A Multiscale Framework for Bayesian Inference in Elliptic Problems

by

Matthew David Parno

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

Submitted to the School of Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Computation for Design and Optimization

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2011

© Massachusetts Institute of Technology 2011. All rights reserved.

Author .................................................................

School of Engineering
May 14, 2011

Certified by .............................................................

Youssef Marzouk
Boeing Assistant Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by .............................................................

Nicolas Hadjiconstantinou
Associate Professor of Mechanical Engineering
Director, Computation for Design and Optimization
A Multiscale Framework for Bayesian Inference in Elliptic Problems

by

Matthew David Parno

Submitted to the School of Engineering on May 14, 2011, in partial fulfillment of the requirements for the degree of Master of Science in Computation for Design and Optimization

Abstract

The Bayesian approach to inference problems provides a systematic way of updating prior knowledge with data. A likelihood function involving a forward model of the problem is used to incorporate data into a posterior distribution. The standard method of sampling this distribution is Markov chain Monte Carlo which can become inefficient in high dimensions, wasting many evaluations of the likelihood function. In many applications the likelihood function involves the solution of a partial differential equation so the large number of evaluations required by Markov chain Monte Carlo can quickly become computationally intractable.

This work aims to reduce the computational cost of sampling the posterior by introducing a multiscale framework for inference problems involving elliptic forward problems. Through the construction of a low dimensional prior on a coarse scale and the use of iterative conditioning technique the scales are decouples and efficient inference can proceed. This work considers nonlinear mappings from a fine scale to a coarse scale based on the Multiscale Finite Element Method. Permeability characterization is the primary focus but a discussion of other applications is also provided. After some theoretical justification, several test problems are shown that demonstrate the efficiency of the multiscale framework.

Thesis Supervisor: Youssef Marzouk
Title: Boeing Assistant Professor of Aeronautics and Astronautics
Acknowledgments

I would not be where I am without Julie, who helps me keep things in perspective and maintain some sort of balance between work and play. I would also like to acknowledge the support of my parents; who have helped me stay on track and have pointed me towards quotes such as “Make friends with pain and you’ll never be alone,” from Ken Chlouber (founder of the Leadville 100) when I needed to dig deep. Of course my dad was talking about running the LeanHorse 100, but at times I think the same philosophy is useful in grad school.

Additionally I would like to thank Sandia National Laboratories and the Office of Science Graduate fellowship for generous funding of this work.
2 Multiscale Background 46

2.1 Multiscale Simulation Methods ........................................... 48
  2.1.1 Model Elliptic Equation ................................................. 48
  2.1.2 Upscaling ................................................................. 52
  2.1.3 Homogenization .......................................................... 55
  2.1.4 Variational Methods ..................................................... 58
  2.1.5 Multiscale Finite Element Methods ................................. 59

2.2 Multiscale Sampling methods ............................................. 63
  2.2.1 Coupled Metropolis Sampling ......................................... 63
  2.2.2 Proposal Filtering with MsFEM ....................................... 65

3 Multiscale inference framework 67

3.1 Conditional Independence .................................................. 67
3.2 Multiscale MCMC formulation ........................................... 70
3.3 Upscaling the Prior ......................................................... 71
  3.3.1 Proof of positive definiteness in one dimension ............... 77
3.4 Iterative Nonlinear Conditioning ....................................... 79
  3.4.1 Linear conditioning ..................................................... 80
  3.4.2 Nonlinear conditioning ............................................... 82
  3.4.3 MCMC along a constraint ............................................ 89
3.5 Multiscale framework overview ......................................... 91
3.6 Implementation details .................................................... 92
  3.6.1 Jacobian in One Dimension ......................................... 92
  3.6.2 Other notes ............................................................... 93
3.7 Generalization to other multiscale settings .......................... 93

4 Numerical Results 94

4.1 Test Cases ................................................................. 94
  4.1.1 Problem Description .................................................... 94
  4.1.2 Verification with single scale inference results ............... 96
  4.1.3 Draw from Exponential Prior ....................................... 100
4.1.4 First layer of SPE10 ........................................... 102
4.1.5 Fifth layer of SPE10 .......................................... 103
4.1.6 Summary of test cases ....................................... 104

5 Future work and conclusions .................................. 106
  5.0.7 Future work ................................................ 106
  5.0.8 Use in dynamic problems ................................ 106
  5.0.9 Spatial decoupling during iterative conditioning .... 107
5.1 Finale ............................................................. 107

A Philosophical Asides ........................................... 109
  A.1 Epistemic and Aleatory Uncertainty ..................... 109
  A.2 Bayesian vs. Frequentist .................................. 110
List of Figures

1-1 The danger of simply using a point estimate for the calibrated parameters. In the blue, noise in the data will not result in a large error in the calibrated model. However, the same amount of noise in the red could result in a calibrated parameter somewhere in the range of $[-1.6, 1.6]$. A Bayesian approach allows this uncertainty to be captured and presented as part of the final inference result. 21

1-2 The likelihood function constrains the likely regions of the parameter space. When the error is large, i.e. $\gamma$ is large for $\Sigma_{ee} = \gamma I$, the likelihood functions does not provide much new information and the posterior will be largely governed by the prior. 25

1-3 Illustration of rejection sampling. Samples of the green distribution are generated by sampling from a distribution proportional to the red function (Gaussian in this case) and accepting with the probability defined in (1.17). 28

1-4 Example of a three state discrete Markov chain with transition probabilities denoted by $p_{ij}$. If you squint hard enough you can imagine a similar system in, $\mathbb{R}^m$, where we have an infinite number of states. 31

1-5 Example of Metropolis-Hastings MCMC sampling a simple Gaussian mixture in 2 dimensions. Shown is the MCMC chain after 50 iterations, the density estimate after 10000 steps, and the chain after 2000 steps. 33

1-6 Example of Metropolis-Hastings samples for increasing proposal variance. The contours show the true distribution, which has a mean of $(0, -2)$. 37
1-7 Example of Metropolis-Hastings chains for increasing proposal variance. Blue is the x component of the chain and green is the y component. For small proposals, high acceptance rates exist but the jumps are small and the chain inefficiently explores the space (the meandering nature in the first plot). However, for too large a proposal, the frequent plateaus indicate that the acceptance probability is low.

1-8 Comparison of the DRAM family of adaptive algorithms. Blue is the x component of the chain and green is the y component. The initial proposal size was set to $C_0 = 2I$. Three DR stages were used, each shrinking the previous proposal by a factor of 4. The covariance was adapted with AM every 20 steps.

2-1 Example of groundwater observations with multiscale behavior. Not only are yearly patterns present, but daily length scales exist as well. Data comes from USGS well measurements in Bingham County, Idaho.

2-2 Conservation of mass in a porous media. When coupled with Darcy’s equation as a constitutive law, this gives the familiar pressure equation.

2-3 Example of multiscale field and corresponding pressures. The one dimensional case has a dirichlet condition on the left and a homogeneous Neumann condition on the right. In the two dimensional case, all boundaries are fixed with Dirichlet conditions. Clearly, the pressure fields are much smoother than the permeability fields. The one dimensional case shows the smoothing effect of the elliptic operator and the two dimensional setting again shows the relative smoothness of the pressure field compared to the permeability field.

2-4 Example of geometric, arithmetic, and harmonic averaging. After inspection, note that the geometric means visually seems to be between the arithmetic and harmonic means, as required by the Wiener bounds. This field is $\log_{10}(K)$ for the 10th layer of the popular SPE10 dataset.
2-5 Comparison of Homogenization process and upscaling process. While
upscaling builds a coarse permeability and uses the same governing
equation, homogenization finds a homogenized permeability by putting
constraints on the coarse operator. ................................. 56

2-6 Illustration of MsFEM Basis function. This illustrates the basis func-
tion evaluated over one coarse element. ............................. 61

2-7 Illustration of the Multiscale Metropolis MCMC method. Here, steps
labeled with MCMC correspond to steps with standard single scale
proposals. It is also possible to have more than one MCMC step before
a swap is performed. .................................................. 64

3-1 Validation of conditional independence assumption. ................. 69

3-2 Illustration of full multiscale inference using procedure using MsFEM. 71

3-3 Quantile-Quantile plot comparing sample quantiles to true log-normal
quantiles. The distribution is an exact fit when the scatter points are
exactly linear. .......................................................... 74

3-4 Distribution of log(e) based on Monte-Carlo sampling of a piecewise
constant log-normal $k$ field with exponential prior. .................. 75

3-5 Distribution of log(e) based on Monte-Carlo sampling of a piecewise
constant log-normal $k$ field with Gaussian prior. .................. 76

3-6 Mapping from $\pi(Y)$ to $\pi(\log(e))$ for exponential and Gaussian covari-
ance kernels in $\pi(Y)$. The fine scale correlation length, $L_f$, has been
scaled by the coarse element size $h$ to allow for arbitrary discretiza-
tions. Notice the $x$ axis of the power plots is in the opposite direction
to the other plots. .................................................. 78

3-7 Illustration of $\pi(k|A)$ as a constraint on the prior. We wish to sample
from the “slice” of $\pi(k)$ along the constraint. ....................... 80

3-8 Comparison of high order ODE integration with SCKF for sampling
from a constraint. The densities are plotted over $x$. .................. 86
3-9 Illustration of integration paths for nonlinear conditioning and linear conditioning ........................................ 88

3-10 Illustration of conditioned proposals in two dimensions on a cubic constraint. In both cases, $\lambda = 0.01.$ ........................................ 91

4-1 Location of pressure observations and posterior covariance comparison using the fully multiscale approach, single scale with FEM forward solver, and single scale with MsFEM forward simulation. ............... 97

4-2 Comparison of posterior mean and variance between standard single scale approaches and the multiscale framework. The left plot shows the posterior means $\pm 2\sigma^2$ ........................................ 98

4-3 Results for multiscale inference using a draw from the prior. The prior kernel is defined by $\rho = 1, \ d = 0.2, \ \sigma = 1.$ Twenty fine elements were used in each coarse element and 5 coarse elements were used. ............ 101

4-4 Results for multiscale inference using a slice of layer 1 of the SPE 10 dataset. The chain was downsampled by 50 before iterative conditioning. 103

4-5 Results for multiscale inference using a slice of layer 5 of the SPE 10 dataset. The chain was downsampled by 50 before iterative conditioning. 105
List of Tables

1.1 Example of Metropolis-Hastings MCMC performance on two dimensional Banana function. Proposal is diagonal Gaussian. 100 trials with 1000 Samples after a burn in period of 200 steps was used. . . . . . . 36

3.1 Comparison of acceptance rates for MCMC on cubic constraint with $\epsilon = 1e - 2$. The bold rates are the combinations that result in the smallest density reconstruction error from 500 nonlinear conditioning runs and 500 MCMC steps. . . . . . . . . . . . . . . . . . . . . 91

4.1 Comparison of single scale performance with multiscale performance. $ESS$ is the effective sample size, so $ESS/N$ is the effective number of samples related to the total number of fine scale samples generated and $ESS/S$ is the number of effective samples per second. . . . . . . 99

4.2 Summary of multiscale inference tests. $K_c$ is the number of coarse elements, $K_f$ is the number of fine elements in each coarse element. $d$ is the correlation length of the prior, $p$ is the prior power, $\sigma$ is the prior variance, $\mu_Y$ is the prior mean, $N_d$ is the number of data points. . . . 100

4.3 Summary of multiscale inference performance. $ESS$ is the effective sample size, so $ESS/N$ is the effective number of samples related to the total number of fine scale samples generated and $ESS/S$ is the number of effective samples per second. Additionally, acceptance is the acceptance rate of the coarse MCMC chain. . . . . . . . . . . . . . . . . 104
0.1 Introduction

Humans have been trying to understand the world around us for millennia. Take for example the Babylonian mathematicians around 1700 BCE, who put together a tablet of Pythagorean triples [1]. Obviously, we are now far beyond the computing power of a few mathematicians with clay tablets and a base 60 number system. With the advent of computers, our predictive capabilities have increased dramatically. We can now perform simulations on huge supercomputers or large distributed systems to simulate large fusion reactors [74] or protein folding in a cell [6]. Advanced simulations like these and even much less sophisticated mathematical models serve two purposes: the model allows scientists and engineers to make predictions about how a system behaves and the model gives insight into why the system behaves as it does. For example, modeling glacier dynamics is important not only for predicting human influence on the environment [13] but also to help researchers answer fundamental questions about why glaciers behave the way they do [57]. However, nearly all models contain at least a few tunable parameters or parameter fields that are not known exactly. The parameters considered here usually represent physical characteristics of the system being simulated. Examples include chemical reaction rates in combustion simulations and permeability fields in groundwater models.

Regardless of the physical situation, it is usually necessary to tune model parameters to ensure model predictions agree with physical observations. Accurate calibration is critical for informative predictions. Additionally, in some situations, finding suitable parameters is the end goal. In order to understand the mechanics of glacier flow, glaciologists need to find basal stresses under a glacier. Obviously these cannot be measured directly, but their effect on glacial motion can be modeled. Thus, finding basal stresses is the end goal; future simulations using these values is a secondary objective. In practice, there is usual some error in observational data, and as the glaciology example shows, available observations may be indirect. The fundamental problem here of using noisy observations to find appropriate parameter values is usually called an inference problem, inverse problem, calibration problem,
or parameter estimation problem. The rest of this chapter will describe many of the existing methods for parameter estimation, which can be quite costly for large scale problems. To reduce the computational complexity, chapter 3 will introduce multiscale techniques for simulating groundwater flow. The application of these techniques in solving the inference problem will then be discussed. It will be shown that multiscale concepts can be used when solving an inference problem to dramatically reduce the required computational effort. The final chapter will discuss the performance of the multiscale procedure on a variety of test problems.

Historically, computational restrictions have forced simulation based engineering analysis and design to use deterministic methods. However, input data to numerical simulations are rarely known exactly in real engineering problems and understanding uncertainties in numerical predictions is paramount to creating robust designs. With the recent explosion of computational resources, there has been significant work towards efficiently characterizing the uncertainty in numerical predictions, especially in the context of large partial differential equation based mathematical models. In most cases, the uncertainty in these problems comes from incomplete knowledge of a model parameter or field. In a hydrology setting, the uncertain field is often subsurface permeability. The permeability field is rarely known precisely and forward uncertainty quantification methods are required to characterize model predictions of pressure, saturation, etc... Before these forward predictions can be used in engineering analysis, a probabilistic description of the field should be constructed from any available information. That is, an inference problem needs to be solved before the model can be used for prediction. Mathematically, an inverse problem is to find parameters \( k \) based on limited data \( d \) and a function \( G(k) \) that maps the parameter to a predicted \( \tilde{d} \):

\[
\tilde{d} = G(k) \quad (1)
\]

Because \( G(k) \) usually depends on a simulation or model of the physical system generating \( d \), we cannot hope to fully capture all the physics and processes that generated \( d \). The prediction \( \tilde{d} \) is only expected to approximately represent the potentially noisy
data $d$. Adding an error term, $\epsilon$ is one approach to incorporate this approximation:

$$ d = G(k) + \epsilon \quad (2) $$

This is an example of an additive error model. This choice is not unique, multiplicative error models are also common. In that case,

$$ d = (1 + \epsilon)G(k) \quad (3) $$

Other error models are also used depending on the specific application. Specifically, the error model can be tailored to acknowledge unmodeled physics or systematic measurement error. In this work, the additive error model will be used. However, all of many methods presented throughout this text can also be applied to situations with more sophisticated error models. Authors will often cite two classes of uncertainty that go into $\epsilon$: epistemic uncertainty and aleatoric uncertainty. No distinction between the two will be made here because I feel in the macroscale PDE setting studied here, the distinction is superfluous. An interested reader should refer to appendix A.1 for a detailed discussion.
The only source of knowledge is experience.

Albert Einstein

Chapter 1

Inference Background

1.1 Deterministic Inversion Overview

1.1.1 Inverse Problem Formulation

Let $k$ be a finite dimensional model parameter. The problem of characterizing $k$ from limited data $d$ can be rephrased as finding an estimate $k^*$ that minimizes the difference between $G(k)$ and $d$. The simplest and probably most well known approach is to find $k$ that minimizes the $L_2$ error:

$$
    k^* = \arg\min_k \|G(k) - d\|_2^2 = \arg\min_k (G(k) - d)^T (G(k) - d)
$$

(1.1)

which is simply nonlinear least squares. Various forms of nonlinear least squares, often with some sort of regularization, remain the standard formulation of deterministic inversion. A slightly more general weighted form is often used:

$$
    k^* = \arg\min_k (G(k) - d)^T W (G(k) - d)
$$

(1.2)

where $W$ is some weighting matrix representing the differences in measurement error or importance of each piece of data. In instances where the dimension of $d$ is less than the dimension of $k$, or when large variations in $k$ cause relatively small changes in $G(k)$, the inverse problem can become underdetermined or ill-posed and many
different values of $k$ will give nearly identical objective values. The study of such ill-posed problems really begins with Hadamard in 1902. He was the first person to formally define what it means for a problem to be well posed, or “un problème bien posé” in the original paper, [37]. There he states that a problem is well posed if the following three conditions are satisfied:

- The solution, $k^*$, exists
- The solution, $k^*$, is unique
- The solution, $k^*$, is stable

Here, stability implies the solution $k^*$ depends continuously on the data. When any one of these conditions is not satisfied, the problem is called ill-posed. Being primarily concerned with under determined problems in this text, the condition that the inverse problem solution be unique is our primary concern. Much of the work in inverse problems can be interpreted as attempts to transform an ill-posed problem into “un problème bien posé” by modifying the formulation. Additional work then focuses on efficient solution methods.

In order to ensure a uniquely solvable problem, more information needs to be used in the optimization problem formulation. A standard methodology introduced by Tikhonov in [76] and now widely used to solve ill-posed problems, is to add a regularization term to the objective, resulting in

$$k^* = \text{argmin}_k (G(k) - d)^T W (G(K) - d) + k^T (\Gamma^T \Gamma) k$$

where $\Gamma$ is called the Tikhonov matrix. When $\Gamma$ is properly chosen, the added regularization problem ensures the well-posedness of the problem. This new problem is sometimes called a damped least squares problem. Choosing the Tikhonov matrix is important for efficiently solving the problem with a nonlinear optimization technique. More information on choosing $\Gamma$ can be found in [4], where a general discussion of least squares methods is given. Tikhonov regularization is also used in the well known software package (PEST) developed by John Doherty for calibration problems arising
in subsurface flow. The PEST manual, [14], provides a description of nonlinear least squares with Tikhonov regularization in a practical setting.

The introduction of the Tikhonov penalty term can also be motivated through the Bayesian paradigm. This will be discussed in more detail in later sections; however, the interested reader should note that $\Gamma$ is related to the Bayesian prior distribution for $k$.

Compressive sensing techniques in the image and signal processing literature can also be viewed as a regularization to combat ill-posedness. In those problems, a limited number of samples are used to reconstruct a signal or image in a large dimensional space. In that setting, the number of nonzero terms in $k$ is used as a penalty to regularize the system. See [11] for an introduction to this topic.

1.1.2 Optimization Approaches

Solving the formulation in (1.3) can be difficult in real-world applications where $G(k)$ is a highly nonlinear model, and the data $d$ have significant noise. The speed of Newton’s method makes it the usual first choice for optimization algorithms, when Hessian information is available. However, obtaining accurate Hessians to (1.3) is infeasible in most real problems. Thus, Quasi-Newton methods that approximate the Hessian are often used. Standard Quasi-Newton methods for nonlinear least squares are the Gauss-Newton and the related Levenberg-Marquadt algorithms. These methods use the Jacobian of $G(k)$ and the residual to find an approximate Newton descent direction. First, define the residual vector, $r(k)$ as:

$$ r(k) = (G(k) - d) $$  \hspace{1cm} (1.4)

Now, the original weighted least squares problem in (1.2) can be rewritten as

$$ k^* = \arg\min_{k} r(k)^TWr(k) $$  \hspace{1cm} (1.5)
Clearly, the residual is nonlinear in $k$ because of the nonlinear forward model $G(k)$. Thus, linearizing $r(k)$ about a point $k^-$ gives:

$$ r(k) \approx r(k^-) + J(k^-)(k - k^-) $$

(1.6)

where $J(k^-)$ is the Jacobian matrix of $G(k)$ at $k^-$. Using this approximation in the least squares problem, we have:

$$ r(k)^T W r(k) \approx \left[ r(k^-) + J(k^-)(k - k^-) \right]^T W \left[ r(k^-) + J(k^-)(k - k^-) \right] $$

(1.7)

which is simply a linear weighted least squares problem. The solution, $k^+$ to this is just a solution to the normal equations and can be written as:

$$ k^+ = (J^T W J)^{-1} W J [J k^- - r(k^-)] $$

$$ = k^- - (J^T W J)^{-1} W J r(k^-) $$

(1.8)

where the dependence of the Jacobian on $k^-$ was dropped for clarity of expression.

