UNCERTAINTY QUANTIFICATION USING MULTISCALE METHODS FOR POROUS MEDIA FLOWS

A Dissertation

by

PAUL FRANCIS DOSTERT

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2007

Major Subject: Mathematics

UNCERTAINTY QUANTIFICATION USING MULTISCALE METHODS FOR POROUS MEDIA FLOWS

A Dissertation

by

PAUL FRANCIS DOSTERT

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Yalchin Efendiev
Committee Members,	Akhil Datta-Gupta
	Richard Ewing
	Raytcho Lazarov
	Bani Mallick
Head of Department,	Al Boggess

December 2007

Major Subject: Mathematics

ABSTRACT

Uncertainty Quantification Using Multiscale Methods
for Porous Media Flows. (December 2007)

Paul Francis Dostert, B.S., James Madison University;
M.S., Texas A&M University
Chair of Advisory Committee: Dr. Yalchin Efendiev

In this dissertation we discuss numerical methods used for uncertainty quantification applications to flow in porous media. We consider stochastic flow equations that contain both a spatial and random component which must be resolved in our numerical models. When solving the flow and transport through heterogeneous porous media some type of upscaling or coarsening is needed due to scale disparity. We describe multiscale techniques used for solving the spatial component of the stochastic flow equations. These techniques allow us to simulate the flow and transport processes on the coarse grid and thus reduce the computational cost. Additionally, we discuss techniques to combine multiscale methods with stochastic solution techniques, specifically, polynomial chaos methods and sparse grid collocation methods.

We apply the proposed methods to uncertainty quantification problems where the goal is to sample porous media properties given an integrated response. We propose several efficient sampling algorithms based on Langevin diffusion and the Markov chain Monte Carlo method. Analysis and detailed numerical results are presented for applications in multiscale immiscible flow and water infiltration into a porous medium.

ACKNOWLEDGMENTS

First and foremost I would like to thank Dr. Yalchin Efendiev for all his guidance and support throughout my graduate career. His many insights and ideas have been invaluable in my preparation to begin my career as a mathematician. I would like to acknowledge Dr. Raytcho Lazarov and Dr. Richard Ewing for not only serving on my committee, but also convincing me to study at Texas A&M. In addition, I wish to thank the entire numerical analysis group for helping me prepare to begin my research and write my dissertation.

I would like to thank Dr. Thomas Yizhao Hou, Dr. Wuan Luo, and Dr. Victor Ginting for their help in my research. I also wish to thank Dr. Akhil Datta-Gupta and Dr. Bani Mallick for serving on my committee and providing many insightful comments and questions.

I would also like to recognize the wonderful mathematics professors at James Madison University. Specifically I'd like to thank Dr. James Sochacki, who encouraged me to pursue a career in mathematics. I also wish to express my gratitude to the professors and staff of the Department of Mathematics at Texas A&M. I would like to thank Mr. Art Belmonte for his help in preparing me to teach mathematics at the university level. I also wish to thank Ms. Monique Stewart for her incredible patience and knowledge with the university administration. I also thank the students in the mathematics department, specifically Dimitar Trenev, Veselin Dobrev, Dr. Dylan Copeland, Dr. Troy Henderson, and Dr. Taejong Kim.

My research was partially supported by the NSF under DMS-0327713, EIA-05401.36, DMS-062113, DMS-0216275 and by the DOE under DE-FG02-05ER25669.

Last, but certainly not least, I'd like to thank my wife, Laura, and my family for their incredible patience and encouragement throughout my graduate career.

TABLE OF CONTENTS

CHAPTER Page

Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

Uncertainties in the detailed description of reservoir lithofacies, porosity, and permeability are major contributors to uncertainty in reservoir performance forecasting. Making decisions in reservoir management requires a method for quantifying uncertainty. Large uncertainties in reservoirs can greatly affect the production and decision making on well drilling. Better decisions can be made by reducing the uncertainty. Thus, quantifying and reducing the uncertainty is an important and challenging problem in subsurface modeling. Additional dynamic data, such as the production data, can be used in achieving more accurate predictions. Previous findings show that dynamic data can be used to improve the predictions and reduce the uncertainty. Therefore, to predict future reservoir performance, the reservoir properties, such as porosity and permeability, need to be conditioned to dynamic data. In general it is difficult to calculate this posterior probability distribution because the process of predicting flow and transport in petroleum reservoirs is nonlinear. Instead, we estimate this probability distribution from the outcomes of flow predictions for a large number of realizations of the reservoir. It is essential that the permeability (and porosity) realizations adequately reflect the uncertainty in the reservoir properties, i.e., we correctly sample this probability distribution.

The prediction of permeability fields based on dynamic data is a challenging problem because permeability fields are typically defined on a large number of grid blocks. The Markov chain Monte Carlo (MCMC) method and its modifications have been used previously to sample the posterior distribution of the permeability field.

This dissertation follows the style of the Journal of Computational Physics.

Oliver et al. [36, 37] proposed the randomized maximum likelihood method, which generates unconditional realizations of the production and permeability data and then solves a deterministic gradient-based inverse problem. The solution of this minimization problem is taken as a proposal and accepted with probability one because the rigorous acceptance probability is very difficult to estimate. In addition to the need of solving a gradient-based inverse problem, this method does not guarantee a proper sampling of the posterior distribution. Developing efficient and rigorous MCMC calculations with high acceptance rates remains a challenging problem.

In this dissertation, we study multiscale methods for stochastic porous media flow equations and their as applications to uncertainty quantification. We propose an approach where traditional MCMC algorithms are modified by the use of multiscale methods to coarsen the flow equations spatially. We combine this with sparse collocation or polynomial chaos techniques to obtain solutions in the high dimensional stochastic space.

In Chapter II, we cover some preliminary background material. We introduce two porous media flow equations that will be studied throughout the dissertation. First, we present the equations for two-phase immiscible flow which have many applications in petroleum engineering and reservoir modeling. Second, we present Richards' equation, often used in hydrology, which models the flow of water in unsaturated soils. We then introduce the Karhunen-Loève expansion, which allows us to represent the realizations of the random field information in each of our equations. Lastly, we present a brief introduction to homogenization.

In Chapter III, we consider the techniques needed to solve the stochastic equations arising in porous media. We first discuss polynomial chaos expansions. We expand the unknown variables in the porous media equations as multidimensional Hermite polynomials. We derive a system of equations which, when solved, recover the coefficients in the polynomial expansion. Once these coefficients are found, then for any stochastic input, an inexpensive approximation to the solution of the porous media equations can be made using the Hermite polynomial expansion. Next, we discuss sparse grid collocation methods. In these methods, we solve the porous media equations for stochastic variables at some sparse grid points in high dimensional stochastic space. Using the data at these sparse grid points, we can approximate the solution for any stochastic input by using multivariate polynomial interpolation. In contrast to polynomial chaos methods, sparse grid collocation methods do not require the solution to a large system of coefficients.

In Chapter IV, we combine multiscale methods with the stochastic solution methods described in Chapter III. We first derive the upscaled equations for the polynomial chaos system using single-phase flow. We then present analysis for the use of sparse grid collocation within multiscale finite element methods. We consider two approaches. In the first approach we compute basis functions at some sparse points in stochastic space. Instead of solving for basis functions for each stochastic variable, we interpolate using pre-computed basis functions based on interpolation. In the second approach we use the family of basis functions for a given set of stochastic variables without interpolating them to a particular realization.

In Chapter V we present details of the uncertainty quantification problems under consideration. Our general goal is to obtain a set of fields that reproduce some given or measured response. We consider the uncertainty quantification problems in the context of sampling using Markov chain Monte Carlo methods. Various algorithms using Langevin diffusion, multiscale models, and collocation methods are presented and analyzed.

In Chapter VI we present numerical results for the sampling problems introduced in Chapter V. We discuss the strengths and weaknesses for each algorithm and present corroborating numerical results. Results for single-phase flow, two-phase flow, and Richards' equation are presented.

Lastly, in Chapter VII, we summarize our findings and present possibilities for future research.

CHAPTER II

BACKGROUND MATERIAL

In this chapter, we introduce some background material that is necessary for our later chapters. We first introduce the two different classes of porous media flow equations. Secondly, we introduce the Karhunen-Loève Expansion, which will be used extensively throughout the later discussions. Lastly, we present some basic concepts of homogenization and discuss the numerical methods involved in the porous media equations.

2.1. Prototypical Examples of Porous Media Flows

We consider two prototypical examples of porous media flows. The first model, referred to as two-phase immiscible flow, has applications in petroleum reservoir simulation. The second, Richards' equation, describes the infiltration of water flow into a porous media whose pore space is filled with air and water.

2.1.1. Two-Phase Immiscible Flow

We consider two-phase flow in a reservoir (denoted by Ω) under the assumption that the displacement is dominated by viscous effects; i.e., we neglect the effects of gravity, compressibility, and capillary pressure. Porosity is considered to be constant. The two phases will be referred to as water and oil, designated by subscripts w and o, respectively. We write Darcy's law for each phase as follows:

$$v_j = -\frac{k_{rj}(S)}{\mu_j} K \cdot \nabla p, \tag{2.1}$$

where v_j is the phase velocity, μ_j is the phase porosity, and k_{rj} is the relative permeability to phase j ($j = o, w$). We denote the permeability as K tensor, the water saturation (volume fraction) as S and p as the pressure. We will assume that the permeability tensor is diagonal, thus $K = kI$, where k is a scalar and I is the unit tensor. In this work, a single set of relative permeability curves is used. Combining Darcy's law with a statement of conservation of mass allows us to express the governing equations in terms of the so-called pressure and saturation equations:

$$\nabla \cdot (\lambda(S) k \nabla p) = h, \tag{2.2}$$

$$\frac{\partial S}{\partial t} + v \cdot \nabla f(S) = 0, \tag{2.3}$$

where λ is the total mobility, h is the source term, $f(S)$ is the flux function, and v is the total velocity, which are respectively given by:

$$\lambda(S) = \frac{k_{rw}(S)}{\mu_w} + \frac{k_{ro}(S)}{\mu_o}, \tag{2.4}$$

$$f(S) = \frac{k_{rw}(S)/\mu_w}{k_{rw}(S)/\mu_w + k_{ro}(S)/\mu_o}, \tag{2.5}$$

$$v = v_w + v_o = -\lambda(S) k \cdot \nabla p. \tag{2.6}$$

Throughout our discussions, the above descriptions will be referred to as the two-phase flow problem. We define single-phase flow as the same equation, but with $k_{rw}(S) = S$ and $k_{ro}(S) = 1 - S$.

2.1.2. Richards' Equation

We consider Richards' equation which describes the infiltration of water into a porous media whose pore space is filled with air and water. The equation describing Richards' equation under some assumptions is given by

$$D_t \theta(u) - \operatorname{div}(k(x, u) D_x(u + x_3)) = 0 \quad \text{in}\, \Omega, \tag{2.7}$$

where $\theta(u)$ is the volumetric water content and u is the pressure. The following are assumed ([39]) for (2.7): (1) the porous medium and water are incompressible; (2) the temporal variation of the water saturation is significantly larger than the temporal variation of the water pressure; (3) air phase is infinitely mobile so that the air pressure remains constant, in this case it is atmospheric pressure which equals zero; (4) neglect the source/sink terms.

Constitutive relations between θ and u and between k and u are developed appropriately, which consequently gives nonlinear behavior in (2.7). The relation between the water content and pressure is referred to as the moisture retention function. The equation written in (2.7) is called the coupled-form of Richards' equation. In other literature this equation is also called the mixed form of Richards' equation, due to the fact that there are two variables involved in it, namely, the water content θ and the pressure head u. Taking advantage of the differentiability of the soil retention function, one may rewrite (2.7) as follows:

$$C(u)D_t u - \operatorname{div}(k(x,u)D_x(u+x_3)) = 0 \quad \text{in}\, \Omega, \tag{2.8}$$

where $C(u) = d\theta/du$ is the specific moisture capacity. This version is referred to as the head-form (h-form) of Richards' equation. Another formulation of the Richards' equation is based on the water content θ,

$$D_t\theta - \operatorname{div}(D(x,\theta)D_x\theta) - \frac{\partial k}{\partial x_3} = 0 \quad \text{in}\, \Omega, \tag{2.9}$$

where $D(\theta) = k(\theta)/(d\theta/du)$ defines the diffusivity. This form is called the θ-form of Richards' equation.

The sources of nonlinearity of Richards' equation comes from the moisture retention and relative hydraulic conductivity functions, $\theta(u)$ and $k(x,u)$, respectively. Perhaps the most widely used empirical constitutive relations for the moisture content and hydraulic conductivity is due to the work of van Genuchten [43]. He proposed a method of determining the functional relation of relative hydraulic conductivity to the pressure head by using the field observation knowledge of the moisture retention. In turn, the procedure would require curve-fitting to the proposed moisture retention function with the observational data to establish certain parameters inherent to the resulting hydraulic conductivity model. There are several widely known formulations of the constitutive relations:

Haverkamp model [25] :

$$\theta(u) = \frac{\alpha\,(\theta_s - \theta_r)}{\alpha + |u|^{\beta}} + \theta_r,$$
$$k(x,u) = K_s(x)\frac{A}{A + |u|^{\gamma}}.$$

van Genuchten model [43] :

$$\theta(u) = \frac{\alpha\,(\theta_s - \theta_r)}{\left[1 + (\alpha|u|)^{n}\right]^{m}} + \theta_r,$$
$$k(x,u) = K_s(x)\frac{\left\{1 - (\alpha|u|)^{n-1}\left[1 + (\alpha|u|)^{n}\right]^{-m}\right\}^{2}}{\left[1 + (\alpha|u|)^{n}\right]^{m/2}}.$$

Exponential model [44] :

$$\theta(u) = \theta_s\, e^{\beta u}, \tag{2.10}$$
$$k(x,u) = K_s(x)\, e^{\alpha u}.$$

Irmay model [28] :

$$\theta(u) = \theta_r + (\theta_s - \theta_r)\, e^{\alpha u}, \tag{2.11}$$
$$k(x,u) = K_s(x)\left(\frac{\theta - \theta_r}{\theta_s - \theta_r}\right)^{m}.$$

In each of the above constitutive relations θ_r is the residual moisture content when the soil is very dry and θ_s is the saturated moisture content when the soil is fully saturated. α, β and m are fitting parameters. Generally one uses the fitting parameters to fit the proposed function with experimental or observed data.

The variable K_s in the above models is known as the saturated hydraulic conductivity, which we will generally refer to as simply the saturated conductivity. It has been observed that the saturated conductivity has a broad range of values, which together with the functional forms presented above, confirm the nonlinear behavior of the process. Furthermore, the water content and saturated conductivity approach zero as the pressure head goes to very large negative values.

2.2. Karhunen-Loève Expansion (KLE)

In many practical applications one wishes to express the permeability k in (2.2) or the saturated conductivity K_s in the constitutive relations for $k(x, u)$ in (2.7) as an expansion of some parameters, rather than simply a function in physical space. In this section we discuss one particular type of expansion, known as the Karhunen-Loève expansion [31, 45], or the KLE. For simplicity, the KLE will be explained with respect to permeability, but the same explanation holds for saturated conductivity.

Using the KLE, a permeability field can be expanded in terms of an optimal L^2 basis. By truncating the expansion we can represent the permeability matrix by a small number of random parameters. Denote $Y(x, \omega) = \log[k(x, \omega)]$, where the random element ω is included to remind us that k is a random field. For simplicity, we assume that $E[Y(x, \omega)] = 0$. Suppose $Y(x, \omega)$ is a second order stochastic process with $E \int_\Omega Y^2(x, \omega) dx < \infty$, where E is the expectation operator. Given an orthonormal basis $\{\phi_k\}$ in $L^2(\Omega)$, we can expand $Y(x, \omega)$ as a general Fourier series

$$Y(x, \omega) = \sum_{k=1}^{\infty} Y_k(\omega) \phi_k(x), \qquad Y_k(\omega) = \int_\Omega Y(x, \omega) \phi_k(x) dx.$$

We are interested in the special L^2 basis $\{\phi_k\}$ which makes the random variables Y_k uncorrelated. That is, $E(Y_iY_j) = 0$ for all $i \neq j$. Denote the covariance function of Y as $R(x, y) = E\left[Y(x)Y(y)\right]$. Then such basis functions $\{\phi_k\}$ satisfy

$$E[Y_iY_j] = \int_\Omega \phi_i(x)dx \int_\Omega R(x, y)\phi_j(y)dy = 0, \qquad i \neq j.$$

Since $\{\phi_k\}$ is a complete basis in $L^2(\Omega)$, it follows that $\phi_k(x)$ are eigenfunctions of $R(x, y)$:

$$\int_\Omega R(x, y)\phi_k(y)dy = \lambda_k\phi_k(x), \qquad k = 1, 2, \dots, \tag{2.12}$$

where $\lambda_k = E[Y_k^2] > 0$. Furthermore, we have

$$R(x, y) = \sum_{k=1}^{\infty} \lambda_k\phi_k(x)\phi_k(y). \tag{2.13}$$

Denote $\theta_k = Y_k/\sqrt{\lambda_k}$, then θ_k satisfy $E(\theta_k) = 0$ and $E(\theta_i\theta_j) = \delta_{ij}$. It follows that

$$Y(x, \omega) = \sum_{k=1}^{\infty} \sqrt{\lambda_k}\theta_k(\omega)\phi_k(x), \tag{2.14}$$

where ϕ_k and λ_k satisfy (2.12). We assume that the eigenvalues λ_k are ordered as $\lambda_1 \geq \lambda_2 \geq \dots$. The expansion (2.14) is called the Karhunen-Loève expansion (KLE). In the KLE, the L^2 basis functions $\phi_k(x)$ are deterministic and resolve the spatial dependence of the permeability field. The randomness is represented by the scalar random variables θ_k. After we discretize the domain Ω by a rectangular mesh, the continuous KLE is reduced to finite terms. Generally, we only need to keep the leading order terms (quantified by the magnitude of λ_k) and still capture most of the energy of the stochastic process $Y(x, \omega)$. For an N-term KLE approximation $Y_N = \sum\limits_{k=1}^{N} \sqrt{\lambda_k}\theta_k\phi_k$, define the energy ratio of the approximation as

$$e(N) := \frac{E\|Y_N\|^2}{E\|Y\|^2} = \frac{\sum_{k=1}^{N} \lambda_k}{\sum_{k=1}^{\infty} \lambda_k}. \tag{2.15}$$

If $\lambda_k, k = 1, 2, \ldots,$ decay very fast, then the truncated KLE would be a good approximation of the stochastic process in the L^2 sense.

We consider two different types of permeability fields $k(x, \omega)$. First, let us suppose $k(x, \omega)$ is a log-normal homogeneous stochastic process, then $Y(x, \omega)$ is a Gaussian process and θ_k are independent standard Gaussian random variables. We assume that the covariance function of $Y(x, \omega)$ has the form

$$R(x, y) = \sigma^2 \exp\Big(-\frac{|x_1 - y_1|^2}{2L_1^2} - \frac{|x_2 - y_2|^2}{2L_2^2}\Big). \tag{2.16}$$

In the above formula, L_1 and L_2 are the correlation lengths in each dimension, and $\sigma^2 = E(Y^2)$ is a constant. We also consider the case when $k(x, \omega)$ is a log-exponential homogeneous stochastic process. In this case, we have a covariance function of $Y(x, \omega)$ of the form

$$R(x, y) = \sigma^2 \exp\Big(-\frac{|x_1 - y_1|}{L_1} - \frac{|x_2 - y_2|}{L_2}\Big). \tag{2.17}$$

In either case, we first solve the eigenvalue problem (2.12) numerically on the rectangular mesh and obtain the eigenpairs $\{\lambda_k, \phi_k\}$. We then truncate the KLE using an appropriate number of terms so that the energy ratio (2.15) is still high. We can then sample $Y(x, \omega)$ from the truncated KLE (2.14) by generating Gaussian random variables θ_k.

In Figure 2.1 we plot the eigenvalues and energy ratio for the KLE with a normal covariance function (2.16). We assume $L_1 = 0.5$, $L_2 = 0.1$ and $\sigma^2 = 2.0$. The convergence rate of the KLE depends only on the smoothness of the covariance function, thus for the normal (Gaussian) covariance function, very few terms are needed. In this particular example we capture more than 95% of the energy using under 20 terms in the KLE. In Figure 2.2 we plot six example eigenvectors. The top three plots show the first three eigenvectors (1, 2, and 3) in the KLE. The bottom three plots show the

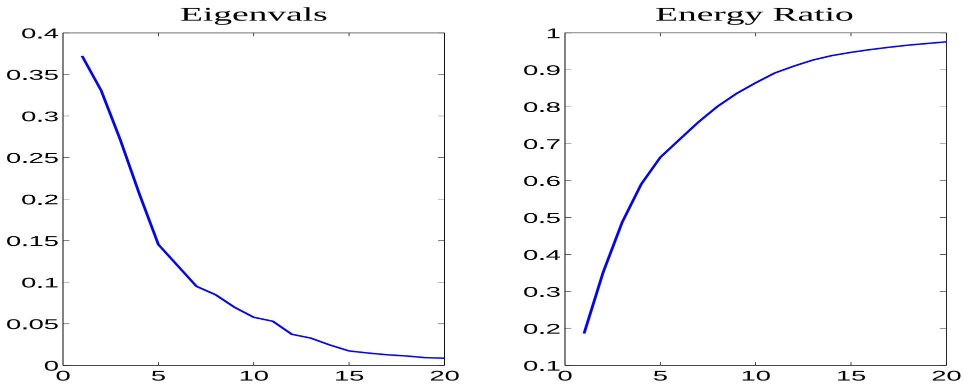
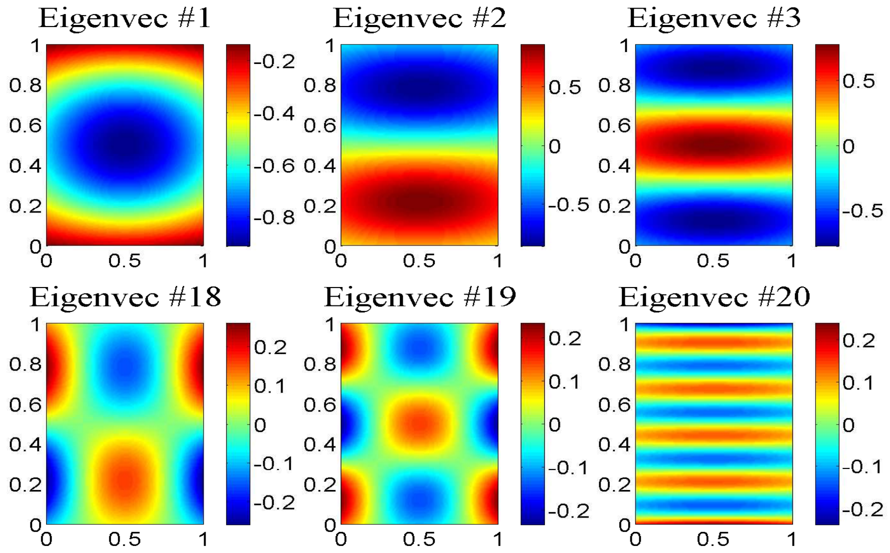


Fig. 2.1. Left: The first 20 eigenvalues for the normal covariance function (2.16). Right: The energy ratio (2.15) using 1 through 20 terms. In each case $L_1 = 0.5$, $L_2 = 0.1$ and $\sigma^2 = 2.0$.

last three eigenvectors (18, 19, and 20) in the KLE. Note the first three eigenvectors contain large scale features while the last three eigenvectors contain relatively small scale features. We again note that, in addition to the scale of the features, the latter eigenvectors also correspond to eigenvalues of smaller magnitude. While there is a large benefit in using normal covariance functions due to the fast convergence, we are somewhat limited in what types of fields we can generate using this KLE. From the previous eigenvector plots, it is clear that very small scale features cannot be reproduced. The smallest scales we can hope to reproduce are those shown in the plots for eigenvectors 18, 19, and 20. In Figure 2.3 we plot four random permeability fields generated using the KLE with normal covariance. Note that these permeability fields are generally quite smooth with fairly large features, consistent with the eigenvector expansion.

In Figure 2.4 we plot the eigenvalues and energy ratio for the KLE with the exponential covariance function (2.17). We again assume $L_1 = 0.5$, $L_2 = 0.1$ and $\sigma^2 = 2.0$. We now must keep over 200 eigenvalues in the KLE to capture just over 90% of the energy. While we must keep many more terms in this expansion, in many



Eigenvec #1

Eigenvec #2

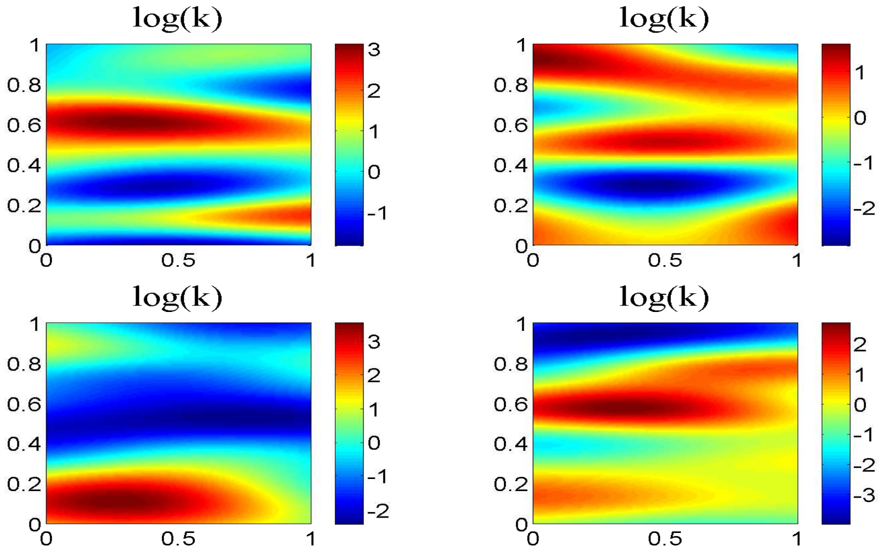
Eigenvec #3

Eigenvec #18

Eigenvec #19

Eigenvec #20

Fig. 2.2. Six example eigenvectors generated using the KLE for the normal covariance function (2.16) with 20 terms. We use $L_1 = 0.5$, $L_2 = 0.1$ and $\sigma^2 = 2.0$.



log(k)

log(k)

log(k)

log(k)

Fig. 2.3. Four example permeability fields generated using the KLE for the normal covariance function (2.16) with 20 terms. In each permeability we use $L_1 = 0.5$, $L_2 = 0.1$ and $\sigma^2 = 2.0$.

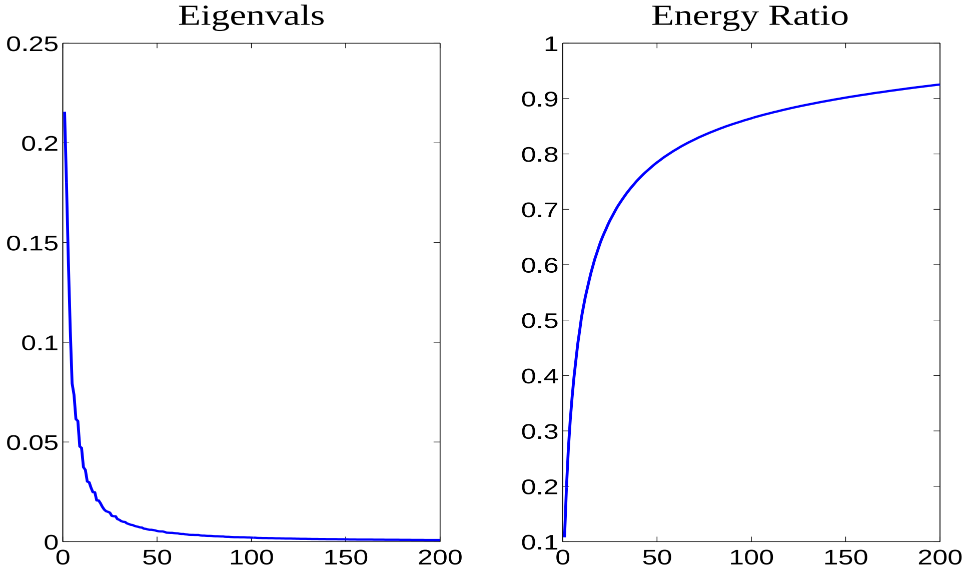


Fig. 2.4. Left: The first 200 eigenvalues for the exponential covariance function (2.17). Right: The energy ratio (2.15) using 1 through 200 terms. In each case $L_1 = 0.5$, $L_2 = 0.1$ and $\sigma^2 = 2.0$.

practical applications one may wish to generate permeability fields which are not nearly as smooth as when using normal covariance. In Figure 2.5 we plot six example eigenvectors. The top three plots show eigenvectors 1, 10, and 20 in the KLE. The bottom three plots show eigenvectors 180, 190, and 200 in the KLE. As with the normal covariance, the first eigenvectors in the KLE contain large scale information. As we descend through the eigenvector expansion, we find smaller scale features. In Figure 2.6 we show four random permeability fields generated using the KLE with exponential covariance. Note the small scale features are consistent with the features found in the plots for the tail of the eigenvector expansion.

2.3. Numerical Homogenization and Multiscale Methods

In this section we discuss the upscaling procedures that will be used to solve our coarse-scale equations. For the coarse-scale models, we consider only upscaling of the

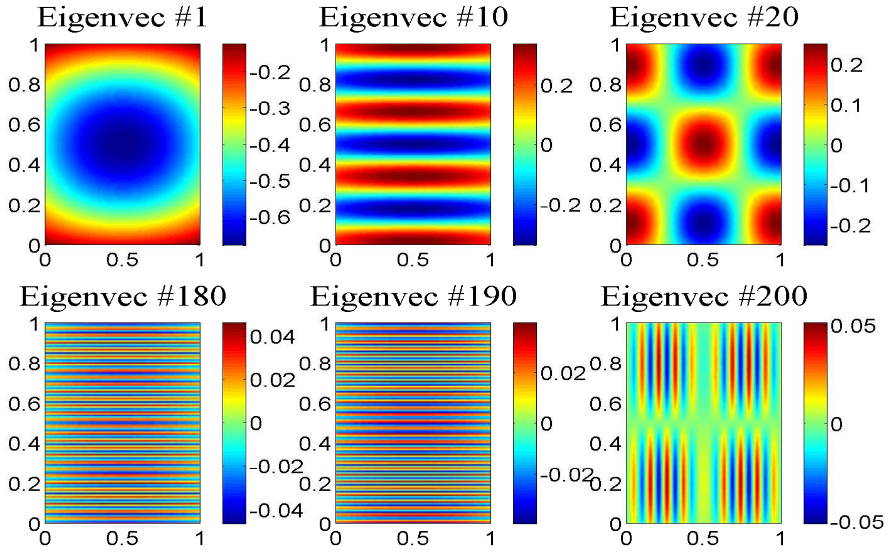


Fig. 2.5. Six example eigenvectors generated using the KLE for the exponential covariance function (2.17) with 200 terms. We use $L_1 = 0.5$, $L_2 = 0.1$ and $\sigma^2 = 2.0$.

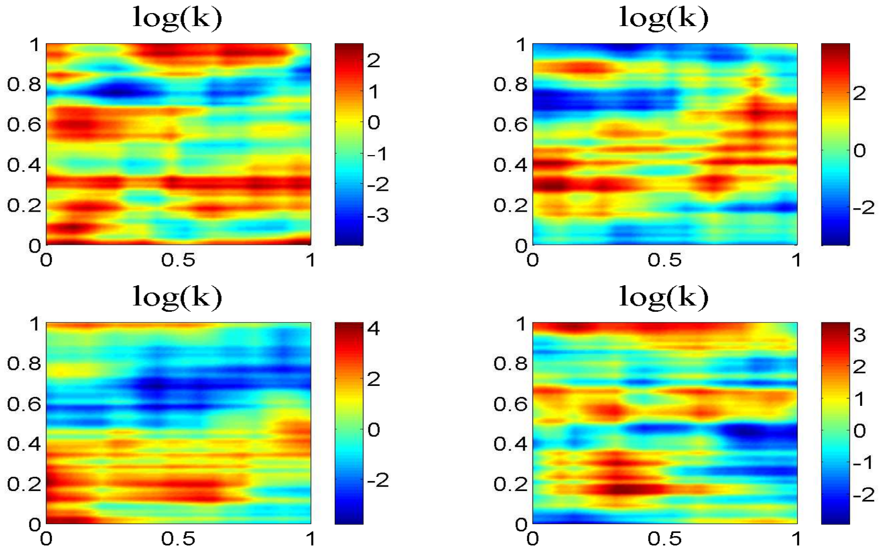


Fig. 2.6. Four example permeability fields generated using the KLE for the exponential covariance function (2.17) with 200 terms. In each permeability we use $L_1 = 0.5$, $L_2 = 0.1$ and $\sigma^2 = 2.0$.

elliptic equations, which we refer to as flow-based upscaling. We begin our discussion with the presentation of the coarse-scale equations in terms of upscaled permeability fields and two-phase flow. Next, we introduce the coarse-scale models for Richards' equation. The process of deriving coarse-scale models is similar for each.

2.3.1. Two-Phase Flow Case

Let us first discuss traditional upscaling procedures, where the coarse-scale pressure equation is of the same form as the fine-scale equation (2.2) but with an equivalent grid block permeability tensor replacing the fine-scale permeability field (see e.g., [9]). For two-phase flow the proposed coarse-scale model consists of the upscaling of the pressure equation (2.2) first. We then compute the coarse-scale velocity and coarse-scale saturation.

We consider single-phase Darcy's flow in a domain Ω

$$v = -k(x)\nabla p, \;\; \nabla \cdot v = f, \tag{2.18}$$

where v is velocity, k is the permeability, p is pressure, and f is the source. To capture the subgrid effects in two-phase flow simulations, typically, flow based subgrid capturing methods are used. These approaches compute the equivalent coarse grid permeability. We will briefly discuss the main idea of these approaches next. There are numerous ways to construct an equivalent grid block permeability tensor, k^*. For a coarse grid element K we define k^* by

$$k^* \left\langle \nabla \phi \right\rangle_K = \left\langle k \nabla \phi \right\rangle_K , \tag{2.19}$$

where

$$\langle \cdot \rangle_K = \frac{1}{K} \int_K (\cdot) \; dx$$

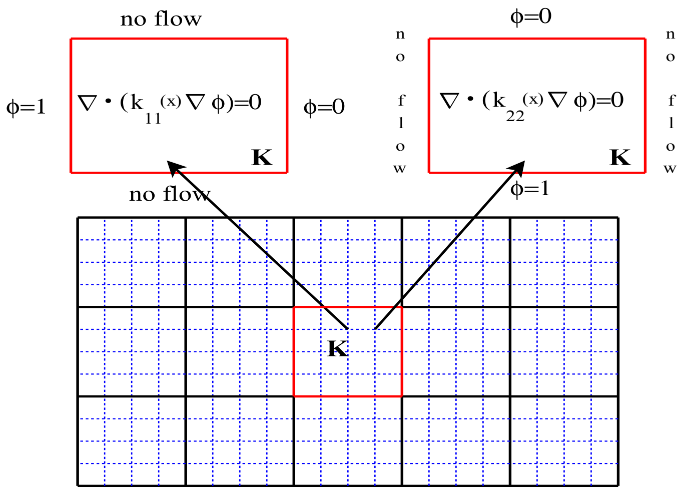


Fig. 2.7. Example boundary conditions for local upscaling problem. The upper left demonstrates boundary conditions in the x-direction and the upper right demonstrates boundary conditions in the y-direction.

is the volume average over K. In (2.19) ϕ is computed from the solution to (2.18) with some prescribed boundary conditions, as in Figure 2.7. Note that we must solve the local problems for each of the d dimensions in order to compute the full upscaled tensor k^*. In solving the local problems one can use various local boundary conditions, e.g. periodic. We refer to [20] for details.

We may consider various upscaling procedures, which we refer to as local, oversampled, and global. In a local upscaling procedure (2.18) is solved on the fine grid within each individual coarse block K. In an oversampled upscaling procedure (2.18) is solved in a slightly larger domain than the coarse block K. In a global upscaling procedure, (2.18) is solved globally on the fine-scale domain. In each procedure k^* is calculated within each coarse grid block from the calculated values of ϕ according to (2.19). It has been shown that local upscaling can introduce some errors due to the fact that the small scale features of the local solution along the boundaries are not consistent with those of the global solution. In order to improve the method, one needs to impose oscillatory boundary conditions for the local problems in a way that these oscillations embody the fine-scale features of the solution. Oversampling methods are introduced for this reason (see e.g.,[20]). The main idea of the oversampling technique is to use the local solutions in larger regions to compute the effective permeability over the target coarse grid block following (2.19). Oversampling reduces subgrid capturing error, because the boundary is outside the coarse block K and hence has less influence. One can further improve upscaling methods by using global information which takes into account non-local effects. Global upscaling approaches are particularly accurate for problems with strong non-local effects. Once the permeability is upscaled, the two-phase flow and transport equations are solved on the coarse grid.

We now discuss a different method to obtain upscaled equations, known as the multiscale finite volume element method (MsFVEM). This model is similar to the single-phase upscaling method. However, instead of coarsening the absolute permeability, we use pre-computed multiscale finite element basis functions. The key idea of the method is the construction of basis functions on the coarse grids such that these basis functions capture the small scale information on each of these coarse grids. The method that we use follows its finite element counterpart presented in [27]. The basis functions are constructed from the solution of the leading order homogeneous elliptic equation on each coarse element with some specified boundary conditions. Thus if we consider a coarse element K which has d vertices, the local basis functions $\phi^i, i = 1, \cdots, d$ satisfy the following elliptic problem:

$$\begin{aligned} -\nabla \cdot (k \cdot \nabla \phi^i) = 0 \quad \text{in } K, \\ \phi^i = g^i \quad \text{on } \partial K, \end{aligned} \tag{2.20}$$

for some function g^i defined on the boundary of the coarse element K. Hou et al.

[27] have demonstrated that a careful choice of boundary condition would guarantee the performance of the basis functions to incorporate the local information and hence improve the accuracy of the method. We note that in previous findings, the function g_i for each i is chosen either to vary linearly along ∂K, to be the solution of the local one-dimensional problems [29], or to be the solution of the problem in a slightly larger domain. If function g^i for each i varies linearly along ∂K then, for example, in the case of a constant diagonal tensor the solution of (2.20) would be a standard linear/bilinear basis function.

Referring to Figure 2.8, we define our problem as follows. Let $\mathcal{K}^h$ denote the collection of coarse elements (rectangles in our case) K. Consider a coarse element K and let ξ_K denote its center. Element K is divided into four rectangles of equal area by connecting ξ_K to the midpoints of the element's edges. We denote these quadrilaterals by K_ξ, where $\xi \in Z_h(K)$ are the vertices of K. Also, we denote $Z_h = \bigcup_K Z_h(K)$ and $Z_h^0 \subset Z_h$ the vertices which do not lie on the Dirichlet boundary of Ω. The control volume V_ξ is defined as the union of the quadrilaterals K_ξ sharing the vertex ξ. We note that, as usual, we require $\phi^i(\xi_j) = \delta_{ij}$. Finally, a nodal basis function associated with the vertex ξ in the domain Ω is constructed from the combination of the local basis functions that share this ξ and zero elsewhere. These nodal basis functions are denoted by $\{\psi_\xi\}_{\xi \in Z_h^0}$.

