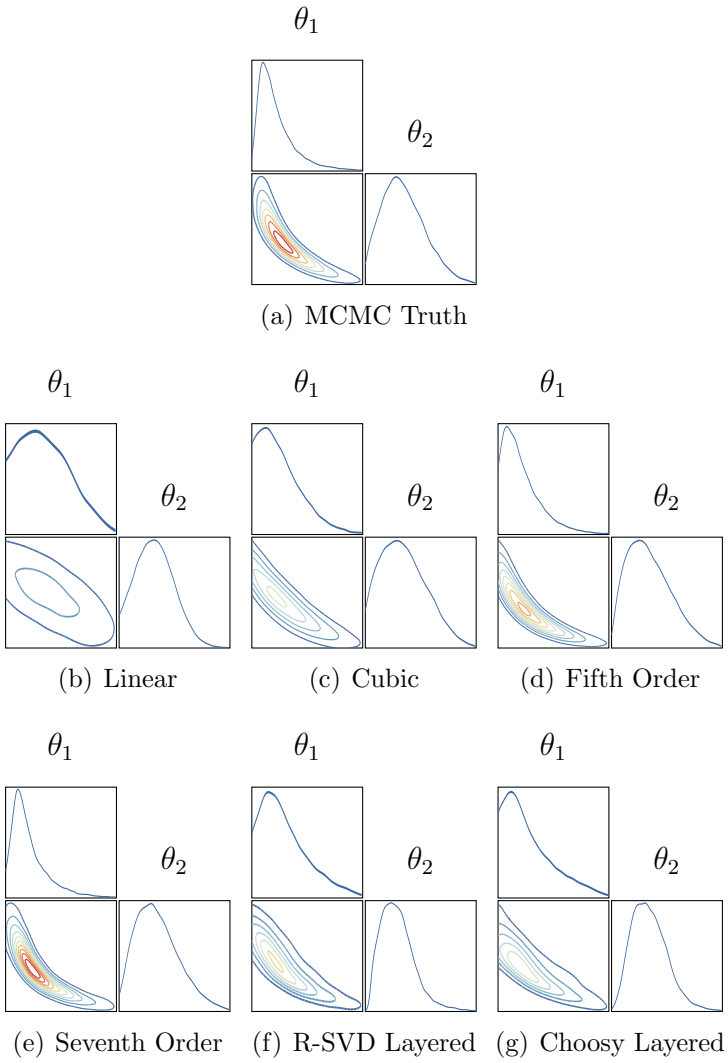


(e) Seventh Order (f) R-SVD Layered (g) Choosy Layered

Figure 5-4: Approximate BOD posterior densities constructed with 5000 samples. Contour levels and color scales are constant for all figures.

captures the posterior kurtosis quite well. Figure 5-7 shows the seventh order posterior distribution. The plot looks nearly identical to the MCMC gold-standard. However, the timing results given below show that even with this comparable accuracy, our use of maps results in much faster posterior sampling.

Like the cubic map, the layered maps have a hard time representing the posterior. Looking at the joint distributions in Figures 5-6 and 5-7, we can see that the layered maps capture the d marginal $\pi(d)$ (top 5 rows of each plot), but have trouble capturing the θ components (bottom 2 rows of each plot). This is particularly obvious when looking at the the θ_1-θ_2 joint marginal, which should be an uncorrelated Gaussian density. This is likely a result of setting the off-diagonal transformation $A^i_{\theta d} = 0$ in (5.10). While computationally convenient, this choice of rotation does not allow the transformation A^i to span all directions in the joint (d, θ) space.



(a) MCMC Truth

(b) Linear (c) Cubic (d) Fifth Order

(e) Seventh Order (f) R-SVD Layered (g) Choosy Layered

Figure 5-5: Approximate BOD posterior densities constructed with 50000 samples. Contour levels and color scales are constant for all figures.

5.4.2 Timing

From Table 5.1 and Figures 5-4 and 5-5, we see that our map-based approach can accurately characterize the posterior. However, our goal is to decrease the online computational cost of sampling the posterior compared to MCMC. In Table 5.2, we show various timings of the map-based approach. These times are compared to the gold-standard adaptive Metropolis MCMC scheme. The online time shows how long each method takes to generate 30000 independent samples of the posterior. For MCMC, we use the average amount of time it takes to generate a chain with an effective sample size (ESS) of 30000.

The polynomial transport maps are clearly more efficient than the adaptive Metropolis MCMC sampler. In fact, even if a much more advanced MCMC sampler were used (see Chapter 4), this conditional map approach would still outperform the MCMC

Table 5.2: Efficiency of offline inference maps on BOD problem. The online time is the time required after observing d to generate the equivalent of 30000 independent samples. Again, all polynomial maps were constructed from Hermite polynomials.

Map Type	Training Samples	Offline time (sec)			Online time (sec)
		Rotation	$\tilde{T}$ construction	$\tilde{F}$ regression	
MCMC "Truth"		NA			591.17
Linear	5000	NA	0.46	0.18	2.60
	50000	NA	4.55	1.65	2.32
Cubic	5000	NA	4.13	1.36	3.54
	50000	NA	40.69	18.04	3.58
Fifth Order	5000	NA	22.82	8.40	5.80
	50000	NA	334.25	103.47	6.15
Seventh Order	5000	NA	145.00	40.46	8.60
	50000	NA	1070.67	432.95	8.83
Layered R-SVD	5000	0.03	67.27	52.26	315.45
	50000	0.14	463.81	521.54	315.28
Layered Choosy	5000	181.23	60.05	50.38	315.65
	50000	352.91	491.28	500.17	306.90

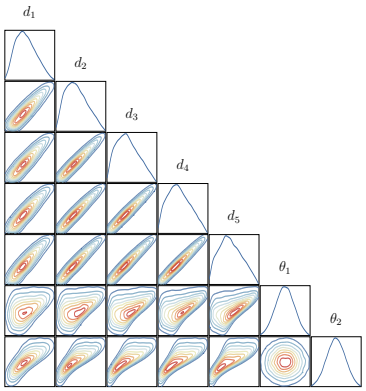
sampler (in terms of ESS) because we are generating *independent* samples. However, we must stress that the samples from our approach are still only *approximate* posterior samples. The approximation can be quite good (as shown in Table 5.1), but we provide no accuracy guarantee. That said, one of the approximate maps constructed here could easily be used as a good proposal mechanism in either an MCMC setting or importance sampling framework. A map constructed using the offline techniques in this chapter could provide an excellent initial map in the adaptive map-accelerated MCMC scheme of Chapter 4. Exploring this combination is left to future work.

5.5 Discussion

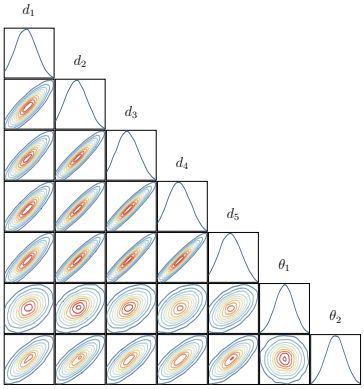
The efficient online sampling demonstrated on the BOD example is possible because we characterized the joint distribution of (d, θ) before any data was observed. Block lower triangular transport maps provide an efficient way of describing the joint distribution that also allow for efficient posterior sampling of $\pi(\theta|d)$. One view of our approach is that we are constructing a family of maps for all posterior distributions – the family is parameterized by the data. With this view, a natural question is, "How can we construct maps to all posteriors without enumerating all possible observations?" The answer is that we have implicitly assumed that the posterior density exists and changes smoothly with the data. In some sense, this smoothness is what allows us to "interpolate" between posterior distributions.

From the BOD example, we see that for small dimensional problems, direct use of total-order polynomials allows us to accurately characterize the posterior. While layered maps have a lot of promise for extending this work to high dimensional problems, more research into the choice of the off-diagonal transformation $A^i_{\theta d}$ is required to make these methods practical. Regardless, our approach can produce good ap-

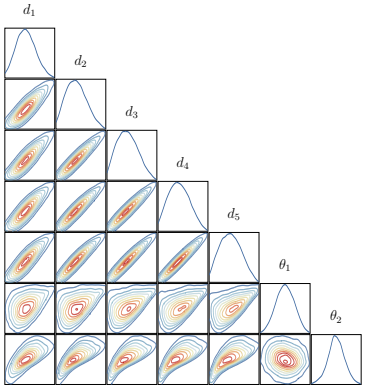
proximations to the posterior density in almost two orders of magnitude less online time than adaptive Metropolis MCMC. On top of that, our approach is much more scalable to large parallel architectures than standard sampling methods. In addition to parallelizing the map construction, the maps could be replicated and used on multiple nodes or GPUs for massively parallel sampling. Each posterior sample could in principle be generated in parallel without any forward model evaluations. This lightweight massively parallel computation is ripe for a GPU implementation. In fact, we are not aware of another method for posterior sampling that scales in this manner. Fast approximate using these techniques may open of a range of exciting possibilities, including online system identification, fast Bayesian inference on mobile devices, and near real-time Bayesian experimental design.



