AN EFFICIENT COMPUTATIONAL FRAMEWORK FOR UNCERTAINTY QUANTIFICATION IN MULTISCALE SYSTEMS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Xiang Ma

January 2011

AN EFFICIENT COMPUTATIONAL FRAMEWORK FOR UNCERTAINTY QUANTIFICATION IN MULTISCALE SYSTEMS

Xiang Ma, Ph.D.

Cornell University 2011

To accurately predict the performance of physical systems, it becomes essential for one to include the effects of input uncertainties into the model system and understand how they propagate and alter the final solution. The presence of uncertainties can be modeled in the system through reformulation of the governing equations as stochastic partial differential equations (SPDEs). The spectral stochastic finite element method (SSFEM) and stochastic collocation methods are the most popular simulation methods for SPDEs. However, both methods utilize global polynomials in the stochastic space. Thus when there are steep gradients or finite discontinuities in the stochastic space, these methods converge slowly or even fail to converge.

In order to resolve the above mentioned issues, an adaptive sparse grid collocation (ASGC) strategy is developed using piecewise multi-linear hierarchical basis functions. Hierarchical surplus is used as an error indicator to automatically detect the discontinuity region in the stochastic space and adaptively refine the collocation points in this region. However, this method is limited to a moderate number of random variables. To address the solution of high-dimensional stochastic problems, a computational methodology is further introduced that utilizes the High Dimensional Model Representation (HDMR) technique in the stochastic space to represent the model output as a finite hierarchical correlated function expansion in terms of the stochastic inputs starting from lower-order

to higher–order component functions. An adaptive version of HDMR is also developed to automatically detect the important dimensions and construct higher-order terms using only the important dimensions. The ASGC is integrated with HDMR to solve the resulting sub-problems.

Uncertainty quantification for fluid transport in porous media in the presence of both stochastic permeability and multiple scales is addressed using the developed HDMR framework. In order to capture the small scale heterogeneity, a new mixed multiscale finite element method is developed within the framework of the heterogeneous multiscale method in the spatial domain. Several numerical examples are considered to examine the accuracy of the multiscale and stochastic frameworks developed.

A summary of suggestions for future research in the area of stochastic multiscale modeling are given at the end of the thesis.

BIOGRAPHICAL SKETCH

The author was born in the city of Leshan, Sichuan province, China in March, 1982. After completing his high school education from Leshan NO.1 Middle School, the author was admitted into the Bachelor's program in Automation at University of Science and Technology of China (USTC) in 2000, from where he received his Bachelor's degree in June, 2004. Then the author entered the doctoral program at the Sibley School of Mechanical and Aerospace Engineering, Cornell University and was awarded a special Master's degree in May, 2008.

This thesis is dedicated to my parents Ma, Maorong and Du, Fuzhen for their constant support and encouragement towards academic pursuits during my school years.

ACKNOWLEDGEMENTS

First, I would like to express my sincere gratitude and appreciation to my research advisor, Professor Nicholas Zabaras, for his insightful and interesting suggestions, for innumerable critical comments not only on the work but also on the writing and presentation style over the last four years. Not only he introduced me to a splendor of fascinating technique subjects spanning from the finite element method to Bayesian inference to uncertainty quantification, he discretely taught me many other related non-technical subjects. I believe what I learnt from him will have a direct impact on my future professional career.

I would also like to thank Professors Subrata Mukherjee, Phaedon-Stelios Koutsourelakis and Gennady Samorodnitsky for serving on my special committee and for their encouragement and suggestions during the course of this work.

This research was supported by the Computational Mathematics program of AFOSR (grant F49620-00-1-0373), the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research, the Computational Mathematics program of the National Science Foundation (NSF) (award DMS-0809062) and an OSD/AFOSR MURI09 award on uncertainty quantification. The computing for this research was supported by the NSF through TeraGrid resources provided by NCSA under grant number TG-DMS090007. I would like to thank the Sibley School of Mechanical and Aerospace Engineering for having supported me through a teaching assistantship for part of my study at Cornell. The parallel computing codes were developed based on an open source scientific computation package `PETSc`. The academic license that allowed for these developments is appreciated. Finally, I would like to thank fellow MPDC members and other friends for their support during my life at Cornell.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Over the past few decades there has been considerable interest among the scientific community in studying physical processes with stochastic inputs. These stochastic input conditions arise from uncertainties in boundary and initial conditions as well as from inherent random material heterogeneities. To accurately predict the performance of physical systems, it becomes essential for one to include the effects of input uncertainties into the model system and understand how they propagate and alter the final solution. The presence of uncertainties can be modeled in the system through reformulation of the governing equations as stochastic ordinary/partial differential equations (SODEs/SPDEs). The goal of my thesis is to develop an efficient computational framework that accounts for uncertainties and the inherently random nature of various complex systems which will give the researchers a new guideline on the analysis of experimental results.

In the past decade, there has been tremendous progress in posing and solving SPDEs with the methods used usually classified into three major groups. The first group refers to sampling methods. The most traditional one is the Monte Carlo (MC) method. Its convergence rate does not depend on the number of independent input random variables. Furthermore, MC methods are very easy to implement given a working deterministic code. However, the statistical approach becomes quickly intractable for complex problems in multiple random dimensions. This is because the number of realizations required to acquire good statistics is usually quite large. Furthermore, the number of realizations changes with the variance of the input parameters and the truncation errors

are hard to estimate. This has in part been alleviated by improved sampling techniques like Latin hypercube sampling [1]. The second group of methods consists of moment/perturbation methods, e.g. KL-based moment-equation approach [2, 3, 4]. These methods can deal with large number of inputs. However, they are limited to small fluctuations and do not provide high-order statistics of the solution.

Significant emphasis is given recently on the non-perturbative methods in the third group. The first approach in this group for quantifying uncertainty is the spectral stochastic finite element method (SSFEM) [5]. In this method, we project the dependent variables of the model onto a stochastic space spanned by a set of complete orthogonal polynomials and then a Galerkin projection scheme is used to transform the original stochastic problem into a system of coupled deterministic equations. These polynomials are functions of a set of random variables $\boldsymbol{\xi}(\theta)$ where θ is a realization of the random event space. In the original work of Wiener [6], Gaussian random variables were used with Hermite polynomials. Some of the early applications of SSFEM are presented in [5, 7, 8, 9]. This scheme has been extended to include other random distributions leading to generalized polynomial chaos expansions (gPC) [10]. The gPC was successfully applied to model uncertainty propagation in various applications [11, 12, 13]. Error bounds and convergence studies [14] have shown that these methods exhibit fast convergence rates with increasing orders of expansions. These convergence studies assume that the solution is sufficiently smooth in the random space. Also, the computed absolute error may become unacceptably large during long-term integration. In addition, when the solution exhibits a discontinuous dependence on the input random parameters, the gPC may converge slowly or even fail to converge. This is due to the global polyno-

mial expansion used in the gPC which cannot resolve the local discontinuity in the random space, the well-known Gibbs phenomenon which occurs in spectral decompositions of discontinuous functions.

Thus, more efficient and robust schemes are needed to address the presence of discontinuities in the solution in the random space. In [14, 15, 16], finite element basis functions were used in the random space to approximate locally the stochastic dependence of the solution. In [17], the authors have successfully applied this method to capture unstable equilibrium in natural convection. The wavelet basis expansion method was also utilized to address this problem [18, 19]. The multi-element generalized polynomial chaos method (ME-gPC) was introduced to address discontinuities in the random space while preserving the convergence rate of the gPC method [20, 21, 22]. The main idea of the ME-gPC method is to decompose the space of random inputs into disjoint random elements, then employ a gPC expansion in each element. All of the above methods employ a Galerkin projection in the random space to transform the corresponding stochastic equations to a set of deterministic algebraic equations. The coupled nature of the resulting equations for the unknown coefficients in the spectral expansion makes the solution of the stochastic problem extremely complex as the number of stochastic dimensions and/or the number of expansion terms increase, the so called *curse of dimensionality*. In fact, computational complexity of the problem increases combinatorially with the number of stochastic dimensions and the number of expansion terms. In addition, it is required to develop a stochastic simulator, which is a non-trivial task especially if the underlying ODEs/PDEs have complicated nonlinear terms.

There have been recent efforts to couple the fast convergence of the Galerkin

methods with the decoupled nature of MC sampling, the so called stochastic collocation method. This framework represents the stochastic solution as a polynomial approximation. This interpolant is constructed via independent function calls to the deterministic problem at different interpolation points. This strategy has emerged as a very attractive alternative to the spectral stochastic paradigm. However, the construction of the set of interpolation points is nontrivial, especially in multi-dimensional random spaces. In [23], a methodology was proposed wherein the Galerkin approximation is used to model the physical space and a collocation scheme is used to sample the random space. A tensor product rule was used to interpolate the variables in stochastic space using products of one-dimensional (1D) interpolation functions based on Gauss quadrature points. Though this scheme leads to the solution of uncoupled deterministic problems as in the MC method, the number of realizations required to build the interpolation scheme increases as power of the number of random dimensions. On the other hand, the sparse grid resulting from the Smolyak algorithm depends weakly on dimensionality [24]. Sparse grids has been applied in many fields, such as high-dimensional integration [25], interpolation [26, 27, 28] and solution of PDEs [29]. For an in depth review, the reader may refer to [30]. In [31, 32, 33], the authors used the Smolyak algorithm to build sparse grid interpolants in high-dimensional stochastic spaces based on Lagrange interpolation polynomials. Using this method, interpolation schemes can be constructed with orders of magnitude reduction in the number of sampled points to give the same level of approximation (up to a logarithmic factor) as interpolation on a uniform grid. Hereafter, this method is referred as conventional sparse grid collocation (CSGC) method.

Error estimates for Smolyak algorithm based stochastic collocation methods

have been given in [32, 33], where assuming smoothness of the solution in random space they were shown to achieve fast convergence, similar to stochastic Galerkin methods. However, it is noted that some stochastic sparse grid collocation methods, e.g. [31, 33], utilize the Lagrange polynomial interpolant, which is a global polynomial basis in the random space. Therefore, as is the case with gPC that uses orthogonal global polynomials, these methods fail to capture local behavior in the random space. To this end, we concentrate on stochastic collocation strategies which utilize basis functions with local support, the same idea as in [14, 15, 16, 18, 19, 20], in order to resolve successfully discontinuities in the random space. In addition, we also seek for an adaptive collocation strategy which can refine the sparse grid only locally around the discontinuity region. It is noted that, for the current existing polynomial interpolation methods, e.g. [31, 33], the set of interpolation points are either Clenshaw-Curtis or Gaussian quadrature points, which are pre-determined. So this leads to grids with no substantial room for adaptivity.

Therefore, an adaptive framework utilizing local interpolant/basis functions offers greater promise in efficiently and accurately representing high-dimensional non-smooth stochastic functions. Towards this idea, the authors in [20] proposed an adaptive version of the ME-gPC, where decay rate of local variance was used as an error indicator to adaptively split the random element into two parts along each dimension similar to the *h*-adaptive approach in the deterministic finite element method. In order to utilize the decoupled nature of the collocation algorithm, they later extended this method to the multi-element probabilistic collocation method (ME-PCM), where tensor product or sparse grid collocation is used in each random element [34]. Then the collocation solution is projected back onto the PC basis such that one can employ the same

adaptive criterion as in ME-gPC. These are still dimension-dependent methods, where the number of random elements increases fast with the number of random dimensions. The same problem also exists for the Stochastic Galerkin [16] and Wiener-Haar expansion [19] methods. Thus, there is also a need for an adaptive framework that scales linearly ($O(N)$) with increasing dimensionality instead of the ($O(2^N)$) scaling of current adaptive stochastic methods, where N is the stochastic dimension. In Chapter 3, we utilize a piecewise multi-linear hierarchical basis sparse grid interpolation approach towards adaptivity that addresses the issues of locality and curse-of-dimensionality. This borrows ideas directly from wavelet-based representation of functions [29, 35, 36, 37, 38], where the coefficients of the representation are used as error indicators. However, in [29, 35, 36, 37, 38], the multi-dimensional interpolation grid is constructed through tensor product of one-dimensional wavelet expansions and therefore it is not suitable for high dimensions. The method introduced in this work is different from the adaptive wavelet method since it employs a different adaptation strategy based on the Smolyak algorithm for constructing the interpolation grid. The basic idea here is to use a piecewise linear-hat function as a hierarchical basis function by dilation and translation on equidistant interpolation nodes. Then the stochastic function can be represented by a linear combination of these basis functions. The corresponding coefficients are just the hierarchical increments between two successive interpolation levels (hierarchical surpluses) [27, 30]. The magnitude of the hierarchical surplus reflects the local regularity of the function. For a smooth function, this value decreases to zero quickly with increasing interpolation level. On the other hand, for a non-smooth function, a singularity is indicated by the magnitude of the hierarchical surplus. The larger this magnitude is, the stronger the singularity. Thus, the hierarchical surplus

serves as a natural error indicator for the sparse grid interpolation. When this value is larger than a predefined threshold, we simply add the $2N$ neighboring points to the current point. A key motivation towards using this framework is its linear scaling with dimensionality, in contrast to the N-dimensional tree (2^N) scaling of the h-type adaptive framework (e.g. the framework in [20]). In addition, such a framework guarantees that a user-defined error threshold is met. We will also show that it is rather easier with this approach to extract realizations, higher-order statistics, and the probability density function (PDF) of the solution.

It is noted here that, in previous works, there exists the so called dimension-adaptive (anisotropic) sparse grid methods employing the concept of generalized sparse grids, which was originally developed in [39] and further extended to interpolation in [28]. In recent papers [40, 41], the authors have applied this method to various stochastic problems. In this framework, the structure of the solution was detected on-the-fly to sample the space in a non-isotropic way. The most sensitive dimension is detected and adaptively sampled. Then *all* of the interpolation points from the next level are added along this dimension. Error bounds and convergence issues for the anisotropic sparse grid collocation technique are discussed in [41]. However, this framework requires the underlying discontinuity to be aligned along the lines of the underlying sparse grid, which is not the case in most problems. The method introduced in Chapter 3 is different from the above adaptive strategy. We only add locally around the current point the $2N$ neighboring points from the next interpolation level instead of all of the interpolation points along only one dimension. In this way, besides the detection of important dimensions, additional singularities and local variations in a stochastic function can be found and resolved [29, 30, 42]. It is also noted

that the work in [41] uses Lagrange polynomial interpolation and thus cannot resolve discontinuities.

In Chapter 3, it is shown that the ASGC can successfully resolve stochastic discontinuity problems and solve stochastic elliptic problems up to 100 dimensions when the weights of each dimension are highly anisotropic and thus when the ASGC places more points only along the first few important dimensions. However, it is also shown in the chapter that when the importance of each dimension weighs equally, ASGC cannot solve the problem accurately even with a moderate (21) stochastic dimensionality. In this case, the effect of ASGC is nearly the same as of the CSGC and thus the convergence rate deteriorates. As is well known, in realistic random heterogeneous media often we deal with a very small correlation length and this results in a rather high-dimensional stochastic space with nearly the same weights along each dimension. In this case, all the previously mentioned stochastic methods are obviously not applicable.

These modeling issues for high-dimensional stochastic problems motivate the development of the so called "High Dimensional Model Representation" (HDMR) technique in Chapter 4. Hereafter, this method is referred as conventional HDMR. It is a general set of quantitative assessment and analysis tools for capturing the high-dimensional relationships between sets of input and output model variables. It was originally developed as a methodology to create an efficient fully equivalent operational model of the original chemical systems [43, 44, 45, 46, 47, 48]. It expresses the model output as an additive hierarchical superposition of correlated functions with increasing numbers of input variables, i.e. $1, 2, \ldots$ up to the total number of input variables. A systematic mapping procedure between the inputs and outputs is prescribed to

reveal the hierarchy of correlations amongst the input variables. At each new level of HDMR, higher-order correlated effects of the input variables are introduced. If these higher-order correlated effects between the input variables have negligible effect upon the output, the HDMR approximation is accurate enough by using only lower-order (usually up to the third-order) component functions. This is the case for most realistic physical systems and is the ansatz that the HDMR is based upon. Depending on the way that one determines the hierarchical component functions in HDMR, there are particularly two types of HDMR: ANOVA-HDMR and CUT-HDMR [43]. ANOVA-HDMR is the same as the analysis of variance (ANOVA) decomposition used in statistics and is useful for measuring the contributions of variance of each component function to the overall variance and therefore for computing the global sensitivity [49]. It involves high-dimensional integration and thus is computationally expensive. On the other hand, the CUT-HDMR expansion is a finite exact representation of the model output in the hyperplane passing through a reference point in the input variable space, which is different from gPC that has infinite terms. It only involves function evaluations at the sample points and is more computationally efficient than ANOVA-HDMR. Therefore, it is our focus in Chapter 4.

The model output can be considered as a function taking value over the parameter space supported by the input variables. Therefore, it is also a multivariate function approximation/interpolation method as well as a means to analyze the relevant statistics of the random output, which is the same idea as stochastic collocation method where the solution to the SPDEs is regarded as a stochastic function in the stochastic input space. It is thus straightforward to apply CUT-HDMR in the random space to construct the stochastic input-output mapping. The most important aspect of this method is the capability to find a way to nu-

merically represent each component function of HDMR. In [50], CUT-HDMR is applied to a transport model to represent the stochastic response and MC analysis is used on the obtained approximation to obtain statistics. However, each CUT-HDMR component function was numerically represented as a low-dimensional look-up table over its variables. To obtain an approximate value, one needs to search and interpolate in the table. In [51, 52], CUT-HDMR is derived from a Taylor expansion and is used to find moments of the solution using the so called moment-based quadrature rule in stochastic mechanics. Although the name "dimension reduction method" is used in these papers, it is another form of HDMR. Later, the same author applied this method to fracture and reliability analysis, where component functions are interpolated using tensor product Lagrange polynomials and again MC analysis is used on the obtained approximation to obtain statistics [53, 54]. The author in [55] also applied CUT-HDMR to reliability analysis however the interpolant is constructed through moving least squares (MLS) approximation which is only limited up to three dimensions. Most of the above applications approximate each component function on a tensor-product uniform sampling space thus they are very expensive and not very accurate implementations. In addition, all the previous applications are still limited to relatively low-number of stochastic dimensions (< 12).

In [56], the authors have applied CUT-HDMR, under the name "Anchored-ANOVA", to find the mean and variance of the solution using multi-element probabilistic collocation method and integrating the resultant HDMR expansion term by term. In [57], the authors have applied the same method for the computation of high-dimensional integrals using quadrature schemes with applications to finance. They also developed a dimension-adaptive version of HDMR

to find the important component functions and related it with the anisotropic sparse grid method. Motivated by their work, we develop a general framework to combine the strength from both HDMR and ASGC and apply it to uncertainty quantification. We first redefine the way to compute the error indicator in the formulation of ASGC. The use of the magnitude of the hierarchical surplus as the error indicator is too sharp and may result in non-terminating algorithms. The new error indicator incorporates the information from both the basis function and the surplus. This guarantees that the refinement will stop at a sufficient interpolation level. Then HDMR is used to decompose the high-dimensional stochastic problem into several lower-dimensional sub-problems. Each low-dimensional sub-problem is solved by ASGC in a *locally-adaptive* way within only related dimensions. In this way, an efficient low-dimensional stochastic reduced-order model is constructed and any model output in the stochastic space can be interpolated. By using ASGC, the interpolation of component functions is done efficiently by summing the corresponding hierarchical surplus and basis function as compared with that of using a numerical table. Mean and variance can also be obtained analytically through integrating the basis functions without any MC analysis. In addition, the ASGC provides a linear combination of tensor products chosen in such a way that the interpolation error is nearly the same as for full-tensor product (numerical table) in higher dimensions. In practice, HDMR is often truncated into a lower-order representation. However, for a very large stochastic dimension (> 100), even second-order expansion will have too many component functions. For example, 125251 component functions are needed for a second-order expansion of a 500-dimensional problem. Therefore, we need to find a way to construct only the important component functions. Motivated by the work in [57], in Chapter 4, we also develop a dimension-

adaptive version of HDMR to detect the important component functions. This is defined in two steps: First, the first-order HDMR is constructed and a weight associated with each term is defined to identify the most important dimensions. Then, higher-order components functions are constructed which consist only of these important dimensions. This method to our knowledge is the first approach which can solve high-dimensional stochastic problems by reducing the dimensions from truncation of HDMR and resolve low-regularity by local adaptivity through ASGC. It is noted here that the adaptivity in our paper is different from that used in [57]. The definition of the error indicator is not the same. The authors in [57] did not identify the important dimensions and thus there is a significant computational overhead in finding the important component functions for high-dimensional problems. It is also noted that in [52, 56], the CUT-HDMR is written in an explicit form which is not suitable for adaptive construction. On the other hand, in our work, a recursive form of an HDMR component function is introduced which is the basis for our adaptive implementation.

In the final part of the thesis, we utilize the computational strategies developed to analyze flow through random heterogeneous media with the multiscale permeability variation. We link stochastic analysis and multiscale method to investigate this problem. A stochastic mixed finite element heterogeneous multiscale method is formulated to incorporated the effects of the multiscale stochastic permeability. To achieve this, a new multiscale mixed finite element is developed which is based on the framework of heterogeneous multiscale method [58, 59].

The organization of this thesis is as follows: In Chapter 2, the basic mathematical theory to represent uncertainty and a model problem definition is intro-

duced. In Chapter 3, the adaptive sparse grid collocation method is developed. In Chapter 4, stochastic high dimensional model representation technique is combined with ASGC to address the stochastic problem of high dimensions. In Chapter 5, the development of the stochastic mixed finite element heterogeneous multiscale method is detailed. Finally, in Chapter 6, conclusions of this thesis work and suggestions for future research are summarized.

CHAPTER 2

MATHEMATICAL PRELIMINARIES AND PROBLEM DEFINITION

In this chapter, the mathematical framework of stochastic PDEs is formulated. In particular, this chapter provides overview of basic ideas of uncertainty quantification used extensively in the later parts of this thesis viz. the Karhunen-Loève (K-L) expansion, the Generalized polynomial chaos expansion technique, the stochastic collocation method and the multi-element based methods.

2.1 Mathematical preliminaries

Let us define a complete probability space $(\Omega, \mathcal{F}, \mathcal{P})$ with sample space Ω which corresponds to the outcomes of some experiments, $\mathcal{F}$ the σ-algebra of subsets in Ω (these subsets are called events) and $\mathcal{P} : \mathcal{F} \rightarrow [0, 1]$ the probability measure [60]. In this framework, a single real-valued random variable Y is defined as a function that maps the probability space Ω to $\mathbb{R}$, i.e.,

$$Y : \Omega \rightarrow \mathbb{R}, \tag{2.1}$$

which assigns to each element ω of Ω a real value $Y(\omega)$. We define $y = Y(\omega), \omega \in \Omega$, a *realization* of Y. In this thesis, we will restrict ourselves to continuous random variables. For a single-valued random variable Y, the set of values of Y for all $\omega \in \Omega$ is called the image $Y(\Omega)$ of Ω, i.e.

$$\Gamma_Y \quad = \quad \{Y(\omega) : \omega \in \Omega\} \subseteq \mathbb{R}. \tag{2.2}$$

That is, Γ_Y is actually the range (of all values) of Y on the real line or subset of the real line and therefore it is sometimes also called the state space of Y. Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be a real-valued function. The composition $U = f \circ Y$ is a function

from Ω into $\mathbb{R}$, defined by

$$U(\omega) \quad = \quad f\left(Y(\omega)\right) \quad \text{for all} \quad \omega \in \Omega, \tag{2.3}$$

with the state space Γ_U. For each $u = U(\omega) \in \Gamma_U$, we have

$$u = f(y) \quad \text{for all} \quad y \in \Gamma_Y. \tag{2.4}$$

Thus, the function $U = f \circ Y$ also defines a random variable thus called *stochastic function* since it is also a function of random variables..

The above definition can be generalized to vectors of random variables. Now, let us assume that $\{Y_i\}_{i=1}^N$ are components of $\mathbf{Y} : \Omega \to \mathbb{R}^N$ and their images $\Gamma_i \equiv Y^i(\Omega)$ are bounded intervals in $\mathbb{R}$ for $i = 1, \ldots, N$. Then the stochastic state space of $\mathbf{Y}$ is defined as

$$\Gamma \quad \equiv \quad \prod_{i=1}^{N} \Gamma^i \subset \mathbb{R}^N, \tag{2.5}$$

with the joint PDF as $p(\mathbf{y})$. For example, if Y_i is independent uniform random variable in $[-1, 1]$, then $\Gamma = [-1, 1]^N$. In the work of this thesis, it is assumed that the space Γ is bounded.

2.2 A general problem definition

In this section, we follow the notation in [61]. Define a complete probability space $(\Omega, \mathcal{F}, \mathcal{P})$ with sample space Ω which corresponds to the outcomes of some experiments, $\mathcal{F} \subset 2^\Omega$ is the σ-algebra of subsets in Ω (these subsets are called events) and $\mathcal{P} : \mathcal{F} \to [0, 1]$ is the probability measure. Also, define D as a d-dimensional bounded domain $D \subset \mathbb{R}^d$ $(d = 1, 2, 3)$ with boundary ∂D. We are

interested to find a stochastic function $u : \Omega \times D \to \mathbb{R}$ such that for $\mathcal{P}$-almost everywhere (a.e.) $\omega \in \Omega$, the following equation holds:

$$\mathcal{L}(\boldsymbol{x}, \omega; u) = f(\boldsymbol{x}, \omega), \quad \forall \boldsymbol{x} \in D, \tag{2.6}$$

and

$$\mathcal{B}(\boldsymbol{x}; u) = g(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \partial D, \tag{2.7}$$

where $\boldsymbol{x} = (x_1, \ldots, x_d)$ are the coordinates in $\mathbb{R}^d$, $\mathcal{L}$ is (linear/nonlinear) differential operator, and $\mathcal{B}$ is a boundary operator. In the most general case, the operators $\mathcal{L}$ and $\mathcal{B}$ as well as the driving terms f and g, can be assumed random. We assume that the boundary has sufficient regularity and that f and g are properly defined such that the problem in Eqs. (2.6)-(2.7) is well-posed $\mathcal{P}$ -a.e. $\omega \in \Omega$.

2.2.1 The finite-dimensional noise assumption and the Karhunen-Loève expansion

Any second-order stochastic process can be represented as a random variable at each spatial and temporal location. Therefore, we require an infinite number of random variables to completely characterize a stochastic process. This poses a numerical challenge in modeling uncertainty in physical quantities that have spatio-temporal variations, hence necessitating the need for a reduced-order representation (i.e. reducing the infinite-dimensional probability space to a finite-dimensional one). Such a procedure, commonly known as a 'finite-dimensional noise assumption' [23, 31], can be achieved through any truncated spectral expansion of the stochastic process in the probability space. One such choice is the Karhunen-Loève (K-L) expansion [5].

For example, let the force term $f(\boldsymbol{x}, \omega)$ be a second-order stochastic process, and its correlation function be $R(\boldsymbol{x}_1, \boldsymbol{x}_2)$, where $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are spatial coordinates. By definition, the correlation function is real, symmetric, and positive definite. All its eigenfunctions are mutually orthonormal and form a complete set spanning the function space to which $f(\boldsymbol{x}, \omega)$ belongs. Then the truncated K-L expansion takes the following form:

$$f(\boldsymbol{x}, \omega) = \mathbb{E}[f](\boldsymbol{x}) + \sum_{i=1}^{N} \sqrt{\lambda_i}\phi_i(\boldsymbol{x})Y_i(\omega), \tag{2.8}$$

where $\{Y_i(\omega)\}_{i=1}^{N}$ are uncorrelated random variables. If the process is a Gaussian process, then they are standard identically independent $N(0,1)$ Gaussian random variables. Also, $\phi_i(\boldsymbol{x})$ and λ_i are the eigenfunctions and eigenvalues of the correlation function, respectively. They are the solutions of the following eigenvalue problem:

$$\int_D R(\boldsymbol{x}_1, \boldsymbol{x}_2)\phi_i(\boldsymbol{x}_2)d\boldsymbol{x}_2 \quad = \quad \lambda_i\phi_i(\boldsymbol{x}_1). \tag{2.9}$$

The number of terms needed to approximate a stochastic process depends on the decay rate of the eigenvalues. Generally, a higher correlation length would lead to a rapid decay of the eigenvalues.

Following a decomposition such as the K-L expansion, the random inputs can be characterized by a set of N random variables, e.g.

$$\begin{aligned} \mathcal{L}(\boldsymbol{x}, \omega; u) &= \mathcal{L}(\boldsymbol{x}, Y_1(\omega), \cdots, Y_N(\omega); u), \\ f(\boldsymbol{x}, \omega) &= f(\boldsymbol{x}, Y_1(\omega), \cdots, Y_N(\omega)). \end{aligned} \tag{2.10}$$

Hence, by using the Doob-Dynkin lemma [62], the solution of Eqs. (2.6) and (2.7) can be described by the same set of random variables $\{Y_i(\omega)\}_{i=1}^{N}$, i.e.

$$u(\boldsymbol{x}, \omega) = u(\boldsymbol{x}, Y_1(\omega), \cdots, Y_N(\omega)). \tag{2.11}$$

Thus, the use of the spectral expansion guarantees that the finite-dimensional noise assumption is satisfied and effectively reduces the infinite probability space to a N-dimensional space.

When using the K-L expansion, we here assume that we obtain a set of mutually independent random variables. The issue of non-independent random variables can be resolved by introducing an auxiliary density function [23]. In this thesis, we assume that $\{Y_i(\omega)\}_{i=1}^N$ are independent random variables with probability density function ρ_j. Let Γ_i be the image of Y_i. Then

$$\rho(\mathbf{Y}) = \prod_{i=1}^{N} \rho_i(Y_i), \quad \forall\ \mathbf{Y} \in \Gamma, \tag{2.12}$$

is the joint probability density of $\mathbf{Y} = (Y_1, \cdots, Y_N)$ with support

$$\Gamma \equiv \prod_{i=1}^{N} \Gamma_i \in \mathbb{R}^N. \tag{2.13}$$

Then the problem in Eqs. (2.6) and (2.7) can be restated as: Find the stochastic function $u : \Gamma \times D \to \mathbb{R}$ such that

$$\mathcal{L}(\boldsymbol{x}, \mathbf{Y}; u) = f(\boldsymbol{x}, \mathbf{Y}), \ \ (\boldsymbol{x}, \mathbf{Y}) \in D \times \Gamma, \tag{2.14}$$

subject to the corresponding boundary conditions

$$\mathcal{B}(\boldsymbol{x}, \mathbf{Y}; u) = g(\boldsymbol{x}, \mathbf{Y}), \ \ (\boldsymbol{x}, \mathbf{Y}) \in \partial D \times \Gamma. \tag{2.15}$$

We emphasize here that the dimensionality N of the space Γ is usually determined by the number of the independent random variables Y_i, for example from the K-L expansion in Eq. (2.8). In addition, we also assume without loss of generality that the support of the random variables Y_i is $\Gamma^i = [0, 1]$ for $i = 1, \cdots, N$ and thus the bounded stochastic space is a N-hypercube $\Gamma = [0, 1]^N$, since any bounded stochastic space can always be mapped to the above hypercube.

Therefore, the original infinite-dimensional stochastic problem is restated as a finite-dimensional problem. Then we can apply any stochastic method (gPC expansion or stochastic collocation) in the random space and the resulting equations become a set of deterministic equations in the physical space that can be solved by any standard deterministic discretization technique, e.g. the finite element method. In the next two sections, the basic theories of gPC expansion and current stochastic collocation method are briefly reviewed.

2.3 Generalized polynomial chaos expansion

The theory and properties of the gPC expansion have been well documented in various references [5, 10, 12]. In this approach, the solutions are regarded as random processes. The gPC expansion is used to represent the variables in terms of orthogonal polynomials in the stochastic space Γ. This is written as

$$u(\boldsymbol{x}, \omega) = \sum_{i=0}^{N_p} u_i(\boldsymbol{x}) \Phi_i(Y_i(\omega)), \tag{2.16}$$

where u is the solution of Eqs. (2.14) and (2.15). Here u_i are deterministic coefficients and the summation in the equation above has been truncated to $(N_p + 1)$ terms. The random basis functions $\{\Phi_i\}$ are chosen according to the type of random variable $\{Y_i\}$ that has been used to describe the random input. For example, if Gaussian random variables are chosen then the Askey based orthogonal polynomials $\{\Phi_i\}$ are chosen to be Hermite polynomials, if Y_i are chosen to be uniform random variables, then $\{\Phi_i\}$ must be Legendre polynomials [10]. The total number of expansions terms is determined by the stochastic dimension (N) and the highest-order (p) of the orthogonal polynomials as follows:

$$N_p + 1 = \frac{(N + p)!}{N!p!}. \tag{2.17}$$

The gPC is used to expand all the solution dependent random processes in terms of the orthogonal polynomials. Substituting these expansions into the governing equation, Eq. (2.14), gives:

$$\mathcal{L}(\sum_{i=0}^{N_p} u_i \Phi_i; \boldsymbol{x}, \mathbf{Y}) = f(\boldsymbol{x}, \mathbf{Y}). \tag{2.18}$$

Following this, a Galerkin projection of the above equation onto each polynomial basis Φ_i is conducted:

$$\langle \mathcal{L}(\sum_{i=0}^{N_p} u_i \Phi_i : \boldsymbol{x}, \mathbf{Y}), \ \ \Phi_j \rangle = \langle f(\boldsymbol{x}, \mathbf{Y}), \ \ \Phi_j \rangle, \quad j = 0, \ldots, N_p, \tag{2.19}$$

where $\langle a, b \rangle$ is the inner product of the functions a and b over the ensemble ($\langle a, b \rangle = \int_\Omega ab \, dY$). A similar procedure is also applied on the boundary condition Eq. (2.15). By using the orthogonality of the polynomial basis, we can obtain a set of $(N_p + 1)$ coupled equations for the solution u. By utilizing the polynomial chaos expansion followed by the Galerkin projection, the randomness has been transferred from the dependent variables to the basis polynomials. The resulting governing equations for the expansion coefficients are deterministic.

This methodology has been very successful in solving SPDEs [5, 10, 11, 12, 13, 20, 63]. We will restrict our discussion to the disadvantages of this method to motivate the development of ASGC. Substantial effort has to be put in to convert a validated deterministic code into a gPC-based stochastic one. The global support of the gPC expansion does not allow modeling of SPDEs with steep gradients or discontinuities in the random space. The coupled nature of the resulting algebraic equations in the expansion coefficients makes the implementation of the gPC method non-trivial. Finally, as the number of stochastic dimensions or the order of polynomials used increases, the efficiency of the method is reduced.

2.4 Stochastic collocation method

In this section, we review the fundamentals of stochastic collocation method. For more details, the interest readers may refer to [31, 33, 40]. The basic idea of the stochastic collocation approach is to build an interpolation function for the unknown stochastic solutions using their values at a *pre-determined* set of points in the stochastic space. These points are called the collocation points. The Galerkin projection is then applied to find that interpolation function which minimizes the projected error of the approximated system. Denote by $\mathbf{Y}$ any point in the random space $\Gamma \subset \mathbb{R}^N$, by Π_N, the space of all N-variate polynomials and by Π_N^p, the subspace of polynomials of total degree at most p. The problem of interpolation can be stated as follows: Given a set of nodes $\Theta_N = \{\mathbf{Y}_i\}_{i=1}^M$ in the N-dimensional random space Γ and the smooth function $f : \mathbb{R}^N \to \mathbb{R}$, find the polynomial $\mathcal{I}f$ such that $\mathcal{I}f(\mathbf{Y}_i) = f(\mathbf{Y}_i), \forall i = 1, \ldots, M$.

The polynomial approximation $\mathcal{I}f$ can be expressed using the Lagrange interpolation polynomials as follows:

$$\mathcal{I}f(\mathbf{Y}) = \sum_{i=1}^{M} f(\mathbf{Y}_i) L_i(\mathbf{Y}) \tag{2.20}$$

where $L_i(\mathbf{Y}_j) = \delta_{ij}$. Now, once the interpolating polynomial have been generated using the nodes $\{\Theta_N\}$, the value of the stochastic function at any point $\mathbf{Y} \in \Gamma$ is approximately $\mathcal{I}f(\mathbf{Y})$. The Lagrange interpolated value of u, denoted by $\hat{u}$ is as follows:

$$\hat{u} = \sum_{i=1}^{M} u(\mathbf{Y}_i) L_i(\mathbf{Y}) \tag{2.21}$$

Substituting this into the governing equation, Eq. (2.14), gives

$$\mathcal{L}\left(\boldsymbol{x}, \sum_{i=1}^{M} u(\mathbf{Y}_i) L_i(\mathbf{Y})\right) = f(\boldsymbol{x}, \mathbf{Y}). \tag{2.22}$$

The interpolation form of the solution immediately leads to M decoupled deterministic systems

$$\mathcal{L}(\boldsymbol{x}, u(\mathbf{Y}_i)) = f(\boldsymbol{x}, \mathbf{Y}_i), \quad i = 1, \ldots, M. \tag{2.23}$$

The collocation method collapses the $(N + d)$-dimensional problem to solve M deterministic problems in d dimensions. The kth-moment of the random solution can then be obtained by

$$\langle u^k \rangle = \sum_{i=1}^{M} u^k(\mathbf{Y}_i) \int_{\Gamma} L_i(\mathbf{Y}) \rho(\mathbf{Y}) d(\mathbf{Y}). \tag{2.24}$$

Choice of collocation points include tensor product of zeros of orthogonal polynomials [23] or sparse grid approximations [31, 33, 40]. In the current implementations, the sparse grid points are chosen as Gauss quadrature points according to the random distributions. So, the integration value of the basis function coincides the Gauss quadrature weights. Therefore, the interpolation function is not explicitly formulated. Instead, the statistics of the solution is directly calculated. As is well known, the global polynomial interpolation cannot resolve local discontinuity in the stochastic space. In addition, the quadrature points are pre-determined thus leaving no space of allowing adaptivity. Based on these observations, adaptive sparse grid collocation method which utilizes local linear interpolation is developed in next chapter.

2.5 Multi-element generalized polynomial chaos

The regularity of the solution with respect to the stochastic space Γ affects the convergence rate of gPC expansion, especially of approximations of global support. However, such regularity is usually not known *a priori* in many problems.

This has motivated the development of the multi-element based stochastic domain decomposition method. The basic idea is to divide the stochastic space into several disjoint elements and solve each subproblem using gPC or conventional probabilistic collocation method, which results in the ME-gPC and ME-PCM methods, respectively. In this section, we briefly discuss the Multi-element (ME) based method, since it is one of the current stochastic methods developed for resolving discontinuities in the random space. The description here closely mirrors that in [20, 34].

Let $\mathbf{Y}$ be a random variable defined on the random space Γ and the components of $\mathbf{Y}$ be independent identically distributed random variables. We define a decomposition B of Γ with M disjoint elements as follows:

$$B = \begin{cases} \Gamma_i = [a_1^i, b_1^i) \times [a_2^i, b_2^i) \times \ldots \times [a_N^i, b_N^i), \\ \Gamma = \bigcup_{i=1}^{M} \Gamma_i, \\ \Gamma_{i_1} \bigcap \Gamma_{i_2} = \emptyset \ \text{ if } \ i_1 \neq i_2, \end{cases} \tag{2.25}$$

where $i, i_1, i_2 = 1, 2, \cdots, M$. Based on this decomposition, we define the following indicator random variables:

$$I_{\Gamma_i} = \begin{cases} 1, & \text{if } \ \mathbf{Y} \in \Gamma_i, \\ 0, & \text{otherwise,} \end{cases} \qquad i = 1, 2, \cdots, M. \tag{2.26}$$

Then, a local random vector $\mathbf{Y}_i$ is defined in each element Γ_i subject to a conditional PDF

$$\hat{f}_i(\mathbf{Y}_i | I_{\Gamma_i} = 1) = \frac{f(\mathbf{Y}_i)}{\Pr(I_{\Gamma_i} = 1)}, \tag{2.27}$$

where $J_i = \Pr(I_{\Gamma_i} = 1) > 0$.

Let us assume that the gPC expansion of a random field in element i is

$$\hat{u}_i(\mathbf{Y}_i) = \sum_{j=0}^{N_p} \hat{u}_{i,j} \Phi_{i,j}(\mathbf{Y}_i), \tag{2.28}$$

where p is the highest expansion order of the orthogonal polynomial and the number of expansion term N_p is given by Eq. (2.17). From the orthogonality of gPC, the local variance is given by

$$\sigma_{i,p}^2 = \sum_{j=1}^{N_p} \hat{u}_{i,j}^2 \langle \Phi_{i,j}^2 \rangle. \tag{2.29}$$

The approximate global mean $\bar{u}$ and variance $\bar{\sigma}^2$ can be expressed as

$$\bar{u} = \sum_{i=1}^{M} \hat{u}_{i,0} J_i, \qquad \bar{\sigma}^2 = \sum_{i=1}^{M} \left[\sigma_{i,p}^2 + (\hat{u}_{i,0} - \bar{u})^2 \right] J_i. \tag{2.30}$$

A local decay rate of relative error of the approximation in each element is defined as follows:

$$\eta_i = \frac{\sum_{j=N_{p-1}+1}^{N_p} \hat{u}_{i,j}^2 \langle \Phi_{i,j}^2 \rangle}{\sigma_{i,p}^2}. \tag{2.31}$$

A random element will be split into two equal parts when the following condition is satisfied

$$\eta_i^{\alpha} J_i \geq \theta_1, \quad 0 < \alpha < 1, \tag{2.32}$$

where α is a prescribed constant usually set to $1/2$. For high-dimensional random inputs, we use another threshold parameter θ_2 to choose the most sensitive random dimension in this element. The sensitivity for each random dimension is defined as

$$r_k = \frac{\left(\hat{u}_{k,p} \right)^2 \langle \Phi_{k,p}^2 \rangle}{\sum_{j=N_{p-1}+1}^{N_p} \hat{u}_{i,j}^2 \langle \Phi_{i,j}^2 \rangle}, \qquad k = 1, 2, \cdots, N, \tag{2.33}$$

where the subscript $\cdot_{k,p}$ denotes the mode consisting only of random dimension Y_k with polynomial order p. All random dimensions which satisfy

$$r_k \geq \theta_2 \cdot \max_{j=1,\cdots,N} r_j, \quad 0 < \theta_2 < 1, \quad k = 1, 2, \cdots, N, \tag{2.34}$$

will be split into two equal random elements while all other random dimensions remain unchanged. Therefore, it is seen that although the most sensitive

random dimension is detected, this method still requires a large number of elements when the random dimension is high. At the same time, the number of expansion terms also increases very quickly, see Eq. (2.17). So it is still a dimension dependent method [21]. We implemented this method in order to compare computational cost and accuracy in the examples of Chapter 3. It is noted here, that due to the assumption of uniform distribution, the local conditional PDF Eq. (2.27) is still uniform. Thus, we do not need to reconstruct the orthogonal polynomials in each random element. In general, the orthogonality is not satisfied because the PDF is also discretized simultaneously and one needs to numerically construct the orthogonal polynomials on the fly [21].

CHAPTER 3

SOLVING STOCHASTIC PARTIAL DIFFERENTIAL EQUATIONS: AN ADAPTIVE HIERARCHICAL SPARSE GRID COLLOCATION ALGORITHM

In this chapter, the development of adaptive sparse grid collocation algorithm is detailed. Most content of this chapter is from the work in [61]. The contribution of this work is as follows: (1) We utilize the concepts of hierarchical sparse grid collocation. This provides a new point of view on the sparse grid collocation method leading to the concept of adaptivity; (2) We develop a locally-refined adaptive sparse grid collocation method with $2N$ linear scaling for the refinement, which further reduces the *curse of dimensionality*; (3) By purely based on the interpolation, it is shown that this method not only can calculate easily the mean and the variance, but also can extract the realization of the solution as a function of the random variables in order to examine its local behavior. This is another issue not addressed in earlier works [31, 32, 33, 34, 40, 41].

3.1 Hierarchical stochastic sparse grid collocation method

The basic idea of this method is to have a finite element approximation for the spatial domain and approximate the multi-dimensional stochastic space using interpolation functions on a set of collocation points $\{\mathbf{Y}_i\}_{i=1}^{M} \in \Gamma$. Suppose we can find a finite element approximate solution u of the deterministic solution of the problem in Eqs. (2.14)-(2.15) for each point $\mathbf{Y}_i$, we are then interested in constructing an interpolant of u by using linear combinations of the solutions $u(\cdot, \mathbf{Y}_i)$. The multi-dimensional interpolation can be constructed through either

full-tensor product of 1D interpolation rule or by the so called sparse grid interpolation method based on the Smolyak algorithm [24]. Since in the full tensor product case the number of support points grows very quickly as the number of stochastic dimensions increases, we will mainly focus on the sparse grid method and discuss the proposed adaptivity algorithm.

3.1.1 Smolyak algorithm

The Smolyak algorithm provides a way to construct interpolation functions based on a minimal number of points in multi-dimensional space. Using the Smolyak method, univariate interpolation formulae are extended to the multivariate case by using tensor products in a special way. This provides an interpolation strategy with potentially orders of magnitude reduction in the number of support nodes required. The algorithm provides a linear combination of tensor products chosen in such a way that the interpolation error is nearly the same as for full-tensor product in higher dimensions.

Let us consider a smooth function $f : [0,1]^N \to \mathbb{R}$. In the 1D case ($N = 1$), we consider the following interpolation formula to approximate f:

$$\mathcal{U}^i(f) = \sum_{j=1}^{m_i} f(Y_j^i) \cdot a_j^i, \tag{3.1}$$

with the set of support nodes

$$X^i = \left\{ Y_j^i \mid Y_j^i \in [0,1] \text{ for } j = 1, 2, \ldots, m_i \right\}, \tag{3.2}$$

where $i \in \mathbb{N}$, $a_j^i \equiv a_j(Y_j^i) \in C([0,1])$ are the interpolation nodal basis functions, and m_i is the number of elements of the set X^i. We assume that a sequence of formulae Eq. (3.1) is given with different i. In the multivariate case ($N > 1$), the

tensor product formulae are

$$(\mathcal{U}^{i_1} \otimes \cdots \otimes \mathcal{U}^{i_N})(f) = \sum_{j_1=1}^{m_1} \cdots \sum_{j_N=1}^{m_N} f(Y_{j_1}^{i_1}, \ldots, Y_{j_N}^{i_N}) \cdot (a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_N}^{i_N}), \tag{3.3}$$

which serve as building blocks for the Smolyak algorithm.

The Smolyak algorithm constructs the sparse interpolant $A_{q,N}$ using products of 1D functions. $A_{q,N}$ is given as [26, 27, 28]

$$\mathcal{A}_{q,N}(f) = \sum_{q-N+1 \le |\mathbf{i}| \le q} (-1)^{q-|\mathbf{i}|} \cdot \binom{N-1}{q-|\mathbf{i}|} \cdot (\mathcal{U}^{i_1} \otimes \cdots \otimes \mathcal{U}^{i_N}), \tag{3.4}$$

with $q \ge N$, $\mathcal{A}_{N-1,N} = 0$ and where the multi-index $\mathbf{i} = (i_1, \ldots, i_N) \in \mathbb{N}^N$ and $|\mathbf{i}| = i_1 + \cdots + i_N$. Here i_k, $k = 1, \ldots, N$, is the level of interpolation along the k-th direction. The Smolyak algorithm builds the interpolation function by adding a combination of 1D functions of order i_k with the constraint that the sum total ($|\mathbf{i}| = i_1 + \ldots + i_N$) across all dimensions is between $q-N+1$ and q. The structure of the algorithm becomes clearer when one considers the incremental interpolant, Δ^i given by [26, 27, 28]

$$\mathcal{U}^0 = 0, \quad \Delta^i = \mathcal{U}^i - \mathcal{U}^{i-1}. \tag{3.5}$$

The Smolyak interpolation $\mathcal{A}_{q,N}$ is then given by

$$\begin{aligned} \mathcal{A}_{q,N}(f) &= \sum_{|\mathbf{i}| \le q} (\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_N})(f) \\ &= \mathcal{A}_{q-1,N}(f) + \sum_{|\mathbf{i}| = q} (\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_N})(f). \end{aligned} \tag{3.6}$$

To compute the interpolant $\mathcal{A}_{q,N}(f)$ from scratch, one needs to compute the function at the nodes covered by the sparse grid $\mathcal{H}_{q,N}$:

$$\mathcal{H}_{q,N} = \bigcup_{q-N+1 \le |\mathbf{i}| \le q} (X^{i_1} \times \cdots \times X^{i_N}). \tag{3.7}$$

The construction of the algorithm allows one to utilize all the previous results generated to improve the interpolation (this is immediately obvious from

Eq. (3.6)). By choosing appropriate points for interpolating the 1D function, one can ensure that the sets of points X^i are nested ($X^i \subset X^{i+1}$). To extend the interpolation from level $i-1$ to i, one only has to evaluate the function at the grid points that are unique to X^i, that is, at $X^i_\Delta = X^i \backslash X^{i-1}$. Thus, to go from an order $q-1$ interpolation to an order q interpolation in N dimensions, one only needs to evaluate the function at the differential nodes $\Delta\mathcal{H}_{q,N}$ given by

$$\Delta\mathcal{H}_{q,N} = \bigcup_{|\mathbf{i}|=q} (X^{i_1}_\Delta \otimes \cdots \otimes X^{i_N}_\Delta). \tag{3.8}$$

3.1.2 Choice of collocation points and the nodal basis functions

It is more advantageous to choose the collocation points in a nested fashion to obtain many recurring points with increasing q. One of the choices is the Clenshaw-Curtis grid at the non-equidistant extrema of the Chebyshev polynomials [31, 33, 41]. For any choice of $m_i > 1$, the sets $X^i = \{Y^i_1, \cdots, Y^i_{m_i}\}$ are given by

$$m_i = \begin{cases} 1, & \text{if } i = 1, \\ 2^{i-1}+1, & \text{if } i > 1, \end{cases} \tag{3.9}$$

$$Y^i_j = \begin{cases} (-\cos(\pi(j-1)/(m_i-1))+1)/2, & \text{for } j = 1, \ldots, m_i, \text{ if } m_i > 1, \\ 0.5, & \text{for } j = 1, \text{ if } m_i = 1. \end{cases} \tag{3.10}$$

With this selection, the resulting sets are nested, i.e., $\mathcal{H}_{q,N} \subset \mathcal{H}_{q+1,N}$. The corresponding univariate nodal basis functions are Lagrange characteristic polynomials.

$$a^i_j = \begin{cases} 1, & \text{for } i = 1, \text{ and} \\ \prod\limits_{\substack{k=1 \\ k\neq j}}^{m_i} \dfrac{Y - Y^i_k}{Y^i_j - Y^i_k}, & \text{for } i > 1 \text{ and } j = 1, \ldots, m_i. \end{cases} \tag{3.11}$$

It is noted that by using this grid, the support nodes are pre-determined as in Eq. (3.10). Thus, this grid is not suitable if we want to use adaptivity. Therefore, we propose to use the Newton-Cotes grid using equidistant support nodes. By using equidistant nodes, it is easy to refine the grid locally. However, it is well known that for Lagrange polynomial interpolation on equidistant nodes, the error may not go to zero as the number of nodes increases due to the well-known Runge's phenomenon [28]. To this end, we propose to use the linear hat function as the univariate nodal basis function [25]. The piecewise linear hat function has a local support in contrast to the global support of the polynomial in Eq. (3.11), so it can be used to resolve discontinuities in the stochastic space.

We first consider the 1D interpolation rule Eq. (3.1) with the support nodes defined as

$$m_i = \begin{cases} 1, & \text{if } i = 1, \\ 2^{i-1} + 1, & \text{if } i > 1, \end{cases} \tag{3.12}$$

$$Y_j^i = \begin{cases} \dfrac{j-1}{m_i - 1}, & \text{for } j = 1, \ldots, m_i, \text{ if } m_i > 1, \\ 0.5, & \text{for } j = 1, \text{ if } m_i = 1. \end{cases} \tag{3.13}$$

It is noted that the resulting grid points are also nested and the grid has the same number of points as the Clenshaw-Curtis grid.

In the linear setting, the simplest choice of 1D basis function is the standard linear hat function [27, 29, 30]:

$$a(Y) = \begin{cases} 1 - |Y|, & \text{if } Y \in [-1, 1], \\ 0, & \text{otherwise.} \end{cases} \tag{3.14}$$

This mother of all piecewise linear basis functions can be used to generate an arbitrary a_j^i with local support $[Y_j^i - 2^{1-i}, Y_j^i + 2^{1-i}]$ by dilation and translation, i.e.,

$$a_1^1 = 1 \text{ for i} = 1, \text{and} \tag{3.15}$$

$$a_j^i = \begin{cases} 1 - (m_i - 1) \cdot |Y - Y_j^i|, & \text{if } |Y - Y_j^i| < 1/(m_i - 1), \\ 0, & \text{otherwise,} \end{cases} \tag{3.16}$$

for $i > 1$ and $j = 1, \ldots, m_i$. The N-dimensional multilinear basis functions can be constructed using tensor products as follows:

$$a_{\mathbf{j}}^{\mathbf{i}}(\mathbf{Y}) := a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_N}^{i_N} = \prod_{k=1}^{N} a_{j_k}^{i_k}, \tag{3.17}$$

where the multi-index $\mathbf{j} = (j_1, \ldots, j_N) \in \mathbb{N}^N$ and j_k, $k = 1, \ldots, N$, denotes the location of a given support node in the k-th dimension from Eq. (3.13). Thus, we define the functional space

$$V^{\mathbf{i}} := \operatorname{span}\left\{ a_{\mathbf{j}}^{\mathbf{i}} : \mathbf{i} \in \mathbb{N}^N, \mathbf{j} \in \mathbb{N}^N, j_k = 1, \ldots, m_{i_k}, k = 1, \ldots, N \right\}, \tag{3.18}$$

as the space of piecewise multi-linear functions for a give multi-index $\mathbf{i}$. Then the family of functions $\{a_{\mathbf{j}}^{\mathbf{i}}\}$ is just the standard *nodal basis* of the finite dimensional space $V^{\mathbf{i}}$. If we want to apply the Smolyak algorithm using nodal basis, it is straightforward to use Eq. (3.4). It can be rewritten as

$$\mathcal{A}_{q,N}(f) = \sum_{q-N+1 \le |\mathbf{i}| \le q} (-1)^{q-|\mathbf{i}|} \cdot \binom{N-1}{q-|\mathbf{i}|} \cdot \sum_{\mathbf{j}} f(Y_{j_1}^{i_1}, \ldots, Y_{j_N}^{i_N}) \cdot a_{\mathbf{j}}^{\mathbf{i}}. \tag{3.19}$$

Now we can define the sparse grid interpolation space V_Γ as

$$V_\Gamma := \bigoplus_{q-N+1 \le |\mathbf{i}| \le q} V^{\mathbf{i}}. \tag{3.20}$$

It is noted that the coefficients of the approximation in the stochastic space mainly depend on the function values at the interpolation points. Thus they do not give much information about the regularity of the solution in the random space. Therefore, the interpolation formulae provided above are not appropriate for an adaptive implementation.

3.1.3 From nodal basis to multivariate hierarchical basis

Let us return to the incremental interpolation formula Eq. (3.6). This formula takes advantage of the nested nature of the grid points, $X^{(i)} \subset X^{i+1}$ [27]. Here, we follow closely [27] to provide a clear development of the derivation of the hierarchical basis and the hierarchical surpluses.

We start from the 1D interpolation formula Eq. (3.1) using nodal basis as discussed in the previous section. By the definition of Eq. (3.5), we can write:

$$\Delta^i(f) \;=\; \mathcal{U}^i(f) - \mathcal{U}^{i-1}(f). \tag{3.21}$$

With

$$\mathcal{U}^i(f) = \sum_{Y_j^i \in X^i} a_j^i \cdot f(Y_j^i), \text{ and } \mathcal{U}^{i-1}(\mathrm{f}) = \mathcal{U}^i(\mathcal{U}^{i-1}(\mathrm{f})), \tag{3.22}$$

we obtain [27]

$$\begin{aligned} \Delta^i(f) &= \sum_{Y_j^i \in X^i} a_j^i \cdot f(Y_j^i) - \sum_{Y_j^i \in X^i} a_j^i \cdot \mathcal{U}^{i-1}(f)(Y_j^i) \\ &= \sum_{Y_j^i \in X^i} a_j^i \cdot (f(Y_j^i) - \mathcal{U}^{i-1}(f)(Y_j^i)), \end{aligned} \tag{3.23}$$

and, since $f(Y_j^i) - \mathcal{U}^{i-1}(f)(Y_j^i) = 0, \forall\ Y_j^i \in X^{i-1}$, we obtain

$$\Delta^i(f) \;=\; \sum_{Y_j^i \in X_\Delta^i} a_j^i \cdot (f(Y_j^i) - \mathcal{U}^{i-1}(f)(Y_j^i)), \tag{3.24}$$

recalling that $X_\Delta^i = X^i \setminus X^{i-1}$. Clearly, X_Δ^i has $m_\Delta^i = m_i - m_{i-1}$ points, since $X_{i-1} \subset X_i$. By consecutively numbering the elements in X_Δ^i, and denoting the j-th point of X_Δ^i as Y_j^i, we can re-write the above equation as [27]

$$\Delta^i(f) \;=\; \sum_{j=1}^{m_\Delta^i} a_j^i \cdot \underbrace{(f(Y_j^i) - \mathcal{U}^{i-1}(f)(Y_j^i))}_{w_j^i}. \tag{3.25}$$

Here, we define w_j^i as the 1D hierarchical surplus, which is just the difference between the function values at the current and the previous interpolation levels. We also define the set of functions a_j^i as the hierarchical basis functions. Fig. 3.1 shows the comparison of the nodal and the hierarchical basis functions [27]. Fig. 3.2 shows a comparison of the nodal and hierarchical interpolation in 1D [27].