The Gauss-Newton method will use $k^+$ as the next step in an iterative process that builds this linear approximation and solves the resulting linear least squares problem at each step. Compare (1.8) with the iteration in Newton’s method:

$$ k^+ = k^- - H^{-1} \nabla (r(k^-)^T W r(k^-)) $$

$$ = k^- - H^{-1} W J r(k^-) $$

(1.9)

Clearly, the Gauss-Newton method is Newton’s method with an approximate Hessian given by $J^T W J$. In fact, another derivation of the Gauss-Newton algorithm given in [52], obtains the Gauss-Newton step simply by looking at a Taylor approximation of the Hessian. To apply the Gauss-Newton method, the matrix $J^T W J$ needs to be nonsingular and well conditioned. This can be quite limiting in practice. [52]. The Levenberg-Marquadt algorithm addresses this issue by introducing a regularization.
parameter, \( \lambda \) to (1.8), resulting in:

\[
k^+ = k^- + (J^T W J + \lambda I)^{-1} W J \left[ G(k^-) - d \right]
\]  

(1.10)

Combined with a line search and adaptation strategy for \( \lambda \), this can perform quite well even when starting far from a local minima. Adapting the regularization parameter \( \lambda \) so that \( \lambda \to 0 \) as \( k^+ \to k^* \), as in PEST, [14], allows the Levenberg-Marquadt method to recover the speed of the Gauss-Newton algorithm in a basin of attraction around the optimum. It should also be noted that the Levenberg-Marquadt algorithm can be interpreted as Gauss-Newton with a trust region, so near the optimum, the trust region is large and does not restrict the step, resulting in pure Gauss-Newton iterations. See [52] for a rigorous discussion on the convergence of these methods.

In addition to these classical methods, there has also been some recent work in [9] and [10] using preconditioned Newton-Krylov methods for largescale PDE-constrained optimization problems. These methods use a very sophisticated adaptation of Newton’s method to address the problem of inverse problems when the underlying partial differential equations are known.

The classical methods above require Jacobian information and the work of Biros and Ghattas in [9, 10] requires Hessian information. In some instances adjoint methods can be used to efficiently obtain this higher order information and these derivative based methods can be used. In addition to [9, 10] examples using adjoints can be found in glacial dynamics calibration, [63]. A control theoretic approach using adjoints is often implemented in the glaciology community to infer basal boundary conditions [63, 49]. Additional information on adjoint methods for largescale optimization can be found in [8]. Unfortunately, for many intriguing calibration problems, adjoints are not a viable option due to complicated models or black box simulations. While the naive approach would be to use finite difference approximations\(^1\), derivative free optimization tools are a significantly more efficient option. Varying from simple pattern search methods to sophisticated hybrid algorithms, there is no clear winner

\[^1\text{This is in what PEST does by default}\]
when it comes to derivative free techniques; however, comparison papers such [62, 25] and application papers such as [58, 40] give guidance for choosing a derivative free algorithm.

1.2 Bayesian Inference Overview

In its simplest form, deterministic inversion gives a single point estimate for the calibrated parameters. Noisy data and approximate models bring into question how useful the parameter estimate will be for future predictions. Some deterministic methods can use Hessian information at the optimum to approximate local covariance but in nonlinear problems this approximation does not adequately quantify uncertainty in the solution.² Figure 1-1 illustrates in a simple setting the large effect that unquantified noise can have on a parameter estimate. The solid lines are illustrations of the least squares objective function and the the dashed lines show some uncertainty in the objective values coming from noise in the data. Deterministic inversion ignores the error and would only minimize the solid lines.

In the blue case, there is a sharp valley and any objective within the dashed lines would produce a similar least squares estimate for $k$. An objective like this could occur for a hyperbolic or nearly hyperbolic PDE forward model where $k$ could be an initial condition. On the other hand, the red line has a much broader valley, perhaps coming from an elliptic or parabolic PDE model where the data is from a much smoother field than the parameter $k$. In the wider case, within the dashed lines the least squares estimate for $k$ could be in a wide range $\approx [-1.7, 1.7]$. However, without incorporating the data noise into the parameter estimation problem problem, a deterministic least squares estimate would give $k = 0$ for both systems and a user would have no idea how reliable a forward prediction with the estimated $k$ will be. This can be particularly troublesome when estimating a parameter from smooth data, such as pressure data.

²For a linear model using Gaussian random variables in a Bayesian formulation, the inverse of the Hessian corresponds to the posterior covariance. In nonlinear models or non Gaussian distributions, the posterior distribution will not be Gaussian and cannot be completely described by a point estimate and covariance matrix.
Figure 1-1: The danger of simply using a point estimate for the calibrated parameters. In the blue, noise in the data will not result in a large error in the calibrated model. However, the same amount of noise in the red could result in a calibrated parameter somewhere in the range of \([-1.6, 1.6]\). A Bayesian approach allows this uncertainty to be captured and presented as part of the final inference result.

In a hydrology context, and then using that estimate in a hyperbolic system, such as tracer transport. Variability in the parameter estimate is dissipated by an elliptic operator, but when used in a hyperbolic operator, the forward predictions could be far from the truth. Bayesian inference is a probabilistic tool that allows uncertainty to be represented during the entire inference process. The goal is to build a probability density for \(k\) that is conditioned on the data. Before proceeding, it is important to note that in many senses, Bayesian methods can be interpreted as generalization of more commonly used frequentist methods. Sec A.2 in the appendix provides a brief justification for the Bayesian interpretation as a useful alternative to frequentist interpretations.
1.3 Bayes’ Rule

In addition to the derivation below, good introductions to Bayesian inference and Bayes’ can be found in [70, 29, 51]. However, a much more thorough discussion using manifolds is given in [75]. The latter approach allows for nonlinear data and model spaces which are not considered here. Additionally, [75] and [51] give a discussion on the Bayesian interpretation of least-squares problems.

1.3.1 Brief History

In words, Bayes’ rule (often called Bayes’ theorem) updates a prior belief of a parameter’s value with data to create a posterior belief. The degrees of belief are quantified with probability distributions. Reverand Thomas Bayes first introduced his theorem\(^3\) in the 18th century with application to the Binomial distribution. It was then generalized to arbitrary distributions by Laplace.\(^4\) However, Laplace limited his choice of prior distribution to uniform distributions. After some controversy and friction between frequentists and Bayesians (see appendix A.2 for more information), in the early 20th century, the works of Harold Jeffrey’s and Edwin Jaynes pushed Bayesian inference into the mainstream. Good textbooks by these authors can be found in [46] and [45]. As pointed out in [68], the widespread adoption of Markov chain Monte Carlo (MCMC) methods in the Bayesian setting did not occur until the late 1980s or early 1990s. The application of MCMC to sampling of posterior distributions, helped launch Bayesian statistics towards real applications and approach its current state as a general inference framework.

---

\(^3\)Bayes in fact did not publish his work, his friend Richard Price actually published Bayes’ work after his death. Interestingly, Richard Price thought that Bayes’ theorem could help prove the existence of god. See [5] for the original letter by Price accompanying the original essay by Bayes.

\(^4\)Laplace actually reinvented Bayes’ rule from scratch and showed its use for general densities, it was only later that he discovered the original work of Bayes.
1.3.2 Derivation

In its modern presentation, Bayes’ rule relies on the ideas of conditional probability and the interpretation of a probability as a degree of belief. Define a probability space \((\Omega, U, \mu)\), where \(\Omega\) is the sample space, \(U\) is a \(\sigma\)-algebra defined on \(\Omega\) and \(\mu\) is a probability measure on \(U\). A usual choice of \(U\) is the Borel \(\sigma\)-algebra. Any subset, \(A \subset U\), is called an event, so \(U\) represents all events that could occur. In the inference setting, \(U\) is all sufficiently well behaved sets of \(k\). The probability of an event occurring is computed from the probability measure as:

\[
P(A) = \int_A d\mu(\omega) \quad (1.11)
\]

for \(\omega \in \Omega\). Note that several probability measures can be defined over \(U\). This is in fact how degrees of belief will be defined. The prior degree of belief will be defined as a prior measure on \(U\) and the posterior degree of belief will be defined as a posterior measure on \(U\). The remainder of this text deals exclusively with real valued random variables, so we will assume \(\mu\) has a density \(\pi\) with respect to Lebesgue measure.

Now, let \(k\) be the parameter of interest and \(d\) be available data. The goal is to find the density \(\pi(k|d)\). Using the law of total probability this can be rewritten as

\[
\pi_{k|d}(k|d) = \frac{\pi_{k,d}(k,d)}{\pi_{d}(d)}
\]

Expanding the joint distribution in the other order gives:

\[
\pi_{k|d}(k|d) = \frac{\pi_{d,k}(d|k)\pi_{k}(k)}{\pi_{d}(d)}
\]

Now rewriting the denominator:

\[
\pi_{k|d}(k|d) = \frac{\pi_{d,k}(d|k)\pi_{k}(k)}{\int_{U} \pi_{d,k}(d|k)\pi_{k}(k)dk} \quad (1.12)
\]

Since the denominator does not depend on \(k\), it is not required during inference\(^5\) and

\(^5\)The denominator, called the evidence is not needed during inference but is a critical quantity
(1.12) is usually written more compactly as:

$$\pi_{k|d}(k|d) \propto \pi_{d|k}(d|k)\pi_k(k)$$  \hspace{1cm} (1.13)

which is the familiar Bayes’ rule of conditioning. The prior density, hereafter just referred to as the prior, is \(\pi_k(k)\) and the posterior is \(\pi_{k|d}(k|d)\). Additionally, \(\pi_{d|k}(d|k)\) is referred to as the likelihood function or just likelihood. The likelihood is usually chosen to be of a particular form based on the noise in \(d\). The difficulty in Bayesian inference methods is how to characterize \(\pi_{k|d}(k|d)\). In practical applications, the posterior can rarely be expressed in an analytic form and needs to be approximated numerically.

### 1.4 Bayesian Inference

As in the deterministic setting, denote the forward model parameterized by \(k\) as \(G(k)\). Incorporating a similar error,

$$d = G(k) + \epsilon$$

where we will assume \(\epsilon \sim N(0, \Sigma_{\epsilon\epsilon})\). The distribution of \(\epsilon\) is called the error model. While an additive Gaussian error model is used here, applications with more systematic error exist and will generally use non-additive and/or non-Gaussian error models. For example, in signal processing much of the noise comes from other communication systems and has more structure than the simple additive Gaussian model used. In that situation more suitable error models exist, see [60] for more information on general noise models in the signal processing context.

In the additive Gaussian error case, the likelihood is given by:

$$\pi_{d|k}(d|k) = \frac{1}{\sqrt{(2\pi)^k|\Sigma_{\epsilon\epsilon}|}} \exp \left[ -\frac{1}{2}(G(k) - d)^T \Sigma_{\epsilon\epsilon}^{-1} (G(k) - d) \right]$$  \hspace{1cm} (1.15)

Evaluating \(G(k)\) may require solving a partial differential equation or calling a black

in model comparison and Bayesian experimental design
Figure 1-2: The likelihood function constrains the likely regions of the parameter space. When the error is large, i.e. $\gamma$ is large for $\Sigma_{\epsilon\epsilon} = \gamma I$, the likelihood functions do not provide much new information and the posterior will be largely governed by the prior.

box model where no analytical form exists for the likelihood. However, given a particular $k$, the likelihood can be evaluated and the posterior can then be evaluated. The likelihood functions acts as a weighting of the prior measure that ensures the posterior will respect the data. Therefore, when the data is extremely noisy, it is easier for a prediction to “agree” with the data and the posterior will be more similar to prior than if little noise was present. Figure 1-2 shows the effect of noise on a potential likelihood function. As more noise is added, more portions of the parameter space have significant probability and the prior begins to play a larger role. This has an interesting analogy with Tikhonov regularization. As the problem becomes more ill-posed in the Hadamard sense, the regularization term plays a more important role in the optimization.

As will be shown later in section 1.6.3, even though the posterior density does
not have an analytic form, samples can still be drawn from the posterior distribution. These samples can then be used as an approximation to the posterior for use in future computations. While sampling methods are the focus of this work, it is important to note that alternatives exist. A variational approximation to the posterior can also be found.

1.5 Variational Bayes

The idea behind variational Bayesian methods is to approximate the posterior, $\pi_{k|d}(k|d)$ by a different parametric distribution $\tilde{\pi}_\theta(k; \theta)$ by choosing $\theta$ to minimize the Kullback-Leibler divergence between $\pi_{k|d}(k|d)$ and $\tilde{\pi}_\theta(k; \theta)$. Formally, we have

$$\theta^* = \arg\min_{\theta} \int_U \tilde{\pi}_\theta(k; \theta) \log \frac{\tilde{\pi}_\theta(k; \theta)}{\pi_{k|d}(k|d)} dk$$

(1.16)

The approximate distribution $\tilde{\pi}_\theta(k; \theta)$ is usually composed of exponential families or another analytically tractable distributions. This is similar to finding the basis function coefficients that minimize the energy norm in finite element methods. For a more thorough introduction to variational Bayesian approaches, [55] provides an introductory chapter.

1.6 Sampling Methods

Given a sufficient number of samples from a distribution, any required statistic can be computed. This includes event probabilities and distribution moments. Sampling methods are designed to generate samples of a distribution for use in this type of calculation. There are a variety of sampling methods, each with its own niche in Bayesian analysis. However, Markov chain Monte Carlo (MCMC) methods are predominantly used in the inference setting described above. To contrast MCMC with other sampling approaches, this section gives an overview of a few popular methods. The number of methods is incredibly large, the methods described here are only meant to give a flavor of the field. More detailed information can be found in [29]
1.6.1 Rejection Sampling

Rejection sampling is a simple way to sample from an arbitrary distribution in any dimension. However, the emptiness of high dimensional spaces causes the method to become inefficient even for moderately sized problems. Suppose that we wish to generate samples from a general distribution with density $\pi(k)$; however, we cannot directly from $\pi(k)$. Introduce an alternative, easy to sample, density $q(x)$. To generate a sample of $\pi(k)$, we can first sample, $k'$ from $q(k')$ and accept this sample as a sample of $\pi(k)$ with probability:

$$P_{\text{accept}} = \frac{\pi(k')/q(k')}{c}$$

where $c$ is the maximum ratio between the distributions:

$$c = \max_k \frac{\pi(k)}{q(k)}$$

Figure 1-3 illustrates the rejection sampling process. The numerator of (1.17) in the figure is given by $\beta/\alpha$. The normalization ensures the acceptance probability is always less than one.

1.6.2 Importance Sampling

Consider the problem of estimating the expectation of $f(x)$ over the distribution $\pi(x)$. That is, we wish to estimate

$$E_{\pi}[f(x)] = \int f(x)d\pi(x) = \int f(x)\pi(x)dx$$
Instead of sampling from $\pi(x)$ directly, we can sample from an alternative distribution $q(x)$ and estimate the expectation as:

$$D_\pi[f(x)] = \int f(x) \frac{\pi(x)}{q(x)} q(x) dx$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} f(x^i) \frac{\pi(x^i)}{q(x^i)}$$

where $N$ is the number of samples and the $x^i$ are iid samples taken from the distribution $q(\cdot)$. Introducing the weights, $w(x^i) = \frac{\pi(x^i)}{q(x^i)}$, the approximate expectation is simply a sum of weighted samples. In standard Monte Carlo estimates, $w(x^i) = 1$. Now, instead of needing to pull samples from $\pi(x)$ which may not be possible when $\pi(x)$ is a complicated distribution, only samples of $q(x)$ are needed. To make importance sampling worthwhile, $q(x)$ is usually chosen from an easily sampleable distribution such as a Gaussian. Another desirable feature of importance sampling is that the weights, $w^i$, also define the normalization factor in Bayes’ rule with no additional work. Let

$$\pi(x) = \frac{\pi_d(k|x)\pi_k(k)}{z}$$
then,

\[ z = \int \pi_d|k(d|k)\pi_k(k)dx = \int \frac{\pi_d|k(d|k)\pi_k(k)}{q(x)}q(x)dx \approx \frac{1}{N} \sum_{i=1}^{N} w^i \]

While this is a useful aspect of importance sampling, the choice of \( q(x) \) is not always obvious. Usually the distribution should approximate \( \pi(x) \) in some way, but \( \pi(x) \) could depend on the solution of PDE and be difficult to approximate. Also important for effective implementation is the weight variance, \( \text{var} \left[ \frac{\pi(x)}{q(x)} \right] \). A small variance means the weights are approximately constant and therefore imply \( q(x) \) is a relatively good approximation to \( \pi(x) \). Alternatively, \( \text{var} \left[ \frac{\pi(x)}{q(x)} \right] \) could be used as a metric of the proposal efficiency. In fact, minimizing this variance produces the optimal \( q(x) \) in terms of reducing the estimate variance. The goal is then to find a \( q(x) \) that is in some sense close to the optimal proposal to minimize the weight variance; this will increase the method’s sampling efficiency.

In some applications outside of inference, such as estimating the probability of rare events, importance sampling is currently the only feasible approach. Importance sampling allows a user to push sample points into very low probability regions that would otherwise go unsampled. This is advantageous when trying to accurately compute the probability of an event with small probability. In the rare event case, standard Monte Carlo sampling would waste many samples in high probability regions without gaining any information about the event of interest.

1.6.3 Markov chain Monte Carlo introduction

The difficulty in using importance sampling or rejection sampling to estimate distribution-wide quantities is that high dimensional parameter spaces are quite empty, and these methods can waste a lot of time sampling in low probability regions that are unimportant from an inference perspective. Markov chain Monte Carlo (MCMC) is similar to rejection sampling and standard Monte Carlo methods, but in a Markov chain. The
advantage of MCMC is once this method finds a high probability region, it will stay in or near that region and more effectively sample the parameter space. However, this advantage comes at the cost of correlated samples.

A Markov chain is a discrete time random process where the current state only depends on the previous state. More formally, if we have a chain with steps $1..n$, denoted by $\{k_1, k_2, ..., k_n\}$, the Markov property states $k_n$ and $k_1, ..., k_{n-2}$ are conditionally independent given $k_{n-1}$:

$$\pi(k_n|k_1, ..., k_{n-1}) = \pi(k_n|k_{n-1})$$

Note that unlike many introductions to Markov chains, each state $k_1, ... k_n$ is not in assumed to be in a discrete space, rather, the chain searches the parameter space $\mathbb{R}^m$ where $m$ is the number of parameters being inferred. However, for illustration, consider for a moment a discrete one dimensional Markov chain with three states, as shown in figure 1-4. When the chain is in state $i$, the probability of moving to state $j$ is given by $p_{ij}$.

As the number of steps goes to infinity, it is easy to visualize the probability of being in each of the states $s_1, s_2$ or $s_3$ convergences to quantities $P_1, P_2, P_3$ that depend on the Markov chain transition probabilities. This discrete distribution is called the stationary distribution of the Markov chain. To ensure the convergence of MCMC, the chain must be ergodic, meaning that no transient states exist and the system is aperiodic. Figure 1-4 illustrates these properties. A state is transient if there is a positive probability of never returning to that state. The opposite of a transient state is a recurrent state. A chain where every state is recurrent is termed a recurrent chain. This property is required for an MCMC chain. The period of a state is the greatest integer $d$ such that the probability of returning to the state in $d$ steps is zero. For a state $i$, the period is given by

$$d_i = \max n \ s.t. \ p_{ii}^{(n)} = 0$$

where $p_{ii}^{(n)}$ is the probability of returning to state $i$ after $n$ steps. A state is aperiodic
Figure 1-4: Example of a three state discrete Markov chain with transition probabilities denoted by $p_{ij}$. If you squint hard enough you can imagine a similar system in, $\mathbb{R}^m$, where we have an infinite number of states.

if $d = 1$ and like a recurrent chain, a chain is aperiodic if all states are aperiodic. Clearly, the last chain in figure 1-4 does not satisfy this property, since $d_1 = d_2 = d_3 = 3$. While not as easy to visualize, the concept of a stationary distribution can be expanded to $\mathbb{R}^m$. For more information on Markov chains, [33] provides a detailed discussion of discrete and continuous Markov processes outside of MCMC.