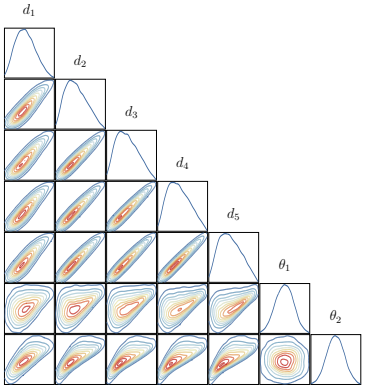
(a) True Joint Distributions



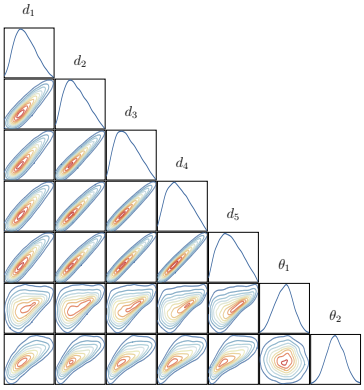
(b) Linear



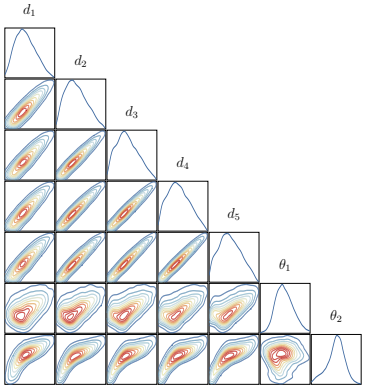
(c) Cubic



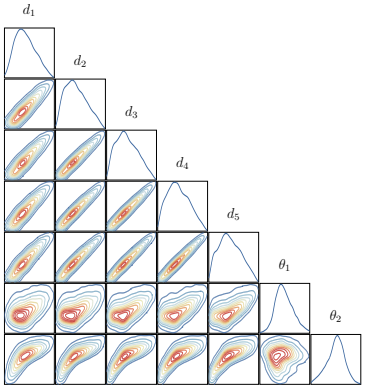
(d) Fifth Order Map



(e) Seventh Order

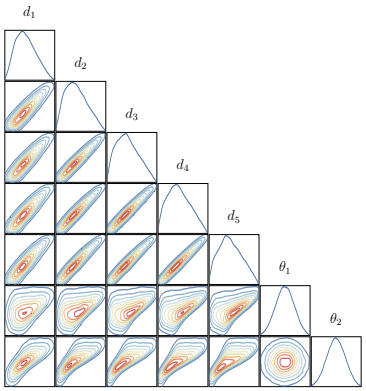


(f) R-SVD Layered

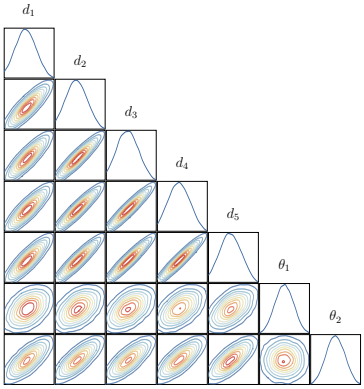


(g) Choosy Layered

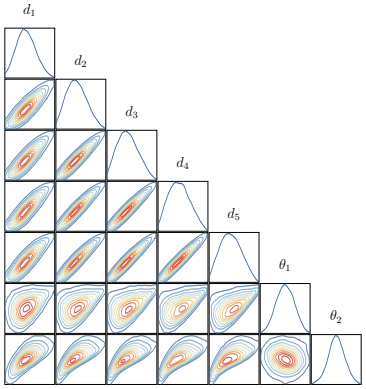
Figure 5-6: Approximate BOD joint densities constructed with 5000 samples.



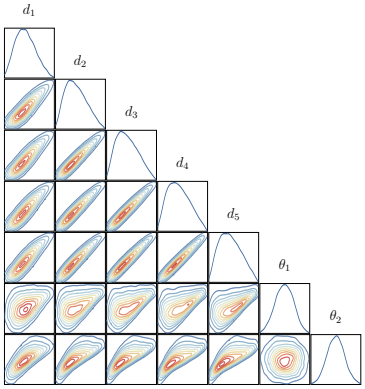
(a) True Joint Distributions



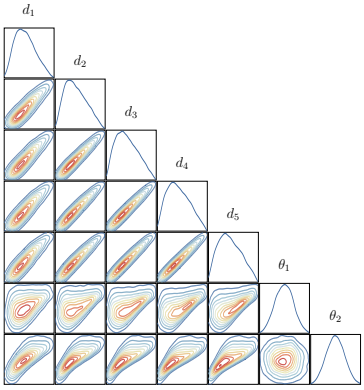
(b) Linear



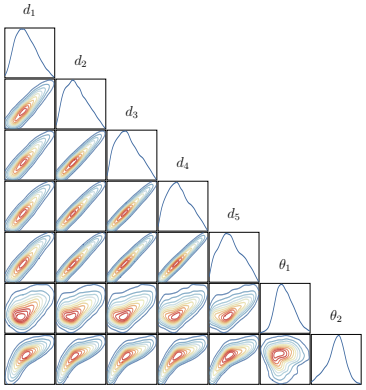
(c) Cubic



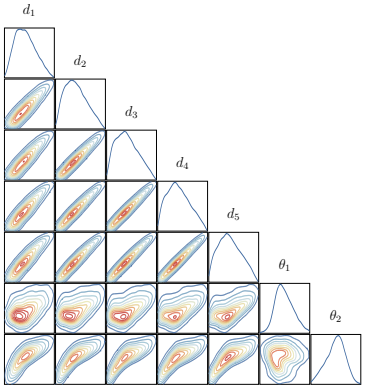
(d) Fifth Order



(e) Seventh Order



(f) R-SVD Layered



(g) Choosy Layered

Figure 5-7: Approximate BOD joint densities constructed with 50000 samples.

Chapter 6

MUQ: A software framework for uncertainty quantification

Algorithm developers are continually creating new, more efficient, techniques for Markov chain Monte Carlo (MCMC), nonlinear programming, and other problems throughout computational science and engineering. Unfortunately though, outside the realm of commercial software packages (e.g., CPLEX, GUROBI), there is often a long lag time between when a new algorithm is developed and when, if at all, it is broadly applied by users. This poses a problem for both algorithmic developers and algorithm users. Users can obviously benefit from faster, more efficient, algorithms coming from the developer community. On the other side, algorithmic developers are likely to produce more efficient tools when focusing on specific user problems. So the question is: How can we create a stronger link between algorithmic research and applications by minimizing the lag time between algorithm development and widespread algorithm use? As an initial attempt at tackling this problem in the uncertainty quantification community, we have developed the MIT Uncertainty Quantification software library, nicknamed MUQ.

MUQ has two target audiences: (1) people who need to *use* uncertainty quantification tools to analyze a model, solve an inverse problem, or perform some of data analysis and (2) people that wish to *develop* new uncertainty quantification tools that will be useful to a variety of users. Our goal is to create a software framework that is easy for both users *and* developers to work with and expand.

Users need to be able to easily implement new models, developers need to be able to easily implement new algorithms, and both groups need an expressive and easy to use model-algorithm interface. In Section 6.1 we outline our approach for constructing models, then in Section 6.2 we show how MUQ facilitates the easy development and testing of new MCMC algorithms, and finally in Section 6.3, we provide some high level conclusions and general trends to consider in future work.

6.1 Structured modeling

In the context of this work, we use the term "model" to describe a function taking M model parameters as input and having N outputs. In our setting, a model is a numerical approximation to some physical or stochastic process. It is important to point that here, a model does not necessarily refer to a mathematical description of the physical process, but rather a numerical approximation to the mathematical description. Let $f : \mathbb{R}^M \to \mathbb{R}^N$ be a numerical model of some physical process. For example, f could be a subsurface flow model that takes a permeability field as input and produces predictions of pressure at N locations. In a more complicated situation, f may include several coupled PDE models – perhaps the output of a subsurface flow model is fed as the input to a contaminant transport model. At any rate, to be used by optimization algorithms or MCMC methods, the model f needs to be implemented (i.e. coded up) with an input-output interface that the optimization and MCMC algorithms can interpret. Moreover, if any algorithm requires additional structure (e.g., block definitions of the input parameters or derivative information), this information also needs to be incorporated into the model-algorithm interface.

6.1.1 Existing algorithm-model interfaces

There are many options for how to define models in a way that algorithms can understand. On a high level, the main variation between these options is in how much flexibility should be given to the person or group implementing the model. Flexibility can include the constraints on the model complexity, any requirements on the programming language used to implement the model, and even if the model has to return derivative information such as Jacobian matrices or adjoint gradients. The most flexible interfaces, such as that used by the DAKOTA package from Sandia[2], only require that the implementation takes an input vector and returns an output vector. Additional flexibility is provided in DAKOTA by performing model evaluations using file i/o and system calls. While this type of "black-box" approach gives users the most flexibility in defining their models, it makes it difficult for algorithms to extract model structure and can also make it difficult for users to implement a model because everything in the model needs to be written from scratch.

On the opposite end of the flexibility spectrum from DAKOTA are algebraic modeling systems such as GAMS [94] or AMPL [35] in the optimization community, and STAN [96] or BUGS [69] in the MCMC community. These tools define a new domain specific modeling language. To define a model in these frameworks, a user defines an input file that describes model parameters and any algebraic relationships between them. This type of model specification is usually quite easy to specify and requires almost no programming experience. Moreover, because all operations are explicitly given defined in the input file, the optimization or MCMC software can extract any information it needs, e.g. sensitivities, linearities, etc... Unfortunately though, requiring the model to be implemented in such an algebraic modeling language restricts the type of model that can be implemented (e.g., no PDE based systems) and also prevents users from employing "black-box" models or legacy code.

One of our visions for MUQ is to provide a modeling framework that lies between the extremes of DAKOTA and algebraic modeling systems. We want to provide enough flexibility to use black-box models or existing code, but simultaneously expose some model structure to the algorithms. We also acknowledge that modeling is in itself an iterative process and the modeling framework should allow for models to be incrementally constructed, tested, and adapted. With these goals in mind, we adopted a graph-based framework for constructing models. In this setting, the full model f is separated into many smaller components that are defined on nodes of the graph. While somewhat similar to the way BUGS internally handles models, our framework is much more flexible and allows non-algebraic or black-box simulations to be included in the graph.

6.1.2 MUQ's modeling framework

To give a feel for our modeling framework, consider an implementation of a simple elliptic PDE with a lognormal permeability field. The physical model is defined by

$$-\nabla \cdot (\kappa \nabla p) = f, \tag{6.1}$$

where κ is an input to the model representing the permeability field, p is the output of the model representing the pressure field, and f is and input to the model that characterizes a recharge term. Now, assume the permeability κ is parameterized through some other parameter (perhaps log-permeability) θ_1 and f depends on two other parameters (perhaps rainfall and temperature) θ_2 and θ_3. Graphically, this relationship is shown in Figure 6-1. Figure 6-1 also shows an observation $h(p)$ that could represent a limited number of borehole pressure observations.

