MULTILEVEL UNCERTAINTY QUANTIFICATION TECHNIQUES USING MULTISCALE METHODS

A Dissertation

by

XIAOSI TAN

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

DOCTOR OF PHILOSOPHY

Chair of Committee,	Yalchin Efendiev
Committee Members,	Eduardo Gildin
	Raytcho Lazarov
	Jianxin Zhou
Head of Department,	Emil Straube

December 2015

Major Subject: Mathematics

ABSTRACT

In this dissertation, we focus on the uncertainty quantification problems in subsurface flow models which can be computationally demanding because of the large number of unknowns in forward simulations. First, we propose a general framework for the uncertainty quantification of quantities of interest for high-contrast single-phase flow problems. It is based on the Generalized Multiscale Finite Element Method (GMsFEM) and Multilevel Monte Carlo (MLMC) methods. The former provides a hierarchy of approximations at different resolutions, whereas the latter gives an efficient way to estimate quantities of interest using samples on different levels. By suitably choosing the number of samples at different levels, one can use less of expensive forward simulations on the fine grid, while more of inexpensive forward simulations on the coarse grid in Monte Carlo simulations. Further, we describe a Multilevel Markov Chain Monte Carlo (MLMCMC) method, which sequentially screens the proposal with different levels of approximations and reduces the number of evaluations required on the fine grid, while combining the samples at different levels to arrive at an accurate estimate. The framework seamlessly integrates the multiscale feature of the GMsFEM with the multilevel feature of the MLMC methods, and our numerical experiments illustrate its efficiency and accuracy in comparison with standard Monte Carlo estimates.

We also propose a multiscale space-parameter separation model reduction method for handling uncertainties in forward problems. The method is based on the idea of separation of variables. This involves seeking the solution in terms of an expansion, where each term is a separable function of space and parameter variables. To find

each term in the expansion, we solve a minimization problem associated with the forward problem. The minimization is performed successively for each term consisting of a separable function. In this proposed approach, we need to solve the PDE repeatedly, where we use GMsFEM to speed up the computation. We discuss how the GMsFEM can be used in this context and how the computational gain can be achieved. We present numerical results, which illustrate the efficiency and accuracy of our method.

We also discuss efficient sampling techniques for uncertainty quantification in inverse problems. In particular, we consider Approximate Bayesian computation (ABC) and develop a Multilevel Approximate Bayesian computation (MLABC) by using a hierarchy of forward simulation models within the MLMC framework. This approach improves the MLMCMC approach. In this part of the dissertation, we develop a mixed Generalized Multiscale Finite Element Method (GMsFEM) for solving parameter-dependent two-phase flow problems with transport model. A hierarchy of approximations at different resolutions can be provided by this mixed GMsFEM. ABC can be incorporated in different levels to reduce the computational cost, and to produce an approximate solution by ensembling at different levels.

ACKNOWLEDGEMENTS

The completion of my dissertation and subsequent Ph.D. has been a long journey. Much has happened and changed in the time I've been involved with this project. I would like to express my greatest appreciation to everyone who was with me during the entire or part of my life through Ph.D. study.

First and foremost, I would like to thank my advisor, Dr. Yalchin Efendiev, for his consistent guidance, support and all the research resources and opportunities he introduced to me, without which this dissertation would never have been accomplished.

I would like to thank Dr. Raytcho Lazarov, Dr. Eduardo Gildin and Dr. Jianxin Zhou, for not only serving on my committee, but also for their insightful and valuable suggestions and comments.

I would like to extend my sincere thanks to my collaborators, Dr. Longfei Gao, Dr. Richard Gibson, Dr. Nilabja Guha, Dr. Bangti Jin and Dr. Michael Presho. Thanks for their thoughtful guidance, patient help and discussions in our collaborations.

I would like to thank the professors and staff of the Department of Mathematics at Texas A&M University. In addition, I'm very grateful to the entire numerical analysis group for providing a great environment for research. I also want to thank Ms. Monique Stewart for her incredible patience and knowledge with the university administration.

I would like to thank my fellow graduate students, Dr. Yan Li, Dr. Jia Wei, Dr. Yuping Yang, Dr. Yang Qi, Dr. Fang Wang, Dr. Kainan Wang, Dr. Lihua Zuo, Dr. Guanglian Li and Yanfang Yang, not only for the discussions and shared thoughts that inspired my research, but also for their friendship and great help outside my

academic life. My gratitude is also extended to my friends, Yilin Song, Xi Cheng, Dizi Shi and Hyunae Lim, for being encouraging, for believing in me, and for the many precious memories along the way.

Last, but certainly not least, I would like to thank my parents, Jianxin Lu and Qun Tan, my grandmother, Tao Zhang, and everyone in my family, for their patience and encouragement throughout my graduate career.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Uncertainties exist inherently in subsurface flow problems in heterogeneous porous media. Mathematical models include uncertainties that come from the modeling errors, which reveal the differences between the models we choose and the real physical facts. For example, uncertainties in the description of subsurface lithofacies, porosity and permeability are major contributors to the uncertainties in reservoir performance forecasting, which can greatly affect the prediction of production and reservoir management. We will focus on this kind of uncertainties, which mainly come from the model parameters in our mathematical models. The parameters, such as permeability and porosity, are typically high-contrast with multiscale nature. The uncertainties associated with these quantities need to be quantified and reduced for better predictions. Our goal is to efficiently quantify and reduce these uncertainties, so that we can get a model with less uncertainties in the parameters, which can provide better predictions.

There are two main types of uncertainty quantification problems that we will consider in the dissertation. The first is the forward uncertainty propagation. The second is the inverse uncertainty quantification. The forward propagation of uncertainties is the quantification of uncertainties in the model outputs that are propagated from uncertain input parameters to the solution space. All possible parameter values that are consistent with the available measurement data and other known geological descriptions should be considered to study the propagation of the uncertainties through the model and how it affects the simulation results. Different approaches have been proposed to evaluate the quantities of interest of the model outputs. For instance, Monte Carlo sampling [15, 72, 32, 59] is one of the popular approaches,

which requires generating a set of independent realizations from the parameter space and solving the equations for each sample in the set. Based on obtained results from multiple realizations, one can infer statistical quantities, such as mean, variance, and so on. Other approaches, such as the response surface method [42, 5], Galerkin-type polynomial approximation methods [6, 24, 11] and spectral decomposition methods [69, 65, 62, 38], also require repeatedly solving a huge amount of deterministic PDEs with high dimensional parameter space. In approaches such as polynomial chaos approximation [66, 82, 84, 83, 37], multiscale high dimensional problems will impose heavy computational challenge due to a huge amount of degrees of freedom. Approaches such as POD [53] can achieve dimension reduction, but at the cost of computing a large space of global snapshots. More discussions can be found in [85, 23].

The inverse problem aims at estimating the discrepancy between experiment and mathematical model and the values of unknown parameters, given some experimental measurements and simulation results. In these problems, the observed data is used to estimate the input parameters and associated uncertainties. The calibrated parameters are then used to make predictions for other output variables in a forward modeling procedure. Bayesian framework based methods [73, 67, 60, 32, 25] are often used to solve the inverse problems. In this framework, the inverse problem is solved from a probabilistic point of view by seeking the posterior distribution of parameters given the observations using Bayes' theorem. For subsurface flow models, this task can be done by running a large number of trial simulations with different parameter values, hence is prohibitively expensive. More discussions can be found in [85, 23, 80, 78, 71, 47]. One of the prominent approaches for solving Bayesian inverse problem is Markov Chain Monte Carlo [73, 60]. There are many other approaches for solving inverse problems [12, 77, 50, 51]. In the dissertation, we will mostly focus

on Bayesian approaches.

From the above, we can see that although a number of approaches for both forward and inverse uncertainty quantification problems have been developed, there exist several difficulties and challenges, especially in the case of high-contrast input parameters with multiscale nature. The reservoir parameters are typically defined on a large number of grid blocks. This leads to a high computational cost when we try to resolve the system. For Monte Carlo simulations and Markov Chain Monte Carlo (MCMC) methods, forward model needs to be solved for each sample. A huge number of samples will be needed to get a proper evaluation of the mean, variance, or other statistics in the Monte Carlo simulations. Moreover, to guarantee the convergence of MCMC, the computational cost in the forward simulations is the most expensive part in the whole algorithm. Similarly, chaos expansion methods also involves multiple solves of the forward model. Different methods have been developed to accelerate the forward models, for example, multiscale finite element methods [45, 46, 49, 68, 43], etc. On the other hand, the low acceptance rate and local-trapping issues [56, 57, 58] of MCMC need to be overcome in the simulations, which also increase the cost of the forward simulation. Several modifications of MCMC have been proposed to improve these issues [32, 35, 58, 25, 60, 16, 41].

In the dissertation, we try to propose several efficient methods for both the forward and inverse uncertainty quantification for multiscale problems. Permeability fields of interest will be parametrized. To accelerate the forward simulation, we propose a Generalized Multiscale Finite Element Method (GMsFEM) [27, 26, 19] for stochastic PDEs. We consider both continuous Galerkin and mixed versions of GMsFEM. A mixed GMsFEM [19, 55] follows the mixed finite element framework [1, 3, 48] will be described for solving two-phase flow with transport model. This technique can be paired with Multilevel Monte Carlo (MLMC) [21, 39] methods and separation of

variables [54, 8], which forms computationally efficient framework for forward uncertainty quantification. We also combine the multilevel methods with sampling process like MCMC and Approximation Bayesian Computation (ABC) [79, 22, 13, 63, 64] to reach computational savings for inverse uncertainty quantification problems.

In Section 2, we cover the preliminary background materials. We introduce the porous media flow equations that will be studied throughout the dissertation. We will consider these equations for single- and two-phase flows. We then introduce the forward uncertainty quantification and Monte Carlo methods. Inverse uncertainty quantification is also introduced and the Bayesian framework is presented. Basic coarse-grid solution techniques to solve porous media flow equations and the parametrization of the parameters are also briefly discussed.

In Section 3, we present a framework for uncertainty quantification of quantities of interest based on the Generalized Multiscale Finite Element Method (GMsFEM) and Multilevel Monte Carlo (MLMC) methods. The GMsFEM provides a hierarchy of approximations to the solution, and the MLMC provides an efficient way to estimate quantities of interest using samples on multiple levels. In MLMC methods, a respective number of samples are used at different levels. More realizations are used at the coarser levels with inexpensive forward computations, and fewer samples are needed at the finer and more expensive levels due to the smaller variances. Therefore, the coupled GMsFEM-MLMC framework naturally integrates the multilevel feature of the MLMC with the multiscale nature of the high-contrast flow problem. MLMC is also considered jointly with Multilevel Markov Chain Monte Carlo (MLMCMC) methods, in which GMsFEM provides the mapping between the levels. We obtain samples from hierarchical posteriors corresponding to multilevel approximations which can be used for rapid computations within a MLMC framework to estimate the expected value. We apply the proposed methods to single phase flow

and numerical results show the efficiency of the framework in comparison with the standard MCMC estimates.

In Section 4, we propose an approach for forward uncertainty quantification that follows the idea of separation of variables, where separable basis functions are computed based on optimizations. By separating the parameter (that represents the uncertainties in the permeability field) and the space variables, the original high dimensional problem is decomposed into a sequence of low dimensional problems. The solution can be approached with a small number of terms in the expansion, which achieves a global model reduction. GMsFEM will be utilized here to solve the parameter-independent PDE in the expansion process repeatedly in order to speed up the computation, which achieves a local model reduction. Several numerical results are presented to show the accuracy of the solution achieved by the first few terms in this model reduction approach. Adaptivity is briefly discussed in the end.

In Section 5, we discuss the Approximate Bayesian computation (ABC) which can be viewed as an approximation of MCMC and can be utilized in inverse uncertainty quantification problems. In ABC, there is no burden of expensive likelihood computation and it is adaptable in many senarios. Here, ABC is implemented in a multiscale scenario, and the quantities of interest are estimated in a multilevel framework which follows the idea of MLMC. We also apply GMsFEM to provide hierarchical posteriors and mapping between levels. A multilevel ABC (MLABC) algorithm is proposed and numerical results are shown for both single-phase and two-phase flow models. We compare MLABC with standard MCMC and MLMCMC to show its computational efficiency. We discuss the convergence of MLABC in the end.

Lastly, in Section 6, we summarize our findings and present possibilities for future research.

2. BACKGROUND

2.1 Model problem

A significant part of the computational expense in uncertainty quantification problems is to solve the forward flow problem through multiscale geologic models repeatedly. In the dissertation, we will consider coupled flow and transport equations in heterogeneous porous media. In particular, we will study single- and two-phase flows, where single-phase flow can be considered as a special case of the two-phase flow systems. First, we briefly describe the two-phase flow equations. We consider two-phase flow in a subsurface formation (denoted by D) under the assumption that the displacement is dominated by viscous effects. The effects of gravity, compressibility and capillary pressure are neglected. The two phases are referred to as water and oil, designated by subscripts w and o, respectively. Darcy's law can be written as follows for each phase, with all quantities dimensionless:

$$v_j = -\frac{\kappa_{rj}(S)}{\nu_j}\kappa \cdot \nabla u, \tag{2.1}$$

where v_j, $j = w, o$, is the phase velocity, ν_j, $j = w, o$, is the phase viscosity, κ is the permeability tensor, κ_{rj} is the relative permeability to phase j, S is the water saturation (volume fraction) and u is the pressure. Combining Darcy's law with a statement of conservation of mass allows us to express the governing equations in terms of pressure and saturation equations:

$$\nabla \cdot (\lambda(S)\kappa\nabla u) = Q_s, \tag{2.2}$$

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

where $\lambda(S)$ is the total mobility, Q_s is the source term, v is the total velocity and $f_s(S)$ is the fraction flux of water, which are respectively given by:

$$\begin{aligned} \lambda(S) &= \frac{\kappa_{rw}(S)}{\nu_w} + \frac{\kappa_{ro}(S)}{\nu_o}, && (2.4) \\ v &= v_w + v_o = -\lambda(S)\kappa \cdot \nabla u, && (2.5) \\ f_s(S) &= \frac{\kappa_{rw}(S)/\nu_w}{\kappa_{rw}(S)/\nu_w + \kappa_{ro}(S)/\nu_o}. && (2.6) \end{aligned}$$

The above description is referred to as the fine model of the two-phase flow problem. For the single-phase flow, we have $\lambda(S) = 1$ and $f_s(S) = S$. Throughout, the porosity is assumed to be constant.

For some examples in this dissertation, we will focus on the uncertainty quantification of a single-phase, high-contrast flow model. Let $D \subset \mathbb{R}^d$ $(d = 2, 3)$ be an open bounded domain, with a boundary ∂D. The model equation reads:

$$-\nabla \cdot (\kappa(x, \mu)\nabla u) = f \quad \text{in} \quad D, \tag{2.7}$$

subject to suitable boundary conditions. Here f is the source term, u is the pressure within the medium, and $\kappa(x, \mu)$ is the heterogeneous spatial permeability field with multiple scales and high contrast, where μ represents the dependence on a multi-dimensional random parameter, typically resulting from a finite-dimensional noise assumption on the underlying stochastic process. $\kappa(x, \mu)$ can be parametrized by Karhunen-Loéve expansion [61, 81, 75] as:

$$\kappa(x, \mu) = \sum_{j=1}^{\infty} \sqrt{\lambda_j}\theta_j(\mu)\Phi_j(x), \tag{2.8}$$

in which $\theta_j(\mu)$'s are uncorrelated random variables and λ_j's are eigenvalues of the

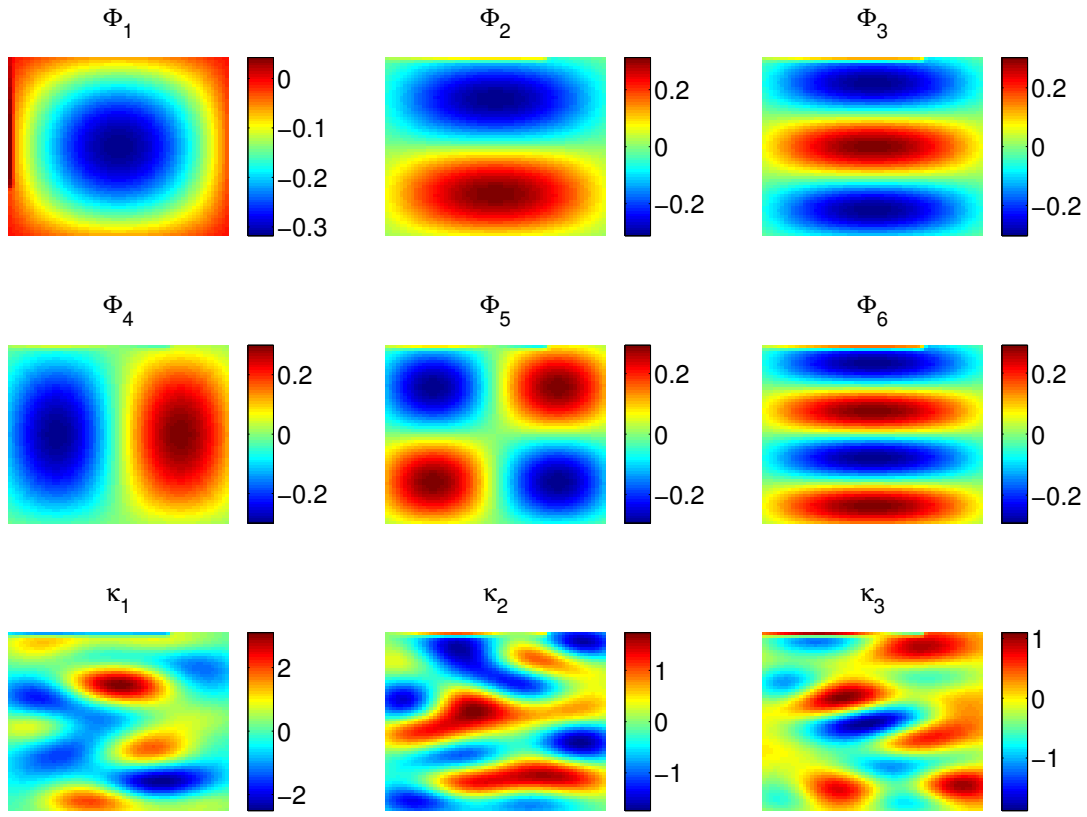


Figure 2.1: Top and middle row: The first 6 basis Φ_j used in the permeability parametrization (2.8). Bottom row: Examples of log-permeability field generated by (2.8).

covariance matrix (see discussions in Section 2.5). To demonstrate a permeability parameterization, in Figure 2.1, we show the first 6 basis (which we will commonly have, see later sections for details) we use in the permeability parametrization and some examples of log of κ's. More details on permeability parametrization can be found in Section 2.5.

2.2 Monte Carlo methods for forward problem

As introduced in Section 1, forward uncertainty quantification is a process to study how the parameter uncertainties (such as permeability field) propagate through

the model and affect the simulation results. In our model problems (2.3) and (2.7), the parameter $\kappa(x, \mu)$ is expressed as a stochastic process, which can be characterized by the corresponding probability distribution. With stochastic parameters, the output of the forward model also becomes random with uncertainties that propagate from the inputs. Hence, the target of forward uncertainty quantification is to reduce the uncertainties and make reliable predictions.

Various methods for forward uncertainty quantification are mentioned in Section 1. Among these methods, Monte Carlo simulation is one of the most straightforward methods for solving stochastic equations. This widely used approach is conceptually simple and is based on the idea of approximating stochastic process by a large number of equally probable realizations. In the following we will briefly introduce the Monte Carlo method and show a simple example of forward uncertainty quantification.

Let $X(\mu)$ be a random variable. We are interested in the efficient computation of the expected value of X, denoted by $\mathbb{E}[X]$. In our calculations, X is a function of the permeability field κ, e.g., the solution to (2.7) evaluated at measurement points. To compute an approximation to $\mathbb{E}[X]$ by Monte Carlo method, one first generates a number M of independent realizations of the random variable X, denoted by $\{X^m\}_{m=1}^M$, and then approximates the expected value $\mathbb{E}[X]$ by the arithmetic mean

$$E_M(X) := \frac{1}{M} \sum_{m=1}^{M} X^m.$$

Now we define the Monte Carlo integration error $e_M(X)$ by

$$e_M(X) = \mathbb{E}[X] - E_M(X).$$

Then the central limit theorem asserts that for large M, the Monte Carlo integration

error

$$e_M(X) \sim \mathrm{Var}[X]^{1/2} M^{-1/2} \nu, \tag{2.9}$$

where ν is a standard normal random variable, and $\mathrm{Var}[X]$ is the variance of X. Hence the error $e_M(X)$ in Monte Carlo integration is of order $O(M^{-1/2})$ with a constant depending only on the variance $\mathrm{Var}[X]$ of the integrand X [73].

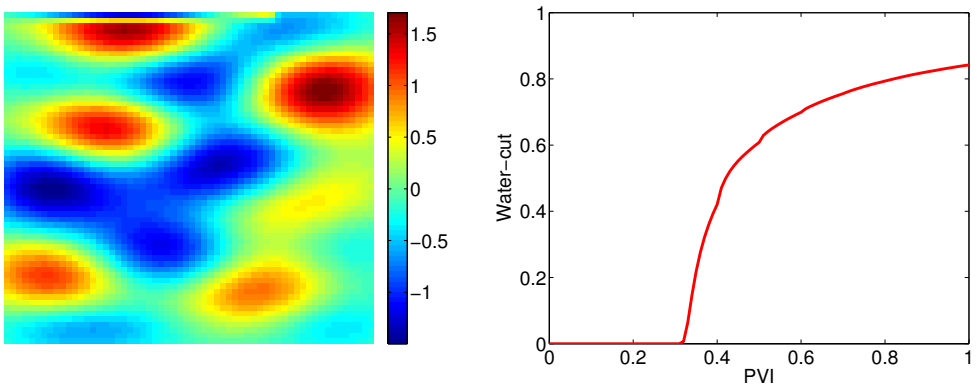
For example, in a two-phase water-oil flow problem modeled as (2.3), the permeability κ is the input parameter with uncertainties. Various production related information can be evaluated by solving the forward problem. In later sections of this dissertation, these outputs of the model problem that we consider will be denoted by F. Both pressure field and water-cut will be used in this dissertation. Water-cut means the fraction of water produced in relation to the total production rate. We compute the saturation S at different pore volume injected (PVI). PVI represents dimensionless time and is computed via

$$\mathrm{PVI} = \int \frac{Q}{V_p} dt,$$

where V_p is the total pore volume of the system, $Q = \int_{\partial D^{out}} v \cdot n ds$ is the total flow rate and ∂D^{out} is the outflow boundary. The water-cut (denoted by $F(t)$ in this section) for a two-phase water-oil flow is defined as the fraction of water in the produced fluid and is given by q_w/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, i.e.,

$$F(t) = \frac{\displaystyle\int_{\partial D^{out}} v_n f(S) dl}{\displaystyle\int_{\partial D^{out}} v_n dl}, \tag{2.10}$$

where ∂D^{out} is the outflow boundaries and v_n is normal velocity field.



(a) $\log(\kappa)$. (b) Water-cut.

Figure 2.2: An example of a permeability realization and the corresponding output water-cut by solving (2.3).

In Figure 2.2, we show an example of permeability field and the corresponding water-cut output in model (2.3). In this example, κ is assumed to be a log-Gaussian field which is parametrized as (2.8). Following the Monte Carlo method, we can generate multiple realizations of κ and take the sample mean of the output water-cuts to be the estimator.

2.3 Uncertainty quantification inverse problem

The observations of flow behavior is another important source of information which helps improve the estimation of parameters as well as the credibility of predictions. The uncertainty quantification inverse problem is a process which infers the input parameters from the observations on some of the output variables. The observed data is denoted by F_{obs}. For example, in model problem (2.7), we can do uncertainty quantification by sampling the permeability field conditioned on the observed pressure data F_{obs}. The pressure is an integrated response, and the map from the pressure to the permeability field is not one-to-one. So there may exist many different permeability realizations that equally reproduce the given pressure data

F_{obs}. In practice, the measured pressure data F_{obs} inevitably contains measurement errors. For a given permeability field κ, we denote the pressure as $F(\kappa)$, which can be computed by solving the model equation (2.7) on the fine grids. The computed pressure $F(\kappa)$ will contain also modeling error, which induces an additional source of errors, apart from the inevitable measurement error. By assuming the combined error as a random variable ϵ we can write the model as

$$F_{obs} = F(\kappa) + \epsilon. \tag{2.11}$$

For simplicity, the noise ϵ will be assumed to follow a normal distribution $\mathcal{N}(0, \sigma_f^2 I)$, i.e., the likelihood $p(F_{obs}|\kappa)$ is assumed be of the form

$$p(F_{obs}|\kappa) \propto exp\left(-\frac{\|F(\kappa) - F_{obs}\|^2}{2\sigma_f^2}\right).$$

We will represent the permeability field κ through the now classical Karhunen-Loéve expansion (KLE) [61], which we describe in more details in Section 2.5. Our goal is to generate permeability fields κ consistent with the observed pressure data F_{obs}. Let the permeability field κ be parameterized by $\theta_1, \theta_2, \cdots, \theta_n$ completely, and we assume that $\theta_1, \theta_2, \cdots, \theta_n$ are independent and identically distributed (i.i.d.). By Bayes' theorem the posterior distribution can be written as

$$\begin{aligned}\pi(k) = p(\kappa|F_{obs}) &\propto p(F_{obs}|\kappa)p(\kappa) \\ &= p(F_{obs}|\kappa)p(\theta_1, \theta_2, \cdots, \theta_n) \\ &= p(F_{obs}|\kappa)p(\theta_1)p(\theta_2)\cdots p(\theta_n).\end{aligned}$$

In the expression for the posterior distribution $\pi(\kappa)$, $p(F_{obs}|\kappa)$ is the likelihood function, incorporating the information in the data F_{obs}, and $p(\kappa)$ is the prior. Further,

we may also incorporate other prior information, e.g., that the permeability field κ is known at some spatial locations corresponding to the wells.

