Accelerating Bayesian Inference in Computationally Expensive Computer Models Using Local and Global Approximations

by

Patrick Raymond Conrad

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2014

Author .....
Department of Aeronautics and Astronautics
March 10, 2014

Certified by .....
Youssef M. Marzouk
Associate Professor, Department of Aeronautics and Astronautics
Thesis Supervisor

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

Certified by .....
Patrick Heimbach
Senior Research Scientist, Department of Earth, Atmospheric and Planetary Sciences
Committee Member

Accepted by .....
Paulo C. Lozano
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Accelerating Bayesian Inference in Computationally Expensive Computer Models Using Local and Global Approximations

by

Patrick Raymond Conrad

Submitted to the Department of Aeronautics and Astronautics
on March 10, 2014, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Computational models of complex phenomena are an important resource for scientists and engineers. However, many state-of-the-art simulations of physical systems are computationally expensive to evaluate and are *black box*—meaning that they can be run, but their internal workings cannot be inspected or changed. Directly applying uncertainty quantification algorithms, such as those for forward uncertainty propagation or Bayesian inference, to these types of models is often intractable because the analyses use many evaluations of the model. Fortunately, many physical systems are well behaved, in the sense that they may be efficiently approximated with a modest number of carefully chosen samples. This thesis develops global and local approximation strategies that can be applied to black-box models to reduce the cost of forward uncertainty quantification and Bayesian inference.

First, we develop an efficient strategy for constructing global approximations using an orthonormal polynomial basis. We rigorously construct a Smolyak pseudospectral algorithm, which uses sparse sample sets to efficiently extract information from loosely coupled functions. We provide a theoretical discussion of the behavior and accuracy of this algorithm, concluding that it has favorable convergence characteristics. We make this strategy efficient in practice by introducing a greedy heuristic that adaptively identifies and explores the important input dimensions, or combinations thereof. When the approximation is used within Bayesian inference, however, it is difficult to translate the theoretical behavior of the global approximations into practical controls on the error induced in the resulting posterior distribution.

Thus, the second part of this thesis introduces a new framework for accelerating MCMC algorithms by constructing local surrogates of the computational model within the Metropolis-Hastings kernel, borrowing ideas from deterministic approximation theory, optimization, and experimental design. Exploiting useful convergence

characteristics of local approximations, we prove the ergodicity of our approximate Markov chain and show that it samples asymptotically from the exact posterior distribution of interest. Our theoretical results reinforce the key observation underlying this work: when the likelihood has some local regularity, the number of model evaluations per MCMC step can be greatly reduced, without incurring significant bias in the Monte Carlo average. We illustrate that the inference framework is robust and extensible by describing variations that use different approximation families, MCMC kernels, and computational environments. Our numerical experiments demonstrate order-of-magnitude reductions in the number of forward model evaluations used in representative ODE or PDE inference problems, in both real and synthetic data examples.

Finally, we demonstrate the local approximation algorithm by performing parameter inference for the ice-ocean coupling in Pine Island Glacier, Antarctica. This problem constitutes a challenging domain for inference and an important application in climate science. We perform simulated inference, comparing synthetic data to predictions from the MIT General Circulation Model, a state-of-the-art ocean simulation. The results reveal some information about parameter sensitivity, but we ultimately conclude that richer data is necessary to constrain the model parameters. In this example, applying our approximation techniques reduced the cost of the inference by a factor of five to ten, taking weeks instead of months, providing evidence that our techniques can make Bayesian inference on large-scale computational models more tractable.

Thesis Supervisor: Youssef M. Marzouk
Title: Associate Professor, Department of Aeronautics and Astronautics

Committee Member: Karen E. Willcox
Title: Professor, Department of Aeronautics and Astronautics

Committee Member: Patrick Heimbach
Title: Senior Research Scientist, Department of Earth, Atmospheric and Planetary Sciences

Acknowledgments

An undertaking as challenging as a thesis is only made possible by the support of a large community, and I am grateful to the many people who helped me along the way. First, I would like to thank Prof. Youssef Marzouk for his constant support during the last few years. He introduced me to fascinating problems in uncertainty quantification and then gave me freedom to explore my interests. Without his guidance and insight, I could not have completed this work.

I would like to thank my committee, Dr. Patrick Heimbach and Prof. Karen Willcox, and thesis readers, Prof. Heikki Haario and Prof. Natesh Pillai, for their many contributions to this work. Patrick introduced me to glaciers and climate science, and made it possible to work on such a fascinating and challenging scientific problem. Karen's thoughtful comments focused and clarified my research and its presentation. Heikki's careful reading provided many useful ideas, and his previous work on the Delayed Rejection Adaptive Metropolis algorithm appears throughout my experiments. I am grateful to Natesh for his enthusiasm for my work, his suggestions from a statistician's perspective, and along with Aaron Smith, for developing the proofs of convergence of the local approximation framework.

I benefited from the many research discussions and support of everyone in the Uncertainty Quantification lab and reading group, and the wider Aerospace Computational Design Lab. I cannot name everyone, but you all helped create a wonderful environment to work in. I owe a special thanks to Tarek Moselhy for helping refine the earliest versions of this work. I would like to thank Matt Parno for his collaboration on the MUQ software, within which I implemented my algorithms. Eric Dow was kind enough to allow me to tinker with the cluster so it would work the way I wanted. Sophia Hasenfus helps to keep the group running smoothly, making all our lives easier.

I would like to thank my sponsors, who generously supported this work, enabling me to design the course of research as I saw fit. Specifically, I was supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program, the National Science Foundation Graduate Research Fellowship under Grant No. 1122374, and the Scientific Discovery through Advanced Computing (SciDAC) program funded by US Department of Energy, Office of Advanced Scientific Computing Research, under award number DE-SC0007099.

To my friends, roommates, and teammates, it has been wonderful to share my time in Cambridge with you. Thank you for keeping my life filled with fun, laughter, and adventure. Finally, I have to express my profound gratitude to my family for your tireless support. My grandparents encouraged a passion for learning that has carried me this far. My brother and sister have been constant allies in all aspects of my life. My parents generously invested their time and energy to give me the education, independence, and creativity that made all my achievements possible.

Contents

Chapter 1

Introduction

In a variety of scientific fields, researchers have developed detailed computational models that facilitate analysis of real-world systems; examples range from ocean modeling and atmospheric science to chemical kinetics. Broadly, the field of uncertainty quantification (UQ) is concerned with analyzing the predictions and implications of these computational models as they interact with randomness or uncertainty. Although increasing computational resources have dramatically improved the physical fidelity of available models, state-of-the-art models tend to remain expensive—and hence uncertainty quantification with these models can be prohibitively expensive. For example, consider that Monte Carlo simulation, a foundation for many uncertainty quantification algorithms, explores the response of a model by collecting many output samples; this process multiplies the computational cost of a single model by several orders of magnitude and makes Monte Carlo analyses infeasible for many systems of interest.

Fortunately, it is often possible to mitigate the expense of UQ analyses (e.g., uncertainty propagation, global sensitivity analysis, Bayesian inference) by exploiting the fact that many models are "well-behaved" in the sense of exhibiting some *regularity* in their parameter dependence. So-called surrogate modeling techniques use this regularity to efficiently approximate the model, e.g., by constructing an interpolant or other functional approximation, and thereby reduce the overall cost of evaluating the model's behavior. We can understand the savings by splitting the algorithm into two phases: first, use the structure of the model to efficiently construct an accurate surrogate approximating the outputs of interest, and second, use the surrogate model to inexpensively perform the desired UQ analysis. In principle, if the surrogate is sufficiently accurate, the overall analysis can produce accurate results at dramatically reduced cost. Moreover, the two phases need not be so distinctly separated, as we will show in subsequent chapters. This work thus aims to design algorithms that construct and use approximations to accelerate Bayesian inference, as motivated by an application to the Pine Island glacier in Antarctica.

1.1 Motivating example: Bayesian inference in the Pine Island Glacier setting

The Pine Island Glacier (PIG) is an important outlet along the Amundsen coast of the western Antarctic Ice Sheet (WAIS), which has become a focus for researching the interaction of the ice and ocean systems [76, 62]. Recent efforts have modeled the ocean flow in the cavity under Pine Island Ice Shelf and its thermal coupling to the floating ice shelf within the MIT General Circulation Model (MITgcm), a state of the art ocean simulator [29]. However, there are significant uncertainties in some parameter values of this model; to make the model useful, we must infer these parameters from observed data, in what is sometimes referred to as a *calibration* procedure.

Classical approaches typically solve an optimization problem to select a single set of parameters to use with the computational model [67], and while such techniques are widely used, the selection of a single parameter set can be limiting. In domains where data is precious and the model is complex, such as the Pine Island setting, we argue that Bayesian inference provides a richer analysis. Bayesian inference produces a distribution over the parameters, where the probability of a parameter set is interpreted as our degree of belief in the parameters values. This distribution provides insights into how much information is contained in the data, how certain the estimates are, and what parameters of the system remain coupled.

Unfortunately, performing Bayesian inference on computationally expensive models, such as the MITgcm, is typically infeasibly expensive. Bayesian inference is commonly performed with Markov chain Monte Carlo (MCMC) algorithms, which require many evaluations of the ocean model to compare the model outputs to observations. Our objective for this work is to develop approximation algorithms that can be used to accelerate Bayesian inference, as performed with MCMC, which can be applied to the PIG problem.

To formally state the problem, begin by assuming that we would like to infer a vector of parameters $\theta \in \Theta \subseteq \mathbb{R}^n$. We have observed data $\mathbf{d} \in \mathbb{R}^d$, and a forward model $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^d$ that simulates the data for an input parameter configuration, which is assumed to be a computationally expensive model. The computational task in Bayesian inference is to draw samples from a posterior distribution that specifies our belief in various possible parameters. The posterior distribution is written with unnormalized density:

$$p(\theta|\mathbf{d}) \propto \mathcal{L}(\theta|\mathbf{d}, \mathbf{f}(\theta))p(\theta)$$

where the likelihood, $\mathcal{L} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$, is a misfit density that compares the real and simulated data, and the prior distribution, $p : \mathbb{R}^d \to \mathbb{R}^+$, specifies our knowledge of the parameter values before observing the data. MCMC algorithms repeatedly evaluate $p(\theta|\mathbf{d})$ and therefore require many runs of our expensive forward model, $\mathbf{f}$, which becomes computationally intractable. Surrogate approaches ameliorate this cost by constructing an approximation of the forward model, $\tilde{\mathbf{f}}$, that is accurate, yet inexpensive to construct and use, and then use $\tilde{\mathbf{f}}$ during inference.

1.2 Properties of approximation algorithms for Bayesian inference

While a full description of the prior work on integrating surrogates into Bayesian inference is postponed until later sections, we now highlight some important properties of surrogate algorithms that help to shape the context of our work. In an effort to select in a setting that is broadly applicable, we focus on approximations of *black box models* using *adaptive* approximation algorithms. Approximation methods for computational models can broadly be classified as *intrusive* or *non-intrusive*. Intrusive approximations assume that the governing equations of a model may be inspected and modified for the purposes of approximation. For example, reduced basis methods and other projection-based reduced order models project the governing equations onto a lower-dimensional subspace of the system state, and solve these modified equations [74, 82]. These approaches can be quite sophisticated and efficient, but are not applicable in the general case.

The MITgcm is an example of a black box model: it can be run at desired inputs, but it is a large, complex computational package, so it neither feasible to analyze the structure of the model, nor to rewrite the solver. Therefore, we handle it non-intrusively, assuming that we may only probe it by running the model and observing the outputs pointwise. Electing to limit this work to non-intrusive models makes it possible to link these techniques with essentially any computational model scientists might provide. Common families of approximations used in a non-intrusive setting include polynomial approximations, radial basis function approximation, and Gaussian process regression, several of which are explored in this work.

We classify approximations as either *local* or *global* : we consider an approximation to be local when only a limited collection of nearby samples is used to construct the approximation at a particular point, and call it global when every available sample is used, regardless of proximity. Global approximations can leverage high-order expansions to provide rapid convergence over the entire input space, but typically impose strict requirements on the regularity of the function over the entire space. For example, global polynomials converge rapidly for smooth functions, but their convergence rate deteriorates rapidly if the functions or its derivatives are discontinuous anywhere. In contrast, local approximations typically converge more slowly, but do so under much looser conditions. Some useful methods do not fall cleanly in either category, but are hybrids: two examples from the Gaussian process literature are treed GPs [47], which divide up the space over which models are constructed, or compactly supported covariance kernels [68], which create global fits where a measure of locality is imposed.

Adaptivity is an important tool to making algorithms useful in practice, as it helps circumvent paradoxical situations where the algorithm can only be efficient if the answer is already known. Controlling the deployment of adaptivity is complex, however, and this work deals with several important features of integrating adaptivity into surrogate-based inference algorithms. First, the adaptivity of the approximation can occur either during a pre-processing step or during the inference itself. Second,

adaptation can occur with respect to the prior or the posterior; the prior is simpler to work with, but the posterior is the actual target of interest. Third, there are many features that an adaptive algorithm might seek to exploit, of which, we consider two options: (1) identifying which parameters are important or tightly coupled can assist in the efficient allocation of resources, or (2) the posterior often exhibits significant concentration within the prior when the data is informative, allowing the approximation to focus on this small region of parameter space. Either exploits detailed knowledge about the problem that is not generally accessible *a priori*. Adaptivity makes it possible for the algorithm to discern the structure of the problem and to tailor its ongoing efforts based on what it learns, which is a key feature of the methods we propose.

1.3 Bayesian inference and global approximations

In a non-intrusive setting, a known strategy for reducing the overall cost of inference is to separate the inference into two stages. First, design a sample set $\mathcal{S} := \{\theta, \mathbf{f}(\theta)\}$, from which we construct a surrogate model $\tilde{\mathbf{f}}$. Second, substitute this surrogate, which may be cheaply evaluated, into the distribution to create an approximate posterior,

$$\tilde{p}(\theta|\mathbf{d}) \propto \mathcal{L}(\theta|\mathbf{d}, \tilde{\mathbf{f}}(\theta))p(\theta),$$

which can be used instead during MCMC. Since performing MCMC requires no further evaluations of the forward model and evaluating the surrogate is inexpensive by design, the cost of inference is essentially dictated by the size of the sample set, $|\mathcal{S}|$, needed to make an accurate approximation. Thus, simulating long MCMC chains, as is typically necessary to explore the posterior, becomes affordable whenever constructing the approximation is feasible. This approach has some formal justification, in that under mild assumptions, a good approximation of the forward model implies a good approximation of the posterior, and hence the samples are useful [23].

Within this framework, we modify existing non-intrusive techniques for building global polynomial approximations, so-called polynomial chaos expansions, to build a highly adaptive algorithm while proving that it maintains theoretical properties necessary for its correctness. *Smolyak algorithms* are a general construction for extending one dimensional algorithms into tensor product spaces, and can be make highly efficient when the dimensions are loosely coupled by focusing on only the couplings that are important. However, such information is not typically known *a priori*, so the adaptive algorithm incrementally finds and explores the relevant couplings, allowing for highly customized approximations.

Polynomial chaos expansions are a useful tool for many tasks in uncertainty quantification, but are not ideally suited for use in inference. Although there are known rates of convergence of these approximations in some cases, which allows us to conclude that the approximate posterior converges in the limit, it is difficult to control the bias of a practical sampling algorithm. This difficulty arises because this strategy

creates two separate sources of error in the inference: the quality of the approximation and the finite length of the MCMC chain. Either error can be refined, by using more samples or simulating a longer chain, respectively, but it is not obvious how to efficiently tune the algorithms to collaborate. Furthermore, it is difficult to provide adaptation to the specific demands of the posterior; whenever the data is informative, the posteriors is constrained to a small region within the support of the prior, making approximation with respect to the entire prior inefficient.

1.4 Bayesian inference and local approximations

To address the weaknesses of the global approximation described above, the primary contribution of this work is a new framework for integrally combining local approximations into MCMC. The fundamental goal is to construct approximations that are adapted to the posterior instead of the prior. Since determining the structure of the posterior is the goal of the inference, it is naturally not feasible to perform this adaptation in a pre-processing step, as with global approximations. Instead, the construction of the approximation is interleaved with the inference, refining the regions the MCMC determines to be important. Instead of adapting to information about which dimensions are important, this work focuses on leveraging posterior concentration, aiming to build an approximation over only the small region supported by the posterior.

In this framework, the sample set is no longer fixed, but is indexed with the MCMC step, $\mathcal{S}_i$, and is possibly expanded by running the forward model at new points selected by an experimental design procedure at each step. This produces a sequence of approximate forward models, $\tilde{\mathbf{f}}_i$ and a corresponding sequence of approximation posterior densities $\tilde{p}_i(\theta|\mathbf{d})$. Previous work has considered interleaving the experimental design with MCMC in this fashion, but only during the initial, finite, burn-in portion of the MCMC chain, after which the approximation is fixed. Instead, we propose to continue this refinement forever, which allows us to take the novel and important theoretical step of proving that the sampler produces exact samples asymptotically. This theory also allows us to make claims about the error at finite length chains, providing a control over the bias lacking in the previous efforts.

The construction of this framework is facilitated by the shift to local approximations, that is, those which are only constructing using a small set of neighboring samples, rather than the entire set $\mathcal{S}_i$. Local approximations are appropriate because they work well with highly adapted, unstructured, multi-scale sample sets. Furthermore, they converge under loose conditions, which makes our proof of the asymptotic exactness of the sampler possible.

1.5 Summary of thesis contributions

The objective of this work is reduce the cost of performing Bayesian inference on realistic, computationally expensive forward models. We develop a novel, adaptive

algorithm for constructing theoretically sound, prior-adapted, global polynomial approximations. Then we develop a framework for using local surrogates in inference that automatically refines the surrogate based on the specific needs of the inference. Furthermore, we prove that a representative instance of the framework asymptotically (as the number of MCMC samples approaches infinity) draws samples from the exact posterior distribution, recovering a standard convergence result for MCMC even in an approximation-based setting. We demonstrate the robustness of our framework in practice by exploring a variety of instances of this algorithm, including differing samplers and underlying approximation families. Finally, this approximation type is applied to a difficult real-world inference problem, the Pine Island Glacier setting. We summarize the contributions of this work as follows:

- Develop an adaptive and non-intrusive algorithm for constructing global polynomial chaos expansions, proving that it maintains favorable theoretical error properties and showing substantial efficiency gains due to the adaptive scheme.
- Describe a novel framework for using local approximations within MCMC and a corresponding experimental design algorithm tailored to inference.
- Prove that the MCMC framework based on local approximations produces asymptotically exact samples.
- Demonstrate that the local approximation framework can be used successfully with a variety of MCMC samplers and approximation families, and that it can incorporate derivative information and parallel chains. Show order of magnitude efficiency gains on representative problems.
- Apply local approximation methods to perform inference in the sub-ice shelf cavity setting of Pine Island Glacier using the MIT General Circulation Model (MITgcm) as the forward model.

1.6 Organization

The thesis is organized as follows. Chapter 2 develops an adaptive, theoretically sound approach for constructing non-intrusive polynomial chaos expansions, which is shown to perform well in numerical experiments. Chapter 3 motivates the shift to a local approximations and develops our framework for integrating them into MCMC. It also proves the exactness of the approximate samplers and demonstrates their effectiveness on numerical examples. Then, Chapter 4 further explores the robustness of the framework by proposing variations on the algorithm, exploring alternate approximation types, other samplers, and parallelism, and shows their usefulness with numerical experiments. Next, Chapter 5 develops an inference problem in the Pine Island Glacier setting and applies the local approximation algorithms to perform inference. Finally, Chapter 6 summarizes this work and discusses potential future research.

Chapter 2

Adaptive Smolyak pseudospectral approximations[1]

A central issue in the field of uncertainty quantification is understanding the response of a model to random inputs. When model evaluations are computationally intensive, techniques for *approximating* the model response in an efficient manner are essential. Approximations may be used to evaluate moments or the probability distribution of a model's outputs, or to evaluate sensitivities of model outputs with respect to the inputs [73, 117, 107]. Approximations may also be viewed as *surrogate models* to be used in optimization [89] or inference [80], replacing the full model entirely.

Often one is faced with black box models that can only be evaluated at designated input points. We will focus on constructing multivariate polynomial approximations of the input-output relationship generated by such a model; these approximations offer fast convergence for smooth functions and are widely used. One common strategy for constructing a polynomial approximation is interpolation, where interpolants are conveniently represented in Lagrange form [4, 118]. Another strategy is projection, particularly orthogonal projection with respect to some inner product. The results of such a projection are conveniently represented with the corresponding family of orthogonal polynomials [11, 73, 119]. When the inner product is chosen according to the input probability measure, this construction is known as the (finite dimensional) polynomial chaos expansion (PCE) [44, 104, 33]. Interpolation and projection are closely linked, particularly when projection is computed via discrete model evaluations. Moreover, one can always realize a change of basis [38] for the polynomial resulting from either operation. Here we will favor orthogonal polynomial representations, as they are easy to manipulate and their coefficients have a useful interpretation in probabilistic settings.

This chapter discusses *adaptive Smolyak pseudospectral approximation*, an accurate and computationally efficient approach to constructing multivariate polynomial chaos expansions. Pseudospectral methods allow the construction of polynomial approximations from point evaluations of a function [11, 9]. We combine these methods

[1] The material in this chapter is adapted from [20].

with *Smolyak's algorithm*, a general strategy for sparse approximation of linear operators on tensor product spaces, which saves computational effort by weakening the assumed coupling between the input dimensions. Gerstner & Griebel [43] and Hegland [52] developed adaptive variants of Smolyak's algorithm for numerical integration and illustrated the effectiveness of on-the-fly heuristic adaptation. We extend their approach to the pseudospectral approximation of functions. Adaptivity is expected to yield substantial efficiency gains in high dimensions—particularly for functions with anisotropic dependence on input parameters and functions whose inputs might not be strongly coupled at high order.

Previous attempts to extend pseudospectral methods to multivariate polynomial approximation with sparse model evaluations employed ad hoc approaches that are not always accurate. A common procedure has been to use sparse quadrature, or even dimension-adaptive sparse quadrature, to evaluate polynomial coefficients directly [115, 73]. This leads to at least two difficulties. First, the truncation of the polynomial expansion must be specified independently of the quadrature grid, yet it is unclear how to do this, particularly for anisotropic and generalized sparse grids. Second, unless one uses excessively high-order quadrature, significant aliasing errors may result. Constantine *et al.* [22] provided the first clear demonstration of these aliasing errors and proposed a Smolyak algorithm that does not share them. That work also demonstrated a link between Smolyak pseudospectral approximation and an extension to Lagrange interpolation called *sparse interpolation*, which uses function evaluations on a sparse grid and has well characterized convergence properties [83, 6].

The first half of this chapter performs a theoretical analysis, placing the solution from [22] in the broader context of Smolyak constructions, and explaining the origin of the observed aliasing errors for general (e.g., anisotropic) choices of sparse grid and quadrature rule. We do so by using the notion of polynomial exactness, without appealing to interpolation properties of particular quadrature rules. We establish conditions under which tensorized approximation operators are exact for particular polynomial inputs, then apply this analysis to the specific cases of quadrature and pseudospectral approximation; these cases are closely related and facilitate comparisons between Smolyak pseudospectral algorithms and direct quadrature. Section 2.1 develops *computable* one-dimensional and tensorized approximations for these settings. Section 2.2 describes general Smolyak algorithms and their properties, yielding our principal theorem about the polynomial exactness of Smolyak approximations, and then applies these results to quadrature and pseudospectral approximation. Section 2.3 compares the Smolyak approach to conventional direct quadrature. Our error analysis of direct quadrature shows why the approach goes wrong and allows us to draw an important conclusion: in almost all cases, direct quadrature is not an appropriate method for constructing polynomial expansions and should be superseded by Smolyak pseudospectral methods.

These results provide a rigorous foundation for *adaptivity*, which is the second focus of this chapter. Adaptivity makes it possible to harness the full flexibility of Smolyak algorithms in practical settings. Section 2.4 introduces a fully adaptive algorithm for

Smolyak pseudospectral approximation, which uses a single tolerance parameter to drive iterative refinement of both the polynomial approximation space and the corresponding collection of model evaluation points. As the adaptive method is largely heuristic, Section 2.5 demonstrates the benefits of this approach with numerical examples.

2.1 Full tensor approximations

Tensorization is a common approach for lifting one-dimensional operators to higher dimensions. Not only are tensor products computationally convenient, but they provide much useful structure for analysis. In this section, we develop some essential background for computable tensor approximations, then apply it to problems of (i) approximating integrals with numerical quadrature; and (ii) approximating projection onto polynomial spaces with pseudospectral methods. In particular, we are interested in analyzing the errors associated with these approximations and in establishing conditions under which the approximations are *exact*.

2.1.1 General setting

Consider a collection of one-dimensional linear operators $\mathcal{L}^{(i)}$, where (i) indexes the operators used in different dimensions. In this work, $\mathcal{L}^{(i)}$ will be either an integral operator or an orthogonal projector onto some polynomial space. We can extend a collection of these operators into higher dimensions by constructing the tensor product operator

$$\mathcal{L}^{(\mathbf{d})} := \mathcal{L}^{(1)} \otimes \cdots \otimes \mathcal{L}^{(d)}. \tag{2.1}$$

The one-dimensional operators need not be identical; the properties of the resulting tensor operator are constructed independently from each dimension. The bold parenthetical superscript refers to the tensor operator instead of the constituent one-dimensional operators.

As the true operators are not available computationally, we work with a convergent sequence of computable approximations, $\mathcal{L}_m^{(i)}$, such that

$$\|\mathcal{L}^{(i)} - \mathcal{L}_m^{(i)}\| \to 0 \text{ as } m \to \infty \tag{2.2}$$

in some appropriate norm. Taking the tensor product of these approximations provides an approximation to the full tensor operator, $\mathcal{L}_{\mathbf{m}}^{(\mathbf{d})}$, where the level of the approximation may be individually selected in each dimension, so the tensor approximation is identified by a multi-index $\mathbf{m}$. Typically, and in the cases of interest in this work, the tensor approximation will converge in the same sense as the one-dimensional approximation as all components of $\mathbf{m} \to \infty$.

An important property of approximation algorithms is whether they are *exact* for some inputs; characterizing this set of inputs allows us to make useful statements at finite order.

Definition 2.1.1 (Exact Sets)**.** *For an operator $\mathcal{L}$ and a corresponding approximation $\mathcal{L}_m$, define the exact set as $\mathcal{E}(\mathcal{L}_m) := \{f : \mathcal{L}(f) = \mathcal{L}_m(f)\}$ and the half exact set $\mathcal{E}_2(\mathcal{L}_m) := \{f : \mathcal{L}(f^2) = \mathcal{L}_m(f^2)\}$.*

The half exact set will help connect the exactness of a quadrature rule to that of the closely related pseudospectral operators. This notation is useful in proving the following lemma, which relates the exact sets of one-dimensional approximations and tensor approximations.

Lemma 2.1.2. *If a tensor approximation $\mathcal{L}_{\mathbf{m}}^{(\mathbf{d})}$ is constructed from one-dimensional approximations $\mathcal{L}_m^{(i)}$ with known exact sets, then*

$$\mathcal{E}(\mathcal{L}_{m_1}^{(1)}) \otimes \cdots \otimes \mathcal{E}(\mathcal{L}_{m_d}^{(d)}) \subseteq \mathcal{E}(\mathcal{L}_{\mathbf{m}}^{(\mathbf{d})}) \tag{2.3}$$

Proof. It is sufficient to show that the approximation is exact for an arbitrary monomial input $f(\mathbf{x}) = f^{(1)}(x^{(1)})f^{(2)}(x^{(2)})\cdots f^{(d)}(x^{(d)})$ where $f^{(i)}(x^{(i)}) \in \mathcal{E}(\mathcal{L}_{m_i}^{(i)})$, because we may extend to sums by linearity:

$$\begin{aligned} \mathcal{L}_{\mathbf{m}}^{(\mathbf{d})}(f^{(1)}\cdots f^{(d)}) &= \mathcal{L}_{m_1}^{(1)}(f^{(1)}) \otimes \cdots \otimes \mathcal{L}_{m_d}^{(d)}(f^{(d)}) \\ &= \mathcal{L}^{(1)}(f^{(1)}) \otimes \cdots \otimes \mathcal{L}^{(d)}(f^{(d)}) = \mathcal{L}^{(\mathbf{d})}(f). \end{aligned}$$

The first step uses the tensor product structure of the operator and the second uses the definition of exact sets. □

2.1.2 Multi-indices

Before continuing, we must make a short diversion to multi-indices, which provide helpful notation when dealing with tensor problems. A multi-index is a vector $\mathbf{i} \in \mathbb{N}_0^d$. An important notion for multi-indices is that of a *neighborhood*.

Definition 2.1.3 (Neighborhoods of multi-indices)**.** *A* forward neighborhood *of a multi-index $\mathbf{k}$ is the multi-index set $n_f(\mathbf{k}) := \{\mathbf{k} + \mathbf{e}_i : \forall i \in \{1 \dots d\}\}$, where $\mathbf{e}_i$ are the canonical unit vectors. The* backward neighborhood *of a multi-index $\mathbf{k}$ is the multi-index set $n_b(\mathbf{k}) := \{\mathbf{k} - \mathbf{e}_i : \forall i \in \{1 \dots d\}, \mathbf{k} - \mathbf{e}_i \in \mathbb{N}_0^d\}$.*

Smolyak algorithms rely on multi-index sets that are *admissible*.

Definition 2.1.4 (Admissible multi-indices and multi-index sets)**.** *A multi-index $\mathbf{k}$ is admissible with respect to a multi-index set $\mathcal{K}$ if $n_b(\mathbf{k}) \subseteq \mathcal{K}$. A multi-index set $\mathcal{K}$ is admissible if every $\mathbf{k} \in \mathcal{K}$ is admissible with respect to $\mathcal{K}$.*

Two common admissible multi-index sets with simple geometric structure are *total order* multi-index sets and *full tensor* multi-index sets. One often encounters total order sets in the sparse grids literature and full tensor sets when dealing with tensor grids of points. The total order multi-index set $\mathcal{K}_n^t$ comprises those multi-indices that lie within a d-dimensional simplex of side length n:

$$\mathcal{K}_n^t := \{\mathbf{k} \in \mathbb{N}_0^d : \|\mathbf{k}\|_1 \leq n\} \tag{2.4}$$

The full tensor multi-index set $\mathcal{K}^f_{\mathbf{n}}$ is the complete grid of indices bounded term-wise by a multi-index $\mathbf{n}$:

$$\mathcal{K}^f_{\mathbf{n}} := \{\mathbf{k} \in \mathbb{N}_0^d : \forall i \in \{1 \dots d\},\, k_i < n_i\} \tag{2.5}$$

2.1.3 Integrals and quadrature

Let $X^{(i)}$ be an open or closed interval of the real line $\mathbb{R}$. Then we define the weighted integral operator in one dimension as follows:

$$\mathcal{I}^{(i)}(f) := \int_{X^{(i)}} f(x) w^{(i)}(x)\, dx \tag{2.6}$$

where $f : X^{(i)} \to \mathbb{R}$ is some real-valued function and $w^{(i)} : X^{(i)} \to \mathbb{R}^+$ is an integrable weight function. We may extend to higher dimensions by forming the tensor product integral $\mathcal{I}^{(\mathbf{d})}$, which uses separable weight functions and Cartesian product domains.

Numerical quadrature approximates the action of an integral operator $\mathcal{I}^{(i)}$ with a weighted sum of point evaluations. For some family of quadrature rules, we write the "level m" quadrature rule, comprised of $p^{(i)}(m) : \mathbb{N} \to \mathbb{N}$ points, as

$$\mathcal{I}^{(i)}(f) \approx \mathcal{Q}^{(i)}_m(f) := \sum_{j=1}^{p^{(i)}(m)} w^{(i)}_j f(x^{(i)}_j). \tag{2.7}$$

We call $p^{(i)}(m)$ the growth rate of the quadrature rule, and its form depends on the quadrature family; some rules only exist for certain numbers of points and others may be tailored, for example, to produce linear or exponential growth in the number of quadrature points with respect to the level.

Many quadrature families are exact if f is a polynomial of a degree $a^{(i)}(m)$ or less, which allows us to specify a well-structured portion of the exact set for these quadrature rules:

$$\mathbb{P}_{a^{(i)}(m)} \subseteq \mathcal{E}(\mathcal{Q}^{(i)}_m) \tag{2.8}$$

$$\mathbb{P}_{\text{floor}(a^{(i)}(m)/2)} \subseteq \mathcal{E}_2(\mathcal{Q}^{(i)}_m), \tag{2.9}$$

where $\mathbb{P}_a$ is the space of polynomials of degree a or less. It is intuitive and useful to draw the exact set as in Figure 2-1. For this work, we rely on quadrature rules that exhibit polynomial accuracy of increasing order, which is sufficient to demonstrate convergence for functions in L^2 [11].

Tensor product quadrature rules are straightforward approximations of tensor product integrals that inherit convergence properties from the one-dimensional case. The exact set of a tensor product quadrature rule includes the tensor product of the constituent approximations' exact sets, as guaranteed by Lemma 2.1.2 and depicted in Figure 2-2.

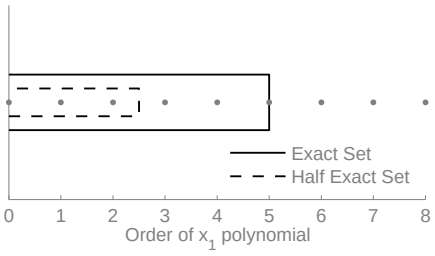


Figure 2-1: Consider, one-dimensional Gaussian quadrature rule with three points, $\mathcal{Q}_3^{(i)}$, which is exact for fifth degree polynomials. This diagram depicts the exact set, $\mathcal{E}(\mathcal{Q}_3^1)$, and half exact set, $\mathcal{E}_2(\mathcal{Q}_3^1)$, of this quadrature rule.

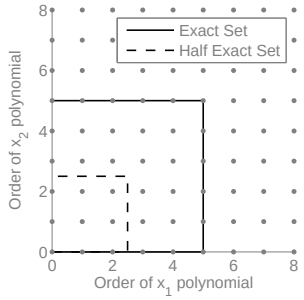


Figure 2-2: Consider a two dimensional quadrature rule constructed from three point Gaussian quadrature rules, $\mathcal{Q}_{(3,3)}^{(\mathbf{2})}$. This diagram depicts the exact set, $\mathcal{E}(\mathcal{Q}_{(3,3)}^{(\mathbf{2})})$, and half exact set, $\mathcal{E}_2(\mathcal{Q}_{(3,3)}^{(\mathbf{2})})$.

2.1.4 Polynomial projection

A polynomial chaos expansion approximates a function with a weighted sum of orthonormal polynomials [114, 119]. Let $\mathcal{H}^{(i)} := L^2\left(X^{(i)}, w^{(i)}\right)$ be the separable Hilbert space of square-integrable functions $f : X^{(i)} \to \mathbb{R}$, with inner product defined by the weighted integral $\langle f, g\rangle = \mathcal{I}^{(i)}(fg)$, and $w^{(i)}(x)$ normalized so that it may represent a probability density. Let $\mathbb{P}_n^{(i)}$ be the space of univariate polynomials of degree n or less. Now let $\mathcal{P}_n^{(i)} : \mathcal{H}^{(i)} \to \mathbb{P}_n^{(i)}$ be an orthogonal projector onto this subspace, written in terms of polynomials $\{\psi_j^{(i)}(x) : j \in \mathbb{N}_0\}$ orthonormal with respect to the inner product of $\mathcal{H}^i$:

$$\mathcal{P}_n^{(i)}(f) := \sum_{j=0}^{n} \left\langle f(x), \psi_j^{(i)}(x) \right\rangle \psi_j^{(i)}(x) = \sum_{i=0}^{n} f_j \psi_j^{(i)}(x). \tag{2.10}$$

The polynomial space $\mathbb{P}_n^{(i)}$ is of course the *range* of the projection operator. These polynomials are dense in $\mathcal{H}^{(i)}$, so the polynomial approximation of any $f \in \mathcal{H}^{(i)}$ converges in the L^2 sense as $n \to \infty$ [11, 119]. If $f \in \mathcal{H}^{(i)}$, the coefficients must satisfy $\sum_{i=0}^{\infty} f_i^2 < \infty$.

Projections with finite degree n omit terms of the infinite series, thus incurring *truncation error*. We can write this error as

$$\left\| f - \mathcal{P}_n^{(i)}(f) \right\|_2^2 = \left\| \sum_{j=n+1}^{\infty} f_j \psi_j^{(i)} \right\|_2^2 = \sum_{j=n+1}^{\infty} f_j^2 < \infty. \tag{2.11}$$

Hence, we may reduce the truncation error to any desired level by increasing n, removing terms from the sum in (2.11) [11, 54].

The d-dimensional version of this problem requires approximating functions in the Hilbert space $\mathcal{H}^{(\mathbf{d})} := \mathcal{H}^{(1)} \otimes \cdots \otimes \mathcal{H}^{(d)}$ via a tensor product basis of the univariate polynomials defined above:

$$\mathcal{P}_{\mathbf{n}}^{(\mathbf{d})}(f) = \sum_{i_1=0}^{n_1} \cdots \sum_{i_d=0}^{n_d} \langle f \Psi_{\mathbf{i}} \rangle \, \Psi_{\mathbf{i}} \tag{2.12}$$

where $\Psi_{\mathbf{i}}(\mathbf{x}) := \prod_{j=1}^{d} \psi_{i_j}^{(j)}\left(x^{(j)}\right)$. The multi-index $\mathbf{n}$ tailors the range of the projection to include a rectangular subset of polynomials.

As in the one-dimensional case, truncation induces error equal to the sum of the squares of the omitted coefficients, which we may similarly reduce to zero as $n_i \to \infty$, $\forall i$. The multivariate polynomial expansion also converges in an L^2 sense for any $f \in \mathcal{H}^{(\mathbf{d})}$ [11]

2.1.5 Aliasing errors in pseudospectral approximation

The inner products defining the expansion coefficients above are not directly computable. Pseudospectral approximation provides a practical non-intrusive algorithm by approximating these inner products with quadrature. Define the pseudospectral

approximation in one dimension as

$$\begin{aligned}\mathcal{S}_m^{(i)}(f) &:= \sum_{j=0}^{q^{(i)}(m)} \mathcal{Q}_m^{(i)}\left(f\psi_j^{(i)}\right)\psi_j^{(i)}(x) \\ &= \sum_{j=0}^{q^{(i)}(m)} \tilde{f}_j\psi_j^{(i)}(x) \end{aligned} \tag{2.13}$$

where $q^{(i)}(m)$ is the polynomial truncation at level m, to be specified shortly [11, 54]. Pseudospectral approximations are constructed around a level m quadrature rule, and are designed to include as many terms in the sum as possible while maintaining accuracy. Assuming that $f \in L^2$, we can compute the L^2 error between the pseudospectral approximation and an exact projection onto the same polynomial space:

$$\left\|\mathcal{P}^{(i)}_{q^{(i)}(m)}(f) - \mathcal{S}_m^{(i)}(f)\right\|_2^2 = \left\|\sum_{j=0}^{q^{(i)}(m)} f_j\psi_j^{(i)} - \sum_{k=0}^{q^{(i)}(m)} \tilde{f}_k\psi_k^{(1)}\right\|_2^2 = \sum_{j=0}^{q^{(i)}(m)} (f_j - \tilde{f}_j)^2 \tag{2.14}$$

This quantity is the *aliasing error* [11, 54]. The error is non-zero because quadrature in general only approximates integrals; hence each $\tilde{f}_i$ is an approximation of f_i. The pseudospectral operator also incurs truncation error, as before, which is orthogonal to the aliasing error. We can expand each approximate coefficient as

$$\begin{aligned}\tilde{f}_j &= \mathcal{Q}_m^{(i)}\left(f\psi_j^{(i)}\right) \\ &= \sum_{k=0}^{\infty} f_k \mathcal{Q}_m^{(i)}\left(\psi_j^{(i)}\psi_k^{(i)}\right) \\ &= \sum_{k=0}^{q^{(i)}(m)} f_k \mathcal{Q}_m^{(i)}\left(\psi_j^{(i)}\psi_k^{(i)}\right) + \sum_{l=q^{(i)}(m)+1}^{\infty} f_l \mathcal{Q}_m^{(i)}\left(\psi_j^{(i)}\psi_l^{(i)}\right). \end{aligned} \tag{2.15}$$

The first step substitutes in the polynomial expansion of f, which we assume is convergent, and rearranges using linearity. The second step partitions the sum around the truncation of the pseudospectral expansion. Although the basis functions are orthonormal, $\langle\psi_j^{(i)}, \psi_k^{(i)}\rangle = \delta_{jk}$, we cannot assume in general that the approximation $\mathcal{Q}_m^{(i)}\left(\psi_j^{(i)}\psi_k^{(i)}\right) = \delta_{jk}$. Now substitute (2.15) back into the aliasing error expression:

$$\sum_{j=0}^{q^{(i)}(m)} (f_j - \tilde{f}_j)^2 = \sum_{j=0}^{q^{(i)}(m)} \left(f_j - \sum_{k=0}^{q^{(i)}(m)} f_k\, \mathcal{Q}_m^{(i)}\left(\psi_j^{(i)}\psi_k^{(i)}\right) - \sum_{l=q^{(i)}(m)+1}^{\infty} f_l\, \mathcal{Q}_m^{(i)}\left(\psi_j^{(i)}\psi_l^{(i)}\right)\right)^2 \tag{2.16}$$

This form reveals the intimate link between the accuracy of pseudospectral approximations and the polynomial accuracy of quadrature rules. All aliasing is attributed to the inability of the quadrature rule to determine the orthonormality of the basis polynomials, causing one coefficient to corrupt another. The contribution of the first

two parenthetical terms on the right of (2.16) is called *internal aliasing*, while the third term is called *external aliasing*. Internal aliasing is due to inaccuracies in $\mathcal{Q}(\psi_j^{(i)}\psi_k^{(i)})$ when *both* $\psi_j^{(i)}$ and $\psi_k^{(i)}$ are included in the expansion, while external aliasing occurs when *only one* of these polynomials is included in the expansion. For many practical quadrature rules (and for all those used in this work), if $j \neq k$ and $\psi_j^{(i)}\psi_k^{(i)} \notin \mathcal{E}(\mathcal{Q})$, and hence the discrete inner product is not zero, then $\|\mathcal{Q}(\psi_j^{(i)}\psi_k^{(i)})\|_2$ is $\mathcal{O}\left(1\right)$ [108]. As a result, the magnitude of an aliasing error typically corresponds to the magnitude of the aliased coefficients.

