The Holy Trinity of Stochastic Modeling and Uncertainty Quantification: High-Dimensionality, Sparsity and Rare Events.

by

Xiu Yang

A dissertation submitted in partial fulfillment of the

requirements for the degree of Doctor of Philosophy

in the Division of Applied Mathematics at Brown University

PROVIDENCE, RHODE ISLAND

May 2014

Abstract of "The Holy Trinity of Stochastic Modeling and Uncertainty Quantification: High-Dimensionality, Sparsity and Rare Events." by Xiu Yang, Ph.D., Brown University, May 2014

In this thesis, we present new developments to the three most important issues in stochastic modeling and uncertainty quantification (UQ): high-dimensionality, sparsity and rare events.

With respect to *high-dimensionality*, we present an adaptive Analysis Of VAriance (ANOVA) method to construct the generalized polynomial chaos (gPC) expansion of the system hierarchically and apply this method to study compressible supersonic flow over rough surface in aerodynamics and a ring oscillator in circuit simulations. We demonstrate that for both examples even a draconian truncation of the ANVOA expansion leads to accurate solutions in terms of mean and standard deviation.

With respect to *sparsity*, we consider the gPC coefficients of the system as a "signal". If this signal is sparse, we employ compressive sensing method to "recover" it with high accuracy but low computational cost. We propose a method combining the reweighted ℓ_1 minimization and Chebyshev sampling strategy to compute the gPC expansion of stochastic partial differential equations (PDE) efficiently. This method is able to exploit information from a limited number of realizations, hence it is suitable for problems with expensive deterministic solver. Furthermore, we apply this method to construct the response surface of particle based simulation models and use this response surface in a Bayesian inference framework to identify parameters of molecular dynamics models. This approach provides a general framework for mescoscale model calibration, and it leads to parametric compression and dimensionality reduction.

With respect to *rare events*, we firstly consider the problem of narrow escape of a Brownian particle in a boundary domain. A potential is imposed to trap the particle in it. We use PDE constrained optimization method to optimize the shape of the potential, hence to maximize the mean first passage time. We also consider a stochastic problem with disparate correlations length in adjacent domains. Instead of solving the problem globally, we use domain decomposition method to reduce the complexity of the problem in each subdomain and use PDE constrained optimization

to obtain the solution. This method does not require an explicit expression of the information transferred across the interface of subdomains and can lead to accurate results.

© Copyright 2014 by Xiu Yang

This dissertation by Xiu Yang is accepted in its present form

by the Division of Applied Mathematics as satisfying the

dissertation requirement for the degree of Doctor of Philosophy.

Date_____ _____

George Em Karniadakis, Advisor

Recommended to the Graduate Council

Date_____ _____

Boris Rozovsky, Reader

Date_____ _____

Xiaoliang Wan, Reader

Approved by the Graduate Council

Date_____ _____

Peter M. Weber, Dean of the Graduate School

Acknowledgments

First of all, I would like to thank my advisor Professor George Em Karniadakis. It is George who provides me the opportunity to study in this world class research group at a prestigious university. During the journey to a Ph.D., I benefit tremendously from his unparalleled vision in science and insight in mathematics. I will always treasure his guidance which is a previous asset for me. He not only introduces to me cutting edge research directions, but also offers me numerous travelling opportunities and encourages me to collaborate with distinguished researchers in different research areas.

Secondly, I would like to express my thank to my thesis readers Professor Boris Rozovsky and Professor Xiaoliang Wan for sparing their time and providing valuable comments on the thesis. I would also like to thank my master advisor Professor Huazhong Tang from Peking University. Further, I am also privileged to take or audit courses by many professors at Brown University.

Further, I would like to thank my collaborators Professor Chryssostomos Chryssostomidis, Professor Changsheng Chen, Professor Luca Daniel, Professor Guang Lin, Professor Daniele Venturi, Professor Xiangxiong Zhang, Dr. Minseok Choi, Dr. Huan Lei, Mr. Zheng Zhang. I would like to thank those who provide inspiring ideas and help to my research including: Professor Chi-Wang Shu, Professor Emmanuel Candès, Professor Michael Saunders, Professor Hui Wang, Professor Alireza Doostan, Professor Jianfeng Lu, Professor Xiang Zhou, Professor Zhongqiang Zhang, Professor Yue Yu, Professor Pengfei Xue, Dr. Xiaodong Li, Dr. Xiangrui Meng, Dr. Ewout van der Berg, Dr. Xingjie Li, Dr. Zhen Li, Dr. Xin Bian, Dr. Xuejin Li, Mr. Mingge Deng, Mr. Zheng Zhang, Mr. Yuhang Tang, Mr. Xiaoxing Cheng, Miss Heyrim Cho. I would also like to thank all Crunch Group members and all my friends at Brown University.

Last but not the least, my utmost gratitude goes to my family members. My parents and parents in law are always proud of me and encourage me. My beloved wife Stella Chen sacrifices a lot to accompany me in this small city for six years. It would not have been possible for me to

finish the journey to a Ph.D. alone. I would also like to thank my host family Dee Bradley and Charles Bradley, Mari Anne Snow and Charlie Zechel. They make me feel at home in America.

This work was supported by the following grants:

- Brown graduate school fellowship
- Brown graduate school teaching assistant
- Department of Energy, DE-SC0002542
- Office of Naval Research, N00014-07-1-0446
- Department of Energy, DE-FG02-07ER25818
- Air Force MURI, FA9550-09-0613

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview of uncertainty quantification method

The past decade has witnessed the explosive development of *uncertainty quantification* (UQ). Based on the pioneering work by Ghanem [57], Karniadakis and Xiu [157], the spectral-based method for uncertainty quantification called *generalized polynomial chaos* (gPC) became tremendously popular in numerical methods for stochastic ordinary or partial differential equations. The enormous saving on the computational cost compared with Monte Carlo method, at least for low dimensions, makes this method a new favor for studying physical phenomena with randomness. Later Xiu and Hesthaven proposed a non-intrusive method called *probabilistic collocation method* (PCM) [156], which can be considered as a "smart Monte Carlo" method as it utilizes the sparse grid points. It treats the solver of specific problems as a black-box and users only need to run the solver with specifically selected sampling points then do the post processing. This method further enhances the power of the gPC method as it avoid the Galerkin projection, and can be applied to much wider areas. This powerful tool also improves methods in other areas. For example, the ensemble Karman filter method [90, 91], the Bayesian inference method [110, 109], etc.

However, there are two main concerns on the aforementioned method: *high-dimensionality* and

discontinuity. Unlike the physical space, which is usually three-dimensional, the dimension of the random space can be very high since a physical system may include many random parameters. Also, the popular Karhunen-Loève (KL) expansion [137] for representing random fields may include large number of random variables when the correlation length is very small with respect to the size of the physical domain. If we construct the gPC basis based on the tensor product rule, then the total number N of gPC expansion up to truncation order P with dimension D is $N = \begin{pmatrix} D+P \\ D \end{pmatrix}$, hence the number of basis increases exponentially as D increases. This make it prohibitive to apply gPC method when D is large. Similarly, the number of the sparse grid points increases exponentially as dimension increases which diminishes the advantage of this method over the Monte Carlo method. Therefore, due to the *curse of high-dimensionality*, both gPC and sparse grid method lose their efficiency. The other regards the discontinuity in the random space. Both gPC and sparse grid method are based on the interpolation approximation, therefore it is challenging if the function is of low regularity. The work by Wan and Karniadakis [149, 150] first addressed this problem and proposed the *multi-element* method, which decomposes the random space adaptively to build a fine mesh where the regularity of the object function is low. There are also works on this issue with sparse grid method, e.g., [11, 74]. However, the curse of high-dimensionality still constrains the efficiency of these methods.

1.2 Adaptive method for high-dimensional method

Since the efficiencies of gPC and sparse grid method are lower for high-dimensional problems, one choice for us is to retreat to the classical Monte Carlo method and its variants e.g., quasi-Monte Carlo. However, in practice, most physical systems are *sparse*, which means that in the gPC expansion for the system we only need a few terms to capture the main property of the system. Just a few years ago, the word *sparse*, which is widely used in signal processing and has become well known in UQ community, was not used in to this area, although the concept was used. Many works

discuss how to hierarchically build gPC expansion or to adaptive select the sparse grid points, e.g., [105, 106, 116, 103, 60, 14]. A useful approach to design the adaptivity criterion in the adaptive method is based on the sensitivity analysis of the system and this idea originates from applying the gPC to analyze the system's sensitivity, e.g., [18, 19, 138, 17]. A popular method in sensitivity analysis called Analysis Of VAriance (ANVOA) [134] has attracted attention in this respect. Based on this idea different types of adaptive criteria are proposed e.g., [107, 164, 93]. We present an adaptive method based on ANOVA in Chapter 2. We apply this method to a compressible flow problem and a ring oscillator model. For the compressible flow, the effects of random geometric perturbation (simulating random roughness) on the scattering of a strong shock wave is investigate both analytically and numerically. For the ring oscillator model in a circuit, the oscillation frequency is studied. Our study shows that for these systems, the mean and the standard deviation of the system can be captured accurately even with a draconian truncation of the ANOVA expansion.

1.3 Sparsity in stochastic modeling

The sparsity of the system means that if we approximate the system with gPC expansion with all the basis functions up to the truncation order, only a small portion of these functions make major contribution. In other words, if we consider the gPC coefficients as a vector, then only a small portion of the entries of it is relatively large. In signal processing processing and imaging processing community, the recovering of a sparse signal is an important topic. Based on the trigonometric or wavelet basis, the sound or an image can be compressed into a sparse *signal*. Here the sparsity is an intrinsic property of the sound or image, otherwise we will hear noise or see a chaotic image. The famous *compressive sensing* method has been successfully applied to recover the sparse signal. To the best of our knowledge, the earliest work on applying compressive sensing method to recover a sparse gPC expansion is [94, 39]. The latter is the first paper with this idea in the UQ community. Unlike the adaptive method, this method does not require design of any criterion. We include

all the basis function up to the truncation order and let the compressive sensing method pick important basis automatically. Here we need the sparsity of the physical system with respect to the gPC expansion. This assumption is reasonable since in many systems, even if these systems include many random variables, these random variables as well as the interactions of these random variables are not equally important. Otherwise, we will observe phenomena like white noise or chaos, which is not the problem we are investigating. In Chapter 3 we discuss the sparsity of the gPC expansion. We firstly apply this method to solve stochastic PDEs, the solutions of which are sparse based on analytical results. Then we apply this method to construct the response surface of particle models with a few simulations. With this response surface we utilize Bayesian inference to identify the parameters in these models. This is the first work with gPC based compressive sensing method to build surrogate model, which is employed to infer the parameter in a complex system.

1.4 Rare events

Rare events is an important topic physics, chemistry, biology, engineering, finance, etc. Rare means the probability of the appearance of the event is very small, e.g., $10^{-3}, 10^{-4}$ or even smaller. It is very useful in chemical engineering, climate, civil engineering, insurance, etc. The purpose is to compute the reaction rate, probability of catastrophe, accident, etc. Based on this study, we can find an approach to accelerate the reaction, select proper strength of the building material, set appropriate premium amount, etc. The most popular method in this area is the important sampling (IS), and a very useful theory is the large deviation theory (LDT), e.g., [33, 23, 147, 84, 35]. The gPC method can also be applied to compute the rare probability, e.g., [92, 89]. The main idea is to evaluate gPC surrogate model in the area where rare events do not appear while near the area the rare event take place, the exact model is evaluated based on important sampling. This type of problem can be considered as approximating a high-dimensional integral in special cases where the integration result is very small. Another type of rare events problems are related to the small

perturbations in the system. These perturbations may arises on the boundary, body force, etc. For example, in the study of chemical reactions, Langevin system and its variants are widely used to describe the motion of the particles; the source of perturbation in this case is the white noise in the system. Many efforts have been made to find the transition path or to accelerate the simulation of the chemical reactions, e.g., [7, 153, 124, 154, 111, 125]. Also, an interesting problem is the perturbation originating from the propagation of uncertainty across different domains or different scales. Hence in Chapter 4 we present a preliminary work on the domain decomposition method to solve stochastic PDEs in adjacent domains with different correlation lengths. This method is based on gPC expansion and PDE constrained optimization. Moreover, we address in this chapter the narrow escape problem. We maximize the mean first passage time of a Brownian particle from a boundary domain by applying PDE constraint optimization. Since the MFPT is related to the reaction rate in many problems the understanding of the optimal shape of the trapping potential is very useful.

Chapter 2

ANOVA Method for High-Dimensionality

2.1 Introduction to ANOVA method

The ANOVA decomposition was introduced by Fisher in 1921 (see e.g. [46]) and employed for studying U-statistics by Hoeffding in 1948 [71]. ANOVA has also previously been used in the context of uncertainty quantification in [155] and was formulated in terms of polynomial chaos for solving high-dimensional stochastic PDEs in [49, 105]. In particular, in [49] ANOVA was combined with a multi-element probabilistic collocation method (PCM) to represent each expansion term for greater control of accuracy and efficiency of the discrete representation. ANOVA involves splitting a multi-dimensional function into its contributions from different groups of sub-dimensions. The underlying idea involves the splitting of a one-dimensional function approximation space into the constant subspace and the remainder space. The associated splitting for multi-dimensional cases is formed via a tensor-product construction. In practice, one essentially truncates the ANOVA-type decomposition at a certain dimension ν, thereby dealing with a series of low-dimensional

($\leq \nu$) approximation problems in lieu of one high-dimensional problem. This type of truncation can make high-dimensional approximation tractable for functions with high nominal dimension N but only low-order correlations amongst input variables (i.e., $\nu \ll N$). However, even with severe truncations, e.g., $\nu = 2$, the ANOVA decomposition of a high-dimensional function includes many thousands of terms, e.g. for a 1000-dimensional function ($N = 1000$) and $\nu = 2$ we need to evaluate close to half a million terms. To this end, in the current work we aim to reduce further the associated computational complexity by adaptively evaluating the terms in the ANOVA expansion and keep only the most important ones. This approach can reduce the number of terms in the aforementioned example to about 100 – a reduction of more than three orders of magnitude! Clearly, the accuracy of the representation will also be reduced as a function of the truncation dimension and this is problem dependent, hence we need to consider different flow systems – both viscous and inviscid – in order to evaluate the relationship between economical adaptive ANOVA representations and accuracy in the main statistics, i.e. the mean and variance of the stochastic solution.

The ANOVA method is widely used in statistics. The same idea can be used for interpolation and integration of high dimensional problems as well as stochastic simulations ([134, 49]). Consider an integrable function $f(\boldsymbol{x})$, $\boldsymbol{x} = (x_1, x_2, \cdots, x_N)$ defined in $I^N = [0,1]^N$, then we have:

Definition 2.1.1. The representation of $f(\boldsymbol{x})$ in a form

$$f(\boldsymbol{x}) = f_0 + \sum_{s=1}^{N} \sum_{j_1 < \cdots < j_s} f_{j_1 \cdots j_s}(x_{j_1}, \cdots, x_{j_s}) \tag{2.1.1}$$

or equivalently

$$f(\boldsymbol{x}) = f_0 + \sum_{j_1 \leq N} f_{j_1}(x_{j_1}) + \sum_{j_1 < j_2 \leq N} f_{j_1,j_2}(x_{j_1}, x_{j_2}) + \cdots + f_{1,2,\cdots,N}(x_1, x_2, \cdots, x_N) \tag{2.1.2}$$

is called ANOVA decomposition, if

$$f_0 = \int_{I^N} f(\boldsymbol{x}) \mathrm{d}\mu(\boldsymbol{x}), \tag{2.1.3}$$

and

$$\int_I f_{j_1 \cdots j_s} \mathrm{d}\mu(x_{j_k}) = 0 \quad \text{for} \quad 1 \leq k \leq s. \tag{2.1.4}$$

Here $1 \leq j_1 < j_2 < \cdots < j_s \leq N, j \leq s \leq N$. We call $f_{j_1}(x_{j_1})$ the first-order term (or first-order component function), $f_{j_1,j_2}(x_{j_1,j_2})$ the second-order term (or second-order component function), etc.

Property 2.1.2. An important property of ANOVA decomposition is the orthogonality of its terms:

$$\int_{I^N} f_{j_1,\cdots,j_s} f_{k_1,\cdots,k_l} \mathrm{d}\mu(\boldsymbol{x}) = 0, \tag{2.1.5}$$

if $(j_1, \cdots, j_s) \neq (k_1, \cdots, k_l)$. This is a direct consequence of (2.1.4). The terms in the ANOVA-decomposition are computed as follows:

$$f_{_S} = \int_{I^{N-|S|}} f(\boldsymbol{x}) \mathrm{d}\mu(\boldsymbol{x}_{_{S^c}}) - \sum_{T \subset S} f_{_T}(\boldsymbol{x}_{_T}), \tag{2.1.6}$$

where $S = \{j_1, j_2, \cdots, j_s\}$.

Property 2.1.3. The variance of f is the sum of the variances of all the decomposition terms:

$$\sigma^2(f) = \sum_{s=1}^{N} \sum_{|S|=s} \sigma^2(f_{_S}), \tag{2.1.7}$$

where $\sigma^2(f_S) = \int_{I^N} f_S^2 d\mu(x)$. It is *very important* that (2.1.7) holds only when the measure used in the integral of computing the variance is the same as that used in the ANOVA decomposition (2.1.6). It is probable that $\sigma(f)$ is different from the exact value computed by the standard definition of

variance, i.e., the integral with Lebesgue measure.

Remark 2.1.4. When applying the ANOVA method to stochastic simulation, we denote the highest dimension for the component functions we use in (2.1.2) with ν and approximate $f(\boldsymbol{x})$ with

$$f(\boldsymbol{x}) \approx f_0 + \sum_{j_1 \leq N} f_{j_1}(x_{j_1}) + \sum_{j_1 < j_2 \leq N} f_{j_1,j_2}(x_{j_1}, x_{j_2}) + \cdots + \sum_{j_1 < j_2 < \cdots < j_\nu \leq N} f_{j_1,j_2,\cdots,j_\nu}(x_{j_1}, x_{j_2}, \cdots x_{j_\nu}). \tag{2.1.8}$$

Here N is called *nominal dimension*, ν is called the truncation or effective dimension, and we use μ to denote the number of collocation points, which coincide with the quadrature points for performing the integration in each dimension.

After we obtain the ANOVA decomposition of a function f, we can approximate the integration of each component function with the values of these functions at the quadrature points based on the corresponding weights. We can also use these quadrature points as the collocation (sampling points) in the probabilistic collocation method (PCM).

2.2 Anchored ANOVA

Computing the ANOVA decomposition, i.e., the constant term from (2.1.3) and high-order terms from (2.1.6), can be very expensive for high dimensional problems or complicated $f(\boldsymbol{x})$. Therefore, we use the Dirac measure instead of Lebesgue measure in integrations, i.e., $\mathrm{d}\mu(\boldsymbol{x}) = \delta(\boldsymbol{x}-\boldsymbol{c})\mathrm{d}\boldsymbol{x}$, $\boldsymbol{c} \in I^N$. The point "$\boldsymbol{c}$" is called "***anchor point***" and this method is called "***anchored-ANOVA***". Hence, for the constant term we have

$$f_0 = f(\boldsymbol{c}). \tag{2.2.1}$$

Different choice of anchor points can lead to different approximation of ANOVA decomposition to a function; the authors in [168] defined different weights depending on the norm for tensor product functions, proved that the anchor point satisfying certain condition minimizes the error estimate of the approximation of ANOVA decomposition and showed some numerical examples. It

has been shown in the stochastic problems [107, 159] that ANOVA decomposition with $nu = 2$ has a good accuracy if the anchor point is chosen as the mean with respect to the probability density function considered. To this end we choose the zero point as the anchor point in this paper.

2.3 Adaptive ANOVA

Although the standard ANOVA can potentially reduce the computational complexity substantially, it still requires a very large number of sampling points to compute all the terms for a high nominal dimension N. For example, for nominal dimension $N = 100$, the number of sampling points for truncation dimension $\nu = 2$ and number of collocation points per direction $\mu = 3$ is $1+3\times\begin{pmatrix}100\\1\end{pmatrix}+3^2\times\begin{pmatrix}100\\2\end{pmatrix} = 44851$. An efficient way of solving this problem is to develop an adaptive ANOVA decomposition. To this end, we replace the nominal dimension by an ***active dimension*** D_i for each subgroup, i.e., we modify (2.1.8) to be

$$f(\boldsymbol{x}) \approx f_0+\sum_{j_1\le D_1} f_{j_1}(x_{j_1})+\sum_{j_1<j_2\le D_2} f_{j_1,j_2}(x_{j_1},x_{j_2})+\cdots+\sum_{j_1<j_2<\cdots<j_\nu\le D_\nu} f_{j_1,j_2,\cdots,j_\nu}(x_{j_1},x_{j_2},\cdots x_{j_\nu}). \tag{2.3.1}$$

In the flow problems we consider here we truncate the expansion after $\nu = 2$ and in most cases we let $D_1 = N$. We then develop adaptivity criteria to determine the active dimension D_2; see also [107, 168]. These criteria depend on the low-order statistics, i.e., mean and variance of the first-order terms in the ANOVA decomposition since they are easy to compute. Specifically, if the nominal dimension is N, the number of collocation points needed for a simulation based on the first-order terms is $\mu N + 1$.

Next, we present how to obtain an estimate of mean and variance of first-order terms in the ANOVA decomposition. Consider $f_1(x_1)$ for instance. Let $\boldsymbol{c}_{-1} = (c_2, c_3, \cdots, c_N)$ and $q_1^1, q_1^2, \cdots, q_1^\mu$ be the quadrature points for one-dimensional approximation (the weights for these quadrature

points are generated as well, e.g., Gaussian quadrature). Since we use the Dirac measure when computing the component term, according to (2.1.6) we have the value of f_1 at the quadrature point:

$$f_1(q_1^1) = f(q_1^1, c_2, c_3, c_4, \cdots, c_N) - f_0, \tag{2.3.2}$$

and we can readily approximate the mean and standard deviation of f_1. By running the deterministic solver at the sampling point $(q_1^1, \boldsymbol{c}_{-1})$ we can obtain $f(q_1^1, c_2, c_3, c_4, \cdots, c_N)$, while the constant term f_0 is obtained by running the deterministic solver at the pre-selected anchor point. Next, we can compute $f_1(q_1^1)$ by (2.3.2) and since the weight for this sampling point can be computed based on the quadrature point q_1^1, the mean and variance of f_1 can be obtained. This process can be repeated for all quadrature points to obtain $f_1(q_1^i), i = 1, \cdots, \mu$.

Similarly we can compute the second-order terms, e.g., consider the term $f_{1,2}$ at the point (q_1^i, q_2^j) with $i, j = 1, \cdots, \mu$. We can use (2.1.6) to obtain:

$$f_{1,2}(q_1^i, q_2^j) = f(q_1^i, q_2^j, c_3, c_4, \cdots, c_N) - f_1(q_1^i) - f_2(q_2^j) - f_0, \tag{2.3.3}$$

where

$$f_1(q_1^i) = f(q_1^i, c_2, c_3, \cdots, c_N) - f_0,$$
$$f_2(q_2^j) = f(c_1, q_2^j, c_3, \cdots, c_N) - f_0,$$

and the weights w_i, w_j for these quadrature points are known.

We can now summarize the above steps in the following algorithm:

Now we have obtained the values of function f at all the collocation points and since each of the points is equipped with a corresponding weight we can estimate the mean, standard deviation

Algorithm 1 Adaptive (anchored) ANOVA

1: Select the anchor point $\boldsymbol{c}$ and run the deterministic solver at this sampling point to obtain the constant term in the ANOVA expansion (according to equation (2.2.1)) for the quantity of interest f_0.
2: Select μ collocation points q_1^i according to the probability measure and generate corresponding weights $w_i, i = 1, \cdots, \mu$. Run the deterministic solver at the sampling points to obtain all first-order terms in the ANOVA decomposition using equation (2.3.2).
3: Compute the necessary statistical values of mean and standard deviation required in the adaptivity criterion to determine the active dimension D_2 for selecting the important second-order terms.
4: Run the deterministic solver at the sampling points and compute the second-order terms in the ANOVA decomposition using equation (2.3.3).
5: Repeat the above steps, if needed, for further selection of high-order terms based on the computed statistical values from first- and second-order terms.

and other statistical values of f. For example, the mean of f is approximated by

$$\int_{I^N} f(\boldsymbol{x})\mathrm{d}x \approx \sum_i f(\boldsymbol{q}^i)w_i, \tag{2.3.4}$$

where $f(\boldsymbol{x}_i)$ is obtained by the deterministic solver, $\boldsymbol{q}^i$ are collocation points and w_i are corresponding weights.

Remark 2.3.1. In step 2 we compute all the first-order terms because we set the active dimension $D_1 = N$. This can be modified if $D_1 < N$. Also, by setting $\nu = 2$ we have that the highest order we use is two, but this can be modified if a larger ν is required; similar to step 3, we compute the necessary statistical values for the proper criterion to select higher order terms.

Remark 2.3.2. In practice, we do not need all the second-order terms in (2.3.1) so we use fewer terms as shown in step 3, e.g., for the applications in this paper we approximate $f(\boldsymbol{x})$ by:

$$f(\boldsymbol{x}) \approx f_0 + \sum_{j_1 \le D_1} f_{j_1}(x_{j_1}) + \sum_{(j_1,j_2)\in\mathcal{F}} f_{j_1,j_2}(x_{j_1}, x_{j_2}), \tag{2.3.5}$$

where

$$\mathcal{F} = \{(j_1, j_2) | j_1 < j_2 < D_2, f_{j_1,j_2} \text{ satisfies an adaptivity criterion.}\}$$

Similarly, if $\nu > 2$ we can continue using proper adaptivity criteria to decide set $\mathcal{G}, \cdots$ such that (2.3.1) can be simplified as

$$f(\boldsymbol{x}) \approx f_0 + \sum_{j_1 \leq D_1} f_{j_1}(x_{j_1}) + \sum_{(j_1,j_2)\in\mathcal{F}} f_{j_1,j_2}(x_{j_1}, x_{j_2}), \sum_{(j_1,j_2,j_3)\in\mathcal{G}} f_{j_1,j_2,j_3}(x_{j_1}, x_{j_2}, x_{j_3}) + \cdots \quad (2.3.6)$$

We list three possible adaptive criteria here:

Criterion 1: Let $T_1 = \sum_{j=1}^{N} \sigma^2(f_j)$, which is the sum of the variances of all the first-order terms. The active dimension D_2 should satisfy:

$$\sum_{j=1}^{D_2} \sigma^2(f_j) \geq pT_1, \quad (2.3.7)$$

where p is a proportionality constant with $0 < p < 1$ and is very close to 1. This criterion is similar to the criterion used in [24] where $\sigma^2(f)$ instead of T_1 is used on the right-hand-side of (2.3.7) and p is set to be 0.99. As an example, Figure 2.3 shows the standard deviation for the first-order terms $f_j(x_j)$ in the ANOVA decomposition of the extra lift and drag for the compressible problem we study in the current work. Since we are interested in these two variables we should consider them simultaneously. It is also clear that since we replace $\sigma^2(f)$ with T_1 we should use larger p, because typically the sum of variances of the first-order terms should be less than that of the function itself. According to Remark 2.3.2, we need to find a set $\mathcal{F}$, which is accomplished by computing

$$\eta_{j_1,j_2} = \frac{\sigma^2(f_{j_1,j_2})}{\sum_{j=1}^{D_1} \sigma^2(f_j)}, \quad (2.3.8)$$

and bounding η_{j_1,j_2} with a predefined error threshold θ_2^1 (superscript i means creterion i).

Criterion 2: Ma and Zabaras ([107]) use the mean of component function $f_{\boldsymbol{u}}$ as the indicator to determine the active ANOVA terms. Let

$$\eta_j = \frac{\mathbb{E}(f_j)}{f_0(\boldsymbol{x})}, \quad (2.3.9)$$

where the pre-defined error threshold θ_1^2 is used to bound η, i.e., $\eta \leq \theta_1^2$. If η_j are monotonically decreasing with respect to j or monotonically decreasing from some j and we set $D_2 \geq j$, then (2.3.9) can be equivalently written as

$$\sum_{i=j}^{D_2} \mathbb{E}(f_j) \geq p \sum_{j=1}^{N} \mathbb{E}(f_j), \tag{2.3.10}$$

where p is a proportionality constant with $0 < p < 1$. For the supersonic flow we study in the paper η_j are monotonically decreasing from some j. Several choices of p and corresponding D_2 are listed in Table 2.5. Then, for a further selection of the second-order terms, Ma and Zabaras also used the mean of the component functions:

$$\eta_{j_1,j_2} = \frac{\mathbb{E}(f_{j_1,j_2})}{\sum_{j=0}^{D_1} E(f_j)}, \tag{2.3.11}$$

where η_{j_1,j_2} is bounded by a pre-defined error threshold θ_2^2.

Remark 2.3.3. The description for Criterion 2 is different from that in [107] but they are equivalent for the oblique shock problem we study in this paper.

Criterion 3: Zhang et. al. proposed a criterion based on both mean and variance of component functions. Let

$$\gamma_j = \frac{\sigma^2(f_j)}{\mathbb{E}(f_j)^2}, \quad \text{if} \quad \mathbb{E}(f_j) \neq 0, \quad 1 \leq j \leq N. \tag{2.3.12}$$

A pre-defined threshold θ_1^3 is set and we only use the first-order terms with $\eta_j > \theta_1^3$. Similar to criterion 2, if η_j is monotonically decreasing with respect to j or monotonically decreasing from some j and we set $D_2 \geq j$ then (2.3.12) can equivalently be written as

$$\sum_{j=1}^{D_2} \eta_j \geq p(1 + \sum_{j=1}^{N} \eta_j), \tag{2.3.13}$$

When selecting second-order terms, we set a pre-defined threshold θ_2^3 and use the criterion

$$\frac{\gamma_j \gamma_k}{1 + \sum_{j=1}^{N} \gamma_j + \sum_{j \neq k} \gamma_j \gamma_k} \geq \theta_2^3, \tag{2.3.14}$$

to determine which terms to use.

Remark 2.3.4. When we employ the above criteria to applications we replace the mean and variance of component function f_j with their L_2 norm values on the physical domain.

Here we give a very simple example to demonstrate the procedure of estimating the mean and variance of a function by considering a 3-dimensional Sobol function.

Example 1.

$$f(\boldsymbol{x}) = \prod_{k=1}^{3} f^{(k)}(x_k), \quad f^{(k)}(x_k) = \frac{|4x_k - 2| + k^2}{1 + k^2}, \tag{2.3.15}$$

where x_k are i.i.d. random variables and $x_k \sim U[0, 1]$. It is easy to obtain that $\mathbb{E}(f) = 1, \sigma^2(f) = 0.1014$. For the anchored-ANOVA decomposition, we choose an anchor point $\boldsymbol{c} = (c_1, c_2, c_3)$ such that $f^{(k)}(c_k) = (0.1\lambda_k^2 + \tau_k^2)/\tau_k$ and as in Example 1, τ_k and λ_k^2 are the mean and variance of $f^{(k)}$. We set $\mu = 4$, i.e., for each x_k we use four sampling points $q_i, i = 1, 2, 3, 4$, which are selected as follows: q_1, q_2 are Gauss quadrature points on $[0, 0.5]$, and q_3, q_4 are Gauss quadrature points on $[0.5, 1]$. Notice that, this is only for demonstration purposes; for practical problems, one can either use quadrature points on the entire domain or select quadrature points based on the property of the problem itself.

We first compute f_0:

$$f_0 = f(\boldsymbol{c}) = \prod_{k=1}^{3} f^{(k)}(c_k) = \prod_{k=1}^{3} \frac{0.1\lambda_k^2 + \tau_k^2}{\tau_k} = 1.010.$$

Then, we compute $f_j(q_i), j = 1, 2, 3, i = 1, 2, 3, 4$, e.g.,

$$f_1(q_2) = f(q_2, c_2, c_3) - f_0.$$

Hence the mean and variance of f_j can be estimated:

$$\mathbb{E}(f_j) \approx \sum_{i=1}^{4} f_1(q_i)w_i, \quad \sigma^2(f_j) \approx \sum_{i=1}^{4} \left(f_j(q_i)\right)^2 w_i - \left(\sum_{i=1}^{4} f_j(q_i)w_i\right)^2.$$

The results are shown in Table 2.1 As in the paper we set $D_1 = N = 3$, then we can apply the

Table 2.1: Mean and variance of the first-order terms in anchored-ANOVA decomposition of $f(x)$ in (2.3.15).

	f_1	f_2	f_3
$\mathbb{E}$	$-8.3472e-3$	$-1.3449e-3$	$-3.3656e-4$
σ^2	$8.3611e-2$	$1.3566e-2$	$3.3982e-3$

criteria to find D_2.

Criterion 1:

$$\frac{\sum_{i=1}^{1}\sigma^2(f_i)}{\sum_{i=1}^{3}\sigma^2(f_i)} = 0.8313, \quad \frac{\sum_{i=1}^{2}\sigma^2(f_i)}{\sum_{i=1}^{3}\sigma^2(f_i)} = 0.9662, \quad \frac{\sum_{i=1}^{3}\sigma^2(f_i)}{\sum_{i=1}^{3}\sigma^2(f_i)} = 1,$$

or equivalently consider

$$\frac{\sigma^2(f_1)}{\sum_{i=1}^{3}\sigma^2(f_i)} = 0.8313, \quad \frac{\sigma^2(f_2)}{\sum_{i=1}^{3}\sigma^2(f_i)} = 0.1349, \quad \frac{\sigma^2(f_3)}{\sum_{i=1}^{3}\sigma^2(f_i)} = 0.0338.$$

So if we set $p = 0.85$ (or $\theta_1^1 = 0.2$ for instance) we have $D_2 = 1$ while if we set $p = 0.90$ (or $\theta_1^1 = 0.1$) we have $D_2 = 2$. From this criterion we observe that x_1 is the most important while x_3 is the least important.

Criterion 2:

$$\frac{|\mathbb{E}(f_1)|}{f_0} = 8.264e-3, \quad \frac{|\mathbb{E}(f_2)|}{f_0} = 1.332e-3, \quad \frac{|\mathbb{E}(f_3)|}{f_0} = 3.332e-4.$$

So if we set $\theta_1^2 = 1e-3$ we have $D_2 = 2$ while if we set $\theta_1^2 = 1e-4$ we have $D_2 = 3$. From this criterion we again observe that x_1 is the most important while x_3 is the least important.

Criterion 3:

$$\gamma_1 = 13/12, \quad \gamma_2 = 76/75, \quad \gamma_3 = 301/300.$$

Here we see that *criterion 3* gives the same prediction as *criterion 1* or *2*.

According to *criterion 1* or *2*, $D_2 = 1$ or $D_2 = 2$ for different thresholds. When $D_2 = 1$, no second-order terms will be included. Therefore, the approximation for f is :

$$f(\boldsymbol{x}) \approx f_0 + f_1(x_1) + f_2(x_2) + f_3(x_3).$$

Now we can approximate $\mathbb{E}(f)$ by the sum of the mean of all the component functions and so we approximate $\sigma^2(f)$ similarly. The error of $\mathbb{E}(f)$ is $1.43e-5$ and the error of $\sigma^2(f)$ is $8.62e-4$.

In order to include the second-order terms, we proceed as follows.

Criterion 1: since $D_2 = 2$ we only need $f_{1,2}$ and if $D_2 = 3$ we compute

$$\frac{\sigma^2(f_{1,2})}{\sum_{i=1}^{3}\sigma^2(f_i)} = 1.107e-2, \quad \frac{\sigma^2(f_{1,3})}{\sum_{i=1}^{3}\sigma^2(f_i)} = 2.772e-3, \quad \frac{\sigma^2(f_{2,3})}{\sum_{i=1}^{3}\sigma^2(f_i)} = 4.494e-4.$$

We observe here that the interaction of x_1 and x_2 is the most important. If we set $\theta_2^1 = 0.1$ we obtain the approximation:

$$f(\boldsymbol{x}) \approx f_0 + f_1(x_1) + f_2(x_2) + f_3(x_3) + f_{1,2}(x_1, x_2).$$

Criterion 2: similar to the case of *criterion 1*, since $D_2 = 2$ we only need $f_{1,2}$ and if $D_2 = 3$ we compute

$$\frac{\mathbb{E}(f_{1,2})}{\sum_{i=0}^{3}\mathbb{E}(f_i)} = 1.089e-5, \quad \frac{\mathbb{E}(f_{1,3})}{\sum_{i=0}^{3}\mathbb{E}(f_i)} = 2.727e-6, \quad \frac{\mathbb{E}(f_{2,3})}{\sum_{i=0}^{3}\mathbb{E}(f_i)} = 4.393e-7.$$

We observe that again the interaction of x_1, x_2 is the most important. If we set $\theta_2^2 = 1e-5$ for

criterion 2, we obtain the same approximation as in *criterion 1*:

$$f(\boldsymbol{x}) \approx f_0 + f_1(x_1) + f_2(x_2) + f_3(x_3) + f_{1,2}(x_1, x_2).$$

For this approximation, the error of $\mathbb{E}(f)$ is $3.26e-6$ and the error of $\sigma^2(f)$ is $2.51e-4$.

Criterion 3: we set $D_2 = 3$ and compute

$$\frac{\gamma_1\gamma_2}{1+\sum_{i=1}^3 \gamma_i + \sum_{j\neq k}\gamma_j\gamma_k} = 1.009e-3,$$
$$\frac{\gamma_1\gamma_3}{1+\sum_{i=1}^3 \gamma_i + \sum_{j\neq k}\gamma_j\gamma_k} = 2.522e-4,$$
$$\frac{\gamma_2\gamma_3}{1+\sum_{i=1}^3 \gamma_i + \sum_{j\neq k}\gamma_j\gamma_k} = 4.035e-5.$$

Again, we observe the same importance order as in *criteria 1* and *2*.

Example 2. We present an example for a 1000-dimensional function to illustrate the proposed adaptive criteria using standard ANOVA and anchored ANOVA. Consider the Sobol function [134]

$$f(x) = \prod_{k=1}^{N} f^{(k)}(x_k), \tag{2.3.16}$$

where $N = 1000, f^{(k)}(x_k) = \frac{|4x_k-2|+a_k}{1+a_k}$ and $a_k = k^2$. Note that the coefficients a_k in this paper is different from those in [134] where $a_k = O(k)$. We consider three cases, where $a_k = O(1), O(k)$ and $O(k^2)$, in [169] and explain how different coefficient lead to different convergence rate with respect to the truncation dimension. See [169] for detail. Then the mean and variance of the function are readily computed: $\mathbb{E}(f) = 1$ and $\sigma^2(f) = 0.10395$. We test criteria 1 and 3 using standard ANOVA and criteria 1 and 2 using anchored ANOVA. For *anchored* ANOVA we use an anchor point $c = (c_1, c_2, ..., c_N)$ such that it satisfies $f^{(k)}(c_k) = \frac{\lambda_k^2+\tau_k^2}{\tau_k}$ where λ_k^2, τ_k is the variance and the mean of $f^{(k)}$, respectively [169]. We compute the error of the variance defined by $|\sigma^2(f) - \sum_S \sigma^2(f_S)|$ and estimate how many terms are selected among all second-order terms. Without adaptivity we

need 499,500 = $\binom{1000}{2}$ second-order terms.

Note that criterion 2 cannot be used for standard ANOVA since the mean of each ANOVA terms is always zero as in (2.1.4), hence no terms will be selected. Both Tables 2.2 and 2.3 show that just a few second-order terms out of $499,500$ terms are needed to achieve at least 10^{-3} accuracy for the variance error. In order to get better accuracy we would need smaller threshold and hence more terms. For standard ANOVA in Table 2.2 both criterion 1 and 3 give the same order of magnitude of error with almost the same number of second-order terms selected. This is also true for anchored ANOVA as shown in Table 2.3. However, the error for anchored ANOVA is larger than the one for standard ANOVA even if it needs more terms since the anchored ANOVA approximation to the function is worse than the standard ANOVA approximation. Tables 2.2 and 2.3 also show that $\theta_2^i, (i = 1, 2, 3)$ does not have a significant impact on the variance error for either the standard or the anchored ANOVA.

Table 2.2: Variance error and size of selected set $\mathcal{F}$ using *standard* ANOVA for the Sobol function. $p = 0.999$. $|\mathcal{F}|$ is the cardinality of $\mathcal{F}$.

Criterion 1	θ_2^1	$1e-7$	$1e-6$	$1e-5$	$1e-4$
	error	$2.0170e-05$	$2.05217-05$	$2.3990e-05$	$5.2835e-05$
	$\lvert\mathcal{F}\rvert$	34	25	16	8
Criterion 3	θ_2^3	$1e-7$	$1e-6$	$1e-5$	$1e-4$
	error	$4.6856e-05$	$1.2148-04$	$2.3682e-04$	$5.5905e-04$
	$\lvert\mathcal{F}\rvert$	15	6	3	1

Table 2.3: Variance error and size of selected set $\mathcal{F}$ using *anchored* ANOVA for the Sobol function. The anchor point is chosen as described in the text. $p = 0.9999$.

Criterion 1	θ_2^1	$1e-7$	$1e-6$	$1e-5$	$1e-4$
	error	$7.0763e-03$	$7.0751-03$	$7.0648e-03$	$7.0243e-03$
	$\lvert\mathcal{F}\rvert$	76	45	19	8
Criterion 2	θ_2^2	$1e-7$	$1e-6$	$1e-5$	$1e-4$
	error	$7.0747e-03$	$7.0636e-03$	$7.0118e-03$	$6.7630e-03$
	$\lvert\mathcal{F}\rvert$	42	18	7	2

2.4 Application to stochastic compressible flow

Stochastic modeling of fluid mechanics problems allows for a broader and more comprehensive understanding of the flow physics compared to deterministic formulations, and it can potentially lead to new approaches in designing thermo-fluidic equipment that operates robustly under uncertain conditions. While significant progress has been made to date in advancing stochastic CFD, most of the processes simulated so far have been modeled by low-dimensional representations of the inputs and outputs, often limiting our ability in probing the intriguing physics of the flow, e.g., the effect of random disturbances of large amplitude and/or small correlation length on flow stability and on momentum and heat transport. These effects require high-dimensional representations in the parametric or random space that often render such simulations computationally prohibitive. Towards this end, progress can be made if the computational complexity of the stochastic problem is reduced by some orders of magnitude, e.g. by estimating the so-called "effective dimensionality" of the flow system, as was done for financial systems using quasi-Monte Carlo theory [151, 133]. In this section we employ the functional ANOVA method [60, 14] and evaluate its effectiveness as a dimension-reduction technique compressible flow problems. This is motivated by the idea that for many flow systems, only relatively low order correlations between dimensions will significantly impact the solution.

We consider a flow problem, which have been well-studied with deterministic CFD: shock scattering by a wedge surface. We consider a random geometric boundary representing roughness. This prototype problem is inviscid compressible flow, specifically, supersonic flow past a rough wedge. For the smooth wedge, the shock path and pressure distribution can be obtained by simple analytical formulas. However, complex shock dynamics is observed when considering a random rough wedge surface. Lighthill [96] and Chu [32] used first-order perturbation analysis to study the case of weak interactions. The first-order theory is adequate only for very small roughness height and does not provide a measure of the *mean* extra force due to roughness, which is assumed to have

zero mean, and hence the first-order theory predicts zero mean forces. Lin et al. [97, 98] developed a second-order perturbation method to overcome the disadvantages of first-order method and developed semi-analytical formulas to describe the solution for small roughness height. Figure 2.1 (from [97]) shows pressure contours of one realization for two different Mach numbers; here we will only focus on the high Mach number case. Similarly to the incompressible flow example, here too we will compare the ANOVA based solution to corresponding solutions obtained by the sparse grid and the quasi-Monte Carlo methods.

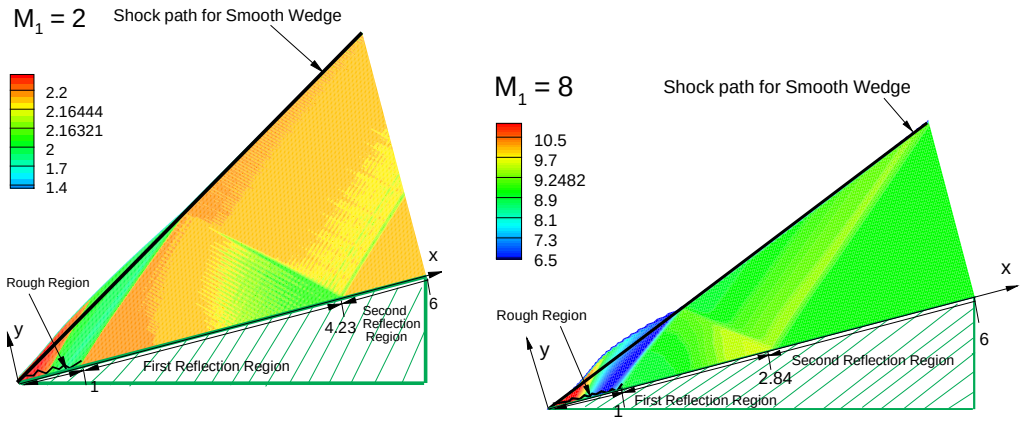


Figure 2.1: Pressure contours for $M_1 = 2$ (left) and $M_1 = 8$ (right), where M_1 is the Mach number of the uniform inflow from left. Roughness height $\varepsilon = 0.01$, and correlation length of roughness is $A = 0.1$ but we also consider the case of $A = 0.01$. The pressure contours for $M_1 = 8$ are stretched 4 times perpendicular to the wedge surface for visualization purposes.

2.4.1 Stochastic perturbation analysis

In this section we briefly describe the first- and second-order perturbation methods to obtain analytical solutions for small roughness height; details can be found in [98]. We assume that: (1) The random wedge roughness is small, and correspondingly the perturbation of the shock slope is small. (2) The oblique shock is attached to the wedge. (3) The flow between the shock and the wedge is adiabatic.

The isentropic flow between the wedge and the shock is governed by the Euler equations along

with the isentropic condition (2D case):

$$\begin{cases} u\dfrac{\partial p}{\partial x}+\upsilon\dfrac{\partial p}{\partial y}+\rho c^2\left(\dfrac{\partial u}{\partial x}+\dfrac{\partial \upsilon}{\partial y}\right)=0, \\ u\dfrac{\partial u}{\partial x}+\upsilon\dfrac{\partial u}{\partial y}+\dfrac{1}{\rho}\dfrac{\partial p}{\partial x}, \\ u\dfrac{\partial \upsilon}{\partial x}+\upsilon\dfrac{\partial \upsilon}{\partial y}+\dfrac{1}{\rho}\dfrac{\partial p}{\partial y}, \\ u\dfrac{\partial s}{\partial x}+\upsilon\dfrac{\partial s}{\partial y}=0, \end{cases} \tag{2.4.1}$$

where the first equation of Eq. (2.4.1) is based on the fact that $c^2=\frac{\partial p}{\partial \rho}$. On the wedge surface ($y=\varepsilon h(x;\omega)$), the slip boundary condition ($\frac{\upsilon_w}{u_w}=\varepsilon\frac{\partial h}{\partial x}$) is employed, where υ_w and u_w are the velocity perpendicular and parallel to the wedge. According to the Rankine-Hugoniot relations, we have:

$$\begin{aligned} &\frac{P_2}{P_1}=1+\frac{2\gamma}{1+\gamma}(M_1^2\sin^2\chi-1), \qquad \frac{\rho_1}{\rho_2}=\frac{u_2}{u_1}=\frac{(\gamma-1)M_1^2\sin^2\chi+2}{(\gamma+1)M_1^2\sin^2\chi}, \\ &\tan(\chi-\theta)=\tan\chi\frac{(\gamma-1)M_1^2\sin^2\chi+2}{(\gamma+1)M_1^2\sin^2\chi}\equiv T(M_1,\chi). \end{aligned} \tag{2.4.2}$$

First-order theory

The domain we consider is between the perturbed shock and the wedge surface. We use the subscript '2' for the flow state after the shock and take the x-axis along the surface of the 'unperturbed' wedge. Let

$$\frac{u}{W_2}=1+\varepsilon w', \quad \frac{\upsilon}{W_2}=\varepsilon\upsilon, \quad \frac{p}{P_2}=1+\varepsilon p', \quad \frac{\rho}{\rho_2}=1+\varepsilon\rho', \quad \frac{s}{S_2}=1+\varepsilon s'. \tag{2.4.3}$$

On the wedge surface, we have $\upsilon_w'=\frac{\partial h}{\partial x}$ and using the Rankine-Hugoniot relations (Eq. (2.4.2)), we obtain the interface conditions after the shock:

$$\upsilon_s'=\theta'=F(M_1,x_0)\chi', \quad \frac{\sqrt{M_2^2-1}}{\gamma M_2}p_s'=G(M_1,\chi_0)\chi', \tag{2.4.4}$$

where

$$
\begin{aligned}
&F(M_1,\chi_0) \equiv \frac{\mathrm{d}\theta}{\mathrm{d}\chi} = 1 - \frac{1}{1+T_0^2}\frac{\partial T}{\partial \chi},\\
&G(M_1,\chi_0) \equiv \frac{\sqrt{M_2^2-1}}{\gamma M_2^2 P_2}\frac{\mathrm{d}P_2}{\mathrm{d}\chi} = \frac{\sqrt{M_2^2-1}}{\gamma M_2^2}\frac{2\gamma M_1^2 \sin 2\chi_0}{1-\gamma+2\gamma M_1^2 \sin^2\chi_0},\\
&T = \tan(\chi-\theta), \quad T_0 = \tan(\chi_0-\theta_0), M_1 = \frac{W_1}{c_1}, \quad M_2^2 = M_1^2\frac{P_1\rho_2\cos^2\chi_0}{P_2\rho_1\cos^2(\chi_0-\theta_0)}.
\end{aligned}
\tag{2.4.5}
$$

To quantify the region of validity for the stochastic perturbation analysis, we use the standard deviation of the stochastic roughness as a measure. To simplify the problem with small perturbation, we have employed the following assumption:

$$
\varepsilon \ll \min\left(\frac{1}{\sigma(w')}, \frac{1}{\sigma(v')}, \frac{1}{\sigma(\rho')}, \frac{1}{\sigma(p')}\right) \tag{2.4.6}
$$

where σ denotes the standard deviation. The region of validity with respect to the roughness amplitude ε for the stochastic perturbation analysis is discussed in Appendix C of [98]. Substituting Eq. (2.4.6) into the steady Euler equations (2.4.1) we obtain the linearized small perturbation equations:

$$
\begin{cases}
\dfrac{\partial v'}{\partial y} + \dfrac{M_2^2-1}{\gamma M_2^2}\dfrac{\partial p'}{\partial x} = 0,\\
\dfrac{\partial}{\partial x}\left(w' + \dfrac{1}{\gamma M_2^2}p'\right) = 0,\\
\dfrac{\partial v'}{\partial x} + \dfrac{1}{\gamma M_2^2}\dfrac{\partial p'}{\partial y} = 0,\\
\dfrac{\partial}{\partial x}(p' - \gamma\rho') = 0.
\end{cases}
\tag{2.4.7}
$$

The results for the first-order method are:

$$
\chi'(x;\omega) = q\sum_{n=0}^{\infty}(-r)^n \frac{\partial h}{\partial x}(x;\omega)\Big|_{x=\alpha\beta^n x}, \tag{2.4.8}
$$

where

$$
q = \frac{2}{F+G}, \quad r = \frac{F-G}{F+G}, \quad \alpha = 1-\frac{T_0}{m}, \quad \beta = \frac{m-T_0}{m+T_0}, \quad m = \frac{1}{\sqrt{M_2^2-1}}.
$$

The perturbed pressure p'_w on the rough wedge surface is:

$$p'_w(x;\omega) = \frac{\gamma M_2^2}{\sqrt{M_2^2-1}}\left(\frac{\partial h(x;\omega)}{\partial x} + 2\sum_{n=1}^{\infty}(-r)^n \frac{\partial h}{\partial x}(x;\omega)\Big|_{x=\beta^n x}\right). \tag{2.4.9}$$

The perturbed shock path z' is computed by:

$$z'(x;\omega) = (1+T_0^2)\int_0^x \chi'(x_1;\omega)\mathrm{d}x_1. \tag{2.4.10}$$

Furthermore, the perturbed p', υ', ρ' and w' at any location (x, y) after the shock can also be obtained (see [98]).

Second-order theory

Using an iterative procedure we start with $z'(x;\omega)$ (Eq.(2.4.10)) and $p'_w(x;\omega)$ (Eq.(2.4.9)) and upon convergence we obtain the corrected perturbed shock path $\widetilde{z'}(x;\omega)$ and corresponding corrected pressure $\widetilde{p'_w}(x;\omega)$. To this end, we have to re-define β in Eq. (2.4.9), i.e., $\beta = x_m/x$, where x_m, x are the distances from the apex of a reflection pair of points on the wedge surface ($x_m < x$). Let

$$\frac{u}{W_2} = 1+\Delta w, \quad \frac{\upsilon}{W_2} = \Delta\upsilon, \quad \frac{p}{P_2} = 1+\Delta p, \quad \frac{\rho}{\rho_2} = 1+\Delta\rho, \quad \frac{s}{S_2} = 1+\Delta s,$$

where $\Delta w = \varepsilon\widetilde{w'} + \varepsilon^2 w''$, etc., are the total first- and second-order corrections. The second-order small perturbation equations are:

$$\begin{cases} \dfrac{\partial \upsilon''}{\partial y} + \dfrac{M_2^2-1}{\gamma M_2^2}\dfrac{\partial p''}{\partial x} = R_1(x,y), \\ \dfrac{\partial}{\partial x}\left(w'' + \dfrac{1}{\gamma M_2^2}p''\right) = R_2(x,y), \\ \dfrac{\partial \upsilon''}{\partial x} + \dfrac{1}{\gamma M_2^2}\dfrac{\partial p''}{\partial y} = R_3(x,y), \\ \dfrac{\partial}{\partial x}\left(p'' - \gamma\rho''\right) = R_2(x,y), \end{cases} \tag{2.4.11}$$

where

$$
\begin{aligned}
R_1(x,y) &= -\frac{1}{\gamma M_2^2}(-M_2^2\widetilde{p'} + \widetilde{\rho'} + (M_2^2+1)\widetilde{w'})\frac{\partial \widetilde{w'}}{\partial x} + \upsilon'\left(\frac{\partial \widetilde{w'}}{\partial y} - \frac{1}{\gamma}\frac{\partial \widetilde{p'}}{\partial y}\right),\\
R_2(x,y) &= \frac{(\widetilde{w'} + \widetilde{\rho'})}{\gamma M_2^2}\frac{\partial \widetilde{p'}}{\partial x} - \upsilon'\frac{\partial \widetilde{w'}}{\partial y},\\
R_3(x,y) &= \frac{1}{\gamma M_2^2}\left((\widetilde{w'} + \widetilde{\rho'})\frac{\partial \widetilde{p'}}{\partial y} + (M_2^2-1)\upsilon'\frac{\partial \widetilde{p'}}{\partial x}\right),\\
R_4(x,y) &= \frac{1}{2}\frac{\partial}{\partial x}(\widetilde{p'}^2 - \gamma\widetilde{\rho'}^2) - \upsilon'\frac{\partial}{\partial y}(\widetilde{p'} - \gamma\widetilde{\rho'}).
\end{aligned}
$$

We note that the $\widetilde{\ }$ denotes a converged first-order state that is corrected due to the shock path update. According to the Rankine-Hugoniot relations, we obtain p_s'' and υ_s'' on the shock path based on the first-order corrected shock path:

$$
\upsilon_s'' = F(M_1,\chi_0)\chi'' + F_2(M_1,\chi_0)\widetilde{\chi'}^2, \qquad \frac{\sqrt{M_2^2-1}}{\gamma M_2^2}p_s'' = G(M_1,\chi_0)\chi'' + G_2(M_1,\chi_0)\widetilde{\chi'}^2,
$$

where

$$
\begin{aligned}
G_2(M_1,\chi_0) &= \frac{2M_1^2\sqrt{M_2^2-1}\cos^2\chi_0}{M_2^2(1-\gamma+2\gamma M_1^2\sin^2\chi_0)},\\
F_2(M_1,\chi_0) &= -\frac{1}{2(1+T_0^2)}\frac{\partial^2 T}{\partial \chi^2} + T_0(1-F(M_1,\chi_0))^2 + F(M_1,\chi_0)Q(M_1,\chi_0),\\
Q(M_1,\chi_0) &= \frac{2\cos(\chi_0-\theta_0)(\sin\theta_0 - M_1^2\sin^2\chi_0\sin(2\chi_0-\theta_0))}{(\gamma+1)M_1^2\cos\chi_0\sin^2\chi_0}.
\end{aligned}
$$

It can be shown ([98]) that

$$
\begin{aligned}
\chi''(x;\omega) = \sum_{n=0}^{\infty}(-r)^n\Bigg[&- r_2\widetilde{\chi'}^2(\beta^n x;\omega) - r_3\widetilde{\chi'}^2(\beta^{n+1}x;\omega) + 2r_4\upsilon_w''(\alpha\beta^n x;\omega)\\
&+ r_4\left(\int_{\alpha\beta^n x}^{\beta^n x} R_+(x_1,mx_1)\mathrm{d}x_1 - \int_{\beta^{n+1}x}^{\alpha\beta^n x} R_-(x_1,-mx_1)\mathrm{d}x_1\right)\Bigg],
\end{aligned}
\tag{2.4.12}
$$

where

$$\upsilon''_w(x;\omega) = \widetilde{w'_w}(x;\omega)\frac{\partial h(x;\omega)}{\partial x}, \quad r_2 = \frac{F_2+G_2}{F+G}, \quad r_3 = \frac{F_2-G_2}{F+G}, \quad r_4 = \frac{1}{F+G},$$
$$R_\pm(x_1,y_1) = R_3(x_1,y_1) \pm \frac{1}{\sqrt{M_2^2-1}}R_1(x_1,y_1).$$

The second-order perturbation equations for pressure is

$$\begin{aligned} p''_w(x;\omega) =& \frac{\gamma M_2^2}{\sqrt{M^2-1}}\sum_{n=0}^{\infty}(-r)^n\left[2\frac{FG_2-F_2G}{F+G}\widetilde{\chi'}^2\left(\frac{\beta^{n+1}}{\alpha}x;\omega\right)+\upsilon''_w(\beta^n;\omega)\right. \\ & \left. - r\upsilon''_w(\beta^{n+1}x;\omega) - r\int_{\beta^{n+1}x}^{\frac{\beta^{n+1}}{\alpha}x} R_+(x_1,mx_1)\mathrm{d}x_1 - \int_{\frac{\beta^{n+1}}{\alpha}x}^{\beta^n x} R_-(x_1,-mx_1)\mathrm{d}x_1\right]. \end{aligned} \tag{2.4.13}$$

while the shock path is

$$z(x;\omega) = \varepsilon\widetilde{z'} + \varepsilon^2 z'' = \varepsilon(1+T_0^2)\int_0^x \widetilde{\chi'} + \varepsilon(\chi'' + T_0\widetilde{\chi'}^2\mathrm{d}x_1), \tag{2.4.14}$$

where $\widetilde{z'}$ and z'' are defined implicitly here. Hence the perturbed pressure on the wedge surface is given by

$$\Delta p_w(x;\omega) = \varepsilon\widetilde{p'} + \varepsilon^2 p''_w. \tag{2.4.15}$$

We note that $\Delta p_w, \widetilde{p'}_w, p''_w$ are normalized by P_2, which is the pressure of the base flow after the shock.