2.4 Coarse-grid solution techniques

Both forward and inverse uncertainty quantification problems involve solving the model flow problem for the quantities of interest repeatedly, which requires large computational cost due to the high dimensionality in spatial space. To reduce the computational cost of the fine model, coarse-grid solution techniques such as Multiscale Finite Element Method (MsFEM) are utilized to solve the flow problem on a coarse grid. As for the coarse-scale model, we will consider single-phase flow based multiscale simulation methods. In this technique, multiscale basis functions are calculated. These basis functions are coupled through a variational formulation of the problem [27, 31, 36, 33, 28, 30, 4]. For multi-phase flow and transport simulations, the conservative fine-scale velocity is often needed. For this reason, the mixed MsFEM is used. In this section, we will have an overview of the coarse-grid solution technique for the model problem (2.7).

To discretize the model equation (2.7), we first introduce the notion of fine and coarse grids. Let $\mathcal{T}^H$ be a conforming triangulation of the computational domain D into finite elements (triangles, quadrilaterals, tetrahedra, etc.). We refer to this partition as the coarse grid and assume that each coarse subregion is further partitioned into a connected union of fine grid blocks. The fine grid partition will be denoted by $\mathcal{T}^h$. We use $\{x_i\}_{i=1}^{N_v}$ (where N_v is the number of coarse nodes) to denote the vertices of the coarse mesh $\mathcal{T}^H$, and define the neighborhood ω_i of the node x_i by

$$\omega_i = \bigcup \{K_j \in \mathcal{T}^H; \quad x_i \in \overline{K}_j\}. \tag{2.12}$$

See Figure 2.3 for an illustration of neighborhoods and elements subordinated to the

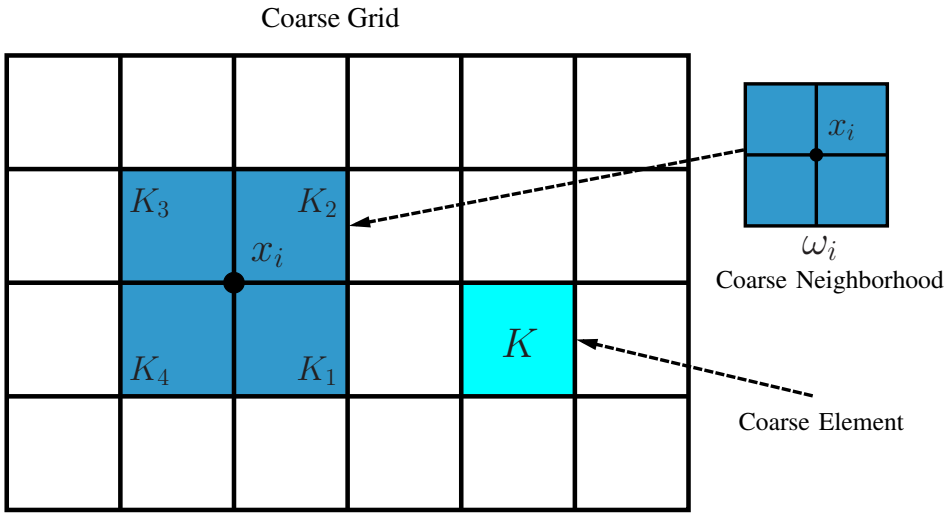


Figure 2.3: Illustration of a coarse neighborhood and coarse elements.

coarse discretization.

Our objective is to seek the multiscale basis functions for each node x_i that represent the spatial heterogeneities and uncertainties. We denote the multiscale basis function corresponding to the node x_i as ψ_i^j, and assume that the basis functions are supported in ω_i. As in standard finite element method, once multiscale basis functions are constructed (see Figure 2.4 for illustration), we will look for $u_0 = \sum_{ij} c_{ij}\psi_i^j$, where c_{ij} are determined from

$$a(u_0, v) = f(v), \quad \text{for all } v \in V_0,$$

in which $V_0 = \text{span}(\psi_i^j)$,

$$a(u, v) = \int_D \kappa(x)\nabla u(x)\nabla v(x)dx \quad \text{for all } u, v \in H_0^1(D),$$

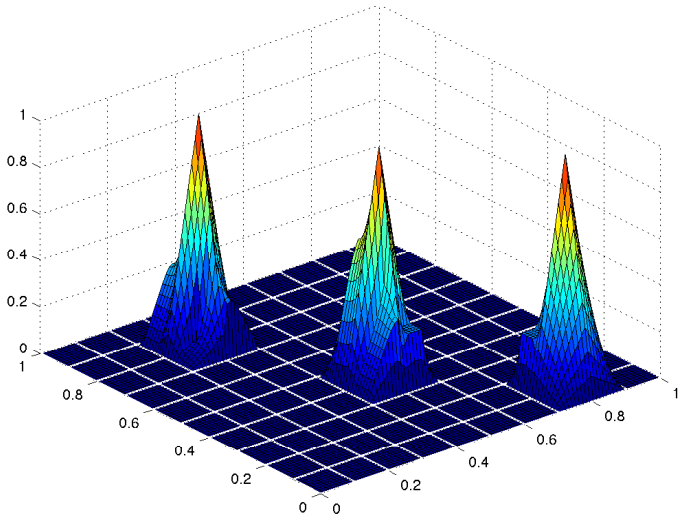


Figure 2.4: Illustration of some multiscale basis functions.

and

$$f(v) = \int_D f(x)v(x)dx \quad \text{for all } v \in H_0^1(D).$$

A GMsFEM for stochastic problems will be introduced later in Section 3 which has a similar structure. The main difference between the two approaches is that we systematically enrich coarse spaces in GMsFEM and generalize it by considering an input space consisting of parameters and source terms. This method can be used to solve parameter-dependent problems with computational saving, while providing a hierarchy of solutions.

2.5 Permeability parametrization

To obtain a permeability field in terms of an optimal L^2 basis, we use the Karhunen-Loève expansion (KLE) [61, 81, 75]. For our numerical tests, we truncate the expansion and represent the permeability matrix by a finite number of random parameters. We consider the random field $Y(x, \omega) = \log[\kappa(x, \mu)]$, where μ repre-

sents randomness. We assume a zero mean $\mathbb{E}[Y(x,\mu)] = 0$, with a known covariance operator $R(x,y) = \mathbb{E}\left[Y(x)Y(y)\right]$. Then we expand the random field $Y(x,\mu)$ as

$$Y(x,\mu) = \sum_{k=1}^{\infty} Y_k(\mu)\Phi_k(x),$$

with

$$Y_k(\mu) = \int_D Y(x,\mu)\Phi_k(x)dx.$$

The functions $\{\Phi_k(x)\}$ are eigenvectors of the covariance operator $R(x,y)$, and form a complete orthonormal basis in $L^2(D)$, i.e.,

$$\int_D R(x,y)\Phi_k(y)dy = \lambda_k\Phi_k(x), \qquad k = 1,2,\ldots, \tag{2.13}$$

where $\lambda_k = \mathbb{E}[Y_k^2] > 0$. We note that $\mathbb{E}[Y_iY_j] = 0$ for all $i \neq j$. By denoting $\theta_k = Y_k/\sqrt{\lambda_k}$ (whence $\mathbb{E}[\theta_k] = 0$ and $\mathbb{E}[\theta_i\theta_j] = \delta_{ij}$), we have

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

where Φ_k and λ_k satisfy (2.13). The randomness is represented by the scalar random variables θ_k. After discretizing the domain D into a rectangular mesh, we truncate the KLE (2.14) to a finite number of terms. In other words, we keep only the leading-order terms (quantified by the magnitude of λ_k), and capture most of the energy of the stochastic process $Y(x,\mu)$. An example of descending eigenvalues in a KLE is shown in Figure 2.5. For an N-term KLE approximation

$$Y_N = \sum_{k=1}^{N} \sqrt{\lambda_k}\theta_k\Phi_k,$$

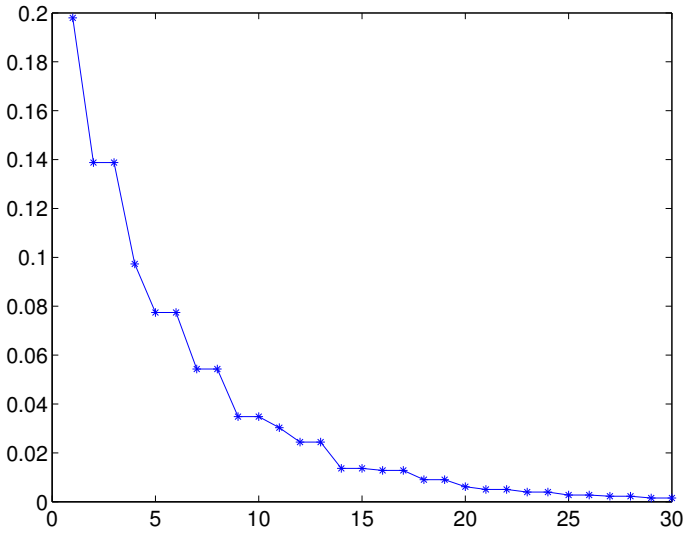


Figure 2.5: Example of descending eigenvalues in the KL expansion.

the energy ratio of the approximation is defined by

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

If the eigenvalues $\{\lambda_k\}$ decay very fast, then the truncated KLE with the first few terms would be a good approximation of the stochastic process $Y(x, \mu)$ in the L^2 sense.

3. MULTILEVEL MARKOV CHAIN MONTE CARLO [*]

3.1 Introduction

In this section, we present a framework for uncertainty quantification of quantities of interest based on the Generalized Multiscale Finite Element Method (GMsFEM) and Multilevel Monte Carlo (MLMC) methods. The GMsFEM provides a hierarchy of approximations to the solution, and the MLMC provides an efficient way to estimate quantities of interest using samples on multiple levels. Therefore, the framework naturally integrates the multilevel feature of the MLMC with the multiscale nature of the high-contrast flow problem.

GMsFEM is a generalized framework that follows the Multiscale Finite Element Method (MsFEM) which is a class of coarse-grid solution techniques introduced in Section 2.4. The GMsFEM achieves efficiency via coarse space enrichment, which is split into two stages, following an offline-online procedure (see also [8, 14, 68, 74]). At the first stage of the computation, a larger-dimensional (relative to the online space) parameter-independent offline space is formed. The offline space accounts for a suitable range of parameter values that may be used in the online stage, and constitutes a one-time preprocessing step. The offline space is created by first generating a set of "snapshots" in which a number of localized problems are solved on each coarse subdomain for a number of parameter values. The offline space is then obtained through solving localized eigenvalue problems that use averaged parameter quantities within the space of snapshots. A number of eigenfunctions are kept in order to form the offline space. At the online stage, we solve analogous eigenvalue problems

[*]Reprinted with permission from "Multilevel Markov chain Monte Carlo method for high-contrast single-phase flow problems" by Y. Efendiev, B. Jin, P. Michael and X. Tan, *Communications in Computational Physics*, 17(01):259-286, 2015. Copyright 2015 by Global Science Press.

using a fixed parameter value within the offline space to form a reduced-order online space. A notable advantage of the GMsFEM construction is that the flexible coarse space dimension naturally provides a hierarchy of approximations to be used within the MLMC framework. Further, we avoid unnecessary large-dimensional eigenvalue computations for each parameter realization.

The main idea of Multilevel Monte Carlo (MLMC) [44, 39, 40, 9, 21] is to use a respective number of samples at different levels to compute the expected values of quantities of interest. In these techniques, more realizations are used at the coarser levels with inexpensive forward computations, and fewer samples are needed at the finer and more expensive levels due to the smaller variances. By suitably choosing the number of realizations at each level, one can obtain a multilevel estimate of the expected values at much reduced computational efforts.

In this work, we couple the GMsFEM with the MLMC methods to arrive at a general framework for the uncertainty quantification of the quantities of interest in multiscale flow problems. Specifically, we take the dimension of the multiscale space to be the MLMC level, where the accuracy of the global coarse-grid simulations depends on the dimension of the multiscale coarse space. The convergence with respect to the coarse space dimension plays a key role in selecting the number of samples at each level of MLMC. We take different numbers of online basis functions to generate the multiscale coarse spaces, running more forward coarse-grid simulations with the smaller dimensional multiscale spaces and fewer simulations with larger dimensional multiscale spaces. By combining these simulation results in a MLMC framework one can achieve better accuracy at the same cost as the classical Monte Carlo (MC) method.

Further, we will consider the use of MLMC jointly with Multilevel Markov Chain Monte Carlo (MLMCMC) methods following [52]. The main idea of MLMCMC ap-

proach is to condition the quantities of interest at one level (e.g., at a finer level) to that at another level (e.g., at a coarser level). The multiscale model reduction framework provides the mapping between the levels, and it can be used to estimate the expected value. Specifically, for each proposal, we run the simulations at different levels to screen the proposal and accept it conditionally at these levels. In this manner, we obtain samples from hierarchical posteriors corresponding to our multilevel approximations which can be used for rapid computations within a MLMC framework.

The rest of this section is organized as follows. In Section 3.2, we describe the two-stage procedure of the GMsFEM for high-contrast single-phase flow problems. We shall discuss the offline and online computations in detail. In Section 3.3 we discuss the idea of Multilevel Monte Carlo methods, and also the crucial issue of complexity analysis. The algorithm for coupling the GMsFEM with the MLMC is described. Then, in Section 3.4, we describe a Multilevel Markov Chain Monte Carlo method for generating samples from hierarchical posterior distributions, which can be used in the MLMC framework. A preliminary analysis of the convergence of the MLMCMC algorithm is also provided. In Section 3.5, we present numerical examples to illustrate the efficiency of the framework, in comparison with the standard MCMC estimates.

3.2 GMsFEM for stochastic problems

In this section we describe the offline-online computational procedure for the efficient construction of GMsFEM coarse spaces. To discretize the model equation (2.7), the notion of fine and coarse grids is introduced in Section 2.4. The GMsFEM has a structure similar to MsFEM mentioned in Section 2.4, the main difference is that we enrich coarse spaces in GMsFEM and generalize it by considering an input

space consisting of parameters and source terms. At the offline stage, one generates the snapshot set, and constructs a low-dimensional offline space by model reduction. At the online stage, for each input parameter μ, one first computes multiscale basis functions and then solves for a coarse-grid problem for any force term and boundary condition. Below we describe the offline and online procedures in more detail.

3.2.1 Offline computation

At the offline stage, we first construct a snapshot space $V_{\text{snap}}^{\omega_i}$ on each coarse neighborhood ω_i in the domain (cf. Figure 2.3). The construction involves solving a set of localized problems for various choices of input parameters. Specifically, we solve the following eigenvalue problems on each ω_i:

$$A(\mu_j)\psi_{l,j}^{\omega_i,\text{snap}} = \lambda_{l,j}^{\omega_i,\text{snap}} S(\mu_j)\psi_{l,j}^{\omega_i,\text{snap}} \quad \text{in } \omega_i, \tag{3.1}$$

where $\{\mu_j\}_{j=1}^{J}$ is a set of parameter values to be specified. Here we consider only Neumann boundary conditions, but other boundary conditions are also possible. The matrices $A(\mu_j)$ and $S(\mu_j)$ in (3.1) are respectively defined by

$$\begin{aligned} A(\mu_j) &= [a(\mu_j)_{mn}] = \int_{\omega_i} \kappa(x,\mu_j)\nabla\phi_n \cdot \nabla\phi_m dx, \\ S(\mu_j) &= [s(\mu_j)_{mn}] = \int_{\omega_i} \widetilde{\kappa}(x,\mu_j)\phi_n\phi_m dx, \end{aligned} \tag{3.2}$$

where ϕ_n denotes the standard bilinear, fine-scale basis functions and $\widetilde{k}$ will be described below, cf. (3.5). We note that (3.1) is the discrete counterpart of the continuous Neumann eigenvalue problem

$$-\text{div}(\kappa(x,\mu_j)\nabla\psi_{l,j}^{\omega_i,\text{snap}}) = \lambda_{l,j}^{\omega_i,\text{snap}}\widetilde{\kappa}(x;\mu_j)\psi_{l,j}^{\omega_i,\text{snap}} \quad \text{in } \omega_i.$$

For notational simplicity, we omit the superscript ω_i. For each ω_i, we keep the first L_i eigenfunctions of (3.1) corresponding to the lowest eigenvalues to form the snapshot space

$$V_{\text{snap}} = \text{span}\{\psi_{l,j}^{\text{snap}} : \quad 1 \leq j \leq J,\ 1 \leq l \leq L_i\}.$$

We then stack the snapshot functions into a matrix

$$R_{\text{snap}} = \left[\psi_1^{\text{snap}}, \ldots, \psi_{M_{\text{snap}}}^{\text{snap}}\right],$$

where $M_{\text{snap}} = J \times L_i$ denotes the total number of snapshots used in the construction.

Next we construct the offline space $V_{\text{off}}^{\omega_i}$, which will be used to efficiently (and accurately) construct a set of multiscale basis functions for each μ value at the online stage. To this end, we perform a dimensionality reduction of the snapshot space using an auxiliary spectral decomposition. Specifically, we seek a subspace of the snapshot space such that it can approximate any element of the snapshot space in a suitable sense. The analysis in [29] motivates the following eigenvalue problem in the space of snapshots:

$$A^{\text{off}}\Psi_k^{\text{off}} = \lambda_k^{\text{off}} S^{\text{off}}\Psi_k^{\text{off}}, \tag{3.3}$$

where the matrices A^{off} and S^{off} are defined by

$$\begin{aligned} A^{\text{off}} = [a_{mn}^{\text{off}}] &= \int_{\omega_i} \overline{\kappa}(x,\mu)\nabla\psi_m^{\text{snap}} \cdot \nabla\psi_n^{\text{snap}} dx = R_{\text{snap}}^T \overline{A} R_{\text{snap}}, \\ S^{\text{off}} = [s_{mn}^{\text{off}}] &= \int_{\omega_i} \overline{\overline{\kappa}}(x,\mu)\psi_m^{\text{snap}}\psi_n^{\text{snap}} dx = R_{\text{snap}}^T \overline{S} R_{\text{snap}}, \end{aligned}$$

respectively. Here $\overline{\kappa}(x,\mu)$ and $\overline{\overline{\kappa}}(x,\mu)$ are domain-based, parameter-averaged coefficients, and $\overline{A}$ and $\overline{S}$ denote fine scale matrices for the averaged coefficients. To generate the offline space, we choose the smallest M_{off} eigenvalues to (3.3), and take

the corresponding eigenvectors in the space of snapshots by setting

$$\psi_k^{\text{off}} = \sum_j \Psi_{kj}^{\text{off}} \psi_j^{\text{snap}}, \quad k = 1, \ldots, M_{\text{off}}$$

to form the reduced snapshot space, where Ψ_{kj}^{off} are the coordinates of the vector Ψ_k^{off}. We then create the offline matrix

$$R_{\text{off}} = \left[\psi_1^{\text{off}}, \ldots, \psi_{M_{\text{off}}}^{\text{off}}\right]$$

to be used in the online computation.

Remark 3.2.1. *At the offline stage the bilinear forms in* (3.3) *are chosen to be parameter-independent, such that there is no need to construct the offline space for each μ value.*

3.2.2 Online computation

Next for a given input parameter μ value, we construct the associated online coarse space $V_{\text{on}}^{\omega_i}(\mu)$ on each coarse subdomain ω_i. In principle, we want this to be a low-dimensional subspace of the offline space for computational efficiency. The online coarse space will be used by the continuous Galerkin finite element method for solving the original global problem. In particular, we seek a subspace of the offline space such that it can approximate any element of the offline space in an appropriate sense. We note that at the online stage, the bilinear forms are parameter-dependent. The analysis in [29] motivates the following eigenvalue problem in the offline space:

$$A^{\text{on}}(\mu)\Psi_k^{\text{on}} = \lambda_k^{\text{on}} S^{\text{on}}(\mu)\Psi_k^{\text{on}}, \tag{3.4}$$

where the matrices $A^{\text{on}}(\mu)$ and $S^{\text{on}}(\mu)$ are defined by

$$A^{\text{on}}(\mu) = [a^{\text{on}}(\mu)_{mn}] = \int_{\omega_i} \kappa(x,\mu)\nabla\psi_m^{\text{off}} \cdot \nabla\psi_n^{\text{off}} dx = R_{\text{off}}^T A(\mu) R_{\text{off}},$$
$$S^{\text{on}}(\mu) = [s^{\text{on}}(\mu)_{mn}] = \int_{\omega_i} \widetilde{\kappa}(x,\mu)\psi_m^{\text{off}}\psi_n^{\text{off}} dx = R_{\text{off}}^T S(\mu) R_{\text{off}},$$

respectively. Note that $\kappa(x,\mu)$ and $\widetilde{\kappa}(x,\mu)$ are now parameter-dependent. To generate the online space, we choose the eigenvectors corresponding to the smallest M_{on} eigenvalues of (3.4), and set

$$\psi_k^{\text{on}} = \sum_j \Psi_{kj}^{\text{on}} \psi_j^{\text{off}}, \quad k = 1, \ldots, M_{\text{on}},$$

where Ψ_{kj}^{on} are the coordinates of the vector Ψ_k^{on}.

Remark 3.2.2 (Adaptivity in the parameter space)**.** *We note that one can use adaptivity in the parameter space to avoid computing the offline space for a large range of parameters and compute the offline space only for a short range of parameters and update the space. To demonstrate this concept, we assume that the parameter space Λ can partitioned into a number of smaller parameter spaces Λ_i, $\Lambda = \bigcup_i \Lambda_i$, where Λ_i may overlap with each other. Furthermore, the offline spaces are constructed for each Λ_i. In the online stage, depending on the online value of the parameter, we can decide which offline space to use. This reduces the computational cost at the online stage. In many applications, e.g., in nonlinear problems, one may remain in one of Λ_i's for many iterations and thus use the same offline space to construct the online space. Moreover, one can also adaptively add multiscale basis functions at the online stage using error estimators. This is a subject of our future research.*

3.2.3 Global coupling mechanism

To incorporate the online basis functions into a reduced-order global formulation of the original problem (2.7), we begin with an initial coarse space $V^{\text{init}}(\mu) = \text{span}\{\chi_i\}_{i=1}^{N_v}$ (N_v denotes the number of coarse nodes). The functions χ_i are standard multiscale partition of unity functions defined by

$$\begin{aligned} -\text{div}\big(\kappa(x;\mu)\,\nabla\chi_i\big) = 0 \quad & K \in \omega_i, \\ \chi_i = g_i \quad & \text{on } \partial K, \end{aligned}$$

for each coarse element $K \in \omega_i$, where g_i is a bilinear boundary condition. Next we define the summed, pointwise energy $\widetilde{\kappa}$ as, cf. (3.2),

$$\widetilde{\kappa} = \kappa \sum_{i=1}^{N_v} H^2 |\nabla\chi_i|^2, \tag{3.5}$$

where H is the coarse mesh size. In order to construct the global coarse grid solution space we multiply the partition of unity functions χ_i by the online eigenfunctions $\psi_k^{\omega_i,\text{on}}$ from the space $V_{\text{on}}^{\omega_i}(\mu)$ to form the basis functions

$$\psi_{i,k} = \chi_i \psi_k^{\omega_i,\text{on}} \quad \text{for } 1 \le i \le N_v \text{ and } 1 \le k \le M_{\text{on}}^{\omega_i}, \tag{3.6}$$

where we recall that $M_{\text{on}}^{\omega_i}$ denotes the number of online basis functions kept for each ω_i. The basis constructed in (3.6) is then used within a global continuous Galerkin formulation. Now we define the online spectral multiscale space as

$$V_{\text{on}}(\mu) = \text{span}\{\psi_{i,k} : 1 \le i \le N_v,\ 1 \le k \le M_{\text{on}}^{\omega_i}\}, \tag{3.7}$$

and using a single index notation, we write $V_{\text{on}}(\mu) = \text{span}\{\psi_i\}_{i=1}^{N_c}$ where N_c denotes the total number of basis functions in the coarse scale formulation. Using the online basis functions, we define the operator matrix $R = [\psi_1, \ldots, \psi_{N_c}]$, where ψ_i represents the vector of nodal values of each basis function defined on the fine grid. To solve (2.7) we seek $u(x, \mu) = \sum_i u_i \psi_i(x, \mu) \in V_{\text{on}}$ such that

$$\int_D \kappa(x, \mu) \nabla u \cdot \nabla v dx = \int_D f \, v dx \text{ for all } v \in V_{\text{on}}. \tag{3.8}$$

The above equation yields the discrete form

$$A(\mu)u = F, \tag{3.9}$$

where

$$A(\mu) := [a_{IJ}] = \int_D \kappa(x, \mu) \nabla \psi_I \cdot \nabla \psi_J dx$$

is a coarse stiffness matrix,

$$F := [f_I] = \int_D f \, \psi_I dx$$

is the coarse forcing vector, P_c denotes the vector of unknown pressure values, and ψ_I denotes the coarse basis functions that span V_{on}. We note that the coarse system may be rewritten using the fine-scale system and the operator matrix R. In particular, we may write $A(\mu) = R^T A^f(\mu) R$ and $F = R^T F^f$, where

$$\begin{aligned} A^f(\mu) &:= [a_{ij}] = \int_D \kappa(x; \mu) \nabla \phi_i \cdot \nabla \phi_j dx, \\ F^f &:= [f_i] = \int_D f \, \phi_i dx, \end{aligned}$$

and ϕ_i are the fine-scale bilinear basis functions. Analogously, the operator matrix R may be used to map coarse scale solutions back to the fine grid.