Having described the basis functions, we denote by V^h the space of our approximate pressure solution which is spanned by the basis functions $\{\psi_\xi\}_{\xi \in Z_h^0}$. Now we may formulate the finite dimensional problem corresponding to the finite volume element formulation of (2.2). A statement of mass conservation on a control volume V_ξ is formed from (2.2), where now the approximate solution is written as a linear combination of the basis functions. Assembly of this conservation statement for all control volumes would give the corresponding linear system of equations that can

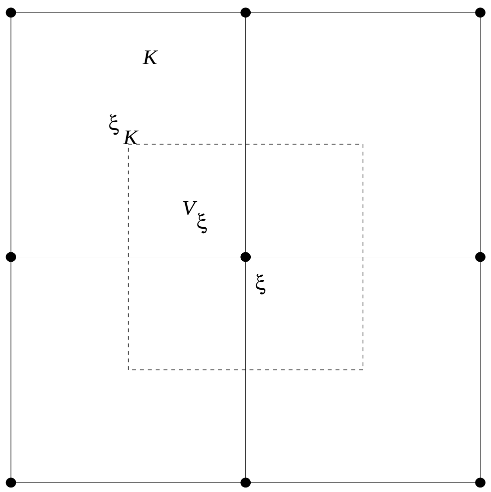


Fig. 2.8. Nodal points and grid for the MsFVEM.

be solved accordingly. The resulting linear system has incorporated the fine-scale information through the involvement of the nodal basis functions on the approximate solution. Specifically, the problem now is to seek $p^h \in V^h$ with $p^h = \sum_{\xi \in Z_h^0} p_\xi \psi_\xi$ such that

$$\int_{\partial V_\xi} \lambda(S) k \cdot \nabla p^h \cdot n \, dl = \int_{V_\xi} f \, dA \tag{2.21}$$

for every control volume $V_\xi \subset \Omega$. Here n defines the unit normal vector on the boundary of the control volume, ∂V_ξ, and S is the fine-scale saturation field at this point. We note that, concerning the basis functions, a vertex-centered finite volume difference is used to solve (2.20), and a harmonic average is employed to approximate the permeability k at the edges of fine control volumes.

Furthermore, the pressure solution may be used to compute the total velocity field at the coarse-scale level, denoted by $\overline{v} = (\overline{v}_x, \overline{v}_z)$ via (2.6). In general, the following equations are used to compute the velocities in horizontal and vertical directions, respectively:

$$\overline{v}_x = -\frac{1}{h_z} \sum_{\xi \in Z_h^0} p_\xi \left(\int_E \lambda(S) k_x \frac{\partial \psi_\xi}{\partial x} dz \right), \tag{2.22}$$

$$\overline{v}_z = -\frac{1}{h_x} \sum_{\xi \in Z_h^0} p_\xi \left(\int_E \lambda(S) k_z \frac{\partial \psi_\xi}{\partial z} dx \right), \tag{2.23}$$

where E is the edge of V_ξ. Furthermore, for the control volumes V_ξ adjacent to the Dirichlet boundary (which are half control volumes), we can derive the velocity approximation using the conservation statement derived from (2.2) on V_ξ. One of the terms involved is the integration along part of the Dirichlet boundary, while the rest of the three terms are known from the adjacent internal control volumes calculations. The detailed analysis of the two-scale finite volume method can be found in [22].

As for the upscaling of the saturation equation, we only use the coarse-scale velocity to update the saturation field on the coarse grid, i.e.,

$$\frac{\partial \overline{S}}{\partial t} + \overline{v} \cdot \nabla f(\overline{S}) = 0, \tag{2.24}$$

where $\overline{S}$ denotes the saturation on the coarse grid. In this case the upscaling of the saturation equation does not take into account subgrid effects. This kind of upscaling techniques in conjunction with the upscaling of absolute permeability are commonly used in applications (see e.g. [10, 11, 12]). The difference of our approach is that the coupling of the small scales is performed through the finite volume element formulation of the pressure equation. One can also try to upscale the saturation equation and couple it to MsFVEM. This is done in [15, 14]. Due to the strong distant effects, the upscaling of the saturation equation is difficult. In this dissertation, we will not consider any subgrid treatment for the saturation equation, though the proposed approaches can be used in conjunction with upscaled saturation equation in uncertainty quantification problems. One can also solve the saturation equation on the fine grid combined with the coarse grid solution of the pressure equation as it is done in [29, 17].

2.3.2. Richards' Equation Case

For Richards' equation, because we are interested in mass conservative schemes, a finite volume formulation of the global problem will be used. The coarse-scale equations will be described for only the finite volume formulation and for a rectangular mesh. For (2.7), our goal is to find $u^h \in S^h$ such that

$$\int_{V_z} (\theta(\eta^{u_h}) - \theta^{n-1})\, dx - \Delta t \int_{\partial V_z} K(x, \eta^{u_h})\, D_x u_{\epsilon,h} \cdot n\, dl = 0 \quad \forall z \in Z_h^0, \tag{2.25}$$

where θ^{n-1} is the value of $\theta(\eta^{u_h})$ evaluated at time step $n-1$, and $u_{\epsilon,h} \in V_\epsilon^h$ is a function that satisfies the boundary value problem:

$$\begin{aligned} -\mathrm{div}(K(x, \eta^{u_h}) D_x u_{\epsilon,h}) &= 0 \quad \text{in } K \in S^h, \\ u_{\epsilon,h} &= u_h \quad \text{on } \partial K. \end{aligned} \tag{2.26}$$

Here V_z is the control volume surrounding the vertex $z \in Z_h^0$ and Z_h^0 is the collection of all vertices that do not belong to the Dirichlet boundary as before. At this point, we can perform single-phase upscaling procedures to the elliptic equation exactly the same as in the previous description of two-phase flow upscaling.

We again consider the use of the MsFVEM. The MsFVEM can be applied to Richards' equations of general form as it was shown in [17], however, the MsFVEM offers a great advantage when the nonlinearity and heterogeneity of $k(x,u)$ is separable, i.e.,

$$k(x,u) = K_s(x)\, k_r(u). \tag{2.27}$$

In this case, the local basis functions become linear and the corresponding space V_ϵ^h is a linear space. We consider the same exact setup as the MsFVEM for two-phase immiscible flow, again referring to Figure 2.8. The local basis functions $\phi_i, i = 1, \cdots, d$ are set to satisfy the elliptic problem given by (2.20) and we again require $\phi_i(\xi_j) = \delta_{ij}$.

We would like to note that one can use an approximate solution of (2.20) when it is possible. For example, in the case of periodic or scale separation cases, the basis functions can be approximated using homogenization expansion (see [18]). This type of simplification is not applicable for the problems we consider.

Now, we may formulate the finite dimensional problem. We want to find a $u_{\epsilon,h} \in V_\epsilon^h$ with $u_{\epsilon,h} = \sum_{z\in Z_h^0} p_z \phi_i$ such that

$$\int_{V_z} (\theta(\eta^{u_h}) - \theta^{n-1})\, dx - \Delta t \int_{\partial V_z} K_s(x)\, k_r(\eta^{u_h})\, D_x u_{\epsilon,h} \cdot n\, dl = 0 \tag{2.28}$$

for every control volume $V_z \subset \Omega$. To this equation we can directly apply a linearization procedure, as described in [23]. Let us denote

$$r^m = u_{\epsilon,h}^m - u_{\epsilon,h}^{m-1}, \quad m = 1, 2, 3, \cdots, \tag{2.29}$$

where $u_{\epsilon,h}^m$ is the iterate of $u_{\epsilon,h}$ at the iteration level m. Thus, we want to find $r^m = \sum\limits_{z\in Z_h^0} r_z^m \psi_z$ such that for $m = 1, 2, 3, \cdots$ until convergence we have

$$\int_{V_z} C(\eta^{u_h, m-1})\, r^m\, dx - \Delta t \int_{\partial V_z} K_s(x)\, k_r(\eta^{u_h, m-1})\, D_x r^m \cdot n\, dl = R^{h,m-1}, \tag{2.30}$$

with

$$R^{h,m-1} = -\int_{V_z} (\theta(\eta^{u_h,m-1}) - \theta^{n-1})\, dx + \Delta t \int_{\partial V_z} K_s(x)\, k_r(\eta^{u_h,m-1})\, D_x u_{\epsilon,h}^{m-1} \cdot n\, dl. \tag{2.31}$$

The superscript m at each of the functions means that the corresponding functions are evaluated at an iteration level m. Numerical examples presented in [18] show that MsFVEM can provide an accurate approximation for the solution.

CHAPTER III

SOLUTION TECHNIQUES FOR STOCHASTIC EQUATIONS

In this chapter, we consider different techniques used for solving the stochastic equations arising in porous media. After using the KLE to express the permeability or saturated conductivity, we can write our model equations as stochastic equations. Using the KLE we write $k(x,\theta)$ and $K_s(x,\theta)$ for the permeability and conductivity, respectively, where θ is a Gaussian random variable. Our goal in this section is not to compare the solution techniques for the stochastic equations to each other, but instead provide a description of how both methods can be used for our porous media equations.

3.1. Polynomial Chaos Methods

In this section, we will give some necessary information on polynomial chaos methods. In the following sections, we will consider the applications. Consider the Hilbert space $L^2(R,\mu)$, where μ is the Gaussian measure

$$\mu(dx) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx.$$

Here dx denotes the regular Lebesgue measure. The inner product in space $L^2(R,\mu)$ is defined as

$$(f,\ g) = \int_{-\infty}^{+\infty} f(x)g(x)\mu(dx) = \int_{-\infty}^{+\infty} f(x)g(x)\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx.$$

We define the unnormalized Hermite polynomial as

$$P_n(x) = (-1)^n e^{\frac{x^2}{2}} \frac{d^n}{dx^n}\left(e^{-\frac{x^2}{2}}\right) \tag{3.1}$$

and the normalized Hermite polynomial as

$$H_n(x) = (n!)^{-\frac{1}{2}} P_n(x) = (n!)^{-\frac{1}{2}} (-1)^n e^{\frac{x^2}{2}} \frac{d^n}{dx^n} \left(e^{-\frac{x^2}{2}} \right), \tag{3.2}$$

where $n = 0, 1, 2, \ldots$. It is well known that $H_n(x)$ form a complete orthonormal bases in $L^2(R, \mu)$. Thus, for any $f(x) \in L^2(R, \mu)$, there exists a Fourier-Hermite expansion

$$f(x) = \sum_{n=0}^{\infty} f_n H_n(x), \qquad f_n = (f,\ H_n). \tag{3.3}$$

Further, from the isometric property of the Fourier expansion, we have

$$\|f\|_2^2 = (f, f) = \sum_{n=0}^{\infty} f_n^2.$$

In the space $L^2(R, \mu)$, we can also view the variable x as a unit Gaussian random variable. Then the inner product can be interpreted as an expectation with respect to the Gaussian random variable. Specifically for the Fourier-Hermite expansion (3.3), we have

$$f_n = (f,\ H_n) = E[f(x)H_n(x)], \qquad Ef^2(x) = \sum_{n=0}^{\infty} f_n^2.$$

Further,

$$E[H_n(x)] = \int_{-\infty}^{+\infty} H_n(x) d\mu(x) = (H_n,\ 1) = 0, \qquad \text{if} \qquad n \neq 0. \tag{3.4}$$

Therefore we have

$$Ef(x) = f_0.$$

We can expand a function $f(x)$ of Gaussian random variable into Fourier-Hermite polynomials. The mean of $f(x)$ is the first coefficient of the expansion, and the second moment of $f(x)$ is the summation of the square of each coefficients. Because of this fact, the Fourier-Hermite expansion provides a useful tool to study Gaussian randomness.

Let us now recall some useful results about Hermite polynomials which will be used in later discussions. Denote

$$\psi(x,\ t) = e^{-\frac{t^2}{2}+xt}.$$

$\psi(x,\ t)$ is usually called the generating function of Hermite polynomials because

$$\psi(x,\ t) = e^{\frac{x^2}{2}} e^{-\frac{(t-x)^2}{2}} = e^{\frac{x^2}{2}} \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \frac{d^n}{dx^n} \left(e^{-\frac{x^2}{2}} \right) t^n = \sum_{n=0}^{\infty} \frac{P_n(x)}{n!} t^n. \tag{3.5}$$

That is, the coefficients of the Taylor expansion of $\psi(x,\ t)$ are exactly the unnormalized Hermite polynomials.

If we differentiate the unnormalized Hermite polynomial (3.1) directly we get

$$P_n'(x) = xP_n(x) - P_{n+1}. \tag{3.6}$$

On the other hand, by differentiating both sides of equation (3.5), we have

$$\frac{\partial}{\partial x}\psi(x,\ t) = t\psi(x,\ t) = \sum_{n=0}^{\infty} \frac{P_n(x)}{n!} t^{n+1} = \sum_{n=0}^{\infty} \frac{P_n'(x)}{n!} t^n.$$

Shifting the summation index we get

$$P_n'(x) = nP_{n-1}(x). \tag{3.7}$$

Combining (3.6) and (3.7) we get

$$P_{n+1}(x) - xP_n(x) + nP_{n-1}(x) = 0, \qquad n = 1,\ 2, \ldots, \tag{3.8}$$

with $P_{-1}(x) = 0$, $P_0(x) = 1$. From (3.2) we have

$$H_{n+1}(x) - x\sqrt{\frac{1}{n+1}} H_n(x) + \sqrt{\frac{n}{n+1}} H_{n-1}(x) = 0, \tag{3.9}$$

$$H_{-1}(x) = 0, \qquad H_0(x) = 1.$$

Since the product of Hermite polynomials is still a polynomial, it can be expanded as a linear combination of Hermite polynomials. We have the following theorem.

Theorem 3.1.1. *For any nonnegative integers α and β, denote $\alpha \wedge \beta = \min\{\alpha,\ \beta\}$. We have*

$$H_\alpha(x)H_\beta(x) = \sum_{p \le \alpha\wedge\beta} B(\alpha, \beta, p) H_{\alpha+\beta-2p}(x), \tag{3.10}$$

where

$$B(\alpha, \beta, p) = \binom{\alpha}{p}\binom{\beta}{p} p! \frac{\sqrt{(\alpha+\beta-2p)!}}{\sqrt{\alpha!\beta!}}. \tag{3.11}$$

Proof. From equation (3.5) we have

$$\psi(x,\ t)\psi(x,\ s) = \sum_{\alpha=0}^{\infty}\sum_{\beta=0}^{\infty} \frac{P_\alpha(x)P_\beta(x)}{\alpha!\beta!} t^\alpha s^\beta. \tag{3.12}$$

On the other hand

$$\begin{aligned}
\psi(x,t)\psi(x,s) &= e^{-\frac{t^2+s^2}{2}+(t+s)x} = e^{ts}e^{-\frac{(t+s)^2}{2}+(t+s)x} \\
&= \sum_{p=0}^{\infty} \frac{(ts)^p}{p!} \sum_{k=0}^{\infty} \frac{P_k(x)}{k!}(t+s)^k \\
&= \sum_{p=0}^{\infty} \sum_{k=0}^{\infty} \frac{P_k(x)}{p!} \sum_{0\le m \le k} \frac{1}{k!}\binom{k}{m} t^{m+p}s^{k+p-m}.
\end{aligned}$$

If we let $k = m + \nu$, then $m \le k$ is equivalent to $\nu \ge 0$. Then the above formula can be rewritten as

$$\psi(x,t)\psi(x,s) = \sum_{p=0}^{\infty} \sum_{m=0}^{\infty} \sum_{\nu=0}^{\infty} \frac{P_{m+\nu}(x)}{p!\ m!\ \nu!} t^{m+p}s^{\nu+p}.$$

Denote $m + p = \alpha$ and $\nu + p = \beta$. Since $m = \alpha - p \ge 0$ and $\nu = \beta - p \ge 0$ we have

$p \leq \alpha \wedge \beta$. Then the previous summation changes to

$$\begin{aligned}\psi(x,t)\psi(x,s) &= \sum_{\alpha=0}^{\infty}\sum_{\beta=0}^{\infty}\left(\sum_{\substack{m+p=\alpha \\ \nu+p=\beta}} \frac{P_{m+\nu}(x)}{p!\; m!\; \nu!}\right) t^{\alpha}s^{\beta} \\ &= \sum_{\alpha=0}^{\infty}\sum_{\beta=0}^{\infty}\sum_{p\leq\alpha\wedge\beta}\frac{P_{\alpha+\beta-2p}(x)}{p!\;(\alpha-p)!\;(\beta-p)!}t^{\alpha}s^{\beta}.\end{aligned}$$

Comparing the above equation with (3.12) we get

$$P_{\alpha}(x)P_{\beta}(x) = \sum_{p\leq\alpha\wedge\beta}\frac{\alpha!\beta!}{p!\;(\alpha-p)!\;(\beta-p)!}P_{\alpha+\beta-2p}(x). \tag{3.13}$$

Note $P_n(x) = (n)^{\frac{1}{2}}H_n(x)$. Substituting this into the above formula immediately gives us (3.10). □

Suppose $\theta_1, \theta_2, \ldots$ are independent unit Gaussian random variables. Denote the Gaussian vector as $\theta = (\theta_1, \theta_2, \ldots)$. Define the multi-index set as

$$\mathcal{J} = \left\{\alpha = (\alpha_1, \alpha_2, \ldots) \mid \alpha_i \in \{0,1,2,\ldots\},\; |\alpha| = \sum_{i=1}^{\infty}\alpha_i < \infty\right\}.$$

For each $\alpha \in \mathcal{J}$, define the multi-variable Hermite polynomial of θ as

$$T_{\alpha}(\theta) = \prod_{i=1}^{\infty} H_{a_i}(\theta_i). \tag{3.14}$$

Note that each $T_{\alpha}(\theta)$ only involves a finite number of factors. The multi-variable Hermite polynomial $T_{\alpha}(\theta)$ is usually called the Wick polynomial.

Now, let us introduce some notation that will be used in the context of chaos expansions. For any $\alpha, \beta \in \mathcal{J}$, denote $\alpha \wedge \beta = \min\{\alpha_i,\ \beta_i\}$ and $\alpha \vee \beta = \max\{\alpha_i,\ \beta_i\}$. We say $\beta \leq \alpha$ if $\beta_i \leq \alpha_i$ for all $i \geq 1$. The operation $\alpha \pm \beta$ is also defined for each component. We denote $\alpha! = \prod_i \alpha_i!$.

The following properties of the Wick polynomials are especially important:

1. The set $\{T_\alpha, \quad \alpha \in \mathcal{J}\}$ are orthonormal bases:

$$E[T_\alpha T_\beta] = \begin{cases} 0 & \text{if} \quad \alpha \neq \beta \\ 1 & \text{if} \quad \alpha = \beta. \end{cases}$$

2. Since $T_0 = 1$, where 0 denotes the zero index, we have $E[T_0] = 1$, and

$$E[T_\alpha] = E[T_\alpha T_0] = 0 \quad \text{if} \quad \alpha \neq 0.$$

3. For any function $f(\theta_1, \theta_2, \ldots)$ with $E(f^2) < +\infty$, we have

$$f(\theta_1, \theta_2, \ldots) = \sum_{\alpha \in \mathcal{J}} f_\alpha T_\alpha, \qquad f_\alpha = E(fT_\alpha),$$

   and

$$Ef = f_0, \qquad Ef^2 = \sum_{\alpha \in \mathcal{J}} f_\alpha^2.$$

4. Theorem 3.1.1 can be generalized to multi-variable Hermite polynomials easily.

$$T_\alpha T_\beta = \sum_{p \leq \alpha \wedge \beta} B(\alpha, \beta, p) T_{\alpha+\beta-2p}. \tag{3.15}$$

It is important that the product of the Hermite polynomials can be expressed as a linear combination of the Hermite polynomials themselves. The following lemma is especially useful in practice.

Lemma 3.1.1. *Suppose u, v have the Fourier-Hermite expansion*

$$u = \sum_{\alpha \in \mathcal{J}} u_\alpha T_\alpha, \qquad v = \sum_{\beta \in \mathcal{J}} v_\beta T_\beta.$$

Then, the product uv has the following expansion:

$$uv = \sum_{\theta \in \mathcal{J}} \left(\sum_{p \in \mathcal{J}} \sum_{0 \leq \gamma \leq \theta} C(\theta,\ \gamma,\ p)\, u_{\theta-\gamma+p}\, v_{\gamma+p} \right) T_\theta, \tag{3.16}$$

where

$$C(\theta,\ \gamma,\ p) = \left[\binom{\theta}{\gamma} \binom{\gamma+p}{p} \binom{\theta-\gamma+p}{p} \right]^{\frac{1}{2}}. \tag{3.17}$$

Proof. From property (3.15), we have

$$\begin{aligned} uv &= \sum_{\alpha \in \mathcal{J}} \sum_{\beta \in \mathcal{J}} u_\alpha v_\beta T_\alpha T_\beta \\ &= \sum_{\alpha \in \mathcal{J}} \sum_{\beta \in \mathcal{J}} u_\alpha\ v_\beta \sum_{p \leq \alpha \wedge \beta} \binom{\alpha}{p} \binom{\beta}{p} p! \frac{\sqrt{(\alpha+\beta-2p)!}}{\sqrt{\alpha!\ \beta!}} T_{\alpha+\beta-2p}. \end{aligned}$$

Changing the variables $\theta = \alpha+\beta-2p$, $\gamma = \beta - p$ and substituting $\alpha = \theta + p - \gamma$, $\beta = \gamma + p$, we immediately obtain formula (3.16), which completes the proof. □

3.2. Collocation Methods

In this section, we introduce pertinent information on collocation methods for use in later sections. Suppose we wish to approximate functions $f : [-1,1]^N \to \mathbb{R}$ using only the known values of f at some locations in $[-1,1]^N$. One may consider two different problems in this situation: the first where the known values are given by scattered data in $[-1,1]^N$ and the other where the approximation is based on values at previously chosen points (gridded data). We will consider only the latter, since interpolation from scattered data in high dimensions remains a challenging problem. For simplicity, we consider approximation via Lagrange interpolation in high dimensions.

First, recall one dimensional Lagrange interpolation. We have a function $f : [-1,1] \to \mathbb{R}$ given at nodes $\theta_j, j = 1, \ldots, M$ for some $M \in \mathbb{N}$, and we wish to find a polynomial of degree $M-1$ that interpolates the function at the given nodes. In order to construct this polynomial we form the Lagrange polynomials $L_k(\theta)$ such that $L_k\left(\theta_j\right)=\delta_{jk}$ and $L_k(\theta)\in\Pi_{M-1}$. Then our polynomial approximation is given by

$$\mathcal{I}(f)(\theta)=\sum_{j=1}^{M} f\left(\theta_j\right) L_j(\theta).$$

In multiple dimensions, let us define interpolation for each dimension $i=1,\ldots,N$ by

$$U_i(f)(\theta)=\sum_{j=1}^{M_i} f\left(\theta_j^i\right) L_j^i(\theta),$$

where M_i is the number of nodes in the i^{th} dimension. A logical multidimensional extension of Lagrange interpolation in $\mathbb{R}^N$ is to use a simple tensor product of the one dimensional interpolants. Such an approximation is given by

$$\left(U_{i_1}\otimes\cdots\otimes U_{i_N}\right)(f)=\sum_{j_1=1}^{M_{j_1}}\cdots\sum_{j_N=1}^{M_{j_N}} f\left(\theta_{j_1}^{i_1},\ldots,\theta_{j_N}^{i_N}\right)\cdot\left(L_{j_1}^{i_1}\otimes\cdots\otimes L_{j_N}^{i_N}\right). \qquad (3.18)$$

If we were to use the above formulation as our multidimensional interpolant, then we need $M_{j_1}\cdots M_{j_N}$ function values. Specifically, if we were to use M nodes in each dimension, then M^N values are required for full grid interpolation. In many practical applications, it is not unusual to have $N\geq 10$. For example, if $N=10$ and $M=4$ we have 4^{10} or $1,048,576$ values needed for interpolation. We are generally interested in applications in which a single function value results from the solution to a nonlinear PDE system, thus full tensor product interpolation is prohibitively expensive. Sparse grid collocation methods can be used to alleviate this problem.

3.2.1. Sparse Grid Collocation

We now consider sparse grid collocation methods, specifically the Smolyak algorithm introduced in [41]. The Smolyak algorithm is a linear combination of product formulas chosen so that an interpolation property for $N = 1$ is preserved for $N > 1$. We let $|\mathbf{i}| = i_1 + \cdots + i_N$ for $\mathbf{i} \in \mathbb{N}^N$. Using notation from [35, 46] we define the Smolyak algorithm by

$$A(q, N) = \sum_{q-N+1 \leq |\mathbf{i}| \leq q} (-1)^{q-|\mathbf{i}|} \cdot \begin{pmatrix} N-1 \\ q-|\mathbf{i}| \end{pmatrix} \cdot (U_{i_1} \otimes \cdots \otimes U_{i_N}). \tag{3.19}$$

Note that we must evaluate f at only sparse values given by

$$H(q, N) = \bigcup_{q-N+1 \leq |\mathbf{i}| \leq q} (\Theta_{i_1} \times \cdots \times \Theta_{i_N}), \tag{3.20}$$

where $\Theta_i = \left\{\theta_1^i, \ldots, \theta_{M_i}^i\right\}$ are the set of points used by U_i. This leads us to $n(k + N, N) \approx \frac{2^k}{k!} \cdot N^k$ nodes used by $A(N + k, N)$. Here the k term determines how many nodes will be used. For a fixed N, we define $A(N + k, N)$ as k^{th} order Smolyak interpolation. The smallest order, of course, is $k = 1$, in which case we will have only $2N + 1$ nodes.

As suggested by numerous sources [35, 46, 4], we consider Smolyak formulas that are based on the extrema of Chebyshev polynomials. We choose

$$\theta_j^i = -\cos\frac{\pi \cdot (j-1)}{M_i - 1}, \; j = 1, \ldots, M_i$$

and define $\theta_1^i = 0$ for $M_i = 1$. We also choose $M_1 = 1$ and $M_i = 2^{i-1} + 1$ for $i > 1$. This has the benefit of making our nodal sets nested, thus $H(q, N) \subset H(q + 1, N)$.

Using the Smolyak formulas and Lagrange interpolation, we have that $A(N + k, N)$ is exact for all polynomials of degree k. Using techniques described in [4] we have the one dimensional error estimate given by

$$\|f - U_i(f)\|_\infty \leq E_{M_i - 1}(f) \cdot (1 + \Lambda_{M_i}),$$

where E_M is the error of the best approximation by polynomials $p \in \mathbb{P}(M, 1)$ and Λ_M

is the Lebesgue constant for the Chebyshev polynomials. We have the estimate

$$\Lambda_M \leq \frac{2}{\pi} \log(M-1) + 1$$

for $M \geq 2$. In multiple dimensions we define the space

$$F_N^k = \left\{ f : [-1,1]^N \to \mathbb{R} \,|\, D^\alpha f \, \text{continuous if } \; \alpha_i \leq k \,\forall\, i \right\}$$

with norm

$$\|f\| = \max \left\{ \|D^\alpha f\|_\infty \,|\, \alpha \in \mathbb{N}_0^N, \, \alpha_i \leq k \right\}.$$

We find an interpolation error of

$$\|I_N - A(N+k,N)\| \leq c_{N,k} \cdot n^{-k} \cdot (\log n)^{(k+2)(d-1)+1}, \tag{3.21}$$

where I_N is the identity operator [4, 46, 35].

3.3. Application to Single-Phase Flow

We consider approximations to the stochastic flow equations using both polynomial chaos and collocation methods. In this section, to ease both notations and computations, we consider only single-phase flow given by

$$\begin{aligned} &\nabla \cdot (k \nabla p) = h, \\ &v = -k \nabla p, \\ &\tfrac{\partial S}{\partial t} + v \cdot \nabla S = 0. \end{aligned} \tag{3.22}$$

We assume the spatial domain is given by Ω and the permeability, k, is given by the KLE. We further assume that the permeability satisfies the strong elliptic condition. Thus there are constants $k_{\min}$ and $k_{\max}$ such that

$$0 < k_{\min} \leq k(x) \leq k_{\max} < \infty. \tag{3.23}$$

As we will see later, the lower bound of the permeability is important in proving the polynomial chaos matrix equation for the elliptic system is symmetric and positive definite. Note k is a function of both space and the multidimensional Gaussian random variable θ. Further, v, p and S are also functions of θ. We will generally omit this notation and simply write $k(x)$ instead of $k(x, \theta)$ and likewise for v, p, and S.

3.3.1. Polynomial Chaos

We assume the permeability field k is given by

$$k(x) = \kappa(x) + \delta e^{Y(x)}, \tag{3.24}$$

where $\kappa(x)$ is a deterministic function, $\delta > 0$ is a constant, and $Y(x)$ is a homogeneous isotropic Gaussian field. We assume

$$0 < a \leq \kappa(x), \ \ x \in \Omega, \tag{3.25}$$

which is a legitimate assumption from the strong elliptic condition (3.23). Since $\delta > 0$ we have

$$0 < a \leq \kappa(x) \leq k(x), \ \ x \in \Omega.$$

Using these assumptions, we can show the elliptic equation in (3.22) has a unique weak solution in a corresponding Hibert space. It is this property we wish to retain in the polynomial chaos formulation.

We expand $k(x)$ into Fourier-Hermite polynomials of θ using the KLE (2.14). We find

$$k(x) = \sum_{\alpha \in \mathcal{J}} E\left[e^{Y(x,\theta)} T_\alpha(\theta)\right] T_\alpha(\theta) \triangleq \sum_{\alpha \in \mathcal{J}} k_\alpha(x) T_\alpha(\theta). \tag{3.26}$$

We now wish to find an expression for the $k_\alpha(x)$ coefficients in terms of the KLE. To do this, we need the following lemma.

Lemma 3.3.1. *Suppose $H_n(x)$ is the n-th order normalized Hermite polynomial and θ is a standard Gaussian random variable, then*

$$E\left[H_n(\theta+a)\right]=\frac{a^n}{\sqrt{n!}}\,,$$

where a is an arbitrary deterministic number.

Proof. From (3.5) we can write

$$\psi(x+a,t)=\sum_{n=0}^{\infty}\frac{P_n(x+a)}{n!}t^n,$$

but we also have

$$\begin{aligned}\psi(x+a,t)=e^{-\frac{t^2}{2}+xt}e^{at}&=\left(\sum_{i=0}^{\infty}\frac{a^i}{i!}t^i\right)\left(\sum_{j=0}^{\infty}\frac{P_j(x)}{j!}t^j\right)\\&=\sum_{n=0}^{\infty}\left(\sum_{i=0}^{n}\frac{a^i}{i!}\frac{P_{n-i}(x)}{(n-i)!}\right)t^n.\end{aligned} \tag{3.27}$$

Equating the two formulas, we can conclude

$$P_n(x+a)=\sum_{i=0}^{n}\frac{a^i}{i!}\frac{n!}{(n-i)!}P_{n-i}(x).$$

Note that $E\left[P_k(\theta)\right]=\delta_{k,0}$ because $P_k(x)$, $k=0,1,\ldots$ are orthogonal with respect to Gaussian measure and $P_0(x)=1$. The desired result follows, as

$$E\left[H_n(\theta+a)\right]=\frac{E\left[P_n(\theta+a)\right]}{\sqrt{n!}}=\frac{a^n}{\sqrt{n!}}.$$

□

Using the previous lemma, we can calculate the coefficients in the expansion of k.

Lemma 3.3.2. *If $k(x)=\kappa(x)+\delta e^{Y(x)}$ and k has the expansion given by (3.26) then*

$$k_0(x)=\kappa(x)+\delta e^{\sigma^2/2}$$

and

$$k_\alpha(x) = e^{\sigma^2/2} \prod_{i=1}^{\infty} \frac{\left(\sqrt{\lambda_i}\phi_i(x)\right)^{\alpha_i}}{\sqrt{\alpha_i!}}, \ \alpha \neq 0. \tag{3.28}$$

Proof. From the previous lemma

$$\begin{aligned} E\left[e^{Y(x,\theta)} T_\alpha(\theta)\right] &= \prod_{i=1}^{\infty} \left(\int_{-\infty}^{\infty} e^{\sqrt{\lambda_i}\theta_i\phi_i} H_{\alpha_i}\left(\theta_i\right) e^{-\frac{-\theta_i^2}{2} d\theta_i} \right) \qquad (3.29) \\ &= \prod_{i=1}^{\infty} e^{\frac{1}{2}\lambda_i\phi_i^2} E\left[H_{\alpha_i}\left(\theta_i + \sqrt{\lambda_i}\phi_i\right)\right] \\ &= e^{\sigma^2/2} \prod_{i=1}^{\infty} \frac{\left(\sqrt{\lambda_i}\phi_i\right)^{\alpha_i}}{\sqrt{\alpha_i!}}, \end{aligned}$$

where the last equation comes from the fact that $\sigma^2 = E\left(Y^2\right) = \sum \lambda_i \phi_i^2$. We now have

$$k_0(x) = E\left[k(x)\right] = \kappa(x) + E\left[\delta e^{Y(x)}\right] = \kappa(x) + \delta e^{\sigma^2/2}$$

and

$$k_\alpha(x) = E\left[\delta e^{Y(x)} T_\alpha(\theta)\right] = \delta e^{\sigma^2/2} \prod_{i=1}^{\infty} \frac{\left(\sqrt{\lambda_i}\phi_i(x)\right)^{\alpha_i}}{\sqrt{\alpha_i!}}.$$

□

Since the permeability $k(x)$ depends on the Gaussian random variables θ_i so do p, v, and S in (3.22). We can expand each of these and find

$$\begin{aligned} p(x) &= \sum_{\alpha\in\mathcal{J}} p_\alpha(x) T_\alpha(\theta), \\ v(x) &= \sum_{\alpha\in\mathcal{J}} v_\alpha(x) T_\alpha(\theta), \qquad (3.30) \\ S(x,t) &= \sum_{\alpha\in\mathcal{J}} S_\alpha(x,t) T_\alpha(\theta). \end{aligned}$$

Since we are interested in the expansion for single-phase flow, we multiply both sides of the elliptic equation in (3.22) by $T_\alpha(\theta)$ and take expectations. We find

$$\begin{aligned} E\left[hT_\alpha\right] &= \nabla\cdot E\left[\left(k\nabla p\right)T_\alpha\right] \\ &= \nabla\cdot E\left[k\left(\sum_{\gamma\in\mathcal{J}} p_\gamma(x)T_\gamma\right)T_\alpha\right] \\ &= \nabla\sum_{\gamma\in\mathcal{J}}\cdot E\left[kT_\alpha T_\gamma\right]\nabla p_\gamma. \end{aligned}$$

We denote

$$A_{\alpha,\gamma}(x) = E\left[kT_\alpha T_\gamma\right]$$

and using (3.15) we get

$$\begin{aligned} A_{\alpha,\gamma}(x) &= \sum_{p\le\alpha\wedge\beta} B(\alpha,\beta,p)E\left(kT_{\alpha+\gamma-2p}\right) \\ &= \sum_{p\le\alpha\wedge\beta} B(\alpha,\beta,p)k_{\alpha+\gamma-2p}. \end{aligned} \tag{3.31}$$

Using all of the above, we can now write the governing equations for the polynomial chaos coefficients for single-phase flow:

$$\begin{aligned} \nabla\cdot\left(\sum_{\gamma\in\mathcal{J}} A_{\alpha,\gamma}(x)\nabla p_\gamma(x)\right) &= hI_{\alpha=0}, \\ v_\alpha &= \sum_{\gamma\in\mathcal{J}} A_{\alpha,\gamma}(x)\nabla p_\gamma(x), \\ \frac{\partial S_\alpha}{\partial t} + \sum_{p\in\mathcal{J}}\sum_{0\le\beta\le\alpha} C\left(\alpha,\beta,p\right)\nabla\cdot\left(v_{\alpha-\beta+p}S_{\beta+p}\right) &= 0. \end{aligned} \tag{3.32}$$

In the first equation, the indicator function $I_{\alpha=0}$ is 1 if $\alpha = 0$ and 0 otherwise. System (3.32) is an infinite system, and we truncate it for numerical purposes. In [32] it is shown that this system must be truncated carefully in order to preserve the strong elliptic condition. Instead of simply truncating the index set for k and using the inherited truncation for p, v, and S, we truncate the index for the expansion of p first.

We have

$$p(x) = \sum_{\alpha \in \mathcal{J}_{M,N}} p_\alpha(x) T_\alpha(\theta),$$

where $\mathcal{J}_{M,N}$ is the truncated multi-index set given by

$$\mathcal{J}_{M,N} = \left\{ \alpha = (\alpha_1, \ldots \alpha_N) \mid \alpha_i \in \{0, 1, 2, \ldots\},\ |\alpha| = \sum_{i=1}^{N} \alpha_i \leqslant M \right\}.$$

Note, here N corresponds to the dimension of the KLE, and M represents the order of the polynomial chaos approximation. Clearly, v and S can be expressed similarly. The difference now is that in the formula (3.31) the summation includes all eligible terms even if the index $\alpha + \gamma - 2p \notin \mathcal{J}_{M,N}$. Due to the explicit formula for k_α in (3.28), is it quite easy to determine these high order terms.

It is necessary that the truncated polynomial chaos system (3.32) preserves the properties of the original elliptic equation in (3.22). Since the original equation is symmetric and positive definite, we expect the system to be as well.

Lemma 3.3.3. *The elliptic system in* (3.32) *is symmetric and positive definite.*

Proof. For symmetry, we wish to show:

$$A_{\alpha,\beta}(x) = A_{\beta,\alpha}(x).$$

This is a direct result from the construction of $A_{\alpha,\beta}$. Recall $A_{\alpha,\beta}(x) E\left(k T_\alpha T_\beta\right).$

As for positive definiteness, we must show for any u_α, $\alpha \in \mathcal{J}_{M,N}$ that

$$0 < a|u|^2 \leq \sum_{\alpha,\beta \in \mathcal{J}_{M,N}} u_\alpha^T A_{\alpha,\beta}(x) u_\beta,\ x \in \Omega,$$

where $|u|^2 = \sum_{\alpha \in \mathcal{J}_{M,N}} |u_\alpha|^2$. From the definition of $A_{\alpha,\beta}$ we have

$$\begin{aligned}
\sum_{\alpha,\beta \in \mathcal{J}_{M,N}} u_\alpha^T A_{\alpha,\beta}(x) u_\beta &= \sum_{\alpha,\beta \in \mathcal{J}_{M,N}} u_\alpha^T E\left[k T_\alpha T_\beta\right] u_\beta \\
&= E\left[\left(\sum_{\alpha \in \mathcal{J}_{M,N}} u_\alpha T_\alpha\right)^T k \left(\sum_{\beta \in \mathcal{J}_{M,N}} u_\beta T_\beta\right)\right] \\
&\geq a\, E\left[\left(\sum_{\alpha \in \mathcal{J}_{M,N}} u_\alpha T_\alpha\right)^T \left(\sum_{\beta \in \mathcal{J}_{M,N}} u_\beta T_\beta\right)\right] \\
&= a|u|^2,
\end{aligned}$$

where the inequality comes from (3.25). □

Even with a truncated index, the number of terms in the truncated polynomial chaos expansion increase exponentially with respect to both N and M. A typical truncation may have $N = 6$ and $M = 4$. Then the finite polynomial expansion, $\mathcal{J}_{M,N}$, has 210 coefficients. However, the simple truncation is not optimal. The polynomial expansion decays both in Gaussian variable θ_i and Wick polynomial order. Note that for Wick polynomial T_α, $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_K)$, the component α_i denotes the order of the Hermite polynomial for random variable θ_i. Instead of using Hermite polynomials of the same order for all θ_i, $i \leq M$, it is better to use lower order polynomials for θ_i with higher subscripts. In addition to the truncation $\sum_{i=1}^N \alpha_i \leq M$, we introduce some extra constraints such as $\alpha_i \leq M - i$. This idea is similar to the sparse tensor product in high dimensional finite element method and also similar to our sparse collocation methods. For $N = 6$ and $M = 4$, we can use the maximum fourth order Hermite polynomials for θ_1, θ_2, but only third order Hermite polynomials for θ_3, second order Hermite polynomials for θ_4 and first order Hermite polynomials for θ_5, θ_6. We can also decouple the random variables θ_5, θ_6 from the rest of our random variables $\theta_1, \theta_2, \theta_3, \theta_4$. With the above compressions, the number of coefficients will be reduced dramatically, from 210 to 66 in the sparse truncation.