MCMC constructs a Markov chain so that the stationary distribution of the chain is equal in distribution to $\pi(k)$, the distribution we wish to sample. In this way, when an MCMC chain is run sufficiently long, the states at each step of the chain can be used as samples of $\pi(k)$. The simplest way to do this is with the Metropolis-Hastings rule, which is simply a method for constructing an appropriate transition kernel. A transition kernel is the continuous generalization of the transition probabilities in
The transition kernel for the Metropolis-Hastings rule is:

$$T(k^{n+1}; k^n) = q(k^{n+1}; k^n)\alpha(k^{n+1}; k^n)$$  \hspace{1cm} (1.18)$$

where $k^n$ is the current state of the chain at the $n$th step, $k^{n+1}$ is the state at the next step, $q(k^{n+1}; k^n)$ is called the proposal distribution, and $\alpha(k^{n+1}; k^n)$ is the acceptance probability of moving from $k^n$ to $k^{n+1}$. Under some technical constraints on the proposal ensuring chain ergodicity, this kernel will ensure the stationary distribution of the chain is $\pi(k)$. In particular, the support of the proposal must span the parameter space to ensure the chain is aperiodic. In simple cases an isotropic Gaussian with mean $k^n$ is often chosen for $q(k^{n+1}; k^n)$. In implementation, the transition kernel is implemented by taking a sample of the proposal and accepting or rejecting that sample according to the Metropolis-Hastings acceptance rule defined below. After drawing a proposed move $k'$ from $q(k; k^n)$, we accept $k'$ as the next step in the chain with probability $\alpha$ where

$$\alpha = \min(\gamma, 1)$$  \hspace{1cm} (1.19)$$

and

$$\gamma = \frac{\pi(k')q(k^n; k')}{\pi(k^n)q(k'; k^n)}$$  \hspace{1cm} (1.20)$$

In the case of symmetric proposals, this simplifies to

$$\alpha = \frac{\pi(k')}{\pi(k^n)}$$  \hspace{1cm} (1.21)$$

which was the original rule put forth by Metropolis and his colleagues in [59] before the generalization to (1.20) by Hastings in [38]. Algorithm 1 shows the general process of sampling with the Metropolis-Hastings rule and figure 1-5 shows a chain from MCMC using a symmetric Gaussian proposal. Clearly, the samples in the first illustration do not represent the desired distribution, but after 2000 samples, the chain seems to be converging on the correct distribution. While the MCMC chain is guaranteed to converge asymptotically to $\pi(k)$, there is no clear method for choosing the number of steps needed to effectively represent $\pi(k)$. Note also that by picking a starting point,
we are imposing bias on the early chain. To overcome this, a fixed number of initial steps are usually treated as burn-in samples and discarded.

The proposal distribution \( q(\cdot) \) plays an important role in how efficiently the chain explores the parameter space and will be discussed in much more detail in section 1.8. More information about convergence rates and technical conditions on the proposal can also be found in [31].

**Algorithm 1 Metropolis-Hastings MCMC**

**Require:** Distribution to sample, \( \pi(k) \)
- number of samples to generate, \( N_s \)
- proposal distribution, \( q(k; k^n) \)
- starting point \( k^0 \)

1: for \( n = 0 : N \) do
2: Generate sample \( k' \) from \( q(k; k^n) \)
3: Evaluate
   \[ \alpha = \min \left\{ 1, \frac{\pi(k')q(k^n; k')}{\pi(k^n)q(k'; k^n)} \right\} \]
4: Generate uniform random variable, \( u \sim U(0, 1] \)
5: if \( u < \alpha \) then
6: \( k^{n+1} = k' \)
7: else
8: \( k^{n+1} = k^n \)
9: end if
10: end for
11: Return \( k^i \) for \( i = 1..N \)

Figure 1-5: Example of Metropolis-Hastings MCMC sampling a simple Gaussian mixture in 2 dimensions. Shown is the MCMC chain after 50 iterations, the density estimate after 10000 steps, and the chain after 2000 steps.
1.7 Dynamic Problems

Until this point only static problems have been discussed. In static problems all data is available at once, and no direct treatment of time or order of the data is included in the inference formulation. However, many important problems are better treated dynamically. The dynamic problems exist when the data becomes available as time goes on. A good example of a dynamic inference problem is estimating the parameters in a weather simulation. Data such as radar measurements and wind speed observations are made over time and the model needs to be updated as the information becomes available. At time $t_0$ there is a certain amount of available observations and then at future times, say $t_1$, $t_2$, etc., more observations are made and need to be incorporated into the inference procedure. The inference procedure is thus recursive. Consider the posterior for $k$, at time $t_n$. Assuming a Markov property for $k$, the estimate of $k$ and $t_n$ only depends on $k_{n-1}$ and $d_n$, so we have:

$$
\pi(k_n|k_{n-1}, k_{n-2}, ..., k_1, k_0) = \pi(k_n|k_{n-1})
$$

(1.22)

Since this same property holds for $k_{n-1}$, we have

$$
\pi(k_n|d_n, d_{n-1}, ..., d_1) \propto \pi(k_0) \prod_{i=1}^{n} \pi(d_n|k_n)\pi(k_n|k_{n-1})
$$

(1.23)

Clearly, the posterior at time $n - 1$ becomes the prior in the next step. An obvious way to estimate the distribution at time $n$ is to treat the whole problem as a static problem and do MCMC. However, this recursive structure can be taken advantage of to develop more efficient dynamic algorithms.

An extensive variety of dynamic inference algorithms exist. Examples include the Kalman filter and its variants, sequential Monte Carlo methods, expectation propagation, and sequential Monte Carlo methods. Within the field of sequential Monte Carlo methods lie sequential importance sampling and the ever popular particle filtering. Going into detail on all of these methods could fill several volumes, and although interesting, much of the work is unrelated to this text. Interested readers
should consider [22] for an introduction to the Kalman filter and a detailed analysis of the Ensemble Kalman filter. Additionally, [17] and [54] are good references for sequential Monte Carlo.

### 1.8 Advanced MCMC

As alluded to earlier, the proposal distribution in Metropolis-Hastings style MCMC can dramatically effect the number of samples required to represent $\pi(k)$. Each evaluation of the posterior requires a computationally expensive forward evaluation meaning it is crucial to reduce the required number of steps in the MCMC chain. One measure of chain efficiency is the acceptance ratio: the fraction of proposed steps that were accepted with the Metropolis-Hastings rule. The acceptance rate will be 1 when the proposal is equal in distribution to $\pi(k)$. It then seems that a user should always strive to maximize the acceptance ratio. However, this logic is misleading. Consider a case where the proposal is an isotropic Gaussian density and $\pi(k)$ is a non-Gaussian density described by transforming a bivariate Gaussian in $k_1$ and $k_2$ to the coordinates:

$$
\begin{align*}
\tilde{k}_1 &= ak_1 \\
\tilde{k}_2 &= \frac{k_2}{a} + b \left(\tilde{k}_1^2 + a^2\right)
\end{align*}
$$

With $a = 1$, $b = 1$, and the correlation of the Gaussian set to $\rho = 0.9$. This gives the Banana shaped density in figure 1-6. Conveniently, the determinant of the transformation Jacobian is 1, making analytically computing expectations simple. Table 1.1 shows the proposal variance, average sample mean, and acceptance rate for 100 runs of Metropolis-Hastings MCMC with 1000 samples in the chain. The true mean of the Banana shaped distribution is $(0, -2)$. Clearly the case with the smallest proposal size has the largest acceptance rate, but it also has the largest error in the mean estimate. A random walk driven by a narrow proposal takes small steps and slowly wanders through parameter space. This means that even with a
high acceptance rate, the chain has not sufficiently explored the parameter space. A slightly wider proposal, such as 1 in this example, sacrifices rejected samples for more aggressive moves through parameter space. This allows the chain to better search all of the high probability regions of $\pi(k)$ and better estimate distribution quantities. Figure 1-7 shows a chain for each of the proposal sizes. The wandering behavior of first chain is indicative that the proposal variance could be too small. Furthermore, the plateaus shown in the last chain indicate that the chain is getting stuck because the proposal is too large for many proposals to be accepted. The history of an ideal chain would appear as white noise with a range spanning the parameter space. A chain is said to be mixing well if it appears as white noise. In figure 1-7 the middle proposal sizes are mixing better than chains with extreme proposal variances but still do not show great mixing characteristics.

Table 1.1: Example of Metropolis-Hastings MCMC performance on two dimensional Banana function. Proposal is diagonal Gaussian. 100 trials with 1000 Samples after a burn in period of 200 steps was used.

<table>
<thead>
<tr>
<th>Proposal Size</th>
<th>Estimated Mean</th>
<th>Acceptance Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truth</td>
<td>[0.0, 2.0]</td>
<td>NA</td>
</tr>
<tr>
<td>0.25</td>
<td>[0.1405, −1.6454]</td>
<td>0.615</td>
</tr>
<tr>
<td>0.5</td>
<td>[0.0822 − 1.8571]</td>
<td>0.4783</td>
</tr>
<tr>
<td>1</td>
<td>[−0.0453, −2.0301]</td>
<td>0.2717</td>
</tr>
<tr>
<td>2</td>
<td>[−0.0061, −2.0049]</td>
<td>0.1233</td>
</tr>
</tbody>
</table>

Figure 1-6 also shows why the mean estimate from the small proposal chain is inaccurate. The proposal is so small that even after 1000 steps, the chain has only covered part of the parameter space. If the chain was run longer, samples would eventually be taken on the lower left leg of the banana and a better estimate would be achieved. Clearly, the chains using wider proposals explore more of the distribution.

The ability to look at the history of an MCMC chain as in figure 1-7 and decipher what is happening (my proposal is too small or too large, or my chain is stuck in local mode of the distribution) is a black art. Some information, however, can be derived through the squint test. In this test, a user focuses hard on the chain history, squints, and hopes for divine intervention. It is often successful in practice, especially when the user tilts her head to the side while squinting.
Figure 1-6: Example of Metropolis-Hastings samples for increasing proposal variance. The contours show the true distribution, which has a mean of $(0, -2)$.

Clearly, choosing the right proposal is critical to efficient sampling and the correct proposal is often not obvious apriori. Solutions to this include adaptively adjusting the proposal by using previous samples as well as using derivative (or higher order) information about the posterior. Strictly speaking, adaptive algorithms break the Markov property needed for ergodicity, but in certain cases, it can still be shown that the adaptive proposal remains ergodic and yields a chain with $\pi(k)$ as a stationary distribution.

### 1.8.1 Delayed Rejection Adaptive Metropolis MCMC

From the name is should be clear that the two components of Delayed Rejection Adaptive Metropolis (DRAM) are delaying rejection (DR) and adapting the proposal covariance (AM). Delaying rejection constitutes a local adaptation of the proposal
Figure 1-7: Example of Metropolis-Hastings chains for increasing proposal variance. Blue is the x component of the chain and green is the y component. For small proposals, high acceptance rates exist but the jumps are small and the chain inefficiently explores the space (the meandering nature in the first plot). However, for too large a proposal, the frequent plateaus indicate that the acceptance probability is low.

to the posterior while adapting the proposal covariance is a more global adaptation strategy. Here we will begin with a description of DR, continue on to AM, and explain how the two approaches can be combined into an efficient MCMC sampling algorithm.

A large proposal in the Metropolis-Hastings framework allows a chain to more efficiently move through a large parameter space. However, the width of the proposal also results in many rejected proposals. After a rejection, instead of staying at the same location and trying again, what if a different proposal distribution was tried? Ideally the second proposal would allow the algorithm to move bit by bit while continuing to try the original proposal allows the method to move around the distribution
\[ \pi(k) \] more aggressively. This is precisely the idea behind DR-MCMC, originally proposed in [61]. If the initial proposal \( k'_1 \) from \( q_1(k; k^n) \) is rejected, another proposal, \( k'_2 \) from a new distribution \( q_2(k; k'_1, k^n) \) is tried. Naturally, the probability of acceptance has a new form:

\[
\alpha(k'_2; k^n, k'_1) = \min \left\{ 1, \frac{\pi(k'_2)q_1(k'_1; k'_2)q_2(k^n; k'_2, k'_1)[1 - \alpha_1(k'_1; k'_2)]}{\pi(k^n)q_1(k'_1; k^n)q_2(k'_2, k'_1, k^n)[1 - \alpha_1(k^n; k'_1)]} \right\} \quad (1.25)
\]

where \( \alpha_1(k'_1; k^n) \) is the original acceptance probability:

\[
\alpha_1(k'_1; k^n) = \min \left\{ \frac{\pi(k'_1)q_1(k^n; k'_1)}{\pi(k^n)q_1(k'_1; k^n)} \right\} = \min \left\{ 1, \frac{N_1}{D_2} \right\}
\]

This expression may look complicated and it is easy to get the ordering of the proposals incorrect. However, recognize that the numerator describes moving from the proposed point back to the previously rejected point, and the denominator describes the forward action, going from the current point to the next proposal. Thus, in the numerator we have, \( q_2(k^n; k'_2, k'_1) \) which is the density \( q_2(\cdot) \) parameterized by \( k'_2 \) and \( k'_1 \) and on the denominator we find \( q_2(k'_2, k'_1, k^n) \) which is parameterized by \( k'_1, k^n \) and evaluated at the most recent proposal \( k'_2 \). Any number of DR stages can be used. The general formula for the acceptance probability can be (painfully) written as:

\[
\alpha_i(k'_i; k^n, k_1, \ldots, k_{i-1}) = \min \left\{ 1, \frac{\pi(k'_i)q_1(k'_{i-1}; k'_i)q_2(k'_{i-2}; k'_{i-1}, k'_i)\cdots q_i(k^n; k_1, k_2, \ldots, k_i)}{\pi(k^n)q_1(k'_1; k^n)q_2(k'_2, k^n, k_1)\cdots q_i(k'_i, k^n, k_1, \ldots, k_{i-1})} \frac{[1 - \alpha_i(k'_{i-1}; k'_i)])[1 - \alpha_{i-2}(k'_{i-2}; k'_{i-1}, k'_i)]\cdots[1 - \alpha_1(k'_1; k_2, \ldots, k_i)]}{[1 - \alpha_1(k'_1; k^n)][1 - \alpha_2(k'_2; k^n, k_1)\cdots[1 - \alpha_{i-1}(k'_{i-1}; k_1, \ldots, k_{i-1}, k^n)]} \right\} \quad (1.26)
\]

Furthermore, at the \( i \)th stage, a recursive formula for the denominator exists:

\[
D_i = q(k'_i; k_{i-1}, \ldots, k_1, k^n)(D_{i-1} - N_{i-1}) \quad (1.27)
\]

A common methodology is simply to shrink the proposal by some factor at each stage.
In this way, when a sufficient number of stages is used, eventually a narrow proposal will be found that should ensure an accepted move. This can also be interpreted as a local approximation of the proposal to $\pi(\cdot)$. If the chain is currently in a wide region of high probability, a large proposal will be advantageous. When the chain is in a narrow slit of high probability, a narrow proposal will be more appropriate. The DR methodology allows the proposal to adapt to these features. However, caution needs to be taken in how many DR stages are used. In the inference setting, each stage requires an evaluation of the forward simulation, which can be quite computationally expensive and excessive DR stages may make the inference intractably slow.

The adaptive Metropolis method introduced in [35] and developed further in [36] provides a more global adaptation strategy. As an MCMC chain progresses, it evaluates $\pi(k)$ at many points and builds up a large number of samples to approximate $\pi(k)$. As the chain progresses, using the previous samples to generate an efficient proposal distribution could be beneficial but the Markov property no longer holds. However, under some conditions, the Markov property can be broken and ergodicity can still be proven. The adaptive Metropolis algorithm, AM-MCMC, does just this. Using the previous samples of the chain, AM builds an efficient proposal covariance matrix while ensuring the chain remains ergodic. Let $n_0$ be the number of burn in samples used before adaptation. The AM adapted covariance at step $n$ is then

$$C_n = \begin{cases} 
C_0, & n < n_0; \\
 s_d \text{Cov}(k^1, \ldots, k^{n-1}) + s_c \epsilon I, & n > n_0
\end{cases}$$

(1.28)

where $s_d$ is a parameter depending on the dimension $d$ of $k$ and $\epsilon$ is a small constant ensuring the covariance stays positive definite. In some circles $\epsilon$ is referred to as a nugget. $C_0$ is the initial proposal covariance matrix used in the first $n_0$ steps. To reduce computational cost, the incremental formula:

$$C_{n+1} = \frac{n-1}{n} C_n + \frac{s_d}{n} \left( n \bar{k}_{n-1} \bar{k}_{n-1}^T - (n+1) \bar{k}_n \bar{k}_n^T + k^n (k^n)^T + \epsilon I \right)$$

(1.29)

should be used to compute the covariance. Here $\bar{k}_n$ is the average of the chain until
step $n$:

$$k_n = \frac{1}{k} \sum_{i=1}^{n} k_i$$

In [36] this update was proved to be ergodic. In practice the update is not performed every step. Through experience it has been found that updating every few steps improves mixing more dramatically than adapting every step. Furthermore, in the case of when $\pi(\cdot)$ is Gaussian, Gelman et al. showed in [28] that $s_d = (2.4)^2/d$ optimizes the mixing properties of the chain. This is also a good starting point for tuning the method on non-Gaussian distributions.

First combined in [34], DR+AM=DRAM provides both the local adaptation of DR and global adaptation of AM, providing better mixing and efficiency than either of the methods individually. As the authors of [34] point out, there are many ways of combining DR and AM. A straightforward approach uses all samples in the chain for the AM part, regardless of what DR stage they were accepted at. AM does not use the intermediate DR samples, just the final step of the chain at each instance in time. Furthermore, the proposal at each DR stage is taken to be a scaled version of the adapted AM proposal. For example, if the covariance of the AM proposal is $C_n$, then the covariance of the $i$th DR stages will be $\alpha_i C_n$ for some $\alpha_i < 1$. See [34] for a proof showing this combination remains ergodic. The see the effects of DR, AM, and the combination, DRAM, we go back to sampling the banana problem. Just for motivation, consider an initial isotropic covariance with variance 2, i.e. $C_0 = 2I$. In the previous examples we showed that this proposal was too large for efficient sampling with Metropolis-Hastings. However, as figure 1-8 shows, each of the adaptive methods, DR and AM, independently provide better mixing.

The combination, however, produces even superior mixing, as seen by the much larger jumps taken with DRAM. Also note the wandering behavior of DR with a few large jumps. DR is replacing the plateaus seen in the MH chain with small steps. The large initial proposal means the AM algorithm has a hard time adapting initially because little of the parameter space has been explored. However, when combined with DR, more samples are initially accepted and a good covariance adaptation occurs. As
Figure 1-8: Comparison of the DRAM family of adaptive algorithms. Blue is the x component of the chain and green is the y component. The initial proposal size was set to $C_0 = 2I$. Three DR stages were used, each shrinking the previous proposal by a factor of 4. The covariance was adapted with AM every 20 steps.

a reminder, MCMC is a stochastic algorithm so these chains are only representations of general features shown by DR, AM, and DRAM. If the same code was run a second time, different chains would be produced.

1.8.2 Langevin MCMC

DRAM adapts the proposal based on previous samples and in some sense the local structure of $\pi(\cdot)$. An alternative to the shrinkage used in DR, is to use more information about the distribution to build a more efficient proposal. One approach, called Langevin MCMC, takes advantage of gradient information to nudge the proposal mean towards high probability regions of $\pi(\cdot)$. Assume the gradient of log $[\pi(\cdot)]$ can
be computed. Following the detailed discussion in [69], a general Langevin diffusion $X(t)$ is defined as the solution to the stochastic differential equation

$$dk(t) = b(k(t))dt + \sigma(k(t))dB(t)$$  \hspace{1cm} (1.30)$$

where $B$ is Brownian motion and $a(k(t)) = \sigma(k(t))\sigma^T(k(t))$ is a symmetric positive definite matrix. Notice in this continuous form, $\sigma(k(t))$ acts like a Cholesky decomposition of a covariance matrix in discrete space. A commonly used special case of Langevin diffusion is

$$dk(t) = \frac{1-2d}{2} \nabla \log \pi(k(t))dt + dB(t)$$  \hspace{1cm} (1.31)$$

By removing $\sigma(k(t))$ the Brownian motion has become isotropic. Note that the stationary distribution of $X(t)$ for this equation is $\pi(k)$, so solving this equation exactly would give $\pi(k)$. However, in general, this SDE cannot be solved analytically, so a discretization needs to be made to solve it numerically. Using a forward Euler discretization in time gives the simple equation:

$$k' = k^n + \epsilon^2 \nabla \log \pi(k^n) + \epsilon z^n$$  \hspace{1cm} (1.32)$$

where $z$ is a sample of iid standard normal random variables and $\epsilon$ is the discretization time step. Discretizing the system introduces error into the approximation and no guarantee exists that the stationary distribution will remain $\pi(k)$ as in the continuous case. Thus, the Metropolis-Hastings rule can be applied to account for any error in the discretization. In this setting, $k'$ is used as the proposed point in the Metropolis-Hastings acceptance rule. All the Langevin proposal is doing is uniformly scaling and shifting an isotropic Gaussian density (the $z$ term). Using the gradient ensures the shift is towards a higher probability region, but no correlation between components of $k$ is taken into account in the proposal covariance. As [69] and [32] point out, more information about parameter correlation can be incorporated through
a preconditioning matrix $M$, to give:

$$k' = k^n + \frac{\epsilon^2}{2} M \nabla \log \pi(k^n) + \epsilon \sqrt{M} z^n$$

(1.33)

Notice that this a discrete version of the more general Langevin diffusion defined in (1.30). How to choose $M$ is unclear, especially when different areas of parameter space are correlated differently (think banana). In the inference setting, it seems that some structure of $G(k)$ could be used to choose $M$ at step in a similar fashion to the approximate Hessian in the Levenberg-Marquardt optimization algorithm.