3.3 Multilevel Monte Carlo methods

As was mentioned earlier, one standard approach for exploring posterior distributions is the Monte Carlo method, especially Markov Chain Monte Carlo (MCMC) methods. Here, generating each sample requires the solution of the forward model, which is unfortunately very expensive for many practical problems defined by partial differential equations, including high-contrast flows. Therefore, it is imperative to reduce the computational cost of the sampling step. We shall couple the Multilevel Monte Carlo with the multiscale forward solvers to arrive at a general framework for uncertainty quantification of high-contrast flows.

3.3.1 MLMC-GMsFEM framework

The MLMC approach was first introduced by Heinrich in [44] for finite- and infinite-dimensional integration. Later on, it was applied to stochastic ODEs by Giles [40, 39]. More recently, it has been used for PDEs with stochastic coefficients [9, 21]. We now briefly introduce the MLMC approach in a general context, and derive our MLMC-GMsFEM framework for uncertainty quantification.

Let $X(\omega)$ be a random variable. In Section 2.2, Monte Carlo (MC) method is introduced to approximate the quantity of interest, $X(\omega)$, in forward uncertainty quantification. In this work, we are interested in MLMC methods. The idea is to compute the quantity of interest $X = X_L$ using the information on several different levels. Here we couple the MLMC with the GMsFEM, where the level is identified with the size of the online space. We assume that L is the level of interest, and computing many realizations at this level is very expensive. Hence we introduce levels smaller than L, namely $L-1, \ldots, 1$, and assume that the lower the level is,

the cheaper the computation of X_l is, and the less accurate X_l is with respect to X_L. By setting $X_0 = 0$, we decompose X_L into

$$X_L = \sum_{l=1}^{L} (X_l - X_{l-1}) .$$

The standard MC approach works with M realizations of the random variable X_L at the level of interest L. In contrast, within the MLMC approach, we work with M_l realizations of X_l at each level l, with $M_1 \geq M_2 \geq \cdots \geq M_L$. We write

$$\mathbb{E}[X_L] = \sum_{l=1}^{L} \mathbb{E}[X_l - X_{l-1}] ,$$

and next approximate $\mathbb{E}[X_l - X_{l-1}]$ by an empirical mean:

$$\mathbb{E}[X_l - X_{l-1}] \approx E_{M_l}(X_l - X_{l-1}) = \frac{1}{M_l} \sum_{m=1}^{M_l} \left(X_l^m - X_{l-1}^m \right) , \tag{3.10}$$

where X_l^m is the m-th realization of the random variable X computed at the level l (note that we have M_l copies of X_l and X_{l-1}, since $M_l \leq M_{l-1}$). Then the MLMC approach approximates $\mathbb{E}[X_L]$ by

$$E^L(X_L) := \sum_{l=1}^{L} E_{M_l} (X_l - X_{l-1}) . \tag{3.11}$$

We note that the realizations of X_l used with those of X_{l-1} to evaluate $E_{M_l}(X_l - X_{l-1})$ do not have to be independent of the realizations of X_l used with those of X_{l+1} to evaluate $E_{M_{l+1}}(X_{l+1} - X_l)$. In our context, the permeability field samples used for computing $E_{M_l}(X_l - X_{l-1})$ and $E_{M_{l+1}}(X_{l+1} - X_l)$ do not need to be independent.

We would like to mention that the MLMC can also be interpreted as a multilevel control variate, following [76]. Specifically, suppose that $X = X_L$ on the level L is the

quantity of interest. According to the error estimate (2.9), the error is proportional to the product of $M_L^{-1/2}$ and the variance $\mathrm{Var}[X_L]$ of X_L. Let X_{L-1} be a cheaper pointwise approximation, e.g., the finite element approximation on a coarser grid, to X_L with known expected value. Then it is natural to use X_{L-1} as the control variate to X_L [73] and to approximate the expected value $\mathbb{E}[X_L]$ by

$$\begin{aligned}\mathbb{E}[X_L] &= \mathbb{E}[X_L - X_{L-1}] + \mathbb{E}[X_{L-1}] \\ &\approx E_{M_L}(X_L - X_{L-1}) + \mathbb{E}[X_{L-1}].\end{aligned}$$

Here we approximate the expected value $\mathbb{E}[X_L - X_{L-1}]$ by a Monte Carlo estimate, which, according to the error estimate (2.9), will have a small error, if the approximations X_L and X_{L-1} are close to each other. More generally, with a proper choice of weights, the latter condition can be relaxed to high correlation. In practice, the expected value $\mathbb{E}[X_{L-1}]$ may be still nontrivial to evaluate. In the spirit of classical multilevel methods, we can further approximate the expected value $\mathbb{E}[X_{L-1}]$ by

$$\mathbb{E}[X_{L-1}] \approx E_{M_{L-1}}(X_{L-1} - X_{L-2}) + \mathbb{E}[X_{L-2}],$$

where X_{L-2} is a cheap approximation to X_{L-1}. By applying this idea recursively, one arrives at the MLMC estimate as described in (3.11).

Now we can give the outline of the MLMC-GMsFEM framework, cf. Algorithm 1. Here, the offline space is fixed and preprocessed. The level of the samples is determined by the size of the online multiscale basis functions. The larger the online multiscale space V_{on} is, the higher the solution resolution is, but the more expensive the computation is; the smaller the online multiscale space V_{on} is, the cheaper the computation is, but the lower the solution resolution is. The MLMC approach as described above provides a framework for elegantly combining the hierarchy of

approximations from the GMsFEM, and leveraging the expensive computations on level L to those lower level approximations. In addition, we note that the samples $\{\kappa^m\}_{m=1}^{M_l}$ used in the Monte Carlo estimate $E_{M_l}(X_l - X_{l-1})$ are identical on every two consecutive levels, i.e., the permeability samples $\{\kappa^m\}_{m=1}^{M_l}$ used in the Monte Carlo estimates on two consecutive levels are nested. For a prescribed error bound ϵ, one can choose the number of samples M_l by equating the error terms for each level.

Algorithm 1 MLMC-GMsFEM

1. Offline computations:
   - Construct the snapshot space;
   - Construct a low-dimensional offline space by model reduction.
2. Multi-level online computations for estimating an expectation at level l, $1 \leq l \leq L$:
   - Generate M_l realizations of the permeability $\{\kappa_l^m\}_{m=1}^{M_l}$ (from $\{\kappa_{l-1}^m\}_{m=1}^{M_{l-1}}$);
   - For each realization κ_l^m, compute online multiscale basis functions;
   - Solve the coarse-grid problem for X_l^m;
   - Calculate the arithmetic mean $E_{M_l}(X_l - X_{l-1})$ by (3.10).
3. Output the MLMC approximation $E^L(X)$ by (3.11).

3.3.2 Cost analysis

In the following, we are interested in the root mean square errors:

$$e_{MLMC}(X_L) = \sqrt{\mathbb{E}[\|\mathbb{E}[X_L] - E^L(X_L)\|^2]},$$

$$e_{MC}(X_L) = \sqrt{\mathbb{E}[\|\mathbb{E}[X_L] - E_{M_L}(X_L)\|^2]},$$

for the MLMC estimate $E^L(X_L)$ and the MC estimate $E_{M_L}(X_L)$, respectively, with an appropriate norm depending on the quantity of interest (e.g., the absolute value for any entry of the permeability coefficient, and the L^2-norm of the solution). For the error estimation, we will use the fact that for any random variable X and any norm, $\mathbb{E}[\|\mathbb{E}[X] - E_M(X)\|^2]$ defines a norm on the error $\mathbb{E}[X] - E_M(X)$, and further, there holds the relation

$$\mathbb{E}[\|\mathbb{E}[X] - E_M(X)\|^2] = \frac{1}{M}\mathbb{E}[\|X - \mathbb{E}[X]\|^2].$$

In the analysis, we will be dealing with solutions at different scales. In the MLMC framework, we denote the scale hierarchy by $H_1 \geq H_2 \geq \cdots \geq H_L$. The number of realizations used at the level l for the scale H_l is denoted by M_l. We take

$$M_1 \geq M_2 \geq \cdots M_L.$$

For the MLMC approach, the error reads

$$\begin{aligned}
e_{MLMC}(X_L) &= \sqrt{\mathbb{E}[\|\mathbb{E}[X_L] - E^L(X_L)\|^2]} \\
&= \sqrt{\mathbb{E}[(\mathbb{E}[\sum_{l=1}^{L}(X_l - X_{l-1})] - \sum_{l=1}^{L} E_{M_l}(X_l - X_{l-1}))^2]} \\
&= \sqrt{\mathbb{E}[(\sum_{l=1}^{L}(\mathbb{E} - E_{M_l})(X_l - X_{l-1}))^2]} \\
&\leq \sum_{l=1}^{L} \sqrt{\mathbb{E}[((\mathbb{E} - E_{M_l})(X_l - X_{l-1}))^2]} \\
&\leq \sum_{l=1}^{L} \frac{1}{\sqrt{M_l}} \sqrt{\mathbb{E}[(X_l - X_{l-1} - \mathbb{E}(X_l - X_{l-1}))^2]},
\end{aligned}$$

where the second last line follows from the triangle inequality for norms, and the last

line follows from (2.9). Next we rewrite $X_l - X_{l-1} = (X_l - X) + (X - X_{l-1})$, and since $M_l \leq M_{l-1}$, we deduce

$$\begin{aligned}
e_{MLMC}(X_L) &\leq \sum_{l=1}^{L} \frac{1}{\sqrt{M_l}} \Big(\sqrt{\mathbb{E}[(X_l - X - \mathbb{E}(X_l - X))^2]} \\
&\qquad + \sqrt{\mathbb{E}[(X_{l-1} - X - \mathbb{E}(X_{l-1} - X))^2]} \Big) \\
&= \frac{1}{\sqrt{M_L}} \sqrt{\mathbb{E}[(X_L - X - \mathbb{E}(X_L - X))^2]} + \frac{1}{\sqrt{M_1}} \sqrt{\mathbb{E}[X^2]} \\
&\qquad + \sum_{l=1}^{L-1} \left(\frac{1}{\sqrt{M_{l+1}}} + \frac{1}{\sqrt{M_l}} \right) \sqrt{\mathbb{E}[((X_l - X) - E(X_l - X))^2]} \\
&\leq \sum_{l=1}^{L} \frac{2}{\sqrt{M_{l+1}}} \sqrt{\mathbb{E}[(X_l - X)^2]} + \frac{1}{\sqrt{M_1}} \sqrt{\mathbb{E}(X^2)} \\
&\leq \sum_{l=1}^{L} \frac{2}{\sqrt{M_{l+1}}} \delta_l + \frac{1}{\sqrt{M_1}} \sqrt{\mathbb{E}[X^2]},
\end{aligned}$$

where the second last line follows from the inequality

$$\sqrt{\mathbb{E}[((X_l - X) - E(X_l - X))^2]} \leq \sqrt{\mathbb{E}[(X_l - X)^2]},$$

a direct consequence of the bias-variance decomposition, and the assumption $M_{L+1} \leq M_L$. Here we denote $\sqrt{\mathbb{E}[(X_l - X)^2]}$ as δ_l. As mentioned in Section 3.3.1, the lower the level l is, the less accurate the approximation X_l is with respect to X_L, hence we will have $\delta_1 > \delta_2 > \cdots > \delta_L$. To equate the error terms we choose

$$M_l = M \begin{cases} \left(\frac{1}{\delta_L}\right)^2 \mathbb{E}[X^2], & l = 1, \\ \left(\frac{\delta_{l-1}}{\delta_L}\right)^2, & 2 \leq l \leq L+1, \end{cases}$$

where $M > 1$ is a fixed positive integer. Then we end up with

$$e_{MLMC}(X_L) \leq \frac{(2L+1)}{M}\delta_L.$$

In principle, for a prescribed error bound ϵ such that $e_{MLMC}(X_L) \leq \epsilon$, one can deduce from the formula the proper choice of the number N of samples, for any given level L.

3.4 Multilevel Markov chain Monte Carlo

One of the most popular and versatile methods for numerically exploring posterior distributions arising from the Bayesian formulation is the Markov chain Monte Carlo (MCMC) method. The basic idea is to construct a Markov chain with the target distribution as its stationary distribution. However, the sampling step remains very challenging in high-dimensional spaces. One powerful idea of improving the sampling efficiency is preconditioning, first illustrated in [18, 32], and more recently extended in [7]. In the latter work, some theoretical properties, e.g., asymptotic confidence interval, of a multistage version of the two-level algorithm ([32]) are also established.

3.4.1 MLMCMC with GMsFEM

The standard Metropolis-Hastings algorithm generates samples from the posterior distribution $\pi(\kappa) = p(\kappa|F_{obs})$ (introduced in Section 2.3), cf. Algorithm 2. Here $\mathcal{U}(0,1)$ is the uniform distribution over the interval $(0,1)$. As described in Section 2.5, the permeability field κ is determined by the parameters θ_k's. Hence, given the current sample κ^m, parameterized by its parameters θ_k's, one can generate the proposal κ by generating the proposal for θ_k's first, i.e., draw θ_k from distribution $q_{\theta_k}(\theta_k|\theta_k^m)$ respectively for each k (in view of the independence among all the θ_k's), for some proposal distributions $q_{\theta_k}(\theta_k|\theta_k^m)$, and then form the proposal for the entire

permeability field κ.

Algorithm 2 Metropolis-Hastings MCMC

Specify κ_0 and M;
for $m = 0 : M$ **do**
  Generate the entire permeability field proposal κ from $q(\kappa|\kappa^m)$;
  Compute the acceptance probability $\gamma(\kappa^m)$ by (3.12);
  Draw $u \sim \mathcal{U}(0, 1)$;
  if $\gamma(\kappa^m, \kappa) \leq u$ **then**
    $\kappa^{m+1} = \kappa$;
  else
    $\kappa^{m+1} = \kappa^m$.
  end if
end for

The transition kernel $K_r(\kappa^m, \kappa)$ of the Markov chain generated by Algorithm 2 is given by

$$K_r(\kappa^m, \kappa) = \gamma(\kappa^m, \kappa)q(\kappa|\kappa^m) + \delta_{\kappa^m}(\kappa)\left(1 - \int \gamma(\kappa^m, \kappa)q(\kappa|\kappa^m)d\kappa\right),$$

where $q(\kappa|\kappa^m)$ denotes the proposal distribution and $\gamma(\kappa^m, \kappa)$ denotes the acceptance probability for the proposal κ defined by

$$\gamma(\kappa^m, \kappa) = \min\left\{1, \frac{q(\kappa^m|\kappa)\pi(\kappa)}{q(\kappa|\kappa^m)\pi(\kappa^m)}\right\}. \tag{3.12}$$

Now we integrate the multilevel idea with the Metropolis-Hastings algorithm and

the GMsFEM. Like before, we start with the telescopic sum

$$\begin{aligned}\mathbb{E}_{\pi_L}[F_L] &= \int F_L(x)\pi_L(x)dx \\ &= \int F_0(x)\pi_0(x)dx + \sum_{l=1}^{L}\int (F_l(x)\pi_l(x) - F_{l-1}(x)\pi_{l-1}(x))dx,\end{aligned}$$

where π_l denotes the approximate target distribution at level l, and π_0 is our initial level. We note that after the initial level each expectation involves two measures, π_l and π_{l-1}, which is different from the case of the MLMC (see [52]). Therefore, we rewrite the integration using a product measure as

$$\begin{aligned}\int (F_l(x)\pi_l(x) - F_{l-1}(x)\pi_{l-1}(x))dx &= \int F_l(x)\pi_l(x)dx - \int F_{l-1}(y)\pi_{l-1}(y)dy \\ &= \int\int (F_l(x) - F_{l-1}(y))\pi_l(x)\pi_{l-1}(y)dxdy \\ &= \mathbb{E}_{\pi_l,\pi_{l-1}}[F_l(x) - F_{l-1}(y)].\end{aligned}$$

Therefore, we have

$$\mathbb{E}_{\pi^L}[F_L] = \mathbb{E}_{\pi_0}[F_0] + \sum_{l=1}^{L}\mathbb{E}_{\pi_l,\pi_{l-1}}[F_l - F_{l-1}]. \tag{3.13}$$

The idea of our multilevel method is to estimate each term of the right hand side of equation (3.13) independently. In particular we can estimate each term in (3.13) by an MCMC estimator. The first term $\mathbb{E}_{\pi_0}[F_0]$ can be estimated using the standard MCMC estimator in Algorithm 2. We estimate the expectation $\mathbb{E}_{\pi_l,\pi_{l-1}}[F_l(x) - F_{l-1}(y)]$ by the sample mean

$$\mathbb{E}_{\pi_l,\pi_{l-1}}[F_l(x) - F_{l-1}(y)] \approx \frac{1}{M_l}\sum_{m=1}^{M_l}(F_l(x_l^m) - F_{l-1}(y_l^m)), \tag{3.14}$$

where the samples $\{(y_l^m, x_l^m)\}_{m=1}^{M_l}$ are drawn from the product measure $\pi_{l-1}(y) \otimes \pi_l(x)$. Next we describe an efficient preconditioned MCMC method for generating samples from the product measure $\pi_{l-1}(y) \otimes \pi_l(x)$, extending our earlier work [32].

Here we introduce a multilevel MCMC algorithm by adapting the proposal distribution $q(\kappa|\kappa_m)$ to the target distribution $\pi(\kappa)$ using the GMsFEM with different sizes of the online space which we call different levels, cf. Algorithm 3. The process modifies the proposal distribution $q(\kappa|\kappa_m)$ by incorporating the online coarse-scale information. Let $F_l(\kappa)$ be the output (pressure/water-cut) computed by solving the online coarse problem at level l for a given κ. The target distribution $\pi(\kappa)$ is approximated on level l by $\pi_l(\kappa)$, with $\pi(\kappa) \equiv \pi_L(\kappa)$. Here we have

$$\pi_l(\kappa) \propto \exp\left(-\frac{||F_{obs} - F_l(\kappa)||^2}{2\sigma_l^2}\right) \times p(\kappa). \tag{3.15}$$

In the algorithm we still keep the same offline space for each level. From level 0 to level L, we increase the size of the online space as we go to a higher level, which means for any levels l, $l+1 \leq L$, samples of level l are cheaper to generate than that of level $l+1$. This idea underlies the cost reduction using the multilevel estimator. Hence the posterior distribution for coarser levels $\pi_l, l = 0, \ldots, L-1$ do not have to model the measured data as faithfully as π_L, which in particular implies that by choosing suitable value of σ_l^2 it is easier to match the result $F_l(\kappa)$ with the observed data. We denote the number of samples at level l by M_l, where we will have $M_0 \leq \cdots \leq M_L$. As was discussed above, our quantity of interest can be approximated by the telescopic sum (3.13). We denote the estimator of $\mathbb{E}_{\pi_0}[F_0]$ at the initial level by $\widehat{F}_0$. Then by the MCMC estimator we have

$$\widehat{F}_0 = \frac{1}{M_0} \sum_{m=1}^{M_0} F_0(x_0^m).$$

Here x_0^m denotes the samples we accepted on the initial level (after discarding the samples at the burn-in period). Similarly we denote the estimator of the differences $\mathbb{E}_{\pi_l,\pi_{l-1}}[F_l(x) - F_{l-1}(y)]$ by $\widehat{Q}_l$. Then with (3.14) we have

$$\widehat{Q}_l = \frac{1}{M_l}\sum_{m=1}^{M_l}(F_l(x_l^m) - F_{l-1}(y_l^m)),$$

where the samples $\{(y_l^m, x_l^m)\}_{m=1}^{M_l}$ are drawn from the product measure $\pi_{l-1}(y) \otimes \pi_l(x)$. Finally denote the estimator of $\mathbb{E}_{\pi_L}[F_L]$ or our full MLMCMC estimator by $\widehat{F}_L$, then the quantity of interest $E_{\pi_L}[F_L]$ is approximated by

$$\widehat{F}_L = \widehat{F}_0 + \sum_{l=1}^{L}\widehat{Q}_l. \tag{3.16}$$

3.4.2 Convergence analysis

In this part, we briefly analyze the convergence property of the multilevel MCMC algorithm, cf. Algorithm 3. Specifically, we discuss the detailed balance relation and the ergodicity of the Markov chain, following the general convergence theory in [73].

To this end, we denote

$$\begin{aligned}
&\mathcal{E}^l = \{\kappa : \pi_l(\kappa) > 0\}, \quad l = 1, 2, \ldots, L, \\
&\mathcal{D} = \{\kappa : q_{l-1}(\kappa|\kappa_l^m) > 0 \quad \text{for some } \kappa_l^m \in \mathcal{E}^l\}.
\end{aligned} \tag{3.17}$$

The set $\mathcal{E}^l$ is the support of the distributions $\pi_l(\kappa)$ at level l. The set $\mathcal{E}^L$ is the support of the target distribution $\pi(\kappa) = \pi_L(\kappa)$ at the finest level. The set $\mathcal{D}$ is the set of all the proposals which can be generated by the proposal distribution $q_{l-1}(\kappa|\kappa_l^m)$. To sample from $\pi(\kappa)$ correctly, it is necessary that $\mathcal{E}^L \subseteq \mathcal{E}^{L-1} \subseteq \ldots \subseteq \mathcal{E}^1 \subseteq \mathcal{D}$ (up to a set of zero measure). Otherwise, if one of these conditions is not true, say, $\mathcal{E}^{l+1} \not\subseteq \mathcal{E}^l$,

Algorithm 3 Multilevel Metropolis-Hastings MCMC

Given κ_m, draw a trial proposal κ from distribution $q(\kappa|\kappa_1^m) = q_0(\kappa|\kappa_1^m)$;
Compute the acceptance probability

$$\rho_1(\kappa_1^m, \kappa) = \min\left\{1, \frac{q_0(\kappa_1^m|\kappa)\pi_1(\kappa)}{q_0(\kappa|\kappa_1^m)\pi_1(\kappa_1^m)}\right\};$$

$u \sim \mathcal{U}(0,1)$;
if $u < \rho_1(\kappa_1^m, \kappa)$ **then**
  $\kappa_1^{m+1} = \kappa$ (at the initial level);
else
  $\kappa_1^{m+1} = \kappa_1^m$ (at the initial level);
end if
for $l = 1 : L-1$ **do**
  if κ is accepted at level l **then**
    Form the proposal distribution q_l (on the $l+1$th level) by

$$q_l(\kappa|\kappa_{l+1}^m) = \rho_l(\kappa_{l+1}^m, \kappa)q_{l-1}(\kappa|\kappa_{l+1}^m) + \delta_{\kappa_{l+1}^m}(1 - \int \rho_l(\kappa_{l+1}^m, \kappa)q_{l-1}(\kappa|\kappa_{l+1}^m)d\kappa_{l+1}^m);$$

    Compute the acceptance probability

$$\rho_{l+1}(\kappa_{l+1}^m, \kappa) = \min\left\{1, \frac{q_l(\kappa_{l+1}^m|\kappa)\pi_{l+1}(\kappa)}{q_l(\kappa|\kappa_{l+1}^m)\pi_{l+1}(\kappa_{l+1}^m)}\right\} = \min\left\{1, \frac{\pi_l(\kappa_{l+1}^m)\pi_{l+1}(\kappa)}{\pi_l(\kappa)\pi_{l+1}(\kappa_{l+1}^m)}\right\};$$

    $u \sim \mathcal{U}(0,1)$;
    if $u < \rho_{l+1}(\kappa_{l+1}^m, \kappa)$ **then**
      $\kappa_{l+1}^{m+1} = \kappa$ and go to next level (if $l = L-1$, accept κ and set $\kappa_L^{m+1} = \kappa$);
    else
      $\kappa_{l+1}^{m+1} = \kappa_{l+1}^m$, and break.
    end if
  end if
end for

then there will exist a subset $A \subset (\mathcal{E}^{l+1} \setminus \mathcal{E}^l)$ such that

$$\pi_{l+1}(A) = \int_A \pi_{l+1}(\kappa)d\kappa > 0 \quad \text{and} \quad \pi_l(A) = \int_A \pi_l(\kappa)d\kappa = 0,$$

which means no element of A can pass the level l and A will never be visited by the Markov chain $\{\kappa_{l+1}^m\}$. Thus, the distribution at level $l+1$, i.e., $\pi_{l+1}(\kappa)$ is not sampled properly.