Based on these expressions for pressure, we can determine the perturbed lift and drag force on the wedge (non-dimensionalized by $P_2 d$):

$$\begin{aligned} \Delta L(x;\omega) &= \varepsilon\left[h\sin\theta_0 - \cos\theta_0\int_0^x \widetilde{p'}_w\mathrm{d}x_1 + \varepsilon\int_0^x\left(\widetilde{p'}_w\frac{\partial h}{\partial x_1}\sin\theta_0 - p''_w\cos\theta_0\right)\mathrm{d}x_1\right] \\ \Delta D(x;\omega) &= \varepsilon\left[h\cos\theta_0 + \sin\theta_0\int_0^x \widetilde{p'}_w\mathrm{d}x_1 + \varepsilon\int_0^x\left(\widetilde{p'}_w\frac{\partial h}{\partial x_1}\cos\theta_0 + p''_w\sin\theta_0\right)\mathrm{d}x_1\right] \end{aligned} \tag{2.4.16}$$

2.4.2 Scattering of Shock Waves

In this section we consider inviscid flow dynamics and focus on the scattering of a strong shock wave due to random roughness on the surface of a half-wedge (see Figures 2 and 9). Of physical interest here is the effect of roughness on the induced forces on the wedge surface while of numerical interest is the exact form of adaptive ANOVA representation required for different levels of accuracy. The scattering of shock wave is quite different from that of the classical oblique shock problem due to the presence of random roughness.

Mathematical modeling of random roughness

The roughness length is denoted by d, and we normalize all lengths by d. The roughness starts from the apex of the wedge and the end part of the wedge is smooth. We denote the half wedge angle by θ_0, the unperturbed shock angle for smooth wedge by χ_0, the shock angle for rough wedge at location x by $\chi(x;\omega)$ and the angle between υ_s and u_s by $\theta(x;\omega)$ satisfying $\tan\theta = \frac{\upsilon_s}{u_s}$, where υ_s and u_s are the velocity right after the shock perpendicular and parallel to the wedge, respectively. For a smooth wedge, χ and θ degenerate to χ_0 and θ_0. We denote the incoming flow velocity by W_1 with its normal component $u_1 = W_1 \sin\chi$ and we also denote the velocity there by W_2 and its normal component to the shock by $u_2 = W_2 \sin(\chi-\theta) = W_1 \cos\chi \tan(\chi-\theta)$. Here, u_1, u_2 are the normal components of the velocities going into and coming out of the shock. Figure 2.2 shows a sketch of a typical perturbed shock path induced by the randomly rough boundary.

The random roughness is modeled based on a non-dimensional random process $h_m(x;\omega)$ expressed by the KL decomposition:

$$h_m(x;\omega) = \bar{h}_m(x) + \sum_{i=1}^{\infty} \sqrt{\lambda_i}\psi_i(x)\xi_i(\omega), \tag{2.4.17}$$

where $\bar{h}_m(x)$ is the mean, $\{\xi_i(\omega)\}$ is a set of uncorrelated random variables with zero mean and unit variance, and ψ_i and λ_i are the eigenvectors and corresponding eigenvalues of the covariance

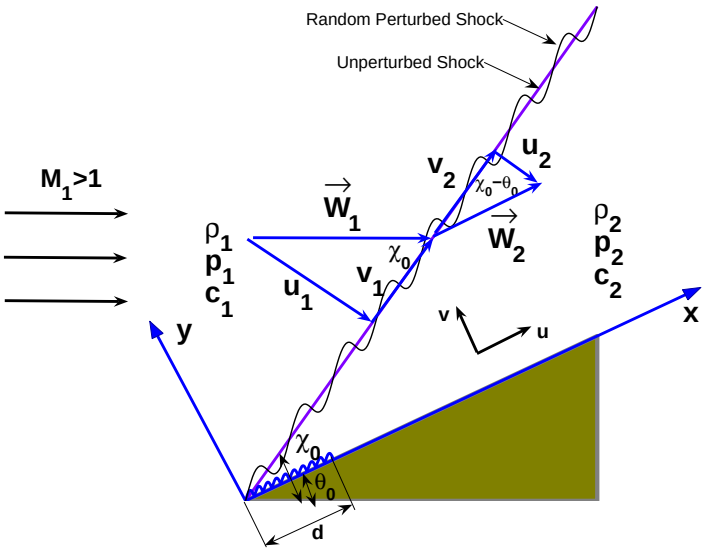


Figure 2.2: Sketch of supersonic flow past a wedge with rough surface: Definition of coordinate system and notation; shown is also a perturbed shock path and the location of the unperturbed shock corresponding to a smooth wedge surface.

kernel $R_{hh}(x_1, x_2)$:

$$\int R_{hh}(x_1, x_2)\psi_i(x_2)dx_2 = \lambda_i\psi_i(x_1). \tag{2.4.18}$$

We assume that $\bar{h}_m(x) = 0$, and the roughness height is

$$y(x;\omega) = \varepsilon h(x;\omega) = \varepsilon \frac{h_m}{\max_x(\sigma(h_m))}. \tag{2.4.19}$$

The covariance kernel $R_{hh}(h_m(x_1;\omega), h_m(x_2;\omega))$ is obtained based on the solution of [98]

$$\frac{\mathrm{d}^4 h_m}{\mathrm{d}x^4} + k^4 h_m = f(x),$$

where x is normalized by the roughness length d, $k = d/A$, A is the correlation length, and the random forcing term $f(x)$ is white noise, i.e., $\mathbb{E}\{f(x_1)f(x_2)\} = \delta(x_1 - x_2)$. Without losing generality, we set $d = 1$. The roughness starts from the apex of wedge and the required boundary conditions

for this case are: $h_m(0;\omega) = h'_m(0;\omega) = h_m(1;\omega) = h'_m(1;\omega) = 0$. The eigenvectors are obtained as the solution of the homogeneous equation

$$\frac{\mathrm{d}^4\psi}{\mathrm{d}x^4} - \Lambda^4\psi = 0$$

with boundary condition $\psi(0) = \psi(1) = \psi'(0) = \psi'(1) = 0$ and Λ is the eigenvalue of operator $\mathrm{d}^4/\mathrm{d}x^4$. Such boundary conditions are chosen due to the assumption for second-order perturbation analysis, which states the random roughness and other perturbed quantities are small *and* smooth in the computational domain. The stochastic process $h_m(x;\omega)$ can then be represented by the KL expansion:

$$h_m(x;\omega) = \sum_{n=1}^{\infty} \frac{1}{\Lambda_n^4 + k^4}\psi_n(x)\xi_n(\omega), \tag{2.4.20}$$

where

$$\psi_n(x) = \cos\Lambda_n x - \cosh\Lambda_n x - \frac{\cos\Lambda_n - \cosh\Lambda_n}{\sin\Lambda_n - \sinh\Lambda_n}(\sin\Lambda_n x - \sinh\Lambda_n x).$$

Here, Λ_n is obtained by solving $\cos\Lambda_n \cosh\Lambda_n = 1$, where $\{\xi_n(\omega)\}$ is a set of uncorrelated random variables with zero mean and unit variance. In practice, the KL expansion is truncated according to the following criterion:

$$\sum_{n=1}^{N} \frac{\Lambda_n}{\Lambda_n^4 + k^4} \geq \alpha \sum_{n=1}^{\infty} \frac{\Lambda_n}{\Lambda_n^4 + k^4}, \tag{2.4.21}$$

where $\alpha = 0.9; 0.95; 0.99$. The number of dimensions for different values of the correlation length and for different values of α are listed in Table 2.4.

Table 2.4: Number of dimensions for different values of correlation length at various truncation levels.

α	$A = 1$	$A = 0.1$	$A = 0.01$	$A = 0.001$
90%	2	7	79	798
95%	3	10	112	1133
99%	10	25	253	2537

Computational results

We consider here the case of Mach number 8 and we follow the set up in [98]; for $A = 0.1$, the nominal dimension $N = 12$ (capturing more than 95% of energy in (2.4.21)), $\varepsilon = 0.003$, $\theta_0 = 14.7436\,^\circ$, $\chi_0 = 20.5755\,^\circ$, and ξ_n used in KL expansion are *uniformly* distributed on $[-\sqrt{3}, \sqrt{3}]$. The contours of pressure for one realization are shown in Figure 2.1. We set $\mu = 3$ in the ANOVA decomposition and let the anchor point $\boldsymbol{c} = 0$. This is because from the results in [97] and [98] we can see that the means for extra lift and drag are very close to 0 and according to the theory in [168], the mean is a good choice (maybe optimal) for anchor point; other choices of anchor points are discussed in [168].

The standard deviation based on the first-order terms f_j for the extra lift and drag are shown in Figure 2.3. We use these values in step 3 of the algorithm to determine the necessary second-order terms based on criterion 1. Similarly, for criterion 2, we can compute the mean of the first-order terms to be used in step 3; the pattern of the means (not shown in this paper) is similar to the pattern of standard deviations. Both the means and standard deviations are monotonically decreasing when $j \geq 2$, hence we can use (2.3.7) and (2.3.10) to adaptively select terms. We have also tested criterion 3 but it does not work well for this specific problem since the error is about three times than using criterion 1 or 2, so we do not report any corresponding results. The important dimensions captured by criterion 3 is quite different from those captured by criterion 1 or 2. The main reason is that the mean of each component function, i.e., the denominator in (2.3.12), is very close to zero, which makes criterion 3 not reliable for this case.

Table 2.5: Threshold p and corresponding active dimension D_2 for different criteria.

Criterion 1	p	0.997	0.998	0.999
	D_2	5	6	7
Criterion 2	p	0.97	0.98	0.99
	D_2	5	6	7

Several choices of threshold p and corresponding D_2 for criteria 1 and 2 are listed in Table 2.5.

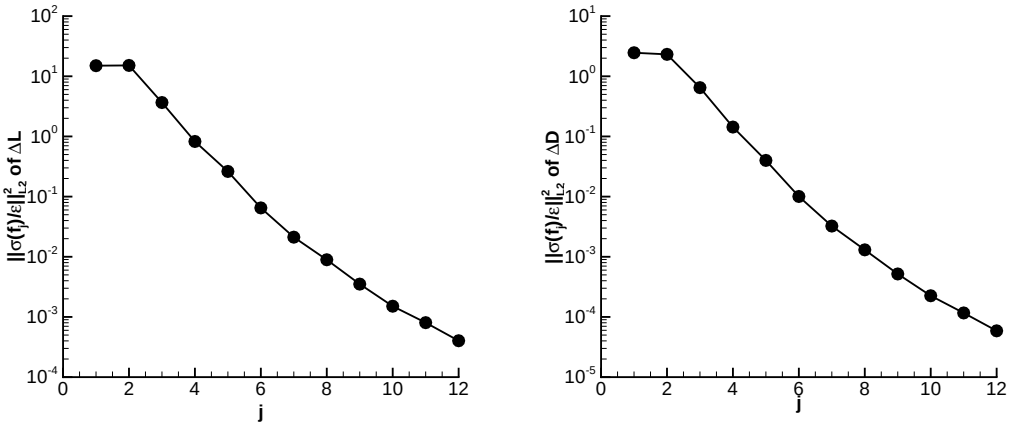


Figure 2.3: Standard deviation of extra lift (left) and drag (right) for first-order terms f_j of ANOVA decomposition. $A = 0.1, N = 12, \mu = 3$. Only the first-order terms are needed when selecting the second-order terms. This is the third step in Algorithm 1.

We set $D_2 = 6$ in order to compare with the sparse grid method and demonstrate the convergence of the adaptive method. Our tests show that $\theta_2^1 = 5.0 \times 10^{-5}$ (i.e., 0.005%) is a good threshold for criterion 1 and $\theta_2^2 = 1.0 \times 10^{-3}$ (i.e., 0.1%) is a good threshold for criterion 2. With these thresholds, the estimates of mean and standard deviation of the extra lift and drag are more accurate than for other values. Both criteria lead to the same selection of second-order terms, and hence a good approximation of $f(\boldsymbol{x})$ turns out to be

$$f(\boldsymbol{x}) \approx f_0 + \sum_{j=1}^{12} f_j(x_j) + \sum_{j=2}^{5} f_{1,j}(x_1, x_j) + \sum_{j=3}^{6} f_{2,j}(x_2, x_j) + \sum_{j=4}^{5} f_{3,j}(x_3, x_j), \tag{2.4.22}$$

where $\boldsymbol{x}$ refers to the random point, and $f(\boldsymbol{x})$ denotes either the extra lift $\Delta L(\boldsymbol{x})$ or extra drag $\Delta D(\boldsymbol{x})$. The number of collocation points used in the different simulations are shown in Tables 2.6 and 2.7 for the sparse grid method and adaptive ANOVA method, respectively. We can see that even though the nominal dimension is not large, the number of collocation points required by the sparse grid method increases very quickly as the level increases. For the adaptive ANOVA method, since the active dimension D_2 and the number of collocation points μ are small, the increase of the sampling points is much slower. For comparison, we note that if we use the standard ANOVA

method with $\mu = 3, \nu = 2$, i.e., we use all the second-order terms, the number of points is 631.

Table 2.6: Number of collocation points for Smolyak sparse grid method: $A/d = 0.1$ and nominal dimension $N = 12$.

Level 1	Level 2	Level 3
25	313	2649

Table 2.7: Number of collocation points for adaptive ANOVA method : $A/d = 0.1$, nominal dimension $N = 12$, active dimension $D_1 = N = 12, D_2 = 6$, $\mu = 3$, $\theta_2^1 = 5.0 \times 10^{-5}$ for criterion 1 or $\theta_2^2 = 1.0 \times 10^{-3}$ for criterion 2.

f_j	$f_j + f_{1,j}$	$f_j + f_{1,j} + f_{2,j}$	$f_j + f_{1,j} + f_{2,j} + f_{3,j}$
37	73	109	127

The computational details are summarized below:

Algorithm 2 Summary of computing shock problem with dimension $N = 12$

1: Select the anchor point $\boldsymbol{c} = 0$ and run the deterministic solver at this sampling point to obtain the constant term f_0 in the ANOVA decomposition (according to equation (2.2.1)). In this case f is the extra pressure Δp.
2: For this case, set $\mu = 3$ and quadrature points $q_i^{1,2,3} = -1, 0, 1, i = 1, 2, \cdots, N$ since ξ_i are uniform random variables on $[-\sqrt{3}, \sqrt{3}]$. then the corresponding weights are $w^{1,2,3} = 5/9, 8/9, 5/9$. Run the deterministic solver at the sampling points $(q_1^{1,2,3}, 0, \cdots, 0), (0, q_2^{1,2,3}, 0, \cdots, 0), \cdots, (0, \cdots, 0, q_N^{1,2,3})$ to obtain the value of all first-order terms at quadrature points $q_i^{1,2,3}, i = 1, 2, \cdots, N$ with equation (2.3.2).
3: Compute the necessary statistical values of mean and standard deviation required in the adaptivity criterion based on quadrature value of first order terms f_i and corresponding weights to determine the active dimension D_2 for selecting the important second-order terms.
4: Use tensor product rule to obtain new sampling points: $(q_1^1, q_2^1, 0, \cdots, 0), (q_1^1, q_2^2, 0, \cdots, 0), (q_1^1, q_2^3, 0, \cdots, 0), (q_1^2, q_2^1, 0, \cdots, 0), \cdots, (0, \cdots, 0, q_{N-1}^3, q_N^3)$ and run the deterministic solver at these the sampling points then compute the value of second-order terms at quadrature points with equation (2.3.3).
5: Use either Criterion 1 or Criterion 2 to decide the important second order terms. Then finally we get the ANOVA decomposition of f with $\nu = 2$.

Figure 2.4 compares the standard deviation of the extra lift and drag obtained by the sparse grid method and the adaptive ANOVA method with the quasi-Monte Carlo as the reference solution (50,000 sampling points). We can see that the results by level 1 sparse grid method are the least accurate, especially for the extra drag. The results by the level 2 sparse grid method are better than those of level 1 sparse grid, especially for the estimate in the first reflection region ([97]), i.e., $x \in [1, 2.84]$. In the second reflection region, i.e., $x \in [2.84, 6]$ the estimates of both level 1 and

level 2 sparse grid method are very close. The results of level 3 sparse grid method are better than those of the former two lower level sparse grid method. This can be observed more distinctly in the second reflection region in the plot of Figure 2.4 for the standard deviation of the extra lift. The results by the adaptive ANOVA method are very close to the results by the level 3 sparse grid and, in fact, if we zoom in the plot of the extra lift we can see that the former one is slightly better than the latter. For the estimate of the standard deviation of the extra drag, it is hard to discern differences between level 2, level 3 sparse grid, and the adaptive ANOVA method. We also study the error more systematically and show the relative error of the standard deviation in Figure 2.5, where "Lvl" means level, $f_j = f_0 + \sum_{j=1}^{12} f_j(x_j)$, $f_{1,j} = \sum_{j=2}^{5} f_{1,j}(x_1, x_j)$, $f_{2,j} = \sum_{j=3}^{6} f_{2,j}(x_2, x_j)$, $f_{3,j} = \sum_{j=4}^{5} f_{3,j}(x_3, x_j)$, i.e., we decompose Eq. (2.4.22) into four parts and add them one by one to see the difference. We can see that the adaptive ANOVA method with terms $f_j + f_{1,j} + f_{2,j} + f_{3,j}$ leads to a better estimate of the standard deviation than the level 3 sparse grid. The x-axis in Figure 2.5 is the number of collocation points (or sampling points) and since the deterministic solver with different collocation points costs almost the same, the computational cost of adaptive ANOVA is less than 1/20 of level 3 sparse grid method (see the exact number of collocation points in Table 2.6 and 2.7). Similarly, Figure 2.6 compares the mean of the extra lift and drag obtained by different methods. Again, we observe that the results by the level 1 sparse grid method are worse than those of others. However, the difference between the estimate by level 2, level 3 sparse grid method, and the adaptive ANOVA method is too small to be seen on the plot so we only plot the results of adaptive ANOVA method. A systematic study of the error of the mean is presented in Figure 2.7, which shows that the estimates by the adaptive ANOVA with terms $f_j + f_{1,j} + f_{2,j} + f_{3,j}$ are almost as accurate as those by level 3 sparse grid method. Moreover, Figures 2.5 and 2.7 also show the convergence of the adaptive ANOVA method as we add the terms in Eq. (2.4.22) one by one. Figures 2.8 and 2.9 show the convergence of the adaptive ANOVA method visually for the standard deviation and mean, respectively. We can see that for this case, the second-order terms play an important role in the approximation, otherwise, the results by using only f_0 and the

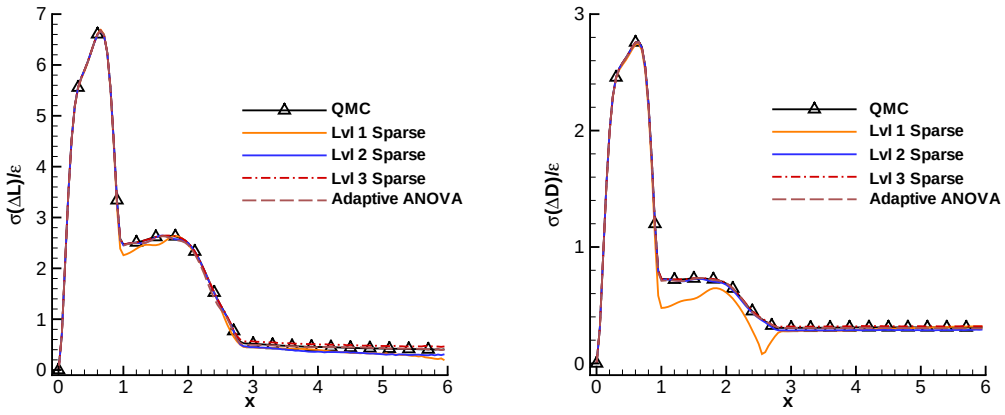


Figure 2.4: Comparison of standard deviation of extra lift (left) and drag (right) for different level of sparse grid method and adaptive ANOVA method. $A = 0.1, D_1 = N = 12, D_2 = 6, \mu = 3$.

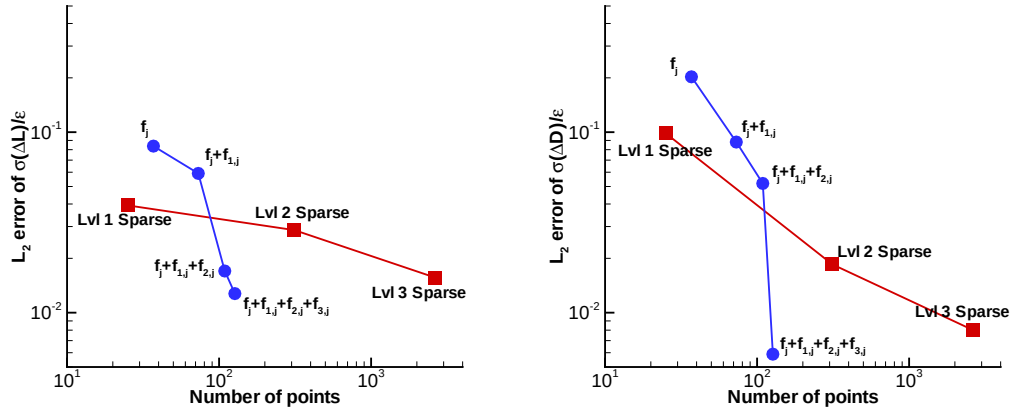


Figure 2.5: Error of standard deviation of extra lift (left) and drag (right) for different level of sparse grid method and adaptive ANOVA method. $A = 0.1, D_1 = N = 12, D_2 = 6, \mu = 3$.

first-order terms f_j are even worse than the level 1 sparse grid method.

Next, we compare the difference of the results by using different active dimensions listed in Table 2.5 (the reference solution is obtained by quasi-Monte Carlo with 50,000 realizations). Our tests show that for criterion 1, $\theta_2^1 = 5.0 \times 10^{-5}$ is a good choice for active dimension $D_2 = 6, 7$ since it leads to small error. With this θ_2^1, when we set the active dimension to be either 6 or 7, we use the same second-order component functions, i.e., f is approximated as in (2.4.22). For $D_2 = 5$,

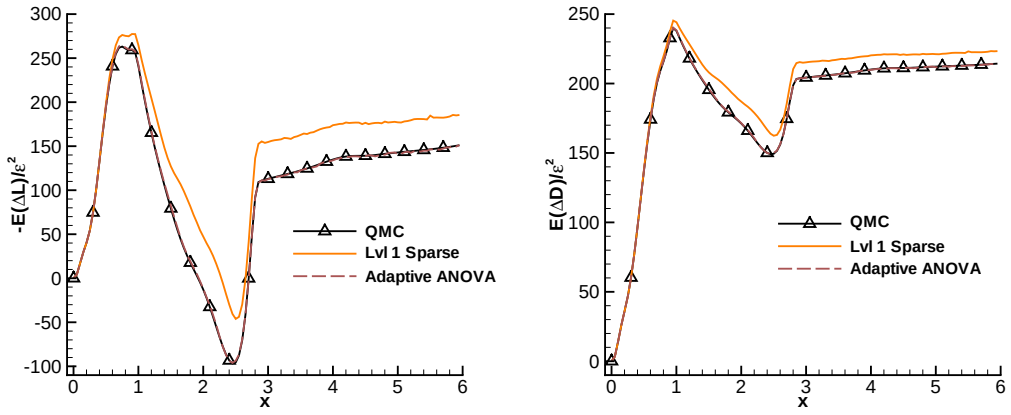


Figure 2.6: Comparison of mean of extra lift (left) and drag (right) for different level of sparse grid method and adaptive ANOVA method. $A = 0.1, D_1 = N = 12, D_2 = 6, \mu = 3$. The curve corresponding to adaptive ANOVA coincides with the QMC curve.

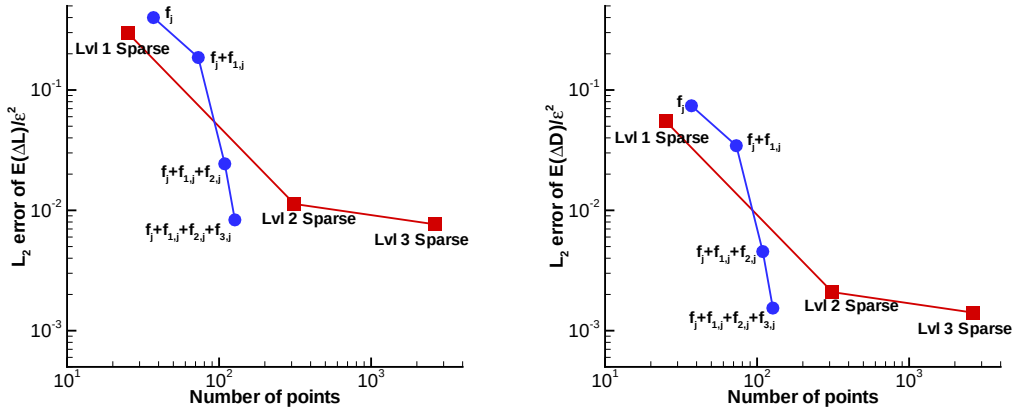


Figure 2.7: Error of mean of extra lift (left) and drag (right) for different level of sparse grid method and adaptive ANOVA method. $A = 0.1, D_1 = N = 12, D_2 = 6, \mu = 3$.

$\theta_2^1 = 2.0 \times 10^{-5}$ is one of the optimal choices and f is approximated well by

$$f(\boldsymbol{x}) \approx f_0 + \sum_{j=1}^{12} f_j(x_j) + \sum_{j=2}^{5} f_{1,j}(x_1, x_j) + \sum_{j=3}^{5} f_{2,j}(x_2, x_j) + \sum_{j=4}^{5} f_{3,j}(x_3, x_j) + f_{4,5}(x_4, x_5). \quad (2.4.23)$$

Similarly, for criterion 2, we set $\theta_2^2 = 1.0 \times 10^{-3}$ for $D_2 = 6, 7$ and $\theta_2^2 = 5.0 \times 10^{-4}$ for $D_2 = 5$.

With these settings, f is also approximated as in (2.4.22) for $D_2 = 6, 7$ and (2.4.23) for $D_2 = 5$.

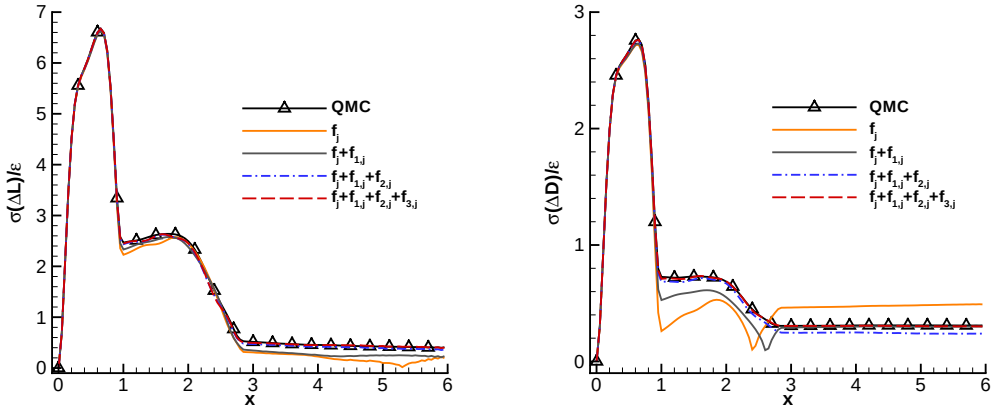


Figure 2.8: Convergence of standard deviation of extra lift (left) and drag (right) for adaptive ANOVA method. $A = 0.1, D_1 = N = 12, D_2 = 6, \mu = 3$.

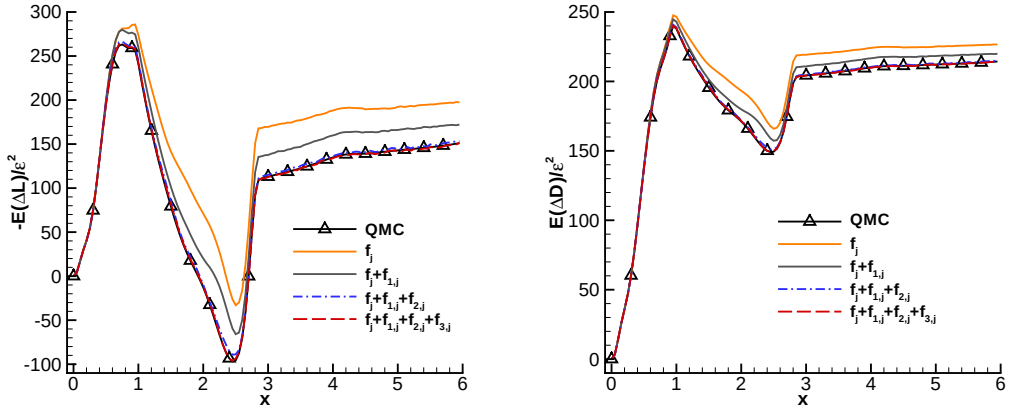


Figure 2.9: Convergence of mean of extra lift (left) and drag (right) for adaptive ANOVA method. $A = 0.1, D_1 = N = 12, D_2 = 6, \mu = 3$.

Here, we compare the results of $D_2 = 5$ and 6. The number of collocation points we use is the same, i.e., 127. The relative L_2 errors of the standard deviation and mean for the extra lift and drag are shown in Table 2.8. We can see that the selection of p (or $\theta_1^{1,2}$), which determines the active dimension and affects the selection of threshold $\theta_2^{1,2}$, will affect the accuracy.

We also employ a fifth-order WENO scheme [75] combined with a mapping technique for solving partial differential equations on random domains [158] to compute the numerical solution as in [98]. We use 900×300 grid points on the domain $[0, 6] \times [0, 0.9]$, and a steady state is achieved by time-

Table 2.8: L_2 errors of standard deviation and mean for extra lift and drag for the adaptive ANOVA method with different active dimensions.

D_2	$\sigma(\Delta L)/\varepsilon$	$\sigma(\Delta D)/\varepsilon$	$\mathbb{E}(\Delta L)/\varepsilon^2$	$\mathbb{E}(\Delta D)/\varepsilon^2$
5	1.2967e-2	7.6705e-3	1.3198e-2	2.4486e-3
6	1.2756e-2	5.8900e-3	8.3316e-3	1.5411e-3

marching. The stochastic simulations are based on the probabilistic collocation method (PCM) and both sparse grid and adaptive ANOVA are used. In Figure 2.10 we observe that in the roughness region ($x \in [0, 1]$) the analytical results and numerical results are almost the same. In the first reflection region ($x \in [1, 2.84]$) the errors are mainly from the numerical solver as we see that the numerical results deviate from the corresponding analytical results. In the second reflection region ($x \in [2.84, 6]$) the sampling error dominates as the numerical results are close to the corresponding analytical solution. Also, the difference between the results by the analytical level 1 sparse grid and the analytical adaptive ANOVA method is almost the same as the corresponding difference using the numerical solver. Figure 2.11 shows the errors of standard deviation of extra lift and drag (numerical solutions minus reference solutions) along the wedge obtained with different sampling points. We see that the adaptive ANOVA solutions have the smallest error compared to numerical solutions obtained from other methods. Also, we observe that near the interface $x = 1$ (where the rough surface ends) and at $x = 2.84$ (where the first reflection region ends), the numerical errors are relatively larger. Also, due to the singularity around the apex ($x = 0$), the numerical errors are relatively large close to $x = 0$. We also include the error of analytical solution obtained by the level 2 sparse grid method in Figure 2.11 and compared with the numerical results we can conclude that the numerical error dominates. Figure 2.12 presents the L_2 norm of the errors of the standard deviations of extra lift and drag of analytical and numerical solutions. We can observe that the gaps between the analytical solution and numerical solution reflect the numerical errors. When we increase the level of the sparse grid method, the accuracy improves only a little since the numerical errors dominate unless a finer mesh is used; however such simulations are very expensive. We also notice that the results by the adaptive ANOVA method is better than those by the level 3 sparse

grid method.

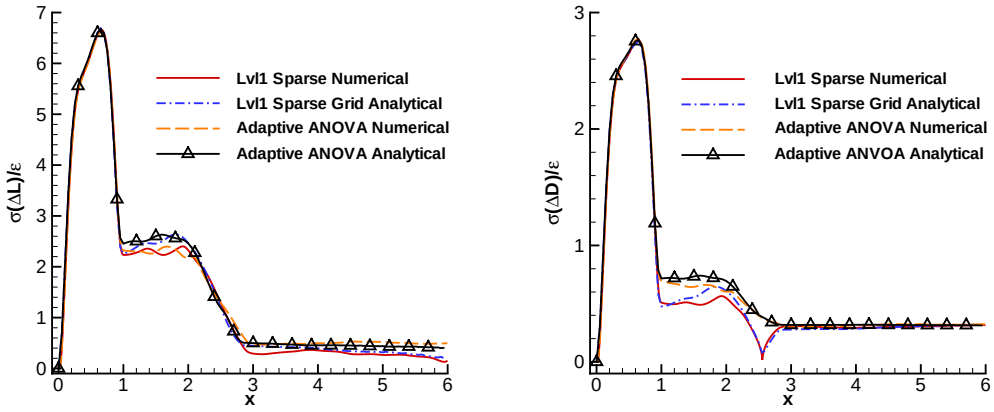


Figure 2.10: Comparison of analytical solution and numerical solution of the standard deviation of extra lift (left) and drag (right) with sparse grid method and adaptive ANOVA method. $A = 0.1, N = 12$.

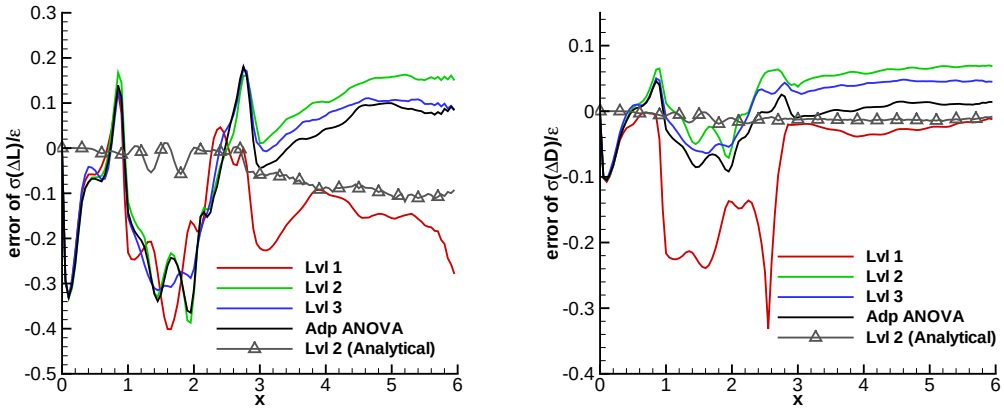


Figure 2.11: Comparison of the error of standard deviation of numerical solution for extra lift (left) and drag (right) (numerical solutions minus reference solutions) with sparse grid method and adaptive ANOVA method ($f_j + f_{1,j} + f_{2,j} + f_{3,j}$). The analytical solution obtained with level 2 sparse grid is also plotted for comparison. $A = 0.1, N = 12$.

A comparison of contours between level 1 sparse grid and adaptive ANOVA of the standard deviation of the extra pressure is also presented in Figure 2.13. We can observe that in the roughness region, where the standard deviation is higher the shock is highly perturbed as we see in the single realization case of Figure 2.1. The interface between the roughness region and the first reflection

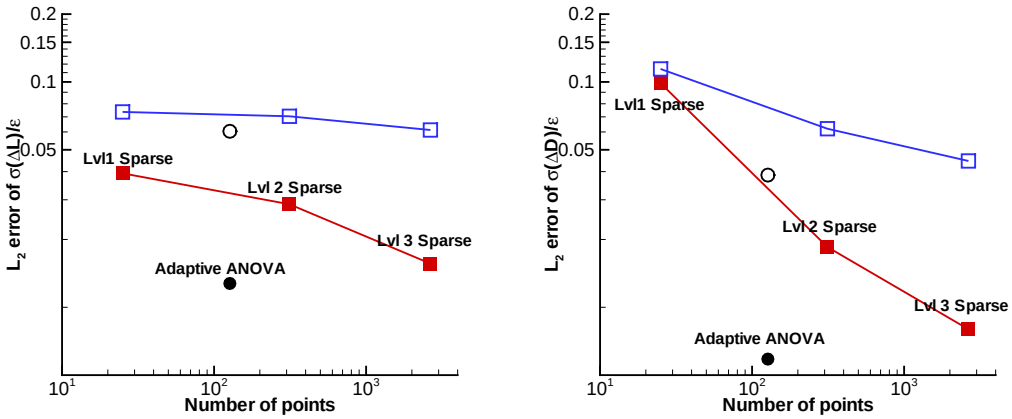


Figure 2.12: Comparison of the error of standard deviation of analytical solution and numerical solution for extra lift (left) and drag (right) with sparse grid method and adaptive ANOVA method. The solid squares are errors of analytical solutions by sparse grid method; the hollow squares are errors of numerical solutions by sparse grid method; the solid circles are analytical solution by adaptive ANOVA using $f_j + f_{1,j} + f_{2,j} + f_{3,j}$; the hollow circles are numerical solution by adaptive ANOVA using $f_j + f_{1,j} + f_{2,j} + f_{3,j}$. $A/d = 0.1, N = 12$.

region is very distinct while the interface between the first and second reflection region is not as clear. This is similar to the single realization shown in Figure 2.1. It is hard to see any difference between the two plots in Figure 2.13 except that at the shock near the right end of the computational domain the standard deviation by the adaptive ANOVA method is larger than that by level 1 sparse grid method.

Finally, we consider a case with correlation length 0.01 with corresponding nominal dimension $N = 100$ (about 95% in (2.4.21)), $\varepsilon = 1.0 \times 10^{-6}$ on physical domain $[0, 5]$. Since the roughness height ε is very small the results by sparse grid and ANOVA are both very accurate. We only present the standard ANOVA method with $\mu = 3, \nu = 1$ in Figure 2.14. The computational details are summarized below:

We can see the results by quasi-Monte Carlo 100,000 realizations) and ANOVA are very close to each other. The errors of standard deviation and mean by both methods are $\mathcal{O}(10^{-4})$. The ANOVA method shows very little improvement to sparse grid method (about $\mathcal{O}(10^{-6})$) and is different from the last case, since using only first-order terms is sufficient here. For this case, since the correlation

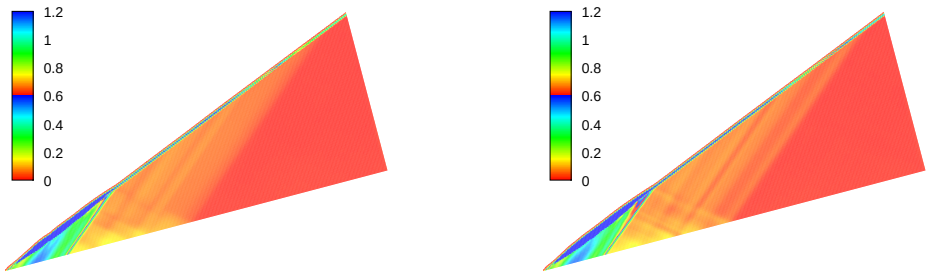


Figure 2.13: Contours of the standard deviation of the normalized extra pressure by level 1 sparse grid method (left) and adaptive ANOVA method (right). $A = 0.1, N = 12$.

Algorithm 3 Summary of computing shock problem with dimension $N = 100$

1: Select the anchor point $\boldsymbol{c} = 0$ and run the deterministic solver at this sampling point to obtain the constant term f_0 in the ANOVA decomposition (according to equation (2.2.1)). In this case f is the extra pressure Δp.

2: For this case, set $\mu = 3$ and quadrature points $q_i^{1,2,3} = -1, 0, 1, i = 1, 2, \cdots, N$ since ξ_i are uniform random variables on $[-\sqrt{3}, \sqrt{3}]$. then the corresponding weights are $w^{1,2,3} = 5/9, 8/9, 5/9$. Run the deterministic solver at the sampling points $(q_1^{1,2,3}, 0, \cdots, 0), (0, q_2^{1,2,3}, 0, \cdots, 0), \cdots, (0, \cdots, 0, q_N^{1,2,3})$ to obtain the value of all first-order terms at quadrature points $q_i^{1,2,3}, i = 1, 2, \cdots, N$ with equation (2.3.2). Since $\nu = 1$ we already get the ANOVA decomposition.

length is small, the change of standard deviation is very sharp at $x = 1$ and $x = 2.84$, i.e., the interface of the roughness region and first reflection as well as the interface of the first and second reflection regions. Also, $x = 1$ is the peak point for the mean while $x = 2.84$ is the point where sharp change appears. Comparing Figures 2.14, 2.15 with Figures 2.4, 2.6 we can observe that a dramatic effect of correlation length on the physics of the scattering problem as the change at the interface of different regions becomes sharp (see the range of roughness region, first reflection region and second reflection region in Figure 2.1). Moreover, when the correlation length is very small, e.g., 0.01, the random roughness at points between 0 and 1 can be approximately treated as independent random variables with the same variance, therefore we observe three very flat steps in the figure of the standard deviation.

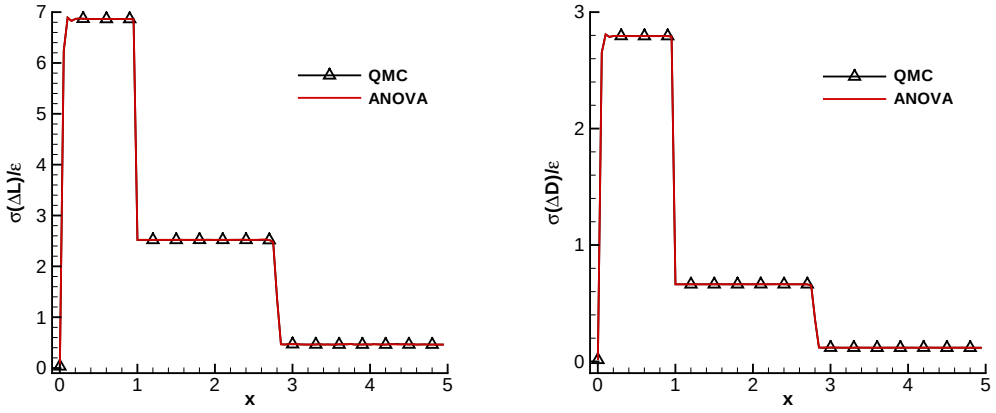


Figure 2.14: Plots of standard deviation of extra lift (left) and drag (right) for ANOVA method. $A = 0.01, N = 100, \mu = 3, \nu = 1$.

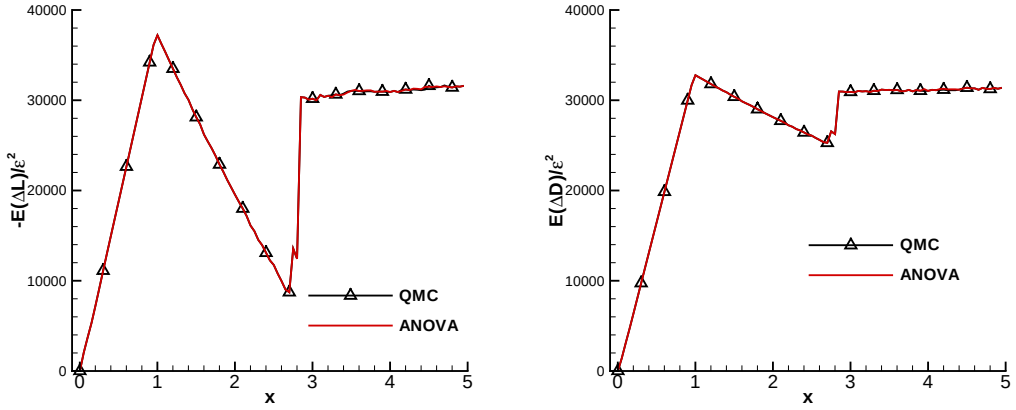


Figure 2.15: Plots of mean of extra lift (left) and drag (right) for ANOVA method. $A = 0.01, N = 100, \mu = 3, \nu = 1$.

2.5 Application to circuit simulation

In this section we investigate an application of ANOVA method in electronic engineering–the circuit model. The system we consider is a ring oscillator, which is a device composed of odd number of not gates whose output oscillated between two voltage levels representing true and false [1]. A schematic of the system is shown in Figure 2.16. This ring oscillator is consists of seven complementary metal-oxide-semiconductor (CMOS) inverters. The schematic of each inverter is shown on the right: it has

one p-type metal-oxide-semiconductor field-effect transistor (MOSFET) and one n-type MOSFET, loaded by a resistor and a capacitor. Here Vdd is the static supply voltage, and GND is the ground. We are interested in the mean and standard deviation of the oscillator's period.

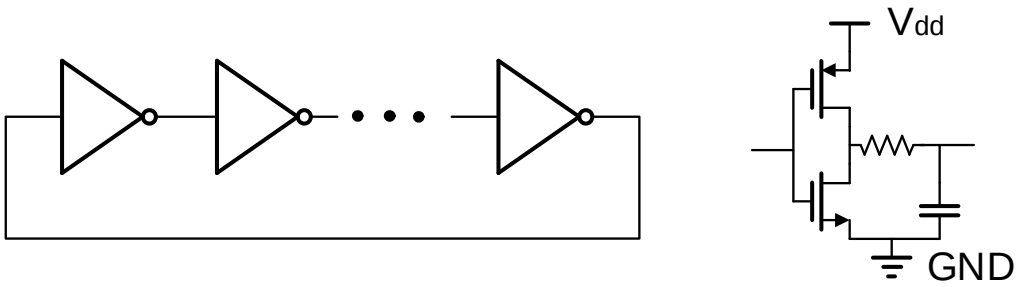


Figure 2.16: Schematic of the 7-stage ring oscillator.

This circuit model contains 57 parameters as follows:

1. The first parameter is the temperature (Gaussian variable), which can influence the flat-band voltage and further lead to the variation of transistor's threshold voltage.

2. The second to 8th parameters are threshold voltage variation for all n-type MOSFET (under zero drain-body bias voltage) , which is Gaussian.

3. The 9th to 15th parameters are threshold voltage variation for all p-type MOSFET (under zero drain-body bias voltage), which is Gaussian.

4. The 16th to 29th parameters describe the gate oxide thickness of all transistors (Gaussian variables).

5. The 30th to 43th parameters describe the uncertainty of the channel width of all transistors. They are uniformly distributed.

6. The 44th to 57th parameters represent the uncertainties of the channel length of all transistors. They are uniformly distributed.

2.5.1 Stochastic testing method

For a general nonlinear circuit with random parameters, we obtain the differential algebraic equation (DAE) by applying modified nodal analysis (MNA) [69]:

$$\frac{\mathrm{d}\boldsymbol{q}(\boldsymbol{x}(t,\boldsymbol{\xi}),\boldsymbol{\xi})}{\mathrm{d}t}+\boldsymbol{f}(\boldsymbol{x}(t,\boldsymbol{\xi}),bx)=\boldsymbol{B}\boldsymbol{u}(t), \tag{2.5.1}$$

where $\boldsymbol{u}(t)\in\mathbb{R}^m$ is the input signal, $\boldsymbol{x}\in\mathbb{R}^n$ is nodal voltages and branch currents, $\boldsymbol{q}\in\mathbb{R}^n$, $\boldsymbol{f}\in\mathbb{R}^n$ represent the charge/flux term and current/voltage terms, respectively. $\boldsymbol{\xi}=(\xi_1,\xi_2,\cdots,\xi_l)$ denotes the random variables describing the device-level uncertainties, which are assumed to be independent. Matrix $\boldsymbol{B}$ is assumed to be independent of $\boldsymbol{\xi}$. We approximate $\boldsymbol{x}$ with a truncated gPC expansion:

$$\boldsymbol{x}(t,\boldsymbol{\xi})\approx\sum_{k=1}^{K}\hat{\boldsymbol{x}}_k(t)\psi_k(\boldsymbol{\xi}). \tag{2.5.2}$$

Let

$$\boldsymbol{X}(t)=\hat{\boldsymbol{x}}_1(t),\hat{\boldsymbol{x}}_2(t),\cdots,\hat{\boldsymbol{x}}_K(t),$$

we aim to solve the following deterministic DAE:

$$\frac{\mathrm{d}\boldsymbol{Q}(\boldsymbol{X}(t))}{\mathrm{d}t}+\boldsymbol{F}(\boldsymbol{X}(t))=\tilde{B}\boldsymbol{u}(t), \tag{2.5.3}$$

where

$$\boldsymbol{Q}(\boldsymbol{X}(t))=\begin{pmatrix}\boldsymbol{q}(\hat{\boldsymbol{x}}(t,\boldsymbol{\xi}^1),\boldsymbol{\xi}^1)\\ \vdots\\ \boldsymbol{q}(\hat{\boldsymbol{x}}(t,\boldsymbol{\xi}^K),\boldsymbol{\xi}^K)\end{pmatrix},\ \boldsymbol{F}(\boldsymbol{X}(t))=\begin{pmatrix}\boldsymbol{f}(\hat{\boldsymbol{x}}(t,\boldsymbol{\xi}^1),\boldsymbol{\xi}^1)\\ \vdots\\ \boldsymbol{f}(\hat{\boldsymbol{x}}(t,\boldsymbol{\xi}^K),\boldsymbol{\xi}^K)\end{pmatrix},\ \tilde{B}=\begin{pmatrix}B\\ \vdots\\ B\end{pmatrix}. \tag{2.5.4}$$

Eq. (2.5.3) are solved by stochastic testing method [171, 172]. This method consists of two key parts: 1. solve the coupled DAE; 2. select the testing nodes (i.e., collocation points). The first

parts consists of Newton's iterations and the details are represented in [171, 172]. We list in 4 the algorithm in [171] to select testing nodes. Here K is the total number of testing nodes, which is the same as the total number of gPC basis, $\boldsymbol{\psi} = (\psi_1, \psi_2, \cdots, \psi_K)$ is the vector of the gPC basis functions. We generate $\hat{n} = P + 1$ 1-D quadrature points first, where P is the highest polynomial order. Then we use tensor product rule to generate $\hat{N} = \hat{n}^l$ quadrature points in l-D random space as the candidate points and implement Algorithm 4 to select testing quadrature.

Algorithm 4 Testing nodes selection

Construct $\hat{N}$ 1-D Gaussian quadrature nodes and weights;
Reorder the quadrature nodes according to the size of the corresponding weights with a descending order, i.e., [w,ind]=sort(w,'descend');
Set $\boldsymbol{\xi}^1 = \boldsymbol{\xi}_k$ and $V = \boldsymbol{\psi}(\boldsymbol{\xi}_k)/\|\boldsymbol{\psi}(\boldsymbol{\xi}_k)\|$ with $k = \text{ind}(1)$, set $m = 1$;
for $j = 2, \cdots, \hat{N}$ **do**
  $k = \text{ind}(j)$, $\boldsymbol{v} = \boldsymbol{\psi}(\boldsymbol{\xi}_k) - V(V^T\boldsymbol{\psi}(\boldsymbol{\xi}_k))$;
  if $\|\boldsymbol{v}\|/\|\boldsymbol{\psi}(\boldsymbol{\xi}_k)\| > \beta$
    $V = [V; \boldsymbol{v}/\|\boldsymbol{v}\|], m = m + 1$;
    $\boldsymbol{\xi}^m = \boldsymbol{\xi}_k$;
    if $m \geq K$, break, **end if**;
  end if
end for

2.5.2 Adaptive ANOVA results

In this case, we consider a up to second-order ANOVA decomposition, i.e., $\nu = 2$. For each component function, we use a third-order gPC expansion to approximate, e.g.,

$$f_{12}(x_1, x_2) \approx \sum_{|\boldsymbol{\alpha}|=0}^{3} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(x_1, x_2).$$

Each component function is simulated with stochastic shooting Newton solver [170, 172] to obtain the gPC coefficients.

The total number of the component functions for a complete third-order ANOVA decomposition is

$$C_{57}^0 + C_{57}^1 + C_{57}^2 = 1654.$$

We firstly compute the results by using third-order gPC expansion, which includes $C_{57+3}^3 = 34220$ terms. The results are compared with those by Monte Carlo method with 10^6 realizations, and the difference is negligible for this problem. Hence, we use the third-order gPC results as the reference solution.

For this specific case, based on the physical properties, we need to keep all the first-order component function in the ANOVA decomposition since they are all important for the oscillation frequency of the oscillator. Then we use **Criterion 1** in Section 2.3 to adaptively choose second-order component functions. We set threshold $\theta_1^1 = 0.001$ and only consider index j such that

$$\frac{\sigma^2(f_j)}{\sum_{k=1}^{57}\sigma^2(f_k)} > \theta_1^1$$

when constructing second-order terms. Based on this criterion, we only consider $j = 1, 2, \ldots, 15$. A plot of $\sigma^2(f_j)/\sum_{k=1}^{57}\sigma^2(f_k)$ is shown in Figure 2.17. With this selection of f_j, we have

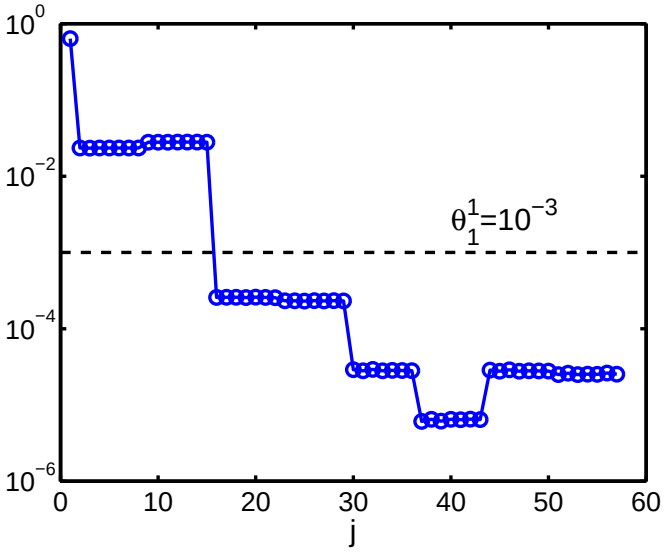


Figure 2.17: $\sigma^2(f_j)/\sum_{k=1}^{57}\sigma^2(f_k)$ of the first order component functions in the ANOVA decomposition of ring the oscillator model.

$$\frac{\sum_{j=1}^{15}\sigma^2(f_j)}{\sum_{k=1}^{57}\sigma^2(f_k)} = 0.996.$$

Now that $\mathcal{F} = \{(i,j)|1 \leq i < j \leq 15\}$, we have $|\mathcal{F}| = 105$. Figure 2.18 shows $\sigma^2 f_{i,j} / \sum_{k=1}^{57} \sigma^2(f_k)$ of 105 sets of (i,j). We set $\theta_2^1 = 5 \times 10^{-7}, 10^{-7}, 10^{-8}$ and present the error of the mean and the

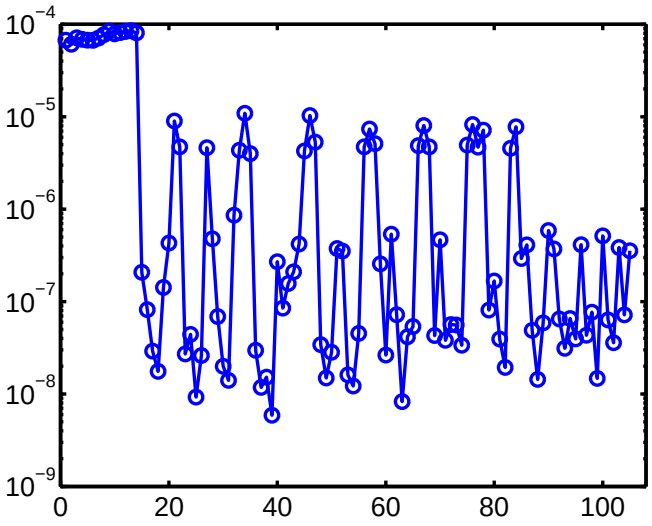


Figure 2.18: $\sigma^2(f_{i,j}) / \sum_{k=1}^{57} \sigma^2(f_k), 1 \leq i < j \leq 15$ of the second-order component functions in the ANOVA decomposition of ring the oscillator model. The total number of $f_{i,j}$ is 105.

standard deviation in Figure 2.19. We can see that as more terms are includes in the ANOVA decomposition, i.e., the smaller the θ_2^1 is, the more accurate estimate of the mean and the standard deviation we can obtain. We can also notice that the improvement of the estimate of the mean is much greater than that of the standard deviation. The small improvement in the estimate of the standard deviation can be understood from Figure 2.18. It is clear that the first 14 terms of $f_{i,j}$ make major contribution to the estimate of standard deviation while the contribution of other $f_{i,j}$ are very small. We also notice that for this case, in order to obtain a good estimate of the mean, we need a very small θ_2^1, which in this case is 10^{-8}. With this selection of θ_2^1, we include 102 second-order component functions in the ANOVA decomposition.

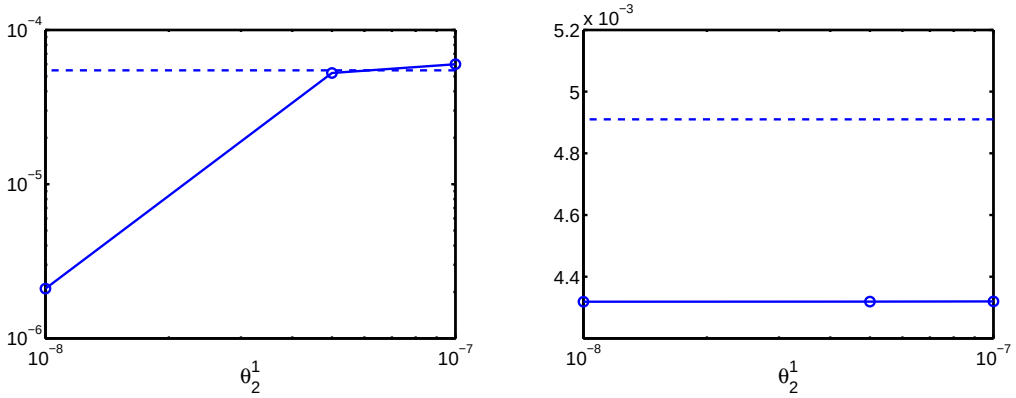


Figure 2.19: Relative error of the mean and the standard deviation with different θ_2^1 for the ring oscillator model. Dash lines are the results without including second-order terms.