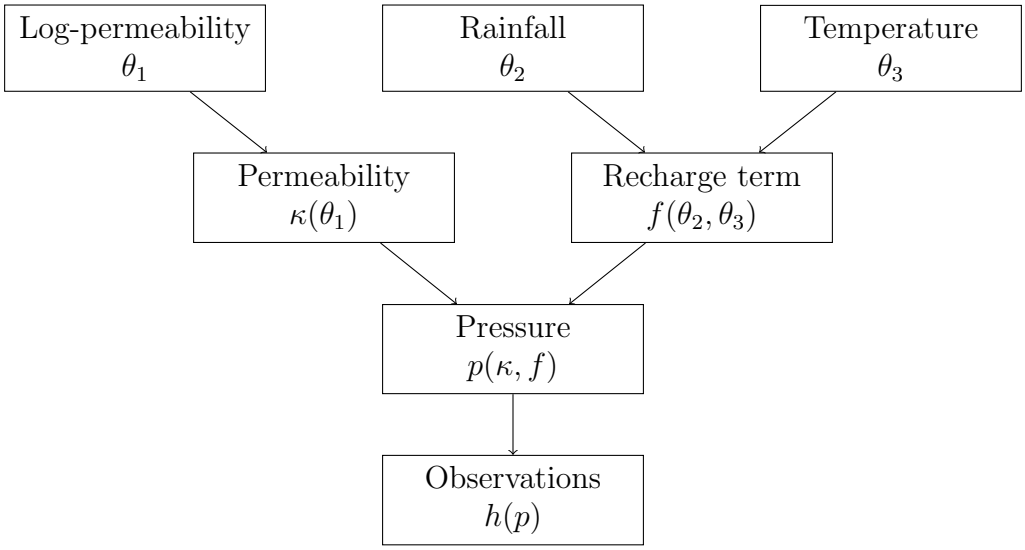


Figure 6-1: Illustration of graphical modeling approach. MUQ uses graphs like this to define models, compute derivatives with the chain rule, and extract additional problem structure.

Drawing the graph in Figure 6-1 is a common way for users to represent information flow through a model. MUQ uses a similar internal representation of models. In MUQ, each node of the graph represents a model component. These components may require the solution of a PDE (as in the pressure node of Figure 6-1), a system call to some black-box simulator, or the component could perform some simple task like adding a scalar to the input vector. While each model component can contain a variety of operations, the large scale model structure is captured by the graph. This enables us to automatically propagate sensitives through the model (via the chain rule) or extract other problem structure such as model linearity. While DAKOTA treats models as "black-boxes" and algebraic systems ensure the model is an entirely known "white-box," our graph-based modeling framework provides an intermediate "gray-box." By sacrificing some of the ease-of-use of algebraic systems, we have gained the full flexibility of file i/o based frameworks. Importantly though, we have not lost the ability to extract problem structure and pass this information on to algorithms.

Our graph-based models make extracting problem structure possible for algorithms, but this approach can also be useful from a modeler's perspective. Our framework allows nodes on the graph (i.e. model components) to easily be swapped for new model components. For example, a node containing a computationally expensive PDE simulation can be swapped for a more efficient polynomial chaos approximation. After the swap, the graph topology remains the same but the model will now be faster to evaluate. Swapping nodes also allows modelers to try different physics or model traits and ultimately to find the combination of nodes that they believe matches reality. Using MUQ in this setting is similar to using Legos. There are many pieces available, and modelers can slowly build up very sophisticated models by combining the right pieces. Each time a piece is changed, only one line of code will need to change. Adding new pieces into the graph also only requires one or two additional lines of code. Thus, it is easy for modelers to incrementally build up the model. Moreover, MUQ provides some python bindings that allow this all to be completed without recompiling.

6.2 Implementing Markov chain Monte Carlo

As we mentioned earlier, our goal for MUQ is to provide both a modeling framework and to facilitate easy algorithmic development. MUQ has many tools ranging from optimization to polynomial chaos to regression, but here we will focus on the structure of MUQ's MCMC stack. Just like decomposing a model into its main components facilitates easy model development, decomposing an MCMC algorithm into its fundamental components enables more efficient algorithm development.

The two main components of a general MCMC algorithm are the states of the chain and the transition kernel. For most Metropolis-Hastings style MCMC algorithms, the kernel can also be separated into the proposal and the accept-reject step of the Metropolis-Hastings rule. The MCMC algorithms in MUQ also follow this decomposition. We use objects (i.e., purely virtual `c++` classes) to define abstract interfaces for the chain, the kernel, and the proposal. The chain can call any available

kernel with the current location, and the kernel returns a new point in the parameter space as the next state in the chain. If the kernel is based on the Metropolis-Hastings rule, it will in turn call a proposal to get a proposed point, which is then accepted or rejected. The proposal knows nothing about the Metropolis-Hastings kernel and the Metropolis-Hastings kernel knows nothing about the chain. It is this complete decomposition that makes our MCMC implementation so extensible. For example, MUQ currently has 5 different proposals that we designed for use with the Metropolis-Hastings kernel. However, after implementing a new delayed rejection kernel, we were able to use any combination of the same 5 proposals to define variations of delayed rejection MCMC. In MUQ, changing the type of proposal simply involves changing a parameter flag, which encourages users to try many different algorithms and find the one that works the best for their problem. In fact, the only difference between running delayed rejection with an adaptive-Metropolis proposal, and running delayed rejection with a combination of MALA and a random walk, is a single string. This flexibility allowed us to define all of the algorithms used for comparison in Chapter 4 by changing one or two parameters.

The decomposition between the three MCMC components also prevents developers from having to "reinvent the wheel" every time a new algorithm comes out. If someone comes up with a new proposal mechanism, we need only write a new proposal class with the same interface as the other proposals. The bookkeeping of the chain and the accept/reject step of the kernel remain the same and no new code is required outside of the proposal. This type of easy implementation is not present in other MCMC packages that we are aware of and I believe our unique approach will enable algorithm developers to easily construct new algorithms that can immediately be used by modelers and other MUQ users. Moreover, because very little code changes with the additional algorithms, fewer bugs are likely to be introduced, saving valuable time for both developers and users.

6.3 Conclusions

By providing easy to use extensible interfaces for modelers, users, and algorithm developers, MUQ aims to reduce the lag time between algorithm improvements and widespread use. We believe that our code structure will enable this and will also allow researchers on both ends of the spectrum to focus on the problems that interest them, without having to worry about the messy bookkeeping and interfacing required to link advanced algorithms with complicated models.

While we only gave a high level overview of our modeling framework and MCMC implementation, MUQ has tools for many other areas in statistics and uncertainty quantification, including transport map construction, stochastic optimization, regression, polynomial chaos, random fields, Monte Carlo, and hierarchical matrices. These components currently range in maturity and ease of use, but we are constantly striving to improve MUQ's capabilities to help users efficiently "MUQ around" in uncertainty quantification.

MUQ is an open source library distributed at `https://bitbucket.org/mituq/`

`muq` under a BSD license.

Chapter 7

Conclusions and future work

Mathematically, Bayes' rule is an unassuming expression for combining multiple types of data. However, as we have shown, characterizing the Bayesian posterior can be a difficult computational task, often requiring an intractable number of expensive model evaluations. This thesis addresses this issue with new exact and approximate posterior sampling strategies, all of which rely on the efficient construction of transport maps.

Map construction

A fundamental tool used throughout this thesis is the *construction of transport maps from samples*. In Chapter 2 we introduced a map-construction techniques based on the solution of convex optimization problems. This technique can efficiently construct a single lower triangular map with good accuracy for low to moderate dimensional problems. However, the computational cost of accurately characterizing a large dimensional distribution can become intractable. Fortunately, the approximation quality of the map can be balanced against computational cost by strategically choosing the terms in the map parameterization. More approximate diagonal and separable maps are more efficient to construct in high dimensional spaces, but can yield crude approximations when used alone.

To overcome this issue, we introduced another map-construction technique based on the composition of simple maps interleaved with parameter space rotations. This layered approach has the potential to scale to very high dimensional problems. After studying the convergence of this approach on a small banana shaped distribution, we demonstrated its efficacy by characterizing a large Besov random field. In both the banana and Besov examples, random rotations that favor non-Gaussian directions or alternating between principal components and completely random rotations yielded the best approximation in the fewest number of layers.

Approximate sampling

Recall our first thesis objective:

- To create a framework for approximate Bayesian inference that uses prior samples and extensive offline computation to enable fast Bayesian inference in the

context of nonlinear inverse problems with computationally intensive forward models.

We have developed two approximate methods aimed at this goal: the mutliscale approach of Chapter 3, and the conditional map approach introduced in Chapter 5.

Our multiscale approach is applicable to large dimensional problems that exhibit multiple spatial or temporal scales. By describing the multiscale nature of the system as a conditional independence assumption, we were able to separate the posterior sampling problem into two stages: (i) sampling a small coarse scale posterior distribution, and (ii) generating fine scale posterior samples for each coarse posterior sample. Both of these stages are facilitated by a block lower triangular transport map that is constructed offline from joint prior samples of the fine and coarse variables. Using MsFEM to define a set of coarse parameters, we found good agreement between our multiscale method and a gold-standard MCMC method on a 100 dimensional problem. We then successfully applied our approach to a problem from porous media with more than 10000 spatially-distribution parameters. Such a large problem is intractable for existing MCMC methods, but our approach was able to generate approximate posterior samples in only a few hours.

Our second approximate approach, constructing the conditional map, was introduced in Chapter 5. This method uses joint samples of the data-parameter distribution to construct a block lower triangular map that can subsequently be used for near real-time posterior sampling. This approach is unique in that it requires no information other than the joint prior samples and can therefore perform almost all necessary computations before any particular data are observed. Because this method does not require any likelihood evaluations (only samples), it can be viewed as an approximate Bayesian computation (ABC) method. However, most ABC methods cannot exploit as much offline computation. Our results show that our conditional map can achieve good posterior accuracy and generate posterior samples in about two orders of magnitude less time than a standard MCMC method. We anticipate that further development of the layered block lower triangular map will extend this approach to higher dimensional problems. A theoretical analysis relating the accuracy of the joint map to the accuracy of the posterior conditional map will also be a useful guide for future development.