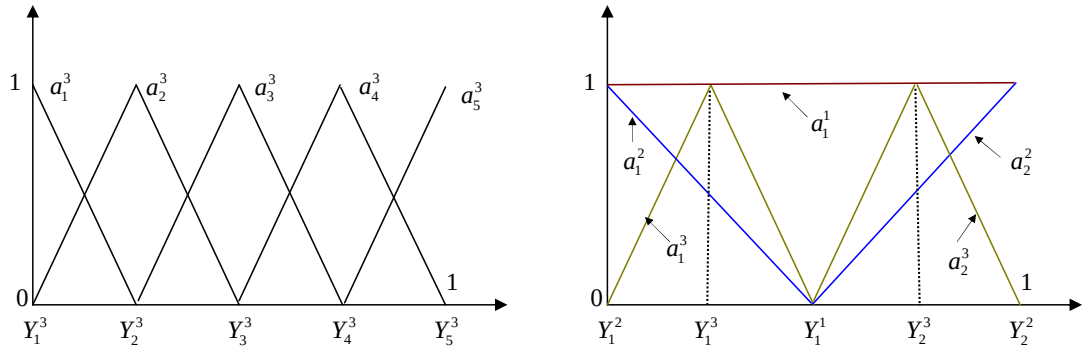


Figure 3.1: Nodal basis functions $a_j^3, Y_j^3 \in X^3$ (left) and hierarchical basis functions a_j^i with the support nodes $Y_j^i \in X_\Delta^i, i = 1, 2, 3$ (right) for the Newton-Cotes grid.

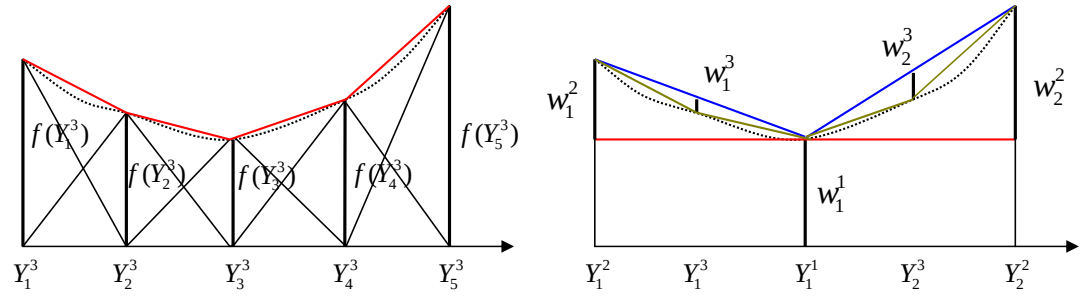


Figure 3.2: Nodal (left) versus hierarchical (right) interpolation in 1D.

For example, in Fig. 3.2, if we work in the nodal basis of interpolation level 3, then the function f is approximated as from Eq. (3.1)

$$f = f(Y_1^3)a_1^3 + f(Y_2^3)a_2^3 + f(Y_3^3)a_3^3 + f(Y_4^3)a_4^3 + f(Y_5^3)a_5^3. \tag{3.26}$$

On the other hand, the hierarchical basis for the same interpolation level from

Eq. (3.25) is given as follows:

$$f = w_1^1 a_1^1 + w_1^2 a_1^2 + w_2^2 a_2^2 + w_1^3 a_1^3 + w_2^3 a_2^3. \tag{3.27}$$

Please note the different numbering used for the hierarchical and nodal basis functions. It is seen that the hierarchical basis utilizes only some of the nodal basis functions form level 1 to 3 instead of all of the nodal basis functions in level 3. It is for this reason that we refer to this representation as the 'hierarchical basis'.

For the multi-dimensional case, we define a hierarchical difference space

$$W^{\mathbf{i}} := V^{\mathbf{i}} \setminus \bigoplus_{k=1}^{N} V^{\mathbf{i}-\mathbf{e}_k}, \tag{3.28}$$

where $\mathbf{e}_k$ denotes the k-th unit vector. To complete this definition, we formally set

$$V^{\mathbf{i}-\mathbf{e}_k} = 0, \quad \text{if } i_k = 0. \tag{3.29}$$

Thus, through a new multi-index set

$$B_{\mathbf{i}} := \left\{ \mathbf{j} \in \mathbb{N}^N : Y_{j_k}^{i_k} \in X_{\Delta}^{i_k} \text{ for } j_k = 1, \ldots, m_{\Delta}^{i_k}, k = 1, \ldots, N \right\}, \tag{3.30}$$

we can obtain another basis of $V^{\mathbf{i}}$, the *hierarchical basis*

$$\left\{ a_{\mathbf{j}}^{\mathbf{i}} : \mathbf{j} \in B_{\mathbf{k}}, \mathbf{k} \leq \mathbf{i} \right\}, \tag{3.31}$$

which also leads to

$$W^{\mathbf{i}} := \operatorname{span} \left\{ a_{\mathbf{j}}^{\mathbf{i}} : \mathbf{j} \in B_{\mathbf{i}} \right\}. \tag{3.32}$$

It is clear that the following decomposition holds [29, 30]

$$V^{\mathbf{i}} := \bigoplus_{k_1=1}^{i_1} \cdots \bigoplus_{k_N=1}^{i_N} W^{\mathbf{k}} := \bigoplus_{\mathbf{k} \leq \mathbf{i}} W^{\mathbf{k}}. \tag{3.33}$$

This equation provides another view on the nodal basis function space $V^{\mathbf{i}}$. Note that in Eqs. (3.31) and (3.33), '$\leq$' refers to the element-wise relation for multi-indexes.

We next obtain the sparse grid interpolation formula for the multivariate case in a hierarchical form. From Eq. (3.6), we can write:

$$\mathcal{A}_{q,N}(f) = \mathcal{A}_{q-1,N}(f) + \Delta\mathcal{A}_{q,N}(f), \tag{3.34}$$

$$\Delta\mathcal{A}_{q,N}(f) = \sum_{|\mathbf{i}|=q} (\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_N}), \tag{3.35}$$

with $\mathcal{A}_{N-1,N} = 0$. This can be further simplified as

$$\mathcal{A}_{q-1,N}(f) = \sum_{|\mathbf{i}|\leq q-1} (\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_N}), \tag{3.36}$$

and

$$\begin{aligned}\Delta\mathcal{A}_{q,N}(f) = \sum_{|\mathbf{i}|=q} \sum_{\mathbf{j}\in B_{\mathbf{i}}} & (a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_N}^{i_N}) \\ & \cdot (f(Y_{j_1}^{i_1}, \ldots, Y_{j_N}^{i_N}) - \mathcal{A}_{q-1,N}(f)(Y_{j_1}^{i_1}, \ldots, Y_{j_N}^{i_N})).\end{aligned} \tag{3.37}$$

Here, we define

$$w_{\mathbf{j}}^{\mathbf{i}} = f(Y_{j_1}^{i_1}, \ldots, Y_{j_N}^{i_N}) - \mathcal{A}_{|\mathbf{i}|-1,N}(f)(Y_{j_1}^{i_1}, \ldots, Y_{j_N}^{i_N}), \tag{3.38}$$

as the hierarchical surplus, which is just the difference between the function value at a point in the current level of interpolation and the corresponding value at the previous interpolation level [27, 29]. Eq. (3.37) actually provides a hierarchical subspace splitting of V_Γ:

$$V_\Gamma := \bigoplus_{|\mathbf{i}|\leq q} W^{\mathbf{i}}. \tag{3.39}$$

Thus, we can work either in the nodal basis functional space or the hierarchical basis space. For smooth functions, the hierarchical surpluses tend to zero as the

interpolation level tends to infinity as shown in Fig. 3.2. On the other hand, for non-smooth functions, steep gradients/finite discontinuities are indicated by the magnitude of the hierarchical surplus. The bigger the magnitude is, the stronger the underlying discontinuity is. Therefore, the hierarchical surplus is a natural candidate for error control and implementation of adaptivity.

3.1.4 Interpolation error

As a matter of notation, the interpolation function used will be denoted $A_{N+k,N}$, where k is called the level of the Smolyak interpolation. This is because we always start the construction from the N-dimensional multi-index $\mathbf{i} = (1, \ldots, 1)$. We consider the interpolation error in the space

$$F_N := \left\{ f : [0,1]^N \to \mathbb{R}, D^{|\mathbf{m}|} f \text{ continues}, m_i \leq 2, \forall i \right\}, \tag{3.40}$$

where $\mathbf{m} \in \mathbb{N}_0^N$ and $D^{|\mathbf{m}|}$ is the usual N-variate partial derivative of order $|\mathbf{m}|$:

$$D^{|\mathbf{m}|} = \frac{\partial^{|\mathbf{m}|}}{\partial Y_1^{m_1} \cdots Y_N^{m_N}}. \tag{3.41}$$

Then the interpolation error in the maximum norm is given by [26, 27, 28]

$$\|f - A_{q,N}(f)\|_\infty = O\left(M^{-2} |\log_2 M|^{3(N-1)}\right), \tag{3.42}$$

where $M = dim(\mathcal{H}(q, N))$ is the number of interpolation points.

3.1.5 From hierarchical interpolation to hierarchical integration

Any function $u \in \Gamma$ can now be approximated by the following reduced form from Eq. (3.37):

$$u(\boldsymbol{x}, \mathbf{Y}) \quad = \quad \sum_{|\mathbf{i}| \leq q} \sum_{\mathbf{j} \in B_{\mathbf{i}}} w_{\mathbf{j}}^{\mathbf{i}}(\boldsymbol{x}) \cdot a_{\mathbf{j}}^{\mathbf{i}}(\mathbf{Y}). \tag{3.43}$$

This expression can be considered as an approximate solution of the problem in Eqs. (2.14) and (2.15). It is just a simple weighted sum of the value of the basis functions for all collocation points in the sparse grid. Therefore, we can easily extract the useful statistics of the solution from it. For example, we can sample independently N times from the uniform distribution $[0, 1]$ to obtain one random vector $\mathbf{Y}$, then we can place this vector into the above expression to obtain one realization of the solution. In this way, it is easy to plot realizations of the solution as well as its PDF. On the other hand, if the Smolyak algorithm Eq. (3.4) is used based on the cubature rule [32], although it is easy to calculate the mean and variance, it is difficult to extract the value of the solution at a particular point in the random space. This is one of the advantages of applying the stochastic collocation method based on the present interpolation rule, which allows us to obtain a visualization of the solution dependence on the random variables. After obtaining the expression in Eq. (3.43), it is also easy to extract the mean and variance analytically, leaving only the interpolation error.

The mean of the random solution can be evaluated as follows:

$$\mathbb{E}[u(\boldsymbol{x})] \quad = \quad \sum_{|\mathbf{i}| \leq q} \sum_{\mathbf{j} \in B_{\mathbf{i}}} w_{\mathbf{j}}^{\mathbf{i}}(\boldsymbol{x}) \cdot \int_{\Gamma} a_{\mathbf{j}}^{\mathbf{i}}(\mathbf{Y}) d\mathbf{Y}, \tag{3.44}$$

where the probability density function $\rho(\mathbf{Y})$ is 1 since the stochastic space is a

unit hypercube $[0,1]^N$. The 1D integral can be evaluated analytically:

$$\int_0^1 a_j^i(Y)dY = \begin{cases} 1, & \text{if } i = 1, \\ \frac{1}{4}, & \text{if } i = 2, \\ 2^{1-i}, & \text{otherwise.} \end{cases} \tag{3.45}$$

This is independent of the location of the interpolation point and only depends on the interpolation level in each stochastic dimension due to the translation and dilation of the basis functions. Since the random variables are assumed independent of each other, the value of the multi-dimensional integral is simply the product of the 1D integrals. Denoting $\int_\Gamma a_{\mathbf{j}}^{\mathbf{i}}(\mathbf{Y})d\mathbf{Y} = I_{\mathbf{j}}^{\mathbf{i}}$, we can rewrite Eq. (3.44) as

$$\mathbb{E}[u(\boldsymbol{x})] = \sum_{|\mathbf{i}|\leq q} \sum_{\mathbf{j}\in B_{\mathbf{i}}} w_{\mathbf{j}}^{\mathbf{i}}(\boldsymbol{x}) \cdot I_{\mathbf{j}}^{\mathbf{i}}. \tag{3.46}$$

Thus, the mean is just an arithmetic sum of the product of the hierarchical surpluses and the integral weights at each interpolation point.

To obtain the variance of the solution, we need to first obtain an approximate expression for u^2, i.e.,

$$u^2(\boldsymbol{x}, \mathbf{Y}) = \sum_{|\mathbf{i}|\leq q} \sum_{\mathbf{j}\in B_{\mathbf{i}}} v_{\mathbf{j}}^{\mathbf{i}}(\boldsymbol{x}) \cdot a_{\mathbf{j}}^{\mathbf{i}}(\mathbf{Y}). \tag{3.47}$$

Then the variance of the solution can be computed as:

$$\begin{aligned} \text{Var[u}(\boldsymbol{x})] &= \mathbb{E}[u^2(\boldsymbol{x})] - (\mathbb{E}[u(\boldsymbol{x})])^2 \\ &= \sum_{|\mathbf{i}|\leq q} \sum_{\mathbf{j}\in B_{\mathbf{i}}} v_{\mathbf{j}}^{\mathbf{i}}(\boldsymbol{x}) \cdot I_{\mathbf{j}}^{\mathbf{i}} - \left(\sum_{|\mathbf{i}|\leq q} \sum_{\mathbf{j}\in B_{\mathbf{i}}} w_{\mathbf{j}}^{\mathbf{i}}(\boldsymbol{x}) \cdot I_{\mathbf{j}}^{\mathbf{i}} \right)^2. \end{aligned} \tag{3.48}$$

3.2 Adaptive sparse grid collocation (ASGC)

As discussed in Section 3.1.3, the magnitudes of the hierarchical surpluses decay to zero quickly as the level of interpolation increases assuming a smooth

function in the stochastic space. If the smoothness condition is not fulfilled, an adaptive sparse grid is preferred, which for example may place more points around the discontinuity region and less points in the region of smooth variation. One way to perform adaptation and refinement is on the level of the hierarchical subspaces $W^{\mathbf{i}}$ in Eq. (3.32). This leads to the so-called dimension-adaptive (anisotropic) sparse grids [39, 40, 41]. This approach detects important dimensions and places all the collocation points from the hierarchical subspace $W^{\mathbf{i}+\mathbf{e}_j}$ along the important dimension j. Thus, this method is not suitable if we want to look at the local behavior of the stochastic function. Alternatively, the adaptation and refinement process can be performed on the level of the single hierarchical basis functions $a^{\mathbf{i}}_{\mathbf{j}}$ from Eq. (3.31). We then obtain a method which, besides the detection of important dimensions, identifies and resolves singularities and local non-smooth variations in the stochastic function [29, 30, 42]. In this section, we focus on the latter method and develop an adaptive sparse grid stochastic collocation algorithm based on the error control of the hierarchical surpluses.

3.2.1 The ASGC algorithm

Before discussing the algorithm, let us first introduce some notation. The $1D$ equidistant points of the sparse grid in Eq. (3.13) can be considered as a tree-like data structure as shown in Fig. 3.3. It is noted that special treatment is needed here going from level 2 to level 3. For the nodes 0 and 1 in level 2, we only add one point along the dimension (there is only one son here instead of two sons as is the case for all other subsequent levels of interpolation). Then, we can consider the interpolation level of a grid point Y as the depth of the tree $D(Y)$.

For example, the level of point 0.25 is 3. Denote the father of a grid point as $F(Y)$, where the father of the root 0.5 is itself, i.e., $F(0.5) = 0.5$.

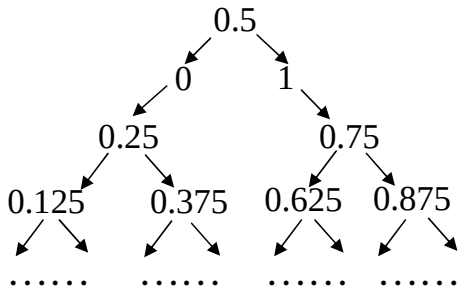


Figure 3.3: $1D$ tree-like structure of the sparse grid.

Thus, the conventional sparse grid in the $N-$dimensional random space can be reconsidered as

$$\mathcal{H}_{q,N} = \left\{ \mathbf{Y} = \{Y_1, \ldots, Y_N\} | \sum_{i=1}^{N} D(Y_i) \leq q \right\}. \tag{3.49}$$

We denote the sons of a grid point $\mathbf{Y} = (Y_1, \ldots, Y_N)$ by

$$\begin{aligned} \text{Sons}(\mathbf{Y}) = \{ & \mathbf{S} = (S_1, S_2, \ldots, S_N) | (F(S_1), S_2, \ldots, S_N) = \mathbf{Y}, \text{or} \\ & (S_1, F(S_2), \ldots, S_N) = \mathbf{Y}, \ldots, \text{or } (\mathrm{S_1}, \mathrm{S_2}, \ldots, \mathrm{F(S_N)}) = \mathbf{Y} \}. \end{aligned} \tag{3.50}$$

From this definition, it is noted that, in general, for each grid point there are two sons in each dimension, therefore, for a grid point in a N-dimensional stochastic space, there are $2N$ sons. It is also noted that, the sons are also the *neighbor points* of the father. Recall from the definition of grid points from Eq. (3.13) and the definition of hierarchical basis from Eq. (3.31) that the neighbor points are just the support nodes of the hierarchical basis functions in the next interpolation level. By adding the neighbor points, we actually add the support nodes from the next interpolation level, i.e., we perform interpolation from level $|\mathbf{i}|$ to level $|\mathbf{i}| + 1$. Therefore, in this way, we refine the grid locally while not violating the developments of the Smolyak algorithm Eq. (3.37).

The basic idea here is to use hierarchical surpluses as an error indicator to detect the smoothness of the solution and refine the hierarchical basis functions $a_{\mathbf{j}}^{\mathbf{i}}$ whose magnitude of the hierarchical surplus satisfies $|w_{\mathbf{j}}^{\mathbf{i}}| \geq \varepsilon$. If this criterion is satisfied, we simply add the $2N$ neighbor points of the current point from Eq. (3.50) to the sparse grid. An example of a case with two random variables is shown in Fig. 3.4. It is noted that the growth of the points scales linearly with increasing dimensionality rather than the $O(2^N)$ tree-like scaling of the standard $h-$type adaptive refinement as in a random element-based framework, e.g. in ME-gPC.

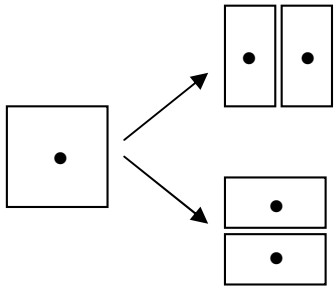


Figure 3.4: An example of nodes and supports of a locally refined sparse grid in 2D random domain.

In the Smolyak construction, we always perform the interpolation level by level. For each level, we first calculate the hierarchical surplus for each point, then we check whether the adaptive criterion $|w_{\mathbf{j}}^{\mathbf{i}}| \geq \varepsilon$ is satisfied. If so, we generate the $2N$ neighboring points. There is a possibility that the neighbors have already been generated by other points. Therefore, it is critical to keep track of the uniqueness of the newly generated neighboring points. We refer to these newly generated neighboring points as active points. To this end, we use the data structure <set> from the standard template library in C++ to store all the active points and we refer to this as the active index set. <set> is a kind of sorted associative container that stores unique elements (keys). When inserting

a new element, this data structure will check if the new element already exists. If so, it will not insert the element. If not, the element is inserted according to the ordering of the elements in the <set>. Due to the sorted nature of the <set>, the searching and inserting is always very efficient. Another advantage of using this data structure is that it is easy for a parallel code implementation. Since we store all of the new points from the next level in the <set>, we can evaluate the surplus for each point in parallel, which increases the performance significantly.

In addition, when the discontinuity is very strong, the hierarchical surpluses may decrease very slowly and the algorithm may not stop until a sufficiently high interpolation level. However, from Eq. (3.45), it is seen that the weights $I_{\mathbf{j}}^{\mathbf{i}}$ decrease very quickly as the level of interpolation increases. The same is true with the hierarchical surpluses. The contribution of this term to the mean and the variance may be neglected in comparison to a certain desired accuracy level of the statistics. Therefore, a maximum interpolation level is always specified as another stopping criterion. It is noted here that the definition of the level of the Smolyak interpolation for the ASGC method is the same as that of the conventional sparse grid even if not all points are included. The first hierarchical surplus is always the function value at the point $(0.5, \ldots, 0.5)$. There is a possibility that the function value may be zero and thus the refinement terminates immediately. In order to avoid an early stop for the refinement process, we always refine the first level and keep a provision on the first few hierarchical surpluses [29]. Therefore, let $\varepsilon > 0$ be the parameter for the adaptive refinement threshold. We propose the following iterative refinement algorithm beginning with a coarsest adaptive sparse grid $\mathcal{G}_{N,N}$, i.e., we begin with the N-dimensional multi-index $\mathbf{i} = (1, \ldots, 1)$, which is just a point $(0.5, \ldots, 0.5)$.

1. Set level of Smolyak construction $k = 0$.

2. Construct the first level adaptive sparse grid $\mathcal{G}_{N,N}$.

   - Calculate the function value at the point $(0.5, \ldots, 0.5)$;
   - Generate the $2N$ neighbor points and add them to the active index set;
   - Set $k = k + 1$.

3. While $k \leq k_{max}$ and the active index set is not empty:

   - Copy the points in the active index set to an old index set and clear the active index set.
   - Calculate in parallel the hierarchical surplus of each point in the old index set according to

     $$w_{\mathbf{j}}^{\mathbf{i}} = f(Y_{j_1}^{i_1}, \ldots, Y_{j_N}^{i_N}) - \mathcal{G}_{N+k-1,N}(f)(Y_{j_1}^{i_1}, \ldots, Y_{j_N}^{i_N}). \tag{3.51}$$

     Here, we use all of the existing collocation points in the current adaptive sparse grid $\mathcal{G}_{N+k-1,N}$. This allows us to evaluate the surplus for each point from the old index set in parallel.

   - For each point in the old index set, if $|w_{\mathbf{j}}^{\mathbf{i}}| \geq \varepsilon$
     - Generate $2N$ neighbor points of the current active point according to Eq. (3.50);
     - Add them to the active index set.
   - Add the points in the old index set to the existing adaptive sparse grid $\mathcal{G}_{N+k-1,N}$. Now the adaptive sparse grid becomes $\mathcal{G}_{N+k,N}$.
   - $k = k + 1$.

4. Calculate the mean and the variance, the PDF and if needed realizations of the solution (see Section 3.1.5).

Remark 1. In practice, instead of using $w_{\mathbf{j}}^{\mathbf{i}}$, it is sometimes preferable to use $v_{\mathbf{j}}^{\mathbf{i}}$ from Eq. (3.47) as the error indicator. This is because the hierarchical surplus $v_{\mathbf{j}}^{\mathbf{i}}$ is related to the calculation of the variance. In principle, we can consider it like a local variance. Thus, it is more sensitive to the local variation of the stochastic function than $w_{\mathbf{j}}^{\mathbf{i}}$. In all but the first example in Section 5.4, the set threshold ϵ refers to the surpluses $v_{\mathbf{j}}^{\mathbf{i}}$. Recall that for accurate calculation of the variance in Eq. (3.48), both the function and its square are interpolated independently, see Eqs. (3.46) and (3.47), respectively.

Remark 2. The algorithm developed here is different from the original algorithm in [29]. In [29], the sparse grid introduced is based on the so-called maximum-norm-based sparse grid [27, 28]. It assumes that the function value vanishes on the boundary and the hierarchical surplus is calculated through a N-dimensional stencil which gives the coefficients for a linear combination of function values at the collocation points. Generally, this kind of sparse grid is not very suitable for high-dimensional stochastic spaces [27, 28].

Remark 3. It is also noted that, in the adaptive sparse grid $\mathcal{G}$, we also keep some points whose surpluses are smaller than the threshold when they are generated from their fathers whose surpluses are larger than the threshold. In other words, we want to keep the adaptive sparse grid balanced, the so called balanced adaptivity, see [64]. This is different from the algorithm in [29], where all of the points whose hierarchical surplus is less than the threshold are omitted.

3.2.2 Convergence and accuracy of the adaptive collocation method

For a threshold ε and a fixed level $q = N + k$ of the sparse grid interpolation, the approximation $u^q_{\mathcal{H}}(\boldsymbol{x}, \mathbf{Y})$ from Eq. (3.43) of the conventional sparse grid method can be rewritten as a sum of two terms $u^q_{\mathcal{G}}$, representing the interpolation on the adaptive sparse grid $\mathcal{G}$ and $u^q_<$ that accounts for all of the missing points (see also Remark 3) whose hierarchical surpluses are below the threshold ε. Since for any piecewise N-linear basis function $a^{\mathbf{i}}_{\mathbf{j}}(\mathbf{Y})$, $\|a^{\mathbf{i}}_{\mathbf{j}}\|_\infty = 1$ [30], we can show that the error between the adaptive sparse grid interpolation and that of using conventional sparse grid is

$$\begin{aligned} \|u^q_{\mathcal{H}} - u^q_{\mathcal{G}}\|_\infty = \left\|u^q_<\right\|_\infty &= \sum_{|\mathbf{i}|\le q} \sum_{\substack{\mathbf{j}\in B_{\mathbf{i}} \\ |w^{\mathbf{i}}_{\mathbf{j}}|<\varepsilon}} |w^{\mathbf{i}}_{\mathbf{j}}(\boldsymbol{x})| \cdot \left\|a^{\mathbf{i}}_{\mathbf{j}}(\mathbf{Y})\right\|_\infty \\ &\le \varepsilon M_2, \end{aligned} \tag{3.52}$$

where M_2 is the number of all missing points. When decreasing the threshold ε, the number of missing terms M_2 also decreases (as the tolerance is reduced, more points are locally refined). Therefore, we can see that indeed the approximation of the adaptive sparse grid interpolation converges to the conventional interpolation case when decreasing the threshold ε. Accordingly, the interpolation error when using the adaptive spare grid collocation method can be approximated by

$$\|u - u^q_{\mathcal{G}}\|_\infty = \left\|u - u^q_{\mathcal{H}} + u^q_{\mathcal{H}} - u^q_{\mathcal{G}}\right\|_\infty \le \left\|u - u^q_{\mathcal{H}}\right\|_\infty + \left\|u^q_{\mathcal{H}} - u^q_{\mathcal{G}}\right\|_\infty . \tag{3.53}$$

The first term in the equation above is the interpolation error of the conventional sparse grid collocation method (see Eq. (3.42)). The second term is the error between the conventional and adaptive sparse grid collocation methods that

was shown to be of the order of $O(\varepsilon)$. Numerical investigation of these errors are provided in Section 3.3.1.

Hereafter, for convenience, we use CSGC to denote the conventional sparse grid collocation method from Eq. (3.37) using multi-linear basis functions and ASGC to denote the adaptive sparse grid collocation method from the algorithm introduced in this section with the same basis functions.

3.3 Numerical examples

This section consists of four examples. The first example is used to demonstrate the failure of the dimension-adaptive method when the singularity is not aligned along the grid. On the second example, we compare our method with MC and the multi-element based method on a benchmark problem involving stochastic discontinuity. In the third example, we assess the ability of ASGC to detect the important dimension in a high dimensional stochastic elliptic problem. In the last example, Rayleigh-Bénard instability is considered to showcase the use of the method in physical problems.

3.3.1 Approximation of function with regularized line singularity

In this section, we demonstrate the ability of the ASGC method in interpolating given functions. The computed results are compared with the CSGC method.

We consider the function on $[0,1]^2$:

$$f(x,y) = \frac{1}{|0.3 - x^2 - y^2| + \delta}, \tag{3.54}$$

where $\delta = 10^{-1}$. We first construct the interpolant $\mathcal{A}_{q,2}(f)$, then we randomly generate 1000 points in $[0,1]^2$ and finally compute the error as follows:

$$e = \max_{i=1,\ldots,1000} |f(\boldsymbol{x}_i) - \mathcal{A}_{q,2}(f)(\boldsymbol{x}_i)|. \tag{3.55}$$

The function of interest has a line singularity that is not along the grid lines, see Fig. 3.5. From the convergence plot with respect to ε on the left of Fig. 3.6, it is seen that the error converges nearly exponentially fast with respect to ε. On the right of Fig. 3.6, the convergence rate is shown with respect to the needed number of points for different thresholds. For example, it is noted that for threshold $\varepsilon = 10^{-3}$ more points are needed than when using the other two thresholds shown but a higher level of accuracy is obtained. Also note that much less points are needed in the ASGC than in the CSGC to achieve the same accuracy. The highest accuracy achieved for ASGC is 6.09×10^{-3}, where the interpolation level is 19 and the number of points is 16659 as opposed to 6029313 points using the same level of CSGC. The evolution of the adaptive grid for threshold $\varepsilon = 10^{-3}$ is shown in Fig. 3.7. The line of discontinuity is automatically detected by the ASGC method.

Since the line singularity is not along any dimension, it is expected that the dimension-adaptive (anisotropic) sparse grid method [39, 41] fails in this case. The results are shown in Fig. 3.8, where the algorithm is implemented using the MatLab Sparse Grid Interpolation Toolbox developed by Klimke [65]. From the convergence plot, it is interesting to note that the convergence rate is nearly the same as that of the CSGC method in Fig. 3.6. This is because the line singularity

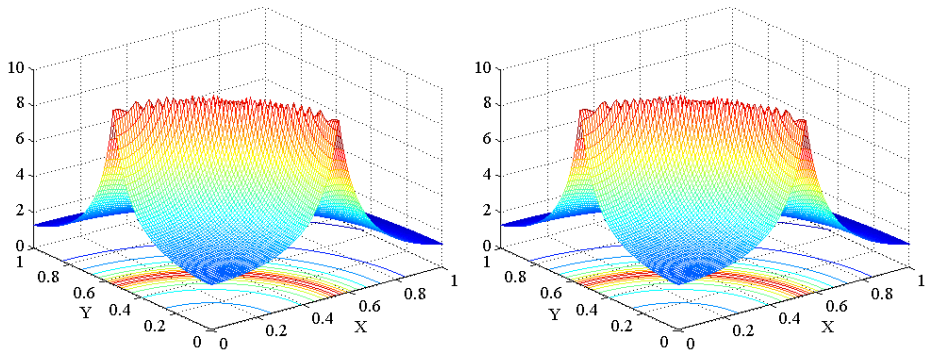


Figure 3.5: Line singularity: Comparison of the exact (left) and interpolant (right) functions using the ASGC method with threshold $\varepsilon = 10^{-3}$.

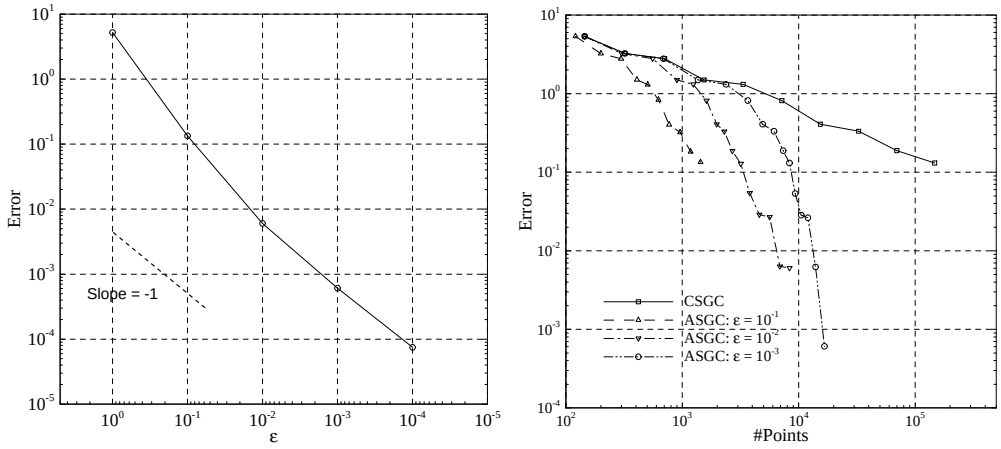


Figure 3.6: Line singularity: (left) Convergence of the ASGC method with respect to the threshold ε; (right) Comparison of the interpolation error for conventional and adaptive sparse grid interpolation using different threshold ε values.

results in the same importance of both dimensions and the anisotropic method thus puts points in all dimensions. This is seen from the sparse grid in Fig. 3.8, where the grid is nearly the same as the full tensor product case. Therefore, this example verifies that if the singularity is not exactly along the dimensions, the dimension-adaptive sparse grid method is not applicable and identifies the need to develop a different adaptive strategy that works directly on the hierarchical basis as the one presented in this chapter.

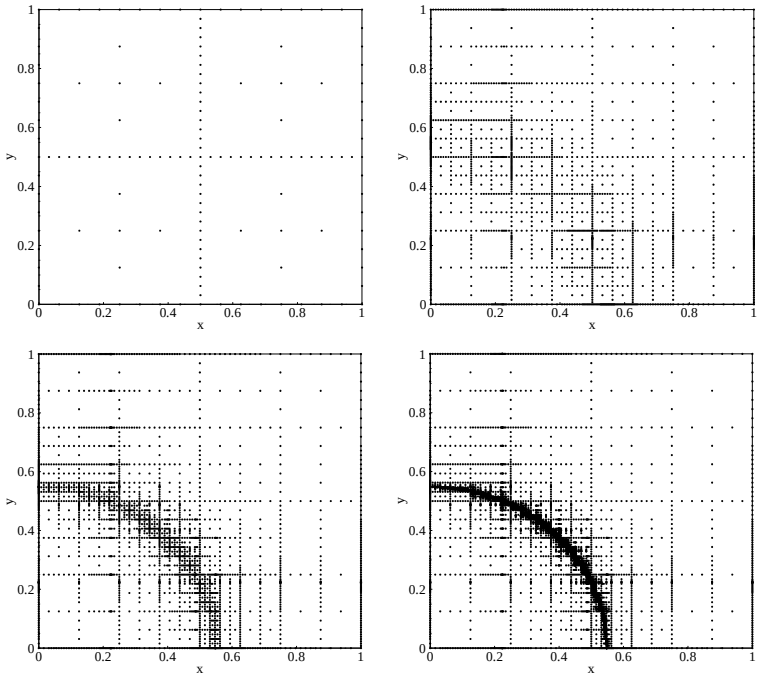


Figure 3.7: Line singularity: The evolution of the adaptive sparse grid with threshold $\varepsilon = 10^{-3}$.

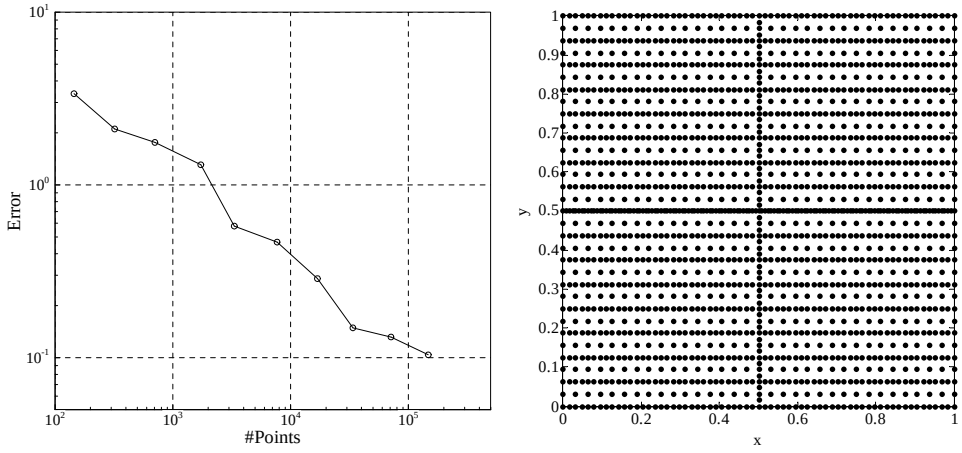


Figure 3.8: Line singularity: (left) Convergence of the dimension-adaptive method; (right) Dimension-adaptive sparse grid.

3.3.2 A dynamical system with discontinuity

This problem that was originally addressed using Wiener-Haar expansion in [18] has an analytical solution. We consider the following governing equa-

tion for a particle moving under the influence of a potential field and friction force:

$$\frac{d^2X}{dt^2} + f\frac{dX}{dt} = -\frac{dh}{dx}, \tag{3.56}$$

with parameters $f > 0$, dh/dx and two initial conditions: $X(t = 0) = x_0$ and $dX/dt(t = 0) = v_0$. If we set the potential field as $h(x) = (35/8)x^4 - (15/4)x^2$, then the differential equation has two stable fixed points ($x = \pm\sqrt{15/35}$) and an unstable fixed point at $x = 0$. The stochastic version of this problem with random initial position in the interval $[x_1, x_2]$ can be expressed as

$$\frac{d^2X}{dt^2} + f\frac{dX}{dt} = -\frac{35}{2}X^3 + \frac{15}{2}X, \tag{3.57}$$

with stochastic initial condition

$$X(t = 0, Y) = X_0 + \Delta X Y, \quad \frac{dX}{dt}|_{t=0} = 0, \tag{3.58}$$

where $X(t, Y)$ denotes the response of the stochastic system, $X_0 \equiv (x_1 + x_2)/2$, $\Delta X = |x_1 - x_2|/2$, and Y is uniformly distributed random variable over $[-1, 1]$. If we choose $X_0 = 0.05$, $\Delta X = 0.2$ and a relatively large friction coefficient $f = 2$, a steady-state solution is achieved in a short time. The analytical steady-state solution is given by

$$\begin{cases} X(t \to \infty, Y) = -\sqrt{15/35}, & Y < -0.25, \\ X(t \to \infty, Y) = \sqrt{15/35}, & Y > -0.25. \end{cases} \tag{3.59}$$

This results in the following statistical moments, $\mathbb{E}[X(t \to \infty, Y)] = \frac{1}{4}\sqrt{15/35}$ and $\mathrm{Var}[X(t \to \infty, Y)] = \frac{45}{112}$. In the following computations, the time integration of Eq. (3.57) is performed using a fourth-order Runge-Kutta scheme and a time step $\Delta t = 0.001$ is used.

Failure of gPC and Lagrange polynomial interpolation

First, we apply the gPC method to the model problem (Fig. 3.9). It is shown that, no matter what the expansion order is, the gPC method cannot give accurate representation of the discontinuous behavior of the solution. The gPC solution results in unphysical oscillations in the realization of X at $t = 10$ as shown in the middle column of Fig. 3.9. For the steady-state solution at $t = 25$, these oscillation still exist, see left plot in Fig. 3.10. On the right of Fig. 3.10 similar results are shown obtained with the sparse grid collocation method using Lagrange polynomials. The existence of these wiggles is a typical characteristic of the Gibbs phenomenon when applying global spectral decomposition to problems with discontinuities in the random space [18]. The corresponding PDF of the solution X at $t = 10$ is also shown on the right column of Fig. 3.9. The analytical PDF should be two Dirac masses with unequal strength located at the two equilibrium points. However, it can be seen from Fig. 3.9 that the gPC solution exhibits a broad spectrum far from the equilibrium points and there are several unphysical peaks.

To further illustrate this point, convergence of the mean and the variance at steady-state are shown in Fig. 3.11. It is obvious that the results fail to converge with increasing expansion orders. A similar plot is presented in Fig. 3.12 using the sparse grid collocation method with Lagrange polynomials. The Lagrange polynomials as the chaos polynomials, are of global support, so similar oscillations exist in the solution of X at $t = 10$ (especially when the interpolation level is 5). Unlike the gPC, increasing the interpolation level can significantly reduce the wiggles in the solution, which is noted for the steady-state solution at $t = 25$ in the right plot in Fig. 3.10. For level of interpolation 10, the solution at $t = 10$ in

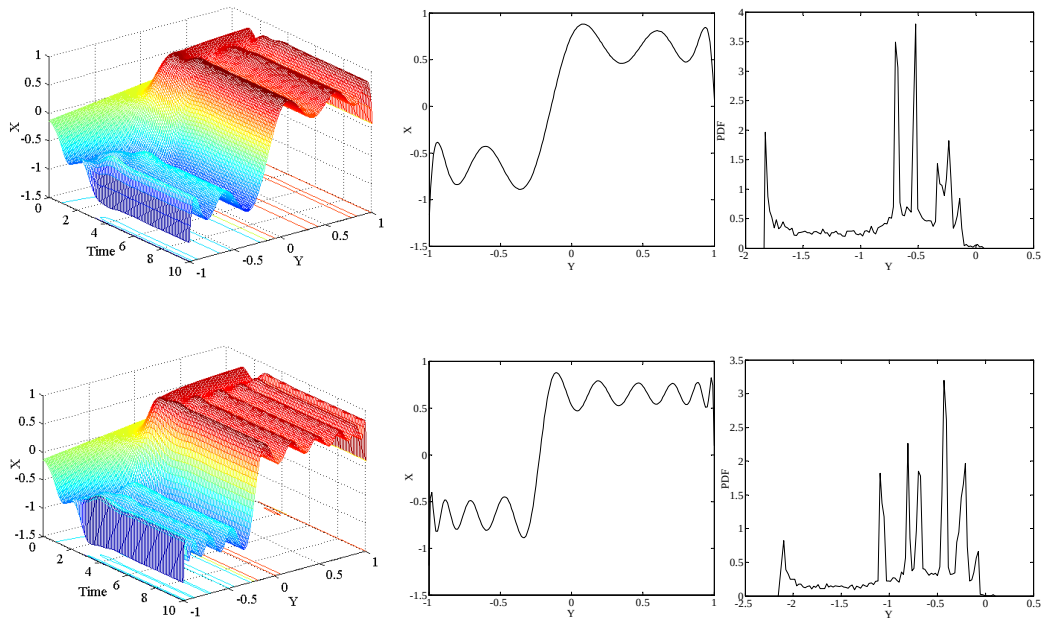


Figure 3.9: gPC solution of the model problem. The left column shows the evolution of $X(t, Y)$ for $0 \leq t \leq 10$. The solution at $t = 10$ as a function of Y is plotted in the middle column and the corresponding PDF is shown in the right column. Results are obtained for gPC expansion orders 10 (top) and 20 (bottom).

Fig. 3.12 suggests that there exists a discontinuity in the solution and there are two equilibrium points. However, near the discontinuity point of the solution, there are still several wriggles, which can be further noted in the corresponding PDF. These wiggles will not disappear even for higher interpolation levels due to the Gibbs phenomenon. Better predictions of the low-order moments are obtained than the gPC method as shown in Fig. 3.13. Though the results indeed converge, the converged values are not accurate.

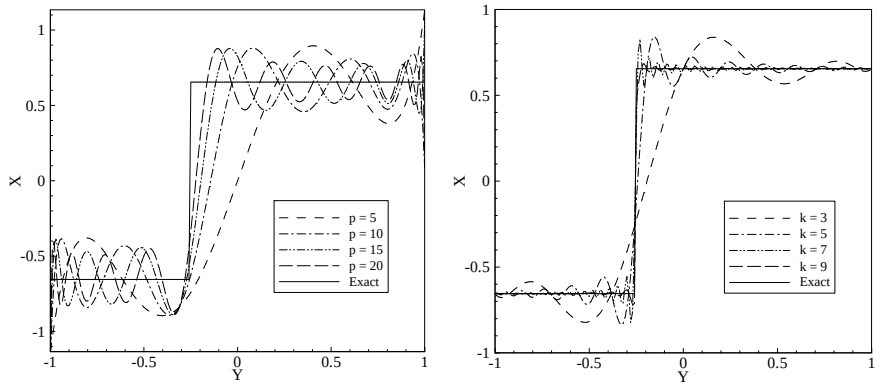


Figure 3.10: Steady-state solution at $t = 25$ of the model problem. Left: gPC method with different expansion orders; Right: Conventional sparse grid collocation method using Lagrange polynomials with different interpolation levels.

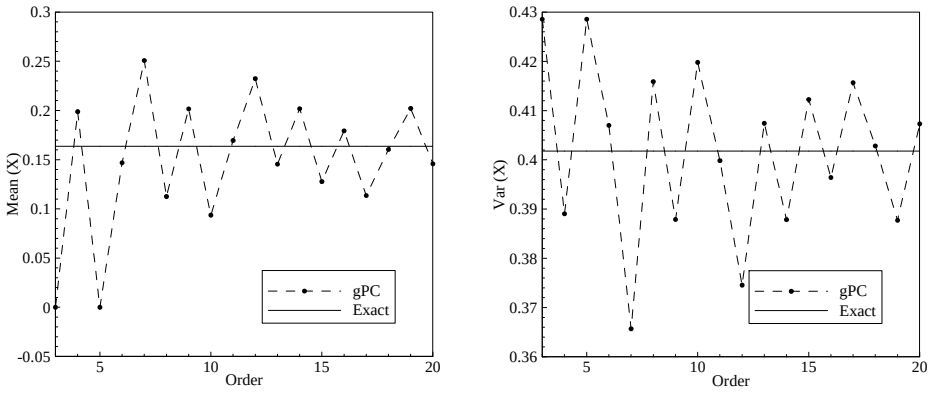


Figure 3.11: Mean (left) and variance (right) using the gPC expansion for the model problem with expansion orders from 3 to 20.

Conventional sparse grid collocation with linear basis functions and adaptivity

We now apply the CSGC method with linear basis functions to the model problem. The linear basis functions have local support, so it is expected that they should correctly resolve the discontinuity. The results are shown in Figs. 3.14 and 3.15. It is indicated that with increasing interpolation level, the CSGC method can provide an adequate representation of the discontinuous behav-

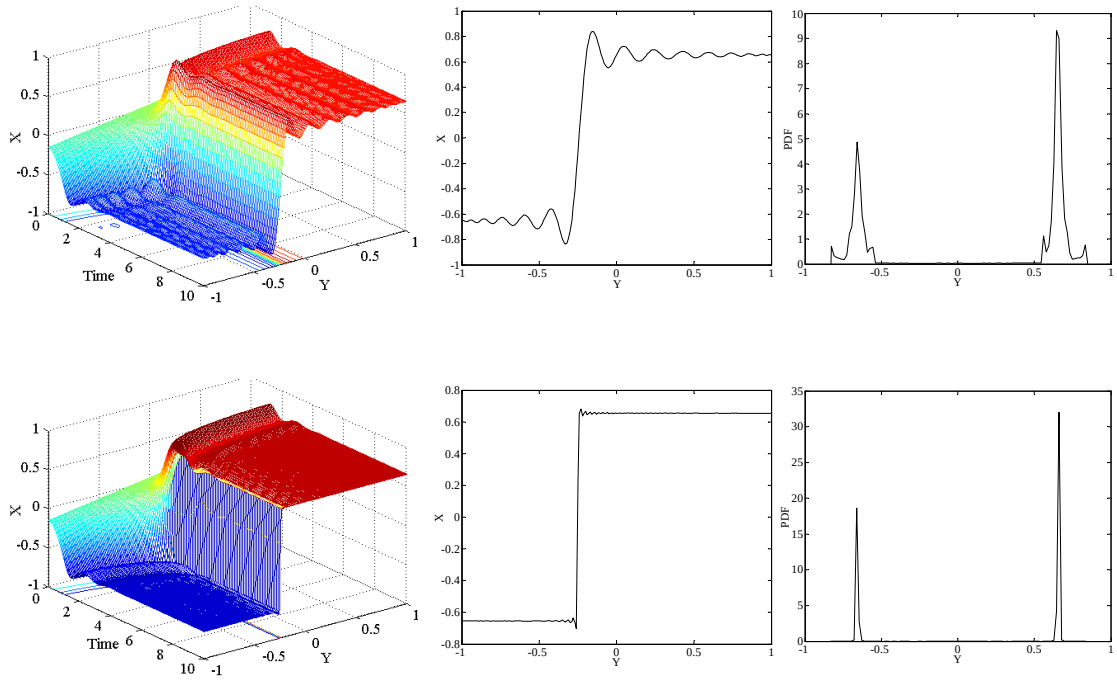


Figure 3.12: Solution of the model problem using the sparse grid collocation method with Lagrange polynomial interpolation. The left column shows the evolution of $X(t, Y)$ for $0 \leq t \leq 10$. The solution at $t = 10$ as a function of Y is plotted in the middle column and the corresponding PDF is shown in the right column. Results are obtained for interpolation levels 5 (top) and 10 (bottom).

ior of the solution. Note that unlike the solutions obtained in the last section, there are no oscillations in the solutions shown here. Even for low interpolation level, the computed solution provides an insight to the existence of discontinuity. In the right plot of Fig. 3.15, the PDF is indeed two Dirac masses with unequal strength. Therefore, basis functions of local support are able to resolve the discontinuity. This is the same idea as the Wiener-Haar expansion [18], where piecewise constant representation of the stochastic support space was used. However, note that the CSGC method is expected to be faster than the Wiener-Haar expansion approach, since it only requires repeated calls to a deterministic solver, while the Wiener-Haar expansion results in coupled equations

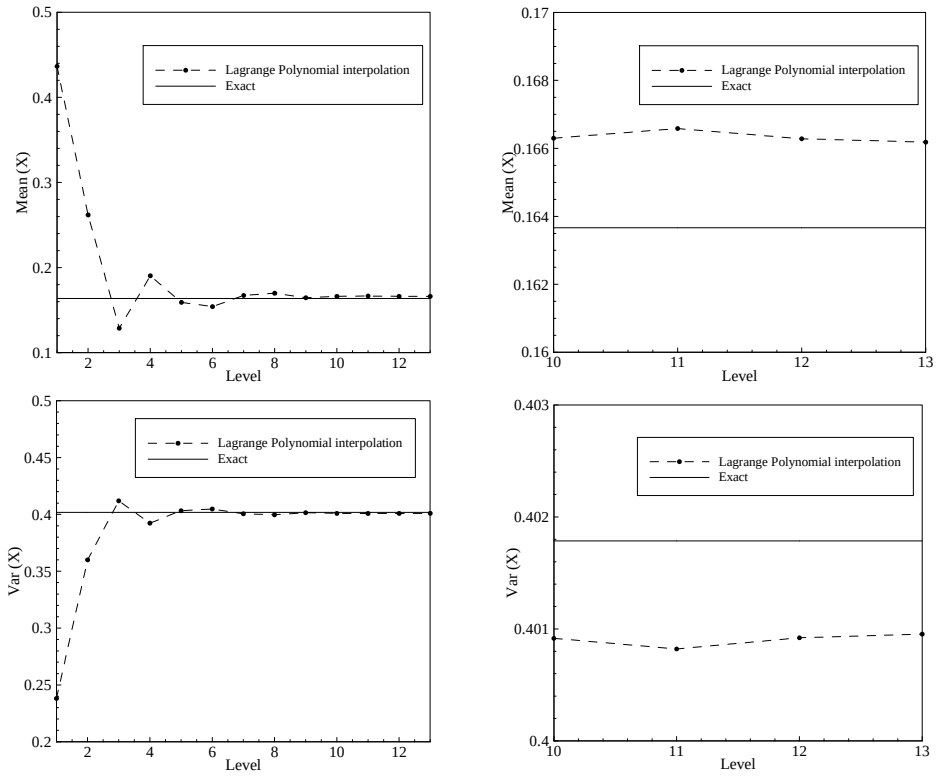


Figure 3.13: Mean (top) and variance (bottom) using sparse grid collocation with Lagrange polynomial basis. The results in the right column are the detailed view from interpolation level 10 to 13.

as is the case with the gPC expansion.

The error of the mean decreases very fast to about 10^{-12} at interpolation level 3. So we only show the error of the variance in Fig. 3.16, where comparison with the results of the ASGC is also made. Here, it is noted that since the solution is very smooth far from the discontinuity point, the magnitude of the hierarchical surpluses are also very small in the first three levels. In order to avoid early stop of the refinement, we always refine the first three levels. For the CSGC method, a maximum interpolation level of 15 is used and 32769 points are needed to reach accuracy of 1×10^{-5}. For the ASGC method, a maximum interpolation level of 30 is used. Only 115 number of points are required to achieve accuracy

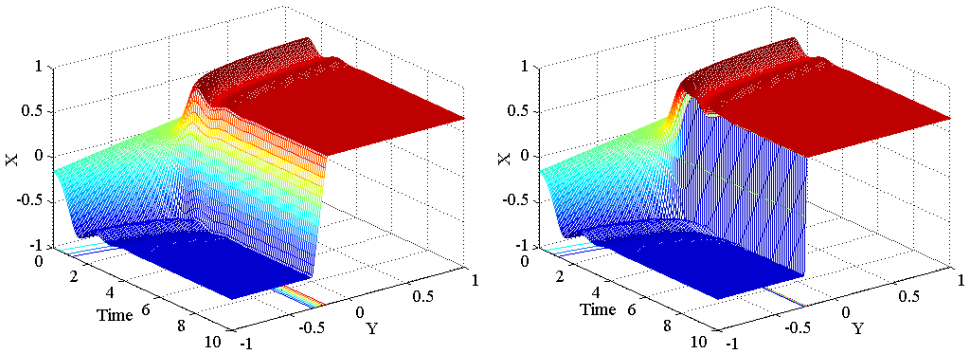


Figure 3.14: Solution of the model problem using multi-linear sparse grid collocation method. The left column shows the evolution of $X(t, Y)$ for $0 \leq t \leq 10$. Results are obtained for interpolation levels 5 (left) and 10 (right).

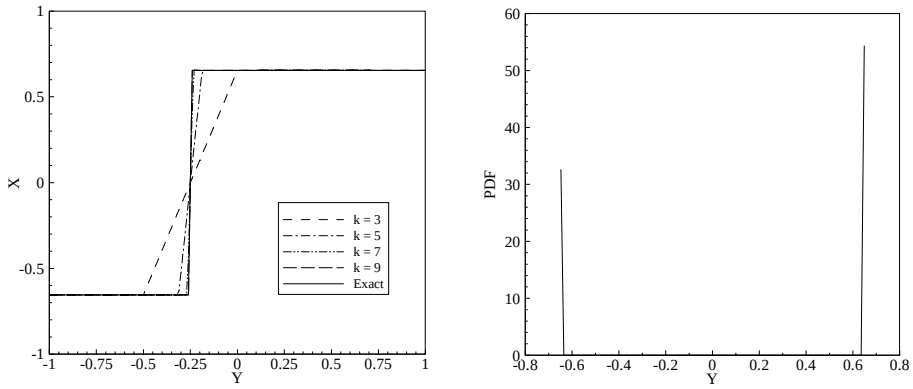


Figure 3.15: Left: Steady-state solution at $t = 25$ of the model problem with different interpolation levels. Right: The corresponding PDF using interpolation level 10.

of 3×10^{-10}. This is because the magnitude of the hierarchical surpluses are very small in the smooth part of the stochastic support space, i.e. about order of 10^{-15}, and only points near the discontinuity point (-0.25) are refined. This is noted in the right plot of Fig. 3.16 (only first 20 levels are shown), where the points are only refined after the appearance of collocation point -0.25, where the magnitude of the hierarchical surplus is very large. As mentioned before, for very strong discontinuities, the magnitude of the hierarchical surplus decreases very slowly, thus we need to specify a maximum interpolation level to force

the refinement to stop. If we use the CSGC to achieve the same accuracy, an interpolation level of 30 must be used, which corresponds to 1.0737×10^9 number of collocation points. This shows the advantage of the ASGC method.