We consider a polynomial chaos approximation to the single-phase flow problem. We use a 32×32 grid in $\Omega = [0,1]^2$. We assume $p = 1$ and $S = 1$ on $x = 0$, $p = 0$ on $x = 1$ and no flow boundary conditions on the lateral boundaries $y = 0$ and $y = 1$. We choose $N = 5$ and $M = 3$ which, after truncation, results in 37 terms in $\mathcal{J}_{M,N}$. We solve the system (3.32) using a finite volume method. We then choose a random $\theta \in \mathbb{R}^5$ and reconstruct the pressure, velocity, and saturation according to (3.30). We calculate the permeability for θ and solve the single-phase problem (3.22) directly using a finite volume method. In Figure 3.1 we plot the saturations at a fixed time using both methods. The left figure represents the saturation from the direct solution to (3.22), while the right represents the saturation reconstructed from the polynomial chaos expansion. While the saturation front is duplicated quite well by the polynomial chaos approximation, we notice some slight oscillations along the flow front. These approximations are satisfactory in practice.

3.3.2. Sparse Grid Collocation

We again consider the single-phase system (3.22). In contrast to the previous polynomial chaos method, sparse grid collocation techniques do not require a solution to a system of coefficients in order to approximate our desired quantities. Indeed, for the single-phase system, two-phase system, or Richards' equation, the process in approximation via sparse grid collocation is identical.

For (3.22), let us suppose we wish to approximate p, given some permeability k. Our permeability is found from the KLE, and hence k and p are functions of the multidimensional stochastic variable θ. To obtain the approximation to p at any fixed θ, we interpolate using the values of p for each $\hat{\theta}$ in the set $H\left(q,N\right)$, given by (3.20). Thus, for each $\hat{\theta} \in H\left(q,N\right)$ we must calculate $k(x,\hat{\theta})$ and solve (3.22) for $p(x,\hat{\theta})$.

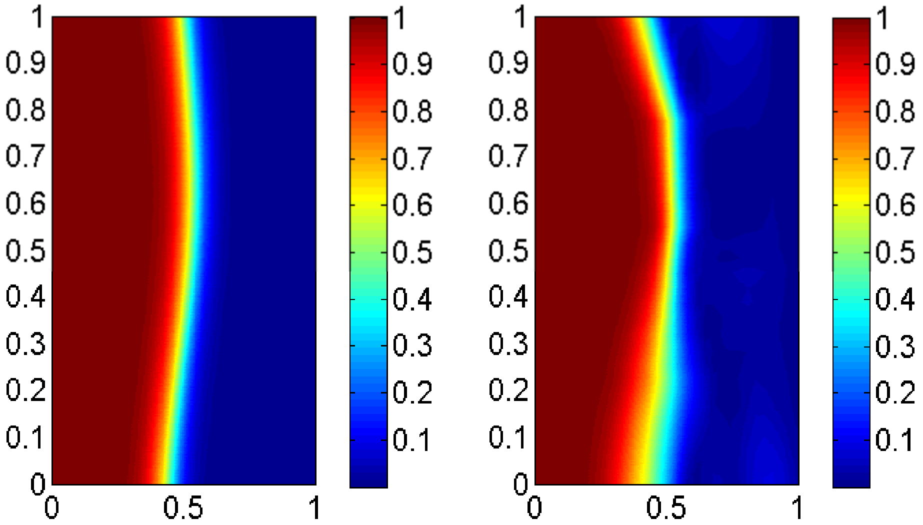


Fig. 3.1. Left: Saturation profile at a fixed time for the direct solve to (3.22). Right: Reconstructed saturation profile at the same fixed time using the polynomial chaos expansion.

Once we have the values of $p(x, \hat{\theta})$ then for any other θ, we calculate $A\left(q, N\right)(p)(\theta)$ according to (3.19).

If we wish to approximate v and S as well, we simply need to save their values at each $\hat{\theta} \in H\left(q, N\right)$ while solving (3.22) and interpolate similarly. As mentioned previously, we choose $\hat{\theta}$ based on the extrema of Chebyshev polynomials. By transformation, we can choose $\hat{\theta}$ for any hypercube. By the nature of interpolation, we will restrict ourselves to approximations of θ that are within this hypercube.

We consider sparse collocation approximations to the single-phase flow problem. We use a 61×61 grid in $\Omega = [0, 1]^2$. We assume $p = 1$ and $S = 1$ on $x = 0$, $p = 0$ on $x = 1$ and no flow boundary conditions on the lateral boundaries $y = 0$ and $y = 1$. We take $N = 9$ and choose 1^{st} order Smolyak interpolation, $q = N + 1$. For $N = 9$, 1^{st} order Smolyak interpolation requires us to use only 19 nodes. Thus only 19 single-phase solutions are needed to generate the necessary data for sparse grid

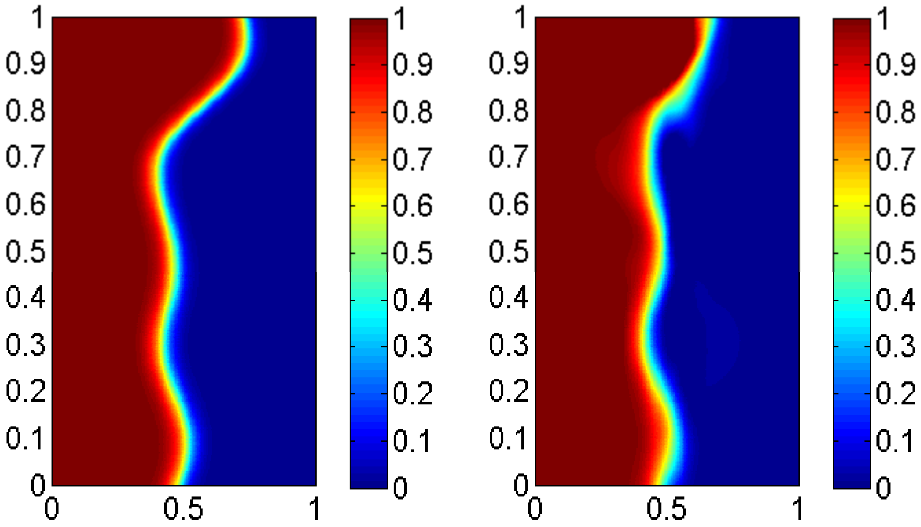


Fig. 3.2. Left: Saturation profile at a fixed time for the direct solve to (3.22). Right: Interpolated saturation profile at the same fixed time using 19 values.

collocation. We generate nodes for Smolyak interpolation based on the hypercube $[-2.5, 2.5]^N$. We choose a specific θ and generate the corresponding permeability. We solve (3.22) directly and compare the results with the interpolated values. In Figure 3.2 we plot the corresponding saturations for a fixed time snapshot. The left plot represents the saturation from the direct solution to (3.22), while the right is interpolated saturation. The two saturations are nearly identical, with only a slight fluctuation along the saturation front.

3.4. Application to Richards' Equation

3.4.1. Polynomial Chaos

We consider Richards' equation in the form given by (2.7), rewritten as

$$\frac{\partial f}{\partial t}(u) - \nabla \cdot \left[K_s K_r \nabla (u + x_3)\right] = 0. \tag{3.33}$$

Here we have written the conductivity as $K = K_s K_r$ where the term K_r depends on which constitutive relation we choose. Note we have written the volumetric water content as $f(u)$ instead of $\theta(u)$ to avoid confusion with the random variables θ in the KLE.

We expand the saturated conductivity as a Fourier-Hermite polynomial of θ,

$$K_s(\theta) = \sum_{\alpha \in \mathcal{J}} K_{s\alpha} T_\alpha(\theta).$$

Since u depends on K_s we also write

$$u(\theta) = \sum_{\alpha \in \mathcal{J}} u_\alpha T_\alpha(\theta).$$

We must also compute the expansions of $f\,(u)$ and K_r. For the exponential model, we have $f\,(u) = f_s e^{bu}$ and $K_r = e^{au}$ (where again we write f_s for the saturated moisture content instead of θ_s). To approximate e^{bu} we must use a Taylor series expansion about $u = 0$. Using a fifth order expansion, we have the estimate

$$e^{bu} \approx 1 + b\left(u + \frac{b}{2}\left(u + \frac{b}{3}\left(u + \frac{b}{4}\left(u + \frac{b}{5}\right)\right)\right)\right).$$

In order to expand e^{bu} as a Fourier-Hermite series, we must recursively apply the formula in Lemma 3.1.1. Similarly we can expand e^{au}. Putting these results together, we write

$$K\,(x,u) = K_s(x)K_r(u) \triangleq \sum_{\alpha \in \mathcal{J}} g_\alpha\,(x,t)\,T_\alpha\,(\theta)\,.$$

With the above definitions, we may write the governing equations for the polynomial chaos coefficients for the exponential model of Richards' equation as

$$f_s \frac{df_\alpha}{dt} - \sum_{p \in \mathcal{J}} \sum_{0 \leq \beta \leq \alpha} \nabla \cdot \left[g_{\alpha-\beta+p} \nabla\left(u_{\beta+p} + x_3\right)\right] = 0.$$

We generally use an implicit scheme in time to solve the Richards' equation. A combination of an implicit formulation along with the recursive applications of Lemma 3.1.1 make this particular formulation expensive. The derivation of the polynomial chaos system is similar for the van Genuchten and Haverkamp models. Both models lead to a nonlinear implicit system which contains recursive applications of the Fourier-Hermite polynomial multiplication formulas.

We now consider a simpler model using the constitutive relation given by Irmay [28]. If we assume $m = 1$ then we will greatly increase the ease at which we can solve the Richards' equation. If we assume this K_r is the same as the exponential constitutive relation, then we have

$$K_r = e^{au} = \left(\frac{f - f_r}{f_s - f_r}\right),$$

which leads to the equation

$$f = f_r + (f_s - f_r)\, e^{au}. \tag{3.34}$$

Note we have again substituted f for θ. As shown in Tracy [42] this model is at least physically reasonable when compared with the van Genuchten model. We also will consider boundary conditions on $\Gamma = \Gamma_D \cup \Gamma_N$ given by

$$u(\mathbf{x}) = \psi(\mathbf{x}),\ x \in \Gamma_D,$$
$$[K_s e^{au} \nabla (u + x_3)] \cdot n = Q(\mathbf{x}),\ x \in \Gamma_N.$$

Here we have $\psi(\mathbf{x})$ a randomly prescribed pressure head and $Q(\mathbf{x})$ is a randomly prescribed flux.

We consider a Kirchhoff transformation given by

$$\Phi(\mathbf{x}) = \int_{-\infty}^{u(\mathbf{x})} e^{at}\, dt = \frac{1}{a} e^{au}. \tag{3.35}$$

Thus we have

$$\nabla\Phi\left(\mathbf{x}\right)=\nabla\frac{1}{a}e^{au}=e^{au}\nabla u,$$

which gives us

$$\nabla\cdot\left[K_s e^{au}\nabla\left(u+x_3\right)\right]=\nabla\cdot\left(K_s\nabla\Phi\right)+a\frac{\partial}{\partial x_3}\left(K_s\Phi\right).$$

Inserting the formula for f from the Irmay model, we find

$$\frac{\partial f}{\partial t}\left(u\right)-\nabla\cdot\left[K_s e^{au}\nabla\left(u+x_3\right)\right]=\frac{\partial}{\partial t}\left(f_r+\left(f_s-f_r\right)a\Phi\right)-\nabla\cdot\left(K_s\nabla\Phi\right)-a\frac{\partial}{\partial x_3}\left(K_s\Phi\right)=0,$$

which leads to the following PDE:

$$\left(f_s-f_r\right)a\frac{\partial\Phi}{\partial t}-\nabla\cdot\left(K_s\nabla\Phi\right)-a\frac{\partial}{\partial x_3}\left(K_s\Phi\right)=0. \tag{3.36}$$

After transformation, our boundary conditions are given by

$$\Phi\left(\mathbf{x}\right)=\tfrac{1}{a}e^{a\psi(\mathbf{x})},\ x\in\Gamma_D,$$
$$\left[\left(K_s\nabla\Phi\right)+a\tfrac{\partial}{\partial x_3}\left(K_s\Phi\right)\right]\cdot n=Q\left(\mathbf{x}\right),\ x\in\Gamma_N.$$

We again assume that the saturated conductivity can be expressed as

$$K_s(x)=\sum_{\alpha\in\mathcal{J}}K_{s\alpha}(x)T_\alpha\left(\theta\right).$$

Since Φ depends on K_s, we can also express Φ as a Fourier-Hermite polynomial. Thus we write

$$\Phi\left(x,t\right)=\sum_{\alpha\in\mathcal{J}}\Phi_\alpha\left(x,t\right)T_\alpha\left(\theta\right).$$

Using the multiplicative rule for Fourier-Hermite expansions we may write our system as

$$\left(f_s-f_r\right)a\frac{\partial\Phi_\alpha}{\partial t}-\sum_{\beta\in\mathcal{J}}\left[\nabla\cdot\left(A_{\alpha,\beta}\nabla\Phi_\beta\right)+a\frac{\partial}{\partial x_3}\left(A_{\alpha,\beta}\Phi_\beta\right)\right]=0,\ \alpha\in\mathcal{J},$$

where

$$A_{\alpha,\beta} = \sum_{\beta \in \mathcal{J}} \sum_{p \leq \alpha \wedge \beta} B\left(\alpha, \beta, p\right) K_{s\,\alpha+\beta-2p} \,.$$

Since our original boundary conditions are given by

$$\begin{gathered} \Phi\left(\mathbf{x}\right) = \tfrac{1}{a} e^{a\psi(\mathbf{x})},\ x \in \Gamma_D, \\ \left[\left(K_s \nabla \Phi\right) + a\tfrac{\partial}{\partial x_3}\left(K_s \Phi\right)\right] \cdot n = Q\left(\mathbf{x}\right),\ x \in \Gamma_N, \end{gathered}$$

then for the polynomial chaos system we have

$$\begin{aligned} \Phi_\alpha\left(\mathbf{x}\right) &= \frac{1}{a} e^{a\psi(\mathbf{x})} \delta_{\alpha 0},\ x \in \Gamma_D, \\ \sum_{\beta \in \mathcal{J}} \left[\nabla \cdot \left(A_{\alpha,\beta} \nabla \Phi_\beta\right) + a\frac{\partial}{\partial x_3}\left(A_{\alpha,\beta} \Phi_\beta\right)\right] \cdot n &= 0,\ x \in \Gamma_N. \end{aligned} \tag{3.37}$$

We consider a polynomial chaos approximation to the above formulation of Richards' equation using the Irmay model. We use a 32×32 grid in $\Omega = [0, 1]^2$. We assume a pressure head of $u = 1$ on $y = 0$, $u = 0$ on $x = 1$, and no flow boundary conditions on the side boundaries $x = 0$ and $x = 1$. We choose $N = 5$ and $K = 2$ which, after truncation, results in 17 terms in $\mathcal{J}_{M,N}$. We solve (3.37) using an implicit finite volume method. We choose a random $\theta \in \mathbb{R}^5$, calculate K_s and solve (3.36) using an implicit finite volume method as well. In Figure 3.3 we compare the pressure head using both methods. The reconstructed pressure matches the original pressure profile almost exactly. These approximations will be satisfactory for the uncertainty quantification problems discussed later.

3.4.2. Sparse Grid Collocation

We again consider Richards' equation, but now we focus on the exponential model. In contrast to the previous polynomial chaos method, sparse grid collocation techniques can be applied regardless of which constitutive relation or what form of Richards' equation we desire to use.

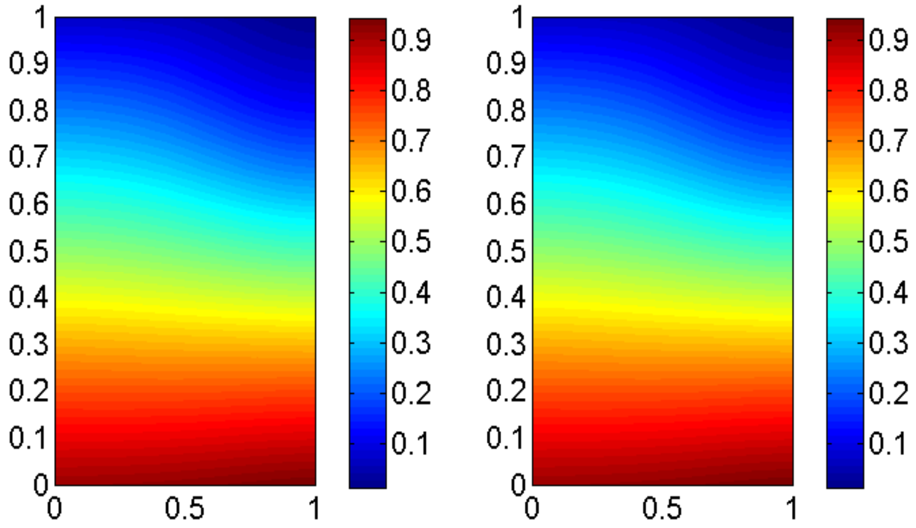


Fig. 3.3. Left: Pressure profile for the direct solve to (3.36). Right: Reconstructed pressure profile using the polynomial chaos expansion.

As in the previous case of single-phase flow, we wish to approximate u given some saturated conductivity K_s. Since K_s is found using the KLE, k and u are functions of the multidimensional stochastic variable θ. We interpolate using the values of u for each $\hat{\theta} \in H\left(q, N\right)$ where H is given by (3.20). We calculate $K_s(x, \hat{\theta})$ and solve Richards' equation for each $u(x, \hat{\theta})$. We choose $\hat{\theta}$ based on the extrema of Chebyshev polynomials, transformed to a hypercube centered at the origin.

We consider a 49×49 grid in $\Omega = [0, 1]^2$. We assume a pressure head of $u = -20$ on $y = 0$, $u = 0$ on $x = 1$, and no flow boundary conditions on the side boundaries $x = 0$ and $x = 1$. We choose first order Smolyak interpolation, thus for $N = 9$ we use only 19 nodes. We choose a specific θ and generate the corresponding K_s. We solve Richards' equation directly using an implicit finite volume solver and compare the results with the interpolated values. In Figure 3.4 we compare the pressure head from each method. The left figure represents the pressure from the direct solution, while the right represents the interpolated pressure. There is very little difference

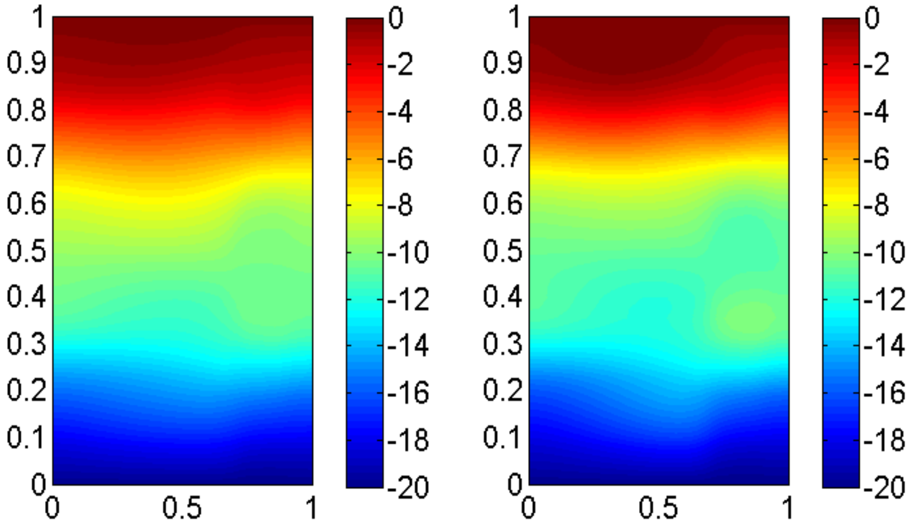


Fig. 3.4. Left: Pressure profile for the direct solve to Richards' equation. Right: Interpolated pressure profile using 19 values.

between the two. This close approximation allows us to use the collocation methods in the uncertainty quantification problems discussed in the later chapters.

CHAPTER IV

MULTISCALE METHODS FOR STOCHASTIC POROUS MEDIA FLOWS

In this chapter, our aim is to combine multiscale methods with the stochastic equations for porous media flows. We use the methods presented in the previous chapter to approximate the stochastic portion of the equations, while using multiscale methods to solve in the spatial dimensions.

4.1. Upscaling Methods

We first consider upscaling methods, as described in Section 2.3. We derive the upscaled equations for the polynomial chaos method for single-phase flow. Next, we discuss applications of upscaling using collocation methods.

4.1.1. Polynomial Chaos Equations

Recall the system given by (3.32). In most applications the permeability k has multi-scale structures, and thus the stiffness coefficients $A_{\alpha,\beta}(x)$ do as well. We wish to capture the effects of the small scales by deriving an upscaled system. We now derive the upscaled equations in the framework of homogenizaton as it is usually done for the scalar flow equations. The obtained method is applicable to more general cases [20].

We write the elliptic system in (3.32) as

$$\frac{\partial}{\partial x_i}\left(A^{ij}_{\alpha,\beta}\left(x,\frac{x}{\varepsilon}\right)\frac{\partial}{\partial x_j}P_\beta\right)=0, \tag{4.1}$$

where we assume that $A_{\alpha,\beta}\left(x,\frac{x}{\varepsilon}\right)$ is periodic. We further assume k has a nonzero lower bound, and thus $A_{\alpha,\beta}(x)$ has a nonzero bound as well [32]. For ease of notation let us assume summation over the indices throughout. We denote $y=\frac{x}{\varepsilon}$ and assume

$x \in \Omega$ and $y \in Y$. We perform a multiscale expansion of P_β and find

$$P_\beta = P_\beta^0\left(x, \frac{x}{\varepsilon}\right) + \varepsilon P_\beta^1\left(x, \frac{x}{\varepsilon}\right) + \varepsilon^2 P_\beta^2\left(x, \frac{x}{\varepsilon}\right) + \dots . \tag{4.2}$$

Note for any $f\left(x, \frac{x}{\varepsilon}\right)$ we have

$$\frac{\partial}{\partial x} f\left(x, \frac{x}{\varepsilon}\right) = \frac{\partial}{\partial x} f\left(x, \frac{x}{\varepsilon}\right) + \frac{1}{\varepsilon}\frac{\partial}{\partial y} f\left(x, \frac{x}{\varepsilon}\right).$$

Using the chain rule in (4.1) we get

$$\left[\frac{\partial}{\partial x_i} + \frac{1}{\varepsilon}\frac{\partial}{\partial y_i}\right]\left(A_{\alpha,\beta}^{ij}(x,y)\left[\frac{\partial}{\partial x_j} + \frac{1}{\varepsilon}\frac{\partial}{\partial y_j}\right] P_\beta(x,y)\right) = 0. \tag{4.3}$$

If we substitute the expansion for P_β back into our pressure equation we get

$$\begin{gathered}
\frac{\partial}{\partial x_i}\left(A_{\alpha,\beta}^{ij}(x,y)\frac{\partial}{\partial x_j}P_\beta^0(x,y)\right) + \frac{1}{\varepsilon}\frac{\partial}{\partial y_i}\left(A_{\alpha,\beta}^{ij}(x,y)\frac{\partial}{\partial x_j}P_\beta^0(x,y)\right) + \\
\frac{1}{\varepsilon}\frac{\partial}{\partial x_i}\left(A_{\alpha,\beta}^{ij}(x,y)\frac{\partial}{\partial y_j}P_\beta^0(x,y)\right) + \frac{1}{\varepsilon^2}\frac{\partial}{\partial y_i}\left(A_{\alpha,\beta}^{ij}(x,y)\frac{\partial}{\partial y_j}P_\beta^0(x,y)\right) + \\
\varepsilon\frac{\partial}{\partial x_i}\left(A_{\alpha,\beta}^{ij}(x,y)\frac{\partial}{\partial x_j}P_\beta^1(x,y)\right) + \frac{\partial}{\partial y_i}\left(A_{\alpha,\beta}^{ij}(x,y)\frac{\partial}{\partial x_j}P_\beta^1(x,y)\right) + \\
\frac{\partial}{\partial x_i}\left(A_{\alpha,\beta}^{ij}(x,y)\frac{\partial}{\partial y_j}P_\beta^1(x,y)\right) + \frac{1}{\varepsilon}\frac{\partial}{\partial y_i}\left(A_{\alpha,\beta}^{ij}(x,y)\frac{\partial}{\partial y_j}P_\beta^1(x,y)\right) + \\
\varepsilon^2\frac{\partial}{\partial x_i}\left(A_{\alpha,\beta}^{ij}(x,y)\frac{\partial}{\partial x_j}P_\beta^2(x,y)\right) + \varepsilon\frac{\partial}{\partial y_i}\left(A_{\alpha,\beta}^{ij}(x,y)\frac{\partial}{\partial x_j}P_\beta^2(x,y)\right) + \\
\varepsilon\frac{\partial}{\partial x_i}\left(A_{\alpha,\beta}^{ij}(x,y)\frac{\partial}{\partial y_j}P_\beta^2(x,y)\right) + \frac{\partial}{\partial y_i}\left(A_{\alpha,\beta}^{ij}(x,y)\frac{\partial}{\partial y_j}P_\beta^2(x,y)\right) + \dots = 0.
\end{gathered}$$

Collecting the ε^{-2} terms we get

$$\frac{\partial}{\partial y_i}\left(A_{\alpha,\beta}^{ij}(x,y)\frac{\partial}{\partial y_j}P_\beta^0(x,y)\right) = 0.$$

We multiply both sides by $P_\alpha^0(x,y)$ and integrate by parts. We use the fact that Y is periodic and that $A_{\alpha,\beta}$ has a lower bound. Hence

$$0 = \int_Y A_{\alpha,\beta}^{ij}(x,y)\frac{\partial}{\partial y_j}P_\beta^0(x,y)\frac{\partial}{\partial y_i}P_\alpha^0(x,y)\,dy \geq \int_Y \sum_\alpha \left|\frac{\partial}{\partial y_i}P_\alpha^0(x,y)\right|^2 dy,$$

which implies that

$$P^0_\beta(x,y) = P^0_\beta(x)\,. \tag{4.4}$$

When we collect the ε^{-1} terms, we find

$$\frac{\partial}{\partial y_i}\left(A^{ij}_{\alpha,\beta}(x,y)\frac{\partial}{\partial x_j}P^0_\beta(x,y)\right)+\frac{\partial}{\partial x_i}\left(A^{ij}_{\alpha,\beta}(x,y)\frac{\partial}{\partial y_j}P^0_\beta(x,y)\right)$$
$$+\frac{\partial}{\partial y_i}\left(A^{ij}_{\alpha,\beta}(x,y)\frac{\partial}{\partial y_j}P^1_\beta(x,y)\right)=0.$$

Since $P^0_\beta(x,y) = P^0_\beta(x)$, the second term is zero and we may separate the first term. We get

$$\frac{\partial}{\partial y_i}A^{ij}_{\alpha,\beta}(x,y)\frac{\partial}{\partial x_j}P^0_\beta(x)+\frac{\partial}{\partial y_i}\left(A^{ij}_{\alpha,\beta}(x,y)\frac{\partial}{\partial y_j}P^1_\beta(x,y)\right)=0.$$

We note that this can be written as $LP^1_\beta(x,y) = f(x,y)\,g(x)$ where L is a linear operator. Thus, we may write $P^1_\beta(x,y) = N(x,y)\,g(x)$ where $LN(x,y) = f(x,y)\,.$ Thus

$$P^1_\beta(x,y) = N^j_{\beta,\gamma}(x,y)\frac{\partial}{\partial x_j}P^0_\gamma(x)$$

and $N^j_{\beta,\gamma}$ satisfies

$$\frac{\partial}{\partial y_i}\left(A^{ij}_{\alpha,\beta}(x,y)\frac{\partial}{\partial y_j}N^k_{\beta,\gamma}(x,y)\right)=-\frac{\partial}{\partial y_i}A^{ik}_{\alpha,\gamma}(x,y)\,. \tag{4.5}$$

Collecting ε^0 terms, we have

$$\frac{\partial}{\partial x_i}\left(A^{ij}_{\alpha,\beta}(x,y)\frac{\partial}{\partial x_j}P^0_\beta(x)\right)+\frac{\partial}{\partial x_i}\left(A^{ij}_{\alpha,\beta}(x,y)\frac{\partial}{\partial y_j}P^1_\beta(x,y)\right)+ \tag{4.6}$$
$$\frac{\partial}{\partial y_i}\left(A^{ij}_{\alpha,\beta}(x,y)\frac{\partial}{\partial x_j}P^1_\beta(x,y)\right)+\frac{\partial}{\partial y_i}\left(A^{ij}_{\alpha,\beta}(x,y)\frac{\partial}{\partial y_j}P^2_\beta(x,y)\right)=0.$$

Let $\langle\cdot\rangle$ be an averaging over Y, and note that

$$\langle\nabla_y\cdot f\rangle=\int_{\partial Y} f(x,y)\cdot n\,dl=0$$

if $f(x,y)$ is periodic. If we average (4.6) over a period, then we find that the last two terms are zero. Then (4.6) reduces to

$$\frac{\partial}{\partial x_i}\left(\left\langle A^{ij}_{\alpha,\beta}(x,y)\right\rangle \frac{\partial}{\partial x_j} P^0_\beta(x,y)\right) + \frac{\partial}{\partial x_i}\left\langle A^{ij}_{\alpha,\beta}(x,y)\frac{\partial}{\partial y_j}P^1_\beta(x,y)\right\rangle =$$
$$\frac{\partial}{\partial x_i}\left(\left\langle A^{ij}_{\alpha,\beta}(x,y)\right\rangle \frac{\partial}{\partial x_j} P^0_\beta(x)\right) + \frac{\partial}{\partial x_i}\left\langle A^{ij}_{\alpha,\beta}(x,y)\frac{\partial}{\partial y_j}N^k_{\beta,\gamma}(x,y)\frac{\partial}{\partial x_k}P^0_\gamma(x)\right\rangle = 0.$$

If we let

$$\left(A^{ij}_{\alpha,\beta}\right)^* = \left\langle A^{ij}_{\alpha,\beta}(x,y)\right\rangle + \left\langle A^{ij}_{\alpha,\beta}(x,y)\frac{\partial}{\partial y_j}N^k_{\beta,\gamma}(x,y)\right\rangle \tag{4.7}$$

then the previous equation for the ε^0 terms reduces to

$$\frac{\partial}{\partial x_i}\left(A^{ij}_{\alpha,\beta}\right)^* \frac{\partial}{\partial x_j}P^0_\beta(x) = 0. \tag{4.8}$$

Then equation (4.8) is the upscaled equation for the leading order terms P^0_β, with $\left(A^{ij}_{\alpha,\beta}\right)^*$ as the stiffness coefficients.

In order to compute $\left(A^{ij}_{\alpha,\beta}\right)^*$, we must first find $N^k_{\beta,\gamma}(x,y)$. In order to apply the explicit upscaled formulations, we partition the domain Ω into coarse blocks $\Omega = \bigcup_{k=1}^{K}\Omega_k$. Then we solve the equation on each coarse block Ω_k. Referring back to (4.5) we write

$$\frac{\partial}{\partial y_i}\left(A^{ij}_{\alpha,\beta}(x,y)\frac{\partial}{\partial y_j}N^k_{\beta,\gamma}(x,y) + A^{ik}_{\alpha,\gamma}(x,y)\right) = 0,\ x\in\Omega_k. \tag{4.9}$$

If we write $A^{ik}_{\alpha,\beta}(x,y) = A^{ij}_{\alpha,\gamma}(x,y)\,\delta_{\beta,\gamma}\delta_{jk}$ then we can simplify the above to

$$\frac{\partial}{\partial y_i}A^{ij}_{\alpha,\gamma}(x,y)\left(\frac{\partial}{\partial y_j}N^k_{\beta,\gamma}(x,y)+\delta_{\beta,\gamma}\delta_{jk}\right) = \frac{\partial}{\partial y_i}A^{ij}_{\alpha,\gamma}(x,y)\frac{\partial}{\partial y_j}\left(N^k_{\beta,\gamma}(x,y)+\delta_{\beta,\gamma}y_k\right) = 0.$$

Thus, we need to solve the local problems

$$\frac{\partial}{\partial x_i}A^{ij}_{\alpha,\gamma}\frac{\partial}{\partial x_j}\varphi_{\beta,\gamma} = 0,\ x\in\Omega_k. \tag{4.10}$$

Once we have $\varphi_{\beta,\gamma}$ then we compute the stiffness coefficient on each block by the local average

$$\left(A^{ij}_{\alpha,\beta}\right)^*_{\Omega_k} = \left\langle A^{ij}_{\alpha,\gamma}\left(x,y\right)\frac{\partial}{\partial y_j}\varphi_{\beta,\gamma}\right\rangle_{\Omega_k}. \tag{4.11}$$

One can also consider global upscaling as it is done for scalar equations. Global upscaling is particularly important if there is no apparent scale separation. Here, we refer to [5, 26] and references therein for some single-phase flow results using global upscaling. The main idea of the global upscaling is to compute the effective permeability from the global solution instead of local solutions. As we mentioned earlier, these approaches are similar to oversampling method when the oversampling domain is taken to be the entire domain. One can also do global upscaling for the polynomial chaos system, similar to scalar elliptic equations. We note that this type of work is not done previously. To do the global upscaling, we solve (4.9) or (4.10) over the entire domain (i.e, Ω_k is replaced by Ω). Then, (4.10) is used to compute the upscaled coefficients in each coarse grid block.

Next, we present some representative numerical examples using polynomial chaos expansions and upscaling. We consider the same polynomial chaos approximation to the single-phase flow problem as in Section 3.3.1. We start with a 32 × 32 grid and use global upscaling to form the solution on an 8 × 8 grid for $\Omega = [0,1]^2$. We repeat the same process from Section 3.3.1 and solve the upscaled version of (3.32). We fix a θ and solve the single-phase problem (3.22) using (4.11). In Figure 4.1 we compare the saturations for a fixed-time snapshot using each solution technique. On the left, we have the saturation from the solution of (3.22) on the full 32 × 32 grid. The middle plot contains the saturation from the solution to (3.22) using global upscaling. The right figure contains the saturation from the globally upscaled polynomial chaos approximation using (4.11). The saturation front is duplicated quite well by the polynomial chaos approximation, which is satisfactory for our uncertainty

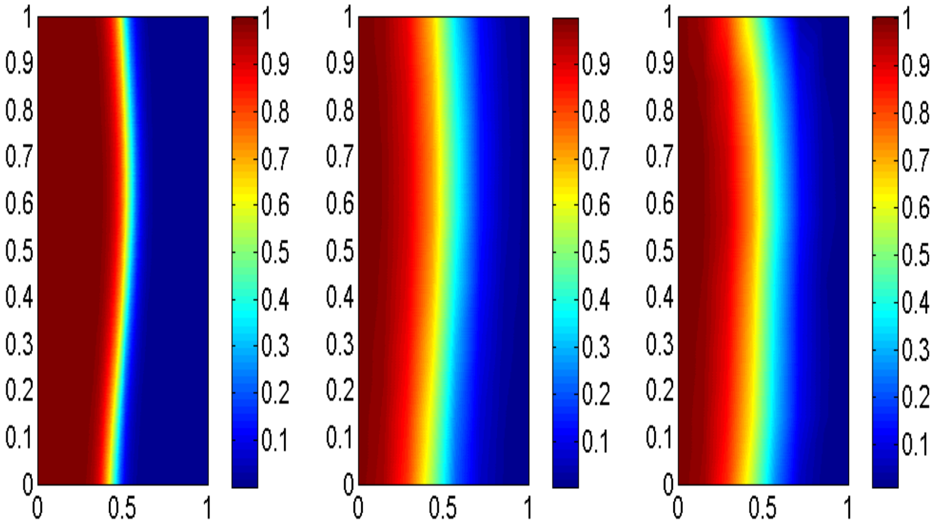


Fig. 4.1. Left: Saturation profile at a fixed time for the direct solve to (3.22) for the full 32 × 32 grid. Middle: Saturation profile at the same fixed time using global upscaling to the 8 × 8 grid with a direct solve to (3.22). Right: Reconstructed saturation profile at the same fixed time using the globally upscaled polynomial chaos expansion on an 8 × 8 grid.

quantification problems.

4.1.2. Collocation Methods

In this section, our aim is to show that collocation methods can be used in conjunction with single-phase upscaling. Using methods mentioned in Section 2.3, we formulate the single-phase upscaling problem for flow in porous media. We calculate a permeability $k(x, \theta)$ using the KLE, then calculate the upscaled permeability $k^*(x, \theta)$ using solutions to local (or global) problems. If we wish to solve this problem for numerous permeabilities, then the upscaled permeability k^* must be recomputed for each new permeability. We instead wish to consider a collocation method, where, using the Smolyak algorithm, we compute a set of upscaled permeabilities at some fixed reference nodes in stochastic space. Thus we compute $k^*(x, \theta_i)$ for each $\theta_i \in H(q, N)$. Then each time we choose a new permeability k, the upscaled permeability k^* is approximated by interpolation based on the reference upscaled permeabilities by

$$\tilde{k}^*(x, \theta) = \sum_i k^*(x, \theta_i)\, \beta_i(\theta).$$

We note the weights are computed apriori, and they are Lagrange polynomials in our particular collocation techniques.

We wish to compare only the interpolated upscaled permeability $\tilde{k}^*$ and the calculated upscaled permeability k^*. In the previous chapter, we considered interpolation of both pressure and saturation for the normal covariance (2.16). For the same reasons as mentioned in the previous chapter, we have a good correspondence between the interpolated coarse-scale permeability, $\tilde{k}^*$, and the original coarse-scale permeability, k^*. The two permeabilities are very close to each other because of the smoothness of the fields generated using normal covariance. We do not present these results here, and instead focus on using exponential covariance (2.17). Another reason the interpolation worked so well in the previous chapter is because the stochastic dimension for the KLE with normal covariance is relatively small due to the reduced truncation. For example, in Section 3.3.2 we interpolate in only 9 stochastic dimensions after using a reduced KLE. In this section, we generate a KLE using exponential covariance where we choose $L_1 = 0.5$, $L_2 = 0.05$ and $\sigma^2 = 2.0$ in (2.17). A truncated KLE with $N = 500$ terms captures over 95% of the energy for this case. For interpolation, we use only 1^{st} order Smolyak. Since $N = 500$ we require $1,001$ reference values for 1^{st} order Smolyak interpolation. If we were to consider 2^{nd} order Smolyak interpolation as well, then for just $N = 100$ we would have to generate $20,201$ reference values. To even compute the number of nodes necessary for $N = 500$ with 2^{nd}

order interpolation would require us to modify the traditional Smolyak algorithm due to computational memory issues.

We consider the single-phase elliptic problem where boundary conditions are given by $p = 1$ on $x = 0$, $p = 0$ on $x = 1$ and no flow boundary conditions on the lateral boundaries $y = 0$ and $y = 1$. We consider a 100×100 fine grid and coarse grids of size 1×1, 5×5 and 10×10. We restrict the stochastic domain to the hypercube $[-2.5, 2.5]^{500}$ for Smolyak interpolation. We solve the single-phase upscaling problem for each of the three coarse grids to generate the $1,001$ reference values of k^*. We then choose $1,000$ random values of $\theta \in [-2.5, 2.5]^{500}$ and compute both k^* and $\tilde{k}^*$. To test the importance of the higher order terms in the KLE to interpolation, we also compute an interpolated permeability based on only the first 100 terms in the KLE. Let us denote this as $\tilde{k}^*_{100}$

In Figure 4.2 we compare the empirical cumulative distribution function (CDF) for each of the three coarse grids. In each plot the CDF value on the y-axis represents the proportion of permeabilities that are less than or equal to the value on the axis. In each plot, the CDF from the two interpolated upscaled permeabilities matches the reference upscaled permeability almost exactly. This indicates that our interpolation is statistically accurate, and also indicates that the trailing 400 terms in the KLE do not have a significant effect on the upscaled permeabilities in this case. In Table 4.1 we show the mean, median, and standard deviation for each of the CDFs. We note both interpolation methods work very well.