Exact sampling

While our first thesis objective was focused on approximate methods, our second objective was to produce an efficient but *exact* sampling method. Recall the exact thesis objective:

- To rigorously formulate a computationally efficient, broadly applicable, and statistically exact sampling scheme for non-Gaussian Bayesian inference problems: in particular, a scheme that targets posterior distributions with varying local correlation structures while requiring no derivative information.

Clearly, our adaptive MCMC technique from Chapter 4 completes this objective. We developed a provably ergodic MCMC scheme that can sample posterior distributions,

but can also be applied to more general probability densities. Transport maps again played a key roll in this algorithm. In the MCMC context, we used a transport map to capture posterior structure and subsequently define an efficient Metropolis-Hastings proposal mechanism. The algorithm adaptively constructs the transport map from previous MCMC states and requires no additional derivative information. Our results show multiple order-of-magnitude efficiency improvements over existing state-of-the-art MCMC samplers, even over samplers that exploit derivative information. This impressive performance stems from our unique combination of transport maps and independence proposals.

Our current implementation of this MCMC scheme use a single lower triangular map; however, the layered transport maps of Section 2.7 provide a natural path to extending this MCMC method to higher dimensions. Additionally, future applications may find posterior gradient or Hessian information useful for both guiding the reference space proposal, and for developing better approximations to the expectation in (2.9) using the sensitivity enhanced Monte Carlo methods of [38].

Smoothness extensions

The transport map formulations in Chapter 2 (and subsequently all the algorithms reviewed above) do not allow target distributions with point masses. Unfortunately, such distributions commonly arise in real geophysical applications (e.g., snow depth is zero during the summer but will have some positive distribution in the winter). Developing approaches for handling point masses in the sample-based construction of Chapter 2 could open up many new applications such as Bayesian system identification of river systems, or even real-time full waveform inversion from acoustic data.

Closing remarks

The utility of transport maps in our algorithms should be clear from the results given in Chapters 3, 4, and 5. However, we believe that our unique approach to constructing transport maps from samples is a more fundamentally useful tool that can be applied to many additional areas in statistics and uncertainty quantification. This means that future work building upon the ideas of Chapter 2 will impact many disciplines. To this end, our future work will focus on both developing a scalable parallel transport map implementation (via CUDA and MPI), and tackling larger dimensional problems by building upon the layered map ideas outlined in Section 2.7.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

Detailed MCMC convergence analysis

Section 4.4 provides a high level overview of the convergence properties for our map-accelerated MCMC algorithm. In this appendix, we elaborate on the descriptions of Section 4.4 with a more technical analysis. In particular, we elaborate on the proof of Theorem 1. Much of the analysis in this chapter follows the proof of Lemma 6.1 in [9].

A.1 Setting the stage: bounding the target proposal

To goal of this section is to find two zero mean Gaussian densities that bound the map-induced target density q_θ. We assume throughout this appendix that the target density $\pi(\theta)$ is finite, continuous, and super exponentially light (see (4.17)). We also assume the reference proposal density $q_r(r'|r)$ is a Gaussian random walk with a location bounded drift term $m(r)$ and fixed covariance Σ. Such a proposal takes the form

$$q_r(r'|r) = N(r + m(r), \Sigma). \tag{A.1}$$

For this proposal, we can follow [9] and show that there are two zero mean Gaussian densities g_1 and g_2 as well as two scalars k_1 and k_2 such that $0 < k_1 < k_2 < \infty$ and

$$k_1 g_1(r' - r) \leq q_r(r'|r) \leq k_2 g_2(r' - r). \tag{A.2}$$

Now, we will use (A.2) to bound the target space proposal q_θ. The following steps yield an upper bound

$$\begin{aligned} q_\theta(\theta'|\theta) &= q_r(\tilde{T}(\theta')|\tilde{T}(\theta))|\partial\tilde{T}(\theta')| && \text{(A.3)} \\ &\leq q_r(\tilde{T}(\theta')|\tilde{T}(\theta))d_{max}^D && \text{(A.4)} \\ &\leq k_2 g_2(\tilde{T}(\theta') - \tilde{T}(\theta))d_{max}^D && \text{(A.5)} \\ &\leq \left(k_2 d_{max}^D\right) g_2\left(d_{min}\left(\theta' - \theta\right)\right) && \text{(A.6)} \\ &= k_U g_U\left(\theta' - \theta\right), && \text{(A.7)} \end{aligned}$$

where g_U is another zero mean Gaussian. The step from (A.4) to (A.5) is a consequence of (A.2). While the step from (A.5) to (A.6) stems from the lower norm bound in (2.12) and because g_2 is a Gaussian with zero mean, which implies that $g_2(x_1) > g_2(x_2)$ when $\|x_1\| < \|x_2\|$. Notice that k_U does not depend on the particular coefficients of the map $\tilde{T}$, it only depends on the lower bound in (2.12). A similar process can be used to obtain the following lower bound

$$\begin{aligned} q_\theta(\theta'|\theta) &= q_r(\tilde{T}(\theta')|\tilde{T}(\theta))|\partial\tilde{T}(\theta')| \\ &\geq q_r(\tilde{T}(\theta')|\tilde{T}(\theta))d_{min}^D \\ &\geq k_1 g_1(\tilde{T}(\theta') - \tilde{T}(\theta))d_{min}^D \\ &\geq \left(k_1 d_{min}^D\right) g_1\left(d_{max}\left(\theta' - \theta\right)\right) \\ &= k_L g_L\left(\theta' - \theta\right). && \text{(A.8)} \end{aligned}$$

The bounds given in (A.7) and (A.8) are a fundamental component of the convergence proofs below. In fact, with these bounds in place, we can following the proof of Lemma 6.2 in [9] almost exactly.

A.2 SSAGE

To show that our adaptive scheme is ergodic, we need to show two things:

1. Diminishing adaptation
2. Containment

As we described in Section 4.4, the diminishing adaptation condition is easy to show for our approach under some mild continuity constraints. However, containment is more difficult to check. Directly assessing containment is difficult, but a more easily verifiable condition is Simultaneous Strongly Aperiodic Geometric Ergodicity (SSAGE). Importantly, [85] prove that SSAGE implies containment (also see [11] for a nice overview of containment in adaptive MCMC). SSAGE is similar to the usual minorization and drift conditions for non-adaptive MCMC, but applies to all proposals simultaneously. Recall that $\mathcal{X}_\theta$ is the set of all possible values for θ and γ are the coefficients defining the map $\tilde{T}(\theta) = \tilde{T}_\gamma(\theta)$. The formal definition of SSAGE is then

Definition 3 (SSAGE). *SSAGE is the condition that there is a set $C \in \mathcal{B}(\mathcal{X}_\theta)$, a function $V : \mathcal{X}_\theta \to [1, \infty)$ with $\sup_{x \in C} V(x) < \infty$, as well as three scalars $\delta > 0$, $\lambda < 1$, and $b < \infty$ such that the following two conditions hold:*

- *(Minorization) For each γ, there exists a measure $\nu_\gamma(\cdot)$ on C with $P_\gamma(x, A) \geq \delta\nu_\gamma(A)$ for all $x \in C$ and $A \in \mathcal{B}(\mathbb{R}^D)$.*
- *(Drift) $\int_{\mathbb{R}^D} V(x)P_\gamma(x, dx) \leq \lambda V(x) + bI_C(x)$ for all γ and x*

The following sections show that our adaptive approach satisfies these two conditions. Th map $\tilde{T}_\gamma(\theta)$ induces a target space proposal that is combined with the Metropolis-Hastings rule to obtain a transition kernel denoted by $P_\gamma(\theta, \cdot)$. Note that γ will be used as a subscript throughout the following text to indicate a dependence on a particular choice of map. In many cases our goal will be to construct results that do not depend on γ.

For the following analysis, assume $\pi(x) > 0$ for all finite x and let $V(x) = c_V\pi^{-\alpha}(x)$, where c_V is chosen so that $\min V(x) = 1$. Also, choose the set C to be a ball with radius $R_C > 0$, i.e., $C = B(0, R_C)$. Clearly, because we assume $\pi(x) > 0$, we immediately obtain $\sup_{x \in C} V(x) < \infty$.

A.3 Minorization

Our goal in this section is to find two things: (i) a scalar δ that does not depend on γ and (ii) a nontrivial measure $\nu_\gamma(\cdot)$ such that the following minorization condition holds

$$P_\gamma(x, \cdot) \geq \delta\nu_\gamma(\cdot). \tag{A.9}$$

To define δ, we refer to the form of the Metropolis-Hastings transition kernel given by

$$P_\gamma(x, dy) = \alpha_\gamma(x, y)q_{\theta,\gamma}(y|x)dy + r_\gamma(x)\delta_x(dy),$$

where

$$r_\gamma(x) = 1 - \int \alpha_\gamma(x, y)q_{\theta,\gamma}(y|x)dy,$$

and α is the Metropolis-Hastings acceptance rate. The acceptance rate is defined by

$$\alpha_\gamma(x, y) = \min\left\{1, \frac{\pi(y)q_{\theta,\gamma}(x|y)}{\pi(x)q_{\theta,\gamma}(y|x)}.\right\}$$

Let τ be the minimum acceptance rate over all $x, y \in C$ and over all of the map-induced proposal densities. In other words, τ is defined by

$$\tau = \inf_\gamma \inf_{x,y \in C} \min\left\{1, \frac{\pi(y)q_{\theta,\gamma}(x|y)}{\pi(x)q_{\theta,\gamma}(y|x)}\right\}. \tag{A.10}$$

Notice that lower bound in (A.8) ensures we always have a positive acceptance rate $\tau > 0$ because

$$\pi(y)q_{\theta,\gamma}(x|y) \geq \pi(y)k_L g_L(x-y) > 0 \quad \forall x, y \in C.$$

Now, this minimum acceptance rate can be substituted back into the transition kernel to obtain

$$\begin{aligned} P_\gamma(x, dy) &= \alpha_\gamma(x, y)q_{\theta,\gamma}(y|x)dy + r_\gamma(x)\delta_x(dy) \\ &\geq \tau q_{\theta,\gamma}(y|x)dy + r(x)\delta_x(dy). \end{aligned} \tag{A.11}$$