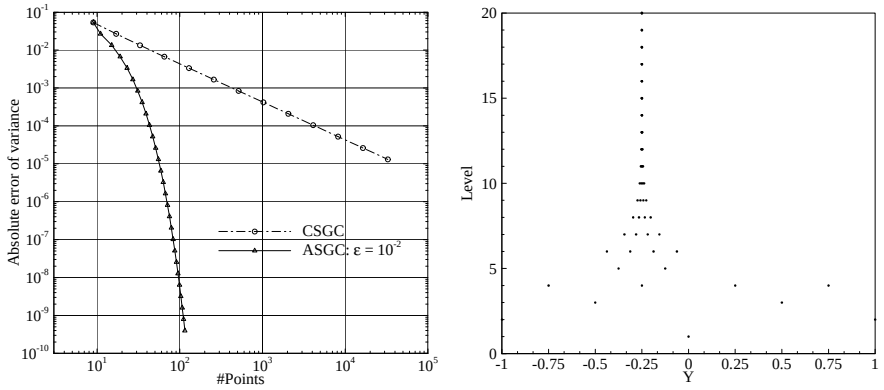


Figure 3.16: Left: Comparison of the absolute error of the variance obtained using the conventional and adaptive sparse grid methods. Right: The corresponding adaptive sparse grid.

Highly discontinuous solution

Now we consider a more difficult problem as in [18]. Specifically, we set $f = 0.05$, $X_0 = 1$ and $\Delta X = 0.1$. All other conditions remain as before. The reduction of the friction coefficient, together with higher initial energy will result in the oscillation of the particle for several cycles between each of the two stable equilibrium points. The results using the CSGC method with linear basis functions are shown in Fig. 3.17. An interpolation level 10 can successfully resolve several discontinuities in each cycle. It is noted from Fig. 3.18 that both the gPC and the collocation method with Lagrange polynomials fail to provide an accurate representation of the behavior of the solution. Unlike the solution in the previous section, the failure of Lagrange polynomial interpolation is much more obvious. Overall in this section the accuracy and efficiency of the sparse

grid collocation method with piecewise multi-linear basis functions is verified through a simple dynamical problem involving discontinuities. We proceed next to apply the ASGC method to several benchmark problems.

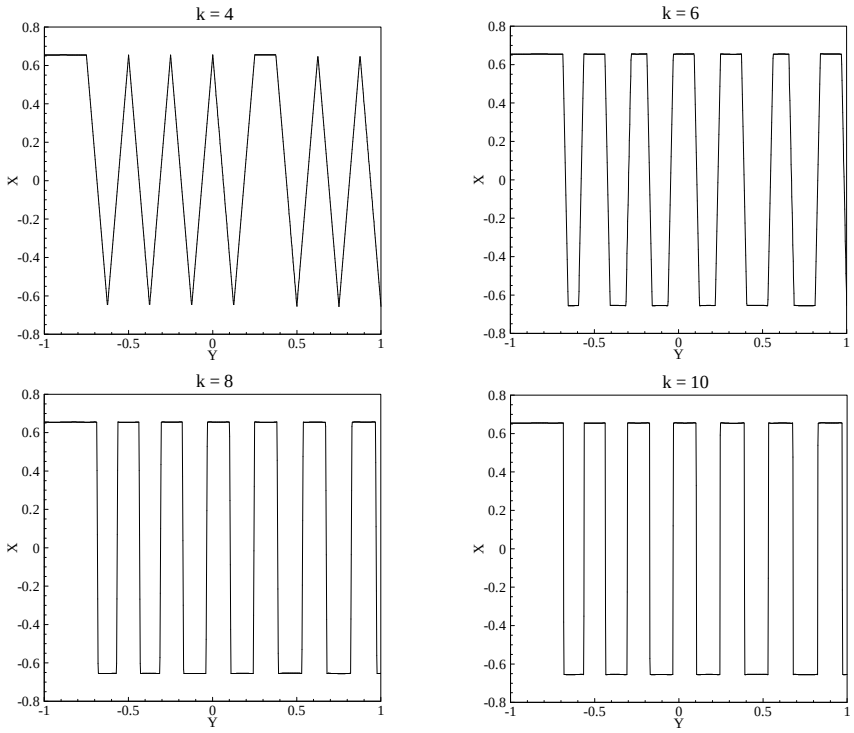


Figure 3.17: Steady-state solution at $t = 250$ of the highly discontinuous solution with different interpolation levels using the CSGC method with piece-wise multi-linear basis functions.

3.3.3 Kraichnan-Orszag (K-O) problem

The transformed Kraichnan-Orszag three-mode problem can be expressed as [20]

$$\begin{aligned} \frac{dy_1}{dt} &= y_1 y_3, \\ \frac{dy_2}{dt} &= -y_2 y_3, \\ \frac{dy_3}{dt} &= -y_1^2 + y_2^2, \end{aligned} \tag{3.60}$$

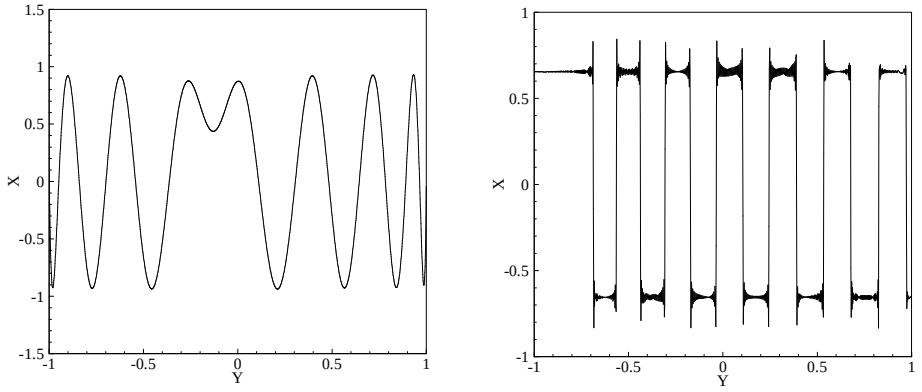


Figure 3.18: Steady-state solution at $t = 250$ of the highly discontinuous solution using gPC with expansion order 20 (left) and conventional sparse grid collocation with Lagrange polynomial interpolation of interpolation level 10 (right).

subject to initial conditions

$$y_1(0) = Y_1(0;\omega), \qquad y_2(0) = Y_2(0;\omega), \qquad y_3(0) = Y_3(0;\omega). \tag{3.61}$$

This problem shows a bifurcation on the parameter $y_1(0)$ and $y_2(0)$. The deterministic solutions of the problem are periodic, and the period goes to infinity if the initial conditions are located at the planes $y_1 = 0$ and $y_2 = 0$, i.e. discontinuity occurs when the initial conditions cross these two planes [20]. Here, we choose the random initial conditions subject to the uniform distribution $Y \sim U(-1, 1)$. In this formulation, the initial conditions cross the discontinuity plane and thus as expected the gPC method fails in computing the solution to this problem. This problem was originally solved using ME-gPC and ME-PCM in [20, 21, 34]. Here, we are addressing this problem using the ASGC method. The time integration of Eq. (3.60) is performed using a fourth-order Runge-Kutta scheme. In all computations described in this section, a time step $\Delta t = 0.01$ was used. Error convergence and comparison of computation cost with both ME-gPC and ME-PCM are conducted.

One-dimensional random input

At first, we study the following random initial conditions:

$$y_1(0) = 1.0, \qquad y_2(0) = 0.1Y(0;\omega), \qquad y_3(0) = 0. \tag{3.62}$$

In Fig. 3.19, we show the evolution of the variance within the time interval $[0, 30]$ (short time behavior). For comparison, the results of gPC are also included. The 'exact' solution is obtained using a quasi-random Sobol (MC-SOBOL) sequence with 10^6 iterations. Due to the discontinuity, the result from MC-SOBOL is much more accurate than the standard MC simulation directly sampling from the uniform distribution. It is seen that the gPC begins to fail at time $t \approx 8$, while the ASGC method converges even with a large threshold $\varepsilon = 0.1$. From the adaptive sparse grid in Fig. 3.19, it is noted that even though most of the points are refined as a result of the small threshold ε, most of the refinement after level 8 occurs around the discontinuity point $Y = 0.0$. The refinement stops at level 16, which corresponds to 425 number of points, while the conventional sparse grid requires 65537 points.

The maximum error of the variance of y_1, y_2 and y_3 at $t = 30$ is tabulated in Table 3.1. The maximum error of the variance is defined as $\max_{i=1,2,3} |\mathrm{Var}(y_i) - \mathrm{Var}(y_{i,MC})|$ at $t = 30$. The 'exact' solution is taken as the results given by MC-SOBOL 10^6 iterations. For each threshold ε, we show the level when the refinement stops, the corresponding number of collocation points and the error. It is seen that, with decreasing threshold, the stopping interpolation levels and the number of collocation points increase. At the same time, the accuracy becomes better and we can approximately obtain an error of the order of 0.01ε. We have also tabulated the error using MC-SOBOL and MC with the same number of samples. Although the MC-SOBOL is a quasi-MC method with better convergence than

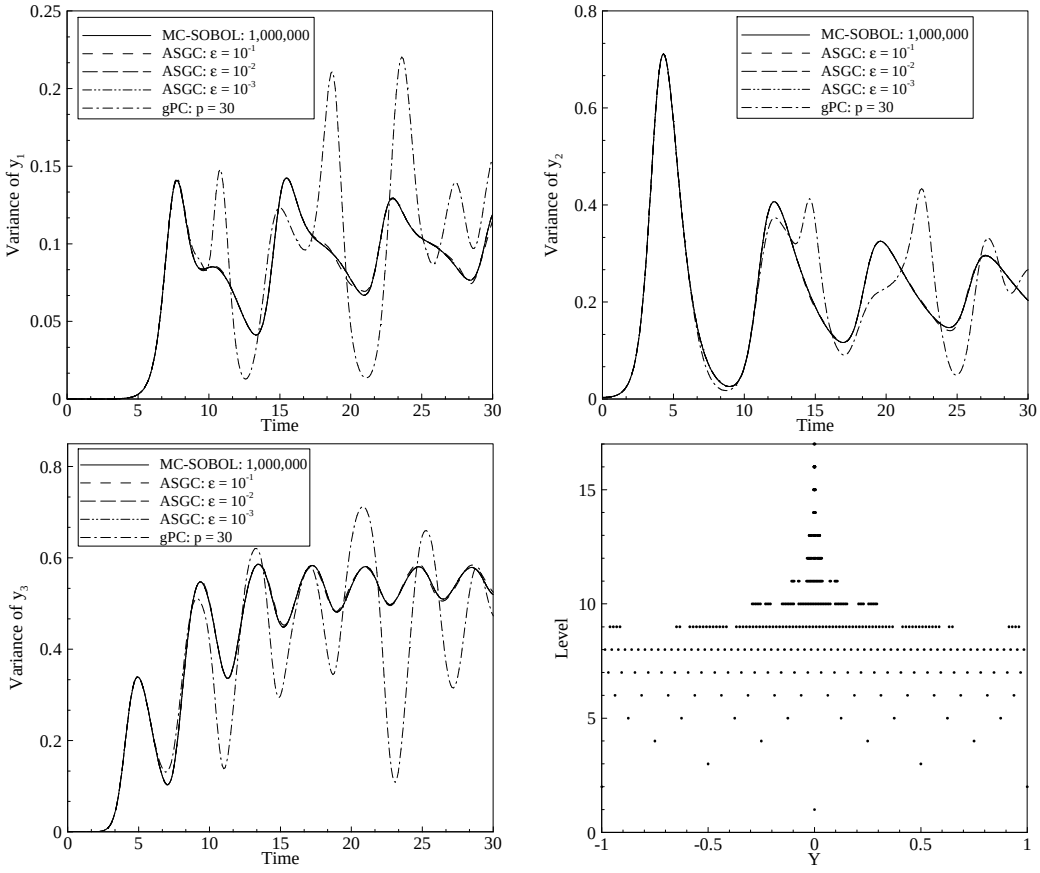


Figure 3.19: Evolution of the variance of the solution for 1D random input. Top left: y_1, Top right: y_2, Bottom left: y_3, Bottom right: Adaptive sparse grid with $\varepsilon = 10^{-2}$.

the standard MC method, from the table it can be seen that the ASGC procedure approximately leads to 2 and 1 orders of magnitude reduction in the error, as compared to MC and MC-SOBOL, respectively. We then compare the computational cost between the ASGC and multi-element based methods (Table 3.2). The error level is achieved by decreasing the error threshold ε in AGSC and θ_1 in both ME-gPC and ME-PCM. It is noted that we conduct the ME-gPC using a third-order expansion while we use linear basis functions in ASGC. In ME-PCM, a level 3 Clenshaw-Curtis sparse grid is used in each element. From the results, we note that for comparable accuracy, both the ASGC method and

ME-PCM are much faster than the *h*-adaptive ME-gPC. Although using higher-order gPC expansion can reduce the number of random elements, the increase of the expansion terms results in more computation time. On the other hand, the computational time is nearly the same for ASGC and ME-PCM. However, it is more meaningful to compare the number of function calls since both methods are based on a collocation algorithm. Many more points are needed for the ME-PCM than the ASGC to achieve the same accuracy.

Table 3.1: Comparison of maximum error of the variance of y_1, y_2 and y_3 at $t = 30$ for the 1D K-O problem for the same number of sample points.

ASGC				MC	MC-SOBOL
Threshold	Level	#Points	Error	Error	Error
10^{-1}	11	117	5.13×10^{-3}	4.62×10^{-2}	4.27×10^{-2}
10^{-2}	16	425	2.34×10^{-4}	1.40×10^{-2}	1.28×10^{-3}
10^{-3}	17	1381	2.08×10^{-5}	7.71×10^{-3}	5.00×10^{-4}
10^{-4}	26	5349	2.59×10^{-6}	2.85×10^{-3}	9.54×10^{-5}

The long-term behavior of the solution is presented in Fig. 3.20 within the time interval $[0, 100]$. The corresponding realizations at different times as a function of random variable Y are given in Fig. 3.21. These realizations are reconstructed using hierarchial surpluses according to Eq. (3.43). It is seen that at early times, the discontinuity has not yet been developed, which explains the reason the gPC is accurate at earlier times. With increasing time, the discontinuity is growing stronger and the solution is very oscillatory. Thus, many more interpolation points are needed to successfully resolve the discontinuity.

Table 3.2: Comparison of computational costs for the 1D K-O problem.

Error level	ASGC		h-adaptive ME-gPC		h-adaptive ME-PCM	
	Time(s)	#Points	Time(s)	#Elems	Time(s)	#Points
10^{-2}	0.03	21	13.75	16	0.38	126
10^{-3}	0.25	117	51.11	58	0.87	288
10^{-4}	1.70	425	82.02	92	7.12	2304
10^{-5}	13.65	1381	97.42	110	14.26	4608

Two-dimensional random input

In this section, we study the K-O problem with 2D random input:

$$y_1(0) = 1.0, \qquad y_2(0) = 0.1Y_1(0;\omega), \qquad y_3(0) = Y_2(0;\omega). \tag{3.63}$$

Now, instead of a point, the discontinuity region becomes a line. In Fig. 3.22, we show the evolution of the variance of y_1, y_2, y_3 and the adaptive sparse grid. We restrict the maximum interpolation level to 20, which is sufficient for achieving desired accuracy. As before, we also include for comparison the result obtained by the gPC with order $p = 10$. From these results, it can be seen that even though the gPC fails at a larger time, the adaptive collocation method converges to the reference solution given by MC-SOBOL with 10^6 iterations. From the adaptive sparse grid, we can see that more points are placed around the line $Y_1 = 0$, since we know that the discontinuity crosses the plane $Y_1 = 0$. The maximum error of the variance of y_1, y_2 and y_3 at $t = 10$ is tabulated in Table 3.3. The 'exact' solution is taken as the results given by MC-SOBOL as before. Due to the increased discontinuity, the reduction of the error from ASGC compared with

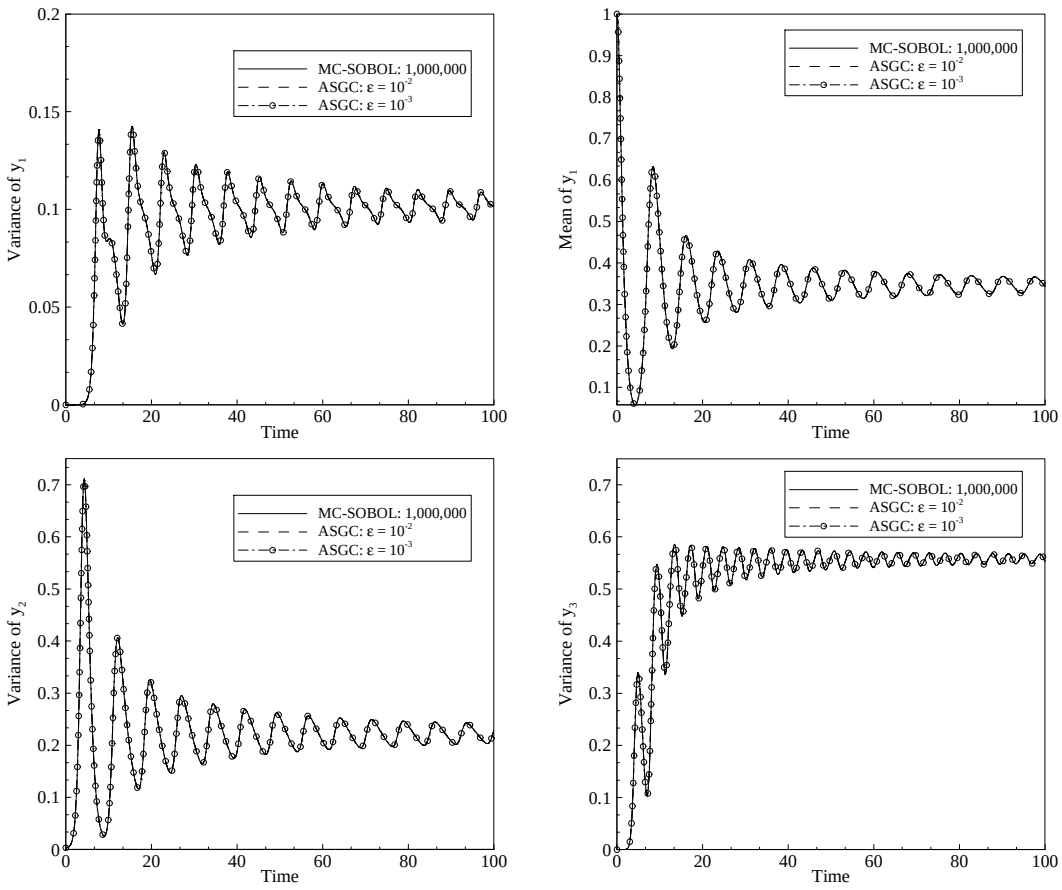


Figure 3.20: Long term behavior of the solution for 1D random input. Top left: Variance of y_1, Top right: Mean of y_1, Bottom left: Variance of y_2, Bottom right: Variance of y_3.

that of the MC-SOBOL is not as significant as in the one-dimensional input case. However, the error is still much better than that of the standard MC method. The comparison of the computational cost is given in Table 3.4. The error level is defined the same as before. We still use a third-order expansion for ME-gPC and level 3 Clenshaw-Curtis sparse grid for ME-PCM and vary θ_1 while fixing $\theta_2 = 10^{-2}$. In this situation, the speed up of the ASGC and ME-PCM with respect to the ME-gPC is still obvious. It is interesting to note that the ME-PCM is much faster than ASGC, although the number of collocation points is about ten times more than that of ASGC. This is because the computational time of the

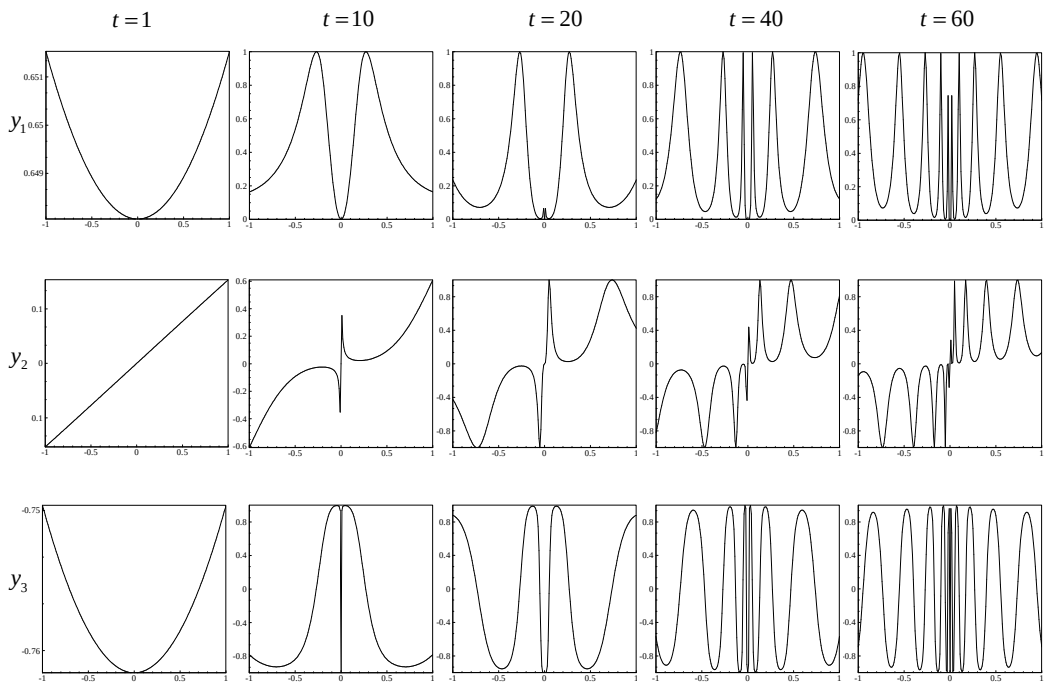


Figure 3.21: Realizations of the solution (y_1, y_2, y_3) for the 1D K-O problem as a function of the random variable Y at different times.

deterministic problem may be ignored since it is a simple ODE. Most of the time in the ASGC is spent on hierarchical surplus calculation and communication between different processors due to MPI parallelization. Therefore, in terms of functional evaluations, ME-PCM is much more expensive than the ASGC.

Three-dimensional random input

In this section, we study the K-O problem with 3D random input:

$$y_1(0) = Y_1(0;\omega), \qquad y_2(0) = Y_2(0;\omega), \qquad y_3(0) = Y_3(0;\omega). \tag{3.64}$$

This problem is much more difficult than any of the other problems examined previously. This is due to the strong discontinuity (the discontinuity region now consists of the planes $Y_1 = 0$ and $Y_2 = 0$) and the moderately-high dimension.

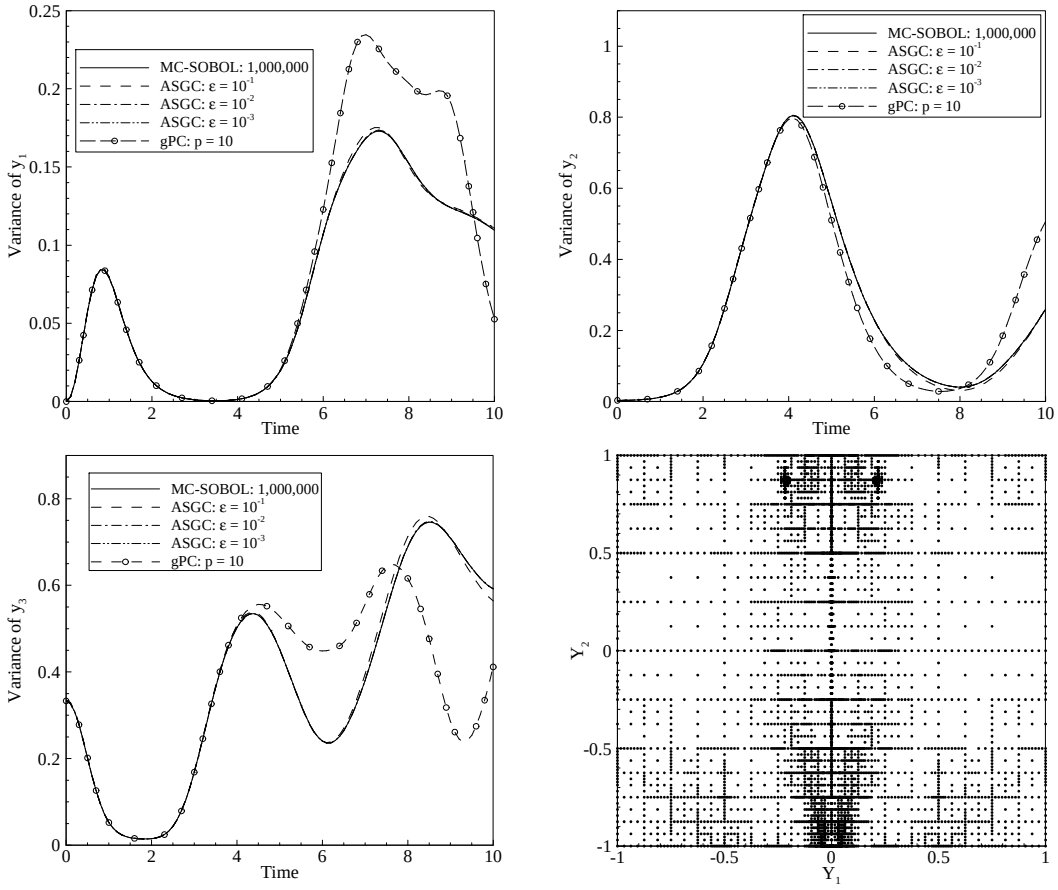


Figure 3.22: Evolution of the variance of the solution for 2D random input. The maximum interpolation level is 20. Top left: y_1, Top right: y_2, Bottom left: y_3, Bottom right: Adaptive sparse grid with $\varepsilon = 10^{-2}$.

It can be verified from comparison with the result obtained by MC-SOBOL that unlike the previous results, here 2×10^6 iterations are needed to correctly resolve the discontinuity. Due to the symmetry of y_1 and y_2 in Eq. (3.60) and the corresponding random input, the variances of y_1 and y_2 are the same. Therefore, in Fig. 3.23 we only show the results for y_1 and y_3. The maximum interpolation level is set at 15. Finally, in order to show the implementations of *h*-adaptive multi-element based methods are correct, we provide the results from both methods in Fig. 3.24. The computation results for the ASGC, ME-gPC and ME-PCM methods are shown in Tables 3.5–3.7, respectively, where we fix $\theta_1 = 10^{-4}$

Table 3.3: Comparison of maximum error of the variance of y_1, y_2 and y_3 at $t = 10$ for the 2D K-O problem for the same number of sample points.

ASGC			MC	MC-SOBOL
Threshold	#Points	Error	Error	Error
10^{-1}	903	3.95×10^{-2}	4.62×10^{-2}	1.30×10^{-3}
10^{-2}	6457	1.50×10^{-4}	1.21×10^{-2}	9.43×10^{-4}
10^{-3}	23099	1.83×10^{-5}	3.34×10^{-3}	5.25×10^{-5}
10^{-4}	79213	5.43×10^{-6}	1.26×10^{-3}	6.46×10^{-5}

Table 3.4: Comparison of computational costs for the 2D K-O problem.

Error level	ASGC		h-adaptive ME-gPC		h-adaptive ME-PCM	
	Time(s)	#Points	Time(s)	#Elems	Time(s)	#Points
10^{-2}	5.82	1575	20.89	22	0.78	232
10^{-3}	17.13	2251	34.69	86	1.78	3944
10^{-4}	27.01	6457	118.29	408	11.17	67280
10^{-5}	144.48	23099	383.57	1612	29.50	213324

and $\theta_2 = 10^{-3}$. The error is defined as the maximum of the absolute error of the variance of y_2 and y_3 at time $t = 10$ from the 'exact' solution given by MC-SOBOL with 2×10^6 iterations. It is seen that although the ASGC method has larger error compared with MC-SOBOL and the convergence rate is not optimal, it has better accuracy than the standard MC method. In addition, it is still much faster than the ME-gPC for a comparable accuracy of the order of 10^{-4}. Due to the strong discontinuity in this problem, it took much longer time for both the

ASGC and ME-PCM to arrive at the same accuracy than in the 2D problem. It is interesting to note that ME-PCM is the fastest method in this case. However, as before, to achieve the same accuracy as the ASGC, many more points are required for the ME-PCM. The advantage of ME-PCM is its p-type convergence such that the error quickly drops to the order of 10^{-6} when interpolation level increases to 8 in each element. However, the number of function evaluations is 10 times more than that of the reference solution. Therefore, the efficiency of all the methods over the MC method is not as obvious as in the previous two examples. In this extreme case, the MC-SOBOL is more favorable.

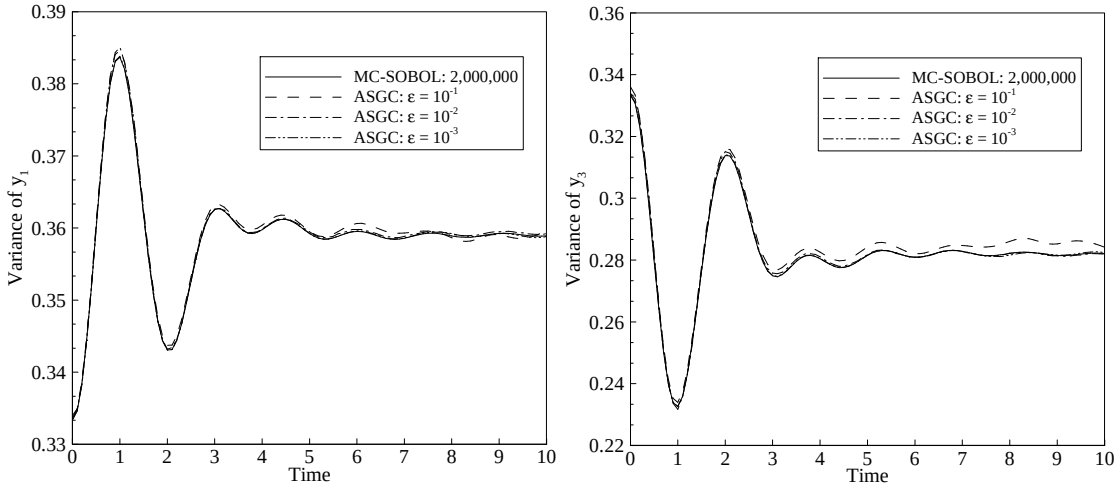


Figure 3.23: Evolution of the variance of $y_1 = y_2$ (left) and y_3 (right) for the 3D K-O problem using ASGC.

3.3.4 Stochastic elliptic problem

In this section, we compare the convergence rate of the CSGC and ASGC methods through a stochastic elliptic problem in two spatial dimensions. As shown in the previous examples, the adaptive sparse grid collocation method can accurately capture the non-smooth region of the stochastic space. Therefore, when

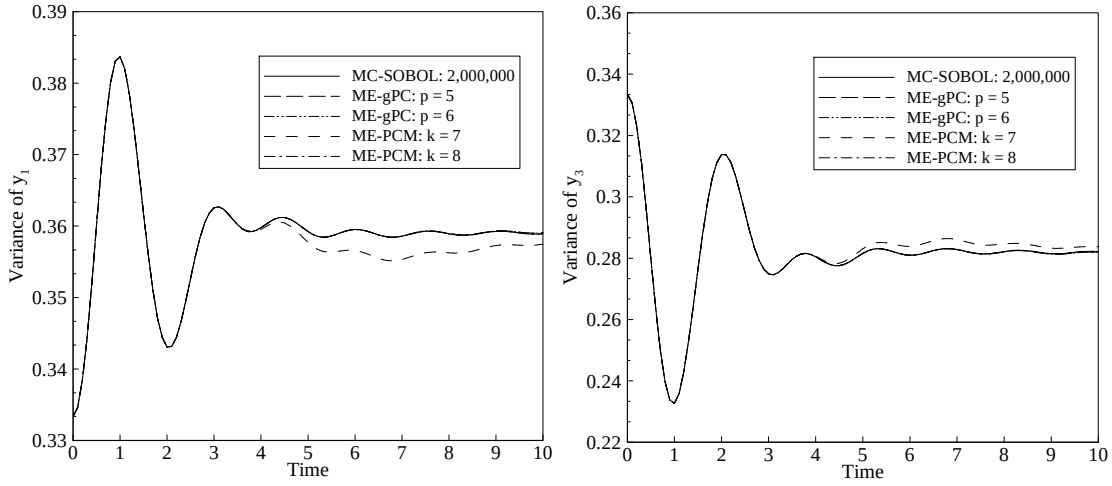


Figure 3.24: Evolution of the variance of $y_1 = y_2$ (left) and y_3 (right) for the 3D K-O problem using both ME-gPC and ME-PCM. Both results are obtained with $\theta_1 = 10^{-4}$ and $\theta_2 = 10^{-3}$.

Table 3.5: Computational results for 3D K-O problem using ASGC.

ASGC				MC	MC-SOBOL
Threshold	#Points	Error	Time	Error	Error
10^{-1}	46953	1.95×10^{-3}	0.09 hour	2.72×10^{-3}	6.23×10^{-5}
10^{-2}	210177	2.95×10^{-4}	0.85 hours	8.99×10^{-4}	6.61×10^{-5}
10^{-3}	498025	1.58×10^{-4}	3 hours	5.47×10^{-4}	2.74×10^{-5}

the non-smooth region is along a particular dimension (i.e. one dimension is more important than others), the ASGC method is expected to identify and resolve it. In this example, we demonstrate this ability of the ASGC method to detect important dimensions when each dimension weighs unequally. This is similar to the dimension-adaptive method, especially in high stochastic dimension problems.

Table 3.6: Computational results for 3D K-O problem using *h*-adaptive ME-gPC.

order p	*h*-adaptive ME-gPC		
	#Elements	Error	Time
3	5584	1.10×10^{-3}	10 hours
4	3256	5.85×10^{-4}	22.5 hours
5	2336	3.29×10^{-4}	39 hours
6	1624	2.98×10^{-4}	108 hours

Table 3.7: Computational results for 3D K-O problem using *h*-adaptive ME-PCM.

Level k	*h*-adaptive ME-PCM		
	#Points	Error	Time
6	590364	4.15×10^{-3}	0.05 hour
7	14305746	1.53×10^{-3}	0.48 hours
8	15788608	6.78×10^{-6}	0.55 hours

Here, we adopt the model problem from [33]:

$$\begin{aligned} -\nabla \cdot (a_N(\omega,\cdot)\nabla u(\omega,\cdot)) &= f_N(\omega,\cdot), \qquad \text{in } D\times\Gamma, \\ u(\omega,\cdot) &= 0, \qquad \text{on } \partial D\times\Gamma, \end{aligned} \tag{3.65}$$

with the physical domain $D = \left\{\boldsymbol{x} = (x,y) \in [0,1]^2\right\}$. To avoid introducing large errors from physical discretization, we take a deterministic smooth load $f_N(\omega,x,y) = \cos(x)\sin(y)$ with homogeneous boundary conditions. Therefore, we assume that there are no substantial errors from the physical discretization.

The deterministic problem is solved using the finite element method with 900 bilinear quadrilateral elements. Furthermore, as in [33], in order to eliminate the errors associated with a numerical Karhunen-Loève expansion solver and to keep the random diffusivity strictly positive, we construct the random diffusion coefficient $a_N(\omega, x)$ with 1D spatial dependence as

$$\log\left(a_N(\omega, x) - 0.5\right) = 1 + Y_1(\omega)\left(\frac{\sqrt{\pi}L}{2}\right)^{1/2} + \sum_{n=2}^{N} \xi_n \phi_n(x) Y_n(\omega), \tag{3.66}$$

where

$$\xi_n := \left(\sqrt{\pi}L\right)^{1/2} \exp\left(\frac{-(\lfloor \frac{n}{2} \rfloor \pi L)^2}{8}\right), \quad \text{if} \quad n > 1, \tag{3.67}$$

and

$$\phi_n(x) := \begin{cases} \sin\left(\frac{\lfloor \frac{n}{2} \rfloor \pi x}{L_p}\right), & \text{if } n \text{ even}, \\ \cos\left(\frac{\lfloor \frac{n}{2} \rfloor \pi x}{L_p}\right), & \text{if } n \text{ odd}. \end{cases} \tag{3.68}$$

$Y_n(\omega), n = 1, \ldots, N$, are independent uniformly distributed random variables in the interval $[-\sqrt{3}, \sqrt{3}]$. The parameter L_p in Eq. (3.68) can be taken as $L_p = \max\{1, 2L_c\}$ and the parameter L in Eqs. (3.66) and (3.67) is $L = L_c/L_p$. This expansion is similar to a Karhunen-Loève expansion of a 1D random field with stationary covariance

$$\operatorname{cov}\left[\log(a_N - 0.5)\right](x_1, x_2) = \exp\left(\frac{-(x_1 - x_2)^2}{L_c^2}\right). \tag{3.69}$$

Small values of the correlation length L_c correspond to slow decay in Eq. (3.66), i.e., each stochastic dimension weighs almost equally. On the other hand, large values of L_c result in fast decay rates, i.e., the first several stochastic dimensions corresponding to large eigenvalues weigh most importantly. By using the expansion Eq. (3.66), it is assumed that we are given an analytic stochastic input. Thus there is no truncation error. It is different from the discretization of a random field using the K-L expansion, where for different correlation

lengths we keep different terms accordingly. In this example, we fix N and change L_c to adjust the importance of each stochastic dimension. In this way, we want to investigate the effect of correlation length L_c on the ability of the ASGC method to detect important dimensions.

To study the convergence of the algorithm, we consider a problem where the interpolation level increases linearly. We estimate the $L^2(D)$ approximation error for the mean and the variance. Specifically, to estimate the computation error in the q-th level, we fix the dimension N and compare the results at two consecutive levels, e.g. the error for the mean is $E\left[\mathcal{A}_{q,N}(u_N) - \mathcal{A}_{q+1,N}(u_N)\right]$. Similar error is defined for the variance. The results are shown in Fig. 3.25 for different correlation lengths at $N = 11$. Each symbol denotes one interpolation level. To compare the convergence rate between the CSGC and ASGC methods, we choose the same maximum interpolation level for both methods. Then we decrease the threshold ε until the ASGC method arrives approximately at the same accuracy as the CSGC method. In [33], the authors proved for the stochastic elliptic problem that the convergence rate for the CSGC is nearly exponential. Since a linear basis can be considered as polynomial of order one, it is seen in Fig. 3.25 that the error for CSGC indeed decreases nearly exponentially which verifies the result in [33]. For small correlation lengths, the effects on the convergence rate for both CSGC and ASGC are nearly the same. On the other hand, for large correlation length, if we adopt the ASGC method, much less number of collocation points is required to achieve the same accuracy as the CSGC method. This is because more points are placed along the important dimensions which are associated with large eigenvalues. Therefore, the larger correlation lengths have positive effects on the rate of convergence, while decreasing L_c leads to a deterioration of the rate of convergence of the ASGC method due to equal weighting of all

directions. It is also noted that, for small correlation length, we need a small threshold to achieve the desired accuracy. Smaller correlation length indicates a smoother stochastic space. The surplus also decreases very fast. Therefore, for a larger threshold, the refinement stops earlier.

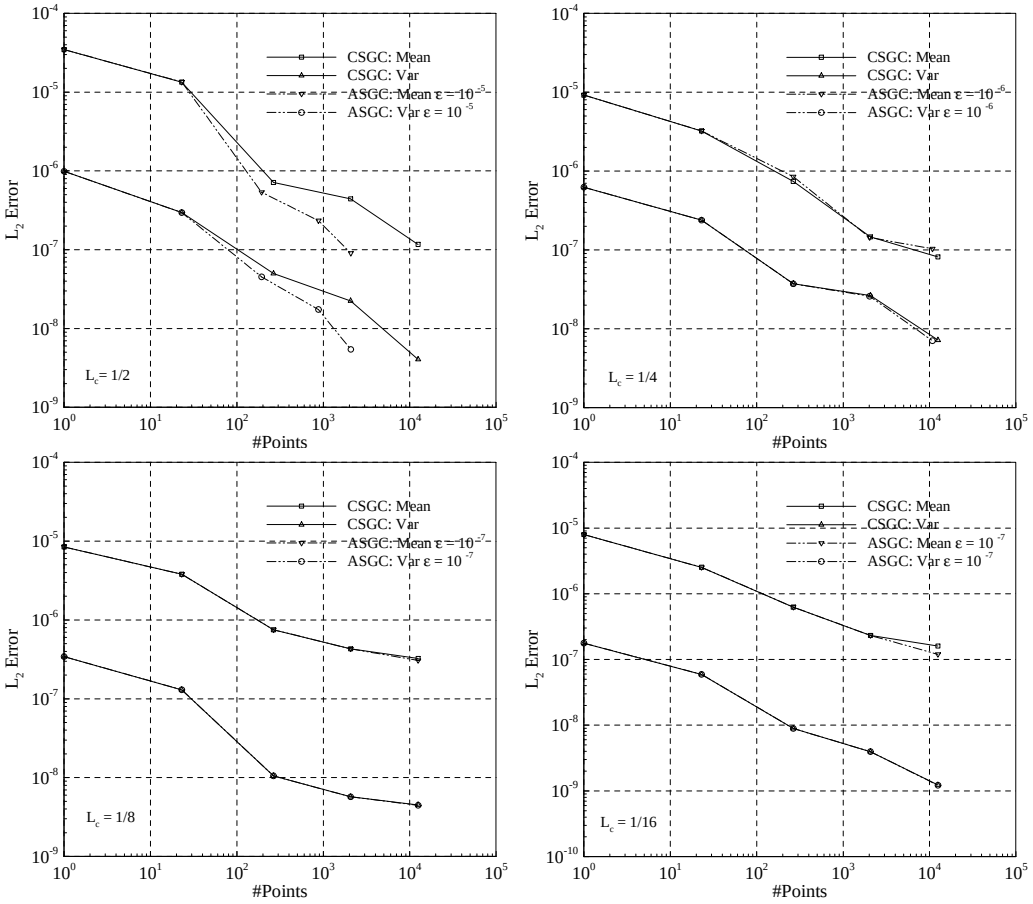


Figure 3.25: The convergence of the stochastic elliptic problem in $N = 11$ dimensions for different correlation lengths $L_c = 1/2, 1/4, 1/8, 1/16$, using both the CSGC and ASGC methods.

Next, in Fig. 3.26 we study some higher-dimensional case. Due to the rapid increase in the number of collocation points, we focus on a moderate correlation length $L_c = 0.6$ so that the ASGC is effective. From this figure, it is seen that for $N \geq 25$, the ASGC method successfully detects the important dimensions and terminates the refinement automatically. The expansion in Eq. (3.66)

is dominated by the first several terms which depend on the chosen correlation length. Since we choose a moderate large value of L_c, the important expansion terms associated with large eigenvalues are almost the same for the four cases considered in Fig. 3.26. Thus, the error level that can be achieved is nearly identical for the four cases. However, many more points are needed for increasing dimensions. Therefore, it is rather difficult to solve this problem using the CSGC method. For example, when $N = 75$ and $\varepsilon = 10^{-6}$, the refinement using the ASGC method stops at level 8 and the corresponding number of points is 276913. On the other hand, the number of points required for the CSGC method with the same interpolation level is 3.5991×10^{11}.

In order to further verify our results, we compare the mean and the variance when $N = 75$ using the AGSC method with $\varepsilon = 10^{-6}$ with the 'exact' solution given by MC simulation with 10^6 samples. The comparison is shown in Fig. 3.27 over the entire physical domain. The relative error is defined as $\frac{|E(\mathcal{A}_{q,N}(u_N)) - E(u_{MC})|}{|E(u_{MC})|}$. The maximum relative error for the mean is 9.89×10^{-4}, while the maximum relative error for the variance is 5.29×10^{-3}. Therefore, the ASGC method is indeed a very accurate method comparable to the MC method even in high stochastic dimensions. The computational time is about 0.5 hours, which is much less than the time needed by the MC method which took about 2 hours on 10 processors.

We also compare the convergence rate with the standard MC method. $L^2(D)$ error is computed by comparing the solution with a reference solution computed from 10^6 MC samples. The results are shown in Fig. 3.28. As expected, when the correlation length is large, the ASGC method is most effective. With the same number of points, the error of the ASGC is nearly one order lower

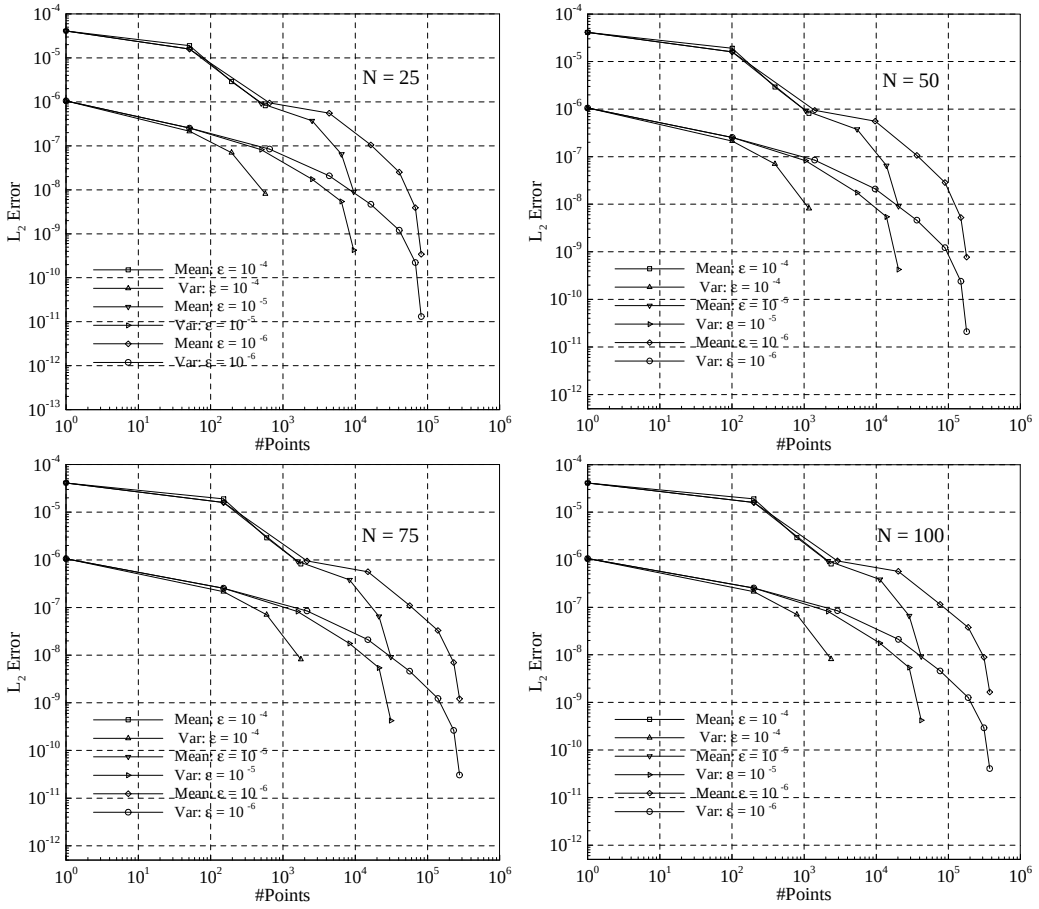


Figure 3.26: The convergence of the stochastic elliptic problem in $N = 25, 50, 75, 100$ dimensions for correlation length $L_c = 0.6$ using the ASGC method.

than that of the MC method. When the correlation length decreases, the effect of the ASGC is nearly the same as that of the CSGC. Thus, the convergence rate becomes slower as seen in the right plot of Fig. 3.28, where the ASGC error is nearly one-order larger than the MC error. This is due to the performance of the CSGC method suffering from increasing number of dimensions (unlike MC method) as a result of the weak dependence on the dimensionality in the logarithmic term of the error bound as indicated in Eq. (3.42). To achieve a desired accuracy, we have to increase the interpolation level. However, the number of collocation points will increase excessively fast as shown in Fig. 3.29, and there-

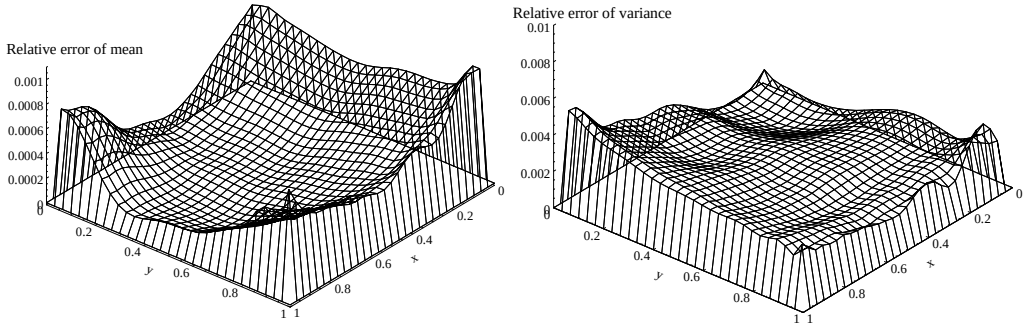


Figure 3.27: Relative error of the mean (left) and variance (right) when $N = 75$ using the ASGC method with error threshold $\varepsilon = 10^{-6}$. The 'exact' results are given by MC with 10^6 iterations.

fore the problem becomes prohibitively expensive. We also implemented the ME-PCM version of this problem that however was not appropriate for such high dimensional problem. When we start solving this problem with only one element, due to the large local variance, the method tends to split the element in every dimension, which results in 2^{25} new elements thus exceeding computer memory. Therefore, the multi-element based domain decomposition method depends much more on the dimensionality than the ASGC does.

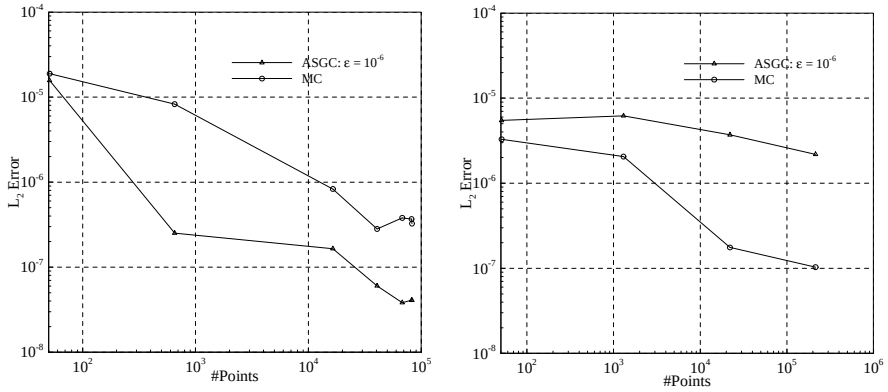


Figure 3.28: $L^2(D)$ error in the mean versus the number of collocation points (or samples of the Monte Carlo method) for $N = 25$. Left: $L_c = 0.6$, Right: $L_c = 1/16$.

In summary, the results show that, besides indicating local variance in the

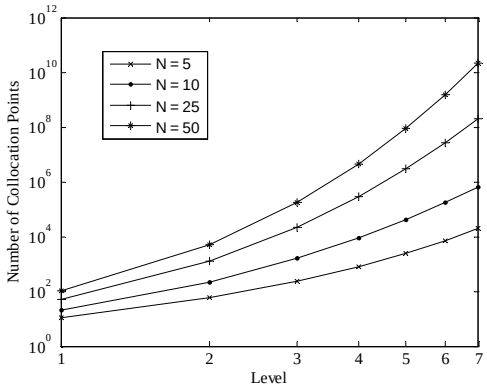


Figure 3.29: Number of collocation points versus interpolation level for the CSGC.

solution, the ASGC method is also an efficient numerical approach to detect the important dimensions in the stochastic space. This plays the same role as in the dimensional adaptive method, in particular in problems with high stochastic dimension. However, the performance of the method depends on the smoothness of the stochastic space. If the stochastic space is rather smooth (i.e. all stochastic dimensions are equally important), the MC method is still the best choice for problems with a high number of stochastic dimensions. Finally, it is also noted that the adaptation discussed here is not the same as that in the dimensional adaptive method [39, 41]. Besides adding the points along the important dimensions, we here also add all the associated neighboring points.

3.3.5 Application to Rayleigh-Bénard instability

Finally, we study the well-known stochastic Rayleigh-Bénard problem with random boundary temperature around the neighborhood of the critical point. Natural convection starts when the fluid buoyancy effect due to temperature gradients exceeds the stabilizing viscous effect. The state where these opposing

effects neutralize each other is called the critical point. Below this point, the viscous effect dominates, the fluid flow vanishes and heat transfer takes place purely by conduction. Above this point, the buoyancy effect dominates, fluid-flow is initiated and heat transfer is by conduction and convection. When the system inputs fluctuate about the critical point, the input-output relationship becomes highly non-linear and possibly discontinuous due to drastic change in the governing dynamics. This problem was previously solved using the Wiener-Haar expansion [18] and the stochastic Galerkin method [17]. Here, we study it using the ASGC method.

Deterministic problem

Consider a 2D bounded domain $D \subset \mathbb{R}^2$ with a boundary $\partial D_d \bigcup \partial D_n$. Dirichlet boundary conditions are applied on ∂D_d, while Neumann boundary conditions are applied on ∂D_n. The deterministic problem consists of finding the velocity $\boldsymbol{u}$, pressure p and temperature θ such that the following non-dimensional governing equations are satisfied:

$$\nabla \cdot \boldsymbol{u} = 0, \tag{3.70}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\nabla p + Pr \nabla^2 \boldsymbol{u} + F(\boldsymbol{u}, \theta), \tag{3.71}$$

$$\frac{\partial \theta}{\partial t} + \boldsymbol{u} \cdot \nabla \theta = \nabla^2 \theta, \tag{3.72}$$

where $F(\boldsymbol{u}, \theta)$ is the forcing function in the Navier-Stokes equations and Pr is the Prandtl number of the fluid. In the problems considered later, $F(\boldsymbol{u}, \theta)$ is the Bousinnessq approximated buoyant force term $-RaPr\theta \boldsymbol{g}$, where Ra is the thermal Rayleigh number and $\boldsymbol{g}$ is the gravity vector.

Here, we want to study the stochastic formulation of this problem. The phys-

ical domain is taken to be a closed cavity $[0,1]^2$ filled with air ($Pr = 0.7$). No-slip conditions are imposed on the boundary. The vertical walls are assumed to be adiabatic. The top wall is maintained under a deterministic cold temperature $\theta_c = -0.5$. The bottom wall temperature is assumed to be a random hot temperature θ_h. The statistics of θ_h are assumed to be such that both stable and unstable modes occur with finite probability. We set $Ra = 2500$, which is larger than the critical Rayleigh number, so that convection can be initialized by varying the hot wall temperature.

Under these conditions the problem is to find stochastic functions that describe the velocity $\boldsymbol{u} \equiv \boldsymbol{u}(\boldsymbol{x}, t, \omega) : D \times [0,T] \times \Gamma \to \mathbb{R}^2$, pressure $p \equiv p(\boldsymbol{x}, t, \omega) : D \times [0,T] \times \Gamma \to \mathbb{R}$ and temperature $\theta \equiv \theta(\boldsymbol{x}, t, \omega) : D \times [0,T] \times \Gamma \to \mathbb{R}$, such that the following equations are satisfied:

$$\nabla \cdot \boldsymbol{u}(\cdot, \omega) = 0, \quad \text{in } D \times [0,T] \times \Gamma, \tag{3.73}$$

$$\begin{aligned} \frac{\partial \boldsymbol{u}(\cdot,\omega)}{\partial t} + \boldsymbol{u}(\cdot,\omega) \cdot \nabla \boldsymbol{u}(\cdot,\omega) = -\nabla p(\cdot,\omega) + Pr \nabla^2 \boldsymbol{u}(\cdot,\omega) \\ + F(\boldsymbol{u}(\cdot,\omega), \theta(\cdot,\omega)), \quad \text{in } D \times [0,T] \times \Gamma, \end{aligned} \tag{3.74}$$

$$\frac{\partial \theta(\cdot,\omega)}{\partial t} + \boldsymbol{u}(\cdot,\omega) \cdot \nabla \theta(\cdot,\omega) = \nabla^2 \theta(\cdot,\omega), \quad \text{in } D \times [0,T] \times \Gamma. \tag{3.75}$$

The deterministic governing Eqs. (3.70)-(3.72) are solved using the second-order stabilized projection finite element method developed in [63]. The spatial domain is discretized using 40×40 bilinear quadrilateral finite elements. Prior to stochastic simulation, several deterministic computations were performed in order to find out the range where the critical point lies in. These simulations were conducted by perturbing the hot wall temperature from the purely conductive solution. We monitor the time evolution of the average kinetic energy in the

field as illustrated in Fig. 3.30. It is seen that after a short time, the kinetic energy exhibits growth (heat convection mode) or decay (heat conduction mode) until steady-state is arrived, depending on the hot wall temperature. Therefore, the critical temperature lies in the range $[0.5, 0.55]$. In addition, we also monitor the steady-state Nusselt number which denotes the rate of heat transfer:

$$Nu \equiv \frac{1}{\theta_h - \theta_c} \int_0^1 \frac{\partial \theta}{\partial y}|_{y=0} dx. \tag{3.76}$$

Clearly, in the conductive (stable) regime, $Nu = 1$. For temperature larger than the critical value, convection is initialized and heat transfer enhancement occurs so that $Nu\,(\theta_h) > 1$. The results are shown in Table 3.8. Obviously, when the hot wall temperature is larger than 0.55, heat convection begins. This again verifies the critical hot wall temperature lies in the range $[0.5, 0.55]$. However, the exact critical value is not known to us. So, we now try to capture this unstable equilibrium using the ASGC method.