In principle, both types of aliasing error are driven to zero by sufficiently powerful quadrature, but we are left to select $q^{(i)}(m)$ for a particular quadrature level m. External aliasing is $\mathcal{O}\left(1\right)$ in the magnitude of the truncation error, and thus it is driven to zero as long as $q^{(i)}(m)$ increases with m. Internal aliasing could be $\mathcal{O}\left(1\right)$ with respect to the function of interest, meaning that the procedure neither converges nor provides a useful approximation. Therefore, the obvious option is to include as many terms as possible while setting the internal aliasing to zero.

For quadrature rules with polynomial exactness, we may accomplish this by setting $q^{(i)}(m) = \text{floor}(a^{(i)}(m)/2)$. This ensures that the internal aliasing of $\mathcal{S}_m^{(i)}$ is zero, because $\forall j, k \leq q^{(i)}(m)$,
$\psi_j^{(i)}\psi_k^{(i)} \in \mathcal{E}(\mathcal{Q}_m^{(i)})$. Equivalently, a pseudospectral operator $\mathcal{S}_m^{(i)}$ using quadrature $\mathcal{Q}_m^{(i)}$ has a range corresponding to the half exact set $\mathcal{E}_2(\mathcal{Q}_m^{(i)})$. Alternatively, we may justify this choice by noting that it makes the pseudospectral approximation exact on its range, $\mathbb{P}^{(i)}_{q^{(i)}(m)} \subseteq \mathcal{E}(\mathcal{S}_m^{(i)})$.

Given this choice of $q^{(i)}(m)$, we wish to show that the pseudospectral approximation converges to the true function, where the magnitude of the error is as follows:

$$\left\|f - \mathcal{S}_m^{(i)}(f)\right\|_2^2 = \sum_{j=0}^{q^{(i)}(m)} \left(\sum_{k=q^{(i)}(m)+1}^{\infty} f_k \mathcal{Q}\left(\psi_j^{(i)}\psi_k^{(i)}\right) \right)^2 + \sum_{l=q^{(i)}(m)+1}^{\infty} f_l^2. \tag{2.17}$$

The two terms on right hand side comprise the external aliasing and the truncation error, respectively. We already know that the truncation error goes to zero as $q^{(i)}(m) \to \infty$. The external aliasing also vanishes for functions $f \in L^2$, as the truncated portion of f likewise decreases [108]. In the case of Gaussian quadrature rules, a link to interpolation provides precise rates for the convergence of the pseudospectral operator based on the regularity of f [11].

As with quadrature algorithms, our analysis of pseudospectral approximation in one dimension is directly extensible to multiple dimensions via full tensor products. We may thus conclude that $\mathcal{S}_{\mathbf{m}}^{(\mathbf{d})}$ converges to the projection onto the tensor product polynomial space in the same sense. The exact set follows Lemma 2.1.2, and hence the tensor product approximation inherits zero internal aliasing if suitable one-dimensional operators are used.

2.2 Smolyak algorithms

Thus far, we have developed polynomial approximations of multivariate functions by taking tensor products of one-dimensional pseudospectral operators. Smolyak algorithms avoid the exponential cost of full tensor products when the input dimensions are not fully coupled, allowing the use of a telescoping sum to blend different lower-order full tensor approximations.

Example 2.2.1. *Suppose that $f(x, y) = x^7 + y^7 + x^3y$. To construct a polynomial expansion with both the x^7 and y^7 terms, a full tensor pseudospectral algorithm would estimate all the polynomial terms up to x^7y^7, because tensor algorithms fully couple the dimensions. This mixed term is costly, requiring, in this case, an 8×8 point grid for Gaussian quadratures. The individual terms can be had much more cheaply, using 8×1, 1×8, and 4×2 grids, respectively. Smolyak algorithms help realize such savings in practice.*

This section reviews the construction of Smolyak algorithms and presents a new theorem about the exactness of Smolyak algorithms built around arbitrary admissible index sets. We apply these results to quadrature and pseudospectral approximation, allowing a precise characterization of their errors.

2.2.1 General Smolyak algorithms

As in Section 2.1, assume that we have for every dimension $i = 1 \dots d$ a convergent sequence $\mathcal{L}_k^{(i)}$ of approximations. Let $\mathcal{L}$ denote the collection of these sequences over all the dimensions. Define the difference operators

$$\Delta_0^{(i)} := \mathcal{L}_0^{(i)} = 0, \tag{2.18}$$

$$\Delta_n^{(i)} := \mathcal{L}_n^{(i)} - \mathcal{L}_{n-1}^{(i)}. \tag{2.19}$$

For any i, we may write the exact or "true" operator as the telescoping series

$$\mathcal{L}^{(i)} = \sum_{k=0}^{\infty} \mathcal{L}_k^{(i)} - \mathcal{L}_{k-1}^{(i)} = \sum_{k=0}^{\infty} \Delta_k^{(i)}. \tag{2.20}$$

Now we may write the tensor product of the exact operators as the tensor product of the telescoping sums, and interchange the product and sum:

$$\begin{aligned} \mathcal{L}^{(1)} \otimes \cdots \otimes \mathcal{L}^{(d)} &= \sum_{k_1=0}^{\infty} \Delta_{k_1}^{(1)} \otimes \cdots \otimes \sum_{k_d=0}^{\infty} \Delta_{k_d}^{(d)} \\ &= \sum_{\mathbf{k}=0}^{\infty} \Delta_{k_1}^{(1)} \otimes \cdots \otimes \Delta_{k_d}^{(d)} \end{aligned} \tag{2.21}$$

Smolyak's idea is to approximate the tensor product operator with truncations of this sum [102]:

$$A(\mathcal{K}, d, \mathcal{L}) := \sum_{\mathbf{k} \in \mathcal{K}} \Delta_{k_1}^{(1)} \otimes \cdots \otimes \Delta_{k_d}^{(d)}. \tag{2.22}$$

We refer to the multi-index set $\mathcal{K}$ as the *Smolyak multi-index set*, and it must be admissible for the sum to telescope correctly. Smolyak specifically suggested truncating with a total order multi-index set, which is the most widely studied choice. However, we can compute the approximation with any admissible multi-index set. Although the expression above is especially clean, it is not the most useful form for computation. We can reorganize the terms of (2.22) to construct a weighted sum of the tensor operators:

$$A(\mathcal{K}, d, \mathcal{L}) = \sum_{\mathbf{k} \in \mathcal{K}} c_{\mathbf{k}} \, \mathcal{L}_{k_1}^{(1)} \otimes \cdots \otimes \mathcal{L}_{k_d}^{(d)}, \tag{2.23}$$

where $c_{\mathbf{k}}$ are integer *Smolyak coefficients* computed from the combinatorics of the difference formulation. One can compute the coefficients through a simple iteration over the index set and use (2.22) to determine which full tensor rules are incremented or decremented. In general, these coefficients are non-zero near the leading surface of the Smolyak multi-index set, reflecting the mixing of the most accurate constituent full tensor approximations.

If each sequence of one-dimensional operators converges, then the Smolyak approximation converges to the tensor product of exact operators as $\mathcal{K} \to \mathbb{N}_0^d$. For the isotropic simplex index set, some precise rates of convergence are known with respect to the side length of the simplex [112, 113, 111, 100, 101]. Although general admissible Smolyak multi-index sets are difficult to study theoretically, they allow detailed customization to the anisotropy of a particular function.

2.2.2 Exactness of Smolyak algorithms

In the one-dimensional and full tensor settings, we have characterized approximation algorithms through their exact sets—those inputs for which the algorithm is precise. This section shows that if the constituent one-dimensional approximations have nested exact sets, Smolyak algorithms are the ideal blending of different full tensor approximations from the perspective of exact sets; that is, the exact set of the Smolyak algorithm contains the union of the exact sets of the component full tensor approximations. This result will facilitate subsequent analysis of sparse quadrature and pseudospectral approximation algorithms. This theorem and our proof closely follow the framework provided by Novak and Ritter [84, 85, 6], but include a generalization to arbitrary Smolyak multi-index sets.

Theorem 2.2.2. *Let $A(\mathcal{K}, d, \mathcal{L})$ be a Smolyak algorithm composed of linear operators with nested exact sets, i.e., with $m \leq m'$ implying that $\mathcal{E}(\mathcal{L}_m^{(i)}) \subseteq \mathcal{E}(\mathcal{L}_{m'}^{(i)})$ for $i = 1 \ldots d$,*

where $\mathcal{K}$ is admissible. Then the exact set of $A(\mathcal{K}, d, \mathcal{L})$ contains

$$\begin{aligned}\mathcal{E}\left(A(\mathcal{K}, d, \mathcal{L})\right) &\supseteq \bigcup_{\mathbf{k}\in\mathcal{K}} \mathcal{E}\left(\mathcal{L}_{k_1}^{(1)} \otimes \cdots \otimes \mathcal{L}_{k_d}^{(d)}\right) \\ &\supseteq \bigcup_{\mathbf{k}\in\mathcal{K}} \mathcal{E}(\mathcal{L}_{k_1}^{(1)}) \otimes \cdots \otimes \mathcal{E}(\mathcal{L}_{k_d}^{(d)}). \end{aligned} \tag{2.24}$$

Proof. We begin by introducing notation to incrementally build a multi-index set dimension by dimension. For a multi-index set $\mathcal{K}$ of dimension d, let the restriction of the multi-indices to the first i dimensions be $\mathcal{K}^{(i)} := \{\mathbf{k}_{1:i} = (k_1, \ldots, k_i) : \mathbf{k} \in \mathcal{K}\}$. Furthermore, define subsets of $\mathcal{K}$ based on the i^{th} element of the multi-indices, $\mathcal{K}_j^{(i)} := \{\mathbf{k}_{1:i} : \mathbf{k} \in \mathcal{K}^{(i)} \text{ and } k_{i+1} = j\}$. These sets are nested, $\mathcal{K}_j^{(i)} \supseteq \mathcal{K}_{j+1}^{(i)}$, because $\mathcal{K}$ is admissible. Also let $k_i^{\max}$ denote the maximum value of the i^{th} component of the multi-indices in the set $\mathcal{K}$.

Using this notation, one can construct $\mathcal{K}$ inductively,

$$\mathcal{K}^{(1)} = \{1, \ldots, k_1^{\max}\} \tag{2.25}$$

$$\mathcal{K}^{(i)} = \bigcup_{j=1}^{k_i^{\max}} \mathcal{K}_j^{(i-1)} \otimes j, \quad i = 2 \ldots d. \tag{2.26}$$

It is sufficient to prove that the Smolyak operator is exact for an arbitrary f with tensor structure, $f = f_1 \times \cdots \times f_d$. Suppose there exists a $\mathbf{k}^*$ such that $f \in \mathcal{E}(\mathcal{L}_{\mathbf{k}^*}^{(\mathbf{d})})$. We will show that if $\mathcal{K}$ is an admissible multi-index set containing $\mathbf{k}^*$, then $A(\mathcal{K}, d, \mathcal{L})$ is exact on f. We do so by induction on the dimension i of the Smolyak operator and the function.

First, consider the $i = 1$ case. $A(\mathcal{K}^{(1)}, 1, \mathcal{L}) = \mathcal{L}_{k_1^{\max}}^{(1)}$, where $k_1^{\max} \geq k_1^*$. Hence $\mathcal{E}(A(\mathcal{K}^{(1)}, 1, \mathcal{L})) = \mathcal{E}(\mathcal{L}_{k_1^{\max}}^{(1)})$.

For the induction step, we construct the $(i+1)$-dimensional Smolyak operator in terms of the i-dimensional operator:

$$A(\mathcal{K}^{(i+1)}, i+1, \mathcal{L}) = \sum_{j=1}^{k_{i+1}^{\max}} A(\mathcal{K}_j^{(i)}, i, \mathcal{L}) \otimes (\mathcal{L}_j^{(i+1)} - \mathcal{L}_{j-1}^{(i+1)}). \tag{2.27}$$

This sum is over increasing levels of accuracy in the $i+1$ dimension. We know the level required for the approximate operator to be exact in this dimension; this may be expressed as

$$\mathcal{L}_j^{(i+1)}(f_{i+1}) = \mathcal{L}_{j-1}^{(i+1)}(f_{i+1}) = \mathcal{L}^{(i+1)}(f_{i+1}) \text{ when } j - 1 \geq k_{i+1}^*. \tag{2.28}$$

Therefore the sum (2.27) can be truncated at the k_{i+1}^* term, as the differences of higher

terms are zero when applied to f:

$$A(\mathcal{K}^{(i+1)}, i+1, \mathcal{L}) = \sum_{j=1}^{k^*_{i+1}} A(\mathcal{K}^{(i)}_j, i, \mathcal{L}) \otimes (\mathcal{L}^{(i+1)}_j - \mathcal{L}^{(i+1)}_{j-1}). \tag{2.29}$$

Naturally, $\mathbf{k}^*_{1:i} \in \mathcal{K}^{(i)}_{k^*_{i+1}}$. By nestedness, $\mathbf{k}^*_{1:i}$ is also contained in $\mathcal{K}^{(i)}_j$ for $j \leq k^*_{i+1}$. The induction hypothesis then guarantees

$$f_1 \otimes \cdots \otimes f_i \in \mathcal{E}(A(\mathcal{K}^{(i)}_j, i, \mathcal{L})), \ \forall j \leq k^*_{i+1}. \tag{2.30}$$

Applying the $(i+1)$-dimensional Smolyak operator to the truncated version of f yields

$$\begin{aligned} & A(\mathcal{K}^{(i+1)}, i+1, \mathcal{L})(f_1 \otimes \cdots \otimes f_{i+1}) \\ = & \sum_{j=1}^{k^*_{i+1}} A(\mathcal{K}^{(i)}_j, i, \mathcal{L})(f_1 \otimes \cdots \otimes f_i) \otimes (\mathcal{L}^{(i+1)}_j - \mathcal{L}^{(i+1)}_{j-1})(f_{i+1}). \end{aligned} \tag{2.31}$$

Since each of the i-dimensional Smolyak algorithms is exact, by the induction hypothesis, we replace them with the true operators and rearrange by linearity to obtain

$$\begin{aligned} A(\mathcal{K}^{(i+1)}, i+1, \mathcal{L})(f_1 \otimes \cdots \otimes f_{i+1}) &= \mathcal{L}^{(\mathbf{i})}(f_1 \otimes \cdots \otimes f_i) \otimes \sum_{j=1}^{k^*_{i+1}} (\mathcal{L}^{(i+1)}_j - \mathcal{L}^{(i+1)}_{j-1})(f_{i+1}) \\ &= \mathcal{L}^{(\mathbf{i})}(f_1 \otimes \cdots \otimes f_i) \otimes \mathcal{L}^{(i+1)}_{k^*_{i+1}}(f_{i+1}). \end{aligned} \tag{2.32}$$

The approximation in the $i+1$ dimension is exactly of the level needed to be exact on the $(i+1)^{\text{th}}$ component of f. Then (2.32) becomes

$$\mathcal{L}^{(\mathbf{i})}(f_1 \otimes \cdots \otimes f_i) \otimes \mathcal{L}^{(i+1)}(f_{i+1}) = \mathcal{L}^{(\mathbf{i+1})}(f_1 \otimes \cdots \otimes f_{i+1}) \tag{2.33}$$

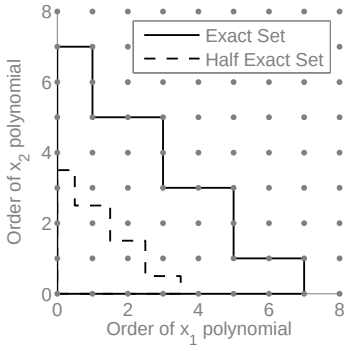
Thus the Smolyak operator is precise for f, and the claim is proven. □

2.2.3 Smolyak quadrature

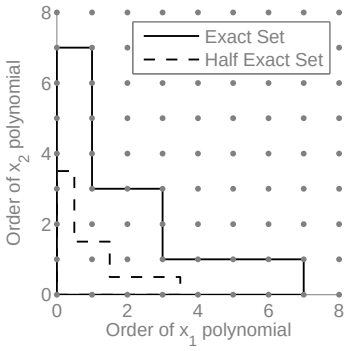
We recall the most familiar use of Smolyak algorithms, sparse quadrature. Consider a family of one-dimensional quadrature rules $\mathcal{Q}^{(i)}_k$ in each dimension $i = 1 \ldots d$; denote these rules by $\mathcal{Q}$. The resulting Smolyak algorithm is written as:

$$A(\mathcal{K}, d, \mathcal{Q}) = \sum_{\mathbf{k} \in \mathcal{K}} c_{\mathbf{k}} \mathcal{Q}^{(\mathbf{d})}_{\mathbf{k}}. \tag{2.34}$$

This approximation inherits its convergence from the one-dimensional operators. The set of functions that are exactly integrated by a Smolyak quadrature algorithm is described as a corollary of Theorem 2.2.2.



(a) The exact set for a level-four Smolyak quadrature in two dimensions, based on *linear* growth Gaussian quadrature rules.



(b) The exact set for a level-three Smolyak quadrature in two dimensions, based on *exponential* growth Gaussian quadrature rules.

Figure 2-3: The exact set diagram for two Smolyak quadrature rules, and the corresponding basis for a Smolyak pseudospectral approximation. $\mathcal{E}(\mathcal{Q})$ is shown in a solid line, $\mathcal{E}_2(\mathcal{Q})$ is the dashed line. The staircase appearance results from the superposition of rectangular full tensor exact sets.

Corollary 2.2.3. *For a sparse quadrature rule satisfying the hypotheses of Theorem 2.2.2,*

$$\mathcal{E}\left(A(\mathcal{K}, d, \mathcal{Q})\right) \supseteq \bigcup_{\mathbf{k}\in\mathcal{K}} \mathcal{E}(\mathcal{Q}_{\mathbf{k}}^{(\mathbf{d})}) \tag{2.35}$$

Quadrature rules with polynomial accuracy do have nested exact sets, as required by the theorem. An example of Smolyak quadrature exact sets is shown in Figure 2-3.

2.2.4 Smolyak pseudospectral approximation

Applying Smolyak's algorithm to pseudospectral approximation operators yields a sparse algorithm that converges under similar conditions as the one-dimensional operators from which it is constructed. This algorithm is written as

$$A(\mathcal{K}, d, \mathcal{S}) = \sum_{\mathbf{k}\in\mathcal{K}} c_{\mathbf{k}} \mathcal{S}_{\mathbf{k}}^{(\mathbf{d})}. \tag{2.36}$$

The Smolyak algorithm is therefore a sum of different full tensor pseudospectral approximations, where each approximation is built around the polynomial accuracy of a single full tensor quadrature rule. It is not naturally expressed as a set of formulas for the polynomial coefficients, because different approximations include different polynomials. The term $\Psi_{\mathbf{j}}$ is included in the Smolyak approximation if and only if $\exists \mathbf{k} \in \mathcal{K} : \Psi_{\mathbf{j}} \in \mathcal{E}_2(\mathcal{Q}_{\mathbf{k}}^{(\mathbf{d})})$. Here, $\mathcal{Q}_{\mathbf{k}}^{(\mathbf{d})}$ is the full tensor quadrature rule used by the full tensor pseudospectral approximation $\mathcal{S}_{\mathbf{k}}^{(\mathbf{d})}$. As in the full tensor case, the half exact

set of a Smolyak quadrature rule defines the range of the Smolyak pseudospectral approximation.

Once again, the Smolyak construction guarantees that the convergence of this approximation is inherited from its constituent one-dimensional approximations. Our choices for the pseudospectral operators ensure nestedness of the constituent exact sets, so we may use Theorem 2.2.2 to ensure that Smolyak pseudospectral algorithms are exact on their range.

Corollary 2.2.4. *If the constituent one-dimensional pseudospectral rules have no internal aliasing and satisfy the conditions of Theorem 2.2.2, then the resulting Smolyak pseudospectral algorithm has no internal aliasing.*

We additionally provide a theorem that characterizes the external aliasing properties of Smolyak pseudospectral approximation, which the next section will contrast with direct quadrature.

Theorem 2.2.5. *Let $\Psi_{\mathbf{j}}$ be a polynomial term included in the expansion provided by the Smolyak algorithm $A(\mathcal{K}, d, \mathcal{S})$, and let $\Psi_{\mathbf{j}'}$ be a polynomial term not included in the expansion. There is no external aliasing of $\Psi_{\mathbf{j}'}$ onto $\Psi_{\mathbf{j}}$ if any of the following conditions is satisfied: (a) there exists a dimension i for which $j_i' < j_i$; or (b) there exists a multi-index $\mathbf{k} \in \mathcal{K}$ such that $\Psi_{\mathbf{j}}$ is included in the range of $\mathcal{S}_{\mathbf{k}}^{(\mathbf{d})}$ and $\Psi_{\mathbf{j}'}\Psi_{\mathbf{j}} \in \mathcal{E}(\mathcal{Q}_{\mathbf{k}}^{(\mathbf{d})})$, where $\mathcal{Q}_{\mathbf{k}}^{(\mathbf{d})}$ is the quadrature rule used in $\mathcal{S}_{\mathbf{k}}^{(\mathbf{d})}$.*

Proof. If condition (a) is satisfied, then $\Psi_{\mathbf{j}}$ and $\Psi_{\mathbf{j}'}$ are orthogonal in dimension i, and hence that inner product is zero. Every quadrature rule that computes the coefficient $f_{\mathbf{j}}$ corresponding to basis term $\Psi_{\mathbf{j}}$ is accurate for polynomials of at least order $2\mathbf{j}$. Since $j_i' + j_i < 2j_i$, every rule that computes the coefficient can numerically resolve the orthogonality, and therefore there is no aliasing. If condition (b) is satisfied, then the result follows from the cancellations exploited by the Smolyak algorithm, as seen in the proof of Theorem 2.2.2. □

These two statements yield extremely useful properties. First, any Smolyak pseudospectral algorithm, regardless of the admissible Smolyak multi-index set used, has no internal aliasing; this feature is important in practice and not obviously true. Second, while there is external aliasing as expected, the algorithm uses basis orthogonality to limit which external coefficients can alias onto an included coefficient. The Smolyak pseudospectral algorithm is thus a practically "useful" approximation, in that one can tailor it to perform a desired amount of work while guaranteeing reliable approximations of the selected coefficients. Computing an accurate approximation of the function only requires including sufficient terms so that the truncation and external aliasing errors are small.

2.3 Comparing direct quadrature to Smolyak pseudospectral approximation

The current UQ literature often suggests a *direct quadrature* approach for constructing polynomial chaos expansions [116, 73, 32, 60]. In this section, we describe this approach and show that, in comparison to a true Smolyak algorithm, it is inaccurate or inefficient in almost all cases. Our comparisons will contrast the theoretical error performance of the algorithms and provide simple numerical examples that illustrate typical errors and why they arise.

2.3.1 Direct quadrature polynomial expansions

At first glance, direct quadrature is quite simple. First, choose a multi-index set $\mathcal{J}$ to define a truncated polynomial expansion:

$$f \approx \sum_{\mathbf{j} \in \mathcal{J}} \tilde{f}_{\mathbf{j}} \Psi_{\mathbf{j}}. \tag{2.37}$$

The index set $\mathcal{J}$ is typically admissible, but need not be. Second, select any d-dimensional quadrature rule $\mathcal{Q}^{(\mathbf{d})}$, and estimate every coefficient as:

$$\tilde{f}_{\mathbf{j}} = \mathcal{Q}^{(\mathbf{d})}(f \Psi_{\mathbf{j}}). \tag{2.38}$$

Unlike the Smolyak approach, we are left to choose $\mathcal{J}$ and $\mathcal{Q}^{(\mathbf{d})}$ independently, giving the appearance of flexibility. In practice, this produces a more complex and far more subtle version of the truncation trade-off discussed in Section 2.1. Below, we will be interested in selecting $\mathcal{Q}$ and $\mathcal{J}$ to replicate the quadrature points and output range of the Smolyak approach, as it provides a benchmark for achievable performance.

Direct quadrature does not converge for every choice of $\mathcal{J}$ and $\mathcal{Q}^{(\mathbf{d})}$; consider the trivial case where $\mathcal{J}$ does not grow infinitely. It is possible that including far too many terms in $\mathcal{J}$ relative to the polynomial exactness of $\mathcal{Q}^{(\mathbf{d})}$ could lead to a non-convergent algorithm. Although this behavior contrasts with the straightforward convergence properties of Smolyak algorithms, most reasonable choices for direct quadrature do converge, and hence this is not our primary argument against the approach.

Instead, our primary concern is aliasing in direct quadrature and how it reduces performance at finite order. Both internal and external aliasing are governed by the same basic rule, which is just a restatement of how we defined aliasing in Section 2.1.5.

Remark 2.3.1. *For a multi-index set $\mathcal{J}$ and a quadrature rule $\mathcal{Q}^{(\mathbf{d})}$, the corresponding direct quadrature polynomial expansion has no aliasing between two polynomial terms if $\Psi_{\mathbf{j}} \Psi_{\mathbf{j}'} \in \mathcal{E}(\mathcal{Q}^{(\mathbf{d})})$.*

The next two sections compare the internal and external aliasing with both theory and simple numeric examples.

2.3.2 Internal aliasing in direct quadrature

As an extension of Remark 2.3.1, direct quadrature has no internal aliasing whenever every pair $\mathbf{j}, \mathbf{j}' \in \mathcal{J}$ has no aliasing. We can immediately conclude that for any basis set $\mathcal{J}$, there is some quadrature rule sufficiently powerful to avoid internal aliasing errors. In practice, however, this rule may not be a desirable one.

Example 2.3.2. *Assume that for some function with two inputs, we wish to include the polynomial basis terms $(a, 0)$ and $(0, b)$. By Remark 2.3.1, the product of these two terms must be in the exact set; hence, the quadrature must include at least a full tensor rule of accuracy (a, b). Although we have not asked for any coupling, direct quadrature must assume full coupling of the problem in order to avoid internal aliasing.*

Therefore we reach the surprising conclusion that direct quadrature inserts significant coupling into the problem, whereas we selected a Smolyak quadrature rule in hopes of leveraging the absence of that very coupling—making the choice inconsistent. For most sparse quadrature rules, we cannot include as many polynomial terms as the Smolyak pseudospectral approach without incurring internal aliasing, because the quadrature is not powerful enough in the mixed dimensions.

Example 2.3.3. *Let $\mathbf{X}$ be the two-dimensional domain $[-1, 1]^2$. Select a uniform weight function, which corresponds to a Legendre polynomial basis. Let $f(x, y) = \psi_0(x)\psi_4(y)$. Use Gauss-Legendre quadrature and an exponential growth rule, such that $p^{(i)}(m) = 2^{m-1}$. Select a sparse quadrature rule based on a total order multi-index set $\mathcal{Q}^2_{\mathcal{K}_5^t}$. Figure 2-4 shows the exact set of this Smolyak quadrature rule (solid line) along with its half-exact set (dashed line), which encompasses all the terms in the direct quadrature polynomial expansion.*

Now consider the $\mathbf{j} = (8, 0)$ polynomial, which is in the half-exact set. The product of the $(0, 4)$ and $(8, 0)$ polynomial terms is $(8, 4)$, which is not within the exact set of the sparse rule. Hence, $(0, 4)$ aliases onto $(8, 0)$ because this quadrature rule has limited accuracy in the mixed dimensions.

Using both the Smolyak pseudospectral and direct quadrature methods, we numerically compute the polynomial expansion for this example. The resulting coefficients are shown in Figure 2-5. Even though the two methods use the same information and project f onto the same basis, the Smolyak result has no internal aliasing while direct quadrature shows significant internal aliasing. Although both methods correctly compute the $(0, 4)$ coefficient, direct quadrature shows aliasing on $(8, 0)$ as predicted, and also on $(10, 0)$, $(12, 0)$, and $(14, 0)$. In this case, direct quadrature is unable to determine the order of the input function or even whether the input is function of x_1 or x_2. Alternating terms are computed correctly because of the parity of the functions.

The $\mathcal{O}\big(1\big)$ errors observed in this simple example demonstrate why it is crucial to eliminate internal aliasing in the construction of one-dimensional pseudospectral approximations and to ensure that the full tensor and Smolyak algorithms inherit that property. More complex functions demonstrate the same type of error, except

that the errors resulting from multiple source terms are superimposed. Examples of the latter are given by Constantine *et al.* [22].

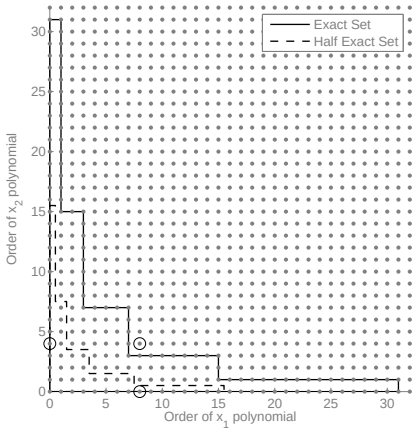


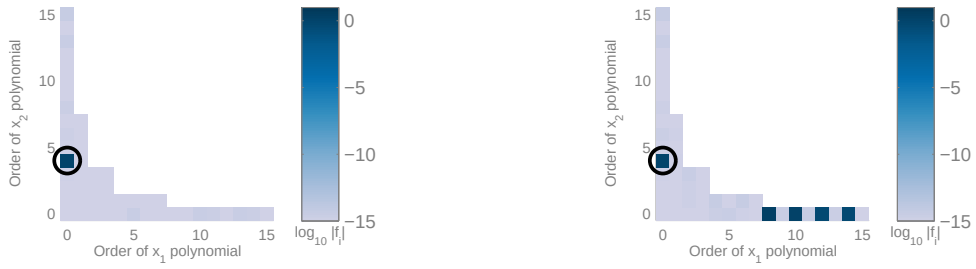
Figure 2-4: The exact set and polynomials included in the direct quadrature construction from Example 2.3.3.

There are some choices for which direct quadrature has no internal aliasing: full tensor quadrature rules and, notably, Smolyak quadrature constructed from one-dimensional Gaussian quadrature rules with $p^{(i)}(m) = m$, truncated according to an isotropic total-order multi-index set. However, many useful sparser or more tailored Smolyak quadrature rules, e.g., based on exponential growth quadrature rules or adaptive anisotropic Smolyak index sets, will incur internal aliasing if the basis selection matches the range of the Smolyak algorithm. This makes them a poor choice when a comparable Smolyak pseudospectral algorithm uses the same evaluation points and produces an approximation with the same polynomial terms, but is guaranteed by construction to have zero internal aliasing. Alternately, it is possible to select a sufficiently small polynomial basis to avoid internal aliasing, but this approach requires unnecessary conservatism that could easily be avoided with a Smolyak pseudospectral approximation.

2.3.3 External aliasing

The difference in external aliasing between direct quadrature and Smolyak pseudospectral approximation is much less severe. Both methods exhibit external aliasing from terms far outside the range of the approximation, as such errors are a necessary consequence of using finite order quadrature. Since the methods are constructed from similar constituent one-dimensional quadrature rules, aliasing is of similar magnitude when it occurs.

Comparing Theorem 2.2.5, condition *(b)*, and Remark 2.3.1, we observe that if the direct quadrature method has no external aliasing between two basis terms, the equivalent Smolyak pseudospectral algorithm will not either. Yet the two methods



(a) Smolyak Pseudospectral

(b) Direct Quadrature

Figure 2-5: Numerical results for Example 2.3.3; each color square indicates the log of the coefficient magnitude for the basis function at that position. The circle identifies the correct non-zero coefficient.

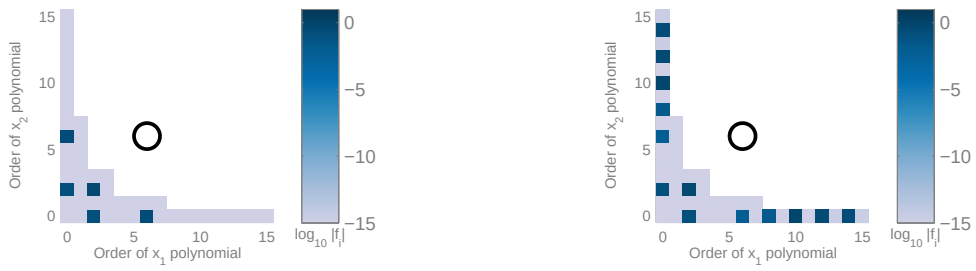
perform differently because of their behavior on separable functions. Condition *(a)* of Theorem 2.2.5 provides an additional condition under which external aliasing will not occur under a Smolyak pseudospectral algorithm, and thus it has strictly less external aliasing in general.

Example 2.3.4. *If we repeat Example 2.3.3 but choose f to be a polynomial outside the approximation space, $f = \psi_6(x)\psi_6(y)$, we obtain the results in Figure 2-6. Now every non-zero coefficient is the result of external aliasing. Direct quadrature correctly computes some terms because of either parity or the few cases where Remark 2.3.1 is satisfied. However, the Smolyak approach has fewer errors because the terms not between $(0,0)$ and $(6,6)$ are governed by condition (a) of Theorem 2.2.5, and hence have no external aliasing.*

This example is representative of the general case. Direct quadrature always incurs at least as much external aliasing as the Smolyak approach, and the methods become equivalent if the external term causing aliasing is of very high order. Although both methods will always exhibit external aliasing onto coefficients of the approximation for non-polynomial inputs, the truncation can in principle be chosen to include all the important terms, so that the remaining external aliasing is acceptably small.

2.3.4 Summary of comparison

Compared to the Smolyak pseudospectral approach, direct quadrature yields larger internal *and* external aliasing errors. Because of these aliasing errors, direct quadrature is essentially unable to make efficient use of most sparse quadrature rules. The Smolyak pseudospectral approach, on the other hand, is guaranteed never to have internal aliasing if the one-dimensional pseudospectral operators are chosen according to simple guidelines. We therefore recommend against using direct quadrature. The remainder of the chapter will focus on extensions of the basic Smolyak pseudospectral approach.



(a) Smolyak Pseudospectral

(b) Direct Quadrature

Figure 2-6: Numerical results for Example 2.3.4; each color square indicates the log of the coefficient magnitude for the basis function at its position. The circle indicates the correct non-zero coefficient. The Smolyak pseudospectral approach has fewer terms corrupted by external aliasing in this case.

2.4 Adaptive polynomial approximations

When constructing a polynomial approximation of a black-box computational model, there are two essential questions: first, which basis terms should be included in the expansion; and second, what are the coefficients of those basis terms? The Smolyak construction allows detailed control over the truncation of the polynomial expansion and the work required to compute it. Since we typically do not have *a priori* information about the functional relationship generated by a black-box model, we develop an adaptive approach to tailor the Smolyak approximation to this function, following the dimension-adaptive quadrature approaches of Gerstner & Griebel [43] and Hegland [52].

The Smolyak algorithm is well suited to an adaptive approach. The telescoping sum converges in the same sense as the constituent one-dimensional operators as the index set grows to include $\mathbb{N}_0^d$, so we can simply add more terms to improve the approximation until we are satisfied. We separate our adaptive approach into two components: local improvements to the Smolyak multi-index set and a global stopping criterion.

2.4.1 Dimension adaptivity

Dimension adaptivity is responsible for identifying multi-indices to add to a Smolyak multi-index set in order to improve its accuracy. A standard refinement approach is simply to grow an isotropic simplex of side length n. [43] and [52] instead suggest a series of greedy refinements that customize the Smolyak algorithm to a particular problem.

The refinement used in [43] is to select a multi-index $\mathbf{k} \in \mathcal{K}$ and to add the forward neighbors of $\mathbf{k}$ that are admissible. The multi-index $\mathbf{k}$ is selected via an error indicator

$\epsilon(\mathbf{k})$. We follow [43] and assume that whenever $\mathbf{k}$ contributes strongly to the result of the algorithm, it represents a subspace that likely needs further refinement.

Let $\mathbf{k}$ be a multi-index such that $\mathcal{K}' := \mathcal{K} \cup \mathbf{k}$, where $\mathcal{K}$ and $\mathcal{K}'$ are admissible multi-index sets. The triangle inequality (for some appropriate norm, see Section 2.4.3) bounds the change in the Smolyak approximation produced by adding $\mathbf{k}$ to $\mathcal{K}$, yielding a useful error indicator:

$$\|A(\mathcal{K}', d, \mathcal{L}) - A(\mathcal{K}, d, \mathcal{L})\| \leq \|\Delta_{k_1}^1 \otimes \cdots \otimes \Delta_{k_d}^d\| =: \epsilon(\mathbf{k}). \tag{2.39}$$

Conveniently, this error indicator does not change as $\mathcal{K}$ evolves, so we need only compute it once. At each adaptation step, we find the $\mathbf{k}$ that maximizes $\epsilon(\mathbf{k})$ and that has at least one admissible forward neighbor. Then we add those forward neighbors.

2.4.2 Termination criterion

Now that we have a strategy to locally improve a multi-index set, it is useful to have a global estimate of the error of the approximation, ϵ_g. We cannot expect to compute the exact error, but even a rough estimate is useful. We follow Gerstner & Griebel's choice of global error indicator

$$\epsilon_g := \sum \epsilon(\mathbf{k}), \tag{2.40}$$

where the sum is taken over all the multi-indices that are eligible for local adaptation at any particular step (i.e., that have admissible forward neighbors) [43]. The algorithm may be terminated when a particular threshold of the global indicator is reached, or when it falls by a specified amount.

2.4.3 Error indicators and work-considering algorithms

Thus far we have presented the adaptation strategy without reference to the problem of polynomial approximation. In this specific context, we use the $L^2(\mathbf{X}, w)$ norm in (2.39), because it corresponds to the convergence properties of pseudospectral approximation and thus seems an appropriate target for greedy refinements. This choice is a heuristic to accelerate performance—albeit one that is simple and natural, and has enjoyed success in numerical experiments (see Section 2.5). Moreover, the analysis of external aliasing in Theorem 2.2.5 suggests that, in the case of pseudospectral approximation, significant missing polynomial terms alias onto some of the *included* lower-order coefficients, giving the algorithm a useful indication of which direction to refine. This behavior helps reduce the need for smoothness in the coefficient pattern. Section 2.5.1 provides a small fix that further helps with even or odd functions.

One is not required to use this norm to define $\epsilon(\mathbf{k})$, however, and it is possible that other choices could serve as better heuristics for some problems. Unfortunately, making definitive statements about the properties or general utility of these heuristic refinement schemes is challenging. The approach described above is intended to be broadly useful, but specific applications may require experimentation to find better

choices.

Beyond the choice of norm, a commonly considered modification to $\epsilon(\mathbf{k})$ is to incorporate a notion of the computational effort required to refine $\mathbf{k}$. Define $n(\mathbf{k})$ as the amount of work to refine the admissible forward neighbors of $\mathbf{k}$, e.g., the number of new function evaluation points. [43] discusses an error indicator that provides a parameterized sliding scale between selecting the term with highest $\epsilon(\mathbf{k})$ and the lowest $n(\mathbf{k})$:

$$\epsilon_{w,1}(\mathbf{k}) = \max\left\{w\frac{\epsilon(\mathbf{k})}{\epsilon(\mathbf{1})}, (1-w)\frac{n(\mathbf{1})}{n(\mathbf{k})}\right\}. \tag{2.41}$$

Here $w \in [0,1]$ is the tuning parameter, and $\epsilon(\mathbf{1})$ and $n(\mathbf{1})$ are the indicator and cost of the first term. Putting $w = 0$ considers only the standard error indicator and $w = 1$ considers only the cost. A different indicator with a similar intent is

$$\epsilon_{w,2}(\mathbf{k}) = \epsilon(\mathbf{k}) - \tilde{w}n(\mathbf{k}), \tag{2.42}$$

where $\tilde{w} > 0$ describes a conversion between error and work. Both of these methods will sometimes select terms of low cost even if they do not appear to provide immediate benefit to the approximation. Yet we find both methods to be challenging to use in practice, because of the difficulty of selecting the tuning parameter. One can remove this particular tuning requirement by taking a ratio:

$$\epsilon_{w,3}(\mathbf{k}) = \frac{\epsilon(\mathbf{k})}{n(\mathbf{k})}. \tag{2.43}$$

This indicator looks for "efficient" terms to refine—ones that are expected to yield greater error reduction at less cost—rather than simply the highest-error directions. We performed some numerical experiments with these methods, but none of the examples demonstrated significant improvement. Furthermore, poor choices of tuning parameters can be harmful because they can essentially make the algorithm revert to a non-adaptive form. We do not give detailed results here for those experiments because they are not particularly conclusive; some types of coefficient patterns may benefit from work-considering approaches, but this remains an open problem.

