ADAPTIVE CONSTRUCTION OF SURROGATE FUNCTIONS FOR VARIOUS COMPUTATIONAL MECHANICS MODELS

by

Anindya Bhaduri

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

September, 2019

Abstract

In most science and engineering fields, numerical simulation models are often used to replicate physical systems. An attempt to imitate the true behavior of complex systems results in computationally expensive simulation models. The models are more often than not associated with a number of parameters that may be uncertain or variable. Propagation of variability from the input parameters in a simulation model to the output quantities is important for better understanding the system behavior. Variability propagation of complex systems requires repeated runs of costly simulation models with different inputs, which can be prohibitively expensive. Thus for efficient propagation, the total number of model evaluations needs to be as few as possible. An efficient way to account for the variations in the output of interest with respect to these parameters in such situations is to develop black-box surrogates. It involves replacing the expensive high-fidelity simulation model by a much cheaper model (surrogate) using a limited number of the high-fidelity simulations on a set of points called the design of experiments (DoE).

ABSTRACT

The obvious challenge in surrogate modeling is to efficiently deal with simulation models that are expensive and contains a large number of uncertain parameters. Also, replication of different types of physical systems results in simulation models that vary based on the type of output (discrete or continuous models), extent of model output information (knowledge of output or output gradients or both), and whether the model is stochastic or deterministic in nature. All these variations in information from one model to the other demand development of different surrogate modeling algorithms for maximum efficiency.

In this dissertation, simulation models related to application problems in the field of solid mechanics are considered that belong to each one of the above-mentioned classes of models. Different surrogate modeling strategies are proposed to deal with these models and their performance is demonstrated and compared with existing surrogate modeling algorithms. The developed algorithms, because of their non-intrusive nature, can be easily extended to simulation models of similar classes, pertaining to any other field of application.

Advisor: Lori Graham-Brady

Co-advisor: Michael D. Shields

Reader: Tamas Budavari

Acknowledgments

First of all, I am deeply grateful and privileged to have worked with an extraordinary advisor, Dr. Lori Graham-Brady. She gave me the freedom to explore new ideas, however unconventional or offbeat they may be, always encouraging discussions and providing valuable feedback. That instilled in me a great deal of confidence as a researcher and helped me move forward in my Ph.D. journey. Her kind and considerate personality, along with her amazing work-life balance, is something which is really admirable, and something that I would like to emulate in my future life.

I would like to sincerely thank my co-advisor Dr. Michael D. Shields. I started working with him from my second year and I have learnt a lot from the discussions in our three-way meetings that we had every alternate Thursday. In my first semester here at Johns Hopkins, I did a course titled “Probabilistic Methods in Civil Engineering” which allowed me to gain a lot of interest in the field of uncertainty propagation. That course helped me shape and mold my research interest and, in turn, the current body of work.

ACKNOWLEDGMENTS

I have had the pleasure of working with some really amazing collaborators during my Ph.D. years. My first collaboration was with Dr. Yanyan He (currently a faculty at New Mexico Institute of Mining and Technology) and Dr. Mike Kirby from the University of Utah. I then had the opportunity to work with Dr. Jasmine Gardner (currently a postdoctoral researcher at Uppsala University) and Dr. Cameron Abrams from Drexel University. My third collaboration involved David Brandyberry and Dr. Philippe Geubelle from University of Illinois at Urbana-Champaign. For my last project, I also worked with Christopher Meyer, Dr. Bazle Haque and Dr. Jack Gillespie from the University of Delaware. It goes without saying, that without the contributions of these people, most of my Ph.D. work would not have been possible.

Many thanks to the Civil Engineering department staff members: Lisa, Shamija, Amanda, Debbie, Carla, and Jess as well as the Hopkins Extreme Materials Institute (HEMI) staff members: Katie, Bess, Scott, and Matt for all their help throughout these years.

I would like to thank my DQE, GBO and thesis committee members: Dr. Somnath Ghosh, Dr. Sauleh Siddiqui, Dr. Noah Cowan, and Dr. Tamas Budavari, for taking time out of their busy schedule to provide valuable suggestions and feedback.