For most practical proposal distributions $q_{l-1}(\kappa|\kappa_l^m)$, such as random walk samplers, the condition $\mathcal{E}^1, \ldots, \mathcal{E}^L \subseteq \mathcal{D}$ is naturally satisfied. To show the inclusion $\mathcal{E}^{l+1} \subseteq \mathcal{E}^l$ for any level l, notice that if the precision parameters σ_{l+1} and σ_l are chosen to be relatively small, then $\pi_{l+1}(\kappa)$ and $\pi_l(\kappa)$ are very close to zero for most proposals. From the numerical point of view, the proposal κ is very unlikely to be accepted if $\pi_{l+1}(\kappa)$ and $\pi_l(\kappa)$ are close to zero. Consequently the support of the distributions should be interpreted as

$$\mathcal{E}^{l+1} = \{\kappa : \pi_{l+1}(\kappa) > \delta\} \quad \text{and} \quad \mathcal{E}^l = \{\kappa : \pi_l(\kappa) > \delta\},$$

where δ is a small positive number. If $\kappa \in \mathcal{E}^{l+1}$, then $\pi_{l+1}(\kappa) > \delta$ and $\|F_{obs} - F_{l+1}(\kappa)\|^2/2\sigma_{l+1}^2$ is not very large. To make $\kappa \in \mathcal{E}^l$, $\|F_{obs} - F_l(\kappa)\|^2/2\sigma_l^2$ should not be very large either. If $\|F_{obs} - F_l(\kappa)\|$ is bounded by $\|F_{obs} - F_{l+1}(\kappa)\|$ up to a multiplicative constant, then the condition $\mathcal{E}^{l+1} \subseteq \mathcal{E}^l$ can be satisfied by choosing the parameter σ_l properly. For our model, the coarser level quantity is indeed bounded by the fine level quantity. Thus, the condition $\mathcal{E}^L \subseteq \mathcal{E}^{L-1} \subseteq \ldots \subseteq \mathcal{E}^1 \subseteq \mathcal{D}$ is satisfied.

Let

$$Q_l(\kappa_l^m, \kappa) = \rho_l(\kappa_l^m, \kappa)q_{l-1}(\kappa|\kappa_l^m) + \delta_{\kappa_l^m}(1 - \int \rho_l(\kappa_l^m, k)q_{l-1}(\kappa|\kappa_l^m)d\kappa_l^m) \qquad (3.18)$$

denote the transition kernel of the Markov chain at level l. As in a regular MCMC method, we can show that $Q_l(\kappa_l^m, \kappa)$ satisfies the detailed balance condition at level l, i.e.,

$$\pi_l(\kappa_l^m) Q_l(\kappa_l^m, \kappa) = \pi_l(\kappa) Q_l(\kappa, \kappa_l^m), \tag{3.19}$$

for any $\kappa, \kappa_l^m \in \mathcal{E}^l$. In fact, the equality (3.19) is obviously true when $\kappa = \kappa_l^m$. If $\kappa \neq \kappa_l^m$, then $Q_l(\kappa_l^m, \kappa) = \rho_l(\kappa_l^m, \kappa) q_{l-1}(\kappa|\kappa_l^m)$, we have

$$\begin{aligned}
\pi_l(\kappa_l^m) Q_l(\kappa_l^m, \kappa) &= \pi_l(\kappa_l^m) \rho_l(\kappa_l^m, \kappa) q_{l-1}(\kappa|\kappa_l^m) \\
&= \min\left(\pi_l(\kappa_l^m) q_{l-1}(\kappa|\kappa_l^m), \pi_l(\kappa) q_{l-1}(\kappa_l^m|\kappa)\right) \\
&= \min\left(\frac{\pi_l(\kappa_l^m) q_{l-1}(\kappa|\kappa_l^m)}{\pi_l(\kappa) q_{l-1}(\kappa_l^m|\kappa)}, 1\right) \pi_l(\kappa) q_{l-1}(\kappa_l^m|\kappa) \\
&= \rho_l(\kappa, \kappa_l^m) \pi_l(\kappa) q_{l-1}(\kappa_l^m|\kappa) = \pi_l(\kappa) Q_l(\kappa, \kappa_l^m).
\end{aligned}$$

So the detailed balance condition (3.19) is always satisfied. Using (3.19) we can easily show that $\pi(A) = \int Q_l(\kappa, A) d\kappa$ for any $A \in \mathcal{B}(\mathcal{E}^l)$, where $\mathcal{B}(\mathcal{E}^l)$ denotes all measurable subsets of $\mathcal{E}^l$. Thus, $\pi_l(k)$ is indeed the stationary distribution of the transition kernel $Q_l(\kappa_l^m, \kappa)$.

In a regular MCMC method, cf. Algorithm 2, the proposal $q(\kappa|\kappa^m)$ is usually chosen to satisfy $q(\kappa|\kappa^m) > 0$ for any $(\kappa^m, \kappa) \in \mathcal{E} \times \mathcal{E}$, which guarantees that the resulting MCMC chain is irreducible. Similarly the irreducibility holds for multilevel MCMC at each level l if $q_{l-1}(\kappa|\kappa_l^m) > 0$ for any $(\kappa_l^m, \kappa) \in \mathcal{E}^l \times \mathcal{E}^l$. We already have $\mathcal{E}^L \subseteq \mathcal{E}^{L-1} \subseteq \ldots \subseteq \mathcal{E}^1$ holds, which means $\rho_l(\kappa_l^m, \kappa) > 0$, and also for common choices of the proposal distribution, we have $q_{l-1}(\kappa|\kappa_l^m)$ positive, which guarantees the irreducibility of the chain at each level.

To prove the convergence of the distribution, we need to show that the chain is aperiodic. Recall that a simple sufficient condition for aperiodicity is that the

transition kernel $Q(\kappa^m, \{\kappa^m\}) > 0$ for some $\kappa^m \in \mathcal{E}$. In other words, the event $\{\kappa^{m+1} = \kappa^m\}$ happens with a positive probability. For our multilevel MCMC at finest level l, consider the transition kernel (3.18), we have

$$\begin{aligned} Q_l(\kappa_l^m, \{\kappa_l^m\}) &= 1 - \int_{\kappa \neq \kappa_l^m} \rho_l(\kappa_l^m, \kappa) q_{l-1}(\kappa|\kappa_l^m) d\kappa_l^m \\ &= 1 - \int_{\kappa \neq \kappa_l^m} \rho_l(\kappa_l^m, \kappa) \rho_{l-1}(\kappa_l^m, \kappa) \dots \rho_1(\kappa_l^m, \kappa) q_0(\kappa|\kappa_l^m) d\kappa_l^m. \end{aligned}$$

Hence $Q_l(\kappa_l^m, \{\kappa_l^m\}) \equiv 0$ requires $\rho_s(\kappa_s^m, \kappa) = 1$ for $s = 1, \dots, l$, for almost all $\kappa \in \mathcal{D}$. which means that all the proposals generated by $q_0(\kappa_1^m, \kappa)$ are correct samples for distributions at all levels. In this case it does not make sense to use the MCMC method since we can sample directly from $q(\kappa|\kappa^m)$. Thus in practice we can always safely assume that the chain generated by the multilevel MCMC is aperiodic. As a result the Markov chain generated by MLMCMC converges.

In Algorithm 3, the specific proposal distribution q_l can be computed easily and at no additional cost, as we can simplify the acceptance probability for level $l+1$ to

$$\rho_{l+1}(\kappa_{l+1}^m, \kappa) = \min\left\{1, \frac{\pi_l(\kappa_{l+1}^m)\pi_{l+1}(\kappa)}{\pi_l(\kappa)\pi_{l+1}(\kappa_{l+1}^m)}\right\}. \tag{3.20}$$

This is true when $\kappa_{l+1}^m = \kappa$, so we will demonstrate this for the case $\kappa_{l+1}^m \neq \kappa$. In this case, $q_l(\kappa|\kappa_{l+1}^m) = \rho_l(\kappa_{l+1}^m, \kappa) q_{l-1}(\kappa|\kappa_{l+1}^m)$, then,

$$\begin{aligned} \rho_{l+1}(\kappa_{l+1}^m, \kappa) &= \min\left\{1, \frac{q_l(\kappa_{l+1}^m|\kappa)\pi_{l+1}(\kappa)}{q_l(\kappa|\kappa_{l+1}^m)\pi_{l+1}(\kappa_{l+1}^m)}\right\} \\ &= \min\left\{1, \frac{\rho_l(\kappa, \kappa_{l+1}^m) q_{l-1}(\kappa_{l+1}^m|\kappa)\pi_{l+1}(\kappa)}{\rho_l(\kappa_{l+1}^m, \kappa) q_{l-1}(\kappa|\kappa_{l+1}^m)\pi_{l+1}(\kappa_{l+1}^m)}\right\}. \end{aligned}$$

Assume for simplicity $q_{l-1}(\kappa_{l+1}^m|\kappa)\pi_l(\kappa) > q_{l-1}(\kappa|\kappa_{l+1}^m)\pi_l(\kappa_{l+1}^m)$, then

$$\rho_l(\kappa_{l+1}^m, \kappa) = 1 \quad \text{and} \quad \rho_l(\kappa, \kappa_{l+1}^m) = \frac{q_{l-1}(\kappa|\kappa_{l+1}^m)\pi_l(\kappa_{l+1}^m)}{q_{l-1}(\kappa_{l+1}^m|\kappa)\pi_l(\kappa)}.$$

Using these relations we obtain the desired formula (3.20). Similarly, in the case of $q_{l-1}(\kappa_{l+1}^m|\kappa)\pi_l(\kappa) < q_{l-1}(\kappa|\kappa_{l+1}^m)\pi_l(\kappa_{l+1}^m)$, then

$$\rho_l(\kappa, \kappa_{l+1}^m) = 1 \quad \text{and} \quad \rho_l(\kappa_{l+1}^m, \kappa) = \frac{q_{l-1}(\kappa_{l+1}^m|\kappa)\pi_l(\kappa)}{q_{l-1}(\kappa|\kappa_{l+1}^m)\pi_l(\kappa_{l+1}^m)}.$$

With these relations we also deduce that (3.20) holds.

3.5 Numerical results

In our numerical examples, we consider permeability fields described by two-point correlation functions, and use Karhunen-Loève expansion (KLE) to parameterize the permeability fields as described in Section 2.5. Then we apply the MLMC and MLMCMC with the GMsFEM algorithms described earlier. The permeability field κ in this section is assumed to follow a log-normal distribution with a known spatial covariance, with the correlation function $R(x, y)$ given by

$$R(x, y) = \sigma^2 \exp\Big(-\frac{|x_1 - y_1|^2}{2l_1^2} - \frac{|x_2 - y_2|^2}{2l_2^2}\Big), \tag{3.21}$$

where l_1 and l_2 are the correlation lengths in x_1- and x_2-direction, respectively, and $\sigma^2 = \mathbb{E}[Y^2]$ is a constant that determines the variation of the permeability field. Thus we have the following prior random field by N-term KLE:

$$Y(x, \mu) = \sum_{k=1}^{N} \sqrt{\lambda_k}\theta_k(\mu)\Phi_k(x), \quad \theta_k \sim N(0, 1). \tag{3.22}$$

3.5.1 MLMC

In our simulations, we evaluate the performance of the MLMC method on computing the expected values of our quantity of interest F. In particular, we consider the stationary, single-phase flow model (2.7) on the unit square domain $D = (0, 1)^2$ with $f \equiv 1$ and linear boundary conditions. The forward problem is solved with the GMsFEM, and the fine grid and coarse grid are chosen to be 50×50 and 5×5, respectively. The quantity of interest F for this set of simulations is the fine scale pressure field. We consider the following two Gaussian covariance functions:

- Isotropic Gaussian field with correlation lengths $l_1 = l_2 = 0.1$ and a stochastic dimension 5;
- Anisotropic Gaussian field with correlation lengths $l_1 = 0.1$ and $l_2 = 0.05$, and a stochastic dimension 5.

In both cases, we use the variance $\sigma^2 = 2$, and keep $N = 5$ terms in the final KL expansion where the θ_k coefficients are drawn from a normal distribution with zero mean and unit variance.

We denote by F_l the fine scale pressure field at level l in MLMC. The level of our interest is $L = 3$. As stated in Algorithm 1, we generate M_l realizations at level l of the permeability field, solve the model problems by choosing N_l eigenvalues to generate the online space in the GMsFEM, and compute the MLMC approximation of $\mathbb{E}[F_L]$ by (3.11). We compare MLMC with the standard MC at the level L of interest with the same amount of cost. Hence we choose

$$\widehat{M} = \frac{\sum_{l=1}^{L} N_l^2 M_l}{N_L^2}$$

as the number of samples in the standard MC algorithm. We use the arithmetic mean of M_{ref} samples of the pressure field as reference and compute the relative L^2-errors

$$(e^{rel}_{MLMC})[F_L] = \frac{\|\mathbb{E}^{ref}_{M_{ref}}[F_L] - \mathbb{E}^{L}[F_L]\|_{L^2(D)}}{\|\mathbb{E}^{ref}_{M_{ref}}[F_L]\|_{L^2(D)}},$$

$$(e^{rel}_{MC})[F_L] = \frac{\|\mathbb{E}^{ref}_{M_{ref}}[F_L] - \mathbb{E}^{MC}_{\widehat{M}}[F_L]\|_{L^2(D)}}{\|\mathbb{E}^{ref}_{M_{ref}}[F_L]\|_{L^2(D)}}.$$

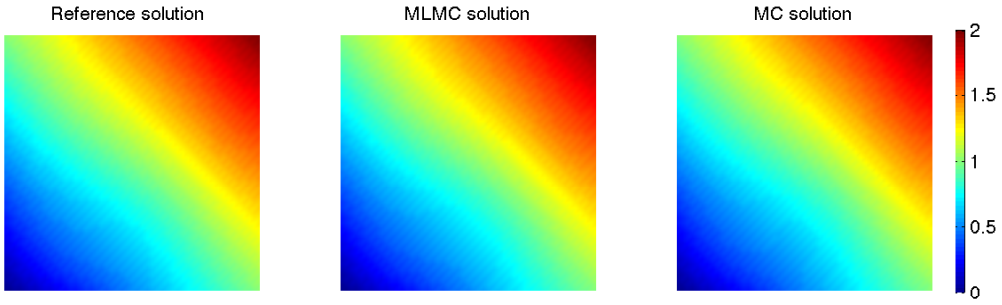
For the simulations use $N_1 = 4$, $N_2 = 8$, and $N_3 = 16$ eigenfunctions for the online space construction. We respectively set the number of samples at each level to be $M_1 = 128$, $M_2 = 32$, and $M_3 = 8$, and equate the computational costs for the MLMC and MC relative error comparisons. With this choice of realizations for MLMC, we use $\widehat{M} = 20$ permeability realizations for the standard MC forward simulations. The parameters we have used and the respective relative errors are summarized in Table 3.1. Figure 3.1 illustrates expected pressure fields for different correlation lengths and different methods (MLMC and MC). For both covariances, we observe that the MLMC approach yields errors which are about 1.5 times smaller than those resulting from the MC approach. We note that the gain is larger for the isotropic case than for the anisotropic case.

3.5.2 MLMCMC

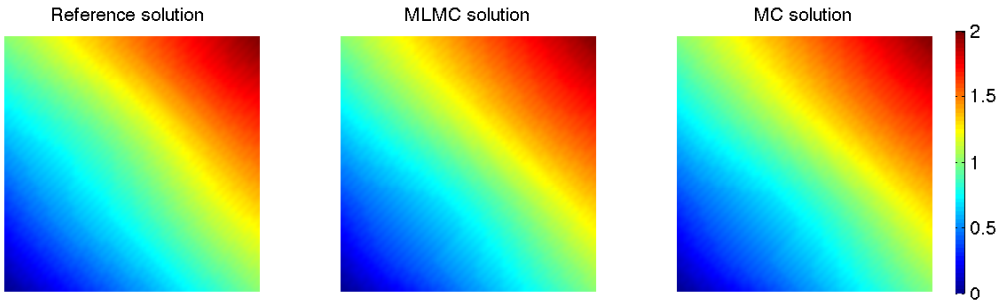
In our MLMCMC experiments we also consider the model problem (2.7) on $D = (0,1)^2$ with $f \equiv 1$ and linear boundary conditions. The prior permeability distribution $p(\kappa)$ is also parameterized by KLE as above. The "observed" data F_{obs} is obtained by generating a reference permeability field (indicated as reference solution in Figure 3.5), solving the forward problem with the GMsFEM, and evaluating the pressure at nine points away from the boundary. The locations of the reference pressures are shown in Figure 3.2. We note that the data generated by the GMsFEM

Table 3.1: Parameters and errors for the estimates by MLMC vs. MC.

	Isotropic Gaussian	Anisotropic Gaussian
(N_1, N_2, N_3)	$(4, 8, 16)$	$(4, 8, 16)$
(M_1, M_2, M_3)	$(128, 32, 8)$	$(128, 32, 8)$
$\widehat{N}$	16	16
$\widehat{M}$	24	24
M_{MCref}	5000	5000
e^{rel}_{MLMC}	0.0431	0.0653
e^{rel}_{MC}	0.0802	0.0952
$e^{rel}_{MC}/e^{rel}_{MLMC}$	1.86	1.45



(a) Isotropic Gaussian.



(b) Anisotropic Gaussian.

Figure 3.1: Pressure field solutions for different methods and correlation lengths.

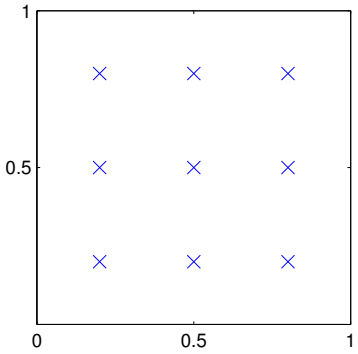


Figure 3.2: The points where pressure is evaluated.

is very close to that by the standard finite element method on a refined mesh, with a relative error less than 0.1%, and thus the "inverse crime" is not present.

Our proposal distribution is a random walker sampler in which the proposal distribution depends on the previous value of the permeability field and is given by $q(\kappa|\kappa_n) = \kappa_n + \delta\epsilon_n$ where ϵ_n is a random perturbation with mean zero and unit variance, and δ is a step size. The random perturbations are imposed on the θ_k coefficients in the KL expansion.

We consider two examples, one with isotropic Gaussian field of correlation length $l_1 = l_2 = 0.1$, the other with anisotropic Gaussian field of correlation lengths $l_1 = 0.05$, $l_2 = 0.1$. For both examples we use $\delta = 0.2$ in the random walk sampler. We again use the level $L = 3$, and for each level l we take the same number of KLE terms, $N = 5$ for the tests. For the GMsFEM, we take the number of eigenvalues to generate the online space at each level as $N_1 = 4, N_2 = 8, N_3 = 16$. We take our quantities of interest F as the pressure values at the same nine points and use them in order to compute the acceptance probabilities as shown in Algorithm 3.

For the MLMCMC examples, we run Algorithm 3 until $P_4 = 1000$ total samples pass the final level of acceptance. We note that 300 initial accepted samples are

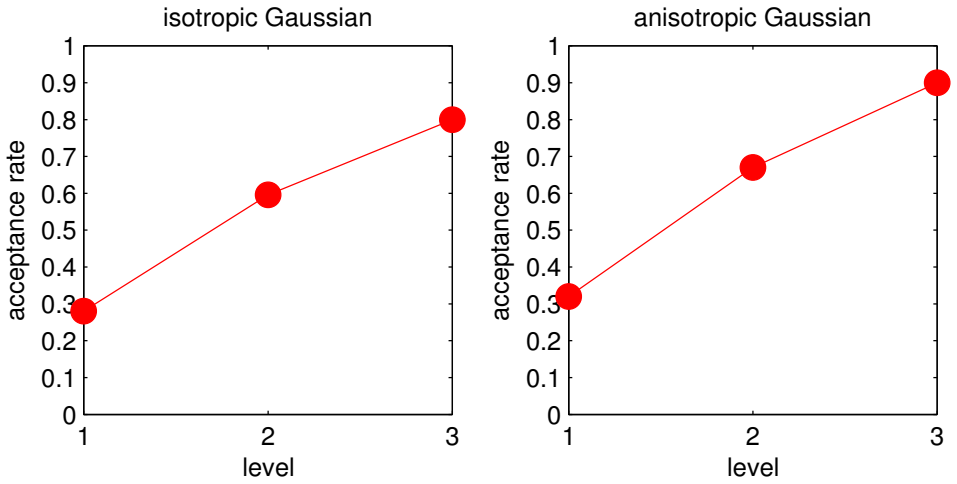


Figure 3.3: Acceptance rate of multilevel sampler with both isotropic and anisotropic trials.

discarded as burn-in. The acceptance rates of the multilevel sampler are shown in Figure 3.3. To compute the acceptance rates, we assume that P_1, P_2, and P_3 samples are proposed for respective levels L_1, L_2, and L_3. Then, the rate at the l-th level is the ratio P_{l+1}/P_l. Most notably, the results in Figure 3.3 show that the acceptance rate increases as l increases. In particular, for more expensive (larger) levels, we observe that it is much more probable that a proposed sample will be accepted. This is an advantage of the multilevel method, due to the fact that fewer proposals are wasted on more expensive computations. We also show a set of plots in Figure 3.4 that illustrate the errors $E_k = \|F_{obs} - F_L(\kappa)\|$, cf. (3.15), of the accepted samples on the finest level. In Figure 3.5 we plot some of the accepted permeability realizations that have passed all levels of computation. We note that the general shapes of the accepted the fields do not necessarily match that of the reference field, reinforcing the notion that the problem is ill-posed due to the fact that a variety of proposals may explain the reference data equally well.

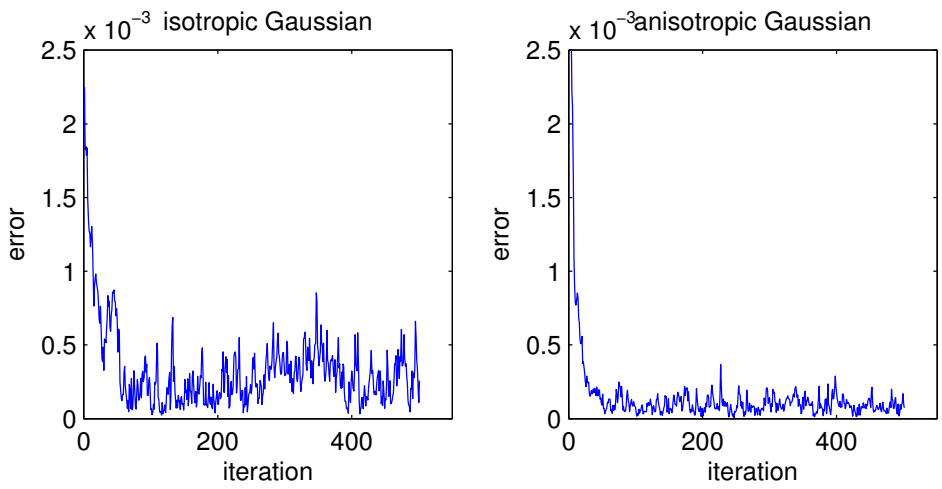


Figure 3.4: Plots of iteration vs. error with both isotropic and anisotropic trials.