Methods like Langevin MCMC require knowledge of the distribution gradient. In some cases, analytic forms or adjoint methods could be used to compute the necessary derivatives efficiently. However, in many situations involving sophisticated simulations or commercial code, evaluating $\pi(k)$ must be treated as a black box operation and derivative based samplers like these cannot be used. However, the task of shifting the proposal towards higher probability regions without derivative information is eerily similar to ideas in derivative free optimization (DFO). Methodology from local DFO methods, such as pattern search, its generalization generating set search, and other methods such as implicit filtering or Nelder Mead, could prove to be useful within an MCMC framework.

### 1.8.3 Other MCMC research

The MCMC literature is extensive and could not be completely discussed here. Any introductory text on MCMC usually includes the Gibbs sampler. However, Gibbs sampling is not general enough for most PDE constrained problems and will not be discussed here. More advanced sampling methods include hybrids where multiple MCMC chains are run simultaneously on the same distribution and information is passed between the chains to improve mixing. Examples of using evolutionary algorithms to do the swapping can be found in [18] and [72]. Furthermore, an MCMC sampler combined with differential evolution can be found in [77]. Still along the lines of combining optimization with MCMC, [30], uses Hessian information to build
an efficient proposal mechanism. This work is geared towards large PDE systems, similar to [9, 10], where adjoint methods can be used to efficiently find derivative information. In a very different approach, [32] shows how to use Riemann Manifolds to dramatically improve the sampling efficiency of Langevin and Hamiltonian MCMC methods. Multiscale MCMC methods can also be used to improve sampling efficiency. These will be discussed in future sections after multiscale modeling has been introduced.

1.9 Inference Summary

The general Bayesian inference procedure is to build a posterior distribution that represents uncertainty in parameters of interest conditioned on data. However, in sophisticated models, the relationship between the parameters and observations can be highly nonlinear and difficult to compute (i.e. solving a PDE). This means that an analytic form for the posterior is rarely available and sampling methods such as MCMC need to be used. The computational expense of the forward simulation also dictates that the MCMC sampler be as efficient as possible. From the previous discussions, it is obvious that this is not a trivial task and many evaluations of the posterior are still needed to accurately represent the distribution, especially in the case of high dimensions. For large problems, running many simulations is intractable. In order to make these problems feasible, it is therefore necessary to reduce the computational expense of each forward evaluation. Multiscale simulation techniques, the topic of the next chapter, are one way of achieving this.
Chapter 2

Multiscale Background

Imagine you are on a beach staring out at the ocean. You see the big waves rolling in and carrying surfers towards the shore, but you will also see smaller cross waves and little ripples that seem to move independently of the big waves. Also, if you were to sit in that spot long enough, the tide may come in and soak your blanket. Each of these observations represents a different length scale. Over short lengths, the ripples and small cross waves are observed. On a slightly larger time scale are the surfing waves, and on an even longer scale is the tide. This is a prime example of a multiscale system. Scale here effectively refers to the correlation length of the system.

It turns out that like the ocean, many natural systems exhibit multiscale behavior. Examples include permeability fields in subsurface flow as well as reactions in chemical kinetics. Figure 2-1 gives an example of multiscale behavior in groundwater flow. Plotted is water depth below the surface elevation at a USGS monitoring station in Idaho. The first plot clearly shows yearly correlation lengths while the second plot\(^1\) shows distinct daily patterns. As an aside, the daily patterns may in fact be due to an interesting effect called an earth tide. This occurs as the moon passes over a point on the earth and causes a dilation force on the bedrock. The force slightly stretches the media, opening up more space and reducing the pressure. Details of this phenomena can be found in [56] and the references therein.

\(^1\)Note that the second plot is based on provisional data measured the week before this thesis was written.
Figure 2-1: Example of groundwater observations with multiscale behavior. Not only are yearly patterns present, but daily length scales exist as well. Data comes from USGS well measurements in Bingham County, Idaho.

Taking advantage of multiscale features when developing solution methods and sampling methods can dramatically improve efficiency. By concentrating on the scales of interest, it is sometimes possible to reduce much of the computational effort required for simulation. For example, in the ocean example mentioned above, certain applications may only be concerned with tidal effects while other may be heavily de-
dependent on mid sized waves and their erosion effects. This chapter will begin with a
discussion of multiscale simulations in groundwater flow and will later turn to current
multiscale sampling strategies. Before proceeding, I would like to emphasize that the
multiscale tools developed here are in no way restricted to porous media flow. Many
models with multiscale behavior can fit into this framework, especially in situations
where the model output is much smoother than model input.

2.1 Multiscale Simulation Methods

2.1.1 Model Elliptic Equation

The focus of this work is on applications in porous media flow. A basic model used
as a building block for many more sophisticated simulations is the pressure equation.
To see where this model comes from, consider the conservation of mass for fluid flow
through a small volume of porous media as shown in figure 2-2.

\[ \rho u_y + \frac{\partial (\rho u_x)}{\partial y} dy \]

\[ \rho u_x + \frac{\partial (\rho u_x)}{\partial x} dx \]

Figure 2-2: Conservation of mass in a porous media. When coupled with Darcy’s
equation as a constitutive law, this gives the familiar pressure equation.

Using the simple relationship

\[ \frac{\partial M}{\partial t} = \text{Mass In} - \text{Mass Out} \]
with the mass flows in figure 2-2, leads to

\[
\frac{\partial (\rho \phi)}{\partial t} \, dxdy = - \frac{\partial (\rho u_x)}{\partial x} \, dxdy - \frac{\partial (\rho u_y)}{\partial y} \, dxdy
\]

where \( \rho \) is the density of the fluid, \( u_x \) is the inflow velocity in the \( x \) direction, \( u_y \) is the inflow velocity in the \( y \) direction. The term \( \frac{\partial (\rho u_x)}{d} \, x \) is the rate of change in the \( x \) direction times the \( x \) distance traveled. Since we are working in the limit as \( dx \to 0 \), no higher order terms are needed. Also, \( \phi \) is the porosity of the media. This term is needed because the fluid only occupies the pore space, so \( \rho \phi dxdy \) is the mass in this representative volume and \( \frac{\partial (\rho \phi)}{\partial t} \, dxdy \) is the mass rate of change. Assume the density is constant in time. Thus, we have

\[
\rho \frac{\partial \phi}{\partial t} + \rho \nabla \cdot (u) = 0
\]

\[
\Rightarrow \frac{\partial \phi}{\partial t} + \nabla \cdot (u) = 0
\]

where the area terms \( dxdy \) have been cancelled out. Note that these equations are on the continuum scale, no longer are the definitions only for the representative volume element. Assuming that \( \phi \) is a linear function of pressure \( p \), i.e \( \phi = pC_t \) where \( C_t \) is the compressibility of the media, we have

\[
C_t \frac{\partial p}{\partial t} + \nabla \cdot u = 0 \quad (2.1)
\]

Obviously this equation is underdetermined; a constitutive law is needed. Here, Darcy’s law is used to complete the system:

\[
u = -\frac{\tilde{k}}{\mu} (\nabla p) \quad (2.2)
\]

where \( \tilde{k} \) is the permeability of the media and \( \mu \) is the dynamic viscosity of the fluid. The previous two equations constitute the conservative form of the pressure equation. In this work, conservation is not critical.\(^2\) The system considered here is the steady

\(^2\)Conservation is not critical in this study because we are using the pressure equation as a proof
state system after non-dimensionalization, given by:

\[- \nabla \cdot (k \nabla p) = 0 \quad (2.3)\]

where \(k\) is now a spatially varying field representing the scaling of an intrinsic permeability field. From here on, \(k\) will be referred to simply as the permeability. Clearly, the pressure is a nonlinear function of the permeability. Thus, to use the predicted pressure from (2.3) as a tool for engineering design or water resource planning, not only do boundary conditions need to be tailored to the region of interest, but the permeability, \(k(x)\), needs to be characterized as well. Typically, very few direct measurements of permeability exist and indirect observations of pressure must be used in conjunction with (2.3) to infer the permeability field. In this work, it will be assumed that no direct observations exist and \(m\) observations of the pressure \(p(x)\) have been taken at a limited number of locations, \(\{x_1, x_2, \ldots, x_m\} \in D\), where \(D\) is the spatial domain of interest. This vector of observations will be denoted \(d\). Ill-posedness of the inverse problem arises when estimating \(k(x)\) from \(d\) because the mapping \(p(x) = G(k(x))\) defined by (2.3) acts as a nonlinear lowpass filter, removing high frequency effects of \(k(x)\) from \(p(x)\). This high frequency information cannot be recovered during inference.\(^3\) This means that potentially large, high frequency changes in permeability will result in only minor pressure changes. Figure 2-3 demonstrates this property for a one dimensional system and a two dimensional layer of the SPE10 dataset.\(^{12}\) The pressure is found using the two log(\(k\)) fields on the left. In one dimension, the high frequency blue field is just the green field plus some correlated noise. Clearly the high frequency addition does not have a large impact on the pressure in the right plot. Intuitively it seems that because the pressure field is smoother, and can thus be more easily represented, only some of the information in the permeability

---

\(^3\)For continuous fields, if \(k(x)\) has continuous derivatives of order \(m\), \(p(x)\) will have \(m + 1\) continuous derivatives.
field should be needed to construct the pressure field. That is, only certain scales of \( k \) have a significant impact on \( p \).

\[
\log(k)
\]

\[
x
\]

\[
y
\]

Permeability, \( \log_{10}(k) \)

\[
y
\]

Pressure, \( p \)

\[
x
\]

Figure 2-3: Example of multiscale field and corresponding pressures. The one dimensional case has a dirichlet condition on the left and a homogeneous Neumann condition on the right. In the two dimensional case, all boundaries are fixed with Dirichlet conditions. Clearly, the pressure fields are much smoother than the permeability fields. The one dimensional case shows the smoothing effect of the elliptic operator and the two dimensional setting again shows the relative smoothness of the pressure field compared to the permeability field.

In largescale (regional) simulations, the meshes needed to resolve fine scale features can be prohibitively large for performing global pressure solves. However, being much smoother than permeability, it seems reasonable to solve for pressure on a more computationally tractable coarse mesh. There are several methods of doing this. Often the most straightforward approach, known as upscaling, is to solve the pressure equation on a coarse mesh with a representative coarse permeability field. Once the coarse permeability has been found, the pressure equation is then solved on
the coarse mesh. Note that upscaling the permeability field is an artificial coarsening of the system. Only pressure can be accurately represented on the coarse scale. Furthermore, with upscaling, the pressure is only represented on the coarse mesh, even though some fine scale features may exist and there is no way to find an approximate fine scale pressure field based on the coarse solution.

The desire to solve the pressure equation on a coarse mesh but maintain some fine scale features is met in variational methods such as Multiscale Finite Element Methods, Varition Multiscale methods, and heterogeneous multiscale methods. The idea behind all of these solution strategies is to implicitly coarsen the pressure equation through a variational minimization. These methods allow the pressure to be approximated on a fine mesh, while only solving a small linear system for a coarse representation of the pressure. The following sections describe and contrast upscaling and these variational methods.

2.1.2 Upscaling

In the ocean example, upscaling could be defined as smoothing or averaging the waves into a quantity on the tidal scale. In the porous media setting, upscaling is just taking the average fine scale permeability over a coarse element. Choosing an appropriate average is not trivial. Analytic averages such as the arithmetic or harmonic mean would provide a simple form for the coarse permeability, but do not introduce any physics into the operation. To see that choosing the arithmetic mean is not a good choice, let \( p = G(k) \) represent the nonlinear mapping from the permeability to pressure. We would like the coarse pressure solution to be the average of the fine scale pressure, but

\[
\mathbb{E}[p] = \mathbb{E}[G(k)] \neq G(\mathbb{E}[k])
\]

Clearly, taking the arithmetic average of \( k \) will not give the average pressure as desired and more sophisticated techniques need to be employed for upscaling to be effective. Nevertheless, the harmonic mean and arithmetic mean provide bounds on the effective permeability. As mentioned in the review paper, [66], and the references within, these
means provide the so-called Wiener bounds:

\[ \mu_h \leq k_{\text{eff}} \leq \mu_a \]  \hspace{1cm} (2.4)

where \( k_{\text{eff}} \) is the effective permeability on a coarse element, \( \mu_h \) is the harmonic mean over the coarse element, and \( \mu_a \) is the arithmetic mean. Figure 2-4 shows an example of various means for a layer of the SPE10 dataset. The averages were taken over 10 fine cells in the x-direction and 5 fine cells in the y-direction. After some inspection, it is apparent that on each coarse cell, the geometric mean lies between the harmonic and arithmetic means. In fact, many tighter bounds on the effective mean can be proved under various circumstances. See [66] for more information.

The Wiener bound is not tight, so to overcome the nonlinear relationship between \( k \) and \( p \), a nonlinear upscaling based on model physics needs to be used. On each coarse element, a few local fine scale solves can be used to characterize the effective permeability. Farmer gives a good review of these methods in [23]. To give a general idea of the method, consider a two dimensional problem with quadrilateral coarse elements of size \( h_x \times h_y \). In order to build a coarse permeability tensor, on each coarse element, two local solutions of the steady state pressure equation could be found. The first would enforce homogeneous Neumann (no-flow) boundary conditions on the top and bottom boundaries, with specified pressures at the other boundaries, and the second would switch the flow direction, with the Neumann conditions on the left and right. Once the pressure has been found, the total flux, \( Q \), through one of the Dirichlet boundaries can be found and Darcy’s law will give the effective permeability:

\[ k_{\text{eff}} = \frac{Q h_x}{A \Delta P} \]  \hspace{1cm} (2.5)

where \( k_{\text{eff}} \) is the effective permeability in the \( x \)-direction, \( Q \) is the computed flow rate, \( h_x \) is the horizontal size of the element, \( A = h_x h_y \) is the element area, and \( \Delta P \) is the pressure drop induced by the Dirichlet boundary conditions. In 1961, Warren and Price in [78] first introduced this idea and various extensions have become prevalent in petroleum engineering. In one dimension, choosing appropriate boundary
conditions is trivial and the upscaled permeability can perform quite well. However, in higher dimension, not only does the computational cost increase because local flow simulations need to be computed to fill in the effective permeability tensor, but the choice of boundary conditions for the local solves is also not obvious.

Figure 2-4: Example of geometric, arithmetic, and harmonic averaging. After inspection, note that the geometric means visually seems to be between the arithmetic and harmonic means, as required by the Wiener bounds. This field is \( \log_{10}(K) \) for the 10th layer of the popular SPE10 dataset.

In a fairy-tale world, computed effective permeabilities would not depend on the choice of local boundary conditions. However, we have not fallen into a rabbit hole and the boundary conditions can significantly impact the computed effective per-
meabilities. The methods for alleviating this dependence discussed in [23] include slightly increasing the computational domain size to reduce boundary effects, or using an approximate global solution to choose the boundary conditions. Additionally, some more recent publications, see [39] and [65], couple the mesh-generation process with the upscaling procedure to choose a mesh that reduces the boundary effects. The issue of appropriate boundary conditions for local solves is not unique to upscaling and will be discussed further in section 2.1.5 in the context of the Multiscale Finite Element Method (MsFEM).

The upscaling methods discussed above are useful when the fine scale permeability field is known. However, stochastic upscaling methods also exist when the fine scale permeability is represented as a stochastic field. Kitanidis provides an introduction to stochastic upscaling methods in [53]. The goal of stochastic upscaling is to use information about the fine scale distribution of $k$ to develop effective permeabilities. The main approach discussed in [53] is based on small perturbation theory, where an assumption of small variance leads to the analytic form:

$$k_{eff} = k_g \exp \left( -\frac{\sigma^2}{n} + \frac{\sigma^2}{2} \right)$$ (2.6)

Here, $k_g$ is the geometric mean of the stochastic field $k$, $\sigma^2$ is the variance of $\log(k)$, and $n$ is the domain dimension. Notice that in this case, the effective permeability is no longer stochastic. The randomness only exists on the fine scale. This upscaling operation is a method for taking a log-normal stochastic $k$ field and computing an effective permeability that represents the general structure of the stochastic field. In section 3.3, we will take a different approach in which the coarse quantities are also represented through probability distributions. It will also become clear that a stochastic coarse representation can be useful in a multiscale inference setting.

### 2.1.3 Homogenization

A good upscaling method introduces a physical model (the pressure equation here) to generate a coarse representation of the permeability. The process of upscaling to
Fine scale permeability, $k$

Local pressure solution, $p_l$

Coarse permeability, $k_c$

(a) Upscaling Process

Local pressure solution: $-\nabla \cdot (k \nabla p_l) = 0$

Represent multiscalar field

$k = k(x, y = \frac{x}{\epsilon})$

Multiscale expansion:

$p_\epsilon(x, y, \epsilon) = p_0(x, y) + \epsilon p_1(x, y) + ...$

Build coarse system:

$\left( \frac{1}{\epsilon} A_0 + \frac{1}{\epsilon} A_1 + A_0 \right) p_\epsilon = f$

(b) Homogenization Process

Homogenized permeability, $k^*$

Figure 2-5: Comparison of Homogenization process and upscaling process. While upscaling builds a coarse permeability and uses the same governing equation, homogenization finds a homogenized permeability by putting constraints on the coarse operator.

the coarse scale can be represented graphically as figure 2-5. An alternative analytic approach is homogenization. The object of homogenization is to find a multiscale expansion of the solution field whose coefficients can be found by solving a homogenized (coarsened) system. The expansion considers the action of the elliptic operator on a rapidly oscillation permeability field. The homogenization process is also outlined in figure 2-5. In this setting, the permeability is represented as a periodic field with period $\epsilon$ and the pressure equation becomes

$$A'p = -\nabla \cdot [k(x/\epsilon)\nabla p] = f$$  \hspace{1cm} (2.7)

The operator $A'$ represents the pressure equation (an elliptic operator) built from a permeability field with period $\epsilon$. Let $y = x/\epsilon$ represent a “fast” coordinate. The goal
is then to find a solution expansion of the form

\[ p_\epsilon(x) = p_0(x, y) + \epsilon p_1(x, y) + \epsilon^2 p_2(x, y) + \ldots \]  

(2.8)

The introduction of the dependence of \( p \) on \( y \) requires the use of a new total derivative, given by:

\[ \nabla = \nabla_x + \frac{1}{\epsilon} \nabla_y \]  

(2.9)

Using this in the pressure equation gives:

\[ - (\nabla_x + \frac{1}{\epsilon} \nabla_y) \cdot \left[ k(y)(\nabla_x p + \frac{1}{\epsilon} \nabla_y p) \right] = f \]  

(2.10)

which after some algebra gives the compact form:

\[ A^\epsilon p = \frac{1}{\epsilon^2} A_0 + \frac{1}{\epsilon} A_1 + A_2 \]  

(2.11)

where

\[
A_0 = -\nabla_y \cdot (k(y) \nabla_y) \\
A_1 = -\nabla_y \cdot (k(y) \nabla_x) - \nabla_x \cdot (k(y) \nabla_y) \\
A_2 = -\nabla_x \cdot (k(y) \nabla_x)
\]

The pressure equation is now

\[ \left( \frac{1}{\epsilon^2} A_0 + \frac{1}{\epsilon} A_1 + A_0 \right) (p_0(x, y) + \epsilon p_1(x, y) + \epsilon^2 p_2(x, y) + \ldots) = f \]  

(2.12)

Collecting terms, we find a system of equations:

\[
A_0 p_0 = 0 \quad (2.13) \\
A_0 u_1 + A_1 u_0 = 0 \quad (2.14) \\
A_0 u_2 + A_1 u_1 + A_2 u_0 = f \quad (2.15)
\]
Following [42], it is possible to show that $p_0$ is independent of $y$, and (2.14) can then be written as

$$-\frac{\partial}{\partial y_i} \left( a_{ij}(y) \frac{\partial}{\partial y_j} \right) p_1 = \left( \frac{\partial}{\partial y_i} a_{ij}(y) \right) \frac{\partial p}{\partial x_j}(x)$$

(2.16)

Now define $\chi_j$ as the solution to

$$\frac{\partial}{\partial y_i} \left( a_{ij}(y) \frac{\partial}{\partial y_j} \right) \chi_j = -\frac{\partial}{\partial y_i} a_{ij}(y)$$

(2.17)

After some technical arguments and looking at the limit $\epsilon \to 0$, [42], shows that the homogenized equation becomes

$$-\frac{\partial}{\partial x_i} \left( k^*_{ij} \frac{\partial}{\partial x_j} \right) p = f$$

(2.18)

Note, this is just the usual pressure equation with a rigorous choice of effective permeability! In fact, the homogenized permeability, $k^*$, is given by

$$a^*_{ij} = \frac{1}{|Y|} \left( \int_Y (a_{ij} - a_{ik} \frac{\partial \chi_j}{\partial y_k}) dy \right)$$

(2.19)

The literature on homogenization theory is vast and only the tip of the iceberg of homogenization for elliptic problems was introduced here. An interested reader can find a thorough discussion of homogenization and its relationship with averaging in [64].