I would like to thank the Civil Engineering Graduate Association (CEGA) and all my departmental colleagues. CEGA has been instrumental in making

this department a close-knit group. It has been amazing in organizing events such as numerous movie nights, fall trips, coffee hours, and happy hours over the years. Through these events, I got the opportunity to interact with many departmental colleagues and have made some really good friends in the process.

I would like to thank the past and present members in my research group: Amartya, Noah, Ashwini, Audrey, Farah, Nicolas, David, Junwei, Kirubel, and Mehmet. I have had the chance to engage in a lot of insightful discussions with them that were undeniably helpful for my research. Their inputs and suggestions during our weekly group meetings have been invaluable.

I consider myself fortunate to have been surrounded by a lot of caring and accommodating friends during my stay here. Subhendu, Shinu, Jaseem and Rajiv were my first apartment mates in Baltimore. Their guidance during those initial months really helped me settle down in this new city. I have made some great friends in the department: Jimi, Gary, Dave, Hwanpyo, Sundar, Astrid, and Guanbo and would like to thank them for their constant support. I would like to thank my friends at Johns Hopkins (in no particular order): Sayak, Debmalya, Arijit, Ikbal, Suman, Sayantan, Debadrita, Roshni, Rikhi, Vivek, Subhro, Amartya, Paramita, Debojyoti, Pallabi, Vishal, Sayan, Neha, Prosenjit, Soumya and Debangan. I would also like to acknowledge my friends outside Johns Hopkins (again, in no particular order): Subhabrata, Suvodeep,

ACKNOWLEDGMENTS

Oindrila, Sagnika, Mainak and Ahana. All of these people have provided great support to me during these years. I have gotten tons of valuable advice from them, gained new perspectives in life from numerous discussions with them, and received constant inspiration and motivation from them.

I would like to sincerely thank the people who matter to me the most, my father Mr. Asim Bhaduri, my mother Mrs. Mitali Bhaduri and my sister Mrs. Debaparna Bhaduri-Mukherjee for simply being there. When I look back, I realize that I probably had the best upbringing that one could ask for, and I would like to acknowledge all their sacrifices and efforts in helping me get wherever I am today. All the lessons I learnt from them since childhood have shaped me into a decent (I hope) individual. I would also like to express my gratitude to all my relatives, including my grandparents, aunts, uncles and cousins.

Finally, I wish to thank the Army Research Laboratory for sponsoring my research, accomplished under Cooperative Agreement Number W911NF-12-2-0023 and W911NF-12-2-0022.

Dedication

To Didi, Maa, Baba and Taton

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Simulation models

In the field of solid mechanics, physical experiments help in observing, understanding and characterizing different important physical phenomena. Information obtained from experiments can play an useful role in enhancing the performance of materials by testing under varying conditions. For example, performance of brittle materials such as ceramics, rock, and ice, under compressive dynamic loads, can be improved by understanding how different conditions and the observed mechanisms drive the failure processes, and then proposing an improved design at the structural or the material level. The variability in the conditions from one experiment to the other can arise from different sources, for example, material micro-structure, material properties,

boundary conditions, loading, etc. Assessment of performance under all these variable conditions requires a substantial number of experiments to be performed. However, conducting even a single experiment can sometimes be very expensive, involving significant time and man power, and it is practically infeasible to execute more than a small number of experiments. In experiments, it is also difficult to explicitly control some of the variabilities, for example, the spatial variation of properties in a material specimen. Thus, repeating the same experiment twice with all controllable parameters identical, can give rise to two different outcomes, however small the difference may be. Usage of computational models can serve as an useful complement to the experimental studies. An integrative computational model can be built by combining known input properties and different physical laws governing the relevant mechanisms, with a goal to replicate a complex system response based on experimental observations. A reliable simulation model can be used to better understand the underlying physics of the phenomenon under study. By tweaking different input parameters, valuable knowledge can be extracted about the system under study in order to achieve optimal performance.