On a similar note, using ϵ_g as a termination criteria is also a heuristic. As our experiments in Section 2.5 will show, for most smooth functions ϵ_g is an excellent estimate of the approximation accuracy. In other cases, the indicator can be quite poor; hence one should not rely on it exclusively. In practice, we typically terminate the algorithm based on a combination of elapsed wall clock time, the global error indicator, and an error estimate computed from limited *ad hoc* sampling.

2.5 Numerical experiments

Our numerical experiments focus on evaluating the performance of different quadrature rules embedded within the Smolyak pseudospectral scheme, and on evaluating

performance of the adaptive Smolyak approximation strategy. Aside from the numerical examples of Section 2.3, we do not investigate the performance of direct quadrature any further. Given our theoretical analysis of aliasing errors and the numerical demonstrations in [22], one can conclude without further demonstration that destructive internal aliasing indeed appears in practice.

This section begins by discussing practical considerations in the selection of quadrature rules. Then we evaluate convergence of Smolyak pseudospectral approximation schemes (non-adaptive and adaptive) on the Genz test functions. Next, we approximate a larger chemical kinetic system, illustrating the efficiency and accuracy of the adaptive method. Finally, we evaluate the quality of the global error indicator on all of these examples.

2.5.1 Selection of quadrature rules

Thus far we have sidestepped practical questions about which quadrature rules exist or are most efficient. Our analysis has relied only on polynomial accuracy of quadrature rules; all quadrature rules with a given polynomial accuracy allow the same truncation of a pseudospectral approximation. In practice, however, we care about the cumulative cost of the adaptive algorithm, which must step through successive levels of refinement.

Integration over a bounded interval with uniform weighting offers the widest variety of quadrature choices, and thus allows a thorough comparison. Table 2.1 summarizes the costs of several common quadrature schemes. First, we see that linear-growth Gaussian quadrature is asymptotically much less efficient than exponential-growth in reaching any particular degree of exactness. However, for rules with fewer than about ten points, this difference is not yet significant. Second, Clenshaw-Curtis shows efficiency equivalent to exponential-growth Gaussian: both use n points to reach nth order polynomial exactness [15]. However, their performance with respect to external aliasing differs: Clenshaw-Curtis slowly loses accuracy if the integrand is of order greater than n, while Gaussian quadrature gives $\mathcal{O}\left(1\right)$ error even on $(n+1)$-order functions [108]. This may make Clenshaw-Curtis Smolyak pseudospectral estimates more efficient. Finally, we consider Gauss-Patterson quadrature, which is nested and has significantly higher polynomial exactness—for a given cumulative cost—than the other types [86]. Computing the quadrature points and weights in finite precision (even extended-precision) arithmetic has practically limited Gauss-Patterson rules to 255 points, but we recommend them whenever this is sufficient.

For most other weights and intervals, there are fewer choices that provide polynomial exactness, so exponential-growth Gaussian quadrature is our default choice. In the specific case of Gaussian weight, Genz has provided a family of Kronrod extensions, similar to Gauss-Patterson quadrature, which may be a useful option [42].

If a linear growth rule is chosen and the domain is symmetric, we suggest that each new level include at least two points, so that the corresponding basis grows by at least one even and one odd basis function. This removes the possibility for unexpected effects on the adaptive strategy if the target function is actually even or odd.

	Lin. G			Exp. G			C-C		G-P	
Order	p	a	t	p	a	t	p	a	p	a
1	1	1	1	1	1	1	1	1	1	1
2	2	3	3	2	3	3	3	3	3	5
3	3	5	6	4	7	7	5	5	7	10
4	4	7	10	8	15	15	9	9	15	22
5	5	9	15	16	31	31	17	17	31	46
6	6	11	21	32	63	63	31	31	63	94
m	m	$2m-1$	$m^2-m/2$	2^{m-1}	2^m-1	2^m-1	$2^{m-1}+1$	$2^{m-1}+1$		

Table 2.1: The cost of four quadrature strategies as their order increases: linear growth Gauss-Legendre quadrature (Lin. G), exponential growth Gauss-Legendre quadrature (Exp. G), Clenshaw-Curtis quadrature (C-C), and Gauss-Patterson quadrature (G-P). We list the number of points used to compute the given rule (p), the polynomial exactness (a), and the total number of points used so far (t). For nested rule, (p) = (t), so the total column is omitted.

2.5.2 Basic convergence: Genz functions

The Genz family [40, 41] comprises six parameterized functions, defined from $[-1,1]^d \to \mathbb{R}$. They are commonly used to investigate the accuracy of quadrature rules and interpolation schemes [6, 71]. The purpose of this example is to show that different Smolyak pseudospectral strategies behave roughly as expected, as evidenced by decreasing L^2 approximation errors as more function evaluations are employed. These functions are as follows:

$$\begin{aligned}
\text{oscillatory: } f_1(x) &= \cos\left(2\pi w_1 + \sum_{i=1}^{d} c_i x_i\right) \\
\text{product peak: } f_2(x) &= \prod_{i=1}^{d} \left(c_i^{-2} + (x_i - w_i)^2\right)^{-1} \\
\text{corner peak: } f_3(x) &= \left(1 + \sum_{i=1}^{d} c_i x_i\right)^{-(d+1)} \\
\text{Gaussian: } f_4(x) &= \exp\left(-\sum_{i=1}^{d} c_i^2 (x_i - w_i)^2\right) \\
\text{continuous: } f_5(x) &= \exp\left(-\sum_{i=1}^{d} c_i^2 (|x_i - w_i|)^2\right) \\
\text{discontinuous: } f_6(x) &= \begin{cases} 0 & \text{if } x_1 > w_1 \text{ or } x_2 > w_2 \\ \exp\left(\sum_{i=1}^{d} c_i x_i\right) & \text{otherwise} \end{cases}
\end{aligned}$$

Our first test uses five isotropic and *non-adaptive* pseudospectral approximation strategies. The initial strategy is the isotropic full tensor pseudospectral algorithm, based on Gauss-Legendre quadrature, with order growing exponentially with level. The other four strategies are total-order expansions of increasing order based on the

following quadrature rules: linear growth Gauss-Legendre, exponential growth Gauss-Legendre, Clenshaw-Curtis, and Gauss-Patterson. All the rules were selected so that the final rule would have around 10^4 points.

We consider 30 random realizations of each Genz function in $d = 5$ dimensions; random parameters for the Genz functions are drawn uniformly from $[0, 1]$, then normalized so that $\|\mathbf{w}\|_1 = 1$ and $\|\mathbf{c}\|_1 = b_j$, where j indexes the Genz function type and the constants b_j are as chosen in [6, 71]. This experiment only uses the first four Genz functions, which are in C^∞, as pseudospectral methods have well known difficulties on functions with discontinuities or discontinuous derivatives [11]. Each estimate of L^2 approximation error is computed by Monte Carlo sampling with 10^4 samples. Figure 2-7 plots L^2 error at each stage, where each point represents the mean error over the 30 random functions.

Relatively simple conclusions can be drawn from this data. All the methods show fast convergence, indicating that the internal aliasing issues have indeed been resolved. In contrast, one would expect direct quadrature to suffer from large aliasing errors for the three super-linear growth rules. Otherwise, judging the efficiency of the different rules is not prudent, because differences in truncation and the structure of the test functions themselves obscure differences in efficiency. In deference to our adaptive strategy, we ultimately do not recommend this style of isotropic and function-independent truncation anyway.

To test our *adaptive* approach, Figure 2-8 shows results from a similar experiment, now comparing the convergence of an adaptive Smolyak pseudospectral algorithm with that of a non-adaptive algorithm. To make the functions less isotropic, we introduce an exponential decay, replacing each c_i with $c_i e^{i/5}$, where the c_i are generated and normalized as above. For consistency, both algorithms are based on Gauss-Patterson quadrature. As we cannot synchronize the number of evaluations used by the adaptive algorithm for different functions, we plot individual errors for the 30 random functions instead of the mean error. This reveals the variability in difficulty of the functions, which was hidden in the previous plot. We conclude that the adaptive algorithm also converges as expected, with performance comparable to or better than the non-adaptive algorithm. Even though we have included some anisotropy, these functions include relatively high degrees of coupling; hence, in this case the non-adaptive strategy is a fairly suitable choice. For example, the "product peak" function shows little benefit from the adaptive strategy. Although omitted here for brevity, other quadrature rules produce similar results when comparing adaptive and non-adaptive algorithms.

2.5.3 Adaptivity: chemical kinetics

To further illustrate the benefits of an adaptive Smolyak approach, we build a surrogate for a realistic simulation of a combustion kinetics problem. Specifically, we consider the auto-ignition of a methane-air mixture given 14 uncertain rate parameters. Governing equations for this process are a set of stiff nonlinear ODEs expressing

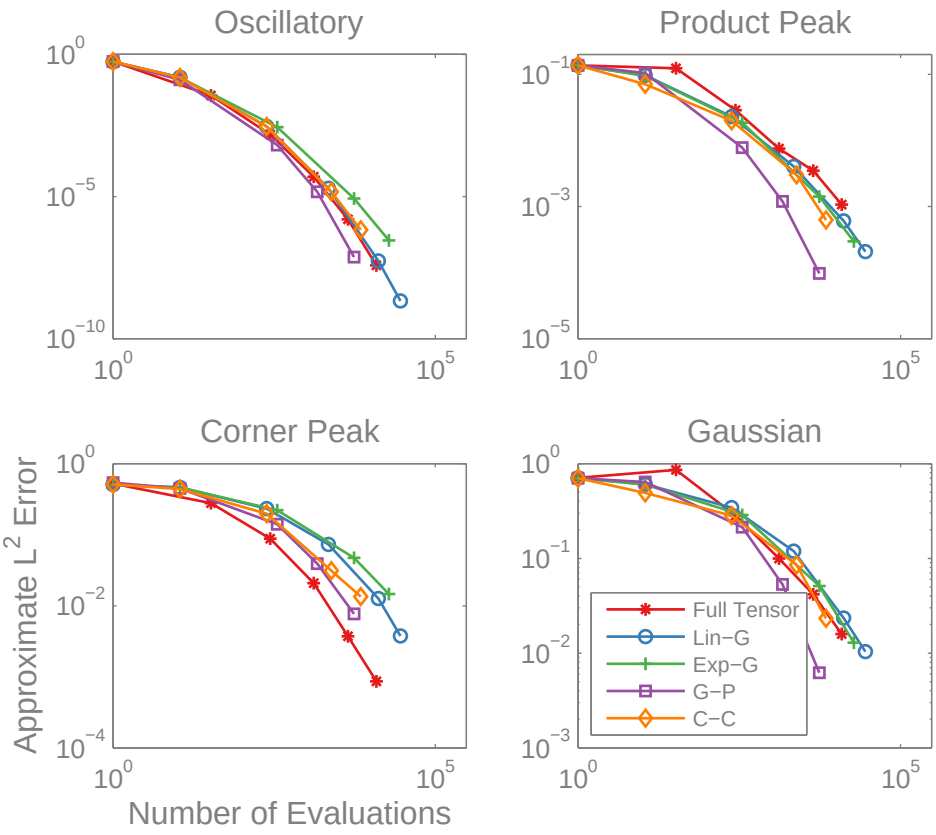


Figure 2-7: Mean L^2 convergence of the non-adaptive isotropic total-order Smolyak pseudospectral algorithm with various quadrature rules, compared to the full tensor pseudospectral algorithm, on the Genz test functions.

conservation of energy and of chemical species [69]. The uncertain rate parameters represent activation energies of reactions governing the conversion of methane to methyl, each endowed with a uniform distribution varying over $[0.8, 1.25]$ of the nominal value. These parameters appear in Arrhenius expressions for the species production rates, with the reaction pathways and their nominal rate parameters given by the GRIMech 3.0 mechanism [48]. The output of interest is the logarithm of the ignition time, which is a functional of the trajectory of the ODE system, and is continuous over the selected parameter ranges. Simulations were performed with the help of the TChem software library [98], which provides convenient evaluations of thermodynamic properties and species production rates, along with Jacobians for implicit time integration.

Chemical kinetics are an excellent testbed for adaptive approximation because, by the nature of detailed kinetic systems, we expect strong coupling between some inputs and weak coupling between others, but we cannot predict these couplings *a priori*. We test the effectiveness of adaptive Smolyak pseudospectral methods based on the four quadrature rules discussed earlier. As our earlier analysis suggested that Gauss-Patterson quadrature should be most efficient, our reference solution is a non-

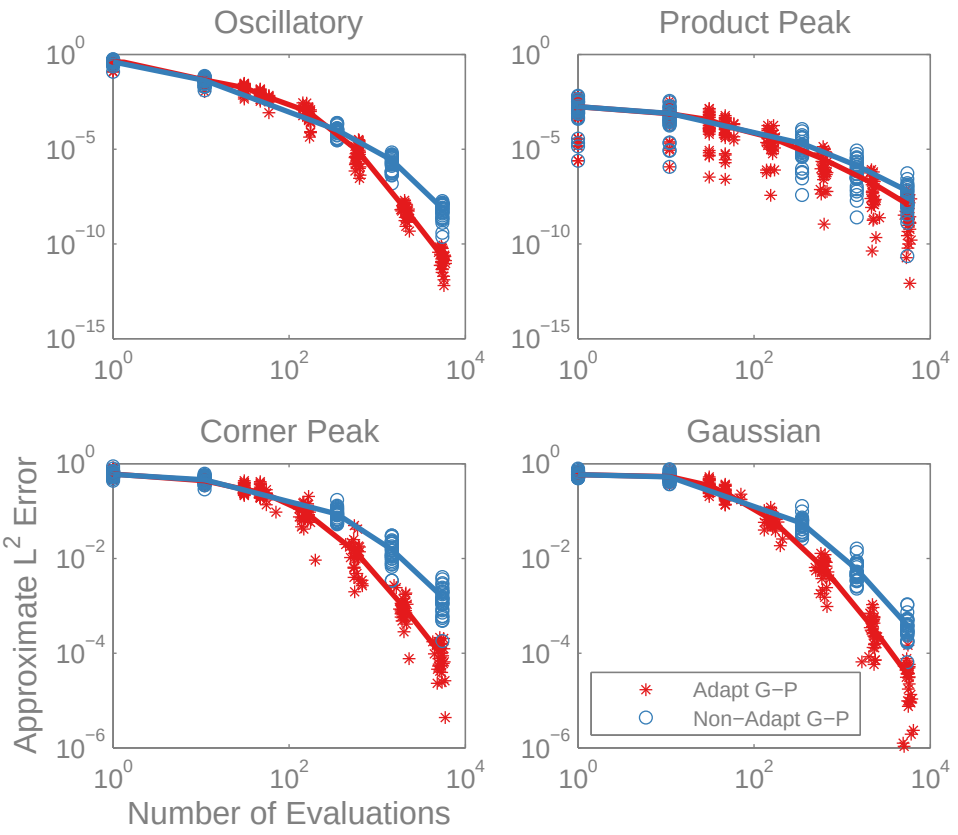


Figure 2-8: L^2 convergence of the adaptive and non-adaptive Gauss-Patterson Smolyak pseudospectral algorithm. Individual results for 30 random instances of the Genz functions are shown.

adaptive Gauss-Patterson total-order Smolyak pseudospectral expansion. We ran the non-adaptive algorithm with a total order index set truncated at $n = 5$ (which includes monomial basis terms up through $\psi_{23}^{(i)}$), using around 40000 point evaluations and taking over an hour to run. We tuned the four adaptive algorithms to terminate with approximately the same number of evaluations.

Figure 2-9 compares convergence of the five algorithms. The L^2 errors reported on the vertical axis are Monte Carlo estimates using 10^4 points. Except for a small deviation at fewer than 200 model evaluations, all of the adaptive methods significantly outperform the non-adaptive method. The performance of the different quadrature rules is essentially as predicted in Section 2.5.1: Gauss-Patterson is the most efficient, exponential growth Gauss-Legendre and Clenshaw-Curtis are nearly equivalent, and linear growth Gauss-Legendre performs worse as the order of the polynomial approximation increases. Compared to the non-adaptive algorithm, adaptive Gauss-Patterson yields more than two orders of magnitude reduction in the error at the same number of model evaluations. Linear growth Gaussian quadrature is initially comparable to exponential growth Gaussian quadrature, because the asymptotic benefits of ex-

ponential growth do not appear while the algorithm is principally using very small one-dimensional quadrature rules. At the end of these experiments, a reasonable number of higher order quadrature rules are used and the difference becomes visible.

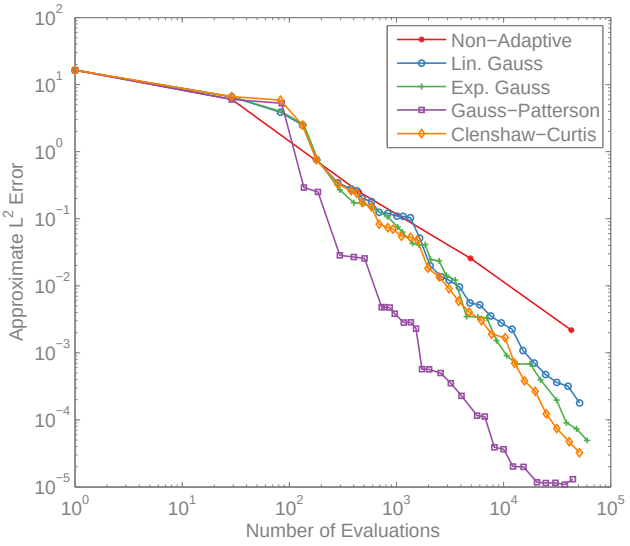


Figure 2-9: L^2 convergence of ignition delay in a 14-dimensional chemical kinetic system; comparing a non-adaptive isotropic total-order Gauss-Patterson-based Smolyak pseudospectral algorithm to the adaptive algorithm with various quadrature rules.

We conclude by illustrating that the adaptive algorithm is effective because it successfully focuses its efforts on high-magnitude coefficients—that is, coefficients that make the most significant contributions to the function. Even though the non-adaptive expansion has around 37,000 terms and the final adaptive Gauss-Patterson expansion only has about 32,000 terms, the adaptive expansion exhibits much lower error because most of the additional terms in the non-adaptive expansion are nearly zero. By skipping many near-zero coefficients, the adaptive approach is able to locate and estimate a number of higher-order terms with large magnitudes. Figure 2-10 depicts this pattern by plotting the difference between the numbers of included terms in the final adaptive Gauss-Patterson and non-adaptive expansions. The adaptive algorithm does not actually add any higher order monomials; neither uses one-dimensional basis terms of order higher than $\psi_{23}^{(i)}$. Instead, the adaptive algorithm adds mixed terms of higher total order, thus capturing the coupling of certain variables in more detail than the non-adaptive algorithm. The figure shows that terms through 30^{th} order are included in the adaptive expansion, all of which are products of non-constant polynomials in more than one dimension.

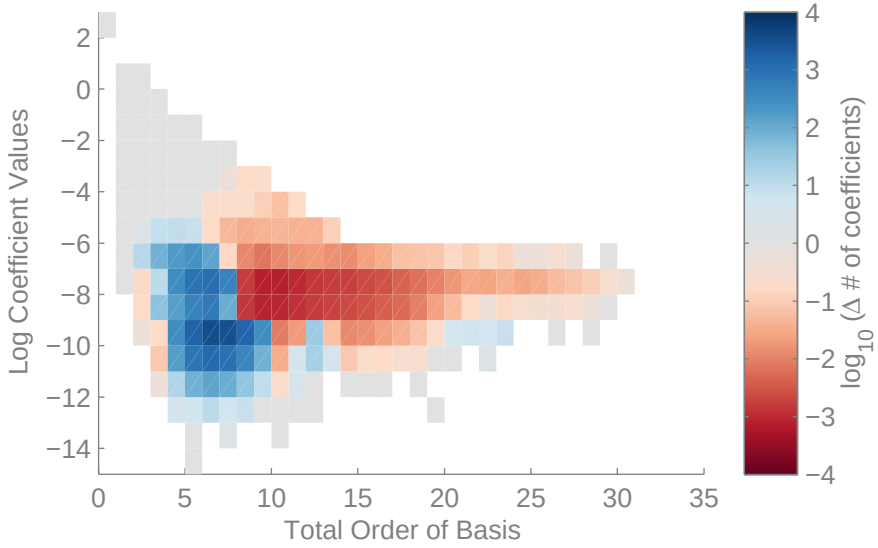


Figure 2-10: The plot depicts the difference between the *number* of coefficients of a particular magnitude and order in the final *adaptive* and *non-adaptive* Gauss-Patterson based expansions. The horizontal axis is the order of the term and the vertical axis specifies $\log_{10}$ of the coefficient value. The color represents $\log_{10}$ of the difference between the two methods, where positive values indicate more terms in the non-adaptive expansion. Hence, the dark blue at $(6, -10)$ indicates that the non-adaptive expansion includes around 3,000 extra terms of magnitude 10^{-10} and the dark red at $(10, -8)$ indicates that the adaptive expansion includes about 1,000 extra terms of magnitude 10^{-8}. Grey squares are the same for both expansions and white squares are not present in either.

2.5.4 Performance of the global error indicator

To evaluate the termination criterion, we collected the global error indicator during runs of the adaptive algorithm for all of the test functions described above, including the slowly converging non-smooth Genz functions omitted before. The discontinuous Genz function does not include the exponential coefficient decay because the discontinuity already creates strong anisotropy. Results are shown for Gauss-Patterson quadrature. The relationship between the estimated L^2 error and the global error indicator ϵ_g is shown in Figure 2-11. For the smooth test functions, ϵ_g is actually an excellent indicator, as it is largely within an order of magnitude of the correct value and essentially linearly related to it. However, the non-smooth Genz functions illustrate the hazard of relying too heavily on this indicator: although the adaptive algorithm does decrease both the errors and the indicator, the relationship between the two appears far less direct.

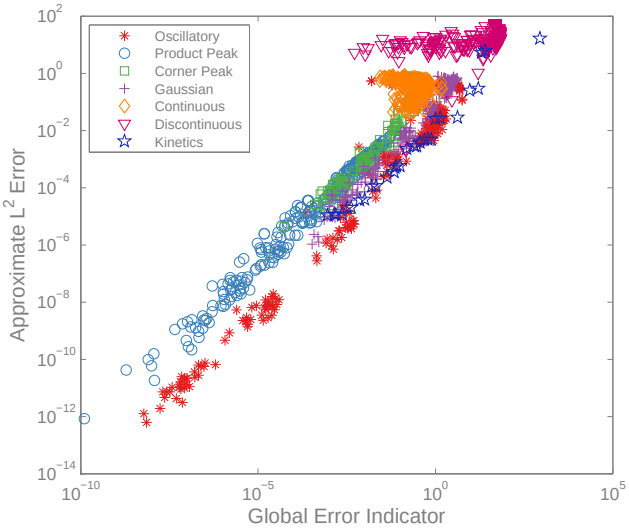


Figure 2-11: Relationship between the termination criterion (2.40) and the estimated L^2 error for every function tested.

2.6 Conclusions

This chapter gives a rigorous development of Smolyak pseudospectral algorithms, a practical approach for constructing polynomial chaos expansions from point evaluations of a function. A common alternative approach, direct quadrature, has previously been shown to suffer from large errors. We explain these errors as a consequence of internal aliasing and delineate the exact circumstances, derived from properties of the chosen polynomial basis and quadrature rules, under which internal aliasing will occur. Internal aliasing is a problem inherent to direct quadrature approaches, which use a single (sparse) quadrature rule to compute a set of spectral coefficients. These approaches fail because they substitute a numerical approximation for only a portion of the algorithm, i.e., the evaluation of integrals, without considering the impact of this approximation on the entire construction. For almost all sparse quadrature rules, internal aliasing errors may be overcome only through an inefficient use of function evaluations. In contrast, the Smolyak pseudospectral algorithm computes spectral coefficients by assembling tensor-product pseudospectral approximations in a coherent fashion that avoids internal aliasing by construction; moreover, it has smaller external aliasing errors. To establish these properties, we extend the known result that the exact set of a Smolyak pseudospectral approximation contains a union of the exact sets of all its constituent tensor-product approximation operators to the case of arbitrary admissible Smolyak multi-index sets. These results are applicable to any choice of quadrature rule and generalized sparse grid, and are verified through numerical demonstrations; hence, we suggest that the Smolyak pseudospectral algorithm is a superior approach in almost all contexts.

A key strength of Smolyak algorithms is that they are highly customizable through

the choice of admissible multi-index sets. To this end, we describe a simple alteration to the adaptive sparse quadrature approaches of [43, 52], creating a corresponding method for adaptive pseudospectral approximation. Numerical experiments then evaluate the performance of different quadrature rules and of adaptive versus non-adaptive pseudospectral approximation. Tests of the adaptive method on a realistic chemical kinetics problem show multiple order-of-magnitude gains in accuracy over a non-adaptive approach. Although the adaptive strategy will not improve approximation performance for every function, we have little evidence that it is ever harmful and hence widely recommend its use.

Chapter 3

Asymptotically exact MCMC algorithms via local approximations[1]

3.1 Introduction

An important strategy for mitigating the computational cost of inference is to recognize that the forward model may exhibit regularity in its dependence on the parameters of interest, such that the model outputs may be approximated with fewer samples than are needed to characterize the posterior via MCMC. Replacing the forward model with an approximation or "surrogate" *decouples* the required number of forward model evaluations from the length of the MCMC chain, and thus can vastly reduce the overall cost of inference [97, 70]. Successful existing approaches typically create high-order global approximations for either the forward model outputs or the log-likelihood function, using, for example, global polynomials [80, 79], radial basis functions [8, 63], or Gaussian processes [97, 70, 91, 99]. Most of these approaches assume that the forward model is deterministic and available only as a black box. We will make the same assumptions here, thus focusing our attention on "non-intrusive" approximation methods that are based only on evaluations of the forward model at selected input points.

Although current methods can provide significant empirical performance improvements, they tend to sacrifice either exact sampling or potential speedups, by either over- or under-utilizing the surrogate. For example, many methods fix the likelihood or forward model approximation after a finite degree of refinement, then use an MCMC algorithm to analyze the resulting posterior without further reference to the true model. The resulting samples are thus necessarily drawn from an approximate posterior distribution. In principle, one might require only that the bias of any posterior expectation computed using samples from this approximate posterior be small relative to the variance introduced by the finite length of the MCMC chain.

[1] The material in this chapter is adapted from [21].

But current methods lack a rigorous approach to controlling this bias [7, 34]. While there do exist theoretical results that allow the convergence rates of the forward model approximation to be transferred to the posterior distribution [23, 79], assessing and controlling the *magnitude* of the error in the posterior remains a challenge. Conversely, other methods limit potential performance improvement by failing to "trust" the surrogate even when it is accurate. Delayed-acceptance schemes, for example, evaluate the full model in a second Metropolis acceptance step within an MCMC iteration, thus screening proposals that have been accepted according to the approximate posterior [91, 14, 28]. These schemes achieve exact sampling while eliminating the need for error analysis of the surrogate, but may require almost as much effort as running MCMC with the full model directly.

Analyzing the error of a forward model approximation would seem to be a useful route towards more efficient exact sampling, or sampling with controlled errors, but this can be quite challenging for the *global* approximation methods used in previous work. Adding to the difficulty of analysis are the complex experimental design heuristics with which these approximations are often paired [91, 7, 34]. Efficiently constructing a surrogate within an inference problem requires that the sample set be tailored simultaneously to the approximation family, the function to be approximated, and the posterior distribution itself. As several of these components are unknown a priori, these requirements favor *sequential* design approaches that interleave characterization of the posterior with refinement of the surrogate. Methods for sequential design in this context are inevitably ad hoc, however. Even when the design heuristics perform well, it is not clear how to establish rigorous error bounds for finite samples or even how to establish convergence for infinite samples, given relatively arbitrary point sets. In contrast, polynomial chaos expansions simplify the experimental design problem by designing sample grids [118, 83, 22, 20] with respect to the prior distribution, which is known to induce a convergent approximation of the posterior density [79]. However, using only prior information is likely to be inefficient; in situations whenever the data are informative such as in large samples, the posterior concentrates on a small fraction of the parameter space relative to the prior. Creating a surrogate that is accurate over the entire prior support thus wastes considerable computational effort. Overall there is a need for efficient approaches with provable convergence properties—such that one can achieve exact sampling while making full use of the surrogate model.

3.1.1 Our contribution

This chapter resolves the above-mentioned issues by proposing a new framework that integrates *local* approximations into Metropolis-Hastings kernels, producing a Markov chain that asymptotically (in the number of MCMC steps) samples from the *exact* posterior distribution. As examples of this approach, we will employ approximations of either the log-likelihood function or the forward model, using local linear, quadratic, or Gaussian process regression. To produce the sample sets used for these local approximations, we will introduce a sequential experimental design procedure that interleaves

infinite refinement of the approximation with the Markov chain's exploration of the posterior. The overall experimental design reflects a combination of guidance from MCMC (so that samples are focused on the posterior) and local space filling heuristics (to ensure well-poised sample sets for local approximation), triggered both by random refinement and by local error indicators of model quality. The result is a practical approach that also permits rigorous error analysis. This concept is inspired by the use of trust region methods in derivative-free optimization [18, 19], wherein local models similarly allow the reuse of model evaluations while enabling refinement until convergence. Optimization and posterior sampling place different requirements on their underlying approximations, however, and thus our integration of local models into MCMC departs from the experimental design procedures used in derivative-free optimization.

Although local approximations cannot be expected to converge as quickly as global (e.g., spectral) approximations of smooth functions, they are simpler to analyze in the present context and provably convergent under relatively straightforward conditions. We use these properties to prove that the resulting MCMC algorithm converges asymptotically to the posterior distribution induced by the exact forward model and likelihood. In this chapter we focus on the specific case of a random-walk Metropolis algorithm coupled with local quadratic approximations of the log-likelihood. We show ergodicity of the MCMC chain with respect to the exact posterior. The proof involves demonstrating that the transition kernel converges quickly as the posterior distribution is explored and as the surrogate is refined. Our arguments are not limited to the random-walk Metropolis algorithm; they apply quite broadly and can be adapted to most other Metropolis-Hastings algorithms and local approximation schemes. Although we do not focus our attention on finite-time error bounds, it is straightforward to propagate such bounds through our arguments. Broadly, *our theoretical results reinforce the notion that it is possible to greatly reduce the number of evaluations of the forward model per MCMC step when the likelihood has some local regularity.* We complement the theory by demonstrating experimental performance improvements of up to several orders of magnitude on inference problems involving ordinary differential equation and partial differential equation forward models, using several different MCMC algorithms and local approximation schemes.

The remainder of this chapter is organized as follows. We describe the new MCMC approach in Section 3.2. A more detailed discussion of related work is deferred to Section 3.2.6, to allow concrete comparisons with our method. Theoretical results, including a proof of ergodicity, are provided in Section 3.3. Section 3.4 then provides empirical assessments of performance in several examples. We emphasize that, while the examples demonstrate strong computational performance, the present implementation is merely a representative of a class of asymptotically exact MCMC algorithms. Section 3.5 discusses several variations on the core algorithm that may be pursued in future work.

3.2 Algorithm description

This section describes our framework for Metropolis-Hastings algorithms based on local approximations, which incrementally and infinitely refine an approximation of the forward model or likelihood as inference is performed. We consider a Bayesian inference problem with posterior density

$$p(\theta|\mathbf{d}) \propto \mathcal{L}(\theta|\mathbf{d}, \mathbf{f})p(\theta),$$

for inference parameters $\theta \in \Theta \subseteq \mathbb{R}^d$, data $\mathbf{d} \in \mathbb{R}^n$, forward model $\mathbf{f} : \Theta \to \mathbb{R}^n$, and probability densities specifying the prior $p(\theta)$ and likelihood function $\mathcal{L}$. The forward model may enter the likelihood function in various ways. For instance, if $\mathbf{d} = \mathbf{f}(\theta) + \eta$, where $\eta \sim p_\eta$ represents some measurement or model error, then $\mathcal{L}(\theta|\mathbf{d}, \mathbf{f}) = p_\eta(\mathbf{d} - \mathbf{f}(\theta))$. Assume that a forward model evaluation is computationally expensive—requiring, for example, a high resolution numerical solution of a partial differential equation (PDE). Also assume that given the parameters and the forward model evaluation, the prior density and likelihood are inexpensive to evaluate, e.g., Gaussian. In such a setting, the computational cost of MCMC is dominated by forward model evaluations, rather than proposal construction or density evaluations.

3.2.1 Overview

Our approach is to use local models in a strategy inspired by work in derivative-free optimization [19] and also related to the LOESS procedure for local regression [16]. To compute the approximation $\tilde{\mathbf{f}}$ at a target point θ, we gather a collection of existing evaluation pairs $(\theta_i, \mathbf{f}(\theta_i))$ at nearby points, then fit a local model using those examples. To control the error of the approximation, the model may need to be refined; this is accomplished by running $\mathbf{f}$ at additional points. We restrict ourselves to the case of incorporating local approximations into Metropolis-Hastings proposals with translation-invariant kernels; other possibilities (e.g., combining local and global approximations, or Hamiltonian Monte Carlo) are left as future work and are briefly described in Section 3.5. We initially describe the framework using local linear or quadratic models; Section 3.2.5 then explains the simple changes required to replace these models with local Gaussian process approximations.

The literature is divided between so-called *direct* and *indirect* surrogate methods [7]—that is, approximations of $\log p(\theta|\mathbf{d})$ or of $\mathbf{f}$, respectively.[2] Both the direct and indirect approaches have certain advantages, and the algorithm presented here can switch between them merely by relabeling the terms subject to approximation. Indeed, it is not obvious whether either approach is superior in general. Practical differences between them include the following: (1) the indirect method produces a higher-order approximation of the posterior density when substituted into the like-

[2] We propose that in direct methods it might be advantageous to approximate only the log-likelihood and to evaluate the prior density without approximation, whenever the latter is inexpensive to evaluate.

lihood; (2) indirect approximation creates surrogates for each model output, which may be numerous and whose errors may accumulate; (3) direct approximation, on the other hand, must only contend with a scalar output; (4) choosing one approach or the other affects which derivatives can be provided to the MCMC algorithm, as well as what kind of adjoint or derivative information from the forward model can be incorporated into the approximation. The numerical experiments performed in this chapter will demonstrate both the direct and indirect approaches. Without loss of generality, we will cast the algorithm description below in terms of approximating the forward model.

In Metropolis-Hastings (MH) algorithms, the forward model is used to compute the probability α of accepting a proposal. In the present context, evaluating the acceptance probability invokes evaluations of the forward model surrogate at both the current and proposed points, θ^- and θ^+, respectively. In keeping with standard reversibility arguments for MH kernels, we consider refinements at either point. Refinement is triggered via two separate criteria: randomly with probability β_m; or when acceptance probability error indicators ϵ^- or ϵ^+ exceed a threshold γ_m. Both refinement thresholds β_m and γ_m decrease asymptotically to zero as the length of the MCMC chain $m \to \infty$. When refinement is triggered, an experimental design phase selects a new point at which to evaluate the forward model. This point is generally near, but distinct from, either the current or proposed points. Intuitively, this procedure should produce a sample set that is space-filling with respect to the high posterior probability regions. Figure 3-1 contrasts this approach with a prior-based sparse grid [20], which can produce approximations that converge quickly with respect to a prior-weighted norm, but is inefficient in this context because it places so few samples in the region of high posterior probability.

Algorithm 3.1 formally describes our approach. The main method, LocallyApproximate, takes as its arguments the function $\mathbf{f}$ to approximate and a pair of points $\theta^+, \theta^- \in \Theta \subseteq \mathbb{R}^d$, and returns local approximations of $\mathbf{f}$ at those two points for use in MCMC. It also takes the existing set of true model evaluations $\mathcal{S} := \{\theta, \mathbf{f}(\theta)\}$,[3] a specification of the posterior (represented as the likelihood $\mathcal{L}$, data $\mathbf{d}$, and prior p), and two constants that govern refinement, $\beta_m, \gamma_m \in [0, 1]$. The algorithm proceeds in three phases: lines 3–4 compute the best approximation using existing samples, lines 5–7 estimate the error in the acceptance probability using cross validation, and lines 8–16 evaluate the model at new points, as necessary. The algorithm uses two supporting subroutines: ConstructApproximation forms a local approximation, optionally withholding a sample for cross validation, and RefineNear performs experimental design to select new points. The details of these steps are given in the following subsections, followed by a discussion of previous work on related algorithms.

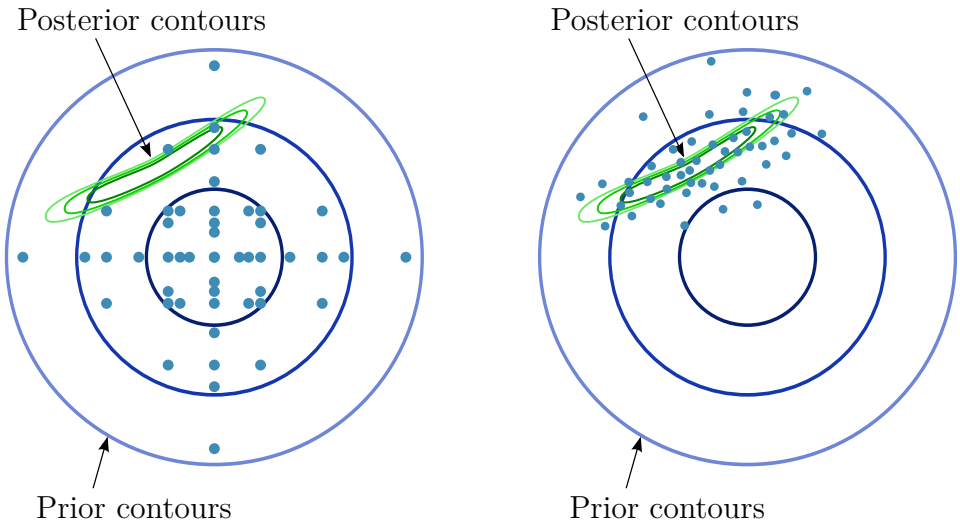
[3]Before MCMC begins, $\mathcal{S}$ needs to be seeded with a sufficient number of samples for the first run. Two simple strategies are to draw these samples either from the prior or near the MCMC starting point, which is often the posterior mode as found by optimization.

Algorithm 3.1 Construct Local Approximation

procedure LocallyApproximate($\mathbf{f}, \theta^+, \theta^-, \mathcal{S}, \mathcal{L}, \mathbf{d}, p, \beta_m, \gamma_m$)
  while forever do
    $\tilde{\mathbf{f}}^+ \leftarrow$ ConstructApproximation($\theta^+, \mathcal{S}, \emptyset$)
      ▷ Compute nominal approximations
    $\tilde{\mathbf{f}}^- \leftarrow$ ConstructApproximation($\theta^-, \mathcal{S}, \emptyset$)
    $\alpha \leftarrow \min\left(1, \frac{\mathcal{L}(\theta|\mathbf{d},\tilde{\mathbf{f}}^+)p(\theta^+)}{\mathcal{L}(\theta|\mathbf{d},\tilde{\mathbf{f}}^-)p(\theta^-)}\right)$
      ▷ Compute nominal acceptance ratio
    $\epsilon^+ \leftarrow \max_j \left|\alpha - \min\left(1, \frac{\mathcal{L}(\theta^+|\mathbf{d},\text{ConstructApproximation}(\theta^+,\mathcal{S},j))p(\theta^+)}{\mathcal{L}(\theta^-|\mathbf{d},\tilde{\mathbf{f}}^-)p(\theta^-)}\right)\right|$
    $\epsilon^- \leftarrow \max_j \left|\alpha - \min\left(1, \frac{\mathcal{L}(\theta^+|\mathbf{d},\tilde{\mathbf{f}}^+)p(\theta^+)}{\mathcal{L}(\theta^-|\mathbf{d},\text{ConstructApproximation}(\theta^-,\mathcal{S},j))p(\theta^-)}\right)\right|$
    if $\epsilon^+ \geq \epsilon^-$ and $\epsilon^+ \geq \gamma_m$ **then**
      ▷ If needed, refine near the larger error
      $\mathcal{S} \leftarrow$ RefineNear($\theta^+, \mathcal{S}$)
    else if $\epsilon^- > \epsilon^+$ and $\epsilon^- \geq \gamma_m$ **then**
      $\mathcal{S} \leftarrow$ RefineNear($\theta^-, \mathcal{S}$)
    else if $u \sim \mathcal{U}(0,1) < \beta_m$ **then**
      ▷ Refine with small probability β_m
      Randomly, $\mathcal{S} \leftarrow$ RefineNear($\theta^+, \mathcal{S}$) or $\mathcal{S} \leftarrow$ RefineNear($\theta^-, \mathcal{S}$)
    else ▷ Accept approximations
      return($\tilde{\mathbf{f}}^+, \tilde{\mathbf{f}}^-, \mathcal{S}$)
    end if
  end while
end procedure

procedure ConstructApproximation($\theta, \mathcal{S}, j$)
  Select R so that $|\mathcal{B}(\theta, R)| = N$, where
  $\mathcal{B}(\theta, R) := \{\{\theta_i, \mathbf{f}(\theta_i)\} \in \mathcal{S} : \|\theta_i - \theta\|_2 \leq R\}$ ▷ Select ball of points
  $\mathcal{B}(\theta, R) \leftarrow$ EnsurePoisedness($\mathcal{B}(\theta, R)$)
  $\tilde{\mathbf{f}} \leftarrow \mathcal{A}^{\sim j}_{\mathcal{B}(\theta,R)}$ ▷ Form approximation,
    ▷ possibly without sample j
  return $\tilde{\mathbf{f}}$
end procedure

procedure RefineNear($\theta, \mathcal{S}$)
  $\theta^* \leftarrow \max_{\|\theta'-\theta\| \leq R} \min_{\theta_i \in \mathcal{S}} \|\theta_i - \theta'\|$ ▷ Optimize near θ
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{\theta^*, \mathbf{f}(\theta^*)\}$ ▷ Grow the sample set
end procedure



(a) Prior-based sparse grid samples. (b) Posterior-adapted samples.