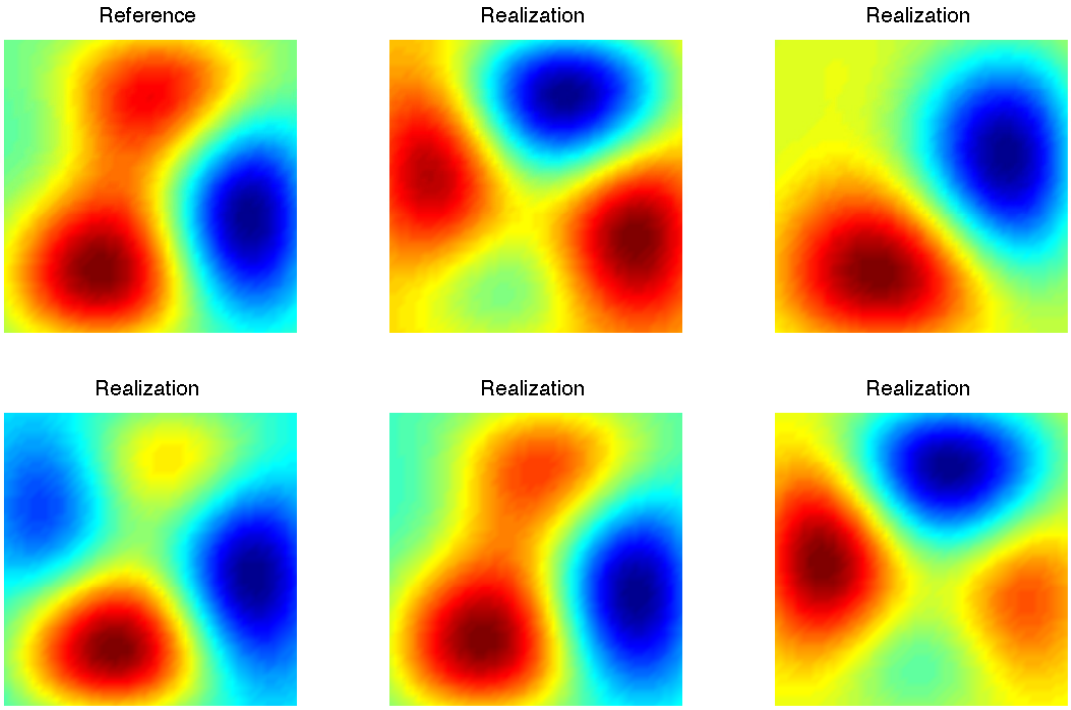


Figure 3.5: Isotropic MLMCMC accepted realizations.

4. MULTISCALE SEPARABLE SPACE-PARAMETER BASIS FUNCTIONS

4.1 Introduction

In this section, we propose an approach that follows the idea of separation of variables where special separable basis functions are computed based on optimization. In particular, we propose special multiscale basis functions which are product of multiscale spatial basis functions and special parameter-dependent functions. These basis functions are calculated as a result of optimization, which is described in the dissertation. It is worthy mentioning that the approach that we propose can be considered as belonging to the Proper Genaralized Decomposition(PGD) [70, 54, 2, 69].

We will combine efficient numerical techniques for handling uncertainties and spatial scales through the proposed model reduction. Specifically, we seek the approximate solution of the model problem (2.7) in the form

$$u(x,\mu) = \sum_i a_i(\mu) v_i(x), \tag{4.1}$$

and find $a_i(\mu)$ and $v_i(x)$ in each term successively. Here, μ is the parameter and x is the space variable. The equations for each a_i and v_i are derived by minimizing the energy related to the original parameter-independent PDE. In this way, the original high dimensional problem is decomposed into a sequence of low dimensional problems. Although we still have to solve the parameter-independent PDEs repeatedly, like in the case of Monte Carlo sampling approach, we recover the approximated solution for all the parameters. To obtain an accurate solution, the required number of terms N in the expansion is not a function of the problem dimension, but rather

depends on the regularity of the exact solution [17]. Thus, the exponential growth of degrees of freedom with respect to the dimension of parameter space can be avoided and the solution can be approached with a small number of N, which achieves a global model reduction.

We note that solving the parameter-independent PDEs repeatedly can still impose prohibitively expensive computational cost. And solving these PDEs accurately won't contribute too much to the overall solution since the iteration result of them only provide the expression of one term in our expansion. Therefore, we apply the Generalized Multiscale Finite Element Method (GMsFEM) [26, 27, 20] to these parameter-independent PDEs in order to speed up the computation.

The main idea of GMsFEM is to generalize the Multiscale Finite Element Method (MsFEM) by systematically enriching the coarse spaces and taking into account small scale information and complex input spaces. This approach divides the computation into an offline-online procedure which is described in details in Section 3.2. To implement this method in this work, firstly we select a number of realizations of the permeability field and create the snapshot space by solving local problems on the coarse subdomain. The offline space is then obtained through solving localized eigenvalue problems that use averaged parameter quantities within the space of snapshots. This offline space should only be computed once and is used for multiple solves in the online stage, which achieves a local model reduction in our iterative solving for each term in the expansion, and brings computational savings.

We present numerical results, which show the convergence of the proposed methods. In our numerical results, we consider several examples of parameter-dependent PDEs and compute the convergence rate as we increase the number of terms in (4.1) and also enrich the multiscale space. First, we observe that the errors associated with adding more terms in (4.1) and the errors due to GMsFEM can be of the same

order in some cases. In general, one needs to balance these errors for computational efficiency. We observe that with only a few terms in the expansion (4.1), we can achieve a fast convergence. This demonstrates that these methods can be more efficient compared to Monte Carlo approaches. Moreover, we discuss adaptive strategies where additional terms are added based on error indicators.

The section is organized as follows. In Section 4.2, we introduce the general framework of the PGD method and propose our global separating algorithm. In Section 4.3, we outline the procedure of the GMsFEM to solve for our parameter-independent problems. Section 4.4 is devoted to the numerical results and some analysis of the method, and we have a brief discussion on the adaptivity of the method.

4.2 Global separating algorithm

We are interested in the solution of the following problem:

$$\begin{aligned} -\nabla \cdot (\kappa(x,\mu)\nabla u(x,\mu)) &= f \text{ in } D, \\ u|_{\partial D} &= 0, \end{aligned} \tag{4.2}$$

where the coefficient κ, and thereby the solution u, depends on the parameter μ.

We seek our solution in the mean sense, i.e., we are looking for our solution u such that

$$u = \arg\min_{v} \int_{\mu}\int_{D} (\frac{1}{2}\kappa|\nabla v|^2 - fv)dxd\mu. \tag{4.3}$$

To numerically solve u from (4.3), we seek an approximation of our solution in the following form:

$$u = \sum_{i=1}^{N_{term}} a_i(\mu)v_i(x), \tag{4.4}$$

where $a_i(\mu)$ depends only on μ, while $v_i(x)$ depends only on x. In the following, we

introduce an algorithm to solve for $a_i(\mu)$ and $v_i(x)$. A similar algorithm has been introduced in [54] to separate among the spatial variables.

As a start, we obtain the first term $a_1 v_1$ by solving the following minimization problem:

$$(a_1, v_1) = \arg\min_{\tilde{a}_1 \tilde{v}_1} \int_\mu \int_D (\frac{1}{2}\kappa |\nabla \tilde{a}_1 \tilde{v}_1|^2 - f \tilde{a}_1 \tilde{v}_1) dx d\mu. \tag{4.5}$$

This can be solved by differentiating with respect to $v_1(x)$ and $a_1(\mu)$, respectively. Specifically, let

$$\begin{aligned} G(a_1(\mu), v_1(x)) &= \int_\mu \int_D (\frac{1}{2}\kappa(x,\mu)|\nabla a_1(\mu) v_1(x)|^2 - f(x) a_1(\mu) v_1(x)) dx d\mu \\ &= \int_\mu \int_D (\frac{1}{2}\kappa(x,\mu)|\nabla v_1(x)|^2 a_1^2(\mu) - f(x) a_1(\mu) v_1(x)) dx d\mu. \end{aligned}$$

For derivative with respect to $v_1(x)$, we consider

$$\begin{aligned} &G(a_1(\mu), v_1(x) + \delta v_1(x)) - G(a_1(\mu), v_1(x)) \\ &\quad = \int_\mu \int_D (\frac{1}{2}\kappa(x,\mu) a_1^2(\mu) \delta \nabla v_1(x) \cdot \nabla v_1(x) - f(x) \delta v_1(x) a_1(\mu)) dx d\mu \\ &\quad = \int_D \Big(\delta \nabla v_1(x) \cdot (\int_\mu \kappa(x,\mu) a_1(\mu) d\mu) \nabla v - \int_\mu a_1(\mu) d\mu f(x) \delta v_1(x) \Big) dx \\ &\quad = \int_D \Big(-\nabla \cdot (\int_\mu \kappa(x,\mu) a_1(\mu) d\mu \nabla v_1(x)) - \int_\mu f(x) a_1(\mu) d\mu \Big) \delta v_1(x) dx. \end{aligned}$$

By setting it to zero we get an equation for v_1,

$$-\nabla \cdot (\int_\mu \kappa a_1^2 d\mu \nabla v_1) = \int_\mu f a_1. \tag{4.6}$$

Also, consider the derivative with respect to $a_1(\mu)$,

$$\begin{aligned} &G(a_1(\mu)+\delta a_1(\mu), v_1(x)) - G(a_1(\mu), v_1(x)) \\ &\qquad = \int_\mu \int_D (\kappa(x,\mu)|\nabla v_1(x)|^2 \delta a_1^2(\mu) - f(x)v_1(x)\delta a_1(\mu))dxd\mu \\ &\qquad = \int_\mu \Big(\int_D (\kappa(x,\mu)|\nabla v_1(x)|^2 a_1(\mu) - f(x)v_1(x))dx \Big)\delta a_1(\mu)d\mu. \end{aligned}$$

By setting it to zero we get an equation for a_1,

$$a_1 = \frac{\int_D f v_1 dx}{\int_D \kappa |\nabla v_1|^2 dx}. \tag{4.7}$$

By the computation above, we arrive at two coupled equations (4.6) and (4.7) for v_1 and a_1. We solve v_1 and a_1 by iterating alternatively between (4.6) and (4.7). When the change of a_1 and v_1 between two neighboring steps is sufficiently small, we stop the iteration.

After obtaining a_1 and v_1, we substitute them into the original PDE (4.2) and then solve for the rest of the solution $u - a_1 v_1$ following the same strategy, which gives us the second tensor product approximation $a_2 v_2$. In particular, a similar minimization problem as (4.5) will be considered for a_2 and v_2. By differentiating with respect to a_2 and v_2 respectively, we will arrive at two coupled equations for a_2 and v_2:

$$-\nabla \cdot (\int_\mu \kappa a_2^2 d\mu \nabla v_2) = \int_\mu f_2 a_2, \tag{4.8}$$

and

$$a_2 = \frac{\int_D f_2 v_2 dx}{\int_D \kappa |\nabla v_2|^2 dx}, \tag{4.9}$$

where

$$f_2 = f + \nabla \cdot (\kappa \nabla a_1 v_1).$$

a_2 and v_2 will also be solved by iterating alternatively between (4.8) and (4.9). We keep doing this until the right hand side f_n is sufficiently small. Eventually, we have our approximation in the form $u = \sum_{i=1}^{N_{term}} a_i(\mu) v_i(x)$.

We summarize the above procedure in Algorithm 4.

Algorithm 4 Multiscale Variable Separation Algorithm

while $(\|f\| > tol_f)$ **do**
  Set $a_i = 1$, $v_i = 1$;
  Set $\delta a = 1$, $\delta v = 1$;
  while $(\delta a > tol_a)$ or $(\delta v > tol_v)$ **do**
    Solve $-\nabla \cdot (\int_\mu \kappa(x,\mu) a_i^2(\mu) d\mu \nabla v(x)) = \int_\mu f a_i$ for v;
    Update $a = \dfrac{\left(\int_D f v_i dx\right)}{\left(\int_D \kappa |\nabla v_i|^2 dx\right)}$;
    Set $\delta a = \|a - a_i\|$, $\delta v = \|v - v_i\|$;
    Set $a_i = a$, $v_i = v$;
  end while
  Update $f = f + \nabla \cdot (\kappa \nabla a_i v_i)$;
  Update $i = i + 1$;
end while

4.3 Separable space-parameter GMsFEM

4.3.1 Overview

We notice that in the above algorithm, for each term $a_i v_i$, we need to solve a PDE alike (4.6) repeatedly. This constitutes a heavy computational load for the overall algorithm. Besides, each term $a_i v_i$ is just a portion of our whole expansion in (4.4), solving them accurately might not benefit us too much on the overall approximation. Therefore, we want to apply the GMsFEM to solve these PDEs in order to accelerate the overall computation. In the following, we briefly explain how to apply the GMsFEM to solve PDE (4.6) in Algorithm 4. The entire procedure is demonstrated in the chart shown in Algorithm 5. More details regarding GMsFEM can be found in [26, 27, 20].

4.3.2 Generalized MsFEM for solving (4.6)

To discretize the PDE (4.6) numerically, the notion of fine and coarse grids introduced in Section 2.4 will be used. See Figure 2.3 for an illustration of neighborhoods and elements subordinated to the coarse and fine discretization. With the above definitions, we proceed to describe the offline-online computational procedure for GMsFEM.

At the offline stage, we first generate the snapshot space, and then a lower-dimensional offline space. This offline space is computed once and used repeatedly at the online stage. At the online stage, we carry out the iterations in Algorithm 4 to solve the terms v_i and a_i alternatively. Specifically, for each given a_i, we solve PDE (4.6) using the constructed offline space and for each given v_i, we evaluate a_i based on the expression (4.7), until the alternation stops.

Algorithm 5 Separable space-parameter GMsFEM

1. Offline computations:
   - Select $\mu_1, \mu_2, \cdots, \mu_N$;
   - Generate snapshot space using $\mu_1, \mu_2, \cdots, \mu_N$;
   - Generate offline space using $\mu_1, \mu_2, \cdots, \mu_N$.
2. Online computations:
   - Compute each v_i and a_i in $u = \sum_{i=1}^{Nterm} a_i(\mu) v_i(x)$;
   - Start global iteration and compute v_i and a_i;
   - The computation of v_i will be performed on a coarse grid using offline space by solving $-\nabla \cdot (\int_\mu \kappa(x,\mu) a_i^2(\mu) d\mu \nabla v(x)) = \int_\mu f a_i$;
   - The computation of a_i will be performed on a coarse grid by $a = \frac{\left(\int_D f v_i dx\right)}{\left(\int_D \kappa |\nabla v_i|^2 dx\right)}$;
   - Use adaptive criteria to select the number of terms and local resolution.

4.3.2.1 Local basis functions

At the offline stage, we first construct a snapshot space $V_{\text{snap}}^{\omega_i}$ on each coarse neighborhood ω_i, which involves solving a set of local problems for a number of selected parameters. This algorithm is the same as the one we discussed in Section 3.2. For completeness, we present the main steps very briefly. We solve the following eigenvalue problems on each ω_i:

$$A(\mu_j)\psi_{l,j}^{\omega_i,\text{snap}} = \lambda_{l,j}^{\omega_i,\text{snap}} S(\mu_j)\psi_{l,j}^{\omega_i,\text{snap}} \quad \text{in } \omega_i, \tag{4.10}$$

where $\{\mu_j\}_{j=1}^{J}$ is a set of selected parameters. The boundary conditions for eigenvalue problem (4.10) can be set as the Neumann type, although other types of boundary conditions can also be considered, see [26, 27, 20].

The matrices $A(\mu_j)$ and $S(\mu_j)$ in (4.10) are defined as

$$\begin{aligned} A(\mu_j) &= [a(\mu_j)_{mn}] = \int_{\omega_i} \kappa(x;\mu_j)\nabla\phi_n \cdot \nabla\phi_m dx, \\ S(\mu_j) &= [s(\mu_j)_{mn}] = \int_{\omega_i} \widetilde{\kappa}(x;\mu_j)\phi_n\phi_m dx, \end{aligned} \tag{4.11}$$

respectively, where ϕ_n denotes the chosen fine-scale basis functions. The choice of $\tilde{\kappa}$ can be found in [26].

We keep the first few eigenfunctions of eigenvalue problem (4.10) corresponding to the lowest eigenvalues as the snapshot basis from ω_i. We then extend these snapshot basis from all the ω_i to the whole domain as our global snapshot basis functions and call the resulting space the snapshot space, denoted as R_{snap}.

After the construction of the snapshot space, we proceed to construct the offline space by performing a dimension reduction within the snapshot space using an auxiliary spectral decomposition. The main objective is to use the offline space to

efficiently (and accurately) construct a set of multiscale basis functions for each μ in the online stage. The procedure for constructing the offline space is similar to the construction of the snapshot space. We set up the local eigenvalue problem in the snapshot space, select the first few eigenfunctions and extend them to the whole domain as our global offline basis functions. Notice that at the offline stage, the bilinear forms are chosen to be parameter independent, hence the coefficients used here are parameter-averaged. The resulting space is called the offline space, denoted as V_{off}.

We use this offline space V_{off} to solve PDE (4.6) repeatedly at the online stage of Algorithm 4.

Remark 4.3.1. *One can further construct an online space V_{on} for each current a_i analogously and solve PDE* (4.6) *in V_{on} at the online stage of Algorithm 4. But in this section, we will use the offline space to save the computation on the online stage without sacrificing the accuracy.*

Remark 4.3.2. *In the discussion above, the global model reduction is performed by approximating the solution with the expansion* (4.4)*, while the local model reduction is performed by GMsFEM. We notice that the idea behind expansion* (4.4) *can also be used in the local model reduction. Instead of generating the snapshot space using realizations of the parameter, we can consider the eigenvalue problem* (4.10) *in the mean sense.*

4.4 Numerical results

In this section, we present several numerical examples to show the performance of the proposed multiscale variable separation method. From our numerical tests, we observe that just with a few expansion terms, our approximation $u = \sum_i a_i v_i$ is

already very close to the best possible solution in our generalized multiscale space. The following results are presented in this section:

- We investigate how additional terms in the global expansion affect the error;
- We show the number of iterations for global solve and its dependence on the choice of coarse space;
- We observe that there is a trade-off between global and local errors, i.e., after some threshold, one needs either increase the number of terms in the global expansion or the dimension of the coarse space.

We present two different examples of testing coefficients. In Example 1, the coefficient $\kappa(x, \mu)$ is taken as the linear combination of two high-contrast permeability fields that are independent of μ (shown in Figure 4.1). In Example 2, the coefficient is generated as the exponential of the permeability field $\bar{\kappa}$ (shown in Figure 4.2), which is also independent of μ .

For each coefficient, we solve the parameter-dependent PDE (2.7) with different approaches and compare their respective solutions, whose definitions are given as below.

- $u_\star^f$: the approximating solution of PDE (2.7) for a specific realization of μ using the fine grid space.
- $u_\star^{ms}$: the approximating solution of PDE (2.7) for a specific realization of μ using the GMsFEM offline space.
- u_n^f: the approximating solution of PDE (2.7) in the mean sense (4.3) with n expansion terms, i.e., $u_n^f = \sum_{i=1}^n a_i(\mu)v_i(x)$, using the fine grid space.

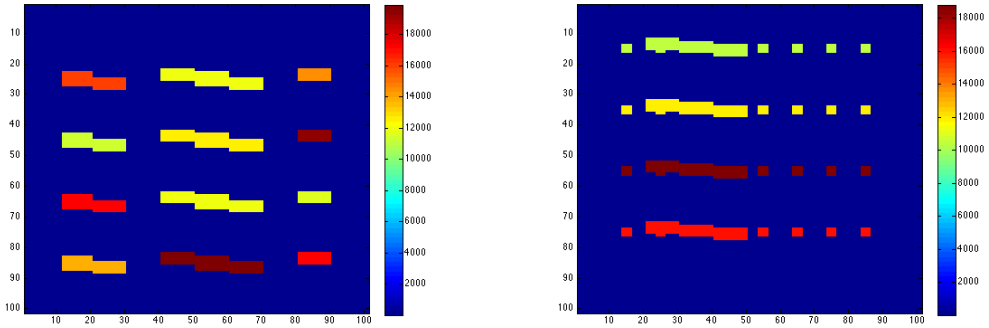
- u_n^{ms}: the approximating solution of PDE (2.7) in the mean sense (4.3) with n expansion terms, using the GMsFEM offline space.

Also, we define the $L^2(\mu, D)$-norm of a function f as $||f||^2_{L^2(\mu,D)} := \int_\mu \int_D f d\mu dx$.

Example 1

In this example, our testing coefficient is generated as follows: We start with two permeability fields that are independent with μ, denoted as $\bar{\kappa}_1$ and $\bar{\kappa}_2$, respectively and shown in Figure 4.1. $\bar{\kappa}_1$ and $\bar{\kappa}_2$ both vary at the range from 1 to 10^4. $\kappa(x, \mu)$ is generated as the linear combination of $\bar{\kappa}_1$ and $\bar{\kappa}_2$:

$$\kappa(x, \mu) = \mu\bar{\kappa}_1 + (1 - \mu)\bar{\kappa}_2.$$



Coefficient $\bar{\kappa}_1$. Coefficient $\bar{\kappa}_2$.

Figure 4.1: Coefficient in Example 1.

We record the results corresponding to different truncation terms in Table 4.1. The errors considered are defined as below:

- The errors of u_n^f for each $n = 1, \ldots 5$, which are computed as $||u_\star^f - u_n^f||_{L^2(\mu,D)}$,

- The errors of u_n^{ms} for each $n = 1, \ldots 5$, which are computed as $||u_\star^f - u_n^{ms}||_{L^2(\mu,D)}$,
- The errors between $u_\star^{ms}$ and u_n^{ms} for each $n = 1, \ldots 5$, which are computed as $||u_\star^{ms} - u_n^{ms}||_{L^2(\mu,D)}$,
- The error of $u_\star^{ms}$, which is computed as $||u_\star^f - u_\star^{ms}||_{L^2(\mu,D)}$. We call this the irreducible error.

The error and N_{iter} in Table 4.1 are the accumulated results up to this term.

Table 4.1: The error and number of iterations for the first 5 terms of the expansion by different methods for Example 1.

Terms	Error	N_{iter}
1	2.18E-02	2
2	1.42E-02	9
3	3.39E-04	12
4	7.38E-05	17
5	6.11E-06	22

(a) $||u_\star^f - u_n^f||_{L^2(\mu,D)}$

Terms	Error	N_{iter}
1	9.46E-02	2
2	9.33E-02	9
3	9.24E-02	13
4	9.24E-02	25
5	9.24E-02	30

(b) $||u_\star^f - u_n^{ms}||_{L^2(\mu,D)}$

Terms	Error
1	2.05E-02
2	1.32E-02
3	6.00E-04
4	4.00E-04
5	1.00E-04

(c) $||u_\star^{ms} - u_n^{ms}||_{L^2(\mu,D)}$

The irreducible error $||u_\star^f - u_\star^{ms}||_{L^2(\mu,D)}$ corresponding to the fixed offline space is 9.24E-02.

In Table 4.1a, we can see that the error drops as the number of iterations increases, reaching the 5th term within 22 iterations and the error reaches $6.11E-06$. This shows that our separation method converges and works well with the fine solution. In Table 4.1c, we can see that the difference between the result of our separation algorithm and the full global model by GMsFEM also reduces fast as the number

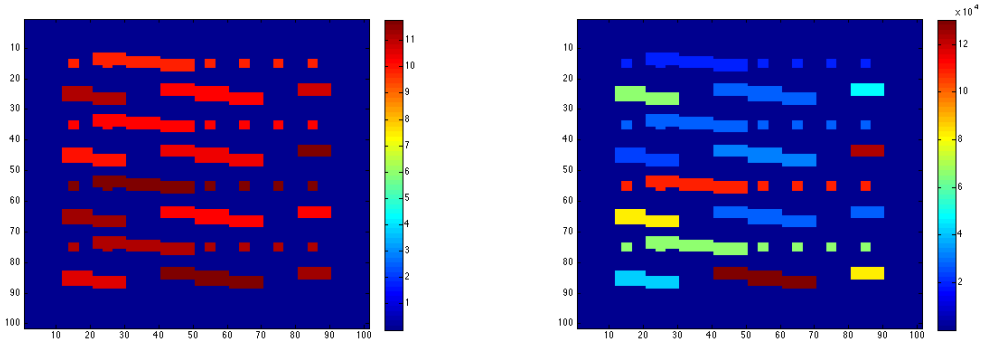
of terms and iterations increases, reaches $1E-04$ at the 5th term. This means our truncated expansion solutions are already very close to best possible solution in our generalized multiscale space. Table 4.1b shows the total error of our approach. We can see that the error also drops by adding terms but it reaches the irreducible error at the 3rd term and does not change a lot after that. This is due to the fixed offline space we use in GMsFEM. But all these results show that we can get a quite accurate approximation of the full parametric space efficiently using the proposed multiscale variable separation method.

Example 2

In this example, our testing coefficient is generated as follows. We start with a fixed permeability fields that is independent with μ, denoted as $\bar{\kappa}$ and shown in the left of Figure 4.2. $\kappa(x,\mu)$ is generated as:

$$\kappa(x,\mu) = e^{(\mu\bar{\kappa})}.$$

The background of $\bar{\kappa}$ is 0 such that the background of κ is 1.



Coefficient $\bar{\kappa}$. Coefficient κ when $\mu = 1$.

Figure 4.2: Coefficient in Example 2.

We record the results corresponding to different truncation terms in Table 4.2. The definitions of errors considered is the same as in Example 1. The error and N_{iter} in Table 4.2 are the accumulated results up to this term.

Table 4.2: The error and number of iterations for the first 5 terms of the expansion by different methods for Example 2.