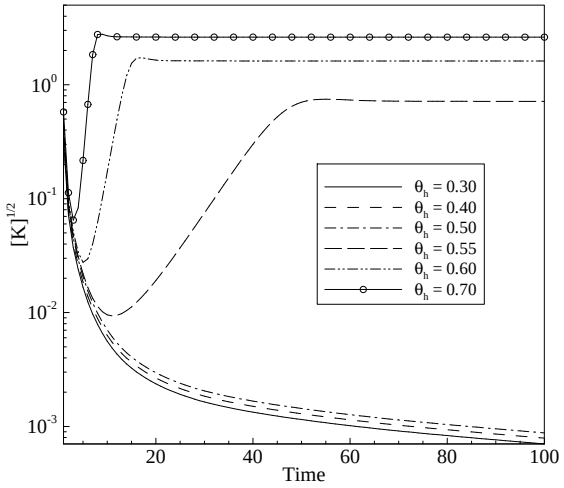


Figure 3.30: Evolution of the average kinetic energy for different hot wall temperatures.

Table 3.8: Steady-state Nusselt number for different hot wall temperatures

θ_h	0.30	0.40	0.50	0.55	0.60	0.70
Nu	1.00000	1.00000	1.00000	1.01496	1.07357	1.17744

Adaptive sparse grid collocation scheme

In this section, we assume the following stochastic boundary condition for the hot wall temperature:

$$\theta_h = 0.4 + 0.3Y, \tag{3.77}$$

where Y is a uniform random variable in the interval $[0, 1]$. Following the discussion above, both a stable and an unstable flow occur for this range of θ_h.

As discussed above, for the conductive regime, $Nu = 1$ and $Nu > 1$ when heat convection occurs. Thus, the difference $\delta Nu \equiv Nu\,(\theta_h) - 1$ provides a measure of the heat transfer enhancement. This result is provided in the left of Fig. 3.31. The result is reconstructed from the hierarchical surplus of the solution. It is noted that, the critical value is about 0.541, since below this value $\delta Nu = 0.0$ and an essentially linear increase of δNu with θ_h is observed beyond this value. This can be further verified from the corresponding adaptive sparse grid in the same figure shown on the right. For the first 6 interpolation levels, it is seen that less points are placed on the left hand side of point 0.55, which is expected due to the pure conduction mode in this region. In this region of the random space, the solution is smooth and the magnitude of the hierarchical surpluses decrease very quickly. On the other hand, almost all points on the right of point 0.55 are refined since heat convection occurs in this regime. After level 6, we can see

that only the points around point 0.541 refine until the refinement stops at interpolation level 12, which suggests larger hierarchical surpluses occur there and that the solution is not very smooth in the neighborhood. Therefore, the critical value for the hot wall temperature is indeed about 0.541, which is consistent with our discussion in Section 3.3.5. In addition, we also plot in Fig. 3.32 the solution of the state variables at point $(0.1, 0.5)$ as a function of hot-wall temperature. At this point the solution exhibits a higher variance in temperature (see also Fig. 3.36 at the end of this section). Again, the same critical point is predicted from Fig. 3.32. Specifically, the velocity vanishes below the point 0.541 and increases with θ_h beyond this points where heat convection occurs. The temperature increases linearly with the hot wall temperature below the critical point which is a typical characteristic of heat conduction. On the other hand, the temperature grows non-linearly beyond this point.

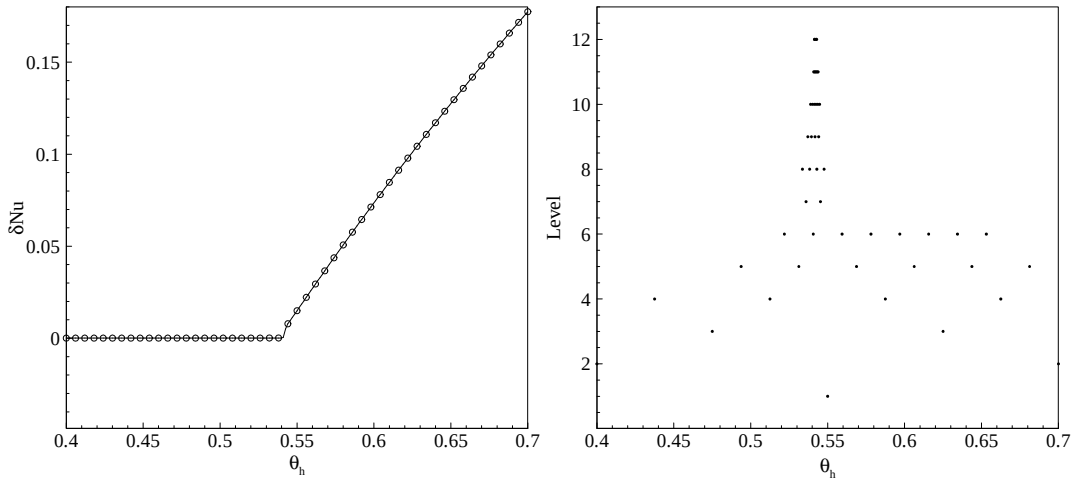


Figure 3.31: Steady-state δNu (left) versus hot-wall temperature using ASGC and the corresponding adaptive sparse grid with threshold $\varepsilon = 0.01$ (right).

To further verify the results, we sample a uniform random variable for hot wall temperature from the conduction regime and reconstruct the solution from

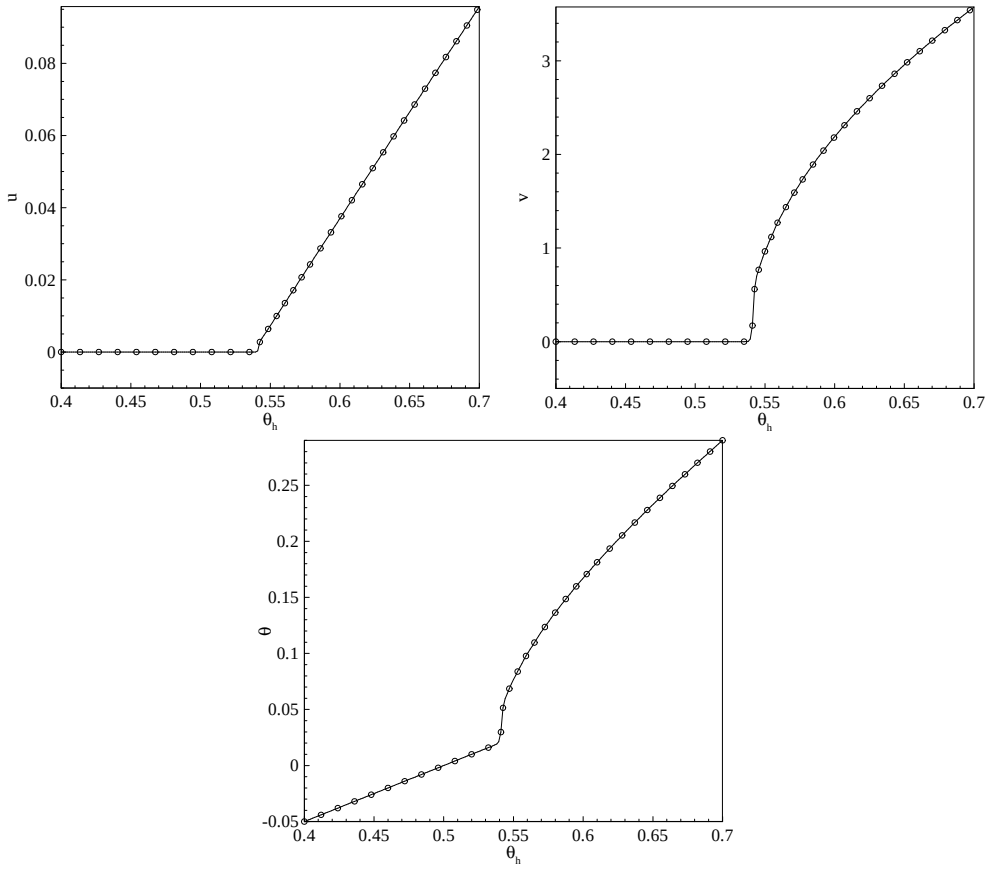


Figure 3.32: Solution of the state variables versus hot-wall temperature at point $(0.1, 0.5)$. Top left: u velocity, Top right: v velocity, Bottom: temperature.

the hierarchical surpluses. At the same time, we run a deterministic problem using the same realization of the random variable. The comparison of the results is shown in Fig. 3.33. The velocities are zero and thus they are not shown. We can see that the contour distribution of the temperature is characterized by parallel horizontal lines which is a typical distribution for heat conduction. The prediction from the collocation method is the same as the deterministic solution. We repeat this process by sampling a hot wall temperature from the convection regime. The results are shown in Fig. 3.34. Again, the results compare very well and correctly predict the convective behavior. It is interesting to note that

the difference of the maximum value of the velocity between the predicted and deterministic solutions is within the order $O(10^{-3})$, which is consistent with the error threshold 10^{-2}.

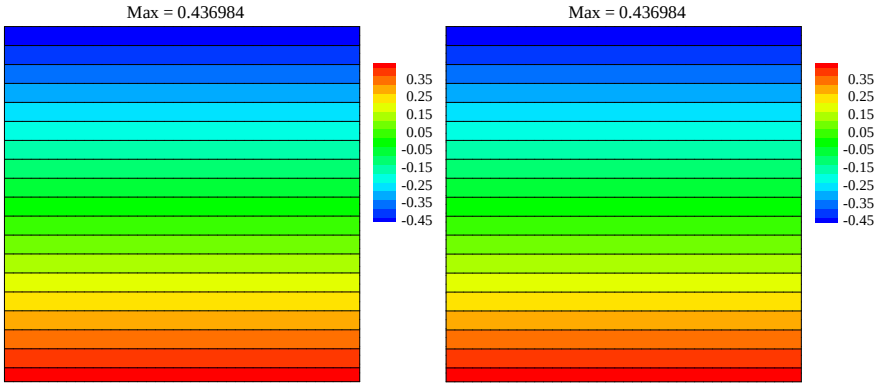


Figure 3.33: Prediction of the temperature when $\theta_h = 0.436984$ using ASGC (left) and the solution of the deterministic problem using the same θ_h.

Finally, we provide the mean and variance in Figs. 3.35 and 3.36, respectively. We also include for comparison the results obtained with the MC-SOBOL sequence with 10^4 iterations. The results compare very well, which again verifies the accuracy of the present method. The number of collocation points, i.e. the number of runs of the deterministic simulator needed for the adaptive sparse grid is 49, while the computation time is about 1.5 hours. However, the computation time for MC-SOBOL is 15 hours. Therefore, the time needed by the MC method is much more than that for the ASGC.

In summary, the ASGC method can successfully capture the unstable equilibrium in natural convection. In addition, it can also predict quite accurately the critical point.

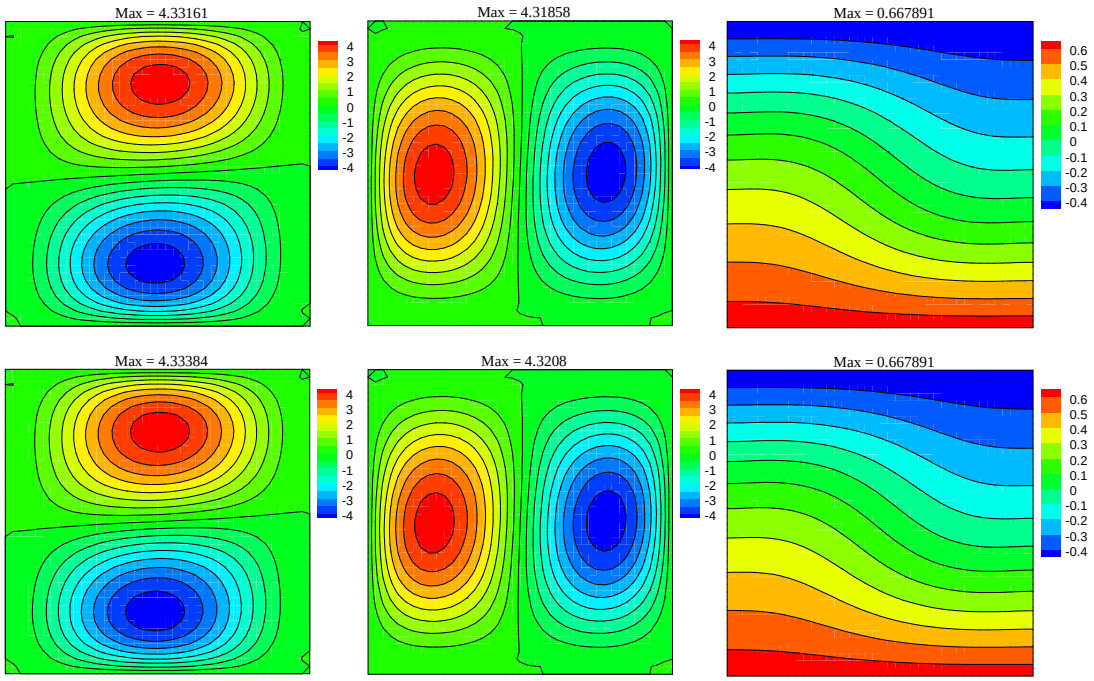


Figure 3.34: Prediction of the u velocity (left column), v velocity (middle column) and temperature (right column) when $\theta_h = 0.667891$ using ASGC (top row) and the solution of the deterministic problem using the same θ_h (bottom row).

3.4 Conclusions

In this chapter, we developed an adaptive hierarchical sparse grid collocation method based on the error control of local hierarchical surpluses. By utilizing multi-linear hierarchical basis functions of local support, this method can resolve successfully discontinuities in the stochastic space. Through numerical examples, we demonstrated that in the presence of discontinuity in the stochastic space, this approach leads to significant reduction in the number of points required to achieve the same level of accuracy as the CSGC method. Unlike the dimension adaptive collocation method developed earlier, the current method refines the sparse grids locally by working directly in the hierarchical basis. Thus, besides the detection of important dimensions as dimension-adaptive methods

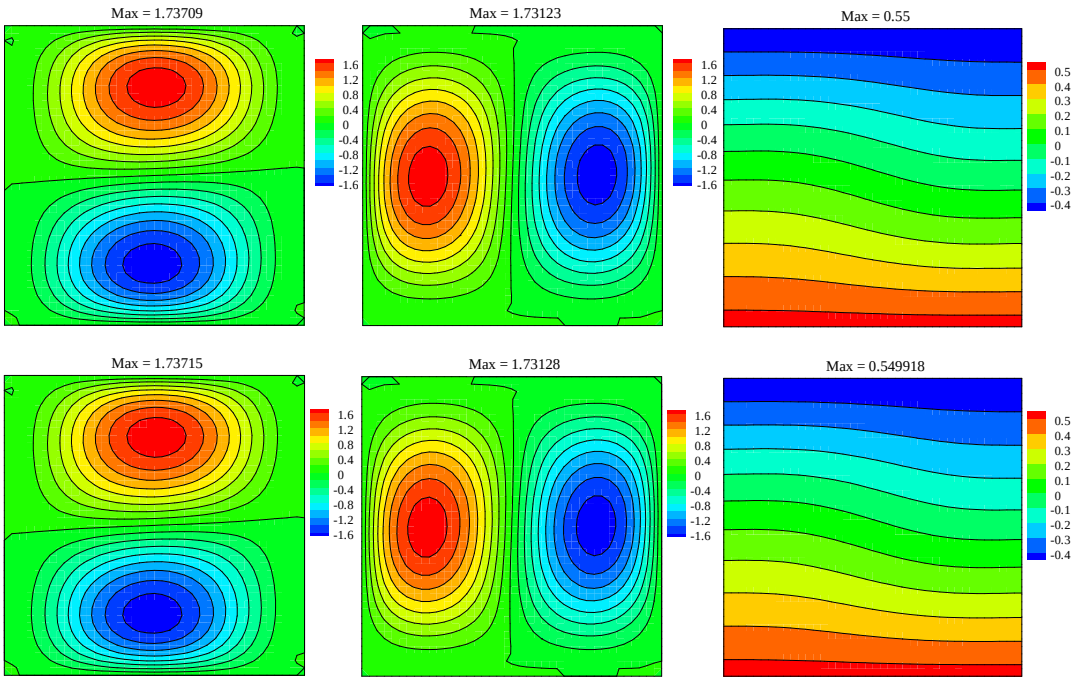


Figure 3.35: Comparison of the mean of the u velocity (left column), v velocity (middle column) and temperature (right column) using ASGC (top row) and MC-SOBOL method with 10^4 iterations (bottom row).

can do, additional singularity and local behavior of the solution can also be revealed. By applying this method to the Rayleigh-Bénard instability, it is shown that the adaptive sparse grid can accurately predict the critical point.

We provided extensive comparisons with the ME-gPC and ME-PCM methods. Due to the decoupled nature of the stochastic collocation method, it is shown that the computational cost of the ASGC and ME-PCM methods is much less than that of the ME-gPC method at least for the low-dimensional problems considered. The ME-PCM is even faster than the ASGC in terms of computational time for low-dimensional problems. However, the ASGC requires much less number of collocation points than the ME-PCM to achieve the same accuracy. Furthermore, the multi-element based method is not suitable for high-

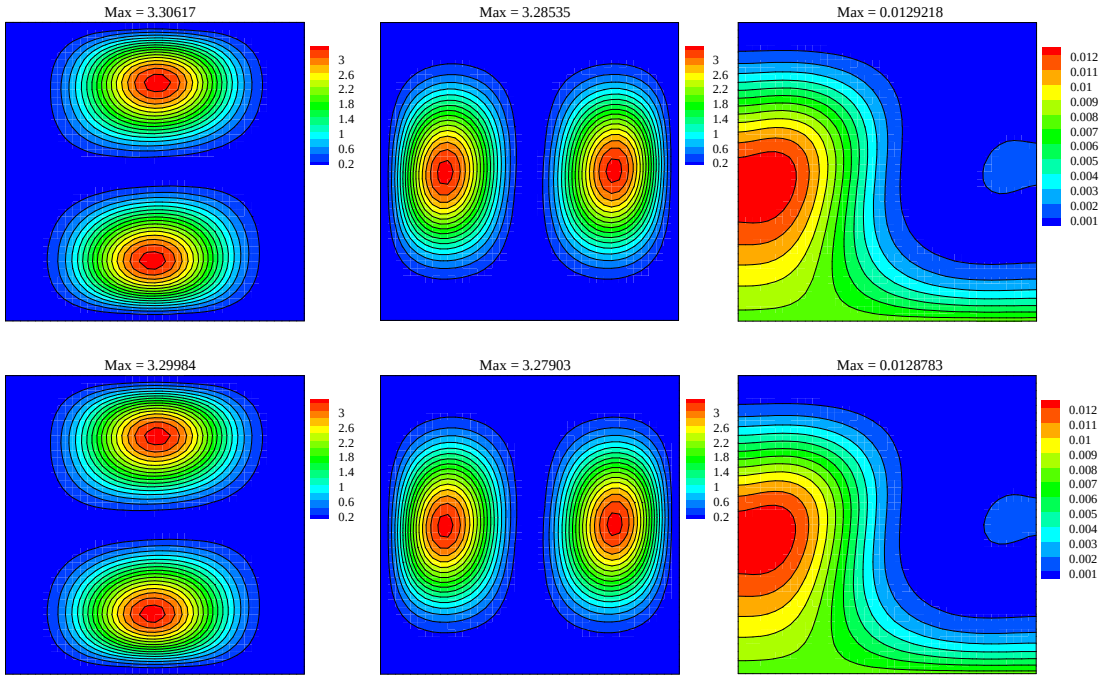


Figure 3.36: Comparison of the variance of the u velocity (left column), v velocity (middle column) and temperature (right column) using ASGC (top row) and MC-SOBOL method with 10^4 iterations (bottom row).

dimensional problems due to its $O(2^N)$ tree-like scaling of the standard $h-$type adaptive refinement.

Solutions are also compared with MC results. For the 3D K-O problem, due to the rather strong discontinuity, the convergence rate of the AGSC is not optimal compared with MC. However, in general, the ASGC can achieve a desired accuracy at a cost much lower than that of the MC method provided that the ASGC is effective in detecting the regularity in the stochastic space. On the other hand, as shown in the stochastic elliptic problem, if each dimension weighs almost equally for a high-dimensional problem, the ASGC is not the best choice. This is because although the Smolyak algorithm depends less on dimensionality than the gPC method, it still suffers with increasing number of dimensions due

to the weak dependence on the dimensions in the logarithmic term of the error bound. A new method is developed to address this issue in the coming chapter.

It is also worth noting that the ASGC method not only gives us the solution statistics the same as using MC, but also calculates an approximate functional representation of the solution in the stochastic space. Therefore, in the context of function approximation, MC is not applicable to solve stochastic problems. Currently, the ME-PCM is based on quadrature rule and only gives us mean and variance. Although it can provide response surface of the solution by projecting the collocation solution onto the gPC basis, it is not as straightforward as the ASGC since it needs to search in the multi-element grid.

CHAPTER 4

SOLVING HIGH-DIMENSIONAL SPDES - AN ADAPTIVE HIGH-DIMENSIONAL STOCHASTIC MODEL REPRESENTATION TECHNIQUE

As is shown in Section 3.3.4, when the importance of each dimension weighs equally, ASGC cannot solve the problem accurately even with a moderate (21) stochastic dimensionality. In this case, the effect of ASGC is nearly the same as of the CSGC and thus the convergence rate deteriorates. In this chapter, the high dimensional model representation technique is utilized to address the high dimensional problem. In addition, an adaptive version of HDMR is also developed for the first time to automatically detect the important dimensions and construct higher-order terms using only the important dimensions. This chapter closely follows the work in [66].

4.1 High dimensional model representations (HDMR)

In this section, the basic concepts of HDMR are introduced following closely the notation in [43, 44, 48, 57]. For a detailed description of the theory applied to deterministic systems, the interesting reader may refer to [44].

Let $f(\mathbf{Y}) : \mathbb{R}^N \rightarrow \mathbb{R}$ be a real valued smooth multivariate stochastic function. Here, it is noted that $f(\mathbf{Y})$ may be also a function of physical coordinate, $f(\mathbf{Y}, \boldsymbol{x})$. From now on, we will omit $\boldsymbol{x}$ to simplify the notation. HDMR represents $f(\mathbf{Y})$ as a finite hierarchical correlated function expansion in terms of the input variables

as [43, 44, 48]

$$f(\mathbf{Y}) = f_0 + \sum_{s=1}^{N} \sum_{i_1<\cdots<i_s}^{N} f_{i_1\cdots i_s}(Y_{i_1}, \ldots, Y_{i_s}), \tag{4.1}$$

where the interior sum is over all sets of s integers $i_1, \ldots, i_s$, that satisfy $1 \le i_1 < i_s \le N$. This relation means that

$$\begin{aligned} f(\mathbf{Y}) &= f_0 + \sum_{i=1}^{N} f_i(Y_i) + \sum_{1\le i_1<i_2\le N} f_{i_1 i_2}(Y_{i_1}, Y_{i_2}) + \cdots \\ &+ \sum_{1\le i_1<\cdots<i_s\le N} f_{i_1\cdots i_s}(Y_{i_1}, \ldots, Y_{i_s}) + \cdots + f_{12\cdots N}(Y_1, \ldots, Y_N). \end{aligned} \tag{4.2}$$

Here, f_0 is the zeroth-order component function which is a constant denoting the mean effect. The first-order component function $f_i(Y_i)$ is a univariate function which represents individual contributions to the output $f(\mathbf{Y})$. It is noted that $f_i(Y_i)$ is general a nonlinear function. The second-order component function $f_{i_1 i_2}(Y_{i_1}, Y_{i_2})$ is a bivariate function which describes the interactive effects of variables Y_{i_1} and Y_{i_2} acting together upon the output $f(\mathbf{Y})$. The higher-order terms reflect the cooperative effects of increasing number of input variables acting together to impact f. The s-th order component function $f_{i_1\cdots i_s}(Y_{i_1}, \ldots, Y_{i_s})$ is a s-dimensional function. The last term $f_{12\cdots N}(Y_1, \ldots, Y_N)$ gives any residual dependence of all input variables cooperatively locked together to affect the output $f(\mathbf{Y})$ [43, 44]. Once all the component functions are suitably determined, then the HDMR can be used as a computationally efficient reduced-order model for evaluating the output. This is the same idea as the stochastic collocation method where we also obtain an approximate representation of $f(\mathbf{Y})$.

Remark 1. The HDMR expansions are based on exploiting the correlation effects of the input variables, which are naturally created by the stochastic input-output mapping, i.e. the SPDE. It is noted that the term "correlation" employed here does not indicate the correlation effects between the input random variables as employed in statistics since in general the random input variables are

independent and thus uncorrelated. Instead, it indicates the impact of these input variables upon the system output when acting together.

Equation (4.1) is often written in a more compact notation [57] as follows:

$$f(\mathbf{Y}) \;=\; \sum_{\mathbf{u}\subseteq\mathcal{D}} f_{\mathbf{u}}(\mathbf{Y}_{\mathbf{u}}), \tag{4.3}$$

for a given set $\mathbf{u} \subseteq \mathcal{D}$, where $\mathcal{D} := \{1, \ldots, N\}$ denotes the set of coordinate indices and $f_{\emptyset}(\mathbf{Y}_{\emptyset}) = f_0$. Here, $\mathbf{Y}_{\mathbf{u}}$ denotes the $|\mathbf{u}|$-dimensional vector containing those components of $\mathbf{Y}$ whose indices belong to the set $\mathbf{u}$, where $|\mathbf{u}|$ is the cardinality of the corresponding set $\mathbf{u}$, i.e. $\mathbf{Y}_{\mathbf{u}} = (Y_i)_{i\in\mathbf{u}}$. For example, if $\mathbf{u} = \{1, 3, 5\}$, then $|\mathbf{u}| = 3$ and $f_{\mathbf{u}}(\mathbf{Y}_{\mathbf{u}})$ implies $f_{135}(Y_1, Y_3, Y_5)$.

The component functions $f_{\mathbf{u}}(\mathbf{Y}_{\mathbf{u}})$ can be derived by minimizing the error functional [43, 44]:

$$\int_{\Gamma}\left[f(\mathbf{Y}) - \sum_{\mathbf{u}\subseteq\{0,\ldots,s\}} f_{\mathbf{u}}(\mathbf{Y}_{\mathbf{u}})\right]^2 d\mu(\mathbf{Y}), \tag{4.4}$$

where $0 \leqslant s \leqslant N$.

The measure $d\mu$ determines the particular form of the error functional and of the component functions. The measure μ induces the projection operator $P_{\mathbf{u}} : \Gamma^N \to \Gamma^{|\mathbf{u}|}$ by [44]

$$P_{\mathbf{u}} f(\mathbf{Y}_{\mathbf{u}}) := \int_{\Gamma^{N-|\mathbf{u}|}} f(\mathbf{Y}) d\mu_{\mathcal{D}\setminus\mathbf{u}}(\mathbf{Y}), \tag{4.5}$$

where $d\mu_{\mathcal{D}\setminus\mathbf{u}}(\mathbf{Y}) := \prod_{i\in\mathcal{D}, i\notin\mathbf{u}} d\mu_i(Y_i)$.

Therefore, the 2^N terms $f_{\mathbf{u}}$ can be recursively defined by [43]

$$f_{\mathbf{u}}(\mathbf{Y}_{\mathbf{u}}) := P_{\mathbf{u}} f(\mathbf{Y}_{\mathbf{u}}) - \sum_{\mathbf{v}\subset\mathbf{u}} f_{\mathbf{v}}(\mathbf{Y}_{\mathbf{v}}), \tag{4.6}$$

and can also be given explicitly by [67]

$$f_{\mathbf{u}}(\mathbf{Y}_{\mathbf{u}}) := \sum_{\mathbf{v}\subseteq\mathbf{u}} (-1)^{|\mathbf{u}|-|\mathbf{v}|} P_{\mathbf{v}} f(\mathbf{Y}_{\mathbf{v}}). \tag{4.7}$$

This formulation is particularly useful to combine with ASGC as will be discussed later. The component functions are orthogonal with respect to the inner product induced by the measure μ,

$$\int_{\Gamma^N} f_{\mathbf{u}}(\mathbf{Y}_{\mathbf{u}}) f_{\mathbf{v}}(\mathbf{Y}_{\mathbf{v}}) d\mu(\mathbf{Y}) = 0, \quad \text{for } \mathbf{u} \neq \mathbf{v}, \tag{4.8}$$

and thus the resulting decomposition Eq. (4.3) is unique for a fixed measure μ. In the next sections, we will present two particularly useful decompositions.

4.1.1 ANOVA-HDMR

In this case, the measure μ is taken as the ordinary Lebesgue measure $d\mu(\mathbf{Y}) = d(\mathbf{Y}) = \prod_{i=1}^{N} dY_i$. With this choice, the actions of the projection operators in the ANOVA-HDMR are given by

$$P_{\mathbf{u}} f(\mathbf{Y}_{\mathbf{u}}) := \int_{\Gamma^{N-|\mathbf{u}|}} f(\mathbf{Y}) d\mathbf{Y}_{\mathcal{D}\setminus\mathbf{u}}. \tag{4.9}$$

More specifically, the first few terms are

$$\begin{aligned} f_0 &= \int_{\Gamma^N} f(\mathbf{Y}) d\mathbf{Y}, \quad f_i(Y_i) = \int_{\Gamma^{N-1}} f(\mathbf{Y}) \prod_{j\neq i} dY_j - f_0 \\ f_{ij}(Y_i, Y_j) &= \int_{\Gamma^{N-2}} f(\mathbf{Y}) \prod_{k\neq i,j} dY_k - f_i(Y_i) - f_j(Y_j) - f_0, \quad \ldots \end{aligned} \tag{4.10}$$

This decomposition is the same as the well-known analysis of variance (ANOVA) decomposition used in statistics. A significant drawback of ANOVA-HDMR is the need to compute the high-dimensional integrals. Even the zeroth-order component function requires a full-dimensional integration in the space. To circumvent this difficulty, a computationally more efficient CUT-HDMR expansion will be introduced in the following section for the stochastic model representation which is the focus of this chapter.

4.1.2 CUT-HDMR

In this work, the CUT-HDMR is adopted to construct the response surface of the stochastic solution. With this method, the measure μ is chosen as the Dirac measure located at a reference point $\bar{\mathbf{Y}} = \left(\bar{Y}_1, \bar{Y}_2, \ldots, \bar{Y}_N\right)$, i.e. $d\mu(\mathbf{Y}) = \prod_{i=1}^{N} \delta(Y_i - \bar{Y}_i) dY_i$. With this choice, the projections Eq. (4.5) become

$$P_{\mathbf{u}} f(\mathbf{Y_u}) := f(\mathbf{Y})|_{\mathbf{Y}=\bar{\mathbf{Y}}\setminus\mathbf{Y_u}}, \tag{4.11}$$

where the notation $\mathbf{Y} = \bar{\mathbf{Y}} \setminus \mathbf{Y_u}$ means that the components of $\mathbf{Y}$ other than those indices that belong to the set $\mathbf{u}$ are set equal to those of the reference point. Equation (4.11) defines a $|\mathbf{u}|$-dimensional function where the unknown variables are those dimensions whose indices belong to $\mathbf{u}$. The component functions of CUT-HDMR are explicitly given as follows [43]:

$$\begin{aligned} f_0 &= f(\bar{\mathbf{Y}}), \;\; f_i(Y_i) = f(\mathbf{Y})|_{\mathbf{Y}=\bar{\mathbf{Y}}\setminus Y_i} - f_0 \\ f_{ij}(Y_i, Y_j) &= f(\mathbf{Y})|_{\mathbf{Y}=\bar{\mathbf{Y}}\setminus(Y_i,Y_j)} - f_i(Y_i) - f_j(Y_j) - f_0, \quad \ldots \end{aligned} \tag{4.12}$$

It is argued in [43, 44] that quite often in typical deterministic physical systems the correlation effects of the higher-order terms amongst the input variables for their action upon the output are weak. Tests on several examples from [43, 44] indicate that only the low-order correlations have a significant impact on the output and thus the few lower-order terms (usually up to third-order) are often sufficient to represent the model to a desired accuracy. The extent of high-order variable cooperativity depends on the choice of input variables [44]. However, the exact factors which determine the correlation effect in stochastic space are unclear and this will be one of the focus points of this work.

It is also interesting to note that the CUT-HDMR can be derived from a Tay-

lor expansion at the reference point [52]:

$$
\begin{aligned}
f(\mathbf{Y}) &= f(\bar{\mathbf{Y}}) + \sum_{j=1}^{\infty} \frac{1}{j!} \sum_{i=1}^{N} \frac{\partial^j f}{\partial Y_i}(\bar{\mathbf{Y}})(Y_i - \bar{Y}_i)^j \\
&+ \sum_{j_1, j_2 > 0}^{\infty} \frac{1}{j_1! j_2!} \sum_{i_1 < i_2} \frac{\partial^{j_1 + j_2} f}{\partial Y_{i_1}^{j_1} \partial Y_{i_2}^{j_2}}(\bar{\mathbf{Y}})(Y_{i_1} - \bar{Y}_{i_1})^{j_1}(Y_{i_2} - \bar{Y}_{i_2})^{j_2} + \cdots
\end{aligned} \tag{4.13}
$$

The infinite number of terms in the Taylor series are partitioned into finite number of groups with each group corresponding to one CUT-HDMR component function. For example, the first-order component function $f_i(Y_i)$ is the sum of all the Taylor series terms which contain and only contain variable Y_i and so on [47]. Any truncated HDMR expansion should provide higher-order accuracy than a truncated Taylor series of the same order.

Choice of the reference point

The convergence property of HDMR is rather sensitive to the choice of the reference point. It is argued in [43] that the reference point can be chosen arbitrarily provided the HDMR expansion is converged. Later, it is pointed out in [48, 68] that in some cases the choice of the reference point is important in order to obtain a good approximation for a fixed-order HDMR and a careless choice may lead to an unacceptable approximation error.

We would like to choose a suitable point such that the error of a fixed-order HDMR approximation is as small as possible and the HDMR expansion order is as low as possible. The authors in [48, 68] proved that a good reference point should satisfy:

$$
\min_{\mathbf{Y} \in \Gamma} \left| f(\bar{\mathbf{Y}}) - \mathbb{E}\left[f(\mathbf{Y}) \right] \right|. \tag{4.14}
$$

However, the mean of the output is not known *a priori*. To this end, they proposed to sample a moderate number of random inputs and compute the mean

of the sample outputs. Then the reference point is chosen as the one among the samples whose output is the closest to the above mean value [48, 68]. It is obvious that this method will be quite expensive if it is used in our problem.

According to the Taylor expansion Eq. (4.13), if we choose the reference point as the mean of the input random vector, then the expectation of the output is close to the function value at the mean vector since the coefficients associated with higher-order terms in the Taylor series are usually much smaller than lower-order terms. It has been shown that in many stochastic mechanics problems [47], a second-order HDMR expansion usually leads to a satisfactory approximation within a desired accuracy if the reference point is chosen as the mean vector. Therefore, unless otherwise stated, we always choose the mean of the random input vector as the reference point in this work.

4.2 Integrating HDMR and ASGC

Although ASGC depends less on dimensionality than the gPC method, it still suffers with increasing number of dimensions [61]. Therefore, for problems with high stochastic dimensionality, to obtain accurate results one needs to use a higher-interpolation level. However, the number of needed points will grow quickly as shown in Fig. 3.29. Integrating HDMR and ASGC is a way to address and overcome this difficulty.

First, we want to discuss the choice of error indictor in ASGC. Previously, the magnitude of the hierarchical surplus is chosen as the error indicator in Section 3.2. However, this error indicator is too sharp and may result in a non-terminating algorithm. We will need to define a new error indicator. The new

error indicator is defined as follows:

$$\gamma_{\mathbf{j}}^{\mathbf{i}} = \frac{\left\| w_{\mathbf{j}}^{\mathbf{i}}(\boldsymbol{x}) \cdot I_{\mathbf{j}}^{\mathbf{i}} \right\|_{L_2(D)}}{\left\| \mathbb{E}_{\|\mathbf{i}\|-N-1}[f] \right\|_{L_2(D)}}. \tag{4.15}$$

Here, the L_2 norm is defined in the spatial domain. This error indicator measures the contribution of each term in Eq. (3.46) to the integration value (mean) relative to the overall integration value computed from the previous interpolation level. In addition to the surpluses, it also incorporates information from the basis functions. This makes the error $\gamma_{\mathbf{j}}^{\mathbf{i}}$ decrease to a sufficiently small value for a large interpolation level. Therefore, for a reasonable error threshold, this error indicator guarantees that the refinement would stop at a certain interpolation level.

Within the framework of CUT-HDMR, let us rewrite Eqs. (4.3) and (4.7) as

$$f(\mathbf{Y}) = \sum_{\mathbf{u}\subseteq\mathcal{D}} f_{\mathbf{u}}(\mathbf{Y}_{\mathbf{u}}) = \sum_{\mathbf{u}\subseteq\mathcal{D}} \sum_{\mathbf{v}\subseteq\mathbf{u}} (-1)^{|\mathbf{u}|-|\mathbf{v}|} f(\mathbf{Y}_{\mathbf{v}})_{\mathbf{Y}=\bar{\mathbf{Y}}\setminus\mathbf{Y}_{\mathbf{v}}}, \tag{4.16}$$

where we define $f(\mathbf{Y}_{\emptyset}) = f(\bar{\mathbf{Y}})$. Therefore, the N-dimensional stochastic problem is transformed to several lower-order $|\mathbf{v}|$-dimensional problems $f(\mathbf{Y}_{\mathbf{v}})_{\mathbf{Y}=\bar{\mathbf{Y}}\setminus\mathbf{Y}_{\mathbf{v}}}$ which can be easily solved by the ASGC as introduced in the last section:

$$f(\mathbf{Y}) = \sum_{\mathbf{u}\subseteq\mathcal{D}} \sum_{\mathbf{v}\subseteq\mathbf{u}} (-1)^{|\mathbf{u}|-|\mathbf{v}|} \sum_{\|\mathbf{i}\|\leqslant N+q} \sum_{\mathbf{j}} w_{\mathbf{v}}^{\mathbf{ij}}(\boldsymbol{x}) \cdot a_{\mathbf{j}}^{\mathbf{i}}(\mathbf{Y}_{\mathbf{v}}), \tag{4.17}$$

where $\|\mathbf{i}\| = i_1 + \cdots + i_{|\mathbf{v}|}$, $w_{\mathbf{v}}^{\mathbf{ij}}(\boldsymbol{x})$ are the hierarchical surpluses for different sub-problems indexed by $\mathbf{v}$ and $a_{\mathbf{j}}^{\mathbf{i}}(\mathbf{Y}_{\mathbf{v}})$ is only a function of the coordinates which belong to the set $\mathbf{v}$. It is noted that the interpolation level q may be different for each sub-problem according to their regularity along the particular dimensions which is controlled by the error threshold ε. In this work, ε is the same for all sub-problems.

Interpolation is done quickly here (with no need to search any tables)

through simple weighted sum of the basis functions and the corresponding hierarchical surpluses. In addition, it is also easy to extract statistics as introduced in Section 3.1.5 by integrating directly the interpolating basis functions. Let us denote

$$J_{\mathbf{u}} = \sum_{\mathbf{v}\subseteq\mathbf{u}} (-1)^{|\mathbf{u}|-|\mathbf{v}|} \sum_{\|\mathbf{i}\|\leqslant N+q} \sum_{\mathbf{j}} w_{\mathbf{v}}^{\mathbf{ij}}(\boldsymbol{x}) \cdot I_{\mathbf{j}}^{\mathbf{i}}, \tag{4.18}$$

as the mean of the component function $f_{\mathbf{u}}$. Then the mean of the HDMR expansion is simply $\mathbb{E}[f(\mathbf{Y})] = \sum_{\mathbf{u}\subseteq\mathcal{D}} J_{\mathbf{u}}$. To obtain the variance of the solution, we can similarly construct an approximation for u^2 and use the formula $\mathrm{Var}[\mathrm{u}(\boldsymbol{x})] = \mathbb{E}\left[\mathrm{u}^2(\boldsymbol{x})\right] - (\mathbb{E}[\mathrm{u}(\boldsymbol{x})])^2$.

Remark 1. It is also possible to use the Smolyak quadrature rule directly to integrate the CUT-HDMR in order to obtain the mean and the variance [56, 57]. However, the method introduced here is much better since it provides a function approximation to the output that can be used as a stochastic reduced-order model [69] with local adaptivity built in its representation.

4.3 The effective dimension of a multivariate stochastic function

Related to HDMR expansions is the concept of the effective dimension of a multivariate function [57, 67]. In [57], the authors have discussed it in the case of integration. Here, we extend this concept to a multivariate stochastic function. Let $\hat{f} := \sum_{\mathbf{u}\subseteq\mathcal{D}} |J_{\mathbf{u}}|$ be the sum of all contributions to the mean value, where $|J_{\mathbf{u}}| = |\int f_{\mathbf{u}} dY_{\mathbf{u}}| \leqslant \|f_{\mathbf{u}}\|_{L_1}$ [57]. Then, for the proportion $\alpha \in (0, 1]$, the *truncation*

dimension is defined as the smallest integer N_t, such that

$$\sum_{\mathbf{u}\subseteq\{1,\ldots,N_t\}} |J_{\mathbf{u}}| \geqslant \alpha\hat{f}, \tag{4.19}$$

whereas, the *superposition dimension* is defined as the smallest integer N_s, such that

$$\sum_{|\mathbf{u}|\leqslant N_s} |J_{\mathbf{u}}| \geqslant \alpha\hat{f}. \tag{4.20}$$

The superposition dimension is also called the *order* of the HDMR expansion. The effective dimensions in the case of interpolation cannot be defined in a unique manner and their definition is part of the algorithmic approach used.

With the definition of effective dimensions, we can thus truncate the expansion in Eq. (4.3). In other words, we take only a subset $\mathcal{S}$ of all indices $\mathbf{u} \subseteq \mathcal{D}$. Here, we assume that the set $\mathcal{S}$ satisfies the following admissibility condition:

$$\mathbf{u} \in \mathcal{S} \text{ and } \mathbf{v} \subset \mathbf{u} \Rightarrow \mathbf{v} \in \mathcal{S}. \tag{4.21}$$

This is to guarantee that all the terms can be calculated according to Eq. (4.7). For example, the set of indices based on the superposition dimension can be defined as $\mathcal{S}_{N_s} := \{\mathbf{u} \subseteq \mathcal{D} : |\mathbf{u}| \leqslant N_s\}$ and the set of indices based on the truncation dimension can be defined as $\mathcal{S}_{N_t} := \{\mathbf{u} \subseteq \{1, \ldots, N_t\}\}$.

Therefore, from Section 4.2, we can define an interpolation formula $\mathcal{A}_{\mathcal{S}}f$ for the approximation of f which is given by

$$\mathcal{A}_{\mathcal{S}}f := \sum_{\mathbf{u}\in\mathcal{S}} \mathcal{A}(f_{\mathbf{u}}). \tag{4.22}$$

Here $\mathcal{A}(f_{\mathbf{u}})$ is the sparse grid interpolant of the component function $f_{\mathbf{u}}$ and $\mathcal{A}_{\mathcal{S}}f$ is the interpolant of the function f using the proposed method with the index set . It is common to refer to the terms $\{f_{\mathbf{u}} : |\mathbf{u}| = l\}$ collectively as the "order-l terms". Then the expansion order, p, for the decomposition Eq. (4.22) is defined as the

maximum of l. Note that the number of collocation points in this expansion is defined as the sum of the number of points for each sub-problem from Eq. (4.17), i.e. $M = \sum_{\mathbf{u}\in\mathcal{S}} M_{\mathbf{u}}$.

Now, let us consider the approximation error of the truncated HDMR expansion with ASGC used for the component functions in Eq. (4.22). To this end, we fix $\alpha \in (0, 1]$ and assume that N_s and N_t, the corresponding superposition and truncation dimensions, are known. We define the index set $\mathcal{S}_{N_t,N_s} := \{\mathbf{u} \subseteq \{1, \ldots, N_t\}, |\mathbf{u}| \leqslant N_s\}$. We can state the following theorem:

Theorem 1. *Let $\mathcal{S} = S_{N_t,N_s}$, and let $\mathcal{A}$ be the ASGC interpolant with the same error threshold ε for all the sub-problems. Then:*

$$|f - \mathcal{A}_{\mathcal{S}} f| \leqslant c(N_s, N_t)\varepsilon + \varepsilon_t, \tag{4.23}$$

for all $f \in F_N$, where $F_N := \left\{f : [0, 1]^N \to \mathbb{R}, D^{\|\mathbf{m}\|} f \text{ continuous}, m_i \leqslant 2, \forall i\right\}$. Here, the constant $c(N_t, N_s)$ depends on the effective dimensions but does not depend on the nominal dimension N. ε_t is the truncation error of Eq. (4.22) according to the definition of effective dimensions.

Proof: The proof of this theorem is similar to that of Theorem 3.3 in [57]. We start with

$$|f - \mathcal{A}_{\mathcal{S}} f| \leqslant |f - f_{\mathcal{S}}| + |f_{\mathcal{S}} - A_{\mathcal{S}} f|, \tag{4.24}$$

where $f_{\mathcal{S}} := \sum_{\mathbf{u}\in\mathcal{S}_{N_t,N_s}} f_{\mathbf{u}}(\mathbf{Y}_{\mathbf{u}})$. The first term on the right hand side is the truncation error and the second term is the interpolation error. According to the definition of effective dimensions, with increasing $\alpha \in (0, 1]$, the approximation approaches the true value. Therefore, for a fixed α, we can denote the truncation error as $|f - f_{\mathcal{S}}| = \varepsilon_t$. From Eq. (4.7), we have the expression

$$f_{\mathbf{u}} - A(f_{\mathbf{u}}) \;=\; \sum_{\mathbf{v}\subseteq\mathbf{u}} (-1)^{|\mathbf{u}|-|\mathbf{v}|} \left(P_{\mathbf{v}} f - A(P_{\mathbf{v}} f)\right). \tag{4.25}$$

According to the ASGC algorithm, it is known that the approximation error is controlled by the error threshold ε. Since we choose the same ε for each sub-problem, we have $|P_{\mathbf{v}} f - A(P_{\mathbf{v}} f)| \leqslant \varepsilon$ [61].

Therefore, we have

$$\begin{aligned} |f_{\mathcal{S}} - A_{\mathcal{S}}(f)| &\leqslant \sum_{\mathbf{u}\in\mathcal{S}} |f_{\mathbf{u}} - A(f_{\mathbf{u}})| \leqslant \sum_{\mathbf{u}\in\mathcal{S}} \sum_{\mathbf{v}\subseteq\mathbf{u}} |P_{\mathbf{v}} f - A(P_{\mathbf{v}} f)| \\ &\leqslant \sum_{\mathbf{u}\in\mathcal{S}} \sum_{\mathbf{v}\subseteq\mathbf{u}} \varepsilon = \sum_{k=1}^{N_s} \binom{N_t}{k} \sum_{j=1}^{k} \binom{k}{j} \varepsilon \leqslant c(N_t, N_s)\varepsilon, \end{aligned} \tag{4.26}$$

with the constant $c(N_t, N_s)$ given as [57]

$$c(N_t, N_s) := \sum_{k=1}^{N_s} \binom{N_t}{k} \sum_{j=1}^{k} \binom{k}{j} \leqslant \sum_{k=1}^{N_s} \binom{N_t}{k} 2^k \leqslant 2^{N_s+1} {N_t}^{N_s},$$

which completes the proof.

Therefore, it is expected that the expansion Eq. (4.22) converges to the true value with decreasing error threshold ε and increasing number of component functions.

4.4 Adaptive HDMR

In practice, the effective dimensions are not known *a priori*. In order to find the effective dimensions, one needs to compute all 2^N component functions. They were originally defined for the representation of pure mathematical functions [48, 57]. However, in our case, calculating effective dimensions is not practical since we need to solve PDEs and thus the computational cost is much higher than that for function evaluation. Note that the total number of component functions for a l-th order expansion is $\sum_{i=0}^{l} \frac{N!}{i!(N-i)!}$, which increases quickly

with the number of dimensions. Therefore, in this section, we would like to develop an adaptive version of HDMR for automatically and simultaneously detecting the truncation and superposition dimensions.

We here assume each component function $f_{\mathbf{u}}$ is associated with a weight $\eta_{\mathbf{u}} \geqslant 0$, which describes the contribution of the term $f_{\mathbf{u}}$ to the HDMR. Using this information, we then want to automatically determine the optimal index set $\mathcal{S}$, which consists of two steps.

At first, we try to find the important dimensions, i.e. the truncation dimension. To this end, we always construct the zeroth- and first-order HDMR expansion where the computational cost is affordable even for very high-dimensions. In this case, the weight is defined as:

$$\eta_i = \frac{\|J_{\{i\}}\|_{L_2(D)}}{\|f_0(\bar{\mathbf{Y}})\|_{L_2(D)}}, \tag{4.27}$$

where $J_{\{i\}} = \int f_i(Y_i) dY_i$ follows the definition in Eq. (4.18) and the L_2 norm is defined in the spatial domain when the output is a function of spatial coordinates. Each first-order component function is only a one-dimensional function which measures the impact on the output when each dimension is acting independently. According to Eqs. (4.12) and (4.27), this weight can be considered as a measurement of the sensitivity of the output when only the ith–dimension is the input. Then we define the important dimensions as those whose weights are larger than a predefined error threshold θ_1. Now, the set $\mathcal{D}$ in Eq. (4.16) only contains these important dimensions instead of all the dimensions. For example, if the important dimensions are 1, 3 and 5, then only the higher-order terms {13}, {15}, {35} and {135} are considered. It is noted that the important dimensions depend on the choice of θ_1. With decreasing θ_1, more dimensions become important and therefore more terms need to be included in the HDMR.

The reason for defining the weight in Eq. (4.27) is as follows. HDMR is aimed to reveal the correlation effects among the input variables as reflected upon the output. The correlation effect is large if the change of the random input within its range will lead to a significant change on the output. According to the definition of first-order component function, $f_i(Y_i) = f(\mathbf{Y})|_{\mathbf{Y}=\bar{\mathbf{Y}}\setminus Y_i} - f_0$, $f_i(Y_i)$ measures the difference between the function value with only input arising from one dimension acting independently and the function value at the reference point. It contains the information about the impact upon the output from this particular dimension. Therefore, if the weight is defined as in Eq. (4.27), it clearly gives us information on the impact that this dimension has when is acting alone upon the output. Only those dimensions which have significant impact on the output are considered as important. Then it is straightforward to argue that if two dimensions are important, there is a possibility that the impact upon the output is still significant if these two dimensions act together. Therefore, we need to consider all the component functions of only these important dimensions. It is noted that a similar definition is also proposed in [57] and is proved to be effective.

However, not all the possible terms are computed. Instead, we adaptively construct higher-order component functions increasingly from lower-order to higher-order in order to reduce the computational cost in the following way. For each computed higher-order term $f_{\mathbf{u}}, |\mathbf{u}| \geqslant 2$, a weight is also defined as

$$\eta_{\mathbf{u}} = \frac{\|J_{\mathbf{u}}\|_{L_2(D)}}{\left\|\sum_{\mathbf{v}\in\mathcal{S},|\mathbf{v}|\leqslant|\mathbf{u}|-1} J_{\mathbf{v}}\right\|_{L_2(D)}}. \tag{4.28}$$

It measures the relative importance with respect to the sum of current integral value which has already been computed in set from the previous order. Similarly, the important component functions are defined as those whose weights are larger than the predefined error threshold θ_1. We put all the important di-

mensions and higher-order terms into a set $\mathcal{T}$, which is called the important set. When adaptively constructing HDMR for each new order, we only calculate the term $f_{\mathbf{u}}$ whose indices satisfy the admissibility relation Eq. (4.21),

$$\mathbf{u} \in \mathcal{D} \text{ and } \mathbf{v} \subset \mathbf{u} \Rightarrow \mathbf{v} \in \mathcal{T}. \tag{4.29}$$

In other words, among all the possible indices, we only want to find the terms which can be computed using the previous known important component functions via Eq. (4.6). In this way, we find those terms which may have significant contribution to the overall expansion while ignoring other trivial terms thus reducing the computational cost for high-dimensional problems.

Let us denote the order of expansion as p. Furthermore, we also define a relative error ρ of the integral value between two consecutive expansion orders p and $p-1$ as

$$\rho = \frac{\left\| \sum_{|\mathbf{u}| \leqslant p} J_{\mathbf{u}} - \sum_{|\mathbf{u}| \leqslant p-1} J_{\mathbf{u}} \right\|_{L_2(D)}}{\left\| \sum_{|\mathbf{u}| \leqslant p-1} J_{\mathbf{u}} \right\|_{L_2(D)}}. \tag{4.30}$$

If ρ is smaller than another predefined error threshold θ_2, we consider that the HDMR has converged and the construction stops.

The above procedure is detailed in Algorithm 1.

It is noted here that we add all the computed indices to set even if their weight is below the threshold θ_1 in order to further improve the accuracy since we have already paid the cost to compute them. This is similar idea to that used in the ASGC algorithm.

Before closing this section, we want to comment again on the definition of the weights in Eqs. (4.27) and (4.28). As discussed before, these weights provide information on the contribution of each component function to the overall expansion. In addition, the weights from the first-order expansion also provide

Algorithm 1: Adaptive construction of the index set

Initialize: Let = $\{\emptyset\}$, $\mathcal{R} = \{\emptyset\}$ and $\mathcal{T} = \{\emptyset\}$. Set $p = 1$.

Construct the zeroth and first-order component functions:

- Solve each sub-problem using the ASGC method with error threshold ε and add all the indices to .
- Compute the weights of each first-order term according to Eq. (4.27). Add those dimensions which satisfy $\eta \geqslant \theta_1$ to set $\mathcal{T}$.

repeat

- $p \leftarrow p + 1$. Construct the set $\mathcal{R}$ whose indices satisfy the admissibility relation Eq. (4.29) for $|\mathbf{u}| = p$.
- If $\mathcal{R} \neq \{\emptyset\}$, for each index $\mathbf{u} \in \mathcal{R}$, solve the corresponding sub-problem using ASGC with error threshold ε and add all the indices to .
- Compute the weight of component functions according to Eq. (4.28). Add those indices which satisfy $\eta \geqslant \theta_1$ to set $\mathcal{T}$ and clear set $\mathcal{R}$.
- Compute the relative error ρ according to Eq. (4.30).

until $\mathcal{R} = \{\emptyset\}$ or $\rho < \theta_2$;

us information on the important dimensions. An important dimension here is defined in a relative sense, which means that its impact upon the output is more significant than that of others. Here, we provide a guideline to select θ_1. Since the expansion of each order only depends on the previous order, we can first only construct the first-order expansion. Then we can sort all dimensions in a descending order according to the value of η_i. Finally, the value of θ_1 is chosen such that only a certain portion of the dimensions are considered important. If

all the weights are the same, we have to include all the dimensions. Although we do not have the situation in the numerical examples considered here, there is a possibility that $\|f_0(\bar{\mathbf{Y}})\|_{L_2} = 0$. In this case, one can simply define $\eta_i = \|J_{\{i\}}\|_{L_2}$.

4.5 Numerical examples

In this section, two sets of numerical examples are considered to illustrate the proposed HDMR technique. In the investigations below we consider both *adaptive HDMR* and *conventional HDMR* where the adaptivity here refers only to the truncation of the expansion. In all examples, the ASGC method is used for the calculation of the component functions. The ASGC method used alone refers to the adaptive sparse grid collocation in [61] applied directly (without HDMR expansion) to the interpolation of the original function of interest or to the solution of the SPDE. Similarly, CSGC refers to the conventional sparse grid interpolation without adaptivity applied to the original function or solution of the SPDE. The first set of examples involves elementary mathematical functions while the second set involves stochastic fluid mechanics problems. Whenever possible, comparisons with alternative methods are provided to evaluate the accuracy, computational efficiency, and convergence of the proposed method. In the following examples, unless otherwise specified, θ_2 is fixed at 10^{-4}.