Again using our lower bound on $q_{\theta,\gamma}(y|x)$ from (A.8), we have

$$\begin{aligned} P_\gamma(x, dy) &\geq \tau k_L g_L(x-y)dy + r_\gamma(x)\delta_x(dy) \\ &\geq \tau k_L g_L(x-y)dy. \end{aligned} \tag{A.12}$$

Thus, for $x \in C$, we have a lower bound on $P_\gamma(x, dy)$ that does not depend on the map parameters γ. Now, we need to remove the dependence of the right hand side on x. Since g_L is a Gaussian density, $\inf_{z\in C} g_L(z-y) > 0$ and we can defined a new density g_{L2} that is not dependent on x. Mathematically, g_{L2} takes the form

$$g_{L2}(y) = \frac{\inf_{x\in C} g_L(x-y)}{\int_{y\in\mathbb{R}^D} \inf_{x\in C} g_L(x-y)dy}.$$

Using this expression yields

$$P_\gamma(x, dy) \geq \tau k_{L2} g_{L2}(y)dy, \tag{A.13}$$

where $k_{L2} = k_L \int_{y\in\mathbb{R}^D} \inf_{x\in C} g_L(x-y)dy$. It may now be tempting to directly use the right hand side of this expression to define the minimization measure ν. However, this expression is only valid for $x \in C$ and $dy \subset C$ and we need the minorization measure to be defined for all measurable sets in $\mathbb{R}^D$. Thus an alternative expression of ν is required. Fortunately, Rosenthal provides a nice example in [93] that can be adapted to this situation. First, set $\delta = \tau k_{L2}$ and define

$$\nu(A) = \frac{\int_{A\cap C} g_{L2}(y)dy}{\int_C g_{L2}(y)dy} \tag{A.14}$$

This expression defines a nontrivial measure and allows us to create a lower bound using (A.13) but on sets outside of C. Combining this expression with (A.13), we obtain

$$P_\gamma(x, A) \geq \delta\nu(A) \tag{A.15}$$

This provides the minimization component of the SSAGE condition. The next section discusses the more intricate drift component.

A.4 Drift

This section is shows that our adaptive algorithm satisfies the drift condition in the SSAGE definition. From the proof of Lemma 6.2 in [9], which resembles the proofs in [55], the following two conditions are equivalent to the SSAGE drift condition

$$\sup_x \sup_\gamma \frac{\int_{\mathbb{R}^D} V(y) P_\gamma(x, dy)}{V(x)} < \infty, \tag{A.16}$$

and

$$\limsup_{\|x\| \to \infty} \sup_\gamma \frac{\int_{\mathbb{R}^D} V(y) P_\gamma(x, dy)}{V(x)} < 1. \tag{A.17}$$

We will therefore satisfy the drift condition by satisfying both of these conditions. First, we will show a bound on $\frac{\int_{\mathbb{R}^D} V(y) P_\gamma(x, dy)}{V(x)}$ that ensures (A.16) is satisfied. The forthcoming simplifications will break the parameter space $\mathcal{X}_\theta$ into multiple regions. The regions are based on the set of guaranteed acceptance, which is given by

$$A_\gamma(x) = \left\{ y \in \mathbb{R}^D \quad | \quad \pi(y) q_{\theta,\gamma}(x|y) \geq \pi(x) q_{\theta,\gamma}(y|x) \right\}, \tag{A.18}$$

and the set of possible rejection, simply defined by

$$R_\gamma(x) = A_\gamma(x)^C \tag{A.19}$$

Now, recall our choice of drift function: $V(x) = c_V \pi^{-\alpha}(x)$ for $\alpha \in (0, 1)$. Plugging

this function into the argument of (A.16) and simplifying yields

$$
\begin{aligned}
\frac{\int_{\mathbb{R}^D} V(y) P_\gamma(x, dy)}{V(x)} &= \frac{\int_{\mathbb{R}^D} \pi^{-\alpha}(y) P_\gamma(x, dy)}{\pi^{-\alpha}(x)} \qquad \text{(A.20)} \\
&= \int_{\mathbb{R}^D} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} P_\gamma(x, dy) \\
&= \int_{A_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x) dy + \int_{R_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} \frac{\pi(y) q_{\theta,\gamma}(x|y)}{\pi(x) q_{\theta,\gamma}(y|x)} q_{\theta,\gamma}(y|x) dy \\
&\quad + \int_{R_\gamma(x)} \left(1 - \frac{\pi(y) q_{\theta,\gamma}(x|y)}{\pi(x) q_{\theta,\gamma}(y|x)}\right) q_{\theta,\gamma}(y|x) dy \\
&= \int_{R_\gamma(x)} q_{\theta,\gamma}(y|x) dy + \int_{A_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x) dy \\
&\quad + \int_{R_\gamma(x)} \left(\frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} - 1\right) \frac{\pi(y) q_{\theta,\gamma}(x|y)}{\pi(x) q_{\theta,\gamma}(y|x)} q_{\theta,\gamma}(y|x) dy \\
&\leq Q_{\theta,\gamma}(\theta, R_\gamma(x)) + \int_{A_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x) dy \\
&\quad + \int_{R_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} \frac{\pi(y) q_{\theta,\gamma}(x|y)}{\pi(x) q_{\theta,\gamma}(y|x)} q_{\theta,\gamma}(y|x) dy \\
&\leq 1 + \int_{A_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x) dy \\
&\quad + \int_{R_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} \frac{\pi(y) q_{\theta,\gamma}(x|y)}{\pi(x) q_{\theta,\gamma}(y|x)} q_{\theta,\gamma}(y|x) dy \qquad \text{(A.21)}
\end{aligned}
$$

Within the region of possible rejection $R_\gamma(x)$, the acceptance rates are all in $[0, 1)$,

which allows us to further bound (A.20) using (A.21) and the following algebra

$$\begin{aligned}
\frac{\int_{\mathbb{R}^D} V(y)P_\gamma(x,dy)}{V(x)} &\leq 1 + \int_{A_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy \\
&\quad + \int_{R_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} \frac{\pi(y)q_{\theta,\gamma}(x|y)}{\pi(x)q_{\theta,\gamma}(y|x)} q_{\theta,\gamma}(y|x)dy \\
&\leq 1 + \int_{A_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy + \int_{R_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy \\
&< 1 + \int_{A_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy + \int_{R_\gamma(x)} \frac{q_{\theta,\gamma}^{-\alpha}(y|x)}{q_{\theta,\gamma}^{-\alpha}(x|y)} q_{\theta,\gamma}(y|x)dy \\
&= 1 + \int_{A_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy + \int_{R_\gamma(x)} q_{\theta,\gamma}^{1-\alpha}(y|x) q_{\theta,\gamma}^{\alpha}(x|y)dy \\
&\leq 1 + \int_{A_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy \\
&\quad + \int_{R_\gamma(x)} \left(k_U g_U(y-x)\right)^{1-\alpha} \left(k_U g_U(x-y)\right)^{\alpha} dy \\
&= 1 + \int_{A_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy + k_U^2 \int_{R_\gamma(x)} g_U(y-x)dy \\
&= 1 + C_R + \int_{A_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy, \qquad \text{(A.22)}
\end{aligned}$$

where the expression in (A.22) is a consequence of the density upper bound in (A.7). Now consider region of guaranteed acceptance $A_\gamma(x)$. A similar application of (A.7) over this region yields

$$\begin{aligned}
\frac{\int_{\mathbb{R}^D} V(y)P_\gamma(x,dy)}{V(x)} &\leq 1 + C_R + \int_{A_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy \\
&= 1 + C_R + \int_{A_\gamma(x)} \frac{\pi^{\alpha}(x)}{\pi^{\alpha}(y)} q_{\theta,\gamma}(y|x)dy \\
&\leq 1 + C_R + \int_{A_\gamma(x)} \frac{q_{\theta,\gamma}^{\alpha}(x|y)}{q_{\theta,\gamma}^{\alpha}(y|x)} q_{\theta,\gamma}(y|x)dy \\
&= 1 + C_R + \int_{A_\gamma(x)} q_{\theta,\gamma}^{\alpha}(x|y) q_{\theta,\gamma}^{1-\alpha}(y|x)dy \\
&\leq 1 + C_R + k_U^2 \int_{A_\gamma(x)} g_U(x-y)dy \\
&= 1 + C_R + C_A \\
&< \infty. \qquad \text{(A.23)}
\end{aligned}$$

Thus, $\frac{\int_{\mathbb{R}^D} V(y)P_\gamma(x,dy)}{V(x)}$ is finite for all x and γ and we have satisfied (A.16). However,

we still need to show (A.17), i.e., that

$$\limsup_{\|x\|\to\infty} \sup_{\gamma} \frac{\int_{\mathbb{R}^D} V(y)P_\gamma(x,dy)}{V(x)} < 1. \tag{A.24}$$

To show this, we will first show that this ratio is less than rejection rate, i.e., that

$$\limsup_{\|x\|\to\infty} \sup_{\gamma} \frac{\int_{\mathbb{R}^D} V(y)P_\gamma(x,dy)}{V(x)} < \limsup_{\|x\|\to\infty} \sup_{\gamma} \int_{R_\gamma(x)} q_{\theta,\gamma}(y|x)dy, \tag{A.25}$$

and then we will show that there is a strictly positive probability of accepting the proposal, which is mathematically stated as

$$\int_{R_\gamma(x)} q_{\theta,\gamma}(y|x)dy < 1. \tag{A.26}$$

Part 1

Our goal in this section is to show (A.25). As we have done before, the left hand side of (A.25) can be broken in the $A_\gamma(x)$ portion and the $R_\gamma(x)$ portion to obtain

$$\begin{aligned}\frac{\int_{\mathbb{R}^D} V(y)P_\gamma(x,dy)}{V(x)} &= \int_{A_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy + \int_{R_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} \frac{\pi(y)q_{\theta,\gamma}(x|y)}{\pi(x)q_{\theta,\gamma}(y|x)} q_{\theta,\gamma}(y|x)dy \\ &\quad + \int_{R_\gamma(x)} \left(1 - \frac{\pi(y)q_{\theta,\gamma}(x|y)}{\pi(x)q_{\theta,\gamma}(y|x)}\right) q_{\theta,\gamma}(y|x)dy. \end{aligned} \tag{A.27}$$