4.2. Multiscale Methods

In this section, we present a multiscale approach for solving stochastic flow equations (elliptic equation) using sparse grid collocation (e.g., [46]). We restrict our

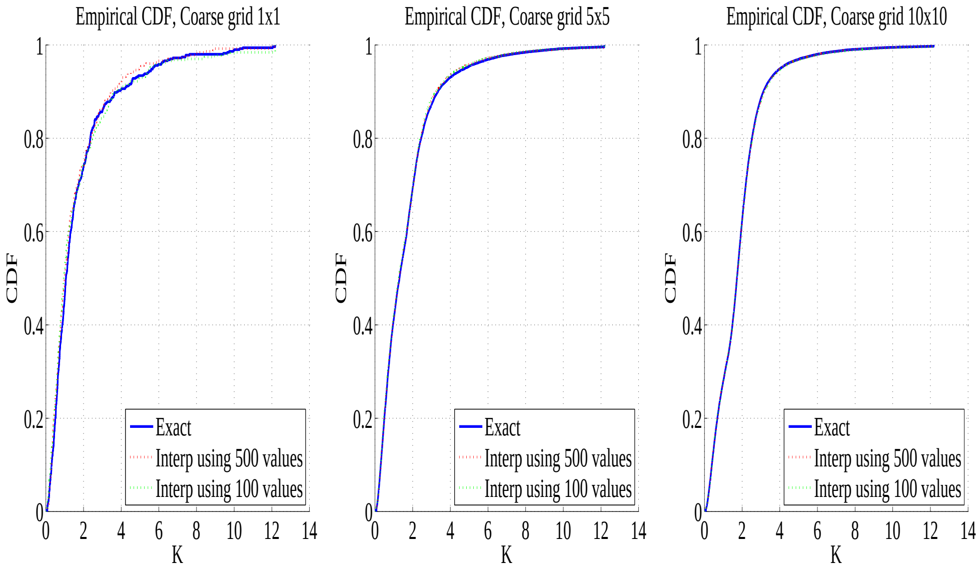


Fig. 4.2. Left: CDF for a coarse grid of size 1 × 1. Middle: CDF for a coarse grid of size 5 × 5. Right: CDF for a coarse grid of size 10 × 10.

Table 4.1. Mean, median, and standard deviation for the CDF using the 1 × 1, 5 × 5, and 10 × 10 coarse grids.

Perm	1 × 1 Coarse Grid			5 × 5 Coarse Grid			10 × 10 Coarse Grid		
	Mean	Med	Std Dev	Mean	Med	Std Dev	Mean	Med	Std Dev
k^*	1.706	1.071	1.912	1.721	1.303	1.707	1.854	1.729	1.444
$\tilde{k}^*$	1.604	0.997	1.808	1.699	1.283	1.688	1.863	1.734	1.451
$\tilde{k}^*_{100}$	1.712	0.954	2.107	1.678	1.264	1.671	1.874	1.739	1.491

discussion to two-phase immiscible flow presented in Section 2.1.1. We assume the solution is computed for some values of $\theta = (\Theta_1, ..., \Theta_N)$, denoted by θ_k, and then interpolated for $\theta \in R^N$. Throughout, we assume that $k(x, \theta)$ smoothly depends on θ, keeping in mind Karhunen-Loève type expansions. The results of this section do not depend on specific collocation methods. We only use the fact that the solution can be approximated using its values at certain locations

$$p(x, \theta) = \sum_i p(x, \theta_i)\beta_i(\theta),$$

where $\beta_i(\theta)$ are the corresponding weights.

Next, we present a multiscale approach. Typically, in the absence of scale separation, one needs to use multiscale models based on global fields [17]. We propose two related approaches. The first approach entails the computation of basis functions at some sparse points θ_k. Denote these basis functions by $\phi_i(x, \theta_k)$. Then, at each θ for which the solution needs to be computed, instead of solving for basis functions at θ, we interpolate it via pre-computed basis functions

$$\phi_i(x, \theta) = \sum_k \phi_i(x, \theta_k)\beta_k(\theta).$$

This approach is efficient, in particular, when interpolation weights can be easily computed. Moreover, this approach is similar to ensemble level upscaling methods where the upscaled permeabilities are computed for selected collocation points and then interpolated for any given θ [7]. Our second approach is different from the first one, and we believe it does not have an analog within traditional upscaling methods. In our second approach, we propose to use the family of basis functions for all θ_k without interpolating them to a particular realization θ. This is an interpolation free approach and has advantages when the number of collocation points are not large and the interpolation formula is not readily available.

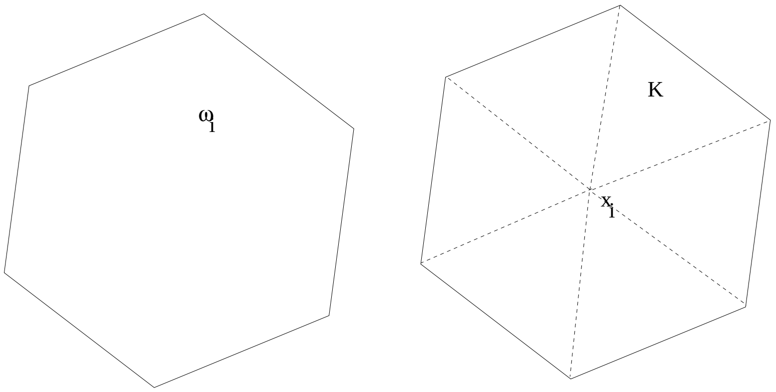


Fig. 4.3. Schematic description of patch

Next, we briefly present an analysis for the second approach. Analysis of the first approach is similar. Our goal is to find a finite number of global fields such that they can be used to approximate the ensemble, i.e., the probability distribution of the solution. For each realization θ_i, we assume that the solution can be represented using the global fields $u_j\left(x,\theta_k\right)$. These global fields, for example, may be global solutions to the single-phase elliptic equation with various boundary conditions or source terms (see [1] for more details). In multiscale methods, our goal is to find basis functions that can be used to approximate the solution for $p(x,\theta_k)$ for any smooth functions $\lambda(x)$ in

$$div(\lambda(x)k(x,\theta_k)\nabla p)=0.$$

Following to [1], our first step is to determine global fields such that the solution smoothly depends on these fields. Let ω_i be a patch (see Figure 4.3) and define $\phi_i^0(x)$ to be a piecewise linear basis function in patch ω_i such that $\phi_i^0(x_j)=\delta_{ij}$. For simplicity of notation, denote $u_1=1$. Then, the multiscale finite element method for each patch ω_i is constructed by

$$\psi_{ijkl}(x)=\phi_i^0(x)u_j\left(x,\theta_k\right), \tag{4.12}$$

where $j = 1, .., N$ and i is the index of nodes (see Figure 4.3). Note that in each K, $\sum_{i=1}^{n} \psi_{ijkl}(x) = u_j(x, \theta_k)$ is the desired single-phase flow solution.

We define the Galerkin finite element space by

$$V_h = span\{\psi_{ijkl}^K : i = 1, \cdots, d, j = 1, \ldots, N; K \in \tau_h\}. \tag{4.13}$$

The weak formulation of (2.2) is to find $p_h \in V_h$ such that

$$(\lambda k \nabla p_h, \nabla q_h) = (f, q_h) \qquad \text{for any } q_h \in V_h, \tag{4.14}$$

where $(\cdot, \cdot)$ denotes inner product in L^2.

To estimate the error between our fine-scale pressure and multiscale pressure, we assume that the pressure field for each θ is a smooth function of $u_i(x, \theta)$. This assumption for each θ has been shown in [1] for channelized two-phase flow and in [38] for more general permeability fields under the assumption that $\lambda(x)$ is a smooth function. Thus, for our analysis we make the following assumption. Assume there exists a sufficiently smooth scalar valued function $\tilde{G}(\eta)$, $\eta \in R^N$ $(G \in C^3)$, such that

$$|p\,(x, \theta) - \tilde{G}(u_1\,(x, \theta)\,, ..., u_N\,(x, \theta))|_{1,\Omega} \leq C\delta, \tag{4.15}$$

where δ is sufficiently small.

Theorem 4.2.1. *Under the assumption (4.15) and $u_i\,(x, \theta) \in W^{1,s}(\Omega)$, $s > 2$, $i = 1, ..., N$, we have*

$$|p\,(x, \theta) - p_h\,(x, \theta)\,|_{1,\Omega} \leq C\delta + Ch^{1-2/s}, \tag{4.16}$$

for each θ.

Remark 4.2.1. *We present the proof of this theorem below, but first note that in a typical application, we assume that θ is in a compact set. In this case, one can obtain from Arzela-Ascoli that (4.16) holds uniformly with respect to θ. We note that one* *can also consider local basis functions following the previous discussion.*

Proof. First, we note the following stability estimate

$$|p - p_h|_{1,\Omega} \leq C|p - c_{ijk}\psi_{ijk}|_{1,\Omega}, \tag{4.17}$$

where c_{ijk} is chosen later. We remind that the index k refers to a realization θ_k, and p is evaluated at an arbitrary θ. The estimate (4.17) can be written as

$$\begin{aligned} |p(x,\theta) - p_h|_{1,\Omega} \leq |p - \tilde{G}(u_1(x,\theta), ..., u_N(x,\theta))|_{1,\Omega} + \\ |\tilde{G}(u_1(x,\theta), ..., u_N(x,\theta)) - G(u_1(x,\theta_k), ..., u_N(x,\theta_k))|_{1,\Omega} + \\ |G(u_1(x,\theta_k), ..., u_N(x,\theta_k)) - c_{ijk}\psi_{ijk}|_{1,\Omega}, \end{aligned} \tag{4.18}$$

where c_{ijk} is chosen later and $G(u_1(x,\theta_k), ..., u_N(x,\theta_k))$ depends on all k, i.e., $G : R^{NM} \to R$,

$$G(u_1(x,\theta_k), ..., u_N(x,\theta_k)) = G(u_1(x,\theta_1), ..., u_N(x,\theta_M)).$$

The first term on the right hand side of (4.18) can be estimated based on our assumption (4.15). To estimate the second term on the right hand side of (4.18), we note that

$$u_i(x,\theta) = \sum_j u_i(x,\theta_k)L_k(\theta) + E, \tag{4.19}$$

where E is the error term.

If we assume (4.19) is exact at $\theta = \theta_k$, then $G = \tilde{G}$ at $\theta = \theta_k$. Thus, from standard interpolation results assuming G is sufficiently smooth with respect to θ, we have

$$|\tilde{G}(u_1(x,\theta), ..., u_N(x,\theta)) - G(u_1(x,\theta_k), ..., u_N(x,\theta_k))|_{1,\Omega} \leq C_G\delta_1, \tag{4.20}$$

where C_G depends on the derivative of $\tilde{G}$ and θ. Provided θ is in a compact set, C_G

is uniformly bounded.

Next, we present the choice of c_{ijk} and the estimate for the third term on the right hand side of (4.18). In each ω_i, we choose c_{ijk} as

$$c_{ijk} = \frac{\partial G}{\partial u_j}(\overline{u}_1^i(\theta_k), ..., \overline{u}_N^i(\theta_k)),\ j \geq 2$$

and

$$c_{i1k} = G(\overline{u}_1^i(\theta_k), ..., \overline{u}_N^i(\theta_k)) - \frac{\partial G}{\partial u_j}(\overline{u}_1^i(\theta_k), ..., \overline{u}_N^i(\theta_k))\overline{u}_j^i(\theta_k),$$

where $\overline{u}_j^i$ is the average of u_j over ω_i for each (θ_k). We note the following Taylor expansion in each ω_i and for each θ_k

$$G(u_1, ..., u_N) = G(\overline{u}_1^i, ..., \overline{u}_N^i) + \frac{\partial G}{\partial u_j}(\overline{u}_1^i, ..., \overline{u}_N^i)(u_j - \overline{u}_j^i) + R_i,$$

where R_i is the remainder given by

$$R_i = \sum_{j,m} \frac{1}{2}\frac{\partial^2 G}{\partial u_m \partial u_j}(\xi_1^i, ..., \xi_N^i)(u_m - \overline{u}_k^i)(u_j - \overline{u}_j^i),$$

where $\xi_m^i = \overline{u}_m^i + \zeta^i(u_m - \overline{u}_k^i)$, $0 < \zeta^i < 1$. Note that we omit the index of θ_k in the remainder for simplicity. Then, it can be shown that in each ω_i (fixed i)

$$\begin{aligned} |G(u_1(x,\theta_k), ..., u_N(x,\theta_k)) - c_{ijk}u_j(x,\theta_k)|_{1,\omega_i} \leq \\ |G(\overline{u}_1(\theta_k), ..., \overline{u}_N(\theta_k)) + \\ \frac{\partial G}{\partial u_j}(\overline{u}_1(\theta_k), ..., \overline{u}_N(\theta_k))(u_j(x,\theta_k) - \overline{u}_j(\theta_k)) - c_{ijk}u_j(x,\theta_k)|_{1,\omega_i} + |R|_{1,\omega_i}. \end{aligned} \tag{4.21}$$

The first term on the right hand side is zero because of the choice of c_{ijk}. Under the assumption that $u_i \in W^{1,s}$ $(s > 2)$, we have the following estimate for the remainder (we ignore the index of θ_k for simplicity):

$$
\begin{aligned}
&|R_i|_{1,\omega_i} \leq \\
&C\sum_{l,j,k}\|\frac{\partial^3 G}{\partial u_l \partial u_j \partial u_k}\nabla u_l(u_j-\overline{u}^i_j)(u_k-\overline{u}^i_k)\|_{0,\omega_i} + C\sum_{l,j\neq l}\|\frac{\partial^2 G}{\partial u_l \partial u_j}(u_j-\overline{u}^i_j)\|_{0,\omega_i} + \\
&C\sum_{l}\|\frac{\partial^2 G}{\partial u_l^2}(u_l-\overline{u}^i_l)^2\nabla u_l\|_{0,\omega_i} \leq \\
&C\sum_{l,j,k} Ch^{2-4/s}|u_j|_{1,\omega_i}|u_k|_{1,\omega_i}|u_l|_{1,\omega_i} + C\sum_{l,j\neq l} h^{1-2/s}|u_j|_{1,\omega_i} + C\sum_{l} h^{2-4/s}|u_l|^2_{1,\omega_i} \leq \\
&C\sum_{l} h^{2-4/s}|u_l|_{1,\omega_i} + C\sum_{l,j\neq l} h^{1-2/s}|u_j|_{1,\omega+i} \leq \\
&Ch^{1-2/s}\sum_{l}|u_l|_{1,\omega_i}.
\end{aligned}
$$

It can be easily shown that

$$
\begin{aligned}
|R_i|_{0,\omega_i} &\leq C\sum_{j,k}\|(u_j-\overline{u}^i_j)(u_k-\overline{u}^i_k)\|_{0,\omega_i} \leq \\
C\sum_{j,k} h^{2-4/s}&|u_j|_{1,\omega_i}|u_k|_{1,\omega_i}h \leq Ch^{3-4/s}\sum_{j}|u_j|_{1,\omega_i}.
\end{aligned}
$$

Following [3], we have

$$
\begin{aligned}
&|G(u_1,...,u_N) - c_{ijk}\psi_{ijk}|^2_{1,\Omega} = \int_\Omega |\nabla(G - c_{ijk}\phi^0_i u_j(x,\theta_k))|^2 dx = \\
&\int_\Omega |\nabla(\phi^0_i(G - c_{ijk}u_j(x,\theta_k)))|^2 dx \leq \\
&C\int_\Omega |(G - c_{ijk}u_j(x,\theta)k))\nabla\phi^0_i|^2 dx + C\int_\Omega |\phi^0_i\nabla(G - c_{ijk}u_j(x,\theta_k))|^2 dx \leq \\
&\frac{1}{h^2}\sum_i\int_{\omega_i}|R_i|^2 dx + \sum_i\int_{\omega_i}|\nabla R_i|^2 dx \leq \\
&C\sum_i\frac{1}{h^2}h^{6-8/s}\sum_j|u_j|^2_{1,\omega_i} + Ch^{2-4/s}\sum_i\sum_j|u_j|^2_{1,\omega_i} \leq Ch^{4-8/s} + Ch^{2-4/s},
\end{aligned}
$$

where we have used the fact that $\sum_i \phi^0_i = 1$ and C depends on the overlapping index of ω_i's. Consequently, we have the following error estimate

$$|p - p_h|_{1,\Omega} \leq C\delta + Ch^{1-2/s}.$$

□

Finally, we would like to note that one can extend the above approach to mixed multiscale finite element methods. In [1], the authors have presented mixed multiscale finite element methods using multiple global information. In mixed multiscale finite element methods, the basis functions for the velocity field are constructed. Taking the global fields to be the single-phase flow velocities at sparse collocation points, one can also obtain the mixed multiscale method for stochastic porous media equations [1].

We present a simple one-dimensional numerical example to demonstrate the efficiency of multiscale finite element methods. We consider a permeability given by

$$k\left(x,\theta\right) = \exp\left(\theta\Phi(x)\right),$$

where $\Phi(x)$ is a highly oscillatory function. Using this $k(x,\theta)$ we wish to solve the one-dimensional version of the single-phase flow equation on $\Omega = [0,1]$

$$\begin{aligned} (k(x,\theta)p'(x,\theta))' &= f(x), \\ p(x,\theta) &= 0 \text{ for } x \,\text{on}\; \partial\Omega. \end{aligned} \tag{4.22}$$

We take $f(x) = -1$. Let us solve this equation exactly for $\Omega = [0,1]$. After one integration we have

$$k\left(x,\theta\right)p'\left(x,\theta\right) = -x + C$$

thus

$$p'\left(x,\theta\right) = \frac{-x}{k\left(x,\theta\right)} + \frac{C}{k\left(x,\theta\right)}.$$

Taking the integral from 0 to 1 of both sides and using the fact that we have zero Dirichlet boundary conditions, we have

$$\int_0^1 p'(x,\theta)\,dx = \int_0^1 \frac{-x}{k(x,\theta)} + \frac{C}{k(x,\theta)}\,dx = 0.$$

Thus

$$C\left\langle \frac{1}{k(x,\theta)} \right\rangle = \left\langle \frac{x}{k(x,\theta)} \right\rangle$$

or

$$C = \frac{\langle x/k(x,\theta)\rangle}{\langle 1/k(x,\theta)\rangle},$$

where $\langle\cdot\rangle$ denotes the integral over 0 to 1.

We now have that the solution to (4.22) can be written as

$$p(x,\theta) = \int \frac{-x}{k(x,\theta)} + \frac{\langle x/k(x,\theta)\rangle}{\langle 1/k(x,\theta)\rangle}\frac{1}{k(x,\theta)}\,dx. \tag{4.23}$$

We wish to compare this exact solution to one obtained by using multiscale finite element methods (MsFEM).

To find the basis function for the MsFEM, we begin by assuming $\Omega = [0,1]$ is partitioned into L segments, with nodes $x_0 = 0$ to $x_L = 1$. We solve

$$\left(k(x,\theta_k)\,\varphi_i'(x,\theta_k)\right)' = 0,\ \varphi_i(x_{i-1}) = 0,\ \varphi_i(x_i) = 1$$

on each $\Omega_i = [x_{i-1},\, x_i]\,,\ i = 1\ldots L$. Similar to the exact solution, we can solve for the basis functions analytically.

Using the basis functions φ_i, we find $\tilde{p}(x,\theta) \approx p(x,\theta)$ at a general θ by using linear combinations of φ_i. In other words, we wish to write

$$\tilde{p}(x,\theta) = \sum_{i,k} \alpha_{ik}\varphi_i(x,\theta_k)\,. \tag{4.24}$$

Plugging this $\tilde{p}$ into (4.22) gives us

$$\left(\sum_{i,k} \alpha_{ik} \varphi_i' \left(x, \theta_k\right) k\left(x, \theta\right)\right)' = -1.$$

We multiply both sides by $\varphi_j\left(x, \theta_m\right)$ and integrate by parts over Ω_j to get

$$\sum_{i,k} \alpha_{ik} \int_{\Omega_j} k\left(x, \theta\right) \varphi_i'\left(x, \theta_k\right) \varphi_j'\left(x, \theta_m\right)\, dx = \int_{\Omega_j} -\varphi_j\left(x, \theta_m\right)\, dx. \tag{4.25}$$

Since $\varphi_i'\left(x, \theta_k\right)$ has support only on Ω_i, we have $\varphi_i'\left(x, \theta_k\right) \varphi_j'\left(x, \theta_m\right) = 0$ unless $j = i-1$, i or $i+1$. Thus we can rewrite (4.25) as

$$\sum_{i,k} \alpha_{ik} \int_{\Omega_{ij}} k\left(x, \theta\right) \varphi_i'\left(x, \theta_k\right) \varphi_j'\left(x, \theta_m\right)\, dx = \int_{\Omega_j} -\varphi_j\left(x, \theta_m\right)\, dx, \tag{4.26}$$

where $\Omega_{ij} = \Omega_j \cap \Omega_i$. We solve the above linear system for α_{ik} and obtain $\tilde{p}$.

We test the above method on a problem with 4, 6 and 8 intervals in $\Omega = [0, 1]$ and $\Phi(x) = \sin(40x)$. We use 3 different expansions for (4.24). First, we use only a single θ value, $\theta_1 = 0$. Next, we take two θ values, $\theta_1 = -2$ and $\theta_2 = 2$. Lastly, we take three θ values, $\theta_1 = -2$, $\theta_2 = 0$, and $\theta_3 = 2$. For a fixed $\theta = -1$ we compare the pressure p, found from the solution to (4.22) given by (4.23), and $\tilde{p}$, the approximate pressure computed by solving the linear system in (4.26). In Figure 4.4 we plot both p and $\tilde{p}$ for each of the above cases. The approximations behave as expected, with an increase in accuracy as the number of θ values used increases. In Table 4.2 we compare the errors, $||\tilde{p} - p||_2$, between p and $\tilde{p}$ as calculated from using collocation to compute the MsFEM basis functions. We observe that with more sampling points one achieves a higher accuracy for the same number of intervals. Moreover, with 3 collocation points, which corresponds to 1^{st} order Smolyak interpolation, one achieves approximately a 3% error for 6 intervals.

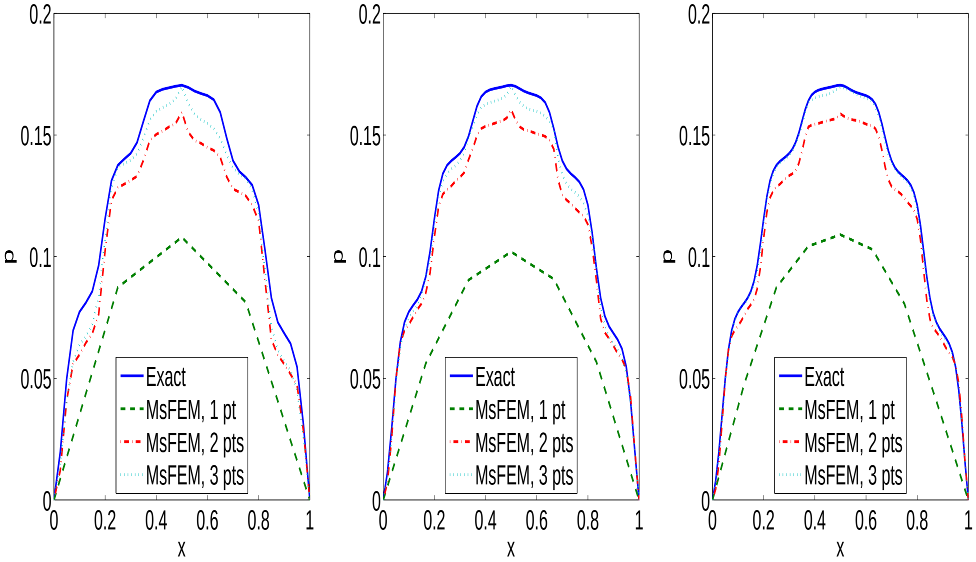


Fig. 4.4. Comparison between exact solution and MsFEM solution. Left: 4 intervals. Middle: 6 intervals. Right: 8 intervals.

Table 4.2. Error $||\tilde{p} - p||_2$ for 4, 6 and 8 intervals using one, two and three θ values.

Number of intervals	$\theta_1 = 0$	$\theta_1 = -2, \theta_2 = 2$	$\theta_1 = -2, \theta_2 = 0, \theta_3 = 2$
4	0.3280	0.0902	0.0534
6	0.4293	0.0792	0.0311
8	0.4397	0.0815	0.0123

CHAPTER V

THEORETICAL RESULTS FOR UNCERTAINTY QUANTIFICATION

The general goal in the uncertainty quantification problems we are interested in is to obtain a set of fields, usually conductivity or permeability, that reproduce some given or measured response. Our response is usually given in terms of some production data, such as oil-cut or average flux values. In both two-phase immiscible flow and Richards' equation, the sampling techniques are similar. The following results are presented in the context of some general unknown field, k, and some given integrated response, F. In our uncertainty quantification problems, we employ the Langevin algorithms within the context of Markov chain Monte Carlo (MCMC) methods for sampling k. Langevin algorithms provide efficient sampling techniques because they use the gradient information of the target distributions. However, the direct Langevin algorithm is very expensive because it requires the computation of the gradients with fine-scale simulations. Based on a coarse-scale model of the problem, we propose an approach where the gradients are computed using inexpensive coarse-scale simulation. These coarse-scale gradients may not be very accurate; therefore, the computed results are first tested with coarse-scale distributions. If the result is accepted at the first stage, then a fine-scale simulation is performed at the second stage to determine the acceptance probability. The first stage of the method modifies the Markov chain generated by the direct Langevin algorithms. It can be shown that the modified Markov chain satisfies the detailed balance condition for the correct distribution. Moreover, we point out that the chain is ergodic and converges to the correct posterior distribution under some technical assumptions. The validity of the assumptions for our application is also discussed.

5.1. Metropolis-Hasting MCMC

The most general problem under consideration consists of sampling an unknown random field k given some related measured data (or integrated response) F. In our cases, there are numerous fields k which reproduce the same response F. From the probabilistic point of view, the problem can be regarded as conditioning the random fields to the measured data with measurement errors. Consequently, our goal is to sample from the conditional distribution $P(k|F)$, where k is the fine-scale field and F is the related data. Using the Bayes formula we can write

$$P(k|F) \propto P(F|k)P(k). \tag{5.1}$$

In the above formula, $P(k)$ is the unconditioned (prior) distribution of the random field. In practice, the measured data contains measurement errors. We assume that the measurement error satisfies a Gaussian distribution, thus, the likelihood function $P(F|k)$ takes the form

$$P(F|k) \propto \exp\Big(-\frac{\|F - F_k\|^2}{\sigma_f^2}\Big), \tag{5.2}$$

where F is the reference data, F_k is the data for the field k, and σ_f is the measurement precision. In most of our applications, F_k is a quantity found using the solution a nonlinear PDE system for the given k. Depending on the data F we must define a relevant norm for $\|F - F_k\|^2$. For example, if both F and F_k are functions of time (denoted by t), the norm $\|F - F_k\|^2$ is defined as the L_2 norm

$$\|F - F_k\|^2 = \int_0^T [F(t) - F_k(t)]^2 \, dt,$$

where T is the time of the available history.

Denote the sampling target distribution as

$$\pi(k) = P(k|F) \propto \exp\Big(-\frac{\|F - F_k\|^2}{\sigma_f^2}\Big)P(k). \tag{5.3}$$

Since different fields may produce the same data, the distribution $\pi(k)$ is a function of k with multiple local maxima. Sampling from the distribution $\pi(k)$ can be accomplished by the MCMC method. For a given proposal distribution $q(y|x)$, the Metropolis-Hasting MCMC algorithm (see, e.g., [40], page 233) consists of the following steps.

Algorithm I (Metropolis-Hasting MCMC, Robert and Casella [40])

- Step 1. At k_n generate Y from $q(Y|k_n)$.
- Step 2. Accept Y as a sample with probability

  $$p(k_n, Y) = \min\left(1, \frac{q(k_n|Y)\pi(Y)}{q(Y|k_n)\pi(k_n)}\right), \tag{5.4}$$

  i.e. take $k_{n+1} = Y$ with probability $p(k_n, Y)$, and $k_{n+1} = k_n$ with probability $1 - p(k_n, Y)$.

The MCMC algorithm generates a Markov chain $\{k_n\}$ whose stationary distribution is $\pi(k)$.

A remaining question is how to choose an efficient proposal distribution $q(k|k_n)$. Two commonly used types of proposal distributions are the independent sampler and the random walk sampler. In the case of the independent sampler, the proposal distribution $q(k|k_n)$ is chosen to be independent of k_n and equal to the prior (unconditioned) distribution. In the random walk sampler, the proposal distribution depends on the previous value of k and is given by

$$k = k_n + \epsilon_n, \tag{5.5}$$

where ϵ_n is a random perturbation with prescribed distribution. If the variance of ϵ_n is chosen to be very large, then the random walk sampler becomes similar to the independent sampler. Although the random walk sampler allows us to accept more realizations, it often gets stuck in the neighborhood of a local maximum of the target distribution.

An important type of proposal distribution can be derived from the Langevin diffusion, as proposed by Grenander and Miller [24]. The Langevin diffusion is defined by the stochastic differential equation

$$dk(\tau) = \frac{1}{2}\nabla \log \pi(k(\tau))d\tau + dW_\tau, \tag{5.6}$$

where W_τ is the standard Brownian motion vector with independent components. It can be shown that the diffusion process $k(\tau)$ has $\pi(k)$ as its stationary distribution. The actual implementation of the Langevin diffusion requires a discretization of the equation (5.6),

$$k_{n+1} = k_n + \frac{\Delta\tau}{2}\nabla \log \pi(k_n) + \sqrt{\Delta\tau}\epsilon_n,$$

where ϵ_n are independent standard normal distributions. However, the discrete solution k_n can have vastly different asymptotic behavior from the continuous diffusion process $k(\tau)$ [40]. In general, the discrete solution k_n does not necessarily have $\pi(k)$ as its stationary distribution. Instead of taking k_n as direct samples, we use them as test proposals for Algorithm I. The samples will be further tested and corrected by the Metropolis acceptance-rejection step (5.4). Consequently, we choose the proposal generator $q(Y|k_n)$ in Algorithm I as

$$Y = k_n + \frac{\Delta\tau}{2}\nabla \log \pi(k_n) + \sqrt{\Delta\tau}\epsilon_n. \tag{5.7}$$

Since ϵ_n are independent Gaussian vectors, the transition distribution of the proposal generator (5.7) is

$$\begin{aligned} q(Y|k_n) &\propto \exp\left(-\frac{\|Y - k_n - \frac{\Delta\tau}{2}\nabla \log \pi(k_n)\|^2}{2\Delta\tau}\right), \\ q(k_n|Y) &\propto \exp\left(-\frac{\|k_n - Y - \frac{\Delta\tau}{2}\nabla \log \pi(Y)\|^2}{2\Delta\tau}\right). \end{aligned} \tag{5.8}$$

The scheme (5.7) can be regarded as a problem-adapted random walk. The gradient information of the target distribution is included to enforce a biased random walk. The use of the gradient information in inverse problems for subsurface characterization is not new. In their original work, Oliver et al. [36, 37] developed the randomized maximum likelihood method, which uses the gradient information of the target distribution. This approach uses unconditional realizations of the data and solves a deterministic gradient-based minimization problem. The solution of this minimization problem is taken as a proposal and is accepted with probability one, since the acceptance probability is very difficult to estimate. In addition to needing to solve a gradient-based inverse problem, this method does not guarantee a proper sampling of the posterior distribution. Thus, developing efficient and rigorous MCMC calculations with high acceptance rates remains a challenging problem. Though the Langevin formula (5.7) resembles the randomized maximum likelihood method, it is more efficient and rigorous, and one can compute the acceptance probability easily. The Langevin algorithms also allow us to achieve high acceptance rates. However, computing the gradients of the target distribution is very expensive. We propose to use the coarse-scale solutions in the computation of the gradients to speed up the

Langevin algorithms.

5.2. Langevin MCMC Method Using Coarse-Scale Models

The major computational cost of Algorithm I is in computing the value of the target distribution $\pi(k)$ for different fields, k. Since generally the map between the k and the response F_k is governed by a PDE system, there is no explicit formula for the target distribution $\pi(k)$. To compute the function $\pi(k)$, we need to solve a nonlinear PDE system on the fine-scale for the given k. For the same reason, we need to compute the gradient of $\pi(k)$ in (5.7) numerically (by finite differences), which involves solving a nonlinear PDE system multiple times. To compute the acceptance probability (5.4), the PDE system needs to be solved one more time. As a result, the direct (full) MCMC simulations with Langevin samples are prohibitively expensive.

To bypass the above difficulties, we design a coarse grid Langevin MCMC algorithm where most of the fine-scale computations are replaced by the coarse-scale ones. Based on a coarse grid model of the distribution $\pi(k)$, we first generate samples from (5.7) using the coarse-scale gradient of $\pi(k)$, which requires solving a PDE system on the coarse grid. Then we further filter the proposals by an additional Metropolis acceptance-rejection test on the coarse grid. If the sample does not pass the coarse grid test, the sample is rejected and no further fine-scale test is necessary. The argument for this procedure is that if a proposal is not accepted by the coarse grid test, then it is unlikely to be accepted by the fine-scale test either. By eliminating most of the "unlikely" proposals with cheap coarse-scale tests, we can avoid wasting CPU time simulating the rejected samples on the fine-scale. We note these procedures have been used before within the context of porous media [15, 13, 19, 33].

To model $\pi(k)$ on the coarse-scale, we define a coarse grid map F_k^* between the field k and the response F. The map F_k^* is determined by solving the PDE system on a coarse grid. Consequently, the target distribution $\pi(k)$ can be approximated by

$$\pi^*(k) \propto \exp\left(-\frac{\|F - F_k^*\|^2}{\sigma_c^2}\right) P(k), \tag{5.9}$$

where σ_c is the measurement precision on the coarse grid, and should be slightly larger than σ_f. Then the Langevin samples are generated from (5.7) using the coarse grid gradient of the target distribution

$$Y = k_n + \frac{\Delta\tau}{2}\nabla \log \pi^*(k_n) + \sqrt{\Delta\tau}\epsilon_n. \tag{5.10}$$

The transition distribution of the coarse grid proposal (5.10) is

$$\begin{aligned} q^*(Y|k_n) &\propto \exp\left(-\frac{\|Y - k_n - \frac{\Delta\tau}{2}\nabla \log \pi^*(k_n)\|^2}{2\Delta\tau}\right), \\ q^*(k_n|Y) &\propto \exp\left(-\frac{\|k_n - Y - \frac{\Delta\tau}{2}\nabla \log \pi^*(Y)\|^2}{2\Delta\tau}\right). \end{aligned} \tag{5.11}$$

To compute the gradient of $\pi^*(k_n)$ numerically, we only need to solve the PDE system on the coarse grid. The coarse-scale distribution $\pi^*(k)$ serves as a regularization of the original fine-scale distribution $\pi(k)$. By replacing the fine-scale gradient with the coarse-scale gradient, we can reduce the computational cost dramatically but still direct the proposals to regions with larger probabilities.

Because of the high dimension of the problem and the discretization errors, most proposals generated by the Langevin algorithms (both (5.7) and (5.10)) will be rejected by the Metropolis acceptance-rejection test (5.4). To avoid wasting expensive fine-scale computations on unlikely acceptable samples, we further filter the Langevin proposals by the coarse-scale acceptance criteria

$$g(k_n, Y) = \min\left(1, \frac{q^*(k_n|Y)\,\pi^*(Y)}{q^*(Y|k_n)\pi^*(k_n)}\right),$$

where $\pi^*(k)$ is the coarse-scale target distribution (5.9). $q^*(Y|k_n)$ and $q^*(k_n|Y)$ are the coarse-scale proposal distributions given by (5.11). Combining all the discussion above, we have the following revised MCMC algorithm [8].

Algorithm II (Preconditioned Coarse-Gradient Langevin Algorithm)

- Step 1. At k_n, generate a trial proposal Y from the coarse Langevin algorithm (5.10).

- Step 2. Take the proposal k as

  $$k = \begin{cases} Y \,\text{with probability}\, g(k_n, Y), \\ k_n \,\text{with probability}\, 1 - g(k_n, Y), \end{cases}$$

  where

  $$g(k_n, Y) = \min\left(1, \frac{q^*(k_n|Y)\pi^*(Y)}{q^*(Y|k_n)\pi^*(k_n)}\right).$$

  Therefore, the proposal k is generated from the effective instrumental distribution

  $$Q(k|k_n) = g(k_n, k)q^*(k|k_n) + \left(1 - \int g(k_n, k)q^*(k|k_n)dk\right)\delta_{k_n}(k). \qquad (5.12)$$

- Step 3. Accept k as a sample with probability

  $$\rho(k_n, k) = \min\left(1, \frac{Q(k_n|k)\pi(k)}{Q(k|k_n)\pi(k_n)}\right), \qquad (5.13)$$

  i.e., $k_{n+1} = k$ with probability $\rho(k_n, k)$, and $k_{n+1} = k_n$ with probability $1 - \rho(k_n, k)$.

Step 2 screens the trial proposal Y by the coarse grid distribution before passing it to the fine-scale test. The filtering process changes the proposal distribution of the algorithm from $q^*(Y|k_n)$ to $Q(k|k_n)$ and serves as a preconditioner to the MCMC method. This is why we call it the preconditioned coarse-gradient Langevin algorithm. We note that testing proposals by approximate target distributions is not a very new idea. Similar strategies have been developed previously in [15, 30, 6, 13, 19, 33].

Note that there is no need to compute $Q(k|k_n)$ and $Q(k_n|k)$ in (5.13) by formula (5.12). The acceptance probability (5.13) can be simplified as

$$\rho(k_n, k) = \min\left(1, \frac{\pi(k)\pi^*(k_n)}{\pi(k_n)\pi^*(k)}\right). \tag{5.14}$$

In fact, this is obviously true for $k = k_n$ since $\rho(k_n, k_n) \equiv 1$. For $k \neq k_n$,

$$\begin{aligned} Q(k_n|k) = g(k, k_n)q(k_n|k) &= \frac{1}{\pi^*(k)} \min\Big(q(k_n|k)\pi^*(k),\ q(k|k_n)\pi^*(k_n)\Big) \\ &= \frac{q(k|k_n)\pi^*(k_n)}{\pi^*(k)} g(k_n, k) = \frac{\pi^*(k_n)}{\pi^*(k)} Q(k|k_n). \end{aligned}$$

Substituting the above formula into (5.13), we immediately get (5.14).

In Algorithm II, the proposals generated by (5.10) are screened by the coarse-scale acceptance-rejection test to reduce the number of unnecessary fine-scale simulations. One can skip that preconditioning step and get the following algorithm.

Algorithm III (Coarse-Gradient Langevin Algorithm)

- Step 1. At k_n, generate a trial proposal Y from the coarse Langevin algorithm (5.10).