Terms	Error	N_{iter}
1	2.64E-01	2
2	6.44E-02	8
3	2.91E-02	22
4	1.52E-02	35
5	5.52E-03	40

(a) $||u_\star^f - u_i^f||_{L^2(\mu,D)}$

Terms	Error	N_{iter}
1	2.79E-01	2
2	1.57E-01	8
3	1.52E-01	20
4	1.51E-01	45
5	1.51E-01	72

(b) $||u_\star^f - u_i^{ms}||_{L^2(\mu,D)}$

Terms	Error
1	2.38E-01
2	4.46E-02
3	1.89E-02
4	1.21E-02
5	1.06E-02

(c) $||u_\star^{ms} - u_i^{ms}||_{L^2(\mu,D)}$

The irreducible error $||u_\star^f - u_\star^{ms}||_{L^2(\mu,D)}$ corresponding to the fixed offline space is 1.51E-01.

With exponential coefficient case, we observe similar results as Example 1. In Table 4.2a, the error drops as the number of iteration increases. The 5th term is reached within 22 iterations for the FEM case, and the error reaches $5.52E-03$. In Table 4.2c, we observe that the difference between the result of our separation algorithm and the full global model solved by GMsFEM reduces fast as the number of terms and iterations increases, reaches $1E-02$ at the 5th term, which means our separation method also converges and the best possible solution is reached rapidly in our generalized multiscale space. For the total error in 4.2b, the error drops very close to the irreducible error within 4 terms and 45 iterations. But it also stays there

due to the fixed offline space we use. These results also show that we can get an accurate approximation of the full parametric space efficiently using the proposed separation algorithm with GMsFEM local model reduction. Hence, from the result above we know that solving with our proposed multiscale variable separation method also produces a good approximation of the solution.

4.4.1 Discussion on adaptivity and applications

In this section, we briefly discuss how one can appropriately choose the number of terms in the stochastic expansion and the corresponding coarse-space dimension.

We define the errors $r = u_\star^f - u_n^{ms}$, $r^{ms} = u_\star^f - u_\star^{ms}$, $r_n^{ms} = u_\star^{ms} - u_n^{ms}$. Notice that r^{ms} can be decomposed as

$$r^{ms} = u_\star^f - u_\star^{ms} = (u_\star^f - u_n^{ms}) - (u_\star^{ms} - u_n^{ms}) = r - r_n^{ms}.$$

One can show that

$$\|r - r_n^{ms}\| \leq R,$$

where $R = f_n + \mathrm{div}(\kappa \nabla r_n^{ms})$ is the residual. Note that $R = f + \mathrm{div}(\kappa \nabla u_\star)$, so this term represents irreducible error.

Also, we have

$$\|u_\star^f - u_n^{ms}\| \leq \|u_\star^{ms} - u_n^{ms}\| + \|u_\star^f - u_\star^{ms}\|.$$

We denote $\|u_\star^{ms} - u_n^{ms}\|$ to be the truncation error and $\|u_\star^f - u_\star^{ms}\|$ to be the multiscale error.

For the efficiency of the computation, it is important to balance the multiscale error and the truncation error. For example, one does not need to achieve a very

high accuracy in the truncation process if multiscale error is large. In general, the multiscale space for each term can be chosen independently. Here, we can fix the offline space for all v_i's calculations.

The following error indicators can be used. For the multiscale error, we can estimate $\|u_\star^{ms} - u_\star^f\|$ by calculating this quantity for some very few selected μ's and the right-hand-side chosen in the offline step. Or one can also use $f + div(\kappa \nabla u_{ms})$ in L^2 norm as an error indicator. However, this estimator may not be robust for high-contrast problems (see [20]). For truncation error, we can solve r_n using multiscale basis functions for selected μ's and estimate $\|r_n\|$.

5. AN APPROXIMATE BAYESIAN APPROACH IN HIGH-CONTRAST FLOW ESTIMATION

5.1 Introduction

In this section, we integrate the MLMC methods with the Approximate Bayesian computation (ABC) to form a multilevel ABC algorithm for the estimation of the underlying quantities of interest in the flow model (2.3). Mixed GMsFEM is coupled in this method, which provides the mapping between the levels following the idea of MLMC-GMsFEM framework introduced in Section 3.

Approximate Bayesian computation (ABC) [22] is a novel way to approximate Bayesian computations, where the likelihood does not have a closed form and is expensive to compute. Given a prior distribution on parameters, ABC methods use realizations from the joint distribution of the data and the prior of the parameters, to approximate the likelihoods without explicitly evaluating them. Using the known data generating mechanisms, given the parameter, a candidate data set is generated. The parameter value is feasible when it is close enough to the observed data. The closeness in that approximating step controls the degree of approximation. There is no burden of likelihood computation in ABC methods, and it is flexible, hence adaptable in many different scenarios. Some of the recent developments include the adaptive approximation [10] and the application in linear regression model [13].

The underlying fine-scale problem is a coupled flow and transport equations. The flow equations require mass conservative discretization to avoid spurious saturation values in the convection equation. For this reason, we use mixed finite element discretization and its multiscale version. A mixed GMsFEM is described in this section for solving the parameter-dependent two-phase flow equation with transport (2.3).

In this method, the pressure field is approximated by piecewise constant basis functions, while the velocity field is approximated by multiscale basis. The multiscale basis are constructed with a offline-online procedure similar to the GMsFEM introduced in Section 3.2. In particular, the snapshot vectors are computed in coarse blocks that share a common face. In the snapshot space, a local spectral problem is solved to identify multiscale basis functions for the velocity field. A hierarchy of approximations to the solution is provided by the resulting mixed GMsFEM, which can be integrated into the Multilevel Monte Carlo framework.

In this section, we implement ABC rejection algorithm in a multiscale scenario, and the underlying quantities of interest are estimated in a multilevel set-up which follows the idea of Multilevel Monte Carlo (see Section 3.3). A multilevel ABC (MLABC) algorithm is proposed and the quantities of interest are estimated from the posterior sample mean generated by the MLABC. We use mixed GMsFEM to solve the forward simulations in the algorithm, where the dimension of the online multiscale space can be chosen to form the MLABC levels. Some relevant results and justifications related to the MLABC algorithm are also proved. Results about the stationary property of the chain at each level generated by the MLABC algorithm are shown and the approximation property is discussed. We implement the proposed methodology in both single-phase and two-phase flow problems. The proposed MLABC shows improvement of computational cost. Also, the simulation results show that a higher number of observations are accepted from the higher (finer) level. Thus, it suggests the scope of a multilevel ABC algorithm in the context of the flow estimation.

The section is arranged as follows. In Section 5.2, we introduce the mixed GMsFEM for solving two-phase flow problem with transport. In Section 5.3, we discuss ABC thoroughly, propose the MLABC algorithm, and write down the stationary

distribution in our multilevel case. In Section 5.4, we compare the approximation and computational gain of the proposed MLABC algorithm with other methods in case of both single- and two-phase flow examples.

5.2 Mixed GMsFEM as a multilevel solver

GMsFEM for the single-phase flow problem (2.7) is introduced in Section 3.2, which achieves an efficient forward model simulation, and also provides a hierarchy of approximations which can be used for constructing Multilevel Monte Carlo estimates. In this section, a mixed GMsFEM for the two-phase flow model problem (2.3) will be introduced.

5.2.1 Preliminaries

We consider the following two-phase flow problem with transport in mixed formulation:

$$\begin{aligned} \kappa(x,\mu)^{-1}v + \nabla u = 0 \quad &\text{in } D, \\ \text{div } (v) = f \quad &\text{in } D, \end{aligned} \tag{5.1}$$

with non-homogeneous Neumann boundary condition $v \cdot n = g$ on ∂D, where $\kappa(x,\mu)$ is the permeability field, D is the computational domain, and n is the ourward unit-normal vector on ∂D. In the mixed GMsFEM considered in this section, we construct the basis functions for the velocity field v with an offline-online procedure. For the pressure field u, we will use piecewise constant approximations.

First, we introduce the following definitions of coarse and fine grids to descretize the model problem (5.1). Let $\mathcal{T}^H$ be a conforming partition of the computational domain D into finite elements (triangles, quadrilaterals, tetrahedra, etc.). This partition is referred to as the coarse grid and assume that each coarse subregion is further partitioned into a connected union of fine grid blocks. We denote the fine grid partition by $\mathcal{T}^h$. We use $\xi^H := \bigcup_{i=1}^{N_e}\{E_i\}$ (where N_e is the number of coarse edges) to

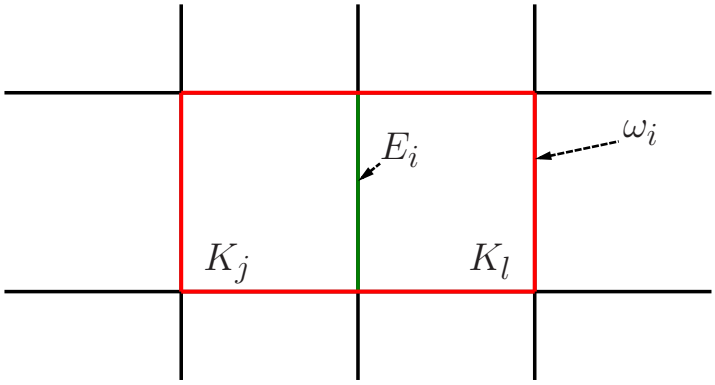


Figure 5.1: Illustration of a coarse edge E_i and the associated coarse neighborhood ω_i.

denote the set of all edges of the coarse mesh $\mathcal{T}^H$, and define the neighborhood ω_i corresponding to the coarse edge E_i by

$$\omega_i = \bigcup \{K_j \in \mathcal{T}^H; \quad E_i \in \partial K_j\}, \tag{5.2}$$

where K_j denotes the coarse element. See Figure 5.1 for an illustration of neighborhoods and elements subordinated to the coarse and fine discretization.

Let Q_H be the space of piecewise constant functions with respect to the coarse grid $\mathcal{T}^H$, the approximation of the pressure u will be obtained in this space. We define the multiscale space for velocity field v as the linear span of all local multiscale basis functions corresponding to edge E_i, which is denoted as

$$V_H = \bigoplus_{\xi^H} \{\psi_i\}.$$

We also define $V_H^0 = V_H \cap \{v \in V_H : v \cdot n = 0 \,\text{on}\, \partial D\}$. The mixed GMsFEM is to

find $(v_H, u_H) \in V_H \times Q_H$ such that

$$\begin{aligned} \int_D \kappa^{-1} v_H \cdot w_H - \int_D \operatorname{div}(w_H) u_H = 0, \quad \forall w_H \in V_H^0, \\ \int_D \operatorname{div}(v_H) q_H = \int_D f q_H, \quad \forall q_H \in Q_H, \end{aligned} \tag{5.3}$$

where $v_H \cdot n = g_H$ on ∂D, and for each coarse edge $E_i \in \partial D$, we have

$$\int_{E_i} (g_H - g) \psi_j \cdot n = 0$$

for all basis functions ψ_j corresponding to the edge E_i.

With the above definition, we proceed to deiscribe the offline-online computational procedure which will construct the multiscale space V_H for the approximation of the velocity field v.

5.2.2 The construction of multiscale basis functions

At the offline stage, we first generate the snapshot space by selecting some realizations, μ_1, ..., μ_J, and then a lower-dimensional offline space is computed. This offline space is computed once and used repeatedly at the online stage. We first construct a snapshot space $V_{\text{snap}}^{E_i}$ on each coarse edge E_i, which involves solving a set of local problems for a number of selected parameters. Specifically, we will find $(v_{l,j}^{(i)}, u_{l,j}^{(i)})$ by solving the following problem on each coarse neighborhood ω_i corresponding to the edge E_i:

$$\begin{aligned} \kappa(\mu_j)^{-1} v_{l,j}^{(i)} + \nabla u_{l,j}^{(i)} = 0 \quad \text{in } \omega_i, \\ \operatorname{div}(v_{l,j}^{(i)}) = \alpha_{l,j}^{(i)} \quad \text{in } \omega_i, \end{aligned} \tag{5.4}$$

subject to the boundary condition $v_{l,j}^{(i)} \cdot n_i = 0$ on $\partial\omega_i$, where n_i denotes the outward unit-normal vector on $\partial\omega_i$. Here $\{\mu_j\}_{j=1}^J$ is a set of selected parameters. The above problem (5.4) will be solved separately in the coarse-grid blocks forming ω_i, therefore

we will need an extra boundary condition on E_i. Notice that E_i can be written as a union of fine-grid edges, $E_i = \bigcup_{l=1}^{L_i} e_l$, where L_i is the total number of fine-grid edges on E_i and e_l denotes a find-grid edge. We define a piecewise constant function $\delta_l^{(i)}$ on E_i as

$$\delta_l^{(i)} = \begin{cases} 1, & \text{on } e_l, \\ 0, & \text{on other fine grid edges on } E_i, \end{cases}$$

for $l = 1, 2, \cdots, L_i$. The remaining boundary condition on the coarse edge E_i for the local problem (5.4) is then taken as

$$v_{l,j}^{(i)} \cdot m_i = \delta_l^{(i)}, \tag{5.5}$$

where m_i is a fixed unit-normal vector on E_i. The constant $\alpha_{l,j}^{(i)}$ in (5.4) is chosen to satisfy the compatibility condition $\int_{K_n} \alpha_{l,j}^{(i)} = \int_{E_i} \delta_l^{(i)}$ for all $K_n \subset \omega_i$.

The collection of the solutions of the above local problems generates the snapshot space. Let $\psi_{l,j}^{i,\text{snap}} := v_{l,j}^{(i)}$ be the snapshot fields and the snapshot space $V_{\text{snap}}^{(i)}$ corresponding to the coarse edge E_i is defined by

$$V_{\text{snap}}^{(i)} = \text{span}\{\psi_{l,j}^{i,\text{snap}} \ : \ 1 \le j \le J, \ 1 \le l \le L_i\}.$$

To simplify the notation, we will use

$$V_{\text{snap}}^{(i)} = \text{span}\{\psi_j^{i,\text{snap}} \ : \ 1 \le j \le M_{\text{snap}}^{(i)}\},$$

where $M_{\text{snap}}^{(i)} = J \times L_i$ is the total number of snapshot fields corresponding to E_i.

After the construction of the snapshot space, we proceed to construct the offline space by performing a dimension reduction within the space of snapshots using some local spectral problems following the general framework of [26]. We consider the

local spectral problem: find a real number $\lambda \geq 0$ and a function $v \in V_{\text{snap}}^{(i)}$ such that

$$a_i(v, w) = \lambda s_i(v, w), \quad \forall w \in V_{\text{snap}}^{(i)}. \tag{5.6}$$

Here we take

$$\begin{aligned} a_i(v, w) &= \int_{\omega_i} \overline{\kappa}^{-1} v \cdot w, \\ s_i(v, w) &= \int_{E_i} \big[u_v\big]\big[u_w\big], \end{aligned} \tag{5.7}$$

where $\overline{\kappa}$ is domain-based, parameter-averaged (equal weight mean) coefficient, (v, u_v), (w, u_w) are solutions of the local problem (5.4), and $\big[u\big]$ denotes the jump of the function u. To generate the offline space, we choose the smallest $M_{\text{off}}^{(i)}$ eigenvalues to (5.6), and take the corresponding eigenvectors, $\Psi_k^{i,\text{off}}$. In the space of snapshots, by setting $\psi_k^{i,\text{off}} = \sum_j \Psi_{kj}^{i,\text{off}} \psi_j^{i,\text{snap}}$ for $k = 1, \cdots, M_{\text{off}}^{(i)}$, we form the reduced snapshot space, where $\Psi_{kj}^{i,\text{off}}$ are the coordinates of the vector $\Psi_k^{i,\text{off}}$. The global offline space is then

$$V_{\text{off}} = \text{span}\{\psi_k^{i,\text{off}} \ : \ 1 \leq k \leq M_{\text{off}}^{(i)}, \ 1 \leq i \leq N_e\},$$

where N_e is the total number of edges in the domain. To simplify the notation, we will use

$$V_{\text{off}} = \text{span}\{\psi_k^{\text{off}} \ : \ 1 \leq k \leq M_{\text{off}}\},$$

where $M_{\text{off}} = \sum_{i=1}^{N_e} M_{\text{off}}^{(i)}$ is the total number of offline basis functions.

One can solve the model problem using the offline space at the online stage. But here we will further construct an online space V_{on} for each given input parameter μ. We want it to be a low-dimensional subspace of the offline space for computational efficiency. In particular, we seek a subspace of the offline space by solving a same eigenvalue problem to (5.6), but it is solved in the offline space V_{off}, and in (5.7) we use the input $\kappa(\mu)$ instead of $\overline{\kappa}$. Similarly, eigenvectors corresponding to the

smallest M_{on} eigenvalues will be chosen. We note that at the online stage, the bilinear forms as in (5.7) will be formed with the input parameter μ, which means that they are parameter-dependent. M_{on} can be chosen such that different size of the online multiscale space will be formed, which provides the hierarchy of approximations. This online space V_{on} is used to solve the model problem.

Remark 5.2.1. *A postprocessing technique is applied in our experiments to obtain conservative velocity fields on the fine-grid. We notice that the solution of* (5.3) *satisfies*

$$\int_{\partial K} v_H \cdot n = \int_K f$$

for every coarse-grid block K. When f has fine-scale oscillation in some coarse blocks, the velocity field needs to be postprocessed in these coarse blocks. A post-processed velocity $v_h^\star$ will be constructed such that conservation on the fine grid is obtained, as following:

$$\int_{\partial \tau} (v_h^\star \cdot n) = \int_{\partial \tau} f, \quad \forall \tau \in \mathcal{T}^h.$$

In particular, for each coarse-grid block K, we find $(v_h^\star, u_h^\star) \in V_h(K) \times Q_h(K)$ such that $v_h^\star \cdot n = v_H \cdot n$ and

$$\begin{aligned} \int_D \kappa^{-1} v_h^\star \cdot w_h - \int_D div(w_h) u_h^\star = 0, \quad \forall w_h \in V_h^0, \\ \int_D div(v_h^\star) q_h = \int_D f q_h, \quad \forall q_h \in Q_h. \end{aligned} \tag{5.8}$$

The offline-online procedure of the mixed GMsFEM is introduced as above for the model problem (5.1). To solve the two-phase flow and transport problem, we solve the flow equation for each time step and use the fine-scale velocity to advance the saturation front. In general, one can solve the saturation equation on a coarse grid

for much faster forward simulations which will be less accurate due to the upscaling errors in the saturation. These approximations can be used in MLABC, which will be investigated in our future work. More details and discussions about mixed GMsFEM can be found in [19].

5.3 MCMC and ABC approximations

Given the observed data F_{obs} (fraction flow/pressure), our aim is to sample from the posterior and estimate the permeability field κ. Because of the PDE structure and the error in the observations, a direct solution is not feasible. A Bayesian framework is used for this purpose. The main idea of the Bayesian approach is to model the field of interest by some appropriate spatial process, and exploit the PDE structure and the error distribution to find the distribution of the permeability field given the observed data.

As mentioned earlier, we do not observe the true data of the quantity of interest. The observed data points are convoluted with some random error. We assume the true data $F(\kappa)$ is observed with some random error ϵ, and the error terms are assumed to be independent normal:

$$\begin{aligned} F_{obs} &= F(\kappa) + \epsilon, \\ \epsilon &\sim N(0, \sigma_f^2). \end{aligned} \tag{5.9}$$

The permeability field κ is parametrized with KLE with the parameters θ_k's as introduced in Section 2.5. Under the assumption of independent priors, the posterior distribution is

$$\pi(\kappa) = p(\kappa|F_{obs}) \propto p(F_{obs}|\kappa)p(\kappa), \tag{5.10}$$

as described in Section 2.3. Because of the underlying PDE, a closed form posterior sampling is not possible. Simple Metropolis Hasting method amounts to solving the PDE at each draw in fine scale which is computationally intensive. A multiscale method is therefore appropriate to save the computational cost, using KLE for parametrization of the random field. A similar parametrization can be adapted for the ABC approximation. Instead of direct sampling using MCMC, in the ABC method, we generate data for the candidate κ, and check if it is close to the true observed data in some appropriate metric. Using ABC samples, we implement MLABC and estimate the κ.

5.3.1 Approximate Bayesian computation

In MCMC, the stationary distribution is achieved by a Gibbs sampling or a similar algorithm, where the balanced equation corresponding to the stationary distribution is satisfied. The Metropolis-Hastings MCMC algorithm is outlined in Algorithm 2. Though this sampling is very useful for many practical purpose to obtain the posterior distribution as the stationary distribution, it may encounter difficulties in case of an inverse problem where we need to solve a underlying PDE at each draw. To reduce the computational cost, an alternative route can be employed through an approximate Bayesian approach.

The main idea behind ABC is to generate pseudo observation $\mathbf{Z}$ at each step given the proposed κ^*. If the distance between the observed data F_{obs} and $\mathbf{Z}$, $d(F_{obs}, \mathbf{Z})$, is less than some chosen $\beta > 0$, then we accept κ^* in the second step. The acceptance criterion for the second step is given in the algorithm below. If F_{obs} has a discrete distribution, then $\beta = 0$ gives the exact posterior density. The choice of the tolerance limit β is subjective and it determines the order of the approximation.

- By the transition kernel $q(.)$ we move $\kappa_m \to \kappa^*$;

- Generate $\mathbf{Z}$ from the likelihood $p(.|\kappa)$ with $\kappa = \kappa^*$;
- If $d(F_{obs}, \mathbf{Z}) < \beta$, the move from $\kappa_m \to \kappa^*$ is accepted with probability

$$\gamma(\kappa_m, \kappa_{m+1)} = \min\{1, \frac{q(\kappa_m|\kappa^*)p(\kappa^*)}{q(\kappa^*|\kappa_m)p(\kappa_m)}\};$$

- If accepted, $\kappa_{m+1} = \kappa^*$, otherwise, $\kappa_{m+1} = \kappa_m$.

The resulting stationary distribution is not exactly $p(\kappa|F_{obs})$. Instead it is given by

$$\pi_\beta(\kappa, \mathbf{Z}|F_{obs}) = \frac{p(\kappa, \mathbf{Z})\mathbf{I}_{d(F_{obs}, \mathbf{Z})<\beta}}{\int_{A_\beta^{F_{obs}} \times \kappa} p(\kappa, \mathbf{Z}) d\mathbf{Z} d\kappa},$$

where $A_\beta^{F_{obs}} = \{\mathbf{Z} : d(F_{obs}, \mathbf{Z}) < \beta\}$. Note that if β approaches zero, the stationary distribution approaches the posterior distribution. Therefore, we obtain an approximate posterior and the choice of β determines the trade-off between computational cost and accuracy.

5.3.2 ABC for multilevel MCMC

In a multilevel scenario, we use a similar approach to ordinary ABC. Let $\beta_1, \ldots \beta_l$ the cut off of the affinity between F_{obs} and the pseudo data points $\mathbf{Z}$'s at the l-th level. The β_i's from different level may be different and produce an approximation of true posterior at each level. The algorithm is formally given in Algorithm 6. Here, κ parametrized by θ_k's is our parameter. The prior for level l is given by the prior on θ_k's and is denoted by $p_l(\kappa) = p(\kappa)$.

Let ξ^l be the set such that the target density at step level l has density value greater than zero. Also, let D^l be the set where the proposal density is greater than zero. Under the normality of (5.10), and the proposal density given by random

perturbation around the current value

$$q_0(\kappa|\kappa_n) = \kappa_n + \delta\epsilon_n,$$

the condition $\xi = \xi_l = D_m = D$ is satisfied. Also, the resulting chain is aperiodic at each level l ([34]). Finally, we can state the following result about the resulting stationary distribution.