1.2 Need for surrogates

Different physical systems can be represented by simulation models of varying complexity, ranging from simple analytical functions to complex partial differential equations that rely on solvers such as finite element or finite difference methods. A continuous increase in computational power has allowed the consideration of phenomena of increasing level of complexity, and also the ability to explore high-dimensional input parameter spaces for achieving optimal performance of the system. However, the above two capabilities are counteracting in the sense that making the model more and more complex reduces its capability to be used for exploratory analysis in the input parameter space. A sophisticated but computationally expensive model, with a potentially large number of parameters explaining the phenomenon of interest, limits the quantification of the parameter influences on the simulator response. For instance, in the automotive industry, expensive simulation models for modeling the behavior of cars during crash-tests exist that require many parameters to capture the highly non-linear mechanical behavior of the system. Thus, it is a significant challenge to use these models for the design of car structures, which require numerous model runs to achieve optimal test performance. In reality, simulation models are validated using finite experimental observations, and in the process, some influence factors tend to be neglected in the model. Thus, the

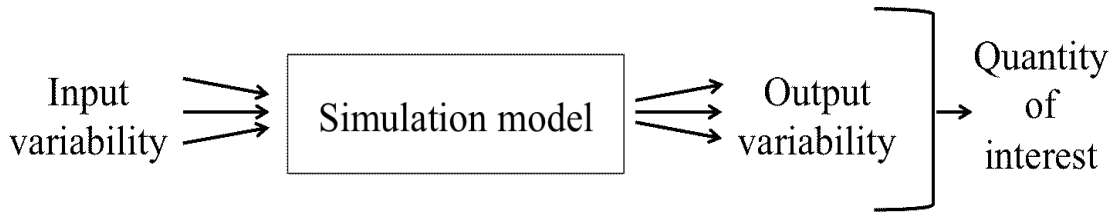


Figure 1.1: Variability propagation using a simulation model

model, however sophisticated and complex it may be, can approximate the real behaviour of the physical system only up to a certain level of accuracy. Due to uncertainties, such as, the inherently stochastic nature of some model parameters, boundary conditions or excitations, and a lack of understanding of the true physics, predictions inevitably deviate from reality. Thus, when it comes down to the choice of a simulation model, there needs to be a trade-off between the accuracy of the model and its computational cost. The model should not be too expensive to be able to run for a significant number of times at different parameter combinations, nor too simple to produce incorrect behaviour leading to erroneous interpretations of the system under study. Thus, the true purpose of using a simulation model is to roughly understand the physical process and also make some related decisions, which is not feasible experimentally.

For a given model with a number of variable input parameters, it may be of interest to assess the corresponding variation in the model output or outputs with a goal of estimating a quantity of interest as shown in figure 1.1. For example, the input parameters can be considered uncertain with a certain distribution and the uncertainty can be propagated through the simulation model

to characterize the corresponding output uncertainties. In this case, a relevant quantity of interest can be the probability density function (PDF) of the output. If the model can be solved cheaply and the number of input parameters is small, the quantity of interest can be estimated by repeated runs of the simulation model. However, if either the model is expensive to solve, or the number of parameters are large, or both, then it becomes impractical to extensively explore the parameter space by evaluating the model for a large number of parameter samples. An efficient way to reduce the overall high computational cost is to use surrogate models, or metamodels. Surrogate models produce computationally cheap mathematical approximations of the input/output relationship for expensive models, using only limited number of model evaluations as shown in figure 1.2. Once a surrogate function is built, outputs can be generated at a fractional cost compared to that obtained by running the actual model, which enables an efficient estimation of the quantity of interest [1–4]. It is, however, important to note that the usefulness of the information obtained from surrogate functions is highly dependent on the accuracy of the simulation model.