Figure 3-1: Schematic of an inference problem with a Gaussian prior and a posterior concentrated therein, with two experimental design approaches superimposed. Points are locations in the parameter space where the forward model is evaluated.

3.2.2 Local polynomial approximation

First, we explain the subroutine ConstructApproximation, which produces an approximation at θ, using a fixed set of samples $\mathcal{S}$, optionally omitting sample j. This section follows Conn *et al.* [19] in constructing either linear or quadratic local models. We construct these models using samples of $\mathcal{S}$ drawn from a ball of radius R centered on θ, $\mathcal{B}(\theta, R) := \{(\theta_i, \mathbf{f}(\theta_i)) \in \mathcal{S} : \|\theta_i - \theta\|_2 \leq R\}$. If this set contains sufficient samples, the low order polynomials are regression models that may easily be fit using least squares. We write the operators that produce the fitted linear or quadratic approximations as $\mathcal{L}^{\sim j}_{\mathcal{B}(\theta^*,R)}$ or $\mathcal{Q}^{\sim j}_{\mathcal{B}(\theta^*,R)}$, respectively. The superscript $\sim j$, if non-empty, indicates that sample j should be omitted; this option is used to support the cross-validation error indicators, described below. The pseudocode uses $\mathcal{A}^{\sim j}_{\mathcal{B}(\theta^*,R)}$ to represent either polynomial fitting algorithm.

It can be shown that the following error bounds hold independently for each output component, for every θ' in the ball [19]:

$$\left| f_i(\theta') - \left(\mathcal{L}^{\sim j}_{\mathcal{B}(\theta,R)}(\theta')\right)_i \right| \leq \kappa_l(\nu_1, \lambda, d) R^2, \tag{3.1a}$$

$$\left| f_i(\theta') - \left(\mathcal{Q}^{\sim j}_{\mathcal{B}(\theta,R)}(\theta')\right)_i \right| \leq \kappa_q(\nu_2, \lambda, d) R^3, \tag{3.1b}$$

where the constants κ are a functions of the Lipschitz constants ν_1, ν_2 of the gradient or Hessian of $\mathbf{f}$, respectively; a constant λ reflecting the geometry of the input sample

set; and the parameter dimension d. Intuitively, λ is small if the points are well separated, fill the ball from which they are drawn, and do not lie near any linear or quadratic paths (for the linear and quadratic approximations, respectively). As long as λ is held below some fixed finite value, the model is said to be λ-*poised*, and these bounds show that the approximations converge as $R \to 0$. The value of λ can be computed, and algorithms exist that can make any set λ-poised by adding a finite number of new points [19]. These simple but rigorous local error bounds form the foundation of our theoretical analysis, and are the reason that we begin with local low order polynomials. Usefully, they are representative of the general case, in that most reasonable local models converge in some sense as the ball size falls to zero.

It remains to precisely specify the choice of radius, R, and the weights used in the least squares regression. The radius R is selected to include a fixed number of points N. A linear model is fully defined by $N_{\text{def}} = d+1$ points and a quadratic is defined by $N_{\text{def}} = (d+1)(d+2)/2$ points; hence, performing a least squares regression requires at least this many samples. Such models are interpolating, but the associated least squares system is often poorly conditioned unless the geometry of the sample set is carefully designed. Conn *et al.* [19] show that adding additional samples can only stabilize the regression problem, so we select $N = \sqrt{d}N_{\text{def}}$, which seems to provide a reasonable balance.

We depart from [19] by performing a weighted regression. If the radius that contains the inner N_{def} samples is R_{def}, then $R > R_{\text{def}}$ and the weight of each sample is:

$$w_i = \begin{cases} 1 & \|\theta_i - \theta\|_2 \leq R_{\text{def}} \\ 0 & \|\theta_i - \theta\|_2 > R \\ \left(1 - \left(\frac{\|\theta_i - \theta\|_2 - R_{\text{def}}}{R - R_{\text{def}}}\right)^3\right)^3 & \text{else} \end{cases} \tag{3.2}$$

This is a variation of the tricube weight function sometimes used with LOESS, a local regression framework [16]. Setting the inner points to have unity weight ensures that the regression is full rank, while subsequently decreasing the weights to zero puts less emphasis on more distant samples. Additionally, this weight function has three continuous derivatives at $\|\theta_i - \theta\|_2 = R$, which ensures that even though $\tilde{\mathbf{f}}$ is generated independently at each point, it will have two continuous derivatives [3].

To satisfy the geometric constraint of the bounds above, Line 21 of Algorithm 3.1 checks that the sample set is at least λ-poised, and adds samples as needed. In practice, the choice of $N > N_{\text{def}}$ and the experimental design procedure used for selecting new points ensures that this constraint is almost never violated, and the error indicator would likely trigger refinement if it is. Hence, we may omit this step in our implementation.

Appendix A precisely details the regressor and the corresponding least squares problem. Multiple outputs are handled by constructing a separate approximation for each one. Fortunately, the expensive step of the least squares problem is identical for all the outputs, so the cost of constructing the approximation scales well with the number of observations.

3.2.3 Triggering model refinement

We separate the model refinement portion of the algorithm into two stages. This section discusses *when* refinement is needed, while Section 3.2.4 explains *how* the refinement is performed.

Refinement is triggered by either of two criteria. The first is random: with probability β_m, the model at the current point θ^- or the proposed point θ^+ is refined. This process fits naturally into MCMC and is essential to establishing the theoretical convergence results in the next section. The second criterion, based on a cross-validation error indicator, is intended to make the approximation algorithm efficient in practice. We compute separate error indicators, ϵ^+ and ϵ^-, that represent error in the acceptance probability due to errors in the local approximations at the proposed and current points, respectively, as computed by cross validation. Refinement is performed whenever these indicators exceed γ_m, at the point whose error indicator is larger. This second criterion is more difficult to analyze, however, and hence the two criteria are complementary.

For a Metropolis-Hastings algorithm, the acceptance probability computed from the forward model approximation $\tilde{\mathbf{f}}$ is

$$\alpha = \min\left(1, \frac{\mathcal{L}(\theta^+|\mathbf{d}, \tilde{\mathbf{f}})p(\theta^+)}{\mathcal{L}(\theta^-|\mathbf{d}, \tilde{\mathbf{f}})p(\theta^-)}\right).$$

(For simplicity, we assume a symmetric proposal distribution above; the generalization to non-symmetric proposals is straightforward.) The error indicators are computed as the maximum change in α that occurs when the jth sample is left out of either local approximation, $\tilde{\mathbf{f}}^+$ or $\tilde{\mathbf{f}}^-$, respectively. For each indicator, the maximum is taken over all $j = 1 \ldots N$ samples used in the weighted approximation; these definitions are written precisely in Algorithm 3.1. We emphasize that the acceptance probability is a natural quantity of interest in this context; it captures the entire impact of the forward model and likelihood on the MH kernel. The cross-validation error indicator is easily computable, summarizes a variety of error sources, and is easily interpretable as an additive error in a probability; these features make it possible for the user to exercise a problem-independent understanding of the threshold to which it is compared, γ_m. In contrast, attempting to control the error in either the forward model outputs or log-likelihood at the current or proposed point is not generically feasible, as their scale and the sensitivity of the MH kernel to their perturbations cannot be known *a priori*.

3.2.4 Refining the local model

If refinement of the local model at a point θ is required, SelectNewPoint selects a single new nearby point θ^*, computes $\mathbf{f}(\theta^*)$, and inserts the new pair into $\mathcal{S}$. This new model evaluation should improve the sample set for the local model $\mathcal{B}(\theta, R)$, either by allowing the radius R to decrease or by improving the local geometry of the sample set.

In previous work on derivative-free optimization, [19] attempts to select a new point optimally to improve a local model based on properties of polynomial interpolation, but is not concerned with the quality of the global set $\mathcal{S}$ because the samples are used to compute a single path to an optimum. In contrast with optimization, however, MCMC will revisit much of the parameter space many times, so our algorithm must ensure that local refinements maintain the global quality of the sample set, that is, the local quality at every nearby location.

Intuitively, local polynomial regression becomes ill-conditioned if the points do not fill the whole ball, or if some points are clustered much more tightly than others. The obvious strategy of simply adding θ to $\mathcal{S}$ is inadvisable because it often introduces tightly clustered points, inducing poorly conditioned regression problems. Instead, a straightforward and widely used type of experimental design is to choose points in a space-filling fashion; doing so near θ naturally fulfills our criteria. Specifically, we select the new point θ^* by finding a local maximizer of the problem:

$$\begin{aligned} \theta^* \quad = \quad & \arg\max_{\theta'} \ \min_{\theta_i \in \mathcal{S}} \|\theta' - \theta_i\|_2, \\ & \text{subject to } \|\theta' - \theta\|_2 \leq R \end{aligned}$$

which places a point within the ball but well separated from all existing samples. Optimization iterations are initialized at $\theta' = \theta$. The constraint ensures that the new sample is used in the improved model, yet the inner minimization operator considers the entire set $\mathcal{S}$ in order to ensure the sample's global quality. Inspection of the constraints reveals that the search in the inner minimization may be simplified to $\theta_i \in \mathcal{B}(\theta^*, 3R)$, as points outside a ball of radius $3R$ have no impact on the optimization. We seek a local optimum of the objective because it is far easier to find than the global optimum but also because it is more likely to be useful, as the global optimum will often be at radius R, which means that the revised model cannot be built over a smaller ball.

Although there is a close relationship between the set of samples where the forward model is evaluated and the posterior samples produced by MCMC, they are distinct and in general the two sets do not overlap. A potential limitation of the optimization approach above is that it might select points outside the support of the prior. This is only problematic if the model is not feasible outside the prior, in which case additional constraints can easily be added.

3.2.5 Local Gaussian process surrogates

Gaussian process (GP) regression underlies an important and widely used class of computer model surrogates, so it is natural to consider its application in the present local approximation framework. Local Gaussian processes have been previously explored in [109, 27, 106, 103, 46]. This section explains how local Gaussian process approximations may be substituted for the polynomial approximations described above, and incrementally refined during MCMC.

The adaptation is quite simple: we define a new approximation operator $\mathcal{G}^{\sim j}_{\mathcal{B}(\theta,R)}$ that may be substituted for the abstract operator $\mathcal{A}^{\sim j}_{\mathcal{B}(\theta,R)}$ in Algorithm 3.1. The samples are unweighted, and as in the polynomial case, the number of samples used for each approximation is a fixed constant N. This choice is relatively unconstrained; in the numerical examples to be shown later, we choose $N = d^{5/2}$, mimicing the choice for quadratic approximations, which performs well in practice. In contrast with low order polynomials, Gaussian processes involve several implementation decisions that can impact the quality of the approximation: one must choose a mean function, a covariance kernel, and a hyperparameter selection strategy. The choices we make here are intended to be simple, not to explore the entire space of options considered in the literature. We select a constant mean function and an anisotropic squared-exponential covariance kernel, that is:

$$C(\mathbf{x}, \mathbf{y}) = \sigma^2 \exp\left(-\sum_{i=1}^{d} \xi_i (x_i - y_i)^2\right),$$

with variance σ^2 and inverse-squared correlation lengths $\xi_i > 0$. Here x_i and y_i are the ith components of $\mathbf{x}, \mathbf{y} \in \Theta$. The length scale parameters are chosen by maximizing the marginal likelihood, as a fully Bayesian approach to inferring these hyperparameters would be prohibitively expensive. Given the inverse length parameters, the mean function is chosen by least squares and the variance σ^2 is estimated from the remaining misfit. In practice, care must be taken in performing these steps to ensure that a high quality surrogate is produced; our implementation borrows from the DACE toolbox [75], which uses parameter rescaling and a custom optimization algorithm to mitigate the impact of local minima, which can cause difficulties in fitting Gaussian processes.

A powerful feature of Gaussian process regressors is that they directly attempt to quantify the variance of their predictions, in the sense that at any point θ, the approximation yields a predictive distribution $\tilde{\mathbf{f}}(\theta) \sim \mathcal{N}(\mu(\theta), \tilde{\sigma}^2(\theta))$ for the model outputs [97]. This renders the leave-one-out error estimation procedure unnecessary. Instead, we define $\mathcal{G}^{\sim j}_{\mathcal{B}(\theta,R)}$ to be the mean $\mu(\theta)$ of the local Gaussian process when $j = \emptyset$ and a draw from the Gaussian predictive distribution otherwise. This definition allows the existing algorithm to leverage the natural uncertainty quantification of Gaussian processes in computing the error indicators ϵ^+ and ϵ^-, without further modification.

As with local polynomials, multiple outputs of the forward model are handled by constructing a separate approximation for each one. Using this simple approach, parameters of the Gaussian process covariance kernel must be inferred independently for each output. Hence, indirect approximation with GPs is not immediately practical for large n; one might instead employ GP approximations designed for high-dimensional outputs [55].

3.2.6 Related work

Having explained our algorithm, we can now provide a more detailed comparison with related work. Previous efforts at mitigating the cost of MCMC with computationally intensive models have focused primarily on global approximations. Gaussian process emulators are perhaps the most widely studied non-intrusive surrogate of this kind, as in Sacks *et al.* [97] and Kennedy & O'Hagan [70]. Samples are typically selected by a space-filling design over the prior (e.g., a Latin hypercube design), possibly followed by sequential design to further refine the approximation in regions of interest. Constructing a GP emulator for a complex computer model may require a large sample set wherein not all the points are well separated, however. Haaland and Qian [49] discuss the difficulties of constructing GPs in this context and analyze a numerically stable and accurate multi-step procedure for doing so. Gramacy and Lee [47] introduce treed Gaussian processes as a strategy for allowing GPs to capture nonstationarity. They develop fully Bayesian methods for partitioning the parameter space, but this approach is likely to be too expensive to interleave with an outer loop of MCMC. Rasmussen [91], on the other hand, employs Gaussian process approximations specifically for the purpose of making Hybrid Monte Carlo (HMC) more affordable, by integrating the associated Hamiltonian dynamics using evaluations of the approximation. The full model is run on the final proposal to correct for errors in the simulated trajectory, but this step limits possible performance improvements. The GP approximation is refined a finite number of times by observing HMC trajectories and triggering new model evaluations whenever the accumulated error estimate exceeds a threshold. This coupling with HMC implicitly introduces an experimental design strategy, but lacks any guarantee of good global approximation properties of the sample set.

Closely related to Gaussian process emulators are radial basis function (RBF) or kernel approximations. Bliznyuk *et al.* construct an RBF interpolant of the log-posterior density directly [7, 8], and explicitly map the high posterior density region, which aids in the experimental design process. Then MCMC is restricted to this high posterior density region, as it is the only region where the approximated density is intended to be accurate; this is a practical, but limiting, solution. Fielding *et al.* [34] extend the approach in [91] by including parallel tempering and by allowing evaluations of the GP surrogate to determine the Metropolis acceptance ratio, at least for higher-temperature chains. Samples are confined to the region of accurate approximation via a penalty introduced into the posterior. The DoIt approximation [63, 64] observes that carefully constructed Gaussian radial basis function approximations of the posterior (or square-root posterior density) can be cheaply and analytically processed to directly evaluate posterior summaries, avoiding MCMC altogether. Experimental design is accomplished by locally optimizing the maximum predicted variance near an existing sample with large leave-one-out cross-validation error.

Another family of approximations involves constructing global polynomial surrogates with respect to the prior measure, referred to as *polynomial chaos expansions* [80, 117, 118, 44]. These approximations can be constructed non-intrusively and efficiently through the use of adaptive Smolyak methods, which evaluate the model on

tailored sparse grids [22, 20]. But these grids are typically not adapted to the posterior, making them inefficient if the data are informative.

Almost all of the approaches above involve creating a single, essentially "global," approximation of the forward model or posterior density. The use of local models as surrogates for inference, on the other hand, has been relatively unexplored. Local models have a long heritage, as typified by LOESS (locally weighted scatterplot smoothing or local regression) [16], which uses local weights to fit low order polynomials [17, 51]. Another well known approximation is multivariate adaptive regression splines (MARS) [37], which computes a global model composed of local functions that are introduced, shifted, and scaled as needed to fit the data. Some advantages of local models as they apply to artificial intelligence were identified in [3] and more recently in an approach that attempts to scale LOESS to high-dimensional spaces [110]. Recent work has also coupled high-order global surrogates with local models derived from MARS [13]. Local models are particularly important in optimization, as they are used in many derivative-free trust region methods [18, 19]. Although this issue is not fully explored in the present work, we also note that local models tend to handle non-smooth functions well, in that the consequent regions of poor approximation can be contained within a decreasing radius of a discontinuity or other non-smooth feature. This behavior contrasts sharply with that of global polynomials, for example, which are known to converge slowly if there is a loss of regularity anywhere in the input domain.

While the present work focuses entirely on "black-box" forward models, where only evaluations of the forward model are available, some interesting approaches leverage the structure of the forward model to reduce the cost of inference. For example, if coarsened versions of a PDE forward solver are available, they can be used to run a family of parallel MCMC chains, where the coarse-scale chains are used to increase the mixing speed of the fine-scale chains [56], or to delay evaluating the fine-scale model until the coarse model suggests that a proposed point should be accepted [14, 31]. Unfortunately, these methods still run the fine-scale model at least once for every new posterior sample, so that the overall cost of inference cannot be dramatically reduced. In two slightly different settings, [72] and [5] observe that sometimes the forward model can be computed incompletely, while still allowing MCMC to decide whether to accept or reject a proposal: the former attempts to process only some of the data in the likelihood computation and the latter tunes the accuracy of a Monte Carlo simulation within the forward model. For certain types of forward models, e.g., particular classes of PDEs or ODEs, reduced order modeling can produce high-quality approximations at significantly reduced computational cost via a pre-processing step that projects the governing equations onto a low-dimensional subspace of the original system state [36, 74].

These previous efforts at creating efficient surrogate-based inference procedures show enormous potential for reducing the computational cost of inference, but also illustrate the subtlety of the problem—as none manage the delicate balance required to achieve practical performance while ensuring exact sampling, or sampling with

known error bounds. While some of the approaches discussed above achieve exact posterior sampling by using the original forward model in a correction step, most other approaches do not sample from the original posterior. Some relevant theory is provided by Cotter *et al.* [23], who prove that good approximation (in a particular global sense) of the forward model or likelihood can ensure that the corresponding approximate posterior is close to the true posterior. These results should apply to a broad range of forward model approximation techniques, but without useful strategies to quantify or control the approximation error, the theory does not necessarily provide practical guidance. The present local approximation scheme, on the other hand, lets us tie construction and refinement of the approximation to the MCMC transition kernel itself. Then, by relying on the convergence properties of local models, we will be able to guarantee asymptotically exact posterior sampling.

3.3 Theory

In this section we show that, under natural assumptions, Algorithm 3.1 converges to the target posterior $p(\theta|\mathbf{d})$ asymptotically. We focus only on the symmetric random walk Metropolis algorithm, although our arguments can be adapted to more sophisticated variants such as the Langevin algorithm, the Hybrid Monte Carlo algorithm, and the Gibbs sampler.

Denote by $L(x, \cdot)$ the kernel on $\mathbb{R}^d$ used to generate new proposals in Algorithm 3.1 and let $\ell(x, dy)$ denote its density. Assume that $L(x, \cdot)$ satisfies

$$L(x, S) = L(x + y, S + y) \tag{3.3}$$

for all points $x, y \in \Theta$ and all sets $S \subset \Theta \subset \mathbb{R}^d$. Similarly, denote by $P(x, \cdot)$ the Metropolis-Hastings kernel (with density $p(x, dy)$) associated with proposal kernel L and target distribution $p(\theta|\mathbf{d})$.

Let $\tilde{P}_t(X_t, \cdot)$ and $\tilde{p}_t(x, dy)$ respectively denote the transition distribution and density associated with the stochastic process X_t evolving according to Algorithm 3.1, ignoring at step t any refinements that would normally be made. Here we assume that $\log p(\theta|\mathbf{d})$ is approximated by quadratic interpolation on the $N = N_{\text{def}}$ nearest points. Denote by $\alpha(x, y)$ and $\tilde{\alpha}_t(x, y)$ the Metropolis-Hastings acceptance ratios associated with P and $\tilde{P}$ respectively. Finally, let R_t be the value of R_{def} at time t and denote by $q_t^1, \ldots, q_t^N$ the points in $\mathcal{S}$ within distance R_t of X_t.

3.3.1 Assumptions

Fix β_t, γ, and λ to be as in Algorithm 3.1. For $\theta \in \Theta$, define the sets

$$A(\theta) = \Big\{ y \,:\, p(y|\mathbf{d}) \geq p(\theta|\mathbf{d}) \Big\}$$

and

$$R^r(\theta) = \Big\{ y \,:\, 2\theta - y \notin A(\theta) \Big\}.$$

Assumptions 3.3.1. *The proposal kernel L and the target $p(\theta|\mathbf{d})$ satisfy the following:*

1. $\liminf_{|\theta|\to\infty} \int \left[1 - \left(\min\left(1, \sqrt{\frac{p(\theta+z|\mathbf{d})}{p(\theta|\mathbf{d})}}\right)\right)^2\right] \ell(\theta, \theta + dz) > 0.$
2. $\limsup_{|\theta|\to\infty} \int_{R^r(\theta)-\theta} \left[\left(\min\left(1, \sqrt{\frac{p(\theta|\mathbf{d})}{p(\theta+z|\mathbf{d})}}\right)\right) - \sqrt{\frac{p(\theta-z|\mathbf{d})}{p(\theta|\mathbf{d})}}\right] \ell(\theta, \theta + dz) \leq 0.$
3. *$p(\theta|\mathbf{d})$ is bounded away from 0 on compact sets.*
4. *There exist $\delta, \epsilon > 0$ so that, for every x, $|x - y| < \delta$ implies $\ell(x, y) > \epsilon$.*

Next we assume that the posterior density $p(\theta|\mathbf{d})$ satisfies the following *Gaussian envelope condition*:

Assumptions 3.3.2. *The posterior density $p(\theta|\mathbf{d})$ satisfies:*

$$\lim_{r\to\infty} \sup_{|\theta|=r} |\log p(\theta|\mathbf{d}) - \log p_\infty(\theta)| = 0 \tag{3.4}$$

where $p_\infty(\theta) = p_\infty(\theta_1, \theta_2, \ldots, \theta_d)$ has the form

$$\log p_\infty(\theta_1, \theta_2, \ldots, \theta_d) = \sum_{1\leq i\leq j\leq d} a_{ij}\theta_i\theta_j$$

with the matrix $[a_{ij}]$ being negative definite.

For $x \in \Theta$, define the *Lyapunov function*

$$V(x) = \frac{1}{\sqrt{p_\infty(x)}}. \tag{3.5}$$

Assumptions 3.3.3. *Let V be as defined in (3.5) and set $V_\epsilon(x) = V(x)^{\frac{1}{1+\epsilon}}$ for $\epsilon > 0$. Assume that there exists some $\epsilon_0 > 0$ so that, for all $\epsilon > \epsilon_0$ and all $x > \mathcal{X}$ sufficiently large, there exists a constant $\mathcal{C}_\epsilon$ such that*

$$\frac{|\int_z (V_\epsilon(x) - V_\epsilon(z))\ell(x, dz)|}{V_\epsilon(x)} \leq \mathcal{C}_\epsilon < \infty. \tag{3.6}$$

Before giving the main result, we briefly discuss the assumptions above.

1. Assumption 3.3.1 constitutes a widely used set of natural conditions which ensure that the Metropolis-Hastings algorithm associated with proposal L is geometrically ergodic [94]. This assumption is easy to verify for a large class of target densities $p(\theta|\mathbf{d})$ and proposal kernels L. Furthermore, Assumption 3.3.1 also

implies that the function V is a Lyapunov function for the chain with proposal distribution L and target p_∞. Indeed, by Theorem 3.2 of [94], we have that a Markov chain Z_t evolving in this way satisfies the inequality

$$\mathbb{E}[V(Z_{t+1})|Z_t = x] \leq \alpha V(x) + b$$

for some $0 < \alpha < 1$ and some $b > 0$.

2. Assumption 3.3.2 signifies a delicate interplay between the deterministic approximation algorithm (in our case, a quadratic interpolation) and the stability of the corresponding MCMC algorithm. This assumption can be much weakened; see Remark 3.3.6 for more elaboration of this point.

3. Assumption 3.3.3 is a mild technical assumption. For example, in one dimension, (3.6) holds if $\ell(x, dy)$ is Gaussian with variance small compared to p_∞.

3.3.2 Ergodicity

We are ready to give the main result of this section. Recall that X_t is the Markov chain on $\Theta \subset \mathbb{R}^d$. The state space Θ need not be compact.

Theorem 3.3.4. *Suppose Assumptions 3.3.1, 3.3.2, and 3.3.3 hold. Then we have*

$$\lim_{t\to\infty} \|\mathcal{L}(X_t) - p(\theta|\mathbf{d})\|_{\text{TV}} = 0.$$

Now, if we assume that Θ is indeed compact, we have the same result under much weaker assumptions:

Theorem 3.3.5. *Suppose Θ is compact and $p(\theta|\mathbf{d})$ is bounded away from 0 on compact sets. Then we have*

$$\lim_{t\to\infty} \|\mathcal{L}(X_t) - p(\theta|\mathbf{d})\|_{\text{TV}} = 0.$$

Remark 3.3.6. *As mentioned before, the Gaussian envelope assumption made in Assumption 3.3.2 can be weakened. We mention without proof that results analogous to Theorem 3.3.4 hold when $\log p_\infty(x)$ from Equation (3.4) is a polynomial of degree m if we also change Algorithm 3.1 to use an approximating polynomial of degree at least m. We also conjecture that this modification to Algorithm 3.1 is not necessary. The only difficulty is in establishing some control over our estimates of $p(X_t|\mathbf{d})$ when X_t is very far from all points of $\mathcal{S}$, as will happen occasionally during real runs. If the approximations made in our algorithm are globally poor, as when we approximate a degree-m polynomial with one of lower degree, an analogue to Lemma 3.3.15 below will not hold.*

Finally, it is easy to check that results analogous to Theorem 3.3.4 hold under the very mild condition that $\log p_\infty(x)$ from Equation (3.4) is concave, if we also change Algorithm 3.1 to use a linear approximation.

Our main tool will be the following consequence of the main Theorem from [88]:

Theorem 3.3.7 (Ergodicity of Adaptive Chains). *Fix a sequence of kernels* $\{K_t\}_{t\in\mathbb{N}}$ *and a kernel of interest* K_∞ *with stationary distribution* π. *Assume that* K_∞ *satisfies*

$$\|K_\infty^t(x,\cdot) - \pi(\cdot)\|_{\text{TV}} \leq C_x(1-\alpha)^t \tag{3.7}$$

for some $\alpha > 0$ *and all* $t > 0$, *with* $C_{\mathcal{U}} \equiv \sup_{x\in\mathcal{U}} C_x < \infty$ *for all compact sets* $\mathcal{U} \subset \Theta$. *Also assume that* $\forall \epsilon > 0$, *there exist compact sets* $\mathcal{A}_\epsilon \subset \mathcal{B}_\epsilon \subset \Theta$ *with the property:*

$$\sup_{x\in\mathcal{A}_\epsilon, t\in\mathbb{N}} K_\infty^t(x, \mathcal{B}_\epsilon^c) \leq \frac{\epsilon \log(1-\alpha)}{4\log\left(\frac{\epsilon}{4C_{\mathcal{A}_\epsilon}}\right)}. \tag{3.8}$$

Furthermore, assume that $\forall \epsilon > 0$, *there exists* $T(\epsilon)$ *so that*

$$\sup_{x\in\mathcal{A}_\epsilon} \mathbb{P}[X_s \notin \mathcal{A}_\epsilon \,\forall\, s \in (S, S+T) | X_S \in \mathcal{A}_\epsilon] \leq \frac{\epsilon}{4} \tag{3.9}$$

for all $T > T(\epsilon)$ *and so that*

$$\sup_{t\geq T(\epsilon),\, x\in\mathcal{B}_\epsilon} \|K_t(x,\cdot) - K_\infty(x,\cdot)\|_{\text{TV}} \leq \frac{\epsilon \log(1-\alpha)}{4\log\left(\frac{\epsilon}{4C_{\mathcal{A}_\epsilon}}\right)}. \tag{3.10}$$

Then, for all $x \in \mathcal{A}_\epsilon$ *and all* $T > 2T(\epsilon) + \frac{\log\left(\frac{\epsilon}{4C_{\mathcal{A}_\epsilon}}\right)}{\log(1-\alpha)}$, *we have that:*

$$\|\mathcal{L}(X_T) - \pi\|_{\text{TV}} < \epsilon$$

when $X_0 = x$.

Remark 3.3.8. *Although there are many assumptions in Theorem 3.3.7, they effectively serve two purposes:*

- *Condition (3.7) implies that the Markov chain corresponding to the kernel* K_∞ *is geometrically ergodic. Inequalities (3.7) and (3.10) ensure that, on compact sets, the limiting chain mixes quickly and the chain of interest remains quite close to it.*

- *The remaining assumptions are a fairly weak collection of inequalities that, taken together, ensure that the chain of interest will eventually spend most of its time in a sufficiently large compact set.*

Taken together, Theorem 3.3.7 says that an approximating chain will converge to its limiting chain if the limiting chain mixes well, the approximation is good on an increasing sequence of compact sets, and eventually the approximating chain remains within these compact sets.

3.3.3 Proof of ergodicity for compact parameter space

In this section we give the proof of Theorem 3.3.5, which is slightly simpler than the proof of Theorem 3.3.4. The main requirement is the following lemma:

Lemma 3.3.9 (Convergence of Kernels)**.** *Consider a chain following the same assumptions as Theorem 3.3.5. For all $\epsilon, \delta > 0$ and compact sets $\mathcal{A} \subset \Theta$, there exists $T(\epsilon, \delta, \mathcal{A}) > 0$ so that:*

$$\mathbb{P}\left(\sup_{t > T(\epsilon,\delta,\mathcal{A})} \sup_{x \in \mathcal{A}} \|P(x,\cdot) - \tilde{P}_t(x,\cdot)\|_{\text{TV}} < \epsilon\right) > 1 - \delta.$$

Proof of Lemma 3.3.9. This follows from two short arguments. First:

Lemma 3.3.10 (Grid Refinements Help)**.** *Fix a compact set $\mathcal{A} \subset \Theta$, a measure ν that is not singular with respect to Lebesgue measure, and $\epsilon, \delta > 0$. Then there exists $M(\epsilon, \delta, \nu) \in \mathbb{N}$ so that any approximation $\hat{p}\,(\theta|\mathbf{d})$ based on at least $M(\epsilon, \delta, \nu)$ independent draws from ν satisfies:*

$$\mathbb{P}[\sup_{\theta \in \Theta} |p(\theta|\mathbf{d}) - \hat{p}(\theta|\mathbf{d})| > \epsilon] < \delta.$$

Proof of Lemma 3.3.10. Say that a collection of points $\mathcal{F}$ covers a compact set $\mathcal{A}$ up to distance $r > 0$ if every point in $\mathcal{A}$ is within r of some point in $\mathcal{F}$. Next, let $\{X_i\}_{i=1}^M$ be a collection of i.i.d. draws from ν, and let $\mathcal{E}_r$ be the event that they cover $\mathcal{A}$ up to distance r. By the multidimensional version of the Glivenko-Cantelli theorem,

$$\lim_{M \to \infty} \mathbb{P}[\mathcal{E}_r] = 1 \tag{3.11}$$

for all $r > 0$.

By results in [19][4], for any $\lambda, \alpha > 0$, there exists a $r = r(\alpha, \lambda) > 0$ so that the approximation $\hat{p}\,(\theta|\mathbf{d})$ based on any λ-poised collection of points that covers $\mathcal{A}$ up to distance r is within α of $p(\theta|\mathbf{d})$. Setting $\alpha = \epsilon$ and combining this with (3.11) completes the proof. □

Next,

Lemma 3.3.11 (Grid Refinements Occur)**.** *There exist an increasing sequence of compact sets $\mathcal{A}_n \subset \Theta$, with $\Theta = \cup_{n>0} \mathcal{A}_n$, with the properties:*

- $\sum_{t=0}^{\infty} \mathbf{1}_{X_t \in \mathcal{A}_n} = \infty$ *with probability 1.*
- *For every n, there exist $k = k(n) > 0$, $\epsilon = \epsilon(n) > 0$ and measure $\mu = \mu_n$ with support equal to $\mathcal{A}_n$ so that, for all $x \in \mathcal{A}_n$, we have:*

$$K_\infty^k(x,\cdot) = \epsilon\mu(\cdot) + (1-\epsilon)r_x(\cdot)$$

[4]The required result is a combination of Theorems 3.14 and 3.16, as discussed in the text after the proof of Theorem 3.16.

for some remainder measure r_x.

Proof of Lemma 3.3.11. Since Θ is compact, we can just set $\mathcal{A}_n = \Theta$ for all $n \in \mathbb{N}$. Since the t'th step of the chain is added to $\mathcal{S}$ in step 12 of Algorithm 3.1 with probability β, independently of the previous steps of the algorithm, the first claim follows from the Borel-Cantelli lemma. The second claim is an immediate consequence of item 4 of Assumption 3.3.1. □

We are now ready to prove Lemma 3.3.9: Choose n so that $\mathcal{A} \subset \mathcal{A}_n$; such an n exists by the compactness of $\mathcal{A}$. By Lemma 3.3.11, we can construct an infinite sequence of random times $\{\tau_i\}_{i\in\mathbb{N}}$ so that X_{τ_i} are an i.i.d. sequence of random draws from some fixed measure μ that is not singular with respect to Lebesgue measure. The result then follows immediately from an application of Lemma 3.3.10. □

We finally prove Theorem 3.3.5:

Proof. Fix some $\epsilon > 0$. It is sufficient to find sets $\mathcal{A}_\epsilon \subset \mathcal{B}_\epsilon$ and time $T(\epsilon)$ that satisfy the conditions of Corollary 3.3.7. Inequality (3.7) follows from Theorem 3.2 of [94] and Assumptions 3.3.1. We set $\mathcal{A}_\epsilon = \mathcal{B}_\epsilon = \Theta$ for all $\epsilon > 0$; thus inequalities (3.8) and (3.9) hold automatically. Finally, the existence of some time $T(\epsilon)$ satisfying inequality (3.10) follows from Lemma 3.3.9. □

Remark 3.3.12. *Lemma 3.3.11 is the only place in the proof of Theorem 3.3.5 in which we use the assumption $\beta > 0$. Since the proof remains correct as stated as long as we have $\sum_t \beta_t = \infty$, Theorem 3.3.5 holds whenever $\beta_t \geq C/t$ for some constant $C > 0$. As will be seen in Example 3.3.17 below, this condition is sharp.*

3.3.4 Drift at infinity

In this section, we show that under the assumptions of Theorem 3.3.4, the chain X_t satisfies a drift condition when $|X_t|$ and t are both sufficiently large. We begin by showing that the approximation $\hat{p}_t(X_t|\mathbf{d})$ of the posterior used at time t is close to $p_\infty(X_t)$ when $|X_t|$ and $|X_t| - R_t$ are sufficiently large:

Lemma 3.3.13 (Approximation at Infinity)**.** *For all $\epsilon > 0$, there exists a constant $\mathcal{X} = \mathcal{X}(\epsilon) > 0$ so that, if $|X_t| - R_t > \mathcal{X}$ and the set $\{q_t^{(1)}, \ldots, q_t^{(N)}\}$ is λ-poised, then*

$$|\log \hat{p}_t(X_t|\mathbf{d}) - \log p_\infty(X_t)| < \epsilon.$$

Proof. Fix $\epsilon > 0$. By (3.4) in Assumption 3.3.2, there exists some $\mathcal{X} = \mathcal{X}(\epsilon)$ so that $x > \mathcal{X}$ implies $|\log(p(x|\mathbf{d})) - \log(p_\infty(x))| < \frac{\epsilon}{\ell\lambda}$. We use this constant in the remainder of the proof.

Denote by $\{f_i\}_{i=1}^{\ell(t)}$ the Lagrange polynomials associated with the set $\{q_t^{(1)}, \ldots, q_t^{(N)}\}$. By Lemma 3.5 of [19], we have

$$|\log(\hat{p}_t(X_t|\mathbf{d})) - \log(p_\infty(X_t))| = |\sum_i f_i(X_t) \log(p(q_t^{(i)}|\mathbf{d})) - \log(p_\infty(X_t))|$$

$$
\begin{aligned}
&\leq |\sum_i \log(p_\infty(q_t^{(i)}))f_i(X_t) - \log(p_\infty(X_t))| \\
&+ \sum_i |\log(p(q_t^{(i)}|\mathbf{d})) - \log(p_\infty(q_t^{(i)}))|\,|f_i(X_t)| \\
&\leq 0 + \ell\lambda \sup_i |\log(p(q_t^{(i)}|\mathbf{d})) - \log(p_\infty(q_t^{(i)}))|,
\end{aligned}
$$

where the first equality of the last line is from the definition of Lagrange polynomials and the second is from Definition 4.7 of [19]. The conclusion now follows from the definition of $\mathcal{X}$ above. □

Next, we show that a good approximation in total variation implies drift:

Lemma 3.3.14 (Drift at Infinity)**.** *For any $\epsilon > \epsilon_0$ and $\mathcal{Y} > 0$, there exists some $\delta, \mathcal{X}, b > 0$ and $0 < \alpha < 1$ so that, if*

$$
\sup_{|y|>\mathcal{Y}} |\log(\tilde{p}_t(y|\mathbf{d})) - \log(p(y|\mathbf{d}))| < \delta, \tag{3.12}
$$

then for all $|x| > \mathcal{X}$, we have:

$$
\mathbb{E}[V_\epsilon(X_{t+1})|X_t = x] \leq \alpha V_\epsilon(x) + b. \tag{3.13}
$$

Proof. Let Z_t be a Metropolis-Hastings Markov chain with the same proposal distribution as X_t but target distribution p_∞, and acceptance probability α_∞. By Theorem 3.2 of [94], Z_t satisfies inequality (3.13) above in the sense that

$$
\mathbb{E}[V(Z_{t+1})|Z_t = x] \leq \alpha V(x) + b
$$

for some $0 < \alpha < 1$ and some $b > 0$. By Jensen's inequality, for all $\epsilon > 0$, we also have

$$
\mathbb{E}[V_\epsilon(Z_{t+1})|Z_t = x] \leq \alpha_\epsilon V_\epsilon(x) + b_\epsilon
$$

for some $0 < \alpha_\epsilon < 1$ and some $b_\epsilon > 0$.

We introduce some general notation to help with the definition of a coupling. For a generic Metropolis-Hastings kernel Q with density $q(x, y)$ and target density $p(\theta|\mathbf{d})$, we represent the associated Metropolis-Hastings chain $\{Z_t\}_{t\in\mathbb{N}}$ by:

$$
\begin{aligned}
Z_{t+1} &= q_t \ : \ \alpha(Z_t, q_t) < U_t \\
Z_{t+1} &= Z_t \ : \ \alpha(Z_t, q_t) \geq U_t,
\end{aligned} \tag{3.14}
$$

where $\alpha(x, y) = \min\left(1, \frac{p(y|\mathbf{d})q(y,x)}{p(x|\mathbf{d})q(x,y)}\right)$ is the *acceptance ratio*, q_t is drawn from $Q(X_t, \cdot)$, and $\{U_s\}_{s\in\mathbb{N}}$ is an i.i.d. sequence of uniform random variables on $[0, 1]$.