- Step 2. Accept Y as a sample with probability

  $$\rho(k_n, Y) = \min\left(1,\ \frac{q^*(k_n|Y)\pi(Y)}{q^*(Y|k_n)\pi(k_n)}\right), \tag{5.15}$$

  i.e. $k_{n+1} = Y$ with probability $\rho(k_n, Y)$, and $k_{n+1} = k_n$ with probability $1 - \rho(k_n, Y)$.

We will demonstrate numerically that Algorithm II is indeed more efficient than

Algorithm III.

Preconditioning the MCMC algorithms by coarse-scale models is studied in [19, 13]. There the independent sampler and random walk sampler are used as the instrumental distribution. Our goal is to show that one can use coarse-scale models, in Langevin algorithms. More specifically, we can use coarse-scale gradients instead of fine-scale gradients in these algorithms. Our numerical experiments show that the coarse-scale distribution somewhat regularizes (smooths) the fine-scale distribution, which allows us to take larger time steps in the Langevin algorithm (5.10). In addition, we employ the preconditioning technique from [19] to increase the acceptance rate of the coarse-gradient Langevin algorithms.

5.2.1. Analysis of the Preconditioned Coarse-Gradient Langevin Algorithm

In this section, we will briefly discuss the convergence property of the preconditioned coarse grid Langevin algorithm. Denote

$$\begin{aligned} &\mathcal{E} = \big\{k;\ \pi(k) > 0\big\}, \\ &\mathcal{E}^* = \big\{k;\ \pi^*(k) > 0\big\}, \\ &\mathcal{D} = \big\{k;\ q^*(k|k_n) > 0 \text{ for any } k_n \in \mathcal{E}\big\}. \end{aligned} \tag{5.16}$$

The set $\mathcal{E}$ is the support of the posterior (target) distribution $\pi(k)$. $\mathcal{E}$ contains all the fields k which have a positive probability of being accepted as a sample. Similarly, $\mathcal{E}^*$ is the support of the coarse-scale distribution $\pi^*(k)$, which contains all the k acceptable by the coarse-scale test. $\mathcal{D}$ is the set of all possible proposals which can be generated by the Langevin distribution $q^*(k|k_n)$. To make the coarse-gradient Langevin MCMC methods sample properly, the conditions $\mathcal{E} \subseteq \mathcal{D}$ and $\mathcal{E} \subseteq \mathcal{E}^*$ must hold (up to a zero measure set) simultaneously. If one of these conditions is violated, say, $\mathcal{E} \not\subseteq \mathcal{E}^*$, then there will exist a subset $A \subset (\mathcal{E} \setminus \mathcal{E}^*)$ such that

$$\pi(A) = \int_A \pi(k)dk > 0 \qquad \text{and} \qquad \pi^*(A) = \int_A \pi^*(k)dk = 0,$$

which means no element of A can pass the coarse-scale test and A will never be visited by the Markov chain $\{k_n\}$. For Langevin algorithms, $\mathcal{E} \subset \mathcal{D}$ is always satisfied since $\mathcal{D}$ is the whole space. By choosing the parameter σ_c in $\pi^*(k)$ properly, the condition $\mathcal{E} \subset \mathcal{E}^*$ can also be satisfied. A typical choice would be $\sigma_c \approx \sigma_f$. More discussions on the choice of σ_c can be found in [19], where a two-stage MCMC algorithm is discussed.

Denote by K the transition kernel of the Markov chain $\{k_n\}$ generated by Algorithm II. Since its effective instrumental proposal is $Q(k|k_n)$, the transition kernel K has the form

$$\begin{aligned} &K(k_n, k) = \rho(k_n, k)Q(k|k_n), \;\; k \neq k_n, \\ &K(k_n, \{k_n\}) = 1 - \textstyle\int_{k \neq k_n} \rho(k_n, k)Q(k|k_n)dk. \end{aligned} \tag{5.17}$$

That is, the transition kernel $K(k_n, \cdot)$ is continuous when $k \neq k_n$ and has a positive probability at the point $k = k_n$. First we show that $K(k_n, k)$ satisfies the detailed balance condition, that is

$$\pi(k_n)K(k_n, k) = \pi(k)K(k, k_n) \tag{5.18}$$

for all k, k_n. The equality is obvious when $k = k_n$. If $k \neq k_n$, then

$$\pi(k_n)K(k_n, k) = \pi(k_n)\rho(k_n, k)Q(k|k_n) = \min\Big(Q(k|k_n)\pi(k_n),\; Q(k_n|k)\pi(k)\Big) =$$
$$\min\left(\frac{Q(k|k_n)\pi(k_n)}{Q(k_n|k)\pi(k)},\; 1\right) Q(k_n|k)\pi(k) = \rho(k, k_n)Q(k_n|k)\pi(k) = \pi(k)K(k, k_n).$$

Using the detailed balance condition (5.18), we can easily show that for any measurable set $A \subset \mathcal{E}$ the expression $\pi(A) = \int K(k, A)dk$ holds. So $\pi(k)$ is indeed the stationary distribution of the transition kernel $K(k_n,\, k)$.

In Algorithm II, the proposal distribution (5.10) satisfies the positivity condition

$$q^*(k|k_n) > 0 \text{ for every } (k_n, k) \in \mathcal{E} \times \mathcal{E}. \tag{5.19}$$

With this property, we can easily prove the following lemma.

Lemma 5.2.1. *If $\mathcal{E} \subset \mathcal{E}^*$, then the chain $\{k_n\}$ generated by Algorithm II is strongly π-irreducible.*

Proof. According to the definition of strong irreducibility, we only need to show that $K(k_n, A) > 0$ for any $k_n \in \mathcal{E}$ and any measurable set $A \subset \mathcal{E}$ with $\pi(A) > 0$. From the formula (5.17) we have

$$\begin{aligned} K(k_n, A) &\geq \int_{A\setminus k_n} K(k_n, k)dk = \int_{A\setminus k_n} \rho(k_n, k)Q(k_n, k)dk \\ &= \int_{A\setminus k_n} \rho(k_n, k)g(k_n, k)q(k|k_n)dk. \end{aligned}$$

In the above inequality, the equal sign holds when $k_n \notin A$. Since $\pi(A) = \int_A \pi(k)dk > 0$, it follows that $m(A) = m(A \setminus k_n) > 0$, where $m(A)$ is the Lebesgue measure. If $\mathcal{E} \subset \mathcal{E}^*$, then both $\rho(k_n, k)$ and $g(k_n, k)$ are positive in A. Combining the positivity condition (5.19), we can easily conclude that $K(k_n, A) > 0$, which completes the proof. □

For the transition kernel (5.17) of Algorithm II, there always exist certain states $\kappa \in \mathcal{E}$ such that $K(\kappa, \{\kappa\}) > 0$. That is, if the Markov chain is on state κ at step n, then it has a positive probability to remain on state κ at step $n + 1$. This condition ensures that the Markov chain generated by Algorithm II is aperiodic. Based on the irreducibility and stability property of Markov chains [40, 34], the following convergence result is readily available.

Theorem 5.2.1. *(Robert and Casella [40]) The Markov chain $\{k_n\}$ generated by the preconditioned coarse-gradient Langevin algorithm is ergodic: for any function $h(k)$,*

$$\lim_{N\to\infty}\frac{1}{N}\sum_{n=1}^{N}h(k_n)=\int h(k)\pi(k)dk. \tag{5.20}$$

Moreover, the distribution of k_n converges to $\pi(k)$ in the total variation norm

$$\lim_{n\to\infty}\sup_{A\in\mathcal{B}(\mathcal{E})}\left|K^n(k_0,A)-\pi(A)\right|=0 \tag{5.21}$$

for any initial state k_0, where $\mathcal{B}(\mathcal{E})$ denote all the measurable subsets of $\mathcal{E}$.

5.3. Langevin MCMC Method Using Collocation

We now consider the use of collocation methods discussed in Section 3.2 along with coarse grid models as a replacement for the MCMC methods using coarse grid models. Instead of repeating the same derivation as for Algorithm II, we simply note that in the previous equations, one can replace the coarse-scale target distribution π^* with an interpolated coarse-scale target distribution $\tilde{\pi}^*$. We will consider two ways to obtain $\tilde{\pi}^*$. The first, we interpolate to find $\tilde{\pi}^*$ based directly on values of π^* at sparse collocation points. Second, we consider interpolation of the coarse-scale response F^* by $\tilde{F}^*$. Based on the interpolated coarse-scale response $\tilde{F}^*$, we obtain an interpolated coarse-scale target distribution $\tilde{\pi}^*$. We can now present the revised MCMC algorithm using interpolation.

Algorithm IV (Preconditioned Interpolated Coarse-Gradient Langevin Algorithm)

- Step 1. At k_n, generate a trial proposal

$$Y=k_n+\frac{\Delta\tau}{2}\nabla\log\tilde{\pi}^*(k_n)+\sqrt{\Delta\tau}\epsilon_n.$$

- Step 2. Take the proposal k as

$$k = \begin{cases} Y \,\text{with probability}\, g(k_n, Y), \\ k_n \,\text{with probability}\, 1 - g(k_n, Y), \end{cases}$$

where

$$g(k_n, Y) = \min\left(1, \frac{\tilde{q}^*(k_n|Y)\tilde{\pi}^*(Y)}{\tilde{q}^*(Y|k_n)\tilde{\pi}^*(k_n)}\right).$$

Therefore, the proposal k is generated from the effective instrumental distribution

$$Q(k|k_n) = g(k_n, k)\tilde{q}^*(k|k_n) + \left(1 - \int g(k_n, k)\tilde{q}^*(k|k_n)dk\right)\delta_{k_n}(k). \qquad (5.22)$$

- Step 3. Accept k as a sample with probability

$$\rho(k_n, k) = \min\left(1, \frac{Q(k_n|k)\pi(k)}{Q(k|k_n)\pi(k_n)}\right), \qquad (5.23)$$

i.e., $k_{n+1} = k$ with probability $\rho(k_n, k)$, and $k_{n+1} = k_n$ with probability $1 - \rho(k_n, k)$.

Note that, as with each of the previous Langevin methods, we must compute the gradient of the logarithm of the target distribution. In Algorithm IV, we must compute $\nabla \log \tilde{\pi}^*(k_n)$ in each step. Previously, since π in Algorithm I and π^* in Algorithm II are not analytical functions, this gradient was computed using finite differences (or computed using an adjoint method). In each of the previous algorithms, this required solving a given PDE system in each dimension. If we are using a target distribution based on Smolyak interpolation, then we can write $\tilde{\pi}^*$ as an analytical function. Hence, we can compute the gradient analytically.

Since the operator in (3.19) is based on tensor products of $U_{i_1}, \ldots, U_{i_N}$ we simply need to take the derivative of each U_i in order to determine the gradient. For a particular stochastic dimension i we write

$$U_i(\theta) = \sum_{j=1}^{M_i} F\left(\theta_j^i\right) L_j^i(\theta).$$

Since i is fixed, $U_i(\theta)$ is a function of only one variable, thus

$$U_i'(\theta) = \sum_{j=1}^{M_i} F\left(\theta_j^i\right) L_j^{i\prime}(\theta).$$

Since the basis functions in Smolyak interpolation are the multidimensional Lagrange polynomials, we have

$$L_j^i(\theta) = \prod_{k=1,k\neq j}^{m_i} \frac{\theta - \theta_k^i}{\theta_j^i - \theta_k^i}.$$

By using an application of the product rule we find the derivative of L_j^i with respect to θ is given by the following

$$L_j^{i\prime}(\theta) = \sum_{s=1,s\neq j}^{m_i} \frac{1}{\theta_j^i - \theta_s^i} \prod_{k=1,k\neq j,s}^{m_i} \frac{\theta - \theta_k^i}{\theta_j^i - \theta_k^i}.$$

In the preconditioned interpolated coarse-gradient Langevin algorithm we must compute $\nabla \log \tilde{\pi}^*(\theta)$. Since we have

$$\nabla \log \tilde{\pi}^*(\theta) = \nabla \left(\frac{-\|F - \tilde{F}_\theta^*\|^2}{\sigma_c^2} \right)$$

then

$$\frac{\partial}{\partial \theta_j} \left(\frac{-\|F - \tilde{F}_\theta^*\|^2}{\sigma_c^2} \right) = -\frac{2}{\sigma_c^2} \|F - \tilde{F}_\theta^*\| \frac{\partial}{\partial \theta_j} \|F - \tilde{F}_\theta^*\|.$$

We write F_θ^k as the reference integrated response where superscript k denotes the dimension if the integrated response is not a scalar (such as a time-dependent response). Similarly we write $\tilde{F}_\theta^{*k}$ as the interpolated coarse-scale integrated response. In most of our practical applications, the integrated response is a vector function, and we use the discrete L^2 norm in the distributions. If we use the L^2 norm and assume the integrated response is M dimensional then we have

$$\nabla \log \tilde{\pi}^*(\theta) = \nabla \left(\frac{-\|F - \tilde{F}_\theta^*\|^2}{\sigma_c^2} \right) = \nabla \left(-\frac{1}{\sigma_c^2} \sum_{k=1}^{M} \left(F^k - \tilde{F}_\theta^{*k} \right)^2 \right)$$

and thus

$$\frac{\partial}{\partial \theta_j} \left(-\frac{1}{\sigma_c^2} \sum_{k=1}^{M} \left(F^k - \tilde{F}_\theta^{*k} \right)^2 \right) = -\frac{1}{\sigma_c^2} \sum_{k=1}^{M} 2 \left(F^k - \tilde{F}_\theta^{*k} \right) \frac{\partial}{\partial \theta_j} \tilde{F}_\theta^{*k}.$$

Writing the Smolyak interpolation formula for $\tilde{F}_\theta^{*k}$ out in terms of U_i functions, we find

$$\begin{gathered} \frac{\partial}{\partial \theta_j} \tilde{F}_\theta^{*k} = \frac{\partial}{\partial \theta_j} A(q,N)(F)(\theta) = \\ \sum_{q-N+1 \leq |\mathbf{i}| \leq q} (-1)^{q-|\mathbf{i}|} \cdot \begin{pmatrix} N-1 \\ q-|\mathbf{i}| \end{pmatrix} \frac{\partial}{\partial \theta_j} \left(U_{i_1} \otimes \cdots \otimes U_j' \otimes \cdots \otimes U_{i_N} \right) (F)(\theta). \end{gathered} \tag{5.24}$$

Since U_j' can be found analytically, we can determine an analytic gradient by finding $\frac{\partial}{\partial \theta_j} A(q,N)(F)(\theta)$ in each of the N stochastic dimensions. This defines the derivative of the coarse-scale interpolated response, $\tilde{F^*}$. This can then be used to calculate $\nabla \log \tilde{\pi}^*$ directly, rather than by finite differences.

5.3.1. Analysis of the Interpolated Preconditioned Coarse-Gradient Langevin Algorithm

As in the analysis of the preconditioned coarse-gradient Langevin algorithm, it can be shown that under mild conditions the modified Markov chain is irreducible and aperiodic. More precisely, denote

$$\begin{aligned} &\mathcal{E} = \left\{ k;\ \pi(k) > 0 \right\}, \\ &\tilde{\mathcal{E}}^* = \left\{ k;\ \tilde{\pi}^*(k) > 0 \right\}, \\ &\mathcal{D} = \left\{ k;\ \tilde{q}^*(k|k_n) > 0 \text{ for any } k_n \in \mathcal{E} \right\}. \end{aligned} \tag{5.25}$$

The set $\mathcal{E}$ is the support of the posterior (target) distribution $\pi(k)$. $\mathcal{E}$ contains all the fields k which have a positive probability of being accepted as a sample. Similarly, $\tilde{\mathcal{E}}^*$ is the support of the interpolated coarse-scale distribution $\tilde{\pi}^*(k)$, which contains all the k acceptable by the interpolated coarse-scale test. $\mathcal{D}$ is the set of all possible proposals which can be generated by the Langevin distribution $\tilde{q}^*(k|k_n)$. To make the interpolated coarse-gradient Langevin MCMC methods sample properly, the conditions $\mathcal{E} \subseteq \mathcal{D}$ and $\mathcal{E} \subseteq \tilde{\mathcal{E}}^*$ must hold (up to a zero measure set) simultaneously. In this case, we can show from the same argument as with Algorithm II that the chain $\{k_n\}$ generated by Algorithm IV is strongly π-irreducible.

CHAPTER VI

NUMERICAL APPLICATIONS FOR UNCERTAINTY QUANTIFICATION

In this chapter, we present numerical results for the uncertainty quantification problems introduced in the previous chapter. We first apply these methods to multiphase immiscible flow, where we will attempt to sample the permeability field given fractional flow measurements. We present a detailed set of results using both fine and coarse models, specifically using Algorithms I, II, and III. We then provide a short set of results which provide evidence that collocation methods, and specifically Algorithm IV, can be used in the context of uncertainty quantification. We next apply the uncertainty quantification methods to Richards' equation, where we will sample the saturated conductivity given some average flux measurements. We provide a short set of results using fine and coarse models, then again show collocation methods can be used. For each equation we use a finite volume method and the MsFVEM discussed in Section 2.3 for the fine-scale and coarse-scale numerical solutions, respectively.

6.1. Two-Phase Immiscible Flow

We consider the two-phase immiscible flow equations introduced in Section 2.1.1. The problem under consideration consists of sampling the permeability field given fractional flow measurements. Fractional flow, $F(t)$ (denoted simply by F in further discussion), is defined as the fraction of oil in the produced fluid and is given by q_o/q_t, where $q_t = q_o + q_w$, with q_o and q_w the flow rates of oil and water at the production edge of the model. More specifically,

$$F(t) = 1 - \frac{\int_{\partial\Omega^{out}} f(S) v_n dl}{\int_{\partial\Omega^{out}} v_n dl},$$

where $\partial\Omega^{out}$ is the outflow boundary and v_n is the normal velocity field. The fractional flow curves will be plotted against pore volumes injected (PVI). PVI is comparable to a dimensionless time and is defined as $\frac{q_t \cdot t}{V_p}$, where t is dimensional time and V_p is the total pore volume of the system.

Typically, the prior information about the permeability field consists of its covariance matrix and the values of the permeability at some sparse locations. Since the fractional flow is an integrated response, the map from the permeability field to the fractional flow is not one-to-one. Hence this problem is ill-posed in the sense that there exist many different permeability realizations for the given production data.

Numerical results for sampling permeability fields using two-point geostatistics are presented. Using the KLE, we can represent the high dimensional permeability field by a small number of parameters. Furthermore, the static data (the values of permeability fields at some sparse locations) can be easily incorporated into the KLE to further reduce the dimension of the parameter space. Imposing the values of the permeability at some locations restricts the parameter space to a subspace (hyperplane). Numerical results are presented for both single-phase and two-phase flows. In all the simulations, we show that the gradients of the target distribution computed using coarse-scale simulations provide accurate approximations of the actual fine-scale gradients. Furthermore, we present the uncertainty assessment of the production data based on sampled permeability fields. Our numerical results show that the uncertainty spread is much larger if no dynamic data information is used. However, the uncertainty spread decreases if more information is incorporated into the simulations.

6.1.1. Coarse vs Fine

In this section we discuss the implementation details of Langevin MCMC method for two-phase immiscible flow and present some representative numerical results. Suppose the permeability field $k(x)$ is defined on the unit square $\Omega = [0,1]^2$. We discretize the domain Ω by a rectangular mesh, hence the permeability field k is represented by a matrix (i.e., k is a high dimensional vector). As for the boundary conditions, we have tested various boundary conditions and observed similar performance for the Langevin MCMC method. In our numerical experiments we will assume $p = 1$ and $S = 1$ on $x = 0$, $p = 0$ on $x = 1$, and no flow boundary conditions on the lateral boundaries $y = 0$ and $y = 1$. We have chosen this type of boundary conditions because they provide a large deviation between coarse-scale and fine-scale simulations for the permeability fields considered. We will consider both single-phase and two-phase flow displacements.

In the simulations, we first generate a reference permeability field using the full KLE of $Y(x,\omega)$. We choose normal covariance and correlation lengths of $L_1 = L_2 = 0.2$ in (2.16) to obtain the corresponding fractional flows. To represent the discrete permeability fields from the prior (unconditioned) distribution, we keep 20 terms in the KLE, which captures more than 95% of the energy of $Y(x,\omega)$. We assume that the permeability field is known at 9 distinct points. This condition is imposed by setting

$$\sum_{k=1}^{20} \sqrt{\lambda_k}\theta_k\phi_k(x_j) = \alpha_j, \tag{6.1}$$

where α_j $(j = 1, \ldots, 9)$ are prescribed constants. For simplicity, we set $\alpha_j = 0$ for all $j = 1, \ldots, 9$. In the simulations we propose eleven θ_i and calculate the rest of the θ_i by solving the linear system (6.1). In all the simulations, we test $5,000$ samples. Because the direct Langevin MCMC simulations are very expensive, we only select a

61×61 fine-scale model for single-phase flow and a 37×37 fine-scale model for two-phase flow. Here 61 and 37 refer to the number of nodes in each direction, since we use a finite element based approach. Typically, we consider 6 or 10 times coarsening in each direction. In all the simulations, the gradients of the target distribution are computed using finite-difference differentiation rule. The time step size $\Delta\tau$ of the Langevin algorithm is denoted by δ. Based on the KLE, the parameter space of the target distribution $\pi(k)$ will change from k to θ in the numerical simulations, and the Langevin algorithms can be easily rewritten in terms of θ.

6.1.1.1. Single-Phase Flow

Our first set of numerical results are for single-phase flows. First, we present a comparison between the fine-scale response surface π and the coarse-scale response surface π^* defined by (5.3) and (5.9), respectively. Because both π and π^* are scalar functions of 11 parameters, we plot the restriction of them to a 2-D hyperplane by fixing the values of the last nine θ values in the reduced KLE. In Figure 6.1, π^* (left figure) and π (right figure) are depicted on such a 2-D hyperplane. It is clear from these figures that the overall agreement between the fine- and coarse-scale response surfaces is good. This is partly because the fractional flow is an integrated response. However, we notice that the fine-scale response surface π has more local features and varies on smaller scales compared to π^*.

In Figure 6.2, we compare the acceptance rates of the Algorithms I, II and III with different coarse-scale precision σ_c. The acceptance rate is defined as the ratio between the number of accepted permeability samples and the number of fine-scale acceptance-rejection tests. Since Algorithm I does not depend on the coarse-scale precision, its acceptance rate is the same for different σ_c. As we can see from the figure, Algorithm II has higher acceptance rates than Algorithm III. The gain in the

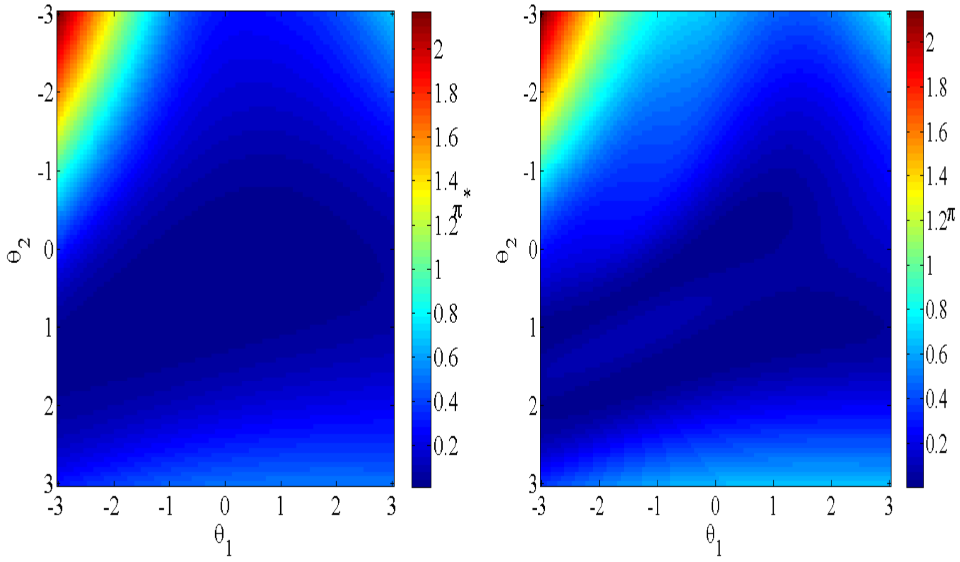


Fig. 6.1. Left: Coarse-scale response surface π^* (defined by (5.9)) restricted to a 2-D hyperplane. Right: Fine-scale response surface π (defined by (5.3)) restricted to the same 2-D hyperplane.

acceptance rates is due to the Step 2 of Algorithm II, which filters unlikely acceptable proposals. To compare the effect of different degrees of coarsening, we plot in Figure 6.2 the acceptance rate of Algorithm II using both 7×7 coarse models and 11×11 coarse models. Since 11×11 coarse models are more accurate, its acceptance rate is higher. In Figure 6.3, we present the numerical results where larger time step δ is used in the Langevin algorithms. Comparing with Figure 6.2, we find that that the acceptance rates for all the three methods decrease as δ increases. In all the numerical results, Algorithm I, which uses the fine-scale Langevin method (5.7), gives a slightly higher acceptance rate than both Algorithms II and III. However, Algorithm I is more expensive than Algorithms II and III since it uses the fine-scale gradients in computing the Langevin proposals. In Figure 6.4, we compare the natural log of CPU time for the different Langevin methods. From the left plot we see that Algorithm I is several times more expensive than Algorithms II and III. In the middle and right plots, we

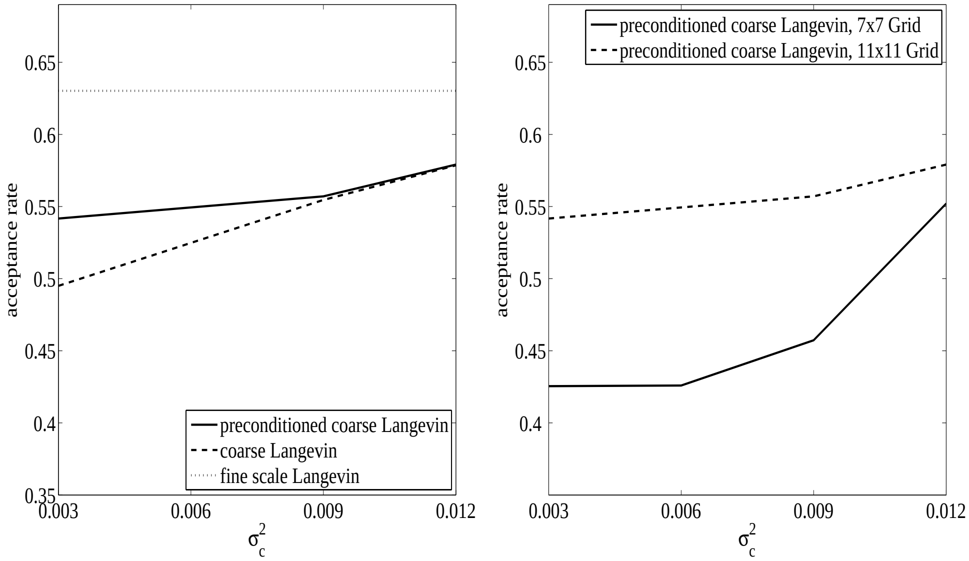


Fig. 6.2. Acceptance rate comparison between Algorithms I, II and III; $\delta = 0.05$, $\sigma_f^2 = 0.003$. In the left plot, the coarse-grid 11×11 is used in the simulation.

compare Algorithms II and III when a different coarse-model and a different time step size δ are used, respectively. We observe that Algorithm II is slightly faster than Algorithm III, which does not use preconditioning.

In all the previous numerical simulations, we choose the fine-scale error precision $\sigma_f^2 = 0.003$. The scaling of the error precision depends on the norm used in (5.3). If one choses σ_f to be very large, then the precision is very low, and consequently, most proposals will be accepted in the direct Langevin algorithm as well as in the coarse-gradient Langevin algorithms. In this case, the acceptance rate of the coarse-gradient Langevin algorithms is still similar to the acceptance rate of the direct Langevin algorithms. Consequently, the speed-up will remain the same. For very large σ_f, the preconditioning step in Algorithm II may not help to improve the acceptance rate, since most proposals will pass the preconditioning step.

Next we compare the fractional flow errors for Algorithm II and Algorithm I in Figure 6.5. Our objective is two-fold. First, we would like to compare the convergence

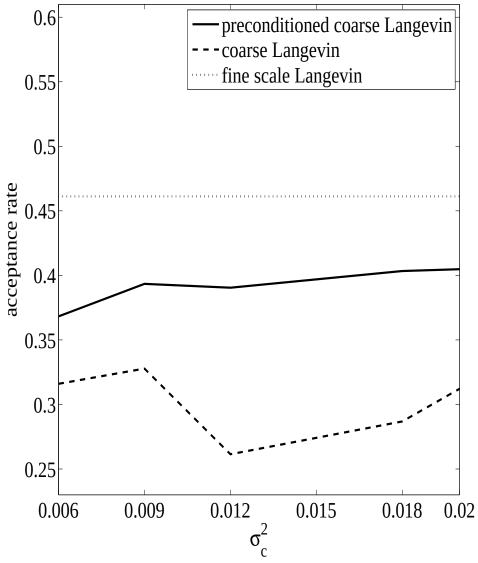


Fig. 6.3. Acceptance rate comparison for Algorithms I, II and III for single-phase flow, $\delta = 0.1$, $\sigma_f^2 = 0.003$.

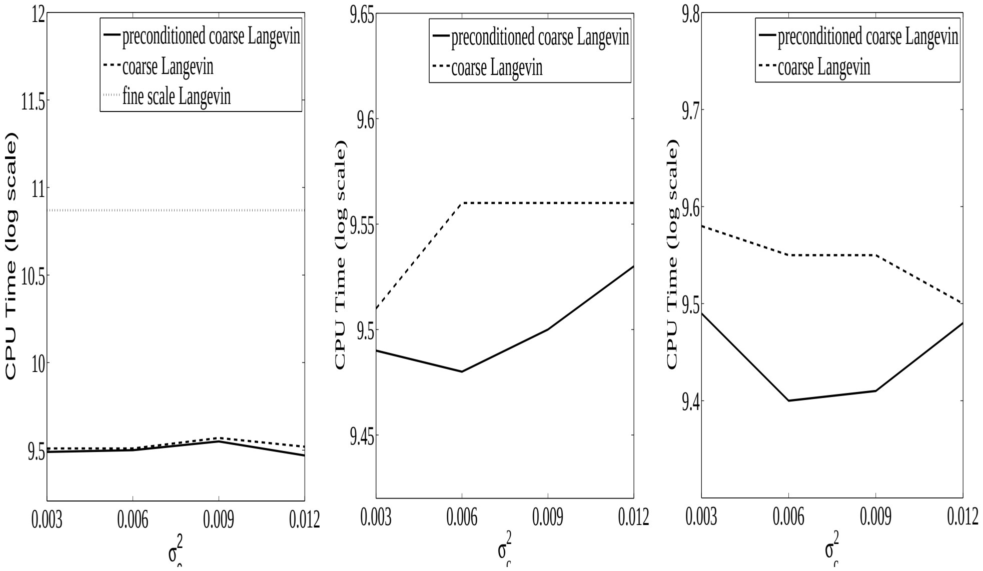


Fig. 6.4. Natural log of CPU time (seconds) comparison for the different Langevin algorithms. Left: $\sigma_f^2 = 0.003$, $\delta = 0.05$, 11×11 coarse-grid. Middle: $\sigma_f^2 = 0.003$, $\delta = 0.05$, 7×7 coarse-grid. Right: $\sigma_f^2 = 0.003$, $\delta = 0.1$, 11×11 coarse-grid.

rates of Algorithm II with that of Algorithm I. Second, we would like to show that the sampled permeability fields give nearly the same fractional flow response as the reference fractional flow data. The left plot represents the error between the accepted fractional flows and the reference fractional flow for both algorithms. It is clear from this figure that both methods converge to the steady state within the same number of iterations. The formal convergence diagnosis has been performed using multiple chains method base convergence diagnosis ([21]). We wish to only compare the chains obtained using both methods to each other, and thus we restrict ourselves to only showing errors vs. the number of iterations. We note that the convergence diagnostics have nothing to do with the rate of convergence. The rate of convergence depends on the second largest eigenvalue of the transition matrix of the Markov chain. For the complex chains, it is not easy to find these eigenvalues. Thus, for simplicity, we instead choose graphical analysis of the samples as in the left plot of Figure 6.5. In the right plot, the fractional flows for sampled realizations are plotted (dotted lines). The fractional flows of the sampled realizations are very close to the reference fractional flow. This is because the error precision is taken to be small ($\sigma_f^2 = 0.003$) in the target distribution.

In Figure 6.6, some permeability realizations sampled from the posterior distribution are plotted. In particular, we plot realizations which do not look very similar to each other and represent the uncertainty range observed in our simulations. We observe that the samples capture some of the features of the reference permeability field. Note that all these permeability fields give nearly the same fractional flows as the reference fractional flow, so they are all eligible samples.

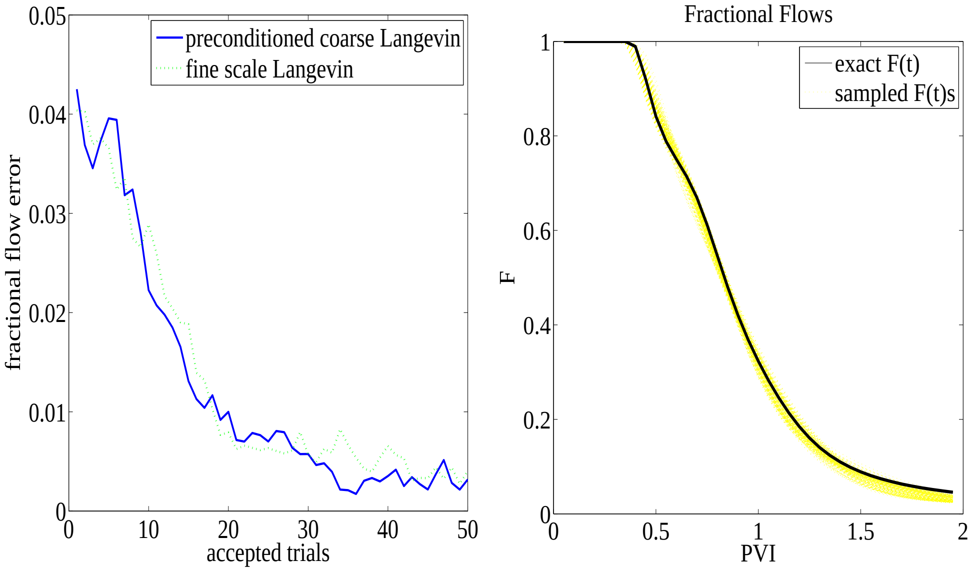


Fig. 6.5. Left: The fractional flow errors for Algorithm I and Algorithm II. Right: The fractional flows of sampled realizations and the reference fractional flow. In these numerical tests, $\delta = 0.05$, $\sigma_f^2 = 0.003$ and 11×11 coarse-scale model is used.

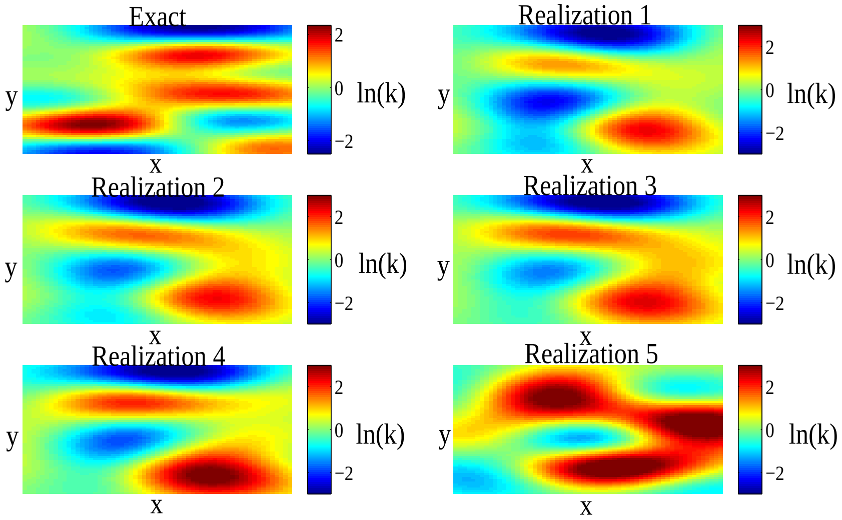


Fig. 6.6. Samples of the permeability realizations. Realizations are selected to represent the uncertainty range in the simulations.

6.1.1.2. Two-Phase Flow

Next we consider two-phase flow simulations. Because two-phase flow simulations are computationally intensive, we restrict our computations to the fine grid 37×37, and the coarse-grid 7×7. In all simulations, we set $\sigma_f^2 = 0.003$ and $\delta = 0.05$. Figure 6.7 shows the response surfaces π and π^* restricted to a two dimensional hyperplane in θ. As in the case of the single-phase flow, π^* approximates π very well in large scales, though π has more variations on small scales. In the left plot of Figure 6.8, acceptance rates for Algorithms I, II and III are compared. As we can see from this figure, the acceptance rates of Algorithm II is very similar to that of Algorithm I. Without preconditioning, Algorithm III has lower acceptance rates than Algorithm II. Comparing the natural log of the CPU time in the right plot of Figure 6.8, we observe that Algorithm II is an order of magnitude faster than Algorithm I. If the resolution of the fine-grid is increased, one can expect an even higher acceleration rate by Algorithm II. In Table 6.1, we compare the acceptance rates and CPU times between Algorithms I, II and III for different coarse grid resolutions. In these numerical results, we have chosen $\sigma_f^2 = 0.003$, though similar results are observed for other values of σ_f^2. We also observe that Algorithm II has higher acceptance rate and lower CPU time compared to Algorithm III for all coarse grid resolutions. For this reason, we will not consider Algorithm III for the remainder our numerical tests.

In Figure 6.9, the fractional flow errors and fractional flows are plotted. In the two-phase flow case, we observe that Algorithm I converges slightly faster than Algorithm II. Finally, in Figure 6.10, we plot some permeability realizations. We selected the samples which do not look very similar to each other and represent the uncertainty range observed in the simulations. This figure illustrates that the sampled permeability realizations capture the main features of the reference permeability field.