To show that this expression is less than the rejection rate $\int_{R_\gamma(x)} q_{\theta,\gamma}(y|x)dy$ as $\|x\| \to \infty$, we will show that the first two integrals (A.27) go to zero as $\|x\| \to \infty$ and that the last integral is bounded by $\int_{R_\gamma(x)} q_{\theta,\gamma}(y|x)dy$. During this derivation, it will prove useful to further decompose $A_\gamma(x)$ and $R_\gamma(x)$ into subsets. This decomposition will be based on a ball of radius R around x, $B(x,R)$, where R implicitly depends on some tolerance $\epsilon > 0$ through the requirement that

$$\int_{B(x,R)} g_U(y-x)dy \geq 1 - \epsilon. \tag{A.28}$$

In addition to this ball, the decomposition of $A_\gamma(x)$ and $R_\gamma(x)$ will also be based on the sets $C_{\pi(x)}$ and $C_{\pi(x)}(u)$ defined by

$$C_{\pi(x)} = \{y \in \mathbb{R}^D : \pi(y) = \pi(x)\}, \tag{A.29}$$

and for $u > 0$,

$$C_{\pi(x)}(u) = \{y + sn(y) : y \in C_{\pi(x)}, -u \leq s \leq u\}. \tag{A.30}$$

You can think of $C_{\pi(x)}$ as a single contour of the target density and $C_{\pi(x)}(u)$ as a narrow region surrounding that contour. Now, we can define the following non-

overlapping subsets of $A_\gamma(x)$ and $R_\gamma(x)$

$$\begin{aligned} A_1(x) &= A_\gamma(x) \cap B(x,R)^c \\ A_2(x) &= A_\gamma(x) \cap B(x,R) \cap C_{\pi(x)}(u) \\ A_3(x) &= A_\gamma(x) \cap B(x,R) \cap C_{\pi(x)}(u)^c \\ R_1(x) &= R_\gamma(x) \cap B(x,R)^c \\ R_2(x) &= R_\gamma(x) \cap B(x,R) \cap C_{\pi(x)}(u) \\ R_3(x) &= R_\gamma(x) \cap B(x,R) \cap C_{\pi(x)}(u)^c. \end{aligned} \tag{A.31}$$

Note that $A_\gamma(x) = A_1(x) \cup A_2(x) \cup A_3(x)$ and $R_\gamma(x) = R_1(x) \cup R_2(x) \cup R_3(x)$. Using this new subsets, reconsider the $A_\gamma(x)$ component of (A.27). We can rewrite the integral from (A.27) as

$$\begin{aligned} \int_{A_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy &= \int_{A_1(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy + \int_{A_2(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy \\ &\quad + \int_{A_3(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy. \end{aligned} \tag{A.32}$$

Recall that we are trying to make show that this integral goes to zero as $\|x\| \to \infty$, so that we can subsequently bound (A.17). Thus, our momentary goal is to show that each of the integrals in (A.32) goes to zero as $\|x\| \to \infty$. Recall the bound used in (A.22), which is repeated here

$$\frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x) \leq k_U^2 g_U(x-y).$$

Applying this bound to the first two integrals in (A.32) yields

$$\begin{aligned} \int_{A_1(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy &\leq k_U^2 \int_{A_1(x)} g_U(x-y)dy \\ \int_{A_2(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy &\leq k_U^2 \int_{A_2(x)} g_U(x-y)dy. \end{aligned} \tag{A.33}$$

A result from [55] will allow us to bound the right hand sides of (A.33) and (A.33) by first bounding the size of $A_1(x)$ and $A_2(x)$.

In the proof of theorem 4.1 from [55], the authors show that for a large radius r_1 and $\|x\| \geq r_1$, the Lebesgue measure of $C_{\pi(x)}(u) \cap B(x,R)$ is bounded by:

$$\lambda\left(C_{\pi(x)}(u) \cap B(x,R)\right) \leq \frac{u}{R}\left(\frac{\|x\|+R}{\|x\|-R}\right)^{D-1} \lambda\left(C_{\pi(x)}(u) \cap B(x,3R)\right) \tag{A.34}$$

where D is the dimension of x and λ is the Lebesgue measure. Notice that as $\|x\| \to \infty$, the right hand size of this expression becomes u/R. Thus, by using the absolute continuity of the Gaussian density $g_U(y-x)$ with respect to Lebesgue measure, we

can use the expression in (A.34) to find a width u and a larger radius $r_2 > r_1$ such that for any $\epsilon > 0$

$$\int_{C_{\pi(x)}(u)\cap B(x,R)} g_U(y-x)dy \leq \epsilon \quad \text{for } \|x\| \geq r_2. \tag{A.35}$$

By applying (A.35) to (A.33) and (A.28) to (A.33) we obtain the upper bounds

$$\begin{aligned} \int_{A_1(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy &\leq k_2^2\epsilon \\ \int_{A_2(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy &\leq k_2^2\epsilon. \end{aligned} \tag{A.36}$$

Applying these expressions to (A.32), shows that as $\|x\| \to \infty$, we can choose a contour width u such that

$$\lim_{\|x\|\to\infty} \int_{A_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy = \lim_{\|x\|\to\infty} \int_{A_3(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy$$

This takes care of the $A_1(x)$ and $A_2(x)$ portions of (A.32). However, we still need to show that the $A_3(x)$ portion goes to zero, i.e., $\lim_{\|x\|\to\infty} \int_{A_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy = 0$. To show this, we will simply show that the size of the set $A_3(x)$ goes to zero as $\|x\| \to \infty$. This will require the super-exponential characteristic of the target density $\pi(\theta)$.

Continuing to follow the proof of lemma 6.2 from [9], for any $r > 0$ and $a > 0$, define

$$d_r(a) = \sup_{\|x\|\geq r} \frac{\pi\left(x + a\frac{x}{\|x\|}\right)}{\pi(x)}$$

As [9] points out, $d_r(a) \to 0$ as $r \to \infty$ because the target density $\pi(\theta)$ is super-exponential. For a particular $r_3 < \infty$, this convergence provides the following bound (taken from [9])

$$\int_{A_3(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x) \leq d_{r_3}(\delta) \quad \text{for all } \|x\| \geq r_3 + R. \tag{A.37}$$

Now, combining (A.36), (A.36), and (A.37), we can finally show

$$\lim_{\|x\|\to\infty} \int_{A_\gamma(x)} \frac{\pi^{-\alpha}(y)}{\pi^{-\alpha}(x)} q_{\theta,\gamma}(y|x)dy = 0. \tag{A.38}$$

Moreover, the same reasoning that got us from (A.32) to (A.38) can be used over the possible rejection region to show that

$$\lim_{\|x\|\to\infty} \int_{R_\gamma(x)} \frac{\pi(y)q_{\theta,\gamma}(x|y)}{\pi(x)q_{\theta,\gamma}(y|x)} q_{\theta,\gamma}(y|x)dy = 0. \tag{A.39}$$

Looking back at (A.27), we can see their is only one remaining part of that integral. This remaining part is given by

$$\lim_{\|x\|\to\infty} \frac{\int_{\mathbb{R}^D} V(y)P_\gamma(x,dy)}{V(x)} = \lim_{\|x\|\to\infty} \int_{R_\gamma(x)} q_{\theta,\gamma}(y|x)dy. \tag{A.40}$$

Thus, we have satisfied the first part in (A.25). The section below will take this result and will verify (A.17) by showing that this term is bounded by 1.

Part 2

Our goal in this section is to show that $\lim_{\|x\|\to\infty} \int_{R_\gamma(x)} q_{\theta,\gamma}(y|x)dy < 1$. Notice that this is equivalent to having a nonzero acceptance probability at the point x. To verify this condition for our adaptive MCMC scheme, we will show that there is a measurable set in the guaranteed acceptance region $W(x) \subset A_\gamma(x)$. Because $W(x)$ is in the guaranteed acceptance region, any y proposed in $W(x)$ will be accepted with probability 1. Also notice that $W(x)$ does not depend on the map coefficients γ.

For a small ball of radius R around x, the following condition holds

$$\begin{aligned} \inf_{y\in B(x,R)} \inf_{\gamma} \frac{q_{\theta,\gamma}(x|y)}{q_{\theta,\gamma}(y|x)} &\geq \inf_{y\in B(x,R)} \frac{k_L g_L(x-y)}{k_U g_U(y-x)} &\text{(A.41)} \\ &\geq c_0, &\text{(A.42)} \end{aligned}$$

for some $c_0 > 0$. The expression in (A.41) is a result of g_L and g_U both having zero mean and positive variance. Now, the fact that $\pi(x)$ is super exponentially light means that for $u \in (0, R)$, there is a radius r_4 such that when $\|x\| > r_4$, we have

$$\pi\left(x - u\frac{x}{\|x\|}\right) \geq \frac{\pi(x)}{c_0}$$

This means that the acceptance probability for $x_1 = x - u\frac{x}{\|x\|}$ is 1 for any map coefficients γ. Mathematically, we have

$$\frac{\pi(x_1)}{\pi(x)}\frac{q_{\theta,\gamma}(x|x_1)}{q_{\theta,\gamma}(x_1|x)} \geq \frac{\pi(x_1)}{\pi(x)}c_0 \geq 1.$$