Continuing with the proof, assume $X_t = x$. Also, let $\{Y_s\}_{s\in\mathbb{N}}$ be a copy of the Metropolis-Hastings chain with proposal kernel L and target distribution $p(\theta|\mathbf{d})$, started at $Y_t = x$. Couple $X_{t+1}, Y_{t+1}, Z_{t+1}$ so that they both make the same choices of

update variables in the representation (3.14). Then, for any $\epsilon > 0$, we have:

$$\begin{aligned}\mathbb{E}[V_\epsilon(X_{t+1})|X_t = x] &= \mathbb{E}[V_\epsilon(Z_{t+1})|Z_t = x] + \mathbb{E}[V_\epsilon(X_{t+1}) - V_\epsilon(Z_{t+1})|X_t = Z_t = x] \\ &\leq \alpha_\epsilon V_\epsilon(z) + b_\epsilon + \mathbb{E}[V_\epsilon(X_{t+1}) - V_\epsilon(Z_{t+1})|X_t = Z_t = x]. \end{aligned} \tag{3.15}$$

By Assumption 3.6 and Equation (3.12), we have for $\epsilon > \epsilon_0$:

$$\begin{aligned}&|\mathbb{E}[V_\epsilon(X_{t+1}) - V_\epsilon(Z_{t+1})]| \\ &= \Big| \int_{z:\alpha_\infty(x,z)>\tilde{\alpha}_t(x,z)} (\alpha_\infty(x,z) - \tilde{\alpha}_t(x,z))\,(V(x) - V(z))\ell(x,dz) \\ &\qquad + \int_{z:\alpha_\infty(x,z)<\tilde{\alpha}_t(x,z)} (\tilde{\alpha}_t(x,z) - \alpha_\infty(x,z))\,(V(z) - V(x))\ell(x,dz)\Big| \\ &\qquad + 2L(x,\mathcal{B}_\mathcal{X}(0)) \sup_{|p|<\mathcal{X}} V_\epsilon(p) \\ &\leq \|P_\infty(x,\cdot) - \tilde{P}_t(x,\cdot)\|_{\mathrm{TV}} \Big| \int_z (V(x) - V(z))\ell(x,dz)\Big| + 2L(x,\mathcal{B}_\mathcal{Y}(0)) \sup_{|p|<\mathcal{Y}} V_\epsilon(p) \\ &\leq \mathcal{C}\delta V_\epsilon(x) + 2L(x,\mathcal{B}_\mathcal{Y}(0)) \sup_{|p|<\mathcal{Y}} V_\epsilon(p). \end{aligned} \tag{3.16}$$

Equations (3.15) and (3.16) together imply that

$$\begin{aligned}\mathbb{E}[V_\epsilon(X_{t+1})|X_t = x] &\leq \alpha_\epsilon V_\epsilon(z) + b_\epsilon + \mathcal{C}\delta V_\epsilon(x) + 2L(x,\mathcal{B}_\mathcal{Y}(0)) \sup_{|p|<\mathcal{Y}} V_\epsilon(p) \\ &= (\alpha_\epsilon + \mathcal{C}\delta)V_\epsilon(x) + b_\epsilon + 2L(x,\mathcal{B}_\mathcal{Y}(0)) \sup_{|p|<\mathcal{Y}} V_\epsilon(p).\end{aligned}$$

We note that

$$\lim_{\mathcal{X}\to\infty} \sup_{|x|>\mathcal{X}} L(x,\mathcal{B}_\mathcal{Y}(0)) = 0, \quad \lim_{\mathcal{X}\to\infty} \frac{\sup_{|p|<\mathcal{Y}} V_\epsilon(p)}{\sup_{|q|>\mathcal{X}} V_\epsilon(q)} = 0.$$

Thus, choosing δ sufficiently small and $\mathcal{X}$ sufficiently large for any fixed $\mathcal{Y}$, the lemma follows immediately. □

Next, we need to show that, for $|X_t|$ and t sufficiently large, we can also have $|X_t| - R_t$ arbitrarily large:

Lemma 3.3.15 (Approximations At Infinity Ignore Compact Sets)**.** *Fix any $\mathcal{X} > 0$, and define*

$$\tau_\mathcal{X} = \sup\Big\{t \,:\, |X_t| > 2\mathcal{X},\, |X_t| - R_t < \mathcal{X}\Big\}.$$

Then

$$\mathbb{P}[\tau_\mathcal{X} < \infty] = 1.$$

Proof. Fix $0 < r_2 < r_1$. Next, define θ_t to be the ray from the origin to X_t. Also

define $\mathcal{B}_r(x)$ to be the ball of radius r around x. Note that, for any $0 < \alpha < 1$,

$$\mathcal{B}_{\alpha|X_t|-\frac{r_1+r_2}{2}}(\alpha X_t) \subset \mathcal{B}_{|X_t|-\frac{r_1+r_2}{2}}(X_t). \tag{3.17}$$

We also note that there exists some $\delta = \delta(r_1, r_2)$ so that, if $p \in B_{r_2}(0)$ and $X_t \notin B_{r_1}(0)$ and the angle between the ray p and θ_t is less than δ, then $p \in \mathcal{B}_{|X_t|-\frac{r_1+r_2}{2}}(X_t)$.

Next, fix a finite covering $\{\mathcal{P}_i\}$ of the surface of $\mathcal{B}_{\frac{r_1+r_2}{2}}(0)$ with the property that any ball $\mathcal{B}$ of radius at least δ in $\frac{r_1+r_2}{2}\mathcal{S}^d$ contains at least one entire set in the covering. We will show that, for every element $\mathcal{P}_i$ of the cover, either $|\mathcal{P}_i \cap \mathcal{S}|$ is eventually larger than N or $|\mathcal{P}_i \cap \{X_t\}_{t\in\mathbb{N}}| < \infty$.

To see this, we introduce a representation of the random variables used in step 12 of Algorithm 3.1. Recall that in this step, X_t is added to $\mathcal{S}$ with probability β, independently of the rest of the history of the walk. We split up the sequence B_t of Bernoulli(β) random variables according to the covering. In particular, for each element $\mathcal{P}_i$ of the covering, let $\{B_t^{(i)}\}_{t\in\mathbb{N}}$ be an i.i.d. sequence of Bernoulli random variables with success probability β. The k'th time that X_t is in $\mathcal{P}_i$, we use $B_k^{(i)}$ as the indicator function in step 12 of Algorithm 3.1. This does not affect the steps that the algorithm takes.

By the Borel-Cantelli lemma, we have for each i that $\mathbb{P}[B_t^{(i)} = 1, \text{infinitely often}] = 1$. We note that, if $B_t^{(i)} = 1$ infinitely often, then $|\mathcal{P}_i \cap \{X_t\}_{t\in\mathbb{N}}| = \infty$ implies that for all $M < \infty$, we have $|\mathcal{P}_i \cap \mathcal{S}| > M$ eventually. Let $\mathcal{C}_i$ be the event that $|\mathcal{P}_i \cap \mathcal{S}| > N$ eventually and let $\mathcal{D}_i$ be the event that $|\mathcal{P}_i \cap \{X_t\}_{t\in\mathbb{N}}| = \infty$. Then this argument implies that

$$\mathbb{P}[\mathcal{C}_i|\mathcal{D}_i] = 1.$$

Since there are only finitely many parts $\mathcal{P}_i$ of the partition, we have

$$\mathbb{P}[\cap_i (\mathcal{C}_i \cup \mathcal{D}_i^c)] = 1. \tag{3.18}$$

Thus, on the almost sure event $\cap_i (\mathcal{C}_i \cup \mathcal{D}_i^c)$, all sets $\mathcal{P}_i$ that X_t visits infinitely often will also contribute points to $\mathcal{S}$ infinitely often.

Let $I = \{i : |\mathcal{P}_i \cap \{X_t\}_{t\in\mathbb{N}}\}$, and let $\tau^-_{r_1,r_2} = \inf\{t : \forall i \in I, |\mathcal{P}_i \cap \mathcal{S}| \geq N\}$. Then, by the above discussion, $\tau_{\mathcal{X}}$ is bounded from above by $\max(\tau^-_{2|\mathcal{X}|,|\mathcal{X}|}, \tau^+_{2|\mathcal{X}|,|\mathcal{X}|})$. Since both of those times are almost surely finite by inequality (3.18), $\tau_{\mathcal{X}}$ is also almost surely finite. This completes the proof. □

Finally, we put these arguments together to show that some compact set is returned to infinitely often:

Lemma 3.3.16 (Infinitely Many Returns)**.** *There exists a compact set which is recurrent with probability 1. Furthermore, for all $\delta > 0$, there exists some compact set*

$\mathcal{A}_\delta \subset \Theta$ *and time* T_δ *so that, for all* S *sufficiently large,*

$$\sup_{x \in \mathcal{A}_\delta} \mathbb{P}[X_s \notin \mathcal{A}_\delta \, \forall s \in (S, S + T_\delta) | X_S \in \mathcal{A}_\delta] \leq \frac{\delta}{4}.$$

Proof. Combining Lemmas 3.3.15 and 3.3.14, there exists some number $\mathcal{X} > 0$ and almost surely finite random time $\tau_{\mathcal{X}}$ so that X_t satisfies a drift condition for all $t > \tau_{\mathcal{X}}$ and X_t outside of the compact set $\mathcal{B}_{\mathcal{X}}(0)$. The existence of a recurrent compact set then follows immediately from Lemma 4 of [95].
The second part of the result follows from noting that the exponential bounds on the return time in Lemma 4 of [95] depend on the starting point $x \in \mathcal{A}_\delta$ only through $V_\epsilon(x)$, which is uniformly bounded in $\mathcal{A}_\delta$ by a constant, and the proof is finished. □

3.3.5 Proof of ergodicity with Gaussian envelopes

Let us now assume that Θ need not be compact. This section proceeds as in Section 3.3.3, with the results of Section 3.3.4 filling in the gaps. First, note that Lemma 3.3.11 holds as stated, with a slightly different proof:

Proof of Lemma 3.3.11 for non-compact Θ. For any compact set $\mathcal{A}_n$, the second claim follows from item 4 of Assumptions 3.3.1. The first claim follows immediately from Lemma 3.3.16. □

Lemma 3.3.10 holds in the Gaussian envelope case exactly as stated, with the same proof. Lemma 3.3.9 now follows for the Gaussian envelope case exactly as stated. We are finally ready to prove Theorem 3.3.4:

Proof of Theorem 3.3.4. Fix $\epsilon > 0$. We will show that the conditions of Corollary 3.3.7. Inequality (3.7) follows from Theorem 5 of [95], Assumptions 3.3.1 and Theorem 3.2 of [94]. Inequality (3.8) follows for t sufficiently large from Markov's inequality, Lemma 3.3.14 and Lemma 3.3.15. Inequality (3.9) follows for S sufficiently large by Lemma 3.3.16. Finally, inequality (3.10) follows from Lemma 3.3.9. □

Example 3.3.17 (Decay Rate for β)**.** *We note that if* β_t *decays too quickly, our sampler may not converge. Consider the proposal distribution* L *that draws i.i.d. uniform samples from* $[0,1]^d$ *and let* $\lambda(\cdot)$ *denote the Lebesgue measure. Consider a target distribution of the form* $p(\theta|\mathbf{d}) \propto \mathbf{1}_{\theta \in G}$ *for set* G *with* $0 < \lambda(G) < 1$ *in Lebesgue measure. If* $\sum_t \beta_t < \infty$*, then by Borel-Cantelli, the probability* $p = p\left(\{\beta_t\}_{t \in \mathbb{N}}\right)$ *that no points are added to* $\mathcal{S}$ *except during the initial choice of reference points or failed cross-validation checks is strictly greater than 0. With probability* $\lambda(G)^k > 0$*, the first* k *reference points are all in* G*. But if both these events happen, all cross-validation checks are passed for any* $\gamma > 0$*, and so the walk never converges; it samples from the measure* λ *forever.*

As pointed out in Remark 3.3.12, we have a converse to this example in the case that Θ *is compact and our proposal distribution is bounded from below on* Θ*. In that situation, we have ergodicity whenever* $\sum_t \beta_t$ *diverges.*

Example 3.3.18 (Decay Rate for γ)**.** *We note that we have not used the assumption that $\gamma < \infty$ anywhere. As pointed out in Example 3.3.17, in a way this is justified—we can certainly find sequences $\{\beta_t\}_{t\in\mathbb{N}}$ and walks that are not ergodic for any sequence $\gamma_t > 0$ converging to zero at any rate.*

In the other direction, there exist examples for which having any reasonable fixed value of γ gives convergence, even with $\beta = 0$. We point out that this depends on the initially selected points; one could be unlucky and choose points with log-likelihoods that happen to lie exactly on some quadratic that does not match the true distribution. Consider a target density $\pi(x) \propto 1 + C\,\mathbf{1}_{x>\frac{1}{2}}$ on $[0,1]$ with independent proposal moves from the uniform measure on $[0,1]$. To simplify the discussion, we assume that our approximation of the density at each point is linear and based exactly on the three nearest sampled points. Denote by $\mathcal{S}_t$ the points which have been evaluated by time t, and let $\mathcal{S}_0 = \{\frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \frac{5}{8}, \frac{6}{8}, \frac{7}{8}\}$. Write $x_1, \ldots, x_{m(t)} = \mathcal{S}_t \cap [0, \frac{1}{2}]$ and $x_{m(t)+1}, \ldots, x_{n(t)} = \mathcal{S}_t \cap [\frac{1}{2}, 1]$. It is easy to check that

$$\|\mathcal{L}(X_{t+1}) - \pi\|_{\mathrm{TV}} \leq x_{m(t)+3} - x_{m(t)-2}. \tag{3.19}$$

It is also easy to see that with probability one, for any $\gamma < \frac{1}{2}$, there will always be a subinterval of $[x_{m(t)-2}, x_{m(t)+3}]$ with strictly positive measure for which a cross-validation check will fail. Combining this with inequality (3.19) implies that the algorithm will converge in this situation, even with $\beta = 0$.

3.4 Numerical experiments

Although the theoretical results in Section 3.3 establish the ergodicity and asymptotic exactness of our MCMC framework, it remains to demonstrate that it performs well in practice. This section provides three examples in which local surrogates produce accurate posterior samples using dramatically fewer evaluations of the forward model than standard MCMC. Additionally, these examples explore parameter tuning issues and the performance of several algorithmic variations.

For each of these examples, we compare the number of evaluations of the forward model to the accuracy of samples from the chain. In the absence of analytical characterizations of the posterior, the error in the chain is estimated by comparing the posterior covariance estimated from a reference MCMC chain—composed of multiple long chains computed *without* any approximation—to posterior covariance estimates computed from chains of Algorithm 3.1. The number of forward model evaluations is a problem-independent proxy for the overall running time of the algorithm, and should be representative of the algorithm's scaling for sufficiently expensive models. Although we presented Algorithm 3.1 in the context of non-adaptive Metropolis-Hastings proposals, here we substitute the delayed-rejection adaptive Metropolis approach of [50], which often performs well without laborious tuning of the proposal distribution. Note that the error indicator for the acceptance probability still reflects the form used in simple MH proposals.

The first example infers six parameters of an ODE model of a genetic circuit, using real experimental data, and suggests how to select values and decay rates for β and γ. The second example inverts for the parameters of the FitzHugh-Nagumo ODE and the third infers the diffusivity field in an elliptic PDE. These latter examples are used to investigate the performance of different types of local approximations (linear, quadratic, Gaussian process) and the difference between direct and indirect approximation.

3.4.1 Genetic toggle switch

This example infers the parameters of a genetic "toggle switch" synthesized in *E. coli* plasmids by Gardner *et al.* [39], and previously used in an inference problem by [79]. This genetic circuit has a bistable response to the concentration of an input chemical, [IPTG]. Figure 3-2 illustrates these high and low responses, where the vertical axis corresponds to the expression level of a particular gene. [39] proposed the following differential-algebraic model for the switch:

$$\begin{aligned} \frac{du}{dt} &= \frac{\alpha_1}{1+v^\beta} - u, \\ \frac{dv}{dt} &= \frac{\alpha_2}{1+w^\gamma} - v, \\ w &= \frac{u}{(1+[\text{IPTG}]/K)^\eta}. \end{aligned} \tag{3.20}$$

The model contains six unknown parameters $Z_\theta = \{\alpha_1, \alpha_2, \beta, \gamma, K, \eta\} \in \mathbb{R}^6$, while the data correspond to observations of the steady-state values $v(t = \infty)$ for six different input concentrations of [IPTG], averaged over several trials each. As in [79], the parameters are centered and scaled around their nominal values so that they can be endowed with uniform priors over the hypercube $[-1, 1]^6$. The measurement errors are independent and Gaussian, with zero mean and variances that differ between the "low" and "high" states of the switch. Further details on the inference problem are given in Appendix B. Figure 3-3 shows marginal posterior densities of the parameters Z_θ in normalized coordinates. These results broadly agree with [79] and indicate that some directions are highly informed by the data while others are largely defined by the prior, with strong correlations among certain parameters.

Now we examine the efficiency and accuracy of the local approximation algorithm in exploring this target posterior. The baseline configuration to which we compare Algorithm 3.1 comprises 30 chains, each run for 10^5 MCMC steps using the true forward model (i.e., with no approximation). In all of the numerical experiments below, we discard the first 10% of a chain as burn-in when computing its covariance. The reference chain used to produce the "truth" covariance is the union of the 30 baseline chains, with the burn-in portion of each removed. The chains are all initialized at the same point in the high posterior density region.

To use the local approximation framework, we must select values for the refinement

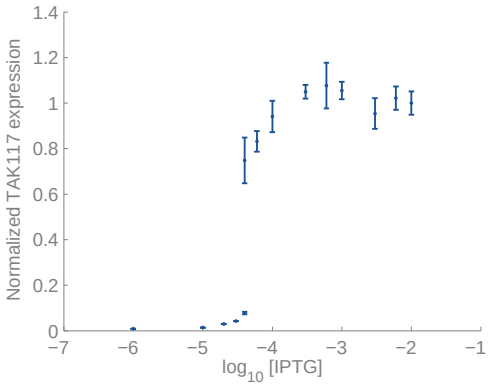


Figure 3-2: Response of the pTAK117 genetic toggle switch to the input concentration of IPTG [39]. The plot shows the mean and standard deviation of the experimentally-observed gene expression levels over a range of input concentrations. Expression levels are normalized by the mean response at the largest IPTG concentration.

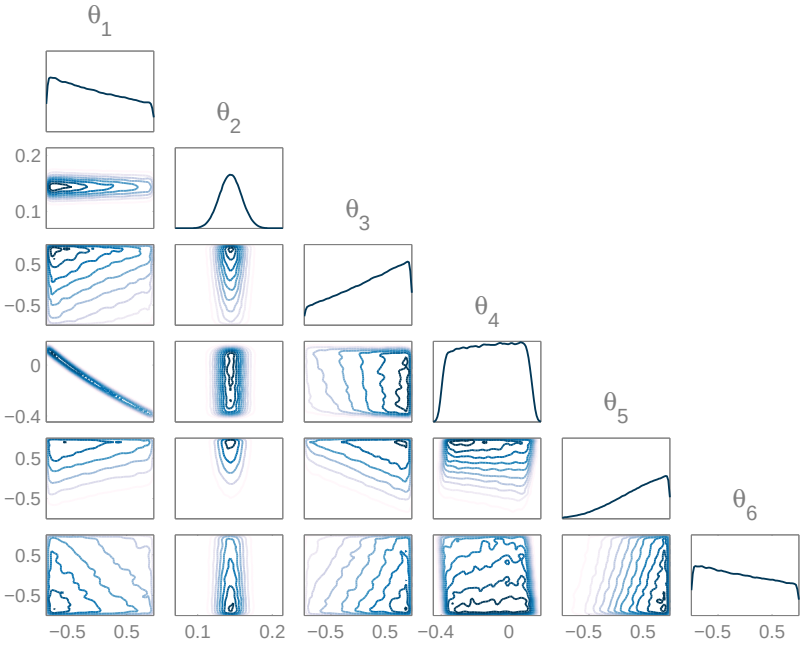


Figure 3-3: One- and two-dimensional posterior marginals of the six parameters in the genetic toggle switch.

parameters β and γ. While the theory does not rely on a particular setting for γ, it suggests that β can decay. Accordingly, we set $\beta_n = c_1 n^{-c_2}$ and $\gamma_n = c_3 n^{-c_4}$, where $c_1, c_3 > 0$ and $0 < c_2, c_4 < 1$, and n is the MCMC step index. Intuitively, the chain will be inexpensive when β is small and when γ is large, and hence if β_n decays relatively quickly and if γ_n decays relatively slowly.

First, we attempt to understand the scale of the refinement parameters by putting the decay rates to zero and setting only one of β_n or γ_n to be nonzero, choosing from $\beta_n \in \{10^{-3}, 10^{-2}, 10^{-1}\}$ and $\gamma_n \in \{10^{-2}, 10^{-1}, 0.5\}$. With these settings, we run Algorithm 3.1 using local quadratic approximations of the forward model outputs (i.e., indirect approximation). Thirty independent chains are run for each parameter setting, with each chain containing 10^5 MCMC steps—identical to the baseline chains. From each chain, we discard 10^4 burn-in samples and compute an estimate of the posterior covariance. We compare this estimate to the covariance of the reference chain by evaluating the Frobenius norm of the difference, then dividing by the Frobenius norm of the reference to provide a relative error measure. The results are summarized in Figure 3-4.

The distribution of errors obtained with the baseline chains, shown in red, reflects both the finite accuracy of the reference chain and the variance resulting from finite baseline chain lengths. Looking at the results produced by Algorithm 3.1, we see that smaller values of β and larger values of γ result in fewer evaluations of the forward model; these values trigger fewer random refinements and allow larger errors in the acceptance probability, respectively. When β-refinement is set to occur at a very low rate, the resulting chain is of poor quality. But higher values of β produce errors that are indistinguishable from those of the baseline chains. Similarly, tighter thresholds on γ produce high-accuracy posterior estimates, yet they use an order of magnitude fewer model evaluations than the baseline cases. Interestingly, even though $\gamma_n = 0.5$ corresponds to a relatively loose constraint on the acceptance probability, it still performs reasonably well. Based on these experiments, $\beta_n = 10^{-2}$ and $\gamma_n = 10^{-1}$ appear to provide a reasonable balance between stable reproduction of the posterior and computational effort. Note that the true model is actually run less than once per MCMC step, even in the baseline results, because proposals falling outside the support of the prior are rejected without actually running the forward model.

Next, we explore possible decay rates for the refinement coefficients. We fix the constants in the expressions for β_n and γ_n to the values determined from the previous numerical experiments: $c_1 = 0.01$ and $c_3 = 0.1$. Then we run experiments for different values of the exponents, using $c_2, c_4 \in \{0.1, 0.5, 0.9\}$. The performance of the resulting chains is shown in Figure 3-5. When γ_n decays relatively quickly (e.g., for $c_4 = 0.5$ or 0.9), the computational costs are much higher and do not improve the results; hence we choose $c_4 = 0.1$. For these parameter settings, the overwhelming majority of refinements are initiated by γ-refinement and the decay rate of β_n appears to have little effect. Thus we chose the relatively faster decay rate $c_2 = 0.9$. These choices yield complete expressions for the refinement parameters: $\beta_n = 0.01 n^{-0.9}$ and $\gamma_n = 0.1 n^{-0.1}$. These decaying rules produce more expensive chains than some of the constant values

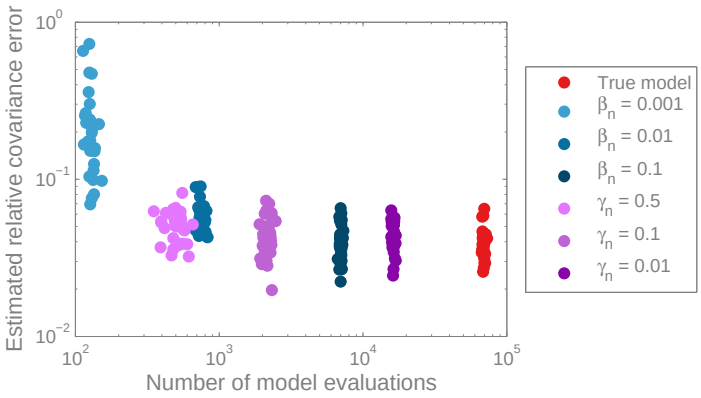


Figure 3-4: Approximate relative covariance errors in MCMC chains for the genetic toggle switch posterior, using constant values of the refinement parameters β_n and γ_n and local quadratic approximations. The parameter given in the legend has the specified value, while the other parameter is zero.

employed earlier because the cross-validation criterion tightens as the chain lengthens, but this approximation strategy should be more robust for difficult problems than one based on a constant threshold. The following numerical examples use these refinement parameters without further tuning. As we will demonstrate, they work quite well in practice.

3.4.2 FitzHugh-Nagumo ODE

This numerical example explores the performance of local linear, quadratic, and Gaussian process approximations. We perform inference for the parameters of the FitzHugh-Nagumo model, a two-dimensional ODE system with three parameters, used to describe the activation and deactivation dynamics of a neuron. The model has been used as a test case for inference by [90, 45]. The governing equations are:

$$\begin{aligned} \frac{dV}{dt} &= V - \frac{V^3}{3} + R, \\ \frac{dR}{dt} &= -\frac{1}{c}(V - a + bR). \end{aligned}$$

In contrast with the previous example, the data now are transient: both states are observed at 100 times equally spaced from $t = 0$ to $t = 20$. The observational error in the likelihood function is a zero-mean Gaussian, where the standard deviation of each component is equal to one-tenth the standard deviation of the 100 observations of that component during the reference run. The data are synthetic, produced by running the model at the nominal parameter values $a = 0.3$, $b = 0.3$, and $c = 2$ and perturbing with a realization of the observation error. The prior on $\theta = (a, b, c)$ is uniform over the box $[0, 10]^3$.

As there are 200 observations, it would be relatively slow to approximate each

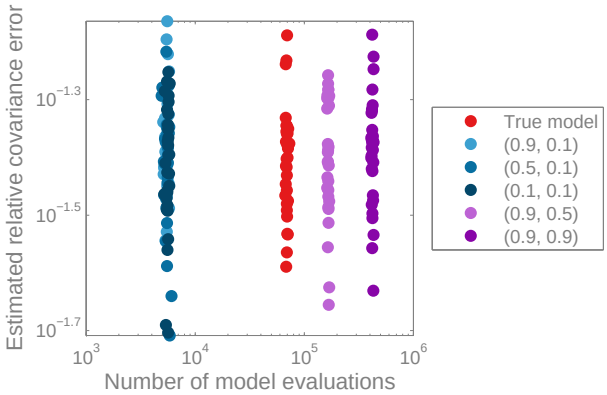


Figure 3-5: Approximate relative covariance errors in MCMC chains for the genetic toggle switch posterior, using decaying refinement parameters and local quadratic approximations. The parameters are $\beta_n = 0.01n^{-c_2}$ and $\gamma_n = 0.1n^{-c_4}$, where the legend specifies (c_2, c_4).

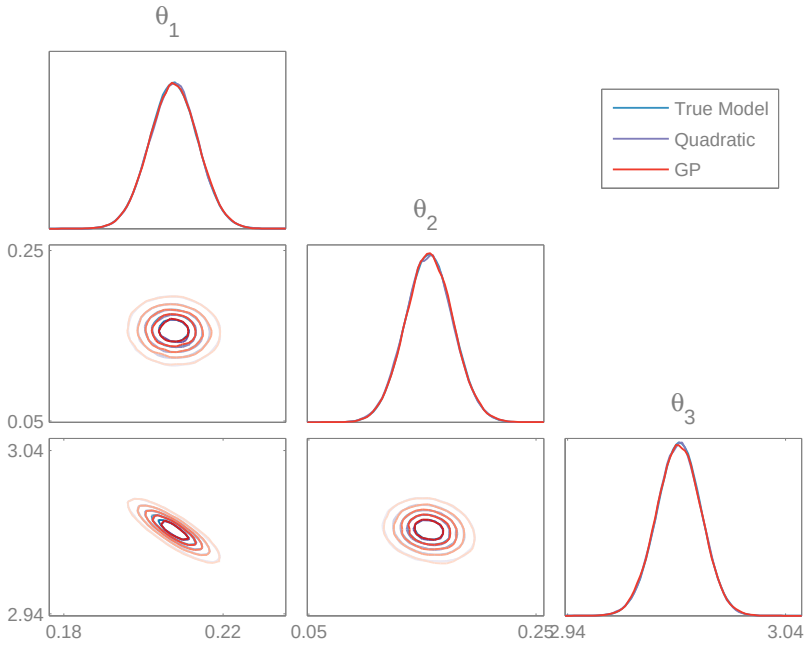


Figure 3-6: One- and two-dimensional posterior marginals of the three parameters in the FitzHugh-Nagumo ODE. We superimpose probability density contours computed via three methods. The "true model" (in blue) refers to the reference chain comprising 30 subchains, while the purple and red lines are obtained from a single run of Algorithm 3.1, using direct approximation via local quadratic or Gaussian process models.

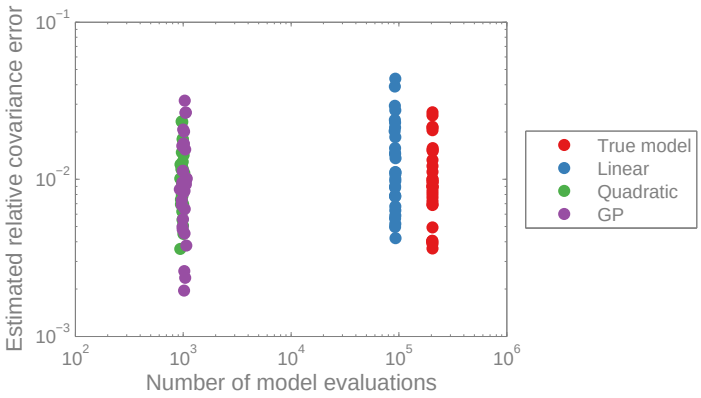


Figure 3-7: Approximate relative covariance errors in the MCMC chains for the FitzHugh-Nagumo parameter posterior, using three different types of local approximations.

corresponding model output individually; instead, the approximation operators are applied to the log-likelihood, i.e., using the direct approach. We run Algorithm 3.1 using the decay schedule for β_n and γ_n found in the previous example. Thirty independent chains are run for each approximation method—local linear, quadratic, and Gaussian process regression—with each chain containing 10^5 samples. A reference chain employing the true model is produced in the same manner described in the previous section (Section 3.4.1). The posterior distribution is illustrated in Figure 3-6. Reconstructions from the reference chain and from single realizations of the approximate chains (produced with quadratic and Gaussian process approximations) are visually indistinguishable.

The computational efficiency and accuracy of the various samplers are summarized in Figure 3-7. For chains of 10^5 samples each, we evaluate the relative error in the posterior covariance as a function of the number of forward model evaluations, exactly as in the previous example. In this case, the linear approximation slightly reduces the number of model evaluations over the non-approximated chains. The quadratic and Gaussian process approximations, on the other hand, reduce the required number of model evaluations by over two orders of magnitude, with no apparent loss of accuracy.

3.4.3 Elliptic PDE inverse problem

We now turn to a canonical inverse problem, involving inference of the diffusion coefficient in an elliptic PDE [30]. As an application of Algorithm 3.1, the goal of this example is to compare direct and indirect approximations, using local linear, quadratic, and Gaussian process approximations, on a non-compact parameter domain. The forward model is given by the solution of an elliptic PDE in two spatial dimensions

$$\nabla_{\mathbf{s}} \cdot (k(\mathbf{s}, \theta) \nabla_{\mathbf{s}} u(\mathbf{s}, \theta)) = 0, \tag{3.21}$$

where $\mathbf{s} = (s_1, s_2) \in [0,1]^2$ is the spatial coordinate. The boundary conditions are

$$\begin{aligned} u(\mathbf{s},\theta)|_{s_2=0} &= s_1, \\ u(\mathbf{s},\theta)|_{s_2=1} &= 1 - s_1, \\ \left.\frac{\partial u(\mathbf{s},\theta)}{\partial s_1}\right|_{s_1=0} &= 0, \\ \left.\frac{\partial u(\mathbf{s},\theta)}{\partial s_1}\right|_{s_1=1} &= 0. \end{aligned}$$

This PDE serves as a simple model of steady-state flow in aquifers and other subsurface systems; k can represent the permeability of a porous medium while u represents hydraulic head. Our numerical solution of (3.21) uses the standard continuous Galerkin finite element method with bilinear basis functions on a uniform 30-by-30 quadrilateral mesh.

The log-diffusivity field $\log k(\mathbf{s})$ is endowed with a Gaussian process prior, with mean zero and an isotropic squared-exponential covariance kernel,

$$C(\mathbf{s}_1, \mathbf{s}_2) = \sigma^2 \exp\left(-\frac{\|\mathbf{s}_1 - \mathbf{s}_2\|^2}{2\ell^2}\right),$$

for which we choose variance $\sigma^2 = 1$ and a length scale $\ell = 0.2$. This prior allows the field to be easily parameterized with a Karhunen-Loève (K-L) expansion [1]:

$$k(\mathbf{s},\theta) \approx \exp\left(\sum_{i=1}^{d} \theta_i \sqrt{\lambda_i} k_i(\mathbf{s})\right),$$

where λ_i and $k_i(\mathbf{s})$ are the eigenvalues and eigenfunctions, respectively, of the integral operator on $[0,1]^2$ defined by the kernel C, and the parameters θ_i are endowed with independent standard normal priors, $\theta_i \sim \mathcal{N}(0,1)$. These parameters then become the targets of inference. In particular, we truncate the Karhunen-Loève expansion at $d = 6$ modes and condition the correponding mode weights $(\theta_1, \ldots, \theta_6)$ on data. Data arise from observations of the solution field on a uniform 11×11 grid covering the unit square. The observational errors are taken to be additive and Gaussian:

$$d_j = u(\mathbf{s}_j, \theta) + \epsilon_j,$$

with $\epsilon_j \sim \mathcal{N}(0, 0.1^2)$.

Because the data in this problem are relatively informative, the posterior shifts and concentrates significantly with respect to the standard normal prior, as shown in Figure 3-8. We also emphasize that even though the PDE is linear, the forward model—i.e., the map from k to u—is nonlinear and hence the posterior is not Gaussian. We also note that, while the design of effective posterior sampling strategies for inverse problems is an enormous and important endeavor [24], the K-L truncation ren-

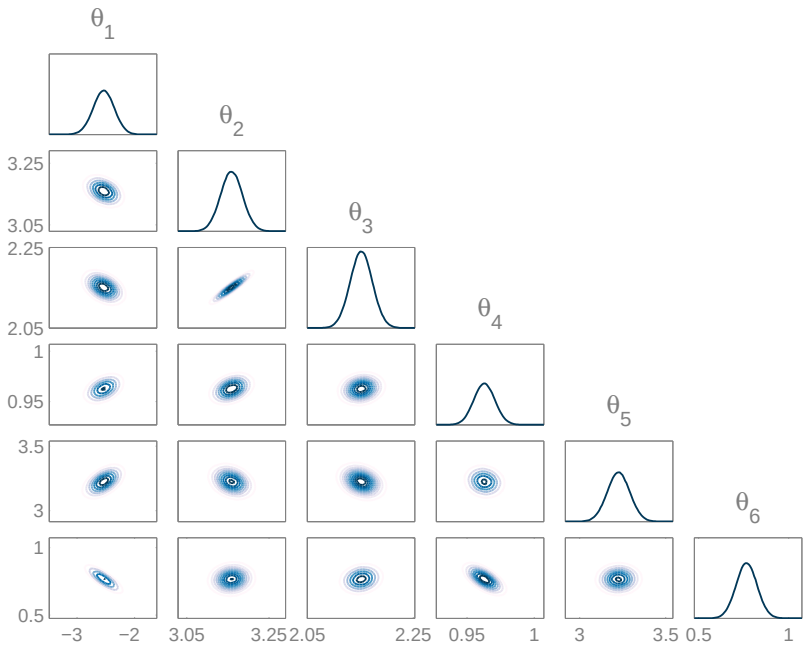


Figure 3-8: One- and two- dimensional posterior marginals of the parameters in the elliptic PDE inverse problem.

ders this problem relatively low-dimensional and the simple delayed rejection adaptive Metropolis sampler used to obtain our results mixes well.

Now we evaluate the performance of various local approximation schemes, using the framework of Algorithm 3.1. We fix the decay schedule of the parameters β_n and γ_n as in the previous examples. Figure 3-9 summarizes the results, which are obtained from the same kind of comparison employed earlier: 30 independent chains of 10^5 samples each, for each approximation scheme and for no approximation, all compared to a reference chain. Within these results, we also contrast the performance of direct approximations with the performance of indirect approximations.

The indirect quadratic approximation shows the best performance, using only 600 model evaluations for 10^5 MCMC samples, instead of 2.4×10^5 evaluations for the true model. This is a 400-fold reduction in computational effort, with no apparent loss in accuracy. For both the linear and quadratic local models, indirect approximation significantly outperforms direct approximation. Indirect Gaussian process regression is not immediately feasible because of the large number of observations, and direct GP approximation produces accurate results for cost roughly equivalent to that of running the true model directly. Linear direct approximation gives inaccurate results even though it uses a large number of evaluations of the true forward model. We take this as evidence that local linear approximations of the log-likelihood are untenably weak in practice, but not that it contradicts the argument in Remark 3.3.6 that this configuration converges in the limit under some assumptions.

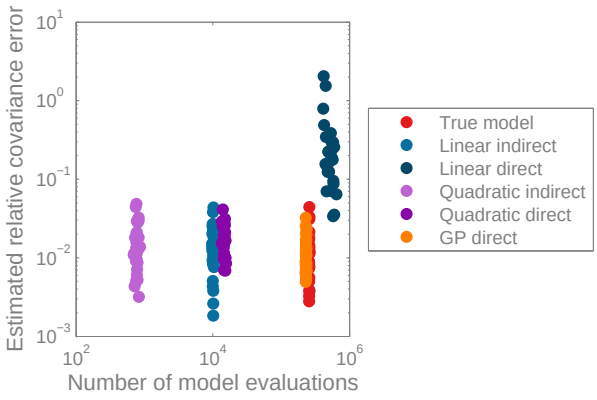


Figure 3-9: Approximate relative covariance errors in the MCMC chains for elliptic inverse problem, using several different local approximation strategies.

3.5 Discussion

We have proposed a new class of MCMC algorithms that construct local surrogates to reduce the cost of Bayesian inference in problems with computationally expensive forward models. These algorithms introduce local approximations of the forward model or log-likelihood into the Metropolis-Hastings kernel and refine these approximations incrementally and infinitely. The resulting Markov chain thus employs a sequence of transition kernels, but asymptotically samples from the exact posterior distribution. We describe variations of the algorithm that employ either local polynomial or local Gaussian process regression, thus spanning two widely-used classes of surrogate models. Numerical experiments demonstrate significant reductions in the number of forward model evaluations used for posterior sampling in ODE and PDE model problems.

Our theoretical and numerical results underscore the notion that local regularity in the forward model or log-likelihood should be harnessed for computational efficiency, and that the number of model evaluations needed to approach exact sampling from the posterior can be much smaller than the number of MCMC samples themselves. Although our convergence arguments can easily be made quantitative, we believe that doing so in a straightforward manner does not capture the largest strength of our algorithm. Looking at the process described in Example 3.3.18, we see that a well-chosen start results in a bias bound that decays almost exponentially in the number of likelihood evaluations and that the number of likelihood evaluations will grow approximately logarithmically in the running time of the process. Our general bounds, however, show only that the bias decays at least as quickly as one over the number of likelihood evaluations. There is a discrepancy here primarily because our general bounds do not take advantage of the fact that the cross-validation approach allows us to evaluate the likelihood primarily in regions where refinement is important. Taking advantage of this fact is not hard to do for particular examples; a more general theory would need to avoid the problems that arise in Example 3.3.17 and similar

constructions.

Chapter 4

Variations on local approximation-based samplers: derivative information and parallel chains

In this chapter, we explore the flexibility of local surrogate-based MCMC by developing three extensions that correspond to important recent trends within the computational science and MCMC literature. First, we loosen the black-box assumption by supplying derivatives of the forward model. Derivatives provide a rich source of information for constructing approximations, and if available cheaply, can reduce the cost of surrogate models. Second, we demonstrate the compatibility of our framework with the important class of modern MCMC algorithms that use derivatives to construct better proposal distributions. As a representative example, we adapt the simplified metric Metropolis adjusted Langevin algorithm (MALA) to use local approximations, producing an algorithm broadly similar to the one for Metropolis-Hastings proposals, but which differs in some important details. Finally, we explore how local approximations can be integrated into a parallel computational environment by sharing the set of samples between parallel chains.

While these extensions are useful and practical in their own right, they serve as case studies that further demonstrate the modularity and stability of the local approximation framework. For example, we show that the MALA sampler can be used successfully in combination with any of the approximation types, whether derivative-based or not. Interestingly, no radical changes to the framework for error indicators or refinement are needed; the obvious translation of cross validation and the acceptance probability based error indicators into these settings is appropriate, and the refinement constants, β_m and γ_m, do not need to be re-tuned. This robustness strengthens the argument that the experimental design process is targeted to natural steps of MCMC.

As the new variations described in this chapter are largely modular, and may be combined with the options from the previous chapter, there is such a large configu-

ration space of algorithms that we cannot exhaustively compare all the possibilities. Furthermore, such an effort would touch on highly problem-dependent, open questions, such as whether DRAM or MALA is more efficient. Instead, we revisit the examples from the last chapter and attempt to provide a small set of illustrative examples to establish that the algorithms are correct and perform as expected.

4.1 Local quadratic models using Jacobians

We have placed our work thus far in a non-intrusive setting because it is a simple and practical model. Often, scientifically interesting computational models are only available as black-box models, and reimplementing them to construct an intrusive approximation is simply not feasible. However, an interesting relaxation to this assumption is to assume that the gradient of the model is available. Although not always available, modern adjoint techniques for PDE solvers allows for the computation of a gradient of the model at a cost approximately equal to an additional linearized forward solve. As the gradient provides d elements for a model with d-dimensional input, the adjoint provides excellent value for the computational effort. Recently, effort has been directed to implementing adjoints for scientific models and using this additional information in analysis algorithms, for an example in an inference context, see [78]. In some cases, automatic differentiation tools can generate adjoints corresponding to existing implementations with relatively little additional work. For example, this strategy is used by the MIT general circulation model (MITgcm) [77], which we use for the Pine Island problem in Chapter 5.

Using derivative information in function approximation is a common idea, for example, classical Hermite polynomial interpolation matches the first m derivatives, and co-kriging fits a Gaussian process to the function evaluations and corresponding gradients [35]. As a representative of this type of approximation, we explore the specific case of quadratic regressors using Jacobians of the model. These approximations are constructed using a simple extension of the quadratic approximations of Section 3.2.2; much of the structure least squares problem is unchanged, except that additional rows are appended to supply the derivative information. Jointly fitting approximation for multiple outputs may allow the use of less derivative information than the entire Jacobian, but we do not explore this additional complexity.