Example Set I - Mathematical Functions

It is noted that according to the right hand side of HDMR Eq. (4.1), if the function has an additive structure, then HDMR is exact, e.g., when $f(\mathbf{Y})$ can be ad-

ditively decomposed into functions $f_i(Y_i)$ of single variables, then the first-order expansion is enough to exactly represent the original function. On the other hand, if a multiplicative nature of the function is dominant then all components of HDMR may be needed to obtain an accurate result. However, if HDMR requires all 2^N components to obtain a desired accuracy, the method becomes very expensive. This example is designed to investigate how function structure and input variability affect the accuracy of the proposed method.

4.5.1 Example 1: Function with additive structure

Let us first consider the following simple function which has an additive structure in $[0,1]^N$:

$$f(\mathbf{Y}) = \sum_{i=1}^{N} Y_i^2. \tag{4.31}$$

where Y_i are uniform random variables with range $[0,1]$. Here, we take $N = 10$ and a third-order conventional HDMR expansion with $\varepsilon = 10^{-4}$. After constructing the HDMR, we generate 100 random points and compare the interpolating results of HDMR with the exact value. The normalized L_2 error is defined as

$$\epsilon_{L_2} = \frac{\sqrt{\sum_{i=1}^{100} \left(f_{HDMR}(\mathbf{Y}_i) - f_{exact}(\mathbf{Y}_i)\right)^2}}{\sqrt{\sum_{i=1}^{100} f_{exact}(\mathbf{Y}_i)^2}} \tag{4.32}$$

The result is shown in Table 4.1. As expected, only first-order HDMR is enough to represent the function exactly. Increasing the expansion order does not improve the accuracy.

We compare next the convergence rate of first-order HDMR with other methods, which is shown in Fig. 4.1. Using HDMR, the number of points required

Table 4.1: Interpolation of the function in Eq. (4.31) using HDMR of different orders with interpolation error threshold $\varepsilon = 10^{-4}$.

$N = 10$	# Points	L_2 Error
1^{st} order HDMR	1291	1.3027×10^{-4}
2^{nd} order HDMR	33016	1.3027×10^{-4}
3^{rd} order HDMR	340336	1.3027×10^{-4}

to achieve a certain level of accuracy is much less than that using other methods. The number of points for CSGC at level 6 is 171425, when going to the next level, the number increases to 652065. The number of points using ASGC alone is 12621. It is also interesting to note that the convergence rate of HDMR is the same as for ASGC, which shows that the error of HDMR is also controlled by ε. To further verify the results numerically, we compare the maximum error with respect to different ε for HDMR and ASGC, respectively, in Table 4.2. As expected the error threshold ε also controls the error of a converged HDMR. This can be partially explained as follows. From Eq. (4.23), the error of HDMR consists of two parts: truncation error of HDMR and interpolation error of ASGC in the component function calculation. When HDMR has converged, the error is only determined by the interpolation error of each low-dimensional component function using ASGC which is also ε, and recall that HDMR is a linear combination of all the low-dimensional functions. Therefore the error is approximately $C\varepsilon$, where C is a constant. This case suggests that a lower-order converged HDMR is a good substitute of ASGC with much less needed number of collocation points.

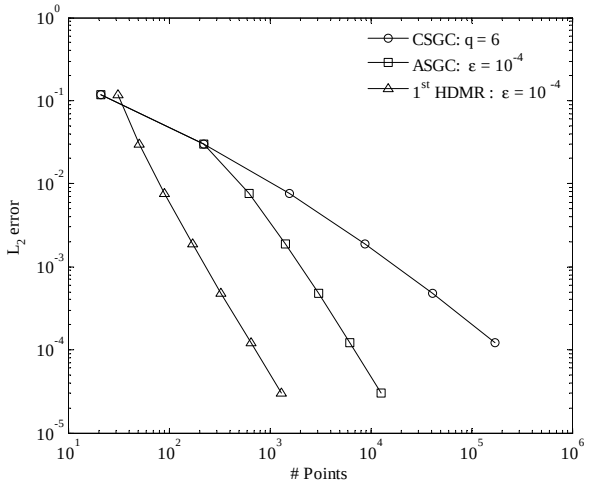


Figure 4.1: L_2 interpolation error versus number of points for the function in Eq. (4.31).

Table 4.2: Maximum error using first-order HDMR and ASGC with various ε.

$N = 10$	HDMR		ASGC	
	# Points	MaxError	# Points	MaxError
$\varepsilon = 10^{-1}$	51	1.3887×10^{-1}	221	1.3887×10^{-1}
$\varepsilon = 10^{-2}$	171	8.3629×10^{-3}	1421	8.3629×10^{-3}
$\varepsilon = 10^{-3}$	331	2.1349×10^{-3}	3021	2.1349×10^{-3}
$\varepsilon = 10^{-4}$	1291	1.3027×10^{-4}	12621	1.3027×10^{-4}
$\varepsilon = 10^{-5}$	5131	8.3142×10^{-6}	51021	8.3142×10^{-6}

4.5.2 Example 2: Function with multiplicative structure

Next we consider the function"product peak" in $[0,1]^N$ from GENZ test package [70] which has multiplicative nature:

$$f(\mathbf{Y}) = \prod_{i=1}^{N} \left(c_i^{-2} + (Y_i - w_i)^2 \right)^{-1}, \tag{4.33}$$

where c_i and w_i are constants and $N = 10$. Y_i are uniform random variables with range $[0, 1]$. The interpolation error is defined the same as before. The integration (mean) value of this function is defined as $I = \int_{[1,0]^{10}} f(\mathbf{Y}) d\mathbf{Y}$. The relative integration error is defined as $\frac{|I_{\text{num}} - I_{\text{exact}}|}{|I_{\text{exact}}|}$ where the exact value is available analytically.

We first examine the convergence of the HDMR with different expansion orders while fixing $\varepsilon = 10^{-6}$, which is shown in Fig. 4.2. Unlike the previous example, we now need all the terms in the HDMR expansion to obtain the exact result due to the multiplicative structure of the function. However, it is seen that the interpolation error decreases quickly with increasing expansion order. At least a fifth-order expansion is needed to achieve accuracy of $O(10^{-3})$ in this case. It is also interesting to note that HDMR converges faster for integration. Since the integration (mean) value is a statistical measure, it possibly suggests that the convergence rate depends more on the statistics of the input than the structure of the function itself. We will further investigate this in the next example.

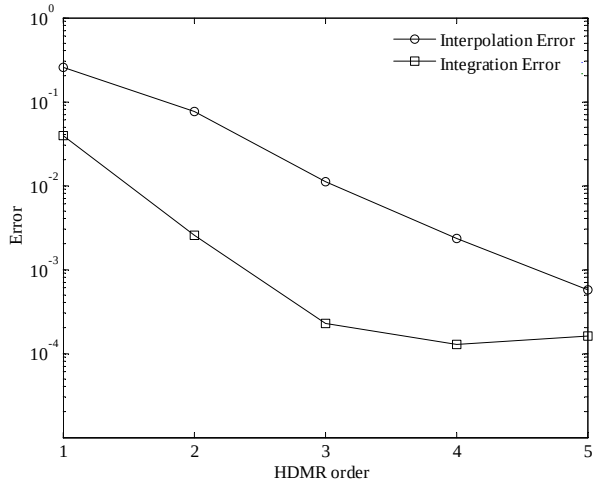


Figure 4.2: Convergence of HDMR for the function in Eq. (4.33) with different orders.

In Tables 4.3 and 4.4 , we also report the convergence of the ASGC used in the

calculation of the component functions for a fixed HDMR expansion order. In the combination of each order and ε, the first column is the L_2 interpolation error and the second is the relative integration error. As expected, the refinement of the ASGC does not improve the accuracy of the interpolation error within each expansion order. This is consistent with the previous results that when HDMR is not converged, the overall error is dominated by the truncation error. On the other hand, the integration value converges with a third-order expansion and for this case refining the ASGC improves the accuracy of the mean value. This suggests that the expansion order and the refinement of ASGC should be increased simultaneously to achieve a satisfactory accuracy, which is consistent with the results reported in [56].

Table 4.3: Relative L_2 interpolation error for various HDMR expansion order and error threshold.

Order	$\varepsilon = 10^{-5}$	$\varepsilon = 10^{-6}$	$\varepsilon = 10^{-7}$
1	2.56×10^{-1}	2.56×10^{-1}	2.56×10^{-1}
2	7.53×10^{-2}	7.57×10^{-2}	7.57×10^{-2}
3	1.07×10^{-2}	1.12×10^{-2}	1.13×10^{-2}

Table 4.4: Relative mean error for various HDMR expansion order and error threshold.

Order	$\varepsilon = 10^{-5}$	$\varepsilon = 10^{-6}$	$\varepsilon = 10^{-7}$
1	3.99×10^{-2}	3.98×10^{-2}	3.98×10^{-2}
2	2.23×10^{-3}	2.55×10^{-3}	2.65×10^{-3}
3	4.02×10^{-4}	2.25×10^{-4}	1.46×10^{-4}

Since each dimension may weigh unequally in this function, it is also interesting to compare the performance of HDMR with that of adaptive methods, i.e. adaptive HDMR and ASGC. The adaptive HDMR is combined with ASGC as described in Section 4.2. We fix $\theta_2 = 10^{-4}$ while varying θ_1. In addition, we choose the same $\varepsilon = 10^{-6}$ for all three methods (conventional fixed-order HDMR, adaptive HDMR and ASGC). The results are given in Table 4.5. For conventional HDMR, we need about 4 million collocation points to obtain an accurate result. On the other hand, it is seen that we can arrive at the same integration error as that of conventional HDMR and ASGC with much less number of collocation points by using adaptive HDMR. However, the interpolation error of adaptive HDMR is larger than that of the other two methods. This is expected since our error indicator for the important component functions is based on the mean value and thus it generally favors the strong interaction between those dimensions that impact the mean of the output. Therefore, due to the multiplicative nature of the function, we need to include all the component functions to compute an accurate value.

Table 4.5: Comparison of performance of conventional (fixed-order) HDMR, adaptive HDMR and ASGC.

	# Terms	# Points	$\epsilon_{\text{interpolation}}$	$\epsilon_{\text{integration}}$
5^{th} HDMR	638	4712870	5.65×10^{-4}	1.62×10^{-4}
HDMR: $\theta_1 = 10^{-3}$	78	83400	6.23×10^{-2}	8.72×10^{-4}
HDMR: $\theta_1 = 10^{-4}$	140	220930	1.47×10^{-2}	1.28×10^{-4}
ASGC: $\varepsilon = 10^{-6}$	N/A	305670	4.23×10^{-4}	1.28×10^{-4}

This example shows that HDMR may not be useful for interpolating arbitrary mathematical functions but it may have a good convergence property for

the approximation of integrals. Although the lack of the importance of higher-order effects for constructing the approximation of input-output relations for most deterministic physical systems is general [43, 44, 45, 46, 47], this feature for the interpolation of stochastic systems deserves further investigation in the next few examples. In addition, the relation between the convergence of the output statistics and the expansion order of HDMR is also important which has not been reported before.

In Fig. 4.3, we also investigate the effects of different choices of the reference point. It is shown that the results are indeed more accurate for a fixed-order expansion if the reference point is near the center of the domain. Therefore, in the remaining examples, we always take the reference point as the mean of the input random vector.

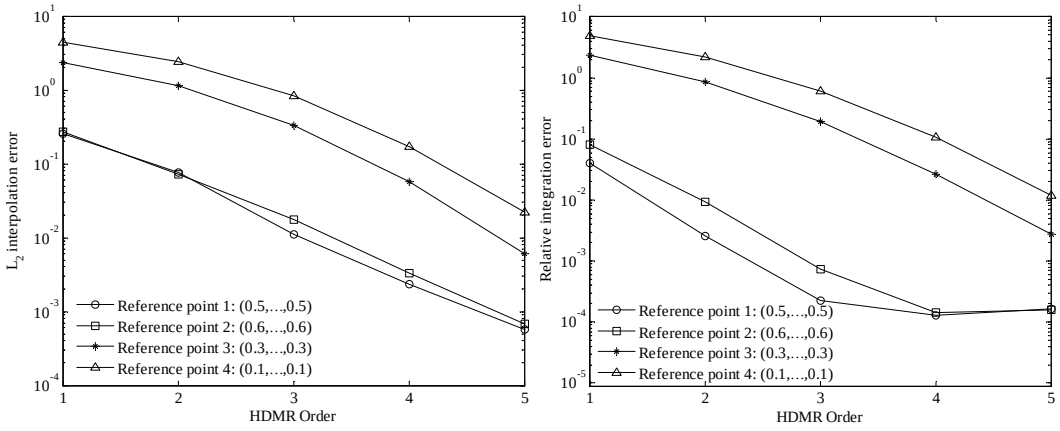


Figure 4.3: Error convergence for four different reference points.

4.5.3 Example 3: Function with different weights among dimensions

The final example in this set entails calculation of the mean and standard deviation of the output

$$f(\mathbf{X}) = \frac{1}{1 + \sum_{i=1}^{10} \alpha_i X_i}, \tag{4.34}$$

where the random input $X_i = \sigma Y_i$ and Y_i are i.i.d. uniform random variables in $[-\sqrt{3}, \sqrt{3}], i = 1, \ldots, 10$. Therefore, the parameter σ controls the standard deviation (std) of the input. The coefficients $\alpha_i, i = 1, \ldots, 10$ adjust the weight of each dimension. The HDMR expansion is employed to compute the mean and standard deviation of the output.

First, an isotropic case is considered by setting $\alpha_i = 0.1, i = 1, \ldots, 10$, i.e. each dimension weighs equally. The results are plotted in Fig. 4.4 for increasing values of standard input deviation. The relative errors are defined the same as before where the reference solution is taken from 10^6 MC samples. Here, conventional (fixed-order) HDMR is used where the error threshold is $\varepsilon = 10^{-6}$. From the figure, it is seen that with increasing σ, a higher-order expansion is needed to obtain an acceptable accuracy of order $O(10^{-3})$. On the other hand, when σ is small, only second-order expansion is enough to give us accurate results. Even first-order expansion is accurate when $\sigma \leqslant 0.1$. When $\sigma \geqslant 0.7$, fourth-order expansion is not enough and thus higher-order terms in the HDMR expansion are needed. The higher the input standard deviation is, the stronger the correlation effects are among the dimensions. Therefore, higher-order expansion is needed to capture these cooperative effects. In function Eq. (4.33), the input is uniform distribution $[0, 1]$. Its input standard deviation is about 0.289, which is not very large. This explains why its mean converges after a third-order expansion.

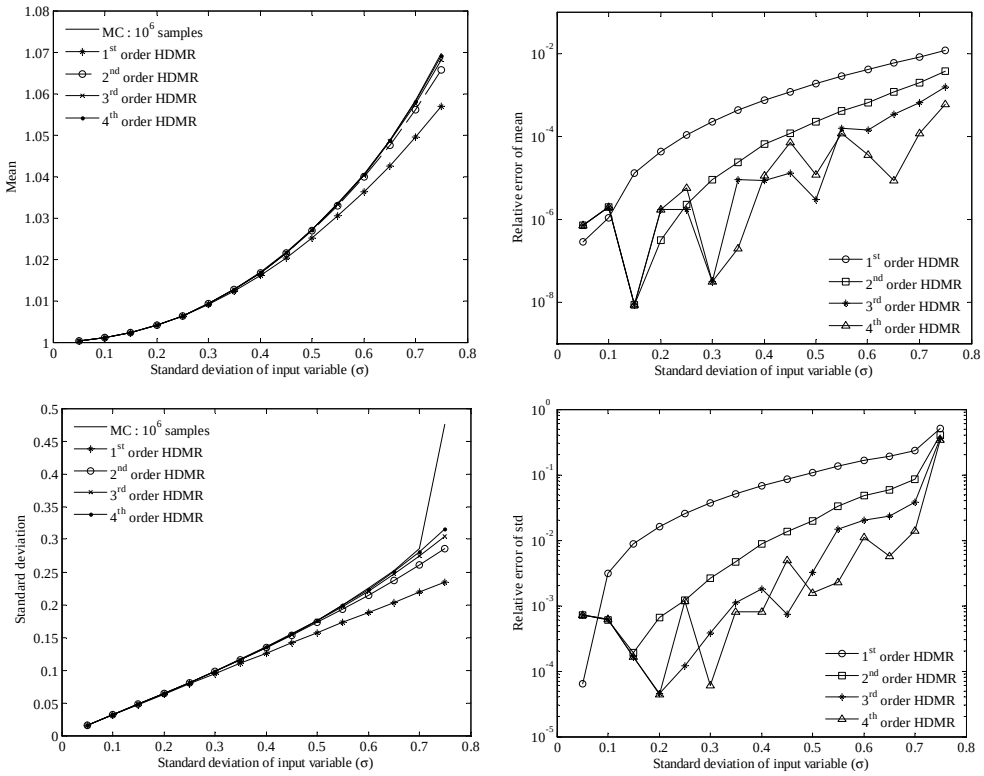


Figure 4.4: Mean and standard deviation of the function in Eq. (4.34) for increasing values of input standard deviation using conventional (fixed-order) HDMR. Top left: Mean; Top right: Error of mean; Bottom left: Standard deviation; Bottom right: Error of standard deviation.

The interpolation error is also given in Fig. 4.5. As expected, due to the non-linear nature of the function, higher-order HDMR expansions usually give better accuracy. But at the same time, the interpolation error for a fixed-order expansion also increases with increasing σ. It is also noted that, as for the integration case, a smaller σ has the effect of reducing the number of the effective dimensions of the function. For example, when $\sigma = 0.3$, the relative error is about 10^{-3} when using only a second-order HDMR.

Next, we consider two anisotropic cases. For Case 1, $\alpha_i = \dfrac{0.1}{2^{i-1}}, i = 1, \ldots, 10$. The ratio between the smallest coefficient and the largest one is about 0.002. For

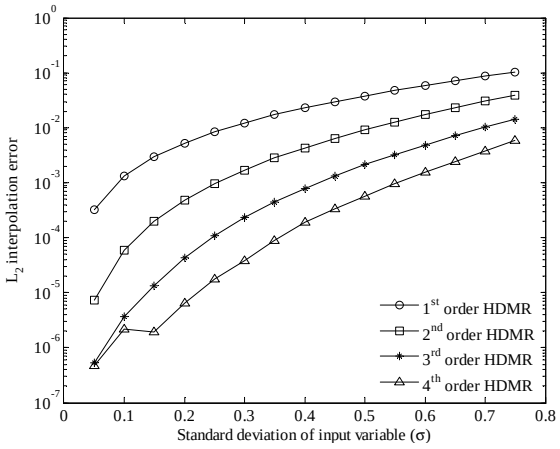


Figure 4.5: Interpolation error of the function in Eq. (4.34) using conventional (fixed-order) HDMR with different expansion orders.

Case 2, $\alpha_i = \dfrac{0.1}{10^{i-1}}, i = 1, \ldots, 10$. The ratio between the smallest coefficient and the largest one is about 10^{-9}. Thus, Case 1 refers to an anisotropic problem with small differences between dimensions while Case 2 refers to a highly anisotropic problem where the first four dimensions are the most important. In these two cases, we again fix the ASGC threshold $\varepsilon = 10^{-6}$ and choose a rather large input standard deviation $\sigma = 2.0$. The reference solution is still taken from 10^6 Monte Carlo samples. The results for Cases 1 and 2 are plotted in Figs. 4.6 and 4.7, respectively. In the left plot of Fig. 4.6, it is seen that the mean and standard deviation converge even for such a large σ. The interpolation error achieves an order of 10^{-4} with only a fourth-order HDMR. Therefore, the different weights between dimensions result in reduction of the number of effective dimensions in the HDMR expansion. This is more obvious in the left plot of Fig. 4.7 where the mean and std errors converge after a second-order expansion and the interpolation error converges after a third-order expansion. This can be explained as follows. If the weights of some dimensions are much larger than those of others, then these dimensions have most of the impact on the output. In this

case, only those component functions which consist of these dimensions provide most of the contribution to HDMR. If the number of important dimensions is small, then less component functions are needed and this leads to an accurate low-order HDMR expansion.

Since each dimension is given a different weight, we also use adaptive HDMR. In the plots on the right of Figs. 4.6 and 4.7, we compare the HDMR (conventional and adaptive) convergence rates for Cases 1 and 2, respectively, with those of ASGC. To plot the convergence of conventional (fixed-order) HDMR, we increase the expansion order from 1 to 5 (each symbol in the figures denotes an expansion order). The convergence of adaptive HDMR is plotted by decreasing θ_1 (each symbol in the figures corresponds to a different θ_1 value). To plot the convergence of ASGC, we plot the error at each interpolation level when constructing the grid level by level. It is seen that for Case 2 adaptive HDMR is much better than both conventional HDMR and ASGC. On the other hand, in Case 1, ASGC is better than the HDMR methods in terms of interpolation. This is expected since, in such a high σ case, there is no significant difference between dimensions and we need a higher-order expansion to capture the interaction effects.

Tables 4.6 and 4.7 show the total number of collocation points and also the number of expansion terms used in HDMR for different θ_1. It is noted that adaptive HDMR requires much less expansion terms and thus less collocation points than conventional HDMR, especially in Case 2. It is also interesting to note that to achieve the same accuracy, the number of expansion terms in Case 2 is less than that of case 1. This is due to the strong anisotropic weights such that only the first five dimensions are important. Therefore, only those terms

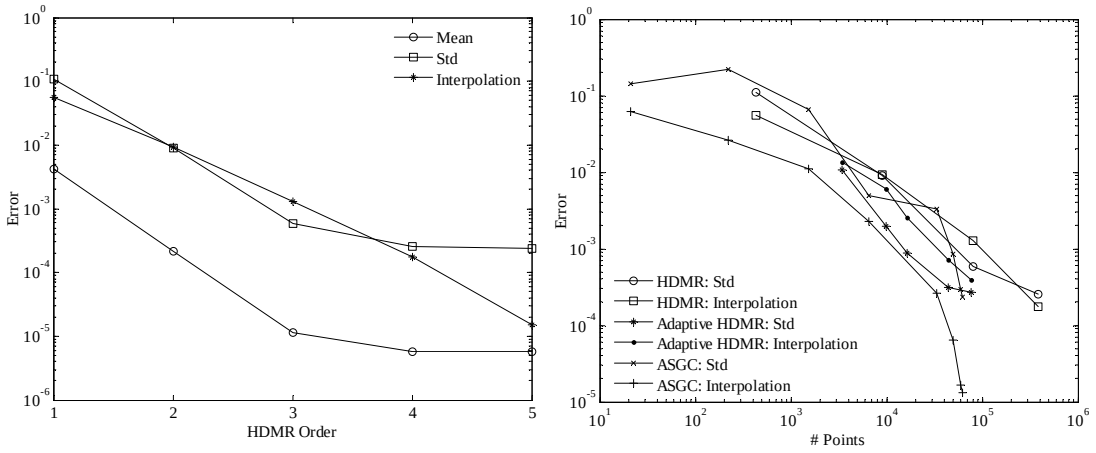


Figure 4.6: Error convergence of the anisotropic Case 1 when $\alpha_i = 0.1/2^{i-1}, i = 1, \ldots, 10$ with $\sigma = 2.0$. Left: Conventional (fixed-order) HDMR. Right: Comparison between conventional (fixed-order) HDMR, adaptive HDMR and ASGC.

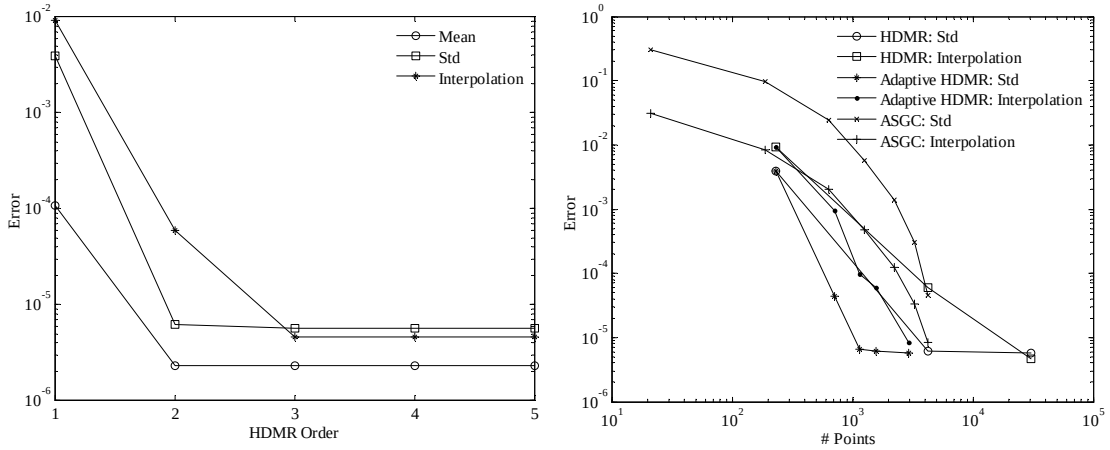


Figure 4.7: Error convergence of anisotropic Case 2 when $\alpha_i = 0.1/10^{i-1}, i = 1, \ldots, 10$ with $\sigma = 2.0$. Left: Conventional (fixed-order) HDMR. Right: Comparison between conventional (fixed-order) HDMR, adaptive HDMR and ASGC.

which involve these dimensions are constructed by adaptive HDMR. In Fig. 4.8, for both cases we also plot the weights, which are defined in Eq. (4.27) for each first-order component function indicating the importance of each dimension. It can be seen that the weights decrease quickly with the indices of the dimensions. In Case 1, the number of important dimensions are 7, 9 and 10 for $\theta_1 = 10^{-5}, 10^{-6}$

and 10^{-7}, respectively. In Case 2, the number of important dimensions are 3, 4 and 5 for $\theta_1 = 10^{-6}, 10^{-7}$ and 10^{-9}, respectively.

Table 4.6: Comparison of performance of fourth-order HDMR and adaptive HDMR for the anisotropic Case 1 where $\alpha_i = 0.1/2^{i-1}, i = 1, \ldots, 10$.

	4^{th} HDMR	Adaptive HDMR		
		$\theta_1 = 10^{-5}$	$\theta_1 = 10^{-6}$	$\theta_1 = 10^{-7}$
# Terms	386	37	64	88
# Points	388891	16584	44095	75877

Table 4.7: Comparison of performance of third-order HDMR and adaptive HDMR for the anisotropic Case 2 where $\alpha_i = 0.1/10^{i-1}, i = 1, \ldots, 10$.

	3^{rd} HDMR	Adaptive HDMR		
		$\theta_1 = 10^{-6}$	$\theta_1 = 10^{-7}$	$\theta_1 = 10^{-9}$
# Terms	176	14	17	22
# Points	30296	1144	1575	2912

Overall, the HDMR reveals the correlations among the input variables as reflected upon the output. Each order-l terms introduce the correlated effects of l-input variables into the expansion. Therefore, the order of HDMR expansion or the total number of component functions depends on the cooperative effects among the input dimensions, which further depends on the input variability and the importance of each dimension as demonstrated through this example set. In general, higher-order expansion will give better accuracy since it captures more correlation effects. However, as shown in this example, HDMR is useful

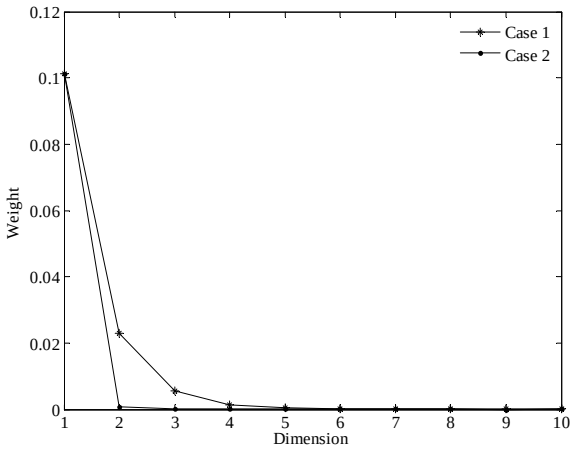


Figure 4.8: The decay of the weights of the first-order expansion terms for the two anisotropic cases.

if it can represent the output to a good accuracy with a sufficiently low-order expansion. For the case of small input variability and strong anisotropic situation, a low-order expansion can lead to a good accuracy. At the same time, by using adaptive HDMR, we can also significantly reduce the computational cost without sacrificing the accuracy. In the next two sections, we will investigate the convergence properties of adaptive HDMR on realistic physical systems and investigate if similar conclusions are obtained.

Example Set II - Fluid-Mechanics Problems

In the following two examples, random fields which are discretized by K-L expansion are introduced to increase the dimension of the stochastic problem. The reference solutions are taken from Monte Carlo simulations. The obtained mean values in these examples compared extremely well with the corresponding MC results for all cases considered and therefore we will focus our discussion only

on the standard deviation. The spatial L_2 normalized error of std σ is defined as

$$\epsilon_{L_2} = \frac{\sqrt{\sum_{i=1}^{n} \left(\sigma_{HDMR}(\boldsymbol{x}_i) - \sigma_{MC}(\boldsymbol{x}_i)\right)^2}}{\sqrt{\sum_{i=1}^{n} \sigma_{MC}(\boldsymbol{x}_i)^2}} \tag{4.35}$$

where n is the number of grid points in the spatial domain.

4.5.4 Example 4: Flow through random heterogeneous media

In this section, we consider flow through random porous media where the permeability is a random field obtained from the K-L expansion of an exponential covariance function. Through this classical problem, we want to investigate the effects of input variability on the accuracy and convergence of HDMR. The problem is defined as follows:

$$\nabla \cdot \boldsymbol{u}(\boldsymbol{x}, \mathbf{Y}) = f(\boldsymbol{x}), \tag{4.36}$$

$$\boldsymbol{u}(\boldsymbol{x}, \mathbf{Y}) = -K(\boldsymbol{x}, \mathbf{Y}) \nabla p(\boldsymbol{x}, \mathbf{Y}), \tag{4.37}$$

where the source/sink term $f(\boldsymbol{x})$ is taken to be deterministic and $K(\boldsymbol{x}, \mathbf{Y})$ is the random permeability. The domain of interest is a quarter-five spot problem in a unit square $D = [0, 1]^2$. Flow is driven by an injection well at the bottom left corner of the domain and a production well at the top right corner. A mixed finite element method is utilized to solve the forward problem [71].

The log-permeability is taken as zero mean random field with a separable exponential covariance function

$$\mathrm{Cov}(\boldsymbol{x}, \boldsymbol{y}) = \sigma^2 \exp\left(-\frac{|x_1 - y_1|}{L} - \frac{|x_2 - y_2|}{L}\right), \tag{4.38}$$

where L is the correlation length and σ is the standard deviation of the random

field. The K-L expansion is used to parameterize the field as

$$\mathbf{Y}(\omega) = \log\left(K(\omega)\right) = \sum_{i=1}^{N} \sqrt{\lambda_i}\phi_i(\boldsymbol{x})Y_i, \tag{4.39}$$

where the eigenvalues $\lambda_i, i = 1, 2, \ldots,$ and their corresponding eigenfunctions $\phi_i, i = 1, 2, \ldots,$ can be determined analytically as discussed in [3, 5]. The Y_i are assumed as i.i.d. uniform random variables on $[-1, 1]$.

In order to investigate the accuracy and applicability of the proposed HDMR approach, we design a series of numerical runs with different correlation lengths L and various degrees of spatial variability σ^2. The first three cases aim to investigate the effects of correlation length ($L = 1.0, 0.5$ and 0.25) on the proposed approach. In these cases, the degree of spatial variability is kept at $\sigma^2 = 1.0$, which corresponds to a moderately high variability. The next three cases are compared against case 3 when $L = 0.25$ to examine the impact of the log-permeability variability ($\sigma^2 = 0.01, 0.25$ and 2.0) ranging from very low to extremely high variability. Monte Carlo simulations are conducted for the purpose of comparison. For each case, the reference solution is taken from 10^6 MC samples and all errors are defined as normalized L_2 errors. In all cases, the threshold θ_2 for the relative error is fixed at $\theta_2 = 10^{-4}$.

Let us first determine the stochastic dimension of our example cases. The eigenvalues and the sum of the eigenvalues as a function of the number of terms included are illustrated in Fig. 4.9 with three different correlation lengths for the case $\sigma^2 = 1.0$. The corresponding eigenfunctions are shown in Fig. 4.10. Based on these figures, the K-L expansions are truncated after 33, 108 and 500 terms, respectively for $L = 1.0, 0.5, 0.25$, which represents $\approx 95\%$ percent of the total variance of the exponential covariance function. The truncation level for the expansion does not change for a fixed correlation length. Therefore, in all cases,

the number of stochastic dimensions is $N = 33, 108$ and 500, respectively for $L = 1.0, 0.5$ and 0.25. In these cases, the ratio between the smallest eigenvalue and the largest eigenvalue is about 10^{-3} which is similar to the moderate anisotropic case 1 in Section 4.5.3. According to previous results on mathematical functions, we would expect that the convergence of the HDMR expansion depends on the input variability and that HDMR can deal with a moderately high stochastic input standard deviation.

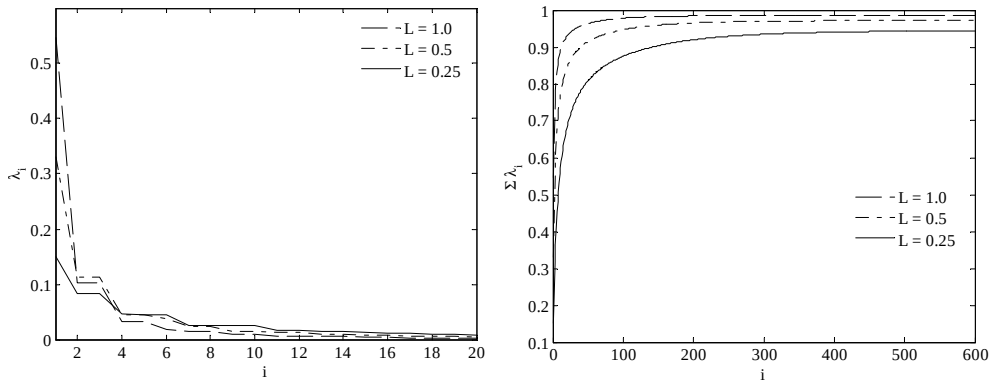


Figure 4.9: Series of eigenvalues and their finite sums for three different correlation lengths at $\sigma^2 = 1.0$.

Effect of correlation length

We fix $\sigma^2 = 1.0$ and consider three different correlation lengths at $L = 1.0, 0.5$ and 0.25. In this way, the input variability is fixed and the change of correlation length adjusts the weights of each random dimension. Thus, we want to investigate the effect of the smoothness of the stochastic space on the accuracy of the HDMR expansion as we did in Section 4.5.3.

We decrease ε and θ_1 simultaneously until the L_2 normalized errors reach an order of $O(10^{-3})$. It is interesting to note that these computed errors are achieved

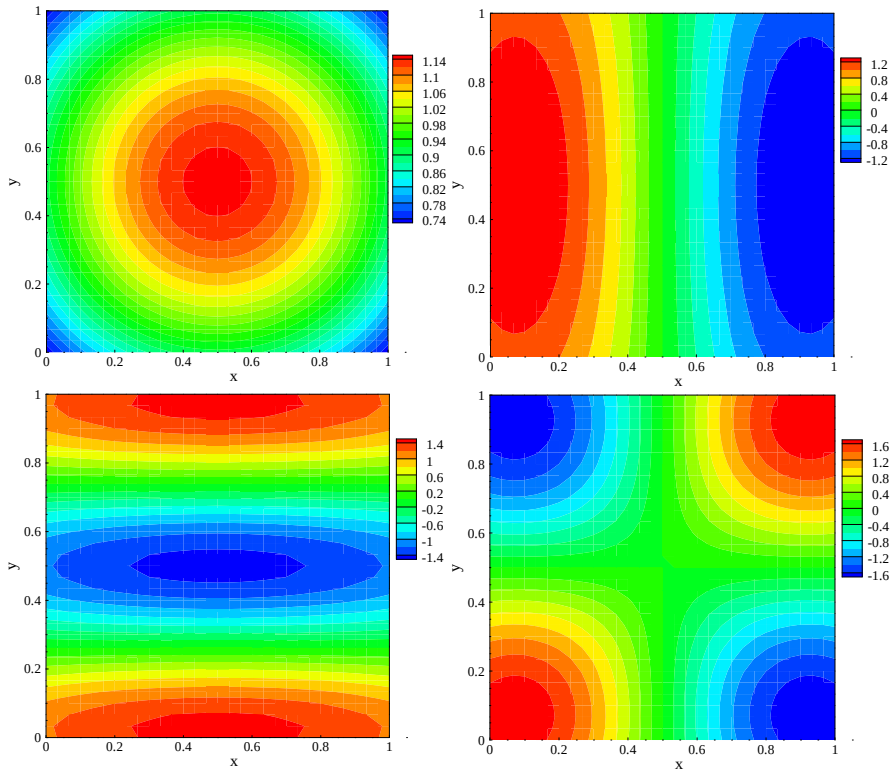


Figure 4.10: Four different eigenfunctions for $L = 1.0$ and $\sigma^2 = 1.0$. Top left: ϕ_1; Top right: ϕ_2; Bottom left: ϕ_5; Bottom right: ϕ_6.

with the same $\varepsilon = 10^{-6}$ and $\theta_1 = 10^{-4}$ for all three correlation lengths. In Fig. 4.11, we compare the standard deviation of the v velocity-component along the line $y = 0.5$ with the results from the MC simulation. It is seen that the two results compare extremely well, where the errors are 1.46×10^{-3}, 1.19×10^{-3} and 1.39×10^{-3}, respectively for $L = 1.0$, $L = 0.5$ and $L = 0.25$.

To investigate the convergence of HDMR, we fix $\varepsilon = 10^{-6}$ while varying θ_1. The PDFs of the v velocity-component at point $(0, 0.5)$, where the standard deviation is large as in Fig. 4.11, are shown in Fig. 4.12. Each PDF is generated by plotting the kernel density estimate of 10000 output samples through sampling the input space and computing the output value through the HDMR approximation. When $\theta_1 = 10^{-2}$, the weights of all first-order terms are smaller

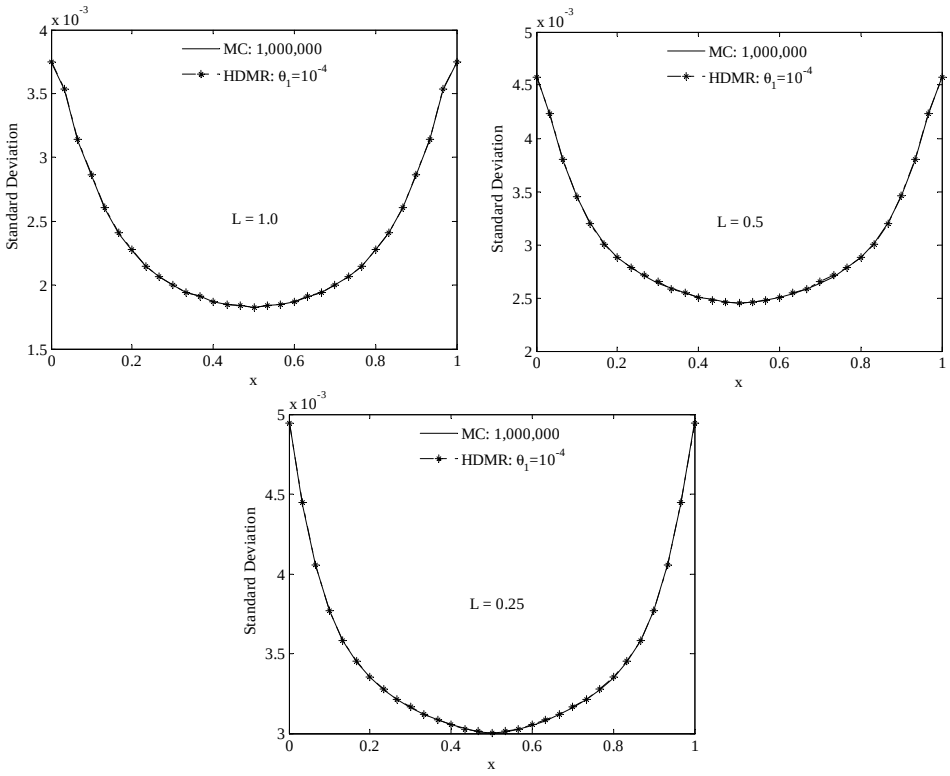


Figure 4.11: Standard deviation of the v velocity-component along the cross section $y = 0.5$ for different correlation lengths where $\sigma^2 = 1.0$, $\varepsilon = 10^{-6}$ and $\theta_1 = 10^{-4}$.

than the threshold and thus there are no higher-order terms. It is seen that the results from only the first-order expansion are not accurate in all three cases which is consistent with the result in Fig. 4.6. This may be explained intuitively as follows. The spatial variability σ^2 determines the total input variability, which further determines the correlation effects among the input variables. The larger the input variability is, the stronger the correlation effects are. The role of HDMR component functions is to capture these input effects upon the output. In other words, the number of component functions needed depends on the input variability. The higher the input variability is, the more component functions are needed for a fixed stochastic dimension. For low-input variability, even first-order expansion may be accurate enough no matter what the stochas-

tic dimension is. In our case, $\sigma^2 = 1.0$ represents a rather high-input variability. Higher-order terms are therefore needed to capture these effects whereas only first-order terms are not enough. The number of important dimensions is shown in Table 4.8. By decreasing θ_1, more dimensions whose weights defined in Eq. (4.27) are larger than θ_1 become important and thus more second-order terms appear. This means more correlation effects are included in the expansion and thus better accuracy. The number of component functions is 287, 889 and 2271 while for the full second-order expansion it is 562, 5887 and 125251, respectively. Thus, the advantage of using adaptive HDMR is obvious, especially for $N = 500$. As expected, all the results converge to those obtained from the MC simulation with decreasing θ_1. The computed PDFs indicate that the corresponding HDMR approximations are indeed very accurate. Therefore, we can obtain any statistic from this stochastic reduced-order model, which is an advantage of the current method over the MC method.

Table 4.8: Number of important dimensions N_i and component functions N_c for different θ_1.

θ_1	$L = 1.0$		$L = 0.5$		$L = 0.25$	
	N_i	N_c	N_i	N_c	N_i	N_c
10^{-2}	0	34	0	109	0	501
10^{-3}	5	44	8	137	11	556
10^{-4}	23	287	40	889	60	2271

The convergence of the L_2 normalized error with respect to the total number of collocation points is shown in Fig. 4.13 by fixing $\theta_1 = 10^{-4}$. The error plots are obtained by decreasing the threshold ε used in the ASGC. Although different correlation lengths result in different truncated stochastic dimensions from the

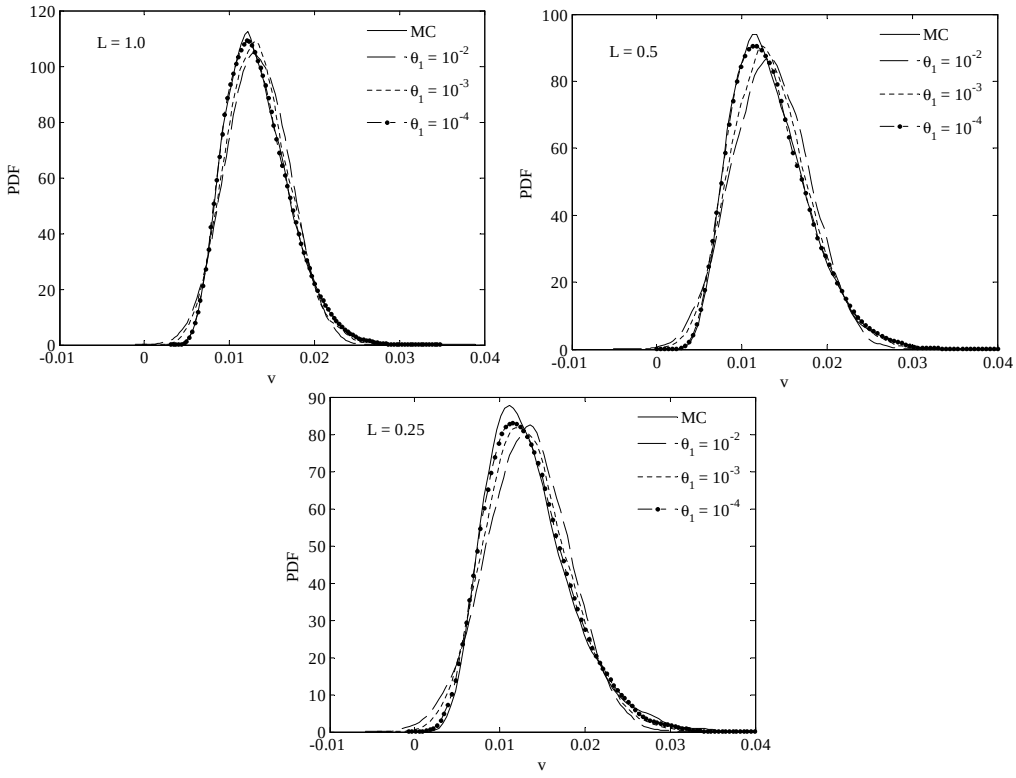


Figure 4.12: PDF of v velocity-component at point $(0, 0.5)$ for different correlation lengths, where $\sigma^2 = 1.0$ and $\varepsilon = 10^{-6}$.

K-L expansion, it is interesting to note that we still have algebraic convergence rates and they are nearly the same for all three cases. This may be explained using Theorem 1. The input variability is the same and so is the superposition dimension. Thus, the constant in Eq. (4.23), which only depends on the effective dimensions, is nearly the same. In addition, the error in Eq. (4.23) exhibits a linear dependence on the threshold ε and as we know the convergence rate of the sparse grid collocation method is algebraic [33, 61]. Thus, the convergence plot here indeed exhibits algebraic rate as indicated from Theorem 1.

In order to further verify the results discussed in this section, we investigate the effect of the spatial variability in the next section.

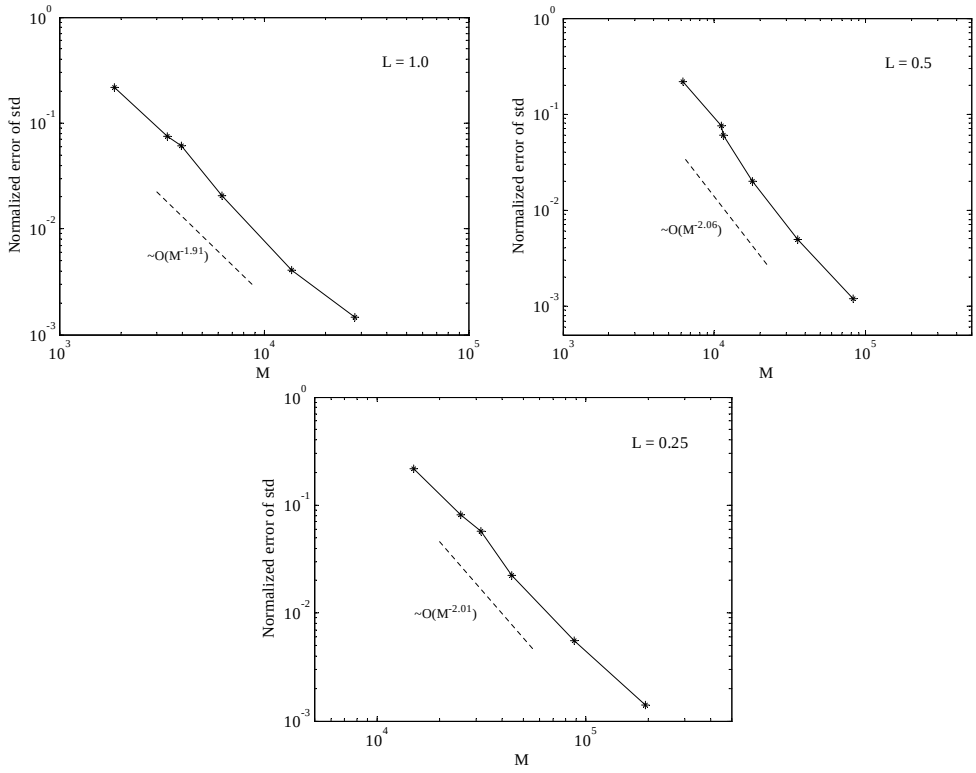


Figure 4.13: Convergence of the L_2 normalized errors of the standard deviation of the v velocity-component for different correlation lengths, where $\sigma^2 = 1.0$ and $\theta_1 = 10^{-4}$.

Effect of the spatial variability σ^2

In this section, we fix the correlation length at $L = 0.25$ such that the weight of each dimension from the K-L expansion is nearly the same. Then we explore the effects of the spatial variability σ^2, from very small variability $\sigma^2 = 0.01$ to very high variability $\sigma^2 = 2.0$. The number of stochastic dimensions is $N = 500$. For such a nearly isotropic situation, as indicated in Fig. 4.4, it is expected that the accuracy of HDMR depends on the input spatial variability.

Fig. 4.14 compares the standard deviation of the v velocity-component along the line $y = 0.5$ with the results obtained from the MC simulation. Again, we obtain very good comparison where the errors are 8.08×10^{-4}, 7.37×10^{-4} and

2.86×10^{-3} for $\sigma^2 = 0.01$, $\sigma^2 = 0.25$ and $\sigma^2 = 2.0$, respectively. Comparing with the result when $\sigma^2 = 1.0$, it can be seen that the error in standard deviation increases with increasing input variability which again verifies our previous results in Fig. 4.4. However, we can still obtain an acceptable accuracy even when the spatial variability is as high as $\sigma^2 = 2.0$. It is interesting to note that the threshold ε needed to achieve the desired accuracy is smaller for $\sigma^2 = 0.01$ and 0.25 than for $\sigma^2 = 1.0$. This is due to the small correlation length and low input variability which results in a rather smooth stochastic space such that the hierarchical surpluses decrease very fast. Therefore, we need a much smaller ε to trigger the adaptivity otherwise the refinement of ASGC for each sub-problem stops earlier. This also shows the ability of ASGC to detect the smoothness of the stochastic space. The increase of the spatial variability also results in the increase of the magnitude of the standard deviation from $\sigma^2 = 0.01$ to $\sigma^2 = 2.0$ in Fig. 4.14, i.e. increasing of the variability in the solution.

Similarly, the convergence of the corresponding PDFs is given in Fig. 4.15. For the case $\sigma^2 = 0.01$, we show the convergence with respect to ε instead. This is because the first-order expansion is accurate enough to represent the solution due to the rather low input variability as expected. Even when the value of θ_1 is as small as 10^{-4}, the weights of all the first-order terms from Eq. (4.27) are still smaller than the threshold. Therefore, there is no need to include second-order terms. Due to the small variability, even a smaller ε can give us a very accurate result as shown in the figure. Compared with Fig. 4.12, although second-order terms are still needed for the cases $\sigma^2 = 0.25$ and 2.0, the PDFs from the first-order terms are not apart from the MC results for the cases $\sigma^2 = 0.25$ and $\theta_1 = 10^{-2}$ when the input variability is moderately high. On the other hand, for $\sigma^2 = 2.0$, the result from the first-order HDMR expansion ($\theta_1 = 10^{-2}$) deviates

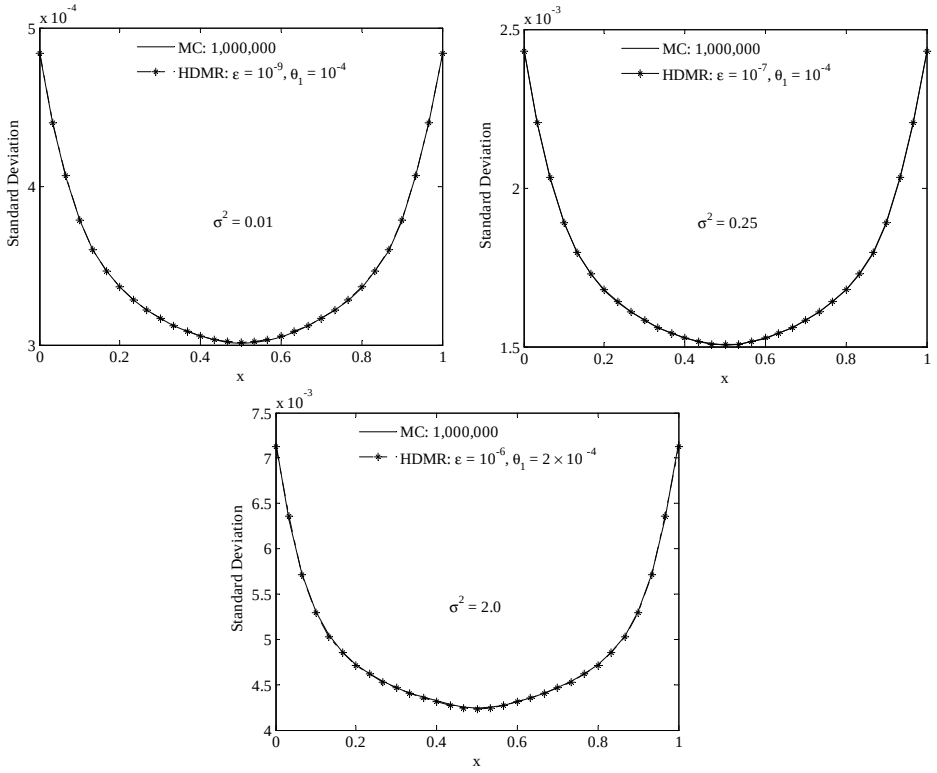


Figure 4.14: Standard deviation of the v velocity-component along the cross section $y = 0.5$ for different σ^2, where $L = 0.25$ and $N = 500$.

significantly from that of the MC results. Indeed, comparing all four cases when $L = 0.25$, it is clear that the PDFs from the first-order expansion deviate gradually from the MC results with increasing input variability σ^2. This reflects that the correlation effects become significant and therefore more higher-order terms are needed to capture these effects, which is again consistent with the result in Fig. 4.5. The improvement of the results is obvious as more terms are included with decreasing θ_1. This numerically verifies our previous discussion that the number of component functions needed depends on the input variability. There are 501, 879 and 2271 component functions for $\sigma^2 = 0.01$, 0.25 and 2.0, respectively. Correspondingly, the total number of collocation points are 9337, 74127, and 249329.

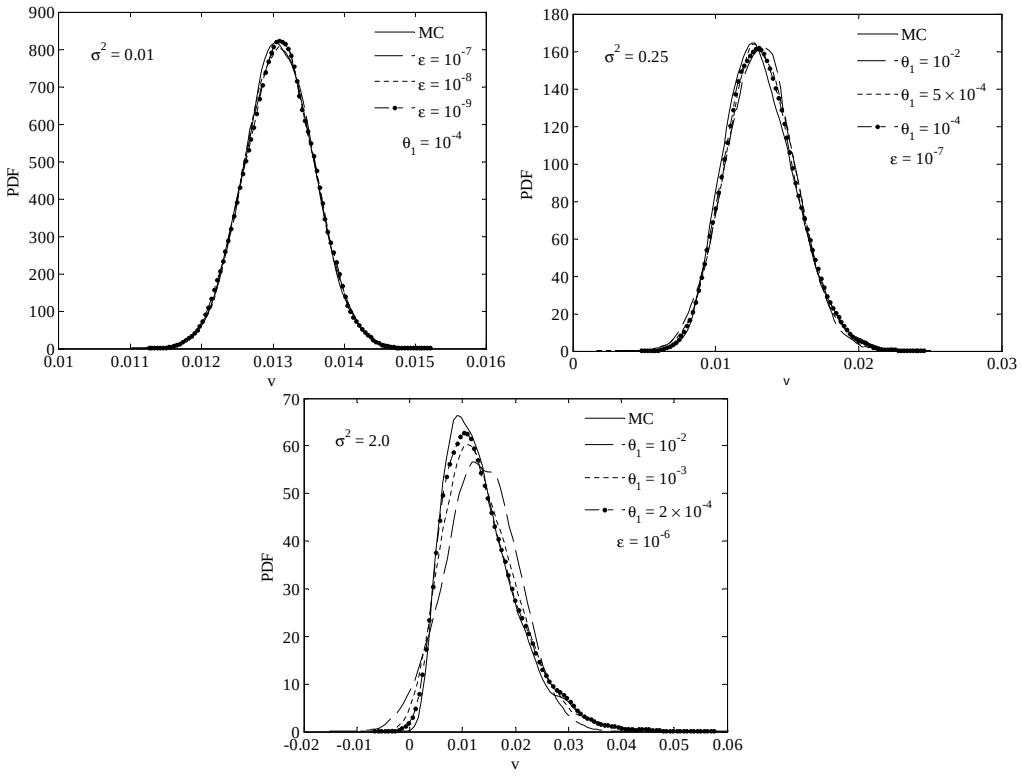


Figure 4.15: PDF of the v velocity-component at point $(0, 0.5)$ for different σ^2, where $L = 0.25$ and $N = 500$.