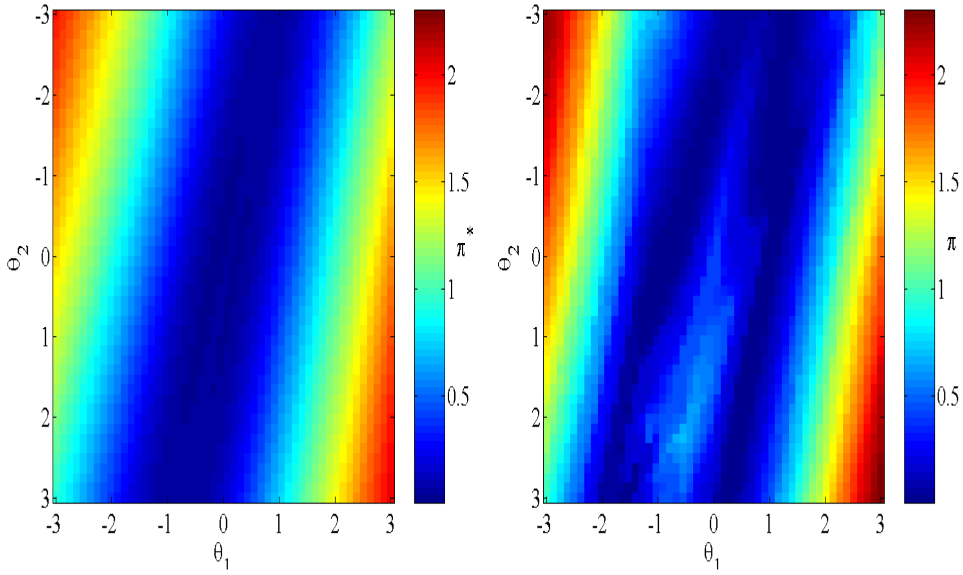


Fig. 6.7. Left: Coarse-scale response surface π^* restricted to 2-D hyperplane. Right: Fine-scale response surface π restricted to the same 2-D hyperplane

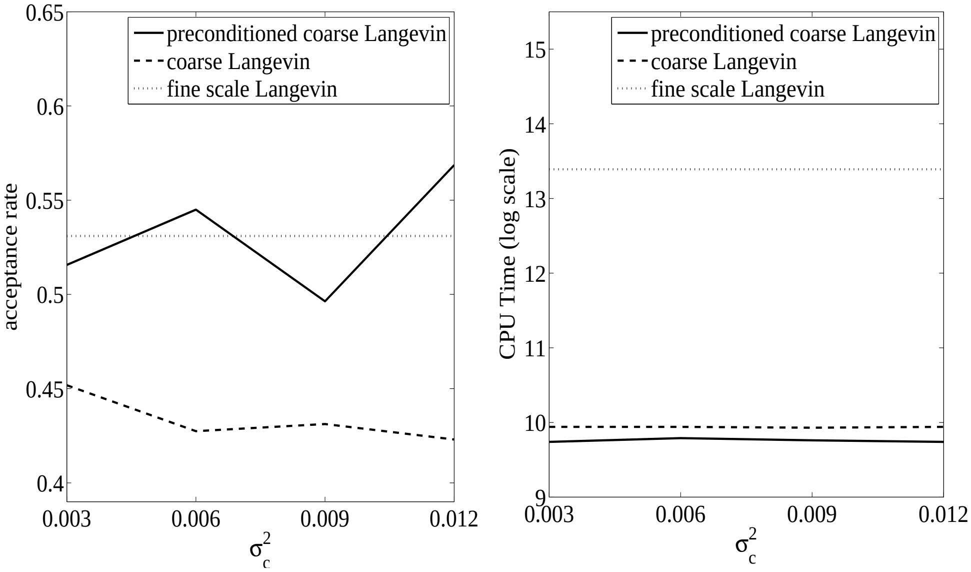


Fig. 6.8. Left: Acceptance rate comparison for Algorithms I, II and III. Right: Natural log of CPU times (seconds) for Algorithms I, II and III. Each plot is for two-phase flow with $\delta = 0.05$, $\sigma_f^2 = 0.003$ and a 7×7 coarse-grid.

Table 6.1. Comparison of Algorithm I, II and III for different coarse-grid resolutions in two-phase flow simulations. $\sigma_f^2 = \sigma_c^2 = 0.003$, $\delta = 0.05$.

coarse grid	coarse		coarse preconditioned		direct	
	accept. rate	CPU	accept. rate	CPU	accept. rate	CPU
4x4	0.47	8527	0.55	7036	0.53	655895
7x7	0.45	21859	0.52	17051	0.53	655895
10x10	0.46	70964	0.57	48653	0.53	655895

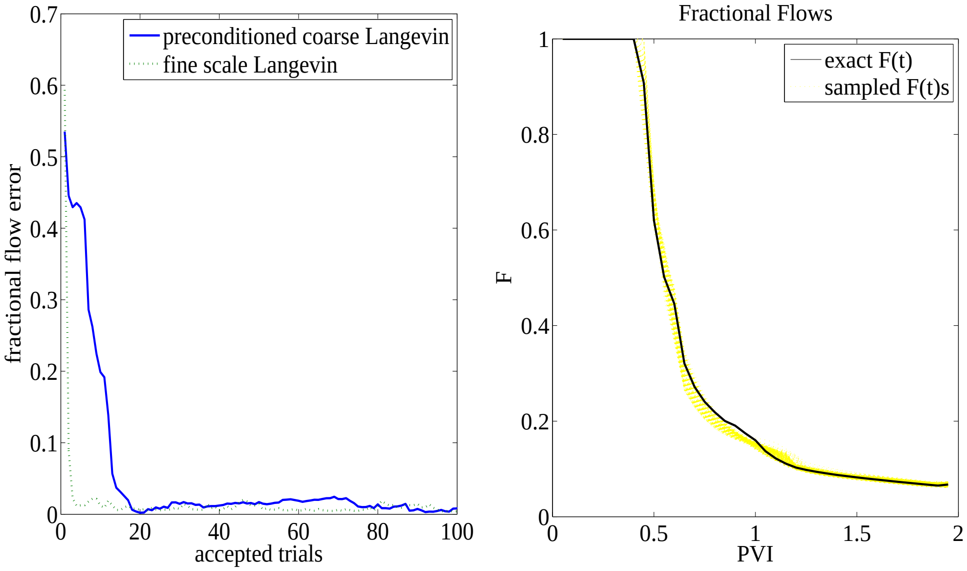


Fig. 6.9. Left: The fractional flow errors for Algorithm I compared with Algorithm II. Right: The fractional flows of sampled realizations and the reference fractional flow. In these numerical tests, $\delta = 0.05$, $\sigma_f^2 = 0.003$ and 7×7 coarse-scale model is used.

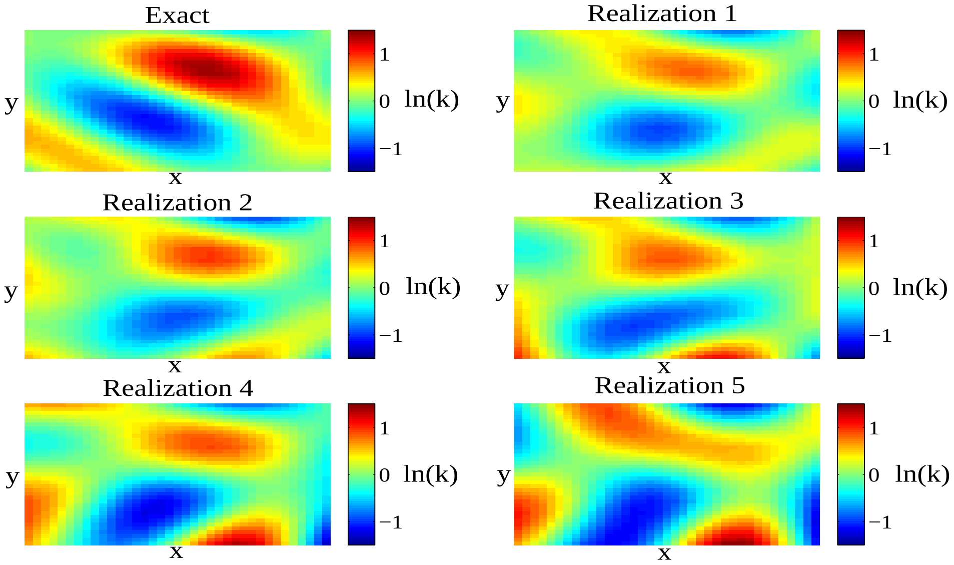


Fig. 6.10. Samples of the permeability realizations. Realizations are selected to represent the uncertainty range in the simulations.

6.1.1.3. Theoretical Computational Costs

Next, we compare the theoretical computational costs of the three Langevin algorithms for the two-phase flow problem. Denote t_f and t_c as the CPU time to solve the PDE system (2.1)-(2.3) on the fine- and coarse-grid respectively. Suppose D is the dimension of the parameter space of the permeability field k, and N is the number of proposals that are tested in all three Langevin algorithms. For each new Y, Algorithm I needs to compute the target distribution $\pi(Y)$ and its gradient $\nabla\pi(Y)$ on the fine-grid. If the gradient is computed by the forward difference scheme, then the PDE system (2.1)-(2.3) needs to be solved on the fine-grid $(D+1)$ times. Therefore, the total computational cost of Algorithm I is $N(D+1)t_f$. For Algorithm III, the gradient of the distribution is computed on the coarse-grid. However, the acceptance test is calculated on the fine-grid for each proposal. Thus, its computational cost is $N(Dt_c + t_f)$. In Algorithm II, the gradient of the distribution is also computed on the coarse-grid, and each new sample is first tested by the coarse-scale distribution. If it passes the coarse-grid acceptance test, then the proposal will be further tested by the fine-scale distribution. Suppose M proposals (out of N) pass the coarse-scale test, then the total computational cost of Algorithm II is $N(D+1)t_c + Mt_f$. Thus, Algorithm II is $\frac{N(D+1)t_f}{N(D+1)t_c+Mt_f}$ times faster than Algorithm I, and $\frac{N(Dt_c+t_f)}{N(D+1)t_c+Mt_f}$ times faster than Algorithm III. In our computations, D is of order of ten because we represent the permeability field by its truncated Karhunen-Loève expansion. If the fine-scale model is scaled up 6 times in each direction, as we did in the numerical experiment, then the coarse-scale model is approximately 36 times faster than the fine-scale model. Indeed, at each time step solving the pressure equation on the coarse grid is approximately 36 times faster than on the fine grid. The same is true for the saturation equation since it is also solved on the coarse grid and with larger time steps. Moreover, in Algorithm II , only a portion of the N proposals can pass the coarse-scale test, where N is usually two times larger than M. Using these estimates, we expect that the CPU time of Algorithm II should be an order of magnitude lower than that of Algorithm I. We indeed observed a similar speedup in our computations, as demonstrated by Figure 6.8.

Note that one can use simple random walk samplers, instead of Langevin sampling, in Algorithm I. We have observed in our numerical experiments that the acceptance rate of the random walk sampler is several times smaller than that of Langevin algorithms. This is not surprising because Langevin algorithms use the gradient information of the target distribution and are problem adapted. One can also use single-phase flow upscaling (as in [9]) in the preconditioning step as it is done in [13]. In general, we have found the multiscale methods to be more accurate for coarse-scale simulations and they can be further used for efficient and robust fine-scale simulations. We would like to mention some other relevant work, [16, 33] where streamline simulations are used for analytical approximation of fractional flows within multistage MCMC. In [33], the authors also compare Langevin algorithms to randomized maximum likelihood method.

6.1.1.4. Predictions

Finally, we present the results demonstrating the uncertainties in the predictions. In our simulations, we use the information of the dynamic data in various time spans. In Figure 6.11, various prediction results are plotted based on information of the dynamic data on $[0, 0.8]$ PVI time (left figure), on $[0, 0.4]$ PVI time (middle figure), and when no dynamic data information is used (right figure). These results are obtained by sampling 50 realizations from the posterior distribution. As we observe from the figure, the uncertainty spread is the largest if no dynamic data information is used. However, the uncertainty spread decreases, as expected, if more information is incorporated into the simulations. In particular, using the dynamic data information up to 0.8 PVI allows us to obtain accurate predictions and reduce the uncertainties. These results allow us to assess the uncertainties in the predictions.

6.1.2. Sparse Grid Collocation

In this section we compare the numerical results using the Algorithm I, Algorithm II, and Algorithm IV. Instead of repeating the same arguments as the previous section, we will focus on a more limited set of results. We consider Algorithm IV with two different interpolated values. First, we consider interpolation of the coarse-scale target distribution π^*, where π^* is found using the MsFVEM as previously. Next, we consider interpolation of the coarse-scale fractional flow, F_k^*, which allows us to obtain an approximate coarse-scale target distribution using (5.9). Lastly, we show the collocation methods can be used for exponential covariance in the KLE. Our aim

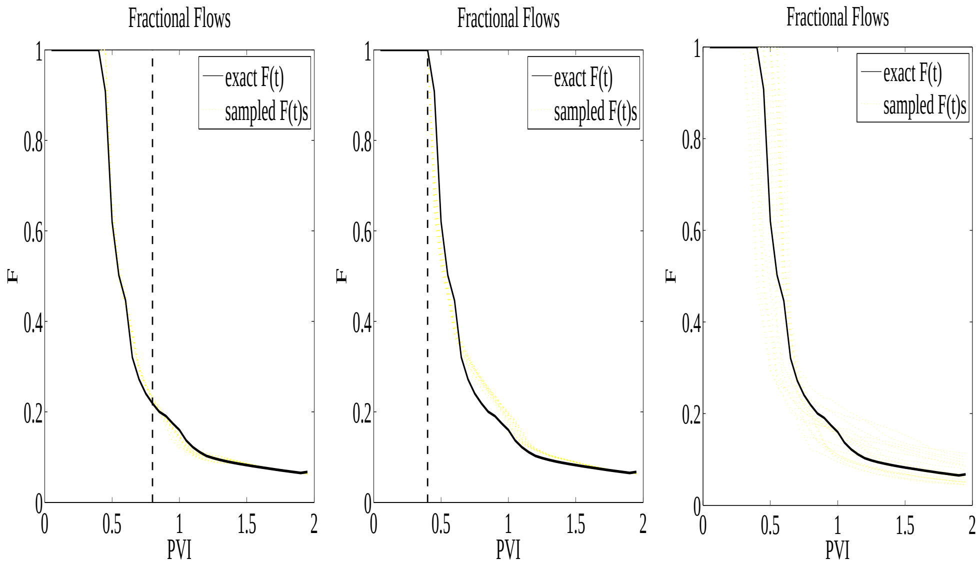


Fig. 6.11. Prediction results using the information about the dynamic data on various time spans. Left: The dynamic data information on $[0, 0.8]$ PVI is used; Middle: The dynamic data information on $[0, 0.4]$ PVI is used; Right: No dynamic data information is used.

is to show that Algorithm IV can be used as an inexpensive alternative to Algorithm II.

6.1.2.1. Interpolation of the Target Distribution

We begin by considering a numerical setup similar to the previous section, where we first generate a reference permeability field using the full KLE of $Y(x, \omega)$ and obtain the corresponding fractional flows. In the KLE we use normal covariance and correlation lengths $L_1 = 0.5$ and $L_2 = 0.1$. We keep 20 terms in the KLE, and assume that the permeability field is known at 11 distinct points. This reduces the dimension of our stochastic space to 9 instead of the 11 dimensions we used previously. This is simply to ease some computations in the collocation methods. We restrict ourselves to the hypercube $[-2.5, 2.5]^9$ in stochastic space. All the simulations are based on $5,000$

samples and a 61×61 fine-scale model is selected. We consider 6 times coarsening in each direction (thus an 11×11 coarse-scale model). The results are based only on two-phase flow. In this section, we consider interpolation of the coarse-scale response surface π^* rather than interpolation of the coarse-scale fractional flow. An advantage to interpolating π^* is that we only have one single value per interpolation node. If we were to interpolate the fractional flow, which is a function of PVI, we would have to interpolate for each value of PVI. In our case, this contains 40 values. Thus we would need to perform 40 interpolation operations as opposed to one. A slight disadvantage to using π^* is that we will not be able to use the analytical formula for the interpolated gradient, given by (5.24). We instead use a gradient based on finite differences, as in the previous section. We present results when using interpolation of the fractional flow in Section 6.1.2.2.

First, we present a comparison between the coarse-scale response surface π^* and the interpolated coarse-scale response surfaces $\tilde{\pi}^*$. We have already noted that the coarse-scale response surface approximates the general properties of the fine-scale response surface well while losing some small fine-scale features, thus we do not compare them again. We consider interpolated surfaces for various levels of interpolation. We define k^{th} order Smolyak interpolation as $A(N+k, N)$ in (3.19). In particular, for $N = 9$ stochastic dimensions, 1^{st} order results in 19 nodes, 2^{nd} order results in 181 nodes, and 3^{rd} order results in $1,177$ nodes. Because both π and π^* are scalar functions of 9 parameters, we plot the restriction of them to a 2-D hyperplane by fixing the values of the last seven θ in the KLE. In Figure 6.12, the coarse-scale response surface π^* is the upper left figure. Interpolated coarse-scale surfaces are in the upper right, lower left, and lower right for 1^{st}, 2^{nd}, and 3^{rd} order Smolyak interpolation, respectively. We note that each interpolation surface matches the overall behavior of π^*. It appears the surface using 2^{nd} order interpolation matches slightly better than

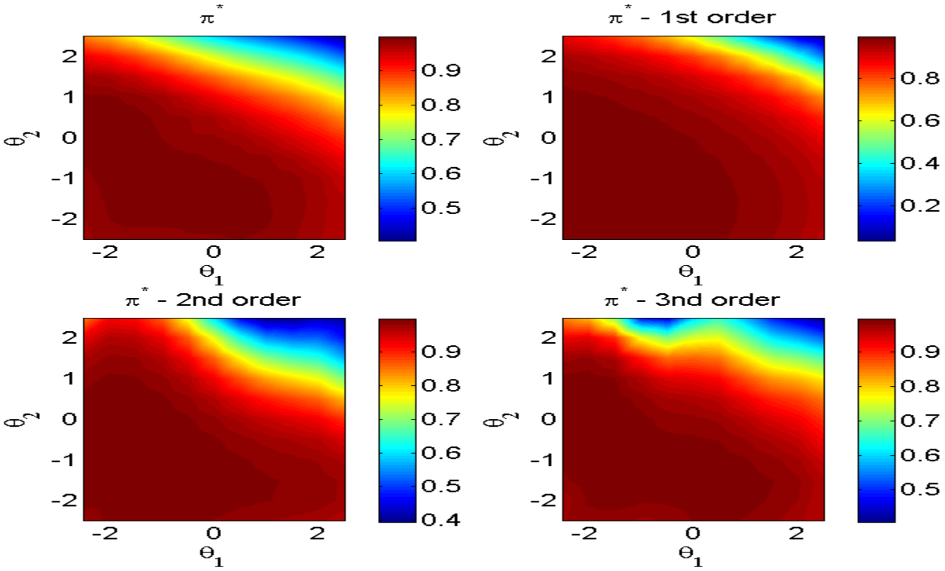


Fig. 6.12. Coarse-scale response surface π^* and interpolated coarse-scale response surfaces $\tilde{\pi}^*$ restricted to a 2-D hyperplane.

the surface using 1^{st} order. The 3^{rd} order is nearly exact in some places, but contains what appears to be some extraneous features. This is likely due to the oscillations that would be observed from a very high dimensional polynomial. Due to the fact that the surface using 1^{st} order interpolation matches well, we will consider 1^{st} order Smolyak interpolation, unless otherwise noted. Note that another reason for choosing low order interpolation is that if we were sampling in much higher dimensions as in Section 4.1.2, then we would not be able to use interpolation higher than 1^{st} order anyway.

In the left plot of Figure 6.13, we compare the acceptance rates of Algorithms I, II and IV with different coarse-scale precisions σ_c. Recall the acceptance rate is defined as the ratio between the number of accepted permeability samples and the number of fine-scale acceptance-rejection tests. Since Algorithm I does not depend on the coarse-scale precision, its acceptance rate is the same for different σ_c. As we can see from the figure, Algorithms II and IV have a slightly higher acceptance

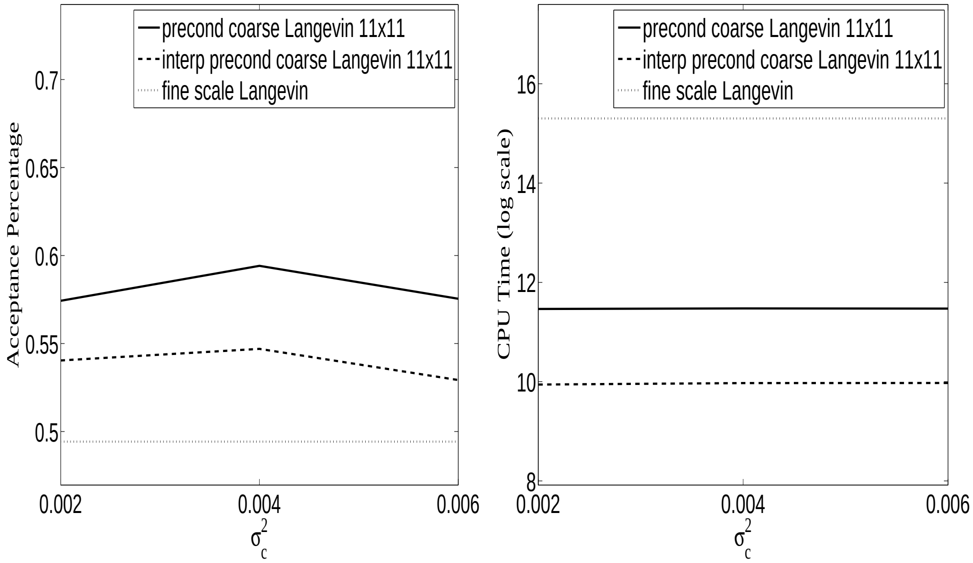


Fig. 6.13. Left: Acceptance rates for Algorithms I, II and IV. Right: Natural log of CPU time (seconds) for Algorithms I, II and IV. In each plot $\delta = 0.05$ and $\sigma_f^2 = 0.002$.

rate as compared to Algorithm I. This is due to the Step 2 of Algorithm II and IV, which filters unlikely acceptable proposals. We also note that Algorithm II has a higher acceptance rate than Algorithm IV. This is expected, as the gradients in the Langevin proposal for Algorithm II will be more precise than the interpolated gradients in Algorithm IV. In the right plot of Figure 6.13, we compare the CPU time (on a log scale) for the different Langevin methods. We see that Algorithm IV is several times faster than Algorithm II, and significantly faster than Algorithm I.

Next we compare the fractional flow errors for Algorithm IV and Algorithm I in Figure 6.14. Our objective is two-fold. First, we would like to compare the convergence rates of Algorithm IV with that of Algorithm I. Second, we would like to show that the sampled permeability fields give nearly the same fractional flow response as the reference fractional flow data. We see from the left plot of Figure 6.14 that both methods converge to the steady state within the same number of

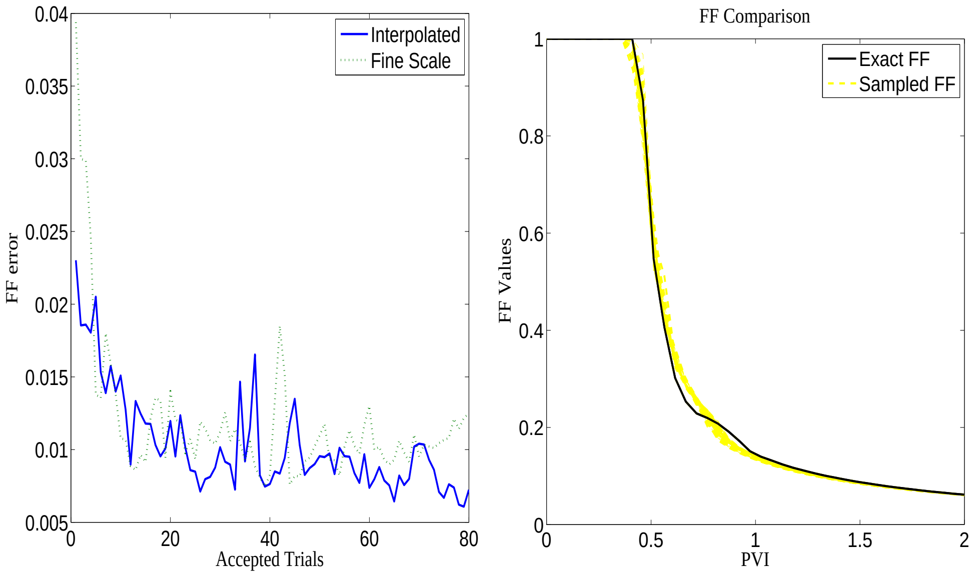


Fig. 6.14. Left: The fractional flow errors for Algorithm I compared with Algorithm IV. Right: The fractional flows of sampled realizations and the reference fractional flow. In these numerical tests, $\delta = 0.05$, $\sigma_f^2 = 0.002$ and an 11×11 coarse-scale model is used.

iterations. In the right plot, the fractional flows for sampled realizations are plotted (dotted lines). The fractional flows of the sampled realizations are very close to the reference fractional flow because the error precision is taken to be small ($\sigma_f^2 = 0.002$) in the target distribution.

In Figure 6.15, some permeability realizations sampled from the posterior distribution are plotted. The upper left plot in Figure 6.15 represents the reference permeability. The remaining plots are three different permeabilities that were accepted using Algorithm IV. We observe that the samples capture some features of the reference permeability field. Note that all these permeability fields give nearly the same fractional flows as the reference fractional flow, so they are all eligible samples.

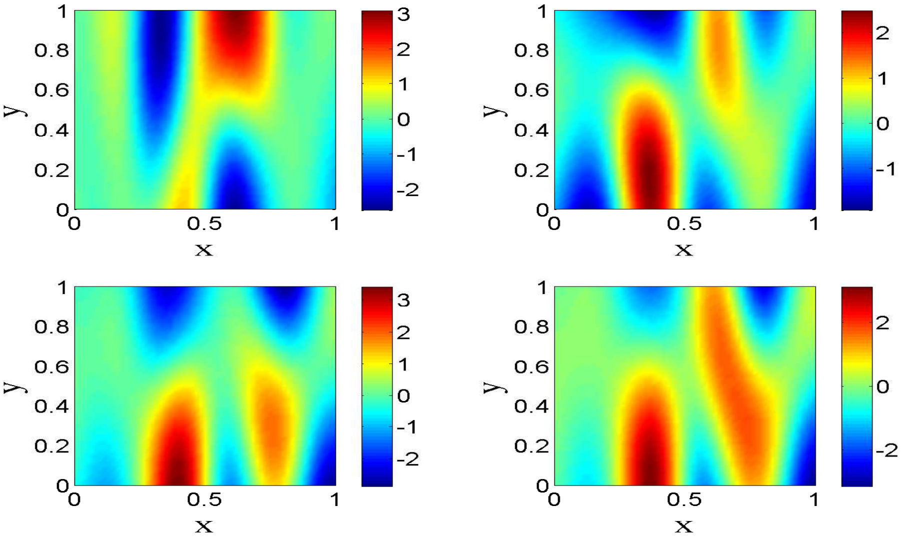


Fig. 6.15. Upper left plot is the reference permeability. The other three plots are examples of accepted permeability realizations.

6.1.2.2. Interpolation of the Fractional Flows

We now consider interpolation of the coarse-scale fractional flow values, F_k^*, rather than interpolation of the coarse-scale target distribution π^*. Since the fractional flow is a function in time, we must now interpolate at each discrete time step. We consider a similar model as the in the previous section. We again have 20 terms in the KLE. We use normal covariance with correlation lengths $L_1 = 0.5$ and $L_2 = 0.1$, and we assume that the permeability field is known at 11 distinct points, resulting in 9 stochastic dimensions. We test $5,000$ samples, a 61×61 fine-scale model, and an 11×11 coarse-scale model. The results are based only on two-phase flow. Since we consider interpolation of the coarse scale fractional flow F^*, we use an analytical formula for the gradient of the target distribution based on (5.24).

First, we present a comparison between the coarse-scale response surface π^* and the interpolated coarse-scale response surfaces $\tilde{\pi}^*$. We again consider interpolated

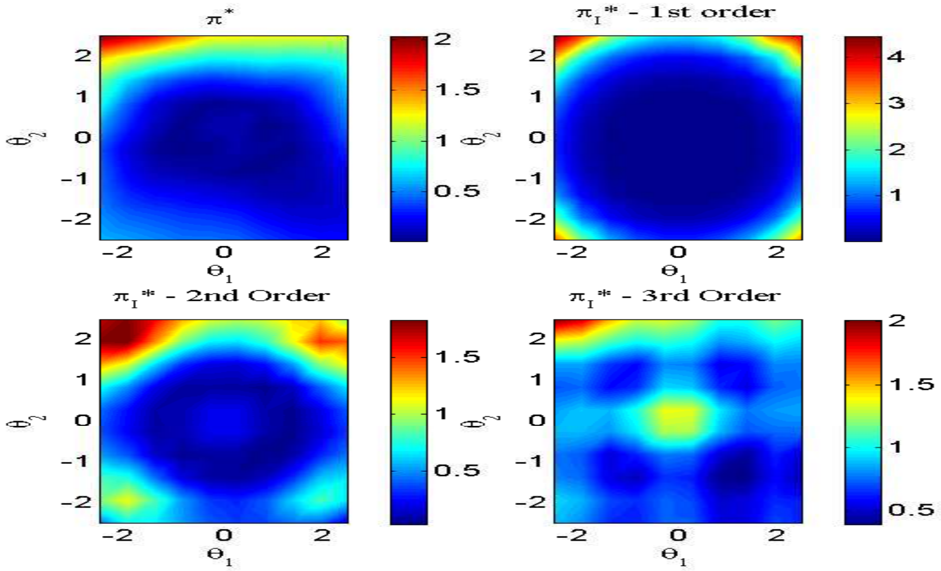


Fig. 6.16. Coarse-scale response surface π^* and interpolated coarse-scale response surfaces $\tilde{\pi}^*$ restricted to a 2-D hyperplane.

surfaces for various levels of interpolation. In Figure 6.16, the coarse-scale response surface π^* is the upper left figure. Interpolated coarse-scale surfaces, π^* are in the upper right, lower left, and lower right for 1^{st}, 2^{nd}, and 3^{rd} order Smolyak interpolation, respectively. Starting at 1^{st} order interpolation, we have a good overall agreement with the coarse surface. However, it appears the 1^{st} order response surface does not capture some local effects, Going to 2^{nd} order interpolation, we appear to capture more local effects, but we find some oscillations, likely due to the higher order polynomial found under 2^{nd} order interpolation. Furthermore, 3^{rd} order interpolation does not reflect the overall behavior of the response surfaces due to numerous oscillations. Due to this fact, for the remainder of our discussion, we consider 1^{st} order Smolyak interpolation unless otherwise noted.

In Figure 6.17, we compare the acceptance rates and CPU time for Algorithms I, II and IV with different coarse-scale precisions σ_c. Algorithms II and IV have a slightly lower acceptance rate as compared to Algorithm I. This is likely due to

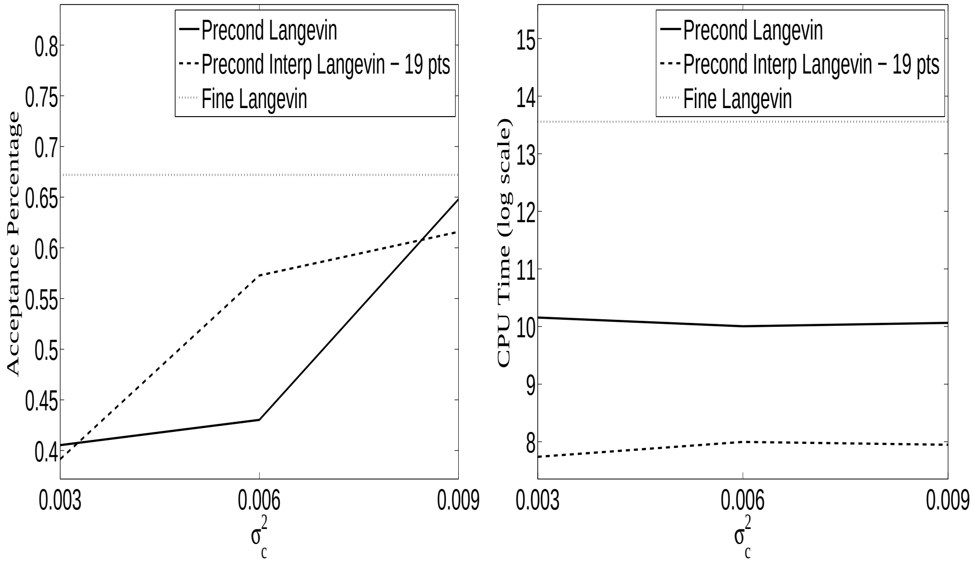


Fig. 6.17. Left: Acceptance rate comparison between the Langevin algorithms. Right: Comparison of the natural log of CPU time (seconds) for the different Langevin algorithms. In each plot $\delta = 0.05$ and $\sigma_f^2 = 0.003$.

the more accurate fine-scale gradients in Algorithm I. In the right plot we see that Algorithm IV is several times faster than Algorithm II, and significantly faster than Algorithm I.

We now consider the fractional flow errors and fractional flow comparisons for Algorithm IV and Algorithm I in Figure 6.18. We see from the left plot that both methods converge to the steady state within the same number of iterations. In the right plot, the fractional flows of the sampled realizations are very close to the reference fractional flow because the error precision is taken to be small ($\sigma_f^2 = 0.003$) in the target distribution. In Figure 6.19, some permeability realizations sampled from the posterior distribution are plotted. We observe that the samples capture some features of the reference permeability field. Note that all these permeability fields give nearly the same fractional flows as the reference fractional flow, so they are all eligible samples.

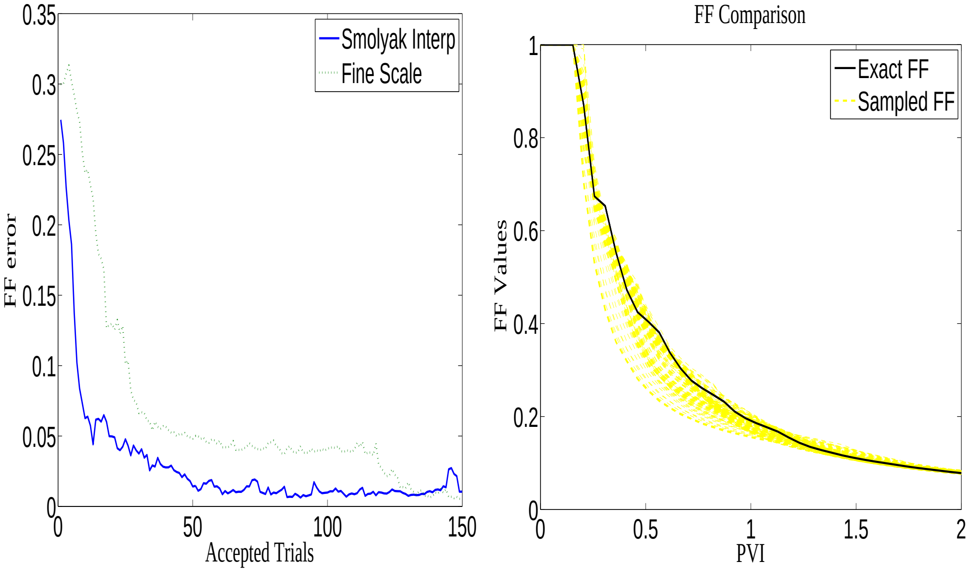


Fig. 6.18. Left: The fractional flow errors for Algorithm I compared with Algorithm IV. Right: The fractional flows of sampled realizations and the reference fractional flow. In these numerical tests, $\delta = 0.05$, $\sigma_f^2 = 0.003$ and 11×11 coarse-scale model is used.

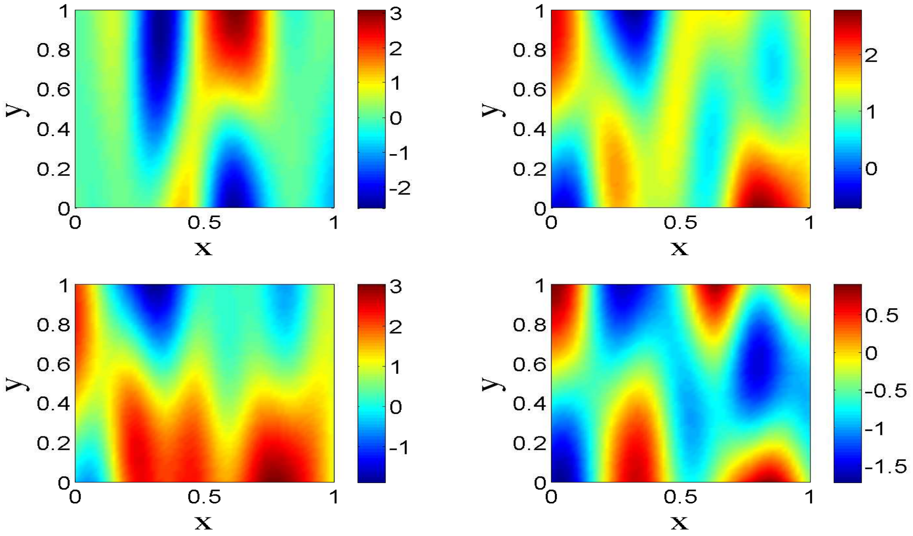


Fig. 6.19. Samples of the permeability realizations. Realizations are selected to represent the uncertainty range in the simulations.

6.1.3. Single-Phase Flow with Exponential Covariance

We introduce numerical results which use the proposed algorithms for a permeability generated using exponential covariance (2.17) in the KLE. As noted in Section 2.2, a proper expansion using exponential covariance contains many more terms than an expansion using normal covariance in order to capture the effects of the desired permeability fields. Moreover, unlike in the case of the normal covariance, the permeability fields for exponential covariance have much rougher features. We wish to show the results using the proposed algorithms in the context of exponential covariance are similar to the previous results which used normal covariance.

We generate a truncated KLE using exponential covariance with correlation lengths $L_1 = 0.5$ and $L_2 = 0.1$. We keep 105 terms and assume the permeability field is known at 5 distinct points. This reduces the dimension of our stochastic space to 100. We restrict ourselves to the hypercube $[-2.5, 2.5]^{100}$ in stochastic space. We again consider a 61×61 fine grid and an 11×11 coarse grid. Recall that in the Langevin algorithms, we require the gradient of the target distribution. Since our stochastic dimension is 100, in Algorithm I we would require 100 fine-scale solutions for each proposal. This would be computationally impossible in our setting. Similarly, in Algorithm II, we require 100 coarse-scale solution for each proposal. For this reason, we do not consider Algorithm I for the numerical results in this section and we restrict ourselves to single-phase flow. Additionally, we consider only 1^{st} order interpolation, since 2^{nd} order would require $20,201$ values.

In Figure 6.20, we compare the coarse-scale response surface π^* with the interpolated coarse-scale surface $\tilde{\pi}^*$. As expected from our previous results, the $\tilde{\pi}^*$ captures some of the important features of π^* while not approximating π^* exactly.

In Figure 6.21 we compare the acceptance rate and CPU time for each algorithm.

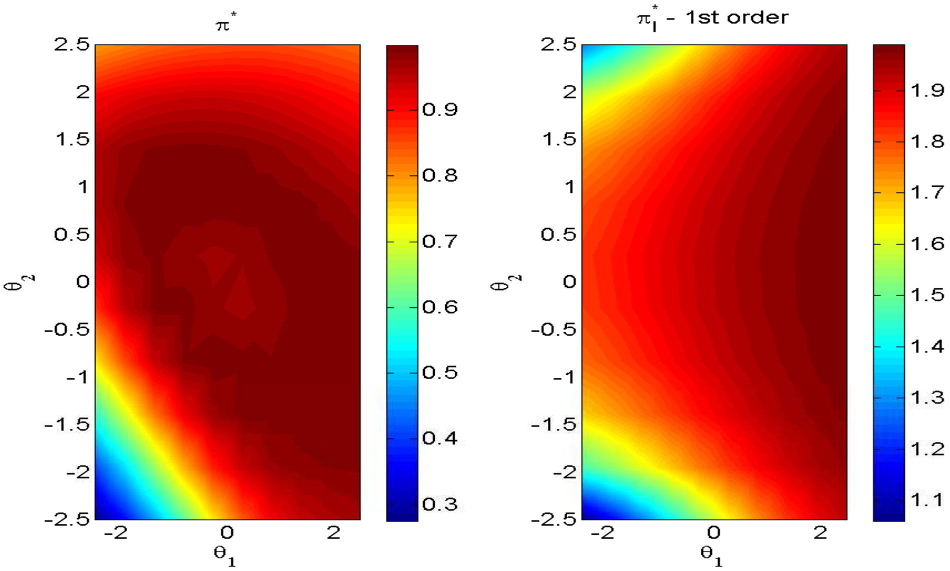


Fig. 6.20. Coarse-scale response surface π^* and interpolated coarse-scale response surface $\tilde{\pi}^*$ restricted to a 2-D hyperplane.