Jacobians are a rich source of information for approximation, but are not always affordable. An adjoint model makes a single gradient inexpensive, but the cost of forming the entire Jacobian using an adjoint technique scales linearly with the output dimension of the model. Considering these costs, we use these approximations in three contexts. First, in the FitzHugh-Nagumo model, the package performing the time integration computes the entire Jacobian of the forward model for reasonable additional cost, hence an indirect approximation is appropriate. Second, some MCMC methods, such as the simplified manifold MALA [45], use the Jacobian of the forward model, as described in the next section. In such cases, the algorithm already needs to pay for the expensive construction of Jacobians of the forward model, and applying

indirect approximations may reduce the number required. Third, direct approximation of the log-likelihood requires a Jacobian of the log-likelihood, which is computable with only a single adjoint computation on the forward model, making it the most generally applicable strategy.

4.1.1 Fitting the gradient-based quadratic

The Jacobian-based quadratic model is fit using the operator $\mathcal{D}^{\sim j}_{\mathcal{B}(\theta,R)}$, which we now describe. As before, assume that the local model has the form

$$\tilde{f}_j(\hat{\theta}) = c_j + \mathbf{l}_j^\top \hat{\theta} + \frac{1}{2}\hat{\theta}^\top H_j \hat{\theta},$$

which is stated in coordinates shifted and scaled so that the point of interest is centered and the samples are drawn from a ball of radius one. Thus, the shifted samples are $\hat{\theta}_i = (\theta_i - \theta)/R$. The regression model is constructed from samples θ_i, evaluations $\mathbf{y}_i = \mathbf{f}(\theta_i)$, and Jacobians $J(\theta_i)$, where

$$J(\theta_i) = \begin{pmatrix} \nabla f_1(\theta_i) \\ \vdots \\ \nabla f_d(\theta_i) \end{pmatrix}.$$

By application of the chain rule, $\hat{J}(\hat{\theta}_i) = J(\hat{\theta}_i)/R$. The quadratic approximation has gradient

$$\nabla \tilde{f}_j(\hat{\theta}_i) = \mathbf{l}_j + H_j \hat{\theta}_i.$$

The weighted least squares problem is augmented with additional rows that attempt to match this gradient to the observed gradients, written compactly as:

$$W' \begin{pmatrix} \Phi \\ \Psi_1 \\ \vdots \\ \Psi_N \end{pmatrix} Z = W' \begin{pmatrix} Y \\ \hat{J}(\theta_1) \\ \vdots \\ \hat{J}(\theta_N) \end{pmatrix}$$

where Φ, Y, and Z are defined as in Appendix A. Just as Φ is the Vandermonde-like matrix for $\tilde{f}(\theta_i)$, Ψ_i is the Vandermonde-like matrix of $\nabla \tilde{f}(\theta_i)$. Recall that

$$Z_j^\top = \begin{pmatrix} c_j & (\mathbf{l}_j)_1 & \cdots & (\mathbf{l}_j)_d & (H_j)_{1,1} & \cdots & (H_j)_{d,d} & (H_j)_{1,2} & \cdots & (H_j)_{d-1,d} \end{pmatrix}.$$

Hence,

$$\Psi_i = \begin{pmatrix} \mathbf{0}_{d\times 1} & I_{d\times d} & A(\hat{\theta}_i) & B(\hat{\theta}_i) \end{pmatrix},$$

where the zero vector corresponds to the constant term, I is the identity matrix corresponding to the linear coefficients, $A(\theta_i)$ is the diagonal matrix

$$A(\hat{\theta}_i) = \begin{pmatrix} \hat{\theta}_{i,1} & 0 & \dots & 0 \\ 0 & \hat{\theta}_{i,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{\theta}_{i,d} \end{pmatrix},$$

corresponding to the squared terms, and $B(\hat{\theta}_i)$ corresponds to the mixed terms. Each row of $B(\hat{\theta}_i)$ is the partial derivative of the mixed coefficient terms with respect to an element of $\hat{\theta}_i$, hence is either zero or an element of $\hat{\theta}_i$:

$$B(\hat{\theta}_i) = \begin{pmatrix} \frac{\partial}{\partial \hat{\theta}_{i,1}} \left(\hat{\theta}_{i,1}\hat{\theta}_{i,2} & \cdots & \hat{\theta}_{i,d-1}\hat{\theta}_{i,d} \right) \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial \hat{\theta}_{i,d}} \left(\hat{\theta}_{i,1}\hat{\theta}_{i,2} & \cdots & \hat{\theta}_{i,d-1}\hat{\theta}_{i,d} \right) \end{pmatrix} = \begin{pmatrix} \hat{\theta}_{i,2} & \cdots & \cdots & 0 \\ \hat{\theta}_{i,1} & \ddots & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & \hat{\theta}_{i,d-1} \end{pmatrix}.$$

The samples are weighted by diagonal matrix W', where the weight for sample i is w_i, as computed by our variant of the tri-cube weight function. Now the ith point appears in several rows, hence the weights appear repeatedly in the complete weight matrix. The diagonal values of the weight matrix are written:

$$\text{diag}(W') = \begin{pmatrix} w_1 & \cdots & w_N & w_1\mathbf{1}_d^\top & \cdots & w_N\mathbf{1}_d^\top \end{pmatrix},$$

which shares the same block structure as the least squares problem.

For this model, because of the extra information provided by the derivatives, the number of points necessary to produce an interpolating model is linear in the input dimension, specifically $N_{\text{def}} = d + 1$, and the total number of points to use in a regression is unchanged, $N = \sqrt{d}N_{\text{def}}$.

This new approximation operator, $\mathcal{D}^{\sim j}_{\mathcal{B}(\theta,R)}$, may be used in Algorithm 3.1 almost without change. Since $\mathcal{S}$ contains the outputs of the model for the evaluated points, it is enriched to hold the Jacobians as well. The procedures for refining the model are unchanged, except that when a new point is added to $\mathcal{S}$, the Jacobian is also computed and stored. The cross validation procedure still omits a single sample, which requires removing the rows for the evaluation and Jacobian at that point; this may be accomplished by setting $w_j = 0$.

4.1.2 Experimental results

We explore the performance of Jacobian-based quadratic approximations by repeating the three experiments introduced in Chapter 3 for DRAM samplers. Since we are interested in providing derivative information in a settings where the computational effort needed to compute the derivatives is comparable to the cost of a run of the model,

as in the case of adjoint methods, we simply expand the cost of one model evaluation to include computing both the output and the derivative. Otherwise, the experimental configuration remains unchanged from Chapter 3. Intuitively, we expect that allowing the Jacobian-based approximation access to this extra information should reduce the number of evaluations.

The genetics model is simple enough that we simply derive the Jacobian of the forward model using symbolic differentiation and compute it directly. This is not typically affordable, but it is still a useful test. The results are shown in Figure 4-1, where the Jacobian-based quadratic provides significant additional savings as compared to the equivalent quadratic approximation.

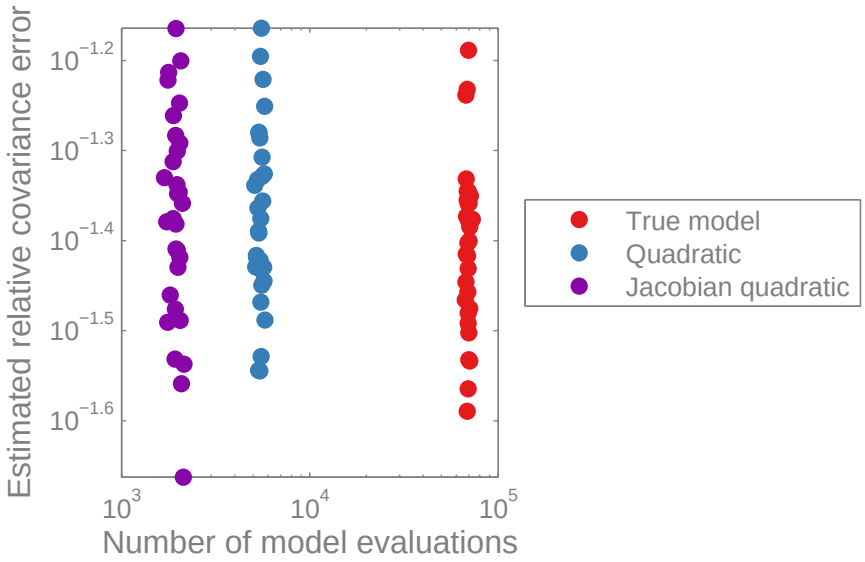


Figure 4-1: Approximate relative covariance errors of the genetic toggle switch example with DRAM based sampler, comparing indirect approximations using quadratic and Jacobian-based quadratic approximations.

In the second example, the package used for solving the FitzHugh-Nagumo ODE, Sundials [57], can be configured to inexpensively compute Jacobians of forward model given analytically specified derivatives of the model, which can easily be formed in this case using symbolic differentiation. Therefore, it is feasible to compute indirect approximations using Jacobian information, and Figure 4-2 demonstrates improved performance.

Finally, the elliptic PDE is self-adjoint, making it simple to provide an adjoint of this model. However, this only allows us to build direct approximations using Jacobian-based quadratic approximations. Figure 4-3 shows improved results when compared to the closest option, direct, quadratic approximation. This result also indicates the effectiveness of indirect approximation, though, as this improved result does not surpass indirect, quadratic approximation without derivatives, as in the previous chapter.

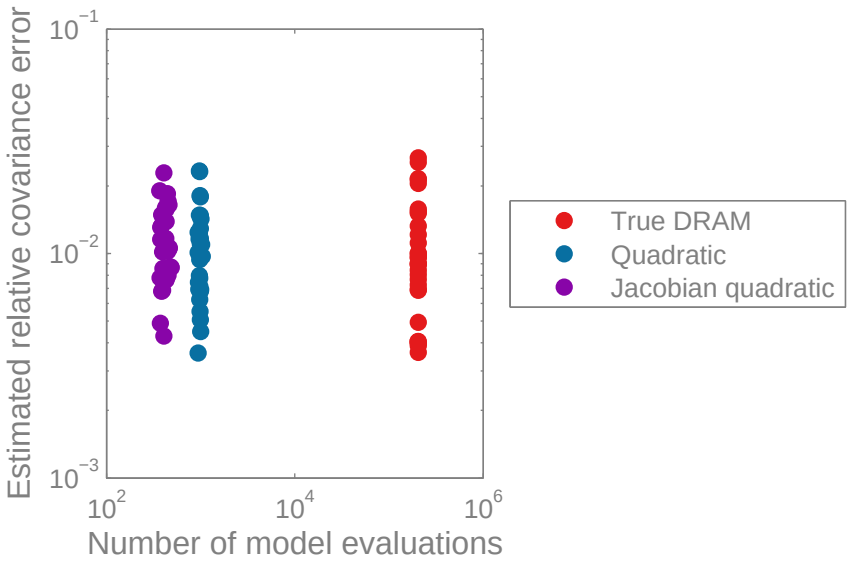


Figure 4-2: Approximate relative covariance errors of the Fitz example with DRAM based sampler, comparing indirect approximations using quadratic and Jacobian-based quadratic approximations.

4.2 An approximate Metropolis Adjusted Langevin sampler

Thus far, we have focused on approximate MCMC algorithms based on the DRAM algorithm, neglecting the important class of MCMC kernels that use derivative information. The two most popular such methods are the Metropolis-adjusted Langevin algorithm (MALA) [45] and Hamiltonian Monte Carlo (HMC) [81, 58]. The goal of these methods is to use derivatives of the posterior density to produce an improved proposal, thus reducing the mixing time of the chain and improving the efficiency of the MCMC algorithm. There are two situations in which local approximations can be usefully combined with derivative-based MCMC. First, when derivatives of the model are not available, we can make these methods feasible by supplying derivatives from local approximations. Second, if derivatives are available, then we may form local approximations using that additional information, as in the Section 4.1, to reduce the number of model and derivative evaluations.

Hamiltonian Monte Carlo typically proposes long-range moves and avoids random-walk behavior of the chain, but does so by taking many sub-moves according to a simulated particle trajectory; this process is quite expensive and reducing this cost motivated Rasmussen's use of approximations [91]. MALA proposes a single step move based on Langevin flows, which biases the proposal towards regions of higher posterior density, and applies a Metropolis correction [92]. We choose to restrict our discussion to MALA for two reasons. First, it is closer in structure to DRAM, so while adapting it to use local approximations requires additional work, much of the

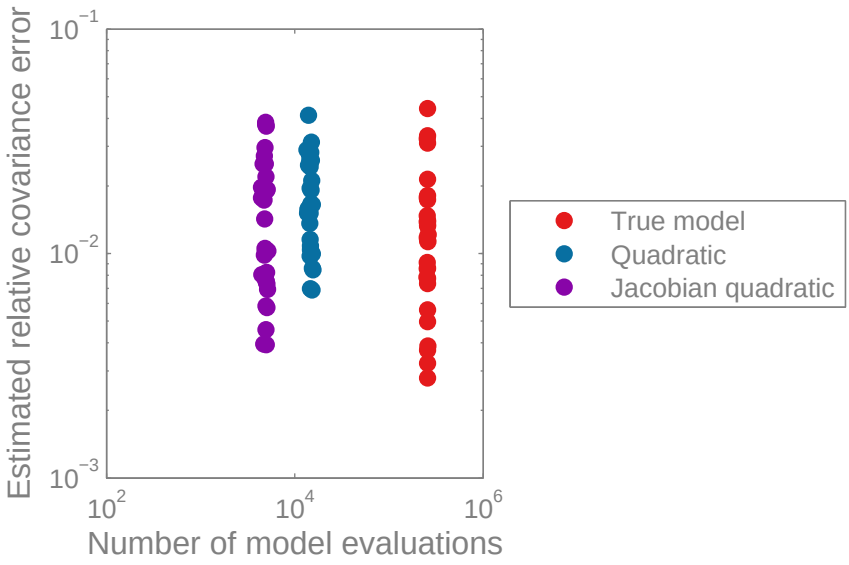


Figure 4-3: Approximate relative covariance errors of the elliptic PDE example with DRAM based sampler, comparing direct approximations using quadratic and Jacobian-based quadratic approximations.

approach is similar. Second, it remains an open question whether the extra effort expended by HMC during the sub-steps is cost-effective in practice [45]. We do not rigorously prove convergence of the sampler developed here, but expect that similar arguments to those in Chapter 3 can be made.

4.2.1 The Metropolis Adjusted Langevin algorithm

We begin by describing the standard Metropolis Adjusted Langevin algorithm (MALA) and a recent variant based on Riemannian manifolds [45]. Consider the Langevin diffusion with stationary distribution $p(\theta|\mathbf{d})$, defined by the stochastic differential equation (SDE),

$$d\theta(t) = \nabla_\theta(\mathcal{L}(\theta|\mathbf{d},\mathbf{f})p(\theta)) + d\mathbf{b}(t),$$

where $\mathbf{b}$ is a d-dimensional Brownian motion. Discretizing this SDE with a first-order Euler strategy suggests a proposal

$$q_t = X_{t-1} + \frac{\epsilon^2}{2}M(X)\nabla_\theta(\mathcal{L}(\theta|\mathbf{d},\mathbf{f})p(\theta)) + \epsilon\sqrt{M(X)}\mathbf{z},$$

where $\mathbf{z} \sim \mathcal{N}(0, I_{d\times d})$, for integration step size, ϵ, and position-dependent mass matrix, $M(X)$, which we may treat as preconditioner. The discretization implies that this proposal is not exact, but it may be corrected with a standard Metropolis step. Hence,

the proposal density is

$$q(x, y) = \mathcal{N}\left(y; x + \frac{\epsilon^2}{2}\nabla_\theta(\mathcal{L}(x|\mathbf{d}, \mathbf{f})p(x)), \epsilon^2 M(x)\right),$$

and the acceptance ratio is

$$\alpha(x, y) = \min\left(1, \frac{p(y|\mathbf{d})q(y, x)}{p(x|\mathbf{d})q(x, y)}\right).$$

We are relatively unconstrained in our choice of the preconditioner, for example, standard MALA corresponds to choosing the identity matrix, $M(\theta) = I$, or we might use domain knowledge to design a useful rescaling or rotation of the parameters. Recent work suggests a sophisticated mechanism for selecting a position-based matrix based on a Riemannian metric induced by the posterior distribution [45]. This method aims to use geometric arguments to extract rich structure of the inference problem, thereby producing high quality proposals. We consider the so-called simplified manifold MALA, which chooses to set the mass matrix to have the value of the following metric,

$$\begin{aligned} M(\theta) &= \left[-\mathbb{E}_{\mathbf{d}|\theta}\left(\frac{\partial^2}{\partial\theta^2}\log p(\theta, \mathbf{d})\right)\right]^{-1} \\ &= \left[-\mathbb{E}_{\mathbf{d}|\theta}\left(\frac{\partial^2}{\partial\theta^2}\left[\log p(\mathbf{d}|\theta, \mathbf{f}) + \log p(\theta)\right]\right)\right]^{-1}. \end{aligned}$$

which is decomposed into the expected Fisher information plus the negative Hessian of the log-prior. In general, computing the expected Fisher information is not trivial, but it is relatively simple for Gaussian likelihoods and priors, that is, when

$$\begin{aligned} p(\mathbf{d}|\theta, \mathbf{f}) &= \mathcal{N}(\mathbf{d}; \mathbf{f}(\theta), \Sigma_{\mathcal{L}}), \\ p(\theta) &= \mathcal{N}(\theta; \mu, \Sigma_p) \end{aligned}$$

for some covariance matrices $\Sigma_{\mathcal{L}} \in \mathbb{R}^{d\times d}$, $\Sigma_p \in \mathbb{R}^{n\times n}$ and mean vector $\mu \in \mathbb{R}^n$. In this case,

$$M(\theta) = \left[J(\theta)^\top \Sigma_{\mathcal{L}}^{-1} J(\theta) + \Sigma_p\right]^{-1}.$$

Girolami *et al.* [45] observe that choosing the preconditioner in this manner can improve the practical performance of MALA. Even standard MALA can be difficult to apply in practice because the needed derivatives must be available and inexpensive; using the Jacobian in the manifold variant potentially makes it prohibitively expensive for many problems of interest. Therefore, this method is a particularly interesting one to adapt into the local approximation framework, as the surrogates can cheaply provide these derivatives. This method is also closely related to the preconditioning

performed in the Stochastic Newton method [78].

Thus far, we have not discussed how to select the step size ϵ; as with MH algorithms, the practical performance of MALA algorithms is highly dependent upon the step size. Scaling results suggest that the optimal step size is one that achieves an acceptance rate for standard MALA algorithms is 0.574 [93]; in practice, we use this theoretical result as a guide for adaptive algorithms that search for a good step size at runtime. For example, the step size may be refined continuously during MCMC, without altering the convergence of the chain, by stochastic approximation [2] or dual weighted averaging [58].

We follow [45] in performing finite, batch adaptation, which is a simple, yet effective strategy: begin by computing the average acceptance probability for a batch of samples, $\bar{\alpha}$. If the average acceptance probability deviates too far from optimal, the step size is altered. Specifically, if $\bar{\alpha} > 0.7$, then ϵ is multiplied by 1.2 and if $\bar{\alpha} < .4$, then ϵ is multiplied by 0.8. In the examples below, we choose batches of 50 samples and halt adaptation after the first 5,000 samples, which appears to be a reasonable choice in practice.

4.2.2 Approximate MALA

Now we can discuss why introducing local approximations into MALA is more complex than the MH case discussed earlier. Before, the selection of θ^+ and θ^- was invariant with respect to $\tilde{\mathbf{f}}$, so both could be passed into Algorithm 3.1 as fixed values. Additionally, we assumed that the proposal ratio $\frac{q(y,x)}{q(x,y)}$ portion of the acceptance probability computation cancels, as it does for MH algorithms, even though the delayed rejection component of DRAM violates this assumption. As the MALA proposal depends upon $\tilde{\mathbf{f}}$ and is highly non-symmetric, neither of these assumptions is possible. Algorithm 4.1 precisely states how the simplified manifold MALA algorithm may be adapted to use and refine local approximations. This algorithm is stated for the most complex case considered, the simplified manifold MALA with Gaussian likelihoods and priors, as simpler variants are easily formed.

The most subtle aspect of the algorithm is that the randomness used within the kernel, $\mathbf{z}$, is drawn before the approximations are built and remains constant even if the model is refined. This means that even though the proposed point θ^+ changes when the approximation is refined, so that under infinite refinement it converges to the proposed point computed with the true model. If $\mathbf{z}$ is re-sampled upon model refinement, samples are effectively rejected because the of the quality of the approximation, whereas acceptance or rejection of samples should only depend upon value of the approximate posterior density.

Cross validation error indicators are computed essentially as before, except that the acceptance probability includes the MALA proposal density. This computation requires carrying the leave-one-out computation through the posterior density, preconditioning matrix, and proposal density, which is compactly accomplished by the subroutines, $\tilde{p}^{\sim j}$, $\tilde{M}^{\sim j}$, and $\tilde{q}^{\sim j}$. When the superscript $\sim j$ is omitted, this im-

plies that $j = \emptyset$. This algorithm also calls the subroutines from Algorithm 3.1, ConstructApproximation and RefineNear, which are unchanged. Once the error indicators are computed, the refinement strategy described here is identical to that used in the previous algorithm.

Naturally, other preconditioning matrices may be substituted, and the choice of the manifold metric used here would need to be altered for non-Gaussian likelihoods or priors, but in principle this is straightforward. If the manifold metric preconditioning is not desired, for example, because the Fisher information matrix is not easily computable, simply replace the computed matrix in $\tilde{M}^{\sim j}$ with an identity matrix. This is also necessary if the approximation is built on the log-likelihood instead of the forward model, as the Jacobian of the forward model would not be available, and the method $\tilde{p}^{\sim j}$ is easily adjusted to reflect that difference.

4.2.3 Experimental results

Once more, we return to our three standard examples to explore the performance of the approximate MALA samplers. We do not expect MALA to outperform DRAM in this setting, as the chains are sufficiently long for either to perform well. Instead, we simply want to demonstrate that the chains mix using a comparable number of samples. We do not present results from the genetic toggle switch example, because the posterior is largely defined by the compact support of the prior, which causes MALA to perform poorly. The discontinuous prior behavior does not appear in the inputs to MALA, so the step size becomes quite small to avoid proposing too many moves outside the prior, but this induces unacceptably slow mixing in the MCMC chain.

The elliptic PDE problem is sampled with the simplified manifold MALA using indirect, quadratic approximations. As shown in Figure 4-4, this sampler uses a small number of samples and accurately reproduces the posterior. This example illustrates the usefulness of the simplified manifold correction. Figure 4-5 depicts the trace of two MCMC chains, one using the manifold correction and the other using standard MALA. The parameters are not scaled uniformly, in that the width of the posterior band is much larger for some parameters than others. In standard MALA, the step size must be set small enough to ensure that the parameter with the smallest posterior range mixes well, but the parameters with larger bands are explored slowly. The ranges are essentially correct, and given a much longer chain and significantly increased computational cost, standard MALA would probably compute the moments correctly. The simplified manifold preconditioning rescales these parameters so that they all mix rapidly, regardless of whether approximations are used. Although it would not typically be affordable to construct indirect approximations using the entire Jacobian, this model is inexpensive enough to make it feasible; Figure 4-6 demonstrates that for both manifold MALA and DRAM, the addition of the Jacobian information reduces the number of model evaluations needed, as expected.

For the FitzHugh-Nagumo problem, we simulate chains using simplified manifold

Algorithm 4.1 Simplified manifold MALA with local approximations

procedure ApproximateMALAKernel($\theta^-, \mathcal{S}, \mathcal{L}, \mathbf{d}, p, \beta_m, \gamma_m$)
  $\mathbf{z} \sim \mathcal{N}(0, I_{d\times d})$ ▷ Draw randomness
  while forever do
    $\theta^+ \leftarrow \theta^- + \frac{\epsilon^2}{2}\tilde{M}(\theta^-)\nabla_\theta \tilde{p}(\theta^-|\mathbf{d}) + \epsilon\sqrt{\tilde{M}^{\sim j}(\theta)}\mathbf{z}$ ▷ Compute proposal
    $\alpha = \min\left(1, \frac{\tilde{p}(\theta^+|\mathbf{d})\tilde{q}(\theta^+,\theta^-)}{\tilde{p}(\theta^-|\mathbf{d})\tilde{q}(\theta^-,\theta^+)}\right)$ ▷ Nominal acceptance probability
    $\epsilon^+ \leftarrow \max_j \left|\alpha - \min\left(1, \frac{\tilde{p}^{\sim j}(\theta^+|\mathbf{d})\tilde{q}(\theta^+,\theta^-)}{\tilde{p}(\theta^-|\mathbf{d})\tilde{q}^{\sim j}(\theta^-,\theta^+)}\right)\right|$
    $\epsilon^- \leftarrow \max_j \left|\alpha - \min\left(1, \frac{\tilde{p}(\theta^+|\mathbf{d})\tilde{q}^{\sim j}(\theta^+,\theta^-)}{\tilde{p}^{\sim j}(\theta^-|\mathbf{d})\tilde{q}(\theta^-,\theta^+)}\right)\right|$
    if $\epsilon^+ \geq \epsilon^-$ and $\epsilon^+ \geq \gamma_m$ **then** ▷ If needed, refine near larger error
      $\mathcal{S} \leftarrow$ RefineNear($\theta^+, \mathcal{S}$)
    else if $\epsilon^- > \epsilon^-$ and $\epsilon^- \geq \gamma_m$ **then**
      $\mathcal{S} \leftarrow$ RefineNear($\theta^-, \mathcal{S}$)
    else if $u \sim \mathcal{U}(0,1) < \beta_m$ **then** ▷ Refine with small probability
      Randomly, $\mathcal{S} \leftarrow$ RefineNear($\theta^+, \mathcal{S}$) or $\mathcal{S} \leftarrow$ RefineNear($\theta^-, \mathcal{S}$)
    else ▷ Accept approximations
      Accept move θ^+ with probability α
    end if
  end while
end procedure

procedure $\tilde{p}^{\sim j}(\theta|\mathbf{d})$
  return $\mathcal{L}(\theta|\mathbf{d}, \text{ConstructApproximation}(\theta, \mathcal{S}, j))p(\theta)$
end procedure

procedure $\tilde{M}^{\sim j}(\theta)$
  $\tilde{\mathbf{f}} \leftarrow$ ConstructApproximation($\theta, \mathcal{S}, j$)
  return $\left[(\nabla\tilde{\mathbf{f}})\Sigma_{\mathcal{L}}^{-1}(\nabla\tilde{\mathbf{f}}) + \Sigma_p\right]^{-1}$
end procedure

procedure $\tilde{q}^{\sim j}(x, y)$
  return $\mathcal{N}\left(y; x + \frac{\epsilon^2}{2}\tilde{M}(\theta^-)\nabla_x \tilde{p}^{\sim j}(x|\mathbf{d}), \epsilon^2\tilde{M}^{\sim j}(x)\right)$
end procedure

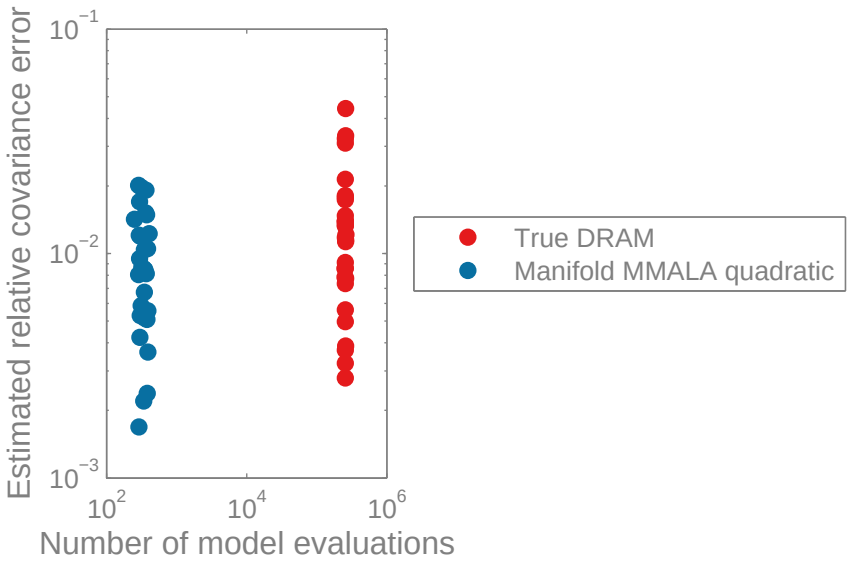
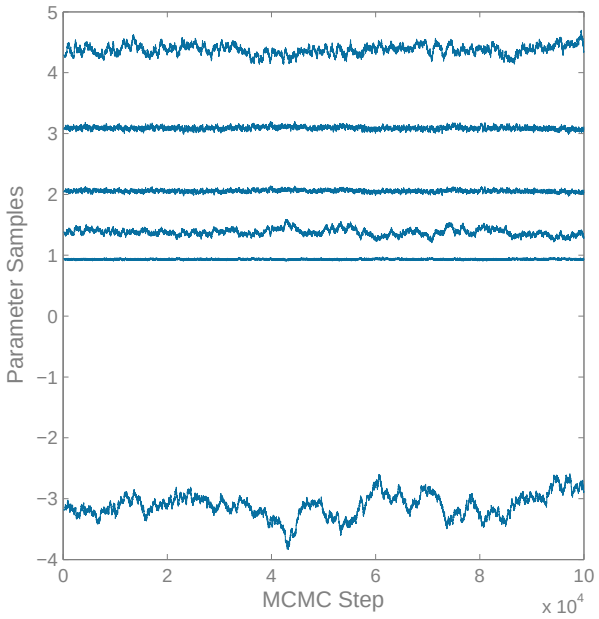


Figure 4-4: Approximate relative covariance errors of the elliptic PDE example, compares reference solution from a DRAM sampler using the true model with manifold MALA using indirect quadratic approximation.

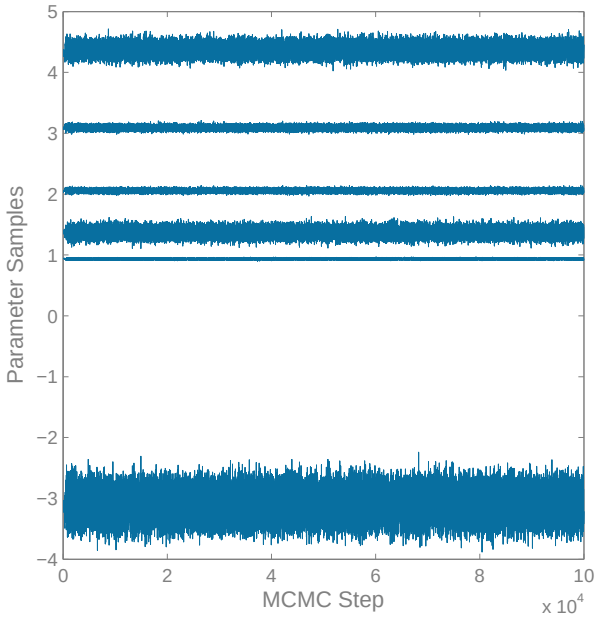
MALA. Using the manifold correction again requires indirect approximation, but this model supports either quadratic or Jacobian-based quadratic approximations. The results are shown in Figure 4-7, showing that the MALA chains accurately draw samples from the posterior. Curiously, the introduction of Jacobian information increases the cost of MALA chains, whereas we expect the extra derivative information to reduce the cost. In this problem, the Jacobians provided are estimated by the Sundials package, and are not guaranteed to be numerically consistent with the forward solution, as in an adjoint method. It is possible that the manifold methods are more sensitive to this inaccuracy than DRAM, hence the poorer performance in that case.

4.3 Sharing local approximations for parallel MCMC

Many computational methods are made useful in practice by running them in a parallel computing environment. Unfortunately, approaches for scaling MCMC to take advantage of a large number of processors are limited by the serial nature of MCMC [96]. Although it is trivial to run multiple, independent chains in parallel, convergence results for MCMC only apply as the chain lengthens, not as more chains are simulated. Hence, each chain must duplicate the effort required for burn-in and must be sufficiently long for its samples to be drawn roughly from its limiting distribution. For more complex distributions, relatively long chains and burn-in periods are needed, and hence simulating many chains becomes less efficient.



(a) Standard MALA.



(b) Simplified Manifold MALA.

Figure 4-5: Traces of the parameter values for two MCMC chains sampling the posterior distribution of the elliptic PDE example, comparing the impact of the simplified manifold correction.

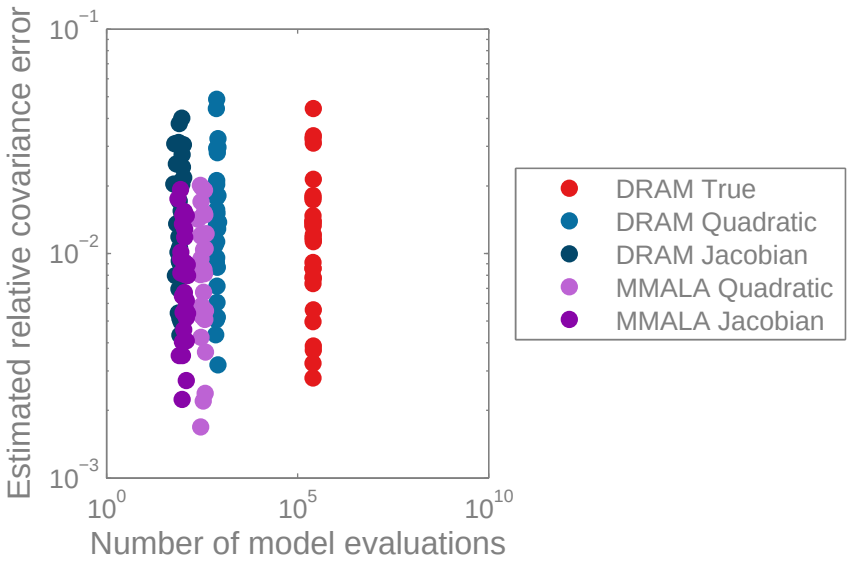


Figure 4-6: Approximate relative covariance errors of the elliptic PDE example, compares reference solution from DRAM sampler using the true model with DRAM or manifold MALA using indirect quadratic and Jacobian-based quadratic approximations.

Although MCMC does not parallelize perfectly, it is often advisable to run a modest number of chains for the purposes of validation. It is difficult to tell whether a single chain has converged enough to become representative of its stationary distribution, but if multiple chains all appear to sample the same distribution, then it becomes much more feasible to conclude that their results are usable [25, 10]. Additionally, exploration of the parameter space can be increased by exchanging information between the chains - if one finds a high posterior density region of interest, this information may be shared with the others [26]. Population MCMC algorithms explore a family of tempered distributions with parallel chains, so that swapping states of the chains can provide long-range moves [12].

Just as in single chain MCMC, any of these parallel approaches requires repeated evaluations of the forward model, which can dominate the overall cost of the algorithm. Direct application of the Algorithm 3.1 as described thus far would require each parallel chain to compute its own approximation, running the forward model as required. However, as each chain eventually needs an accurate approximation of the same function, building them independently appears inefficient. Instead, an obvious improvement is for the parallel chains to collaborate by sharing a common set of evaluations, $\mathcal{S}$. The initialization of $\mathcal{S}$ may be performed in parallel, and thereafter, whenever a refinement occurs, the result is shared with all the chains. Fortunately, this strategy should not alter the theory regarding the convergence of the sampler and requires no fundamental changes to the proposed algorithms, but can be an important improvement in practice, as we discuss subsequently.

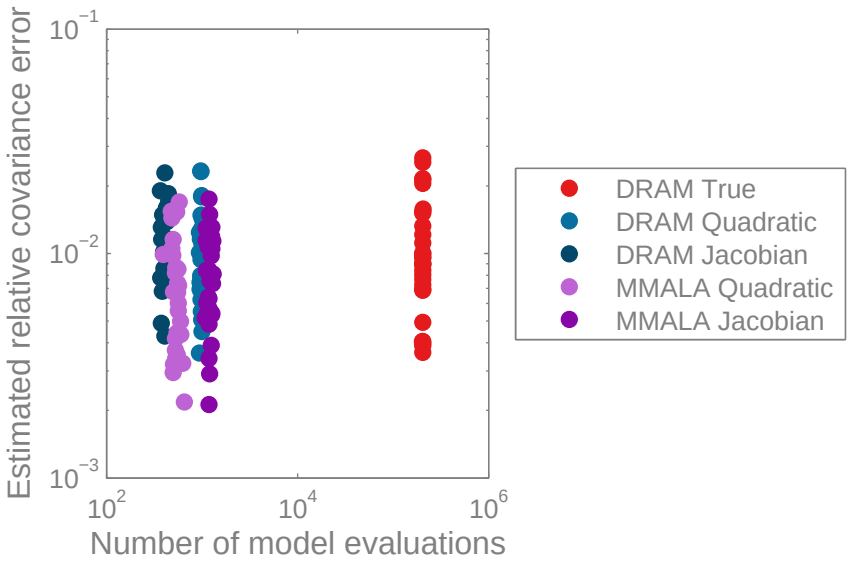


Figure 4-7: Approximate relative covariance errors of the FitzHugh-Nagumo example with indirect approximation, comparing the performance of DRAM or simplified manifold MALA, as combined with quadratic or Jacobian-based quadratic approximations.

Our implementation does not synchronize the chains that are run in parallel. Instead, each chain is free to compute and accept samples at its own speed, sharing forward model evaluations whenever they are completed. This means that the different chains may have slightly different sample sets, for example, if one completes early. Since the refinement criteria is unchanged, the correctness of each chain remains decoupled from the others, so this discrepancy should have little impact on the overall results.

To compare the performance of sharing evaluations in parallel, we again return to the elliptic PDE and FitzHugh-Nagumo models. We compute ten chains in parallel with shared evaluations and compare the collection of chains to ten of the chains computed without shared evaluations. For the elliptic PDE example, we compare DRAM based chains using indirect, quadratic approximation. The results are shown in Figure 4-8. The non-synchronization of the chains causes the parallel chains to have used slightly differing numbers of evaluations, but all ten chains were constructed for the cost of the maximum sample. This contrasts with the serial chains, where cost of each chain is independent and must be summed, therefore, these collections of chains cost 3350 evaluations in serial and 655 in parallel. For the FitzHugh-Nagumo problem, we simulate DRAM chains with indirect Jacobian-based quadratic approximation, as shown in Figure 4-9. In this case the serial chains cost 4098 evaluations and in parallel the cost is reduced to 1087 evaluations. Both examples show that the serial and parallel collections of approximate chains are of similar quality to the true model. The parallel configuration is somewhat more expensive than a single chain, but is

significantly cheaper than a collection of chains of the same size computed without sharing evaluations.

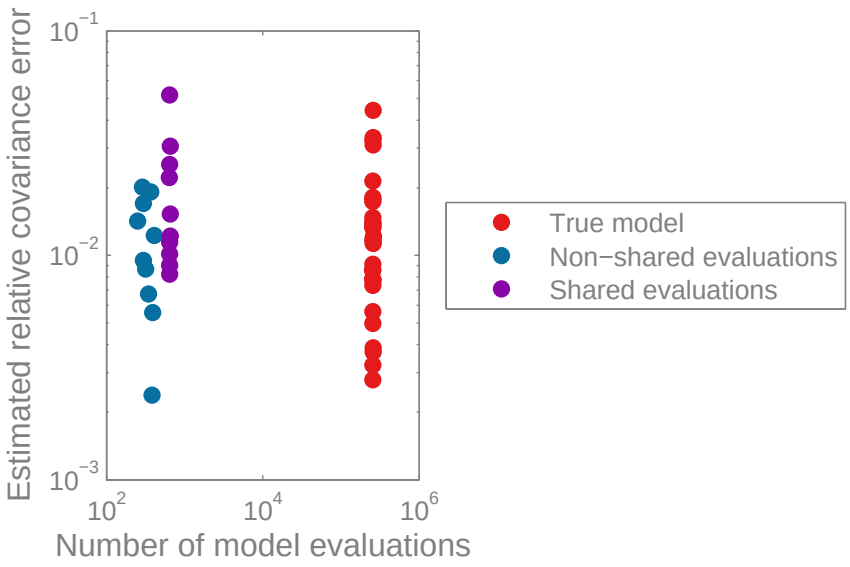


Figure 4-8: Approximate relative covariance errors of elliptic PDE example with a DRAM based sampler and indirect quadratic approximations, comparing chains computed serially or in parallel with shared evaluations.

4.4 Discussion

This chapter seeks to enrich our development of approximate samplers by exploring variations on our framework, showing that a variety of useful algorithms are feasible. Specifically, we introduced derivative-based approximations, MALA-based samplers, and how to construct approximations in parallel. Broadly, these techniques produce correct results and in some cases further improve on the existing performance. Perhaps most interesting is that the adaptation strategies remain essentially unchanged, even with combinations of these variants. Although MALA does not sample all these problems well, the issues discussed with the genetic toggle switch and elliptic PDE problems occur regardless of approximation, and are easily diagnosable as failures of the sampler. This indicates that the proposed framework is widely applicable, is robust, and that future work should be able to develop a rich class of algorithms in this family.