Chapter 3

Sparsity

In this chapter we consider problems which exhibit sparsity, i.e., the quantity of interest is "sparse" in random space, and hence the solution can be accurately represented with only a few terms when linearly expanded into a stochastic, e.g., gPC, basis. This is similar to adaptive gPC approach [104]. It is also similar to dimension reduction methods, where the aim is to retain only the most important dimensions, hence the solution can be approximated economically without significantly sacrificing accuracy [105, 117, 107, 148]. For many high-dimensional problems, careful analysis has shown that the number of basis functions with large coefficients is small relative to the cardinality of the full basis. For example, it has been shown in [142, 14] that under some mild conditions, solutions to *elliptic* SPDEs with high-dimensional random coefficients admit sparse representations with respect to gPC basis; see also [66, 70].

Doostan and Owhadi [39] proposed a method for gPC expansion of sparse solutions to SPDEs based on compressive sampling techniques. This method is non-adapted, provably convergent and well suited to problems with high-dimensional random inputs. They applied it to an elliptic equation with random coefficients and demonstrated good results. Their approach provides a more flexible method than the sparse grid method in that the number of the sampling points at a different level of the sparse grid method is fixed once the dimension is decided while number of the sampling

points of this method is arbitrary (but of course generally speaking, more samples will lead to more accurate results). For example, in one of our numerical examples in this chapter, where $d = 40$, i.e., 40 random variables are considered, level 1 sparse grid method requires 81 samples while level 2 sparse grid method needs 3281 samples. In dynamic data-driven applications, e.g., weather forecast, petroleum engineering, ocean modelings, the system can be extremely complicated and the simulation is very costly, and only a small number, e.g., $\mathcal{O}(100)$, of simulations can be afforded. Therefore, it is impossible to obtain more accurate results than level 1 sparse grid or even the level 1 sparse grid may not be applicable when the dimension is really high. Hence, the compressive sampling technique provides a feasible approach for such type of problems provided that based on physical knowledge, mathematical analysis or experience, the solution is sparse in random space. Our main improvement of the method proposed by Doostan and Owhadi is that by employing the reweighted procedure, which is popular in compressive sensing [29, 79, 162], we can greatly enhance the accuracy of the approximated solution. Furthermore, we combine the technique of selecting sampling points based on Chebyshev measure [123] to enhance the performance of the results.

3.1 Overview of compressive sensing method

Compressive sensing is a signal processing technique for efficiently acquiring and reconstructing a signal by solving a underdetermined system. A very simple model of sending a signal is as follows: given a *signal* $\boldsymbol{c}$, we *code* it with a *measurement matrix* $\boldsymbol{\Psi}$ to obtain *observation* $\boldsymbol{y} = \boldsymbol{\Psi c}$. Notice that $\boldsymbol{\Psi}$ is a $m \times N$ matrix with $m < N$ (or usually $m \ll N$), the length of $\boldsymbol{y}$ is smaller than that of $\boldsymbol{c}$. We send $\boldsymbol{y}$ to the receiver and the receiver, who already knows $\boldsymbol{\Psi}$ in advance, needs to find $\boldsymbol{c}$ with $\boldsymbol{\Psi}$ and $\boldsymbol{y}$, which is called *decoding*. For signal processing and image processing, this is possible since the signal $\boldsymbol{c}$ is *sparse*, which means that most entries of $\boldsymbol{c}$ is zeros. In the famous paper by Candès, Tao, Donoho [28, 37] it is proved that given knowledge about a signal's sparsity, the signal may be reconstructed with fewer samples than the Nyquist-Shannon theorem requires. The key

point is to solve the optimization problem (P_h)

$$(P_h): \qquad \min_c \|\boldsymbol{c}\|_h \quad \text{subject to} \quad \boldsymbol{\Psi c} = \boldsymbol{u}, \tag{3.1.1}$$

where h is usually taken as 0 or 1. When $h = 0$, the "ℓ_0 norm" is defined as the non-zeros entries of a vector:

$$\|\boldsymbol{c}\|_0 = \#\{i | c_i \neq 0\}. \tag{3.1.2}$$

However, if we solve this problem by combinatorial search, i.e., we sweep exhaustively through all possible sparse subsets, the complexity is exponential in m, which is the length of $\boldsymbol{c}$. It has been proven that (P_0) is, in general, NP-hard [113]. Therefore the most popular method for compressive sensing is the ℓ_1 minimization, which relax the optimization problem (P_0) to a convex optimization problem (P_1). Here the ℓ_1 norm is defined as

$$\|\boldsymbol{c}\|_1 = \sum_i |c_i|. \tag{3.1.3}$$

Moreover, if the observation is contaminated by noise, i.e.,

$$\boldsymbol{y} = \boldsymbol{\Psi c} + \boldsymbol{\varepsilon},$$

then we need to modify (P_h) to be

$$(P_{h,\epsilon}): \qquad \min_c \|\boldsymbol{c}\|_h \quad \text{subject to} \quad \|\boldsymbol{\Psi c} - \boldsymbol{u}\|_2 \leq \epsilon, \tag{3.1.4}$$

where $\epsilon = \|\boldsymbol{\varepsilon}\|_2$. In order to solve $(P_{1,\epsilon})$, all the classical convex optimization solver is applicable, e.g., `CVX` [59]. Moreover, algorithms and solvers targeting the sparse recovering, e.g., `SPGL1` package [146], ℓ_1`-MAGIC` [3]. Also, in the recent years, split Bregman method has become popular, e.g., [58, 166, 26, 25].

3.1.1 Greedy algorithm for ℓ_0 minimization

Here we present the greedy method named *Orthogonal Matching Pursuit* (OMP) for solving ℓ_0 minimization problem (P_0) [22]:

Algorithm 5 Approximate the solution of (P_0): $\min_{\boldsymbol{c}} \|\boldsymbol{c}\|_0$ subject to $\boldsymbol{\Psi c} = \boldsymbol{y}$.

1: Initialize $k = 0$ and $\boldsymbol{c}^0 = 0$, residual $\boldsymbol{r}^0 = \boldsymbol{b} - \boldsymbol{\Psi c}^0 = \boldsymbol{y}$, solution support $\mathcal{S}^0 = \text{Support}\{\boldsymbol{c}^0\} = \emptyset$.
2: $k = k + 1$, compute

$$\epsilon(j) = \min_{z_j} \| \boldsymbol{\Psi}_j z_j - \boldsymbol{r}^{k-1} \|_2^2$$

for all j using the optimal choice

$$z_j^* = \boldsymbol{\Psi}_j^T \boldsymbol{r}^{k-1} / \| \boldsymbol{\Psi}_j \|_2^2.$$

3: Find a minimizer j_0 for $\epsilon(j) : \forall j \notin S^{k-1}$, $\epsilon(j_0) < \epsilon(j)$ and update $\mathcal{S}^k = \mathcal{S}^{k-1} \cup \{j_0\}$.
4: Compute $\boldsymbol{c}^k$, the minimizer of $\| \boldsymbol{\Psi c} - \boldsymbol{y} \|_2^2$ subject to $\text{Support}\{\boldsymbol{c}\} = \mathcal{S}^k$.
5: Update the residual: $\boldsymbol{r}^k = \boldsymbol{y} - \boldsymbol{\Psi c}^k$.
6: Stop the iteration if $\|\boldsymbol{r}^k\|_2 < \epsilon_0$, where ϵ_0 is the stopping threshold. Otherwise go to step 2.

If the signal $\boldsymbol{c}$ has k_0 nonzeros, the method requires $\mathcal{O}(k_0 mn)$ flops in general. This is dramatically better than the exhaustive search, which requires $\mathcal{O}(nm^{k_0}k^2)$ flops. Hence, this method can be much more efficient-IF IT WORKS! However, this single-term-at-a-time strategy can fail badly, i.e., there are explicit examples (see [36, 140, 141]) where a simple k-term representation is possible, but this approach yields an n-term (i.e., dense) representation. Many variants on this method are available, e.g., [30, 34, 108, 119, 144]. In the last part of this chapter, we will use OMP method to construct a sparse surrogate model for the DPD.

3.1.2 Reweighted ℓ_1 minimization method

Candès et al. [29] proposed the reweighted ℓ_1 minimization, which employed the weighted norm and iterations to enhance the sparsity of the solution. This algorithm consists of the following four steps:

1. Set the iteration count l to zero and $w_i^{(0)} = 1,\ i = 1, \cdots, N$.

2. Solve the weighted ℓ_1 minimization problem

$$\boldsymbol{c}^{(l)} = \arg\min \|W^{(l)}\boldsymbol{c}\|_1 \quad \text{subject to} \quad \|\boldsymbol{\Psi c} - \boldsymbol{u}\|_2 \leq \epsilon,$$

where W is a diagonal matrix with $W_{j,j}^{(l)} = w_j^{(l)}$.

3. Update the weights: for each $i = 1, 2, \cdots, N$,

$$w_i^{(l+1)} = \frac{1}{|c_i^{(l)}| + \tau}.$$

4. Terminate upon convergence or when l reaches a specified maximum number of iterations $l_{\max}$. Otherwise, increment l and go to step 2.

Remark 3.1.1. According to [29], in step 3, the parameter τ is introduced to provide stability and to ensure that a zero-valued component in $\boldsymbol{c}^{(l)}$ does not strictly prohibit a nonzero estimate at the next step. It should be set slightly smaller than the expected nonzero magnitudes of $\boldsymbol{c}_0$, which is a sparse vector approximating $\boldsymbol{c}$. Empirically, τ is set around $0.1 \sim 0.001$. In step 4, the convergence of $\boldsymbol{c}$ can be tested by comparing the difference between $\boldsymbol{c}^{(l)}$ and $\boldsymbol{c}^{(l+1)}$. Usually, $l_{\max}$ is set so that it controls the number of iterations.

From the description of this algorithm, we notice that it repeats the ℓ_1 minimization for different weighted ℓ_1 norm in different steps. Hence, when $l_{\max}$ is fixed, we actually need $l_{\max}$ additional optimization solves compared to the regular ℓ_1 minimization. A series of tests in [29, 79, 162] demonstrated remarkable performance and broad applicability of this algorithm in the areas of sparse signal recovery, statistical estimation, error correction and image processing with not only sparse but also nearly-sparse representations. Moreover, these tests showed that usually three or four iterations are enough to obtain good performance. Notice that in this algorithm there is no requirement or modification for the measurement matrix, therefore it can be applied to any linear system including (3.2.6). Needell [114] provided an analytical result of the improvement in the error

bound by using the reweighted ℓ_1 minimization over the ℓ_1 minimization under some conditions:

Theorem 3.1.2. *([114]) Assume that $\boldsymbol{\Psi}$ satisfies the RIP condition with $\delta_{2s} < \sqrt{2}-1$. Let $\boldsymbol{c}$ be an arbitrary vector with noisy measurements $\boldsymbol{y} = \boldsymbol{\Psi c} + \boldsymbol{e}$, where $\|\boldsymbol{e}\|_2 < \epsilon$. Assume that the smallest nonzero coordinate η of $\boldsymbol{c}_s$ satisfies $\eta > \frac{4\alpha\epsilon_0}{1-\rho}$, where $\epsilon_0 = 1.2(\|\boldsymbol{c}-\boldsymbol{c}_s\|_2 + \frac{1}{\sqrt{s}}\sigma_s(c)_1) + \epsilon$ and $\|\boldsymbol{c}_s - \boldsymbol{c}\|_1 = \sigma_s(\boldsymbol{c})_1$. Then the limiting approximation from reweighted ℓ_1 minimization satisfies*

$$\|\boldsymbol{c} - \hat{\boldsymbol{c}}\|_2 \leq \frac{4.1\alpha}{1+\rho}\left(\epsilon + \frac{\sigma_{s/2}(c)_1}{\sqrt{s}}\right), \tag{3.1.5}$$

and

$$\|\boldsymbol{c} - \hat{\boldsymbol{c}}\|_2 \leq \frac{2.4\alpha}{1+\rho}\left(\epsilon + \frac{\sigma_s(c)_1}{\sqrt{s}} + \|\boldsymbol{c}-\boldsymbol{c}_s\|_2\right), \tag{3.1.6}$$

where the definitions of $\alpha, \rho, \sigma_s(c)_p$ are the same as in Theorem 3.2.3.

In practice, we only use one to three reweighted iterations, the following lemma in [114] is more useful:

Lemma 3.1.3. *(Single reweighted ℓ_1 minimization [114]): Assume that $\boldsymbol{\Psi}$ satisfies the RIP condition with $\delta_{2s} < \sqrt{2}-1$. Let $\boldsymbol{c}$ be an arbitrary vector with noisy measurements $\boldsymbol{y} = \boldsymbol{\Psi c} + \boldsymbol{e}$, where $\|\boldsymbol{e}\|_2 < \epsilon$. Let $\boldsymbol{w}$ be a vector such that $\|\boldsymbol{w}-\boldsymbol{c}\|_\infty \leq B$ for some constant B. Denote by $\boldsymbol{c}_s$ the vector consisting of the s (where $s \leq |supp(\boldsymbol{c})|$) largest coefficients of $\boldsymbol{c}$ in absolute value. Let η be the smallest coordinate of $\boldsymbol{c}_s$ in absolute value, and set $b = \|\boldsymbol{c}-\boldsymbol{c}_s\|_\infty$. Then when $\eta \geq B$ and $\rho C_1 < 1$, the approximation from reweighted ℓ_1 minimization using weights $\delta_i = 1/(w_i + \tau)$ satisfies*

$$\|\boldsymbol{c} - \hat{\boldsymbol{c}}\|_2 \leq D_1\epsilon + D_2\frac{\sigma_s(c)_1}{\tau}, \tag{3.1.7}$$

where

$$D_1 = \frac{(1+C_1)\alpha}{1-\rho C_1},\ D_2 = C_2 + \frac{(1+C_1)\rho C_2}{1-\rho C_1},\ C_1 = \frac{B+\tau+b}{\eta - B + \tau},\ C_2 = \frac{2(B+\tau+b)}{\sqrt{s}},$$

and $\rho, \alpha, \delta_s(c)_p$ *are as in Theorem 3.2.3.*

3.1.3 Sampling strategy

According to the gPC theory, Legendre polynomial is the appropriate basis for the uniformly distributed random variables [157]. Therefore, if the system involves uniform random variables the solution can be expanded in terms of *Legendre polynomials* and the selection of sampling points are based on the uniform distribution. With this setting, the mutual coherence $\mu(\boldsymbol{\Psi})$ converges to zero almost surely for asymptotically large random sample size m as pointed out in [39], and several theories were proposed to estimate the number of samples for different error level ϵ. Noticing the special structure of the measurement matrix $\boldsymbol{\Psi}$ consisting of Legendre polynomial, Rauhut and Ward [123] proposed a sampling strategy based on the Chebyshev measure $\mathrm{d}\nu(x) = \pi^{-1}(1 - x^2)^{-1/2}\mathrm{d}x$ for recovering the Legendre sparse polynomials from a few samples for 1D (random space) problem. We know that if a random variable X is uniformly distributed on $[0, \pi]$, then $Y = \cos X$ is distributed according to the Chebyshev measure. Hence, in order to generate a sampling point based on Chebyshev measure, we first generate a sampling point x according to $U[0, \pi]$, then $y = \cos(x)$ is the point we need. Equation $(P_{1,\epsilon})$ is now modified as

$$(P_{1,\epsilon}^{Cheb}): \qquad \min_{\boldsymbol{c}} \|\boldsymbol{c}\|_1 \quad \text{subject to} \quad \|\boldsymbol{A}\boldsymbol{\Psi}\boldsymbol{c} - \boldsymbol{A}\boldsymbol{u}\|_2 \leq \epsilon, \tag{3.1.8}$$

where $\boldsymbol{A}$ is an $m \times m$ diagonal matrix with $A_{j,j} = (\pi/2)^{1/2}(1 - \xi_j^2)^{1/4}$ and $\xi_j, j = 1, 2, \cdots, m$ are the sampling points based on Chebyshev measure, $\boldsymbol{\Psi}_{j,n} = L_{n-1}(\xi_j)$ and L_n is the n-th order normalized Legendre polynomial. As was shown in [123], this strategy can improve the accuracy of the recovery for 1-D (random space) problem. Moreover, this method can be extended to a large class of orthogonal polynomials, including the Jacobi polynomials, of which the Legendre polynomials are a special case. The main theoretical conclusion is the following:

Definition 3.1.4. ([123]) Let $\psi_j, j \in [N] \overset{\text{def}}{=} \{1, 2, \cdots, N\}$ be an orthonormal system of functions

on $\mathcal{D}$, i.e.,

$$\int_{\mathcal{D}} \psi_j(x)\psi_k(x)\mathrm{d}\nu(x) = \delta_{j,k}, j, k \in [N].$$

If this orthonormal system is uniformly bounded,

$$\sup_{j\in[N]} \|\psi_j\|_\infty = \sup_{j\in N} \sup_{x\in\mathcal{D}} |\psi_j(x)| \leq K \tag{3.1.9}$$

for some constant $K \geq 1$, we call the systems $\{\psi_j\}$ satisfying this condition *bounded orthonormal systems.*

Theorem 3.1.5. *([123]) (RIP for bounded orthonormal systems). Consider the matrix $\boldsymbol{\Psi} \in \mathbb{R}^{m\times N}$ with entries*

$$\Psi_{l,k} = \psi_k(x_l), \quad l \in [m], k \in [N],$$

formed by i.i.d. samples x_l drawn from the orthogonalization measure ν associated to the bounded orthonormal system $\{\psi_j, j \in [N]\}$ having uniform bound $K \geq 1$ in (3.1.9)*. If*

$$m \geq C\delta^{-2}K^2 s \log^3(s)\log(N), \tag{3.1.10}$$

then with probability at least $1 - N^{-\gamma\log^3(s)}$, the restricted isometry constant δ_s of $\frac{1}{\sqrt{m}}\boldsymbol{\Psi}$ satisfies $\delta_s \leq \delta$. The constants $C, \gamma > 0$ are universal.

Remark 3.1.6. ([123]) Consider the functions

$$Q_n(x) = \sqrt{\frac{\pi}{2}}(1-x^2)^{1/4}L_n(x),$$

where L_n are n-th order normalized Legendre polynomials. According to Lemma 5.1 in [123], $|Q_n(x)| < \sqrt{2+\frac{1}{n}}$ and $\{Q_n(x)\}$ forms a bounded orthonormal system with respect to Chebyshev measure. The matrix $\boldsymbol{\Phi}$ with entries $\Phi_{j,n} = Q_{n-1}(x_j)$, which is used in the constraint $\|\boldsymbol{\Phi c} - \boldsymbol{A u}\| \leq \epsilon$,

can be written as $\boldsymbol{\Phi} = \boldsymbol{A}\boldsymbol{\Psi}$ as in $(P_{1,\epsilon}^{Cheb})$. Then the system $\{Q_n\}$ is uniformly bounded on $[-1,1]$ and satisfies the bound $\|Q_n\|_\infty \leq \sqrt{3}$, i.e., $K = \sqrt{3}$ for this system. Therefore, according to Theorem 3.1.5, we can expect a high probability of good recovery based on a relatively small number of samples compared to regular ℓ_1 minimization.

We can extend the Chebyshev measure based sampling points strategy from 1-D to multi-D. Since $\xi_1, \xi_2, \cdots, \xi_d$ are i.i.d random variables and the basis functions used to present the solution are in the tensor product form (see Eq. (3.3.4)), we modify $(P_{1,\epsilon}^{Cheb})$ by setting $A_{j,j} = \prod_{k=1}^{d'} (\pi/2)^{1/2}(1-(\boldsymbol{\xi}_j)_k)^{1/4}$, i.e., we use the tensor product form in the weight as in [163]. Here $\boldsymbol{\xi}_j$ is the j-th sampling point, which includes d entries and $(\boldsymbol{\xi}_j)_k$ is the k entry of it. Notice that we use d' instead of d here because according to our numerical tests, $d' = d$ does not necessarily provide good performance, hence it is better to take $d' < d$. This is consistent with the conclusion in [163], which shows that the Chebyshev measure may become less efficient in high-dimensional cases. An important reason is that the uniform bound K in Theorem 3.1.5 will increase as d increases, which implies that more samples are needed, see also [163]. More precisely, when generating a d dimensional sampling point $(\xi_1, \xi_2, \cdots, \xi_d)$, we generate the first d' entries $\xi_1, \xi_2, \cdots, \xi_{d'}$ according to Chebyshev measure and the remaining entries $\xi_{d'+1}, \cdots, \xi_d$ according to uniform distribution.

A brief summary of the theorems presented in this section is as follows: Theorem 3.1.5 and Remark 3.1.6 provide the estimate of the size of sampling points we need to obtain a measurement matrix with small RIP constant when the sampling points are based on Chebyshev measure. After obtaining observations and the measurement matrix, Theorem 3.1.2 shows that by employing reweighted iterations, we can expect a lower error bound than that in Theorem 3.2.3, which is the error estimate for regular ℓ_1 minimization. Alternatively, if the number of sampling points is fixed, Chebyshev points allow a larger s (but still relative small compared with N to guarantee sparsity) in the RIP condition, hence the upper bound of the error in Theorem 3.2.3 can be reduced. Then, with the reweighted iterations, this bound can be reduced further.

3.1.4 Cross validation

The cross-validation method we use in this section follows the description in Section 3.5 of [39]. We first divide the m available solution samples to m_r reconstruction and m_v validation samples such that $m = m_r + m_v$. Then repeat the ℓ_1 minimization $(P_{1,\epsilon})$ (not the reweighted iteration) on the reconstruction samples and with multiple values of truncation error tolerance ϵ_r. In this section we test 11 different ϵ_r from $[10^{-4}, 10^{-2}]$ with constant ratio. Next, set $\epsilon = \sqrt{m/m_r}\hat{\epsilon}_r$, where $\hat{\epsilon}_r$ is such that the corresponding truncation error on the m_v validation samples is minimum. Finally, we repeat the above cross-validation algorithm for multiple replications of the reconstruction and validation samples. The estimate of $\epsilon = \sqrt{m/m_r}\hat{\epsilon}_r$ is then based on the values of $\hat{\epsilon}_r$ for which the average of the corresponding truncation errors ϵ_v, over all replications of the validation samples, is minimum. In Section 3.3 we set $m_r \approx 3m/4$ and performed the cross-validation for four replications. In Section 3.4 we set $m_r \approx 2m/3$ and performed the cross-validation for three replications. More details can be found in [39]. Figure 3.1 is a demonstration of the cross-validation with $m_r \approx 2m/3$.

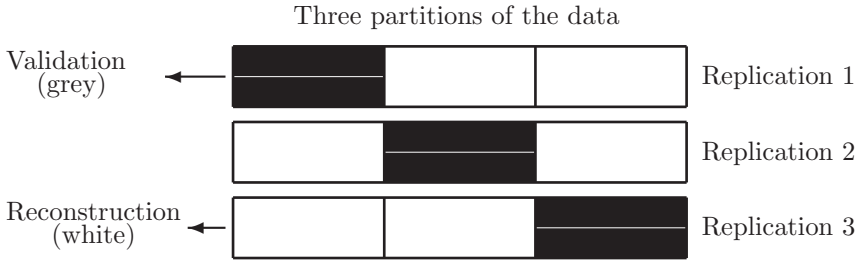


Figure 3.1: Demonstration of the cross-validation. The data is partitioned into three parts. For each replication, the white parts are used in reconstruction and the grey part is used for validation.

3.2 Compressive sensing based polynomial chaos

Consider an underdetermined linear system of equations $\boldsymbol{\Psi c} = \boldsymbol{u}$, where $\boldsymbol{\Psi}$ (the "measurement matrix") is an $m \times N$ matrix with $m < N$ (usually $m \ll N$), and $\boldsymbol{c}$ and $\boldsymbol{u}$ (the "observation")

are vectors of length N and m, respectively. In order to find a "sparse" solution, we consider the following optimization problem:

$$(P_{h,\epsilon}): \qquad \min_{\boldsymbol{c}} \|\boldsymbol{c}\|_h \quad \text{subject to} \quad \|\boldsymbol{\Psi c} - \boldsymbol{u}\|_2 \leq \epsilon, \tag{3.2.1}$$

where we introduce the tolerance ϵ to make $(P_{h,\epsilon})$ more general since in data-driven systems there may be noise in the observations. The sparsity of the solution $\boldsymbol{c}$ is best described by the ℓ_0 "norm" (number of nonzero entries of $\boldsymbol{c}$):

$$\|\boldsymbol{c}\|_0 \stackrel{\text{def}}{=} |\{i : c_i \neq 0\}|,$$

as smaller ℓ_0 "norm" indicates more sparse $\boldsymbol{c}$. In order to avoid the NP-hard problem of exhaustive search to solve $(P_{0,\epsilon})$, a type of greedy algorithm called *orthogonal matching pursuit* (OMP) can be employed [22] as seem at the beginning of this section. Alternatively, one can relax the ℓ_0 "norm" to ℓ_1 norm, therefore replace $(P_{0,\epsilon})$ with a convex optimization problem $(P_{1,\epsilon})$, which is the ℓ_1 minimization method in compressive sensing. If the solution is "nearly sparse" instead of "sparse", i.e., the zero entries of $\boldsymbol{c}$ are replaced by relatively small numbers which will only change $\boldsymbol{\Psi c}$ a little, we can still use $(P_{h,\epsilon})$ by considering this small change as "noise" in the observation. In the problems we consider in this section, the exact solutions are always *nearly sparse*. Next, we recall two fundamental concepts in the ℓ_1 minimization.

Definition 3.2.1. (*Mutual coherences* [22, 38]) The mutual coherence $\mu(\boldsymbol{\Psi})$ of a matrix $\boldsymbol{\Psi} \in \mathbb{R}^{m\times N}$ is the maximum of absolute normalized inner-products of its columns. Let Ψ_j and Ψ_k be two columns of $\boldsymbol{\Psi}$. Then,

$$\mu(\boldsymbol{\Psi}) \stackrel{\text{def}}{=} \max_{1\leq j,k\leq N, j\neq k} \frac{|\Psi_j^T \Psi_k|}{\|\Psi_j\|_2 \|\Psi_k\|_2}. \tag{3.2.2}$$

In other words, the mutual coherence measures how close $\boldsymbol{\Psi}$ is to orthogonal matrix. It is clear that for a general matrix A,

$$0 \leq \mu(\mathsf{A}) \leq 1.$$

For instance, if $\boldsymbol{A}$ is unitary matrix, then $\mu(\boldsymbol{A}) = 0$. However, since we always consider the case of $m < N$ in compressive sensing, $\mu(\boldsymbol{\Psi})$ is strictly positive. It is understood that a measurement matrix with smaller mutual coherence can better recover a sparse solution by compressive sensing techniques, e.g., Lemma 3.4 of [39].

Definition 3.2.2. (*Restricted isometries* [28]) Let $\boldsymbol{\Psi}$ be a matrix with a finite collection of vectors $(\Psi_j)_{j\in J} \in \mathbb{R}^m$ as columns, where J is a set of indices. For every integer $1 \leq s \leq |J|$, we define the s-restricted isometry constant δ_s to be the smallest quantity such that $\boldsymbol{\Psi}_T$ obeys

$$(1-\delta_s)\|\boldsymbol{c}\|_2^2 \leq \|\boldsymbol{\Psi}_T\boldsymbol{c}\|_2^2 \leq (1+\delta_s)\|\boldsymbol{c}\|_2^2 \tag{3.2.3}$$

for all subsets $T \subset J$ of cardinality at most s, and all real coefficients $(c_j)_{j\in T}$. Here $\boldsymbol{\Psi}_T$ is the submatrix of $\boldsymbol{\Psi}$ with column indices $j \in T$ so that

$$\boldsymbol{\Psi}_T\boldsymbol{c} = \sum_{j\in T} c_j\Psi_j.$$

In other words, for any *s-sparse* vector $\boldsymbol{c}$ (the number of non-zero entries is at most s), δ_s measures whether $\boldsymbol{\Psi}_T$ behaves like an orthonormal matrix. Informally, the matrix $\boldsymbol{\Psi}$ is said to have the restricted isometry property (RIP) if δ_s is small for s reasonably large compared to m.

Note that both the concepts of mutual coherence and restricted isometry describe the same property of the measurement matrix, i.e., roughly speaking, how far sparse subsets of the columns of it are from being an isometry. This is a key idea in compressive sensing, and to construct a measurement matrix with small mutual coherence or small restricted isometry constant is crucial. In this section we mainly concentrate on the latter when we refer to theoretical analysis. A well known theorem is the following:

Theorem 3.2.3. *(Sparse recovery for RIP-matrices [27]). Assume that the restricted isometry constant of* $\boldsymbol{\Psi}$ *satisfies* $\delta_{2s} < \sqrt{2}-1$. *Let* $\boldsymbol{c}$ *be an arbitrary signal with noisy measurements* $\boldsymbol{y} =$

$\boldsymbol{\Psi c} + \boldsymbol{e}$, where $\|\boldsymbol{e}\|_2 < \epsilon$. Then the approximation $\hat{\boldsymbol{c}}$ to $\boldsymbol{c}$ from ℓ_1 minimization $(P_{1,\epsilon})$ satisfies

$$\|\boldsymbol{c} - \hat{\boldsymbol{c}}\|_2 \leq C_1\epsilon + C_2\frac{\sigma_s(c)_1}{\sqrt{s}}, \tag{3.2.4}$$

where

$$C_1 = \frac{2\alpha}{1-\rho},\ C_2 = \frac{2(1+\rho)}{1-\rho},\ \rho = \frac{\sqrt{2}\delta_{2s}}{1-\delta_{2s}},\ \alpha = \frac{2\sqrt{1+\delta_{2s}}}{\sqrt{1-\delta_{2s}}},\ \sigma_s(\boldsymbol{c})_p = \inf_{z:\|\boldsymbol{z}\|_0 \leq s} \|\boldsymbol{c} - \boldsymbol{z}\|_p.$$

Remark 3.2.4. Notice that in Theorem 3.2.3 the bound for δ_{2s} is $\sqrt{2} - 1$. There are sharper estimates for this bound, e.g., [50, 51]. Since we are not concentrating on ℓ_1 minimization itself, we quote the original estimate by Candès [27].

Remark 3.2.5. Instead of Eq. (3.2.4), there is another form of estimate [39, 22, 38], which relates the error with the tolerance ϵ and the number of basis N and provides the probability of solutions satisfying the error bound.

In classical compressive sensing method, several choices of the measurement matrix are employed, e.g., Gaussian random matrix, Bernoulli random matrix, etc. For SPDEs, we can expand the solution with gPC basis :

$$u(\boldsymbol{x};\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}}(\boldsymbol{x})\psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}), \tag{3.2.5}$$

where $\boldsymbol{x}$ is the variable in the physical space and $\boldsymbol{\xi}$ is the random vector, $\psi_{\boldsymbol{\alpha}}$ are gPC basis functions, and $\boldsymbol{\alpha}$ are the indices. For each fixed $\boldsymbol{x}$, u is a linear combination of the gPC basis, hence when different sampling points $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \cdots$ are employed, Eq. (3.2.5) can be cast into a linear system:

$$\boldsymbol{\Psi c} = \boldsymbol{u}, \tag{3.2.6}$$

where each entry of the measurement matrix is $\Psi_{i,j} = \psi_{\boldsymbol{\alpha}_j}(\boldsymbol{\xi}_i)$ with $\boldsymbol{\xi}_i$ being sampling points in random space, and each entry of vector $\boldsymbol{u}$ being $u_i = u(\boldsymbol{x};\boldsymbol{\xi}_i)$, i.e., the output of the deterministic

solver with sampling point $\boldsymbol{\xi}_i$. We call these u_i samples of the solution. In [39] the authors expanded the solutions with Legendre polynomial and used Monte Carlo points based on uniform distribution. According to the law of large numbers and the orthogonality of the Legendre polynomials, the mutual coherence converges to zero for asymptotically large random sample sizes m. Also, with this choice of sampling points, the compressive sensing method can be considered as a post processing of Monte Carlo method and the benefit is that we can arbitrarily increase the number of sampling points without wasting any realizations we already obtained. We note that the nested sparse grid method can also reuse the sampling points of the lower levels while the number of additional samples from one level to the next level is fixed.

3.3 Compressive sensing based polynomial chaos for solving elliptic equations

We consider the following problem: let $(\Omega, \mathcal{F}, \mathcal{P})$ be a probability space where $\mathcal{P}$ is a probability measure on the σ-field $\mathcal{F}$. We consider the following SPDE defined on a bounded Lipschitz continuous domain $\mathcal{D} \subset \mathbb{R}^D, D = 1, 2, 3$, with boundary $\partial\mathcal{D}$,

$$\begin{aligned} -\nabla \cdot (a(\boldsymbol{x};\omega)\nabla u(\boldsymbol{x};\omega)) &= f(\boldsymbol{x}) \quad \boldsymbol{x} \in \mathcal{D}, \\ u(\boldsymbol{x};\omega) &= 0 \quad \boldsymbol{x} \in \partial\mathcal{D}, \end{aligned} \tag{3.3.1}$$

where ω is an "event" in probability space Ω. The diffusion coefficients is represented by a Karhunen-Loève (KL) expansion:

$$a(\boldsymbol{x};\omega) = \bar{a}(\boldsymbol{x}) + \sigma_a \sum_{i=1}^{d} \sqrt{\lambda_i}\phi_i(\boldsymbol{x})\xi_i(\omega), \tag{3.3.2}$$

where $(\lambda_i, \phi_i), i = 1, 2, ..., d$ are eigenpairs of the covariance function $C_{aa}(\boldsymbol{x}_1, \boldsymbol{x}_2) \in L_2(\mathcal{D} \times \mathcal{D})$ of $a(\boldsymbol{x};\boldsymbol{\xi})$, $\bar{a}(\boldsymbol{x})$ is the mean and σ_a is the standard deviation. ξ_i are i.i.d uniformly distributed random

variables on $[-1, 1]$.

3.3.1 Theoretical results for the sparsity

Additional requirement on $a(\boldsymbol{x}; \omega)$ are:

- For all $\boldsymbol{x} \in \mathcal{D}$, there exists constants $a_{\min}$ and $a_{\max}$ such that

$$0 < a_{\min} \leq a(\boldsymbol{x}; \omega) \leq a_{\max} \leq \infty \quad \text{a.s.}$$

- The covariance function $C_{aa}(\boldsymbol{x}_1, \boldsymbol{x}_2)$ is piecewise analytic on $\mathcal{D} \times \mathcal{D}$ [14], implying that there exist real constants c_1 and c_2 such that for $i = 1, \cdots, d$,

$$0 \leq \lambda_i \leq c_1 \mathrm{e}^{-c_2 i^{\kappa}},$$

  where $\kappa := 1/D$ and $\boldsymbol{\alpha} \in \mathbb{N}^d$ is a fixed multi-index.

Here the second requirement guarantees the existence of a sparse solution for problem (3.3.1) [14]. With the setting of Eq. (3.3.2), a and u depend on $\boldsymbol{x}$ and $\boldsymbol{\xi}$, so we omit ω and use the notation $a(\boldsymbol{x}; \boldsymbol{\xi})$ and $u(\boldsymbol{x}; \boldsymbol{\xi})$.

In the context of the gPC method, the solution of (3.3.1) is represented by an infinite series of the tensor form:

$$u(\boldsymbol{x}; \boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}_0^d} c_{\boldsymbol{\alpha}}(\boldsymbol{x}) \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}), \tag{3.3.3}$$

where

$$\psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) = \psi_{\alpha_1}(\xi_1) \psi_{\alpha_2}(\xi_2) \cdots \psi_{\alpha_d}(\xi_d), \quad \alpha_i \in \mathbb{N} \cup \{0\}. \tag{3.3.4}$$

In practice we truncate this expression, e.g., up to order P:

$$u(\boldsymbol{x}; \boldsymbol{\xi}) \approx u_P^*(\boldsymbol{x}; \boldsymbol{\xi}) \stackrel{\text{def}}{=} \sum_{|\boldsymbol{\alpha}|=0}^{P} c_{\boldsymbol{\alpha}}(\boldsymbol{x}) \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}). \tag{3.3.5}$$

By selecting m different sampling points $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \cdots, \boldsymbol{\xi}_m$ we obtain m different samples of u: $\boldsymbol{u} = (u_1, u_2, \cdots, u_m)$. Hence, we rewrite (3.3.5) as: $\boldsymbol{u} \approx \boldsymbol{\Psi c}$, where $\Psi_{i,j} = \psi_{\boldsymbol{\alpha}_j}(\boldsymbol{\xi}_i)$ and $\boldsymbol{\Psi}$ is an $m \times N$ matrix with $N = \frac{(P+d)!}{P!d!}$. Now we can use the ℓ_1 minimization to seek a solution $\boldsymbol{c}$ satisfying:

$$(P_{1,\epsilon}): \qquad \min_{c} \|\boldsymbol{c}\|_1 \quad \text{subject to} \quad \|\boldsymbol{\Psi c} - \boldsymbol{u}\|_2 \leq \epsilon, \tag{3.3.6}$$

where ϵ is related to the truncation error. It is clear that the more terms we include in the expansion (3.3.5) the more accurate result we will get, which in turn means we can put smaller ϵ in (3.3.6). Notice that, if ϵ is too large we will only obtain a less accurate result while if ϵ is too small, the so called *over fitting* problem, will cause a less sparse solution, which is also not accurate. Several methods like CoSaMP [115] or Iterative Hard Thresholding [20] avoid the *a priori* knowledge of the noise level ϵ in the context of signal processing or image processing.

In order to prevent the optimization from biasing toward the non-zero entries in $\boldsymbol{c}$ whose corresponding columns in $\boldsymbol{\Psi}$ have large norms, a weighted matrix $\boldsymbol{W}$ can be included in the l_1 norm [22], thus we have

$$(P^W_{1,\epsilon}): \qquad \min_{c} \|\boldsymbol{W c}\|_1 \quad \text{subject to} \quad \|\boldsymbol{\Psi c} - \boldsymbol{u}\|_2 \leq \epsilon, \tag{3.3.7}$$

where $\boldsymbol{W}$ is a diagonal matrix whose $[j, j]$ entry is the ℓ_2 norm of the j-th column of $\boldsymbol{\Psi}$. For the original ℓ_1 minimization, $\boldsymbol{W} = \boldsymbol{I}$, i.e., the identity matrix. This can also be considered as a special case of reweighted ℓ_1 method in that we put a non-identity weight matrix in the first step and employ no additional iterations. Once the coefficients are computed, we obtain the gPC expansion of the solution (3.3.5), and then we can estimate the statistics of the solution, e.g., $\mathbb{E}(u) = c_0, Var(u) = \sum_{|\boldsymbol{\alpha}|=1}^{P} c_{\boldsymbol{\alpha}}^2$ since $\psi_{\boldsymbol{\alpha}}$ are orthonormal with respect to the distribution of $\boldsymbol{\xi}$. We summarize the above descriptions in Algorithm 6. In this section we use "a trial" to denote completing Algorithm 1 once.

Algorithm 6 Reweighted ℓ_1 minimization method for elliptic equation (3.3.1)

1: Generate m sampling points $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \cdots, \boldsymbol{\xi}_m$ based on the distribution of $\boldsymbol{\xi}$ (or based on the Chebyshev measure if Chebyshev sampling points are employed). Run the deterministic solver to solve (3.3.1) for each $\boldsymbol{\xi}_i$ to obtain m samples of the solution $u_1, u_2, \cdots, u_m$. Denote $\boldsymbol{u} = (u_1, u_2, \cdots, u_m)$ and it is the "observation" in $(P_{1,\epsilon})$. The "measurement matrix" $\boldsymbol{\Psi}$ in $(P_{1,\epsilon})$ is $\Psi_{i,j} = \psi_{\boldsymbol{\alpha}_j}(\boldsymbol{\xi}_i)$, where $\psi_{\boldsymbol{\alpha}}$ are the basis functions in (3.3.3). The size of $\boldsymbol{\Psi}$ is $m \times N$, where N is the total number of basis functions depending on P in (3.3.5).

2: Set the tolerance ϵ in $(P_{1,\epsilon})$, set the iteration count $l = 0$ and weight $w_i^{(0)} = 1, i = 1, 2, \cdots, N$. If $(P_{1,\epsilon}^W)$ instead of $(P_{1,\epsilon})$ is implemented in the first step, select appropriate w_i based on the corresponding $(P_{1,\epsilon}^W)$ algorithm. Select the maximum number of the iterations $l_{\max}$, usually 2 or 3 is enough.

3: Solve the weighted ℓ_1 minimization problem

$$\boldsymbol{c}^{(l)} = \arg\min \|\boldsymbol{W}^{(l)}\boldsymbol{c}\|_1 \quad \text{subject to} \quad \|\boldsymbol{\Psi}\boldsymbol{c} - \boldsymbol{u}\|_2 \leq \epsilon,$$

where $\boldsymbol{W}$ is a diagonal matrix with $W_{j,j}^{(l)} = w_j^{(l)}$. If the Chebyshev sampling points are employed, solve

$$\boldsymbol{c}^{(l)} = \arg\min \|\boldsymbol{W}^{(l)}\boldsymbol{c}\|_1 \quad \text{subject to} \quad \|\boldsymbol{A}\boldsymbol{\Psi}\boldsymbol{c} - \boldsymbol{A}\boldsymbol{u}\|_2 \leq \epsilon$$

instead.

4: Update the weights: for each $i = 1, 2, \cdots, N$,

$$w_i^{(l+1)} = \frac{1}{|c_i^{(l)}| + \tau}.$$

5: Terminate upon convergence or when l attains a specified maximum number of iterations $l_{\max}$. Otherwise, increment l and go to step 3.

6: Compute statistics of the solution after obtaining $\boldsymbol{c}$. For instance, $\mathbb{E}(u) = c_0$, $Var(u) = \sum_{i=1}^{N-1} c_i^2$, etc.

3.3.2 Computational algorithm

In this section, we start with a 1D problem and then discuss multi-dimensional problems, where the *dimension* here refers to the random space. More precisely, in the problems we consider below, we refer dimension to "d" in Eq. (3.3.2). We will investigate the relative error in the mean $\varepsilon_m \overset{\text{def}}{=} |c_{|\boldsymbol{\alpha}|=0} - \mathbb{E}(u)|/|\mathbb{E}(u)|$ and the standard deviation $\varepsilon_s \overset{\text{def}}{=} |(\sum_{i=1}^{P} c_{|\boldsymbol{\alpha}|=i}^2)^{1/2} - \sigma(u)|/|\sigma(u)|$ of the solution at specific spatial point. Also, we will check the L_2 error of the numerical solution $\varepsilon_u = \|\tilde{u}_P^* - u\|_2/\|u\|_2$ at specific spatial point, where $\tilde{u}_P^*$ is the truncated gPC expansion with coefficients recovered by our method. The integral is calculated by the sparse grids method. All the ℓ_1 minimizations in this section are achieved by the `SPGL1` package [146].

One-dimensional problem

We consider the problem:

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}x}\left(a(x;\xi)\frac{\mathrm{d}}{\mathrm{d}x}u(x,\xi)\right) = -1, \quad x \in (0,1), \\ &u(0) = u(1) = 0. \end{aligned} \tag{3.3.8}$$

where ξ is uniformly distributed on $[-1,1]$. We set $a(x;\xi) = a(\xi) = 1 + 0.5\xi$. The exact solution for this problem is

$$u(x;\xi) = \frac{x(1-x)}{2a(\xi)} = \frac{x(1-x)}{2+\xi}.$$

We investigate the solution at $x = 0.5$ and omit the symbol x unless confusion arises. The expectation of the solution is $\mathbb{E}(u) = \frac{1}{8}\ln 3$ and the standard deviation is $\sigma(u) = \sqrt{\frac{1}{48} - \frac{1}{64}(\ln 3)^2}$. The coefficients in the gPC expansion

$$u(\xi) = \sum_{j=0}^{\infty} c_j L_j(\xi) \tag{3.3.9}$$

can be computed analytically:

$$c_j = \int_{-1}^{1} u(\xi) L_j(\xi) \mathrm{d}\nu(\xi),$$

where L_j is j-th order *normalized* Legendre polynomial. Figure 3.2 presents the absolute values of c_j while those below 10^{-14} are neglected. We can see that the solution is *nearly sparse* as only eight coefficients are larger than 10^{-5}, which means that very few basis make considerable contribution to the solution. We also point out that it does not matter if we change the order of the basis since the compressive sensing method will detect the pattern of sparsity automatically. Moreover, the threshold 10^{-5} used here is only for demonstration purposes to present the sparsity of the coefficients. For different problems, the threshold can be different.

Uniformly distributed sampling points

We first employ the sampling points ξ_i based on uniform distribution on $[-1,1]$ (will be called "uniform points"). The purpose of this test is to demonstrate the benefit of using reweighted

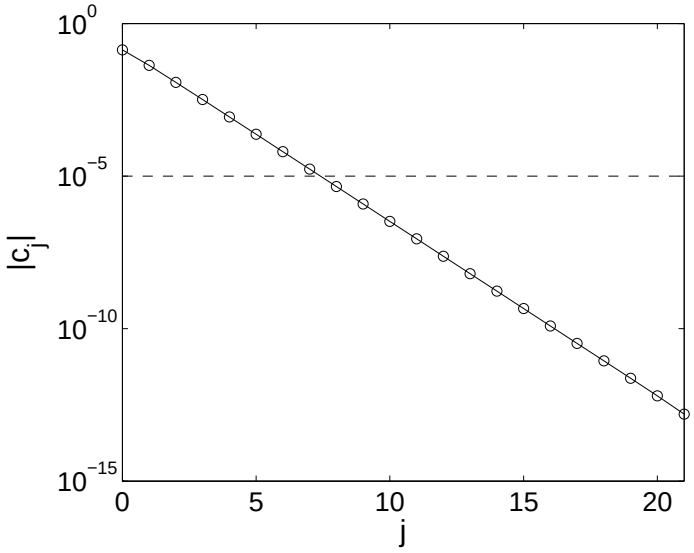


Figure 3.2: 1D example : absolute values of 22 coefficients with the largest magnitude in the gPC expansion for equation (3.3.8) at $x = 0.5$ with $a = 1 + 0.5\xi$ and $\xi \sim U[-1, 1]$. Only 8 of these absolute values are larger than 10^{-5}.

iterations. In our test, ϵ and τ in the reweighted procedure are set empirically. A better choice of ϵ can be obtained by, e.g., cross-validation, which we adopt in the multi-dimensional tests below. We truncate the solution of (3.3.8) with $P = 80$, hence, $N = P + 1 = 81$, and try to recover the coefficients of the following gPC expansion:

$$u(\xi) \approx u^*_{80} = \sum_{j=0}^{80} c_j L_j(\xi). \tag{3.3.10}$$

In this test the truncation error with $P = 80$ is negligible, hence the L_2 error of the solution can be reflected by the ℓ_2 error of the gPC coefficients. More precisely, since $\|u - u^*_P\|_2$ is negligible, $\|u - \tilde{u}^*_P\|_2$ is the same as $\|u^*_P - \tilde{u}^*_P\|_2$ which equals $\left(\sum_{j=0}^{P} |\tilde{c}_j - c_j|^2\right)^{1/2}$ (ℓ_2 error of gPC coefficients) since the gPC basis is orthonormal. Here $\tilde{u}^*_P$ is the approximation of u^*_P with coefficients $\tilde{\boldsymbol{c}}$ obtained by our method. We present the ℓ_2 error by the gPC coefficients by using 10, 15 and 20 uniform points with and without reweighted iterations in Figure 3.3. In all the tests we set $l_{\max}$ to be 2 in Algorithm 6. In order to obtain the histograms of the recovery error, we run 10,000 trials for each

test and set $\epsilon = 10^{-4}, \tau = 8 \times 10^{-2}$. From Figure 3.3 (a), (b), (c) we observe that without using the reweighted iterations, 20 uniform points yield the best recovery in that bins denoting smaller error are higher as the number of samples increases while bins denoting larger error are lower. When the reweighted iterations are employed, i.e., (g), (h), (i), the recovery is much better, which can be observed by comparing (a) and (g), (b) and (h), (c) and (i), respectively. Also, comparing (b) with (g), we observe that with the reweighted iterations, using 10 uniform points can render a comparable result than using 15 uniform points without reweighting. Similarly, comparing (c) with (h), we see that by using 15 uniform points with reweighted iterations we obtain much better results than using 20 uniform points without reweighting.

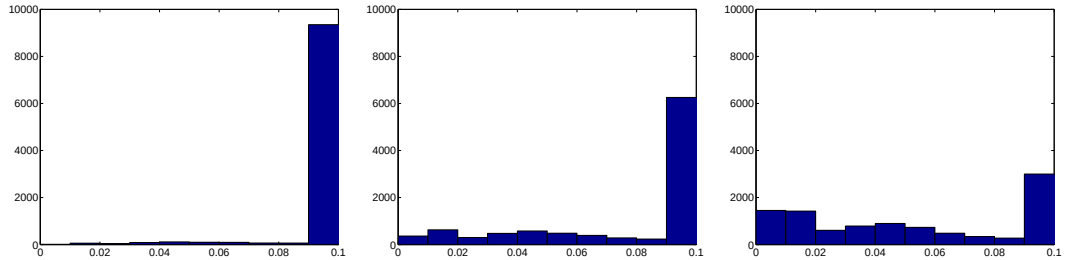
Figure 3.4 presents the improvement from the reweighted iterations by considering the ℓ_2 reconstruction error: $\|\boldsymbol{c} - \boldsymbol{c}^{(2)}\|_2 / \|\boldsymbol{c} - \boldsymbol{c}^{(0)}\|_2$, where $\boldsymbol{c}$ is the vector of exact coefficients. Notice that the upper bound of $\|\boldsymbol{c} - \boldsymbol{c}^{(0)}\|_2$ is given in Theorem 3.2.3 and the upper bound of $\|\boldsymbol{c} - \boldsymbol{c}^{(2)}\|_2$ is close to the one in Theorem 3.1.2. Since we set $l_{\max} = 2$, i.e., we compute $\boldsymbol{c}^{(0)}, \boldsymbol{c}^{(1)}, \boldsymbol{c}^{(2)}$, where $\boldsymbol{c}^{(0)}$ is the results by ℓ_1 minimization; we only use two additional ℓ_1 optimizations. We observe that 56.2% of the 10-sample tests, 72.0% of the 15-sample tests and 68.9% of the 20-sample tests show a reduction of l_2 reconstruction error up to 50% or more, i.e., $\|\boldsymbol{c} - \boldsymbol{c}^{(2)}\|_2 / \|\boldsymbol{c} - \boldsymbol{c}^{(0)}\|_2 \leq 0.5$. If we are more aggressive to check the percentage of reduction of l_2 reconstruction error up to 90% or more, i.e., $\|\boldsymbol{c} - \boldsymbol{c}^{(2)}\|_2 / \|\boldsymbol{c} - \boldsymbol{c}^{(0)}\|_2 \leq 0.1$, the answer is 12.4% for the 10-sample tests, 22.3% for the 15-sample tests and 17.9% for the 20-sample tests. Notice that with the change of the number of the sampling points m, the property of the measurement matrix $\boldsymbol{\Psi}$ changes as well. More precisely, the RIP condition ($\sigma_{2s} < \sqrt{2} - 1$) in Theorems 3.2.3 and 3.1.2 will be satisfied for different s. (If m is too small, then this condition may not be satisfied for any $s \leq N/2$.) Hence, the error bound in Theorems 3.2.3 and 3.1.2 will be different, and the ratios of improvement are different as well. Finally, we also point out that in a few tests (less than 0.4%), the reweighted iterations show worse results.

The observations for the 1-D test are consistent with the conclusions of the ℓ_1 minimization and

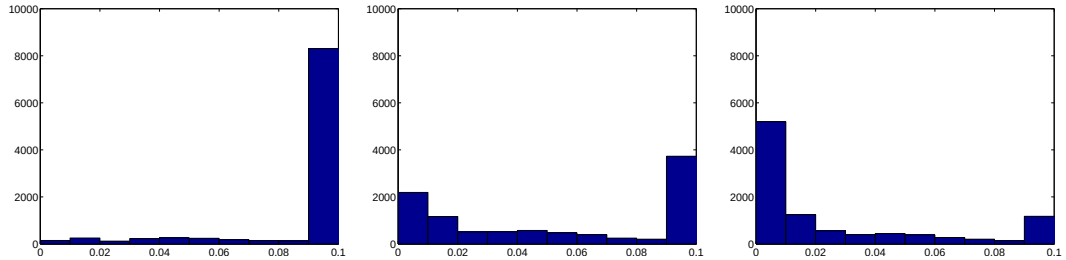
reweighted ℓ_1 minimization in [29] for the deterministic cases in that: (1) more sampling points will provide a higher probability of more accurate recovery; (2) reweighted iterations can enhance the sparsity of the solution, which in turn can improve the accuracy but there are also very small chances that the result is worse than ℓ_1 minimization.

Chebyshev measure based sampling points

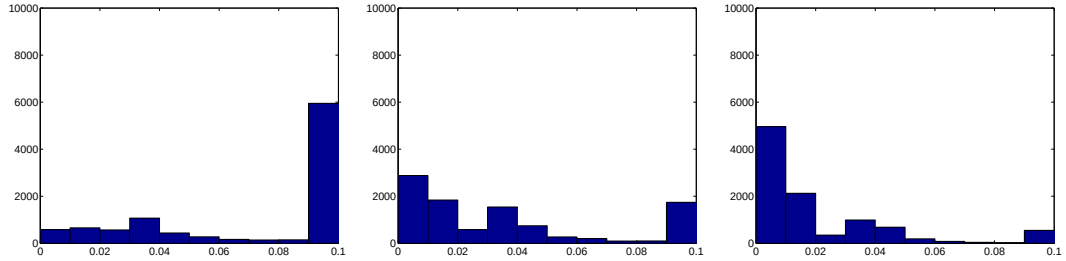
Now we use the sampling points based on Chebyshev measure (will be called "Chebyshev points") to repeat the experiments in the last subsection. Figure 3.3 also presents the relative error in the coefficients by using 10, 15 and 20 Chebyshev points with and without reweighted iterations. Comparing the first two rows in this figure, where we only use the ℓ_1 minimization and keep increasing the number of sampling points, we observe that without reweighted iterations Chebyshev points render better recovery than uniform points. This is consistent with Theorem 3.1.5, which claims that by using Chebyshev points we can reduce the number of sampling points to obtain a measurement matrix satisfying the RIP condition. Comparing the third and the fourth row, where reweighted iterations are employed, we observe that when the Chebyshev points are used, the reweighted iterations still improve the accuracy. We can also fix the number of the sampling points and compare the recovery accuracy with different sampling strategies. Let us consider the middle column for example, where the number of sampling points is fixed to be 15. We can observe from the pattern of the histogram that for this test case, reweighted iterations provide greater enhancement than using the Chebyshev points as (h) (where uniform points and reweighted iterations are employed) shows better results than (e) (where Chebyshev points and ℓ_1 minimization are employed) and the combination of these two ideas yields remarkable improvement as shown in (k). This phenomenon implies that with the same number of sampling points, Chebyshev points allows a larger s to satisfy the RIP condition (Theorem 3.1.5) and reweighted iterations further decreases the upper bound of the recovering error of $\boldsymbol{c}$ (Theorem 3.1.2). Moreover, the result in (k) (15 Chebyshev points with reweighted ℓ_1) is much better than that in (c) (20 uniform points with ℓ_1) and it is also better than the results in (f) (20 Chebyshev points with ℓ_1) and (i) (20 unform



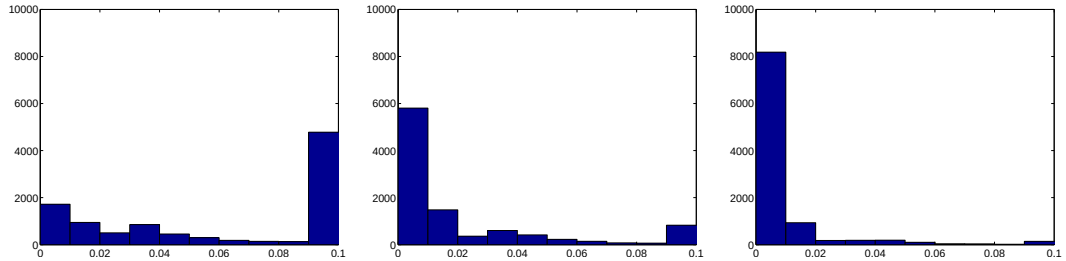
(a) ℓ_1 unif ($m = 10$) (b) ℓ_1 unif ($m = 15$) (c) ℓ_1 unif ($m = 20$)



(d) ℓ_1 Cheb ($m = 10$) (e) ℓ_1 Cheb ($m = 15$) (f) ℓ_1 Cheb ($m = 20$)

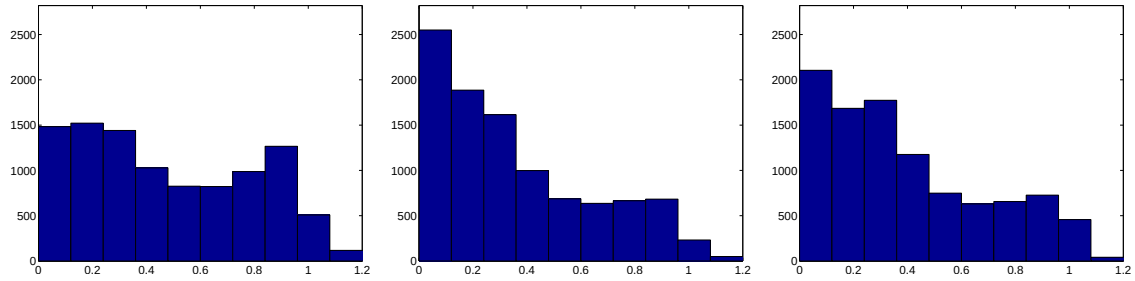


(g) rw ℓ_1 unif ($m = 10$) (h) rw ℓ_1 unif ($m = 15$) (i) rw ℓ_1 unif ($m = 20$)



(j) rw ℓ_1 Cheb ($m = 10$) (k) rw ℓ_1 Cheb ($m = 15$) (l) rw ℓ_1 Cheb ($m = 20$)

Figure 3.3: 1D example: comparison of the relative error of the ℓ_2 norm of the gPC coefficients ($\|\boldsymbol{c}-\tilde{\boldsymbol{c}})\|_2/\|\boldsymbol{c}\|_2$, where $\boldsymbol{c}$ is the exact solution and $\tilde{\boldsymbol{c}}$ is the numerical solution) by using uniform points ("unif") and Chebyshev points ("Cheb") with ℓ_1 minimization and reweighted ℓ_1 minimization ("rw"). Number of basis $N = 81$, number of samples $m = 10$ (left column), $m = 15$ (middle column), $m = 20$ (right column). Total number of trials is 10,000. $l_{\max} = 2, \epsilon = 10^{-4}, \tau = 8 \times 10^{-2}$. x-axis presents the range of the relative error and the histograms demonstrate the number of trials with the relative error in specific ranges.