Lemma 1. *Under the model* (5.10) *and prior* (3.22)*, the stationary distribution corresponding to the MLABC algorithm in Algorithm 6 is given by*

$$\pi_l^{\beta}(\kappa, \mathbf{Z}|F_{obs}) = \frac{p_l(\kappa, \mathbf{Z})\mathbf{I}_{d(F_{obs},Z)<\beta}}{\int\int_{A_{\beta,\kappa}^{F_{obs}}} p(\kappa, \mathbf{Z})d\mathbf{Z}d\kappa},$$

where $\mathbf{Z}$ *is a draw from the field value* κ *at level* l*, and* $A_{\beta,\kappa}^{F_{obs}}$ *denotes the set of observations* $y = F_l(\kappa) + \epsilon$ *which has a distance less than* $\beta = \beta_l$ *from the observed* F_{obs}*.*

Proof. Let c be the value of the integral in the denominator. Without loss of generality, we assume $\rho_{l+1}(\kappa_{l+1}^m, \kappa) \leq 1$. For a transition $(\kappa_{l+1}^m, \mathbf{Z}) \to (\kappa, \mathbf{Z}')$, the transition probability for $\kappa_{l+1}^m \neq \kappa$ is

$$Q_{l+1}(\kappa_{l+1}^m, \kappa) = q_l(\kappa|\kappa_{l+1}^m)\rho_{l+1}(\kappa_{l+1}^m, \kappa).$$

Let $\mathbf{I}$ denotes the indicator function that $\mathbf{Z}$ and $\mathbf{Z}'$ is in $A_{\eta,\kappa_l^{m+1}}^{F_{obs}}$ and $A_{\eta,\kappa}^{F_{obs}}$, respectively. Writing down the stationary condition for level l from Algorithm 6, we have for level

$l+1$,

$$\begin{aligned}
&\pi^{\eta}_{l+1}(\kappa^m_{l+1}, \mathbf{Z}|F_{obs})Q_{l+1}(\kappa^m_{l+1}, \kappa)p_{l+1}(\mathbf{Z}'|\kappa)\mathbf{I} \\
&\quad = c^{-1}p_{l+1}(\mathbf{Z}|\kappa^m_{l+1})p_{l+1}(\kappa^m_{l+1})q_l(\kappa|\kappa^m_{l+1})\rho_{l+1}(\kappa^m_{l+1}, \kappa)p_{l+1}(\mathbf{Z}'|\kappa)\mathbf{I} \\
&\quad = c^{-1}p_{l+1}(\mathbf{Z}|\kappa^m_{l+1})p_{l+1}(\kappa^m_{l+1})q_l(\kappa|\kappa^m_{l+1})\frac{q_l(\kappa^m_{l+1}|\kappa)p_{l+1}(\kappa)}{q_l(\kappa|\kappa^m_{l+1})p_{l+1}(\kappa^m_{l+1})}p_{l+1}(\mathbf{Z}'|\kappa)\mathbf{I} \\
&\quad = c^{-1}p_{l+1}(\mathbf{Z}'|\kappa)p_{l+1}(\kappa)q_l(\kappa^m_{l+1}|\kappa)p_{l+1}(\mathbf{Z}|\kappa^m_{l+1})\mathbf{I} \\
&\quad = c^{-1}p_{l+1}(\mathbf{Z}'|\kappa)p_{l+1}(\kappa)\rho_{l+1}(\kappa, \kappa^m_{l+1})p_{l+1}(\mathbf{Z}|\kappa)\mathbf{I} \\
&\quad = \pi^{\eta}_{l+1}(\kappa, \mathbf{Z}'|F_{obs})Q_{l+1}(\kappa, \kappa^m_{l+1})p_{l+1}(\mathbf{Z}|\kappa^m_{l+1})\mathbf{I}.
\end{aligned}$$

For $\kappa^m_{l+1} = \kappa$, the condition is satisfied trivially. □

Remark 5.3.1. *Therefore, the marginal posterior of κ would be*

$$\pi^{\beta}_{l}(\kappa|F_{obs}) = \frac{p(\theta)t_A(\beta_l, \kappa)}{\int\int_{A^{F_{obs}}_{\beta}} p(\kappa, \mathbf{Z})d\mathbf{Z}d\kappa}, \tag{5.11}$$

where

$$t_A(\beta_l, \kappa) = \int_{A^{F_{obs}}_{\beta_l, \kappa}} p(\mathbf{Z}|\kappa)d\mathbf{Z}.$$

The choice of β_i's determine the order of approximation of the posterior distribution at the l-th level. With β_i approaching zero, the true posterior is achieved. If some regularity conditions are satisfied, the ABC estimate $\pi^{\beta_l}_{l}(\kappa|F_{obs})$ approaches the true posterior $\pi_l(\kappa|F_{obs})$ at the l-th level in some appropriate topology, with β_l converging to zero. A simple result for weak topology can be stated in the form of the following lemma.

Lemma 2. *Suppose $p(\mathbf{Z}|\kappa)$ is a continuous function in $\mathbf{Z}$ and $\sup_{\mathbf{Z},\kappa} p(\mathbf{Z}|\kappa) < \infty$. Then, the ABC estimate $\pi^{\beta_l}_{l}(\kappa|F_{obs})$ converges to the true posterior $\pi_l(\kappa|F_{obs})$ at the*

l-th level in terms of weak topology, as β_l converges to zero.

Proof. Let $p(\mathbf{Z}|\kappa) < c_o < \infty$ for all κ and $\mathbf{Z}$. Let $V_n(\beta)$ be the volume of the n dimensional set $A_\beta^{F_{obs}} = \{\mathbf{Z} : d(F_{obs}, \mathbf{Z}) < \beta\}$. Here, n is the length of the observed vector. Let M be the prior dimension used in KLE. Given $\delta > 0$, there exists a compact set K_c such that $p(K_c') < \delta$.

Given $\delta' > 0$, there exists $\beta' > 0$ such that for all $\beta < \beta'$ and $\mathbf{Z} \in A_\beta^{F_{obs}}$, we have

$$|p(F_{obs}|\kappa) - p(\mathbf{Z}|\kappa)| < \delta'$$

for $\kappa \in K_c$. Let N_β and D_β be the numerator and denominator of (5.11). Hence, we have

$$|D_\beta - m(F_{obs})V_n(\beta)| < 2\delta' V_n(\beta) + 2V_n(\beta)c_o\delta.$$

Here, $m(\cdot)$ denotes the marginal density. Also,

$$|N_\beta(\kappa) - p(F_{obs}, \kappa)V_n(\beta)| \leq \delta' V_n(\beta)$$

on K_c, and on K_c',

$$|N_\beta(\kappa) - p(F_{obs}, \kappa)V_n(\beta)| < 2c_o V_n(\beta).$$

Thus, on K_c,

$$\pi_l^\beta(\kappa|F_{obs}) = \pi_l^\kappa(\kappa|F_{obs}) + O(\delta) + O(\delta').$$

Now for any bounded continuous function of κ, $g(\kappa)$, we have

$$E_{\pi_l^\beta(\kappa|F_{obs})}(g(\kappa)) = E_{\pi_l^\beta(\kappa|F_{obs})}(g(\kappa)\mathbf{I}_{K_c}) + c\delta',$$

and hence,

$$E_{\pi_l^{\eta}(\kappa|F_{obs})}(g(\kappa)) = E_{\pi_l(\kappa|F_{obs})}(g(\kappa)) + O(\delta) + O(\delta').$$

As δ and δ' can be arbitrarily small, it concludes the proof. □

Remark 5.3.2. *For practical purpose, the choice of β_l can be data driven. One approach may be using some small quantile of the posterior distribution of the proposed metric. This can be done by solving the system in the coarse grid, which is computationally cheaper.*

5.3.3 MLABC implementation

The idea of multilevel MCMC is important in context of drawing posterior samples from different levels with higher levessl associated with higher costs. A function of the parameter such as posterior mean may be of interest. The lowest level estimate may be cheaper but less accurate. Thus, using different levels, we can estimate the subsequent improvements of adding a new level and estimate the target function as a telescoping sum.

We start with the telescopic sum

$$\begin{aligned}\mathbb{E}_{\pi_L}[F_L] &= \int F_L(x)\pi_L(x)dx \\ &= \int F_0(x)\pi_0(x)dx + \sum_{l=1}^{L}\int (F_l(x)\pi_l(x) - F_{l-1}(x)\pi_{l-1}(x))dx,\end{aligned}$$

where π_l denotes the approximated target distribution at level l, and π_0 is our initial level. We note that after the initial level each expectation involves two measures, π_l and π_{l-1}. We can rewrite the integration using a product measure and we will get,

$$\mathbb{E}_{\pi^L}[F_L] = \mathbb{E}_{\pi_0}[F_0] + \sum_{l=1}^{L}\mathbb{E}_{\pi_l,\pi_{l-1}}[F_l - F_{l-1}]. \tag{5.12}$$

Algorithm 6 Multilevel ABC

Pick $\beta_1, \beta_2, \ldots, \beta_L$;
Given κ_m, draw a trial proposal κ from distribution $q(\kappa|\kappa_1^m) = q_0(\kappa|\kappa_1^m)$;
Compute the acceptance probability

$$\rho_1(\kappa_1^m, \kappa) = \min\left\{1, \frac{q_0(\kappa_1^m|\kappa)p_1(\kappa)}{q_0(\kappa|\kappa_1^m)p_1(\kappa_1^m)}\right\};$$

Compute $d(F_{obs}, F_1(\kappa))$;
if $d(F_{obs}, F_1(\kappa)) \leq \beta_1$ **then**
  Accept κ with probability ρ_1;
end if
for $l = 1 : L - 1$ **do**
  if κ is accepted at level l **then**
    Form the proposal distribution q_l (on the $l + 1$th level) by

$$q_l(\kappa|\kappa_{l+1}^m) = \rho_l(\kappa_{l+1}^m, \kappa)q_{l-1}(\kappa|\kappa_{l+1}^m) + \delta_{\kappa_{l+1}^m}(1 - \int \rho_l(\kappa_{l+1}^m, \kappa)q_{l-1}(\kappa|\kappa_{l+1}^m)d\kappa_{l+1}^m);$$

    Compute the acceptance probability

$$\rho_{l+1}(\kappa_{l+1}^m, \kappa) = \min\left\{1, \frac{q_l(\kappa_{l+1}^m|\kappa)p_{l+1}(\kappa)}{q_l(\kappa|\kappa_{l+1}^m)p_{l+1}(\kappa_{l+1}^m)}\right\};$$

    Compute $d(F_{obs}, F_l(\kappa))$;
    if $d(F_{obs}, F_l(\kappa)) \leq \beta_l$ **then**
      Accept κ with probability ρ_{l+1};
    end if
    if κ is accepted **then**
      $\kappa_{l+1}^{m+1} = \kappa$ and go to next level (if $l = L - 1$, accept κ and set $\kappa_L^{m+1} = \kappa$);
    else
      $\kappa_{l+1}^{m+1} = \kappa_{l+1}^m$, and break.
    end if
  end if
end for

The idea of our multilevel method is to estimate each term of the right hand side of equation (5.12) independently. The proposed ABC estimator can be used to estimate each term in (5.12). To create different levels, we adapt the proposal distribution $q(\kappa|\kappa_m)$ to the target distribution $\pi(\kappa)$ using the GMsFEM with different sizes of the online space. The process modifies the proposal distribution $q(\kappa|\kappa_m)$ by incorporating the online coarse-scale information. Let $F_l(\kappa)$ be the pressure/water-cut computed by solving online coarse problem at level l for a given κ. The target distribution $\pi(\kappa)$ is approximated on level l by $\pi_l(\kappa)$, with $\pi(\kappa) \equiv \pi_L(\kappa)$. Here, we have

$$\pi_l(\kappa) \propto \exp\left(-\frac{||F_{obs} - F_l(\kappa)||^2}{2\sigma_l^2}\right) \times p(\kappa). \tag{5.13}$$

In the algorithm we still keep the same offline space for each level. From level 0 to level L, we increase the size of the online space as we go to a higher level, which means for any levels l, $l+1 \leq L$, samples of level l are cheaper to generate than that of level $l+1$. This idea underlies the cost reduction using the multilevel estimator. Hence, the posterior distribution for coarser levels $\pi_l, l = 0, \ldots, L-1$ do not have to model the measured data as faithfully as π_L, which in particular implies that by choosing suitable value of σ_l^2 it is easier to match the result $F_l(\kappa)$ with the observed data. We denote the number of samples at level l by M_l, where we will have $M_0 \leq \cdots \leq M_L$. As was discussed above, our quantity of interest can be approximated by the telescopic sum (5.12).

5.4 Numerical results

In this section, we present some numerical examples of the multilevel ABC method. Suppose the permeability field $\kappa(x, \mu)$, where x is defined on the unit square $\Omega = [0, 1]^2$. We assume that the permeability field $\kappa(x, \mu)$ is a log-normal process and its covariance function is known. We use the Karhunen-Loéve expansion (KLE)

to parameterize the permeability field. Then, we apply the multilevel ABC method described earlier and compare the simulation results with multilevel MCMC method. Both single-phase and two-phase flow cases will be considered. First, we briefly recall the permeability parameterization, and then we present numerical results.

In our examples, the permeability field κ is assumed to follow a log-normal distribution with a known spatial covariance, with the correlation function $R(x, y)$ given by

$$R(x, y) = \sigma^2 \exp\Big(-\frac{|x_1 - y_1|^2}{2l_1^2} - \frac{|x_2 - y_2|^2}{2l_2^2}\Big), \tag{5.14}$$

where l_1 and l_2 are the correlation lengths in x_1- and x_2-direction, respectively, and $\sigma^2 = \mathbb{E}[Y^2]$ is a constant that determines the variation of the permeability field.

5.4.1 Single-phase flow

For the first set of numerical tests, we consider the stationary, single-phase flow model (2.7) with $f \equiv 1$ and linear boundary conditions. We apply GMsFEM to solve the forward problem, with a 5×5 coarse grid and a 50×50 fine grid. The "true" data F_{obs} is obtained by generating a reference permeability field, solving the forward problem with the GMsFEM, and evaluating the pressure at nine selected points away from the boundary.

The prior permeability distribution $p(\kappa)$ is parameterized by KL expansion as introduced in Section 2.5. We keep $N = 5$ terms in the KL expansion. The Gaussian field is of correlation length $l_1 = l_2 = 0.1$. Our proposal distribution is a random walker sampler in which the proposal distribution depends on the previous value of the permeability field and is given by $q(\kappa|\kappa_n) = \kappa_n + \delta\epsilon_n$, where ϵ_n is a random perturbation with mean zero and unit variance, and δ is a step size. The random perturbations are imposed on the θ_k coefficients in the KL expansion. The distance $d(F_{obs}, F(\kappa))$ in Algorithm 6 is taken as the L^2-norm. The acceptance criterion β_l is

Table 5.1: Number of accepted samples at each level in MCMC, 3-level MCMC and 3-level ABC.

	MCMC	3-level MCMC	3-level ABC
Total	5534	4723	4197
Level 1		2299	1754
Level 2		1573	1176
Level 3	1000	1000	1000

decided by 25% quantile of the L^2-norm of pseudo data value $\mathbf{Z}$'s from the draws of a coarse level MCMC sampling.

To compare the performance of different methods, we test the same problem with MCMC, 3-level MCMC and 3-level ABC, respectively. For the multilevel methods, the numbers of eigenvalues to generate the online space at each level are taken to be 4,8 and 16. For the 3-level ABC, we set the acceptance criterion $\beta = 0.06$ for all the levels. We run the algorithms until 1000 total samples pass the final level of acceptance. The number of accepted samples are listed in Table 5.1. We can observe that for more expensive levels, it is more probable that a proposed sample will be accepted. In Figure 5.2, we plot the reference permeability field, initial permeability field, and the result posterior mean of the 3 methods. We notice that all 3 sample means are very close to the reference field. Hence, the proposed multi-level ABC method is of higher efficiency.

5.4.2 Two-phase flow and transport

In these simulations, we present the performance of the multilevel ABC method with two-phase flow and transport problems. We consider the model problem (2.3) on $D = (0, 1)^2$ with zero Neumann boundary condition. The flow equation is solved by mixed GMsFEM [19] with a 50×50 fine grid and a 5×5 coarse grid, and the

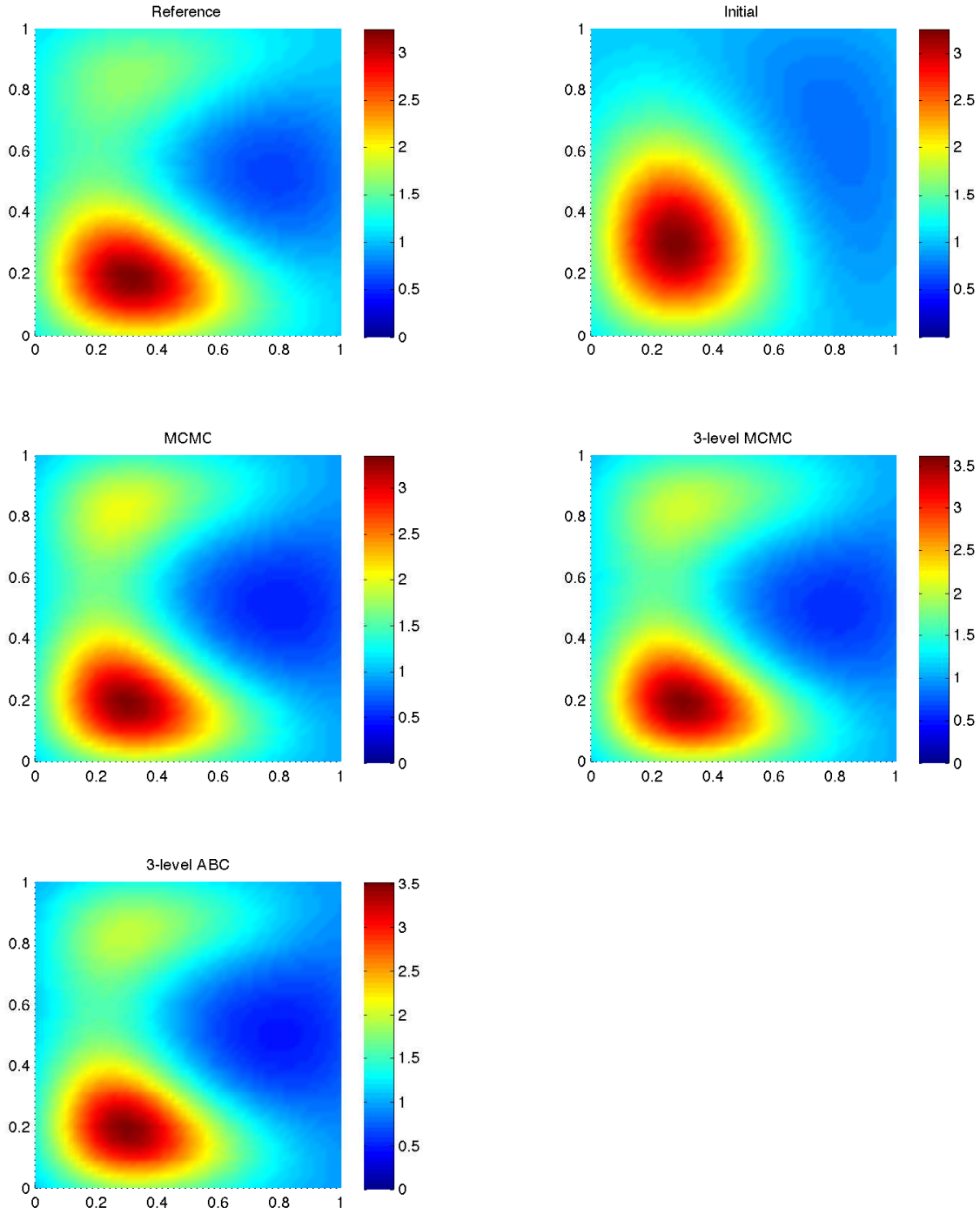


Figure 5.2: Top left: The true log-permeability field. Top right: The initial log-permeability field. Middle Left/ Right: The mean of the sampled log-permeability field from MCMC and 3-level MCMC. Bottom: The mean of the sampled log-permeability field from 3-level ABC.

saturation equation is solved on the fine grid by the finite volume method. The prior permeability distribution $p(\kappa)$ is also parametrized by KLE as above. The quantity of interest is taken to be the water-cut function F. One injecter at $(0, 0)$ and one producer at $(1, 1)$ are considered when we run the forward model in the reference permeability field to get the fractional flow. Two sets of Gaussian covariance functions are considered in this simulation.

- Isotropic Gaussian field with correlation lengths $l_1 = l_2 = 0.1$, and a KL dimension 8;
- Anisotropic Gaussian field with correlation lengths $l_1 = 0.05, l_2 = 0.1$, and a KL dimension 8.

Our proposal distribution is a random walker sampler in which the proposal distribution depends on the previous value of the permeability field and is given by $q(\kappa|\kappa_n) = \kappa_n + \delta\epsilon_n$, where ϵ_n is a random perturbation with mean zero and unit variance, and δ is a step size. The random perturbations are imposed on the coefficients in the KL expansion. The distance $d(F_{obs}, F(\kappa))$ in Algorithm 6 is taken as the L^2-norm. The acceptance criterion β_l is decided by 20% quantile of L^2-norm of pseudo data value $\mathbf{Z}$'s from draws of a coarse level MCMC sampling.

Again, to compare the performance of different methods we run MCMC, 2-level MCMC and 2-level ABC on the same problem. For the multilevel methods, the numbers of eigenvalues to generate the online space at each level are taken to be 2 and 8. For the 2-level ABC test, we set different acceptance criterion β_l for each level. In the isotropic case, $\beta_1 = 0.38$ and $\beta_2 = 0.1$. In the anisotropic case $\beta_1 = 0.4$ and $\beta_2 = 0.15$. We run the algorithms until 2500 total samples are accepted on the second-to-last level.

Table 5.2: Number of accepted samples at each level in MCMC, 2-level MCMC and 2-level ABC for the isotropic case.

	MCMC	2-level MCMC	2-level ABC
Total	2500	8119	4810
Level 1		2500	2500
Level 2	462	497	1218

For the isotropic case, the number of accepted samples are listed in Table 5.2. We can see that the 2-level ABC method has a higher acceptance rate than the other two methods on the more expensive level. In Figure 5.3, we plot the reference fraction flow, the initial sample and the sample mean of each method. We observe that the mean estimate of the fractional flow is very close to the observed data. In Figure 5.4, we plot some of the accepted permeability realizations that have passed all levels of computation in the 2-level ABC method. For the anisotropic case, the number of accepted samples are listed in Table 5.3. We plot the reference fraction flow, the initial sample and the sample mean of each method in Figure 5.5. And in Figure 5.6, we plot some of the accepted permeability realizations that have passed all levels of computation in the 2-level ABC method. Similarly, we observe that the 2-level ABC method has a higher acceptance rate than the other two methods on the more expensive level and the mean estimate of the fractional flow is very close to the observed data. We also notice that the run time of the 2-level ABC of both the tests are around 15% faster than the 2-level MCMC simulations. By these results, we can see that we achieve higher computational efficiency with the proposed multi-level ABC method.

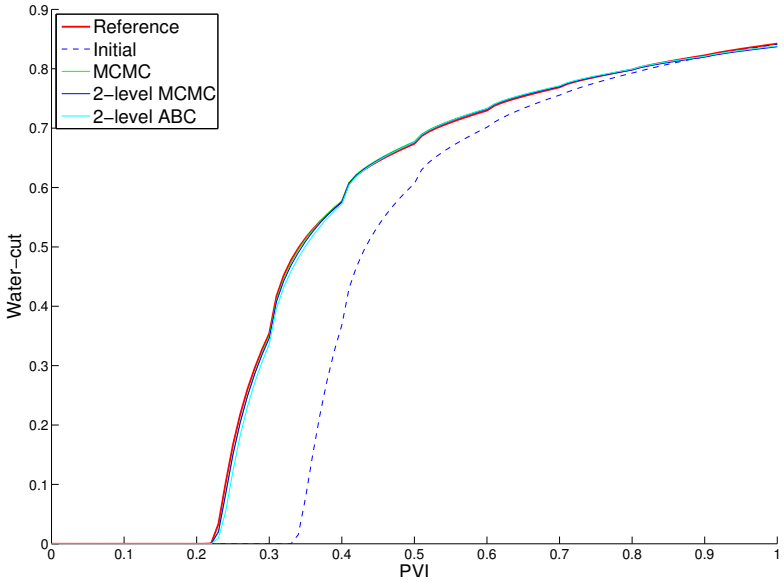


Figure 5.3: Red line designates the reference water cut, the dash line designates the initial water cut and the green, the blue and the cyan line designate water cuts corresponding to mean of the sampled water cuts from MCMC, 2-level MCMC and 2-level ABC, respectively.

Table 5.3: Number of accepted samples at each level in MCMC, 2-level MCMC and 2-level ABC for the anisotropic case.