By our definition of the acceptance region $A_\gamma(x)$, this means that $x_1 \in A_\gamma(x)$. The single point x_1 has zero measure, so its existence does not mean that the rejection rate in (A.40) will be less that 1. We need to further show that there is a measurable set $W(x)$ around x_1. To show this, we will first give a definition of $W(x)$ and will then verify that $W(x) \subset A_\gamma(x)$. For a scalar ϵ arbitrarily small, let $W(x)$ be defined as

$$W(x) = \left\{ x_1 - a\zeta,\, 0 < a < R - u,\, \zeta \in S^{D-1},\, \left\| \zeta - \frac{x_1}{\|x_1\|} \right\| < \frac{\epsilon}{2} \right\},$$

where S^{D-1} is the unit sphere in $\mathbb{R}^D$ dimensions. Without the $\left\| \zeta - \frac{x_1}{\|x_1\|} \right\| < \frac{\epsilon}{2}$ restric-

tion, $W(x)$ would simply be $B(x_1, R-u) \setminus \{x_1\}$. However, this additional restriction forces the vector ζ to point in the same direction as x, which means $W(x)$ is a cone of points closer to the origin than x_1. Now, from the final paragraph of the proof of Lemma 6.2 in [9], we know the curvature condition from (4.18) ensures that the target density is larger in $W(x)$ than x_1. Since x_1 was accepted, this means that everything in $W(x)$ will also be accepted and that $W(x) \subseteq A_\gamma(x)$. This also implies that

$$\begin{aligned}
\lim_{\|x\|\to\infty} \int_{R_\gamma(x)} q_{\theta,\gamma}(y|x)dy &= \lim_{\|x\|\to\infty} \left(1 - \int_{A_\gamma(x)} q_{\theta,\gamma}(y|x)dy\right) \\
&\leq \lim_{\|x\|\to\infty} \left(1 - \int_{W_(x)} q_{\theta,\gamma}(y|x)dy\right) \\
&\leq 1. \qquad \text{(A.43)}
\end{aligned}$$

Notice that this expression guarantees (A.26), which subsequently verifies (A.17). Furthermore (A.16) is verified by (A.23) **so we have satisfied the drift condition!** Combining this with our proof that the minorization condition holds, we have verified that when using a Gaussian reference proposal with bounded mean, the SSAGE condition is satisfied for our adaptive map-accelerated MCMC scheme. This subsequently implies the containment condition and ultimately the ergodicity of our adaptive approach!

Bibliography

[1] J. Aarnes and Y. Efendiev, *Mixed multiscale finite element methods for stochastic porous media flows*, SIAM Journal on Scientific Computing, 30 (2008), pp. 2319–2339.

[2] B. Adams, L. Bauman, W. Bohnhoff, K. Dalbey, M. Ebeida, J. Eddy, M. Eldred, P. Hough, K. Hu, J. Jakeman, L. Swiler, , and D. Vigil, *DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 5.4 User's Manual*, Tech. Rep. Sandia Technical Report SAND2010-2183, Sandia National Laboratories, 2009.

[3] S.-i. Amari and H. Nagaoka, *Methods of Information Geometry*, Oxford University Press, 1993.

[4] C. Andrieu and E. Moulines, *On the ergodicity properties of some adaptive MCMC algorithms*, The Annals of Applied Probability, 16 (2006), pp. 1462–1505.

[5] C. Andrieu and J. Thoms, *A tutorial on adaptive MCMC*, Statistics and Computing, 18 (2008), pp. 343–373.

[6] G. Anthes, *Deep learning comes of age*, Communications of the ACM, 56 (2013), p. 13.

[7] a.R. Barron, *Universal approximation bounds for superpositions of a sigmoidal function*, IEEE Transactions on Information Theory, 39 (1993), pp. 930–945.

[8] T. Arbogast, *Numerical subgrid upscaling of two-phase flow in porous media*, Lecture Notes in Phyics, 552 (2000), pp. 35–49.

[9] Y. F. Atchadé, *An Adaptive Version for the Metropolis Adjusted Langevin Algorithm with a Truncated Drift*, Methodology and Computing in Applied Probability, 8 (2006), pp. 235–254.

[10] K. Bache and M. Lichman, *UCI Machine Learning Repository*, 2013.

[11] Y. Bai, G. Roberts, and J. Rosenthal, *On the Containment Condition for Adaptive Markov Chain Monte Carlo Algorithms*. 2009.

[12] J. M. Bardsley, A. Solonen, H. Haario, and M. Laine, *Randomize-then-optimize: a method for sampling from posterior distributions in nonlinear inverse problems*, tech. rep., 2014.

[13] J. Bear, *Dynamics of Fluids in Porous Media*, Dover Publications, 1988.

[14] A. J. Bell and T. J. Sejnowski, *An information-maximisation approach to blind separation and blind deconvolution*, Neural Computation, 7 (1995), pp. 1004–1034.

[15] N. Bonnotte, *From Knothe's rearrangement to Brenier's optimal transport map*, SIAM Journal on Mathematical Analysis, 45 (2013), pp. 64–87.

[16] Y. Brenier, *Polar Factorization and Monotone Rearrangement of Vector-Valued Functions*, Communications on Pure and Applied Mathematics, XLIV (1991), pp. 375–417.

[17] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, eds., *Handbook of Markov Chain Monte Carlo*, Chapman and Hall, 2011.

[18] T. Bui-Thanh and O. Ghattas, *Analysis of the Hessian for inverse scattering problems: I. Inverse shape scattering of acoustic waves*, Inverse Problems, 28 (2012), p. 055001.

[19] T. Bui-thanh and O. Ghattas, *Analysis of the Hessian for inverse scattering problems: II. Inverse medium scattering of acoustic waves*, Inverse Problems, 28 (2012), p. 055002.

[20] B. Calderhead and M. Girolami, *Statistical analysis of nonlinear dynamical systems using differential geometric sampling methods.*, Interface focus, 1 (2011), pp. 821–35.

[21] J.-F. Cardoso and B. H. Laheld, *Equivariant adaptive source separation*, IEEE Transactions on Signal Processing, 44 (1996), pp. 3017–3030.

[22] G. Carlier, A. Galichon, and F. Santambrogio, *from Knothe's transport to Brenier's map and a continuation method for optimal transport*, SIAM Journal on Mathematical ..., 41 (2010), pp. 2554–2576.

[23] J. A. Christen and C. Fox, *A General Purpose Sampling Algorithm for Continuous Distributions (the t-walk)*, Bayesian Analysis, 5 (2010), pp. 263–282.

[24] P. Comon, *Independent component analysis, A new concept?*, Signal Processing, 36 (1994), pp. 287–314.

[25] S. Cotter, G. Roberts, A. Stuart, and D. White, *MCMC Methods for functions: Modifying old algorithms to make them faster*, arXiv preprint arXiv:1202.0709, (2012).

[26] S. Cotter, G. Roberts, A. Stuart, and D. White, *MCMC Methods for functions: modifying old algorithms to make them faster*, Statistical Science, (2013).

[27] K. Csilléry, M. G. B. Blum, O. E. Gaggiotti, and O. François, *Approximate Bayesian Computation (ABC) in practice.*, Trends in ecology & evolution, 25 (2010), pp. 410–8.

[28] M. Dashti, S. Harris, and A. Stuart, *BESOV PRIORS FOR BAYESIAN INVERSE PROBLEMS*, arXiv preprint arXiv:1105.0889, X (2011), pp. 1–18.

[29] P. Dostert, Y. Efendiev, T. Hou, and W. Luo, *Coarse-gradient Langevin algorithms for dynamic data integration and uncertainty quantification*, Journal of Computational Physics, 217 (2006), pp. 123–142.

[30] W. E, B. Engquist, X. Li, W. Ren, and E. Vanden-Eijnden, *Heterogeneous multiscale methods: A review*, Communications in Computational Physics, 2 (2007), pp. 367–450.

[31] Y. Efendiev, T. Hou, and W. Luo, *Preconditioning markov chain monte carlo simulations using coarse-scale models*, SIAM Journal on Scientific Computing, 28 (2006), pp. 776–803.

[32] EPA, *Contaminated Sediment Remediation Guidance for Hazardous Waste Sites*, tech. rep., United States Environmental Protection Agency, 2005.

[33] I. Epanomeritakis, V. Akçelik, O. Ghattas, and J. Bielak, *A Newton-CG method for large-scale three-dimensional elastic full-waveform seismic inversion*, Inverse Problems, 24 (2008), p. 034015.

[34] G. Evensen, *Data Assimilation: The Ensemble Kalman Filter*, Springer, 2 ed., 2009.

[35] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*, Cengage Learning, 2 ed., 2002.

[36] R. Freeze, J. Massmann, L. Smith, T. Sperling, and B. James, *Hydrogeological decision analysis: 1. A framework*, Ground ..., 28 (1990), pp. 738–766.

[37] I. Friend and R. Westerfield, *Coskewness and capital asset pricing*, The Journal of Finance, XXXV (1980).

[38] V. Garg, *Coupled Flow Systems, Adjoint Techniques and Uncertainty Quantification*, PhD thesis, University of Texas at Austin, 2012.

[39] M. Girolami and B. Calderhead, *Riemann manifold Langevin and Hamiltonian Monte Carlo methods*, Journal Royal Statistical Society B, 73 (2011), pp. 1–37.

[40] C. Gommenginger, P. Thibaut, L. Fenoglio-Marc, G. Quartly, X. Deng, J. Gomez-Enri, P. Challenor, and Y. Gao, *Retracking Altimeter Waveforms Near the Coasts*, in Coastal Altimetry, S. Vignudelli, A. Kostianoy, P. Cipollini, and J. Benveniste, eds., Springer, 2011 editi ed., 2010, ch. 4, pp. 61–102.

[41] H. Haario, M. Laine, A. Mira, and E. Saksman, *DRAM : Efficient adaptive MCMC*, Statistics and Computing, 16 (2006), pp. 339–354.

[42] H. Haario, E. Saksman, and J. Tamminen, *An adaptive Metropolis algorithm*, Bernoulli, 7 (2001), pp. 223–242.

[43] M. Hairer, A. Stuart, and S. Vollmer, *Spectral gaps for a metropolis-hastings algorithm in infinite dimensions*, arXiv preprint arXiv:1112.1392, (2011), pp. 1–39.

[44] A. Hald, *A History of Parametric Statistical Inference from Bernoulli to Fisher, 1713-1935*, 2006.

[45] W. Hastings, *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika, 57 (1970), pp. 97–109.

[46] D. Higdon, H. Lee, and Z. Bi, *A Bayesian approach to characterizing uncertainty in inverse problems using coarse and fine-scale information*, Signal Processing, IEEE Transactions on, 50 (2002), pp. 389–399.