### 2.1.4 Variational Methods

Homogenization and upscaling reduce the degrees of freedom in an elliptic solve and thus reduce the computational cost of the global solve. The idea is that being smoother than the permeability, computing pressure only requires lumped information about the permeability field. The lumped effective permeability however, is a fictitious field developed simply for numerical reasons. The vast literature on upscaling shows that it can be effective, but a more satisfying methodology would maintain the fine scale representation of $k$ and only develop a coarse approximation to $p$, the
field that truly exists on a smoother scale. Multiscale variational methods provide
just such an alternative. Additionally, the variational approaches can often provide a
systematic way of projecting the coarse pressure to a fine scale pressure that exhibits
important features seen in a full fine scale solution. Several variational multiscale
methods exist, including the Variational Multiscale Method (VMM) [44, 50], Hetero-
gegeneous Multiscale Methods (HMM) [19, 20], Subgrid upscaling [3], Multiscale finite
element methods [42, 2], and the Multiscale finite volume method [47, 48]. All of
these methods are very similar and on some specific problems can be equivalent. In
fact, some papers, such as [79] do not even provide a clear distinction between these
methods. Subtleties aside, the basic idea is to perform a limited number of local
pressure solves to build a set of basis functions for use in a coarse scale (often called
global) coupling. Here, we will focus on the multiscale finite element method which
uses the local basis functions in a coarse scale Galerkin finite element formulation.

2.1.5 Multiscale Finite Element Methods

The formulation here follows closely the introduction in [42]. For more general infor-
mation and useful background information on the standard finite element approaches,
see [71].

Let $\Omega$ be the domain of interest, where the pressure equation is to be solved.
Consider a coarse triangulation $T_h$ of $\Omega$ into finite elements. For each element of $T_h$,
a fine mesh can also be created. A simple choice is to use quadrilateral elements for
the coarse grid and then either quadrilaterals or simplex elements on the fine grid.
Let $p$ lie in a function space $X$, usually chosen as $H^1_0(\Omega)$. Then the usual weak form
reads: find $u \in X$ such that:

$$a(u, v) = b(v) \forall v \in X$$

The weak form of the pressure equation is given by

$$a(u, v) = \int_{\Omega} k(x) \nabla u \cdot \nabla v dx = \int_{\Omega} f v dx = b(v) \forall v \in X$$ (2.20)
From here, most finite element methods will discretize the system by defining nodal basis functions, \( \phi^0_i \), (often linear) and only consider \( v \in W_h = \text{span}\{\phi^0_i\} \). However, it is possible to choose these basis functions such that fine scale information is embedded into the coarse scale weak formulation. Consider the basis function \( \phi_i \) defined on the support of \( \phi^0_i \). Since \( \phi^0_i \) is a nodal basis function, the support is over the elements containing node \( i \) as a vertex. Let \( K \) be one of these coarse elements and define \( \phi_i \) to satisfy:

\[
- \nabla \cdot (k(x) \nabla \phi_i) = 0 \quad x \in K
\]

\[
\phi_i = \phi^0_i \quad x \in \partial K
\]

where \( \partial K \) is the coarse element boundary. Clearly, the MsFEM basis function, \( \phi_i \), matches \( \phi^0_i \) on the element boundaries, but solves the homogeneous pressure equation on the element interior. Thus, effects of the fine scale \( k \) on the pressure are embedded into the coarse scale formulation through \( \phi_i \). Figure 2-6 shows an example of an MsFEM basis function over one coarse element. In the case of figure 2-6, the coarse elements are quadrilaterals, so the figure only shows one quarter of a complete nodal basis function.

Let \( P_h = \text{span}\{\phi_i\} \) be the space spanned by the MsFEM basis functions. To discretize the system using MsFEM, we first represent the pressure in terms of the MsFEM basis functions:

\[
p_h = \sum_i p_i \phi_i
\]

and use the following weak formulation:

\[
\sum_K \int_K k \nabla p_h \nabla v_h \, dx = \int_{\Omega} v_h \, dx \quad \forall v_h \in P_h
\]

This is a Galerkin projection; however, a Petrov-Galerkin projection could also be used. Either form will yield a linear system:

\[
A_c p_{\text{nodal}} = b
\]
where $p_{\text{nodal}}$ is a vector of basis function weights, $A_c$ is the coarse stiffness matrix, and $b$ represents source terms. Elements of $A_c$ are given by

$$a_{ij} = \sum_K \int_K k \nabla \phi_j \nabla \phi_i \, dx$$

The integrals inside the summation will be referred to as *elemental integrals*. As pointed out by [42], in implementation it is more straightforward to use the relationship:

$$A_c = D^T A_f D$$

than the integral form. Here, column $i$ of $D$ is a vector holding the discrete basis function $\phi_i$, and $A_f$ is the global fine scale stiffness matrix. $A_f$ would be the result of using a standard finite element approach on the global fine scale mesh. Boundary conditions for (2.25) can be implemented identically to standard finite element approaches. Once the coarse pressure $p_{\text{nodal}}$ has been found, projecting this solution
back to the fine scale is trivial:

\[ p_f^* = Dp_{\text{nodal}} \]  \hspace{1cm} (2.27)

Since \( P_h \) is spanned by a relatively small number of basis functions compared to a full fine scale formulation, solving the resulting linear system is much more computationally tractable. However, local solves are still required to find \( \phi_i \). The local solves are easily parallelized. Local solves on each element are independent of all other elements. Furthermore, the basis functions do not need to be recomputed when the source terms in \( f \) are changed, a useful feature when considering MsFEM as the solver in a well-design framework, or other application where many solves are needed but only \( f \) changes between solves.

In the definition of the MsFEM basis functions the constraint \( \phi_i = \phi_0^i \) on \( \partial K \) was used as the boundary conditions of the local homogeneous pressure equation. However, it is well known that these boundary conditions do not represent any fine scale heterogeneities and significant errors can be produced on problems without a large scale separation in the permeability field. Possible solutions include oversampling, [42, 43], where the local solve is performed on a computational domain larger than a single coarse element and then truncated, or solving one dimensional problems along the boundary, as in [47, 48]. Here we use the latter method and first solve a one dimensional problem along each edge of \( K \). For \( \phi_i \) in \( K \), the pressure is set to 1 at node \( i \) and zero elsewhere for the one dimensional solves. The one dimensional solutions are then used as Dirichlet conditions in the two dimensional solve for \( \phi_i \) on the interior of \( K \).

In practice, even these boundary conditions can introduce error because of a lack of scale separation in the permeability field. Errors may still exist, coming into play when the coarse element size resonates with length scales of the permeability. Overcoming this requires the use of limited global information which can require a global fine scale solve. See [42] for a detailed discussion of global information in MsFEM as well as a thorough error analysis based on homogenization theory.
2.2 Multiscale Sampling methods

So far, this chapter has focused solely on multiscale solution strategies for the pressure equation. However, using multiscale strategies within a sampling method such as MCMC can also be advantageous. This section will give a brief overview of current multiscale MCMC methods and point out where there is room for improvement.

2.2.1 Coupled Metropolis Sampling

Following the derivations in [41] and [24], consider a case where we have two representations of the permeability field, $k_1$ is a fine field, and $k_2$ is a coarsened field. The fine scale field is donned with a Gaussian prior:

$$k_1 \sim N(\mu, \Sigma_1)$$

Furthermore, a linear equation is used to transfer information from the fine scale $k_1$ to the coarse field $k_2$:

$$k_2 = A_1 k_1$$

The linear link $A_1$, usually represents some type of linear average, [24]. As already mentioned in the upscaling section, a linear relationship between fine and coarse permeability fields does not satisfy the problem physics. This is not a killer in the coupled Metropolis setting discussed in [24]. In that case, a linear map is necessary and any errors in the mapping are alleviated by introducing an additive error term to the upscaling. The new multiscale inference procedure outlined in the next chapter avoids this issue by allowing for a nonlinear relationship between scales.

In a two scale problem, the multiscale Metropolis method proposed by Higdon in [41] has two MCMC chains simultaneously exploring the posterior distributions of $k_1$ and $k_2$. In porous media, each scale will have a likelihood distribution, $\pi(d|k_1)$ and $\pi(d|k_2)$, where $d$ are pressure observations. The coarse likelihood will require a simulation using a coarse discretization of the pressure equation with permeability $k_2$. Similarly, the fine scale likelihood will require a more expensive pressure equation
solve using a fine grid with $k_1$. Occasionally, the chains will swap information through an interscale swap proposal, $q(k'_1, k'_2|k^n_1, k^n_2)$. The proposal for $k_2$ is simply taken as the linear upscaling, $Ak_1$. Furthermore, $k_1$ and $k_2$ are assumed to both have normal densities, so the fine proposal will simply be the prior for $k_1$ conditioned on $Ak_1 = k_2$.

So, we have:

$$q(k'_1, k'_2|k^n_1, k^n_2) = q(k'_1|Ak_1 = k^n_1)q(k'_2|k_2 = Ak_1)$$  (2.29)

The density $q(k'_2|k_2 = Ak_1^n)$ is a Dirac measure at $Ak_1^n$. Using this proposal in the Metropolis-Hastings rule gives

$$\alpha = \min \left\{ 1, \frac{\pi(d|k'_2)\pi(k'_2)\pi(d|k'_1)\pi(k'_1)q(k^n_1, k^n_2|k'_1, k'_2)}{\pi(d|k_2)\pi(k_2)\pi(d|k_1)\pi(k_1)q(k'_1, k'_2|k^n_1, k^n_2)} \right\}$$  (2.30)

Figure 2-7 shows a graphical representation of the Coupled Metropolis MCMC.

Figure 2-7: Illustration of the Multiscale Metropolis MCMC method. Here, steps labeled with MCMC correspond to steps with standard single scale proposals. It is also possible to have more than one MCMC step before a swap is performed.

Multiscale Metropolis increases MCMC effectiveness through better mixing of the fine scale MCMC chain and can help the chain explore more of the fine scale parameter space. However, each step of the chain still requires a fine scale simulation to evaluate the fine scale likelihood. Like computing MsFEM basis functions offline for a well-field design problem, we would like to avoid any fine scale simulations beyond offline preprocessing. The basic idea behind Metropolis-Coupled MCMC is that a coarse representation of $k$ is easier to sample because it reduces the problem dimension. This is useful concept that will be expanded later to the point where knowing the coarse quantities alone is sufficient to predict the pressure and no fine scale solves are needed during the MCMC routines.


2.2.2 Proposal Filtering with MsFEM

Taking one step closer to avoiding fine scale simulation altogether is the work on two stage proposals in [16] and expanded to Langevin type proposals in [21]. Conceptually, these methods use a coarse multiscale finite volume (MsFV) simulation to filter out bad proposals before any fine scale simulations need to be performed. The papers are focused on saturation simulations, so the major computational cost is in solving fine scale saturation equations. Thus, a coarse MsFV pressure is used with a crudely averaged coarse saturation solve. While some errors in the coarse solve exist, the information is still valuable for filtering out bad proposal points. Consider a coarse forward model $G^*(k)$ and a likelihood function $\pi^*(k')$ that depends only on outputs from the coarse forward model. Now, let $k'$ be a proposal from a standard Gaussian proposal mechanism. During a preconditioning step, this proposal is accepted as the true proposal with probability

$$\alpha^*(k^n, k') = \min \left\{ 1, \frac{q(k'n|k')\pi^*(k')}{q(k'|k^n)\pi^*(k^n)} \right\}$$

(2.31)

where $q(k|k^n)$ is the usual proposal density. This means that the final, fine scale, proposal comes from the density:

$$Q(k|k^n) = \alpha^*(k^n, k)q(k|k^n) + \left( 1 - \int \alpha(k^n, k)q(k|k^n)dk \right) \delta_{k^n}(k)$$

(2.32)

Using this proposal in the Metropolis-Hastings rule gives a final acceptance probability of

$$\alpha = \min \left\{ 1, \frac{Q(k^n|k)\pi(k)}{Q(k|k^n)\pi(k^n)} \right\}$$

(2.33)

The advantage of this approach is that when $k'$ is rejected in the initial stage, no fine scale saturation solve is necessary and a significant computational savings is experienced. In [21], this procedure is generalized to a Langevin MCMC case where the posterior gradient is approximated using only coarse scale solves. Unfortunately, the advantages in these methods are not as useful in the steady state single flow scenario considered here because some fine scale simulations are still required after
a proposed point passes the initial rejection stage. As will be shown in the next chapter, when only considering the pressure equation, reasonable assumptions and some offline preprocessing allow for a complete decoupling of the fine and coarse scales. This simultaneously reduces the MCMC parameter dimension and the computational expense of a forward evaluation.
Chapter 3

Multiscale inference framework

3.1 Conditional Independence

For steady state single phase flow based on the pressure equation, solving for pressure with MsFEM can be broken into two stages. In the first stage, local solves are used to build a coarse scale stiffness matrix and in the second stage, this stiffness matrix is used to compute the pressure. If desired, the coarse pressure can also be projected onto the fine scale using the same basis functions. In the forward problem, information moves from permeability to coarse stiffness matrix, to pressure. Thus, when solving the inference problem, information should travel from pressure observations, through the coarse scale stiffness matrix, to the permeability field. The previous work in [16, 21] discussed above, takes partial advantage of this to pre-reject bad proposals based only on a coarse approximation. Here we take further advantage of the MsFEM structure to eliminate the need for any global fine scale solves. We focus on a static inference problem where steady state pressure observations, $d$, are given. The goal is then to characterize a permeability field on the fine scale, $k$, conditioned on the data. That is, we wish to find the density $\pi(k|d)$. As usual Bayes’ rule will be used to obtain the posterior in terms of a likelihood and prior:

$$\pi(k|d) \propto \pi(d|k)\pi(k)$$  \hspace{1cm} (3.1)
In the MsFEM setting, the intermediate coarse scale stiffness matrix can also be included in the inference. Thus, the joint posterior of the stiffness matrix $A$ and the permeability $k$ can be represented as:

$$
\pi(k, A|d) \propto \pi(d|k, A)\pi(k, A)
$$

Here, the notation $\pi(k, A|d)$ refers to the joint distribution of the permeability values and the entries in the stiffness matrix. Note that in some instances, the number of variables representing $A$ can be reduced by considering the distribution of elemental integrals. This topic will be discussed later. Expanding the joint prior with the law of total probability gives

$$
\pi(k, A|d) \propto \pi(d|k, A)\pi(k|A)\pi(A) \quad (3.2)
$$

At this point it may seem awkward to place a prior, $\pi(A)$, on the stiffness matrix entries but we will see that this is a critical step in decoupling the fine and coarse scales. Now, concentrate on the likelihood term $\pi(d|k, A)$. This is the probability of the data given both the permeability and the coarse scale stiffness matrix. However, as we know from MsFEM, only the stiffness matrix is needed to construct the pressure, $p$. This is an important observation: when using MsFEM, the stiffness matrix $A$ is “sufficient” for $p$. Sufficiency is an important notion in statistics; see [29] for more information or [15] for a discussion in the context of Jeffreys’ conditioning. Since $d$ is an observation of $p$, we have that $A$ is sufficient to describe $d$. Probabilistically, this idea corresponds to the conditional independence of $d$ and $k$ given $A$:

$$
\pi(d, k|A) = \pi(d|A)\pi(k|A) \quad (3.3)
$$

which implies

$$
\pi(d|k, A) = \pi(d|A) \quad (3.4)
$$
Using this in (3.2) gives

$$\pi(k, A|d) \propto \pi(d|A)\pi(k|A)(A)$$

(3.5)

Figure 3-1 validates the conditional independence assumption. 500000 samples were generated from a prior on $k$ and the corresponding entries of the stiffness matrix and pressure were found. In the joint covariance, three regions are marked. The first columns and last rows are the entries in the stiffness matrix, the next group inward is the pressure, and the large section filling the upper right is the permeability. In the joint covariance, faint correlations can be seen between all the fields. However, looking at the joint conditional distribution $\pi(k, p|A^*)$, no significant correlations are seen between the pressure and permeability. In order to perform the conditioning, a particular stiffness matrix, $A^*$, needed to be used. Several choices were tested and all produced similar results. Note the pressure in figure 3-1 was computed with a fine scale solve. Similar results for MsFEM show that this is not a feature of the solver being used, but a consequence of $A$ being sufficient to describe $p$.

![Joint Covariance](image1)

![Conditioned Covariance](image2)

Figure 3-1: Validation of conditional independence assumption.
3.2 Multiscale MCMC formulation

In order to characterize the new target distribution $\pi(k, A|d)$, MCMC will be used. In order to sample the joint density, a joint proposal, $q(k', A'|k, A)$ will be used. In this setting the Metropolis-Hastings acceptance probability becomes:

$$\alpha = \min\{1, \gamma\}$$

where

$$\gamma = \frac{\pi(d|k', A')\pi(k', A')q(k, A|k', A')}{\pi(d|k, A)\pi(k, A)q(k', A'|k, A)}$$

$$= \frac{\pi(d|A')\pi(k'|A')\pi(A')q(k|A)q(A|A')}{\pi(d|A)\pi(k|A)\pi(A)q(k'|A')q(A'|A)} \quad (3.6)$$

The second equality comes from simply expanding the joint distributions into conditionals and using the conditional independence assumption. This equation looks much more difficult to use than the usual expression when we want $\pi(k|d)$. This equation has both coarse stiffness matrix terms and permeability terms. However, if we could sample exactly from $\pi(k|A)$, the corresponding portion of the proposal could be exact and major simplifications would ensue. Assume therefore

$$q(k|A) = \pi(k|A)$$

then, after some simplification, we have

$$\gamma = \frac{\pi(d|A')\pi(A')q(A|A')}{\pi(d|A)\pi(A)q(A'|A')} \quad (3.7)$$

which is nothing more than a standard Metropolis-Hastings rule in just the low dimensional $A$! Using this acceptance probability, the relatively low dimensional coarse scale stiffness matrix could be inferred without ever needing the high dimensional fine scale field, $k$. Alas, there is no free lunch. The two major assumptions made during this derivation need to be addressed. Specifically,
• a means of evaluating the prior density for the entries in $A$ needs to be found.

• a method to sample from the conditional distribution $\pi(k|A)$ needs to be developed

A prior on $A$ is needed to perform the coarse sampling. Additionally, sampling from $\pi(k|A)$ will allow posterior realizations of $k$ to be generated from each sample of the stiffness matrix. The proposed multiscale sampling strategy is summarized in figure 3-2. In this figure, data is used with a prior on $A$ to generate posterior samples with standard MCMC tools such as DRAM. Then, a nonlinear iterative conditioning procedure is used to generate posterior samples of $k$ for each sample of $A$. The next section will discuss how a prior for $A$ can be developed offline from a prior on $k$. The following section will then introduce a new nonlinear conditioning approach that will allow samples of $\pi(k|A)$ to be generated. Numerical results and analysis will be left to the next chapter.

\[
D = p(d|A) \\
MCMC \rightarrow A^{(i)} = \text{Nonlinear conditioning} \\
\]

Figure 3-2: Illustration of full multiscale inference using procedure using MsFEM.