The acceptance rate of Algorithm IV is slightly lower than that of Algorithm II. This is similar to our previous results and is likely due to the more precise gradients in Algorithm II. We notice a large improvement in CPU time, clearly a result of the need to run 100 solutions for each sample in Algorithm II. Also, note the CPU time contribution from generating the data values for interpolation is negligible in comparison to the total CPU time, since it is the same cost as two samples in Algorithm II.

Next we compare the fractional flow errors and fractional flow comparisons for Algorithm II and Algorithm IV in Figure 6.22. In the left plot the fractional flow errors reduce at approximately the same rate, showing that both methods converge to the steady state within about the same number of iterations. In the right plot, we show the fractional flows of the sampled realizations are very close to the reference fractional flow. This is again because the error precision is taken to be small ($\sigma_f^2 = 0.001$) in the target distribution. In Figure 6.23, some permeability realizations sampled from the

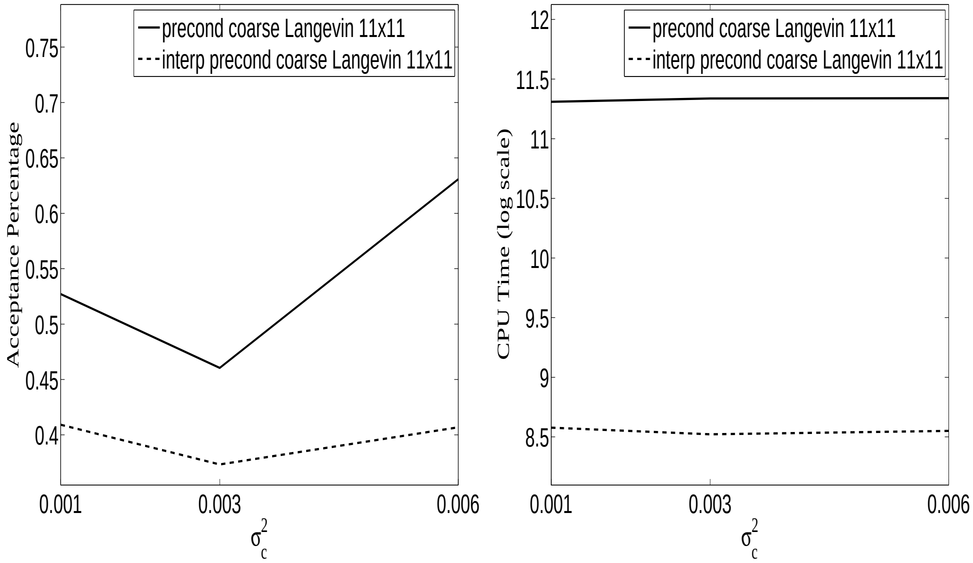


Fig. 6.21. Left: Acceptance rate comparison. Right: Natural log of CPU time (seconds) comparison. In each we compare Algorithm II and Algorithm IV for $\delta = 0.05$, $\sigma_f^2 = 0.001$.

posterior distribution are plotted. The upper left plot in Figure 6.23 represents the reference permeability. The remaining plots are three different permeabilities that were accepted using Algorithm IV.

As we can see from the preceding results, exponential covariance produces results very similar to normal covariance. In general, for multiphase flow, Algorithm IV provides similar sampling performance as Algorithm II while providing a dramatic saving in CPU time.

6.2. Richards' Equation

In this section, we consider the application of uncertainty quantification techniques introduced earlier to Richards' equation. Our problems are motivated by application in soil moisture predictions. Soil moisture conditions are important in determining the amount of infiltration and ground water recharge. Soil moisture is

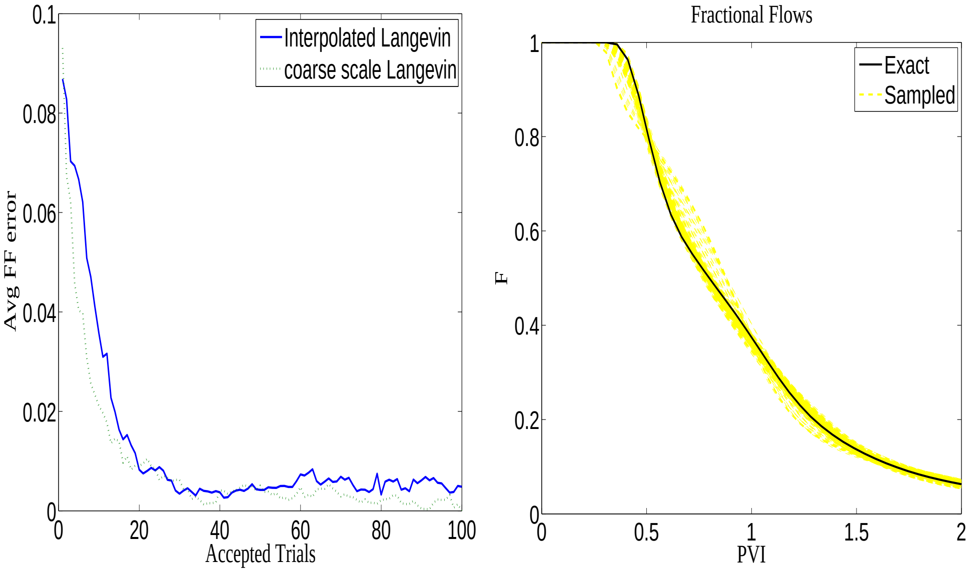


Fig. 6.22. Left: The fractional flow errors for Algorithm II compared with Algorithm IV. Right: The fractional flows of sampled realizations and the reference fractional flow. In these numerical tests, $\delta = 0.05$, $\sigma_f^2 = 0.001$.

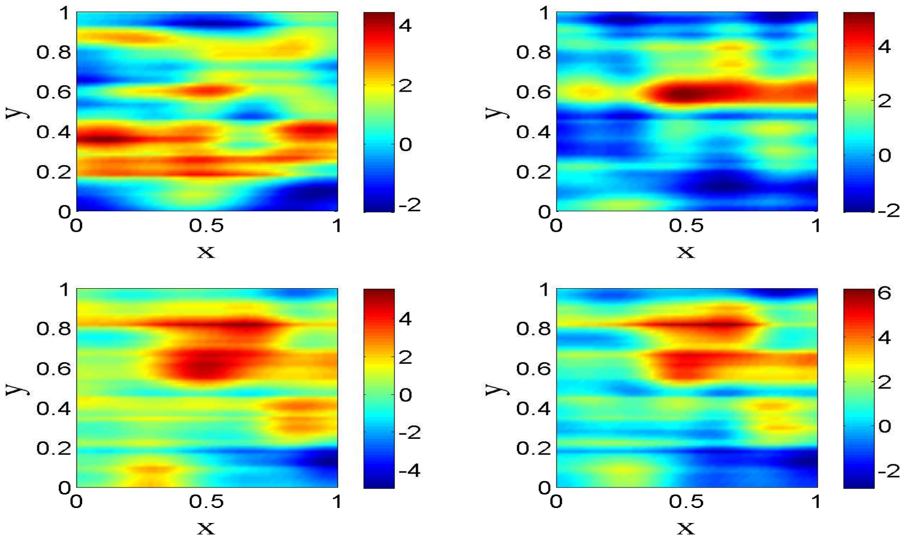


Fig. 6.23. Upper left plot is the reference permeability. The other three plots are examples of accepted permeability realizations.

controlled by factors such as soil type, topography, vegetation, and climate. Soil moisture is typically measured at different scales varying from point scale (*in-situ*) to remote sensing scale (of order several miles). The objective is to predict the soil moisture via prediction of saturated conductivity field. We note that the measure data is coarse-scale data. We apply the procedure described previously to Richards' equation. The multiscale techniques for Richards' equation differ from that of two-phase immiscible flow. In particular, we are not dealing with the saturation equation in Richards' equation. As a consequence, the coarse-scale methods are more accurate for Richards' equation.

Our goal in this section is not to repeat the same very detailed arguments and results made in Section 6.1, but rather to provide evidence that similar results hold for Richards' equation. Recall Richards' equation and the discussed constitutive relations, given in Section 2.1.2. We consider the problem of sampling the saturated conductivity field given an integrated response of the average flux over a certain region at given time steps. Specifically, for a discretized domain $\Omega = [0,1]^2$, let us define the discrete integrated response function, F. In practice, we assume that F contains the average flux on the uppermost boundary of the fine-grid domain for a given set of times. Let us suppose that for each time in a given set of k times, the flux is found across the entire upper cell boundary at n equally spaced intervals as in Figure 6.24. We denote these times as $t_1, \ldots, t_k$. The response is then given by the average flux at each of the n intervals, and each of these k times, in order. In other words,

$$F = \left(Flux_{avg}\left(x_1, t_1\right), \ldots, Flux_{avg}\left(x_n, t_1\right), Flux_{avg}\left(x_1, t_2\right), \ldots, Flux_{avg}\left(x_n, t_k\right)\right) \tag{6.2}$$

We will again be using the MsFVEM for our coarse-scale model. Note, when using the MsFVEM, we can reconstruct the fine-scale flux using the multiscale basis functions.

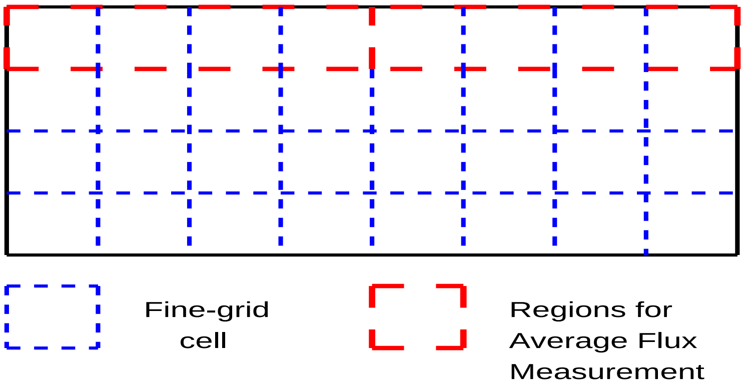


Fig. 6.24. Example of average flux for 4 × 8 grid with $n = 2$ measured intervals.

Thus, the averaging used for F will always be over the fine-grid domain, regardless of whether we are using a fine-scale or coarse-scale model. We remind the reader that Algorithm I refers to fine-scale Langevin MCMC, Algorithm II refers to preconditioned coarse-scale Langevin MCMC, and Algorithm IV refers to preconditioned interpolated coarse-scale Langevin MCMC.

6.2.1. Coarse vs Fine

In the numerical simulations, to represent the discrete saturated conductivity field we generate a KLE using normal covariance and correlation lengths $L_1 = 0.5$ and $L_2 = 0.1$. We keep 20 terms in the KLE and assume that the saturated conductivity is known at 9 sparse locations. For each simulation we test $5,000$ samples in the MCMC. We choose a 49×49 fine-scale model and a 5×5 coarse-scale model. Note these models are vertex-based, thus the number of cells are 48×48 and 4×4 in the fine-scale and coarse-scale model, respectively.

We compare numerical results for the exponential constitutive relation. We choose $\alpha = 0.01$, $\beta = 0.01$, and $\theta_s = 1.0$. In Figure 6.25 we show the boundary conditions and initial conditions for the exponential model. We assume the initial

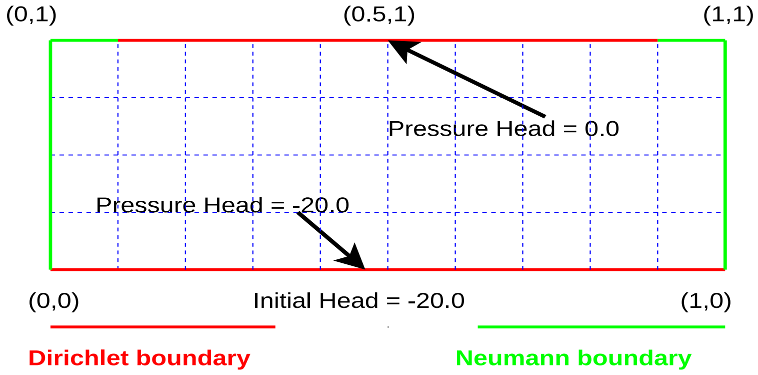


Fig. 6.25. Exponential model boundary and initial conditions.

pressure head is $u = -20$. Dirichlet boundary conditions are given by $u = -20$ on the bottom of Ω, $[0,1] \times 0$, and $u = 0$ on the portion of the top boundary between $x = 0.1$ and $x = 0.9$, $[0.1, 0.9] \times 1$. We assume no flow boundary conditions otherwise. We choose these particular boundary conditions rather than simply no flow on the sides and given pressure heads on the top and bottom so that we can see more variations in the pressure head profiles. Additionally, we scale the average flux response surfaces down by a factor of 20. This is done simply to keep σ_f and σ_c values at approximately the same magnitude as in our previous MCMC investigation for multiphase immiscible flow.

First, we present a comparison between the fine-scale target distribution π and the coarse-scale distribution π^*, defined by (5.3) and (5.9), respectively. We plot the restriction of the target distributions to a 2D hyperplane by fixing the value of 9 of the 11 stochastic dimensions. In Figure 6.26 we show both π and π^*. While the values in the surfaces do not correspond exactly, the overall agreement between the two surfaces appears to be quite good.

Next, we compare acceptance rates of Algorithm I with those of Algorithm II. In the left portion of Figure 6.27 we compare these acceptance rates with different

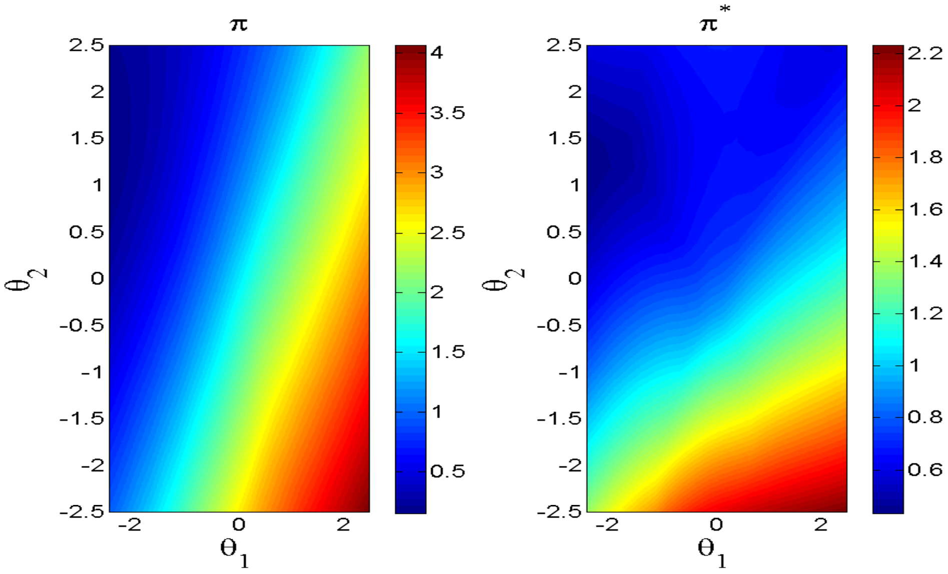


Fig. 6.26. Left: Fine-scale response surface π restricted to the same 2D hyperplane. Right: Coarse-scale response surface π^* restricted to a 2D hyperplane.

coarse scale precision σ_c. Recall the acceptance rate is the ratio between the number of accepted saturated conductivities and the number of fine-scale acceptance- rejection tests. Since Algorithm I does not have a coarse-scale test, it is constant for all σ_c and its acceptance rate is simply the number of accepted saturated conductivities divided by the total number of samples. From Figure 6.27, we notice Algorithm II has a higher acceptance rate than Algorithm I. This is due to the preconditioning step, which filters proposals that are not likely to be accepted by the fine-scale test. In the right portion of Figure 6.27 we compare the CPU time for each of the two algorithms. We notice a significant savings in CPU time when using Algorithm II. This is partly due to the fact that Algorithm II uses a coarse-scale gradient, $\nabla \log \pi^*(k)$, while Algorithm I uses a fine-scale gradient $\nabla \log \pi(k)$. Since $\pi(k)$ and $\pi^*(k)$ are functions determined by simulations, these gradients are computed using finite differences. Thus we must run a simulation in each of the 11 stochastic dimensions, so our CPU savings is partly due to 11 coarse-scale simulations per sample as opposed

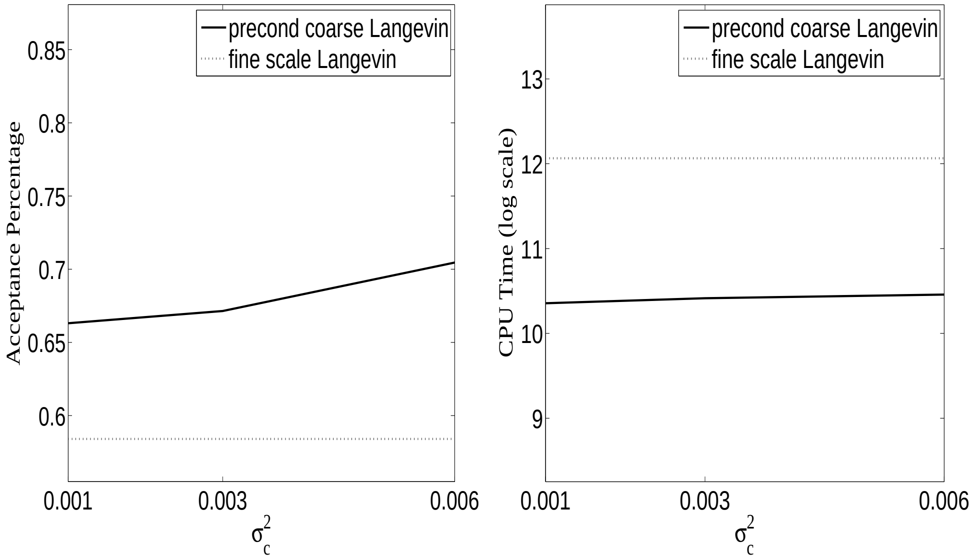


Fig. 6.27. Comparison between Algorithm I and Algorithm II for various σ_c values with $\sigma_f^2 = 0.001$ and $\delta = 0.05$. Left: Acceptance rate. Right: Natural log of CPU time (seconds)

to 11 fine-scale simulations. Additionally, the preconditioning step in Algorithm II helps filter the samples, resulting in fewer fine-scale acceptance-rejection tests.

We now compare the average flux errors for both algorithms in Figure 6.28. First, we would like to compare the convergence rates for each of the algorithms. Second, we wish to show that the sample saturated conductivities have responses which closely match the reference response. The left plot demonstrates that both algorithms converge to steady state for approximately the same number of accepted trials. In the right plot, the average fluxes for the sampled realizations are plotted (dotted lines) along with the reference average flux (solid line). Recall the average flux defined by (6.2) is given by 2 average flux values at 10 discrete time steps. The x-axis is given by this discrete data, where the first 2 values on the x-axis represent the two average flux values at the first time step. The second 2 values represent the average fluxes at the second time step and so on. The sampled realizations closely

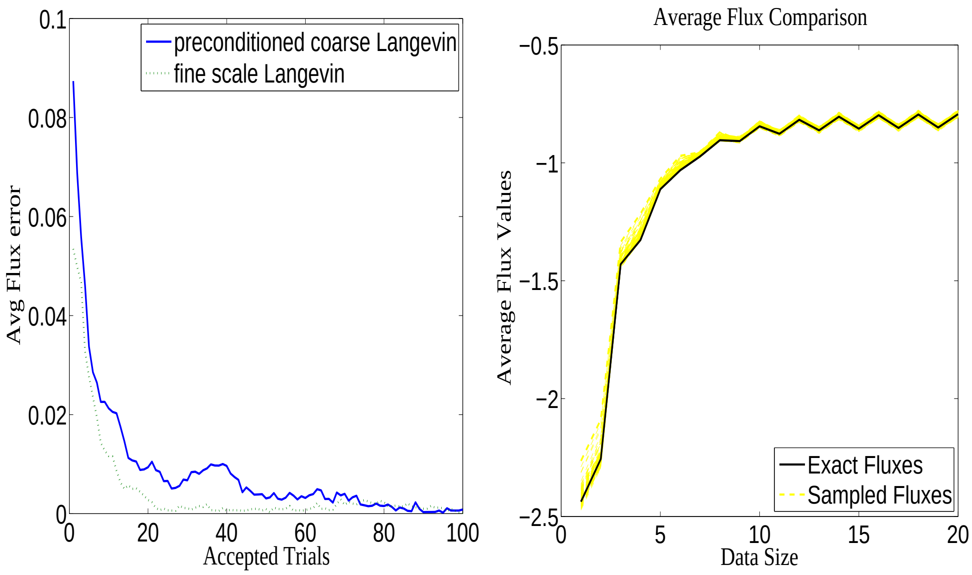


Fig. 6.28. Left: Average flux errors for Algorithm I compared to Algorithm II. Right: The fluxes of sampled realizations and the reference flux.

match the reference flux because the error precision is taken to be small ($\sigma_f^2 = 0.001$) in the target distribution.

Lastly, we present some saturated conductivity realizations sampled from the posterior distribution. In Figure 6.29 we observe that the samples capture some features of the reference saturated conductivity quite well (such as the peak at approximately $(0.75, 1)$) while seemingly ignoring other features (such as the valley from approximately $(0.4, 1)$ to $(0.4, 0.2)$). Note that these accepted saturated conductivity fields correspond to those with errors in Figure 6.28, thus they give nearly the same average flux responses and are all eligible samples.

6.2.2. Sparse Grid Collocation

In this section we briefly compare the numerical results using Algorithms I, II and IV. We consider Algorithm IV with two different interpolated values. First, we consider interpolation of the coarse-scale target distribution π^*. Next we consider

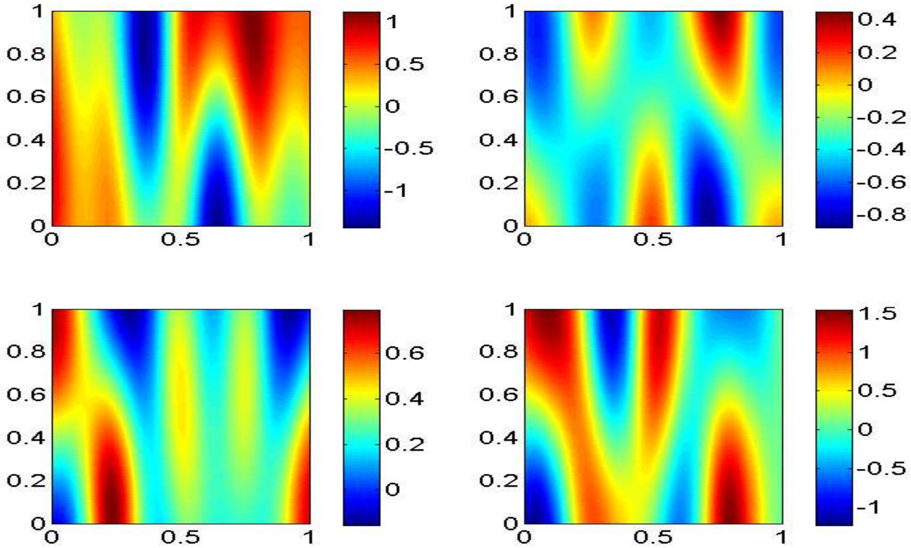


Fig. 6.29. Exact K_s (upper left) and three different accepted conductivites.

interpolation of the average flux found using the MsFVEM model. Our aim is to show that Algorithm IV can be used as an inexpensive alternative to Algorithm II.

6.2.2.1. Interpolation of the Target Distribution

We consider the same exponential constitutive relation as in the previous section, given by (2.11). We again scale our average flux response by 20 as in the previous numerical results. We choose a different exact permeability using a KLE with 20 terms. We again use normal covariance and correlation lengths $L_1 = 0.5$ and $L_2 = 0.1$. We now assume that the saturated conductivity is known at 11 sparse locations rather than 9. We use a 49×49 fine-scale model, a 5×5 coarse-scale model, and we base our results on $5,000$ samples in the MCMC. For variety, we consider a slightly different set of initial conditions and boundary conditions. We assume the initial pressure head is $u = -10$. Dirichlet boundary conditions are given by $u = -10$ on the bottom of Ω, $[0, 1] \times 0$, and $u = 0$ on the portion of the top boundary between $x = 0.3$ and $x = 0.7$,

$[0.3, 0.7] \times 1$. We assume Neumann boundary conditions otherwise.

We again refer to k^{th} order Smolyak interpolation as $A(N+k, N)$ in (3.19). In particular, for $N = 9$ stochastic dimensions, 1^{st} order results in 19 nodes, 2^{nd} order results in 181 nodes, and 3^{rd} order results in $1,177$ nodes. It is worth noting that even for 3^{rd} order, the computational cost in computing the Smolyak nodes is still small compared to the coarse-gradient Langevin MCMC. For example if we are to sample in 9 dimensions using coarse-gradient Langevin, then we must run 10 forward simulations for each sample. After approximately 120 samples, we have already run the coarse-scale simulations more times than if we had generated nodes for 3^{rd} order Smolyak interpolation. We now consider a set of numerical results for Algorithms I, II and IV. We do not present the results in as much detail as in the previous section. We simply wish to provide numerical justification for the use of Algorithm IV.

First, we present a comparison between the coarse-scale target distribution π^* and the interpolated coarse-scale distribution $\tilde{\pi}^*$ for different interpolation orders. We restrict the target distributions to a 2D hyperplane by fixing all but 2 of the stochastic dimensions. In Figure 6.30 we show π^* and $\tilde{\pi}^*$ for each interpolation order. We note that each interpolation surface matches the reference quite well. However, we have some local effects in the 2^{nd} and 3^{rd} order surface that do not appear in the original coarse-scale surface. This is likely due to oscillations from the high order polynomials involved in 2^{nd} and 3^{rd} order interpolation. Due to these oscillations, and the fact that 1^{st} order interpolation approximates quite well, we consider only 1^{st} order interpolation.

Next, we compare acceptance rates of Algorithm I, II and IV. The left plot in Figure 6.31 shows the acceptance rates with different coarse scale precision σ_c. We notice Algorithms II and IV have a higher acceptance rate than Algorithm I. As discussed previously, this is due to the preconditioning step. Also notice Algorithm II

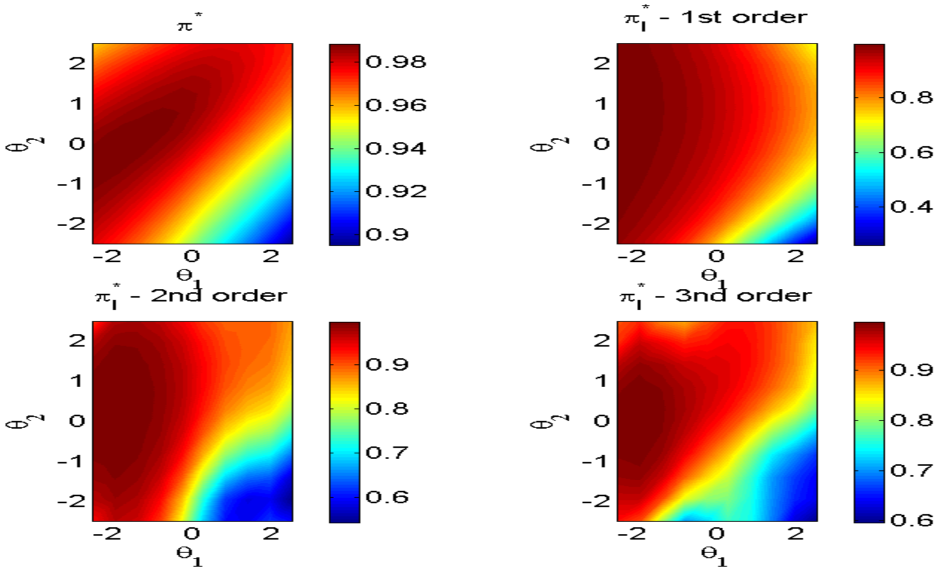


Fig. 6.30. Coarse-scale response surface π^* and interpolated coarse-scale response surfaces $\tilde{\pi}^*$ restricted to a 2D hyperplane.

has a higher acceptance rate than Algorithm IV. This is expected, as the coarse-scale target distribution in Algorithm IV is an approximation of the coarse-scale target distribution in Algorithm II. Thus, there will be some slight errors involved, and the Langevin proposals in Algorithm IV may be directed into a region with slightly lower acceptance probabilities than Algorithm II. In the right portion of Figure 6.31 we compare the CPU time for each of the algorithms. We again notice a savings in CPU time for Algorithm II as compared to Algorithm I. Further, Algorithm IV uses significantly less CPU time than both Algorithms I and II. This is due to the interpolated gradients. Interpolation using 19 points is nearly instantaneous, while in Algorithms I and II we must run a fine or coarse-scale solution in each stochastic dimension to find the gradients in the Langevin algorithm. Let us also note that in Figure 6.31 we have not included the CPU time necessary to generate the interpolation points. Since we use 19 points, we must obtain 19 coarse-scale solutions to generate the interpolation data. Since this is less than the CPU time required for just 2

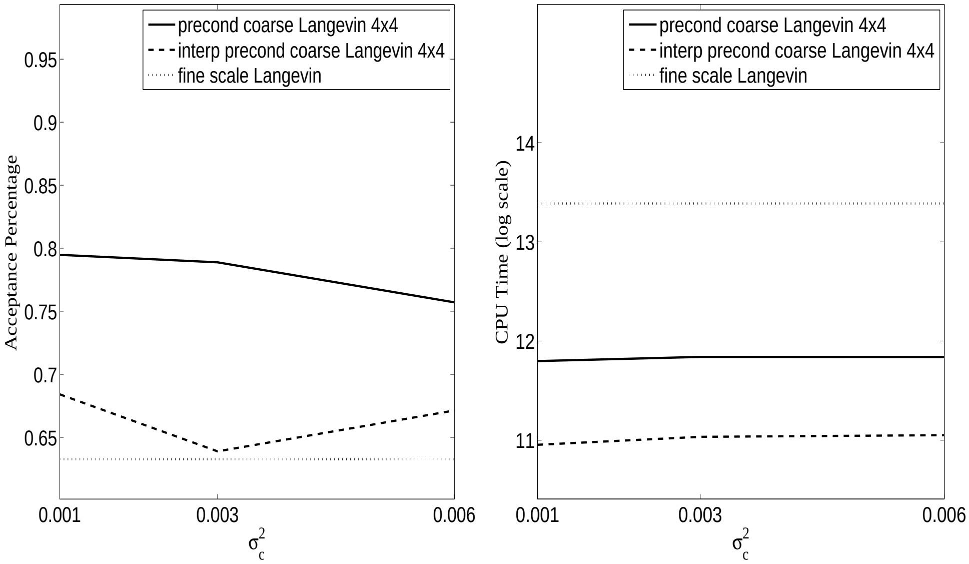


Fig. 6.31. Left: Acceptance rate comparison. Right: Natural log of CPU time (seconds) comparison. Comparison is between Algorithm I, Algorithm II, and Algorithm IV for various σ_c with $\sigma_f^2 = 0.001$ and $\delta = 0.05$.

proposals in Algorithm II, it is negligible with respect to the MCMC CPU time.

In Figure 6.32 the left plot demonstrates that Algorithm IV converges to steady state for approximately the same number of accepted trials as Algorithm I. The right plot demonstrates that the accepted fluxes closely match the reference flux. In Figure 6.33 we compare the reference saturated conductivity (upper left) with three different accepted conductivities. We observe that the samples capture many of the features of the reference saturated conductivity.

We note that we have generally considered a KLE for Richards' equation with normal covariance and correlation lengths given by $L_1 = 0.5$ and $L_2 = 0.1$ in (2.16). We now instead consider correlation lengths $L_1 = 0.2$ and $L_2 = 0.2$. We would not expect this to change our results, but we will briefly present some numerical details verifying this.

In Figure 6.34 we compare π^* with $\tilde{\pi}^*$ for 1^{st} through 3^{rd} order interpolation. As

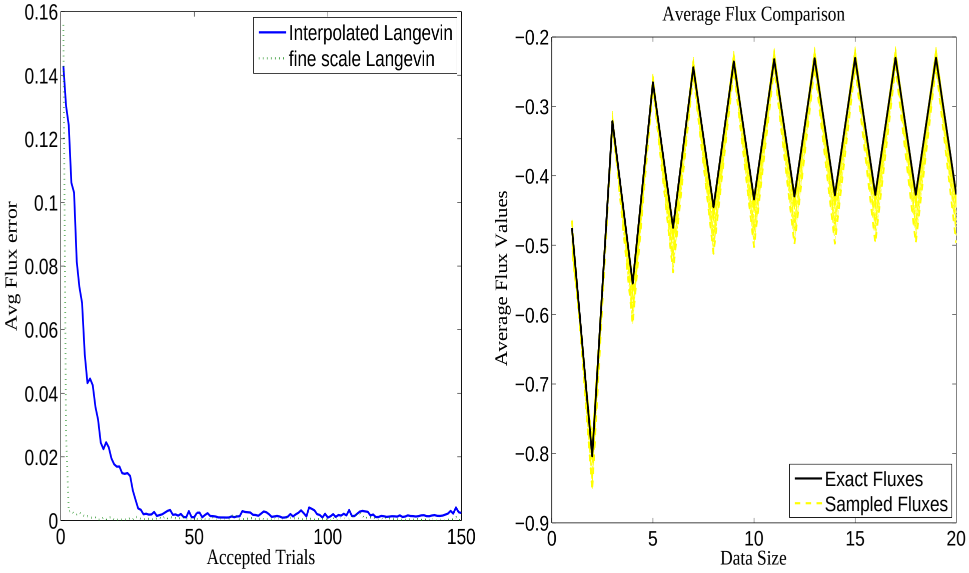


Fig. 6.32. Left: Average flux errors for fine-scale Langevin compared to interpolated Langevin. Right: The fluxes of sampled realizations and the reference flux. In each plot $\sigma_f^2 = 0.001$ and $\delta = 0.05$.

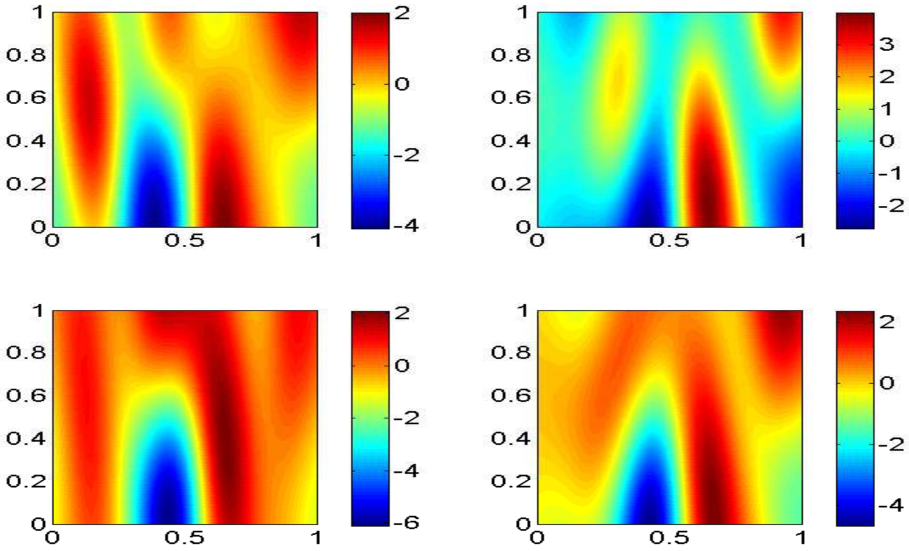


Fig. 6.33. Exact K_s (upper left) and three different accepted conductivites.

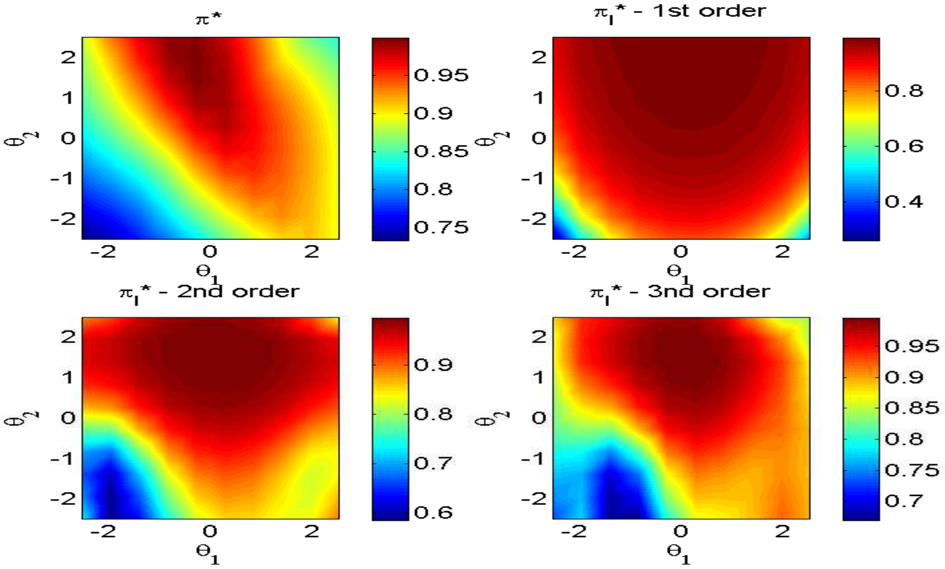


Fig. 6.34. Coarse-scale response surface π^* and interpolated coarse-scale response surfaces $\tilde{\pi}^*$ restricted to a 2D hyperplane.

previously, the interpolation surfaces match the general features of the coarse-scale surface, with the higher order surfaces displaying more extraneous features. In Figure 6.35 we show the acceptance rate (left) and CPU time (right) for Algorithms I, II and IV. We again find Algorithm IV has a similar acceptance rate to Algorithms I and II, while providing a large saving in CPU time.

In Figure 6.36 the left plot demonstrates that Algorithm IV converges to steady state for approximately the same number of accepted trials as Algorithm I. The right plot demonstrates that the accepted fluxes closely match the reference flux. In Figure 6.37 we compare the reference saturated conductivity (upper left) with three different accepted conductivities. The samples capture many of the features of the reference saturated conductivity.

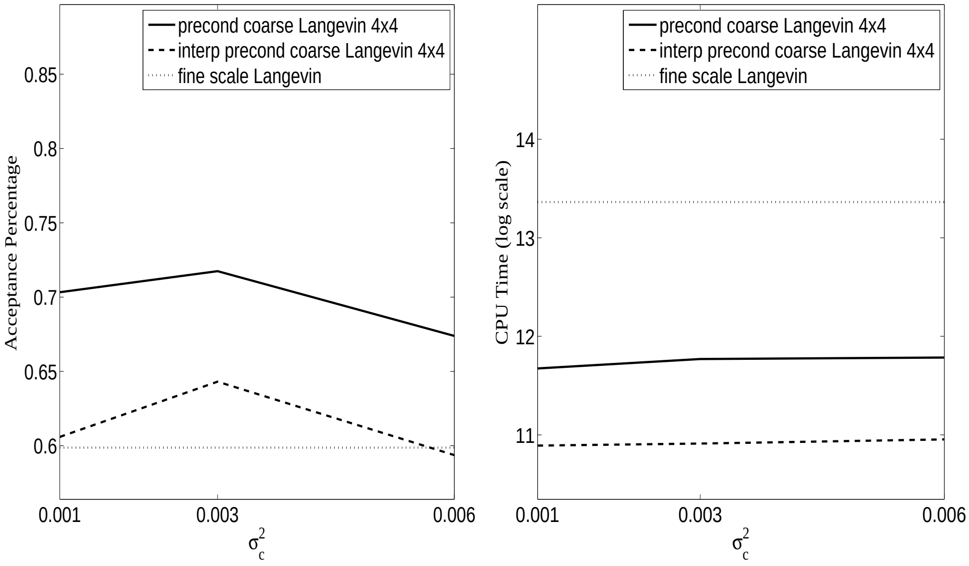


Fig. 6.35. Left: Acceptance rate comparison. Right: Natural log of CPU time (seconds) comparison. Comparison is between Algorithm I, Algorithm II, and Algorithm IV for various σ_c with $\sigma_f^2 = 0.001$, $\delta = 0.05$, $L_1 = 0.2$ and $L_2 = 0.2$.