[47] M. Hoffman and A. Gelman, *The No-U-Turn Sampler : Adaptively setting path lengths in Hamiltonian Monte Carlo*, arXiv preprint arXiv:1111.4246, (2011), pp. 1–30.

[48] T. Homem-de Mello, *On rates of convergence to stochastic optimization problems under non-independent and identically distributed sampling*, SIAM Journal on Optimization, 19 (2008), pp. 524–551.

[49] K. Hornik, *Approximation capabilities of multilayer feedforward networks*, Neural networks, 4 (1991), pp. 251–257.

[50] K. Hornik, M. Stinchcombe, and H. White, *Multilayer feedforward networks are universal approximators*, Neural networks, 2 (1989), pp. 359–366.

[51] T. Hou and Y. Efendiev, *Multiscale Finite Element Methods: Theory and Applications*, Springer, 2009.

[52] T. J. Huges, G. R. Feijoo, L. Mazzei, and J.-B. Quincy, *The variational multiscale method – a paradigm for computational mechanics*, Computer methods in applied mechanics and engineering, 166 (1998), pp. 3–34.

[53] A. Hyvärinen and E. Oja, *Independent Component Analysis: Algorithms and Applications*, Neural Networks, 13 (2000), pp. 411–430.

[54] T. Jaakkola and M. I. Jordan, *Bayesian parameter estimation via variational methods*, Statistics and Computing, 10 (2000), pp. 25–37.

[55] S. r. F. Jarner and E. Hansen, *Geometric ergodicity of Metropolis algorithms*, Stochastic Processes and their Applications, 85 (2000), pp. 341–361.

[56] E. Jaynes, *Probability Theory: The Logic of Science*, Cambridge University Press, 2003.

[57] P. Jenny, S. Lee, and H. Tchelepi, *Adaptive fully implicit multi-scale finite-volume method for multi-phase flow and transport in heterogeneous porous media*, Journal of Computational Physics, 217 (2006), pp. 627–641.

[58] L. Johnson and C. Geyer, *Variable transformation to obtain geometric ergodicity in the random walk metropolis algorithm*, The Annals of Statistics, (2012), pp. 1–30.

[59] G. L. Jones, *On the Markov chain central limit theorem*, Probability Surveys, 1 (2004), pp. 299–320.

[60] R. Juanes and F.-X. Dub, *A locally conservative variational multiscale method for the simulation of porous media flow with multiscale source terms.*, Computational Geosciences, 12 (2008), pp. 273–295.

[61] S. Kim, R. Ma, D. Mesa, and T. Coleman, *Efficient Bayesian Inference Methods via Convex Optimization and Optimal Transport*, in IEEE Symposium on Information Theory, no. 6, 2013.

[62] A. Kleywegt, A. Shapiro, and T. Homem-de Mello, *The sample average approximation method for stochastic discrete optimization*, SIAM Journal on Optimization, 12 (2002), pp. 479–502.

[63] D. E. Knuth, *The Art of Compute Programming: Volume 2: Seminumerical Algorithms*, Addison-Wesley, Boston, MA, 3 ed., 1998.

[64] M. Lassas, E. Saksman, and S. Siltanen, *Discretization-invariant Bayesian inversion and Besov space priors*, arXiv preprint arXiv:0901.4220, (2009), pp. 1–41.

[65] K. Law, *Proposals Which Speed-Up Function Space MCMC*, arXiv preprint arXiv:1212.4767, (2012).

[66] O. Le Maitre and O. M. Knio, *Spectral methods for uncertainty quantification: with applications to computational fluid dynamics.*, Springer, 2010.

[67] W. Li and O. A. Cirpka, *Efficient geostatistical inverse methods for structured and unstructured grids*, Water Resources Research, 42 (2006), pp. 1944–7973.

[68] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer, New York, New York, USA, 2004.

[69] D. Lunn, C. Jackson, N. Best, A. Thomas, and D. Spiegelhalter, *The BUGS Book: A Practical Introduction to Bayesian Analysis*, Chapman and Hall/CRC, 2012.

[70] J. Marin, P. Pudlo, C. Robert, and R. Ryder, *Approximate Bayesian Computational methods*, Statistics and Computing, (2011), pp. 1–28.

[71] P. Marjoram, J. Molitor, V. Plagnol, and S. Tavare, *Markov chain Monte Carlo without likelihoods.*, Proceedings of the National Academy of Sciences of the United States of America, 100 (2003), pp. 15324–8.

[72] T. Marshall and G. Roberts, *An adaptive approach to Langevin MCMC*, Statistics and Computing, (2011).

[73] J. Martin, L. Wilcox, C. Burstedde, and O. Ghattas, *A stochastic newton MCMC method for large-scale statistical inverse problems with application to seismic inversion*, SIAM Journal on Scientific ..., 34 (2012), pp. 1460–1487.

[74] Y. M. Marzouk and H. N. Najm, *Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems*, Journal of Computational Physics, (2009).

[75] R. McCann, *Existence and Uniqueness of Monotone Measure-Preserving maps*, Duke Mathematical Journal, 80 (1995), pp. 309–323.

[76] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, *Equation of State Calculations by Fast Computing Machines*, The Journal of Chemical Physics, 21 (1953), p. 1087.

[77] A. Mira, *On Metropolis-Hastings algorithms with delayed rejection*, Metron, (2001).

[78] G. Monge, *Mémoire sur la théorie des déblais et de remblais*, in Histoire de l'Académie Royale des Sciences de Paris, avec les Mémoires de Mathématique et de Physique pour la même année, 1781, pp. 666–704.

[79] T. E. Moselhy, *Bayesian inference with optimal maps*, Journal of Computational Physics, (2011).

[80] R. M. Neal, *MCMC Using Hamiltonian Dynamics*, in Handbook of Markov Chain Monte Carlo, S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng, eds., Taylor and Francis, Boca Raton, FL, 2011, ch. 5, pp. 113–162.

[81] M. Parno, A. Davis, and P. Conrad, *MIT Uncertainty Quantification (MUQ) library*, 2014.

[82] C. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer, New York, New York, USA, 2004.

[83] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer, 2nd ed., 2004.

[84] G. Roberts, A. Gelman, and W. Gilks, *Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms*, The Annals of Applied Probability, 7 (1997), pp. 110–120.

[85] G. Roberts and J. S. Rosenthal, *Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms*, Journal of applied probability, 44 (2007), pp. 458–475.

[86] G. Roberts and O. Stramer, *Langevin diffusions and Metropolis-Hastings algorithms*, Methodology and computing in applied . . . , (2002), pp. 337–357.

[87] G. Roberts and R. Tweedie, *Exponential convergence of Langevin distributions and their discrete approximations*, Bernoulli, 2 (1996), p. 341.

[88] G. O. Roberts and J. S. Rosenthal, *Optimal scaling of discrete approximations to Langevin diffusions*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 60 (1998), pp. 255–268.

[89] ——, *Optimal scaling for various Metropolis-Hastings algorithms*, Statistical Science, 16 (2001), pp. 351–367.

[90] L. Rockwood, *Introduction to Population Ecology*, Wiley-Blackwell, 1 ed., 2006.

[91] M. Rosenblatt, *Remarks on a Multivariate Transformation*, The Annals of Mathematical Statistics, 23 (1952), pp. 470–472.

[92] J. S. Rosenthal, *A first look at rigorous probability theory.*, World Scientific Publishing, Singapore, 2006.

[93] J. S. Rosenthal, *Markov Chain Monte Carlo Algorithms: Theory and Practice*, in Monte Carlo and Quasi-Monte Carlo Methods 2008, P. L' Ecuyer and A. B. Owen, eds., no. Mcmc, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 157–169.

[94] R. Rosenthal, *GAMS, A user's guide*, tech. rep., 2014.

[95] S. a. Sisson, Y. Fan, and M. M. Tanaka, *Sequential Monte Carlo without likelihoods.*, Proceedings of the National Academy of Sciences of the United States of America, 104 (2007), pp. 1760–5.

[96] Stan Development Team, *Stan: A c++ library for probability and sampling, version 2.4*, 2014.

[97] G. Strang and G. Fix, *An analysis of the Finite Element Method, 2nd ed.*, Wellesly-Cambridge, 2008.

[98] S. Strogatz, *Nonlinear Dynamics and Chaos*, Westview Press, 2001.

[99] A. B. Sullivan, D. M. Snyder, and S. a. Rounds, *Controls on biochemical oxygen demand in the upper Klamath River, Oregon*, Chemical Geology, 269 (2010), pp. 12–21.

[100] a. M. Vershik, *Long History of the Monge-Kantorovich Transportation Problem*, The Mathematical Intelligencer, 35 (2013), pp. 1–9.

[101] V. V. Vesselinov, D. O. Malley, and D. Katzman, *Robust decision analysis for environmental management of groundwater contamination sites*, arXiv preprint arXiv:1311.6014, (2013).

[102] C. Villani, *Optimal Transport: Old and New*, Springer-Verlag, 2009.

[103] J. A. Vrugt, C. ter Braak, C. Diks, B. A. Robinson, J. M. Hyman, and D. Higdon, *Accelerating Markov Chain Monte Carlo Simulation by Differential Evolution with Self-Adaptive Randomized Subspace Sampling*, Mathematical Modeling And Analysis, 10 (2009), pp. 273–290.

[104] B. Wagner and S. Gorelick, *Reliable aquifer remediation in the presence of spatially variable hydraulic conductivity: From data to design*, Water Resources Research, 25 (1989), pp. 2211–2225.

[105] J. Wan and N. Zabaras, *A Bayesian approach to multiscale inverse problems using the sequential Monte Carlo method*, Inverse Problems, 27 (2011), p. 105004.

[106] H. Wilf, *A global bisection algorithm for computing the zeros of polynomials in the complex plane*, Journal of the Association for Computing Machinery, 25 (1978), pp. 415–420.

[107] U. Wolff, *Monte Carlo errors with less errors .*, Computer Physics Communications, 156 (2004), pp. 143–153.

[108] D. Xiu and G. Karniadakis, *The Wiener-Askey Polynomial Chaos for stochastic differential equations*, SIAM Journal on Scientific Computing, 24 (2002), pp. 619–644.