(a) $m = 10$ (b) $m = 15$ (c) $m = 20$

Figure 3.4: 1D example : improvement of the reweighted iterations with uniform points by checking the ℓ_2 error: $\|\boldsymbol{c} - \boldsymbol{c}^{(2)}\|_2 / \|\boldsymbol{c} - \boldsymbol{c}^{(0)}\|_2$, where $\boldsymbol{c}$ is the vector of the exact coefficients. Total number of trials is 10,000. Number of basis $N = 81$, number of samples, $m = 10$ in (a), $m = 15$ in (b), $m = 20$ in (c). $l_{\max} = 2, \epsilon = 10^{-4}, \tau = 8 \times 10^{-2}$. x-axis presents the range of the improvement and the histograms demonstrate the number of trials with the improvement in specific ranges.

points with reweighted ℓ_1). Therefore, the combination of Chebyshev points and the reweighted iterations has the potential to perform good recovery with fewer sampling points compared with the ℓ_1 minimization method.

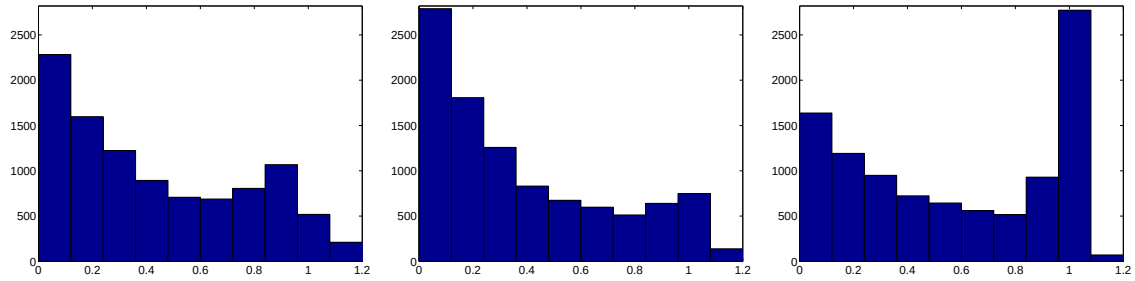
Since the inherent combinatorial nature of the RIP makes it impossible to directly compute the RIP constant of a matrix [16], we present the mutual coherences of the measurement matrices for different cases in Table 3.1 for comparison. We can observe that the mutual coherence is smaller when the Chebyshev points are used.

Table 3.1: Mutual coherence μ of the measurement matrices of 1D tests. Means and standard deviations ("s.d.") of μ in tests with diferent m are presented.

	$m = 10$		$m = 15$		$m = 20$	
	mean	s.d.	mean	s.d.	mean	s.d.
uniform points	0.9282	0.0293	0.8467	0.0461	0.7785	0.0546
Chebyshev points	0.9166	0.0315	0.8193	0.0479	0.7355	0.0536

Moreover, the improvement of the accuracy of recovering the coefficient by using reweighted iterations with Chebyshev points is presented in Figure 3.5. We observe that similar to the conclusion for uniformly distributed ξ, reweighted iterations enhance the accuracy of recovery. Also, we notice that more than 1/4 of the tests with 20 Chebyshev points show almost no improvement, which is different from the result in Figure 3.4(c), where the uniformly distributed sampling points are employed. This difference means that 20 is a relative large number for the Chebyshev points.

It is clear that if m is very large, e.g., $m = N$ we will obtain a very accurate recovery by ℓ_1 minimization, hence, reweighted iterations will provide very little improvement.



(a) $m = 10$ (b) $m = 15$ (c) $m = 20$

Figure 3.5: 1D example: improvement of the reweighted iterations with Chebyshev points by checking the ℓ_2 error $\|\boldsymbol{c} - \boldsymbol{c}^{(2)}\|_2/\|\boldsymbol{c} - \boldsymbol{c}^{(0)}\|_2$, where $\boldsymbol{c}$ is the vector of the exact coefficients. Total number of trials is 10,000. Number of basis $N = 81$, number of samples, $m = 10$ in (a), $m = 15$ in (b), $m = 20$ in (c). $l_{\max} = 2, \epsilon = 10^{-4}, \tau = 8 \times 10^{-2}$. x-axis presents the range of the improvement and the histograms demonstrate the number of trials with the improvement in specific ranges.

To conclude, in the numerical tests of the 1-D problem, we firstly verify the benefit of reweighted iterations and Chebyshev points for the test problem as the results are consistent with those in the corresponding references for deterministic cases, e.g., [29, 123]. We then obtain that the reweighted iterations can make more contribution in the improvement. Finally, the combination of these two techniques can provide remarkable enhancement of the accuracy in the recovery. Notice that this 1-D problem is for demonstration purposes, and it can be solved more accurately by other methods, e.g., gPC with Galerkin projection or a sparse grid method with less computational cost. In the next subsection we will discuss high-dimensional cases.

3.3.3 Multi-dimensional problems

We consider the following elliptic equation which is 1-D in physical space but multi-D in random space:

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}x}\left(a(x;\boldsymbol{\xi})\frac{\mathrm{d}}{\mathrm{d}x}u(x;\boldsymbol{\xi})\right) = -1, \quad x \in (0,1), \\ &u(0) = u(1) = 0. \end{aligned} \tag{3.3.11}$$

where the stochastic diffusion coefficients $a(x;\boldsymbol{\xi})$ is given by the KL expansion in Eq. (3.3.2). Here, $\{\lambda_i\}_{i=1}^d$ and $\{\phi_i(x)\}_{i=1}^d$ are, respectively, the d largest eigenvalues and corresponding eigenfunctions of the Gaussian covariance kernel

$$C_{aa}(x_1,x_2)=\exp\left[-\frac{(x_1-x_2)^2}{l_c^2}\right], \tag{3.3.12}$$

in which l_c is the correlation length of $a(x;\boldsymbol{\xi})$ that dictates the decay of the spectrum of C_{aa}. The random variables $\{\xi_i\}_{i=1}^d$ are assumed to be independent and uniformly distributed on $[-1,1]$. The coefficient σ_a controls the variability of $a(x;\boldsymbol{\xi})$ and we consider two cases here: $(l_c,d)=(1/5,14),(l_c,d)=(1/14,40)$ and set $\bar{a}=0.1,\sigma_a=0.03;\bar{a}=0.1,\sigma_a=0.021$, respectively. Therefore, the requirements for coefficient $a(\boldsymbol{x};\boldsymbol{\xi})$ in Section 3.3.1 are satisfied. Both the sparse grid method with Clenshaw-Curtis abscissas and Monte Carlo method are tested. For each sampling point $\boldsymbol{\xi}_i$ in random space, the deterministic second-order ODE is solved by integrating Eq.(3.3.11) to obtain

$$u'(x)=\frac{a(0)u'(0)-x}{a(x)}. \tag{3.3.13}$$

Again, integrating Eq.(3.3.13) and letting $M=a(0)u'(0)$ we have

$$u(x)=u(0)+\int_0^x\frac{M-s}{a(s)}\mathrm{d}s=\int_0^x\frac{M-s}{a(s)}\mathrm{d}s. \tag{3.3.14}$$

By imposing the boundary condition $u(1)=0$, we can compute M. In order to compute the integrals in Eq. (3.3.14), we split the domain $[0,1]$ into 2000 equi-distance subintervals and use 3 Gaussian quadratures in each subinterval. The reference solution is obtained by level 7 sparse grids method for $d=14$ and level 4 sparse grids method for $d=40$ since this accuracy is sufficient for the demonstrations in this section. For $d=14$ we set $P=3$, i.e., gPC basis up to 3rd-order are employed and the total number of basis is $N=680$. For $d=40$ we set $P=2$, i.e., gPC basis up to 2rd-order are employed and the total number of basis is $N=861$.

Effect of ϵ

We first study the effect of ϵ in $(P_{1,\epsilon})$. It dictates the distance between the exact solution and the approximated one through the projection to the pre-selected basis. In practice, we truncate the right-hand-side of Eq. (3.3.3) to obtain an approximation, therefore there is always an error since the basis is not complete. As pointed out in [39], ideally, we would like to choose $\epsilon \approx \|\boldsymbol{\Psi c} - \boldsymbol{u}\|_2$. Figure 3.6 presents the error in the mean and the standard deviation of the approximated solution with the coefficients recovered from the ℓ_1 minimization, i.e., $(P_{1,\epsilon})$ for $d = 14$. Here ϵ varies from 10^{-4} to 10^{-1} with a fixed ratio and three trials (denoted by different colors) with 120 uniform points each. These three trials are randomly selected from 1000 such trials to demonstrate the effect of ϵ. It is clear that, as ϵ decreases, the error in both mean and standard deviation decreases considerably in the range of $\epsilon \in [10^{-2}, 10^{-1}]$. However, when ϵ continues to decrease, the improvement is very little or there may not be an improvement. This is because ϵ has reached the level of truncation error, and therefore the only way to improve the accuracy in the current setting is to include more terms in the gPC expansion of the solution or increase m. This conclusion, which is an intrinsic property of ℓ_1 minimization method, holds for general cases as mentioned in Section 3.

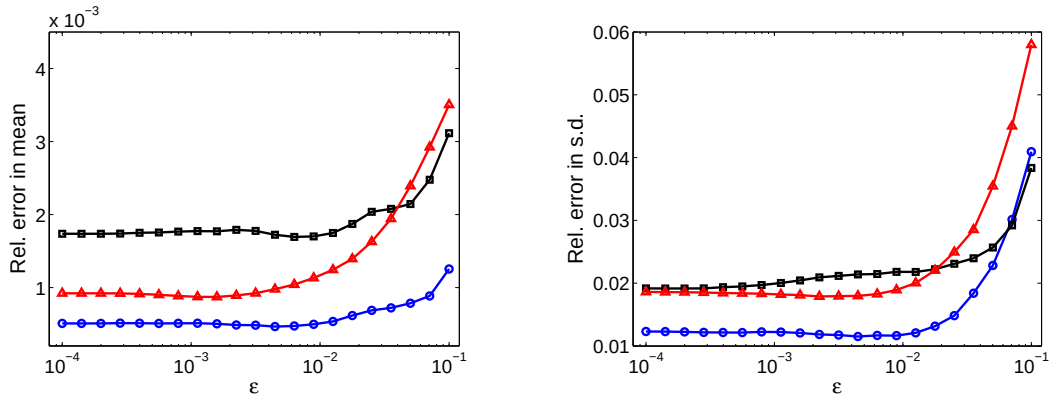


Figure 3.6: 14D example: relative error in the mean (left) and the standard deviation (right) of the approximated solution with the coefficients recovered from ℓ_1 minimization $(P_{1,\epsilon})$. Three trials with 120 uniform points each are presented by different symbols; ϵ varies from 10^{-4} to 10^{-1}. $\bar{a} = 0.1, \sigma_a = 0.03, d = 14$.

These tests imply that for the current problem setting, the selection of ϵ plays an essential role

in a specific range before it descends to the vicinity of the truncation error while very little or even no improvement can be achieved by continuing decreasing ϵ when it is already very close to the truncation error. Moreover, weighted ℓ_1 minimization ($P^W_{1,\epsilon}$) improves the results only a little or has no benefit (not presented here). Therefore, in order to considerably reduce the error further we need other techniques. In the rest part of this section, we will only use ($P_{1,\epsilon}$) in the first step of Algorithm 6.

Reweighted ℓ_1 minimization

As shown for the 1-D test case, reweighted ℓ_1 minimization has the potential of further increasing the sparsity of the solution and therefore reducing the recovery error. Figure 3.7 presents the results of repeating the same trials as in the last subsection but with reweighted ℓ_1 minimizations for the uniform points with $d = 14$. Different colors denote different trials. Dash lines are the result by the ℓ_1 minimization $\epsilon = 10^{-4}$. For the mean, the reweighted ℓ_1 method reduces the relative error by 95% (blue), 70% (black) and 71% (red). For the standard deviation, three different trials show reduction of the error to be 74% (blue), 49% (black), 43% (red). Similar results for $d = 40$ are presented in Figure 3.8. Notice that, ϵ, τ are chosen randomly here. We will employ cross-validation method as in [39] to select parameters in the next subsection. To our knowledge, so far the best theoretical result to estimate the error bound is Theorem 3.1.2 and its related lemmas and theorems in [114].

Quantitative comparisons of results in Figure 3.7 and Figure 3.8 are presented in Table 3.2 and Table 3.3 for $d = 14$ and $d = 40$, respectively. We can observe that for some trials, the improvement is dramatic, e.g., for trial 1 of $d = 14$ case, the error for the mean is reduced by more than 90% while the error for the standard deviation is reduced by 75%. For some trials, the improvement is not that impressive with the reduction of error ranging from 30% to 50%.

Next, we also employ the cross-validation method to obtain a relatively better choice of ϵ (details can be found in 3.1.4). We run 1000 trials of each experiment and the histograms of the L_2 error of the solution at $x = 0.5$ are shown in Figures 3.9 and 3.10. It is clear that the reweighted ℓ_1

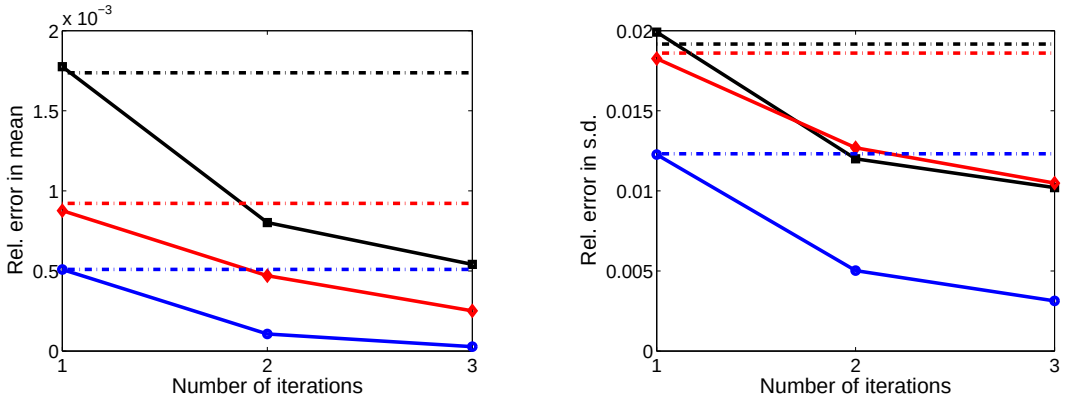


Figure 3.7: 14D example: relative error in the mean and the standard deviation of the approximated solution with the coefficients recovered from reweighted ℓ_1 minimization. Three different trials with 120 uniform points each are tested. Different colors denote different trials, solid lines are the results by the reweighted ℓ_1 minimization and the dash lines are the result with $\epsilon = 10^{-4}$ in ℓ_1 minimization as in Figure 3.6. Here $\epsilon = 10^{-3}, \tau = 10^{-3}$ for all the tests and $\bar{a} = 0.1, \sigma_a = 0.03, d = 14$.

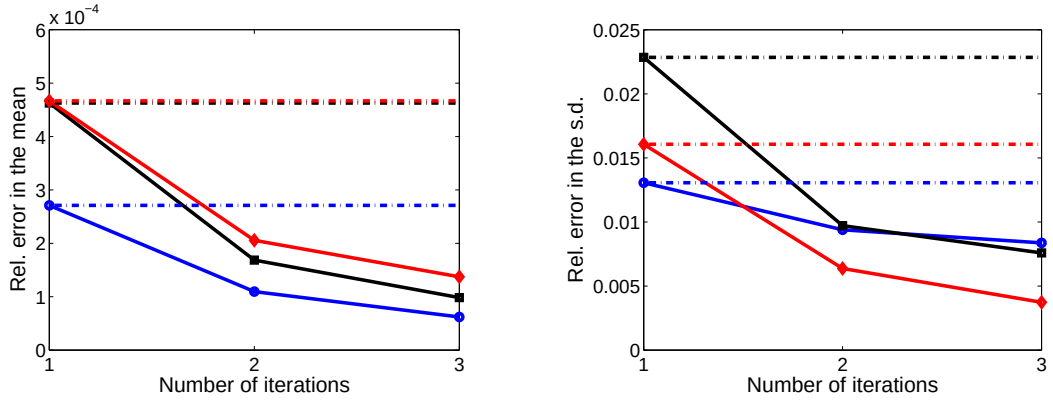


Figure 3.8: 40D example: relative error in the mean and the standard deviation of the approximated solution with the coefficients recovered from reweighted ℓ_1 minimization. Three different trials with 200 uniform points each are tested. Different colors denote different data sets, solid lines are the results by the reweighted ℓ_1 minimization while the dash lines are the result with $\epsilon = 10^{-4}$ in ℓ_1 minimization. Here $\epsilon = 5 \times 10^{-3}, \tau = 10^{-3}$ for all the tests and $\bar{a} = 0.1, \sigma_a = 0.021, d = 40$.

Table 3.2: 14D example: comparison of ℓ_1 and reweighted ℓ_1 minimization with 3 iterations ($l_{\max} = 2$) for the three trials in Figure 3.7. 120 uniform points are employed in each trial. $\bar{a} = 0.1, \sigma_a = 0.03, d = 14, \epsilon = 10^{-3}, \tau = 10^{-3}$.

	Relative error in the mean ε_m		Relative error in the s.d. ε_s	
	ℓ_1	rw ℓ_1	ℓ_1	rw ℓ_1
trial 1	$5.0925e-4$	$2.7083e-5$	$1.2268e-2$	$3.1292e-3$
trial 2	$1.7753e-3$	$5.4107e-4$	$1.9909e-2$	$1.0200e-2$
trial 3	$8.7641e-4$	$2.5095e-4$	$1.8257e-2$	$1.0487e-2$

Table 3.3: 40D example: comparison of ℓ_1 and reweighted ℓ_1 minimization with 3 iterations ($l_{\max} = 2$) for the three trials in Figure 3.8. 200 uniform points are employed in each trial. $\bar{a} = 0.1, \sigma_a = 0.021, d = 40, \epsilon = 10^{-3}, \tau = 10^{-3}$.

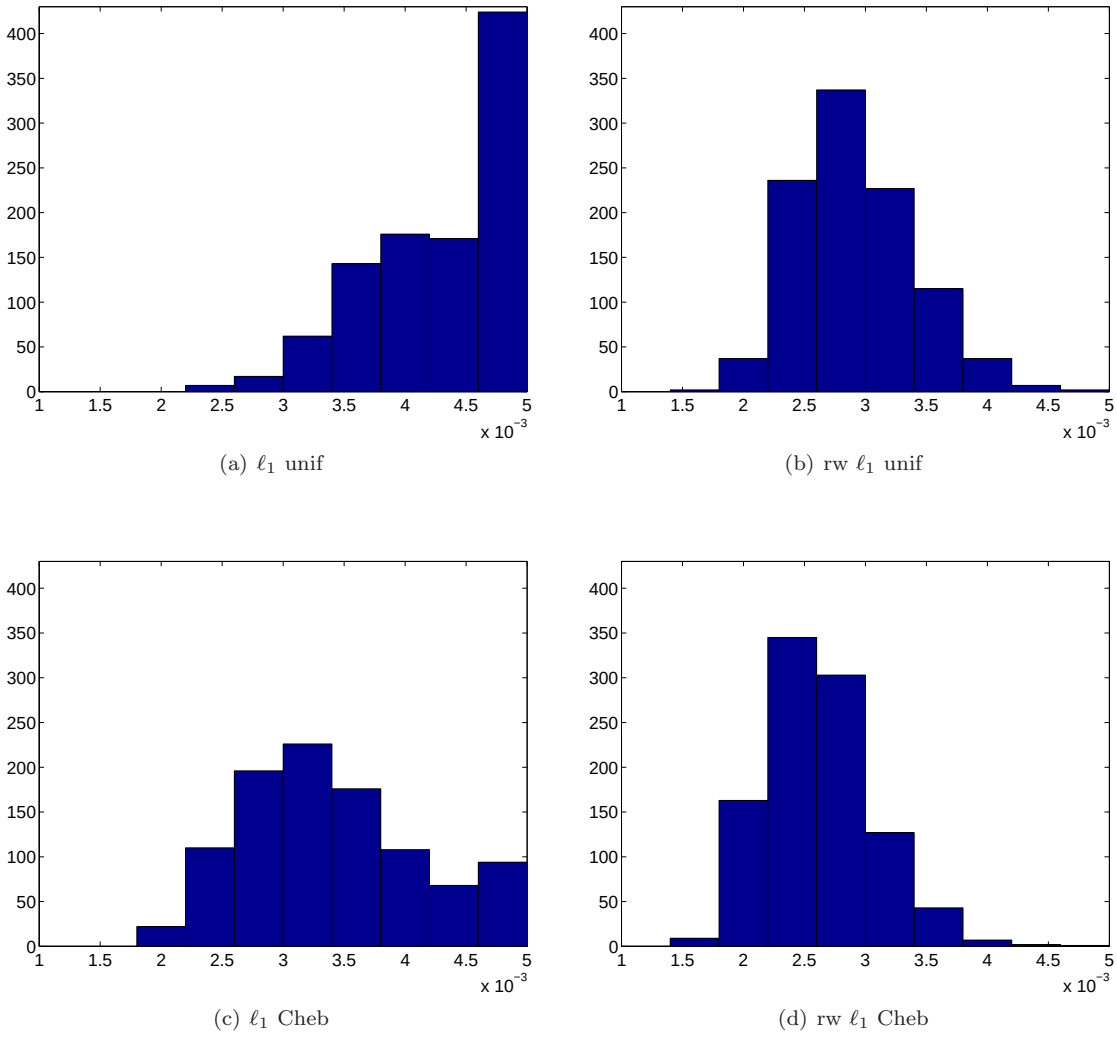
	Relative error in the mean ε_m		Relative error in the s.d. ε_s	
	ℓ_1	rw ℓ_1	ℓ_1	rw ℓ_1
trial 1	$2.7113e-4$	$6.1976e-5$	$1.3064e-2$	$8.3692e-3$
trial 2	$4.6284e-4$	$9.8414e-4$	$2.2842e-2$	$7.5808e-3$
trial 3	$4.6676e-4$	$1.3727e-4$	$1.6061e-2$	$3.7268e-3$

minimization reduces the L_2 error of the solution when uniform points or Chebyshev points are employed. A quantitative comparison of the error of statistics at $x = 0.5$ is shown in Table 3.4 (for $d = 14$) and Table 3.5 (for $d = 40$). As a comparison, e.g., for $d = 14$, the error by level 1 sparse grids method (29 samples) is $9.4387e-4$ for the mean and $5.0068e-2$ for the standard deviation while the corresponding data for level 2 sparse grids method (421 samples) is $3.2826e-5$ and $1.4532e-3$. We can observe that, for the estimate of the mean (i.e., c_0), neither methods guarantee that the accuracy is better than level 1 sparse grids method and few trials reach the accuracy of level 2 sparse grids method. However, for higher requirement of the accuracy, reweighted ℓ_1 minimization performs well. For example, in Table 3.4, 75% tests with reweighted ℓ_1 method show an error in the mean less than $5e-4$, which is about half of the error by the level 1 sparse grid method. Without the reweighted iterations, this percentage is only 7%. Similar conclusions can be drawn for the comparisons of the error of the standard deviation.

Table 3.4: 14D example: number of trials out of 1000 with corresponding error at $x = 0.5$ for different methods for $d = 14$. 120 uniform points are employed in each trial. $l_{\max} = 2, \bar{a} = 0.1, \sigma_a = 0.03, d = 14, \tau = 10^{-3}$. As a comparison, the error of level 1 sparse grids method (29 samples) $9.4387e-4$ for the mean and $5.0068e-2$ for the standard deviation while the corresponding data of level 2 sparse grids method (421 samples) is $3.2826e-5$ and $1.4532e-3$.

Relative error in the mean ε_m			Relative error in the s.d. ε_s		
	ℓ_1	rw ℓ_1		ℓ_1	rw ℓ_1
$< 9.4e-4$	380	975	$< 5e-2$	1000	1000
$< 5e-4$	71	747	$< 1e-2$	167	850
$< 3.3e-5$	0	42	$< 1.5e-3$	2	60

Moreover, we consider the global error of statistics (mean and the standard deviation) over the physical domain $x \in [0, 1]$ by selecting equi-distance points $x_i = 0.05 \times i, i = 1, \cdots, 19$



(a) ℓ_1 unif

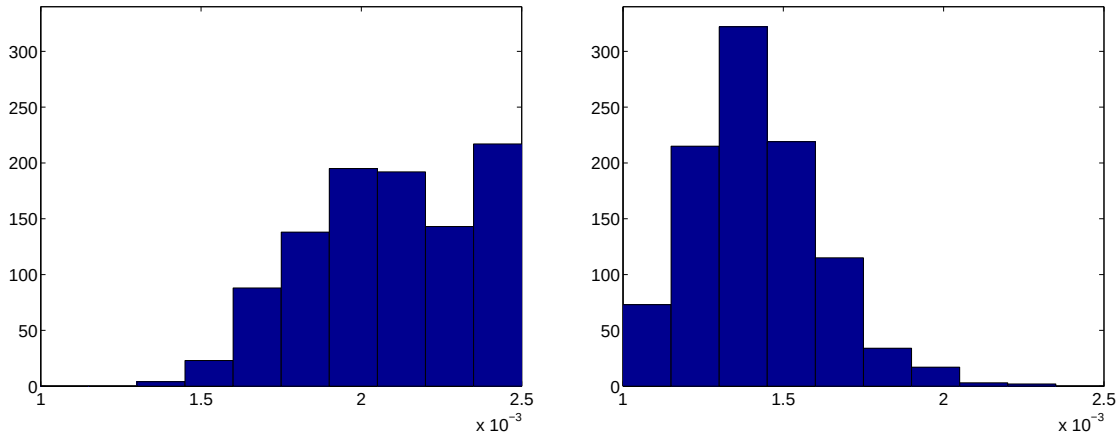
(b) rw ℓ_1 unif

(c) ℓ_1 Cheb

(d) rw ℓ_1 Cheb

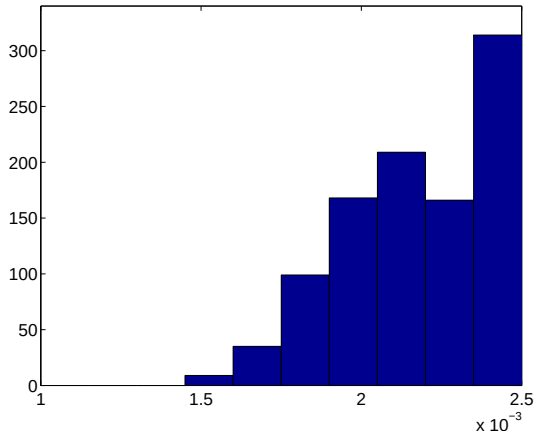
Figure 3.9: 14D example: L_2 error of the solution at $x = 0.5$. x-axis is the range of the error and the histograms demonstrate the number of trials with the error in specific ranges. In each trial, 120 sampling points are employed and $1,000$ trials are tested to present the histograms. "ℓ_1" denotes the standard ℓ_1 minimization; "rw ℓ_1" denotes reweighted ℓ_1 minimization; "unif" denotes that only uniform points are used; "Cheb" denotes that Chebshev points are employed in the sampling points. In (a) and (b) the sampling points are uniform points; in (c) and (d) the first d' dimension of the sampling points are Chebyshev and the remainings are uniform. ϵ is obtained by cross-validation and $l_{\max} = 2, \bar{a} = 0.1, \sigma_a = 0.03, d = 14, d' = 6, \tau = 10^{-3}$.

and computing the error of statistics at each point. Specifically, we compute the l_2 error of the statistics at these points. For example, to investigate the global error of the mean, we compute $\left(\sum_{i=1}^{19} |\mathbb{E}(u)|_{x=x_i} - \mathbb{E}(\tilde{u}_P^*)|_{x=x_i}|^2\right)^{1/2} / \left(\sum_{i=1}^{19} |\mathbb{E}(u)|_{x=x_i}|^2\right)^{1/2}$, where u is the reference solution and $\tilde{u}_P^*$ is the approximated solution. The results are presented in Figure 3.11 (for $d = 14$) and

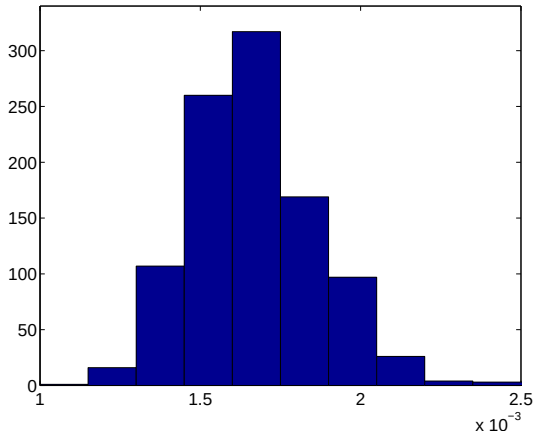


(a) ℓ_1 unif

(b) rw ℓ_1 unif



(c) ℓ_1 Cheb



(d) rw ℓ_1 Cheb

Figure 3.10: 40D example: L_2 error of the solution at $x = 0.5$. x-axis is the range of the error and the histograms demonstrate the number of trials with the error in specific ranges. In each trial, 120 sampling points are employed and $1,000$ trials are tested to present the histograms. "ℓ_1 denotes the standard ℓ_1 minimization; "rw" denotes reweighted ℓ_1 minimization; "unif" denotes that only uniform points are used; "Cheb" denotes that Chebshev points are employed in the sampling points. In (a) and (b) the sampling points are uniform points; in (c) and (d) the first d' dimension of the sampling points are Chebyshev and the remainings are uniform. ϵ is obtained by cross-validation and $l_{\max} = 2, \bar{a} = 0.1, \sigma_a = 0.021, d = 40, d' = 3, \tau = 10^{-3}$.

Figure 3.12 (for $d = 40$). We observe that the reweighted iterations improve the results when uniform points are employed. We list in Table 3.6 the 95% confidence interval of the mean of the global error of statistics by Monte Carlo method (without post processing) and by reweighted ℓ_1 minimization. It is clear that the reweighted ℓ_1 minimization reduces the error of the mean by about 85% for both $d = 14$ and $d = 40$. It reduces the error of the standard deviation by about

Table 3.5: 40D example: number of trials out of 1000 with corresponding error at $x = 0.5$ for different methods for $d = 40$. 200 uniform points are employed in each trial. $l_{\max} = 2, \bar{a} = 0.1, \sigma_a = 0.021, d = 40, \tau = 10^{-3}$. As a comparison, the error of level 1 sparse grids method (81 samples) $4.3655e{-}4$ for the mean and $6.4767e{-}2$ for the standard deviation while the corresponding data of level 2 sparse grids method (3281 samples) is $1.9131e-5$ and $3.6733e-3$.

Relative error in the mean ε_m			Relative error in the s.d. ε_s		
	ℓ_1	rw ℓ_1		ℓ_1	rw ℓ_1
$< 4.4e-4$	682	993	$< 6.5e-2$	1000	1000
$< 1.0e-4$	63	463	$< 1.0e-2$	287	983
$< 1.9e-5$	16	87	$< 3.7e-3$	13	534

65% for $d = 14$ and 70% for $d = 40$. The performance is slightly different for the two cases. There are two main reasons that affect the efficiency of the method in our test cases: (1) for these two different cases, the "sparsity" of the coefficients is different; (2) we expand the solution of $d = 14$ case with $P = 3$ while the solution of $d = 40$ case with $P = 2$ due to computational limitations, which is not adequate for higher accuracy. Since the standard deviation depends on the coefficients of all the Legendre polynomials except for c_0 (which is the mean), the accuracy not only depends on the performance of ℓ_1 minimization but also relies on the gPC expansion of the solution.

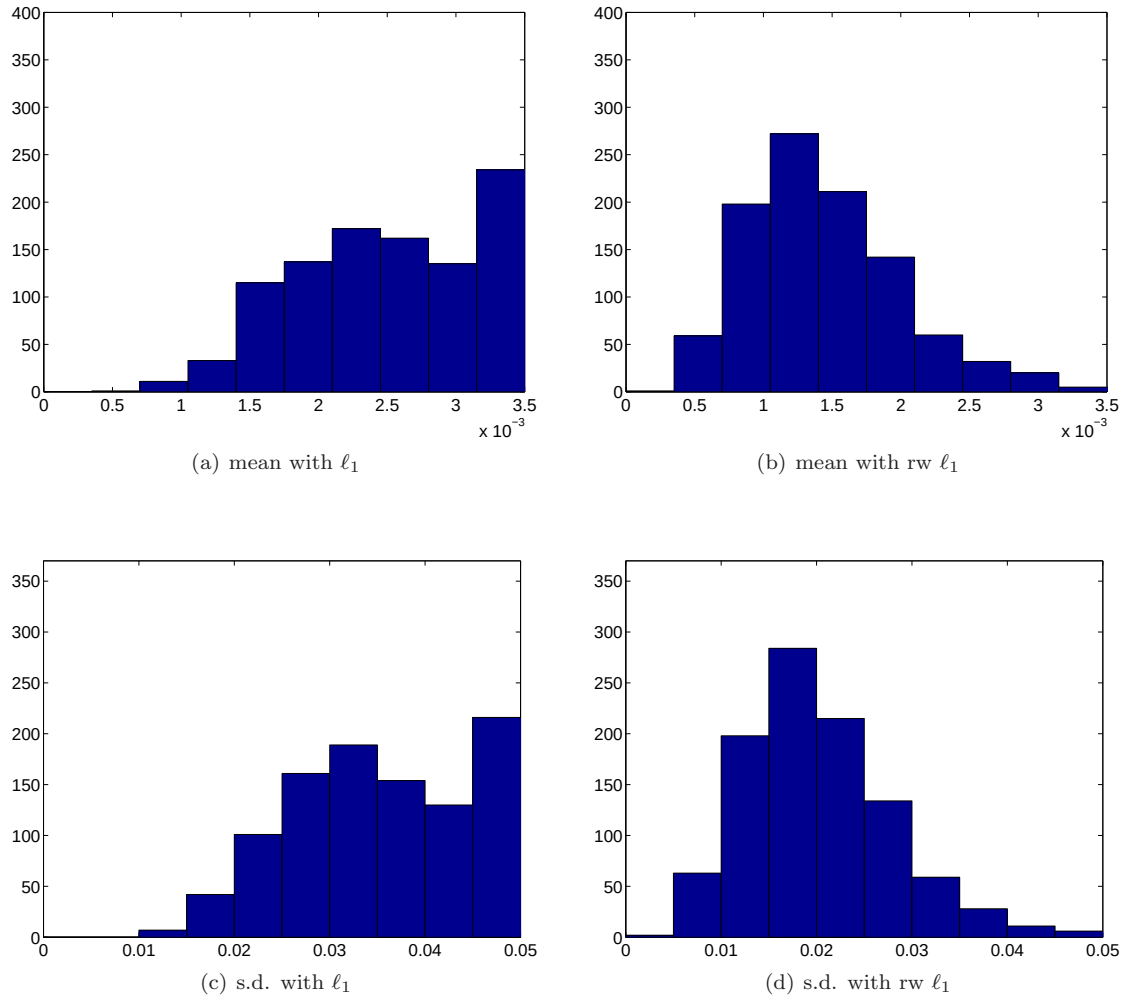
Table 3.6: 95% confidence interval of the mean of relative error of statistics over the entire physical domain when uniform points are employed. "MC" denotes the result without any post processing, "rw ℓ_1" denotes reweighted ℓ_1 minimization.

	Relative error in the mean		Relative error in the s.d.	
	MC	rw ℓ_1	MC	rw ℓ_1
$d = 14$	$[8.8e-3,\ 9.5e-3]$	$[1.4e-3,\ 1.5e-3]$	$[5.8e-2,\ 6.1e-2]$	$[2.0e-2,\ 2.1e-2]$
$d = 40$	$[3.6e-3,\ 3.8e-3]$	$[4.7e-4,\ 4.9e-4]$	$[4.3e-2,\ 4.6e-2]$	$[1.3e-2,\ 1.4e-2]$

Remark 3.3.1. In these tests, there are analytical results guaranteeing that the solutions are sparse in the random space no matter which physical point is considered, hence, we obtain a good global recovery. For problems in which the sparsity varies at different locations, our method is effective on the area with sparse solution while less effective on the area with less sparse solution. This is a multiscale problem and requires further investigation.

Reweighted ℓ_1 minimization combined with Chebyshev points

Similar to the tests of the 1-D case, we also compare the results by different sampling strategies



(a) mean with ℓ_1

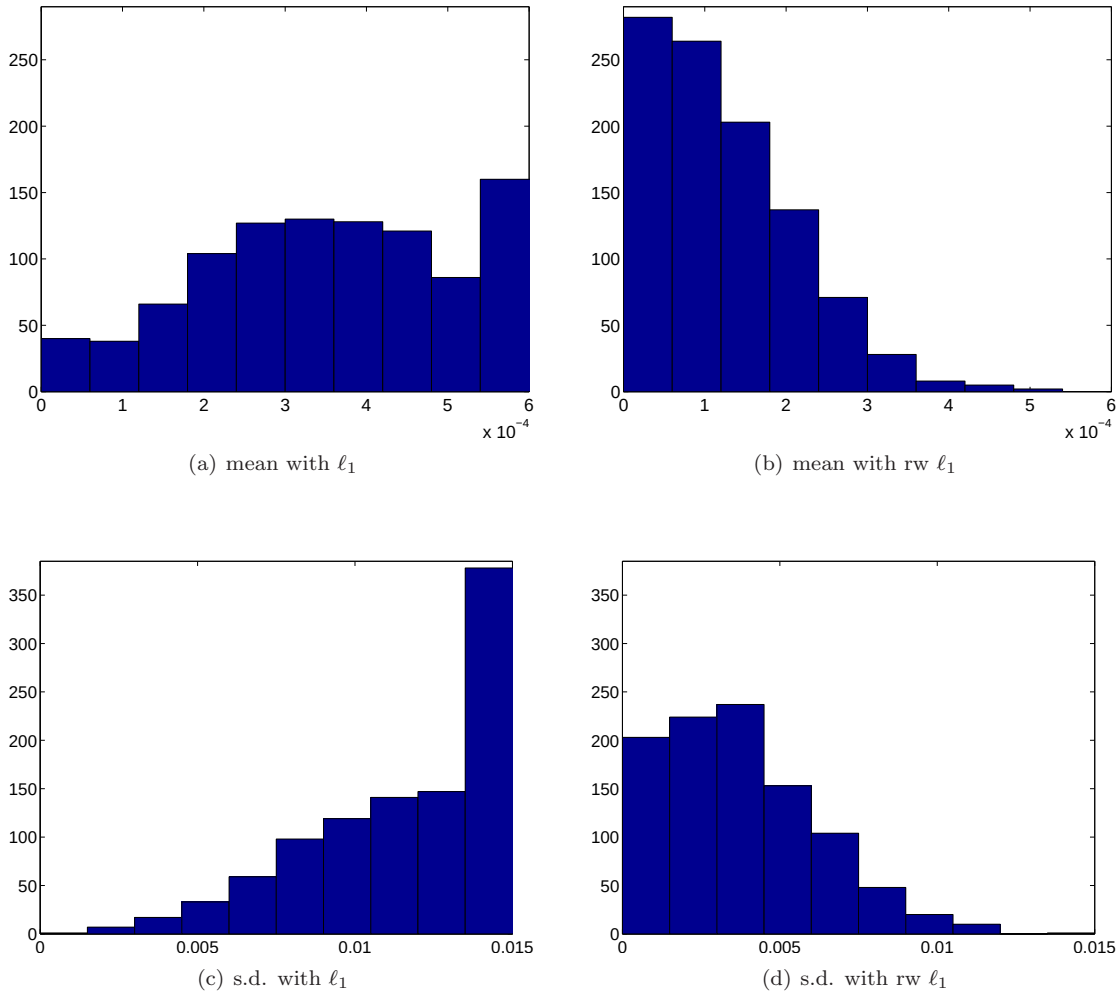
(b) mean with rw ℓ_1

(c) s.d. with ℓ_1

(d) s.d. with rw ℓ_1

Figure 3.11: 14D example: relative (global) error in the mean and the standard deviation ("s.d.") of the approximated solution over the physical domain with the coefficients $\boldsymbol{c}$ recovered from ℓ_1 or reweighted ("rw") ℓ_1 minimization with uniform points. In each trial, 120 sampling points are employed and $1,000$ trials are tested. ϵ is obtained by cross-validation and $l_{\max} = 2, \bar{a} = 0.1, \sigma_a = 0.03, d = 14, \tau = 10^{-3}$.

with or without reweighted iterations. Figure 3.9 presents the results for $d = 14$ case. It is clear that Chebyshev points help to improve the result as (c) shows better results than (a), and (d) shows better results than (b). A very important point is that instead of using Chebyshev point in all the dimensions, we only use for the first d' ($< d$) dimensions and in the remaining dimensions we still use uniform points. We notice that this is the most effective way to apply this sampling strategy in this case. Our numerical tests show that $3 \leq d' \leq 7$ are good choices, and we present the results



(a) mean with ℓ_1

(b) mean with rw ℓ_1

(c) s.d. with ℓ_1

(d) s.d. with rw ℓ_1

Figure 3.12: 40D example: relative error (global) in the mean and the standard deviation ("s.d.") of the approximated solution over the physical domain with the coefficients $\boldsymbol{c}$ recovered from ℓ_1 or reweighted ("rw") ℓ_1 minimization with uniform points. In each trial, 200 sampling points are employed and $1,000$ trials are tested. ϵ is obtained by cross-validation and $l_{\max} = 2, \bar{a} = 0.1, \sigma_a = 0.021, d = 40, \tau = 10^{-3}$.

for $d' = 6$ in this section. For $d = 40$, as shown in Figure 3.10, we observe a different phenomenon, i.e., Chebyshev points may not improve the estimate, e.g., at $x = 0.5$, especially when reweighted ℓ_1 is employed. Our tests show that for the estimate of the solution on the entire physical domain, $1 \leq d' \leq 3$ are good choices, and we present the result of $d' = 3$ for this test case.

Tables 3.7 and 3.8 present comparison between ℓ_1 minimization and reweighted ℓ_1 minimization when Chebyshev points are employed. It is clear that the reweighted iterations improve the accuracy

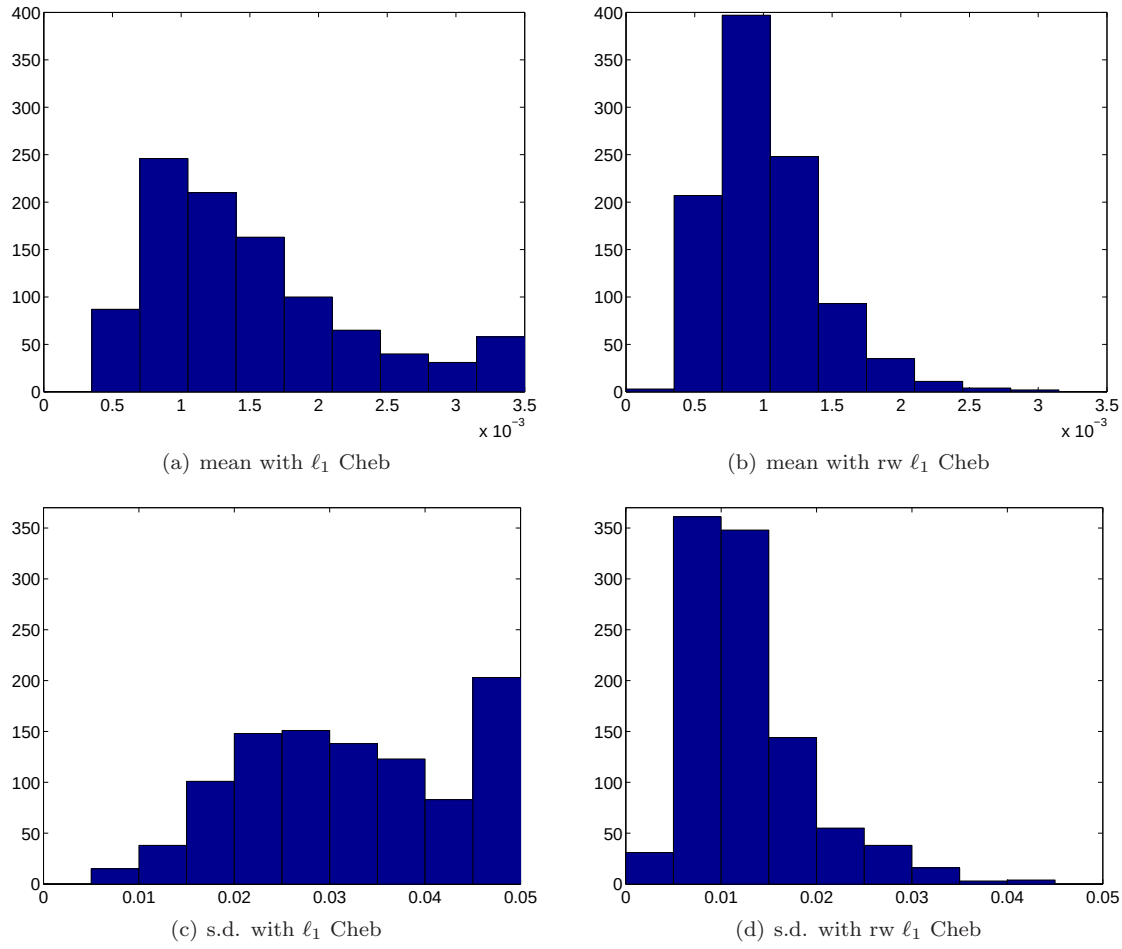
of the recovery. Comparing these two tables with Tables 3.4 and 3.5, we observe that the benefit of employing Chebyshev points is clear for $d = 14$ and the estimate of the standard deviation for $d = 40$. However, for the estimate of the mean for $d = 40$ case when reweighted ℓ_1 is used, there is no improvement by employing Chebyshev points. The global error of statistics over the physical domain is presented in Figure 3.13 and Figure 3.14 for $d = 14$ and $d = 40$, respectively. We can observe the improvement by applying the reweighted iterations. Also, comparing these results with those in Figures 3.11 and 3.12, where only uniform points are employed, we notice that when standard ℓ_1 minimization is used, Chebyshev points improve the results for both $d = 14$ and $d = 40$ case. When reweighted iterations are employed, Chebyshev points improve the results for $d = 14$. However, for $d = 40$ there is improvement for the estimate of the standard deviation but no improvement for the mean. This findings implies that for mildly high dimension, e.g., $d = 14$, the combination of reweighted ℓ_1 and Chebyshev points works well while for higher dimensions, e.g., $d = 40$ only applying reweighted ℓ_1 minimization is sufficient. We list in Table 3.9 the 95%

Table 3.7: 14D example: number of trials out of 1000 with corresponding error at $x = 0.5$ for different methods for $d = 14$. 120 sampling points are employed in each trial. $l_{\max} = 2, \bar{a} = 0.1, \sigma_a = 0.03, d = 14, d' = 6, \tau = 10^{-3}$. As a comparison, the error of level 1 sparse grids method (29 samples) $9.4387e-4$ for the mean and $5.0068e-2$ for the standard deviation while the corresponding data of level 2 sparse grids method (421 samples) is $3.2826e-5$ and $1.4532e-3$.

Relative error in the mean ε_m			Relative error in the s.d. ε_s		
	ℓ_1	rw ℓ_1		ℓ_1	rw ℓ_1
$< 9.4e-4$	920	975	$< 5e-2$	1000	1000
$< 5e-4$	690	792	$< 1e-2$	841	958
$< 3.3e-5$	59	68	$< 1.5e-3$	181	238

Table 3.8: 40D example: number of trials out of 1000 with corresponding error at $x = 0.5$ for different methods for $d = 40$. 200 sampling points are employed in each trial. $l_{\max} = 2, \bar{a} = 0.1, \sigma_a = 0.021, d = 40, d' = 3, \tau = 10^{-3}$. As a comparison, the error of level 1 sparse grids method (81 samples) $4.3655e-4$ for the mean and $6.4767e-2$ for the standard deviation while the corresponding data of level 2 sparse grids method (3281 samples) is $1.9131e-5$ and $3.6733e-3$.

Relative error in the mean ε_m			Relative error in the s.d. ε_s		
	ℓ_1	rw ℓ_1		ℓ_1	rw ℓ_1
$< 4.4e-4$	795	988	$< 6.5e-2$	1000	1000
$< 1.0e-4$	207	414	$< 1.0e-2$	655	998
$< 1.9e-5$	43	70	$< 3.7e-3$	92	754



(a) mean with ℓ_1 Cheb

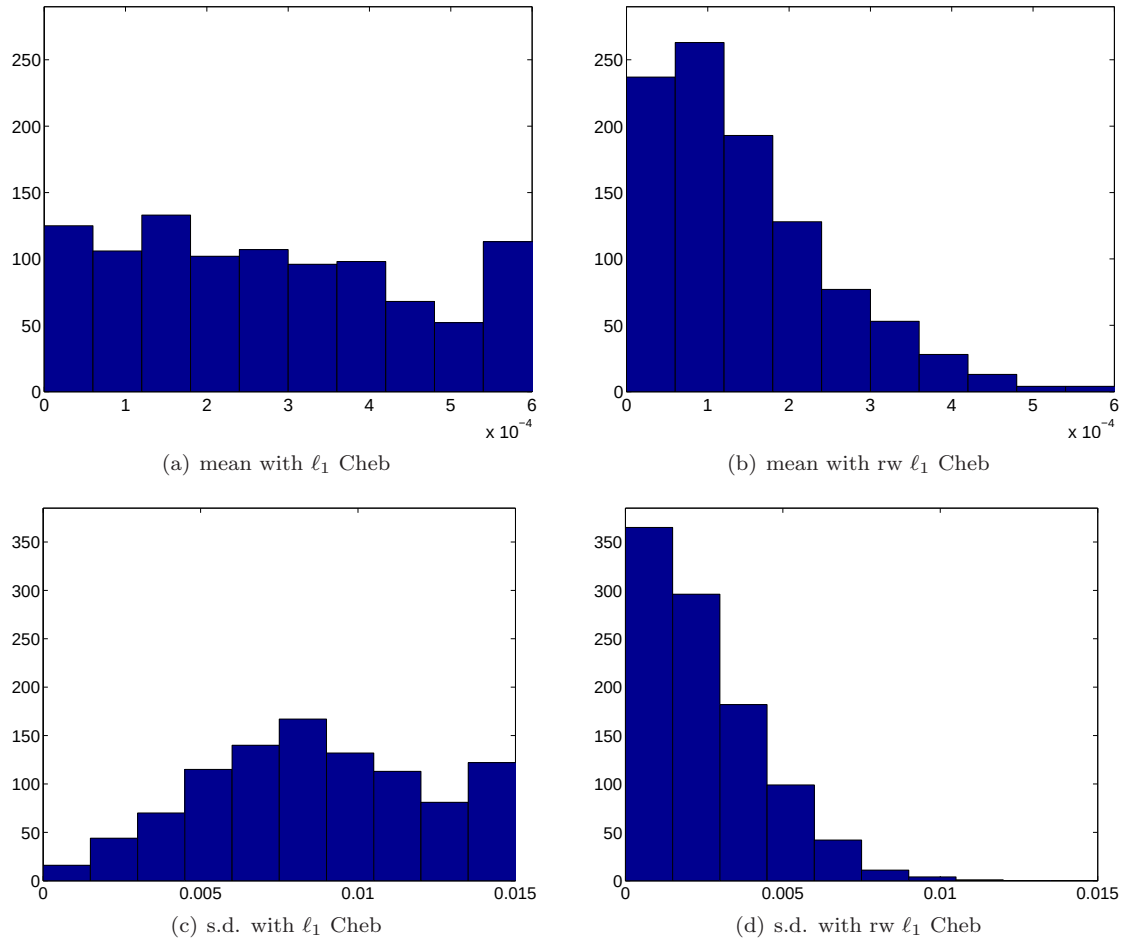
(b) mean with rw ℓ_1 Cheb

(c) s.d. with ℓ_1 Cheb

(d) s.d. with rw ℓ_1 Cheb

Figure 3.13: 14D example: relative (global) error in the mean and the standard deviation ("s.d.") of the approximated solution over the physical domain with the coefficients $\boldsymbol{c}$ recovered from ℓ_1 or reweighted ("rw") ℓ_1 minimization with Chebyshev points. In each trial, 120 sampling points are employed and $1,000$ trials are tested. ϵ is obtained by cross-validation and $l_{\max} = 2, \bar{a} = 0.1, \sigma_a = 0.03, d = 14, d' = 6, \tau = 10^{-3}$.

confidence interval of the mean of the global error of statistics by Monte Carlo method (without post processing) with uniform points and by reweighted ℓ_1 minimization with Chebyshev points. We observe that the reweighted ℓ_1 minimization reduces the error of the mean by nearly 90% for both $d = 14$ and $d = 40$. It reduces the error of the standard deviation by nearly 80% for $d = 14$ and 74% for $d = 40$. Comparing the results in Table 3.9 with those in Table 3.6, we notice that the combination of Chebyshev points and reweighted ℓ_1 minimization is the best choice for $d = 14$, but this is not true for $d = 40$. The reweighted ℓ_1 with uniform points provides better estimate



(a) mean with ℓ_1 Cheb

(b) mean with rw ℓ_1 Cheb

(c) s.d. with ℓ_1 Cheb

(d) s.d. with rw ℓ_1 Cheb

Figure 3.14: 40D example: relative (global) error in the mean and the standard deviation ("s.d.") of the approximated solution over the physical domain with the coefficients $\boldsymbol{c}$ recovered from ℓ_1 or reweighted ("rw") ℓ_1 minimization with Chebyshev points. In each trial, 200 sampling points are employed and $1,000$ trials are tested. ϵ is obtained by cross-validation and $l_{\max} = 2, \bar{a} = 0.1, \sigma_a = 0.021, d = 40, d' = 3, \tau = 10^{-3}$.

of the mean while the reweighted ℓ_1 with Chebyshev points shows better estimate of the standard deviation. These results are consistent with the comparison of Figure 3.12 (b) and Figure 3.14 (b) as well as comparison of Figure 3.12 (d) and Figure 3.14 (d). In this specific case, the exact mean is more than 50 times larger than the exact standard deviation. Hence when we compare the L_2 error at fixed spatial points, the reweighted ℓ_1 method with uniform points performs better. This is consistent with the comparison of Figure 3.10 (b) and (d). The above results imply that for moderately high dimensional (~ 10) problems, combination of reweighted ℓ_1 and Chebyshev points

is a good choice while for higher dimensional problems, reweighted ℓ_1 only might be a better choice than combining it with Chebyshev points.

Table 3.9: 95% confidence interval of the mean of relative error of statistics over the entire physical domain. "MC unif" denotes the results by Monte Carlo method with uniform points and without any post processing; "rw ℓ_1 Cheb" denotes the results by employing reweighted ℓ_1 minimization method and Chebyshev points.

	Relative error in the mean		Relative error in the s.d.	
	MC unif	rw ℓ_1 Cheb	MC unif	rw ℓ_1 Cheb
$d = 14$	$[8.8e-3,\ 9.5e-3]$	$[9.9e-4,\ 1.0e-3]$	$[5.8e-2,\ 6.1e-2]$	$[1.2e-2,\ 1.3e-2]$
$d = 40$	$[3.6e-3,\ 3.8e-3]$	$[4.2e-4,\ 4.4e-4]$	$[4.3e-2,\ 4.6e-2]$	$[1.1e-2,\ 1.2e-2]$

Remark 3.3.2. In this section we use reweighted ℓ_1 minimization, Chebyshev points and the combination of these two approaches to improve the accuracy of the recovery of the coefficients in the gPC expansion of the solution of SPDEs. We point out that in the optimization problem $(P_{1,\epsilon})$ we set the bound of the "noise" by the cross-validation method, for which we do not tune the parameters precisely since our purpose is to demonstrate the improvement by the new method. Therefore, for some trials, the result by cross-validation may not be as accurate as using a randomly or empirically chosen parameter.

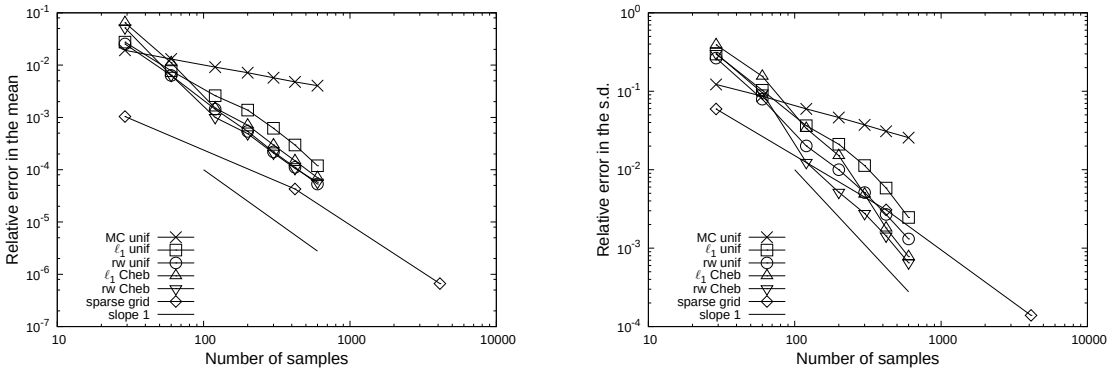
Remark 3.3.3. When applying Chebyshev points to high dimensional problems, we use an empirical parameter $d' < d$ because $d' = d$ is not a good choice. We observe that when using Chebyshev points in ℓ_1 minimization, the results are better than the standard ℓ_1 minimization where only uniform points are employed. As pointed out in [163], when Chebyshev points are applied to high dimensional problems, the number of samples to accurately recover the coefficients scales as 2^d. We reduce this by selecting $d' < d$, that is, employing Chebyshev points in only a few dimensions. Hence we can still take advantage of this sampling strategy especially in moderately high dimensional (~ 10) problems. In our test, we know a priori that the first several dimensions are more important due to the expression of diffusion coefficients (represented by the hierarchical Karhunen-Loève expansion), hence we employ Chebyshev points in these dimensions. For the reweighted ℓ_1 method, Chebyshev points only help in the moderately high dimensional (~ 10) case. An impor-

tant reason is that we employed a lower order gPC expansion, e.g., $P = 2$ for $d = 40$. The upper bound K of the L^∞ norm of the basis is $\sqrt{5}$, which is already very low. Since the Chebyshev points improve the recovery by controlling K (see Remark 3.1.6), it contributes only a little to the case with very low gPC order. This contribution is negligible compared with the improvement from reweighted iterations.