	MCMC	2-level MCMC	2-level ABC
Total	2500	5418	3377
Level 1		2500	2500
Level 2	549	828	1717

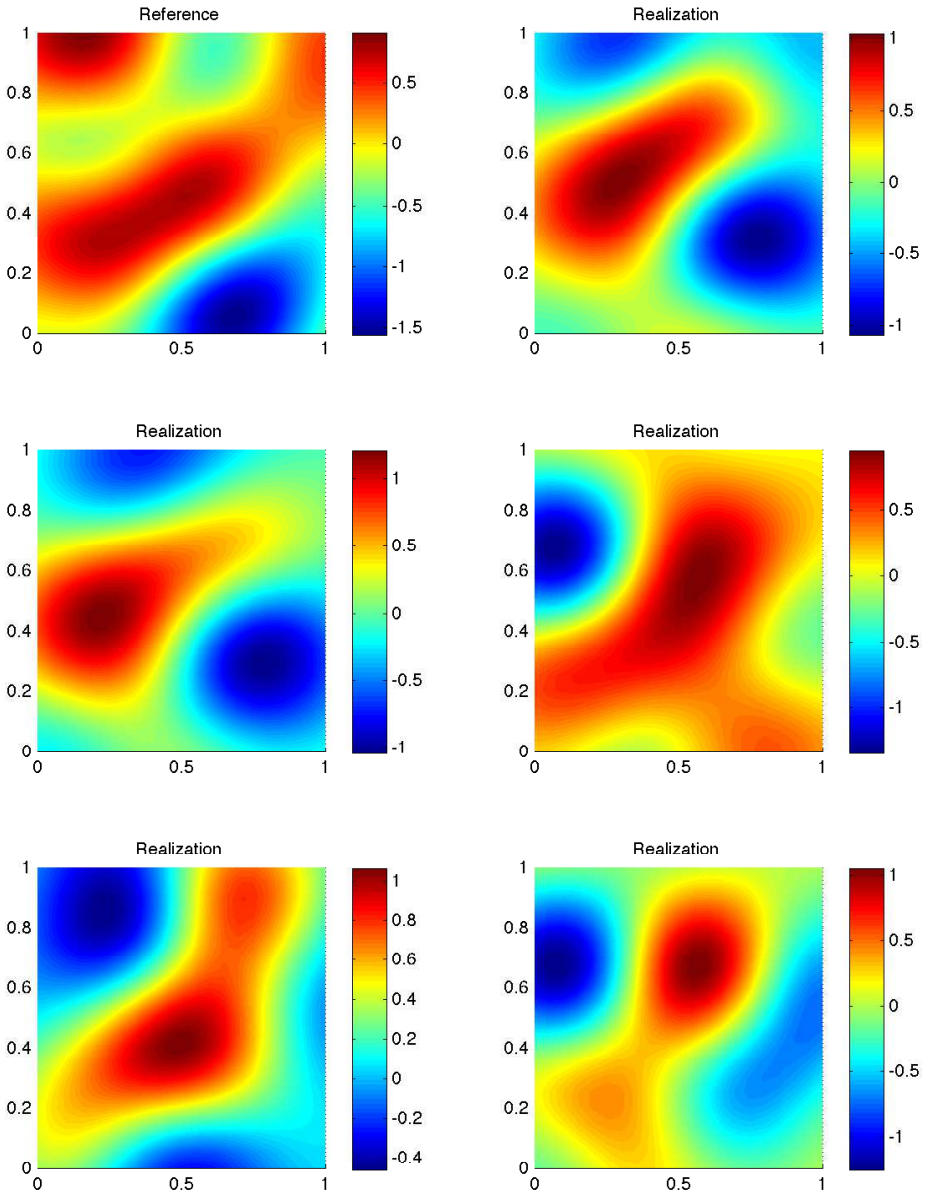


Figure 5.4: Top left: The true log-permeability field. Others: The last five accepted realizations of the log-permeability field for the isotropic case.

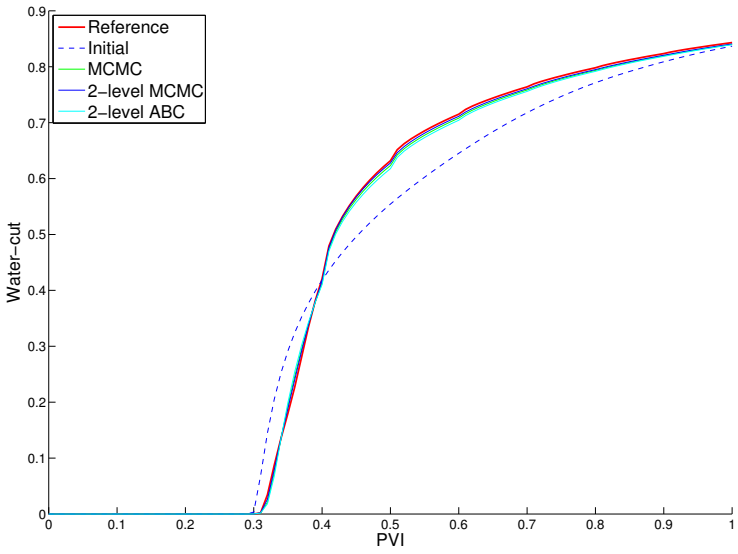


Figure 5.5: The red line designates the reference water cut, the dash line designates the initial water cut and the green, the blue and the cyan line designate water cuts corresponding to the mean of the sampled water cuts from MCMC, 2-level MCMC and 2-level ABC, respectively.

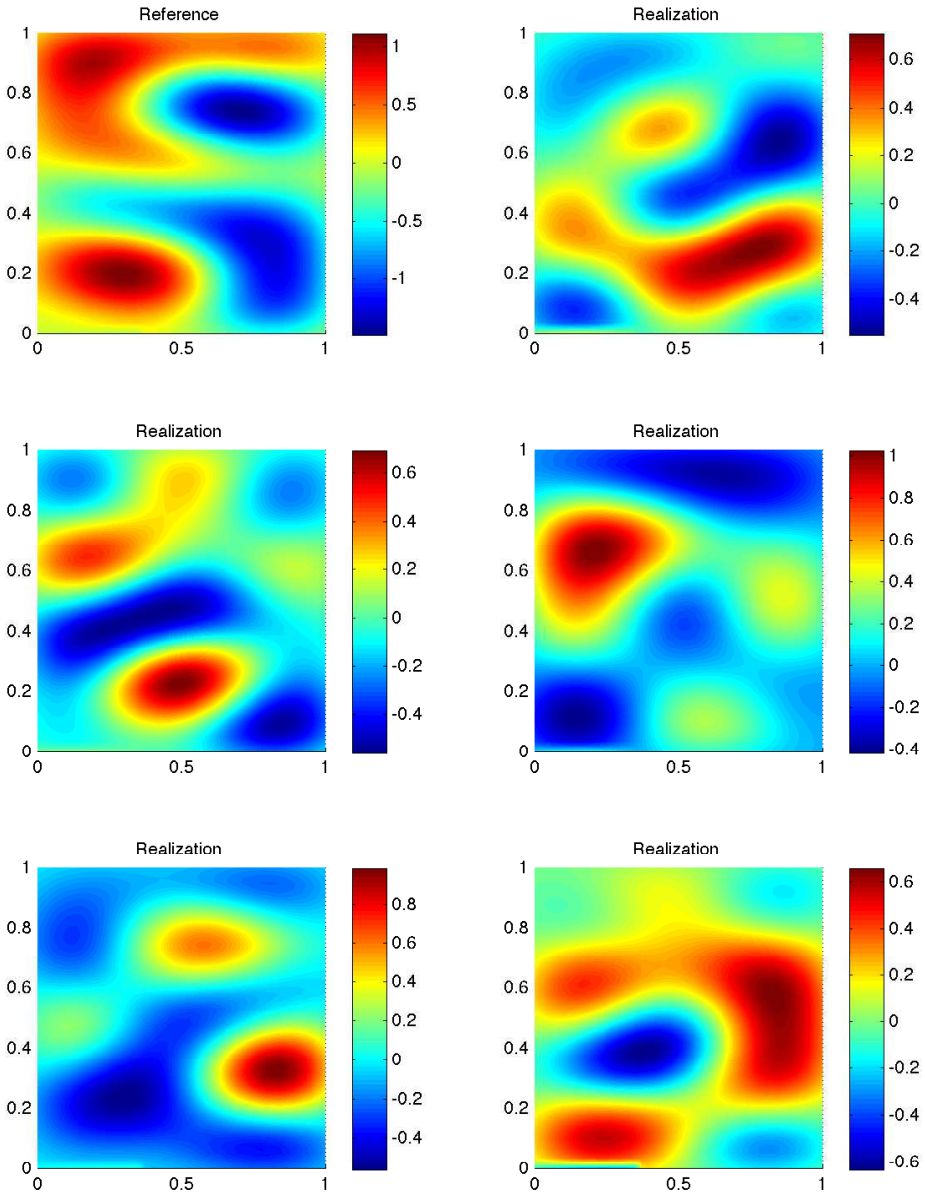


Figure 5.6: Top left: The true log-permeability field. Others: The last five accepted realizations of the log-permeability field for the anisotropic case.

6. SUMMARY

In this dissertation, the uncertainty quantification problems of subsurface flow models in heterogeneous porous media have been studied. Several efficient methods have been proposed for both the forward and inverse uncertainty quantification for multiscale flow problems. In particular, we have proposed a robust framework for the uncertainty quantification for the quantities of interest for high-contrast single-phase flow problems, which combines the Generalized Multiscale Finite Element Method (GMsFEM) and Multilevel Monte Carlo (MLMC) methods. Within this context, GMsFEM provides a hierarchy of approximations at varying levels of accuracies and computational costs, and MLMC offers an efficient way to estimate quantities of interest using samples on respective levels. The number of basis functions in the online GMsFEM stage may readily and adaptively be modified in order to adjust the computational cost and accuracy, and efficiently generate samples at different levels. In particular, it is inexpensive to generate samples through smaller dimensional online spaces with less accuracy, and it is expensive to generate samples through larger dimensional online spaces with a higher level of accuracy. As such, a suitable number of samples at different levels allows us to leverage the expensive computations at finer levels and the less expensive computations at coarser levels, while retaining the accuracy of the final estimates on the output quantities of interest. We additionally describe a Multilevel Markov Chain Monte Carlo (MLMCMC) inverse modeling technique, which sequentially screens the proposal with different levels of GMsFEM approximations. In particular, the method reduces the number of evaluations that are required at finer levels, while combining the samples at varying levels to arrive at an accurate estimate. A number of numerical examples are presented in

order to illustrate the efficiency and accuracy of the multilevel methods as compared to standard Monte Carlo estimates. The analysis and examples of the proposed methodology offer a seamless integration between the flexibility of the GMsFEM online space construction along with the multilevel features of MLMC methods.

We have also proposed an approach for forward uncertainty quantification problems, which combines the separation of variables methods with the GMsFEM to achieve efficiency. In this framework, we construct multiscale basis functions as the product of multiscale spatial basis functions and special parameter-dependent functions, which are computed based on an energy minimization associated with the global problem. The solution is approximated by an expansion of terms with separated parameter and space variables. Via this separation, we decompose the high dimensional problem into a sequence of low dimensional problems, in terms of the parameter and the space variables. GMsFEM is utilized in repeatedly solving the corresponding parameter-independent PDEs in the separation approach, in which the offline space is precomputed and can be used for multiple solves at the online stage. A local model reduction is achieved by GMsFEM for each term in the global expansion, which brings computational savings. In the numerical results, we observe that a fast convergence can be achieved with only a few terms in the expansion.

We have combined the idea of MLMCMC with Approximate Bayesian Computation (ABC) to develop a Multilevel ABC (MLABC) framework for inverse uncertainty quantification problems in both single-phase and two-phase flow problems coupled with transport. A mixed GMsFEM for the parameter-dependent problem is introduced to provide a hierarchy of approximations, which forms the varying levels in MLABC for two-phase flow with transport models. In ABC methods, realizations from the joint distribution of the data and the prior of the parameters are used to approximate the likelihoods without explicitly evaluating them, which

gains computational efficiency without sacrificing the accuracy of the estimation. In the numerical results presented in the dissertation, we have compared the proposed MLABC with standard MCMC estimation and the MLMCMC. We observe that the number of acceptance in the finer and more expensive level has increased significantly in MLABC. This gives a scope for more accurate posterior analysis with more posterior samples from the finer grid. Based on our simulations, the proposed MLABC is faster than the MCMC based methods. Analysis also gives a theoretical justification of the MLABC approximation.

In future, we plan to consider the following research directions. (1) The KL expansion for the permeability field is well-suited for Gaussian distributions. Further work can be done to investigate non-Gaussian fields with different parametrization and different priors in the Bayesian process. (2) The models presented in this dissertation consider single-phase and two-phase systems. The proposed uncertainty quantification methods can be extended to handle other models including seismic wave equation and multi-phase flow with gravity and so on. (3) Rigorous adaptivity techniques for the space-parameter separation method will help to increase the efficiency of the proposed approaches. Approaches that allow appropriately choosing the number of terms in the stochastic expansion and the corresponding coarse-space dimension can improve the methods considered in the dissertation. (4) A more extensive study about the choice of approximation parameters in MLABC will help to consider more complex systems.

REFERENCES

[1] J. E. Aarnes, S. Krogstad, and K.-A. Lie. A hierarchical multiscale method for two-phase flow based upon mixed finite elements and nonuniform coarse grids. *Multiscale Modeling & Simulation*, 5(2):337–363, 2006.

[2] A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modelling of complex fluids: Part II: Transient simulation using space-time separated representations. *Journal of Non-Newtonian Fluid Mechanics*, 144(2):98–121, 2007.

[3] T. Arbogast, G. Pencheva, M. F. Wheeler, and I. Yotov. A multiscale mortar mixed finite element method. *Multiscale Modeling & Simulation*, 6(1):319–346, 2007.

[4] I. Babuška and R. Lipton. Optimal local approximation spaces for generalized finite element methods with application to multiscale problems. *Multiscale Modeling & Simulation*, 9(1):373–406, 2011.

[5] I. Babuška, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM review*, 52(2):317–355, 2010.

[6] I. Babuška, R. Tempone, and G. E. Zouraris. Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM Journal on Numerical Analysis*, 42(2):800–825, 2004.

[7] G. Bal, I. Langmore, and Y. Marzouk. Bayesian inverse problems with Monte Carlo forward models. *Inverse Problems and Imaging*, 7(1):81–105, 2013.

[8] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathématique. Académie des Sciences. Paris*, 339(9):667–672, 2004.

[9] A. Barth, C. Schwab, and N. Zollinger. Multi-level Monte Carlo finite element method for elliptic PDEs with stochastic coefficients. *Numerische Mathematik*, 119(1):123–161, 2011.

[10] M. A. Beaumont, J.-M. Cornuet, J.-M. Marin, and C. P. Robert. Adaptive approximate Bayesian computation. *Biometrika*, page asp052, 2009.

[11] M. Berveiller, B. Sudret, and M. Lemaire. Stochastic finite element: a non intrusive approach by regression. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, 15(1-3):81–92, 2006.

[12] L. Biegler, G. Biros, O. Ghattas, M. Heinkenschloss, D. Keyes, B. Mallick, L. Tenorio, B. van Bloemen Waanders, K. Willcox, and Y. Marzouk. *Large-scale inverse problems and quantification of uncertainty*, volume 712. John Wiley & Sons, 2011.

[13] M. G. Blum and O. François. Non-linear regression models for Approximate Bayesian Computation. *Statistics and Computing*, 20(1):63–73, 2010.

[14] S. Boyaval. Reduced-basis approach for homogenization beyond the periodic setting. *Multiscale Modeling & Simulation*, 7(1):466–494, 2008.

[15] R. E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. *Acta numerica*, 7:1–49, 1998.

[16] Y. Cheng and G. Lin. An improved method of VFSA for global optimization, in progress. 2014.

[17] F. Chinesta, A. Ammar, A. Leygue, and R. Keunings. An overview of the proper generalized decomposition with applications in computational rheology. *Journal of Non-Newtonian Fluid Mechanics*, 166(11):578–592, 2011.

[18] J. A. Christen and C. Fox. Markov chain Monte Carlo using an approximation. *Journal of Computational and Graphical Statistics*, 14(4):795–810, 2005.

[19] E. T. Chung, Y. Efendiev, and C. S. Lee. Mixed Generalized Multiscale Finite Element Methods and Applications. *Multiscale Modeling & Simulation*, 13(1):338–366, 2015.

[20] E. T. Chung, Y. Efendiev, and G. Li. An adaptive GMsFEM for high-contrast flow problems. *Journal of Computational Physics*, 273:54–76, 2014.

[21] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup. Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Computing and Visualization in Science*, 14(1):3–15, 2011.

[22] K. Csilléry, M. G. Blum, O. E. Gaggiotti, and O. François. Approximate Bayesian Computation (ABC) in practice. *Trends in ecology & evolution*, 25(7):410–418, 2010.

[23] G. Dagan and S. P. Neuman. *Subsurface flow and transport: a stochastic approach*. Cambridge University Press, 2005.

[24] M. K. Deb, I. M. Babuška, and J. T. Oden. Solution of stochastic partial differential equations using Galerkin finite element techniques. *Computer Methods in Applied Mechanics and Engineering*, 190(48):6359–6372, 2001.

[25] 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.

[26] Y. Efendiev, J. Galvis, and T. Y. Hou. Generalized Multiscale Finite Element Methods (GMsFEM). *Journal of Computational Physics*, 2013.

[27] Y. Efendiev, J. Galvis, G. Li, and M. Presho. Generalized Multiscale Finite Element Methods: Nonlinear elliptic equations. submitted to CiCP, arxiv.org/abs/1304.5188, 2013.

[28] Y. Efendiev, J. Galvis, and F. Thomines. A systematic coarse-scale model reduction technique for parameter-dependent flows in highly heterogeneous media and its applications. *Multiscale Modeling & Simulation*, 10(4):1317–1343, 2012.

[29] Y. Efendiev, J. Galvis, and X.-H. Wu. Multiscale finite element methods for high-contrast problems using local spectral basis functions. *Journal of Computational Physics*, 230(4):937–955, 2011.

[30] 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):155–174, 2006.

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

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

[33] Y. Efendiev and T. Y. Hou. *Multiscale Finite Element Methods: Theory and applications*. Springer Science & Business Media, New York, 2009.

[34] Y. Efendiev, B. Jin, M. Presho, and X. Tan. Multilevel Markov chain Monte Carlo method for high-contrast single-phase flow problems. *Communications in*

Computational Physics, 17(01):259–286, 2015.

[35] H. P. Flath, L. C. Wilcox, V. Akçelik, J. Hill, B. van Bloemen Waanders, and O. Ghattas. Fast algorithms for Bayesian uncertainty quantification in large-scale linear inverse problems based on low-rank partial Hessian approximations. *SIAM Journal on Scientific Computing*, 33(1):407–432, 2011.

[36] J. Galvis and Y. Efendiev. Domain decomposition preconditioners for multiscale flows in high-contrast media. *Multiscale Modeling & Simulation*, 8(4):1461–1483, 2010.

[37] R. G. Ghanem and P. D. Spanos. Polynomial chaos in stochastic finite elements. *Journal of Applied Mechanics*, 57(1):197–202, 1990.

[38] R. G. Ghanem and P. D. Spanos. *Stochastic Finite Elements: a Spectral Approach*. Springer-Verlag, New York, 1991.

[39] M. B. Giles. Improved multilevel Monte Carlo convergence using the Milstein scheme. In *Monte Carlo and quasi-Monte Carlo methods*, pages 343–358. Springer, Berlin, 2008.

[40] M. B. Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56(3):607–617, 2008.

[41] V. Ginting, F. Pereira, M. Presho, and S. Wo. Application of the two-stage Markov chain Monte Carlo method for characterization of fractured reservoirs using a surrogate flow model. *Computational Geosciences*, 15(4):691–707, 2011.

[42] A. Giunta, J. McFarland, L. Swiler, and M. Eldred. The promise and peril of uncertainty quantification using response surface approximations. *Structures and Infrastructure Engineering*, 2(3-4):175–189, 2006.

[43] I. G. Graham, T. Y. Hou, O. Lakkis, and R. Scheichl. *Numerical analysis of multiscale problems*, volume 83. Springer Science & Business Media, 2012.

[44] S. Heinrich. Multilevel Monte Carlo methods. In S. Margenov, J. Waśniewski, and P. Yalamov, editors, *Lecture Notes in Computer Science*, volume 2179, pages 58–67. Springer-Verlag, Berlin, 2001.

[45] 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):169–189, 1997.

[46] T. J. R. Hughes, G. R. Feijóo, L. Mazzei, and J.-B. Quincy. The variational multiscale method—a paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166(1-2):3–24, 1998.

[47] K. Ito and B. Jin. *Inverse Problems: Tikhonov Theory and Algorithms*. World Scientific, Singapore, 2014.

[48] L. Jiang, I. Mishev, and Y. Li. Stochastic mixed multiscale finite element methods and their applications in random porous media. *Computer Methods in Applied Mechanics and Engineering*, 199(41):2721–2740, 2010.

[49] L. Jiang and M. Presho. A resourceful splitting technique with applications to deterministic and stochastic multiscale finite element methods. *Multiscale Modeling & Simulation*, 10(3):954–985, 2012.

[50] M. C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, pages 425–464, 2001.

[51] M. C. Kennedy and A. OHagan. Supplementary details on Bayesian calibration of computer models. Technical report, Internal Report. URL http://www. shef.

ac. uk/˜ st1ao/ps/calsup. ps, 2001.

[52] C. Ketelsen, R. Scheichl, and A. Teckentrup. A hierarchical multilevel Markov chain Monte Carlo algorithm with applications to uncertainty quantification in subsurface flow. *arXiv preprint arXiv:1303.7343*, 2013.

[53] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical analysis*, 40(2):492–515, 2002.

[54] C. Le Bris, T. Lelievre, and Y. Maday. Results and questions on a nonlinear approximation approach for solving high-dimensional partial differential equations. *Constructive Approximation*, 30(3):621–651, 2009.

[55] G. Li, J. Galvis, and K. Shi. A generalized multiscale finite element method for the Brinkman equation. *Journal of Computational and Applied Mathematics*, 280:294–309, 2015.

[56] F. Liang. Annealing stochastic approximation Monte Carlo algorithm for neural network training. *Machine Learning*, 68(3):201–233, 2007.

[57] F. Liang. Annealing evolutionary stochastic approximation Monte Carlo for global optimization. *Statistics and Computing*, 21(3):375–393, 2011.

[58] F. Liang, C. Liu, and R. Carroll. *Advanced Markov chain Monte Carlo methods: learning from past samples*, volume 714. John Wiley & Sons, 2011.

[59] F. Liang, C. Liu, and R. J. Carroll. Stochastic approximation in Monte Carlo computation. *Journal of the American Statistical Association*, 102(477):305–320, 2007.

[60] J. S. Liu. *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2008.

[61] M. Loève. *Probability theory. II.* Springer-Verlag, New York, fourth edition, 1978.

[62] X. Ma and N. Zabaras. An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations. *Journal of Computational Physics*, 229(10):3884–3915, 2010.

[63] J.-M. Marin, P. Pudlo, C. P. Robert, and R. J. Ryder. Approximate Bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180, 2012.

[64] P. Marjoram, J. Molitor, V. Plagnol, and S. Tavaré. Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26):15324–15328, 2003.

[65] Y. Marzouk, H. N. Najm, and L. A. Rahn. Stochastic spectral methods for efficient Bayesian solution of inverse problems. *Journal of Computational Physics*, 224(2):560–586, 2007.

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

[67] A. Mondal, Y. Efendiev, B. Mallick, and A. Datta-Gupta. Bayesian uncertainty quantification for flows in heterogeneous porous media using reversible jump Markov chain Monte Carlo methods. *Advances in Water Resources*, 33(3):241–256, 2010.

[68] N. C. Nguyen. A multiscale reduced-basis method for parametrized elliptic partial differential equations with multiple scales. *Journal of Computational Physics*, 227(23):9807–9822, 2008.

[69] A. Nouy. A generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 196(45):4521–4537, 2007.

[70] A. Nouy. Proper generalized decompositions and separated representations for the numerical solution of high dimensional stochastic problems. *Archives of Computational Methods in Engineering*, 17(4):403–434, 2010.

[71] D. S. Oliver, A. C. Reynolds, Z. Bi, and Y. Abacioglu. Integration of production data into reservoir models. *Petroleum Geoscience*, 7(S):S65–S73, 2001.

[72] M. Papadrakakis and V. Papadopoulos. Robust and efficient methods for stochastic finite element analysis using Monte Carlo simulation. *Computer Methods in Applied Mechanics and Engineering*, 134(3):325–340, 1996.

[73] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, New York, second edition, 2004.

[74] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics. *Archives of Computational Methods in Engineering*, 15(3):229–275, 2008.

[75] C. Schwab and R. A. Todor. Karhunen–Loève approximation of random fields by generalized fast multipole methods. *Journal of Computational Physics*, 217(1):100–122, 2006.

[76] A. Speight. A multilevel approach to control variates. *Journal of Computational Finance*, 12(4):3–27, 2009.

[77] P. B. Stark and L. Tenorio. A primer of frequentist and Bayesian inference in inverse problems. *Large Scale Inverse Problems and Quantification of Uncer-*

tainty, John Wiley and Sons, 2010.

[78] A. M. Stuart. Inverse problems: a Bayesian perspective. *Acta Numerica*, 19:451–559, 2010.

[79] M. Sunnåker, A. G. Busetto, E. Numminen, J. Corander, M. Foll, and C. Dessimoz. Approximate Bayesian Computation. *PLoS Comput Biol*, 9(1):e1002803, 2013.

[80] M. Tatang and G. McRae. Direct treatment of uncertainty in models of reaction and transport. *Tech. Rep., MIT Technical Report*, 1994.

[81] E. Wong. *Stochastic processes in information and dynamical systems*. McGraw-Hill, 1971.

[82] D. Xiu. *Numerical methods for stochastic computations: a spectral method approach*. Princeton University Press, 2010.

[83] D. Xiu and G. E. Karniadakis. The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002.

[84] D. Xiu and G. E. Karniadakis. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, 187(1):137–167, 2003.

[85] D. Zhang. *Stochastic methods for flow in porous media : coping with uncertainties*. Academic press, San Diego (Calif.), San Francisco, New York, 2002.