### 3.3 Upscaling the Prior

The prior distribution on the permeability field is derived from expert opinions, apriori site characterizations, and an overall geologic understanding of the medium. However, this knowledge does not directly transfer to a prior on $A$. It is not clear how a
knowledge of sediment formations will affect $A$. Consider the case when $k$ is a log-normal random process. That is $k = \exp[Y]$ where $Y \sim GP(\mu_Y, \Sigma_Y)$. The covariance of $Y$ at two locations from a two point covariance kernel. The kernel maps the spatial distance between two permeability regions to a correlation. Multiplying by a stationary variance then gives the covariance. A general covariance kernel may follow an exponential form:

$$
\text{Cov}[Y(x_i), Y(x_j)] = \sigma \exp \left[ -\frac{1}{p} \left( \frac{\|x_i - x_j\|^2}{L} \right) \right] \tag{3.8}
$$

where $p$ is a power parameter, $L$ is a length scale, and $\sigma$ is the variance. When $p = 1$ we have an exponential kernel, and $p = 2$ is a Gaussian kernel. Exponential kernels tend to be much rougher than Gaussian kernels and usually need to be represented on a finer mesh. Representing the prior with a covariance kernel such as this is convenient; only 3 parameters and the mean are needed to completely describe a high dimensional distribution. This section attempts to demonstrate that with some offline Monte Carlo preprocessing, a reference mapping can be built that takes the kernel parameters for $Y$ to parameters describing a log-normal distribution for the elemental integrals. In this setting, $Y$ will be a discrete dimensional field described by a Multivariate Gaussian distribution. This is a simply a discretization of Gaussian process definition. Begin by considering a one dimensional Petrov-Galerkin formulation of MsFEM on a one dimensional mesh with nodes at $x_0, x_1, ...$. Construction of the coarse scale stiffness matrix in this case is given by:

$$
a_{ij} = \int_{x_{i-1}}^{x_i} k(x) \frac{\partial \phi_i}{\partial x} \frac{\partial \phi_j^0}{\partial x} dx + \int_{x_i}^{x_{i+1}} k(x) \frac{\partial \phi_i}{\partial x} \frac{\partial \phi_j^0}{\partial x} dx \tag{3.9}
$$

where $\phi_i$ is the MsFEM basis function defined as 1 at node $i$ and 0 at $x_{i-1}$ and $x_{i+1}$. Within these two elements, $\phi_i$ satisfies:

$$
\frac{\partial}{\partial x} \left( k(x) \frac{\partial \phi_i}{\partial x} \right) = 0
$$
so
\[ k(x) \frac{\partial \phi_i}{\partial x} = c \]
for some constant \( c \). Dividing both sides by \( k(x) \) and integrating over \([x_{i-1}, x_i]\) we see that

\[
\int_{x_{i-1}}^{x_i} \frac{\partial \phi_i}{\partial x} \, dx = \int_{x_{i-1}}^{x_i} \frac{c}{k(x)} \, dx
\]

\[
\Rightarrow \phi_i |_{x_{i-1}}^{x_i} = \int_{x_{i-1}}^{x_i} \frac{c}{k(x)} \, dx
\]

\[
\Rightarrow c = \frac{1}{\int_{x_{i-1}}^{x_i} \frac{1}{k(x)} \, dx}
\]

(3.10)

An equivalent process can be repeated over \([x_i, x_{i+1}]\) to obtain

\[
a_{ij} = \int_{x_{i-1}}^{x_i} \frac{1}{k(x)} \, \partial \phi_i^0 \, dx + \int_{x_i}^{x_{i+1}} \frac{1}{k(x)} \, \partial \phi_j^0 \, dx
\]

\[
= \frac{1}{\int_{x_{i-1}}^{x_i} \frac{1}{k(x)} \, dx} \int_{x_{i-1}}^{x_i} \partial \phi_i^0 \, dx + \frac{1}{\int_{x_i}^{x_{i+1}} \frac{1}{k(x)} \, dx} \int_{x_i}^{x_{i+1}} \partial \phi_j^0 \, dx
\]

(3.11)

Using nodal linear functions for the test basis functions \( \phi_i^0 \) simplifies this expression to:

\[
a_{i,i-1} = - \frac{1}{\int_{x_{i-1}}^{x_i} \frac{1}{k(x)} \, dx}
\]

(3.12)

\[
a_{i,i+1} = - \frac{1}{\int_{x_i}^{x_{i+1}} \frac{1}{k(x)} \, dx}
\]

(3.13)

\[
a_{i,i} = \frac{1}{\int_{x_{i-1}}^{x_i} \frac{1}{k(x)} \, dx} + \frac{1}{\int_{x_i}^{x_{i+1}} \frac{1}{k(x)} \, dx}
\]

(3.14)

Denote the coarse element from \( x_{i-1} \) to \( x_i \) as element \( i \), then the elemental integral for element \( i \) is given by:

\[
e_i = \frac{1}{\int_{x_{i-1}}^{x_i} \frac{1}{k(x)} \, dx}
\]

(3.15)

Note that the stiffness matrix entries are composed of the elemental integrals. Clearly, \( a_{i,i} = e_i + e_{i+1}, a_{i,i-1} = -e_i, \) and \( a_{i,i+1} = -e_{i+1}. \) Thus, a prior on the stiffness matrix, \( \pi(A) \) is equivalent to having a prior on the elemental integrals: \( \pi(e) \). This holds in
one dimension, in higher dimensions more care is required.

![Figure 3-3: Quantile-Quantile plot comparing sample quantiles to true log-normal quantiles. The distribution is an exact fit when the scatter points are exactly linear.](image)

Let $k$ be defined as a piecewise constant field in one dimension following a log-normal distribution, $k = \exp[Y]$ for $Y \sim N(\mu_Y, \Sigma_{YY})$. Furthermore, let $\Sigma_{YY}$ be described through an exponential covariance kernel of the form in (3.8). Taking 400,000 Monte Carlo of $Y$ and computing $e$ with (3.15) shows that $e$ is approximately joint log-normally distributed as well. Figures 3-4 and 3-5 show the joint distribution of the elemental integrals when 5 coarse elements were used with 20 fine elements for each coarse element for a total of 100 fine elements over $[0, 1]$. The fitted covariance kernels are also shown. The densities pass the eyeball test of normality, figure 3-3 also shows that these $e$ are well approximated by log-normal distribution. Through the majority of parameter space, the sample quantiles match what is expected of a log-normal density. However, the tail probabilities are off, as you would expect in an approximation and with a finite number of samples. Each color corresponds to a different elemental integral. The plot only shows a few dimensions but similar results are found in all dimensions.

Therefore, for a prior parameterized by a given length scale, power, and variance, the distribution of elemental integrals can be well approximated by a log-normal distribution of a similar form to the prior. The fitted parameters for $\pi(\log(e))$ were found using maximum likelihood estimation. The optimization was done with a
Figure 3-4: Distribution of $\log(e)$ based on Monte-Carlo sampling of a piecewise constant log-normal $k$ field with exponential prior.

A simple implementation of the Levenberg-Marquadt algorithm with constant Marquadt parameter $\lambda$.

It is infeasible to perform the Monte Carlo sampling and fitting each time a prior for $e$ is needed. However, using a database of previously run samples provides an empirical way to map the parameters describing $\pi(k)$ to parameters characterizing...
Figure 3-5: Distribution of $\log(c)$ based on Monte-Carlo sampling of a piecewise constant log-normal $k$ field with Gaussian prior.

$\pi(e)$. All the work to build the database can be done before an inference is required and reused each time the algorithm is run. Performing the same fitting procedure used in figures 3-4 and 3-5, the surfaces in figure 3-6 were constructed. Some sampling error is visible, especially in the power plots. These mappings are critical to this multiscale framework; given the parameters of the prior, the corresponding parameters of the elemental integrals can be trivially found. Distances between elemental integrals are the distances between the center of each coarse element. It may seem unusual
to represent a vector quantity $e$ as a spatially dependent parameter. However, in a case where $k$ was a continuous random process, the elemental integrals could be interpreted as the reciprocal of a moving average of $\frac{1}{k(x)}$ over a length $\delta x = x_i - x_{i-1}$. The elemental integrals used in the coarse stiffness matrix are then snapshots of a continuously varying random process.

Figure 3-6 shows the coarse length scale is only a function of the fine correlation length. For significant correlation lengths, the coarse variance is also only a function of the fine variance. However, as the fine scale correlation length $d_f$ goes to zero, so does the coarse variance. Conceptually, this is expected. When $\sigma_f \rightarrow 0$, $Y$ becomes white noise on top of $\mu_Y$, so integration on each coarse element will give approximately the same quantity, and $\text{var}[\log(e)] \rightarrow 0$.

### 3.3.1 Proof of positive definiteness in one dimension

Using the information in figure 3-6, a coarse scale prior distribution is defined in terms of the prior on $k$. Using the log-normal $\pi(e)$, samples of the stiffness matrix can easily be generated. It can also be shown that these samples of $A$ are guaranteed to be positive definite in the one dimensional setting. Here we will show that in one dimensional, the coarse scale stiffness matrix is identical to a matrix derived with linear basis functions and a coarse representation of $k$. Thus, coming from a standard finite element discretization of an SPD operator, we conclude that by construction, any sample of $\pi(e)$ will create a SPD stiffness matrix $A$.

Using linear basis functions on the coarse grid in a standard Galerkin projection; the following elemental integrals, $\tilde{e}$, are created:

$$ \tilde{e}_i = \int_{x_{i-1}}^{x_i} \tilde{k}(x) \frac{\partial \phi_0^i}{\partial x} \frac{\partial \phi_0^{i-1}}{\partial x} \, dx $$

(3.16)

Note that a different permeability field $\tilde{k}$ was used here. The field $\tilde{k}$ is assumed to be piecewise constant on the coarse grid, whereas the field $k$ used in the MsFEM formulation is piecewise constant on the fine grid. From standard finite element theory, using (3.16) is guaranteed to produce a symmetric positive definite stiffness
Figure 3-6: Mapping from $\pi(Y)$ to $\pi(\log(e))$ for exponential and Gaussian covariance kernels in $\pi(Y)$. The fine scale correlation length, $L_f$, has been scaled by the coarse element size $h$ to allow for arbitrary discretizations. Notice the $x$ axis of the power plots is in the opposite direction to the other plots.

matrix when $\tilde{k} > 0$. Thus, to guarantee $A$ will be symmetric positive definite for any sample of $\pi(e)$ it suffices to show there exists a field $\tilde{k}$ such that $\tilde{e}_i = d_i$ for any $e$. 

78
Proof Let $e_i^*$ be a sample of $\pi(e)$. On each element, choose

\[ \tilde{k}_i = -h^2 e_i^* \]  

where $h = x_i - x_{i-1}$. Then

\[ \tilde{e}_i = \int_{x_{i-1}}^{x_i} -h^2 e_i^* \frac{\partial \phi_i^0}{\partial x} \frac{\partial \phi_i^0}{\partial x} \, dx \]

\[ = -h^2 e_i^* \int_{x_{i-1}}^{x_i} \frac{\partial \phi_i^0}{\partial x} \frac{\partial \phi_i^0}{\partial x} \, dx \]

\[ = e_i^* h^2 \]  

The last step comes from the fact that $\phi_i^0$ and $\phi_{i-1}^0$ are linear with slopes $\frac{1}{h}$ and $-\frac{1}{h}$ respectively. Thus, the MsFEM stiffness matrix from any sample of $\pi(e)$ is equivalent to a coarse Galerkin discretization of the pressure equation on a permeability field $\tilde{k}$ and is guaranteed to be symmetric positive definite.

In higher dimensions, it is not clear whether a similar proof can be found. Currently, our only solace is that a proposal of $e$ resulting in a non-SPD stiffness matrix may result in an inaccurate pressure and have a very small probability of being accepted.

3.4 Iterative Nonlinear Conditioning

At first glance, generating a sample of $\pi(k|A)$ may seem just as difficult as sampling a posterior. Indeed, using Bayes’ rule would give

\[ \pi(k|A) \propto \pi(A|k)\pi(k) \]

which is no easier to compute than $\pi(k|d)$. However, for each sample $A^{(i)}$, we are conditioning on a point density, which means that $\pi(k|A)$ will only have positive measure on a low dimensional manifold constrained by $A^{(i)}$. The constraint comes from the nonlinear relationship map between $k$ to $A$. Sampling from $\pi(k|A)$ can be seen as sampling from a slice of the prior on $k$. Figure 3-7 illustrates this idea.
Ultimately, the goal is to characterize $k$ through $A$. To do this, it will be convenient to work with Gaussian distributions; however, $k$ and $A$ are distributed log-normally. Thus, we will instead consider $Y = \log(k)$ and $W = \log(e)$ in the following analysis.

### 3.4.1 Linear conditioning

Let $F(Y)$ be the mapping from $Y$ to $W$. Assume for the moment that this mapping is linear: $F(Y) = FY$. In this case, the joint distribution of $Y$ and $W$ is given by

$$
\begin{bmatrix}
W \\
Y
\end{bmatrix} \sim N
\begin{pmatrix}
F\mu_Y \\
\mu_Y
\end{pmatrix},
\begin{pmatrix}
F\Sigma_{YY}F^T & F\Sigma_{YY} \\
\Sigma_{YY}F^T & \Sigma_{YY}
\end{pmatrix}
$$

(3.21)

The dimension of $W$ is equal to the number of coarse elements in the system and will be significantly smaller than the dimension of $Y$, which is equal to the total number of fine elements. The goal here is to express $Y$ in terms of $W$. To do this, we can use the Schur complement of the joint covariance, which can also be viewed here as a block Gaussian elimination. We obtain:

\begin{align}
\mathbb{E}[Y|W] &= \mu_Y + \Sigma_{YY}F^T \left( F\Sigma_{YY}F^T \right)^{-1} (W - F\mu_Y) \quad (3.22) \\
\text{Cov}[Y|W] &= \Sigma_Y - \Sigma_{YY}F^T \left( F\Sigma_{YY}F^T \right)^{-1} F\Sigma_{YY} \quad (3.23)
\end{align}
which is more commonly written as

\[ \mathbb{E}[Y|W] = \mu_Y + K(W - F\mu_Y) \] (3.24)

\[ \text{Cov}[Y|W] = \Sigma_{YY} - K F \Sigma_{YY} \] (3.25)

where \( K = \Sigma_{YY} F^T (F \Sigma_{YY} F^T)^{-1} \) is often called the Kalman gain. Clearly, in the linear Gaussian case a simple analytic form exists for \( \pi(Y|W) \) in terms of the prior mean and covariance. Alternatively, independent samples of \( \pi(Y|W) \) can be generated by taking samples of the prior, and using a modified version of (3.22) to move the sample onto the constraint. The moments of the moved samples are guaranteed to converge to the analytic mean and covariance using the linear update equation:

\[ Y^*_a = Y^* + \Sigma_{YY} F^T (F \Sigma_{YY} F^T)^{-1} (W - FY^*) \] (3.26)

The following proof shows that sampling from the prior and then moving those samples to the constraint with (3.26) will converge in distribution to \( \pi(k|A) \).

**Proof** Let \( Y \sim N(\mu, \Sigma) \) be a multivariate Gaussian random variable with dimension \( N \). Given a linear operator \( F \) of size \( M \times N \) with \( M \leq N \), define the multivariate Gaussian \( W = FY \). Thus, \( \mathbb{E}[Y|W = b] \) is defined in (3.22) and \( \text{Cov}[Y|W = b] \) is given by (3.23). Now, consider the linear operation:

\[ Z = Y + F \Sigma (F \Sigma F^T)^{-1} (b - FY) \]

where \( Z \) is a multivariate Gaussian random variable of dimension \( N \). The expectation of \( Z \) is:

\[ \mathbb{E}[Z] = \mathbb{E}[Y + F \Sigma (F \Sigma F^T)^{-1} (b - FY)] \]

\[ = \mathbb{E}[Y] + F \Sigma (F \Sigma F^T)^{-1} \mathbb{E}[(b - FY)] \]

\[ = \mathbb{E}[Y] + F \Sigma (F \Sigma F^T)^{-1} (b - F \mathbb{E}[Y]) \]

\[ = \mu + F \Sigma (F \Sigma F^T)^{-1} (b - F \mu) \]
and the covariance of $Z$ is:

$$\text{Cov}[Z] = \mathbb{E}[(Z - \mathbb{E}[Z])(Z - \mathbb{E}[Z])^T]$$

$$= \mathbb{E}
\left[
(Y + F\Sigma(F\Sigma F^T)^{-1}(b - FY) - \mu - F\Sigma(F\Sigma F^T)^{-1}(b - F\mu))
\times
(Y + F\Sigma(F\Sigma F^T)^{-1}(b - FY) - \mu - F\Sigma(F\Sigma F^T)^{-1}(b - F\mu))^T
\right]$$

$$= \mathbb{E}
\left[
(Y - \mu - F\Sigma(F\Sigma F^T)^{-1}FY + F\Sigma(F\Sigma F^T)^{-1}F\mu)
\times
(Y - \mu - F\Sigma(F\Sigma F^T)^{-1}FY + F\Sigma(F\Sigma F^T)^{-1}F\mu)^T
\right]$$

$$= \mathbb{E}
\left[
(Y - (\mu + F\Sigma(F\Sigma F^T)^{-1}(FY - F\mu)))
\times
(Y - (\mu + F\Sigma(F\Sigma F^T)^{-1}(FY - F\mu)))^T
\right]$$

$$= \mathbb{E}
\left[
(Y - \mathbb{E}[Y|FY])
\times
(Y - \mathbb{E}[Y|FY])^T
\right]$$

$$= \text{Cov}[Y|FY]$$

The mean and covariance match the mean and covariance of $W$. Therefore, since the mean and covariance of a multivariate Gaussian completely describe the distribution, $Z \overset{d}{=} W$. This implies sampling from $Y$ and then updating the samples is equivalent to sampling from $W$ since the update is linear.

### 3.4.2 Nonlinear conditioning

The idea of taking a sample from the prior and moving it to a representative location is not unique to this work. This concept is also fundamental to the Ensemble Kalman Filter (EnKF) [22]. In a dynamic problem, the EnKF works by running several samples (an ensemble) through the forward model until an observation time is reached. Then the observation is incorporated by using (3.26) on the entire ensemble. The samples are then again propagated through the forward model until another observation is reached. The basic EnKF assumes a linear observation function.\(^1\) That is, even if the forward simulation is nonlinear, the relationship between simulation output and the observational data is linear. Interestingly, sampling $\pi(k|A)$ is identical

\(^1\)Additionally, the EnKF assumes Gaussianity of the ensemble members when incorporating observations.
to incorporating observations in the EnKF with a nonlinear observation function. To surmount the nonlinearity, we follow years of mathematical tradition and simply approximate the nonlinear $F(Y)$ with a linearization around $Y_0$:

$$F(Y) \approx F(Y_0) + J(Y_0)(Y - Y_0) \quad (3.27)$$

where $J(Y_0)$ is the Jacobian matrix of $F(Y)$ evaluated at $Y_0$. The constraint, $F(Y) = W$ can now be approximated with

$$F(Y_0) + J(Y_0)(Y - Y_0) = W \quad (3.28)$$

$$\Rightarrow J(Y_0)Y = W + J(Y_0) - F(Y_0) \quad (3.29)$$

and a linear update can be performed:

$$Y_1 = Y_0 + \Sigma_{YY}J^T(J\Sigma_{YY}J^T)^{-1}(W + JY_0 - F(Y_0) - JY_0)$$

$$= Y_0 + \Sigma_{YY}J^T(J\Sigma_{YY}J^T)^{-1}(W - F(Y_0)) \quad (3.30)$$

where $J = J(Y_0)$. In this situation, $Y_1$ is an approximation to a sample of $\pi(Y|W)$. The linear approximation gets $Y_1$ closer to satisfying $F(Y) = W$, but will not in general satisfy the constraint. To obtain a better approximation, the linearization process is repeated at $Y_1$ and another linear conditioning step can be performed. This iteration is repeated until a point $Y_n$ satisfying $\|F(Y_n) - W\|_2 < \epsilon$ is reached for some small $\epsilon$. Linearization errors are significant in this process and only an approximate sample of $\pi(k|A)$ will be generated. The errors can be corrected after the constraint is reached through additional measures such as MCMC. However, samples more closely representing $\pi(k|A)$ will require less correction and less MCMC will be required. Use of MCMC is discussed more thoroughly in later sections. For now, the focus remains on approximately getting to the constraint.