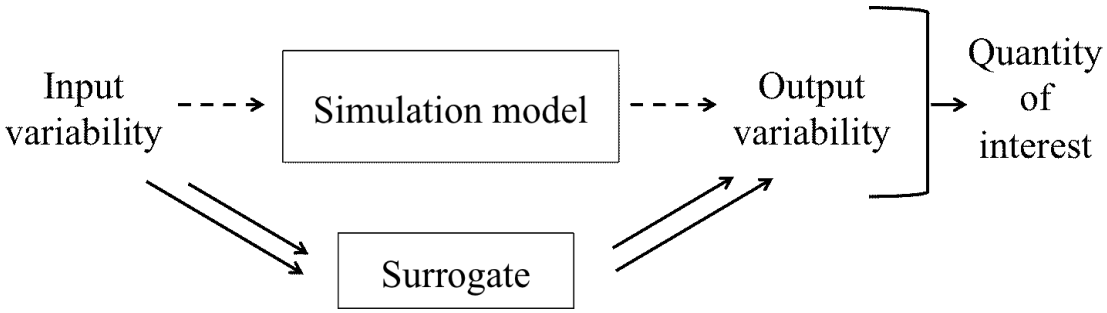


Figure 1.2: Use of surrogates for propagation of variability

Intrusive	Non-intrusive	Semi-intrusive
gPC-Galerkin	Non-intrusive polynomial chaos	POD-Galerkin
gSD-Galerkin	Stochastic collocation interpolation	Hierarchical
	Kriging	
	Support vector machines (SVM)	

Figure 1.3: Classification of surrogates with examples

1.3 Types of surrogates

Surrogates can be generally classified into 3 types (shown in figure 1.3): intrusive, non-intrusive and semi-intrusive. Intrusive surrogates require knowledge of the governing equations of the simulation code for their implementation. On the other hand, non-intrusive surrogates, also known as "black box" surrogates, are sampling based and depend only on the input and output data of the simulation model with no utilization of information about the inner workings of the model or the behaviour of the system. Semi-intrusive surrogates involve both working with the governing equations of the simulation model as well as generating input/output data by explicitly sampling the model. A class of partially non-intrusive surrogates also exists where qualitative trends about the system behaviour can be known from expert opinion [5]. For instance, if it is known in advance that the model output should have a linear variation with the different input parameters, then use of linear surrogates can be very effective. A brief overview of the three categories of surrogate modeling methodologies is given in the following subsections.

1.3.1 Intrusive surrogates

In an intrusive surrogate construction, the solution is represented as a spectral expansion of a finite-dimensional basis function, which is a function of the

random parameters describing the uncertainties. The basis function has associated deterministic coefficients which are unknown. Stochastic Galerkin methods [6, 7] project the given differential equation (governing equation) of a simulation model with random parameters in the continuous infinite-dimensional space onto a finite-dimensional subspace defined in terms of the basis functions. This results in a coupled system of deterministic equations which needs to be solved to obtain the expansion coefficients. The idea behind a Galerkin projection is to minimize the error of the truncated expansion.

Generalized polynomial chaos (gPC) expansion based stochastic galerkin methods (gPC-Galerkin) [8–11] typically use global orthogonal polynomial basis (fixed a *priori*) to represent the solution. The goal is to find a solution in the space of the orthogonal polynomials such that the residue of the governing equations is orthogonal to the space of polynomials. However, the basis polynomials need not be globally smooth. When the solution is non-smooth, wavelet basis [12] or piecewise-polynomial basis [13] are often used. Multi-element [14, 15] approaches are also used with globally smooth polynomials for non-smooth cases. In a generalized spectral decomposition based Galerkin projection approach (gSD-Galerkin) [16], the solution is represented in a low-dimensional basis space where the stochastic basis functions are not known a *priori*, contrary to the gPC approach. The basis is determined along with the unknown deterministic coefficients by solving an "extended" eigenvalue prob-

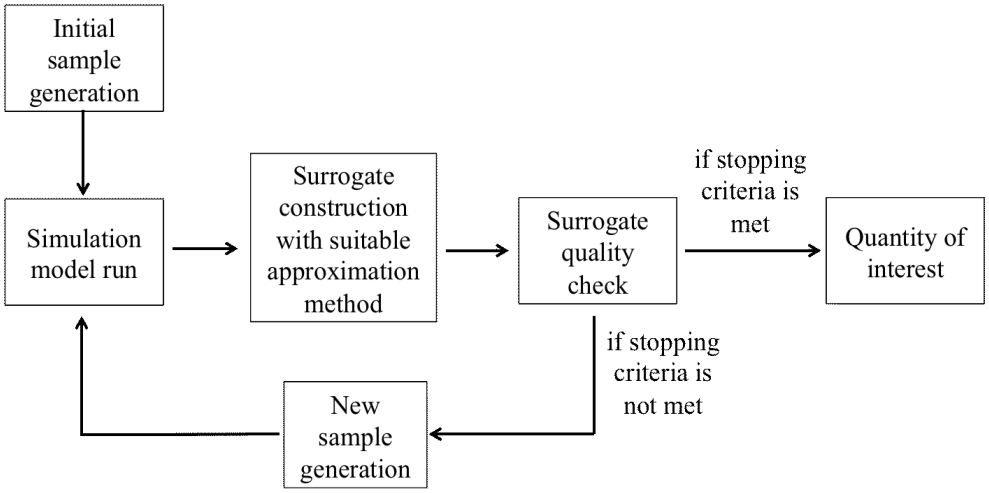


Figure 1.4: General framework of black-box surrogate modeling

lem.