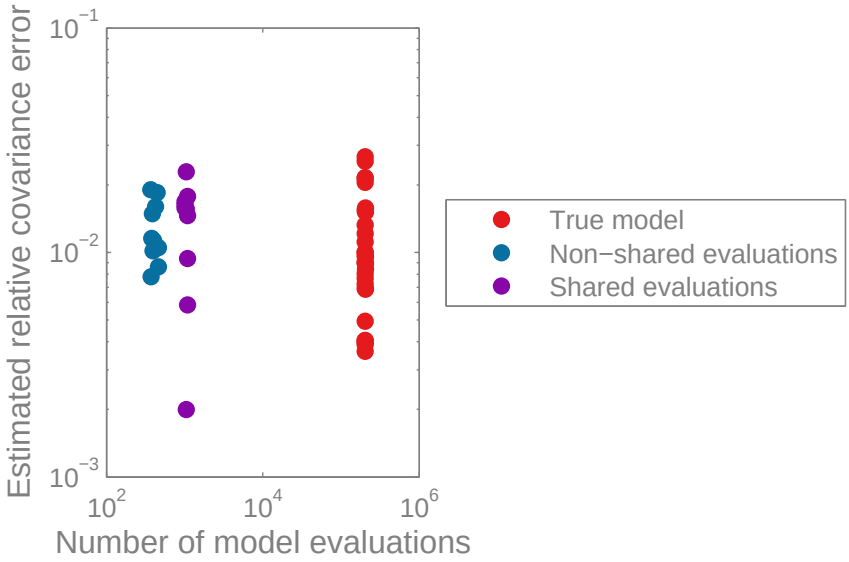


Figure 4-9: Approximate relative covariance errors of FitzHugh-Nagumo example with a DRAM based sampler and indirect Jacobian-based quadratic approximations, comparing chains computed serially or in parallel with shared evaluations.

Chapter 5

Inference in the Pine Island Glacier setting

Pine Island Glacier is an important outlet along the Amundsen coast of the Western Antarctic Ice Sheet (WAIS), which has recently accelerated and thinned [76]. It is a fast flowing ice stream draining the relatively slow flowing Antarctic interior, terminating in a floating ice shelf. Antarctica primarily loses ice mass through such flow into the ocean, and Pine Island Glacier serves to regulate the discharge, making it a primary research interest.

The dynamics of the system and the loss of ice mass demonstrate important couplings to the ocean [62]. Surface water is either frozen or near-freezing year-round, thus contributes little to melting the ice shelf. However, the so-called Circumpolar Deep Water (CDW) is a mass of warm, salty, dense water that resides in the lower portion of the water column. Currents introduce this warm water beneath the ice shelf, where it contributes to melting the glacier [62]. Of particular interest is the interaction of the CDW with the grounding line, the region where the ice transitions from grounded on bedrock to floating; the sharp difference between the dynamics on either side of the grounding line mean that its evolution has a large impact on the overall system [65]. When the salty, warm water is brought into contact with the glacier, it melts ice, cooling and freshening the water. Reduced density causes this fresher, near-freezing water to rise along the underside of the ice shelf, creating a melt plume that has a boundary layer effect at the ice-ocean interface and which emerges at the edge of the glacier at the ocean's surface [105].

Pine Island Glacier has become a widely studied example, as shown by recent modeling efforts [87] and field campaigns to deploy autonomous submarines [62] or drill through the ice [105]. Although these efforts have yielded important results, they also reveal the difficulty of obtaining observations and the large uncertainties in our understanding of the coupling between the ice and the ocean.

In this work, we perform a Bayesian analysis in the Pine Island setting, inferring parameters and computing the predicted melt-rate, which is made feasible by applying our approximate MCMC sampler. Our model is the ocean model for the Pine

Island system developed in [29], and we infer a small number of scalars governing the coupling of the ice and ocean. For this initial work, we restrict ourselves to synthetic observations in a steady-state model, as it is difficult to provide the type of accurate time-varying forcings needed to work with real observations. Even so, this is an important step towards large scale parameter inversion using real data and accurate models. Although the observations are synthetic, their spatial distribution is realistic, hence interesting conclusions can be made, such as how informative different types of observations are and how coupled parameters are given measurement.

5.1 Constructing a Pine Island inference problem

We perform inferences using a model for Pine Island developed by [29], which we briefly review. This model is based on the Massachusetts Institute of Technology general circulation model (MITgcm), a "state-of-the-art scalable finite-volume numerical ocean general circulation model" [53]. This model has been configured to support cavity flow with realistic bathymetry and ice shelf geometry from surveys of the Pine Island Glacier ice shelf and the immediately surrounding bay. The model is available in a fine-resolution (1×1 km^2) and coarse-resolution (4×4 km^2) mode, both of which are used in this study. Figure 5-1 depicts the bathymetry of the fine resolution cavity and Figure 5-2 shows a simulated flow.

The ocean model has been augmented to simulate the exchange of heat and fresh water between the ocean and ice, based on the model given by [59], described in detail by [29]. This combined model uses several constant parameters whose values are the target of our inference and serve as the input to the model, for example, drag coefficients and the bulk parameters governing the exchange of heat and salt near the ice-ocean boundary. Although proposed values exist in the literature, there remains considerable uncertainty, which can result in considerable changes in model predictions [29]. We perform this inference in a synthetic environment, using converged steady-state results using fixed boundary conditions and forcings informed by data from Pine Island.

Collecting oceanographic measurements near the Pine Island ice-shelf is a difficult task, hence data is typically sparse in both space and time. In an effort to simulate realistic inferences, we consider point measurements of temperature and salinity, which are basic oceanographic measurements, collected from two realistic sources. First, research vessels have collected observations from the water column at several locations along the ice front [61, 76]; we simplify this data source to include observations from just below the surface at eight locations along the ice front, similar to those shown in Figure 1 of [61]. Second, a recent expedition drilled through the ice shelf to place sensors in the cavity just below the ice shelf [105]. We simulate this type of data with two observations from the top of the cavity, again similar in location to the actual drilling sites. Both sets of locations are depicted in Figure 5-1. The drilling observations are believed to be especially useful because the behavior of the boundary layer, formed in the ocean near the ice interface, has significant impact on the overall

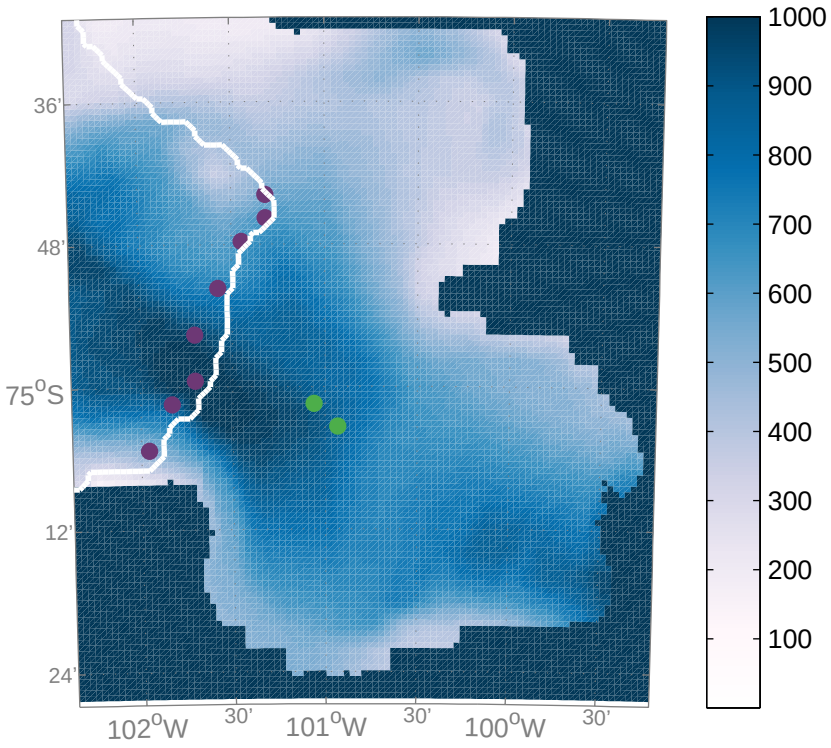


Figure 5-1: A top down view of PIG, depicting locations of observations; white is the ocean surface, dark blue is grounded ice, and the color represents ocean floor depth in meters. The surface measurements are shown with purple dots and the drilling sites with green.

dynamics of the coupled ice-ocean system [105]. We will investigate this by comparing the results of inference with only the surface observations to results of inference using both surface and drilling observations.

When performing evaluations of inference techniques in a synthetic setting, it is important not to simulate the data from the same model as is used during inference. This is known as an "inverse crime" and will produce optimistic predictions about the quality of feasible inferences because of the unrealistically strong match between the data and model [66]. Therefore, we use the fine resolution model to produce the synthetic observations and perform the inference using the coarse scale model.

A fully Bayesian methodology uses the posterior distribution over the parameters to characterize the likely parameters and to induce distributions over predictions. In the Pine Island setting, the principle prediction of interest is how quickly the ice sheet

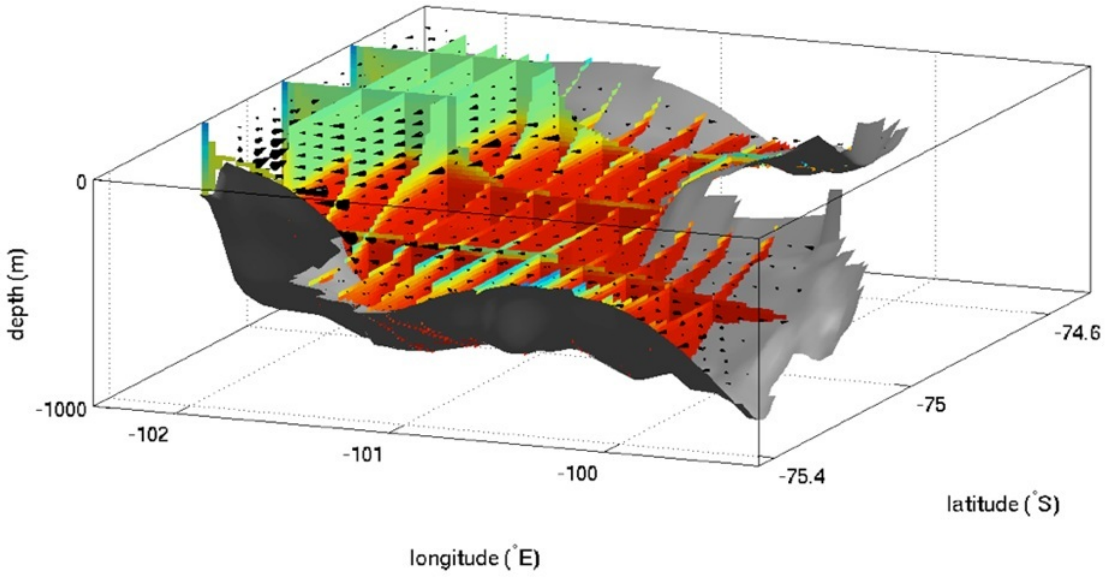


Figure 5-2: Simulated ocean circulation under Pine Island Ice Shelf with the MIT general circulation model

will melt from below, as represented by the instantaneous melt-rate in our model. As with parameter estimates, the difference between prior and posterior distributions of the prediction will help investigate how much about the melt-rate has been learned through the inference.

5.2 Prior and likelihood selection

To conduct our Bayesian inference, we must precisely specify prior and likelihood distributions. The prior summarizes our knowledge of the parameter values without considering the data. We selected nine scalar parameters governing the coupling of the ice to the ocean in our simulation, but three pairs of parameters are set to be equal, leaving six independent quantities to infer. As these parameters are positive and their values are relatively unknown, we endow them with independent log-normal priors. That is, for parameter i, under the prior, $\theta_i \sim \exp \mathcal{N}(\mu, \sigma)$. The literature suggests, along with some expert judgment, nominal values μ' and width parameters σ'. We may relate these as $\mu = \log \mu'$ and $\sigma = \log(\mu' + \sigma'/4) - \mu$. The scaling factor, $1/4$, makes the prior fairly restrictive within these bounds, and is selected to avoid large parameter changes that induce the numerical solver to diverge. Longer simulations and a smaller time-step should resolve this issue, allowing the inference to proceed over a wider prior, but this would make the forward model more expensive to run and is not explored here.

The simulated data are generated from the fine grid model, run at the prior mode, that is, the μ' values. At the coarse scale, the data will simply be the temperature or salinity at the center of the grid cell at the locations of interest, but we have some

Table 5.1: The inference parameters and their priors.

Parameter	Nominal value, μ'	Prior "width" σ'
Drag coefficients	1.5E-3	1.5E-3
Heat & Salt transfer	1.0E-4	0.5E-4
Prandtl Number	13.8	1.
Schmidt Number	2432.	200.
Horizontal Diffusion	5.0E-5	5.0E-5
ZetaN	5.2E-2	0.5E-3

choice in creating corresponding data from the finer scale simulation. The grids of the coarse and fine scales are aligned, so each coarse scale cell contains 4×4 fine scale cells. We create the observation by averaging the results from the center 2×2 fine scale cells. We have not corrupted these observations with pointwise noise, so the complexity of the inference stems from the difference between the two models; this is often the case, so the comparison is still interesting.

The fine scale data is generated from two years of simulation, which is sufficient to create equilibrium, given that the boundary conditions are constant. Much of the effort of the model is spent adjusting the arbitrary initial conditions into physically realistic conditions. Therefore, we speed up the coarse model by creating an initial state for the prior mode, brought into equilibrium with a two year simulation. Then the model used during inference evolves this state forward for three months further with the set of parameters of interest, and observations are collected from the final state.

The likelihood is an uncorrelated normal distribution centered on the perturbed data generated from the fine scale model. This distribution encompasses measurement errors, short-term variability, model error, and sensor location error. From their subsurface observations, [105] provides time-series of the observed temperatures, demonstrating that they vary slowly. Therefore, we only simulate a single observation at each location with relatively low error, which in practice would correspond to short-term averages of measurements drawn from each location. The temperature and salinity measurements have differing variances, $\sigma_t^2 = (0.04\,^\circ\mathrm{C})^2$ and $\sigma_s^2 = (0.1\,\mathrm{psu})^2$, respectively. These variances are derived from observed data [105] and the variation within the model; it is reasonable to have small variance in the temperature observations in this case, because the observations are all drawn water masses that must be very close to the local freezing conditions. The fine scale data are perturbed with an uncorrelated, zero mean Gaussian, with standard deviations equal to one-half of the likelihood standard deviations.

5.3 Computational results

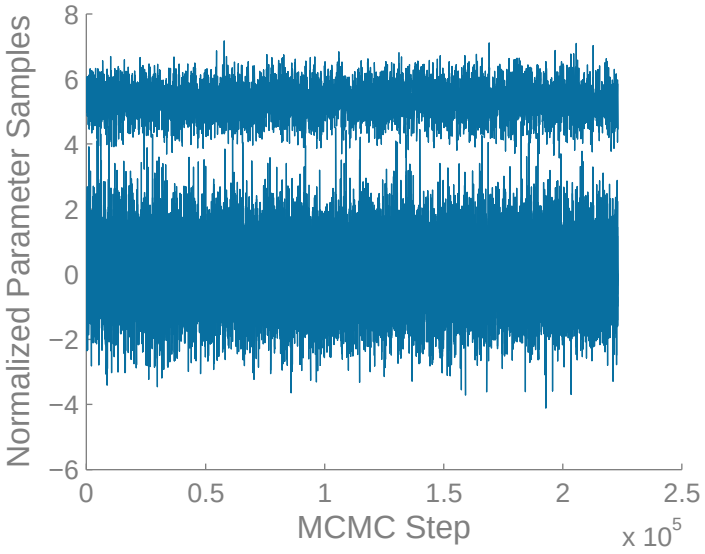
We infer the six parameters of Table 5.1, using the open surface data and the combined open and drilling data. We run 30 chains in parallel with a shared pool of evaluations. The MCMC kernel is DRAM and an indirect approximation is constructed, given the evidence from the previous chapters that suggests this strategy is more efficient. As before, the parallel chains are not synchronized, and since the inference is expensive, we do not wait for a specified length of the chains. Instead, we allowed the MCMC chains to proceed for approximately two weeks and then collected the state of the chains. The first 1,000 samples of each chain is discarded as burn-in, and the remaining chains are kept, which range between lengths of around 8,000-15,000 steps. All of the chains are initialized with an approximation of the *maximum a posteriori* point, as found by numerical derivative-free optimization; this optimization is relatively inexpensive as compared to the cost of MCMC and contributed to our ability to use such a short burn-in.

5.3.1 MCMC convergence and posterior distribution

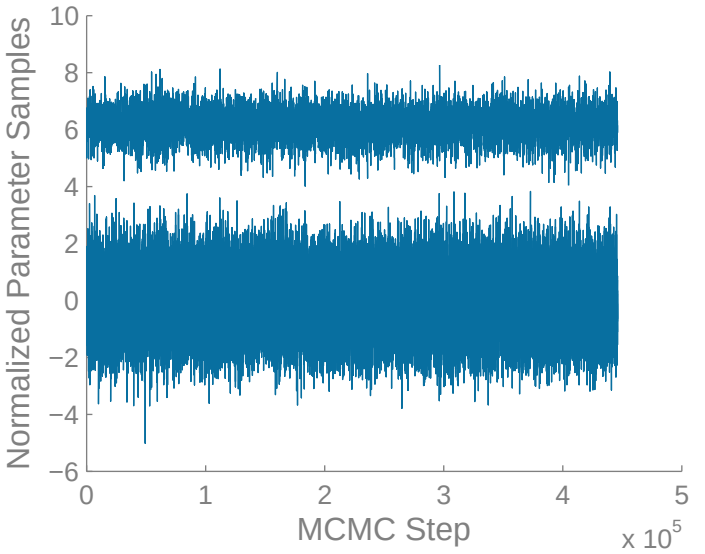
As a first step, we must establish that the MCMC chains have converged. We begin by plotting the traces of the combined chains for both data sets in Figure 5-3, which appear well mixed; visual inspection of the chain is often a good preliminary diagnostic. The parameters are plotted in normalized coordinates, that is, where the prior is a standard normal, $\mathcal{N}(0, 1)$. Additionally, we check that the various chains all compute the same result by comparing the covariances of the 30 individual chains to the covariance of the merged chain. We plot the result in Figure 5-4, as in previous experiments, observing that all the constituent chains are consistent with the combined result for both data sets.

With burn-in removed, the combined, usable chains are approximately 225,000 and 450,000 steps long and were computed with around 53,000 and 52,000 samples, for drilling and surface only, respectively. Therefore, the approximation appears to have reduced the necessary number of model runs by a factor of five to ten, even without considering that DRAM typically uses several model runs per step.

Plotting the marginals in normalized coordinates allows us to easily compare to the circular contours of the standard normal prior, as shown in Figure 5-5. We see that both data types have posteriors of similar size and exhibit little correlation in the two dimensional marginals. They roughly coincide on the inferred distribution of the drag coefficients, which are very similar to the prior. The posterior of the latter four parameters appear to be the prior, regardless of which data set is used, implying that the data provides little information about these parameters. However, the inferred heat and salt transfer coefficient is much higher than suggested by the prior, and varies somewhat between the two observations. As both are far outside the prior range of a standard normal, there must be a significant signal in the data, although the posterior is still not highly concentrated in this parameter value either. Although the parameters are intended to be scale-independent, which would imply that the

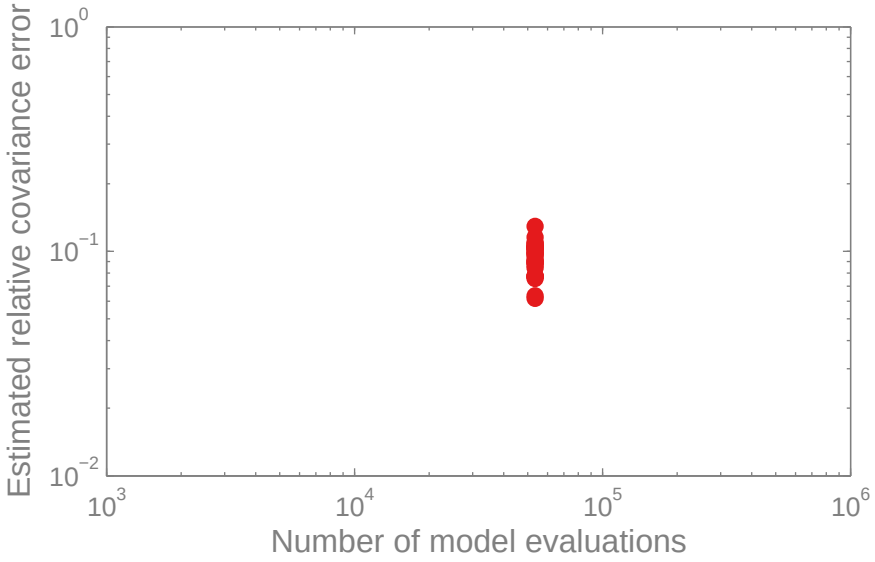


(a) Drilling and surface observations.

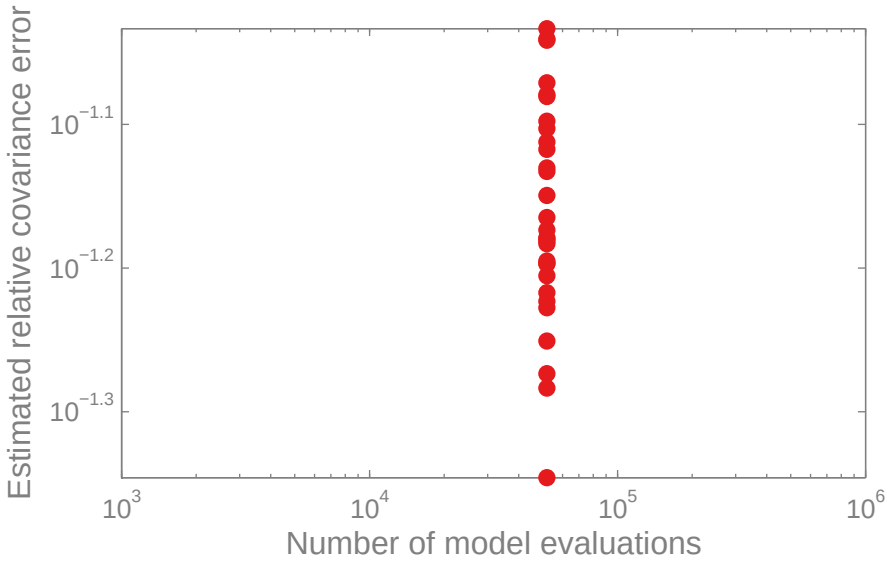


(b) Surface observations only.

Figure 5-3: Traces of the first two components of the combined MCMC chains in normalized coordinates, thinned for visual clarity.



(a) Drilling and surface observations.



(b) Surface observations only.

Figure 5-4: The relative covariance error between the 30 individual MCMC chains and chain formed by merging all 30 chains.

ideal result is for the posterior to concentrate around the values used in the fine-scale, the response of the model to the parameters may have some scale dependence, implying that a shift is appropriate. Figure 5-6 shows the posterior distribution of the parameters in their natural values, which illustrates their non-Gaussianity after the exponential transform.

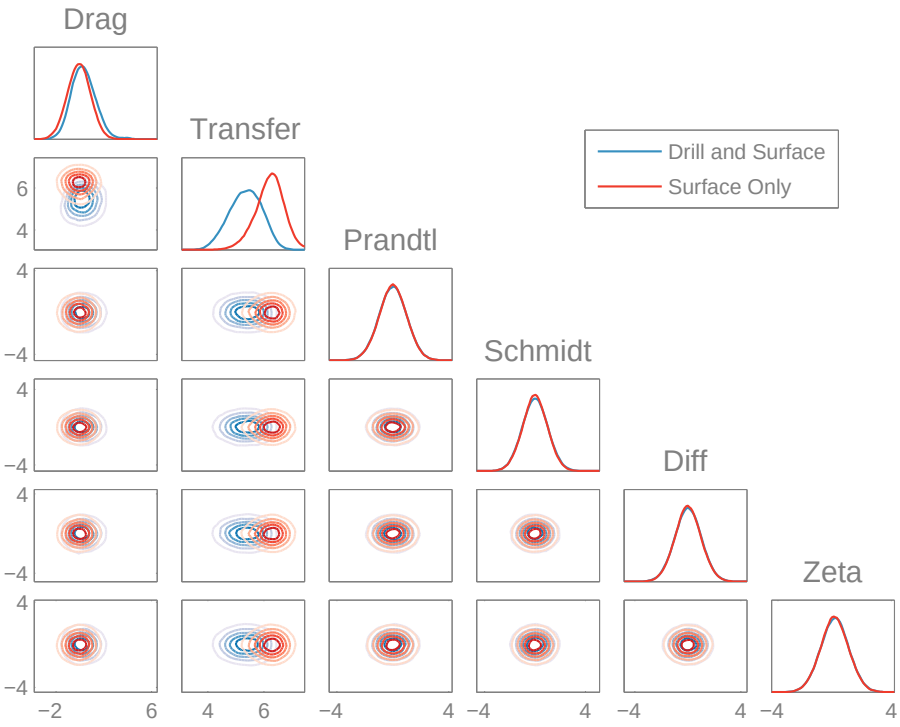


Figure 5-5: Marginals of the Pine Island Glacier posterior in normalized coordinates, for both data sets.

5.3.2 Prior and posterior model predictions

To help interpret these results, we now explore the model outputs of the fine scale model, and the coarse model at the prior mean parameters and the posterior mean parameters for the combined observations [1]. We are interested in the behavior of three fields, the temperature and salinity of the ocean, and the melt-rate under the ice shelf. We plot the temperature and salinity at the highest point in the water column, as

[1] We do not repeat the exercise for the surface-only observations for brevity.

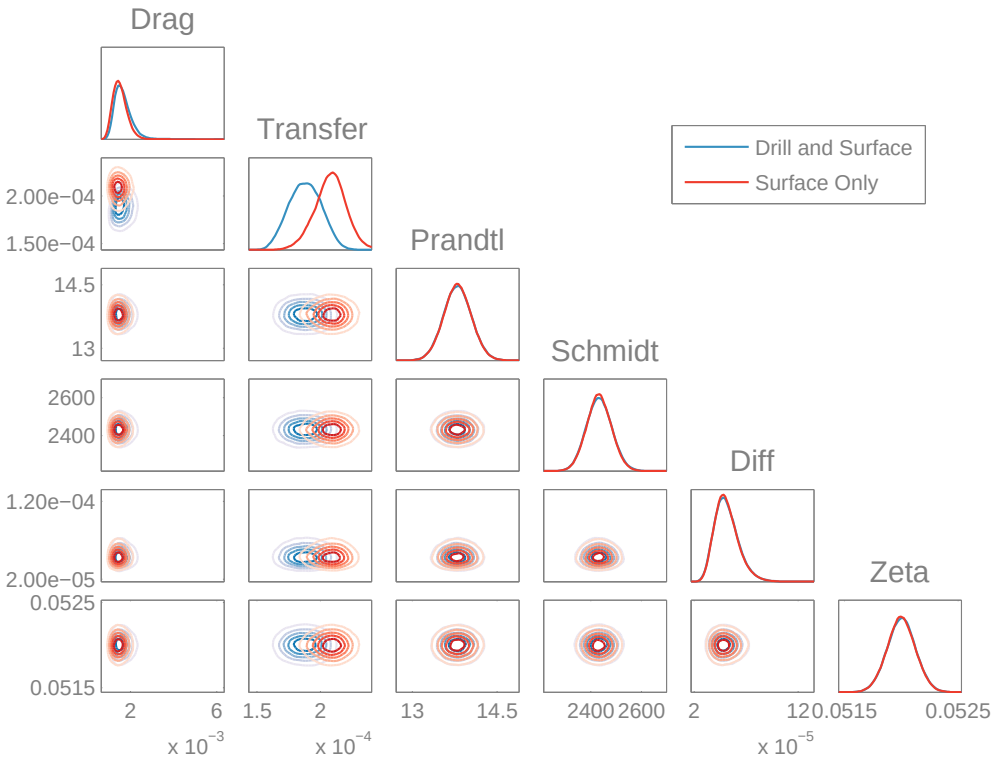


Figure 5-6: Marginals of the Pine Island Glacier posterior in the natural parameter values, for both data sets.

this is representative of the observations provided here for inference, while providing a richer sense of the model output; the melt-rate is naturally a two dimensional field.

Figure 5-7 shows the temperature field. The fine scale results clearly show bands created by the discrete jumps in depth. Both coarse scale models show colder temperatures at the open surface, on the top-left of the domain. Figure 5-8 depicts the salinity fields; the coarse scale models overall appear to have higher salinity and show a pocket of low salinity near the bottom of the domain that is not present in the fine scale. This salinity difference is larger than expected for realistic variations and may suggest inaccuracy in the coarse scale model.

To allow a more detailed comparison, we plot the difference in these fields between the fine scale and the prior mean, which depicts the signal present in the data that is not explained by the prior. Similarly, we plot the difference between the posterior mean and the prior mean; if the shift from the prior to posterior mean perfectly explained the data, these two difference fields would be identical. The temperature differences shown in Figure 5-9 indicate that the posterior mean parameters induce

overall cooler temperatures, which is in agreement with both the drilling locations and the border of the ice front, although there is also evidence of significant differences for most of the open surface and near the grounding line on the right of the domain. The salinity differences in Figure 5-10 suggests that the posterior actually makes progress towards recovering the observed data, for example increased salinity in the center, reduced salinity at the top and right; much of the pattern appears similar, but the magnitude of the posterior correction is smaller overall.

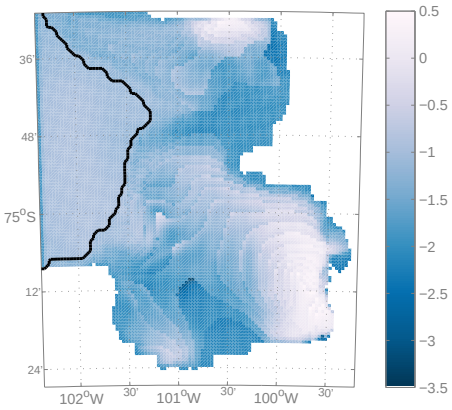
An important Bayesian analysis is to examine the predictive distributions for the observations under the prior and posteriors, as compared to the likelihood distribution, which is centered at the observed data. Figures 5-11 and 5-12 show distributions for the temperature and salinity, respectively, each at two observation locations on the surface. The prior distribution is computed by constructing a high-order, adapted polynomial chaos expansion, constructed as in Chapter 2, which is then sampled under the prior. The posterior values are taken from the approximate observations computed by the MCMC chains. These graphs indicate that the posterior predictive distribution is closer to the likelihood center than the prior, but neither is in strong agreement with the data. The salinity likelihood is much wider than the actual observed variation in the model.

As mentioned earlier, our primary motivation for performing inference on the parameters is to inform our predictions of the melt-rate of Pine Island glacier. Figure 5-13 illustrates the pattern of melting induced in the fine scale, prior mean, and posterior mean cases. While the prior mean and fine scale figures are relatively similar, the posterior mean melt-rates are overall much higher, as a result of the increased salt and heat transfer.

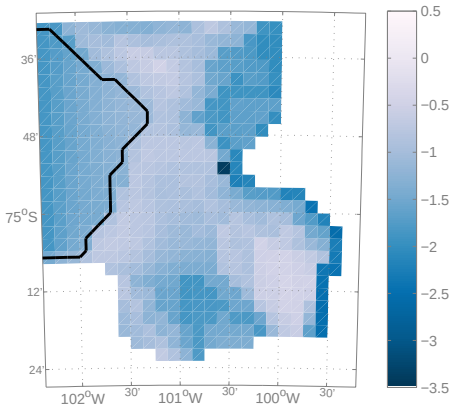
Reducing the target of interest to the integration of the melt rate over the entire ice shelf, we can again plot the predicted distributions, as in Figure 5-14. The posterior values are estimated by the quadratic approximation along with the observed quantities, using the same sample set; the only difference is that because the melt-rate was not directly observed, it does not appear in the computations determining when refinement occurs. The fine scale result lies within the high density region of the prior, but both posteriors are shifted to suggest much higher melt-rates. Our aim in performing inference is to construct a posterior for the parameters that produces good agreement between the posterio predicted melt-rates and the fine-scale truth. Although the observation distributions indicated better agreement with the data under the posterior than the prior, the inference appears to have significantly reduced the quality of this prediction.

5.4 Discussion

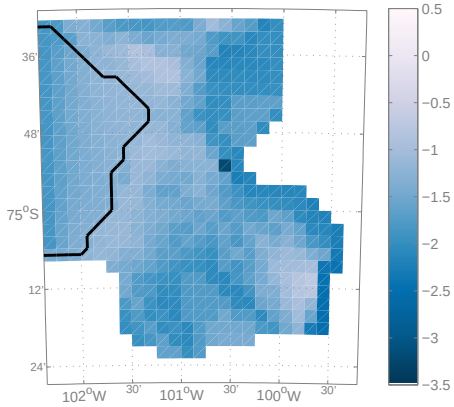
We have successfully applied our approximate samplers to the challenging domain of ice-ocean coupling in Pine Island Glacier. In approximately two weeks, the chains appear to have converged to a usable degree, reducing the computational cost by a factor of five or more. Unfortunately, the data appears to be uninformative, in that



(a) Fine scale.

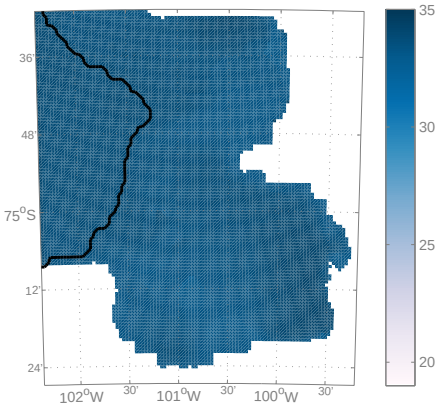


(b) Coarse scale, prior mean.

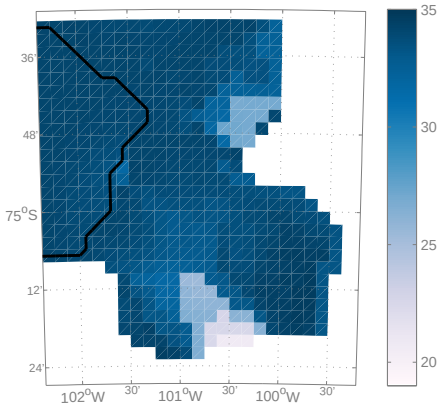


(c) Coarse scale, drilling and surface observation posterior mean.

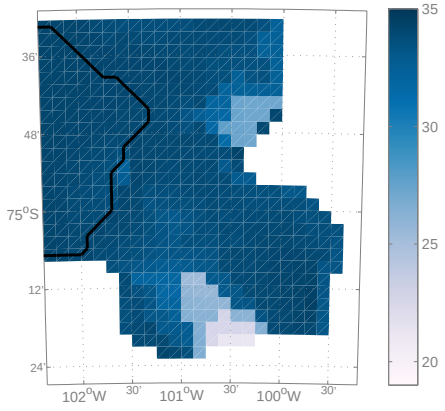
Figure 5-7: The temperature in degrees Celsius at the top of the water column for several configurations.



(a) Fine scale.

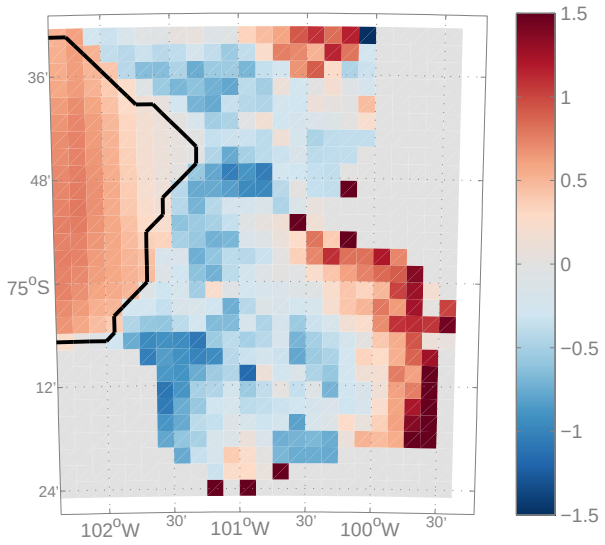


(b) Coarse scale, prior mean.

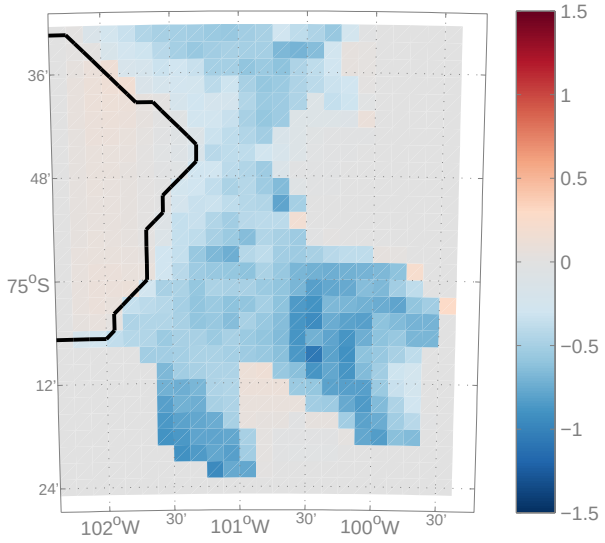


(c) Coarse scale, drilling and surface observation posterior mean.

Figure 5-8: The salinity at the top of the water column for several configurations.

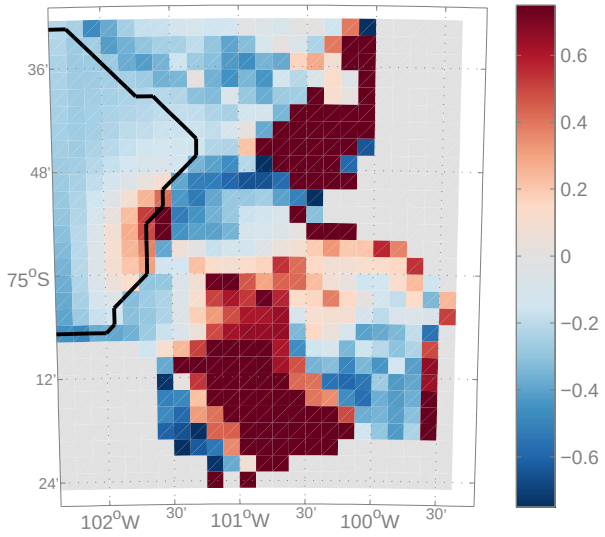


(a) The difference between the fine scale and prior mean.

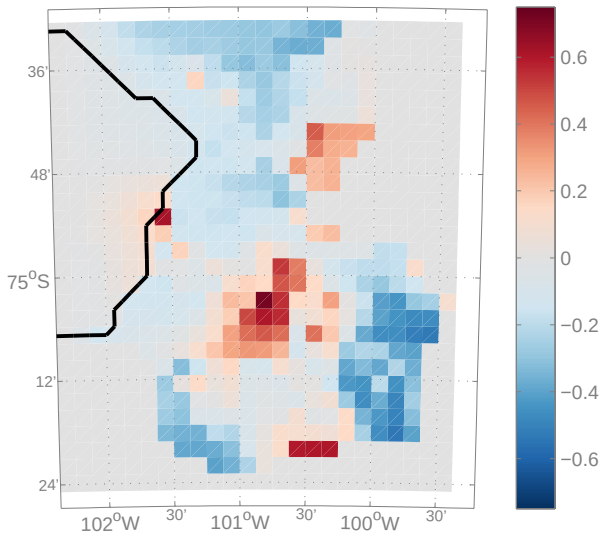


(b) The difference between the posterior mean and prior mean.

Figure 5-9: Comparisons of the temperature at the top of the water column for several configurations.

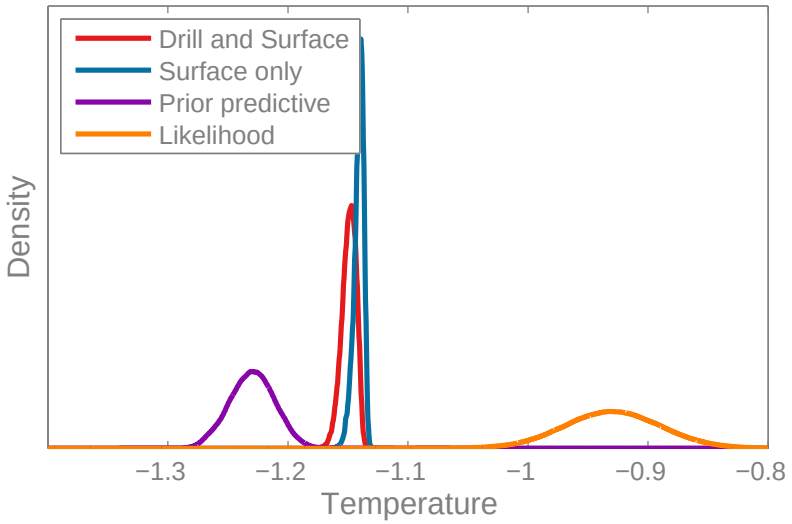


(a) The difference between the fine scale and prior mean.

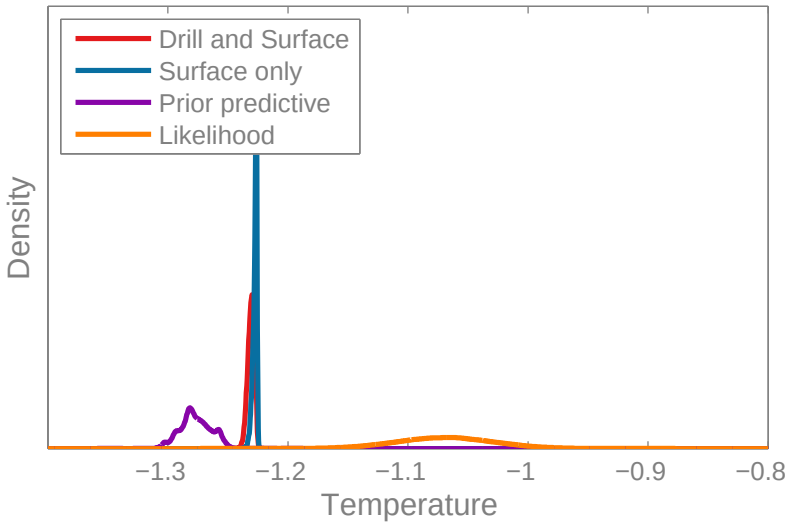


(b) The difference between the posterior mean and prior mean.