Several variations of the simple iterative linearization introduced above can be used. In fact, the linear update step can be viewed as a forward Euler discretization
of the nonlinear ordinary differential equation

\[
\frac{\partial y}{\partial t} = \Sigma_{YY} J^T(y) (J(y) \Sigma_{YY} J^T(y))^{-1} (W - F(y)) \quad (3.31)
\]

with initial condition \(y(t = 0) = Y_0\). By using higher order integration techniques, the linearization error can be reduced. In fact, as the timestep \(\Delta t \to 0\), the high order terms originally truncated in (3.28) become negligible and virtually no approximation error exists. However, computational costs will require the use of a more moderate \(\Delta t\), and some linearization errors will undoubtedly be introduced.

Using high order methods to integrate is similar to the mollified ensemble Kalman filters, where data is incorporated into the system gradually using a continuous time derivation of the EnKF[7]. Another related concept is the idea of nonlinearly constrainted Kalman Filters. Sampling from \(\pi(Y|W)\) is nearly identical to the EnKF update equation in that setting. The nonlinear constrained EnKF has been thor- oughly discussed in [26] and more recently in [80]. The idea proposed by Geeter et al. is to replace the hard constraint \(F(Y) = W\) with a series of weak constraints \(J(Y_{n-1})Y_n = W + \epsilon Y\) for some \(\epsilon \sim N(0, \eta_n \Gamma_0)\). Where

\[
\Gamma_0 = \alpha J \Sigma_{YY} J^T
\]

This follows from the fact that linearly conditioning a Gaussian random variable on observations with variance \(\sigma\) is equivalent to conditioning the random variable on the same observation 10 times with observation variance 10\(\sigma\). In the presence of this article noise, the update equation becomes

\[
Y_{n+1} = Y_n + \Sigma_{YY} J^T (J \Sigma_{YY} J^T + \eta \Gamma_0)^{-1} (W - F(Y_n)) \quad (3.32)
\]

Where the observation error \(\eta \Gamma_0\) is meant to roughly approximate the linearization error at each step. As the iteration process and \(F(Y_n)\) approaches \(W\), the incremental stepsize \(Y_{n+1} - Y_n\) will decrease and a decrease in linearization error will ensue. To
incorporate this into the artificial noise, [26] proposes shrinking $\eta$ with

$$\eta_n = \alpha \exp(-n)$$

In true EnKF fashion, a covariance is approximated at each step, replacing $\Sigma_{YY}$ in subsequent position updates. The complete method for approximately sampling $\pi(Y|W)$ is outlined in algorithm 2.

**Algorithm 2** Smoothly constrained Kalman filter sampling

**Require:** Prior distribution, $\pi(Y) \sim N(\mu_Y, \Sigma_{YY})$
- Error parameter, $\eta$
- Nonlinear constraint $F(Y) = W$, Stopping tolerance $\delta$.  
1: Draw a sample, $Y_0$ from $\pi(Y)$
2: Set initial covariance, $\Sigma_0 = \Sigma_{YY}$
3: Set initial error, $\Gamma_0 = \alpha J(Y) \Sigma_0 J^T(Y)$
4: $n = 0$
5: while $\|F(Y_n) - W\|_2 > \delta$ do
6: $\Gamma = \Gamma_0 \exp(-n)$
7: $n = n + 1$
8: $Y_{n+1} = Y_n + \Sigma_n J^T(Y_n) \left( J(Y_n) \Sigma_n J^T(Y_n) + \Gamma \right)^{-1} \left( W - F(Y_n) \right)$
9: $\Sigma_{n+1} = \Sigma_n - \Sigma_n J^T(Y_n) \left( J(Y_n) \Sigma_n J^T(Y_n) + \Gamma \right)^{-1} J(Y_n) \Sigma_n$
10: end while
11: Return $Y_{n+1}$ as approximate sample of $\pi(Y|W)$

Figure 3-8 shows a comparison of this smoothly constrained Kalman filter (SCKF) approach with a 4th-5th order adaptive Runge-Kutta integration of (3.4.2). The prior distribution is simply $N(0, I)$ in two dimensions with a constraint

$$y = 0.2x^3 - 0.2x + 1$$

Clearly, the SCKF method provides a better initial approximation compared to the Runge-Kutta solver. However, after a few hundred MCMC steps along the constraint, any error is erased and the samples represent the true posterior shown in black. In general, as the problem becomes more nonlinear, more MCMC correction will be needed to ensure the final sample represents $\pi(Y|W)$.

The iterative conditioning procedure can also be viewed as Monte Carlo integra-
Figure 3-8: Comparison of high order ODE integration with SCKF for sampling from a constraint. The densities are plotted over $x$.

For uncorrelated jointly Gaussian random variables, $x_1$ and $x_2$, marginalizing is equivalent to conditioning: $\pi(x_1|x_2) = \pi(x_1)$. This can be generalized to correlated random variables by introducing a linear transformation. Specifically, marginalization in linearly transformed coordinates is equivalent to conditioning in the original coordinates. This can be used to cast “sample and move” linear conditioning as Monte-Carlo integration. Begin by transforming $Y$ to an uncorrelated vector of random variables, $z$, with $Y = \mu_Y + Lz$ where $L$ is a matrix square root (e.g. Cholesky...
factorization) of $\Sigma_{YY}$

$$L^T L = V^T \Sigma_k V$$

The goal now is to find $\pi(z|FLz = W)$. Where $z \sim N(0, I)$. Consider the singular value decomposition of $FL$,

$$FL = USV^T$$

where $V$ contains right eigenvectors, $S$ is the diagonal matrix of singular values, and $U$ contains the left eigenvectors. Since $F$ has fewer rows than columns, some singular values will be zero. Using the SVD in the conditional density gives

$$\pi(z|FLz = W) = \pi(z|USV^Tz = W) = \pi(z|SV^Tz = U^TW)$$

(3.33)

Next, introduce the transformed coordinates, $x = V^Tz$. The new problem becomes to find $\pi(x|Sx = W')$ where $W' = U^TA$ and the prior on $x$ is

$$x \sim N(V^T\mu_k, V^TV)$$

which is equivalent to

$$x \sim N(V^T\mu_k, I)$$

Note that $x$ is uncorrelated and $S$ is diagonal. Also observe that rows of $S$ corresponding to the zero singular values are completely zero. Now, consider the set $Z$ of all nonzero rows of $S$. The conditional distribution can then be decomposed as

$$\pi(x|S'x = W'') = \prod_{i \notin Z} \pi(x_i)$$

(3.34)

where $x_i = \frac{W''}{s_i} \forall i \in Z$. Thus, conditioning in the original coordinates, $\pi(Y|W)$ is simply marginalizing over $x_i$ for $i \in Z$. Back to sampling from $\pi(Y|W)$. It is clear that for a linear map, sampling a $Y$ and moving it to $Y^*$ with (3.26) is a projection of the point onto a subspace. Many samples of $\pi(Y)$ can be mapped to $x$ space and then
moved to the constraint. This process eliminates all $Z$ directions. By elimination, we mean those coordinates have been marginalized out of the final distribution. The sample and move strategy can now be seen as a Monte-Carlo integration over the $Z$ directions. In the nonlinear case, the singular value decomposition of the Jacobian is not constant over the entire parameter space, so the integration directions will not be lines. Instead, the integration directions will be given by (3.4.2). Figure 3-9 illustrates the linear and nonlinear directions along which Monte-Carlo integration is taking place. Any sample of the prior that lands on one of these lines will move to the same location on the constraint. In the linear case, moving along a line to the constraint represents a projection onto the null space of $F$. The norm in this projection is defined by the prior covariance. Thus, moving along a line to a nonlinear constraint is a projection to the nullspace of $F(Y)$ using a warped nonlinear metric. In fact, this nonlinear projection might be related in some way to geodesics and Riemannian manifolds.

![Figure 3-9: Illustration of integration paths for nonlinear conditioning and linear conditioning](image)

Figure 3-9 also gives insight into why the nonlinear conditioning approximations in figure 3-8 show larger peaks than the true posterior. Exact integration of $\pi(Y)$ along each line in 3-9 with a numerical integration rule will give correct density evaluations along the constraint. However, moving random samples of the prior along
the integration path results in more samples where the integration paths converge towards one another, resulting in an overestimation of the density. By incorporating some artificial noise into the system the SCKF results in some diffusion, alleviating the impact of converging integration paths.

### 3.4.3 MCMC along a constraint

The approximate nonlinear conditioning approaches generate a point $Y^\ast$ that is an approximate sample of $\pi(Y|W)$. Linear approximation errors as well as the convergence of integration paths will need to be corrected by performing MCMC on the constraint itself. The constraint can be seen as a nonlinear manifold in a high dimensional space. Moving around such as manifold is non-trivial. Here, we instead consider a slight relaxation by replacing the constraint $F(Y) = W$ with $\|F(Y) - W\|_2 \leq \epsilon$ for a small tolerance $\epsilon$. A likelihood function for satisfying this constraint is given by

$$
\pi(W|Y) = \begin{cases} 
0 & \|F(Y) - W\|_2 > \epsilon \\
1 & \|F(Y) - W\|_2 \leq \epsilon 
\end{cases}
$$

(3.35)

Using Bayes’ rule, the density we need to sample is

$$
\pi(Y|W) \propto \pi(W|Y)\pi(Y)
$$

(3.36)

A naive Metropolis-Hastings or DRAM approach could be used here. However, for small $\epsilon$ these methods will mix poorly and many evaluations of $F(Y)$ will be needed. We can take advantage of the fact that more information is at our disposal then is typically available for Metropolis-Hastings or DRAM. During the nonlinear conditioning, Jacobian approximations to $F(Y)$ were used in an approximate linear conditioning step. During the MCMC, this same information will be available and can be used to build efficient proposal densities adapted to the local topology of $F(Y)$. In fact, an isotropic proposal covariance can be adapted to local structure through the same equations used for nonlinear conditioning. In the absence of noise, updating the
covariance $\Sigma = \sigma^2 I$ is done through

$$\Sigma_a = \Sigma - \Sigma J^T(Y_n) \left( J(Y_n) \Sigma J^T(Y_n) \right)^{-1} J(Y_n) \Sigma$$

Within the MCMC framework, $\Sigma$, should be chosen like a standard Metropolis-Hastings proposal. The updated covariance $\Sigma_a$ will then be tuned to local effects of $F(Y)$. In cases where large global correlations exists, that information should be incorporated into $\Sigma$. Only local adaptation will be introduced here.

The linear conditioning update causes $\Sigma_a$ to become rank deficient. Positive variance will only exist on the plane spanned by the rows of $J(Y_n)$. A Gaussian proposal with this covariance will result in a non-ergodic chain and also cripple the chain’s ability to move around any regions of the parameter space where $F(Y)$ has curvature in directions perpendicular to the rowspace of $J(Y_n)$. Thus, much like the SCKF and Levenberg-Marquadt algorithm, artificial noise is introduced into the system:

$$\Sigma_{prop} = \Sigma - \Sigma J^T(Y_n) \left( J(Y_n) \Sigma J^T(Y_n) + \lambda I \right)^{-1} J(Y_n) \Sigma \quad (3.37)$$

where $\lambda$ controls the width of the proposal in the nullspace of $J(Y_n)$. The choice of $\lambda$ is problem dependent. Generally, $\lambda$ should be less than $\epsilon$ to stay within the constraint tolerance but large enough to account for the linearization error and allow the chain to scoot around high curvature regions of $F(k)$. Figure 3-10 shows two examples of the conditioned proposal. In both cases, $\lambda = 0.01$. The gray outline is the isotropic Gaussian density, $\Sigma$ and the colored contours show the distribution updated with Jacobian information.

Table 3.1 shows the acceptance rates of the above MCMC algorithm using a cubic constraint with $Y \sim N(0, I)$ as a prior in two dimensions. The bold entries represent the proposal variance that minimizes error between the true density and nonlinear conditioned estimate. The estimate comes from iteratively conditioning and correction 500 samples of the prior. Note that even though smaller proposals have a higher acceptance rate, the larger proposals the entire parameter space more effectively. Additionally, for small $\lambda$, the MCMC iteration tended to get trapped in regions between
Figure 3-10: Illustration of conditioned proposals in two dimensions on a cubic constraint. In both cases, $\lambda = 0.01$.

Table 3.1: Comparison of acceptance rates for MCMC on cubic constraint with $\epsilon = 1e - 2$. The bold rates are the combinations that result in the smallest density reconstruction error from 500 nonlinear conditioning runs and 500 MCMC steps.

<table>
<thead>
<tr>
<th>Isotropic Variance, $\sigma$</th>
<th>1.00E-03</th>
<th>1.00E-02</th>
<th>1.00E-01</th>
<th>1.00E+00</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$ 5.00E-04</td>
<td>38.55%</td>
<td>31.92%</td>
<td>19.96%</td>
<td>0.0741</td>
</tr>
<tr>
<td>$\lambda$ 1.00E-03</td>
<td>32.04%</td>
<td>24.16%</td>
<td>16.35%</td>
<td>6.41%</td>
</tr>
<tr>
<td>$\lambda$ 5.00E-03</td>
<td>24.41%</td>
<td>13.01%</td>
<td>9.25%</td>
<td>4.10%</td>
</tr>
</tbody>
</table>

high curvature. The linear approximation used in the proposal creation, prevented the chain from moving through these regions in a finite number of samples, even though asymptotic convergence to $\pi(Y \mid \| F(Y) - W\|_2 < \epsilon)$ is guaranteed.

### 3.5 Multiscale framework overview

A summary of the entire multiscale inference framework presented here is given in algorithm 3. Each iteration of the procedure is more computationally expensive than a single MCMC iteration. However, $W$ and $e$ have many fewer dimensions than $Y$ and $k$. Furthermore, less correlation exists in $W$ compared to $Y$. These two changes allow MCMC to more efficiently sample the posterior. Furthermore, the majority of the computational cost in the multiscale framework comes from the iterative nonlinear conditioning, which is embarrassingly parallel. Once MCMC has generated samples of
all samples of $k$ can be generated simultaneously. More precisely, no communication is needed between iterations of the loop in algorithm 3. When sufficient processing power is available, a new thread can be created for each step of the coarse MCMC chain to perform the nonlinear conditioning. It is also possible to only generate a fine scale realization for every $k$ MCMC steps. This can be useful when a user sees the MCMC chain is not mixing quickly and has limited computational resources.

**Algorithm 3** Basic summary of multiscale inference algorithm

**Require:** Prior distribution, $\pi(Y) \sim N(\mu_Y, \Sigma_{YY})$ and $\pi(W) \sim N(\mu_W, \Sigma_{WW})$

- Pressure observations $d$
- Number of samples $N_s$
- Nonlinear mapping from $Y$ to $W$, $F(Y) = W$

1: Generate $N$ samples of $\pi(W|d)$, $W^1, W^2, ..., W^N$ using DRAM

2: for $i = 1...N$ do
3: Use SCKF or ODE integration to obtain $Y^i$ satisfying $F(Y^i) = W^i$
4: Correct $Y^i$ with $M$ steps of MCMC on the constraint
5: end for
6: Return $\exp(Y^1), ..., \exp(Y^N)$ as samples of $\pi(k|d)$

### 3.6 Implementation details

#### 3.6.1 Jacobian in One Dimension

To perform nonlinear conditioning, the Jacobian of $F(Y)$ is needed. Recall the mapping

$$W_i = \log \left[ \frac{1}{\int_{x_{i-1}}^{x_i} \exp(-Y)dx} \right]$$

Let $Y$ be piecewise constant on each fine element. Let $N_c$ be the number of fine elements and $N_f$ be the number of fine elements in each coarse element. Thus, $Y_{ij}$ is the log permeability in the $j$th fine element of the $i$th coarse element. Using this notation, the mapping becomes:

$$W_i = \log \left[ \frac{1}{\delta x \sum_{j=1}^{N_f} \exp(-Y_{ij})} \right]$$

(3.38)
where $\delta x = x_i - x_{i-1}$. The derivative of $W_i$ with respect to $Y_{ij}$ is then

$$\frac{\partial W_i}{\partial Y_{ij}} = \frac{1}{\exp(Y_{ij}) \left( \sum_{k=1}^{N_f} \exp(-Y_{ik}) \right)}$$

(3.39)

The analytic form in one dimension makes computing the Jacobian trivial. In higher dimensions, where no clear analytic form exists, more effort is required.

### 3.6.2 Other notes

As an implementation note. In cases where $J(k)$ has a row of very small values, the term $J(Y)\Sigma_k J^T(Y)$ can be nearly singular. In this case, a nugget needs to be added to diagonal, $J(Y)\Sigma Y J^T(Y) + \epsilon I$. This is essentially adding some artificial width to the constraint, placing a lower bound on the eigenvalues of $J(Y)\Sigma_k J^T(Y) + \epsilon I$ and reducing the condition number. The value of $\epsilon$ is usually very small, $\approx 10^{-10}$.

### 3.7 Generalization to other multiscale settings

The framework presented here uses MsFEM to map fine scale quantities to coarse scale quantities. However, the framework is not restricted to this case. Any method that provides a coarse quantity sufficient to represent $p$ can apply, even upscaling could be used. MsFEM can be viewed as a local reduced order modeling technique. Other methods that represent the solution by a limited number of basis functions could also be used in this framework. For the multiscale framework to be effective, the coarse parameter should be sufficiently low dimensional for efficient MCMC and the mapping should not be extremely nonlinear. That is, the mapping should at least be continuous to ensure that MCMC can be efficient at correcting for linearization errors acquired during the iterative conditioning.
If you have built castles in the air, your work need not be lost; that is where they should be. Now put the foundations under them.

Henry David Thoreau

Chapter 4

Numerical Results

4.1 Test Cases

This section illustrates the multiscale framework presented in the previous chapter on several one dimensional test problems. Initially, a comparison is done with single scale inference to validate the multiscale approach. Then several examples are given to demonstrate the framework’s applicability in a wide variety of situations. A summary of the additional problems can be found in table 4.3.

4.1.1 Problem Description

All example problems share the same type of boundary conditions, Dirichlet on the left and Neumann on the right. However, each problem consists of a different type of permeability field and varying amounts of data. A fine scale simulation using a true permeability field is combined with additive noise to generate the data. The variance of this noise is 0.01 for all situations, which is approximately 1% of the pressure range for each case. This low noise level was used to ensure that the likelihood has a strong impact and the posterior is sufficiently different than the prior.

In the multiscale framework, DRAM is used for sampling the elemental integrals before iterative conditioning is performed with the SCKF and constrained MCMC. Within DRAM, the initial proposal size was manually tuned based on the well founded
The first test is a verification that the multiscale method produces similar results to standard MCMC approaches. The first example contains a relatively smooth permeability field without a wide scale separation. This is important for practical applications of MCMC. The parameter space is already 50 dimensional in this smooth case and would need to be increased for a permeability with large scale separation. An accurate DRAM characterization in the high dimensional setting would require a prohibitive number of samples.

The remaining tests illustrate the effectiveness of the multiscale inference framework on more complicated permeability fields. A summary of all the problems can be found in table 4.3. As an implementation note, coarse nodes need to be placed at the observation points because the MsFEM basis functions are never explicitly created. In the ensuing examples we have the luxury of creating the data, so it is easy to place observations at coarse node locations. In real situations this is not the case and either the mesh needs to be constructed with knowledge of the observation locations, or interpolation error must be introduced by projecting the data onto known basis functions instead of the MsFEM basis functions.

The standard error of a posterior estimate for a fixed number of samples is minimized when the posterior samples are uncorrelated. However, an MCMC chain generates correlated samples. Thus, in comparing MCMC methods, the autocorrelation of samples is a useful measure of performance. In fact, an integrated autocorrelation can be used to compute the effective sample size of a chain. This quantity represents how many independent samples would be needed to have essentially the same amount of information as the correlated MCMC chain. In the comparisons below, effective sample size (ESS) of a correlated sequence, $\Theta$, with length $N$ is defined as

$$
ESS(\Theta) = \frac{N}{1 + 2 \sum_{k=1}^{\infty} \rho_k(\Theta)}
$$

(4.1)

where $\rho_k(\theta)$ is the autocorrelation of the sequence at lag $k$. This quantity is computed using all samples after an initial burn-in of the chain. Obviously the sum has to be
truncated to a finite lag. Here, the maximum lag, $n$, is chosen so that $\rho_k(\Theta) \leq 0.05$ for all $k > n$. The effective sample size can be computed for each dimension of the chain so a range of effective sample sizes is obtained. Unless otherwise noted by Max ESS, or Min ESS, just $ESS$ will refer to the median effective sample size over all dimensions.

4.1.2 Verification with single scale inference results

Here we compare three different methods. The base case is single scale inference for $k$ using DRAM for sampling with a standard FEM forward solver with linear basis functions on the fine scale. The second single scale approach again uses DRAM for sampling but MsFEM as the forward solver. These methods are compared with the multiscale framework presented in the last chapter. The single scale methods operate with 50 elements spread over $[0, 1]$ and the multiscale approach uses 5 coarse elements with 10 fine elements in each coarse element. Figure 4-1 shows the true pressure with all 4 noisy observations used in this test as well as the posterior covariances obtained with each method. Figure 4-2 concentrates on the posterior mean and variance.