1.3.2 Non-intrusive surrogates

Non-intrusive surrogates are black-box surrogates which involve an empirical approximation of the simulation model based on simulation runs at a set of samples generated in the input parameter space through various sampling techniques [17]. Figure 1.4 illustrates the general concept behind the construction of a black-box surrogate modeling algorithm. Construction of such a surrogate involves the following steps [18,19]: Design of Experiments (DOE), numerical simulation run, surrogate construction, and surrogate quality validation. If there is adaptivity involved in the construction, a surrogate quality check step is usually there which provides feedback from the current surrogate to the design of experiments. DOE involves strategies for sample selec-

tion used for simulation runs [20]. The generation of input/output data is followed by the surrogate construction step using surrogate models, which may be parametric approximation methods, such as polynomial regression [21], non-intrusive polynomial chaos expansion (PCE) [22], and Kriging [23–25]; or non-parametric methods, such as projection-pursuit regression [26, 27], stochastic collocation interpolation [22], radial basis functions (RBF) [28]. When the physics of the problem under study is very complex to deal with, these surrogates are preferred because of their non-intrusive nature. When the cost of running a simulation model is high and there is a large number of associated parameters (dimensions), constructing these surrogates can be a challenge and some dimensionality reduction technique should be adopted to reduce the number of parameters of the problem.

Black-box surrogates can be of two kinds based on the type of output produced from the simulation model. When the outputs are continuous, they are called continuous surrogates, and if the outputs are discrete values, they are called discrete surrogates. Discrete surrogates are used for classification purposes, for example, support vector machines (SVM), k-nearest neighbors (k-NN), logistic regression, artificial neural networks (ANN). Continuous surrogates can be either regression based or interpolation based. In regression surrogates, the exact function values values are not necessarily reproduced at the training data points, and are more suitable for dealing with noisy data; for ex-

ample, linear regression. On the other hand, in interpolation surrogates, the surrogate function passes through the training data points exactly; for example, Kriging. A class of gradient enhanced continuous surrogates also exists where the simulation model produces output as well as gradients of the output, for example, gradient enhanced kriging, gradient enhanced regression, gradient enhanced RBF [29]. A rare class of gradient-only surrogates also exist where only the gradients of the outputs (and no output values) are available from the simulation model [30] .

Black-box surrogates can also be categorized into local, multipoint, and global models [19, 31] based on the usage of data points in the surrogate function construction. Local surrogates are built based on the information of a single data point, and the approximation is only valid in the close proximity of that particular point; for example, Taylor series approximation and intervening variables. Global surrogates, on the other hand, use the entire data set to produce surrogate functions valid over the entire input space of interest. Some approximation methods that can build these surrogates include polynomial regression [21], multivariate adaptive regression splines (MARS) [32], support vector machine (SVM) [33], artificial neural network (ANN) [34], kriging [23–25], and RBF [28]. Multipoint surrogates typically uses two or more data points (but not the entire data set) for surrogate construction; for example, two-point exponential approximation (TPEA) [35], two-point adaptive