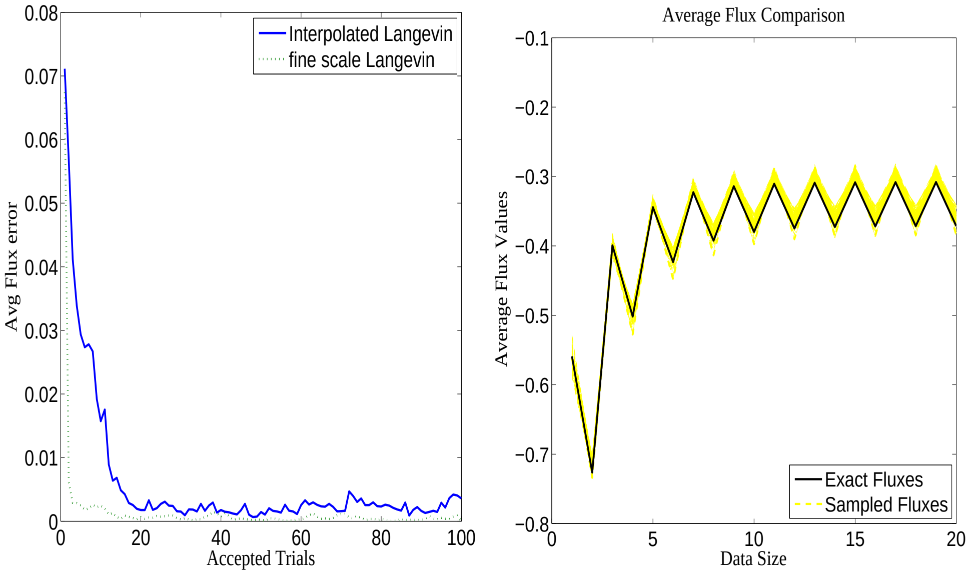


Fig. 6.36. Left: Average flux errors for fine-scale Langevin compared to interpolated Langevin. Right: The fluxes of sampled realizations and the reference flux. In each plot $\sigma_f^2 = 0.001$, $\delta = 0.05$, $L_1 = 0.2$ and $L_2 = 0.2$.

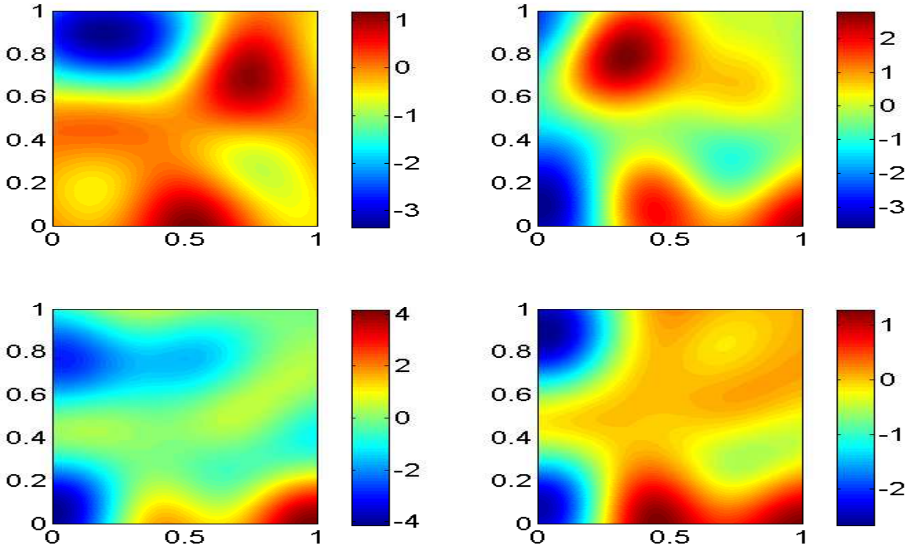


Fig. 6.37. Exact K_s (upper left) and three different accepted conductivites.

6.2.2.2. Interpolation of the Average Fluxes

We perform the same numerical tests as in the previous section, but we consider interpolation of the average flux in Algorithm IV rather than the target coarse-distribution. We generate average flux values at the Smolyak nodes in our stochastic dimensions using the MsFVEM as in (6.24). For each new proposal θ in Algorithm IV we generate, $\tilde{F}^*(\theta)$ by interpolation. We also use the analytic formula for the interpolated gradient, given by (5.24). Once we have the average flux, $\tilde{\pi}^*$ is computed using $\tilde{F}^*$ rather than interpolated as in the previous section.

We again consider the exponential constitutive relation given in Section 2.1.2. We consider the same exact boundary conditions as in the previous section, and we again scale our average flux response by 20. We choose a different exact permeability using a KLE with 20 terms, and we again assume that the saturated conductivity is known at 9 sparse locations. We use a 49×49 fine-scale model, a 5×5 coarse-scale model, and we base our results on $5,000$ samples in the MCMC.

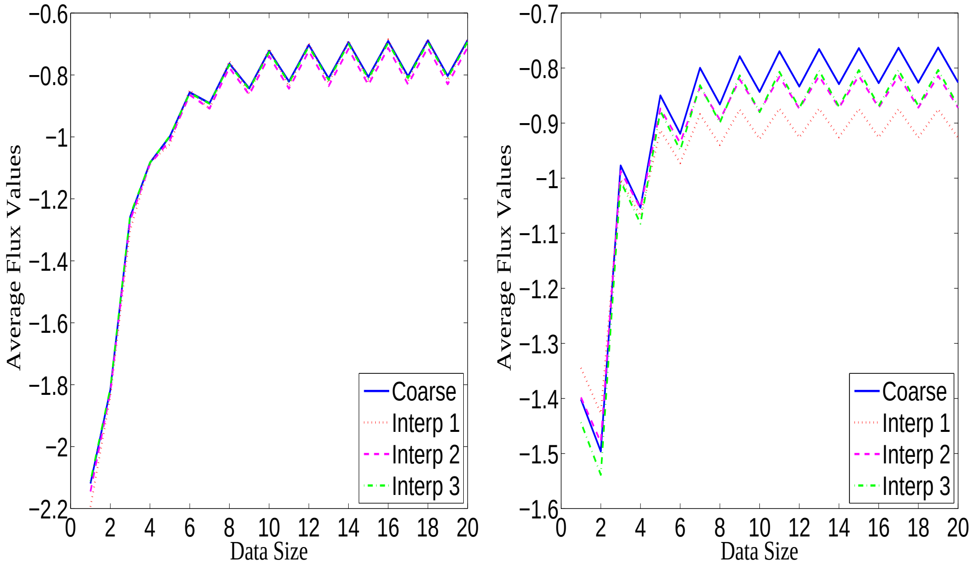


Fig. 6.38. Two typical interpolated and coarse average flux responses.

We first would like to discuss the possible errors involved in interpolation of the average flux, as opposed to interpolation of the coarse-scale target distribution. In Figure 6.38 we compare a set of typical coarse-scale average fluxes and a set of interpolated average fluxes for different order interpolations. Recall the average flux defined by (6.2) is given by 2 average flux values at 10 discrete time steps. As expected, the interpolation surfaces generally match the coarse-scale average flux. Most likely due to the smoothness in stochastic dimension of our response, the increase in interpolation order does not appear to make a significant impact upon our interpolation results. Due to this fact, we generally consider first order interpolation, which requires very little computational effort to generate the Smolyak nodes.

Recall that we must also compute the gradient of the target distribution in the Langevin algorithm. While Smolyak interpolation matches the coarse-scale average flux responses quite well, there is no guarantee that the derivatives of the average fluxes match well. In Figure 6.39 we show the gradients of the target distribution for

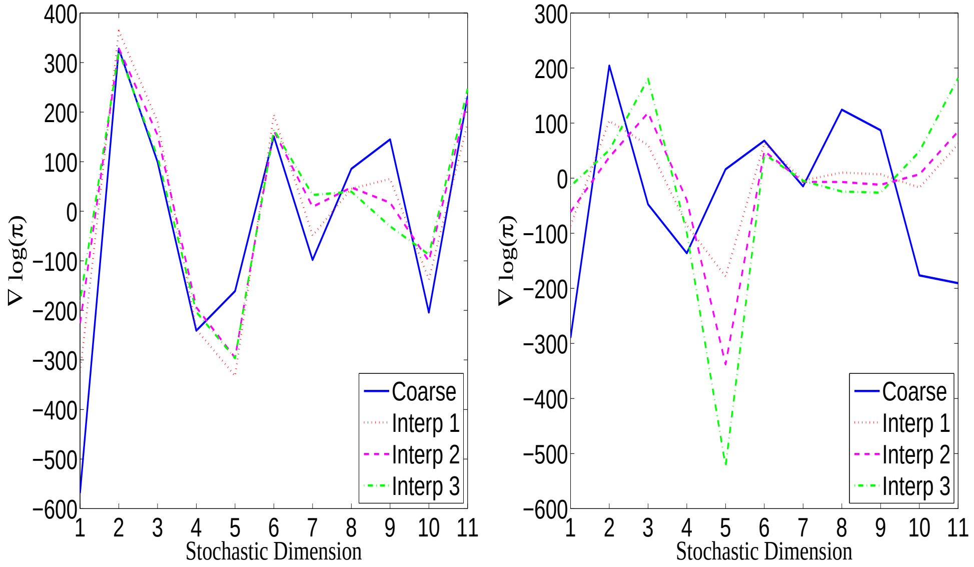


Fig. 6.39. Two typical gradient of the target distribution for both interpolated and coarse average flux responses.

two typical cases, using 1^{st} through 3^{rd} order Smolyak interpolation. In the left plot, the gradients using interpolation closely match the gradient of the coarse distribution. In the right plot, however, the gradients do not match nearly as well. While it may be the case that the interpolated gradient does not match well, note we have similar approximations for 1^{st} through 3^{rd} order, providing further evidence that we may consider only 1^{st} order. For this reason, we expect some trials in Algorithm IV may be rejected simply due to inaccurate gradients in the Langevin proposal.

We present a comparison between the coarse-scale distribution π^*, and the coarse-scale distribution found through interpolation, $\tilde{\pi}^*$. We plot the restriction of the target distributions to a 2D hyperplane by fixing the value of 9 of the 11 stochastic dimensions. In Figure 6.40 we show both π^* and $\tilde{\pi}^*$ for 1^{st}, 2^{nd} and 3^{rd} order Smolyak interpolation. As one would expect, the agreement between π^* and $\tilde{\pi}^*$ is very close in each case. The 3^{rd} order surface tends to have more extraneous effects due to the large degree polynomial involved in the interpolation. We again consider only 1^{st}

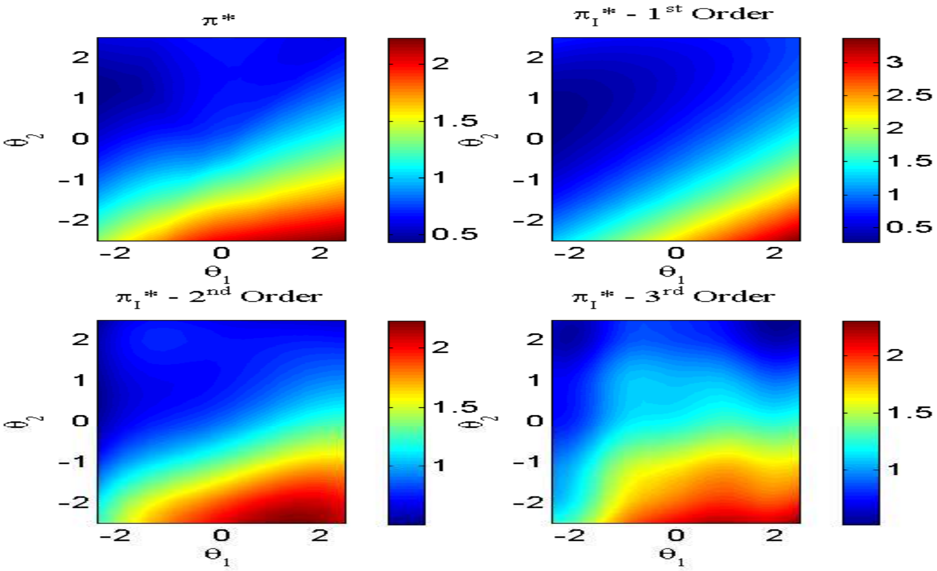


Fig. 6.40. Coarse-scale response surface π^* and interpolated coarse-scale response surfaces $\tilde{\pi}^*$ restricted to a 2D hyperplane.

order interpolation.

In the left portion of Figure 6.41 we compare these acceptance rates with different coarse scale precision σ_c. From the figure, we notice Algorithm II has a higher acceptance rate than Algorithm I while Algorithm IV has a slightly lower acceptance rate. This is likely due to the errors in the gradients using collocation. In the right portion of Figure 6.41 we compare the CPU time for each of the algorithms. We notice a savings in CPU time above Algorithm I when using Algorithm II. We notice an even larger saving when using Algorithm IV.

In Figure 6.42 the left plot demonstrates that Algorithm IV converges to steady state for approximately the same number of accepted trials as the Algorithm I. The right plot demonstrates that the accepted fluxes closely match the reference flux.

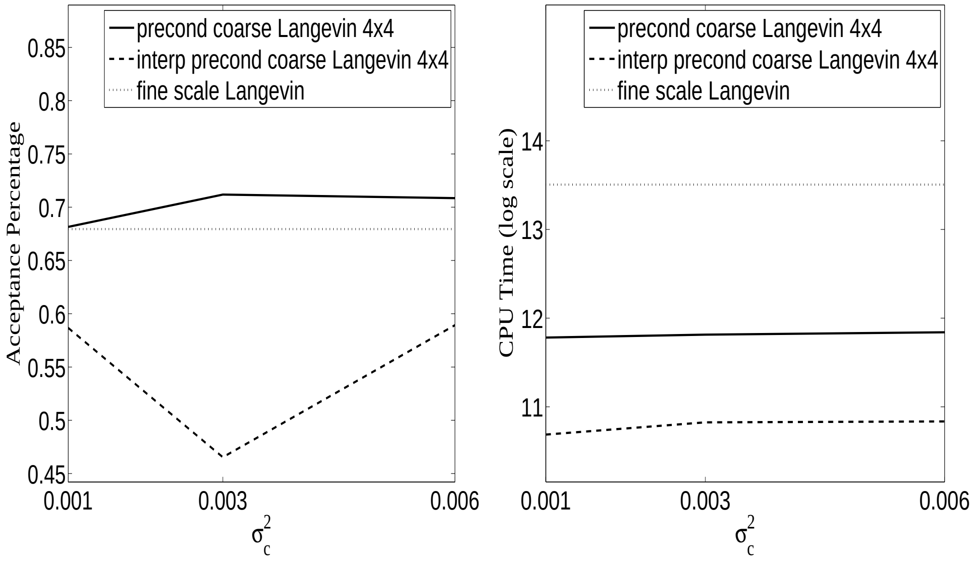


Fig. 6.41. Left: Acceptance rate comparison. Right: Natural log of CPU time (seconds) comparison. Each plot compares Algorithm I, Algorithm II, and Algorithm IV for various σ_c values with $\sigma_f^2 = 0.001$ and $\delta = 0.05$.

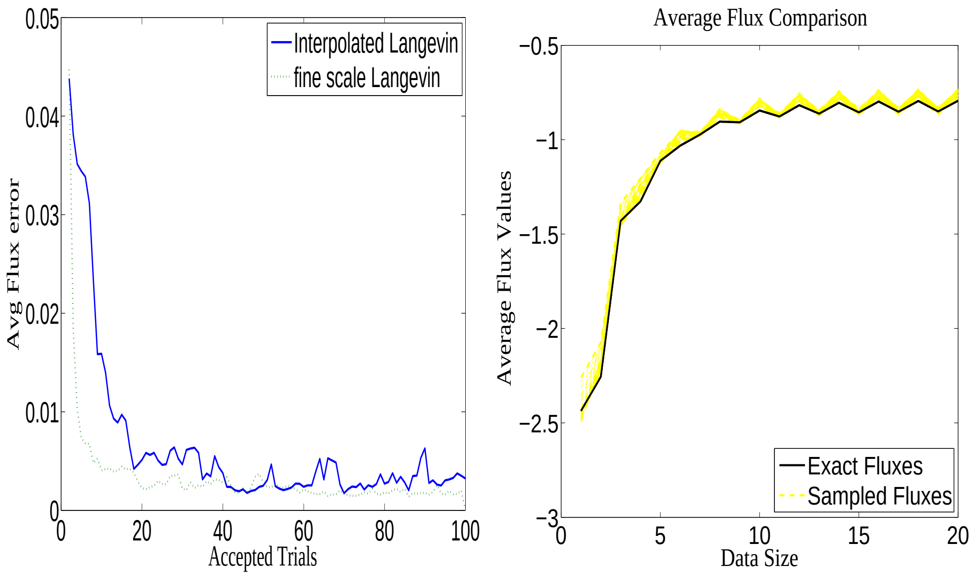


Fig. 6.42. Left: Average flux errors for Algorithms I and IV. Right: The fluxes of sampled realizations and the reference flux. In each plot $\sigma_f^2 = 0.001$ and $\delta = 0.05$.

6.2.3. Exponential Covariance

We now introduce a set of numerical results which use the proposed algorithms for a conductivity generated using the KLE with exponential covariance (2.17). As noted in Section 2.2, the KLE with exponential covariance contains many more terms in the expansion. Moreover, the conductivity fields for exponential covariance have more varied fine-scale features. We wish to show the results using the proposed algorithms in the context of exponential covariance are similar to the previous results which used normal covariance.

We generate a truncated KLE using exponential covariance with correlation lengths $L_1 = 0.2$ and $L_2 = 0.2$. We keep 105 terms and assume the conductivity field is known at 5 distinct points, thus we have 100 stochastic dimensions. We restrict ourselves to the hypercube $[-2.5, 2.5]^{100}$ in stochastic space. We again consider a 49×49 fine grid and a 5×5 coarse grid. Since we must compute the gradient of the target distribution for each 100 dimensions, in Algorithm I we would require 100 fine-scale solutions for each proposal. This is computationally impossible in our setting. Thus we compare only Algorithms II and IV. Additionally, we consider only 1^{st} order interpolation, since 2^{nd} order would require $20,201$ values. We again present our numerical results briefly, since many of the same ideas have already been discussed.

In Figure 6.43 we compare π^* with $\tilde{\pi}^*$. As previously, the interpolation surfaces match the general features of the coarse-scale surface. In Figure 6.44 we show the acceptance rate (left) and CPU time (right) for Algorithms II and IV. We find that the two algorithms have a similar acceptance rate, while Algorithm IV is over ten times faster than Algorithm II. This vast improvement in CPU time is clearly due to the use of interpolated gradients in Algorithm IV, as opposed to coarse-scale gradients in Algorithm II.

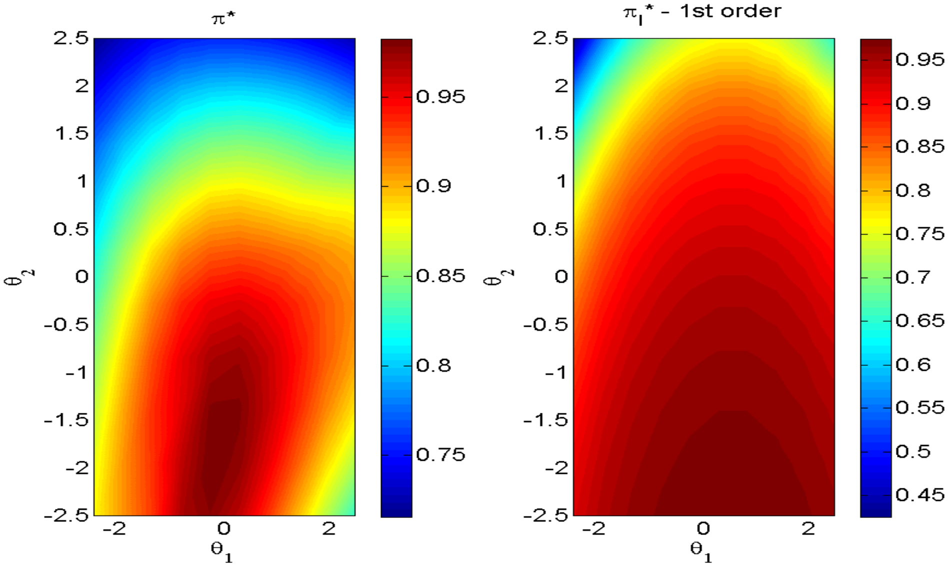


Fig. 6.43. Coarse-scale response surface π^* and interpolated coarse-scale response surfaces $\tilde{\pi}^*$ restricted to a 2D hyperplane.

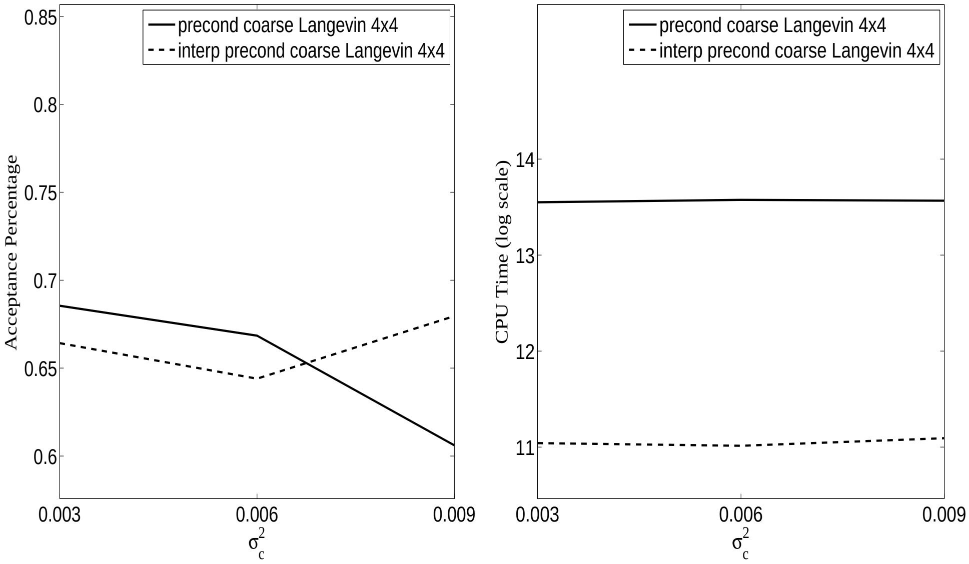


Fig. 6.44. Left: Acceptance rate comparison. Right: Natural log of CPU time (seconds) comparison. Each plot compares Algorithm I and Algorithm IV for various σ_c values with $\sigma_f^2 = 0.001$ and $\delta = 0.05$. We use exponential covariance in the KLE with $L_1 = 0.2$ and $L_2 = 0.2$.

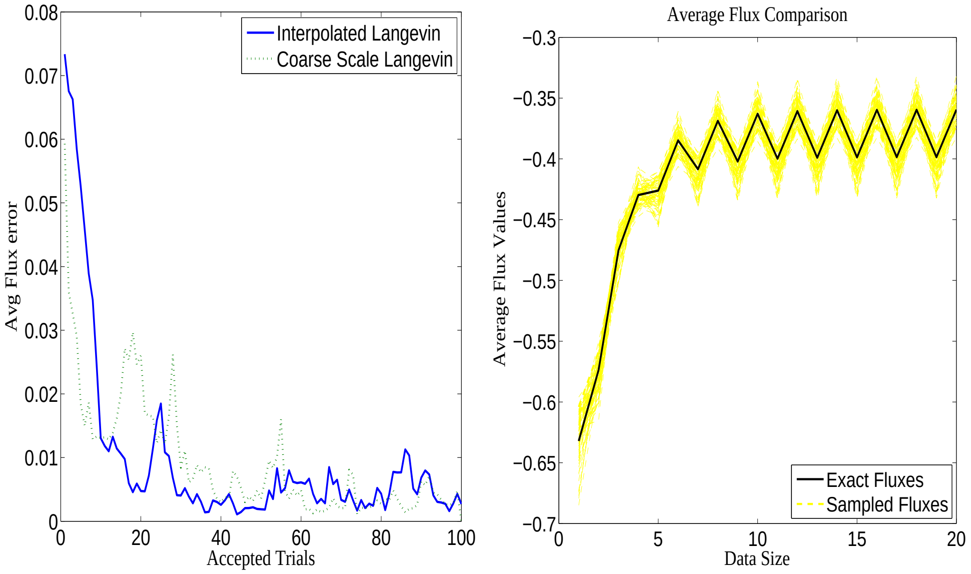


Fig. 6.45. Left: Average flux errors for Algorithm II compared to Algorithm IV. Right: The fluxes of sampled realizations and the reference flux. In each plot we use exponential covariance in the KLE, $\sigma_f^2 = 0.001$, $\delta = 0.05$, $L_1 = 0.2$ and $L_2 = 0.2$.

In Figure 6.45 the left plot demonstrates that Algorithm IV converges to steady state at a similar rate to Algorithm II. The right plot demonstrates that the accepted fluxes closely match the reference flux. In Figure 6.46 we compare the reference saturated conductivity (upper left) with three different accepted conductivities. The samples appear to capture some of the features of the reference saturated conductivity. Note the conductivity has many small features, as opposed to a few large smooth features in the case of normal covariance.

As we can see from the preceding results, exponential covariance produces results very similar to normal covariance for Richards' equation. We find that Algorithm IV provides an acceptance rate similar to Algorithm II while providing a dramatic savings in CPU time.

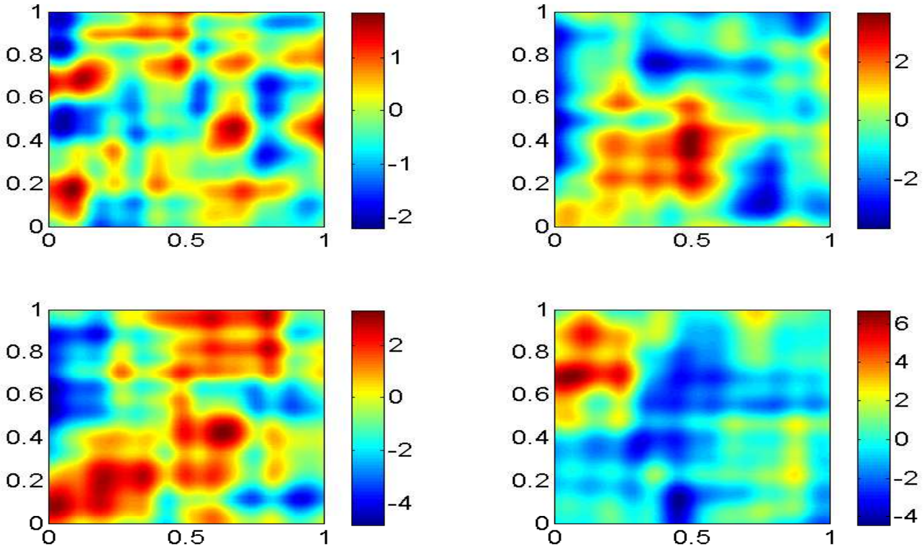


Fig. 6.46. Exact K_s (upper left) and three different accepted conductivites.

CHAPTER VII

CONCLUSIONS

7.1. Conclusions

In this dissertation, we have combined multiscale methods with polynomial chaos expansions and sparse grid collocation techniques to solve uncertainty quantification problems in porous media. We have presented analysis for both polynomial chaos expansions and sparse grid collocation methods. We have discussed and analyzed applications of sparse grid collocation to multiscale finite element methods. We have also presented an application of collocation methods in extremely high dimensions to estimate upscaled permeabilities.

We have applied the various techniques to an uncertainty quantification problem in which we sample subsurface properties given some integrated response. We discussed the preconditioned coarse gradient Langevin Markov chain Monte Carlo algorithm, as well a variant using sparse grid collocation. We have applied these Langevin algorithms, as well as traditional Langevin algorithms, to applications using both two-phase immiscible flow and Richards' equation. We have found the coarse grid based algorithms give similar sampling results as compared to the traditional fine-scale Langevin algorithm. Further, we have shown the coarse grid algorithms provide a dramatic savings in computation time.

7.2. Future Work

While we have presented numerous techniques for applications in uncertainty quantification, we believe there are still many interesting areas for further research. In the area of upscaling combined with collocation, we believe there is a possibility for further research in various petroleum applications. For some permeability models it may be necessary to have a parameterization containing thousands of stochastic variables. In these cases, it may not be possible to use sparse grid collocation methods in the full stochastic dimensions. It may, however, be possible to use collocation for a reduced set of variables and introduce a correction term to account for the remaining variables.

It would be interesting in the future to consider the applications of upscaling discussed in the previous paragraph to uncertainty quantification. In particular, we observed that using fewer eigenvectors (or lower dimensional space), one can approximate statistical properties of flow and transport accurately in some cases. We can consider using these reduced upscaling techniques and resulting response surfaces in uncertainty quantification problems. These techniques will allow us to perform uncertainty quantification on very large models.

One can further consider improving CPU time by re-using the basis functions generated for the MsFVEM. One example is to re-use the basis functions if the permeability field is conditionally accepted for fine-scale simulations. In this case, the basis functions are already available from the coarse-scale simulations. One can also attempt to take advantage of pre-computed basis functions in Langevin algorithms.

One can consider improved Langevin algorithms by using various discretization methods. In our current studies, we have used explicit approaches to discretize Langevin equation. It is known that for linear problems (Gaussian distributions) [33], an implicit discretization of Langevin equations can give samples with acceptance probability 1, and the mixing of the Markov chain is very fast [2]. Though, for general nonlinear probability density functions, there is no clear way to derive more efficient discretization methods for MCMC. For our specific problems one can still consider various other discretization techniques, e.g., semi-implicit discretizations.

In the future, it would be also interesting to test how saturation upscaling affects the efficiency of the proposed sampling methods. We expect improved results if the upscaling method for the saturation is accurate. However, these upscaled equations will be more expensive, because the upscaling of the saturation is typically nonlocal. As a consequence, computation time of coarse-scale Langevin algorithms will be affected. The investigation of various upscaling methods for the saturation and their effect in uncertainty quantification is one of our future goals.

Another future direction is to consider more general prior models where the correlation lengths and the variance are unknown. In this case, one deals with very large uncertainty space and the uncertainty quantification is a challenging problem. We believe the proposed approaches which do not use the interpolation can be applied without much modification to this type of uncertainty quantification problem. However, the use of interpolation techniques seems to be a challenging issue since it requires the parameterization of the prior models. We plan to study these issues in future.

In the area of uncertainty quantification in hydrology, we may be able to apply our techniques to models which are more complicated variants of Richards' equation. For instance one may wish to use complex models that take into account precipitation, solar radiation, air temperature, and vegetation.

REFERENCES

[1] J. E. Aarnes, Y. Efendiev, and L. Jiang, Analysis of multiscale finite element methods using global information for two–phase flow simulations, submitted to Computational Geosciences (2007).

[2] A. Apte, C. K. R. T. Jones, and A. M. Stuart, A Bayesian approach to Lagrangian data assimilation, in preparation (2007).

[3] I. Babuška and J. M. Melenk, The partition of unity method, International Journal for Numerical Methods in Engineering, 40 (1997), 727–758.

[4] V. Barthelmann, E. Novak, and K. Ritter, High dimensional polynomial interpolation on sparse grids, Advances in Computational Mathematics, 12 (4) (2000), 273–288.

[5] Y. Chen, L. J. Durlofsky, M. Gerritsen, and X. H. Wen, A coupled local–global upscaling approach for simulating flow in highly heterogeneous formations, Advances in Water Resources, 26 (2003), 1041–1060.

[6] A. Christen and C. Fox, MCMC using an approximation, Journal of Computational and Graphical Statistics, 14 (4) (2005), 795–810.

[7] Y. Chen and L. J. Durlofsky, An ensemble level upscaling approach for efficient estimation of fine–scale production statistics using coarse–scale simulations, SPE 106086, in Proceedings of the SPE Reservoir Simulation Symposium, Houston, TX, February 26 – February 28, 2007.

[8] P. Dostert, Y. Efendiev, T. Hou, and W. Luo, Coarse–gradient Langevin algorithms for dynamic data integration and uncertainty quantification, Journal of Computational Physics, 217 (1) (2006), 123–142.

[9] L. J. Durlofsky, Numerical calculation of equivalent grid block permeability tensors for heterogeneous porous media, Water Resources Research, 27 (5) (1991), 699–708.

[10] L. J. Durlofsky, Coarse scale models of two phase flow in heterogeneous reservoirs: Volume averaged equations and their relationship to the existing upscaling techniques, Computational Geosciences, 2 (2) (1998), 73–92.

[11] L. J. Durlofsky, R. A. Behrens, R. C. Jones, and A. Bernath, Scale up of heterogeneous three dimensional reservoir descriptions, SPE paper 30709 (1996).

[12] L. J. Durlofsky, R. C. Jones, and W. J. Milliken, A nonuniform coarsening approach for the scale up of displacement processes in heterogeneous media, Advances in Water Resources, 20 (5) (1997), 335–347.

[13] Y. Efendiev, A. Datta-Gupta, V. Ginting, X. Ma, and B. Mallick, An efficient two–stage Markov chain Monte Carlo method for dynamic data integration, Water Resources Research, 41 (12) (2005).

[14] Y. Efendiev and L. Durlofsky, Numerical modeling of subgrid heterogeneity in two phase flow simulations, Water Resources Research, 38(8) (2002), 1128–1138.

[15] Y. Efendiev, L. Durlofsky, and S. Lee, Modeling of subgrid effects in coarse–scale simulations of transport in heterogeneous porous media, Water Resources Research, 36(8) (2000), 2031–2041.

[16] Y. Efendiev, A. Datta-Gupta, X. Ma, and B. Mallick, A modified MCMC using streamline approach, submitted to Mathematical Geology (2005).

[17] Y. Efendiev, V. Ginting, T. Hou, and R. Ewing, Accurate multiscale finite element methods for two–phase flow simulations, Journal of Computational Physics,

220 (1) (2006), 155–174.

[18] Y. Efendiev, T. Hou and V. Ginting, Multiscale finite element methods for nonlinear problems and their applications, Communications in Mathematical Sciences, 2 (4) (2004), 553-589.

[19] Y. Efendiev, T. Hou and W. Luo, Preconditioning Markov chain Monte Carlo simulations using coarse–scale models, SIAM Journal on Scientific Computing, 28 (2) (2006), 776–803.

[20] Y. Efendiev, T. Hou, X. Wu, Convergence of a nonconformal multiscale finite element method, SIAM Journal on Numerical Analysis, 37 (3) (2000), 888–910.

[21] A. Gelman and D. B. Rubin , Inference from iterative simulation using multiple sequences, Statistical Science, 7 (1992), 457–511.

[22] V. Ginting, Analysis of two–scale finite volume element method for elliptic problem, Journal of Numerical Mathematics, 12(2) (2004), 119–142.

[23] V. Ginting, Computational upscaled modeling of heterogeneous porous media flow utilizing finite volume method, Ph.D. dissertation, Texas A&M University, College Station, TX, May 2004.

[24] U. Grenander and M. I. Miller, Representations of knowledge in complex systems (with discussion), Journal of the Royal Statistical Society Series B (Methodological), 56 (4) (1994), 549–603.

[25] R. Haverkamp, M. Vauclin, J. Touma, P. J. Wierenga, and G. Vachaud, A comparison of numerical solution models for one–dimensional infiltration, Soil Science Society of America Journal, 41 (1977), 285–294.

[26] L. Holden and B. F. Nielsen, Global upscaling of permeability in heterogeneous reservoirs: the output least squares (OLS) method, Transport in Porous Media, 40 (2000), 115–143.

[27] T. Y. Hou and X. H. Wu, A multiscale finite element method for elliptic problems in composite materials and porous media, Journal of Computational Physics, 134 (1) (1997), 169–189.

[28] S. Irmay, On the hydraulic conductivity of unsaturated soils, Transactions of the American Geophysical Union, 35 (1954), 463-467.

[29] P. Jenny, S. H. Lee, and H. Tchelepi, Multi–scale finite volume method for elliptic problems in subsurface flow simulation, Journal of Computational Physics, 187 (1) (2003), 47-67.

[30] J. S. Liu, Monte Carlo Strategies in Scientific Computing, Springer–Verlag, New York, 2001.

[31] M. Loève, Probability Theory, 4th Edition, Springer, Berlin, 1977.

[32] W. Luo, Wiener chaos expansion and numerical solutions of stochastic partial differential equations, Ph.D. dissertation, California Institute of Technology, Pasadena, CA, 2006.

[33] X. Ma, M. Al–Harbi, A. Datta–Gupta, and Y. Efendiev, A multistage sampling method for rapid quantification of uncertainty in history matching geological models, SPE paper 102476, SPE Annual Conference and Exhibition, San Antonio, TX, September 24–27, 2004.

[34] S. P. Meyn, R. L. Tweedie, Markov Chains and Stochastic Stability, Springer–Verlag, London, 1996.

[35] F. Nobile, R. Tempone, C. G. Webster, A sparse grid stochastic collocation method for elliptic partial differential equations with random input data, MOX Technical Report 85, Dipartimento di Matematica, Politecnico di Milano, submitted to SIAM Journal of Numerical Analysis (2006).

[36] D. Oliver, L. Cunha, and A. Reynolds, Markov chain Monte Carlo methods for conditioning a permeability field to pressure data, Mathematical Geology, 29 (1) (1997), 61–91.

[37] D. Oliver, N. He, and A. Reynolds, Conditioning permeability fields to pressure data, 5th European Conference on the Mathematics of Oil Recovery, Leoben, Austria, September 3–6, 1996.

[38] H. Owhadi and L. Zhang, Metric based up-scaling, Communications on Pure and Applied Mathematics (in press) (2007).

[39] L. A. Richards, Capillary conduction of liquids through porous mediums, Physics, 1 (5) (1931), 318–333.

[40] C. Robert and G. Casella, Monte Carlo Statistical Methods, Springer–Verlag, New York, 1999.

[41] S. A. Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of functions, Doklady Akademii Nauk SSSR, 4 (1963), 240-243.

[42] F. T. Tracy, Clean two– and three–dimensional analytical solutions of Richards' equation for testing numerical solvers, Water Resources Research, 42 (8) (2006).

[43] M. Th. van Genuchten, A closed–form equation for predicting the hydraulic conductivity of unsaturated soils, Soil Science Society of America Journal, 44 (1980), 892–898.

[44] A. W. Warrick, Time–dependent linearized infiltration: III. Strip and disc sources, Soil Science Society of America Journal, 40 (1976), 639–643.

[45] E. Wong, Stochastic Processes in Information and Dynamical Systems, MCGraw–Hill, New York, 1971.

[46] D. Xiu and J. Hesthaven, High-order collocation methods for differential equations with random inputs, SIAM Journal on Scientific Computing, 27 (3) (2005), 1118–1139.

VITA

Paul Francis Dostert was born in New Haven, Connecticut on March 4, 1978. He received his Bachelor of Science in Mathematics from James Madison University in May 2000. He then began studies at Texas A&M University where he received his Master of Science in Applied Mathematics in December 2002. He received his Ph.D. from the Department of Mathematics in December 2007. Paul Dostert can contacted by writing to: Department of Mathematics, The University of Arizona, 617 N. Santa Rita Ave., P.O. Box 210089, Tucson, AZ 85721-0089, or to the email address dosterpf@gmail.com.