Clearly, all three methods are in fairly good agreement. The mean fields are nearly identical until the high permeability region on the right of the domain and the variances are quite close except at a few locations. The multiscale method slightly overestimates the posterior mean but captures the same covariance trend as the single scale methods. One possibility is the single scale MCMC chains have not sufficiently converged. While possible, this is unlikely. Each single scale chain was run for $2 \times 10^6$ steps, achieving a minimum effective sample size of 3400. A more likely explanation is in the approximate prior distribution on $\pi(e)$ and errors in the nonlinear iterative conditioning. Decreasing the step size of the nonlinear conditioning can help improve the result and the number of MCMC steps along the constraint so that no error in the iterative conditioning exists. However, performing enough MCMC samples so the chain “forgets” its initial point on the constraint is intractable. Thus, the iterative conditioning method should be chosen and tuned to best approximate the generation of samples from $\pi(k|e)$. While some errors are introduced, the computational savings
Figure 4-1: Location of pressure observations and posterior covariance comparison using the fully multiscale approach, single scale with FEM forward solver, and single scale with MsFEM forward simulation.

and applicability of the multiscale framework to problems with large scale separation outweigh this relatively small posterior error.

Although the effect is small, another possible error source also needs to be mentioned. With the strongly correlated prior in this test case, the update matrix from (3.26) can become ill-conditioned and cause issues during the SCKF stage and constrained MCMC stage. In some cases, the ill-conditioning causes the permeability to go far into the tails of the parameter space. It is possible to reject these samples and
Figure 4-2: Comparison of posterior mean and variance between standard single scale approaches and the multiscale framework. The left plot shows the posterior means $\pm 2\sigma^2$
	ry again, as in rejection sampling. However, sometimes the error is not as extreme and catching the inaccurate sample is not possible. This effect may be slightly contributing to the overestimation seen in figure 4-2. The ill-conditioned update occurs predominantly in smooth cases where the prior places a strong restriction on the posterior. With better conditioned prior covariances, this issue becomes insignificant. The smooth example here was used for efficient MCMC comparison but is not the application of interest. The multiscale framework will be most applicable to problems with weak priors and when the permeability field is expected to have short length scales. On average, this reduces the condition number of the linear update and the conditioning errors are less likely to have any impact on the posterior.

In addition to the posterior summaries, table 4.1 shows a performance comparison. The ratio of effective sample size to the total number of samples is shown in the first table while the second table displays how many effectively independent samples were produced each second. Note that the coarse MsFEM chain was thinned by a factor of 10 before performing the iterative conditioning. The timings were performed on a MacBook with 2.4Ghz Core2Duo and 2GB of 1067 MHz DDR3 memory. Focusing on $ESS/N$, it is clear that the multiscale method generates more than an order of
Table 4.1: Comparison of single scale performance with multiscale performance. ESS is the effective sample size, so $ESS/N$ is the effective number of samples related to the total number of fine scale samples generated and $ESS/S$ is the number of effective samples per second.

<table>
<thead>
<tr>
<th>Method</th>
<th>Min $ESS/N$</th>
<th>Max $ESS/N$</th>
<th>Median $ESS/N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEM Forward</td>
<td>$1.7 \times 10^{-3}$</td>
<td>$9.5 \times 10^{-3}$</td>
<td>$7.0 \times 10^{-3}$</td>
</tr>
<tr>
<td>MsFEM Forward</td>
<td>$5.6 \times 10^{-3}$</td>
<td>$9.3 \times 10^{-3}$</td>
<td>$7.2 \times 10^{-3}$</td>
</tr>
<tr>
<td>Multiscale</td>
<td>$1.7 \times 10^{-1}$</td>
<td>$3.1 \times 10^{-1}$</td>
<td>$2.5 \times 10^{-1}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Min $ESS/s$</th>
<th>Max $ESS/s$</th>
<th>Median $ESS/s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEM Forward</td>
<td>0.26</td>
<td>1.54</td>
<td>1.38</td>
</tr>
<tr>
<td>MsFEM Forward</td>
<td>2.03</td>
<td>3.34</td>
<td>2.60</td>
</tr>
<tr>
<td>Multiscale</td>
<td>1.12</td>
<td>2.05</td>
<td>1.64</td>
</tr>
</tbody>
</table>

magnitude more effectively independent samples than the single scale approaches. Not only does the coarse scale chain more effectively explore the parameter space and reduce correlations between samples, but additional randomness is introduced into the system by starting the iterative conditioning from a random sample of the prior. This additional randomness dramatically reduces the chain correlation.

Now concentrating on the timing, the benefit of the multiscale framework is not as prevalent. While still outperforming the standard FEM methods, the gap is not as convincing as the $ESS/N$. However, the single scale MsFEM timing is dependent on the analytical form of the elemental integrals in one dimension. In higher dimensions, MsFEM time is expected to drop closer to the standard FEM approach. These runs were computed in a sequential fashion, each iterative conditioning step was performed one after another, but an additional benefit of the multiscale procedure is its easy parallelization onto nearly as many nodes as available. For example, even for a moderate 80 core cluster, the parallel $ESS/s$ would be over 120. In addition to easy parallelization, the multiscale framework allows the inexpensive coarse chain to generate many more samples of the elemental integrals than is needed to characterize the poster of $k$. The coarse chain can be downsamples to reduce the number of iterative conditioning steps required.

The example problem here is a smooth problem using a Gaussian kernel prior used solely for comparison of the multiscale framework with standard single scale ap-
Smooth priors can be handled efficiently by standard single scale sampling approaches with a dimensionality reduction technique like Karhunen-Loeve expansion. However, many interesting problems do not have smooth permeability fields and can have large scale separations. While problematic for single scale methods, the multiscale framework is ideally suited for this type of problem. The following three test problems illustrate this fact. For a summary of the problems, see table 4.2.

### 4.1.3 Draw from Exponential Prior

Table 4.2: Summary of multiscale inference tests. $K_c$ is the number of coarse elements, $K_f$ is the number of fine elements in each coarse element. $d$ is the correlation length of the prior, $p$ is the prior power, $\sigma$ is the prior variance, $\mu_Y$ is the prior mean, $N_d$ is the number of data points.

<table>
<thead>
<tr>
<th>Trial Name</th>
<th>$K_c$</th>
<th>$K_f$</th>
<th>$d$</th>
<th>$p$</th>
<th>$\sigma$</th>
<th>$\mu_Y$</th>
<th>$N_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp. Prior</td>
<td>10</td>
<td>10</td>
<td>0.2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>SPE Layer 1</td>
<td>11</td>
<td>20</td>
<td>0.075</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>SPE Layer 5</td>
<td>11</td>
<td>20</td>
<td>0.17</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>11</td>
</tr>
</tbody>
</table>

This test uses an exponential prior covariance kernel on a 100 dimensional permeability field to create a more difficult problem beyond the single scale MCMC regime. In order to characterize the field, the number of coarse elements was increased to 10. The fine mesh has 10 fine elements in each coarse element. The number of observations was also increased from 4 to 8 as shown in figure 4-3. In terms of inference results, in the posterior density plot, the thick black line is the true permeability, the dashed lines are posterior samples, the thick green line is the posterior mean, and fuzzy shading is a more complete visualization of the posterior density. Darker shading implies a larger density. The autocorrelation plot is for the chain of permeability samples generated by sequentially taking each step of the elemental integral chain and performing iterative conditioning.

In the results from figure 4-3, the coarse chain was downsamples by 50 before performing the iterative conditioning. Little correlation can be seen within the chain. This is a major advantage over single scale methods, here the level of downsampling can be adaptively chosen to maximize the information after analyzing the correlation.
Figure 4-3: Results for multiscale inference using a draw from the prior. The prior kernel is defined by $p = 1$, $d = 0.2$, $\sigma = 1$. Twenty fine elements were used in each coarse element and 5 coarse elements were used.

of the coarse chain. Thus, with a fixed amount of computational power, a user can be confident that as much information as possible about the posterior was gathered. While not always a good measure of performance, it is encouraging to see the posterior mean agreeing quite well with the true permeability field. Also, note the relative smoothness of the posterior mean compared to the true field.
4.1.4 First layer of SPE10

In some sense, the above test problems were “cheating.” The permeability field was drawn from the prior, which means the prior is related to the true field. This violates the philosophical definition of prior information and does not stress the multiscale framework in the same way as a realistic problem. To take a small step towards reality, this test considers a slice of the SPE10 dataset. The SPE10 dataset is a representative reservoir description developed by the society of petroleum engineers. It consists of 85 layers of permeability and porosity information with $60 \times 220$ cells. In this test, row 25 of layer 1 is used as the true permeability field. The prior used here was adapted from a geostatistical characterization of the entire dataset developed by Sean McKenna at Sandia National Laboratory. See table 4.2 for a complete prior definition. The results shown in figure 4-4. Again, the black line is the true permeability, green is the posterior mean, etc...

The permeability field in this test has a very different structure than the prior realizations used before. There is clearly several inherent scales on a slowly varying field superimposed with high frequency oscillations. Interesting to note is the significant impact of a low permeability area around $x = 0.2$ on the pressure. The small permeability causes a significant drop in the pressure. Being a dominant feature of the entire pressure field, matching the data is very sensitive to matching this drop. The posterior results demonstrate this with less uncertainty (darker shading) near $x = 0.2$. Additionally, throughout the domain the true log permeability is covered by the posterior, showing that with additional observations, the variance could be reduced and the posterior could collapse on the true field.

Even in this one dimensional problem, the fine scale inference problem has 220 dimensions. DRAM would require millions of samples to attempt searching this space. However, in this multiscale framework, only the 11 elemental integral dimensions are searched with DRAM and an acceptance ratio of 23% is achieved. Furthermore, the lack of significant autocorrelation shows that most samples are nearly uncorrelated. In a similar fashion to the previous test, the info the inexpensive coarse MCMC chain
Figure 4-4: Results for multiscale inference using a slice of layer 1 of the SPE 10 dataset. The chain was downsampled by 50 before iterative conditioning.

was run longer and downsampled to provide less correlation between samples of $k$. The correlation plot in figure 4-4 was constructed after downsampling the coarse chain by a factor of 50.

**4.1.5 Fifth layer of SPE10**

The fifth layer of the SPE10 dataset has different characteristics than the first. Here, row 45 of layer 5 is used which has a generally negative slope with a medium correlation length pattern and some high frequency components. An exponential prior is
used and the number of observations is increased to 11. For a more complicated field like this one with a shorter lengthscale, more data is required for the posterior to be significantly different than the prior. Figure 4-5 shows the results. The pressure is dominated by the two distinct drops near the right of the domain. The pressure is again sensitive to the permeability values in these areas. The posterior captures this behavior as expected. There are two low permeability areas in the posterior with a small variance. These bound the highly uncertain area around $x = 0.9$. While the true permeability does not have as large a peak at $x = 0.9$ as the posterior mean, the multiscale method samples effectively enough to capture the uncertainty in this region. The smaller true permeability is well within the high probability region of the posterior.

### 4.1.6 Summary of test cases

Table 4.3: Summary of multiscale inference performance. $ESS$ is the effective sample size, so $ESS/N$ is the effective number of samples related to the total number of fine scale samples generated and $ESS/S$ is the number of effective samples per second. Additionally, acceptance is the acceptance rate of the coarse MCMC chain.

<table>
<thead>
<tr>
<th>Trial Name</th>
<th>ESS/N</th>
<th>ESS/S</th>
<th>Acceptance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp. Prior</td>
<td>0.2267</td>
<td>2.11</td>
<td>0.16</td>
</tr>
<tr>
<td>SPE Layer 1</td>
<td>0.028</td>
<td>0.17</td>
<td>0.23</td>
</tr>
<tr>
<td>SPE Layer 5</td>
<td>0.067</td>
<td>0.082</td>
<td>0.12</td>
</tr>
</tbody>
</table>

These test cases stress the multiscale inference framework with high dimensionality and realistic permeability fields with scale separation. Table 4.3 summarizes the test problems not compared with MCMC. MCMC methods could be applied on these problems but the comparison between the MCMC results and multiscale results would be moot because the MCMC would not be able to sufficiently explore the posterior. Some of these test problems have also been studied with more observation noise, the result is as expected, more noise results in a posterior more similar to the prior.

In the verification of the multiscale method with MCMC results, it's clear that in a smooth setting, the iterative conditioning process can become difficult due to poor condition numbers of the linear update. However, the ability to adaptively down sam-
Figure 4-5: Results for multiscale inference using a slice of layer 5 of the SPE 10 dataset. The chain was downsampled by 50 before iterative conditioning.

ple and easily parallelize the iterative conditioning stage would result in a dramatic increase in the number of effectively independent samples generated. Some will argue that MCMC can also easily be parallelized by running multiple chains simultaneously and then combining the results. This is an improvement, but each chain still suffers an inability to effectively search vast emptiness of large dimensional spaces. Additionally, several coarse MCMC chains could be also used in the multiscale setting, reducing the amount of down sampling required before iteratively conditioning.
Science, my lad, is made up of mistakes, but they are mistakes which it is useful to make, because they lead little by little to the truth.

Chapter 5

Future work and conclusions

5.0.7 Future work

One dimensional examples were given here, but it is important to note the framework is not dependent on the spatial dimension of the parameters being estimated. The exact same formulation exists for any number of spatial dimensions. This text focused on the multiscale framework derivation and a discussion of the formulation intricacies with motivation using a one dimensional field. Compared to the one dimensional setting, no analytical expression exists for the Jacobian information of the two dimensional mapping from $k$ to $e$. However, the Jacobian can be computed by using adjoint methods to first compute the sensitivity of the MsFEM boundary conditions to the log permeability and then computing the joint sensitivity of the elemental integrals to the permeability and boundary condition. Additionally, the number of elemental integrals per element will also increase. It should be noted however, that once Jacobian information is computed, all other parts of the algorithm remain unchanged. Implementation in higher dimensions is just one aspect of future work.

5.0.8 Use in dynamic problems

In addition to expanding this application of the multiscale framework to higher dimensions, the method is also applicable to time dependent parabolic equations. In the porous media context this corresponds to a time dependent pressure equation.
An interesting avenue of research is the use of particle filters or the ensemble Kalman filter to dynamically update the posterior elemental integral distribution and only generate a few fine scale realizations at each timestep. This also has applications in real-time PDE constrained dynamic inference. Being sufficient to describe the coarse pressure, only elemental integrals are needed to evaluate the likelihood and evolve the ensemble forward in time. By dramatically reducing the computational cost of a likelihood evaluation, the multiscale approach would allow for real time inference. Fine scale realizations would not need to be generated in real-time for many applications and a massively parallel post-processing procedure would allow the fine scale realizations to be efficiently generated.

5.0.9 Spatial decoupling during iterative conditioning

The iterative conditioning procedure does not take into account any spatial decomposition. All components of the permeability could be correlated with other permeabilities. However, in many applications, some areas of the spatial domain will not be related and further decomposition could be performed. Using knowledge of uncorrelated parameters would allow the iterative conditioning process itself to be parallelized. The multiple levels of parallelism (independent coarse samples and independent spatial regions) would allow this multiscale framework to capitalize on the advent of highly heterogeneous clusters where multiple levels of parallelism are needed for maximum efficiency.

5.1 Finale

Inference problems found in real-world applications are often characterized by large dimensional parameter spaces and noisy observations. Bayesian inference provides an intuitive paradigm for formulating the inference problem but requires sampling of a high dimension space. Especially in the case of elliptic forward models, high probability regions of the posterior distribution can be small compared to the parameter space, making traditional single scale MCMC difficult. Using a multiscale approach
with a nonlinear mapping from fine parameters to coarse parameters, we showed that when the coarse parameters are sufficient to describe the model output, the scales can be decoupled and the inference process can be broken into two stages. The first stage uses standard MCMC sampling tools to characterize a posterior for the coarse quantities and the second stage takes of the coarse samples and generates a realization of the fine scale parameter that agrees with the coarse quantity.

It was shown in the context of porous media that this multiscale framework can be successfully applied for inference. By decoupling the elemental integrals from the permeability, as shown in the several test cases above, it is possible to efficiently explore a high dimensional posterior. It is important to note that the multiscale procedure used here is not limited to elliptic problems or even hydrology. In fact, the framework should be applicable to many situations involving smoothing operators (like the pressure equation) and/or nonlinear observations.

The unique decoupling of fine and coarse scales with a nonlinear mapping between the two, provides a small step towards tackling large dimensional inference problems in a Bayesian setting. The several examples of permeability estimation shown here indicate the efficacy of this decoupling and point towards a future application in currently intractable real-world inference problems.
Appendix A

Philosophical Asides

A.1 Epistemic and Aleatory Uncertainty

Consider the definitions of epistemic uncertainty and aleatory uncertainty put forth by William Oberkampf of Sandia National Lab:¹

**Aleatory Uncertainty** is an inherent variation associated with the physical system or the environment.

**Epistemic Uncertainty** is an uncertainty that is due to a lack of knowledge of quantities or processes of the system or the environment.

In a quantum system, Heisenberg’s uncertainty principle provides a clear distinction between what is a lack of knowledge and what is an unavoidable variation. However, in a largescale setting such as porous media flow, the distinction is moot. I claim that in this setting, all uncertainty is of the epistemic variety. In theory, if the molecular structure of a porous media was known as well as the exact molecular structure of the fluid involved, I could use first principles to exactly model the flow. There is no inherent variation in this system. At a more basic level, academics may cite the roll of the dice as an example of aleatory uncertainty. However, given the precise origin and

¹The presentation where Dr. Oberkampf makes these definitions can be found at http://www.stanford.edu/group/cits/pdf/lectures/oberkampf.pdf
kinematics of the dice, I again could use the system physics to precisely anticipate
the roll’s outcome. A typical example of aleatory uncertainty given by [73]:

Some examples of aleatory uncertainty are weather or the height of
individuals in a population: these cannot be reduced by gathering further
information.

While intractable, I believe that with much effort, the height of every individual
in a population could be known and with tremendous advances in sensing devices,
the state of the atmosphere could also be measured. I admit that there is a difference
between uncertainties that can easily be reduced by taking more data, uncertainties
that can only be reduced through extreme measures, and uncertainties that can not
be reduced due to physical infeasibility (read Heisenberg’s Uncertainty Principle).
However, excepting systems in the uncertainty principle regime, I do not feel that
there is any useful distinction between types of uncertainties. Even if currently there
is no method of obtaining data that reduces the uncertainty of a system, some endeav-
oring citizen will eventually provide a methodology to do so, and the uncertainty will
again fall into classical definition of epistemic uncertainty. A definition that changes
with the times is no more useful than a definition of pressure that changes with the
season.

A.2 Bayesian vs. Frequentist

Modern statistical methods can approximately be divided into two categories: the
Bayesian methodology and the frequentist methodology. The root of the distinction
is the definition of probability used within the two camps. On the Bayesian side,
probabilities are understood as degrees of belief while a frequentist would view a
probability as the long time chance of a recurring event. As an example, consider
measuring the concentration of sediment at a fixed location in a river. After several
measurements have been taken, a probability distribution approximating the data
is developed. A frequentist would view the density as the the limiting chance of
a measurement taking a particular value. A Bayesian would view the density as a measure of belief of the concentration in the river. The distinction between frequentist and Bayesian views may seem subtle in this example, but interpreting probability as a degree of belief allows for easy extension of statistical descriptions to situations where the variable cannot be viewed as a repeatable or recurring event. Variations in flow rates and other natural processes change what sediment concentration would be seen at any instant in time and a frequentist view makes sense. However, in inverse problems, where the parameter of interest could be static, (i.e. permeability fields), treating these parameters as a random variables in the frequentist setting may require some imagination, whereas a Bayesian would have no issues assigning a probability to the mass of the moon, because it simply represents the current state of knowledge. The point here is that the mass of the moon is not a random event, it is a fixed parameter, but the mass is not known precisely and can be viewed as a random variable in a Bayesian viewpoint but not a frequentist viewpoint.

Debates on the superiority of Bayesian of frequentist statistics can become heated. See [27] for a glimpse into this heated debate. I prefer a more practical perspective. For some applications the Bayesian approach will be more natural than the frequentist approach, and for other applications a frequentist approach will be more suitable. In the setting of PDE constrained inference, I find the Bayesian approach to be better suited, but in many data analysis situations, classical frequentist analysis is preferable.
Bibliography

[1] Yale tablet ybc 7289, 1800–1650 BC. 12


117