The convergence of the L_2 normalized error with respect to the total number of collocation points is shown in Fig. 4.13 by decreasing ε in the ASGC. As expected, the convergence rates deteriorate with increasing σ^2. For $\sigma^2 = 0.01$, the convergence rate is nearly of the order of $O(M^{-2.87})$. This is because first-order HDMR expansion is used and the number of collocation points used in the first-order terms is much lower than that in higher-order terms. The convergence rate for the case $\sigma^2 = 2.0$ is the lowest among all the cases examined so far. However, it is still better than the theoretic MC rate $O(M^{-0.5})$ and the results compare well with that of MC. Comparing all four cases when $L = 0.25$, it is seen that although the stochastic space is smooth, the convergence rate still decreases with increasing spatial variability. Thus, as discussed before, the number of component functions needed for a fixed stochastic dimension depends

more on the input variability for such situation. This provides us a very useful guideline to set up the parameters when applying HDMR to realistic stochastic problems. On the other hand, the regularity of the stochastic space can be effectively resolved with the ASGC method. The proposed method is indeed a very powerful and versatile method to address stochastic PDE problems.

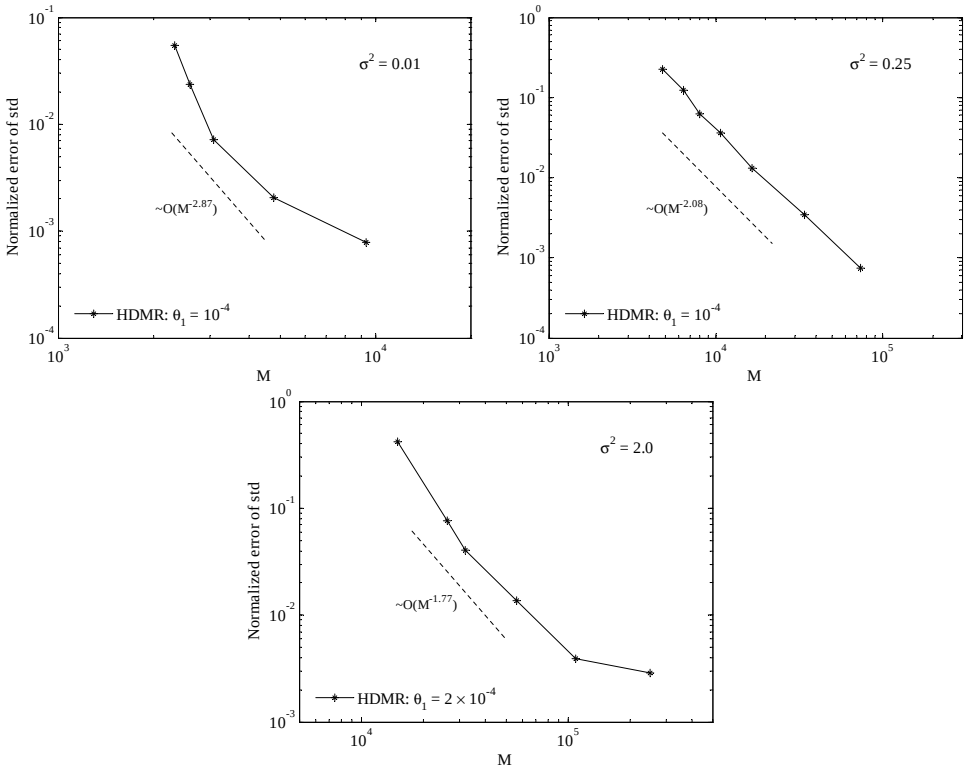


Figure 4.16: Convergence of the L_2 normalized errors of standard deviation of the v velocity-component for different σ^2, where L = 0.25.

Effect of choices of the reference points $\bar{\mathbf{Y}}$

Here, we also investigate the effect of the reference points $\bar{\mathbf{Y}}$ on the convergence of the adaptive HDMR expansion when $\sigma^2 = 1.0$ and $L = 1.0$ as discussed in Section 4.1.2. We fix $\varepsilon = 10^{-6}$ and vary θ_1. The error convergence for three

different reference points is shown in Fig. 4.17. As expected, when the reference point is far from the mean point, the convergence rate is quite slow and many more points are required to achieve a good accuracy.

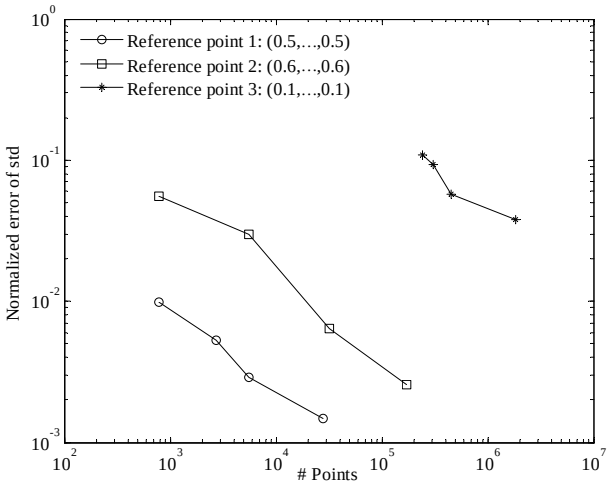


Figure 4.17: Convergence of the L_2 normalized errors of standard deviation of the v velocity-component for three different reference points.

In Table 4.9, we also tabulate the relevant parameters. It is seen from the table and Fig. 4.17 that when the reference point $\bar{\mathbf{Y}} = (0.5, \ldots, 0.5)$, it requires the least number of component functions and thus collocation points to achieve an error of order $O(10^{-3})$. When the reference point is far from the center, higher-order expansion is needed to obtain an accurate result. However, when $\bar{\mathbf{Y}} = (0.1, \ldots, 0.1)$, the error is 3.82×10^{-2} even when using a fourth-order expansion. Thus, we need to increase the expansion order or decrease θ_1. This will significantly increase the computational cost. Overall, the result of HDMR is sensitive to the choice of reference point. According to our experience, the mean vector can always give us a satisfactory result with much less computational cost. It is also interesting to note that when $\bar{\mathbf{Y}} = (0.6, \ldots, 0.6)$, in order to achieve an error of order $O(10^{-3})$, the number of component functions is 880

while the number is 6018 when using conventional HDMR. This again shows the advantage of the adaptive HDMR method.

Table 4.9: Comparison of performance for three different reference points. $\varepsilon = 10^{-6}$ and $\theta_1 = 10^{-4}$. N_i denotes the number of important dimensions and p denotes the expansion order of the HDMR defined in Section 4.3.

$\bar{\mathbf{Y}}$	# Terms	# Points	N_i	p
(0.5,...,0.5)	287	27968	23	2
(0.6,...,0.6)	880	170992	33	3
(0.1,...,0.1)	5526	1854460	33	4

Finally, it is emphasized here that using the ASGC or CSGC methods alone as in [61] to solve these high-dimensional problems is not feasible due to the following two reasons. At first, the number of needed collocation points is significant for such high-dimensional cases and the convergence rate is very small due to the logarithmic term shown in the error estimation. For example, for $N = 500$, the number of points is 167169001 for a third-level interpolation. Secondly, there is a need for large memory storage to store all the high-dimensional multi-indices. Therefore, we are not able to compare the results shown in this chapter with results that can be obtained directly from the ASGC. However, through the numerical examples, it is shown that the method presented here is indeed a useful tool for solving high-dimensional stochastic problems.

4.5.5 Example 5: Stochastic natural convection problem with random boundary conditions

It is interesting to also solve non-linear transient SPDEs using HDMR to investigate if the effective dimensions depend on the type of SPDE. In the following example, a stochastic natural convection problem in a closed cavity is considered.

For ease of analysis, we consider the same non-dimensional form of the governing equations (3.73)-(3.75) in Section 3.3.5. The physical domain is taken to be a closed cavity $[0,1]^2$. The Prandtl number Pr is set to 1.0 and the thermal Rayleigh number Ra is set to 1000. The deterministic governing equations are solved using the second-order stabilized projection finite element method developed in [63]. The spatial domain is discretized using 30×30 bilinear quadrilateral finite elements. Each deterministic simulation is performed until steady-state is reached. No slip boundary conditions are enforced on all four walls. The left wall is maintained at a higher mean temperature of 0.5. The temperature at different points on the left wall is correlated. This is physically feasible through, say, a resistance heater, where the mean temperature remains constant, but material variations cause local fluctuations from this mean temperature. We assume this correlation is a Gaussian correlation function:

$$\mathrm{Cov}(y_1, y_2) = \sigma^2 \exp\left(-\frac{|y_1 - y_2|^2}{L}\right), \tag{4.40}$$

with L being the correlation length and σ the standard deviation. The K-L expansion of the correlation for the input random temperature is performed:

$$\theta(y, \mathbf{Y}) = 0.5 + \sum_{i=1}^{N} \sqrt{\lambda_i}\phi_i(y)Y_i, \tag{4.41}$$

where the Y_i are assumed as i.i.d. uniform random variables on $[-1, 1]$.

We consider three different cases: $L = 0.1, \sigma = 0.05$, $L = 0.1, \sigma = 1.0$ and $L = 1.0, \sigma = 2.0$. The coefficients of variation, which is the ratio of the standard deviation and the mean, are 10%, 200% and 400%, respectively. The decay rate of the eigenvalues of this correlation is much faster than the exponential one of Eq. (5.58). Therefore, we truncate the expansion after 10 terms, i.e. the stochastic dimension is 10. The ratio between the smallest and largest eigenvalues is 10^{-6} and 10^{-14} for $L = 0.1, 1.0$, respectively. Thus, this problem is similar to the highly anisotropic case 2 discussed in Section 4.5.3. It is expected that the problem has a low effective dimensionality and a lower-order expansion may be accurate for large input variability. In addition, due to the highly anisotropic situation, the adaptive HDMR is expected to be most effective.

The mean and standard deviation of the v velocity component are shown in Fig. 4.18 when $L = 0.1$ and $\sigma = 0.05$. We choose $\varepsilon = 10^{-7}$ and $\theta_1 = 10^{-4}$. As expected, for such a small input variability, the construction of HDMR stopped after a first-order expansion even with $\theta_1 = 10^{-4}$. The results compare quite well with the reference solution which is from Monte Carlo simulation with 10^5 samples which shows that the first-order expansion is enough to capture the additive effects when each dimension is acting independently. Since the error of the mean converges very fast, we only show the convergence of error of the standard deviation with decreasing ε in Fig. 4.19. Only 261 collocation points are needed to achieve an error 1.69×10^{-3}, which is a significant computational saving in comparison to MC.

Next, we consider the case $\sigma^2 = 1.0$ and $L = 1.0$. For this case of high input variability, it is expected that higher-order terms are needed. We choose $\varepsilon =$

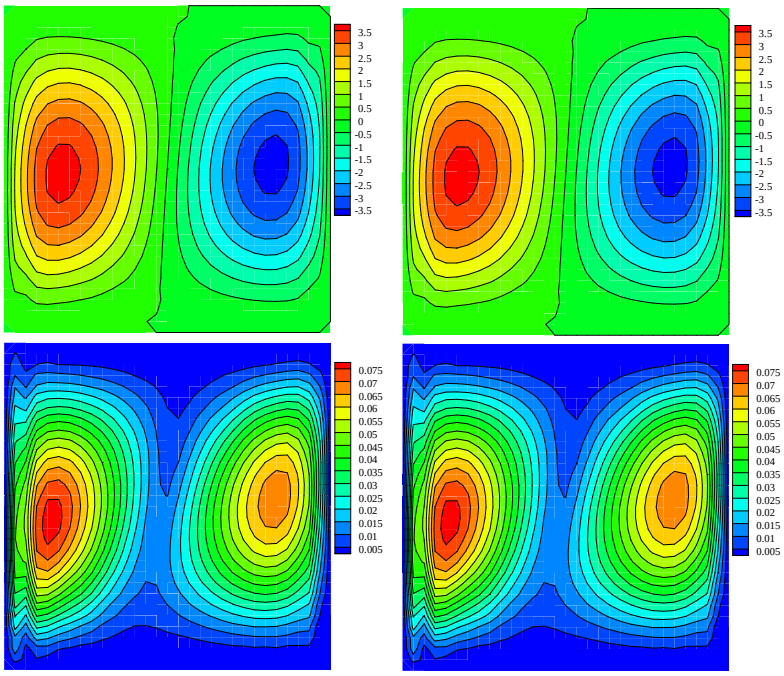


Figure 4.18: Mean (top row) and standard deviation (bottom row) of v velocity when $L = 0.1$ and $\sigma = 0.05$. Left column: HDMR solution, Right column: 10^5 Monte Carlo samples.

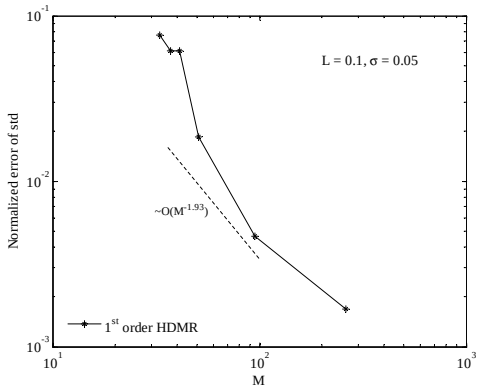


Figure 4.19: Convergence of the L_2 normalized errors of standard deviation of the v velocity-component where $L = 0.1$ and $\sigma = 0.05$.

10^{-5} and $\theta_1 = 10^{-4}$. The construction stopped after order 2. However, not all the terms are needed. Only 17 component functions are in the final HDMR expansion including only 6 second-order component functions which consist of only important dimensions. The total number of terms in conventional HDMR

is 56 which shows the effectiveness of the adaptive method. The mean and standard deviation of u velocity component are given in Fig. 4.20, which again compares well with the MC results. The error convergence is shown in Fig. 4.21. Only 1085 collocation points are needed to achieve an error 1.78×10^{-3}. It is noted that the convergence rate is a little slower than that of $\sigma = 0.05$, which is consistent with Fig. 4.16.

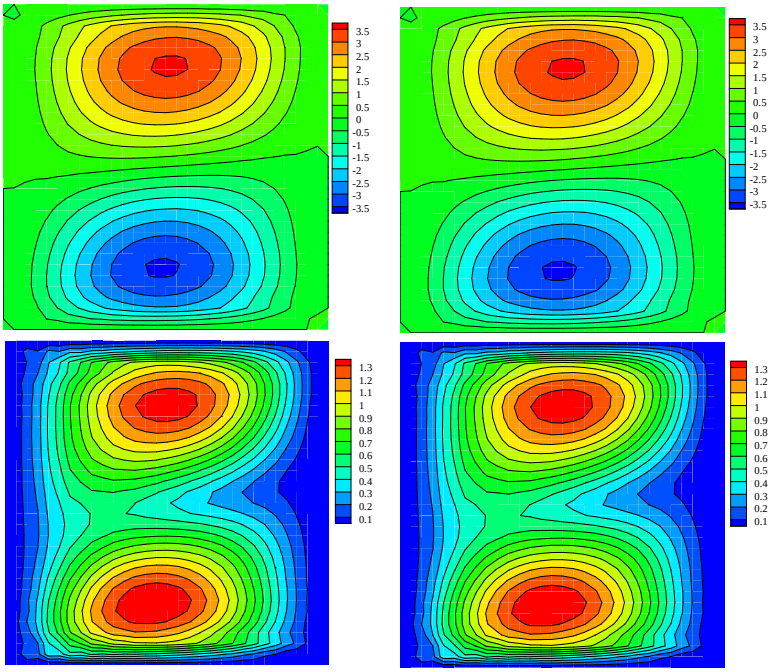


Figure 4.20: Mean (top row) and standard deviation (bottom row) of u velocity when $L = 0.1$ and $\sigma = 1.0$. Left column: HDMR solution, Right column: 10^5 Monte Carlo samples.

Finally, we consider the extreme case $L = 1.0$ and $\sigma = 2.0$. It is noted that this case may not be feasible in a physical sense. However, we still include it to demonstrate the applicability of this method. We choose $\varepsilon = 10^{-5}$ and $\theta_1 = 0.1$ such that only the first two dimensions are important. Only 356 collocation points are required to obtain a good comparison with that of MC results, which is shown in Fig. 4.22. The only second-order component function is f_{12}. By

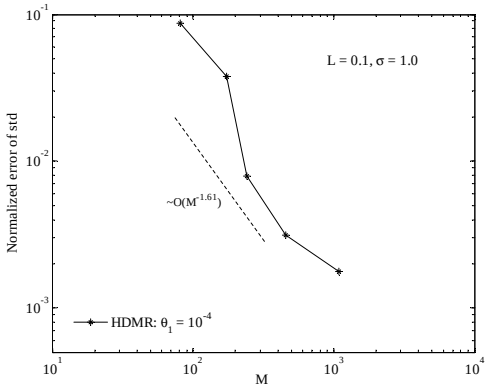


Figure 4.21: Convergence of the L_2 normalized errors of standard deviation of the u velocity-component where $L = 0.1$ and $\sigma = 1.0$.

adding this term to the first-order expansion, the error of temperature standard deviation decreases from 3.25×10^{-3} to 1.86×10^{-3}. This example clearly demonstrates the significance of including the important component function which captures the strong correlation effects among important dimensions. The error convergence is given in Fig. 4.23.

4.6 Conclusions

In this chapter, a novel adaptive high dimensional stochastic model representation technique for solving high-dimensional SPDEs is introduced. This method applies HDMR in the stochastic space and decomposes the original N-dimensional problem into several low-dimensional sub-problems. This has been shown to be more efficient than solving the N-dimensional problem directly. Each sub-problem is solved using ASGC, where a proper error indicator is introduced to adaptively refine locally the collocation points.

Numerical examples involving both elementary mathematical functions and

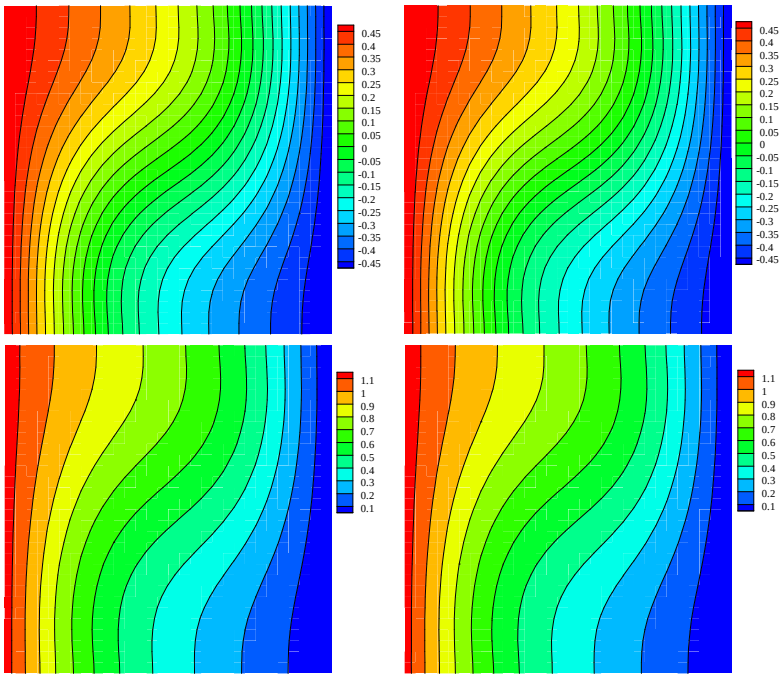


Figure 4.22: Mean (top row) and standard deviation (bottom row) of temperature when $L = 1.0$ and $\sigma = 2.0$. Left column: HDMR solution, Right column: 10^5 Monte Carlo samples.

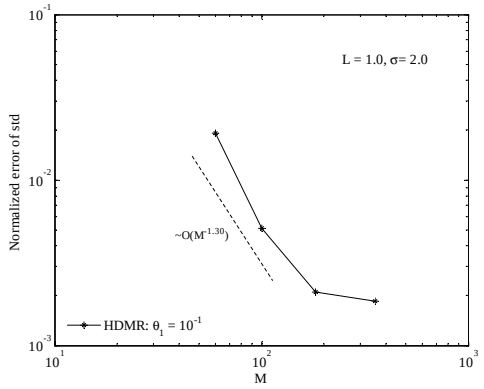


Figure 4.23: Convergence of the L_2 normalized errors of standard deviation of the temperature where $L = 1.0$ and $\sigma = 2.0$.

stochastic fluid mechanics problems have been conducted to verify the accuracy and efficiency of the proposed method. The numerical examples show that the number of component functions needed in the HDMR expansion for a fixed stochastic dimension depends both on the input variability and the important

dimensions no matter how many stochastic dimensions there are. However, it does not depend on the type of the function, i.e., whether the SPDE is linear or nonlinear. Although, in general, lower-order expansion is enough to capture all the input uncertainty to a good accuracy, the number of component functions increases quickly with the dimension. It is impossible to calculate all the terms through conventional HDMR in the case of extremely high-stochastic dimension. However, by using the adaptive version of HDMR, we can effectively solve the problem within a desired accuracy even for problems with high-dimensional high input variability. On the other hand, for small variability, the first-order expansion is accurate enough. Therefore, the HDMR is quite suitable in most real physical applications for simulating high-dimensional stochastic problems in the area of uncertainty quantification where the correlation length is small and the input variability is generally large and solving the N-dimensional problem directly by ASGC is not feasible. It is also shown that the convergence rate of the proposed method is even better than that of MC for the problems considered up to 500 stochastic dimensionality.

Finally, we note that the proposed method is more computationally efficient than applying the ASGC method directly on the function of interest or to represent the solution of an SPDE. This is due mainly to the fact that implementation of HDMR requires less memory storage and avoids the heavy surplus calculation typical of high-dimensional problems. However, it is emphasized that HDMR may not be of practical use for interpolating arbitrary mathematical functions where sometimes all 2^N terms might be required, for example, in the case of a function with multiplicative nature.

CHAPTER 5

A STOCHASTIC MIXED FINITE ELEMENT HETEROGENEOUS MULTISCALE METHOD FOR FLOW IN POROUS MEDIA

In this chapter, a computational methodology is developed to efficiently perform uncertainty quantification for fluid transport in porous media in the presence of both stochastic permeability and multiple scales. In order to capture the small scale heterogeneity, a new mixed multiscale finite element method is developed within the framework of the heterogeneous multiscale method (HMM) in the spatial domain. This new method ensures both local and global mass conservation. Starting from a specified covariance function, the stochastic log-permeability is discretized in the stochastic space using a truncated Karhunen-Loève expansion with several random variables. Due to the small correlation length of the covariance function, this often results in a high stochastic dimensionality. Therefore, HDMR is used in the stochastic space. This results in a set of low stochastic dimensional subproblems which can efficiently solved using the ASGC. This chapter closely follows the work in [72].

5.1 Introduction

Flow through porous media is ubiquitous, occurring from large geological scales down to microscopic scales. Several critical engineering phenomena like contaminant spread, nuclear waste disposal and oil recovery rely on accurate analysis and prediction of these multiscale phenomena. Such analysis is complicated by heterogeneities at various length scales as well as inherent uncertainties. For this reason, to predict the flow and transport in stochastic porous

media, some type of stochastic upscaling or coarsening is needed for computational efficiency by solving these problems on a coarse grid. However, most of the existing multiscale methods are realization based, i.e. they can only solve a deterministic problem for a single realization of the stochastic permeability field. This is not sufficient for uncertainty quantification since we are mostly interested in the statistics of the flow behavior, such as mean and standard deviation. In this paper, we propose a stochastic multiscale approach which resolves both uncertainties and subgrid scales by developing a new multiscale method and adopting a newly developed adaptive high dimensional stochastic model representation technique (HDMR). The goal of the multiscale method is to coarsen the flow equations spatially whereas HDMR is used to address the curse of dimensionality in high dimensional stochastic spaces.

One of the challenging mathematical issues in the analysis of transport through heterogeneous random media is the multiscale nature of the property variations. Complete response evaluation involving full-scale spatial and temporal resolution simulations of multiscale systems is extremely expensive. Computational techniques have been developed that solve for an appropriate coarse-scale problem that captures the effect of the subgrid-scales. The most popular techniques developed for such upscaling fall under the category of multiscale methods viz. the multiscale finite element (MsFEM) method [73, 74], the variational multiscale (VMS) method [75, 76] and the heterogeneous multiscale (HMM) method [58, 59]. The MsFEM was originally developed in [73, 74] for the solution of elliptic equation based problems with multiscale coefficients using conforming linear finite elements. The primal unknown is the nodal value, e.g. the pressure, and one can obtain the velocity by calculating the gradient of the pressure filed given the finite element solution. The result is

generally not accurate and conservation of the flux in each element may be violated, which is an important property for the numerical solution of transport equations in porous media. Therefore, a mixed multiscale finite element method (MMsFEM) that guarantees the local mass conservation at the element level was proposed in [77] using the lowest-order Raviart-Thomas mixed finite element [78]. The basic idea of the method is to construct the multiscale finite element basis functions that incorporate the small scale information through the solution of a local problem in each element and couple them through a global formulation of the problem. However, this work only produces a globally mass conserving velocity field. Later on this work was extended in a number of important ways to guarantee mass conservation on both fine- and coarse-scales [79, 80]. A similar framework utilizing the finite volume method as the global solver was also proposed in [81, 82, 83], which also preserves mass conservation at both scales. The basic idea of the VMS method is to invoke a multiscale split of the solution into a coarse-scale part and a subgrid component. The variational coarse-scale problem is performed and solved using the solution of the localized subgrid problem. Parallel to MMsFEM, a mixed finite element version of VMS was also proposed in [84, 85, 86], which is often called "Numerical subgrid upscaling". A thorough comparison of the above three methods for elliptic problems in porous media flows can be found in [87].

Unlike the MsFEM which was built on the finite element method (FEM), the HMM is a more general methodology for multiscale PDEs (see [59] for a review). The basis idea of HMM consists of two components: selection of a macroscopic solver and estimating the needed macroscale data by solving locally the fine-scale problem. It allows two different sets of governing equations on macro- and micro-scales, e.g. atomistic simulation on micro-scale and continuum simula-

tion on macro-scale [88, 89]. Later on this framework was utilized to solve multiscale elliptic problems with the conforming linear FEM (FeHMM) [90, 91, 92]. The method was analyzed in a series of papers [93, 94, 95]. However, unlike the MMsFEM, there is no discussion of the mixed version of FeHMM except the work in [96], where the author first developed the theory of the mixed finite element version of HMM for the elliptic problem and proved the stability and convergence of this new method. However, the primitive idea in [96] is only a simple extension to the original theory of HMM which in general is not suitable for realistic problems such as flow through porous media. In addition, no numerical implementation was given in [96]. Motivated by the work in [96], in this paper, we first develop and implement the mixed finite element version of HMM with application to flow transport in heterogeneous porous media, which we will call it mixed heterogeneous multiscale method (MxHMM).

However, all of the above mentioned multiscale analyses of such systems inherently assume that the complete multiscale variation of the permeability is known. This assumption limits the applicability of these frameworks since it is usually not possible to experimentally determine the complete structure of the medium at small scales. One way to cope with this difficulty is to view the permeability variation as a random field that satisfies certain statistical correlations. This naturally results in describing the physical phenomena using stochastic partial differential equations (SPDEs). Therefore, in this chapter, we will use both of these developments developed in the last two chapters in the stochastic space together with the newly developed MxHMM for the spatial discretization.

There exist several new stochastic multiscale methods for elliptic problems.

In [97] and [98], the variational multiscale method was extended to a stochastic version using gPC and stochastic collocation method respectively to solve a simple diffusion problem. The stochastic multiscale finite element was also developed in [99] however only an elliptic problem was solved to find the hydraulic head. More related work can be found in [71, 100, 101]. In [71], the stochastic numerical subgrid upscaling method was also developed for the solution of the mixed form of the Darcy's equation using the stochastic collocation method. However, in that work, only the statistics of the coarse-scale velocity and pressure were solved and no flow transport problem was investigated. In [100], a projection method for the solution of stochastic mixed multiscale finite element method was introduced where the velocity solution was projected onto the a set of multiscale velocity basis which are precomputed from an arbitrary number of random realizations. It generally involves the solution of a big linear system of equations to find the projection coefficients if the realization number is large. For each new permeability sample, this method needs to solve one coarse-scale problem again and is generally computationally expensive. In addition, this method cannot give the statistics of the saturation directly and thus was not reported in their work. In [101], this framework was used to sample the permeability given measurements within the Markov chain Monte Carlo method (MCMC) framework and again no statistics of the saturation were reported. However, in real reservoir engineering, we are primarily interested in mean behavior and a measure of uncertainty, e.g. standard deviation, in the saturation of each phase. By using the adaptive HDMR and ASGC developed in the last two chapters, we can obtain not only a surrogate model for the saturation profile but also can extract the statistics of the saturation easily. Therefore, the novel contributions of this chapter are as follows: (1) We develop a new

mixed finite element version of the heterogeneous multiscale method for the simulation of flow through porous media in the spatial domain; (2) We utilize the newly developed HDMR technique to address the *curse of dimensionality* that occurs naturally in this problem due to the heterogeneity of the permeability; (3) Finally, we investigate the effect of the stochastic permeability on various statistics of the saturation using the recently developed adaptive HDMR method.

5.2 Problem definition

In this section, we follow the notation in Chapter 2. Let us define a complete probability space $(\Omega, \mathcal{F}, \mathcal{P})$ with sample space Ω which corresponds to the outcomes of some experiments, $\mathcal{F} \subset 2^{\Omega}$ is the σ-algebra of subsets in Ω and $\mathcal{P} : \mathcal{F} \to [0,1]$ is the probability measure. Also, let us define D as a d-dimensional bounded domain $D \subset \mathbb{R}^d$ $(d = 2, 3)$ with boundary ∂D. The governing equations for immiscible and incompressible two-phase flow in porous media consists of an elliptic equation for fluid pressure and a transport equation for the movement of fluid phases. For simplicity, we will neglect the effects from gravity, capillary forces and assume that the porosity is a constant. The two phases will be referred to as water and oil, denoted as w and o, respectively. The total Darcy velocity $\boldsymbol{u}$ and the pressure p satisfy for $\mathcal{P}$-almost everywhere (a.e.) in Ω the following SPDEs [87]

$$\nabla \cdot \boldsymbol{u} = \bar{q}, \quad \boldsymbol{u} = -K(\boldsymbol{x}, \omega)\lambda_t \nabla p, \ \ \forall \boldsymbol{x} \in D, \tag{5.1}$$

with the following boundary conditions

$$p = \bar{p} \ \text{ on } \ \partial \mathrm{D_p}, \quad \boldsymbol{u} \cdot \boldsymbol{n} = \bar{\boldsymbol{u}} \ \text{ on } \ \partial \mathrm{D_u}. \tag{5.2}$$

The total velocity $\boldsymbol{u} = \boldsymbol{u}_o + \boldsymbol{u}_w$ is a sum of the velocities of oil $\boldsymbol{u}_o$ and water $\boldsymbol{u}_w$. $\bar{q}$ is a volumetric source terms which is assumed 0 throughout this chapter. The random permeability tensor K is assumed to be diagonal and uniformly positive definite. In addition, we will assume K is a stochastic scalar function. The total mobility is given by $\lambda_t = \lambda_w + \lambda_o$, where λ_i models the reduced mobility of phase i due to the presence of the other phase. Without loss of generality, we assume that the boundary conditions are deterministic and that the Neumman condition is homogeneous, $\bar{\boldsymbol{u}} = 0$ on ∂D_u.

Furthermore, to assess the quality of the multiscale model, the unit mobility ratio displacement model is used, i.e. $\lambda_w = S$, $\lambda_o = 1 - S$ and hence $\lambda_t = 1$, where S is the water saturation. Under this assumption, the water saturation equation is given by

$$\frac{\partial S(\boldsymbol{x}, t, \omega)}{\partial t} + \boldsymbol{u} \cdot \nabla S(\boldsymbol{x}, t, \omega) = 0, \quad \forall \boldsymbol{x} \in D, t \in [0, T]. \tag{5.3}$$

Since the permeability K is a stochastic function, all the unknowns p, $\boldsymbol{u}$ and S are also stochastic. Therefore, our complete stochastic model is: Find stochastic functions $\boldsymbol{u} : \Omega \times D \to \mathbb{R}$, $p : \Omega \times D \to \mathbb{R}$ and $S : \Omega \times [0, T] \times D \to \mathbb{R}$ for $\mathcal{P}$-almost everywhere (a.e.) $\omega \in \Omega$ such that the following equations hold:

$$\nabla \cdot \boldsymbol{u}(\boldsymbol{x}, \omega) = 0, \quad \boldsymbol{u}(\boldsymbol{x}, \omega) = -K(\boldsymbol{x}, \omega) \nabla p(\boldsymbol{x}, \omega) \ \ \forall \boldsymbol{x} \in D, \tag{5.4}$$

$$\frac{\partial S(\boldsymbol{x}, t, \omega)}{\partial t} + \boldsymbol{u}(\boldsymbol{x}, t, \omega) \cdot \nabla S(\boldsymbol{x}, t, \omega) = 0, \quad \forall \boldsymbol{x} \in D, t \in [0, T], \tag{5.5}$$

with the boundary conditions

$$p = \bar{p} \ \text{ on } \ \partial \mathrm{D_p}, \quad \boldsymbol{u} \cdot \boldsymbol{n} = 0 \ \text{ on } \ \partial \mathrm{D_u}, \tag{5.6}$$

together with appropriate initial and boundary conditions for S. Computation with this model is much more efficient than using the actual two-phase

flow model because the pressure and saturation equations are effectively decoupled. Throughout this work, the Darcy velocity $\boldsymbol{u}$ is first computed using the mixed finite element heterogeneous multiscale method developed in Section 5.3.1 and then the saturation equation is solved using a upwinding finite element scheme [102] in Section 5.3.2. Although these equations differ from the actual flow equations, they do capture many important aspects of two-phase flow problems. Specifically, the effects of the heterogeneity are often similar in the unit mobility and two-phase flow problems [103].

Geostatistical models often suggest that the permeability field is a weakly or second-order stationary random field such that the mean log-permeability is constant and its covariance function only depends on the relative distance of two points rather than their actual location [77]. Denote $G(\boldsymbol{x}, \omega) = \log(\mathrm{K})$ and its covariance function by $R_G(\boldsymbol{x}_1, \boldsymbol{x}_2)$, where $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are spatial coordinates. Then the truncated K-L expansion takes the following form:

$$G(\boldsymbol{x}, \omega) = \mathbb{E}[G(\boldsymbol{x})] + \sum_{i=1}^{N} \sqrt{\lambda_i}\phi_i(\boldsymbol{x})Y_i(\omega), \tag{5.7}$$

where $\{Y_i(\omega)\}_{i=1}^N$ are uncorrelated random variables. Also, $\phi_i(\boldsymbol{x})$ and λ_i are the eigenfunctions and eigenvalues of the correlation function, respectively.

When using the K-L expansion, we here assume that we obtain a set of mutually independent random variables. Denote the probability density functions of $\{Y_i(\omega)\}_{i=1}^N$ as ρ_i, $i = 1, \ldots, N$. Let Γ_i be the image of Y_i. Then $\rho(\boldsymbol{Y}) = \prod_{i=1}^N \rho_i(Y_i)$ is the joint probability density of $\boldsymbol{Y} = (Y_1, \cdots, Y_N)$ with support $\Gamma \equiv \Gamma_1 \times \Gamma_2 \times \cdots \times \Gamma_N \in \mathbb{R}^N$. Then the stochastic log permeability can be represented by $G(\boldsymbol{x}, \omega) = G(\boldsymbol{x}, Y_1, \ldots, Y_N) = G(\boldsymbol{x}, \boldsymbol{Y})$.

5.2.1 Stochastic variational formulation

By using the Doob-Dynkin lemma [61], the solutions of Eqs. (5.4) and (5.5) can be described by the same set of random variables $\{Y_i(\omega)\}_{i=1}^N$. Following [71], we define appropriate function spaces that encode variations of the function in the physical domain $\mathcal{D}$ and in the stochastic space Γ.

In the physical space, we introduce the following common functional spaces [86, 71]:

$$W \equiv L^2(D) = \left\{ p : \int_D |p|^2 d\boldsymbol{x} = \| p \|^2_{L^2(D)} < +\infty \right\}, \tag{5.8}$$

with inner product

$$(p, q) \equiv (p, q)_{L^2(D)} := \int_D p\, q\, d\boldsymbol{x}, \quad p, q \in L^2(D), \tag{5.9}$$

and

$$H(\mathrm{div}, \mathrm{D}) = \left\{ \boldsymbol{u} : \boldsymbol{u} \in (\mathrm{L}^2(\mathrm{D}))^2, \nabla \cdot \boldsymbol{u} \in \mathrm{L}^2(\mathrm{D}) \right\}, \tag{5.10}$$

with inner product

$$(\boldsymbol{u}, \boldsymbol{v}) \equiv (\boldsymbol{u}, \boldsymbol{v})_{H(\mathrm{div},\mathrm{D})} := \int_D \boldsymbol{u} \cdot \boldsymbol{v}\, d\boldsymbol{x}, \quad \boldsymbol{u}, \boldsymbol{v} \in H(\mathrm{div}, \mathrm{D}). \tag{5.11}$$

We will also make use of the following space:

$$V \equiv H_{0,u}(\mathrm{div}, \mathrm{D}) = \{ \boldsymbol{u} : \boldsymbol{u} \in \mathrm{H}(\mathrm{div}, \mathrm{D}), \boldsymbol{u} \cdot \boldsymbol{n} = 0 \}. \tag{5.12}$$

The duality product is defined as:

$$\langle \bar{u}, \bar{p} \rangle \equiv \langle \bar{u}, \bar{p} \rangle_{\partial D_p} := \int_{\partial D_p} \bar{u}\, \bar{p}\, dx, \quad \bar{u} \in H^{1/2}(D), \quad \bar{p} \in H^{-1/2}(D). \tag{5.13}$$

The functional space in Γ is defined as follows:

$$\mathcal{U} \equiv L^2_\rho(\Gamma) = \left\{ p : \left(\int_\Gamma |p(\boldsymbol{Y})|^2 \rho(\boldsymbol{Y}) d\boldsymbol{Y} \right)^{1/2} < \infty \right\}. \tag{5.14}$$

By taking its tensor product with the previous deterministic spaces, one can form the stochastic functional spaces:

$$\mathcal{W} = \mathcal{U} \otimes W, \;\; \mathcal{V} = \mathcal{U} \otimes V. \tag{5.15}$$

Multiplication of Eqs. (5.4) and (5.5) by appropriate test functions and integration by parts leads to the following weak formulations: Find $\boldsymbol{u} \in \mathcal{H}$, $p \in \mathcal{W}$ such that

$$\begin{aligned} \int_\Gamma (K^{-1}\boldsymbol{u}, \boldsymbol{v})\rho(\boldsymbol{Y})d\boldsymbol{Y} \;\; &- \;\; \int_\Gamma (\nabla \cdot \boldsymbol{v}, p)\rho(\boldsymbol{Y})d\boldsymbol{Y} \\ &= \;\; -\int_\Gamma \langle \boldsymbol{v} \cdot \boldsymbol{n}, \bar{p}\rangle \rho(\boldsymbol{Y})d\boldsymbol{Y}, \;\; \forall \boldsymbol{v} \in \mathcal{V}, \end{aligned} \tag{5.16}$$

$$\int_\Gamma (l, \nabla \cdot \boldsymbol{u})\rho(\boldsymbol{Y})d\boldsymbol{Y} \;\; = \;\; 0, \;\; \forall l \in \mathcal{W}, \tag{5.17}$$

and $S \in \mathcal{W}$ for each $t \in [0, T]$ such that

$$\int_\Gamma \left(\frac{\partial S}{\partial t}, q\right)\rho(\boldsymbol{Y})d\boldsymbol{Y} + \int_\Gamma (\boldsymbol{u} \cdot \nabla S, q)\rho(\boldsymbol{Y})d\boldsymbol{Y} \;\; = \;\; 0, \;\; \forall q \in \mathcal{W}. \tag{5.18}$$

Again we assume without loss of generality that the support of the random variables Y_i is $\Gamma^i = [0, 1]$ for $i = 1, \cdots, N$ and thus the bounded stochastic space is a N-hypercube $\Gamma = [0, 1]^N$, since any bounded stochastic space can always be mapped to the above hypercube.

5.2.2 Stochastic discretization by HDMR

The original infinite-dimensional stochastic problem is now restated as a finite N-dimensional problem. Then we can apply any stochastic method in the random space and the resulting equations become a set of deterministic equations in the physical space that can be solved by any standard deterministic discretization technique, e.g. the finite element method. The solution to the above

SPDEs Eqs. (5.16)-(5.18) can be regarded as stochastic functions taking real values in the stochastic space Γ. For example, we can consider the pressure as a stochastic function $p : \Gamma \to \mathbb{R}$ and we use the notation $p(\boldsymbol{Y})$ to highlight the dependence on the randomness. Then it can be shown that the weak formulation Eqs. (5.16)-(5.18) is equivalent to [23]: for a.e. $\rho \in \Gamma$ the following deterministic weak form equations hold:

$$
\begin{aligned}
(K^{-1}\boldsymbol{u}, \boldsymbol{v}) - (p, \nabla \cdot \boldsymbol{v}) &= -\langle \bar{p}, \boldsymbol{v} \cdot \boldsymbol{n} \rangle, \quad \forall \boldsymbol{v} \in V && (5.19)\\
(l, \nabla \cdot \boldsymbol{u}) &= 0, \quad \forall l \in W && (5.20)\\
\left(\frac{\partial S}{\partial t}, q\right) + (q, \boldsymbol{u} \cdot \nabla S) &= 0, , \quad \forall q \in W && (5.21)
\end{aligned}
$$

This nature is utilized by the stochastic collocation method which constructs the interpolant of the stochastic function in Γ using only the solutions to the above deterministic problems at chosen sample points. Thus, we can simply apply HDMR in the stochastic space and the only thing we need is the deterministic multiscale solver.

5.3 Spatial finite element discretization

As stated in Section 5.2.2, in order to utilize the HDMR, we only need to seek the solution $(\boldsymbol{u}, p, S)$ at each collocation point in the stochastic space Γ. In other words, our goal is reduced to: for each permeability realization $K^{(i)}(\boldsymbol{x}) = K(\boldsymbol{x}, \boldsymbol{Y}_i), i = 1, \ldots, M$, we solve the deterministic problem: find $\boldsymbol{u}^{(i)} \in V$, $p^{(i)} \in W$ and $S^{(i)} \in W$ such that for $i = 1, \ldots, M$

$$
\begin{aligned}
(K^{-1}\boldsymbol{u}^{(i)}, \boldsymbol{v}) - (p^{(i)}, \nabla \cdot \boldsymbol{v}) &= -\langle \bar{p}, \boldsymbol{v} \cdot \boldsymbol{n} \rangle, \quad \forall \boldsymbol{v} \in V, && (5.22)\\
(l, \nabla \cdot \boldsymbol{u}^{(i)}) &= 0, \quad \forall l \in W, && (5.23)
\end{aligned}
$$

$$\left(\frac{\partial S^{(i)}}{\partial t}, q\right) + (q, \boldsymbol{u}^{(i)} \cdot \nabla S^{(i)}) \quad = \quad 0, \quad \forall q \in W. \tag{5.24}$$

In this section, mixed finite element methods are introduced to solve the above equations in the spatial domain. Since the pressure Eqs. (5.22) and (5.23) are effectively decoupled from the saturation Eq. (5.24), we will first introduce the multiscale method to find $\boldsymbol{u}, p$ and then use the upwinding finite element method to find S. To simplify the notation, we will omit the superscript i and assume the deterministic equations are satisfied for an arbitrary permeability sample in the stochastic space.

5.3.1 Mixed finite element heterogeneous multiscale method (MxHMM)

In the porous media flow problem, the heterogeneity of the permeability field will have a great impact on the global flow conditions. In order to resolve the fine-scale velocity accurately with lower computational cost, a multiscale method is needed. In addition, the mixed finite element method is also required to compute the velocity and pressure simultaneously, if we want to have an accurate velocity and ensure mass conservation. We can identify at least two main multiscale methods: multiscale finite element or finite volume methods [77, 79, 81] and the variational multiscale methods [84, 86]. In this section, we will develop a new multiscale method which is based on the framework of the heterogeneous multiscale method [92]. We present the discretization and methodology for a two-dimensional system. Extension to three-dimensions is straightforward.

Consider a partition, $\mathcal{T}_h$ for the domain D into non-overlapping elements e_i, $\mathcal{T}_h = \bigcup_{i=1}^{N_h} e_i$, where N_h is the number of elements of the grid. Define also the skeleton of the partition, $\mathcal{SP}_h = \bigcup_{a=1}^{M_h} \nu_a$, where M_h is the number of element faces denoted by ν_a. The partition $\mathcal{T}_h$ is regarded as the fine-scale grid. The multiscale permeability is defined as a cell-wise constant on this grid. To implement the multiscale method, we also consider a coarse-scale partition of the same domain D. Denote this partition as $\mathcal{T}_c = \bigcup_{i=1}^{N_c} E_i$. Denote by $\mathcal{SP}_c = \bigcup_{a=1}^{M_c} \Lambda_a$ the associated skeleton of the coarse-scale discretization. Here, N_c is the number of coarse elements and M_c is the number of coarse element faces denoted by Λ_a. In order to conserve the mass at the coarse-scale, we also assume for simplicity that the partitions $\mathcal{T}_h$ and $\mathcal{T}_c$ are nested, conforming and consist of rectangular elements. Fig. 5.1 shows a fine grid (finer lines) and a corresponding coarse grid (heavier lines).

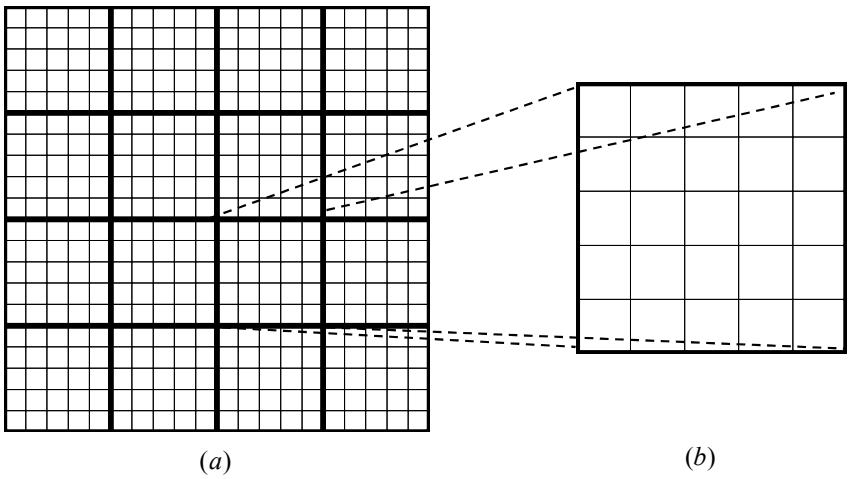


Figure 5.1: Schematic of the domain partition: (a) fine- and coarse-scale grids, (b) fine-scale local region in one coarse element.

Now consider the finite dimensional subspaces on the coarse-scale $V_c \in V$ and $W_c \in W$. The mixed finite element method approximation of Eqs. (5.22)-(5.23) on the coarse-scale reads: Find the coarse-scale $(\boldsymbol{u}_c, p_c) \in V_c \times W_c$ such

that

$$(K^{-1}\boldsymbol{u}_c, \boldsymbol{v}_c) - (p_c, \nabla \cdot \boldsymbol{v}_c) = -\langle p_0, \boldsymbol{v}_c \cdot \boldsymbol{n} \rangle, \quad \forall\, \boldsymbol{v}_c \in V_c, \tag{5.25}$$

$$(l_c, \nabla \cdot \boldsymbol{u}_c) = 0, \quad \forall\, l_c \in W_c. \tag{5.26}$$

Note that V_c and W_c should satisfy the discrete inf-sup condition [104]. In this work, V_c is taken to be the lowest-order Raviart-Thomas space [78], $RT_0(\mathcal{T}_c)$ and W_c is taken to be the space of piece-wise constants on the coarse-scale mesh, $P_0(\mathcal{T}_c)$. Other choices can be found in [104]. Therefore, we define the finite element space for the coarse-scale velocity as:

$$V_c = \left\{ \boldsymbol{u}_c : \boldsymbol{u}_c = \sum_{a=1}^{M_c} \boldsymbol{\psi}_a^c u_a^c, \quad u_a^c = 0, \quad \forall\, \Lambda_a \in \partial D_u \right\}, \tag{5.27}$$

where $\boldsymbol{\psi}_a^c$ is the RT_0 basis functions on the uniform mesh of rectangular elements associated with the coarse element face Λ_a. For a reference element $E = [x_1^L, x_1^R] \times [x_2^L, x_2^R]$ with the area $|E|$, there are four vector RT_0 basis functions with non-zero support:

$$\boldsymbol{\psi}_1^c = \left[(x_1^R - x_1)/(x_1^R - x_1^L), 0\right]^T, \quad \boldsymbol{\psi}_2^c = \left[0, (x_2^R - x_2)/(x_2^R - x_2^L)\right]^T, \tag{5.28}$$

$$\boldsymbol{\psi}_3^c = \left[(x_1 - x_1^L)/(x_1^R - x_1^L), 0\right]^T, \quad \boldsymbol{\psi}_4^c = \left[0, (x_2 - x_2^L)/(x_2^R - x_2^L)\right]^T. \tag{5.29}$$

The basis functions satisfy the properties such that $\boldsymbol{\psi}_i^c \cdot \boldsymbol{n}_j = 1$ if $i = j$, otherwise $\boldsymbol{\psi}_i^c \cdot \boldsymbol{n}_j = 0$ for $i, j = 1, \ldots, 4$. Therefore u_a^c is value of the coarse-scale flux at the middle point of the side Λ_a, i.e. $\boldsymbol{u}_c \cdot \boldsymbol{n}_a = u_a^c$, where $\boldsymbol{n}_a$ is the unit outer normal to the interface Λ_a. The coarse-scale pressure approximation is piecewise constant on the coarse-mesh and $P_0(\mathcal{T}_c)$ is

$$W_c = \left\{ p_c : p_c = \sum_{a=1}^{N_c} \phi_i^c p_i^c \right\}, \tag{5.30}$$

where ϕ_i^c is the coarse-scale pressure basis function for the coarse element i defined as

$$\phi_i^c(\boldsymbol{x}) = \begin{cases} 1, & \text{if } \boldsymbol{x} \in E_i, \\ 0, & \text{if } \boldsymbol{x} \notin E_i. \end{cases} \tag{5.31}$$

p_i^c is the corresponding pressure degree of freedom (the average pressure in coarse element E_i).

It is obvious that all the fine-scale information is included in the bilinear form $(K^{-1}\boldsymbol{u}_c, \boldsymbol{v}_c)$. Denote $A = (A_{ij})$ the global matrix for the bilinear form, where

$$A_{ij} = \int_D \boldsymbol{\psi}_i^c(\boldsymbol{x}) \cdot K^{-1}(\boldsymbol{x}) \boldsymbol{\psi}_j^c(\boldsymbol{x}) \, d\boldsymbol{x}, \tag{5.32}$$

We could evaluate Eq. (5.32) by the 2×2 Gauss quadrature rule: let

$$f_{ij}(\boldsymbol{x}) = \boldsymbol{\psi}_i^c(\boldsymbol{x}) \cdot K^{-1}(\boldsymbol{x}) \boldsymbol{\psi}_j^c(\boldsymbol{x}), \tag{5.33}$$

then

$$A_{ij} = \int_D f_{ij} \, d\boldsymbol{x} \simeq \sum_{E \in \mathcal{T}_c} \sum_{\boldsymbol{\xi}_k \in E} \tau_k f_{ij}(\boldsymbol{\xi}_k), \tag{5.34}$$

where $\boldsymbol{\xi}_k$ and $\tau_k, k = 1, \ldots, 4$ are the quadrature points and weights (including the determinant of the Jacobian matrix) in the coarse element E, respectively. It is obvious that any realization of the permeability field at the quadrature point $K(\boldsymbol{\xi}_k)$ is not able to capture the full information at the subgrid scale in the coarse element since the size of the coarse element is much larger than the characteristic length scale of the multiscale permeability field. Therefore, we need to modify the bilinear form Eq. (5.33) at the quadrature point $\boldsymbol{\xi}_k$ following the framework of the heterogeneous multiscale method [91, 96] as:

$$f_{ij}(\boldsymbol{\xi}_k) = \frac{1}{|E_{\delta_k}|} \int_{E_{\delta_k}} \tilde{\boldsymbol{u}}_{ik}(\boldsymbol{x}) \cdot K^{-1} \tilde{\boldsymbol{u}}_{jk}(\boldsymbol{x}) \, d\boldsymbol{x}, \quad k = 1, \ldots, 4, \tag{5.35}$$

where $\tilde{\boldsymbol{u}}_{ik}(\boldsymbol{x}), i = 1, \ldots, 4$ is the solution to the following local subgrid problem in the sampling domain $E_{\delta_k} \subset E, k = 1, \ldots, 4$:

$$\nabla \cdot \tilde{\boldsymbol{u}}_{ik}(\boldsymbol{x}) = 0, \quad \tilde{\boldsymbol{u}}_{ik}(\boldsymbol{x}) = -K \nabla \tilde{p}_{ik}(\boldsymbol{x}), \quad \forall \, \boldsymbol{x} \in E_{\delta_k}, \tag{5.36}$$

with appropriate boundary conditions which we will discuss below. $\tilde{p}_{ik}(\boldsymbol{x})$ can be considered as the subgrid pressure.

First, we will discuss the choice of the sampling domain E_{δ_k} of the subgrid problem. In the original problem definition of the FeHMM [91, 96], the coefficient of the elliptic equation (here K) is assumed to be periodic. Therefore, the sampling domain was taken around each quadrature point as $E_{\delta_k} = \boldsymbol{\xi}_k + \delta I$, where $I = (-1/2, 1/2)^2$ and δ is equal to one period of the coefficient in the elliptic equation, as in Fig. 5.2(a). However, in general, the permeability is not periodic. If the sampling domain is too small, one cannot capture enough information on the subgrid scale. According to the numerical results in [105], the larger the size of the sampling domain is, the more accurate the computed result is. Therefore, we would like to take the sampling domain to be the same as the coarse element, i.e. $E_{\delta_k} = E, \partial E_{\delta_k} = \partial E$ as in Fig. 5.2(b). In addition, we also assume that the fine grid within each coarse element is the same as the fine-scale grid $\mathcal{T}_h$, where the permeability is defined, see Fig. 5.1(b). In this way, we can ensure global continuity of the flux across the coarse element.

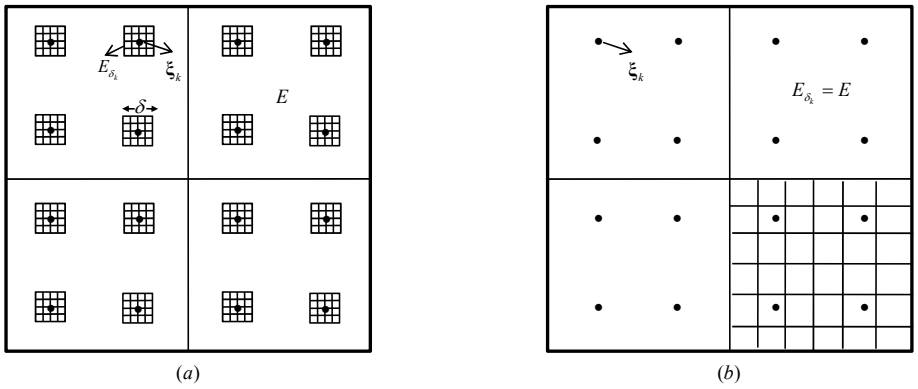


Figure 5.2: (a) Schematic of the original HMM method, where the sampling domain is around the quadrature point. (b) Schematic of the proposed MxHMM method, where the sampling domain is the same as the coarse element.

Remark 1. Unlike the mixed multiscale finite element method [79], where the subgrid problem is limited to the coarse element, the advantage of the heteroge-

neous multiscale method here is that the sampling domain is not limited to the coarse element. In fact, it can be chosen arbitrarily to include as many coarse elements as necessary. However, in the present work we still solve the subgrid problem in only one coarse element. The effect of the size of the sampling domain is reserved for later work.