Comparison of the effectiveness

We now consider an increasing number of random samples to evaluate the solution and compare the global relative error with different methods, including Monte Carlo method and (reweighted) ℓ minimization (with uniform points or Chebyshev points). For each case we compute the results at $x = 0.1, 0.2, \cdots, 0.9$ on the physical domain to obtain the global error. For $d = 14$, we consider $m = \{29, 60, 120, 200, 300, 421, 600\}$. We note that sample size $m = 29$ and $m = 421$ correspond to the number of sparse grid method of level 1 and 2, respectively. In this test, for sample size $m = \{29, 60, 120\}$ we attempt to recover the gPC coefficients of the 3rd-order gPC expansion. For larger sample size m, we also include the first 320 basis function from the 4th-order gPC expansion, resulting in $m = 1000$ (see also Case I in [39]). We test 1000 trials for each sample size m and present the mean of the relative error in Figure 3.15. We observe that when a sufficient number of solution samples is available, which in Figure 3.15 is 120, ℓ_1 minimization shows an advantage over the Monte Carlo method. If reweighted iterations are employed, the ℓ_1 minimization begins to show an advantage with a smaller number of samples as pointed out in [29]. As the number of samples increases, we observe a more considerable improvement over the Monte Carlo method. For example, when $m = 600$, the mean of the relative error in the mean by reweighted ℓ_1 (both uniform points and Chebyshev points) is only about 1.5% of the error by the Monte Carlo method, i.e., our method increases the accuracy by about two orders in the estimate of the mean. Also, for this test case, we notice that with the same number of samples ($m = 421$), the accuracy of the estimate of the mean by our method is close to that by sparse grid method of level 2 while the estimate of

the standard deviation by our method is better. We also present a line of slope 1 in Figure 3.15 to compare the behavior of our method. Theorem 3.6 in [39] provides an upper bound of the L_2 error of the solution, which is approximately $\mathcal{O}(m^{-1/2})$. Figure 3.15 implies that in practice we can expect faster convergence in the estimate of statistics, which is also reflected in Figures 3 and 5 in [39]. Moreover, we notice that the reweighted iterations do not change the rate of convergence substantially, but reduce the error by around $50\% \sim 60\%$ for the mean and $60\% \sim 70\%$ for the standard deviation. We present the error bars of the 1000 trials for different sample size by the Monte Carlo method and (reweighted) ℓ_1 minimization method in Figure 3.16. This figure also implies that reweighted iterations improve the estimate of the solutions, when a sufficient number of samples is available.

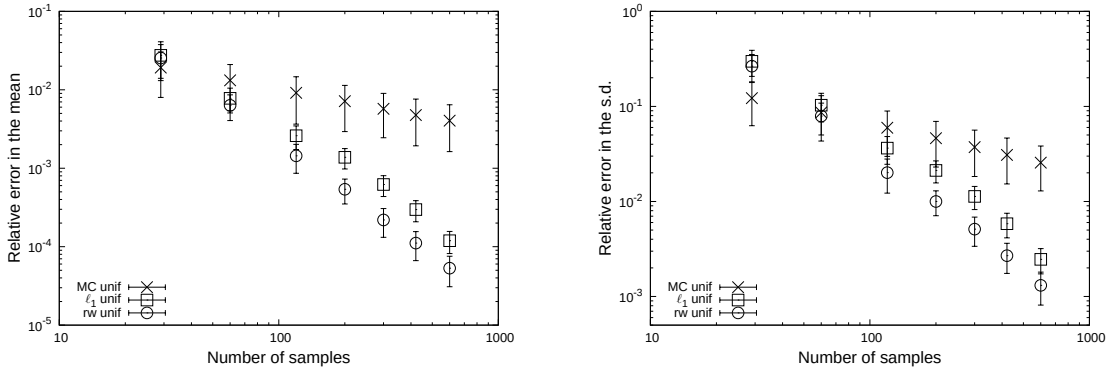


(a) Error in the mean.

(b) Error in the s.d.

Figure 3.15: 14D example: mean of the relative error (global) in the mean and the standard deviation ("s.d.") of the approximated solution over the physical domain by different methods. "unif" means uniform points, "Cheb" means Chebyshev points, "rw" means reweighted method.

Similar results for $d = 40$ are presented in Figures 3.17 and 3.18. In this test we chose $m = \{81, 120, 200, 400, 600, 800, 1000\}$. For sample size $m = \{81, 120, 200\}$, we attempt to recover the gPC coefficients of the 2nd-order gPC expansion. For larger sample size m, we also include the first 639 basis function from the 3rd-order gPC expansion, hence $m = 1500$ (see also Case II in [39]). Figure 3.17 implies that $m = 200$, is the smallest size of the samples in our test which guarantees that ℓ_1 minimization performs better than the Monte Carlo method. However, we note

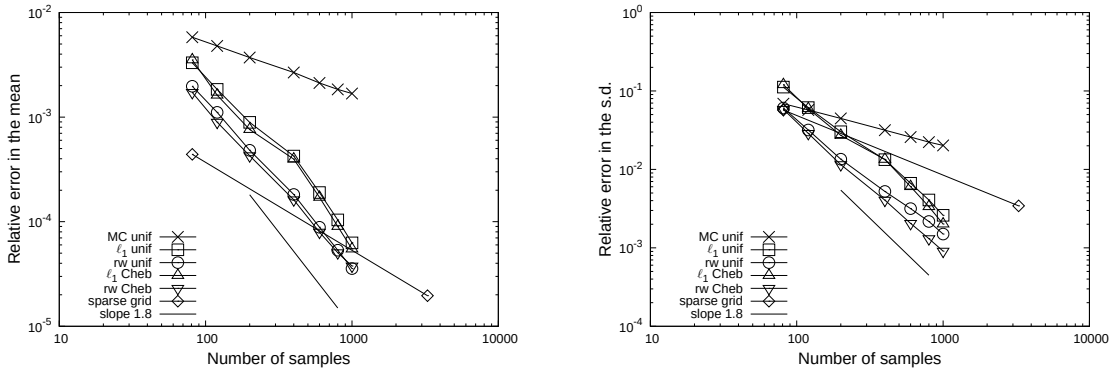


(a) Error bar of the mean.

(b) Error bar of the s.d.

Figure 3.16: 14D example: error bars (mean and standard deviation) of the relative error (global) in the mean and the standard deviation ("s.d.") of the approximated solution over the physical domain by different methods. "unif" means uniform points, "rw" means reweighted method.

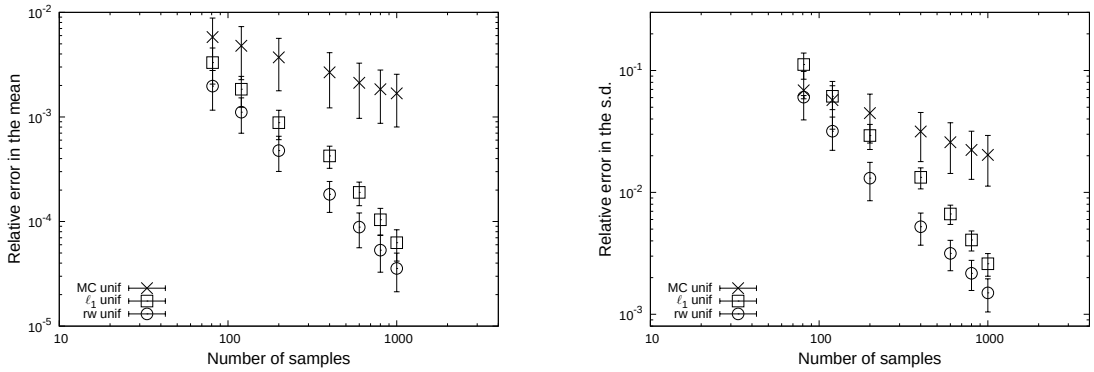
that when the reweighted ℓ_1 minimization is employed, $m = 120$ is sufficient to enable a better estimate. Also, similar to the $d = 14$ case, the reweighted ℓ_1 minimization method, even the standard ℓ_1 minimization, shows an advantage over the sparse grid method when sufficient samples are available. Moreover, we observe an even faster convergence rate than the $d = 14$ case as shown by a line of slope 1.8 for comparison. This again implies that in practice, we may expect a much faster convergence rate than 0.5. We also notice that the reweighted iterations do not change this rate substantially as in $d = 14$ case.



(a) Error in the mean.

(b) Error in the s.d.

Figure 3.17: 40D example: mean of the relative error (global) in the mean and the standard deviation ("s.d.") of the approximated solution over the physical domain by different methods. "unif" means uniform points, "Cheb" means Chebyshev points, "rw" means reweighted method.



(a) Error bar of the mean.

(b) Error bar of the s.d.

Figure 3.18: 40D example: error bars (mean and standard deviation) of the relative error (global) in the mean and the standard deviation ("s.d.") of the approximated solution over the physical domain by different methods. "unif" means uniform points, "rw" means reweighted method.

3.4 Parameter identification in mesoscopic model

3.4.1 Introduction of mesoscopic model

It is well known that many of the macroscopic properties observed in soft matter systems, such as liquid crystals, polymers, and colloids are natural consequences of the physical processes at the microscopic level. Accurate modeling of the corresponding processes enables us to successfully predict the properties of these material systems as well as, in turn, calibrate the parameters of the models. Molecular dynamics (MD) simulation, in conjunction with optimized force fields, has been successfully applied to study various physical systems [77, 136]. However, due to the explicit modeling of individual atomistic particles, the MD simulation method is limited and cannot reach the spatial and temporal scales relevant to the collective motion we are interested in, i.e., at the mesoscale.

To overcome this limitation, an alternative approach is to coarse grain (CG) the system by representing several atomistic particles as a single virtual particle. The essential idea is to eliminate the fast modes and corresponding degrees of freedom (DOF) while only keeping those DOFs relevant to the scale of our interest, represented as the CG particles. The static properties of the CG system is closely related to the governing force field, which has been studied extensively [41, 82, 102, 80, 9,

67, 52]. Remarkably, Espanol [41] modeled the DPD particles by grouping several Lennard Jones (LJ) particles into clusters, and derived the conservative force field from the radial distribution function of the clusters. Bolhuis *et al.* [102] mapped the semi-dilute polymer solution onto soft particles via an effective pairwise potential. Kremer *et al.* [67] and Fukunaga *et al.* [52] extracted the effective force field for complex polymers from the distribution functions of the bond length, bending angle and torsion angle. By carefully choosing the CG sites ("super atoms") and tuning the CG force field parameters, these CG models can reproduce the specified structural properties of the atomistic systems.

However, currently there are still two open questions for constructing the CG model and force field. First, most of the CG force fields are constructed by targeting certain static properties (e.g., the pair and angle distribution functions) of the atomistic system, while the constructed CG force fields show limitation if we consider other static properties (e.g., the equation of state) [8, 86]. In particular, how to construct a CG model with a thermodynamically consistent force field is still an open question. Moreover, the CG force fields are insufficient to reproduce the dynamic properties of the atomistic system. Instead, two additional (dissipative and random) force terms should be introduced to compensate for the eliminated atomistic DOFs during the CG procedure [81]. This is represented by the force terms in mesoscopic simulation methods such as dissipative particle dynamics (DPD) [73, 42]. Computing the dissipative and random force terms directly from the atomistic systems is a non-trivial task. Eriksson *et al.* [40] estimated the dissipative force term by the force covariance function, whereas Lei *et al.* [86] and Hijon *et al.* [68] computed the dissipative force term using the Mori-Zwanzig formulation [112, 173]. While the computed dissipative force terms successfully reproduce the dynamic properties in the dilute and semi-dilute regime, the force terms show limitation in reproducing the dynamic properties for highly correlated systems. Currently, it is still unclear how to determine the optimized CG force terms with correct dynamic properties over the entire density regime.

3.4.2 Parameter induced uncertainty

To circumvent those difficulties discussed above, we study the CG systems by considering an alternative question: what is the *distribution* of those dynamic properties, if we specify the CG force field within certain confidence range instead of a single deterministic value over the parameter space. That is, how do we quantify the propagation of the CG force parameter-induced uncertainty at the microscopic scale throughout the observed properties at the macroscopic scale? Such study enables us to analyze the sensitivity of the macroscopic quantities on individual force parameters, as well as validating the physical model or inferring the force field parameters if the macroscopic quantities at certain parameter values are known. For example, Rizzi *et al.* [126, 127] used the generalized polynomial chaos (gPC) [57, 157] expansion to construct a surrogate model and inferred the three-dimensional force parameters in TIP4P water model by using the macroscopic properties. Koumoutsakos *et al.* [10] implemented the transitional Markov chain Monte Carlo (TMCMC) algorithm to investigate the MD system of liquid and gaseous argon by adopting the kriging algorithm as the adaptive surrogate model.

In this section, we aim to investigate the static and dynamic properties of mesoscopic systems governed by dissipative particle dynamics (DPD) force field within a high dimensional random parameter space. In particular, we study the dynamic properties of two mesoscopic systems: a simple Newtonian fluid system governed by DPD force field over a three-dimensional parameter space, and a non-Newtonian polymer melt system governed by DPD force field over a six-dimensional parameter space. The target property is approximated by a gPC expansion in the form of a linear combination of a set of special basis functions defined in the parameter space. Hence, the estimate of this property with a specific parameter set can be considerably accelerated by evaluating the value of the gPC expansion instead of running costly simulations. Within this framework, we can exploit features of several approaches proposed recently on uncertainty quantification (UQ), e.g., (adaptive) sparse grid method [156, 53, 48, 117, 105] or (adaptive) ANOVA method [107, 47, 169, 164].

All these methods are categorized as probabilistic collocation methods (PCM) as they provide smart strategies to select sampling points to obtain a good estimate of the statistics of the system efficiently. However, in practice, the curse of high-dimensionality is still a big issue. The adaptive methods developed in [126, 164] provide some choices to deal with this issue but they require suitable empirical adaptivity criteria, which may not always be effective.

In recent years, some efforts have been made to propose a non-adaptive but simple and accurate method for high dimensional problems [94, 39, 163, 165] when the system is *sparse*, which within the UQ framework means that only a small portion of the gPC coefficients has relatively large values while the others are close to zero. This idea stems from the compressive sensing area [28, 38, 22]. More precisely, if we know a priori that the vector of gPC coefficients is sparse, we can employ compressive sensing techniques to "recover" these coefficients accurately using only a few (deterministic) simulation data. This method can be considered as post processing of Monte Carlo method (or PCM method), hence it is flexible as we can always incorporate new available data and obtain better estimates. This has an advantage over many of the popular methods, e.g., the standard sparse grid method, which is a golden standard in UQ today, as it requires a fixed number of additional simulations to reach the next accuracy level. This requirement is prohibitive in practice when the dimension of the problem is high. Even with the adaptive method, the selection of adaptivity criteria can be constrained by the limitation of computational resources. Moreover, in MD and DPD simulations, due to the existence of the thermal noise, it is difficult to obtain a highly accurate gPC surrogate model. In other words, even if we can afford running simulations with high order PCM method, we may not be able to reduce the error of the surrogate model. This is different from research on UQ in solving stochastic PDEs, where high accuracy can be expected with high order methods. Therefore, it is more helpful to find a method which can obtain a gPC expansion with small errors (ideally close to the thermal noise level) fast.

After obtaining a good surrogate model of the response surface of the target property, we employ Bayesian inference to explore suitable model parameters in DPD. With the gPC expansion, we are

able to dramatically reduce the cost of evaluating the likelihood, since we do not need to run the whole DPD simulation when this evaluation is needed. This type of gPC based Bayesian inference has been applied in several inverse problems, e.g., [110, 109], and we adopt a similar framework but employ a new technique to construct the gPC expansion. Thus, for practical material design problems, given the desirable target properties $\boldsymbol{P}^t$, we can firstly build a DPD model with empirical constants. Then, we parametrize these constants to obtain a parameter set $\boldsymbol{\theta}$. Next, we run the DPD simulation with different sets of $\boldsymbol{\theta}$ to obtain samples of target properties $\boldsymbol{P}^S$ and build the gPC approximation (surrogate model) of target properties depending on $\boldsymbol{\theta}$ via compressive sensing. Finally, we implement Bayesian inference based on $\boldsymbol{P}^t$ and the gPC expansion to infer suitable $\boldsymbol{\theta}$ for the DPD model. The above procedure can be repeated until the DPD model reaches required the accuracy by increasing the accuracy of the surrogate model or by modifying the range of the parameters. Figure 3.19 provides an overview of the entire procedure of inferring the parameters in the DPD model via gPC expansion and compressive sensing. The difference between $\boldsymbol{P}^t$ and $\boldsymbol{P}^{DPD}$ can be measured by the summation of relative error of each target property.

3.5 Simulation method and physical model

3.5.1 Dissipative Particle Dynamics

Dissipative Particle Dynamics (DPD) [73, 62] is a particle based mesoscopic simulation method. Each DPD particle represents a coarse-grained virtual cluster of multiple atomistic particles [86]. In standard DPD formulation [73], the motion of each particle is governed by

$$\begin{aligned} \mathrm{d}\boldsymbol{r}_i &= \boldsymbol{v}_i \mathrm{d}t \\ \mathrm{d}\boldsymbol{v}_i &= (\boldsymbol{F}_i^C \mathrm{d}t + \boldsymbol{F}_i^D \mathrm{d}t + \boldsymbol{F}_i^R \sqrt{\mathrm{d}t})/m, \end{aligned} \tag{3.5.1}$$

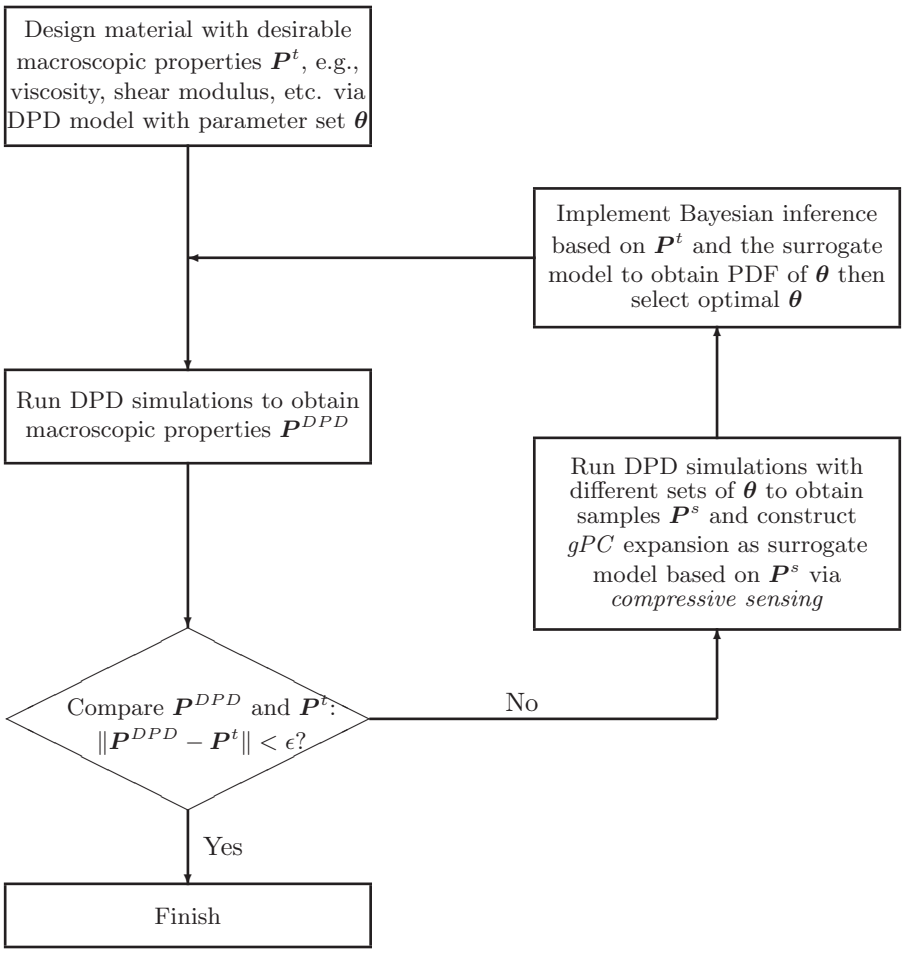


Figure 3.19: Overview of estimating parameters to achieve desired material properties using the DPD model.

where $\boldsymbol{r}_i$, $\boldsymbol{v}_i$, m are the position, velocity, and mass of the particle i, and $\boldsymbol{F}_i^C$, $\boldsymbol{F}_i^D$, $\boldsymbol{F}_i^R$ are the total conservative, dissipative and random forces acting on the particle i, respectively. Under the assumption of pairwise interactions the DPD forces can be decomposed into pair interactions with

the surrounding particles by

$$\begin{aligned}
\boldsymbol{F}_{ij}^{C} &= \begin{cases} a(1.0 - r_{ij}/r_c)\boldsymbol{e}_{ij}, & r_{ij} < r_c, \\ 0, & r_{ij} > r_c, \end{cases} \\
\boldsymbol{F}_{ij}^{D} &= -\gamma w^{D}(r_{ij})(\boldsymbol{v}_{ij} \cdot \boldsymbol{e}_{ij})\boldsymbol{e}_{ij}, \\
\boldsymbol{F}_{ij}^{R} &= \sigma w^{R}(r_{ij})\zeta_{ij}\boldsymbol{e}_{ij},
\end{aligned} \tag{3.5.2}$$

where $\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$, $r_{ij} = |\boldsymbol{r}_{ij}|$, $\boldsymbol{e}_{ij} = \boldsymbol{r}_{ij}/r_{ij}$, and $\boldsymbol{v}_{ij} = \boldsymbol{v}_i - \boldsymbol{v}_j$. r_c is the cut-off radius beyond which all interactions vanish. The coefficients a, γ and σ represent the strength of the conservative, dissipative and random force, respectively. The dissipative and random force terms are coupled with the system temperature by the fluctuation-dissipation theorem [42] as $\sigma^2 = 2\gamma k_B T$. Here, ζ_{ij} are independent identically distributed (*i.i.d.*) Gaussian random variables with zero mean and unit variance. The weight functions $w^D(r)$ and $w^R(r)$ are defined by

$$\begin{aligned}
w^{D}(r_{ij}) &= \left[w^{R}(r_{ij})\right]^2, \\
w^{R}(r_{ij}) &= (1.0 - r_{ij}/r_c)^k,
\end{aligned} \tag{3.5.3}$$

where k is a parameter that determines the extent of dissipative and random force envelopes.

While this method was initially proposed [73] to simulate the complex hydrodynamic processes of isothermal fluid systems, the particle based feature enables us to easily incorporate additional physical models and extend its application to various soft matter and complex fluid systems, such as polymer and DNA suspensions [135, 43, 139], platelet aggregation [122], colloid suspension [21], droplet wetting [95] and blood flow systems [121, 88, 87]. However, we note that the intrinsic relationship between DPD and the atomistic force field is not well understood yet. The force parameters of the mesoscopic models are usually chosen empirically (and often as a specific "point" in parameter space) while the variation of the macroscopic properties over the global parameter space has not been fully studied. Alternatively, in this work, we simulate the Newtonian fluid

and the non-Newtonian polymer melt systems by choosing the force parameters varying within a range of empirical values and we aim to systematically investigate the sensitivity dependence of the dynamic viscosity on individual force parameters.

The Newtonian fluid is modeled by homogeneous DPD particles in a domain of $40 \times 20 \times 20$ with periodic boundary condition. The number density n is 3.0 and the system temperature $k_B T = 1.0$. The force parameters in Eq. (3.5.2) and Eq. (3.5.3) are given by

$$\begin{aligned} a(\xi_1) &= \langle a \rangle + \sigma_a \xi_1, \\ \gamma(\xi_2) &= \langle \gamma \rangle + \sigma_\gamma \xi_2, \\ k(\xi_3) &= \langle k \rangle + \sigma_k \xi_3, \end{aligned} \tag{3.5.4}$$

where ξ_1, ξ_2 and ξ_3 are *i.i.d.* random variables uniformly distributed on $[-1, 1]$. The random force parameter σ is coupled with the dissipative force coefficient γ through $\sigma^2 = 2\gamma k_B T$. Other force parameters are listed in Table 3.10.

Table 3.10: Typical DPD force parameters for Newtonian fluid system. $\langle \star \rangle$ and $\sigma_\star$ represent the average and the magnitude of the random variables for the conservative force magnitude a, dissipative force coefficient γ and the power index of the dissipative force envelop k, respectively.

	a	γ	k
$\langle \star \rangle$	25.0	8.0	0.25
$\sigma_\star$	10.0	2.0	0.15

3.5.2 Polymer model and parameter uncertainty

The polymer melt system is modeled by N_p flexible polymer chains in a domain of $50 \times 20 \times 10$ with periodic boundary conditions with total DPD particle number density $n = 3.0$. Each polymer chain consists of $N_b = 2 \sim 5$ DPD particles connected by the Finitely Extensible Non-Linear Elastic (FENE) potential given by

$$U_{FENE} = -\frac{k_s}{2} r_{max}^2 \log \left[1 - \frac{|\boldsymbol{r}_i - \boldsymbol{r}_j|^2}{r_{max}^2} \right], \tag{3.5.5}$$

where k_s is the spring constant. The spring extension r is limited by its maximum value r_{max} attained when the corresponding spring force becomes infinite. The force parameters of the polymer melt system are given by

$$\begin{aligned} a(\xi_1) &= \langle a \rangle + \sigma_a \xi_1, \\ \gamma(\xi_2) &= \langle \gamma \rangle + \sigma_\gamma \xi_2, \\ k(\xi_3) &= \langle k \rangle + \sigma_k \xi_3, \\ r_c(\xi_4) &= \langle r_c \rangle + \sigma_{r_c} \xi_4, \\ k_s(\xi_5) &= \langle k_s \rangle + \sigma_{k_s} \xi_5, \\ r_{max}(\xi_6) &= \langle r_{max} \rangle + \sigma_{r_{max}} \xi_6 \end{aligned} \tag{3.5.6}$$

where $\xi_1, ..., \xi_6$ are *i.i.d.* random variables uniformly distributed on $[-1, 1]$. To investigate the numerical performance of the compressive sensing method on different parameter spaces, we choose three parameter sets with different variance values, as shown in Table 3.11.

Table 3.11: DPD force parameters for non-Newtonian polymer melt system. $\langle \star \rangle$ and $\sigma_\star$ represent the average and magnitude of the random variables for the values of the conservative force magnitude a, dissipative force coefficient γ, the power index of the dissipative force envelop k, the cutoff distance of DPD interaction r_c, the bond spring constant k_s, and the maximum bond extension distance r_{max}, respectively. The superscript l, m and s represent the three parameter sets with different variance values.

	a	γ	k	r_c	k_s	r_{max}
$\langle \star \rangle$	25.0	8.0	0.25	1.0	25.0	1.0
$\sigma_\star^l$	15.0	4.0	0.18	0.06	15.0	0.06
$\sigma_\star^m$	10.0	2.0	0.15	0.05	10.0	0.05
$\sigma_\star^s$	6.0	1.2	0.08	0.03	6.0	0.03

3.5.3 Shear viscosity rheometer

In traditional approaches, the shear viscosity of fluid system is usually determined either by the linear response theory with calculation of the stress auto-correlation function terms in equilibrium state, or by modeling the steady shear flow with the Lees-Edwards boundary conditions [85].

However, both approaches incorporate the calculation of the stress field, which is extremely noisy in equilibrium or low shear rate regime. Therefore, a large numerical error is anticipated in the regime where the thermal fluctuation dominates over the ensemble average value. Moreover, unlike a simple Newtonian fluid system, polymeric fluids typically exhibit non-Newtonian behavior with shear rate dependent viscosity. As shear rate approaches zero, the normal stress and viscosity value typically approach a constant *low-shear-rate* plateau, which is usually inaccessible due to rheometer limitations [83].

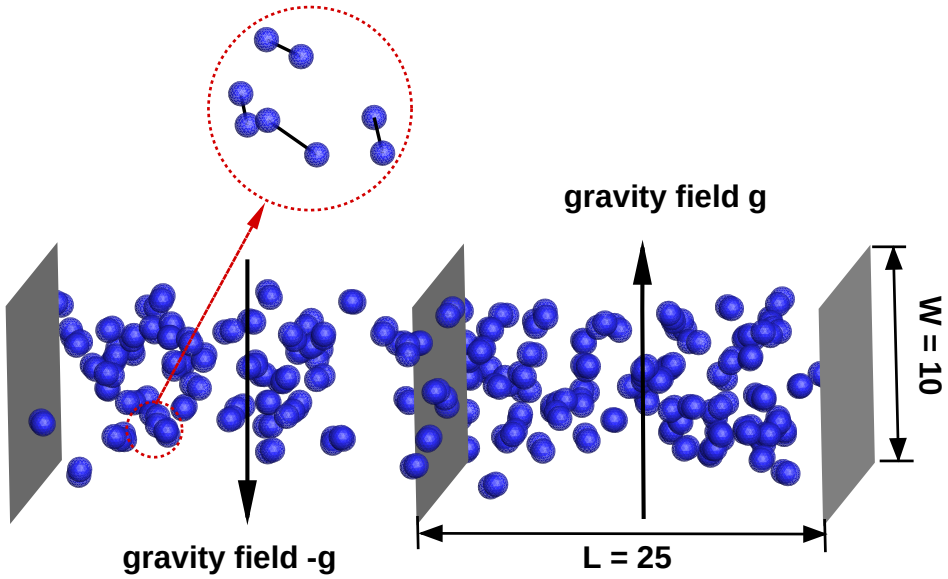


Figure 3.20: Sketch of the simulation polymer melt system. $N_p = 15,000$ polymer chains are placed in the simulation domain with size $50 \times 20 \times 10$ in DPD reduced units (only 2% of the simulation chain is visualized). Each polymer chain consists of $N_b = 2$ DPD particles, represented by the particles in blue color. Individual polymer chains are connected by FENE bond potential, as defined by Eq. (3.5.5).

Alternatively, we compute the *low-shear-rate* viscosity values using the reverse Poiseuille flow rheometer. This approach was initially proposed for computing the shear viscosity for Newtonian flow system. However, a later study showed that this approach can be well extended to study the complex fluid system with non-Newtonian behavior [44]. This approach generates more accurate

results than the previous two methods as it replaces the expensive stress calculation with the calculation of velocity profile. Here we briefly review the method and we refer to [12] and [44] for details.

As shown in Figure 3.20, we simulate $N_p = 15,000$ polymer chains in a domain of $50 \times 20 \times 10$ DPD units with periodic boundary conditions. The domain is divided into two regimes at center ($x = 0$), where an equal but opposite gravity force f between 0.015 and 0.05 is applied on each DPD particle in each half of the domain. At the cross-stream position x and time t, we have

$$\rho \frac{\partial u}{\partial t} = \frac{\partial \tau_{xy}}{\partial x} - fn, \tag{3.5.7}$$

where n is the number density. Under steady state flow, the left hand side of Eq. (3.5.7) vanishes and the shear stress τ_{xy} is linear across the channel with maximum value $fnH/2$ across the virtual wall boundary, where $H = 50$ is the length of the channel.

Having computed the shear stress profile, the shear-rate-dependent viscosity $\eta(x)$ can be calculated by

$$\tau_{xy}(x) = \eta(x)\dot{\gamma}(x), \tag{3.5.8}$$

where $\dot{\gamma}(x)$ is the shear rate across the channel. Therefore, the calculation of shear viscosity is transformed into the calculation of the velocity derivatives across the channel. For the Newtonian fluid system, the shear viscosity across the channel is nearly a constant and the steady flow velocity keeps the quadratic order. For the non-Newtonian polymer melt system, the shear viscosity is inhomogeneous across the channel. In particular, the *low-shear-rate* viscosity values are extracted from the velocity regime with a plateau profiles where a relatively larger numerical error is anticipated. To further reduce the numerical error, we fit the velocity profile of Newtonian fluid and polymer melt system using a 2nd and 4th-order polynomial, respectively:

$$V(x) = V_c \pm C_2(x \pm H/4)^2, \tag{3.5.9a}$$

$$V(x) = V_c \pm C_2(x \pm H/4)^2 \pm C_4(x \pm H/4)^4, \tag{3.5.9b}$$

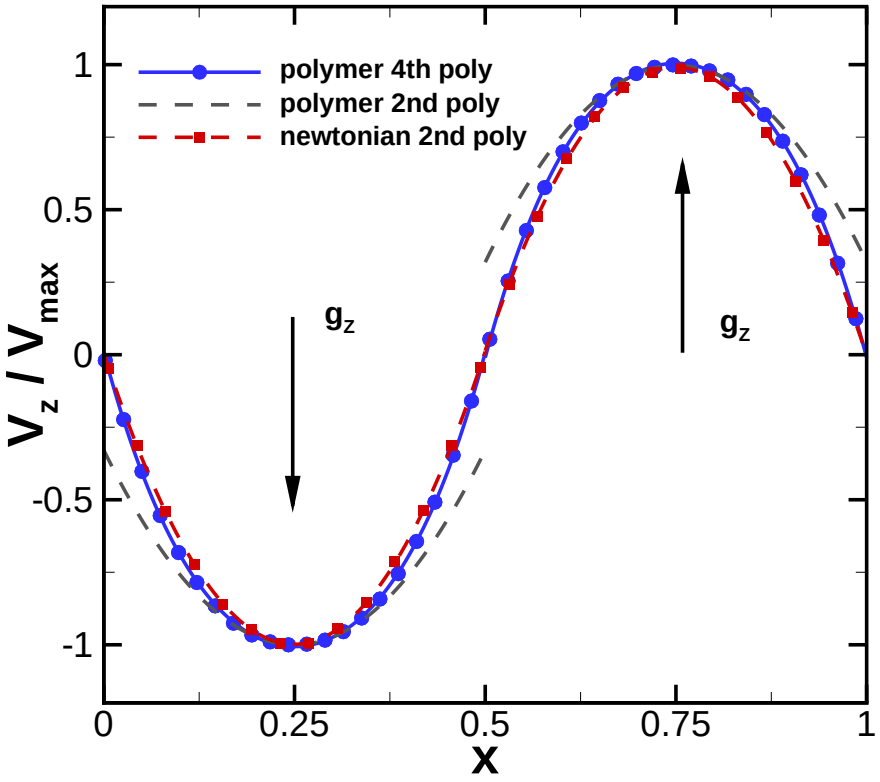


Figure 3.21: Typical steady reverse Poiseuille flow velocity profiles of the Newtonian and polymer melt system. The points represent the direct simulation results. The solid line represents the 4th order polynomial fitting curve for the polymer melt system. The dash lines represent the 2nd order polynomial fitting curves for the polymer melt and Newtonian flow systems.

For each parameter set, we conduct three independent reverse Poiseuille flow simulation: $800,000$ steps are performed for each simulation and the velocity profile sampling is taken during the last $700,000$ steps; a time step $dt = 0.01$ is adopted for all the simulations. Figure 3.21 shows an example of the steady velocity distribution across the channel for both the polymer melt and Newtonian fluid systems. The Newtonian fluid system maintains a near constant viscosity value across the

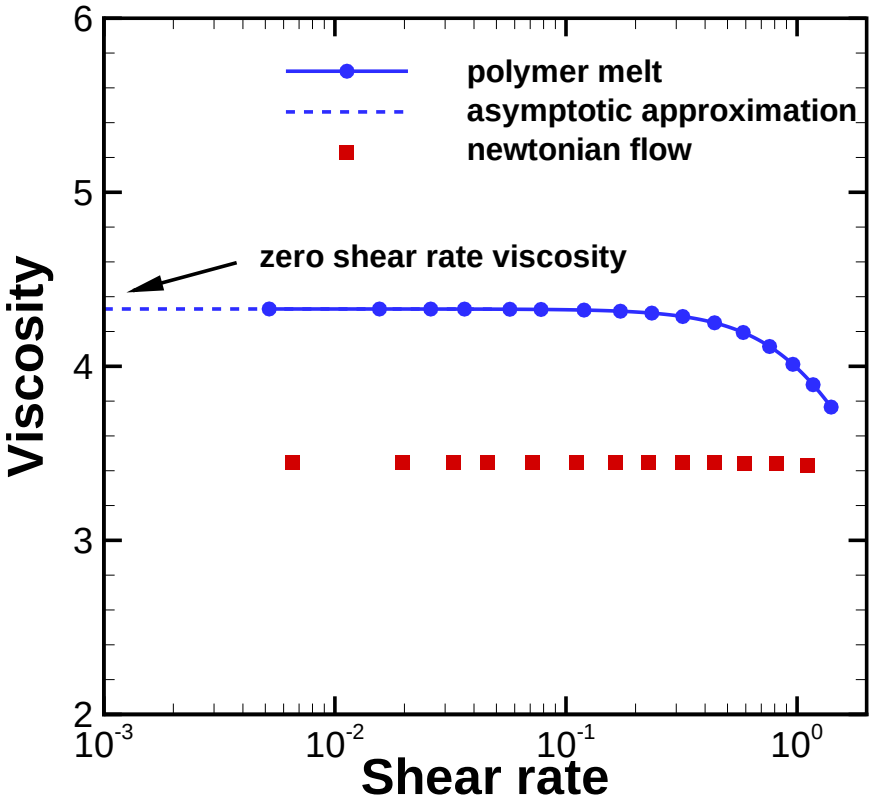


Figure 3.22: *Shear-rate-dependent* viscosity $\mu(\dot{\gamma})$ computed from the simulated velocity profiles for the polymer melt given $a = 20.0$, $\gamma = 8.0$, $k = 0.26$, $r_c = 1.0$, $k_s = 40.0$ and $r_{max} = 1.0$ and Newtonian flow systems given $a = 40.0$, $\gamma = 4.5$ and $k = 0.2$. The *zero-shear-rate* viscosity of the polymer melt system is computed by taking the limit of $\mu(\dot{\gamma})$.

channel and the velocity profile can be well fitted by Eq. (3.5.9a). On the contrary, the polymer melt system exhibits varying viscosity value across channel, resulting in a poor fitting by Eq. (3.5.9a). Instead, high order terms in Eq. (3.5.9b) are essential to capture the non-Newtonian properties. With the fitted velocity profile, the shear viscosity of the Newtonian fluid is determined by $\frac{nf}{2C_2}$ and the *zero-shear-rate* viscosity is computed by taking the limit $\eta_0 = \lim_{x \to \pm H/4} \eta(x)$, where $\eta(x)$ is *low-shear-rate* viscosity determined by the local shear viscosity determined by Eq. (3.5.8) and Eq. (3.5.9), as shown in Figure 3.22.

3.6 Stochastic sparse modeling for mesoscopic model

3.6.1 gPC expansion for target property

In this section, we introduce an efficient approach to approximate the response surface of the target property by generalized polynomial chaos (gPC) depending on the model parameter $\boldsymbol{\theta}$. Due to the linear relation between $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$ in this section (see Eq. (3.5.4) and Eq. (3.5.6)), we consider gPC depending on $\boldsymbol{\xi}$ instead. Given a set of parameter $\boldsymbol{\xi}$, the output $X(\boldsymbol{\xi})$ obtained from the DPD simulation is

$$X(\boldsymbol{\xi}) = \mu(\boldsymbol{\xi}) + \phi, \tag{3.6.1}$$

where $\mu(\boldsymbol{\xi})$ is the value of target property, and ϕ represents the intrinsic thermal noise discussed in the previous section. In order to compute $\mu(\boldsymbol{\xi})$, we simulate multiple replicas for the parameter set $\boldsymbol{\xi}$. As the number of replica increases, the average of $X(\boldsymbol{\xi})$ converge to $\mu(\boldsymbol{\xi})$ as we assume ϕ to be a Gaussian random variable. In the present work, we take three identical independent simulations for each $\boldsymbol{\xi}$ and consider the average $\bar{X}(\boldsymbol{\xi})$ as a good approximation of $\mu(\boldsymbol{\xi})$. With the same approach as in [126], we verify that in the cases we study, the magnitude of the fluctuation is $\mathcal{O}(0.1\%)$ (or smaller) of the target property. Hence, the error from the intrinsic thermal noise is much less than that from quantifying the parameter uncertainty.

Given $\bar{X}(\boldsymbol{\xi})$, we employ gPC to represent the target property:

$$\bar{X}(\boldsymbol{\xi}) = \sum_{|\boldsymbol{\alpha}|=0}^{\infty} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}), \tag{3.6.2}$$

where $\boldsymbol{\alpha}$ is a multi-index, $\boldsymbol{\xi} = (\xi_1, \xi_2, \cdots, \xi_d)$ is a vector of d *i.i.d.* random variables, $\psi_{\boldsymbol{\alpha}}$ is a set of orthonormal polynomials associated with the probability measure ν of $\boldsymbol{\xi}$ and $c_{\boldsymbol{\alpha}}$ is the coefficient.

We truncate the expression (3.6.2) up to polynomial order P, hence $\tilde{X}$ is approximated as:

$$\bar{X}(\boldsymbol{\xi}) \approx \widetilde{X}(\boldsymbol{\xi}) = \sum_{|\boldsymbol{\alpha}|=0}^{P} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}). \tag{3.6.3}$$

We use $\boldsymbol{c}$ to denote the vector of the gPC coefficients, i.e., $\boldsymbol{c} = (c_{\boldsymbol{\alpha}_1}, c_{\boldsymbol{\alpha}_2}, \cdots)$. By sorting the indices $\boldsymbol{\alpha}_i$ we can also write it as $\boldsymbol{c} = (c_0, c_1, \cdots)$, where c_0 is the coefficient of the zero-th order gPC basis function. The gPC representation $\widetilde{X}$ is constructed by computing the coefficients $c_{\boldsymbol{\alpha}}$, which can be accomplished by using the probabilistic collocation method (PCM) [156]. For example, tensor product points or sparse grid points can be employed. After obtaining $\bar{X}^1, \bar{X}^2, \cdots, \bar{X}^M$, which are values of $\bar{X}$ at sampling points $\boldsymbol{\xi}^1, \boldsymbol{\xi}^2, \cdots, \boldsymbol{\xi}^M$, we can compute the gPC coefficients of $\bar{X}$ as follows:

$$c_\alpha = \frac{\int \bar{X}(\boldsymbol{\xi}) \psi_\alpha(\boldsymbol{\xi}) \mathrm{d}\nu(\boldsymbol{\xi})}{\int \psi_\alpha^2(\boldsymbol{\xi}) \mathrm{d}\nu(\boldsymbol{\xi})}, \tag{3.6.4}$$

where ν is the probability measure of $\boldsymbol{\xi}$ and the integrals are multi-variate integrals depending on the dimension of $\boldsymbol{\xi}$ and are approximated by Gauss-type quadratures, e.g.,

$$\int \bar{X}(\boldsymbol{\xi}) \psi_\alpha(\boldsymbol{\xi}) \mathrm{d}\mu(\boldsymbol{\xi}) \approx \sum_{i=1}^{M} \bar{X}^i \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}^i) w^i. \tag{3.6.5}$$

Here w^i is the corresponding weight for $\boldsymbol{\xi}^i$.

3.6.2 Sparsity and compressive sensing

For systems with relatively high dimensions, e.g., the polymer melt model with $d = 6$, the method to compute the gPC coefficients described in Section 3.6.1 may not be efficient in that in order to increase the accuracy, the increment of the number of simulations is fixed and this number can be very large ($\mathcal{O}(10^2) \sim \mathcal{O}(10^3)$ or larger). Hence these methods can be prohibitive for DPD simulations which are usually very costly. Another drawback of this method is that it requires results from all the sampling points. If the DPD code fails at some sampling points, the method

fails, although we may employ other approaches to obtain a less accurate result. To overcome the above difficulties, we compute the gPC expansion by applying the compressive sensing method as post processing for the Monte Carlo method as discussed below.

We first generate M random samples of the parameter sets $\boldsymbol{\xi}^j, j = 1, \cdots, M$ based on the distribution of the random variables. Notice that $\boldsymbol{\xi}^j$ is a random vector with *i.i.d* entries. In this section, we generate M such random vectors based on the uniform distribution $\mathcal{U}[-1, 1]$. Then we input these vectors in the DPD code to obtain M outputs $\bar{\boldsymbol{X}} = (\bar{X}^1, \bar{X}^2, \cdots, \bar{X}^M)$, respectively. Based on (3.6.3), we obtain the following linear system:

$$\begin{pmatrix} \psi_{\boldsymbol{\alpha}_1}(\boldsymbol{\xi}^1) & \psi_{\boldsymbol{\alpha}_2}(\boldsymbol{\xi}^1) & \cdots \\ \psi_{\boldsymbol{\alpha}_1}(\boldsymbol{\xi}^2) & \psi_{\boldsymbol{\alpha}_2}(\boldsymbol{\xi}^2) & \cdots \\ & \vdots & \end{pmatrix} \begin{pmatrix} c_{\boldsymbol{\alpha}_1} \\ c_{\boldsymbol{\alpha}_2} \\ \vdots \end{pmatrix} = \begin{pmatrix} \bar{X}(\boldsymbol{\xi}^1) \\ \bar{X}(\boldsymbol{\xi}^2) \\ \vdots \end{pmatrix} + \boldsymbol{\varepsilon},$$

or equivalently,

$$\boldsymbol{\Psi c} = \bar{\boldsymbol{X}} + \boldsymbol{\varepsilon}, \tag{3.6.6}$$

where $\boldsymbol{\Psi}$ is the "measurement matrix" with entries $\boldsymbol{\Psi}_{i,j} = \psi_{\boldsymbol{\alpha}_j}(\boldsymbol{\xi}^i)$, $\boldsymbol{c}$ is the vector of the gPC coefficients, $\bar{\boldsymbol{X}}$ is the vector consisting of the outputs and $\boldsymbol{\varepsilon}$ is related to the truncation error. Here $\boldsymbol{\Psi}_{\boldsymbol{\alpha}_j}$ is the tensor product of normalized Legendre polynomials since $\boldsymbol{\xi}$ is uniformly distributed. According to the compressive sensing theory, when $\boldsymbol{c}$ is sparse and $\boldsymbol{\Psi}$ satisfies certain conditions, we can recover it from Eq. (3.6.6) by solving the following ℓ_h minimization problem:

$$(P_{h,\delta}): \qquad \min_{c} \|\boldsymbol{c}\|_h \quad \text{subject to} \quad \|\boldsymbol{\Psi c} - \bar{\boldsymbol{X}}\|_2 \leq \delta, \tag{3.6.7}$$

where $h = 1$ or 0, $\delta = \|\boldsymbol{\varepsilon}\|_2$, $\|\boldsymbol{c}\|_1 = \sum |c_i|$ and $\|\boldsymbol{c}\|_0 =$ number of nonzero entries of $\boldsymbol{c}$.

In this section we employ ℓ_0 minimization, i.e., the results of $(P_{0,\delta})$. To solve this minimization problem, we employ the greedy algorithm *orthogonal matching pursuit* (OMP) presented in **Algo-**

rithm 5. The threshold ϵ_0 in Step 6 is set as δ in Eq. (3.6.7) obtained by the cross-validation method (see Section 3.1.4). The algorithm of the entire procedure is presented in **Algorithm 7**. In this section, the OMP method is achieved by `SparseLab` [5]. We use "a trial" to denote completing **Algorithm 1** once. For different trials, we employ different sets of sampling points $(\boldsymbol{\xi}^1, \boldsymbol{\xi}^2, \cdots, \boldsymbol{\xi}^M)$. Finally, since M different samples of $\boldsymbol{\xi}^j$ lead to M different $\bar{X}^j$, we use "number of samples" to refer to M.

Algorithm 7 Compressive sensing method to obtain gPC surrogate model for DPD

1: Generate M sampling points $\boldsymbol{\xi}^1, \boldsymbol{\xi}^2, \cdots, \boldsymbol{\xi}^M$ based on the distribution of $\boldsymbol{\xi}$. Run the DPD code with inputs $\boldsymbol{\xi}^1, \boldsymbol{\xi}^2, \cdots, \boldsymbol{\xi}^M$ to obtain M outputs $\bar{X}^1, \bar{X}^2, \cdots, \bar{X}^M$, respectively. Denote $\bar{\boldsymbol{X}} = (\bar{X}^1, \bar{X}^2, \cdots, \bar{X}^M)$ and it is the "observation" in $(P_{0,\delta})$. The "measurement matrix" $\boldsymbol{\Psi}$ in is constructed as $\boldsymbol{\Psi}_{i,j} = \psi_{\boldsymbol{\alpha}_j}(\boldsymbol{\xi}^i)$, where $\psi_{\boldsymbol{\alpha}_j}$ are the basis functions. The size of $\boldsymbol{\Psi}$ is $M \times N$, where N is the total number of basis functions depending on P in (3.6.3).
2: Set the tolerance δ in $(P_{0,\delta})$ by employing cross-validation method (see Section 3.1.4).
3: Solve the weighted ℓ_0 minimization problem

$$\boldsymbol{c} = \arg\min \|\boldsymbol{c}\|_0 \quad \text{subject to} \quad \|\boldsymbol{\Psi}\boldsymbol{c} - \boldsymbol{u}\|_2 \leq \delta.$$

As we are need to construct an accurate surrogate model to approximate the response surface, we check the relative L_2 error:

$$\varepsilon_0 = \sqrt{\frac{\int |\mu(\boldsymbol{\xi}) - \tilde{X}(\boldsymbol{\xi})|^2 \mathrm{d}\nu(\boldsymbol{\xi})}{\int |\mu(\boldsymbol{\xi})|^2 \mathrm{d}\nu(\boldsymbol{\xi})}}, \tag{3.6.8}$$

where μ is the target property and $\tilde{X}$ is the gPC expansion of $\bar{X}$ (see Eq. (3.6.3)). Since we approximate μ with $\bar{X}$, we check the following error instead:

$$\varepsilon = \sqrt{\frac{\int |\bar{X}(\boldsymbol{\xi}) - \tilde{X}(\boldsymbol{\xi})|^2 \mathrm{d}\nu(\boldsymbol{\xi})}{\int |\bar{X}(\boldsymbol{\xi})|^2 \mathrm{d}\nu(\boldsymbol{\xi})}}. \tag{3.6.9}$$

As mentioned in Section 3.6.1, in our numerical examples, the thermal noise is about 10^{-3} (or smaller) of the target property, i.e., $|\mu - \bar{X}|/|\mu| \sim \mathcal{O}(10^{-3})$. Take the shear-rate viscosity for example, the relative error of approximating $\bar{X}$ by $\tilde{X}$ is $\mathcal{O}(10^{-2})$ (as we will see in this section), namely, $|\bar{X} - \tilde{X}|/|\bar{X}| \sim \mathcal{O}(10^{-2})$, which reflects that $|\mu - \tilde{X}|/|\mu| \sim \mathcal{O}(10^{-2})$. Hence, ε is a good

approximation of ε_0. The integrals in Eq. (3.6.9) are approximated by quadrature rules based on the tensor product or by the sparse grid method. The error by these methods is far below the magnitude of the thermal noise as we use high order quadrature points or a high level sparse grid method to approximate the integral.

3.6.3 Bayesian inference and Markov chain Monte Carlo

3.6.4 Newtonian flow

Figure 3.23 represents the shear viscosity of the Newtonian fluid system measured at the $7 \times 7 \times 7$ Gauss quadrature points with tensor product rule $\boldsymbol{\xi}^i = (\xi_1^i, \xi_2^i, \xi_3^i), i = 1, 2, \cdots, 343$. The positive correlation of the viscosity with ξ_1 and ξ_2 and the negative correlation with ξ_3 can be understood as follows: ξ_1 and ξ_2 determine the magnitude of the conservative and dissipative pairwise force interaction between DPD particles, respectively. Larger values enhance the momentum transport between a target DPD particle with its neighboring particles, resulting in larger shear viscosity values. On the other hand, ξ_3 quantifies the decay of dissipative force between two DPD particles with respect to the distance. A smaller value represents slower decaying of the dissipative interaction, resulting in larger shear viscosity values.

With the value of $\bar{X}$ at these quadrature points we compute a gPC expansion with order $P = 6$ by Eq. (3.6.4) and use it as the reference solution. The integrals in Eq.(3.6.9) are approximated by the $7 \times 7 \times 7$ tensor product Gauss quadrature points. We compute gPC expansions with $3 \times 3 \times 3$ and $4 \times 4 \times 4$ tensor product Gauss quadrature points to construct 2nd-order and 3rd-order gPC expansions of $\bar{X}$ respectively. The corresponding errors are presented in Figure 3.24. We then run Monte Carlo simulations and use the OMP method to construct a six-th order gPC expansion, which contains 84 basis functions. The results of two independent trials are also presented in Figure 3.24. We can observe that the L_2 error by $3 \times 3 \times 3$ and $4 \times 4 \times 4$ tensor product points are already very small. The OMP method shows better accuracy over the $3 \times 3 \times 3$ tensor product points when

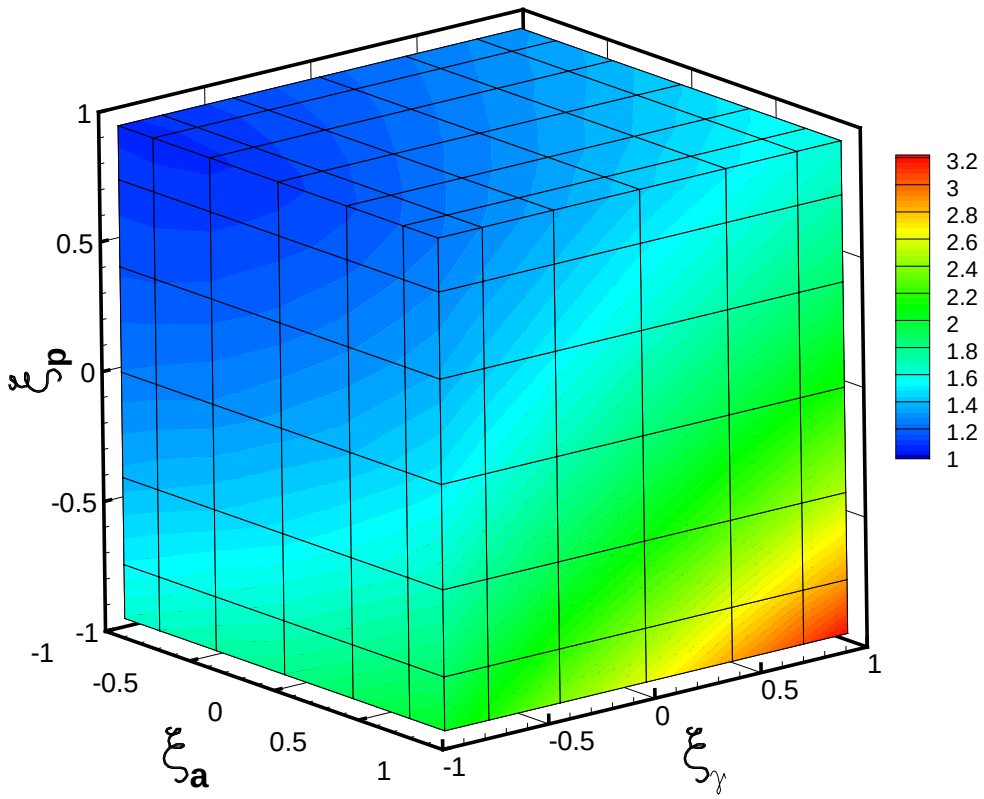


Figure 3.23: Shear viscosity of the Newtonian fluid computed at the different seventh-order tensor product Gauss quadrature points $\boldsymbol{\xi}^i$.

the number of samples is above 52. Hence, if due to the limitation of the computation resource, we are not able to run all the $4 \times 4 \times 4$ tensor product points, we can turn to the compressive sensing based method to obtain more accurate result. Also, generally speaking, with the increase of the number of samples, the accuracy of our method increases; but we should point out that this is a asymptotic property as investigated in [39, 163, 165]. Also, this example is a demonstration of the accuracy and flexibility of our method as we can use an arbitrary number of samples (as long as this number is beyond a certain threshold see [28, 38]) to construct the gPC expansion.

For relatively low dimensional problems (i.e., only 2 or 3 uncertain parameters), standard PCM methods, e.g., the sparse grid method is efficient. Hence, compressive sensing method does not

show substantial advantage. However, for high dimensional problems the benefit of employing compressive sensing method is significant, as we will demonstrate in the example in the next subsection. Furthermore, in order to improve the efficiency of the compressive sensing method, a possible way is to combine this method with special sampling strategies, e.g., see [123, 163, 165], which we will not discuss in this section.

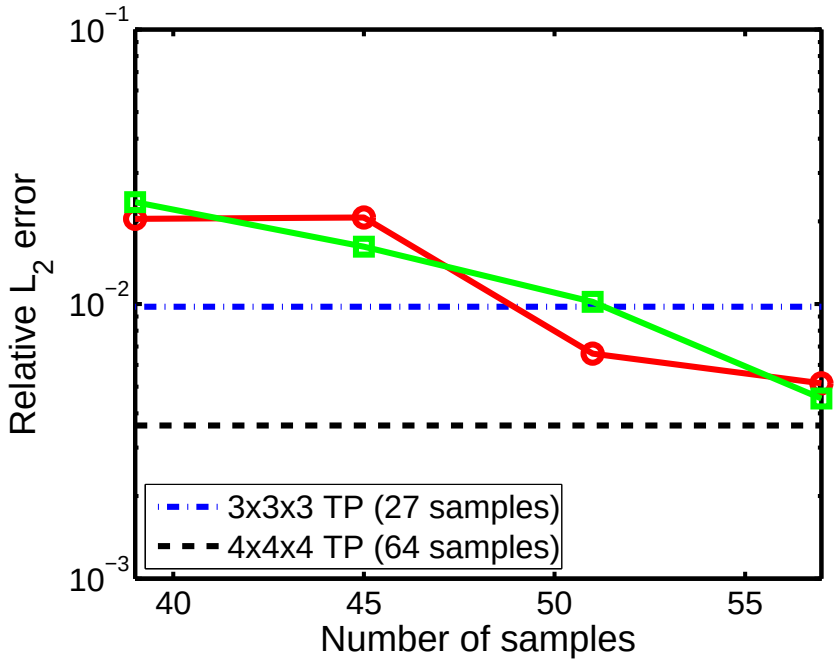


Figure 3.24: L_2 error of the response surface of shear viscosity of the Newtonian fluid. The dotted lines represent the results by $3 \times 3 \times 3$ and $4 \times 4 \times 4$ tensor product (TP) Gauss quadrature points. The symbols "◯" and "□" denote the results by OMP method of two different trials.