Figure 5-10: Comparison of the salinity at the top of the water column for several configurations.

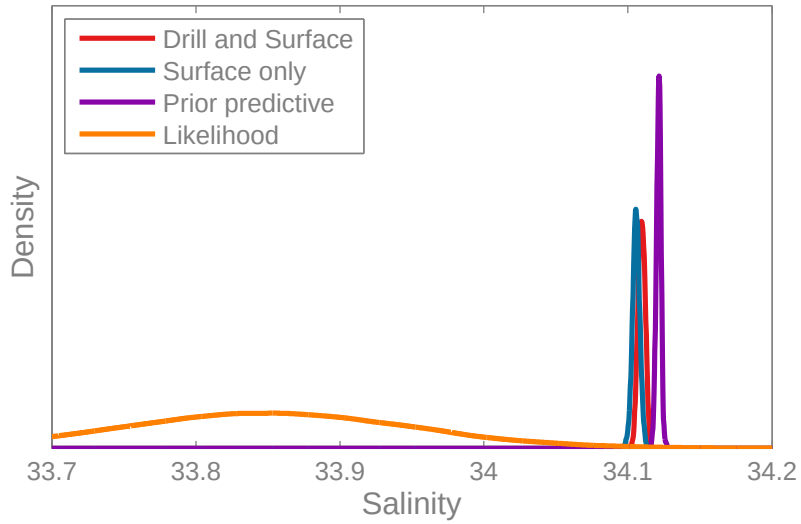


(a) Edge of the ice front.

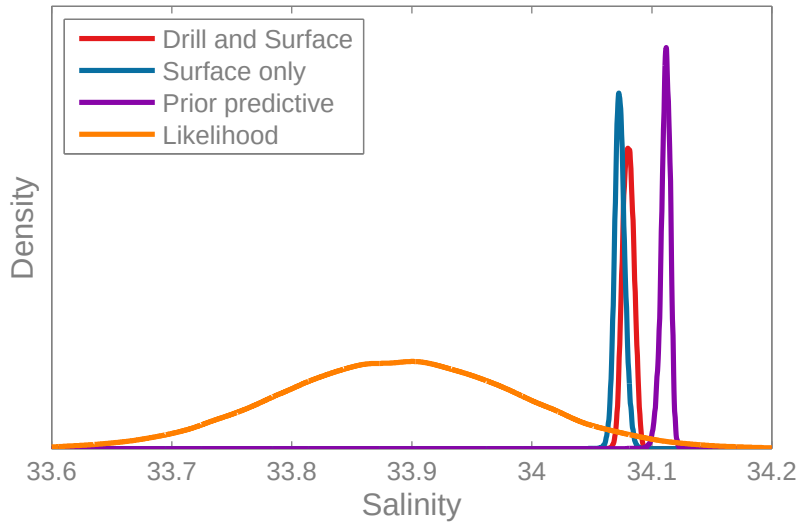


(b) Center of the ice front.

Figure 5-11: Predictive temperature distributions for two surface observation locations.

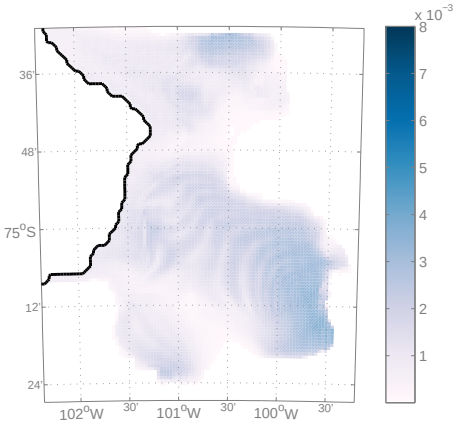


(a) Edge of the ice front.

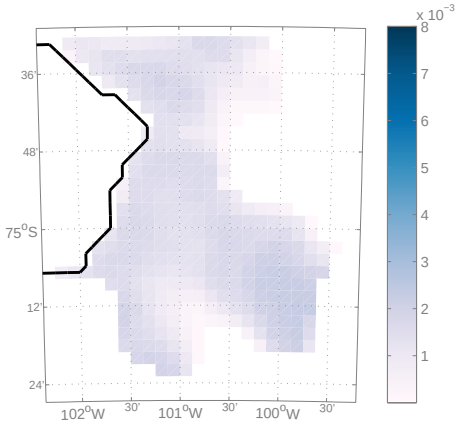


(b) Center of the ice front.

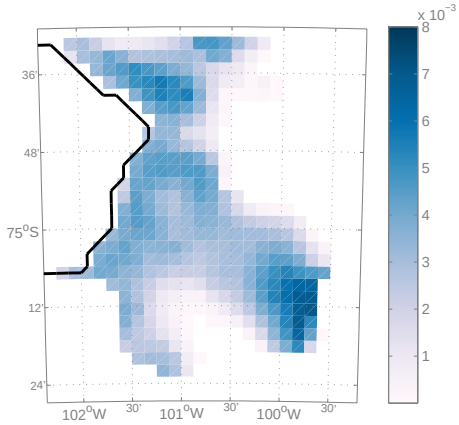
Figure 5-12: Predictive salinity distributions for two surface observation locations. The likelihood height is not normalized for visual clarity.



(a) Fine scale.



(b) Coarse scale, prior mean.



(c) Coarse scale, drilling and surface observation posterior mean.

Figure 5-13: The melt-rate at the top of the water column for several configurations.

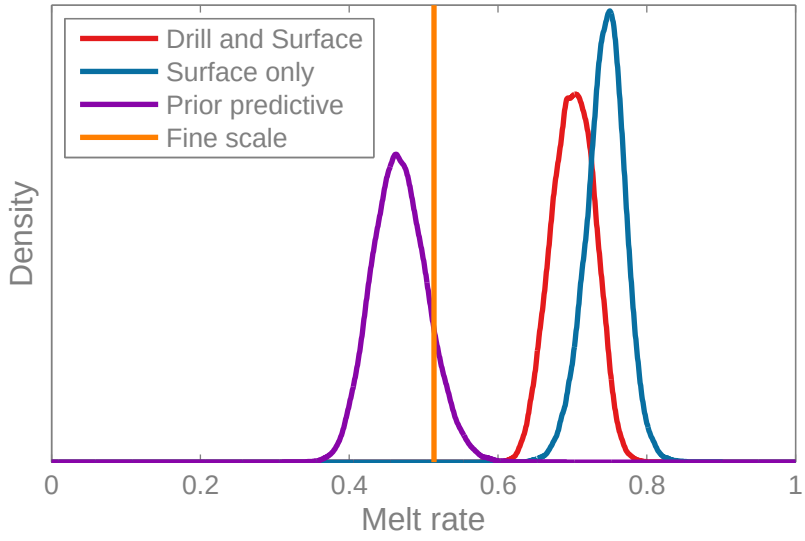


Figure 5-14: Distributions of integrated melt-rate for Pine Island Glacier.

the posterior volume is similar to that of the prior. The posterior is shifted in the transfer coefficients, as compared to the prior, which suggests a strong signal in the data. The drilling data seems to contribute to the inference, in that the two additional observation locations introduce a noticeable difference in the posterior, although the large change in the posterior suggests that the inference result is still sensitive to the choice of observations. We conclude that performing inference with such limited data would be infeasible with real data, but this work serves as an important proof of concept. Including richer data sources, such as the entire water column observations or those from autonomous submarines might improve the quality of the inference. In addition, a richer characterization of model error in the inference might help produce results that produce better consistency between the inferred parameters and melt-rate predictions and those taken from the fine-scale model used at the truth model.

Although the approximation does provide a meaningful improvement in the performance, it is not as favorable as in the previous examples. In initial exploration, we attribute this to small-scale, spurious non-smooth behavior of the model, likely stemming to numeric issues. This makes the model response appear quite challenging to approximate, and when combined with the large posterior volume, induces more refinements than seem necessary given the simplicity of the resulting posterior distribution. Future work may be able to adjust the refinement and approximation procedures to handle this behavior more efficiently.

Chapter 6

Summary and future work

6.1 Summary

Bayesian inference is an important tool for comparing complex models to real-world data, but can be quite expensive in practice when combined with state of the art computational models. Standard Markov chain Monte Carlo algorithms are not efficient in their use of model evaluations in the context of well-behaved models, as the model is called anew at every step, regardless of whether the model has already been called nearby. This work has explored adaptive techniques for constructing approximations of black-box models that are suitable for use within MCMC; constructing surrogate models allow the reuse of model evaluations, reducing the number of times the forward model must be evaluated to perform inference.

We described a theoretically sound approach to non-intrusively constructing global polynomial approximations using a Smolyak pseudospectral algorithm. We provided a clear theoretical statement about the precision of this algorithm, which provided a basis for an adaptive approach that can efficiently allocate resources to explore the important input dimensions or subspaces. This produces a widely applicable algorithm for constructing polynomial chaos expansions in practice, as demonstrated on practical examples.

To produce a theoretically convergent approximate MCMC algorithm, we turned to local approximations that can be interleaved with the exploration performed by MCMC. We proposed a framework for approximate samplers and the corresponding experimental design procedure responsible for determining when new samples of the forward model are needed and for selecting new points. We proved that a representative instance of this framework asymptotically produces samples from the exact posterior distribution. Then we performed a variety of experiments showing that the approach is robust and efficient in practice, showing orders of magnitude improvement in the run-time on sample problems. Critically, we tuned the parameters of the algorithm once and showed that they need not be adjusted even with significant changes in the problem or algorithm, which we argue is feasible because guiding the refinement process based on the acceptance probability is relatively problem- and algorithm-

invariant. For example, we successfully used approximations based on polynomials or Gaussian processes, and samplers based on DRAM or MALA. Furthermore, we extended our framework to incorporate two important tools for large-scale computation: the use of parallelism and derivative information. These techniques provide a foundation for new explorations into MCMC methods that will scale to computationally expensive forward models, which should help their wider adoption throughout the scientific community.

Finally, we demonstrated our techniques by performing inference in the complex scenario of the circulation and melting in the cavity under Pine Island Glacier. The MIT General Circulation Model is an expensive computational model that must be handled as a black box, and hence is not suitable for many standard MCMC techniques. While the inference is successful and the approximation produces significant cost savings, and is an important and successful proof of concept, we primarily conclude that a greater volume of more informative data is necessary to more tightly constrain the parameters of interest.

6.2 Future Work

While the approach to adaptivity in the Smolyak pseudospectral algorithm is deliberately simple, many extensions are possible. For instance, as described in Section 2.4.3, measures of computational cost may be added to the dimension refinement criterion. One could also use the gradient of the L^2 error indicator to identify optimal directions in the space of multi-indices along which to continue refinement, or to avoid adding all the forward neighbors of the multi-index selected for refinement.

If the virtue of local models is that they readily facilitate analysis, their pitfall is that it can be difficult to navigate the wide variety of options carrying similar theoretical guarantees. One could certainly envision using radial basis functions, wavelets, or many other families of approximations within the framework of Algorithm 3.1. Judging from the significant cost reductions exhibited in our numerical experiments, polynomials and Gaussian processes are useful options, but neither is clearly superior in all cases. Furthermore the use of local models offers the opportunity to explore the impact of other relevant inputs, including the choice of weight function, bandwidth, and variable model order, cf. [17, 46]. Much more work is needed to understand the entire "design space" of algorithms. Additionally, we have mostly neglected the cost of building local models, as this is independent of the cost of evaluating the forward model. In practice, the computational effort of constructing a local approximation (given model evaluations) is not always negligible and can increase quickly with the dimension of the parameter space.

We propose several lines of future work to remedy possible limitations of the framework for local approximation for inference. Perhaps the most pressing issue is that the local methods as described will not scale to performing inference in high dimensional spaces. It may be possible to devise sparse approximations that will scale to higher dimensions, as this is a common theme in recent algorithms for working with high di-

mensional functions. In addition, many inference problems are dominated by the prior in all but a few dimensions, which new algorithms attempt to leverage; the approximation could be constructed principally in directions where the posterior is informed by the likelihood, again reducing the number of forward model evaluations required. In some sense, this is similar to the dimension-adaptivity of the pseudospectral approximations, and might serve as a means to reintroduce it into the local approximation framework. Another issue of scaling is the use of derivative information; the current approach can only build approximations from Jacobians, hence adjoints can only be used in direct approximation. Instead, it would be more efficient if adjoints, or possibly even the action of Hessians, could be used in indirect approximations, since we have shown the indirect approach to perform much better. Additionally, further research is needed to understand the impact of noisy function evaluations or derivatives, such as non-adjoint derived Jacobians, and how to safely incorporate them.

Although the theory provides a useful guide that the algorithm behaves well, in practice, determining whether an MCMC algorithm has converged to be representative of the posterior distribution remains a challenge. We have argued that the framework we propose is robust enough that the standard types of tests for chain convergence would reveal any issues in the approximation, just as they identify when the chain is having difficulty exploring the posterior. However, deeper understanding of this behavior is still needed, and the development of practically computable error indicators, perhaps guided by the theoretical finite-sample behavior of the approximate sampler, would be beneficial.

While we have mostly discussed the local approximations as an alternative to global approximations, it is, of course, possible to create hybrids. For example, [13] suggests that it is possible to construct a fixed global approximation as a pre-processing step, then use local approximations to refine it. This type of strategy could combine the high-order convergence of global approximations with the theoretical guarantees and highly adaptive nature of local approximation.

Finally, there are many important types of MCMC kernels we have not combined with approximation strategies. For example, Hamiltonian Monte Carlo is an important derivative based algorithm, but it is not obvious how to adapt the refinement process to the multi-step proposal. Second, the parallel tempering algorithm creates a family of synthetic distributions that are jointly explored, which helps to promote mixing in multi-modal distributions. Finally, the equi-energy sampler is a member of a fascinating class of recently developed adaptive algorithms that appear to have favorable theoretical properties. This sampler revisits previous samples of comparable density to produce long-range moves, which is a reuse of samples that may integrate well with our techniques.

Our study of inference in the Pine Island setting can be improved on both the computational and scientific aspects of the inference. If the computational performance is limited by the small scale behavior of the forward model, as discussed earlier, an approximation family that is designed to be robust to noise may restore the strong results seen on simpler problems. The limited quality of the inference is independent

of the computational approach, and stems from limited data and model error. Future efforts should determine the benefits of including all available data, including full water column observations and autonomous submarine observations. Furthermore, it would be useful to understand how dense the observations must be to tightly constrain the inference and to perform experimental design to select observation locations. Attempting to reduce the model error, or else to infer it from observations, should improve the agreement between the posterior and the true model, and will help us understand the feasible performance if these methods were applied to real data.

Appendix A

Local polynomial regression

Here we provide additional detail about the polynomial regression scheme described in Section 3.2.2. We consider the quadratic case, as the linear case is a simple restriction thereof. For each component f_j of $\mathbf{f}$, the quadratic regressor is of the form

$$\tilde{f}_j(\hat{\theta}) := a_j + b_j^T \hat{\theta} + \frac{1}{2} \hat{\theta}^T H_j \hat{\theta},$$

where $a_j \in \mathbb{R}$ is a constant term, $b_j \in \mathbb{R}^d$ is a linear term, and $H_j \in \mathbb{R}^{d \times d}$ is a symmetric Hessian matrix. Note that a_j, b_j, and H_j collectively contain $M = (d+2)(d+1)/2$ independent entries for each j. The coordinates $\hat{\theta} \in \mathbb{R}^d$ are obtained by shifting and scaling the original parameters θ as follows. Recall that the local regression scheme uses N samples $\{\theta^1, \ldots, \theta^N\}$ drawn from the ball of radius R centered on the point of interest θ, along with the corresponding model evaluations $y_j^i = f_j(\theta^i)$.[1] We assume that the components of θ have already been scaled so that they are of comparable magnitudes, then define $\hat{\theta}^i = (\theta^i - \theta)/R$, so that the transformed samples are centered at zero and have maximum radius one. Writing the error bounds as in (3.1) requires this rescaling along with the $1/2$ in the form of the regressor above [19].

Next, construct the diagonal weight matrix $W = \mathrm{diag}(w^1, \ldots, w^N)$ using the sample weights in (3.2), where we have $R = 1$ because of the rescaling. Then compute the N-by-M basis matrix Φ:

$$\Phi = \begin{pmatrix} 1 & \hat{\theta}_1^1 & \cdots & \hat{\theta}_d^1 & \frac{1}{2}\left(\hat{\theta}_1^1\right)^2 & \cdots & \frac{1}{2}\left(\hat{\theta}_d^1\right)^2 & \hat{\theta}_1^1 \hat{\theta}_2^1 & \cdots & \hat{\theta}_{d-1}^1 \hat{\theta}_d^1 \\ \vdots & & & & & & & & & \vdots \\ 1 & \hat{\theta}_1^N & \cdots & \hat{\theta}_d^N & \frac{1}{2}\left(\hat{\theta}_1^N\right)^2 & \cdots & \frac{1}{2}\left(\hat{\theta}_d^N\right)^2 & \hat{\theta}_1^N \hat{\theta}_2^N & \cdots & \hat{\theta}_{d-1}^N \hat{\theta}_d^N \end{pmatrix}$$

where we ensure that $N > M$. Finally, solve the n least squares problems,

$$\Phi^T W \Phi Z = \Phi^T W Y, \tag{A.1}$$

[1] To avoid any ambiguities, this appendix departs from the rest of the narrative by using a superscript to index samples and a subscript to index coordinates.

where each column of the N-by-n matrix Y contains the samples $\left(y_j^1, \ldots, y_j^N\right)^T$, $j = 1 \ldots n$. Each column z_j of $Z \in \mathbb{R}^{M \times n}$ contains the desired regression coefficients for output j. The least squares problem may be solved stably using a thin SVD of $W\Phi Z$, which may be computed once and reused for all n least squares problems. The cross-validation fit omitting sample i simply removes row i from both sides of (A.1).

Appendix B

Genetic toggle switch inference problem

Here we provide additional details about the setup of the genetic toggle switch inference problem from Section 3.4.1. The six parameters of interest are normalized around their nominal values to have the form

$$Z_i = \bar{\theta}_i(1 + \zeta_i\theta_i),\ i = 1 \dots 6$$

so that each θ_i has prior $\mathcal{U}(-1, 1)$. The values of $\bar{\theta}_i$ are and ζ_i are given in Table B.1. The data are observed at six different values of [IPTG]; the first corresponds to the "low" state of the switch while the rest are in the "high" state. Multiple experimental observations are averaged without impacting the posterior by correspondingly lowering the noise; hence, the data comprise one observation of v/v_{ref} at each concentration, where $v_{\text{ref}} = 15.5990$. The data are modeled as having independent Gaussian errors, i.e., as draws from $\mathcal{N}(d_i, \sigma_i^2)$, where the high- and low-state observations have different standard deviations, specified in Table B.2. The forward model may be computed by integrating the ODE system (3.21), or more simply by iterating until a fixed point for v is found.

Table B.1: Normalization of the parameters in the genetic toggle switch example.

	α_1	α_2	β	γ	K	η
$\bar{\theta}_i$	156.25	15.6	2.5	1	2.0015	2.9618e-5
ζ_i	0.20	0.15	0.15	0.15	0.30	0.2

Table B.2: Data and obervation error variances for the likelihood of the genetic toggle switch example.

[IPTG]	156.25	15.6	2.5	1	2.0015	2.9618e-5
d_i	0.00798491	1.07691684	1.05514201	0.95429837	1.02147051	1.0
σ_i	4.0e-5	0.005	0.005	0.005	0.005	0.005

Bibliography

[1] R. J. Adler, *The geometry of random fields*, SIAM, 1981.

[2] C. Andrieu and J. Thoms, *A tutorial on adaptive MCMC*, Statistics and Computing, 18 (2008), pp. 343–373.

[3] C. G. Atkeson, A. W. Moore, and S. Schaal, *Locally Weighted Learning*, Artificial Intelligence Review, 11 (1997), pp. 11–73.

[4] I. Babuska, F. Nobile, and R. Tempone, *A stochastic collocation method for elliptic partial differential equations with random input data*, SIAM Journal on Numerical Analysis, 45 (2007), pp. 1005–1034.

[5] G. Bal, I. Langmore, and Y. M. Marzouk, *Bayesian Inverse Problems with Monte Carlo Forward Models*, Inverse problems and imaging, 7 (2013), pp. 81–105.

[6] V. Barthelmann, E. Novak, and K. Ritter, *High dimensional polynomial interpolation on sparse grids*, Advances in Computational Mathematics, 12 (2000), pp. 273–288.

[7] N. Bliznyuk, D. Ruppert, C. Shoemaker, R. Regis, S. Wild, and P. Mugunthan, *Bayesian Calibration and Uncertainty Analysis for Computationally Expensive Models Using Optimization and Radial Basis Function Approximation*, Journal of Computational and Graphical Statistics, 17 (2008), pp. 270–294.

[8] N. Bliznyuk, D. Ruppert, and C. A. Shoemaker, *Local Derivative-Free Approximation of Computationally Expensive Posterior Densities*, Journal of Computational and Graphical Statistics, 21 (2012), pp. 476–495.

[9] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*, Dover Publications, 2nd revise ed., 2001.

[10] S. P. Brooks and G. O. Roberts, *Assessing Convergence of Markov Chain Monte Carlo Algorithms*, Statistics and Computing, 8 (1998), pp. 319–335.

[11] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zhang Jr., *Spectral Methods: Fundamentals in single domains*, Springer Berlin Heidelberg, 2006.

[12] O. Cappe, A. Guillin, J.-M. Marin, and C. P. Robert, *Population Monte Carlo*, Journal of Computational and Graphical Statistics, 13 (2004), pp. 907–929.

[13] A. Chakraborty, B. K. Mallick, R. G. Mcclarren, C. C. Kuranz, D. Bingham, M. J. Grosskopf, E. M. Rutter, H. F. Stripling, and R. P. Drake, *Spline-Based Emulators for Radiative Shock Experiments With Measurement Error*, Journal of the American Statistical Association, 108 (2013), pp. 411–428.

[14] J. A. Christen and C. Fox, *Markov chain Monte Carlo Using an Approximation*, Journal of Computational and Graphical Statistics, 14 (2005), pp. 795–810.

[15] C. Clenshaw and A. Curtis, *A method for numerical integration on an automatic computer*, Numerische Mathematik, 2 (1960), pp. 197–205.

[16] W. S. Cleveland, *Robust Locally Weighted Regression and Smoothing Scatterplots*, Journal of the American Statistical Association, 74 (1979), pp. 829–836.

[17] W. S. Cleveland and C. Loader, *Smoothing by local regression: Principles and methods*, in Statistical Theory and Computational Aspects of Smoothing, W. Haerdle and M. G. Schimek, eds., vol. 1049, Springer, New York, 1996, pp. 10–49.

[18] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust Region Methods*, SIAM, 2000.

[19] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*, SIAM, 2009.

[20] P. R. Conrad and Y. M. Marzouk, *Adaptive Smolyak Pseudospectral Approximations*, SIAM Journal of Scientific Computing, 35 (2013), pp. A2643–2670.

[21] P. R. Conrad, Y. M. Marzouk, N. S. Pillai, and A. Smith, *Asymptotically Exact MCMC Algorithms via Local Approximations of Computationally Intensive Models*, Arxiv preprint arXiv:1402.1694v1, (2014), pp. 1–38.

[22] P. G. Constantine, M. S. Eldred, and E. T. Phipps, *Sparse Pseudospectral Approximation Method*, Computer Methods in Applied Mechanics and Engineering, 229-232 (2012), pp. 1–12.

[23] S. L. Cotter, M. Dashti, and A. M. Stuart, *Approximation of Bayesian Inverse Problems*, SIAM Journal of Numerical Analysis, 48 (2010), pp. 322–345.

[24] S. L. Cotter, G. O. Roberts, A. M. Stuart, and D. White, *MCMC methods for functions: Modifying old algorithms to make them faster*, Statistical Science, 28 (2013), pp. 424–446.

[25] M. K. Cowles and B. P. Carlin, *Markov Chain Monte Carlo Convergence Diagnostics : A Comparative Review*, Journal of the American Statistical Association, 91 (1996), pp. 883–904.

[26] R. V. Craiu, J. Rosenthal, and C. Yang, *Learn From Thy Neighbor : Parallel-Chain and Regional Adaptive MCMC*, Journal of the American Statistical Association, 104 (2009), pp. 1454–1466.

[27] N. Cressie, *Statistics for Spatial Data*, John Wiley and Sons, 1991.

[28] T. Cui, C. Fox, and M. J. O'Sullivan, *Bayesian calibration of a large-scale geothermal reservoir model by a new adaptive delayed acceptance Metropolis Hastings algorithm*, Water Resources Research, 47 (2011), p. W10521.

[29] V. Dansereau, P. Heimbach, and M. Losch, *Simulation of sub-ice shelf melt rates in a general circulation model: velocity-dependent transfer and the role of friction*, Journal of Geophysical Research, in review, (2013).

[30] M. Dashti and A. Stuart, *Uncertainty Quantification and Weak Approximation of an Elliptic Inverse Problem*, SIAM Journal of Numerical Analysis, 49 (2011), pp. 2524–2542.

[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] M. S. Eldred and J. Burkardt, *Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification*, AIAA paper 2009–0976, (2009).

[33] O. G. Ernst, A. Mugler, H.-J. Starkloff, and E. Ullmann, *On the convergence of generalized polynomial chaos expansions*, ESAIM: Mathematical Modeling and Numerical Analysis, 46 (2012), pp. 317–339.

[34] M. Fielding, D. J. Nott, and S.-Y. Liong, *Efficient MCMC Schemes for Computationally Expensive Posterior Distributions*, Technometrics, 53 (2011), pp. 16–28.

[35] A. I. Forrester, A. Sóbester, and A. J. Keane, *Multi-fidelity optimization via surrogate modelling*, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 463 (2007), pp. 3251–3269.

[36] M. Frangos, Y. Marzouk, K. Willcox, and B. van Bloemen Waanders, *Surrogate and Reduced-Order Modeling: A Comparison of Approaches for Large-Scale Statistical Inverse Problems, Biegler, Lorenz et al.*, John Wiley and Sons, 2010.

[37] J. Friedman, *Multivariate adaptive regression splines*, The Annals of Statistics, 19 (1991), pp. 1–141.

[38] W. Gander, *Change of basis in polynomial interpolation*, Numerical Linear Algebra with Applications, 12 (2005), pp. 769–778.

[39] T. S. Gardner, C. R. Cantor, and J. J. Collins, *Construction of a genetic toggle switch in Escherichia coli.*, Nature, 403 (2000), pp. 339–42.

[40] A. Genz, *Testing Multidimensional Integration Routines*, in Tools, Methods, and Languages for Scientific and Engineering Computation, B. Ford, J. Rault, and F. Thomasset, eds., North-Holland, 1984, pp. 81–94.

[41] ——, *A Package for Testing Multiple Integration Subroutines*, in Numerical Integration: Recent Developments, Software and Applications, P. Keast and G. Fairweather, eds., D Reidel, 1987, pp. 337–340.

[42] ——, *Fully symmetric interpolatory rules for multiple integrals over infinite regions with Gaussian weight*, Journal of Computational and Applied Mathematics, 71 (1996), pp. 299–309.

[43] T. Gerstner and M. Griebel, *Dimension-Adaptive Tensor-Product Quadrature*, Computing, 71 (2003), pp. 65–87.

[44] R. Ghanem and P. Spanos, *Stochastic finite elements: a spectral approach*, Springer, New York, 1991.

[45] M. Girolami and B. Calderhead, *Riemann manifold Langevin and Hamiltonian Monte Carlo methods*, Journal Royal Statistical Society B, 73 (2011), pp. 1–37.

[46] R. B. Gramacy and D. W. Apley, *Local Gaussian process approximation for large computer experiments*, arXiv preprint, (2013), pp. 1–27.

[47] R. B. Gramacy and H. K. H. Lee, *Bayesian Treed Gaussian Process Models with an Application to Computer Modeling*, Journal of the American Statistical Association, 103 (2008), pp. 1119–1130.

[48] Z. Q. Gregory P. Smith, David M. Golden, Michael Frenklach, Nigel W. Moriarty, Boris Eiteneer, Mikhail Goldenberg, C. Thomas Bowman, Ronald K. Hanson, Soonho Song, William C. Gardiner, Jr., Vitali V. Lissianski, *GRI-MECH 3.0.* `http://www.me.berkeley.edu/gri_mech`.

[49] B. Haaland and P. Z. G. Qian, *Accurate emulators for large-scale computer experiments*, Annals of Statistics, 39 (2011), pp. 2974–3002.

[50] H. Haario, M. Laine, A. Mira, and E. Saksman, *DRAM: efficient adaptive MCMC*, Statistics and Computing, 16 (2006), pp. 339–354.

[51] T. Hastie and C. Loader, *Local Regression: Automatic Kernel Carpentry*, Statistical Science, 8 (1993), pp. 120–143.

[52] M. Hegland, *Adaptive sparse grids*, ANZIAM J, 44 (2003), pp. 335–353.

[53] P. Heimbach and M. Losch, *Adjoint sensitivities of sub-ice shelf melt rates to ocean circulation under Pine Island Ice Shelf, West Antarctica*, Annals of Glaciology, 54 (2010), pp. 59–69.

[54] J. S. Hesthaven, S. Gottlieb, and D. Gottlieb, *Spectral Methods for Time-Dependent Problems*, Cambridge University Press, 2007.

[55] D. Higdon, J. Gattiker, B. Williams, and M. Rightley, *Computer Model Calibration Using High-Dimensional Output*, Journal of the American Statistical Association, 103 (2008), pp. 570–583.

[56] D. Higdon, H. Lee, and C. Holloman, *Markov chain Monte Carlo-based approaches for inference in computationally intensive inverse problems*, in Bayesian Statistics 7, Oxford University Press, 2003, pp. 181–197.

[57] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward, *SUNDIALS : Suite of Nonlinear and Differential / Algebraic Equation Solvers*, ACM Transactions on Mathematical Software, 31 (2005), pp. 363–396.

[58] M. D. Hoffman and A. Gelman, *The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo*, arXiv preprint arXiv:1111.4246, (2011).

[59] D. M. Holland and A. Jenkins, *Modelling Thermodynamic Ice - Ocean Interactions at the Base of an Ice Shelf*, Journal of Physical Oceanography, 29 (1999), pp. 1787–1800.

[60] X. Huan and Y. M. Marzouk, *Simulation-based optimal Bayesian experimental design for nonlinear systems*, Journal of Computational Physics, 232 (2013), pp. 288–317.

[61] S. S. Jacobs, A. Jenkins, C. F. Giulivi, and P. Dutrieux, *Stronger ocean circulation and increased melting under Pine Island Glacier ice shelf*, Nature Geoscience, 4 (2011), pp. 1–5.

[62] A. Jenkins, P. Dutrieux, S. S. Jacobs, S. D. McPhail, J. R. Perrett, A. T. Webb, and D. White, *Observations beneath Pine Island Glacier in WestÂ Antarctica and implications for its retreat*, Nature Geoscience, 3 (2010), pp. 468–472.

[63] V. R. Joseph, *Bayesian Computation Using Design of Experiments-Based Interpolation Technique*, Technometrics, 54 (2012), pp. 209–225.

[64] ——, *A Note on Nonnegative DoIt Approximation*, Technometrics, 55 (2013), pp. 103–107.

[65] I. Joughin, B. E. Smith, and D. M. Holland, *Sensitivity of 21st century sea level to ocean-induced thinning of Pine Island Glacier, Antarctica*, Geophysical Research Letters, 37 (2010), pp. 1–5.

[66] J. Kaipio and E. Somersalo, *Statistical inverse problems: Discretization, model reduction and inverse crimes*, Journal of Computational and Applied Mathematics, 198 (2007), pp. 493–504.

[67] J. P. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, Springer New York, 2005.

[68] C. G. Kaufman, D. Bingham, S. Habib, K. Heitmann, and J. a. Frieman, *Efficient emulators of computer experiments using compactly supported correlation functions, with an application to cosmology*, The Annals of Applied Statistics, 5 (2011), pp. 2470–2492.

[69] R. J. Kee, M. E. Coltrin, and P. Glarborg, *Chemically Reacting Flow: Theory and Practice*, Wiley-Interscience, 2003.

[70] M. Kennedy and A. O'Hagan, *Bayesian calibration of computer models*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63 (2001), pp. 425–464.

[71] A. Klimke and B. Wohlmuth, *Algorithm 847: Spinterp: piecewise multilinear hierarchical sparse grid interpolation in MATLAB*, ACM Transactions on Mathematical Software (TOMS), 31 (2005), pp. 561–579.

[72] A. Korattikara, Y. Chen, and M. Welling, *Austerity in MCMC Land: Cutting the Metropolis-Hastings Budget*, arXiv preprint, (2013), pp. 1–13.

[73] O. P. Le Maitre and O. M. Knio, *Spectral Methods for Uncertainty Quantification*, Springer, 2010.

[74] C. Lieberman, K. Willcox, and O. Ghattas, *Parameter and State Model Reduction for Large-Scale Statistical Inverse Problems*, SIAM Journal on Scientific Computing, 32 (2010), pp. 2523–2542.

[75] S. N. Lophaven, H. B. Nielsen, and J. Sondergaard, *Aspects of the MATLAB Toolbox DACE*, tech. report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2002.

[76] K. D. Mankoff, S. S. Jacobs, S. M. Tulaczyk, and S. E. Stammerjohn, *The role of Pine Island Glacier ice shelf basal channels in deep-water upwelling, polynyas and ocean circulation in Pine Island Bay, Antarctica*, Annals of Glaciology, 53 (2012), pp. 123–128.

[77] J. Marotzke, R. Giering, K. Q. Zhang, D. Stammer, C. Hill, and T. Lee, *Construction of the adjoint MIT ocean general circulation model and application to Atlantic heat transport sensitivity*, Journal of Geophysical Research, 104 (1999), pp. 29529–29547.

[78] J. Martin, L. C. 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 Computing, 34 (2012), pp. 1460–1487.

[79] Y. Marzouk and D. Xiu, *A stochastic collocation approach to Bayesian inference in inverse problems*, Communications in Computational Physics, 6 (2009), pp. 826–847.

[80] Y. M. Marzouk, H. N. Najm, and L. A. Rahn, *Stochastic spectral methods for efficient Bayesian solution of inverse problems*, Journal of Computational Physics, 224 (2007), pp. 560–586.

[81] R. M. Neal, *MCMC using Hamiltonian dynamics*, in Handbook of Markov Chain Monte Carlo, S. Brooks, A. Gelman, G. Jones, and Meng X.-L., eds., Chapman & Hall/CRC, 2010.

[82] N. Nguyen, G. Rozza, D. Huynh, and A. Patera, *Reduced basis approximation and a posteriori error estimation for parametrized parabolic PDEs; Application to real-time Bayesian parameter estimation*, in Large-Scale Inverse Problems and Quantification of Uncertainty, L. Biegler, G. Biros, O. Ghattas, M. Heinkenschloss, D. Keyes, B. Mallick, Y. Marzouk, L. Tenorio, B. van Bloemen Waanders, and K. Wilcox, eds., John Wiley & Sons, Ltd, 2011.

[83] F. Nobile, R. Tempone, and C. G. Webster, *A Sparse Grid Stochastic Collocation Method for Partial Differential Equations with Random Input Data*, SIAM Journal on Numerical Analysis, 46 (2007), p. 2309.

[84] E. Novak and K. Ritter, *High dimensional integration of smooth functions over cubes*, Numerische Mathematik, 75 (1996), pp. 79–97.

[85] ——, *Simple Cubature Formulas with High Polynomial Exactness*, Constructive Approximation, 15 (1999), pp. 499–522.

[86] T. N. L. Patterson, *The Optimum Addition of Points to Quadrature Formulae*, Mathematics of Computation, 22 (1968), p. 847.

[87] A. J. Payne, P. R. Holland, A. P. Shepherd, I. C. Rutt, A. Jenkins, and I. Joughin, *Numerical modeling of ocean-ice interactions under Pine Island Bay's ice shelf*, Journal of Geophysical Research, 112 (2007), p. C10019.

[88] N. Pillai and A. Smith, *Ergodicity of approximate MCMC chains with applications to large data sets*, arXiv preprint, (2014).

[89] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. Kevin Tucker, *Surrogate-based analysis and optimization*, Progress in Aerospace Sciences, 41 (2005), pp. 1–28.

[90] J. O. Ramsay, G. Hooker, D. Campbell, and J. Cao, *Parameter Estimation for Differential Equations: A Generalized Smoothing Approach*, Journal of the Royal Statistical Society. Series B (Methodological), 69 (2007), pp. 741–796.

[91] C. E. Rasmussen, *Gaussian Processes to Speed up Hybrid Monte Carlo for Expensive Bayesian Integrals*, in Bayesian Statistics 7, Oxford University Press, 2003, pp. 651–659.

[92] G. Roberts and O. Stramer, *Langevin diffusions and Metropolis-Hastings algorithms*, Methodology and computing in applied probability, 4 (2002), pp. 337–357.

[93] 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.

[94] G. O. Roberts and R. L. Tweedie, *Geometric convergence and central limit theorems for multidimensional hastings and metropolis algorithms*, Biometrika, 83 (1996), pp. 95–110.

[95] J. Rosenthal, *Minorization conditions and convergence rates for Markov chain Monte Carlo*, JASA, 90 (1995), pp. 558–566.

[96] J. S. Rosenthal, *Parallel computing and Monte Carlo algorithms*, Far east journal of theoretical statistics, 4 (2000), pp. 207–236.

[97] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, *Design and analysis of computer experiments*, Statistical Science, 4 (1989), pp. 409–423.

[98] C. Safta, H. Najm, and O. Knio, *TChem*. `http://www.sandia.gov/tchem/`.

[99] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments*, Springer, New York, 2003.

[100] W. Sickel and T. Ullrich, *The Smolyak algorithm, sampling on sparse grids and function spaces of dominating mixed smoothness*, East J Approx, 13 (2007), pp. 387–425.

[101] ——, *Tensor products of Sobolev-Besov spaces and applications to approximation from the hyperbolic cross*, Journal of Approximation Theory, 161 (2009), pp. 748–786.

[102] S. A. Smolyak, *Quadrature and interpolation formulas for tensor products of certain classes of functions*, Dokl. Akad. nauk SSSR, 4 (1963), pp. 240–243.

[103] E. Snelson and Z. Ghahramani, *Local and global sparse Gaussian process approximations*, in Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS-07), 2007.

[104] C. Soize and R. Ghanem, *Physical Systems with Random Uncertainties: Chaos representations with arbitrary probability measure*, SIAM Journal of Scientific Computing„ 26 (2004), pp. 395–410.

[105] T. P. Stanton, W. J. Shaw, M. Truffer, H. F. J. Corr, L. E. Peters, K. L. Riverman, R. Bindschadler, D. M. Holland, and S. Anandakrishnan, *Channelized ice melting in the ocean boundary layer beneath Pine Island Glacier, Antarctica.*, Science (New York, N.Y.), 341 (2013), pp. 1236–9.

[106] M. L. Stein, Z. Chi, and L. J. Welty, *Approximating likelihoods for large spatial data sets*, Journal of the Royal Statistical Society. Series B (Methodological), 66 (2004), pp. 275–296.

[107] B. Sudret, *Global sensitivity analysis using polynomial chaos expansions*, Reliability Engineering & System Safety, 93 (2008), pp. 964–979.

[108] L. N. Trefethen, *Is Gauss Quadrature Better than Clenshaw-Curtis?*, SIAM Review, 50 (2008), p. 67.

[109] A. V. Vecchia, *Estimation and Model Identification for Continuous Spatial Processes*, Journal of the Royal Statistical Society. Series B (Methodological), 50 (1988), pp. 297–312.

[110] S. Vijayakumar, A. D'Souza, and S. Schaal, *Incremental online learning in high dimensions.*, Neural computation, 17 (2005), pp. 2602–2634.

[111] G. W. Wasilkowski and H. Wozniakowski, *Explicit cost bounds of algorithms for multivariate tensor product problems*, Journal of Complexity, 11 (1995), pp. 1–56.

[112] ——, *Weighted Tensor Product Algorithms for Linear Multivariate Problems*, Journal of Complexity, 15 (1999), pp. 402–447.

[113] ——, *Polynomial-Time Algorithms for Multivariate Linear Problems with Finite-Order Weights: Worst Case Setting*, Foundations of Computational Mathematics, 5 (2005), pp. 451–491.

[114] N. Wiener, *The Homogeneous Chaos*, American Journal of Mathematics, 60 (1938), pp. 897–936.

[115] D. Xiu, *Efficient Collocational Approach for Parametric Uncertainty Analysis*, Communications in Computational Physics, 2 (2007), pp. 293–309.

[116] ——, *Fast numerical methods for stochastic computations: a review*, Communications in computational physics, 5 (2009), pp. 242–272.

[117] ——, *Numerical Methods for Stochastic Computations: A Spectral Method Approach*, Princeton University Press, 2010.

[118] D. Xiu and J. S. Hesthaven, *High-Order Collocation Methods for Differential Equations with Random Inputs*, SIAM Journal on Scientific Computing, 27 (2005), p. 1118.

[119] D. Xiu and G. E. Karniadakis, *The Wiener–Askey Polynomial Chaos for Stochastic Differential Equations*, SIAM Journal on Scientific Computing, 24 (2002), p. 619.