Hence, all the subgrid problems are solved within the same coarse element. The only difference is the applied boundary condition. The boundary condition of the problem in Eq. (5.36) plays a significant role in the accuracy of the multiscale method as discussed in [105], where three different boundary conditions are considered: the periodic boundary condition, Dirichilet boundary condition, and the Neumann boundary condition. However, when mixed finite element formulation is used on the coarse-scale, only the Neumann boundary condition is applicable here. In [96], the following Neumann boundary condition is proposed:

$$\tilde{\boldsymbol{u}}_{ik} \cdot \boldsymbol{n}_{\partial E} = \boldsymbol{\psi}_i^c(\boldsymbol{\xi}_k) \cdot \boldsymbol{n}_{\partial E}, \quad \text{on } \partial \text{E}, \tag{5.37}$$

where $\boldsymbol{\psi}_i^c(\boldsymbol{\xi}_k)$ denotes the value of the i-th coarse-scale RT_0 finite element basis function at the quadrature point $\boldsymbol{\xi}_k, k = 1, \ldots, 4$ and $\boldsymbol{n}_{\partial E}$ denotes the unit outer normal of the coarse element boundary ∂E. According to the definition of RT_0 basis function in Eqs. (5.28)-(5.29), this boundary condition applies a uniform flow with magnitude $\boldsymbol{\psi}_i^c(\boldsymbol{\xi}_k)$ from one side to the opposite side while keeping no-flow conditions on the other two sides. The example of $\boldsymbol{\psi}_1(\boldsymbol{\xi}_1)$ is shown in Fig. 5.3. However, this boundary condition only reflects the local heterogeneity structure within the current coarse element. It does not contain the flow condition across the coarse element interface which is often important in guaranteeing the continuity of flux on the coarse-scale. Therefore, we would like to propose a new boundary condition which reflects the heterogeneous structure

across the coarse element boundary.

For a fine-scale element interface ν_a, denote the two adjacent fine-scale elements as e_i and e_j, i.e. $\nu_a = e_i \bigcap e_j$. According to two-point flux approximation finite volume method, if the element interface is in the y-direction, the element interface transmissibility in the x-dimension is defined by [106]:

$$T_{\nu_a} = 2|\nu_a| \left(\frac{\Delta x_i}{K_i} + \frac{\Delta x_j}{K_j} \right)^{-1}, \tag{5.38}$$

where $|\nu_a|$ is the length of the interface, Δx_i denote the length of element e_i in the x-coordinate direction, and K_i is the permeability in element e_i. Similar expression can be defined in the y-dimension. The fine-scale transmissibility of interface ν_a reflects the flow condition across elements. Denote the total applied flux along the coarse element interface Λ due to the value of the i-th coarse-scale basis functions at the k-th quadrature point as

$$Q_{ik} = \int_\Lambda \boldsymbol{\psi}_i^c(\boldsymbol{\xi}_k) \cdot \boldsymbol{n} \, ds = |\Lambda| \boldsymbol{\psi}_i^c(\boldsymbol{\xi}_k) \cdot \boldsymbol{n}_\Lambda. \tag{5.39}$$

Hence, we modify the boundary condition Eq. (5.37) to:

$$\tilde{\boldsymbol{u}}_{ik} \cdot \boldsymbol{n}|_\Lambda = Q_{ik} \cdot \frac{T_{\nu_a}}{\sum_{\nu_b \subset \Lambda} T_{\nu_b} |\nu_b|}, \quad \text{on } \Lambda \subset \partial \mathrm{E}, \tag{5.40}$$

where Q_{ik} is defined in Eq. (5.39) and T_{ν_a} is the fine-scale transmissibility of interface $\nu_a \subset \Lambda$ as defined in Eq. (5.38). When the interface is in the x-direction, we change the definition of T_{ν_a} accordingly. See for example Q_{11} in Fig. 5.3(b). Therefore, the sum of the flux applied on the fine-scale element is equal to the total flux applied on the same coarse element boundary. We just redistribute the total flux on the coarse-scale element boundary according to the ability to transport the flow at the interface of each fine-scale element. This is clearly a better choice for boundary condition since it determines the flow conditions across the inter-block boundaries.

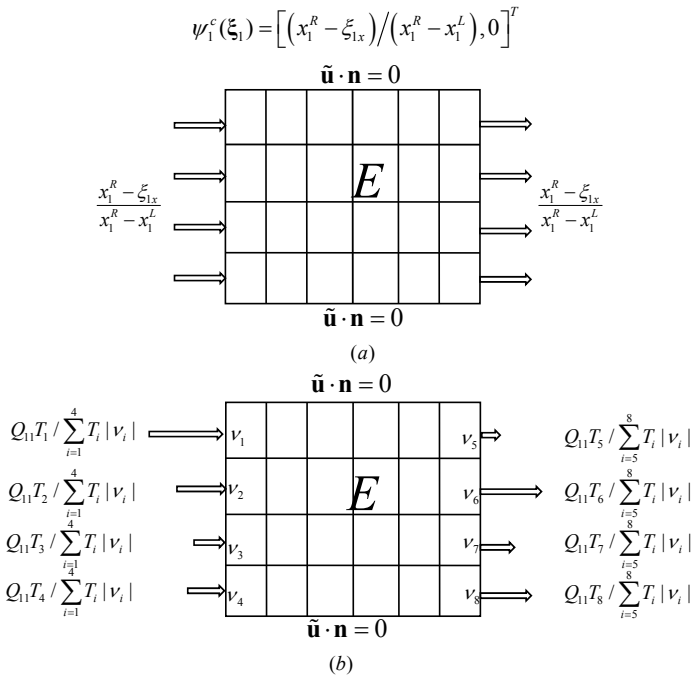


Figure 5.3: Schematic of different boundary conditions. (a) The uniform boundary condition, (b) The modified boundary condition where the flux is scaled according to the fine-scale transmissibilities.

Finally, our subgrid problem in a coarse-element E is defined as follows: For each quadrature point $\boldsymbol{\xi}_k, k = 1, \ldots, 4$, we seek the solution $\tilde{\boldsymbol{u}}_{ik}$ to the following subgrid problem for each coarse-scale RT_0 basis function $\boldsymbol{\psi}_i^c, i = 1, \ldots, 4$:

$$\nabla \tilde{\boldsymbol{u}}_{ik}(\boldsymbol{x}) = 0, \quad \tilde{\boldsymbol{u}}_{ik}(\boldsymbol{x}) = -K \nabla \tilde{p}_{ik}(\boldsymbol{x}), \quad \forall\, \boldsymbol{x} \in E, \tag{5.41}$$

with the Neumman boundary condition defined in Eq. (5.40).

For convenience, we will define the corresponding modified bilinear form as: for any $\boldsymbol{u}_c, \boldsymbol{v}_c \in V_c$

$$A_h(K^{-1}\boldsymbol{u}_c, \boldsymbol{v}_c) := \sum_{E \in \mathcal{T}_c} \sum_{k=1}^{4} \frac{\tau_k}{|E|} \int_E \boldsymbol{U}_k(\boldsymbol{x}) \cdot K^{-1} \boldsymbol{V}_k(\boldsymbol{x})\, d\boldsymbol{x}, \tag{5.42}$$

where $\boldsymbol{U}(\boldsymbol{V})$ is defined through the subgrid problems. The assembling of this bilinear form will be detailed in Section 5.3.2. Therefore, the MxHMM version of

Eqs. (5.25)-(5.26) on the coarse-scale reads: Find the coarse-scale $(\boldsymbol{u}_c, p_c) \in V_c \times W_c$ such that

$$A_h(K^{-1}\boldsymbol{u}_c, \boldsymbol{v}_c) - (p_c, \nabla \cdot \boldsymbol{v}_c) = -\langle p_0, \boldsymbol{v}_c \cdot \boldsymbol{n} \rangle, \quad \forall\, \boldsymbol{v}_c \in V_c, \tag{5.43}$$

$$(l_c, \nabla \cdot \boldsymbol{u}_c) = 0, \quad \forall\, l_c \in W_c, \tag{5.44}$$

with the boundary condition

$$p_c = \bar{p} \ \text{ on } \ \partial \mathrm{D_p}, \quad \boldsymbol{u}_\mathrm{c} \cdot \boldsymbol{n} = 0 \ \text{ on } \ \partial \mathrm{D_u}. \tag{5.45}$$

The major difference between Eqs. (5.25)-(5.26) and Eqs. (5.43)-(5.44) lies in the bilinear form $A_h(\cdot, \cdot)$, which needs solution of the local subgrid problem Eq. (5.41). It is through these subgrid problems and the mixed formulation that the effect of the heterogeneity on coarse-scale solutions can be correctly captured. Unfortunately, it is not trivial to analyze this multiscale method in a general case, but convergence results have been obtained using the homogenization theory in the case of periodic coefficients [96].

Solution of the subgrid problems and assembling the bilinear form

In general, the subgrid problem Eq. (5.41) can be solved through standard or mixed finite element method. In the present setting, since we are only interested in the velocity, the mixed finite element method is preferred. Let $E_h = \mathcal{T}_h(E)$ denote the fine grid defined over one coarse element E. As mentioned before, it coincides with the fine-scale grid $\mathcal{T}_h$. The subgrid-scale velocity functional spaces will be defined on the fine grid E_h of each coarse element:

$$V_E = \left\{ \tilde{\boldsymbol{u}} : \tilde{\boldsymbol{u}} = \sum_{a=1}^{M_E} \boldsymbol{\psi}_a^h \tilde{u}_a^h, \ \ \boldsymbol{\psi}_a^h \in RT_0(E_h) \right\}, \tag{5.46}$$

where M_E is the number of edges in E, and the pressure space is defined similarly:

$$W_E = \left\{ \tilde{p} : \tilde{p} = \sum_{a=1}^{N_E} \phi_i^h \tilde{p}_i^h, \;\; \phi_i^h \in P_0(E_h) \right\}, \tag{5.47}$$

where N_E is the number of elements in E. It is noted that, as the Neumann boundary conditions in Eq. (5.40) are imposed on all boundaries of the coarse element E, an extra constraint must be added to make the subgrid problem well posed. In our implementation, the pressure is prescribed to 0 at one of the elements in E_h.

The mixed finite element method approximation of Eq. (5.41) in coarse element E_i on the subgrid-scale grid reads: Find the subgrid-scale $(\tilde{\boldsymbol{u}}, \tilde{p}) \in V_{E_i} \times W_{E_i}$ such that

$$(K^{-1}\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}) - (\tilde{p}, \nabla \cdot \tilde{\boldsymbol{v}}) = 0, \;\; \forall\, \tilde{\boldsymbol{v}} \in V_{E_i}, \tag{5.48}$$

$$(\tilde{l}, \nabla \cdot \tilde{\boldsymbol{v}}) = 0, \;\; \forall\, \tilde{l} \in W_{E_i}, \tag{5.49}$$

with the boundary condition Eq. (5.40). It is noted that for each coarse element, we need to solve 4 (number of quadrature points) $\times$ 4 (number of basis functions) = 16 subgrid problems. However, the only difference between them is the boundary condition. Therefore, we only need to assemble the stiffness matrix once and solve the problem with different right hand vectors.

Following a standard assembly process for the global matrix of the coarse-scale bilinear form Eq. (5.42), we compute the contribution A_E to the global matrix associated with the coarse element E, where A_E is a 4×4 matrix. Assume the solution of the subgrid problem at the k-th Gaussian point can be written as $\tilde{\boldsymbol{u}}_{ik} = \sum_{j=1}^{M_E} c_{ij}^k \boldsymbol{\psi}_j^h, i = 1, \ldots, 4$. We can write all the solutions as a $4 \times N_E$ matrix, $C_k = (c_{ij}^k)$ where the i-th row contains the subgrid solution corresponding to the

i-th coarse-scale basis function $\boldsymbol{\psi}_i^c$. Therefore, the value of A_E from the k-th Gauss point can be denoted as $A_E^k = (a_E^k)_{ij}$, where

$$\left(a_E^k\right)_{ij} = \frac{\tau_k}{|E|} c_{il} \int_E K^{-1} \boldsymbol{\psi}_l^h \cdot \boldsymbol{\psi}_m^h d\boldsymbol{x}\ c_{jm}. \tag{5.50}$$

Denoting the bilinear form matrix from the subgrid-scale problem as $B^k = (b_{lm}^k)$, $b_{lm}^k = \int_E K^{-1} \boldsymbol{\psi}_l^h \cdot \boldsymbol{\psi}_m^h d\boldsymbol{x}$, we can write:

$$A_E = \sum_{k=1}^{4} \frac{\tau_k}{|E|} C_k B^k (C_k)^T. \tag{5.51}$$

Finally, we would like to comment on the solution of the linear systems resulting from the mixed finite element discretization. The linear system is indefinite, and it is difficult to solve using the common iterative method. In our implementation, we use the Schur complement matrix to solve the pressure first and then solve the velocity [80]. The linear system is solved using preconditioned conjugate gradient method. All the implementations are based on the data structure of the numerical library PETSc [107].

5.3.2 Reconstruction of the fine-scale velocity and solution of transport equation

So far we have described the development of the mixed finite element heterogeneous multiscale method for the solution of the coarse-scale velocity. However, in order to simulate the transport equation accurately, we need to reconstruct the fine-scale velocity using the coarse-scale velocity and the subgrid permeability. It is noted that the coarse-scale velocity is not conservative at the fine-scale. In order to obtain a mass-conservative fine-scale velocity, we solve

Darcy's equation within each coarse element E using Neumann boundary condition given by the coarse-scale flux along the coarse-element boundary. The coarse-scale flux, denoted by Q^c is directly given as the solution of the system of linear equations from the coarse-scale discretization. That is, for each $E \in \mathcal{T}_c$, one solves the fine-scale velocity $\boldsymbol{u}_h$ inside E by [87, 106]

$$\nabla \cdot \boldsymbol{u}_h = 0, \quad \boldsymbol{u}_h = -K\nabla p_h, \ \ \forall \boldsymbol{x} \in E, \tag{5.52}$$

with the boundary condition similar to the one used in Eq. (5.40):

$$\boldsymbol{u}_h \cdot \boldsymbol{n}|_\Lambda = Q^c \cdot \frac{T_{\nu_a}}{\sum_{\nu_b \subset \Lambda} T_{\nu_b} |\nu_b|}, \ \ \text{on } \Lambda \subset \partial \mathrm{E}, \tag{5.53}$$

where Q^c is the coarse-scale flux across the coarse element interface Λ, and T_{ν_a} is the fine-scale transmissibility of interface $\nu_a \subset \Lambda$. Since the mixed finite element method is used to solve the coarse-scale equations, the coarse-scale flux Q^c is obtained directly. Similar to the subgrid problem, Neumann boundary condition is applied on all the boundaries of the coarse element. To obtain a unique solution of the above problem, the pressure is fixed to the coarse-scale pressure p_c in the center element of the mesh E_h. As indicated in [87, 106, 108], this reconstruction step guarantees the continuity of the flux across the fine-scale elements between two coarse blocks and accounts for subgrid heterogeneity. It also forces the sum of the fine grid fluxes to be equal to the corresponding coarse-scale flux. In this way, the resulting fine-scale velocity is conservative on fine-scale grid as well as the coarse-scale grid.

For the solution of the saturation equation, we use the upwinding finite element method [77, 102], which is equivalent to the standard upstream weighted finite volume method. We also approximate the saturation as a piecewise constant in each fine-scale element e, $P_0(\mathcal{T}_h)$, the same as the pressure space. Given the discrete reconstructed fine-scale velocity field $\boldsymbol{u}_h$, for a fine-scale element

$e \in \mathcal{T}_h$. We define the inflow boundary of the element as ∂e_-, if $\boldsymbol{u}_h \cdot \boldsymbol{n} < 0$ on ∂e_- and similarly the outflow boundary as ∂e_+, if $\boldsymbol{u}_h \cdot \boldsymbol{n} \geqslant 0$ on ∂e_+. For any piecewise constant function S_h over the mesh $\mathcal{T}_h$, the upwinding value on ∂e is defined as $\tilde{S}_h$ and is equal to the interior trace of S_h if on ∂e_+ and equal to the exterior trace of S_h if on ∂e_-. In addition, we also assume $\tilde{S}_h = 0$ on $\partial e_- \bigcap \partial D$.

Therefore, the weak formulation of the upwinding scheme is to find $S_h \in W_h$ such that

$$\int_D \frac{\partial S_h}{\partial t} q_h \, d\boldsymbol{x} + \sum_{e \in \mathcal{T}_h} \int_{\partial e} (\boldsymbol{u}_h \cdot \boldsymbol{n}) \tilde{S}_h q_h \, ds = 0, \quad \forall q_h \in W_h. \tag{5.54}$$

Let Δt be the time step and denote by S_i^k the approximation of the water saturation in fine-scale element e_i at time t^k. Then the discrete form of the saturation Eq. (5.54) is:

$$S_i^{k+1} + \frac{\Delta t}{|e_i|} \sum_{j \neq i} f_{ij}(S^{k+1}) q_{ij} = S_i^k. \tag{5.55}$$

Here $|e_i|$ is the area of the element e_i. $f_{ij}(S) = \max\{\text{sign}(q_{ij}) S_i, -\text{sign}(q_{ij}) S_j\}$ is the upwinding water saturation for the interface $\nu_{ij} = \partial e_i \bigcap \partial e_j$. Finally, the flux across the boundary is $q_{ij} = \int_{\nu_{ij}} \boldsymbol{u}_h \cdot \boldsymbol{n}_{ij} \, ds$ where $\boldsymbol{n}_{ij}$ is the unit normal to ν_{ij} pointing from e_i to e_j. It is noted that in Eq. (5.55), only the flux q_{ij} on the each interface is required. This value is directly computed as the solution from our multiscale approach. That's why the method discussed here is better than the stabilized conforming finite element method [109].

It is emphasized again that we consider the transport problem with unit mobility ratio, so the saturation changes will not affect the pressure or velocity. Therefore, we can first compute the fine-scale velocity with our multiscale approach and then solve the transport equation. The flow rate of produced oil at the outlet boundary is denotes as q_o and the flow rate of produced water q_w.

To assess the quality of our multiscale approach, we will use the so called water cut curve F, which defines the fraction of water in the produced fluid, i.e., $F = q_w/(q_w + q_o)$ as a function of time measured in pore volume injected (PVI). The water-cut is defined as

$$F(t) = \frac{\int_{\partial D^{\text{out}}} (\boldsymbol{u}_h \cdot \boldsymbol{n}) S \; ds}{\int_{\partial D^{\text{out}}} (\boldsymbol{u}_h \cdot \boldsymbol{n}) \; ds}, \tag{5.56}$$

where ∂D^{out} refers to the part of the boundary with outer flow, i.e. $\boldsymbol{u}_h \cdot \boldsymbol{n} > 0$. PVI represents dimensionless time and is computed as

$$\text{PVI} = \int Q \; dt/V_p, \tag{5.57}$$

where V_p is the total pore volume of the system, which is equal to the area of the domain D here and $Q = \int_{\partial D^{\text{out}}} (\boldsymbol{u}_h \cdot \boldsymbol{n}) \; ds$ is the total flow rate.

The complete schematic of the stochastic multiscale method for porous media flow is illustrated in Fig. 5.4.

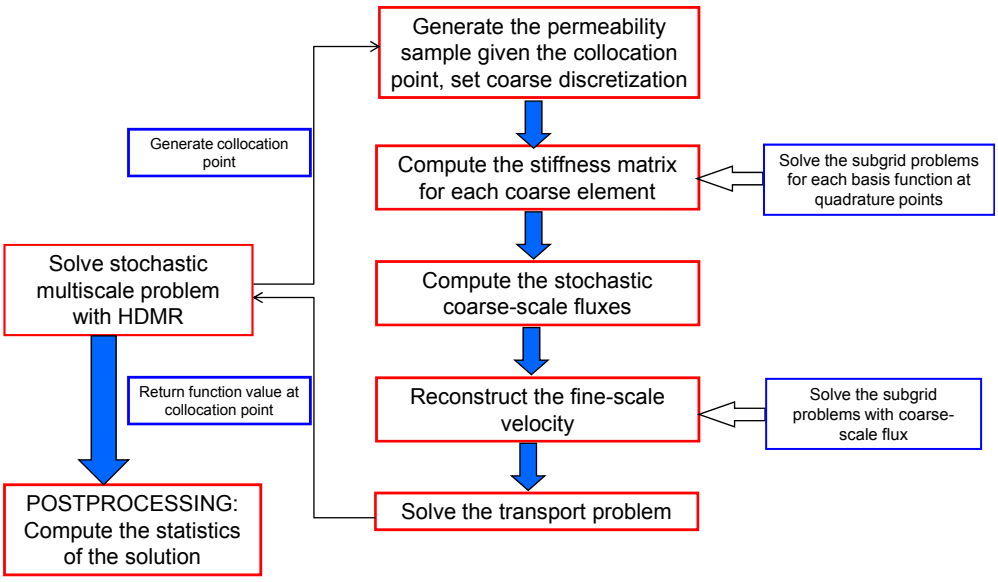


Figure 5.4: Schematic of the developed stochastic multsicale method for porous media flow.

5.4 Numerical Examples

In the first two examples, we solve the problem with deterministic permeability in order to validate the newly developed multiscale method. In the third example, the complete stochastic problem with a known covariance function is addressed.

5.4.1 Simulation in realistic two-dimensional reservoirs

This test case is a two-dimensional problem with a highly heterogeneous permeability. The permeability field shown in Fig. 5.5 is taken from the top layer of the 10-th SPE comparative solution project [110]. The fine grid on which the permeability is defined consists of 60×220 gridblocks. It has Dirichlet boundary conditions $\bar{p} = 100$ on $\{x_2 = 0\}$, $\bar{p} = 0$ on $\{x_2 = 220\}$ and Neumann boundary conditions $\boldsymbol{u} \cdot \boldsymbol{n} = 0$ on both $\{x_1 = 0\}, \{x_1 = 60\}$. We also impose zero initial condition for saturation $S(x, 0) = 0$ and boundary condition $S(x, t) = 1$ on $\{x_2 = 0\}$.

The reference solution is computed on the fine-scale grid using single-scale mixed finite element method directly, as shown in Fig. 5.6(a) and Fig. 5.7(a). We also show the solutions obtained with the MxHMM method on various coarse grids in Figs. 5.6 and 5.7. It is seen that the flow focuses along the region with higher permeability while bypassing the low-permeability areas. At the same time, the velocity field displays significant small-scale structure corresponding to the spatial permeability variations. The multiscale solution successfully captures all the main characters of the fine-scale results and compare very well with the fine-scale solution, with the two results being quite difficult to distinguish

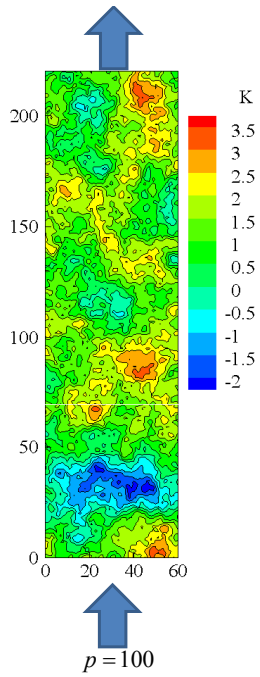


Figure 5.5: Logarithm of the permeability field from the the top layer of the 10-th SPE model, which is defined on 60×220 fine grid.

visually. As a direct measure of the error in the computed velocity field, we consider the L^2–norm: $\|\boldsymbol{u}\|_2 = (\int_D \boldsymbol{u} \cdot \boldsymbol{u}\, d\boldsymbol{x})^{1/2}$, where the corresponding relative error is given as $\delta(\boldsymbol{u}) = \|\boldsymbol{u}_{\text{ref}} - \boldsymbol{u}_{\text{ms}}\|/\|\boldsymbol{u}_{\text{ref}}\|$. The result is given in Table 5.1. In general, the error is larger with coarser grid which is possibly due to some large local error in the high permeability region where the velocity changes quickly.

However, for reservoir simulation the most crucial factor is the transport properties of a velocity field. That is, a large local error in the velocity field may not be crucial as long as the overall transport properties are correct. Therefore, we give the contour plots of the saturation at time 0.4 PVI for various coarse grids in Fig. 5.8. The four multiscale results compare nearly the same as the reference solution. To assess the accuracy of the transport properties, we measure the relative difference in the saturation profile at a given time:

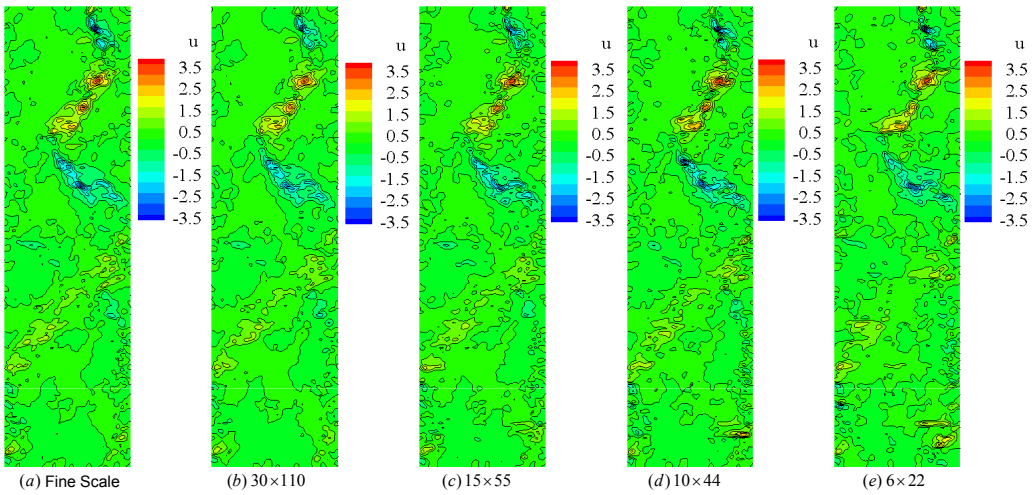


Figure 5.6: Contour plots of the x-velocity component for various meshes: (a) 60×220 fine-scale grid, (b) 30×110 coarse grid, (c) 15×55 coarse grid, (d) 10×44 coarse grid, (e) 6×22 coarse grid.

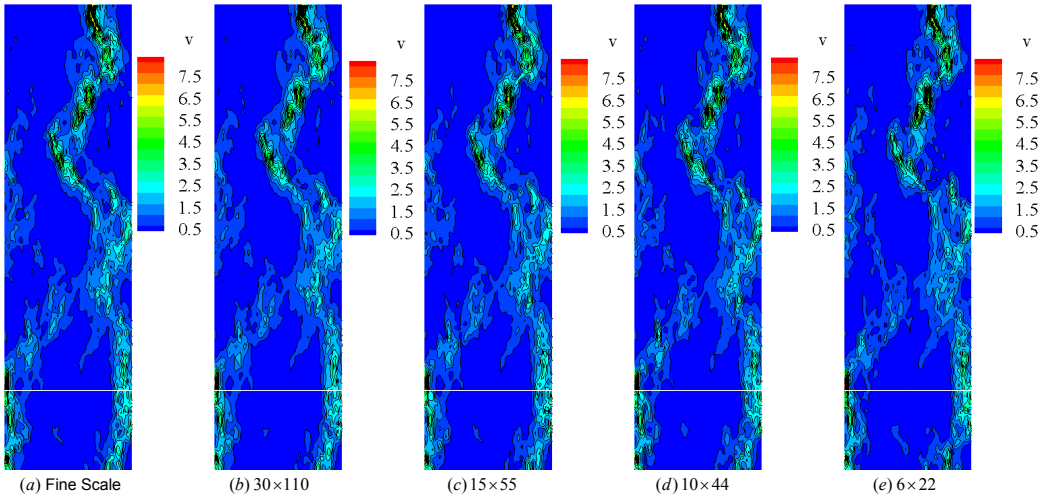


Figure 5.7: Contour plots of the y-velocity component for various meshes: (a) 60×220 fine-scale grid, (b) 30×110 coarse grid, (c) 15×55 coarse grid, (d) 10×44 coarse grid, (e) 6×22 coarse grid.

$\delta(S) = (\int_D |S_{\text{ref}} - S_{\text{ms}}|^2 \, d\boldsymbol{x})^{1/2} / (\int_D |S_{\text{ref}}|^2 \, d\boldsymbol{x})^{1/2}$. The result is given in Table 5.1. It is seen that although the corresponding velocity error is larger for the same coarse grid, the saturation error is significantly smaller.

Finally, we consider the water cut, which is shown in Fig. 5.9. Once again,

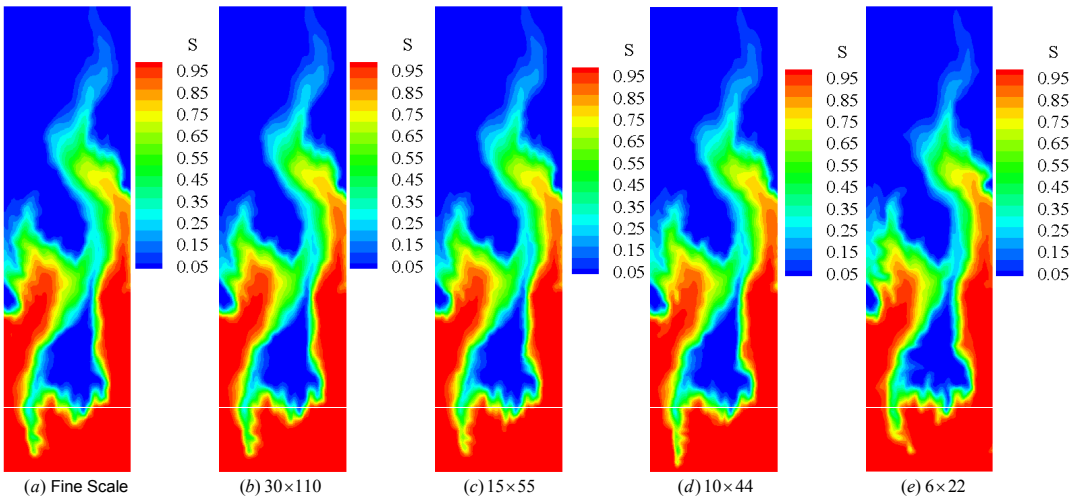


Figure 5.8: Contour plots of Saturation at 0.4 PVI: (a) 60×220 fine-scale grid, (b) 30×110 coarse grid, (c) 15×55 coarse grid, (d) 10×44 coarse grid, (e) 6×22 coarse grid.

the results compare well with the reference solution. Here, we measure the maximum error as $\delta(F) = \max_{t \geqslant 0} |F_{\text{ref}}(t) - F_{\text{ms}}(t)|$. The result is shown in Table 5.1, where the error is quite small. Note that this is a quite strict measure, since the water cut curves tend to be steep right after breakthrough, and thus a small deviation in breakthrough time may give a large value in the error measure.

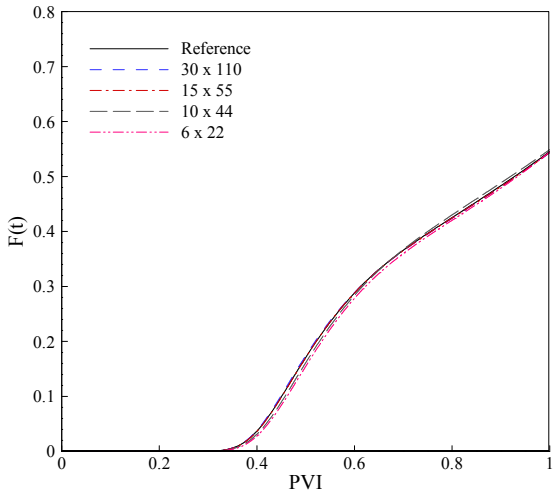


Figure 5.9: Water cut curves for various coarse grids.

Table 5.1: Relative errors for various coarse grids in Example 1.

Errors	30×110	15×55	10×44	6×22
$\delta(\boldsymbol{u})$	0.112	0.159	0.170	0.234
$\delta(S)$	0.025	0.049	0.067	0.124
$\delta(F)$	0.0033	0.0019	0.0101	0.0165

Overall, through this example, it is shown that the introduced multiscale method is quite robust and accurate for different mesh discretizations.

5.4.2 Simulation in a realization sample from a random permeability filed

In this section, we consider only a sample realization from a random permeability field, which can be considered as a deterministic run at a collocation point in a stochastic simulation. The permeability is defined on a 100×100 fine-scale grid, which is shown in Fig. 5.10. Flow is induced from left-to-right with Dirichlet boundary conditions $\bar{p} = 100$ on $\{x_1 = 0\}$, $\bar{p} = 0$ on $\{x_1 = 100\}$ and no-flow homogeneous Neumann boundary conditions on the other two edges. We also impose zero initial condition for saturation $S(x, 0) = 0$ and boundary condition $S(x, t) = 1$ on the inflow boundary $\{x_1 = 0\}$. The reference solution is again taken from the single-scale mixed finite element on the fine-scale grid directly. All the errors are defined the same as before.

In Figs. 5.11 and 5.12, we show the velocity contour plots of the reference

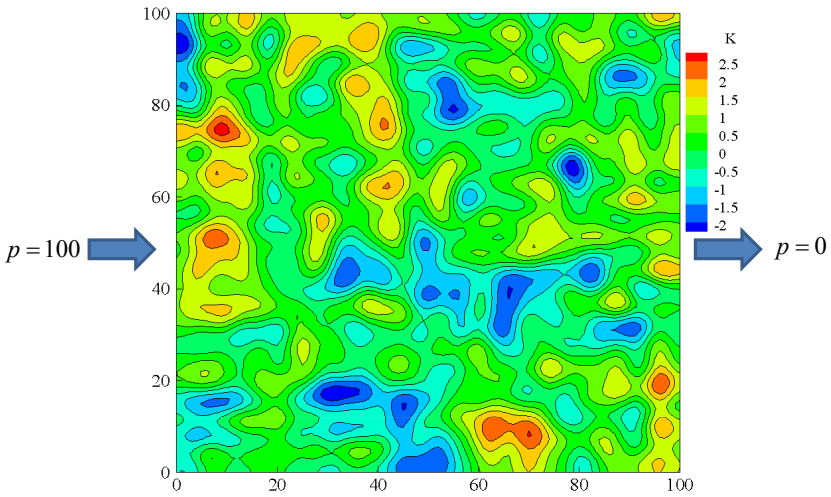
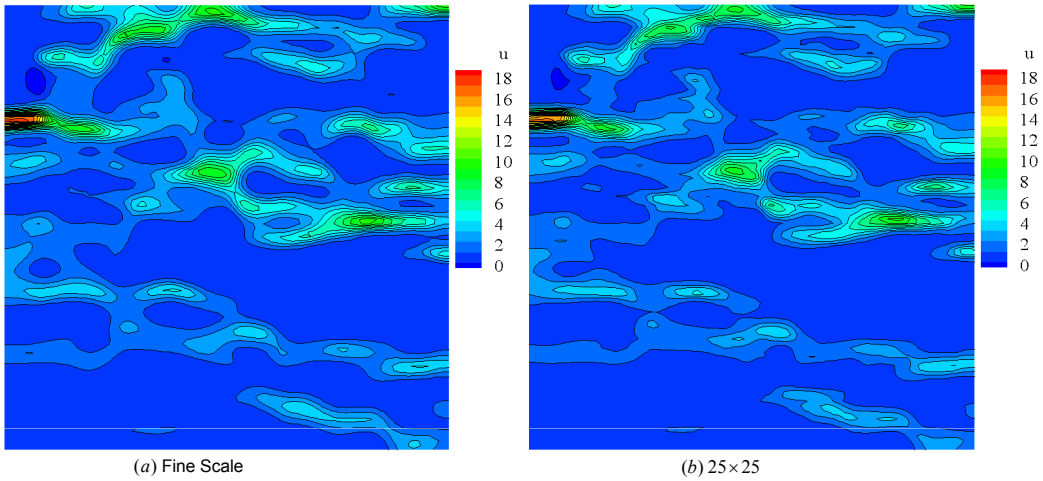


Figure 5.10: Logarithm of the permeability field from one sample of a log-normal permeability filed defined on 100×100 fine-scale grid.

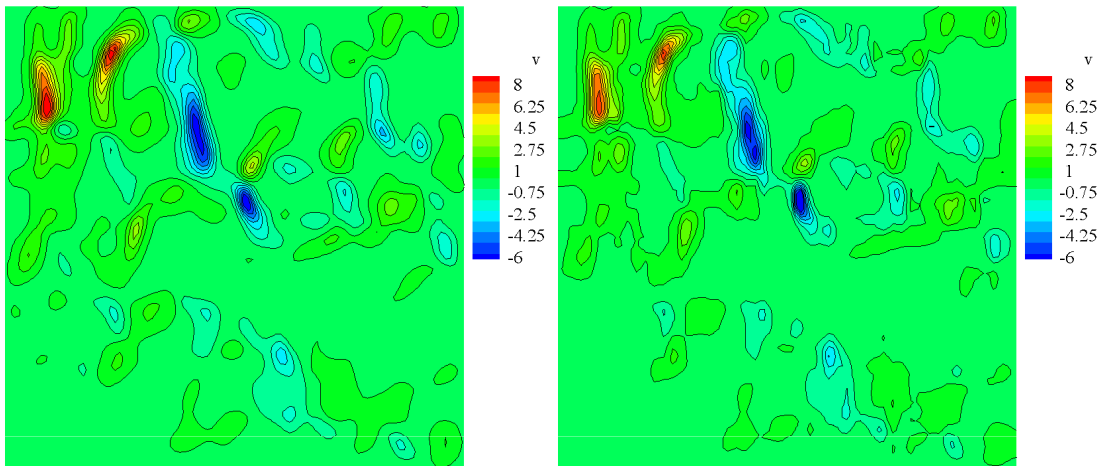
solution and the multiscale solution on a 25×25 coarse grid. The flow tries to go through the high permeable regions and bypass the low permeable regions, which is clearly reflected in the saturation plot at time 0.4 PVI as shown in Fig. 5.13. All the three figures compare well with the reference solutions. The relative errors are shown in Table 5.2. We note the relatively small saturation errors compared with the large velocity errors, which again confirms that the large local velocity errors may not reflect the overall accuracy of the saturation results as long as the multiscale method captures the major feature of the underlying permeability field.

Water cut curves are shown in Fig. 5.14 and the maximum error is given in Table 5.2. All the water cut curves are visually nearly the same. The two deterministic numerical examples successfully validate the introduced multiscale model. Since the stochastic multiscale framework only requires repeated solution of the deterministic problems at different collocation points, it is expected to also have accurate statistics of the solution in the stochastic simulation as



(*a*) Fine Scale (*b*) 25×25

Figure 5.11: Contour plots of the x-velocity component for (a) 100×100 fine-scale grid, (b) 25×25 coarse grid.



(*a*) Fine Scale (*b*) 25×25

Figure 5.12: Contour plots of the y-velocity component for (a) 100×100 fine-scale grid, (b) 25×25 coarse grid.

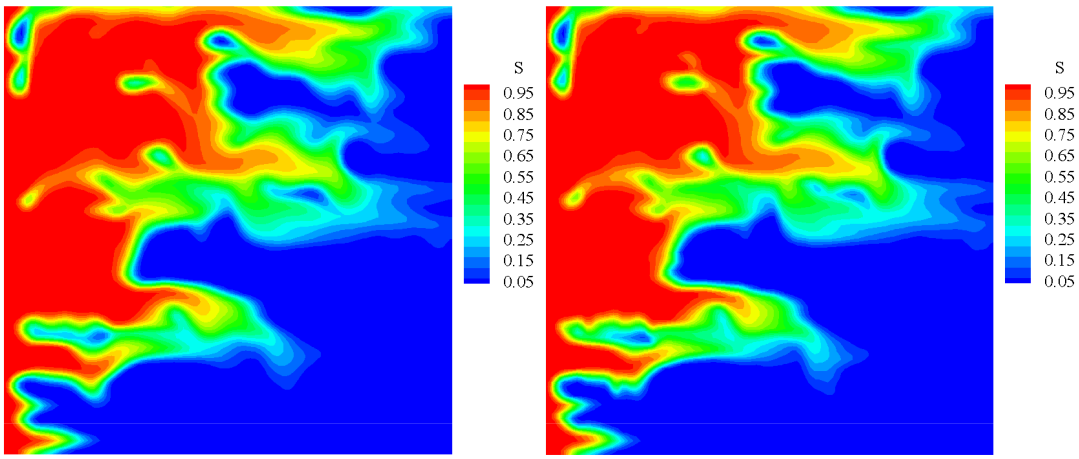


Figure 5.13: Contour plots of Saturation at 0.4 PVI: for (a) 100×100 fine-scale grid, (b) 25×25 coarse grid.

shown in the next example.

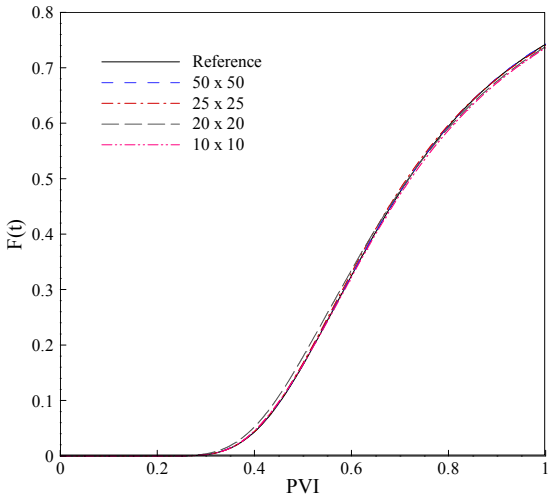


Figure 5.14: Water cut curves for various coarse grids.

5.4.3 Simulation in random permeability field

In the last two examples, we have successfully verified the accuracy of our newly developed multiscale solver. In this example, we are going to investi-

Table 5.2: Relative errors for various coarse grids in Example 2.

Errors	50×50	25×25	20×20	10×10
$\delta(\boldsymbol{u})$	0.060	0.156	0.183	0.324
$\delta(S)$	0.019	0.065	0.089	0.182
$\delta(F)$	0.0017	0.0059	0.0149	0.0079

gate the statistical properties of the transport phenomenons in random heterogeneous porous media. The domain of interest is the unit square $[0,1]^2$. Flow is still induced from left-to-right with Dirichlet boundary conditions $\bar{p} = 1$ on $\{x_1 = 0\}$, $\bar{p} = 0$ on $\{x_1 = 1\}$ and no-flow homogeneous Neumann boundary conditions on the other two edges. We also impose zero initial condition for saturation $S(x,0) = 0$ and boundary condition $S(x,t) = 1$ on the inflow boundary $\{x_1 = 0\}$.

The log-permeability is taken as zero mean random field with a separable exponential covariance function

$$\mathrm{Cov}(\boldsymbol{x},\boldsymbol{y}) = \sigma^2 \exp\left(-\frac{|x_1 - y_1|}{L_1} - \frac{|x_2 - y_2|}{L_2}\right), \tag{5.58}$$

where L_1 and L_2 are the correlation lengths in x and y direction, respectively. σ is the standard deviation of the random field. The K-L expansion is used to parameterize the field as

$$\mathbf{Y}(\omega) = \log\left(K(\omega)\right) = \sum_{i=1}^{N} \sqrt{\lambda_i}\phi_i(\boldsymbol{x})Y_i, \tag{5.59}$$

where the eigenvalues $\lambda_i, i = 1,2,\ldots,$ and their corresponding eigenfunctions $\phi_i, i = 1,2,\ldots,$ can be determined analytically as discussed in [111]. Different probability distributions can be chosen for Y_i. The effects of log permeability

with uniform, beta and Gaussian distributions on the mean and standard deviation of the output were investigated in [112], where the results showed that the three distributions had close peak values of standard deviation. Therefore, without losing the main feature of the output uncertainty, here Y_i are assumed as i.i.d. uniform random variables on $[-1, 1]$.

In this problem, the fine-scale permeability is defined on 64×64 grid and the coarse grid is taken as 8×8. For comparison, the reference solution is taken from 10^6 MC samples, where each direct problem is solved using the fine-scale solver. The stochastic problem is solved using HDMR, where the solution of each deterministic problem at the collocation points is from the multiscale solver. In this way, the accuracy of both multiscale solver and HDMR can be verified. In Chapter 4, the effects of the correlation length and standard deviation have been studied thoroughly. Thus, here we will fix the standard deviation to $\sigma^2 = 1.0$ and investigate the effect of the anisotropy of the random field.

Isotropic random field

In this problem, we take $L_1 = L_2 = 0.1$. Due to the slow decay of the eigenvalue, the Eq. (5.59) is truncated after 100 terms. Therefore, the stochastic dimension is 100. The problem is solved with HDMR where each sub-problem is solved through ASGC. We take $\varepsilon = 10^{-6}, \theta_1 = 5 \times 10^{-5}$ and $\theta_2 = 10^{-4}$.

In Fig. 5.15 we compare the mean and standard deviation at 0.2 PVI. It is interesting to note that although the permeability field shows heterogeneity for different realization, the mean saturation is the same as the solution with homogeneous mean permeability filed. This behavior is called "heterogeneity-

induced dispersion" where the heterogeneity smoothes the water saturation profile in the ensemble sense. Our results again confirms this phenomenon, which was first investigated in [113] through method of moment equations. The figure also indicates that higher water saturation variations are concentrated near displacement fronts, which are areas of steep saturation gradients. Therefore, the comparisons between the MC and HDMR results are only shown around the displacement fronts on the bottom two plots in Fig. 5.15. It is seen that the solutions from HDMR compare quite well with that of MC results. The convergence of HDMR is shown in Table. 5.3, where the normalized error is defined the same as before with MC results as the reference solution. N_i denotes the number of important dimensions and N_c denotes the total number of component functions. The expansion order of HDMR for all three cases is 2. For conventional HDMR, the total number of component functions is 5051. However, by using adaptivity, N_c is reduced to 1047 which clearly demonstrates the advantage of our methods. From the table, it is seen that the results are indeed quit accurate despite the fact that 64-fold upscaling is used to solve the deterministic problem and adaptive methods are used to solve the stochastic problem.

Table 5.3: Convergence of HDMR with different θ_1 at 0.2 PVI for isotropic random field.

θ_1	N_i	N_c	# Points	Error mean	Error std
1×10^{-3}	2	102	1694	7.47×10^{-4}	4.38×10^{-2}
1×10^{-4}	27	452	34379	5.69×10^{-4}	2.06×10^{-2}
5×10^{-5}	44	1047	77988	5.10×10^{-4}	6.66×10^{-3}

Next, we demonstrate the interpolatory properties of our HDMR method.

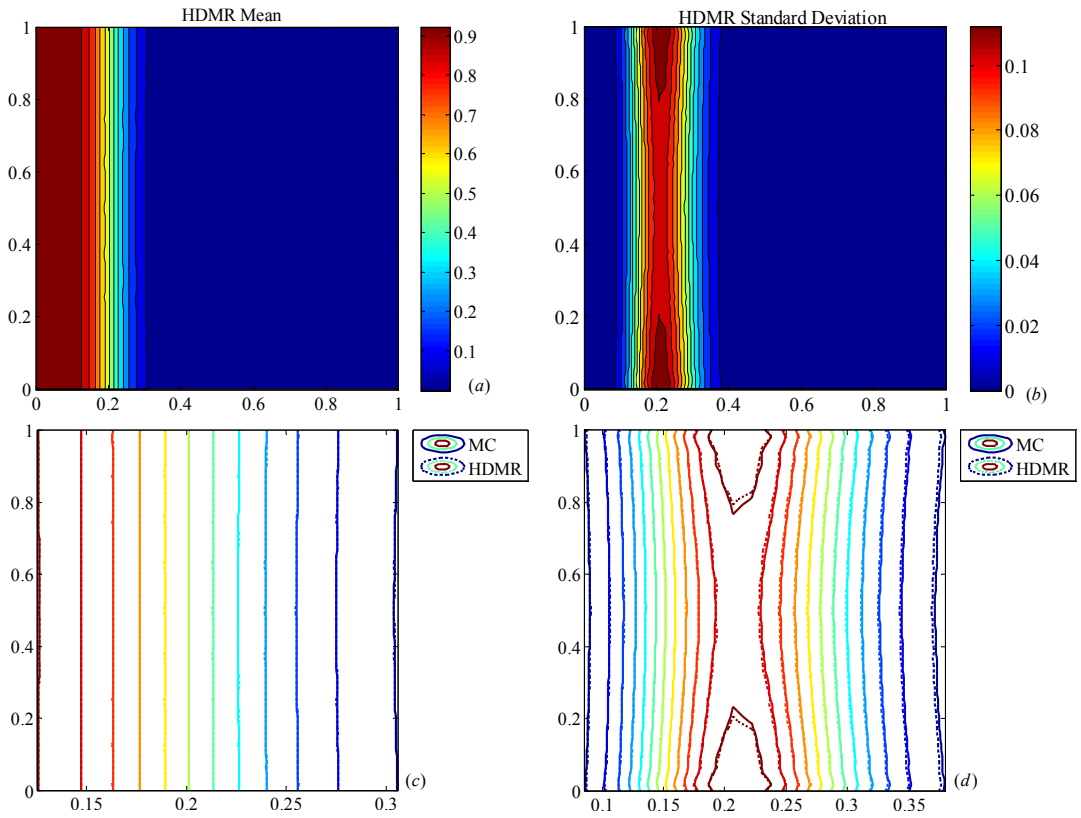


Figure 5.15: Mean and standard deviation of saturation at 0.2 PVI for isotropic random field. Top: Mean (a) and standard deviation (b) form HDMR. Bottom: Comparison of mean (c) and standard deviation (d) between MC and HDMR near the saturation front.

As mentioned before, one of the advantages of HDMR is that it can be served as a surrogate model for the original problem. Realization of the saturation for arbitrary random input can be obtained through HDMR. To verify this property, we randomly generate one input vector and reconstruct the result from HDMR. At the same time, we run a deterministic problem with the fine-scale model and the same realization of the random input vector. The comparison of these results are shown in Fig. 5.16. In addition, in Fig. 5.17, we also plot the probability density function (PDF) and cumulative distribution function (CDF) at point $(0.2, 0)$ where it has the highest standard deviation as indicated from

Fig. 5.15(b). These results indicate that the corresponding HDMR approximations are indeed very accurate. Therefore, we can obtain any statistics from this stochastic reduced-order model, which is an advantage of the current method over the MC method.

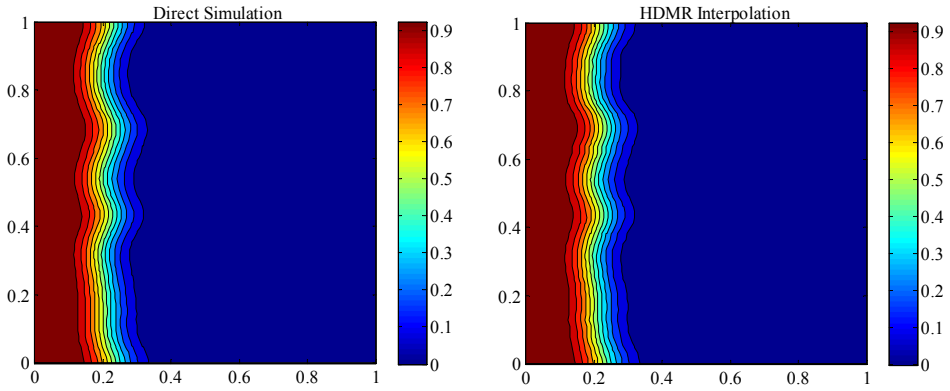


Figure 5.16: Prediction of the saturation profile using HDMR and the solution of the deterministic fine-scale problem with the same input for isotropic random field. Left: Saturation at 0.2 PVI from direct simulation , Right: Saturation at 0.2 PVI reconstructed from HDMR.

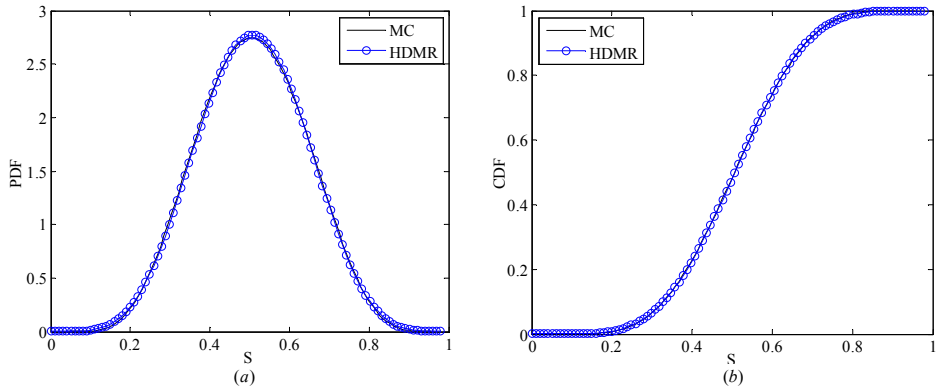


Figure 5.17: Isotropic random field: (a) PDF of the saturation at point (0.2,0) and 0.2 PVI, (b) CDF of the saturation at point (0.2,0) and 0.2 PVI.

Similar results at 0.4 PVI are also given in Figs. 5.18, 5.19 and 5.20 respectively. It is noted that the standard deviation of the saturation becomes larger at later time as is seen from the wider strip of the non-zero regions in the contour

maps at 0.4 PVI in Fig. 5.18. With the increasing of standard deviation, more collocation points are needed to capture the overall uncertainty. Indeed, there are 1229 component functions and 104662 collocation points in this case. From Fig. 5.19, it is seen that the saturation front exhibits a much more significant variation due to the larger standard deviation. Similarly, in Fig. 5.20, we plot the PDF and CDF at point $(0.4, 0)$ where the highest standard deviation happens. It is noted that the spread of the PDF at 0.4 PVI is wider than that of 0.2 PVI which again indicates the larger variation of the saturation at this time step. Thus, it is more difficult to predict the uncertainty with the simulation time increases.

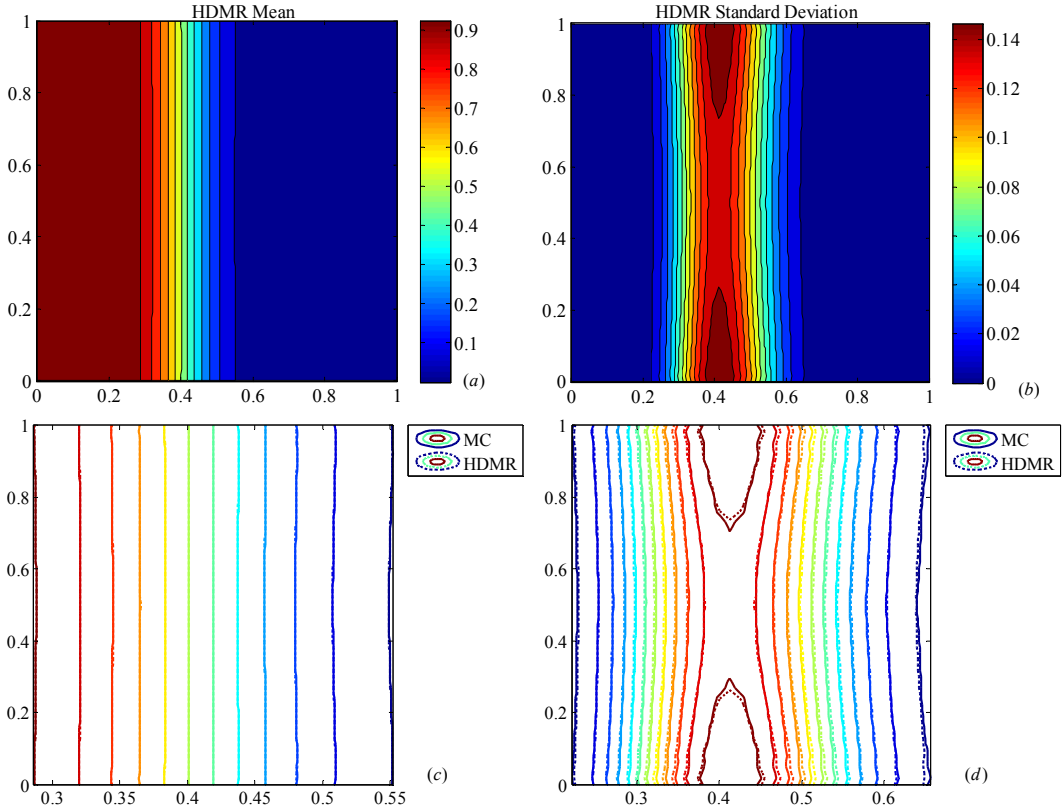


Figure 5.18: Mean and standard deviation of saturation at 0.4 PVI for isotropic random field. Top: Mean (a) and standard deviation (b) form HDMR. Bottom: Comparison of mean (c) and standard deviation (d) between MC and HDMR near the saturation front.