3.6.5 Polymer melt model

Next we consider a polymer melt system with six-dimensional parameter space and study three different cases with different parameter sets as presented in Section 3.5.2. We use the compressive sensing based method to construct a third-order gPC expansion as the surrogate response surface. The total number of basis functions is 84, i.e., the measurement matrix Ψ has $N = 84$ columns. In this example, the L_2 error defined in Eq.(3.6.9) is computed by approximating the integrals

with level 4 sparse grid method. The sparse grid method mentioned in this section is based on Clenshaw-Curtis abscissas.

Since it is impossible to visualize a 7D hyper-plane, we present the response surface of the shear viscosity by fixing part of the entries of $\boldsymbol{\xi}$ in Figure 3.25 to help understand the sensitivity of the parameters. These results are based on the simulation results of the level 4 sparse grid method. For example, in Figure 3.25(a), we set $\xi_3 = \xi_4 = \xi_5 = \xi_6 = 0$ to plot the response surface with respect to ξ_1 and ξ_2. The nearly planar response surface indicates that the momentum transport is proportional to the magnitude of the repulsive and dissipative force interactions. On the other hand, the curved response surfaces in Figure 3.25(b) and Figure 3.25(c) indicate the nonlinear dependence of the shear viscosity on the dissipative interaction profile and the force interaction cut-off range. Moreover, the response surface in Figure 3.25(d) indicates that the shear viscosity of the polymer melt system further depends on the elasticity of the polymer model. To systematically study the dependence of the shear viscosity on individual parameters, we compute the gPC expansion using the OMP method introduced previously with fewer samples and use the results by level 4 sparse grid point as reference solution. Similar procedures are followed for different physical properties.

Figure 3.26 presents the L_2 errors with the parameter set σ_*^l, i.e., the parameter set with the largest perturbation around the mean, hence the response surface is more complicated than the parameter sets σ_*^m and σ_*^s. In this figure, when the number of samples is larger than 75 we also include 46 gPC basis functions of the fourth-order polynomials. We observe that the error of our method is decreasing as more simulation results become available. For example, if only 65 simulations are affordable, our method can provide a result with accuracy about the same level of that by the level-2 sparse grid method. If 75 or more samples are available, the accuracy of our method exceeds that of the level-2 sparse grid method, which requires 85 samples. When 95 or more samples are available, our method yields more accurate results than level-3 sparse grid method. As a comparison, if we employ the level-3 sparse grid method, we need 304 additional simulation results given the 85 simulation results for the level-2 sparse grid method. Hence, our

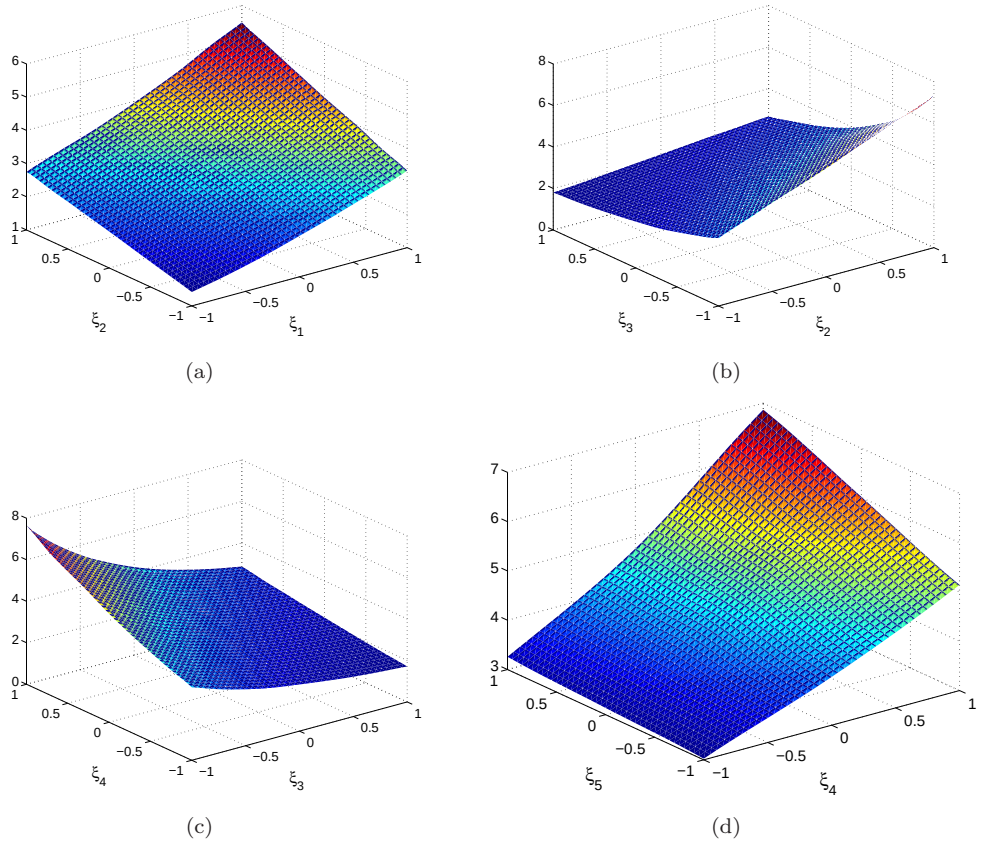


Figure 3.25: Response surface of the shear viscosity by fixing the value of the rest entries of $\boldsymbol{\xi}$. (a) $\boldsymbol{\xi}_3 = \boldsymbol{\xi}_4 = \boldsymbol{\xi}_5 = \boldsymbol{\xi}_6 = 0$, (b) $\boldsymbol{\xi}_1 = \boldsymbol{\xi}_4 = \boldsymbol{\xi}_5 = \boldsymbol{\xi}_6 = 0$, (c) $\boldsymbol{\xi}_3 = \boldsymbol{\xi}_4 = \boldsymbol{\xi}_5 = \boldsymbol{\xi}_6 = 0$, and (d) $\boldsymbol{\xi}_1 = 1.0, \boldsymbol{\xi}_2 = 0.0, \boldsymbol{\xi}_3 = 0.0, \boldsymbol{\xi}_6 = -1.0$.

method is much more flexible than the standard sparse grid method.

Figure 3.27 presents the same error analysis for the parameter sets σ_*^m and σ_*^s and we only employ third-order gPC expansions. It also illustrates the accuracy and flexibility of our method as Figure 3.26 does since the error decreases as the number of samples increases, and there is no restriction on the increment of the number of samples. Moreover, by comparing Figure 3.26 and Figure 3.27, we notice that in this model a smaller perturbation in the parameters yields smaller difference between the results by level-1 and level-2 sparse grid method. This is because the higher order basis in the gPC expansion makes smaller contributions, which in turn implies that the vector of the gPC coefficients is more sparse given a fixed number of basis functions N. Therefore, we

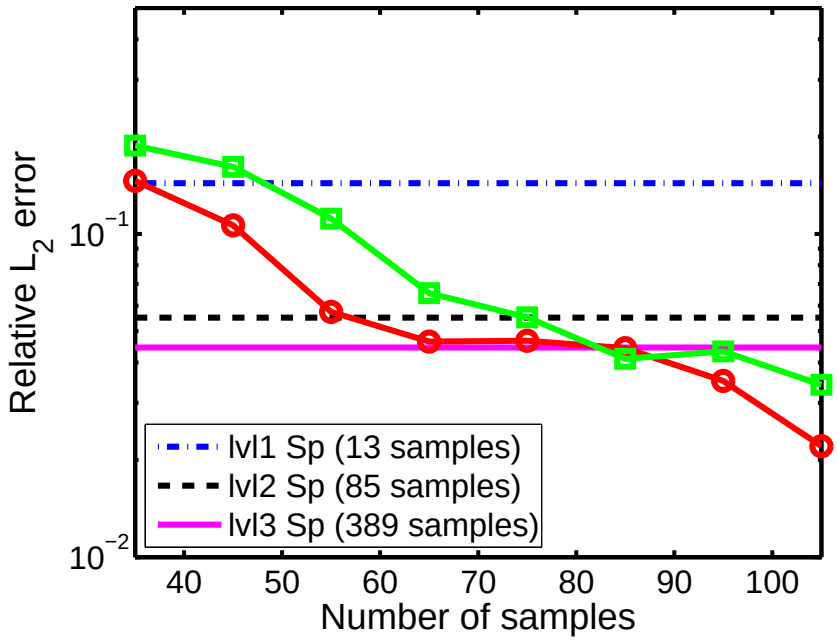
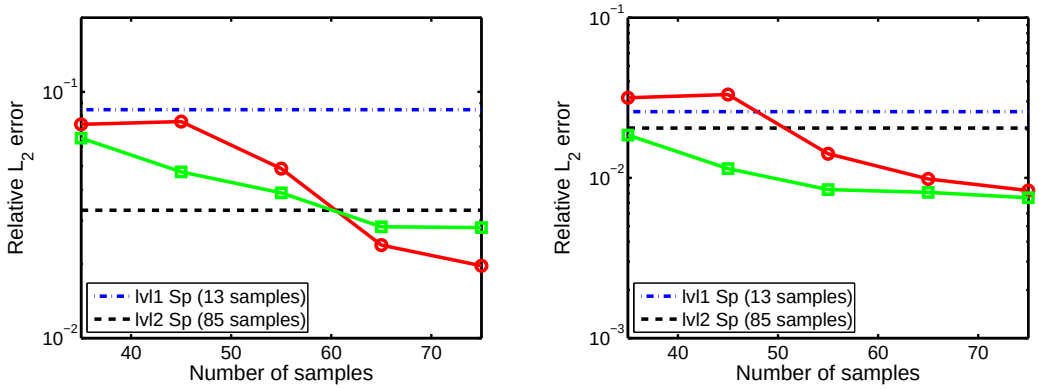


Figure 3.26: L_2 error of the *zero-shear-rate* viscosity of the polymer melt model with parameter sets σ_*^l. Results of two different trials (denoted by "◯" and "□") are presented. Results of sparse grid method ("Sp") of level 1, 2 and 3 are presented for comparisons.

observe that with 55 samples, our method exceeds the level-2 sparse grid method for the parameter set σ_*^s while this is not the case for the other two parameter sets. This comparison implies that the sparser the system is the more efficient our method is. On the other hand, in order to increase the accuracy of the gPC expansion, we may include more basis functions, which helps to reduce the truncation error and also helps to increase the sparsity as long as the higher order basis functions make small contributions. However, we cannot arbitrarily increase the number of basis function since the compressive sensing algorithm will fail if N is too large given fixed number of samples. In this section, we keep the number of basis M to be larger than $0.4N$.



(a) L_2 error with σ_*^m (b) L_2 error with σ_*^s

Figure 3.27: L_2 error of the *zero-shear-rate* viscosity of the polymer melt model with parameter sets σ_*^m (left) and σ_*^s (right). Results of two different trials (denoted by "◯" and "□") are presented. Results of sparse grid method ("Sp") of level 1 and level 2 are presented for comparisons.

3.6.6 Inference of parameters in the model

3D model

Similar to the shear viscosity of the polymer melt system discussed in Section 3.6.4 and Section 3.6.5, we can use the OMP method to construct the response surfaces of various properties of the systems, from which we are able to infer the different force parameters of the mesoscopic model. As a simple demonstration, we consider a polymer melt system with $N_b = 4$, $n = 3.0$ by fixing $(\gamma, k, r_c) = (8.0, 0.25, 1.1)$ while the parameter space (a, k_s, r_{max}) is defined by

$$\begin{aligned} a(\xi_1) &= 25.0 + \sigma_a \xi_1, \\ k_s(\xi_5) &= 60.0 + \sigma_{k_s} \xi_5, \\ r_{max}(\xi_6) &= 0.8 + \sigma_{r_{max}} \xi_6, \end{aligned} \tag{3.6.10}$$

where $(\sigma_a, \sigma_{k_s}, \sigma_{r_{max}}) = (15.0, 30.0, 0.2)$ and $\boldsymbol{\xi} = (\xi_1, \xi_5, \xi_6)$ are *i.i.d* uniform random variables on $[-1, 1]$. In this test we construct a sixth-order gPC expansion with Legendre polynomials as the surrogate model based on 54 samples of DPD simulations.

For the system defined above, we target three bulk properties: *the zero-shear-rate* viscosity

(η), the average value of radius of gyration (R_g), and the pressure (P). Here R_g of an individual polymer is defined by

$$R_g^2 = \sum_{i=1}^{N_b} (\boldsymbol{r}_i - \boldsymbol{r}_c)^2 / N_b \tag{3.6.11}$$

where $\boldsymbol{r}_i$ and $\boldsymbol{r}_c$ represent the position of individual bead i and the center of mass, respectively.

Given the (approximated) response surface of η, P and R_g, we aim to infer the parameter (a, k_s, r_{max}) with the three desirable target properties

$$\boldsymbol{P}^t = (\eta, R_g, P) = (4.457, 0.09862, 15.54).$$

In order to infer a, k_s, r_{max}, we follow the framework in [127] and employ the Bayesian theorem. In this case, $\boldsymbol{\theta} = (a, k_s, r_{max})$ and we denote $(G_1, G_2, G_3) = (\eta, R_g, P)$ as in [127]. The desirable $\boldsymbol{P}^t$ can be written as

$$\boldsymbol{G}_m = \{G_m^k\}_{k=1}^3, \qquad m = 1, 2, 3, \tag{3.6.12}$$

where G_m^k represents the k-th replica for the m-th target property. A direct Bayesian framework is employed to infer the posterior probability density:

$$\pi(\boldsymbol{\theta}|\{\boldsymbol{G}_m\}_{m=1}^3) \propto \mathcal{L}(\{\boldsymbol{G}_m\}_{m=1}^3|\boldsymbol{\theta}) q(\boldsymbol{\theta}), \tag{3.6.13}$$

where $\mathcal{L}$ is the likelihood and q is the prior. Due to the linear relation between $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$ in Eq. (3.6.10), we can equivalently consider the following posterior:

$$\pi(\boldsymbol{\xi}|\{\boldsymbol{G}_m\}_{m=1}^3) \propto \mathcal{L}(\{\boldsymbol{G}_m\}_{m=1}^3|\boldsymbol{\xi}) q(\boldsymbol{\xi}). \tag{3.6.14}$$

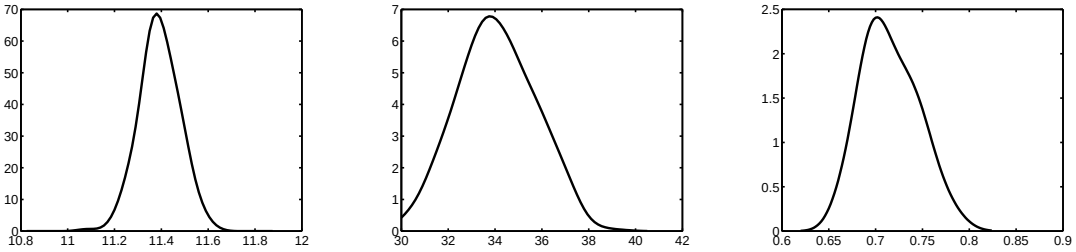
We employ the same posterior as in [127]:

$$\pi(\boldsymbol{\xi}, \tilde{\boldsymbol{\sigma}}^2 | \{\boldsymbol{G}_m\}_{m=1}^3) \propto \prod_{m=1}^{3} \prod_{k=1}^{K} \frac{\exp\left(\frac{[G_m^k - \tilde{X}_m(\boldsymbol{\xi})]^2}{2\tilde{\sigma}_m^2}\right)}{\sqrt{2\pi\tilde{\sigma}_m^2}} q(\tilde{\sigma}_m^2), \tag{3.6.15}$$

where $\tilde{X}_m(\boldsymbol{\xi})$ is the value of the surrogate model at $\boldsymbol{\xi}$ for the m-th observable, and $\{\tilde{\sigma}_m^2\}_{m=1}^3$ are hyperparameters given a Jeffreys prior of the form

$$q(\tilde{\sigma}_m^2) = \frac{1}{\tilde{\sigma}_m^2}, \qquad m = 1, 2, 3. \tag{3.6.16}$$

We use the Metropolis-Hasting Markov chain Monte Carlo (MCMC) method to sample the posterior probability density Eq. (3.6.15) by running 10^5 steps and setting burn in period as $50,000$. The posterior distribution is estimated by kernel smoothing function in MATALB. The results are presented in Figure 3.28. In this case, due to the property of the model and the selection of the macroscopic observables, all the three PDFs are single modal. Thus, these parameters can be selected by maximum *a posteriori* estimation probability (MAP) estimate based on the inferred PDFs. The set $\boldsymbol{\theta}$ we select is listed in Table 3.12. The DPD simulation results $\boldsymbol{P}^{DPD}$ and the



(a) a (b) k_s (c) r_{max}

Figure 3.28: PDFs of a, k_s and r_{max} by Bayesian inference.

Table 3.12: Inferred parameters $\boldsymbol{\theta}$ for the 3D polymer melt system.

	a	k_s	r_{max}
$\boldsymbol{\theta}$	11.35	33.60	0.696

relative errors based on this selected $\boldsymbol{\theta}$ are listed in Table 3.13. It is clear that with the inferred

Table 3.13: Validation of the inferred parameters $\boldsymbol{\theta}$ for the 3D polymer melt system. The DPD simulation results for three target properties are listed. The relative error of each target property is also listed.

	η	R_g	P
$\boldsymbol{P}^{DPD}(\boldsymbol{\theta})$	4.381	0.09893	15.40
$\lvert\boldsymbol{P}_i^t(\boldsymbol{\theta}) - \boldsymbol{P}_i^{DPD}\rvert/\lvert\boldsymbol{P}_i^t\rvert$	1.7%	0.31%	0.89%

parameters, the relative error for each target property is $\mathcal{O}(1\%)$ or smaller, which implies that our selection of $\boldsymbol{\theta}$ is good.

Remark 3.6.1. There are different approaches to infer the model parameters and here we employ a simple and direct one in our demonstration since we focus on how to construct the surrogate model fast. More sophisticated methods, e.g., adaptive MCMC [64], DRAM [63], TMCMC [10] can be applied once the surrogate models are constructed and the desirable properties are given.

6D model

We now study the full polymer melt system with six parameters discussed in Section 3.6.5 with $N_b = 5$, $n = 3.0$ and $k_BT = 1.0$, given the parameter space $(a, \gamma, k, r_c, k_s, r_{max})$ defined by

$$\begin{aligned} a(\xi_1) &= 25.0 + \sigma_a\xi_1, \\ \gamma(\xi_2) &= 8.0 + \sigma_\gamma\xi_2, \\ k(\xi_3) &= 0.25 + \sigma_k\xi_3, \\ r_c(\xi_4) &= 1.35 + \sigma_{r_c}\xi_4, \\ k_s(\xi_5) &= 50.0 + \sigma_{k_s}\xi_5, \\ r_{max}(\xi_6) &= 0.85 + \sigma_{r_{max}}\xi_6, \end{aligned} \tag{3.6.17}$$

where $(\sigma_a, \sigma_\gamma, \sigma_k, \sigma_{r_c}, \sigma_{k_s}, \sigma_{r_{max}}) = (15.0, 4.0, 0.1, 0.05, 30.0, 0.25)$ and $\boldsymbol{\xi} = (\xi_1, \xi_2, ..., \xi_6)$ are *i.i.d* uniform random variables on $[-1, 1]$. We aim to infer all six parameters in the DPD model given macroscopic observables properties. More precisely, we set $\boldsymbol{\theta} = (a, \gamma, k, r_c, k_s, r_{max})$, and target

the following six properties: viscosity at shear-rate $0.06, 0.07, 0.08$ ($\eta_{0.06}, \eta_{0.07}, \eta_{0.08}$), diffusivity ($D$), average of radius of gyration (R_g) and pressure (P). In this test we construct a surrogate model with up to third-order Legendre polynomials as well as 46 fourth-order Legendre polynomials based on 110 samples of DPD simulations. We denote $\boldsymbol{P}^t = (G_1, G_2, G_3, G_4, G_5, G_6) = (\eta_{0.06}, \eta_{0.07}, \eta_{0.08}, D, R_g, P)$, and similar to Equation (3.6.12) the replicas of the properties are written as

$$\boldsymbol{G}_m = \{G_m^k\}_{k=1}^3, \qquad m = 1, 2, 3, 4, 5, 6. \tag{3.6.18}$$

Then we employ Eq. (3.6.15) again by changing the upper range of m to 6 since we have six properties. We set the desirable target property as

$$\boldsymbol{P}^t = (29.12, 27.96, 26.88, 0.003563, 0.1726, 73.69).$$

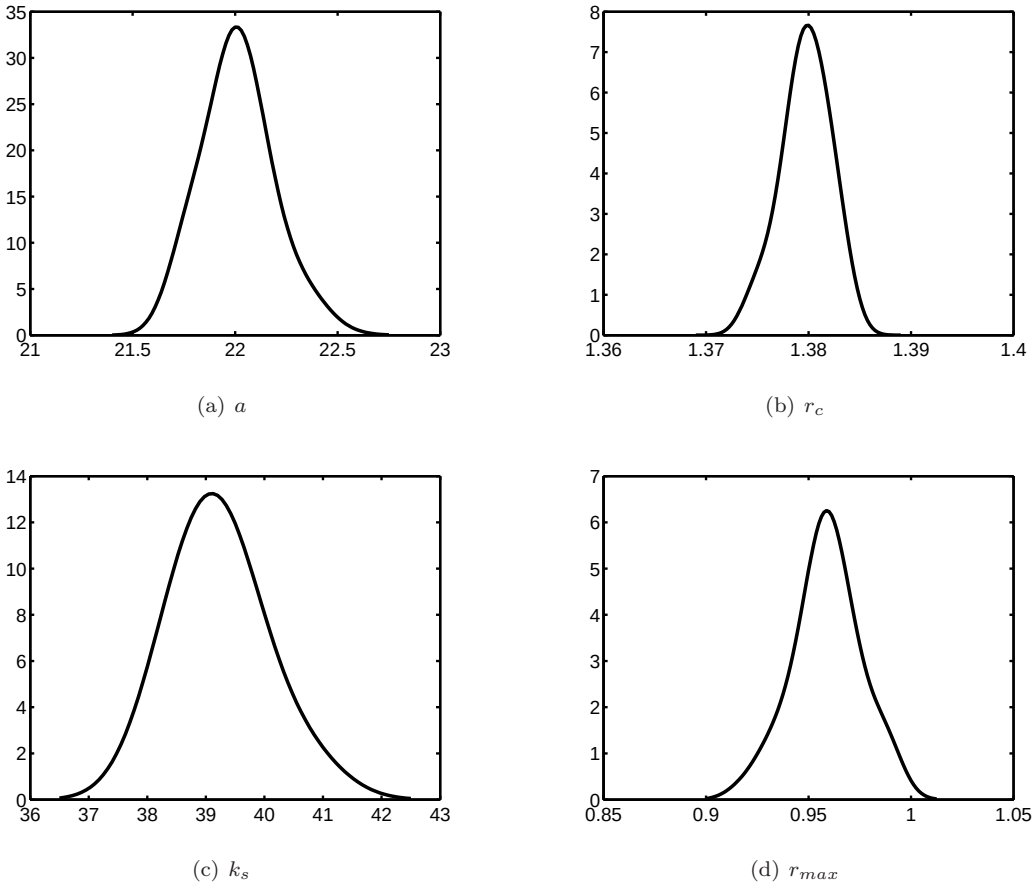
Figure 3.29 presents the inference results for $\xi_1, \xi_4, \xi_5, \xi_6$. We can see that these four parameters can be inferred by MAP. However, for ξ_2 and ξ_3, we obtain bi-modal PDFs. Hence, we plot MCMC samples of (γ, k) in Figure 3.30 to investigate the correlation between these two parameters. This plot reveals a strong correlation between γ and k. We also obtained similar plots to check the possible correlation between other pairs of parameters (not presented here) but we did not observe similar correlations.

Therefore, in order to verify the inference results, we select two parameter sets for test (see Table 3.14). The locations of (γ, k) are presented in Figure 3.30 with square points. Notice that here we

Table 3.14: Two sets of inferred parameters $\boldsymbol{\theta}$ for the 6D polymer melt system.

	a	γ	k	r_c	k_s	r_{max}
$\boldsymbol{\theta}^1$	22	5.76	0.20	1.38	39.2	0.9575
$\boldsymbol{\theta}^2$	22	6.20	0.22	1.38	39.2	0.9575

fixed $\xi_1, \xi_4, \xi_5, \xi_6$ and select two different sets of (ξ_2, ξ_3) according to Figure 3.30. We obtain two sets of target properties from the DPD simulation as presented in Table 3.15. The relative error of



(a) a (b) r_c

(c) k_s (d) r_{max}

Figure 3.29: PDFs of a, r_c, k_s and r_{max} by Bayesian inference.

each target property is also listed. We observe that the relative error is $\mathcal{O}(1\%)$ or smaller, hence, the selections of both parameter sets are good. This implies that in our DPD model we only need to keep γ or k and use the correlation revealed in Figure 3.30 to set the other parameter. Thus, we achieve a model reduction by decreasing the degree of freedom of the model by one. In other words, the Bayesian inference result implies that, in this polymer melt system, we only need five parameters in our DPD model to capture the six target properties we need.

Remark 3.6.2. In Table 3.15, we notice that the relative errors of different bulk properties varies. For example, the error of the pressure is smaller than other bulk properties. This is because: (1) the thermal noise has very little impact on the pressure; (2) the surrogate model for the pressure is more accurate than other properties due to the sparsity of its gPC coefficients; (3) the pressure

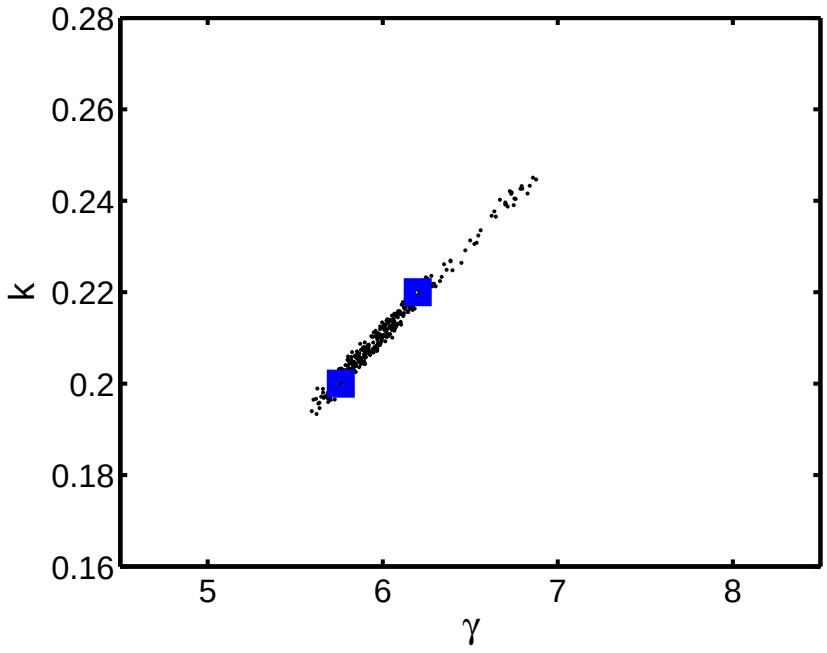


Figure 3.30: MCMC samples of (γ, k). The two square points are specified in Table 3.14.

Table 3.15: Validation of the inferred parameters $\boldsymbol{\theta}^1$ and $\boldsymbol{\theta}^2$ for the 6D polymer melt system. The DPD simulation results for six target properties with different $\boldsymbol{\theta}$ are listed. The relative error of each target property is also listed.

	$\eta_{0.06}$	$\eta_{0.07}$	$\eta_{0.08}$	D	R_g	P
$\boldsymbol{P}^{DPD}(\boldsymbol{\theta}^1)$	29.38	28.05	26.83	0.00358	0.1722	73.65
$\lvert\boldsymbol{P}_i^t(\boldsymbol{\theta}^1) - \boldsymbol{P}_i^{DPD}\rvert/\lvert\boldsymbol{P}_i^t\rvert$	0.89%	0.32%	0.19%	0.48%	0.23%	0.05%
$\boldsymbol{P}^{DPD}(\boldsymbol{\theta}^2)$	29.47	28.13	26.90	0.00354	0.1722	73.65
$\lvert\boldsymbol{P}_i^t(\boldsymbol{\theta}^2) - \boldsymbol{P}_i^{DPD}\rvert/\lvert\boldsymbol{P}_i^t\rvert$	1.20%	0.61%	0.07%	0.65%	0.23%	0.05%

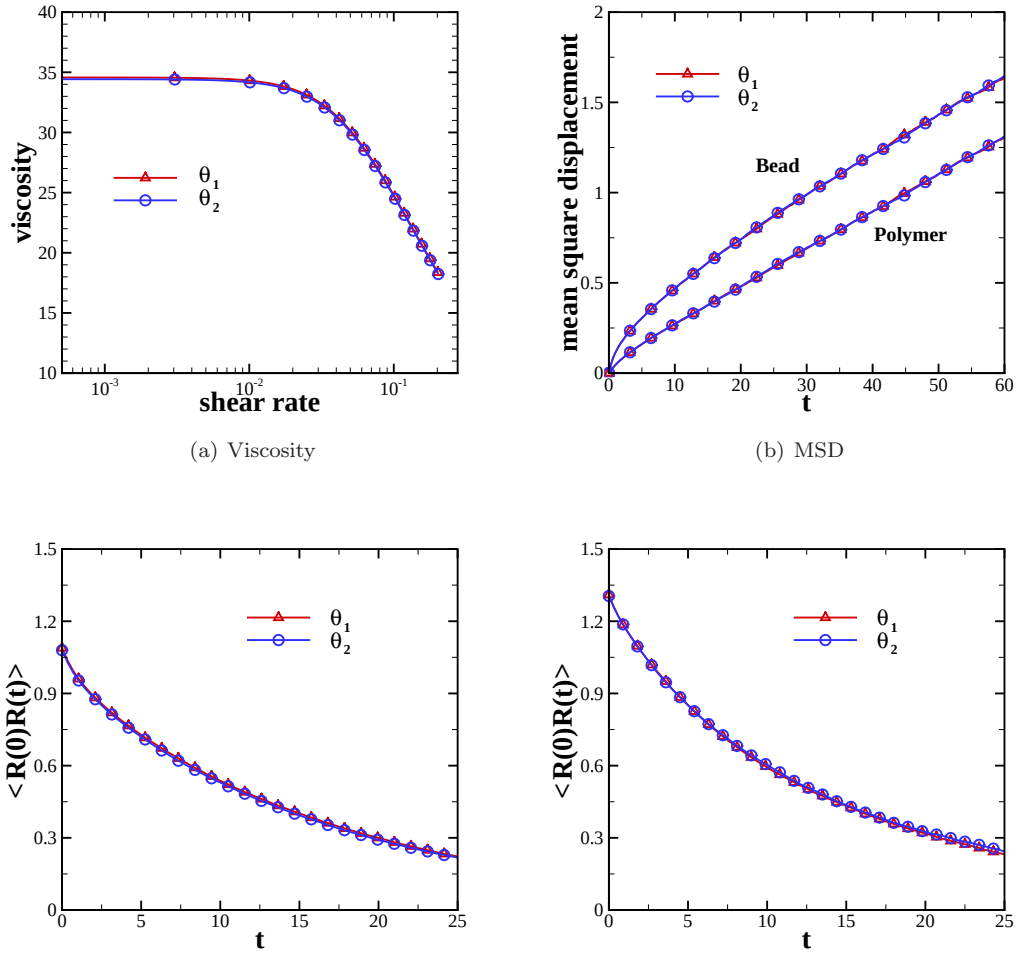
is less sensitive with respect to the parameters within the range of inferred parameter values.

Since the Bayesian inference result indicates that there exists certain parametric redundancy we were not aware of when constructing the mesoscopic model for the polymer melt system in Section 3.5, in order to further validate this hypothesis, we investigate the dynamic properties of the polymer melt system with two different parameter sets $\boldsymbol{\theta}^1$ and $\boldsymbol{\theta}^2$ in Table 3.14. We compute various dynamic properties of the polymer melt system, as shown in Figure 3.31. First, we compute the shear viscosity as a function of shear rate following the method explained in Section 3.5.3. The two response curves show good agreement within the whole shear rate regime. Next, we

consider the bulk diffusivity by computing the mean square displacement of the individual DPD bead as well as the center of mass of individual polymer. Again the two parameter sets generate consistent results. Moreover, we consider the relaxation time of individual polymer determined by the time correlation of the end-to-end vector of individual polymers, e.g., $\langle \mathbf{R}(0)\mathbf{R}(t)\rangle$, where $\mathbf{R}(t)$ represent the instantaneous end-to-end vector of an individual polymer under equilibrium state. The simulation results agree well with each other. Finally, we consider the relaxation process of the polymer melt system from a periodic opposite pressure driven flow as sketched in Figure 3.20. The initial state is obtained by applying equal but opposite body force $g = 0.2$ on individual DPD particles. At $t = 0$, we remove the body force and compute the evolution of individual polymer using $\langle \mathbf{R}(0)\mathbf{R}(t)\rangle$. Figure 3.31 validates that the two parameter sets result in the same simulation results. We have also conducted the above study for other parameter sets, generating similar consistent results. All these results validate that there exists parameter degeneracy in the mesoscopic model of the polymer melt system. Therefore, the present model can be further simplified by parameter compression according to the correlation function identified from the above analysis.

Remark 3.6.3. Based on the framework presented in this section, we may obtain several sets of parameters that are able to capture the target properties. This is not only because there may be correlations between the parameters as studied in the 6D polymer melt system, but it may also be because there are indeed several sets of suitable parameters with no correlation between them given a wide parameter confidence range. Therefore, it is possible that with a different approach to obtain the posterior in the Bayesian inference, we may obtain different sets of parameters. Hence, we can introduce more target properties to select the optimal parameter, or narrow the parameter confidence range according to empirical values, or select the one with the smallest error for the target properties we are more interested in.

To conclude, accurately recovered gPC coefficients of the target properties over the random



(a) Viscosity

(b) MSD

(c) Bulk relaxation time

(d) Poiseuille flow relaxation time

Figure 3.31: The shear viscosity, mean square displacement both individual DPD bead and center of mass of polymer (MSD), bulk relaxation time and relaxation time from reverse Poiseuille flow for the polymer melt systems with the parameter set $\boldsymbol{\theta}^1$ and $\boldsymbol{\theta}^2$ in Table 3.14.

parameter space enables us to calibrate the model parameters with respect to the observed target property values. For models with high dimensional random parameter space, the present work also provides a framework for model optimization/reduction through identification of all possible parameter degeneracies. We note that the dual effects from multiple parameters on target properties of the mesoscopic models may not be easily identified on pure theoretical modeling aspect. Systematically analysis on the intrinsic relationship between those parameters relies on the fully and

accurate access to the response surface of the target properties obtained from the present study.

We also emphasize that several factors not considered in the present work may further affect the performance of present method. First, we note that the present method, theoretically, relies on *a prior* assumption that the solution (gPC coefficients) is "sparse" in gPC basis; otherwise we may not be able to obtain accurate results by means of compressive sensing. However, this condition, in practice, is usually *not that strong* for most mesoscopic model systems. Given a system governed by regular Hamiltonian formulation, the physical properties can usually be well approximated by several orders of polynomial function over parameter space. Second, we emphasize that the parameter degeneracy and model reduction/optimization discussed in the present work depends on the specific physical properties we target to recover. For example, we find the parameter ξ_2 and ξ_3 are degenerated given the polymer melt model and conditions defined in Sec. 3.6.6. However, we note that ξ_2 and ξ_3 may not be degenerated if we specify more target parameters. For instance, if we assume the mesoscopic force field of the polymer melt system weakly depends on the polymer number density (e.g., weak many-body effect, see [86] for details discussion) and infer the parameters by targeting the dynamic properties with number density $n = 3$ and $n = 5$, ξ_2 and ξ_3 may not be degenerated. However, this result is not unexpected since the spirit of coarse-graining is to utilize *simple* formulation to recover *less* number of target properties. The more properties we target, the more parameters we need to incorporate into our model. In practice, we need to calibrate both the model parameter and the formulation according to the target properties we are interested in. Third, we consider the systems with small thermal noise in this paper. For systems with large thermal noise, we may need to modify our method by employing gPC with uncertain coefficients e.g., see [126, 127], or by including another gPC expansion to present the thermal noise. Finally, we note that the present work can not be directly applied to study systems with parameter confidence range across the phase transition regime. Special treatment is needed to take care of the abrupt discontinuity near the phase transition regime. All these issues need further study and will be addressed in the future work.

Chapter 4

Rare Events

4.1 Narrow escape problem

The narrow escape problem has gained increasing interest in biology, biophysics, cellular biology, etc, e.g., [13, 61, 72, 129]. The formulation is the following [2]: a Brownian particle in a bounded domain Ω with reflecting boundary, except for a small window through which it can escape or to be absorbed. Figure 4.1 is an example of this type of problem [120]. The narrow escape problem is that of calculating the mean escape time. This time diverges as the window shrinks, thus rendering the calculation a singular perturbation problem. The motion of the particle is described by the Smoluchowski limit of the Langevin equation:

$$\mathrm{d}\boldsymbol{X}_t = \frac{1}{\gamma}\boldsymbol{F}(x)\mathrm{d}t + \sqrt{2D}\mathrm{d}\boldsymbol{W}_t, \tag{4.1.1}$$

where $\boldsymbol{X}$ is the location of the particle, D is the diffusion coefficient of the particle, γ is the friction coefficient, $\boldsymbol{F}$ is the force and $\boldsymbol{W}_t$ is a Brownian motion. We are interested in the the mean first passage time (MFPT) of the particle. The definition of MFPT is the expectation value of the time T for the Brownian particle starting from $\boldsymbol{X}(0) = \boldsymbol{x} \in \Omega$ to reach the escaping window or the

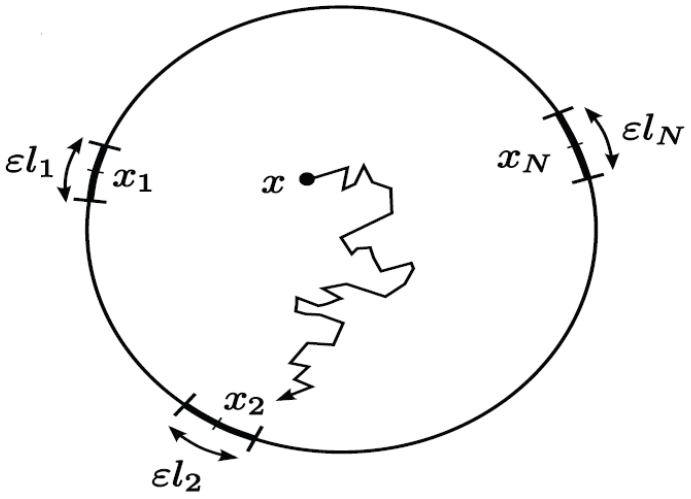


Figure 4.1: Sketch of a Brownian trajectory in the two-dimensional unit disk with absorbing windows on the boundary.

absorbing boundary somewhere in $\partial\Omega$, i.e.,

$$u(\boldsymbol{x}) = \mathbb{E}\{T|\boldsymbol{X}(0) = \boldsymbol{x}\}. \tag{4.1.2}$$

Asymptotic analysis of this problem for some special cases has been studied, e.g., [120, 31, 132, 130, 131]. For example, from [72, 130] we know that for a two-dimensional domain Ω with a smooth boundary with one small window of length $\mathcal{O}(\varepsilon)$ on its boundary, the leading-order expansion for the MFPT is

$$u(\boldsymbol{x}) = \frac{|\Omega|}{\pi D}[-\log\varepsilon + \mathcal{O}(1)]. \tag{4.1.3}$$

Notice that this leading-order result is independent of $\boldsymbol{x}$ and the location of the window on $\partial\Omega$. Here the length of each absorbing arc (see Figure 4.1) is assumed to be εl_j, where $l_j = \mathcal{O}(1)$. Also it is assumed that the windows are well separated in the sense that $|x_i - x_j| = \mathcal{O}(1)$ for $i \neq j$.

4.1.1 Problem set up

The problem we considered here is a variant of the original problem: instead of imposing a reflecting boundary, we add a potential on the domain to trap the particle as long as possible. In other word, we aim to optimize the shape of the potential such that the mean escape time of the particle is maximized. It is well known that (e.g., see [118]) the MFPT of a particle governed by Eq. (4.1.1) in domain Ω starting at points $\boldsymbol{x}$ is the solution of the following boundary value problem:

$$\begin{aligned} D\Delta u(\boldsymbol{x}) + \frac{1}{\gamma}\boldsymbol{F}(\boldsymbol{x})\cdot\nabla u(\boldsymbol{x}) = -1, &\boldsymbol{x}\in\Omega \\ u(\boldsymbol{x}) = 0, &\boldsymbol{x}\in\partial\Omega. \end{aligned} \tag{4.1.4}$$

Now that we consider imposing a potential on the domain to trap the particle, the force can be expressed as $\boldsymbol{F}(\boldsymbol{x}) = -\nabla V(\boldsymbol{x})$, where $V(\boldsymbol{x})$ is the potential we aim to optimize. For simplicity, we set $D = 1, \gamma = 1$. Hence, we consider the Dirichlet boundary condition problem in a bounded domain Ω:

$$\begin{aligned} \Delta u(\boldsymbol{x}) - \nabla V(\boldsymbol{x})\cdot\nabla u(\boldsymbol{x}) &= -1 \qquad \text{in } \Omega, \\ u(\boldsymbol{x}) &= g(\boldsymbol{x}) \qquad \text{on } \partial\Omega, \end{aligned} \tag{4.1.5}$$

where V is the potential of the field. We want to solve the following optimization problem:

$$\max_V \mathcal{J}(V) = \max_V \left\{ \max_{\boldsymbol{x}} u + \text{penalty} \right\}. \tag{4.1.6}$$

We consider two types of penalty terms:

1. Integral penalty, i.e.

$$\text{penalty} = -\lambda \int_\Omega |\nabla V|^2 \mathrm{d}\boldsymbol{x},$$

   where $\lambda > 0$ is a constant.

2. From a Bayesian point of view, consider a Gaussian random field with zero mean and covari-

ance function $\mathcal{C}(\boldsymbol{x}, \boldsymbol{y})$. For grid points $\{\boldsymbol{x}_i\}$, $\{V(\boldsymbol{x}_i)\}$ is a multivariate Gaussian with mean zero and covariance matrix $\mathsf{C}_{ij} = \mathcal{C}(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The corresponding penalty is

$$\text{penalty} = -\lambda \sum_{i,j} V(\boldsymbol{x}_i)(C^{-1})_{ij} V(\boldsymbol{x}_j), \tag{4.1.7}$$

where $\lambda > 0$ is a constant.

4.1.2 Computational Details

In this section, we use a one-dimensional problem to explain the computational details. Consider a 1D problem on $\Omega = [0, 1]$ with an integral penalty:

$$\max_V \mathcal{J}(V) = \max_V \left\{ \max_{\boldsymbol{x}} u - \lambda \int_\Omega |V'|^2 \mathrm{d}x \right\}, \tag{4.1.8}$$

where u solves the equation

$$\begin{aligned} &u''(x) - V'(x) \cdot u'(x) = -1 && \text{in } \Omega, \\ &u(x) = 0 && \text{on } \partial\Omega. \end{aligned} \tag{4.1.9}$$

Solve the BVP problem

We use collocation spectral method to Eq. (4.1.9). We denote the first and second-order derivative matrix with D^1 and D^2 respectively. Let $x_j = \cos(j\pi/N), 0 \le j \le N$ be the collocation points, $\{u_j\}_{j=0}^N$ be the values of u at these points, we have

$$\begin{aligned} &u''(x)\big|_{x=x_j} - V'(x)\big|_{x=x_j} \cdot u'(x)\big|_{x=x_j} = -1, \quad j = 1, \cdots, N-1, \\ &u_0 = 0, \quad u_N = 0. \end{aligned} \tag{4.1.10}$$

Hence we obtain a system of linear equations

$$\sum_{j=1}^{N-1} \left((D^2)_{ij} + (D^1)_{jk} V(x_k) (D^1)_{ij} \right) u_j = -1, \quad j = 1, \cdots, N-1. \tag{4.1.11}$$

Or equivalently,

$$\left(D^2 + \mathrm{diag}(D^1 \boldsymbol{V}) \right) \boldsymbol{u} = -\mathbf{1}, \tag{4.1.12}$$

where $\boldsymbol{V} = (V(x_1), V(x_2), \cdots, V(x_{N-1}))$, $\boldsymbol{u} = (u_1, u_2, \cdots, u_{N-1})$, $\mathbf{1} = (1, 1, \cdots, 1)$. Currently, the maximum is taken over all $x_j,\ j = 0, \cdots, N$. This may need modifying since usually the discrete random walk is considered on a uniform grid. In our computation, we set $N = 50$ and due to the optimization solver `SNPOT` we use, we minimize the following object function instead:

$$\min_V \mathcal{J}(V) = \min_V \left\{ -\max_{\boldsymbol{x}} u + \lambda \int_\Omega |\nabla V|^2 \mathrm{d}x \right\}, \tag{4.1.13}$$

Notice that the spectral method in this problem is sufficiently accurate, we don't need large N, hence, we can reduce that to a smaller number, e.g., 16,20,32, etc. And we can use spline interpolation to plot the graphs. In 2D tests, in order to obtain the result quickly, we set $Nx = Ny = 12$, i.e., 169 points in total. The formulation for a 2D problem is

$$\left(I \otimes D^2 + D^2 \otimes I + \mathrm{diag}(\boldsymbol{V_x}) I \otimes D^1 + \mathrm{diag}(\boldsymbol{V_y}) D^1 \otimes I \right) \boldsymbol{u} = -\mathbf{1}. \tag{4.1.14}$$

Here we stretch the matrix of the solution U to be a long vector u, $\otimes$ is the Kronecker product, $\mathrm{diag}(\boldsymbol{V_x})$ is the diagonal matrix generated from the stretched matrix V_x. For Poisson equations this is a inefficient way to solve it. However, for our problem, $\boldsymbol{V_x}$ and $\boldsymbol{V_y}$ are changing in each optimization iteration, hence we use the most brutal force way to solve the 2D BVP problem. Also remember that D^1, D^2 are parts of the corresponding derivative matrix since we consider Dirichlet problem.

Compute the integral in the penalty term

Since we use Chebyshev points we can use either Fejér (second rule) weights or Clenshaw-Curtis weights. The former does not use the two end points in the integration while the latter one uses. Let $\theta_k = k\pi/N$, and we list the weights below:

- Fejér weights:

$$w_k^F = \frac{4}{N}\sin\theta_k \sum_{j=1}^{[N/2]} \frac{\sin(2j-1)\theta_k}{2j-1}, \quad k = 0, 1, \cdots, N. \tag{4.1.15}$$

- Clenshaw-Curtis weights:

$$w_k^{CC} = \frac{c_k}{N}\left(1 - \sum_{j=1}^{[N/2]} \frac{b_j}{4j^2-1}\cos(2j\theta_k)\right), \quad k = 0, 1, \cdots, N, \tag{4.1.16}$$

  where the coefficients b_j, c_j are defined as

$$b_j = \begin{cases} 1, & j = n/2 \\ \\ 2, & j < n/2 \end{cases}, \qquad c_k = \begin{cases} 1, & k = 0 \text{ or } n \\ \\ 2, & \text{otherwise} \end{cases}. \tag{4.1.17}$$

In practice, we use Clenshaw-Curtis weights to approximate the integral. In 2D tests, we employ tensor product rule to generate collocation points as well as the corresponding weights.

Solve the optimization problem

We use `SNOPT` [4] to solve the optimization problem. The penalty term is necessary otherwise we will obtain irregular solution. When λ is small, e.g., for the integral penalty with $\lambda < 0.075$ we will also see very huge $\max u$ and irregular V'. In the optimization solver, we do not use the gradient of the object function yet. The result will be affected by the magnitude of the penalty. The initial guess is $V(x) = -0.01(\cos^2(\pi x) - 1)$.

Notice that since the penalty only depends on the gradient of V, for this case we only consider

inner points in the optimization problem. For example, for 1D test, if the spatial points are $x_0, x_1, \cdots, x_N$, then $x_0 = 0$ is fixed and we only have N variables in the optimization problem. If the penalty is the covariance kernel prior function, since all the x_i are included in the penalty, hence they are all considered as variables in the optimization problem.

We select several kernel functions to generate Gaussian random fields from [6] for the second type of test, i.e., when $V(x)$ is a Gaussian random field.

4.1.3 Numerical results 1D

In this section we present the results for 1D problem. N is set to be 50 in these tests. As is mentioned before, smaller number is sufficient.

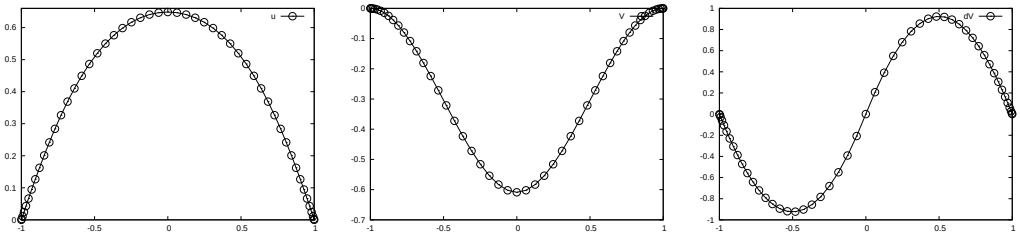
Integral penalty

We present the result by employing integral as the penalty, i.e., Eq. (4.1.8). In Figure 4.2 we present the value of u, V, V' with $N = 50$ and λ varying from 0.1 to 100. We notice that when λ increases from 1 to 100, $\max u$ changes slightly, and both V and V' decrease with the same order as λ increases, namely, V and V' shrink to 10% as λ increases to 1000%. We can observe that $\max u$ converge to 0.5 as we keep increasing λ. This is because as λ increases, the penalty term is decreasing and V' reduces to 0, hence the solution of (4.1.8) converges to the solution of

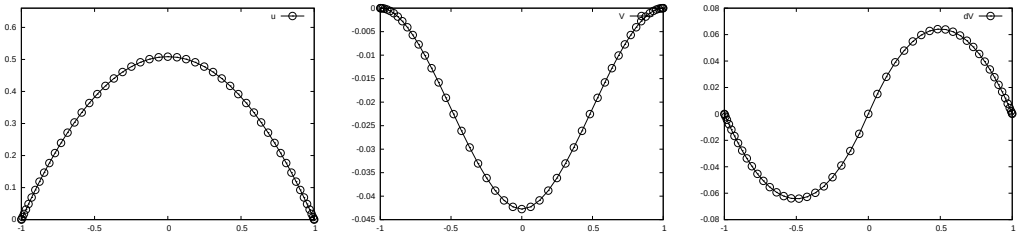
$$\begin{aligned} u''(x) &= -1, \qquad x \in \Omega, \\ u(x) &= 0, \qquad x \in \partial\Omega. \end{aligned} \tag{4.1.18}$$

which is the case when no potential is imposed on the domain Ω. The solution is simply

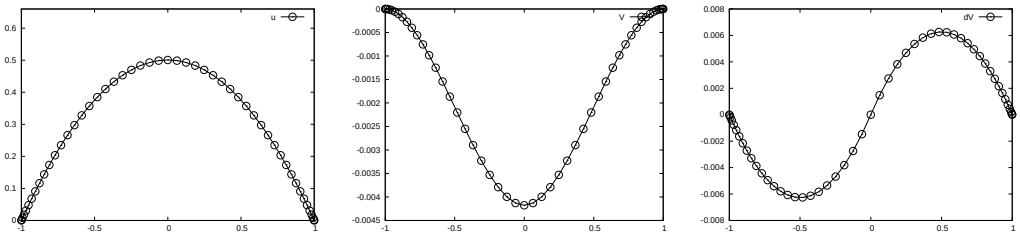
$$u(x) = -\frac{1}{2}(x^2 - x).$$



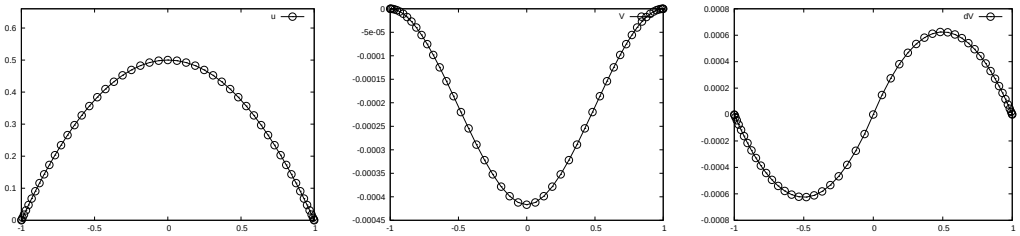
(a) $u(\lambda = 0.1)$ (b) $V(\lambda = 0.1)$ (c) $V'(\lambda = 0.1)$



(d) $u(\lambda = 1.0)$ (e) $V(\lambda = 1.0)$ (f) $V'(\lambda = 1.0)$



(g) $u(\lambda = 10.0)$ (h) $V(\lambda = 10.0)$ (i) $V'(\lambda = 10.0)$



(j) $u(\lambda = 100.0)$ (k) $V(\lambda = 100.0)$ (l) $V'(\lambda = 100.0)$

Figure 4.2: 1D case: u, V and V' for different λ with integral penalty.

In Figure 4.3, the potential V with $\lambda = 1, 10, 100$ are compared with rescaled (and shifted) cosine function. We can observe that the potentials for $\lambda = 10$ and 100 are very close to the cosine function. Here the rescaling is accomplished by fitting the peak and valley of the curve with the cosine function. Hence, it is very probable that the optimal potential is a cosine function. Also, we present $\max u - 0.5$ and $\lambda \int_\Omega |\nabla V|^2 \mathrm{d}x$ in Figure 4.4, which implies that both the "convergence

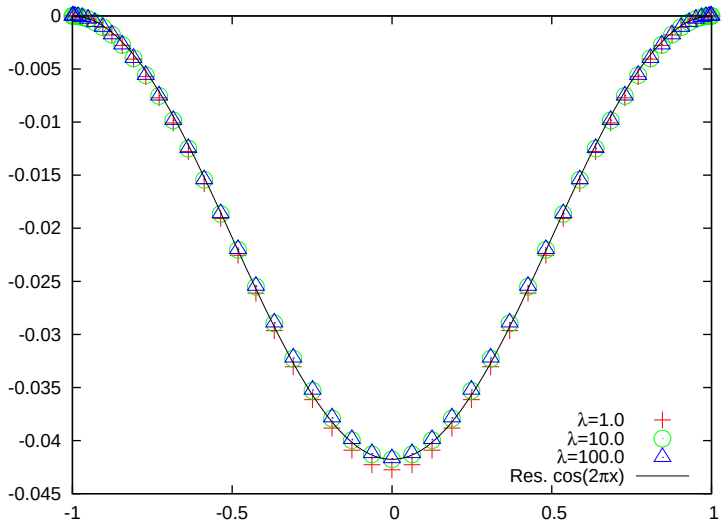
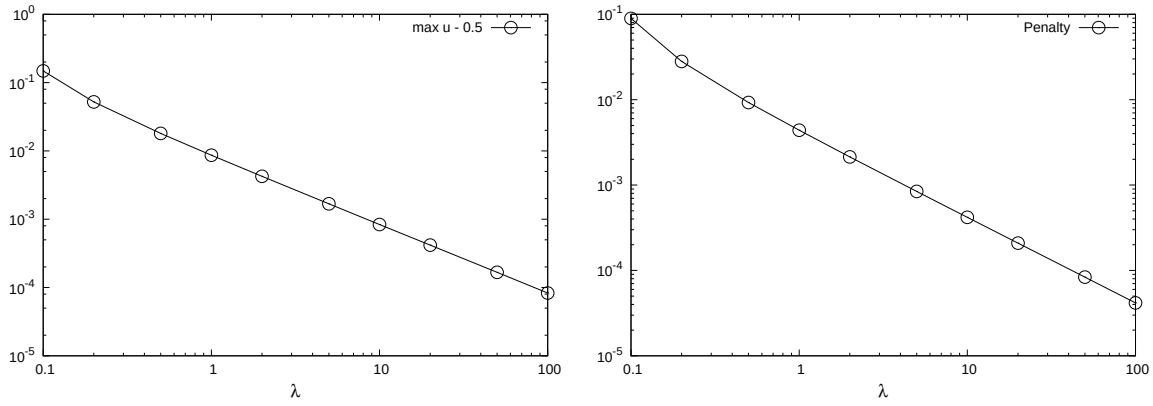


Figure 4.3: Comparison of potential V for different λ and rescaled cosine function.

rate" of $\max u$ and the penalty are 1 with respect to λ. Based on the above observation, we adopt the results with $\lambda = 0.1$ as the optimal shape of $V(x)$.



(a) $\max u - 0.5$ (b) Penalty

Figure 4.4: Change of $\max u - 0.5$ (left) and penalty $\lambda \int_\Omega |\nabla V|^2 \mathrm{d}x$ (right) with respect to λ.

Gaussian random field penalty

We present the results of employing several different covariance matrix bellow. For simplicity, we consider the isotropic Gaussian random fields, i.e., the covariance function depends only on the

distance of two points:

$$\mathcal{C}(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{C}(\tau), \tau = \|\boldsymbol{x} - \boldsymbol{y}\|. \tag{4.1.19}$$

In this case, we use the *correlation function* $\rho(\tau)$ to denote covariance function $\mathcal{C}(\tau)$ as in [6].

Spherical covariance

The correlation function of 1D sphere is

$$\rho(\tau, R) = 1 - \tau/R. \tag{4.1.20}$$

Here we set $R = 2r = 2$ since our computational domain is a 1D ball $B_1(0)$. u, V, V' with different λ are presented in Figure 4.5. In fact these results are very close to those with integral penalty term. We neglect the comparison between V and cosine function here. The optimal shape can be chosen as $V(x)$ with $\lambda = 0.1$.

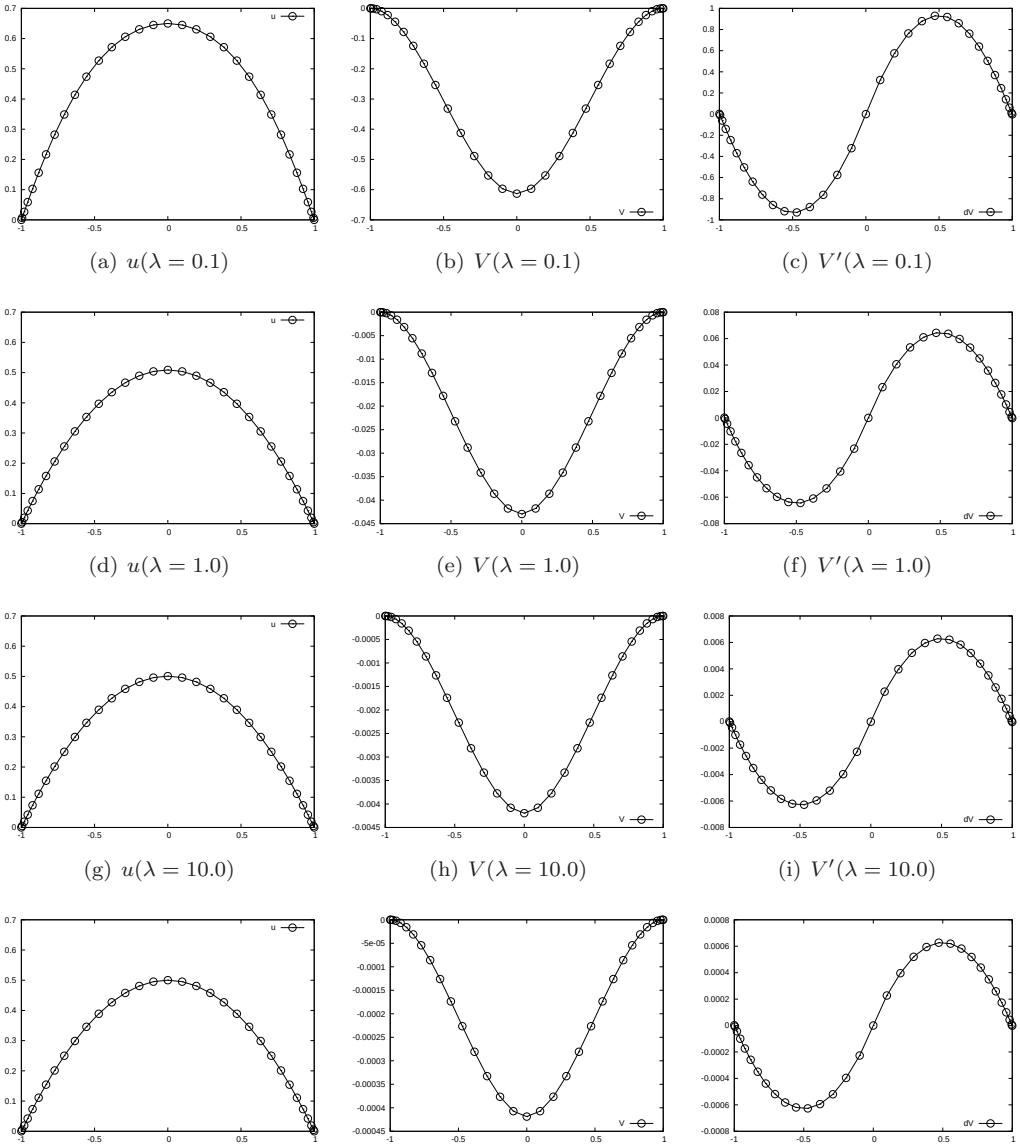
Exponential covariance

The 1D exponential correlation function is

$$\rho(\tau) = \mathrm{e}^{-3(\tau/R)^{\nu}}. \tag{4.1.21}$$

Here we set $R = 1$. Results for $\nu = 1$ are presented in Figure 4.6. The shape of u are still the same as the above two examples. Also, as λ increases, the maximum of u converges to 0.5. Although the potential is different in this case, the change of magnitude of V and V' is still proportional to the change of λ. However, the shape of V is different from the previous two cases. The peaks of curve $V(x)$ are not on the boundary of the domain but are in it. The optimal shape can be chosen as $V(x)$ with $\lambda = 0.1$.

Next we fix $\lambda = 1.0$ and test different ν. In Figure 4.7 we observe that the value of u are similar and it increases as ν increases. For the potential V, the magnitudes of the gradients at the two boundary points decreases as ν increases. We list V' for $\nu \geq 1.0$ for comparison. We can observe



(a) $u(\lambda = 0.1)$ (b) $V(\lambda = 0.1)$ (c) $V'(\lambda = 0.1)$

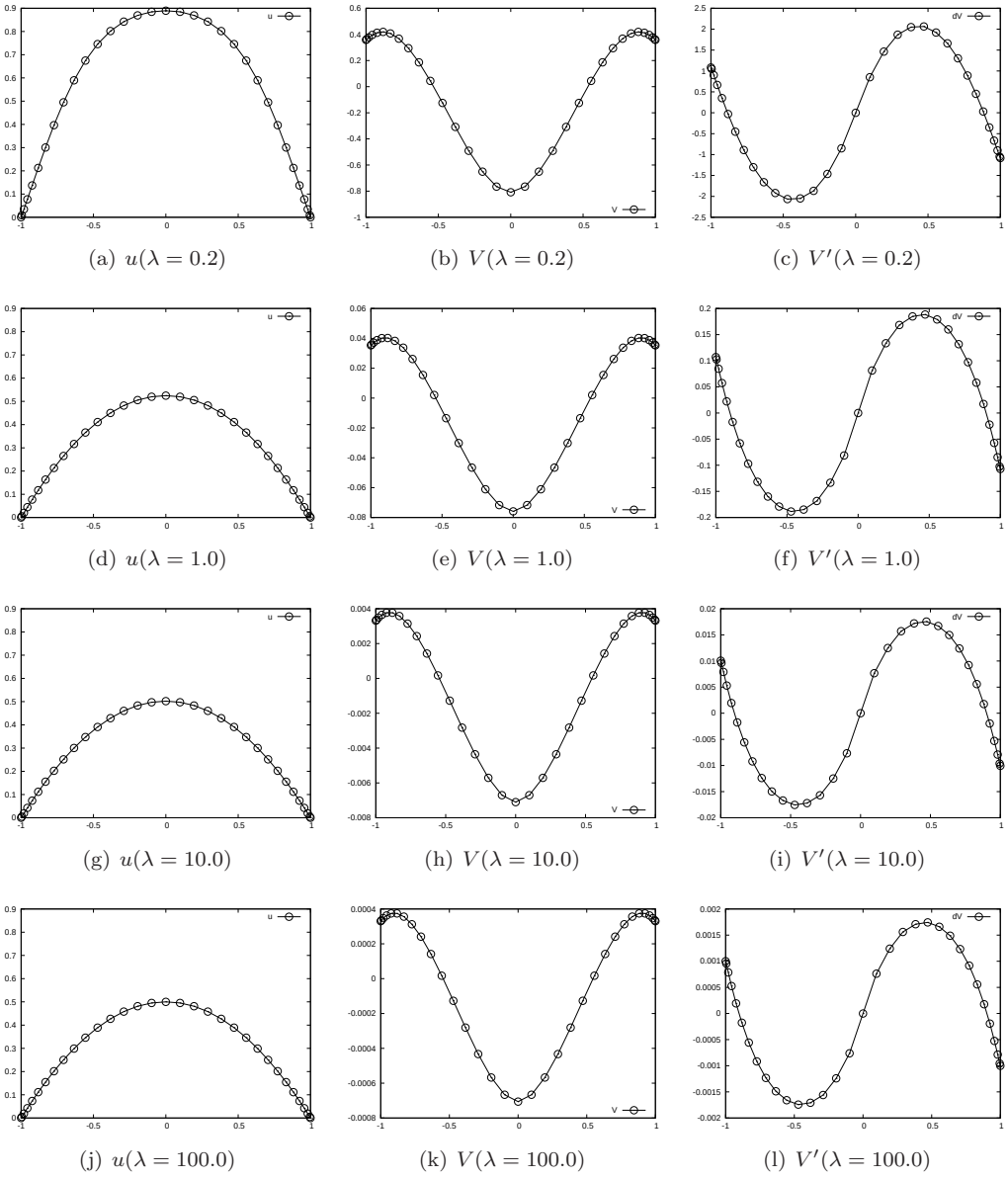
(d) $u(\lambda = 1.0)$ (e) $V(\lambda = 1.0)$ (f) $V'(\lambda = 1.0)$

(g) $u(\lambda = 10.0)$ (h) $V(\lambda = 10.0)$ (i) $V'(\lambda = 10.0)$

(j) $u(\lambda = 100.0)$ (k) $V(\lambda = 100.0)$ (l) $V'(\lambda = 100.0)$

Figure 4.5: 1D case: u, V and V' for different λ with 1D spherical kernel $\rho(\tau) = 1 - \tau/R$,$R = 2$.

that V' becomes less steep near the boundary as ν increases. This can be related to following property of the Gaussian random field with exponential covariance : the larger the ν, the smoother the random field.



(a) $u(\lambda = 0.2)$ (b) $V(\lambda = 0.2)$ (c) $V'(\lambda = 0.2)$

(d) $u(\lambda = 1.0)$ (e) $V(\lambda = 1.0)$ (f) $V'(\lambda = 1.0)$

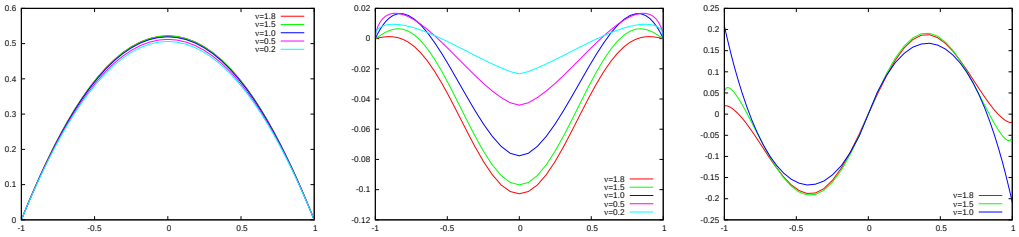
(g) $u(\lambda = 10.0)$ (h) $V(\lambda = 10.0)$ (i) $V'(\lambda = 10.0)$

(j) $u(\lambda = 100.0)$ (k) $V(\lambda = 100.0)$ (l) $V'(\lambda = 100.0)$

Figure 4.6: 1D case : u, V and V' for different λ with exponential kernel $\rho(\tau) = \mathrm{e}^{-3(\tau/R)^{\nu}}, \nu = 1.0, R = 1.0$.

4.1.4 Numerical results 2D

In this section, we present results for 2D tests. The computational domain is $\Omega = [0, 1] \times [0, 1]$. We set $Nx = Ny = 16$, where $Nx + 1$ is the number of Chebyshev collocation points on x-axis and N_y is the number of Chebyshev collocation points on y-axis. Tensor product rule is employed to



(a) u (b) V (c) V'

Figure 4.7: 1D case: u, V and V' for different ν with $\lambda = 1.0$ and exponential kernel $\rho(\tau) = \mathrm{e}^{-3(\tau/R)^{\nu}}$, $R = 1$.

generate 2D collocation points.

Integral penalty

We first present the results of with the integral penalty in Figures 4.8-4.11. The shapes of $V(\boldsymbol{x})$ with different λ are very similar while the magnitudes of $V(\boldsymbol{x})$ and $\nabla V(\boldsymbol{x})$ are different. The selection of $\lambda = 0.1$ is sufficient to provide a regular solution for this case. The optimal choice of the potential can be as that in 4.8(b).

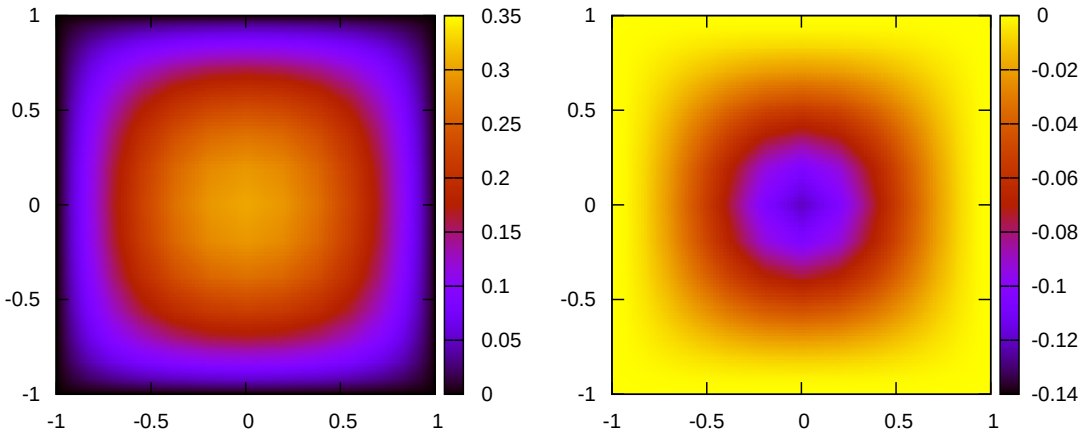
Gaussian random field penalty

In this subsection we only present the results with exponential correlation function since the spherical correlation function corresponds to the case with the domain being a disk. The 2D exponential correlation function is given as:

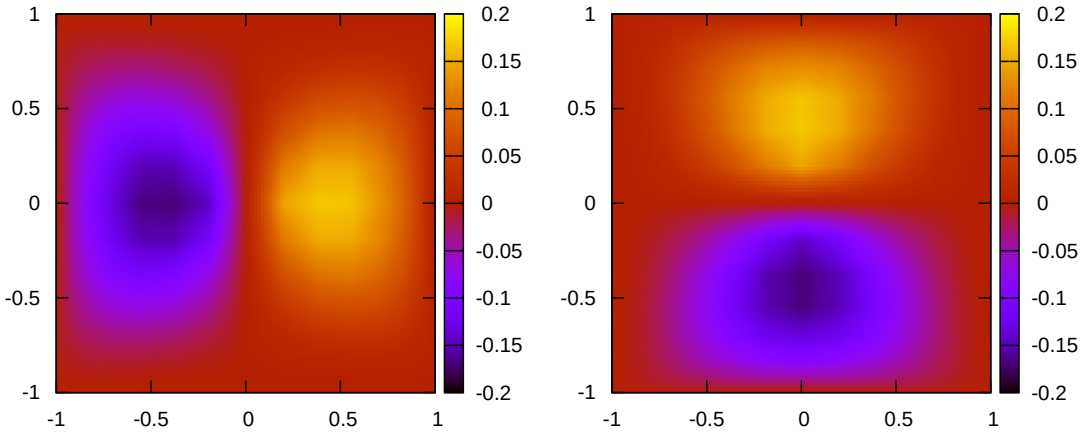
$$\rho(\tau, R) = \mathrm{e}^{-3(\tau/R)^{\nu}}. \tag{4.1.22}$$

Here we set $R = 1, \nu = 1$. The results in Figure 4.12-4.14(where $\lambda = 0.1, 1.0, 10.0$ respectively) imply that the shape of V is similar for different λ as we see in the 1D case. The only difference is the magnitude of the solution $V(x)$ and $\nabla V(x)$. In this case, we choose the results of $\lambda = 0.1$ as the optimal potential.

Now we fix $\lambda = 1.0$ and see the effect of ν in the exponential kernel. The results in Figure 4.15-4.18 imply that larger ν yield less steep $\nabla V(\boldsymbol{x})$ near the center of the domain.



(a) u (b) V

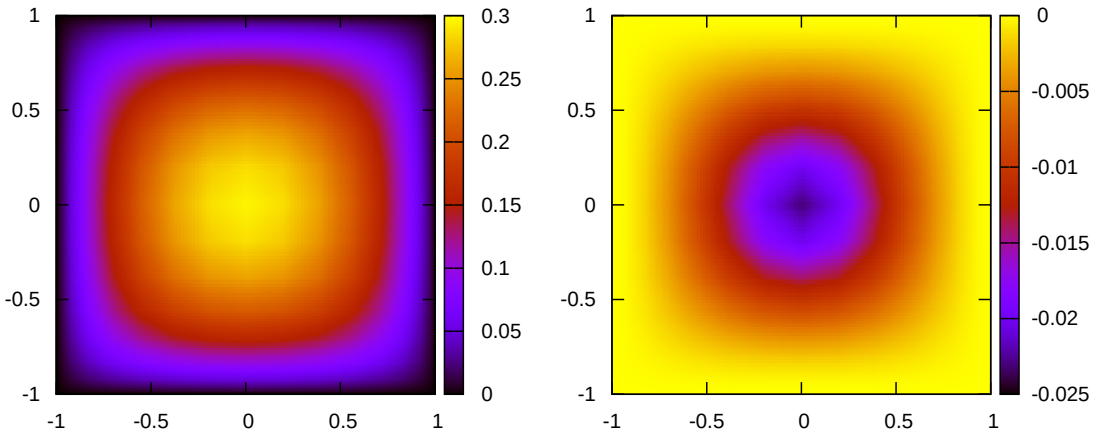


(c) V_x (d) V_y

Figure 4.8: u, V and ∇V for $\lambda = 0.1$ with integral penalty.

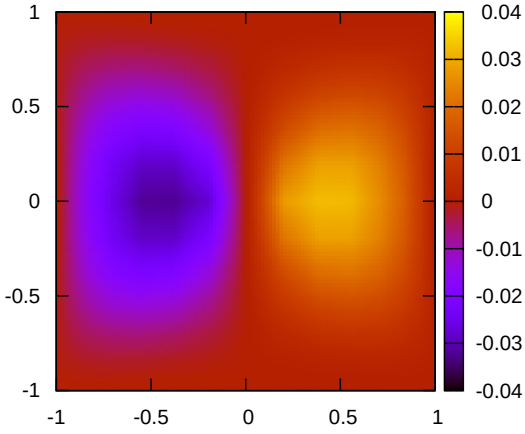
4.2 Stochastic domain decomposition

Schwarz method [78, 145, 15, 143, 160, 161] for parallel domain decomposition of deterministic problems are well-established and have become popular over the last decades. New algorithms based on more effective transmission of information on the interfaces of subdomains are also proposed, e.g., [54, 55, 56, 65]. However, if the governing equation in some subdomains is not deterministic, or if different sources of uncertainty exists in different subdomains, the classical methods is not effective to solve the problem. The most natural approach to apply the classical domain decomposition skill is to first generate sampling points (e.g., based on Monte Carlo method, sparse grid method, etc.)

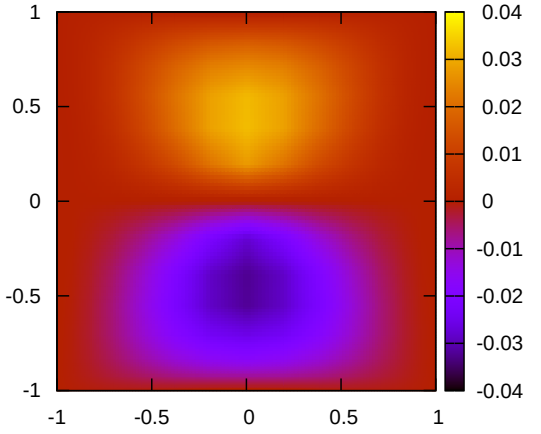


(a) u

(b) V



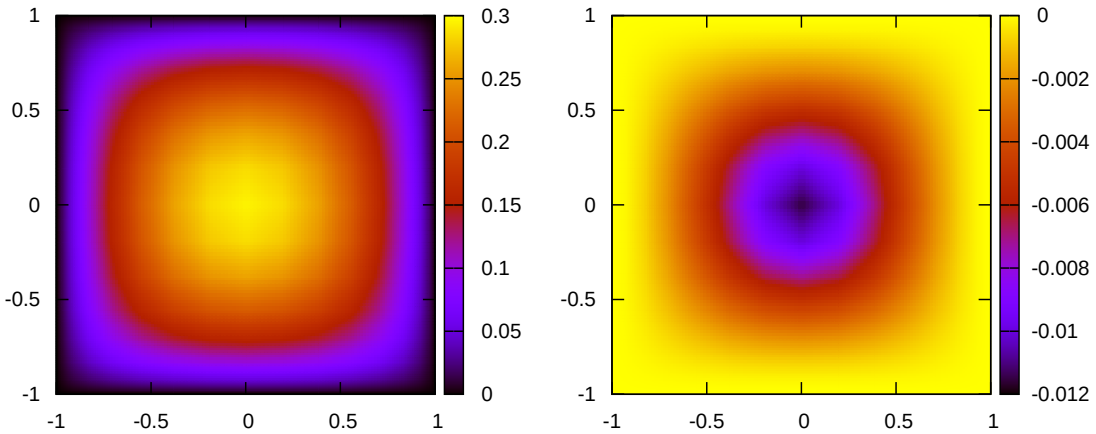
(c) V_x



(d) V_y

Figure 4.9: u, V and ∇V for $\lambda = 0.5$ with integral penalty.

of the system, then for each sampling point, employ the Schwarz method for deterministic problem to solve it. After the solution of each sampling points are collected, some post processing steps are implemented to obtain the estimate of the statistics or other physical properties of the system e.g., [99, 167]. Alternatively, another type of method based on the popular UQ framework is to first implement a global Galerkin projection based on gPC or its variant, then decompose the global linear algebra problem into subproblems and solve them them, e.g., [128, 76]. This type of method relies on the numerical linear algebra skills, e.g., preconditioning, Schur complement, etc. These methods construct global deterministic problems first then use the existing skills to decompose the

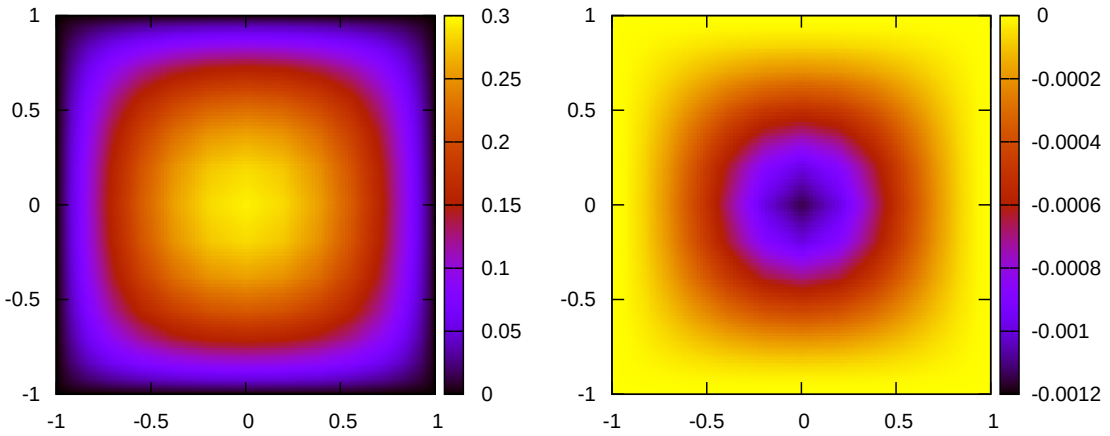


(a) u (b) V

(c) V_x (d) V_y

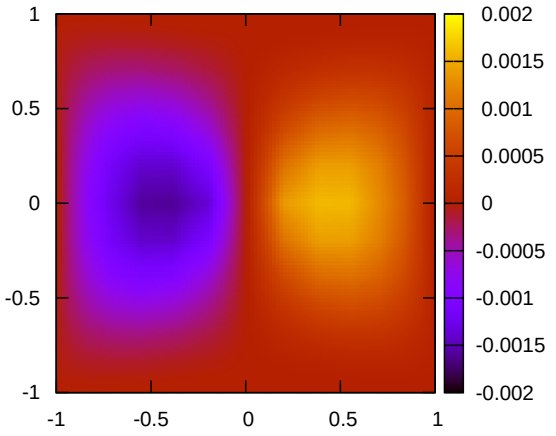
Figure 4.10: u, V and ∇V for $\lambda = 1.0$ with integral penalty.

deterministic problem into subproblems and solve them. The main limitation of these methods is that the constructed deterministic problems can be very costly to solve. For example, for the non-intrusive method, if the dimension of the random space is high, large number of advanced sampling points (e.g., sparse grid points) are needed to obtain sufficiently accurate results due to the curse of high-dimensionality or Monte Carlo sampling points have to be used. For the intrusive method, the linear algebra system obtained from the Galerkin projection is huge due to the number of basis functions needed in the expansion of the solution in the random space. Another concern about the intrusive method is that it is hard to be extended to more general models. In other words,

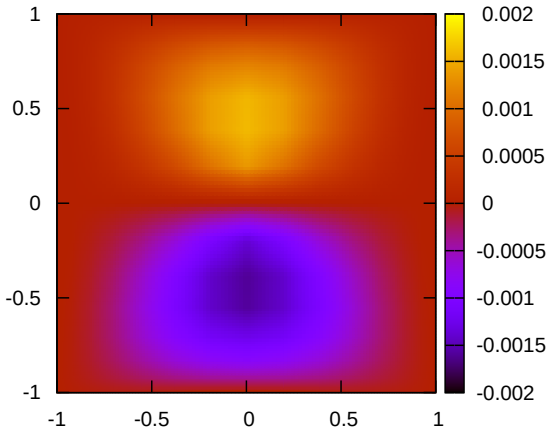


(a) u

(b) V



(c) V_x

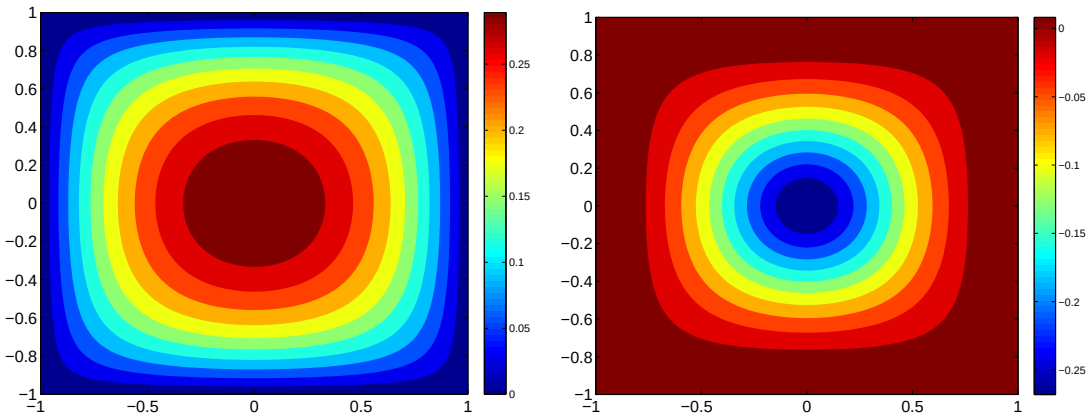


(d) V_y

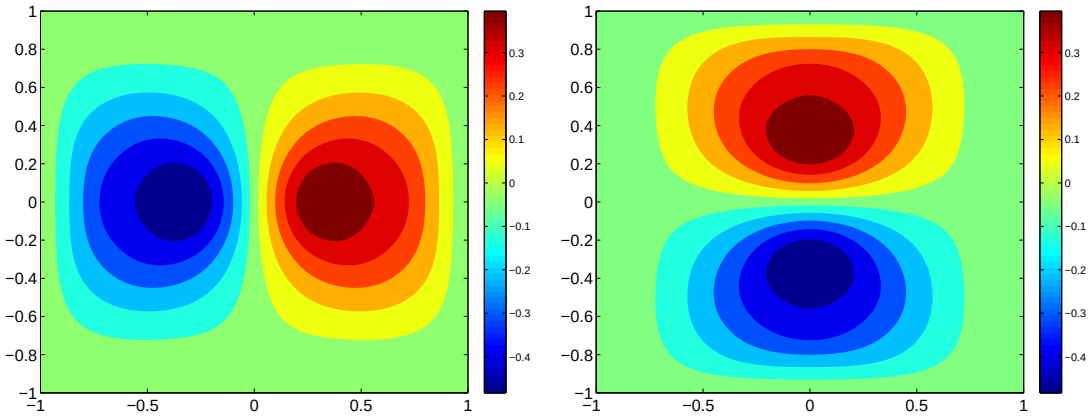
Figure 4.11: u, V and ∇V for $\lambda = 10.0$ with integral penalty.

the Galerkin projection is mainly used for solving PDEs or stochastic PDEs. However, in many applications, the physical system is simulated by particle method, e.g., MD, DPD, SDPD, etc. Hence, it is impossible or very hard to achieve the Galerkin projection or other intrusive method. Therefore we aim to propose a new framework, which is non-intrusive and does not require to construct a huge deterministic system.

Another motivation to consider the domain decomposition method for stochastic problems is that a general way to represent a random field is to use KL expansion. Take a 1D domain for example, given the correlation length for a random field, we need more eigvenfunctions in the KL



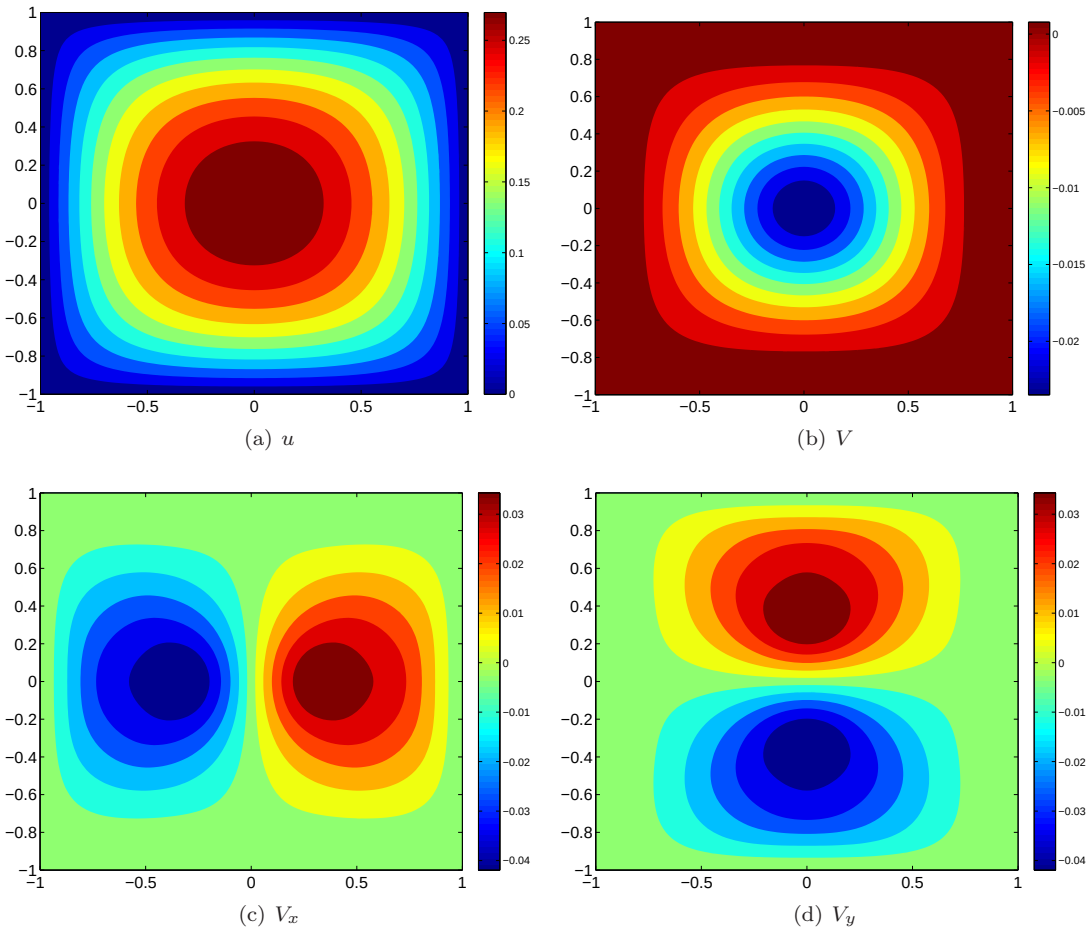
(a) u (b) V



(c) V_x (d) V_y

Figure 4.12: u, V and ∇V for different $\lambda = 0.1$ with $\nu = 1$ in 2D exponential kernel.

expansion for larger domain because the correlation length is smaller with respect to the length of the domain. Hence if we project the global random field into subdomains, we can obtain low-dimensional representation of the random field in each subdomain. Notice that the construction of low-dimensional representation of the random field should be treated carefully. It does not mean that we only need to compute eigenvalues and eigenfunction for the KL expansion within subdomains. Because the correlation between the points in different subdomains will be eliminated, hence some information of the global random field is lost. Therefore, unlike the deterministic domain decomposition problem, the first problem we are face to is to present the problem in each subdomain! This problem is solved in one of our ongoing work to some extend. The next problem

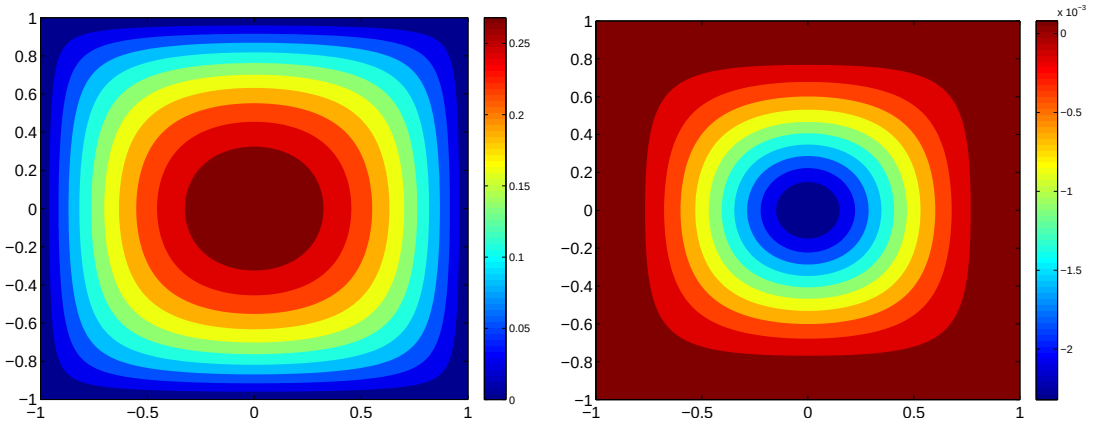


(a) u (b) V

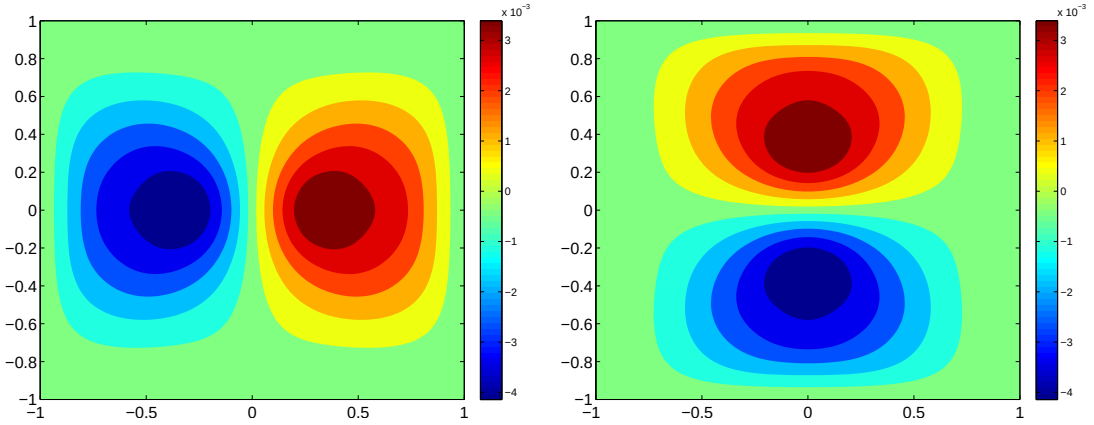
(c) V_x (d) V_y

Figure 4.13: u, V and ∇V for different $\lambda = 1.0$ with $\nu = 1$ in 2D exponential kernel.

is the information to be transferred through the interface of subdomains. This is the most critical part of the stochastic domain decomposition method. Unlike the deterministic cases, where value of the solution (for Dirichlet boundary condition), derivative of the solution (for Neumann boundary condition) or the combination of value and derivative of the solution (for the Robin boundary condition) is passed across the interface, we need to transfer sufficient stochastic/statistical information through the boundary to guarantee the accuracy. However, this step is extremely hard and can even be not achievable in some cases. For example, consider a case in which we apply MD simulation in a subdomain and solve Navier-Stokes equation in the adjacent domain. It is easier to extract information from MD simulation and pass it to the Navier-Stokes solver. This

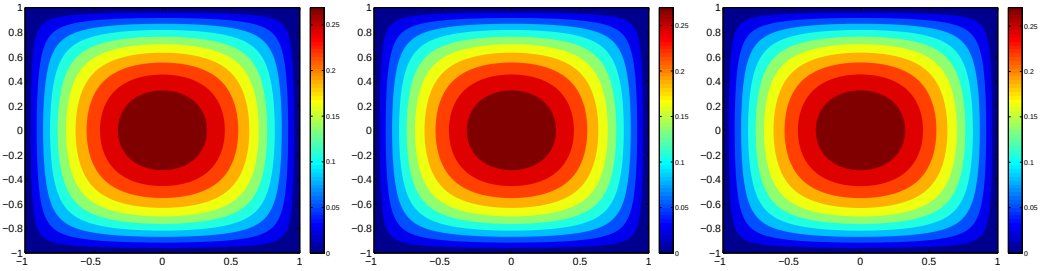


(a) u (b) V



(c) V_x (d) V_y

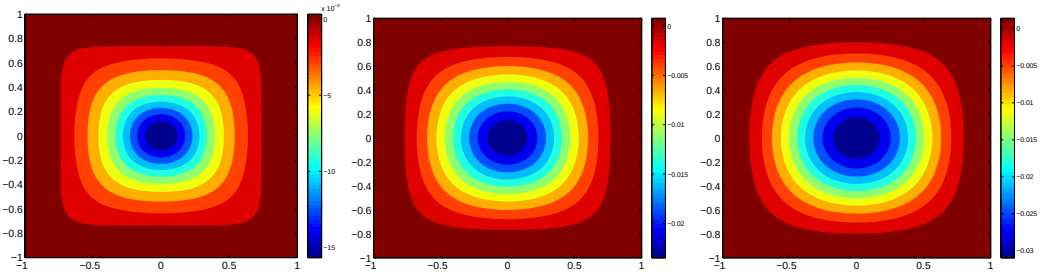
Figure 4.14: u, V and ∇V for different λ with $\nu = 1$ in 2D exponential kernel.



(a) $u(\nu = 0.5)$ (b) $u(\nu = 1.0)$ (c) $u(\nu = 1.5)$

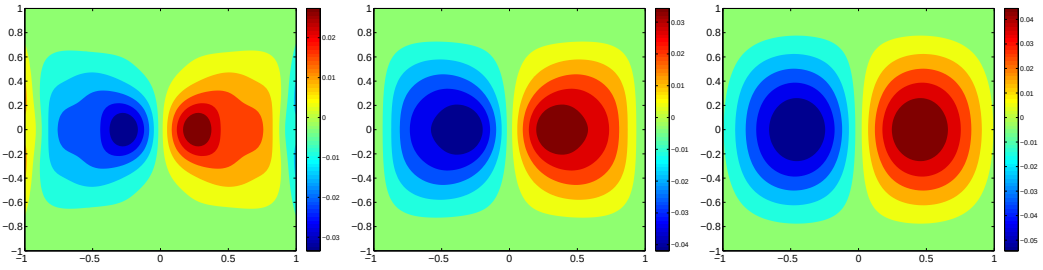
Figure 4.15: u for different ν in 2D exponential kernel with $\lambda = 1$.

is because the macroscopic properties, e.g., viscosity, velocity, can be obtained by take average in the MD simulation. Unlike this up-scaling step, for the down-scaling step, which is transferring information from the Navier-Stokes solver to the MD simulation, we will have infinite number of



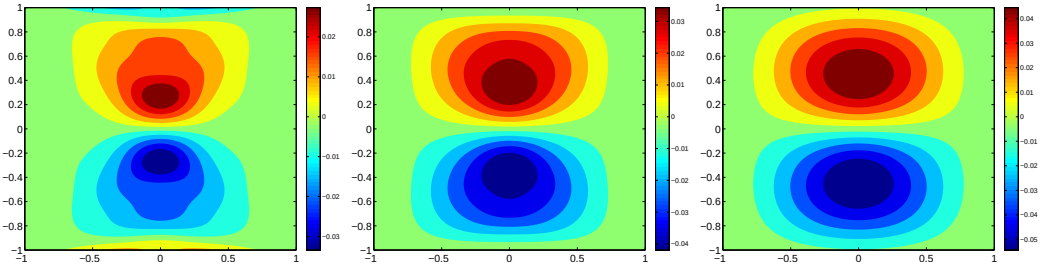
(a) $V(\nu = 0.5)$ (b) $V(\nu = 1.0)$ (c) $V(\nu = 1.5)$

Figure 4.16: V for different ν in 2D exponential kernel with $\lambda = 1$.



(a) $V_x(\nu = 0.5)$ (b) $V_x(\nu = 1.0)$ (c) $V_x(\nu = 1.5)$

Figure 4.17: V_x for different ν in 2D exponential kernel with $\lambda = 1$.



(a) $V_y(\nu = 0.5)$ (b) $V_y(\nu = 1.0)$ (c) $V_y(\nu = 1.5)$

Figure 4.18: V_y for different ν in 2D exponential kernel with $\lambda = 1$.

choice of the configurations. This is because we only have the information of the mean. Although some techniques have been proposed to deal with this problem (e.g., see [152]), these types of constrained method relies on some assumption and lose some accuracy. An alternative solution to this problem is to use the "triple-decker" skill [45], which introduce a mesoscopic layer between the miscroscopic and macroscopic model. It requires constructing the mescopic model which is not easy. Moreover, the convergence of the method by passing specific stochastic boundary condition

across the interface remains to be proved. The techniques for analysing the convergence of the deterministic equation (e.g., [100, 101]) is NOT directly applicable.

In this section we consider the domain decomposition problem with stochastic/deterministic coupling and stochastic/stochastic coupling. More specifically, for the first case, we consider a problem with two non-overlapping subdomains. The governing equation on one subdomain includes random coefficients while the governing equation on the other subdomain is deterministic. For the second case, we consider a problem with overlapping subdomains and the random coefficients have differenet representation on different domains.

4.2.1 Problem set up

We consider the elliptic equation

$$\begin{aligned} &-(a(x;\boldsymbol{\xi})u(x;\boldsymbol{\xi})')' = f(x), \quad x \in (x_l, x_r), \\ &u(x_l) = u_l, u(x_r) = u_r. \end{aligned} \tag{4.2.1}$$

where $\boldsymbol{\xi}$ is i.i.d. $\mathcal{U}[-1,1]$ and $a(x;\boldsymbol{\xi})$ is a random field. The domain is decomposed into several subdomains, and $a(x;\boldsymbol{\xi})$ denotes different random fields in different subdomains. Figure 4.19 is a demonstration of two non-overlapping adjacent domains. In this figure $\tau_{i,j}$ denotes the information transferred from D_i to D_j, and $\tau_{j,i}$ denotes the information from D_j to D_i. Figure 4.20 is a demonstration for two-overlapping domains. Similarly, $\tau_{i,j}$ denotes the information from D_i to D_j and so on. In this thesis, we decompose the entire domain into two subdomains. The idea can be directly extended to dealing with multiple subdomains. We denote the two subdomains D_1 (see D_i in Figures 4.19, 4.20) and D_2 (see D_j in Figures 4.19, 4.20).

The analytic solution for deterministic version of the problem (i.e., when $\boldsymbol{\xi}$ is fix in Eq. (4.2.1)) is

$$u(x) = u_l + \int_{x_l}^{x} \frac{a(x_l)u'(x_l) - \int_{x_l}^{s} f(t)\mathrm{d}t}{a(s)}\mathrm{d}s, \tag{4.2.2}$$

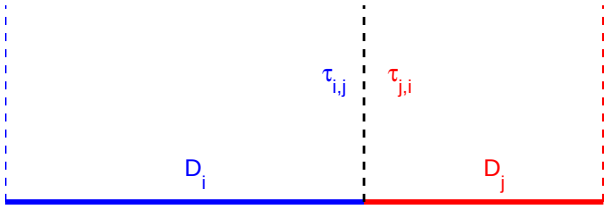


Figure 4.19: Demonstration of two non-overlapping domains.

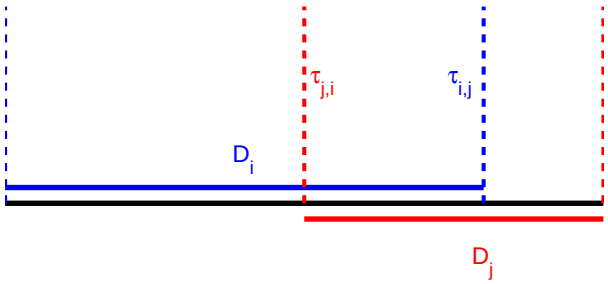


Figure 4.20: Demonstration of two overlapping domains.

where

$$a(x_l)u'(x_l) = \left(u_r - u_l + \int_{x_l}^{x_r} \frac{\int_{x_l}^{s} f(t)\mathrm{d}t}{a(s)} \mathrm{d}s \right) \Big/ \left(\int_{x_l}^{x_r} \frac{\mathrm{d}s}{a(s)} \right). \tag{4.2.3}$$

The reference solution of Eq. (4.2.1) is obtained by PCM method, i.e., by employing Eq. (4.2.2) for different samples of $\boldsymbol{\xi}$, then compute the statistics with corresponding weights for each $\boldsymbol{\xi}$.

4.2.2 Stochastic/deterministic coupling

For the stochastic/deterministic coupling problem, we set the domain $D = [-1, 1]$ and we decompose it into two non-overlapping subdomains (see Figure 4.19). We set

$$D_1 = [-1, \gamma], \quad D_2 = [\gamma, 1].$$

We consider the case when $a(x; \boldsymbol{\xi})$ is a random field in D_1 and it is a constant in D_2. On D_1 we impose Dirichlet boundary condition on both ends while on D_2 we impose Neumann boundary condition on the left end ($x = \gamma$) and Dirichlet boundary condition on the right end ($x = 1$). For

deterministic case, the analytic solution on D_1 is given in Eq. (4.2.2) with

$$a(-1)u'(-1) = \left(u(\gamma) - u(-1) + \int_{-1}^{\gamma} \frac{\int_{-1}^{s} f(t)\mathrm{d}t}{a(s)}\mathrm{d}s\right) \Big/ \left(\int_{-1}^{\gamma} \frac{\mathrm{d}s}{a(s)}\right). \tag{4.2.4}$$

The analytic solution on D_2 ($[\gamma, 1]$)is

$$u(x) = u(1) - \int_{x}^{1} \frac{a(\gamma)u'(\gamma) - \int_{\gamma}^{s} f(t)\mathrm{d}t}{a(s)}\mathrm{d}s, \tag{4.2.5}$$

We use 200 elements with 5 Gauss quadrature points in each to compute the integral, hence the error of solving the deterministic system can be neglected.

We propose an optimization method based on the gPC approximation of the solution at $x = \gamma$. We first approximate $u(\gamma)$ as

$$u_1(\gamma; \boldsymbol{\xi}) \approx \sum_{|\boldsymbol{\alpha}|=0}^{P} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}). \tag{4.2.6}$$

Here the subscript "1" indicates that this expansion will be used as the boundary condition for solving the subproblem on D_1. Notice that if we obtain an accurate gPC expansion in the above gPC approximation, we can solve the Dirichlet-Dirichlet problem on D_1 accurately, then based on the this result, compute an accurate gPC expansion of $u(\gamma; \boldsymbol{\xi})'$, next, solve the Neumann-Dirichlet problem on D_2. Hence the key point is to obtain the gPC coefficients of $u(\gamma; \boldsymbol{\xi})$. This is accomplished by the optimization procedure. We start from a initial guess of $c_{\boldsymbol{\alpha}}$, which can be obtained by solving a deterministic problem (e.g., set $a(x; \boldsymbol{\xi}) = \bar{a}$) on the global domain. This is nothing more than solving the global problem once. After this step, we will obtain c_0 which is the solution of the global problem at $x = \gamma$ and set other entries of $\boldsymbol{c}$ to be zero. Based on $\boldsymbol{c}$ we can solve the subproblem on D_1 first then solve the subproblem on D_2. After the Neumann-Dirichelet problem on D_2 is solved, we can obtain another gPC expansion of $u(\gamma; \boldsymbol{\xi})$:

$$u_2(\gamma; \boldsymbol{\xi}) \approx \sum_{|\boldsymbol{\alpha}|=0}^{P} \tilde{c}_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}). \tag{4.2.7}$$

Here the subscript "2" that this solution is obtained by solving the subproblem on D_2. We aim to minimize the difference between u_1 and u_2, namely $\|u_1(\gamma;\boldsymbol{\xi}) - u_2(\gamma;\boldsymbol{\xi})\|_2$. If we use the same sets of the basis in the gPC expansion in D_1 and D_2, this is equivalent to minimizing $\|c - \tilde{c}\|_2$. As a summary, the problem is set as the following optimization problem:

$$\min \mathcal{J}(\boldsymbol{c}) = \min_{\boldsymbol{c}} \{\|\boldsymbol{c} - \tilde{\boldsymbol{c}}\|_2 \mid \tilde{\boldsymbol{c}} = \mathcal{L}_2(\mathcal{L}_1(\boldsymbol{c})),\} \tag{4.2.8}$$

where $\mathcal{L}_1$ the mapping $\boldsymbol{c} \mapsto \boldsymbol{c}^*$ where c^* is the gPC coefficients of $u_1(x;\boldsymbol{x})'$ on the boundary $x = \gamma$ and $\mathcal{L}_2$ is $\mathcal{L}_2$ is the mapping $\boldsymbol{c}^* \mapsto \tilde{\boldsymbol{c}}$ where $\tilde{\boldsymbol{c}}$ is the gPC expansion of $u_2(x;\boldsymbol{\xi})$ on the boundary $x = \gamma$.

Since we consider a deterministic/stochastic coupling problem, we set the random field $a(x;\boldsymbol{\xi})$ as follows:

$$a(x;\boldsymbol{\xi}) = \begin{cases} 1.0 + \sigma G(x;\boldsymbol{\xi})\mathcal{M}(x;\gamma,\delta) & x < \gamma, \\ 1.0 & x \geq \gamma. \end{cases} \tag{4.2.9}$$

Here $G(x;\boldsymbol{\xi})$ is the KL expansion of a random field on $[-1,\gamma]$, σ is the magnitude of the perturbation which is selected to guarantee the positivity of $a(x;\boldsymbol{\xi})$, $\mathcal{M}(x;\gamma)$ is the following mollifier:

$$\mathcal{M}(x;\gamma,\delta) = \begin{cases} 1.0 & x \leq \gamma - \delta, \\ \exp\left(1.0 - \dfrac{1.0}{1.0 - |[x-(\gamma-\delta)]/\delta|^2}\right) & \gamma - \delta < x < \gamma \\ 0.0 & x \geq \gamma. \end{cases} \tag{4.2.10}$$

We set $\delta = 0.2$ here. Notice that the mollifier is a C^∞ function see Figure 4.21.

We first set $G(x;\boldsymbol{\xi}) = \xi$ where $\xi \sim \mathcal{U}[-1,1]$, i.e., a uniform random variable distributed on $[-1,1]$. We set $\sigma = 0.5$, which implies that the range of $a(x;\boldsymbol{\xi})$ is $[0.5, 1.5]$. γ is set to be 0.5. We solve the elliptic equation on $[-1,1]$ with 9 Gauss quadrature points to obtain the reference solution u_g. Then we approximate $u_1(\gamma)$ with a fifth-th order gPC expansion, hence we have 6 variables

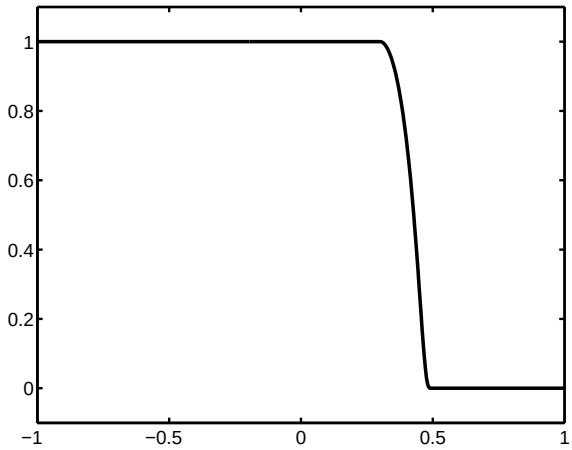
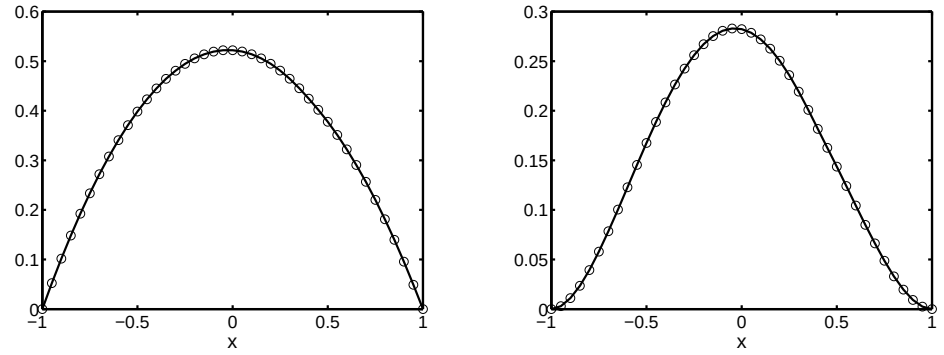


Figure 4.21: Demonstration of mollifier.

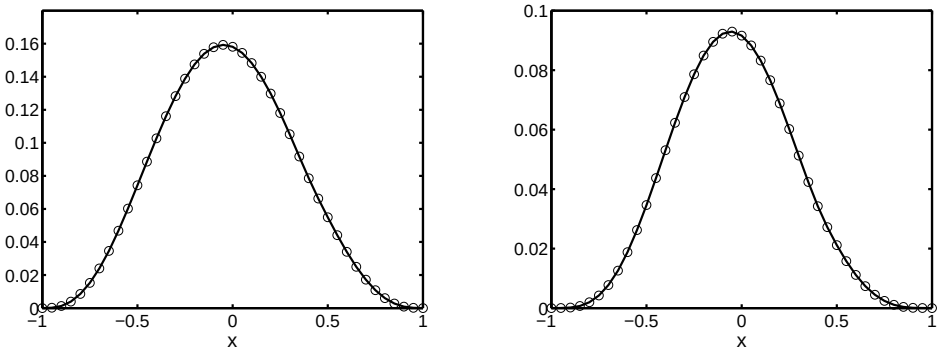
in the optimization problem. The optimization is achieved by `SNOPT` [4] package. We compare the first fourth-order moments of the solution by the domain decomposition method and the reference solution at 21 equi-distance points on $[-1, 1]$:

$$\mu_r(x_i) = \mathbb{E}(u(x_i; \boldsymbol{\xi})^r), \quad i = 0, 1, \cdots, 21; \ r = 1, 2, 3, 4. \tag{4.2.11}$$

Figure 4.22 presents the comparison results, where solid lines are reference solutions and circles are results by optimization-based domain decomposition. Obviously, it is impossible to observe the difference on this figure. We compute the relative l_2 error of each moments over the physical space and present the results in Table 4.1. From this table we observe that the error is small and the error is larger for larger r. Notice that in our test we use the same number of Gauss quadrature point in random space for global problem and the domain decomposition problem, the error here consists of two parts: 1) the truncation error of the gPC approximation of the $u_1(\gamma; \boldsymbol{\xi}), u_1(\gamma; \boldsymbol{\xi})'$ and $u_2(\gamma; \boldsymbol{\xi})$; 2) the accuracy of the optimization solver. If the truncation error is the same as the global solution and if the optimization solver provides the global minimum of the optimization problem, the results by domain decomposition should be the same as the global solution since the solution of this equation is unique.



(a) μ_1 (b) μ_2



(c) μ_3 (d) μ_4

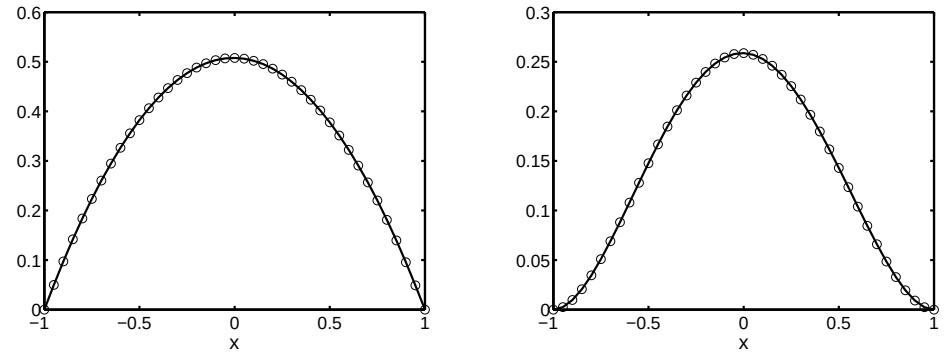
Figure 4.22: 1D case: comparison of the first four moments for the stochastic domain decomposition problem. Solid lines are the reference solution and circles are the results by the decomposition.

Table 4.1: 1D case: relative error (in L^2 norm) of the estimated first four moments.

μ_1	μ_2	μ_3	μ_4
$1.098e-6$	$7.769e-6$	$3.208e-5$	$9.909e-5$

We also test a multi-D case in which we use the KL expansion with Gaussian kernel to represent the random field. We set the correlation length to be 0.2. The KL expansion on $[-1, 0.5]$ requires 10 eigenvalues cover 99% of the "energy", i.e., the summation of all the eigenvalue. Level 3 sparse grid points are employed to compute the global solution and to obtain the gPC expansion in each subdomain in each iterations within the optimization solver. The gPC expansion for $u_1(\gamma; \boldsymbol{\xi})$ is truncated up to $P = 2$, hence we need to identify 66 coefficients. The comparison of the results by different methods are presented in Figure 4.23 and the error of different moments are listed in

Table 4.2. We can see that the results by the domain decomposition method is very accurate.