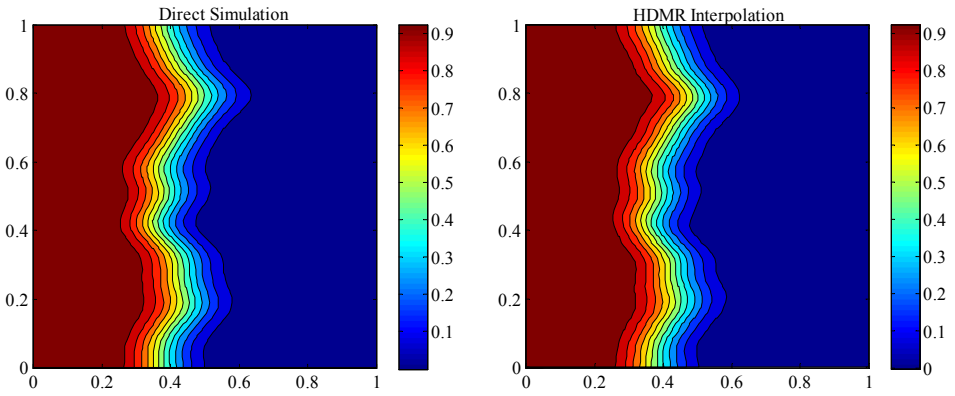


Figure 5.19: Prediction of the saturation profile using HDMR and the solution of the deterministic fine-scale problem with the same input for isotropic random field. Left: Saturation at 0.4 PVI from direct simulation , Right: Saturation at 0.4 PVI reconstructed from HDMR.

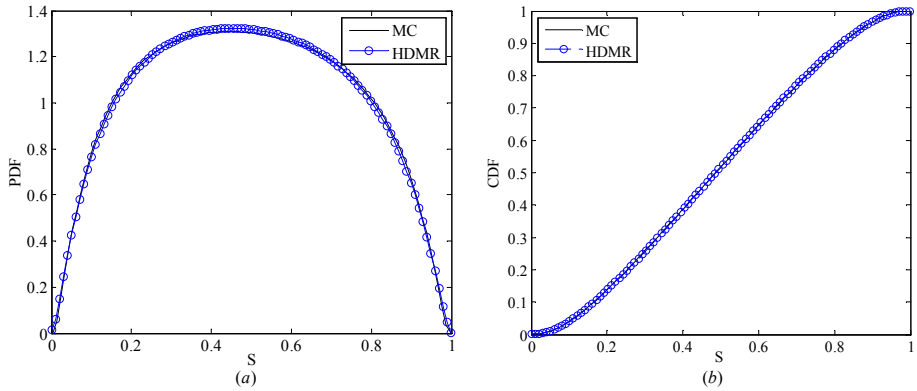


Figure 5.20: Isotropic random field:(a) PDF of the saturation at point (0.4,0) and 0.4 PVI, (b) CDF of the saturation at point (0.4,0) and 0.4 PVI.

Anisotropic random field

In this problem, we take $L_1 = 0.25, L_2 = 0.1$. Due to the increasing of correlation length in x direction, the Eq. (5.59) is truncated after 50 terms. Therefore, the stochastic dimension is 50.

We first solve this problem at time 0.2 PVI using HDMR with ASGC. We

take $\varepsilon = 10^{-6}, \theta_1 = 5 \times 10^{-5}$ and $\theta_2 = 10^{-4}$. The results are shown in Fig. 5.21. It is interesting to note that the shape of contour is nearly the same as that of isotropic random field. Only the values of standard deviation are different. The introduction of anisotropy has the effect of increasing the output uncertainty. The convergence of HDMR shown in Table. 5.4. Again, the HDMR results compare very well with the reference solution. According to our previous numerical results in [66], larger uncertainty requires more expansion terms. Indeed, more expansion terms and collocation points are needed compared with that of the isotropic case. In addition, the highest HDMR expansion order is 3. There are 3 third-order component functions, which indicating the existence of higher-order cooperative effects among the inputs. The reconstruction of the saturation profile is shown in Fig. 5.22. The PDF and CDF at point (0.2,0) are shown in Fig. 5.23.

Table 5.4: Convergence of HDMR with different θ_1 at 0.2 PVI for anisotropic random field.

θ_1	N_i	N_c	# Points	Error mean	Error std
1×10^{-3}	8	79	6199	1.14×10^{-3}	4.69×10^{-2}
1×10^{-4}	38	754	72243	6.95×10^{-4}	1.35×10^{-2}
5×10^{-5}	45	1044	96999	6.51×10^{-4}	1.01×10^{-2}

We are going to show that HDMR is indeed a versatile method where each sub-problem can be solved by any stochastic method. Therefore, we solve the problem at 0.4 PVI using HDMR where each sub-problem is solved with sparse grid based on Gauss-Legendre quadrature rule instead of ASGC. A level 3 sparse grid is chosen for each sub-problem. θ_1 is chosen as 1×10^{-5}. The results are shown in Fig. 5.24. The convergence of HDMR is given in Table. 5.5. In this

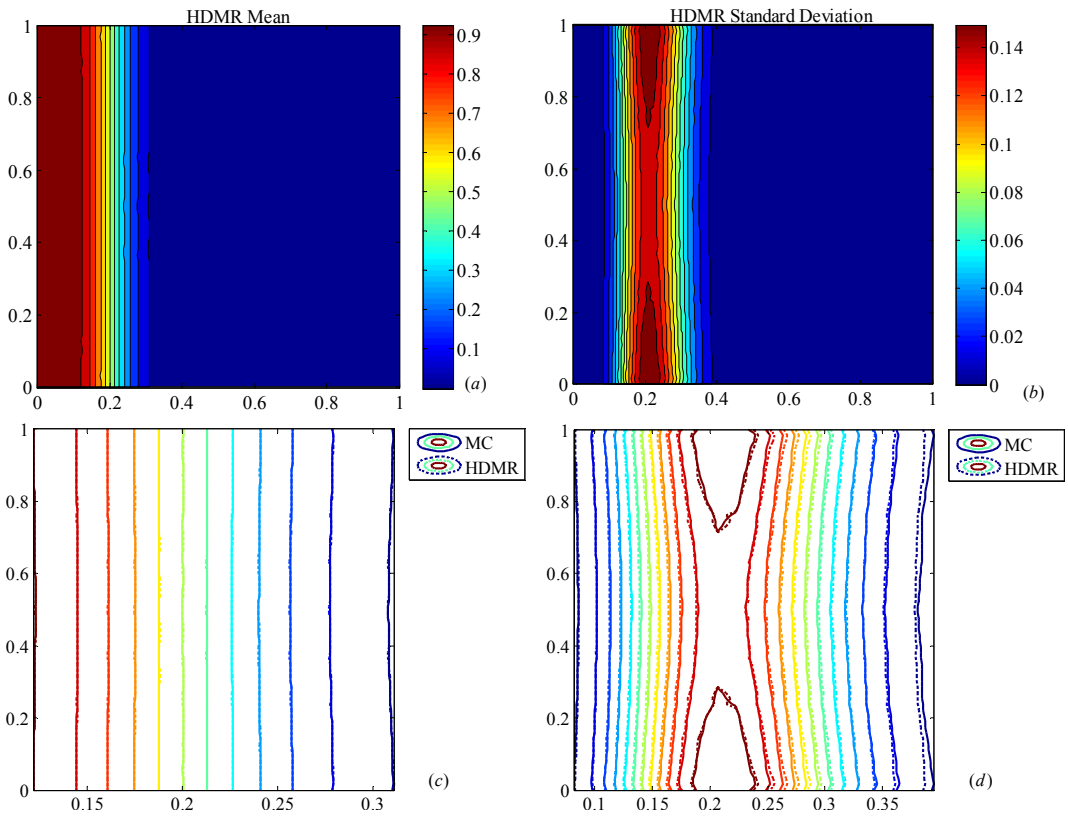


Figure 5.21: Mean and standard deviation of saturation at 0.2 PVI for anisotropic random field. Top: Mean (a) and standard deviation (b) form HDMR. Bottom: Comparison of mean (c) and standard deviation (d) between MC and HDMR near the saturation front.

extreme case, all the 50 dimensions are considered as important and the maximum expansion order is 4. This again is consistent with our previous results in [66]. Higher-order terms are needed to capture the large variability. Without adaptivity, there are 251176 component functions for a 4th order conventional HDMR. The advantage of adaptive HDMR is more impressive in this case. We also solve this problem directly with a 50-dimensional sparse grid based on Gauss-Legendre quadrature rule. The results from level 2 and 3 sparse grids are given in Fig. 5.25. Since the mean saturations are nearly the same, we only show the comparison between standard deviations. For level 2 sparse grid, the

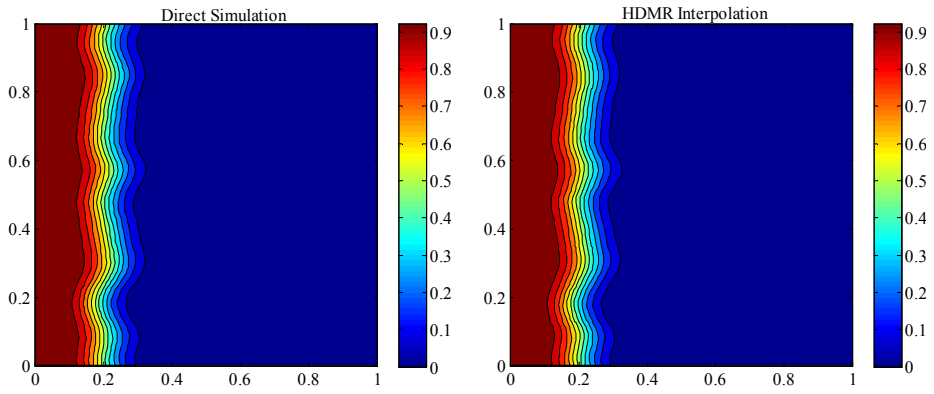


Figure 5.22: Prediction of the saturation profile using HDMR and the solution of the deterministic fine-scale problem with the same input for anisotropic random field. Left: Saturation at 0.2 PVI from direct simulation , Right: Saturation at 0.2 PVI reconstructed from HDMR.

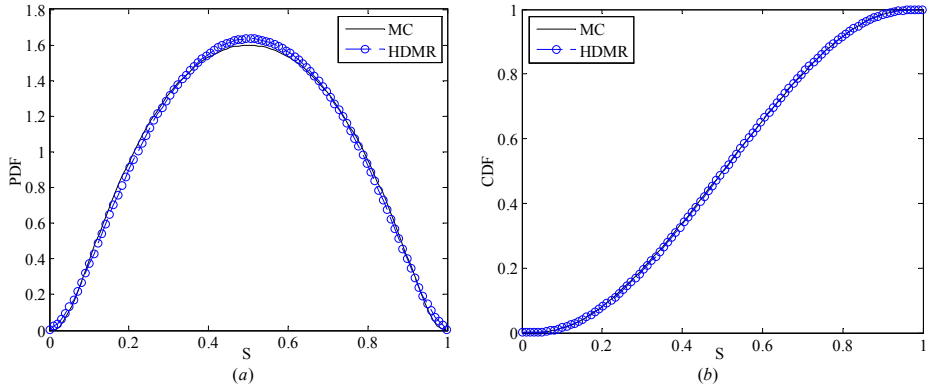


Figure 5.23: Anisotropic random field: (a) PDF of the saturation at point (0.2,0) and 0.2 PVI, (b) CDF of the saturation at point (0.2,0) and 0.2 PVI.

number of collocation points is 5301 with the mean error 8.31×10^{-4} and std error 4.38×10^{-2}. However, when increasing the sparse grid to level 3 with a total number of 192201 collocation points, the mean error increases to 1.90×10^{-3} and std error increases to 7.09×10^{-2}. In other words, the direct sparse grid method fails to converge. It is computationally impossible to increase the sparse grid level to 4 since it would thus require 5402401 collocation points. The failure of convergence may be due to the steep saturation gradient near the displacement front

which it is widely known that the polynomial based quadrature method has difficulty in convergence. From our results, it seems that the adaptive HDMR can reduce the irregularity of the stochastic space through decomposing the dimensions. However, a higher order expansion may be needed and thus increases the computational cost significantly. This interesting phenomenon derves further investigation in our ongoing future work.

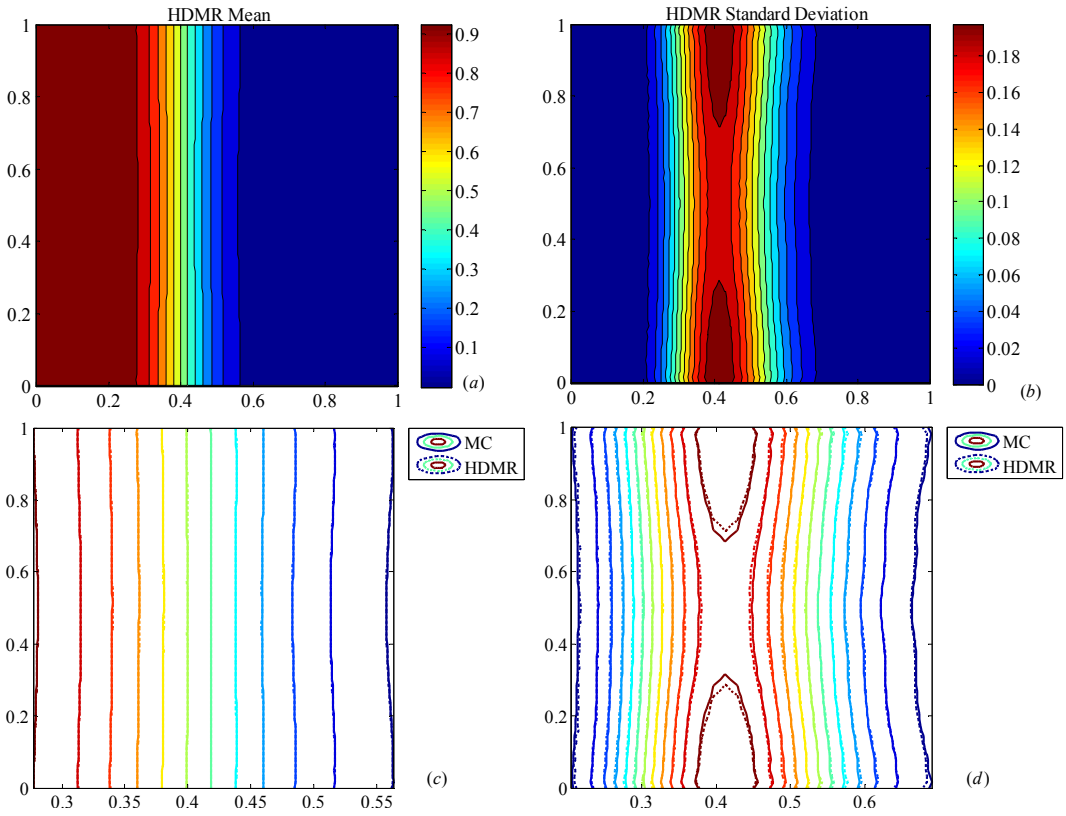


Figure 5.24: Mean and standard deviation of saturation at 0.4 PVI for anisotropic random field. Top: Mean (a) and standard deviation (b) form HDMR. Bottom: Comparison of mean (c) and standard deviation (d) between MC and HDMR near the saturation front. Here each sub-problem is solved using sparse grid based on Gauss-Legendre quadrature rule.

Finally, we want to comment on the computational time of this example. First, in Fig. 5.26, the convergence of standard deviation of the saturation at

Table 5.5: Convergence of HDMR with different θ_1 at 0.4 PVI for anisotropic random field.

θ_1	N_i	N_c	Order	# Points	Error mean	Error std
1×10^{-3}	10	96	2	4126	1.32×10^{-3}	5.17×10^{-2}
1×10^{-4}	38	763	3	54925	7.00×10^{-4}	4.10×10^{-2}
5×10^{-5}	45	1087	3	82407	6.40×10^{-4}	3.21×10^{-2}
1×10^{-5}	50	2050	4	218136	2.97×10^{-4}	1.97×10^{-2}

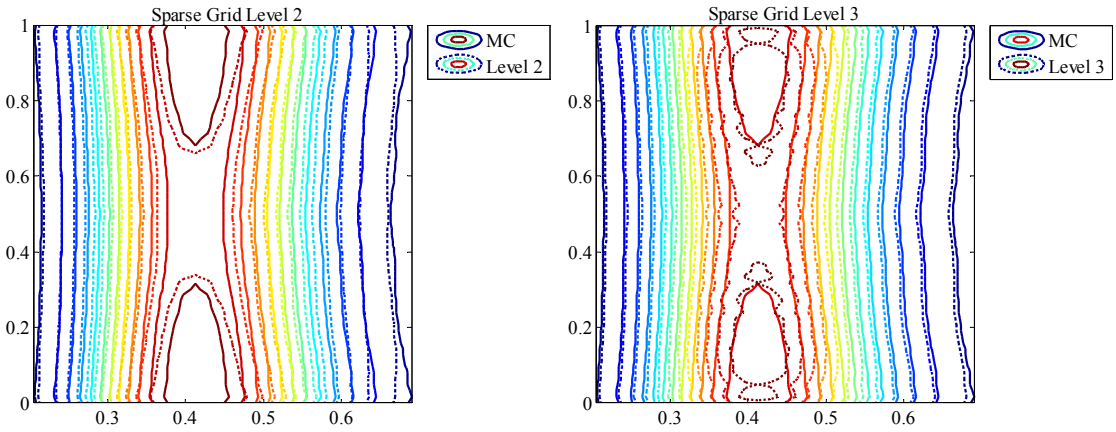


Figure 5.25: Standard deviation of saturation at 0.4 PVI for anisotropic random field using 50-dimensional sparse grid based on Gauss-Legendre rule: Comparison of standard deviation between MC and sparse grid level 2 (left) and 3 (right) near the saturation front.

one point with the number of MC simulations is given. The points are chosen at the place where the largest standard deviation occurs and they are different for different cases. From the figure, it is seen that at least 10^5 MC samples are needed in order to achieve statistical convergence. However, there are still some small oscillations after it. As is well known, the MC convergence rate is $M^{-1/2}$, therefore, to ensure a good comparison with HDMR, we use 10^6 samples eventually. It took about 19 hours on 60 processors while the average computational

time for HDMR is 5 hours on the same number of processors in such a high-dimensional case. It is also noted from the figure that much more points are needed to achieve statistical convergence in the anisotropic case which partially explains the larger variations of saturation as is seen before. Moreover, an interesting observation is that the shapes of the convergence plots are nearly the same at the two time instants for the same random input. This phenomenon suggests that although the convergence rate of MC is independent of number of the stochastic dimension, it does depend on the regularity of the stochastic input space. In general, more number of MC samples is needed for a stochastic space which is not smooth as is seen from the case of anisotropic random field.

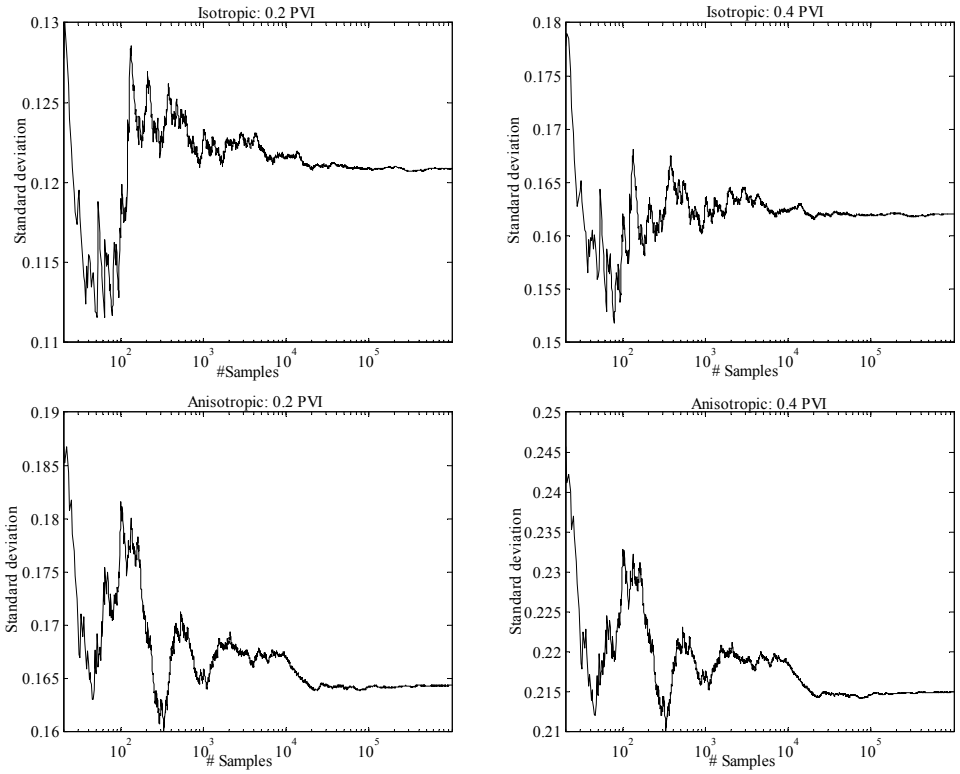


Figure 5.26: Standard deviation of the saturation at the point, where the largest value occurs, obtained from MC simulations versus the number of realizations.

5.5 Conslusions

In the first part of this chapter, a new multiscale methodology using the mixed finite element method is developed for the solution of elliptic equation arising from heterogeneous porous media problem. This multiscale framework is based on the framework of heterogeneous multiscale method which adds a new perspective into the area of numerical multiscale method. A novel boundary condition for the local cell problem is proposed which gives more realistic flow conditions across a coarse-element interface. In addition, a reconstruction method for the fine-scale velocity is also proposed, which ensure the continuity of the mass at both local and global scales. The first two numerical examples verifies the accuracy of the new method.

In the second part of this chapter, we consider the uncertainty quantification when the permeability field is modeled as a random field. The newly developed multiscale method is used as a direct solver within the framework of ASGC and HDMR. Our numerical results in Example 3 compare well with that of MC results with fine-scale solvers, which again verifies the accuracy of both multiscale and HDMR methods. Our study confirms the interesting phenomenon that the introduction of permeability heterogeneity leads to the heterogeneity-induced dispersion. Our results also indicates that the HDMR expansion can be served as an accurate surrogate model for the underlying problem.

CHAPTER 6

CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

In this thesis, we dealt with three important problems relevant to generic uncertainty quantification of complex systems on - (i) how to resolve the discontinuity or steep gradients in the stochastic space, (ii) how to address the problem of *curse of dimensionality* and (iii) how to incorporate uncertainty from different scales. To resolve all these issues, a complete computational strategy was developed. The key aspects of this framework are to utilize the adaptive high-dimensional stochastic model representation technique coupled with adaptive sparse grid collocation for solving the stochastic PDEs involved. The most important rational that has made the developed framework valuable is its non-intrusive character, where only repetitive deterministic simulations are required at a much less number of samples points than that of Monte Carlo method for the moderately high stochastic dimensional problems considered. Therefore, the process is trivially parallelizable, except for the final post-processing. During this work, we have developed a black box stochastic toolkit that can seamlessly link *any deterministic simulator* to facilitate stochastic analysis. In addition, a new mutiscale method for porous media flow is also developed. This is the first instance of a stochastic multiscale treatment of flow through random heterogeneous media that investigates the effect of random permeability on the flow transport properties.

Although the computational framework developed in this thesis works well for the numerical examples examined, there are still several areas where further developments and research are required. Suggestions for the continuation of this study are provided next.

6.1 Solution of SPDEs in high dimensions

The ASGC utilizing linear interpolation exhibits slow convergence rate. Therefore, there is a need to use high order polynomials. There are two possible extensions to resolve this issue.

According to the tree structure of the collocation points, the first extension is to utilize all the ancestors of the current collocation point in the previous level to build higher-order polynomials as in [64]. For example, in Fig. 3.3, all the ancestors of point 0.375 are $0.25, 0, 0.5$. Therefore, we use these four points to construct a Lagrange polynomial of order 4 as in the right plot of Fig. 6.1. The advantage of this method is that it is built on our current algorithm and is easy to implement. However, it is noted that the way to compute the hierarchical surplus in [64] is different from ours and that requires alternative methods for computing the surplus in the case of *hierarchical polynomial interpolation*.

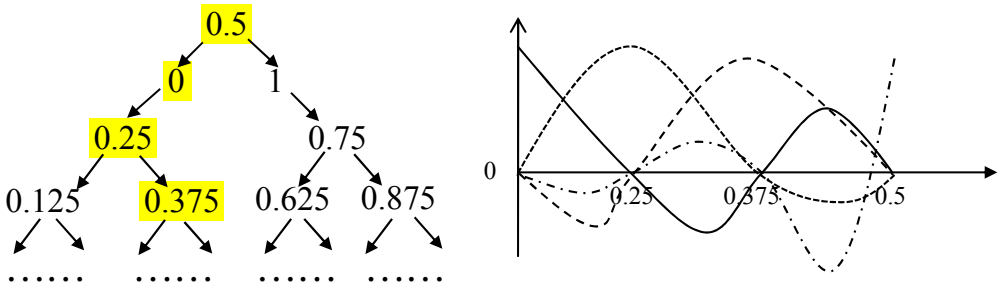


Figure 6.1: Schematic of the proposed hierarchical polynomial interpolation basis function: Left figure shows the needed hierarchical ancestor for point 0.375 if a fourth order polynomial is needed. The right figure shows the hierarchical basis polynomials using these collocation points.

Another possible extension is to use Lagrange polynomial interpolation directly. However, it is well known that Lagrange interpolation is unstable if using equidistant points. Since our ASGC algorithm is only built the equidistant

collocation points, we thus need to utilize a different adaptive strategy, the so called dimensional adaptive sparse grid interpolation. There exists similar work in [39]. The basic idea there is to use integration value of the function as an error indicator to detect the important dimension. If the dimension is considered as important, then we add all the next level collocation points along that dimension. This is different from our ASGC method where we only add local points around the current region. This method is independent of the choice of the collocation point and thus we can choose points which are better suited for Lagrange interpolation. Since the method in [39] is based on numerical integration, we need to adjust their algorithm to our sparse grid hierarchical interpolation based method. Finally, it is noted that the proposed method is quite different from the so called anisotropic sparse grid collocation method [41]. The work in [41] does not utilize the concept of hierarchical interpolation that can explore efficiently the inherent structure of the underlying stochastic sytem.

The stochastic HDMR is an ideal tool for dealing with problems of high stochastic dimension. When combined with ASGC, this method to our knowledge is the first approach which can solve high-dimensional stochastic problems by reducing the dimensions from truncation of HDMR and resolve low-regurlarity by local adaptivity through ASGC. Currently, we choose the same error threshold ε of ASGC for all components. However, in general, the regularity conditions vary for the different order component functions. For example, a small ε is enough for the one-dimensional component function while a larger ε is needed for high-order component functions. Therefore, a possible extension is to adapt strategy which can adjust the ε automatically between different component functions. In a similar way, the weight threshold θ_1 is also needed to be tuned for different order expansions in order to achieve the best possi-

ble computational performance. In addition, the current criteria to choose the important dimension is in some sense heuristic. Stochastic sensitivity analysis may be used to rank the importance of all random inputs [114, 115].

6.2 Multiscale modeling of flow through heterogeneous porous media

In this thesis, as a first step towards this new method, only a single-phase flow and transport problem is considered. A straight forward extension is to model the effect of uncertain permeability on the transport properties of multi-phase flow. It is well known that the saturation front of the multi-phase flow exhibits strong discontinuity, which will certainly increase the computational effort. In addition, unlike single-phase flow, the velocity and saturation equations are tightly coupled in multi-phase flow, where we need to solve for the velocity at each time step when the saturation field is updated. Thus, there is a need to develop adaptive time-integration methods as in [79, 87]. Another issue with the multi-phase flow is in the case of permeability without scale separation. The solution is to use the global flow boundary conditions [79]. In our case, enlarging the sampling domain to include several coarse elements may be an alternative solution.

It is also natural to include well modeling in the multiscale method [86]. To investigate the effect of the random location of the source/wells on the distribution of the saturation profile is a very interesting but realistic topic which has not been discussed before.

6.3 Stochastic input model reduction using kernel PCA

The uncertainty involved in the current thesis is only from the analytic K-L expansion. It is more interesting to consider data-driven stochastic input models from experimental data such as in [71]. However, the non-linear dimensionality reduction methods developed in [116] do have two potential limitations. First, they do not give us the inherent patterns (the eigenfunctions as in the K-L expansion) in the embedded random space. Therefore, they can not provide us a mathematical parametric input model as in K-L expansion, i.e. we want to find the form $\boldsymbol{y} = f(\boldsymbol{\xi})$, where vector $\boldsymbol{y}$ is a realization of a discrete random field, and vector $\boldsymbol{\xi}$, of dimension much smaller than the original input stochastic dimension, is a set of independent random variables with a specified distribution. Secondly, the method in [116] only preserves some notion of the geodesic distances between the sample random fields. This definition of distance does not consider the actual stochastic systems and thus its performance varies dramatically for different problems. In practice, the target stochastic system is always known to us and there is a need to define a distance which is best to determine how similar two realizations of the random field are in terms of spatial properties and the *corresponding stochastic system response*.

A possible remedy to this problem is to use kernel principal component analysis (KPCA) [117, 118]. KPCA has proven to be a powerful tool as a nonlinear feature extractor of classification algorithm [117], pattern recognization [119], image-denosing [120] and statistical shape analysis [121]. Recently, it has been also applied in reservoir engineering in the context of inversion of flow data [122, 123].

Fig. 6.2 demonstrates the basic idea behind nonlinear kernel PCA. Consider an arbitrary random field in R^2, i.e. each realization is a vector of dimension two, $\boldsymbol{y} = (y_1, y_2)^T$. If $\boldsymbol{y}$ is non-Gaussian , y_1 and y_2 can then be nonlinearly related to each other in R^2 (Fig. 6.2, left). If linear PCA or the standard K-L expansion were used, the relationship obtained between y_1 and y_2 would again be modeled as linear, which is clearly not right. Now consider a nonlinear mapping Φ that relates the input space R^2 to another space F

$$\Phi : R^2 \rightarrow F; \quad \mathbf{Y} = \Phi(\boldsymbol{y}); \quad \boldsymbol{y} \in R^2, \mathbf{Y} \in F \tag{6.1}$$

F is called feature space. In the right plot of Fig. 6.2, after this Φ transform, the realizations that were nonlinearly related in R^2 becomes linearly related in the feature space F. Standard linear PCA or the K-L expansion can now be performed in F in order to determine the principal eigenvectors in this space. Now,

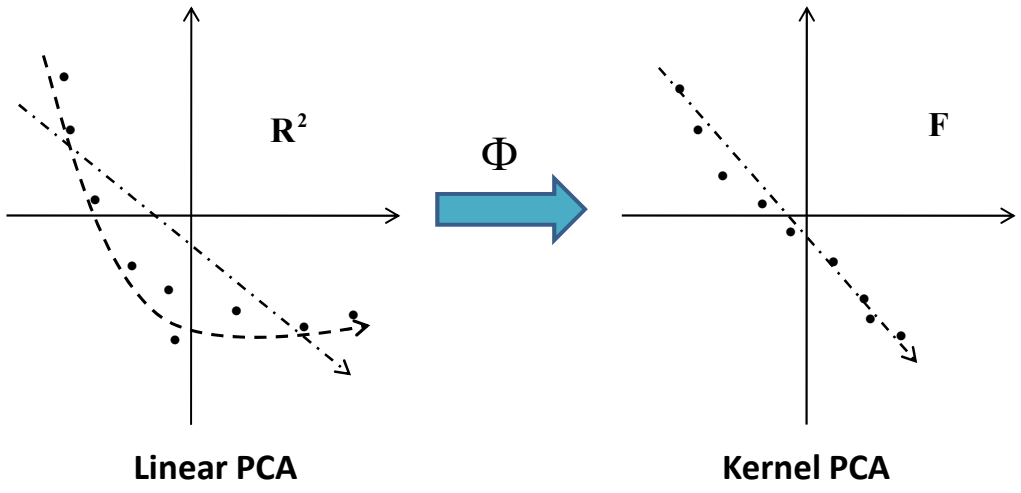


Figure 6.2: Basic idea behind kernel PCA. Left: In this non-Gaussian case, the linear PCA is not able to capture the nonlinear relationship among the realizations in the original space. Right: After the nonlinear mapping Φ, the realizations become linearly related in the feature space F. Standard linear PCA or K-L expansion can now be performed in F.

assume we are given N number of realizations of a random field $\boldsymbol{y}_k, k = 1, \ldots, N$,

where each realization is representation as a high-dimensional vector $\boldsymbol{y}_k \in \mathbb{R}^M$ (e.g. M can be considered as the number of grid points in the discretization). In general, we have $M \gg N$. Realizing that nonlinear kernel PCA is essentially linear PCA in a high dimensional feature space F (as opposed to the input space R^M), all results on linear PCA can be readily generalized for general kernel PCA. The maps of the realizations $\boldsymbol{y}_k, k = 1, \ldots, N_R$ in the feature space F are $\Phi(\boldsymbol{y}_k), k = 1, \ldots, N_R$, and assuming $\Phi(\boldsymbol{y}_k)$ are centered (if not, they can be centered as in [117, 121]), analogous to linear PCA, a kernel eigenvalue problem is formulated as $N_R \lambda \boldsymbol{\alpha} = \mathbf{K}\boldsymbol{\alpha}$. The kernel matrix $\mathbf{K}$ is defined as

$$\mathbf{K}: \quad K_{ij} = \Big(\Phi(\boldsymbol{y}_i) \cdot \Phi(\boldsymbol{y}_j)\Big), \quad i, j = 1, \ldots, N_R \tag{6.2}$$

One significant advantage of using kernels is that there is no need to map explicitly the points from input space $\mathbb{R}^M$ to F; all necessary computations in space F can be carried out using the scalar product of the nonlinear function Φ. This function is called a kernel function and is given by

$$k(\boldsymbol{x}, \boldsymbol{y}) = (\Phi(\boldsymbol{x}) \cdot \Phi(\boldsymbol{y})) \tag{6.3}$$

The kernel function $k(\boldsymbol{x}, \boldsymbol{y})$ calculates the dot product in the feature space F directly from the elements of the input space $\mathbb{R}^M$. The most common kernel function is the Gaussian kernel (radial basis function), which is given by

$$k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\mathrm{d}^2(\boldsymbol{x}, \boldsymbol{y})}{2\sigma^2}\right) \tag{6.4}$$

where $d^2(\boldsymbol{x}, \boldsymbol{y})$ is a distance measure in the input space, which is not necessary a geodesic distance between samples.

The k-th orthonormal eigenvector of the kernel matrix in the feature space can then be shown to be [117] $\boldsymbol{V}_k = \sum_{i=1}^{N_R} \frac{\alpha_{ki}}{\sqrt{\lambda_k}} \Phi(\boldsymbol{y}_i)$ where $\boldsymbol{\alpha}_i = [\alpha_{1i}, \alpha_{2i}, \ldots, \alpha_{N_R i}]^T$

and λ_k are the kth eigenvector and eigenvalue of the kernel matrix $\mathbf{K}$, respectively. Since the K-L expansion is performed in the high dimensional feature space F, the results of the K-L expansion (realizations) thus lie in the feature space, that is, a simulated realization $\mathbf{Y} \in F$. Any realization $\mathbf{Y}$ is a linear combination of maps $\Phi(\boldsymbol{y}_k)$ of the input realizations $\boldsymbol{y}_k, k = 1, \ldots, N_R$ [122]. We can project $\mathbf{Y}$ onto the subspace spanned by the first largest n eigenvectors $\mathbf{Y} = \sum_{i=1}^{n} \boldsymbol{\xi}_i \boldsymbol{V}_i$ where $\boldsymbol{\xi}$ are independent random variables. Therefore, by drawing samples of $\boldsymbol{\xi}$ we obtain different realizations of the underlying random field in the feature space F. Now the dimensionality of the stochastic space successfully reduces to a small number n.

We successfully construct a low-dimensional stochastic input model $\boldsymbol{V}_k = \sum_{i=1}^{N_R} \frac{\alpha_{ki}}{\sqrt{\lambda_k}} \Phi(\boldsymbol{y}_i)$ in the feature space F. However, we are interested in obtaining realizations in the original space of the input random field $\mathbb{R}^M$ because the goal is to obtain a parameterization of the input random field and use it as the input model to our SPDEs. In order to obtain a realization $\boldsymbol{y}$ in the original space of the realizations R^M that corresponds to this simulated realization $\mathbf{Y} \in F$, an inverse Φ map of $\mathbf{Y}$ is required as $\boldsymbol{y} = \Phi^{-1}(\mathbf{Y})$. This is known as the pre-image problem [117, 120, 121]. A fixed point iterative algorithm was proposed in [120] that however was shown not to be very robust. A possible solution is to use local linear interpolation between k-nearest neighbors [116]. Another issue is that we do not know the random distribution of the variables in advance. Thus we need to find a way to estimate the probability density function of the random vector $\boldsymbol{\xi}$. The Maximum likelihood or maximum entropy method can be used to compute the probability distribution function of the underlying random field given only the realizations of the random field [124, 125].

BIBLIOGRAPHY

[1] R. L. Iman, W. J. Conover, Small sample sensitivity analysis techniques for computer models, with an application to risk assessment, Commun. Statist. (1980) 1749–1842.

[2] D. Zhang, Stochastic Methods for Flow in Porous Media: Coping with Uncertainties, Academic Press, San Diego, CA, 2002.

[3] D. Zhang, Z. Lu, An efficient, high-order perturbation approach for flow in random porous media via karhunen-love and polynomial expansions, Journal of Computational Physics 194 (2004) 773 – 794.

[4] Z. Lu, D. Zhang, A comparative study on uncertainty quantification for flow in randomly heterogeneous media using monte carlo simulations and conventional and kl-based moment-equation approaches, SIAM Journal on Scientific Computing 26 (2004) 558–577.

[5] R. Ghanem, P. D. Spanos, Stochastic Finite Elements: A Spectral Approach, Springer - Verlag, New York, 1991.

[6] N. Wiener, The homogeneous chaos, American Journal of Mathematics 60 (1938) 897–936.

[7] R. Ghanem, Probabilistic characterization of transport in heterogeneous media, Computer Methods in Applied Mechanics and Engineering 158 (1998) 199 – 220.

[8] A. Sarkar, R. Ghanem, Mid-frequency structural dynamics with parameter uncertainty, Computer Methods in Applied Mechanics and Engineering 191 (2002) 5499 – 5513.

[9] R. Ghanem, Higher-order sensitivity of heat conduction problems to random data using the spectral stochastic finite element method, Journal of Heat Transfer 121 (1999) 290–299.

[10] D. Xiu, G. E. Karniadakis, The wiener–askey polynomial chaos for stochastic differential equations, SIAM Journal on Scientific Computing 24 (2002) 619–644.

[11] D. Xiu, G. E. Karniadakis, Modeling uncertainty in steady state diffusion

problems via generalized polynomial chaos, Computer Methods in Applied Mechanics and Engineering 191 (43) (2002) 4927 – 4948.

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

[13] D. Xiu, G. E. Karniadakis, A new stochastic approach to transient heat conduction modeling with uncertainty, International Journal of Heat and Mass Transfer 46 (2003) 4681 – 4693.

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

[15] M. K. Deb, I. M. Babuska, J. T. Oden, Solution of stochastic partial differential equations using galerkin finite element techniques, Computer Methods in Applied Mechanics and Engineering 190 (2001) 6359 – 6372.

[16] I. Babuska, R. Tempone, G. E. Zouraris, Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation, Computer Methods in Applied Mechanics and Engineering 194 (2005) 1251 – 1294.

[17] B. V. Asokan, N. Zabaras, Using stochastic analysis to capture unstable equilibrium in natural convection, Journal of Computational Physics 208 (2005) 134 – 153.

[18] O. P. L. Matre, O. M. Knio, H. N. Najm, R. G. Ghanem, Uncertainty propagation using wiener-haar expansions, Journal of Computational Physics 197 (2004) 28 – 57.

[19] O. P. L. Matre, H. N. Najm, R. G. Ghanem, O. M. Knio, Multi-resolution analysis of wiener-type uncertainty propagation schemes, Journal of Computational Physics 197 (2004) 502 – 531.

[20] X. Wan, G. E. Karniadakis, An adaptive multi-element generalized polynomial chaos method for stochastic differential equations, Journal of Computational Physics 209 (2005) 617 – 642.

[21] X. Wan, G. E. Karniadakis, Multi-element generalized polynomial chaos

for arbitrary probability measures, SIAM Journal on Scientific Computing 28 (2006) 901–928.

[22] X. Wan, G. E. Karniadakis, Long-term behavior of polynomial chaos in stochastic flow simulations, Computer Methods in Applied Mechanics and Engineering 195 (2006) 5582 – 5596.

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

[24] S. Smolyak, Quadrature and interpolation formulas for tensor product of certain classes of functions, Soviet Math. Dokl. 4 (1963) 240243.

[25] T. Gerstner, M. Griebel, Numerical integration using sparse grids, Numerical Algorithms 18 (1998) 209232.

[26] V. Barthelmann, E. Novak, K. Ritter, High dimensional polynomial interpolation on sparse grids, Advances in Computational Mathematics 12 (2000) 273288.

[27] A. Klimke, B. Wohlmuth, Algorithm 847: Spinterp: piecewise multilinear hierarchical sparse grid interpolation in matlab, ACM Trans. Math. Softw. 31 (2005) 561–579.

[28] A. Klimke, Uncertainty modeling using fuzzy arithmetic and sparse grids, Ph.D. thesis, Universitt Stuttgart, Shaker Verlag, Aachen (2006).

[29] M. Griebel, Adaptive sparse grid multilevel methods for elliptic pdes based on finite differences, Computing 61 (1998) 151–179.

[30] H.-J. Bungartz, M. Griebel, Sparse grids, Acta Numerica 13 (2004) 147–269.

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

[32] D. Xiu, Efficient collocational approach for parametric uncertainty analysis, Communications in Computational Physics 2 (2007) 293–309.

[33] F. Nobile, R. Tempone, C. G. Webster, A sparse grid stochastic collocation method for partial differential equations with random input data, SIAM Journal on Numerical Analysis 46 (2008) 2309–2345.

[34] J. Foo, X. Wan, G. E. Karniadakis, The multi-element probabilistic collocation method (me-pcm): Error analysis and applications, Journal of Computational Physics 227 (2008) 9572 – 9595.

[35] D. Donoho, Interpolating wavelet transforms, Preprint, Standford University, 1992.

[36] S. Bertoluzza, G. Naldi, A wavelet collocation method for the numerical solution of partial differential equations, Applied and Computational Harmonic Analysis 3 (1996) 1 – 9.

[37] O. V. Vasilyev, S. Paolucci, A dynamically adaptive multilevel wavelet collocation method for solving partial differential equations in a finite domain, Journal of Computational Physics 125 (1996) 498 – 512.

[38] O. V. Vasilyev, S. Paolucci, A fast adaptive wavelet collocation algorithm for multidimensional pdes, Journal of Computational Physics 138 (1997) 16 – 56.

[39] T. Gerstner, M. Griebel, Dimensionadaptive tensorproduct quadrature, Computing 71 (2003) 65 – 87.

[40] B. Ganapathysubramanian, N. Zabaras, Sparse grid collocation schemes for stochastic natural convection problems, Journal of Computational Physics 225 (2007) 652 – 685.

[41] F. Nobile, R. Tempone, C. G. Webster, An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data, SIAM Journal on Numerical Analysis 46 (2008) 2411–2442.

[42] M. Griebel, Sparse grids and related approximation schemes for high dimensional problems, in: Proceedings of the Conference on Foundations of Computational Mathematics, Santander, Spain, 2005.

[43] H. Rabitz, O. F. Alis, J. Shorter, K. Shim, Efficient input–output model representations, Computer Physics Communications 117 (1999) 11 – 20.

[44] H. Rabitz, O. F. Alis, General foundations of high-dimensional model representations, Journal of Mathematical Chemistry 25 (1999) 197 – 233.

[45] O. F. Alis, H. Rabitz, Efficient implementation of high dimensional model representations, Journal of Mathematical Chemistry 29 (2001) 127 – 142.

[46] S. W. Wang, H. L. II, G. Li, H. Rabitz, Fully equivalent operational models for atmospheric chemical kinetics within global chemistry-transport models, Journal Of Geophysical Research 104 (1999) 30417 – 30426.

[47] G. Li, C. Rosenthal, H. Rabitz, High dimensional model representations, The Journal of Physical Chemistry A 105 (2001) 7765–7777.

[48] I. M. Sobol', Theorems and examples on high dimensional model representation, Reliability Engineering and System Safety 79 (2003) 187 – 193.

[49] G. Li, S.-W. Wang, H. Rabitz, S. Wang, P. Jaff, Global uncertainty assessments by high dimensional model representations (hdmr), Chemical Engineering Science 57 (2002) 4445 – 4460.

[50] S. Balakrishnan, A. Roy, M. G. Ierapetritou, G. P. Flach, P. G. Georgopoulos, A comparative assessment of efficient uncertainty analysis techniques for environmental fate and transport models: application to the fact model, Journal of Hydrology 307 (2005) 204 – 218.

[51] S. Rahman, H. Xu, A univariate dimension-reduction method for multidimensional integration in stochastic mechanics, Probabilistic Engineering Mechanics 19 (2004) 393 – 408.

[52] H. Xu, S. Rahman, A generalized dimension-reduction method for multidimensional integration in stochastic mechanics, International Journal for Numerical Methods in Engineering 61 (2004) 1992 – 2019.

[53] H. Xu, S. Rahman, Decomposition methods for structural reliability analysis, Probabilistic Engineering Mechanics 20 (2005) 239 – 250.

[54] S. Rahman, A dimensional decomposition method for stochastic fracture mechanics, Engineering Fracture Mechanics 73 (2006) 2093 – 2109.

[55] R. Chowdhury, B. N. Rao, A. M. Prasad, High-dimensional model representation for structural reliability analysis, Communications in Numerical Methods in Engineering 25 (2009) 301–337.

[56] J. Foo, G. E. Karniadakis, Multi-element probabilistic collocation method in high dimensions, Journal of Computational Physics 229 (2010) 1536 – 1557.

[57] M. Grieble, M. Holtz, Dimension-wise integration of high-dimensional functions with applications to finance, Tech. Rep. INS Prepreint NO. 0809, University of Bonn (November 2008).

[58] W. E, B. Engquist, The heterogeneous multi-scale methods, Comm. Math. Sci 1 (2002) 87–132.

[59] E. Weinan, B. Engquist, X. Li, W. Ren, E. Vanden-Eijnden, Heterogeneous multiscale methods: A review, Communications in Computational Physics 2 (2007) 367–450.

[60] M. Loève, Probability Theory, fourth ed., Berlin: Springer-Verlag, 1977.

[61] X. Ma, N. Zabaras, An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations, Journal of Computational Physics 228 (2009) 3084 – 3113.

[62] B. Oksendal, Stochastic Differential Equations: An introduction with applications, Springer-Verlag, New York, 1998.

[63] X. Ma, N. Zabaras, A stabilized stochastic finite element second-order projection method for modeling natural convection in random porous media, Journal of Computational Physics 227 (2008) 8448 – 8471.

[64] H.-J. Bungartz, S. Dirnstorfer, Multivariate quadrature on adaptive sparse grids, Computing 71 (2003) 89 – 114.

[65] A. Klimke, Sparse Grid Interpolation Toolbox – user's guide, Tech. Rep. IANS report 2007/017, University of Stuttgart (2007).

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

[67] F. Y. Kuo, I. H. Sloan, G. W. Wasilkowski, H. Wozniakowski, On the decompositions of multivariate functions, Mathematics of Computation 79 (2010) 953 – 966.

[68] X. Wang, On the approximation error in high dimensional model representation, in: Proceedings of the 2008 Winter Simulation Conference, Austin, TX, 2008.

[69] X. Ma, N. Zabaras, An efficient bayesian inference approach to inverse problems based on an adaptive sparse grid collocation method, Inverse Problems 25 (2009) 035013.

[70] A. GENZ, A package for testing multiple integration subroutines, Numerical Integration: Recent Developments, Software and Applications, 1987, pp. 337–340.

[71] B. Ganapathysubramanian, N. Zabaras, A stochastic multiscale framework for modeling flow through random heterogeneous porous media, Journal of Computational Physics 228 (2009) 591 – 618.

[72] X. Ma, N. Zabaras, A stochastic mixed finite element heterogeneous multiscale method for flow in porous media, Journal of Computational Physics, Under review.

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

[74] T. Y. Hou, X.-H. Wu, Z. Cai, Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients, Math. Comput. 68 (1999) 913–943.

[75] T. J. R. Hughes, Multiscale phenomena: Green's functions, the dirichlet-to-neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods, Computer Methods in Applied Mechanics and Engineering 127 (1995) 387 – 401.

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

[77] Z. Chen, T. Y. Hou, A mixed multiscale finite element method for elliptic problems with oscillating coefficients, Math. Comput. 72 (2003) 541–576.

[78] P. Raviart, J. Thomas, A mixed finite element method for 2-nd order elliptic problems, in: Galligani, E. Magenes (Eds.), Mathematical Aspects

of Finite Element Methods, Lecture Notes in Mathematics 606, Springer-Verlag, 1977, pp. 292–315.

[79] J. E. Aarnes, On the use of a mixed multiscale finite element method for greater flexibility and increased speed or improved accuracy in reservoir simulation, Multiscale Modeling and Simulation 2 (2004) 421–439.

[80] J. E. Aarnes, V. Kippe, K.-A. Lie, Mixed multiscale finite elements and streamline methods for reservoir simulation of large geomodels, Advances in Water Resources 28 (2005) 257 – 271.

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

[82] P. Jenny, S. H. Lee, H. A. Tchelepi, Adaptive multiscale finite-volume method for multiphase flow and transport in porous media, Multiscale Modeling and Simulation 3 (2005) 50–64.

[83] P. Jenny, S. Lee, H. Tchelepi, Adaptive fully implicit multi-scale finite-volume method for multi-phase flow and transport in heterogeneous porous media, Journal of Computational Physics 217 (2006) 627 – 641.

[84] T. Arbogast, Implementation of a locally conservative numerical subgrid upscaling scheme for two-phase darcy flow, Computational Geosciences 6 (2002) 453 – 481.

[85] T. Arbogast, Analysis of a two-scale, locally conservative subgrid upscaling for elliptic problems, SIAM Journal on Numerical Analysis 42 (2004) 576–598.

[86] R. Juanes, F.-X. Dub, A locally conservative variational multiscale method for the simulation of porous media flow with multiscale source terms, Computational Geosciences 12 (2008) 273–295.

[87] V. Kippe, J. E. Aarnes, K.-A. Lie, A comparison of multiscale methods for elliptic problems in porous media flow, Computational Geosciences 12 (2008) 377–398.

[88] W. Ren, W. E, Heterogeneous multiscale method for the modeling of complex fluids and micro-fluidics, Journal of Computational Physics 204 (2005) 1 – 26.

[89] X. Li, W. E, Multiscale modeling of the dynamics of solids at finite temperature, Journal of the Mechanics and Physics of Solids 53 (2005) 1650 – 1685.

[90] A. Abdulle, C. Schwab, Heterogeneous multiscale fem for diffusion problems on rough surfaces, Multiscale Modeling and Simulation 3 (2005) 195–220.

[91] A. Abdulle, A. Nonnenmacher, A short and versatile finite element multiscale code for homogenization problems, Computer Methods in Applied Mechanics and Engineering 198 (2009) 2839 – 2859.

[92] A. Abdulle, The finite element heterogeneous multiscale method: a computational strategy for multiscale pdes, GAKUTO Int. Ser. Math. Sci. Appl. 31 (2009) 135–184.

[93] W. E, P. Ming, P. Zhang, Analysis of the heterogeneous multiscale method for elliptic homogenization problems, Journal of the American Mathematical Society 18 (2004) 121–156.

[94] A. Abdulle, On a priori error analysis of fully discrete heterogeneous multiscale fem, Multiscale Modeling and Simulation 4 (2005) 447–459.

[95] A. Abdulle, A. Nonnenmacher, A posteriori error analysis of the heterogeneous multiscale method for homogenization problems, Comptes Rendus Mathematique 347 (2009) 1081 – 1086.

[96] Z. Chen, Multiscale methods for elliptic homogenization problems, Numerical Methods for Partial Differential Equations 22 (2006) 317 – 360.

[97] B. V. Asokan, N. Zabaras, A stochastic variational multiscale method for diffusion in heterogeneous random media, Journal of Computational Physics 218 (2006) 654 – 676.

[98] B. Ganapathysubramanian, N. Zabaras, Modeling diffusion in random heterogeneous media: Data-driven models, stochastic collocation and the variational multiscale method, Journal of Computational Physics 226 (2007) 326 – 353.

[99] X. F. Xu, A multiscale stochastic finite element method on elliptic problems involving uncertainties, Computer Methods in Applied Mechanics and Engineering 196 (2007) 2723 – 2736.

[100] J. E. Aarnes, Y. Efendiev, Mixed multiscale finite element methods for stochastic porous media flows, SIAM Journal on Scientific Computing 30 (2008) 2319–2339.

[101] P. Dostert, Y. Efendiev, T. Hou, Multiscale finite element methods for stochastic porous media flow equations and application to uncertainty quantification, Computer Methods in Applied Mechanics and Engineering 197 (2008) 3445 – 3455.

[102] Z. Chen, Q. Du, An upwinding mixed finite element method for a mean field model of superconducting vortices, Mathematical Modelling and Numerical Analysis 34 (2000) 687–706.

[103] L. J. Durlofsky, Upscaling of geocellular models for reservoir flow simulation: A review of recent progress, in: 7 th International Forum on Reservoir Simulation, 2003, pp. 23–27.

[104] F. Brezzi, M. Fortin, Mixed and Hybrid Finite Element Method, Springer - Verlag, New York, 1991.

[105] X. Yue, W. E, The local microscale problem in the multiscale modeling of strongly heterogeneous media: Effects of boundary conditions and cell size, Journal of Computational Physics 222 (2007) 556 – 572.

[106] Y. Gautier, M. Blunt, M. Christie, Nested gridding and streamline-based simulation for fast reservoir performance prediction, Computational Geosciences 3 (1999) 295 – 320.

[107] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, H. Zhang, PETSc Web page, http://www.mcs.anl.gov/petsc (2009).

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

[109] N. Zabaras, D. Samanta, A stabilized volume-averaging finite element method for flow in porous media and binary alloy solidification processes, Advances in Water Resources 60 (2004) 1103 – 1138.

[110] M. A. Christie, M. Blunt, Tenth spe comparative solution project: A com-

parison of upscaling techniques, SPE Reservoir Evaluation and Engineering 4 (2001) 308 – 317.

[111] G. Lin, A. Tartakovsky, An efficient, high-order probabilistic collocation method on sparse grids for three-dimensional flow and solute transport in randomly heterogeneous porous media, Advances in Water Resources 32 (2009) 712 – 722.

[112] G. Lin, A. Tartakovsky, Numerical studies of three-dimensional stochastic darcys equation and stochastic advection-diffusion-dispersion equation, Journal of Scientific Computing 43 (2010) 92 – 117.

[113] D. Zhang, H. Tchelepi, Stochastic analysis of immiscible two-phase flow in heterogeneous media, SPE Journal 4 (1999) 380 – 388.

[114] G. Lin, G. E. Karniadakis, Sensitivity analysis and stochastic simulations of non-equilibrium plasma flow, International Journal for Numerical Methods in Engineering 80 (2009) 738 – 766.

[115] J. Foo, S. Sindi, G. Karniadakis, Multi-element probabilistic collocation for sensitivity analysis in cellular signalling networks, IET Systems Biology 3 (2009) 239–254.

[116] B. Ganapathysubramanian, N. Zabaras, A non-linear dimension reduction methodology for generating data-driven stochastic input models, Journal of Computational Physics 227 (2008) 6612 – 6637.

[117] S. Bernhard, S. Alexander, M. Klaus-Robert, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation 10 (1998) 1299–1319.

[118] S. Bernhard, S. Alexander, Learning with kernels, MIT Press, 2002.

[119] S.-T. J, N. Cristianni, Kernel methods for pattern analysis, Cambridge University Press, 2004.

[120] S. Mika, S. Bernhard, S. Alexander, M. Klaus-Robert, M. Scholz, G. Ratsch, Kernel pca and de-noising in feature spaces, Advances in Neural Information Processing Systems 11, MIT Press, 1999, pp. 536–542.

[121] Y. R. Samuel, S. Dambreville, A. Tannenbaum, Statistical shape analy-

sis using kernel pca, Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning, p. 60641B.

[122] P. Sarma, L. J. Durlofsky, K. Aziz, Kernel principal component analysis for efficient, differentiable parameterization of multipoint geostatistics, Mathematical Geosciences 40 (2008) 3–32.

[123] C. Scheidt, J. Caers, Representing spatial uncertainty using distances and kernels, Mathematical Geosciences 41 (2009) 397–419.

[124] C. Desceliers, R. Ghanem, Maximum likelihood estimation of stochastic chaos representations from experimental data, International Journal for Numerical Methods in Engineering 66 (2006) 978 – 1001.

[125] S. Sankaran, N. Zabaras, A maximum entropy approach for property prediction of random microstructures, Acta Materialia 54 (2006) 2265 – 2276.