(a) μ_1 (b) μ_2

(c) μ_3 (d) μ_4

Figure 4.23: 10D case: comparison of the first four moments for the stochastic domain decomposition problem. Solid lines are the reference solution and circles are the results by the decomposition.

Table 4.2: 10D case: relative error (in L^2 norm) of the estimated first four moments.

μ_1	μ_2	μ_3	μ_4
$3.752e-7$	$2.544e-6$	$8.225e-6$	$1.969e-5$

For the above two cases, we require that the gPC expansion of the solution at the boundary from different subdomain should be the same. This is a very strong requirement, hence yields accurate results but it is costly. For higher dimensional problems, especially for stochastic/stochastic coupling problems, we may relax the requirement to matching the first several moments of the solution from different subdomains. Moreover, with this method we do not explicitly transfer the information across the interface since the optimization solver accomplishes this task in each iteration. However,

for the problems with many subdomains, the complexity of the optimization problem increases very fast as the number of unknown becomes larger. Hence, a more delicate optimization solver is needed. Alternatively, an efficient approach to explicitly transfer the stochastic information across the interface is necessary.

4.2.3 Stochastic/stochastic coupling

In this section, we consider two adjacent domains with different correlation lengths, i.e., $a(x;\boldsymbol{\xi})$ are different random fields in different subdomains. We decompose the domain $D = [0,1]$ into two overlapping domains (see Figure 4.20) $D_1 = [0, 0.8], D_2 = [0.6, 1.0]$. The random fields $a(x;\boldsymbol{\xi})$ is given as the following KL expansion:

$$a(x;\boldsymbol{\xi}) = \bar{a} + \sum_k \sqrt{\lambda_k}\phi_k(x)\xi_k.$$

Here (λ_k, ϕ_k) are eigenpairs of the following modified Gauss covariance kernel:

$$\mathcal{C}(x,y) = \exp\left(-\frac{|x-y|^2}{\left[\frac{l(x)+l(y)}{2}\right]^2}\right), \tag{4.2.12}$$

where the correlation length is given as:

$$l(x) = \mathcal{M}(x;\gamma,\delta)\cdot\left(\frac{1}{10}-\frac{1}{40}\right)+\frac{1}{40} \tag{4.2.13}$$

with $\gamma = 0.8, \delta = 0.2$ (see Figure (4.24)). In other words, on $[0, 0.6]$ the correlation length is $1/10$, on $[0.8, 1.0]$ the correlation length is $1/40$, and on $[0.6, 0.8]$ the correlation length is a C^∞ function connecting $1/10$ and $1/40$. We select the overlapping domain as $[0.6, 0.8]$, i.e., the domain with mollified correlation length. The covariance matrix is given in Figure 4.25. We can see from this Figure that the correlation length on $[0, 0.6]$ and $[0.8, 1.0]$ are constants. $[0.6, 0.8]$ is a transition

area in which the correlation length decreases smoothly. To represent the random field based on the eigenpairs of kernel (4.2.12), we need 21 terms in KL expansion to cover 99.3% of the total energy. A realization of this random field is given in Figure 4.26 as an example.

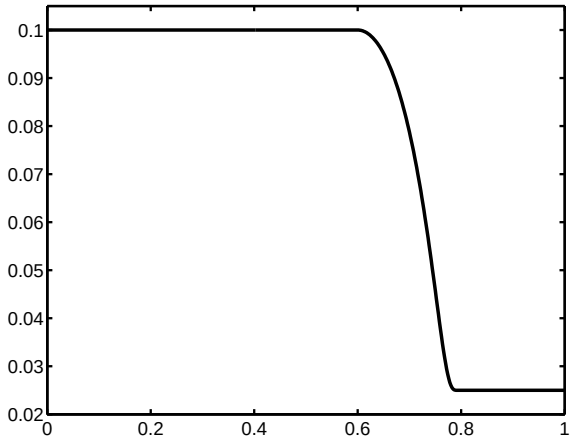


Figure 4.24: Correlation length of the random field of the stochastic/stochastic coupling problem.

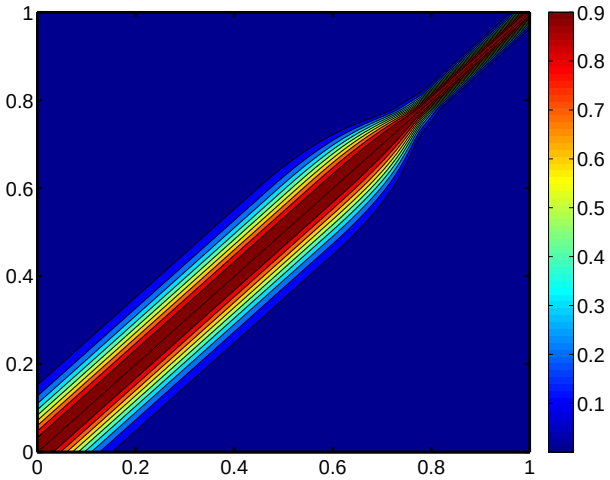


Figure 4.25: Demonstration of covariance kernel $\mathcal{C}(x, y)$ given by Eq.(4.2.12).

In order to represent the random field in subdomains, we firstly construct the covariance matrix C based on the quadrature points. As is well known that to compute the eigenpairs, we discretize the integral equation

$$\int \mathcal{C}(x, y)\phi(y)\mathrm{d}y = \lambda\phi(x)$$

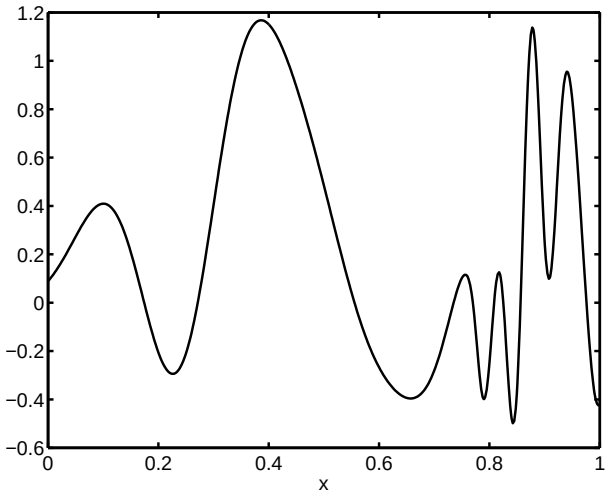


Figure 4.26: One realization of the random field with covariance kernel $\mathcal{C}(x, y)$ given by Eq.(4.2.12).

by approximating the integral with quadrature rule. Hence we need to solve the following eigenvalue problem:

$$C W \boldsymbol{v} = \lambda \boldsymbol{v},$$

where W is a diagonal matrix consists of the weights of the corresponding quadrature points. Here $\boldsymbol{v}$ should be a normalized since $\|\phi\|_2 = 1$. The matrix C we use in this section is presented in Figure 4.25. Here we decompose domain $[0, 1]$ into 100 equi-distance elements and use 3 Gauss quadrature points in each element. The random field on $[0, 1]$ is represented by the following KL expansion:

$$R_g(x; \boldsymbol{\xi}_g) = \bar{a} + \sum_{k}^{K_g} \sqrt{\lambda_{gk}} \phi_{gk}(x) \xi_k, \quad \xi_i \sim \text{i.i.d. } U[-1, 1]. \tag{4.2.14}$$

Next, we decompose the covariance matrix C into three (overlapping) sub-covariance matrix $C_i, i = 1, 2, 3$ as shown in Figure 4.27. Each C_i is the covariance matrix for a subinterval of $[0, 1]$:

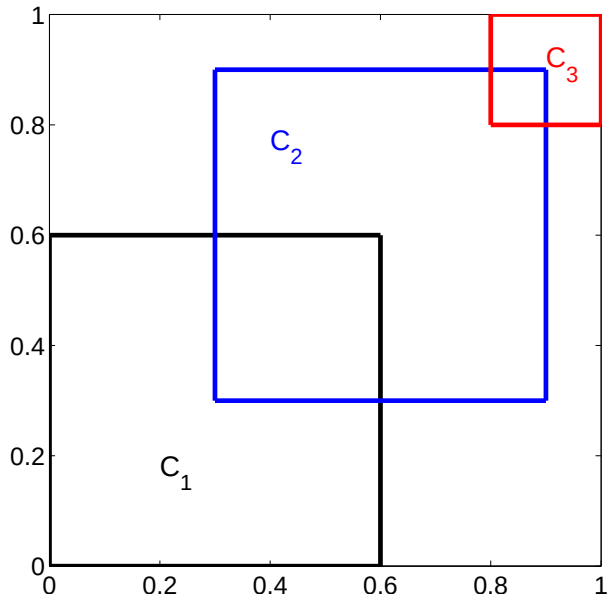


Figure 4.27: Demonstration of decomposing the covariance matrix C.

$$\begin{cases} C_1(x,y): & x,y \in [0,0.6], \\ C_2(x,y): & x,y \in [0.3,0.9], \\ C_3(x,y): & x,y \in [0.8,1.0]. \end{cases} \tag{4.2.15}$$

Notice that with this decomposition we do not keep all the information of the covariance kernel. For example, the correlation between $x = 0.2$ and $y = 0.8$ is neglected. However, the value of covariance between these two points are very small: $\mathcal{C}(0.2, 0.8) \sim \mathcal{O}(10^{-7})$. Due to the correlation length and the size of the sub-covariance matrix, we keep most information of the covariance matrix C. More specifically, if we set $\tilde{C} = C$, and set all the entries of $\tilde{C}$, the corresponding location point (x, y) is not included in any of C_i in Figure 4.27, to be zero, the relative L_2 error $\|C - \tilde{C}\|_2 / \|C\|_2 = 1.96 \times 10^{-11}$. In order to obtain C_i, we multiply the corresponding part of C by a weight matrix H_i:

$$C_i = (C(m_i : n_i, m_i : n_i) \cdot H_i)_p, \quad i = 1, 2, 3.$$

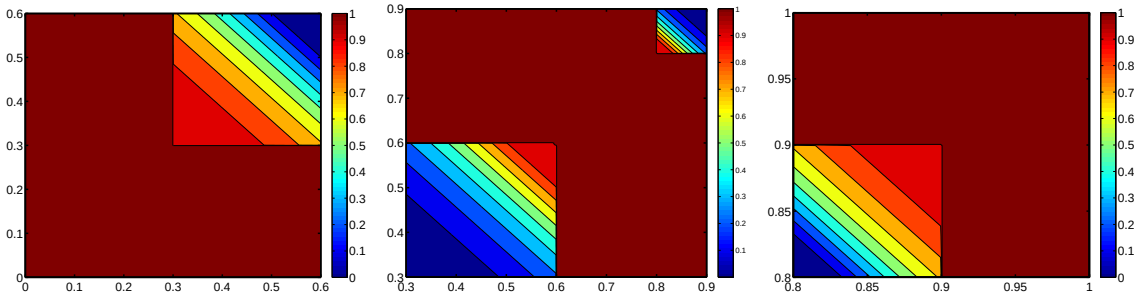
where $(\cdot)_p$ is point-wise product between two matrices (i.e., if $C = (A \cdot B)_p$, then $C_{ij} = A_{ij}B_{ij}$), H_i are weight matrices:

$$H_1 = \begin{cases} \mathcal{M}((x+y)/2; 0.6, 0.3), & (x,y) \in [0.3, 0.6]^2; \\ 1, & (x,y) \in [0, 0.6]^2 / [0.3, 0.6]^2; \end{cases} \tag{4.2.16}$$

$$H_2 = \begin{cases} 1 - \mathcal{M}((x+y)/2; 0.6, 0.3), & (x,y) \in [0.3, 0.6]^2; \\ 1 - \mathcal{M}((x+y)/2; 0.9, 0.1), & (x,y) \in [0.8, 0.9]^2; \\ 1, & (x,y) \in [0.3, 0.9]^2 / ([0.3, 0.6]^2 \cup [0.8, 0.9]^2); \end{cases} \tag{4.2.17}$$

$$H_3 = \begin{cases} \mathcal{M}((x+y)/2; 0.9, 0.1), & (x,y) \in [0.8, 0.9]^2; \\ 1, & (x,y) \in [0.8, 1]^2 / [0.8, 0.9]^2. \end{cases} \tag{4.2.18}$$

Notice that the entries of the weight matrices are 1 except for the area where different C_i overlap. For example, at point $(0.2, 0.2)$ the corresponding entry of the weight matrix is 1 while at point $(0.5, 0.5)$ the corresponding entry of the weight matrix is between 0 and 1. To construct the weight matrix in the overlapping area, we employ the mollifying function. Figure 4.28 presents three weight matrices we use. Each weight matrix corresponds to a block of the covariance matrix C. The three sub-covariance matrices are presented in Figure 4.29. In each subdomain, we solve a



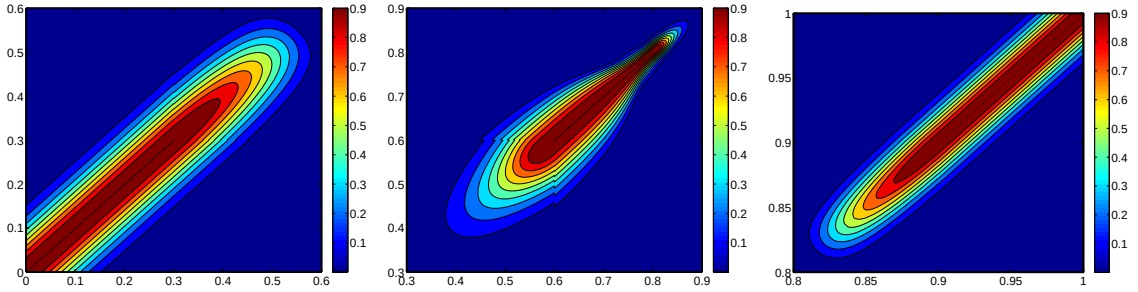
(a) Weight matrix 1 (b) Weight matrix 2 (c) Weight matrix 3

Figure 4.28: Weight matrices.

local eigenvalue problem

$$C_i W(m_i : n_i, m_i, n_i)\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i,$$

where $W(m_i : n_i, m_i, n_i)$ is a block of quadrature weight matrix W corresponding to the block of C, based on which C_i is constructed. Hence, we can obtain three KL expansion based on three



(a) Sub-covariance matrix 1 (b) Sub-covariance matrix 2 (c) Sub-covariance matrix 3

Figure 4.29: Sub covariance matrices.

different C_i:

$$\begin{aligned} KL_1(x;\boldsymbol{\xi}_1) &= \sum_{k=1}^{K_1} \lambda_{1k}\phi_{1k}(x)\xi_{1k}, \quad x \in [0, 0.6]; \\ KL_2(x;\boldsymbol{\xi}_2) &= \sum_{k=1}^{K_2} \lambda_{2k}\phi_{2k}(x)\xi_{2k}, \quad x \in [0.3, 0.9]; \\ KL_3(x;\boldsymbol{\xi}_3) &= \sum_{k=1}^{K_3} \lambda_{3k}\phi_{3k}(x)\xi_{3k}, \quad x \in [0.8, 1]; \end{aligned} \tag{4.2.19}$$

where $\xi_{ik}(i = 1, 2, 3) \sim$ i.i.d. $U[-1, 1]$. Based on these KL expansions, we construct the random field on D_1 and D_2 as follows:

$$\begin{aligned} R_1(x;\boldsymbol{\xi}_1,\boldsymbol{\xi}_2) &= \bar{a} + KL_1(x;\boldsymbol{\xi}_1) + KL_2(x;\boldsymbol{\xi}_2)\mathbf{1}_{[0,0.6]}, \quad x \in D_1; \\ R_2(x;\boldsymbol{\xi}_2,\boldsymbol{\xi}_3) &= \bar{a} + KL_3(x;\boldsymbol{\xi}_3) + KL_2(x;\boldsymbol{\xi}_2)\mathbf{1}_{[0.6,1]}, \quad x \in D_2, \end{aligned} \tag{4.2.20}$$

where $\mathbf{1}_{[\cdot]}$ is the indicator function. With the correlation length given in Eq.(4.2.13), we use $K_g = 21$ random variables in R_g to represent the random field, the relative L_2 error of the reconstructed covariance matrix based on 21 eigenfunction $\phi_{gk}(x)$ is 1.3×10^{-4}. We set $K_1 = 7, K_2 = 10, K_3 = 9$ for KL_1, KL_2, KL_3, respectively. The relative L_2 error of the reconstructed covariance matrix

based on these three KL expansion is 1.4×10^{-4}.

Similar to the stochastic/deterministic coupling, we use optimization method to solve this problem. On both D_1 and D_2 we solve Dirichlet boundary problem to obtain solution $u_1(x; \boldsymbol{\xi}_1, \boldsymbol{\xi}_2)$ and $u_2(x; \boldsymbol{\xi}_2, \boldsymbol{\xi}_3)$, respectively. Here we need to provide the expression of u_1 at the boundary of D_1, i.e., $x = 0.8$ and the expression of u_2 at the boundary of D_2, i.e., $x = 0.6$. Hence, we assume the gPC expansion of u_1 and u_2 at these two points are:

$$\begin{aligned} u_1(0.8; \boldsymbol{\xi}_1, \boldsymbol{\xi}_2) &= \sum_{|\boldsymbol{\alpha}_j|=0}^{P_1} c_{\boldsymbol{\alpha}_j} \psi_{\boldsymbol{\alpha}_j}(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2), \\ u_2(0.6; \boldsymbol{\xi}_2, \boldsymbol{\xi}_3) &= \sum_{|\boldsymbol{\beta}_j|=0}^{P_2} d_{\boldsymbol{\beta}_j} \varphi_{\boldsymbol{\beta}_j}(\boldsymbol{\xi}_2, \boldsymbol{\xi}_3). \end{aligned} \tag{4.2.21}$$

We aim to minimize the difference of the statistics of the u_1 and u_2 at the boundary of D_1 and D_2. More specifically, based on $u_1(0.8; \boldsymbol{\xi}_1, \boldsymbol{\xi}_2)$ we can obtain $u_1(0.6; \boldsymbol{\xi}_1, \boldsymbol{\xi}_2)$ and based on $u_2(0.6; \boldsymbol{\xi}_2, \boldsymbol{\xi}_3)$ we can obtain $u_2(0.8; \boldsymbol{\xi}_2, \boldsymbol{\xi}_3)$. We aim to minimize the following object function:

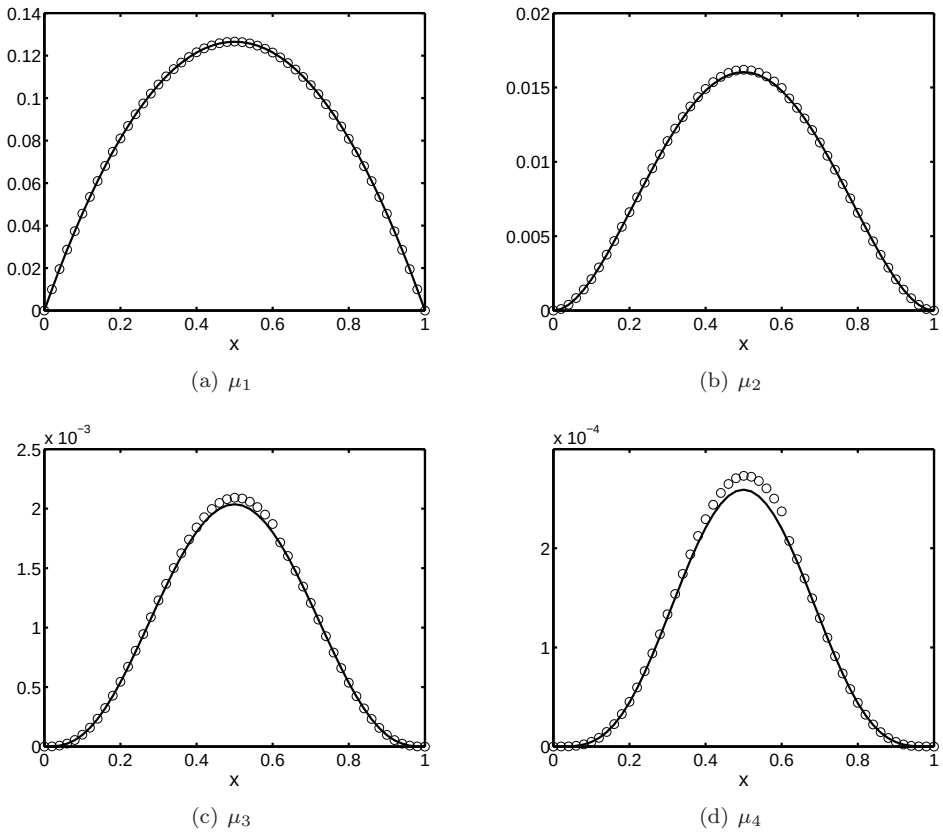
$$\mathcal{J}(u_1, u_2) = \sum_{r=1}^{R} w_r \left[|\mu_r(u_1(0.6)) - \mu_r(u_2(0.6))|^2 + |\mu_r(u_1(0.8)) - \mu_r(u_2(0.8))|^2 \right], \tag{4.2.22}$$

where μ_r is the r-th order moments, w_r is a weight since the magnitude of different order moments are different. Now that u_1 and u_2 at the boundaries are approximated by gPC expansions $u_1(c_{\boldsymbol{\alpha}})$ and $u_2(c_{\boldsymbol{\beta}})$, the objection function can be expressed as the function of $c_{\boldsymbol{\alpha}}$ and $d_{\boldsymbol{\beta}}$ in Eq. (4.2.21):

$$\mathcal{J}(c_{\boldsymbol{\alpha}}, d_{\boldsymbol{\beta}}) = \sum_{r=1}^{R} w_r [|\mu_r(\mathcal{L}_1(c_{\boldsymbol{\alpha}})) - \mu_r(u_2(d_{\boldsymbol{\beta}}))|^2 + |\mu_r(\mathcal{L}_2(c_{\boldsymbol{\beta}})) - \mu_r(u_1(d_{\boldsymbol{\alpha}}))|^2], \tag{4.2.23}$$

where $\mathcal{L}_1$ is the mapping $c_{\boldsymbol{\alpha}} \mapsto u_1(0.6; \boldsymbol{\xi}_1, \boldsymbol{\xi}_2)$ and $\mathcal{L}_2$ is the mapping $d_{\boldsymbol{\beta}} \mapsto u_2(0.6; \boldsymbol{\xi}_1, \boldsymbol{\xi}_2)$ In this problem, we set $P_1 = P_2 = 2$ in Eq. (4.2.21), $R = 2$ in Eq. (4.2.23), $w_r = 20$ in Eq. (4.2.23). We use SNOPT [4] to minimize $J(c_{\boldsymbol{\alpha}}, d_{\boldsymbol{\beta}})$, and the results are presented in Figure 4.30. Here we use level 2 sparse grid method to compute the statistics of u_1 and u_2. The reference solution is

obtained by solving the global problem on $[0, 1]$ with 21 random variables in the expansion of the random field R_g, i.e., $K_g = 21$ in Eq. (4.2.14). The relative errors of the first four moments are listed in Table 4.3. We notice that for the estimate of the higher order moments, the accuracy is decreasing. The error is large near $x = 0.5$ and is small near the boundary. This is because we set $u = 0$ on both ends of the domain.



(a) μ_1 (b) μ_2

(c) μ_3 (d) μ_4

Figure 4.30: 21D case: comparison of the first four moments for the stochastic domain decomposition problem. Solid lines are the reference solution and circles are the results by the decomposition.

Table 4.3: 21D case: relative error (in L^2 norm) of the estimated first four moments.

μ_1	μ_2	μ_3	μ_4
$1.266e-3$	$7.684e-3$	$2.297e-2$	$4.739e-2$

Also, if we compare these results with those in the stochastic/deterministic coupling, the error here is larger. This is because for the stochastic/deterministic coupling, we match the gPC expan-

sion at the boundary, which indicates that we match each trajectory of the local solution at the boundary. However, for the stochastic/stochastic coupling, we only match several moments of local solutions at the boundary, hence, the results is less accurate.

Remark 4.2.1. We use PDE constrained optimization method to solve the stochastic domain decomposition problem. With this method we do not need to provide an explicit form of the information transferred across the boundary.However, if we decompose the entire domain into more subdomains, the computational cost may increase fast since we will have more unknowns in the system. Therefore, an appropriate expression of the information to pass as well as a quick iteration method is required. This will be our future work.

Chapter 5

Summary and Future Work

In this thesis, we present adaptive ANOVA method for high-dimensional problems, gPC based compressive sensing method to obtain a surrogate model efficiently and the narrow escape problem as well as stochastic domain decomposition.

In Chapter 2, we propose a criterion based on the variance of the component functions in (anchored) ANOVA decompositions. Hence the selection pool of the higher order component functions is decided by the lower order component functions. In both compressible flow and ring oscillator models presented in this Chapter, we can see that the most important interaction terms are the second-order interactions, i.e., second-order component functions make main contribution to the ANOVA decomposition. The high-order component functions are neglected, hence the computational cost is dramatically reduced. For the 100 dimensional case of the compressible flow, due to the magnitude of the perturbation, we only need first-order component functions to capture mean and standard deviation of the system. These examples imply that in practical problems, even the nominal dimension of the system is high, a draconian truncation of the ANOVA expansion may lead to accurate solutions.

In Chapter 3, we consider systems with sparse gPC expansions. We test the reweighted ℓ_1 minimization method and also combine the reweighted ℓ_1 minimization with the Cheyshev sampling

strategy to propose a efficient way to construct the gPC expansion. This method can be considered as a post processing of Monte Carlo method if no special sampling strategy is employed, hence, it is flexible. We point out that for high-dimensional problem, the Chebyshev measure based sampling strategy is not a good choice, and reweighted ℓ_1 alone is sufficient. We also notice that the upper bound of the accuracy of this method is not sharp since it predicts that the error is $\mathcal{O}\frac{1}{\sqrt{m}}$, where m is number of samples we use. Our numerical results for stochastic elliptic equations show higher accuracy than this. In the second part of Chapter 3, we use gPC expansion to construct the surrogate model of the response surface of two DPD systems. Based on the physical knowledge of the system we select the macroscopic properties of system such that their gPC expansions are sparse. Hence, we are able to compute the gPC coefficients based on limited number of DPD simulations, which are usually very costly. Further, this accurate response surface enables us to apply Bayesian inference to identify the parameters in the DPD model. Notice that the inference procedure is based on the MCMC, which requires evaluating the system tens of thousands of times. Therefore, our method is very suitable for this type of problems since the surrogate model is accurate and extremely easy to evaluate. This approach provides a general framework for mescoscale model calibration, and it leads to parametric compression and dimensionality reduction.

In Chapter 4, we firstly consider the narrow escape problem. The mean first passage time is maximized by imposing a potential to trap the Brownian particle. This type of research can be applied to many rare events problems where reaction rate is of interest since it can be expressed as a function of the MFPT. Our optimization results show the optimal shape of different types of potentials for 1D and 2D cases. This will help to identity ways to accelerate the reactions in biology, chemistry, etc. Moreover, an optimization based method to solve stochastic domain decomposition problem is proposed. This method is very accurate as long as the gPC expansion of the solution on the boundary is small. It helps to understand the propagation of uncertainties across different domains and different scales. Therefore, it is helpful to study the rare events generated by the small perturbation induced by the uncertainty from different domain or different scales.

In the future, we would like to extend research to multiscale modeling, simulation of complex materials. We would like to develop a systematic approach of model calibration for mesoscale models as well as other multiscale models. We aim to to develop new sampling strategies which are capable of selecting sampling points more efficiently. Next, in order to exploit information from the simulation results, we plan to develop a hybrid ANOVA-compressive sensing method. This method takes advantage of both adaptive ANOVA and compressive sensing based UQ method. More specifically, the ANOVA method helps to determine the important variables and the CS based UQ method constructs a gPC surrogate model with a few simulations results. As the adaptive ANOVA is a hierarchical approach, this new hybrid method will be able to identify additional sampling points that can be tested to further enhance the accuracy of the approximation. In the last step, Bayesian inference is employed to search for suitable parameters and achieve model reduction by identifying existing correlations between parameters. Moreover, we plan to develop algorithms based on the aforementioned methods to study the rare events. One possible approach that we would like to explore in the near future is the following. As CG model allows larger time steps, it can serve as a guidance for the MD simulation to explore the free energy surface. We can use a CG model to first probe the free energy surface (relatively) quickly, and then use MD simulations to detect the detailed structure, e.g., local/global minima, saddle points, etc. In this "prediction-correction" procedure, a suitable CG model is critical. We will aim to design better approaches to determine optimal parameters in the CG model and to select proper collective variables. Further, we will continue our work on the stochastic domain decomposition, which helps to understand the perturbation inducing rare events across scales more efficiently, and propagation of uncertainty in multiphysics problems.

Bibliography

[1] http://en.wikipedia.org/wiki/Ring_oscillator.

[2] http://en.wikipedia.org/wiki/Narrow_escape_problem.

[3] ℓ_1-magic. http://statweb.stanford.edu/~candes/l1magic/.

[4] SNOPT. http://www.sbsi-sol-optimize.com/asp/sol_product_snopt.htm.

[5] SparseLab: seeking sparse solutions to linear system of equations. http://sparselab.stanford.edu.

[6] P. Abrahamsen. *A review of Gaussian random fields and correlation functions*. Norsk Regnesentral/Norwegian Computing Center, 1997.

[7] C. F. Abrams and E. Vanden-Eijnden. Large-scale conformational sampling of proteins using temperature-accelerated molecular dynamics. *Proc. Natl. Acad. Sci.*, 107(11):4961–4966, 2010.

[8] R. L. C. Akkermans and W. J. Briels. Coarse-grained dynamics of one chain in a polymer melt. *J. Chem. Phys.*, 113(15):6409–6422, 2000.

[9] R. L. C. Akkermans and W. J. Briels. Coarse-grained interactions in polymer melts: A variational approach. *J. Chem. Phys.*, 115(13):6210–6219, 2001.

[10] P. Angelikopoulos, C. Papadimitriou, and P. Koumoutsakos. Bayesian uncertainty quantification and propagation in molecular dynamics simulations: A high performance computing framework. *J. Chem. Phys.*, 137(14):144103, 2012.

[11] R. Archibald, A. Gelb, R. Saxena, and D. Xiu. Discontinuity detection in multivariate space for stochastic simulations. *J. Comput. Phys.*, 228(7):2676–2689, 2009.

[12] J. A. Backer, C. P. Lowe, H. C. J. Hoefsloot, and P. D. Iedema. Poiseuille flow to measure the viscosity of particle model fluids. *J. Chem. Phys.*, 122(15):154503–154509, 2005.

[13] O. Bénichou and R. Voituriez. Narrow-escape time problem: Time needed for a particle to exit a confining domain through a small window. *Phys. Rev. Lett.*, 100(16):168105, 2008.

[14] M. Bieri and C. Schwab. Sparse high order FEM for elliptic SPDEs. *Comput. Methods Appl. Mech. Engrg.*, 198(13-14):1149–1170, 2009.

[15] P. Bjorstad and W. Gropp. *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*. Cambridge University Press, 2004.

[16] J. D. Blanchard, C. Cartis, and J. Tanner. Compressed sensing: how sharp is the restricted isometry property? *SIAM Rev.*, 53(1):105–125, 2011.

[17] G. Blatman and B. Sudret. An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Prob. Engrg. Mech.*, 25(2):183–197, 2010.

[18] G. Blatman and B. Sudret. Efficient computation of global sensitivity indices using sparse polynomial chaos expansions. *Reliab. Engrg. Syst. Safe.*, 95(11):1216–1229, 2010.

[19] G. Blatman and B. Sudret. Adaptive sparse polynomial chaos expansion based on least angle regression. *J. Comput. Phys.*, 230(6):2345–2367, 2011.

[20] T. Blumensath and M. E. Davies. Iterative hard thresholding for compressed sensing. *Appl. Comput. Harmon. Anal.*, 27(3):265–274, 2009.

[21] E. S. Boek, P. V. Coveney, H. N. W. Lekkerkerker, and P. van der Schoot. Simulating the rheology of dense colloidal suspensions using dissipative particle dynamics. *Phys. Rev. E*, 55(3):3124–3133, 1997.

[22] A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Rev.*, 51(1):34–81, 2009.

[23] J. A. Bucklew. Large deviation techniques in decision, simulation, and estimation. *Wiley Series in Probability and Mathematical Statistics, New York: Wiley, 1990*, 1, 1990.

[24] R. Caflisch, W. Morokoff, and A. Owen. Valuation of mortgage-backed securities using brownian bridges to reduce the effective dimension. *J. Comput. Finance*, 1:27–46, 1997.

[25] J.-F. Cai, S. Osher, and Z. Shen. Convergence of the linearized bregman iteration for ℓ_1-norm minimization. *Math. Comput.*, 78(268):2127–2136, 2009.

[26] J.-F. Cai, S. Osher, and Z. Shen. Linearized Bregman iterations for compressed sensing. *Math. Comput.*, 78(267):1515–1536, 2009.

[27] E. J. Candès. The restricted isometry property and its implications for compressed sensing. *C. R. Math. Acad. Sci. Paris*, 346(9-10):589–592, 2008.

[28] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Trans. Inf. Theory*, 51(12):4203–4215, 2005.

[29] E. J. Candès, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted l_1 minimization. *J. Fourier Anal. Appl.*, 14(5-6):877–905, 2008.

[30] S. Chen, S. A. Billings, and W. Luo. Orthogonal least squares methods and their application to non-linear system identification. *Internat. J. Control*, 50(5):1873–1896, 1989.

[31] A. F. Cheviakov, M. J. Ward, A. Peirce, and T. Kolokolnikov. An asymptotic analysis of the mean first passage time for narrow escape problems: Part ii: The sphere. *Multiscale Model. Simul.*, 8(3):836–870, 2010.

[32] B.-T. Chu. On weak interaction of strong shock and mach waves generated downstream of the shock. *J. Aeronaut. Sci.*, 19(7):433–446, 1952.

[33] M. Clark, R. D. Cramer, and N. Van Opdenbosch. Validation of the general purpose tripos 5.2 force field. *J. Comput. Chem.*, 10(8):982–1012, 1989.

[34] G. Davis, S. Mallat, and M. Avellaneda. Adaptive greedy approximations. *J. Construct. Approx.*, 13(1):57–98, 1997.

[35] A. Dembo and O. Zeitouni. *Large deviations techniques and applications*, volume 2. Springer, 1998.

[36] R. A. DeVore and V. N. Temlyakov. Some remarks on greedy algorithms. *Adv. Comput. Math.*, 5(2-3):173–187, 1996.

[37] D. L. Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, 2006.

[38] D. L. Donoho, M. Elad, and V. N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Trans. Inf. Theory*, 52(1):6–18, 2006.

[39] A. Doostan and H. Owhadi. A non-adapted sparse approximation of PDEs with stochastic inputs. *J. Comput. Phys.*, 230(8):3015–3034, 2011.

[40] A. Eriksson, M. N. Jacobi, J. Nystrom, and K. Tunstrom. Using force covariance to derive effective stochastic interactions in dissipative particle dynamics. *Phys. Rev. E*, 77(1):016707, 2008.

[41] P. Espanol, M. Serrano, and I. Zuniga. Coarse-graining of a fluid and its relation with dissipative particle dynamics and smoothed particle dynamics. *Internat. J. Modern Phys. C*, 8:899–908, 1997.

[42] P. Espanol and P. Warren. Statistical mechanics of dissipative particle dynamics. *Europhys. Lett.*, 30(4):191–196, 1995.

[43] X. Fan, N. Phan-Thien, S. Chen, X. Wu, and T. Y. Ng. Simulating flow of DNA suspension using dissipative particle dynamics. *Phys. Fluids*, 18(6):063102, 2006.

[44] D. A. Fedosov, B. Caswell, and G. E. Karniadakis. Steady shear rheometry of dissipative particle dynamics models of polymer fluids in reverse Poiseuille flow. *J. Chem. Phys.*, 132:144103, 2010.

[45] D. A. Fedosov and G. E. Karniadakis. Triple-decker: Interfacing atomistic-mesoscopic-continuum flow regimes. *J. Comput. Phys.*, 228(4):1157 – 1171, 2009.

[46] R. Fisher. *Statistical Methods for Research Workers*. Oliver and Boyd, 1925.

[47] J. Foo and G. E. Karniadakis. Multi-element probabilistic collocation method in high dimensions. *J. Comput. Phys.*, 229(5):1536–1557, 2010.

[48] J. Foo, X. Wan, and G. E. Karniadakis. The multi-element probabilistic collocation method (ME-PCM): error analysis and applications. *J. Comput. Phys.*, 227(22):9572–9595, 2008.

[49] J. Y. Foo and G. E. Karniadakis. Multi-element probabilistic collocation in high dimensions. *J. Comput. Phys.*, 229:1536–1557, 2009.

[50] S. Foucart. A note on guaranteed sparse recovery via l_1-minimization. *Appl. Comput. Harmon. Anal.*, 29(1):97–103, 2010.

[51] S. Foucart and M.-J. Lai. Sparsest solutions of underdetermined linear systems via l_q-minimization for $0 < q \leq 1$. *Appl. Comput. Harmon. Anal.*, 26(3):395–407, 2009.

[52] H. Fukunaga, J. Takimoto, and M. Doi. A coarse-graining procedure for flexible polymer chains with bonded and nonbonded interactions. *J. Chem. Phys.*, 116(18):8183–8190, 2002.

[53] B. Ganapathysubramanian and N. Zabaras. Sparse grid collocation schemes for stochastic natural convection problems. *J. Comput. Phys.*, 225(1):652–685, 2007.

[54] M. J. Gander, L. Halpern, and F. Nataf. Optimal schwarz waveform relaxation for the one dimensional wave equation. *SIAM J. Num. Anal.*, 41(5):1643–1681, 2003.

[55] M. J. Gander, F. Magoules, and F. Nataf. Optimized schwarz methods without overlap for the helmholtz equation. *SIAM J. Sci. Comput.*, 24(1):38–60, 2002.

[56] L. Gerardo-Giorda, F. Nobile, and C. Vergara. Analysis and optimization of robin-robin partitioned procedures in fluid-structure interaction problems. *SIAM J. Num. Anal.*, 48(6):2091–2116, 2010.

[57] R. G. Ghanem and P. D. Spanos. *Stochastic finite elements: a spectral approach*. Springer-Verlag, New York, 1991.

[58] T. Goldstein and S. Osher. The split Bregman method for L1-regularized problems. *SIAM J. Imaging Sci.*, 2(2):323–343, 2009.

[59] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming. `http://cvxr.com/cvx/`.

[60] M. Griebel. *Sparse grids and related approximation schemes for higher-dimensional problems*. Proceedings of the conference on Foundations of Computational Mathematics, Santander, Spain, 2005.

[61] I. V. Grigoriev, Y. A. Makhnovskii, A. M. Berezhkovskii, and V. Y. Zitserman. Kinetics of escape through a small hole. *J. Chem. Phys.*, 116(22):9574–9577, 2002.

[62] R. D. Groot and P. B. Warren. Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation. *J. Chem. Phys.*, 107(11):4423–4435, 1997.

[63] H. Haario, M. Laine, A. Mira, and E. Saksman. Dram: efficient adaptive mcmc. *Stat. Comput.*, 16(4):339–354, 2006.

[64] H. Haario, E. Saksman, and J. Tamminen. An adaptive metropolis algorithm. *Bernoulli*, pages 223–242, 2001.

[65] T. Hagstrom, R. P. Tewarson, and A. Jazcilevich. Numerical experiments on a domain decomposition algorithm for nonlinear elliptic boundary value problems. *Appl. Math. Lett.*, 1(3):299–302, 1988.

[66] M. Hansen and C. Schwab. Analytic regularity and best N-term approximation of high dimensional parametric initial value problems. Technical Report 2011-64, Seminar for Applied Mathematics, ETHZ, 2011.

[67] V. A. Harmandaris, N. P. Adhikari, N. F. A. van der Vegt, and K. Kremer. Hierarchical modeling of polystyrene: From atomistic to coarse-grained simulations. *Macromolecules*, 39(19):6708–6719, 2006.

[68] C. Hijon, P. Espanol, E. Vanden-Eijnden, and R. Delgado-Buscalioni. Mori-Zwanzig formalism as a practical computational tool. *Faraday Discuss.*, 144:301–322, 2010.

[69] C.-W. Ho, A. E. Ruehli, and P. A. Brennan. The modified nodal approach to network analysis. *IEEE Trans. Circuits Syst.*, 22(6):504–509, Jun 1975.

[70] V. H. Hoang and C. Schwab. N-term Galerkin Wiener chaos approximations of elliptic PDEs with lognormal Gaussian random inputs. Technical Report 2011-59, Seminar for Applied Mathematics, ETHZ, 2011.

[71] W. Hoeffding. A class of statistics with asymptotically normal distributions. *Annals of Math. Statist.*, 19:293–325, 1948.

[72] D. Holcman and Z. Schuss. Escape through a small opening: receptor trafficking in a synaptic membrane. *J. Stat. Phys.*, 117(5-6):975–1014, 2004.

[73] P. J. Hoogerbrugge and J. M. V. A. Koelman. Simulating microscopic hydrodynamic phenomena with dissipative particle dynamics. *Europhys. Lett.*, 19(3):155–160, 1992.

[74] J. D. Jakeman, R. Archibald, and D. Xiu. Characterization of discontinuities in high-dimensional stochastic problems on adaptive sparse grids. *J. Comput. Phys.*, 230(10):3977–3997, 2011.

[75] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted eno schemes. *J. Comput. Phys.*, 126:202–228, 1996.

[76] C. Jin, X.-C. Cai, and C. Li. Parallel domain decomposition methods for stochastic elliptic equations. *SIAM J. Sci. Comput.*, 29(5):2096–2114, 2007.

[77] W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives. Development and testing of the opls all-atom force field on conformational energetics and properties of organic liquids. *J. Am. Chem. Soc.*, 118(45):11225–11236, 1996.

[78] G. Karniadakis and S. Sherwin. *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, 2013.

[79] M. Khajehnejad, W. Xu, A. Avestimehr, and B. Hassibi. Improved sparse recovery thresholds with two-step reweighted l_1 minimization. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, pages 1603 –1607, june 2010.

[80] T. Kinjo and S. Hyodo. Linkage between atomistic and mesoscale coarse-grained simulation. *Mol. Simulat.*, 33(4-5):417–420, 2007.

[81] T. Kinjo and S. A. Hyodo. Equation of motion for coarse-grained simulation based on microscopic description. *Phys. Rev. E*, 75(5):051109, 2007.

[82] S. H. L. Klapp, D. J. Diestler, and M. Schoen. Why are effective potentials "soft"? *J. Phys-Condens. Mat.*, 16(41):7331–7352, 2004.

[83] R. G. Larson. *The Structure and Rheology of Complex Fluids*. Oxford University Press, 1998.

[84] J. L. Lebowitz and H. Spohn. A gallavotti–cohen-type symmetry in the large deviation functional for stochastic dynamics. *J. Stat. Phys.*, 95(1-2):333–365, 1999.

[85] A. W. Lees and S. F. Edwards. The computer study of transport processes under extreme conditions. *J. Phys. C*, 5:1921–1928, 1972.

[86] H. Lei, B. Caswell, and G. E. Karniadakis. Direct construction of mesoscopic models from microscopic simulations. *Phys. Rev. E*, 81:026704, 2010.

[87] H. Lei and G. E. Karniadakis. Probing vasoocclusion phenomena in sickle cell anemia via mesoscopic simulations. *Proc. Natl. Acad. Sci. USA*, 110(28):11326–11330, 2013.

[88] H. Lei and G. E. Karnidakis. Quantifying the rheological and hemodynamic characteristics of sickle cell anemia. *Biophys. J.*, 102:185–194, 2012.

[89] J. Li, J. Li, and D. Xiu. An efficient surrogate-based method for computing rare failure probability. *J. Comput. Phys.*, 230(24):8683–8697, 2011.

[90] J. Li and D. Xiu. On numerical properties of the ensemble kalman filter for data assimilation. *Comput. Methods Appl. Mech. and Engrg.*, 197(43):3574–3583, 2008.

[91] J. Li and D. Xiu. A generalized polynomial chaos based ensemble kalman filter with high accuracy. *J. Comput. Phys.*, 228(15):5454–5469, 2009.

[92] J. Li and D. Xiu. Evaluation of failure probability via surrogate models. *J. Comput. Phys.*, 229(23):8966–8980, 2010.

[93] W. Li, G. Lin, and D. Zhang. An adaptive anova-based pckf for high-dimensional nonlinear inverse modeling. *J. Comput. Phys.*, 258:752–772, 2014.

[94] X. Li. Finding deterministic solution from underdetermined equation: large-scale performance variability modeling of analog/rf circuits. *IEEE Trans. Comput-Aided Design Integr. Circuits Syst.*, 29(11):1661–1668, 2010.

[95] Z. Li, G.-H. Hu, Z.-L. Wang, Y.-B. Ma, and Z.-W. Zhou. Three dimensional flow structures in a moving droplet on substrate: A dissipative particle dynamics study. *Phys. Fluids*, 25(7):072103, 2013.

[96] M. Lighthill. The flow behind a stationary shock. *Philos. Mag.*, 40:214–220, 1949.

[97] G. Lin, C.-H. Su, and G. E. Karniadakis. Random roughness enhances lift in supersonic flow. *Phys. Rev. Lett.*, 99:104501, Sep 2007.

[98] G. Lin, C.-H. Su, and G. E. Karniadakis. Stochastic modeling of random roughness in shock scattering problems: theory and simulations. *Comput. Methods Appl. Mech. Engrg.*, 197(43-44):3420–3434, 2008.

[99] G. Lin, A. M. Tartakovsky, and D. M. Tartakovsky. Uncertainty quantification via random domain decomposition and probabilistic collocation on sparse grids. *J. Comput. Phys.*, 229(19):6995–7012, 2010.

[100] P.-L. Lions. On the schwarz alternating method. i. In *First international symposium on domain decomposition methods for partial differential equations*, pages 1–42. Paris, France, 1988.

[101] P.-L. Lions. On the schwarz alternating method. iii: a variant for nonoverlapping subdomains. In *Third international symposium on domain decomposition methods for partial differential equations*, volume 6, pages 202–223. SIAM, Philadelphia, PA, 1990.

[102] A. A. Louis, P. G. Bolhuis, J. P. Hansen, and E. J. Meijer. Can polymer coils be modeled as "soft colloids"? *Phys. Rev. Lett.*, 85(12):2522–2525, 2000.

[103] D. Lucor and G. E. Karniadakis. Adaptive generalized polynomial chaos for nonlinear random oscillators. *SIAM J. Sci. Comput.*, 26(2):720–735, 2004.

[104] D. Lucor and G. E. Karniadakis. Adaptive generalized polynomial chaos for nonlinear random oscillators. *SIAM J. Sci. Comput.*, 26(2):720–735, 2004.

[105] X. Ma and N. Zabaras. An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. *J. Comput. Phys.*, 228(8):3084–3113, 2009.

[106] X. Ma and N. Zabaras. An efficient bayesian inference approach to inverse problems based on an adaptive sparse grid collocation method. *Inv. Prob.*, 25(3):035013, 2009.

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

[108] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.*, 41(12):3397–3415, 1993.

[109] Y. M. Marzouk and H. N. Najm. Dimensionality reduction and polynomial chaos acceleration of bayesian inference in inverse problems. *J. Comput. Phys.*, 228(6):1862–1902, 2009.

[110] Y. M. Marzouk, H. N. Najm, and L. A. Rahn. Stochastic spectral methods for efficient bayesian solution of inverse problems. *J. Comput. Phys.*, 224(2):560–586, 2007.

[111] T. F. Miller, E. Vanden-Eijnden, and D. Chandler. Solvent coarse-graining and the string method applied to the hydrophobic collapse of a hydrated chain. *Proc. Natl. Acad. Sci. USA*, 104(37):14559–14564, 2007.

[112] H. Mori. Transport, collective motion, and Brownian motion. *Prog. Theor. Phys*, 33:423, 1965.

[113] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24(2):227–234, 1995.

[114] D. Needell. Noisy signal recovery via iterative reweighted l_1 minimization. In *Proc. Asilomar Conf. on Signal Systems and Computers*, Pacific Grove, CA, 2009.

[115] D. Needell and J. A. Tropp. CoSaMP: iterative signal recovery from incomplete and inaccurate samples. *Appl. Comput. Harmon. Anal.*, 26(3):301–321, 2009.

[116] F. Nobile, R. Tempone, and C. G. Webster. An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Num. Anal.*, 46(5):2411–2442, 2008.

[117] F. Nobile, R. Tempone, and C. G. Webster. An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.*, 46(5):2411–2442, 2008.

[118] B. Øksendal. *Stochastic differential equations*. Springer, 2003.

[119] Y. C. Pati, R. Rezaiifar, and P. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 40–44. IEEE, 1993.

[120] S. Pillay, M. J. Ward, A. Peirce, and T. Kolokolnikov. An asymptotic analysis of the mean first passage time for narrow escape problems: Part i: Two-dimensional domains. *Multiscale Model. Simul.*, 8(3):803–835, 2010.

[121] I. V. Pivkin and G. E. Karniadakis. Accurate coarse-grained modeling of red blood cells. *Phys. Rev. Lett.*, 101(11):118105, 2008.

[122] I. V. Pivkin, P. D. Richardson, and G. E. Karniadakis. Effect of red blood cells on platelet aggregation. *IEEE, Eng. Med. Biol. Mag.*, 28(2):32 –37, march-april 2009.

[123] H. Rauhut and R. Ward. Sparse legendre expansions via l_1-minimization. *J. Approx. Theory*, 164(5):517–533, May 2012.

[124] W. Ren and E. Vanden-Eijnden. Finite temperature string method for the study of rare events. *J. Phys. Chem. B*, 109(14):6688–6693, 2005.

[125] W. Ren, E. Vanden-Eijnden, P. Maragakis, E. Weinan, et al. Transition pathways in complex systems: Application of the finite-temperature string method to the alanine dipeptide. *J. Chem. Phys.*, 123(13):134109, 2005.

[126] F. Rizzi, H. N. Najm, B. J. Debusschere, K. Sargsyan, M. Salloum, H. Adalsteinsson, and O. M. Knio. Uncertainty quantification in MD simulation. Part I: Forward propagation. *Multiscale Model. Simul.*, 10(4):1428–1459, 2012.

[127] F. Rizzi, H. N. Najm, B. J. Debusschere, K. Sargsyan, M. Salloum, H. Adalsteinsson, and O. M. Knio. Uncertainty quantification in MD simulation. Part II: Bayesian inference of force-field parameters. *Multiscale Model. Simul.*, 10(4):1460–1492, 2012.

[128] A. Sarkar, N. Benabbou, and R. Ghanem. Domain decomposition of stochastic pdes: theoretical formulations. *Int. J. Num. Meth. Engrg.*, 77(5):689–701, 2009.

[129] Z. Schuss, A. Singer, and D. Holcman. The narrow escape problem for diffusion in cellular microdomains. *Proc. Natl. Acad. Sci.*, 104(41):16098–16103, 2007.

[130] A. Singer, Z. Schuss, and D. Holcman. Narrow escape, part ii: The circular disk. *J. Stat. Phys.*, 122(3):465–489, 2006.

[131] A. Singer, Z. Schuss, and D. Holcman. Narrow escape, part iii: Non-smooth domains and riemann surfaces. *J. of Stat. Phys.*, 122(3):491–509, 2006.

[132] A. Singer, Z. Schuss, D. Holcman, and R. Eisenberg. Narrow escape, part i. *J. Stat. Phys.*, 122(3):437–463, 2006.

[133] I. Sloan. When are Quasi-Monte Carlo algorithms efficient for high-dimensional integrals? *J. Complexity*, 14:1–33, 1998.

[134] I. Sobol'. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Math. Comput. Simulation*, 55:271–280, 2001.

[135] N. A. Spenley. Scaling laws for polymers in dissipative particle dynamics. *Europhys. Lett.*, 49(4):534, 2000.

[136] T. Spyriouni, I. G. Economou, and D. N. Theodorou. Molecular simulation of α-olefins using a new united-atom potential model:. vapor-liquid equilibria of pure compounds and mixtures. *J. Am. Chem. Soc.*, 121(14):3407–3413, 1999.

[137] H. Stark and J. W. Woods. *Probability, random processes, and estimation theory for engineers.* Prentice-Hall Englewood Cliffs (NJ), 1986.

[138] B. Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliab. Engrg. Syst. Safe.*, 93(7):964–979, 2008.

[139] V. Symeonidis, G. E. Karniadakis, and B. Caswell. Dissipative particle dynamics simulations of polymer chains: Scaling laws and shearing response compared to DNA experiments. *Phys. Rev. Lett.*, 95(7):076001, 2005.

[140] V. N. Temlyakov. Greedy algorithms and m-term approximation with regard to redundant dictionaries. *J. Approx. Theory*, 98(1):117–145, 1999.

[141] V. N. Temlyakov. Weak greedy algorithms. *Adv. Comput. Math.*, 12(2-3):213–227, 2000.

[142] R. A. Todor and C. Schwab. Convergence rates for sparse chaos approximations of elliptic problems with stochastic coefficients. *IMA J. Numer. Anal.*, 27(2):232–261, 2007.

[143] A. Toselli and O. Widlund. *Domain decomposition methods: algorithms and theory*, volume 3. Springer, 2005.

[144] J. A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. Inf. Theory*, 50(10):2231–2242, 2004.

[145] A. Valli, A. Quarteroni, et al. *Domain decomposition methods for partial differential equations*. Number CMCS-BOOK-2009-019. Oxford University Press, 1999.

[146] E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.*, 31(2):890–912, 2008.

[147] S. S. Varadhan, S. S. Varadhan, and S. S. Varadhan. *Large deviations and applications*, volume 46. SIAM, 1984.

[148] D. Venturi, X. Wan, and G. E. Karniadakis. Stochastic low-dimensional modelling of a random laminar wake past a circular cylinder. *J. Fluid Mech.*, 606:339–367, 2008.

[149] X. Wan and G. E. Karniadakis. An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *J. Comput. Phys.*, 209(2):617–642, 2005.

[150] X. Wan and G. E. Karniadakis. Multi-element generalized polynomial chaos for arbitrary probability measures. *SIAM J. Sci. Comput.*, 28(3):901–928, 2006.

[151] X. Wang and I. Sloan. Why are high-dimensional finance problems often of low effective dimension? *SIAM J. Sci. Comp.*, 27(1):159–183, 2005.

[152] E. Weinan. *Principles of multiscale modeling*. Cambridge University Press, 2011.

[153] E. Weinan, W. Ren, and E. Vanden-Eijnden. String method for the study of rare events. *Phys. Rev. B*, 66(5):052301, 2002.

[154] E. Weinan, W. Ren, and E. Vanden-Eijnden. Simplified and improved string method for computing the minimum energy paths in barrier-crossing events. *J. Chem. Phys.*, 126(16):164103, 2007.

[155] C. Winter, A. Guadagnini, D. Nychka, and D. Tartakovsky. Multivariate sensitivity analysis of saturated flow through simulated highly heterogeneous groundwater aquifers. *J. Comput. Phys.*, 217:166–175, 2009.

[156] D. Xiu and J. S. Hesthaven. High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput.*, 27(3):1118–1139, 2005.

[157] D. Xiu and G. E. Karniadakis. The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.*, 24(2):619–644, 2002.

[158] D. Xiu and D. M. Tartakovsky. Numerical methods for differential equations in random domains. *SIAM J. Sci. Comput.*, 28(3):1167–1185, 2006.

[159] H. Xu and S. Rahman. A generalized dimension-reduction method for multidimensional integration in stochastic mechanics. *Int. J. Numer. Methods Eng.*, 61(12):1992 – 2019, 2004.

[160] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Rev.*, 34(4):581–613, 1992.

[161] J. Xu and J. Zou. Some nonoverlapping domain decomposition methods. *SIAM Rev.*, 40(4):857–914, 1998.

[162] W. Xu, M. Khajehnejad, A. Avestimehr, and B. Hassibi. Breaking through the thresholds: an analysis for iterative reweighted l_1 minimization via the grassmann angle framework. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 5498 –5501, march 2010.

[163] L. Yan, L. Guo, and D. Xiu. Stochastic collocation algorithms using l_1-minimization. *Int. J. Uncertain. Quantif.*, 2(3):279–293, 2012.

[164] X. Yang, M. Choi, G. Lin, and G. E. Karniadakis. Adaptive ANOVA decomposition of stochastic incompressible and compressible flows. *J. Comput. Phys.*, 231(4):1587 – 1614, 2012.

[165] X. Yang and G. E. Karniadakis. Reweighted ℓ_1 minimization method for stochastic elliptic differential equations. *J. Comput. Phys.*, 248(1):87 – 108, 2013.

[166] W. Yin, S. Osher, D. Goldfarb, and J. Darbon. Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing. *SIAM J. Imaging Sci.*, 1(1):143–168, 2008.

[167] K. Zhang, R. Zhang, Y. Yin, and S. Yu. Domain decomposition methods for linear and semilinear elliptic stochastic partial differential equations. *Appl. Math. Comput.*, 195(2):630–640, 2008.

[168] Z. Zhang, M. Choi, and G. Karniadakis. Anchor points matter in ANOVA decomposition. In *Spectral and High Order Methods for Partial Differential Equations Lecture Notes in Computational Science and Engineering*, volume 76, pages 347–355. Springer, 2011.

[169] Z. Zhang, M. Choi, and G. E. Karniadakis. Error estimates for the ANOVA method with polynomial chaos interpolation: Tensor product functions. *SIAM J. Sci. Comput.*, 34(2):A1165–A1186, 2012.

[170] Z. Zhang, T. El-Moselhy, P. Maffezzoni, I. Elfadel, and L. Daniel. Efficient uncertainty quantification for the periodic steady state of forced and autonomous circuits. *IEEE Trans. Circuits-II*, 60(10):687–691, Oct 2013.

[171] Z. Zhang, T. A. El-Moselhy, I. M. Elfadel, and L. Daniel. Stochastic testing method for transistor-level uncertainty quantification based on generalized polynomial chaos. *IEEE Trans. Comput-Aided Design Integr. Circuits Syst.*, 32(10):1533–1545, 2013.

[172] Z. Zhang, I. Elfadel, and L. Daniel. Uncertainty quantification for integrated circuits: Stochastic spectral methods. In *Computer-Aided Design (ICCAD), 2013 IEEE/ACM International Conference on*, pages 803–810, Nov 2013.

[173] R. Zwanzig. Ensemble method in the theory of irreversibility. *J. Chem. Phys.*, 33:1338, 1960.