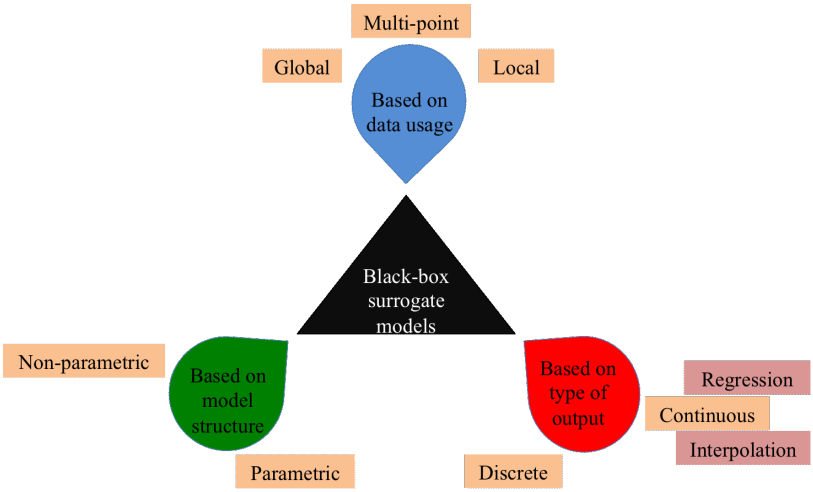


Figure 1.5: Classification of black-box surrogates

nonlinear approximation (TANA) [36], Inverse Distance Weighted (IDW) approach [37]. In fact, if subsets (multiple points) of the entire data set are used to build multiple surrogate patches in the input space, based on any of the above mentioned global approximation methods, like, Kriging or RBF, over the entire input space, such a surrogate construction will fall into the multipoint category.

The different categorizations of black box surrogates discussed above is illustrated in figure 1.5.

1.3.3 Semi-intrusive surrogates

Most reduced-order models and hierarchical models belong to the class of semi-intrusive surrogates. In this type of surrogate construction, changes are made to the governing equations of the simulation model and input/output data

is also generated from the model by treating it as a black box.

Reduced order surrogates use numerical methods to transform the original simulation model to an approximate lower dimensional representation. The solutions of the reduced surrogates are obtained by applying the same Galerkin projection as in the intrusive approach but now in a subspace of lower dimensionality [38, 39]. This reduced subspace is represented via a set of basis vectors, computed using a number of different methods, including Krylov-subspace methods [40–42], approximate balanced truncation [43–47], and proper orthogonal decomposition (POD), also known as Karhunen-Loève expansion in stochastic theory or principle component analysis in statistical analysis [48–52]. This calculation of the reduced subspace usually needs running the original simulation model at different input parameter values, hence the semi-intrusive nature.

Hierarchical surrogates are also known as multi-fidelity, variable-fidelity, or variable-complexity surrogates [53, 54], and they transform any given high-fidelity physics based simulation model to a physics-based models of lower accuracy and reduced computational cost. Different approaches are used for deriving this low-fidelity model; for instance, using simpler models with simplified physics assumptions [55, 56], using the same high-fidelity model but with a coarser grid in a numerical solver like finite element method [57, 58], or using the same high-fidelity models but with a higher residual tolerance [27].

The two models of different fidelity can then be suitably run at desirable set of parameters to eventually obtain the hierarchical surrogate. Here, the derivation of the low-fidelity approach on one hand, and the model runs (of different fidelity) at chosen parameter values on the other, gives the approach a semi-intrusive nature.

1.4 Black-box surrogate formulation

A system is considered whose behaviour is represented by a computational model M which maps the d-dimensional input space D to an s-dimensional output space:

$$M : \mathbf{x} \in D \subset R^d \rightarrow \mathbf{y} \in R^s$$

where $\mathbf{x} = \{x_1, \ldots, x_d\}^T$ is a d-dimensional vector of input variables, D is the design space, and $\mathbf{y} = \{y_1, \ldots, y_s\}^T$ is a s-dimensional vector of output variables. In other words, the input $\mathbf{x}$ and output $\mathbf{y}$ of the simulation model is related by a function M:

$$\mathbf{y} = M(\mathbf{x})$$

It is assumed that M is a black box, implying that the inner workings of the computational model need not be known. This means, for each input vector $\mathbf{x}^{(i)}$, only the corresponding response $\mathbf{y}^{(i)} = M(\mathbf{x}^{(i)})$ is accessible. For example,

when dealing with a complex finite element model (FEM), it might be of interest to just extract outputs for given inputs by ignoring any specifics about the FEM model. In the most general case, the computational model M can be either stochastic or deterministic. In a deterministic model, repeated evaluations with the same input vector $\mathbf{x}^{(0)} \in D$ lead to the same output vector $\mathbf{y}^{(0)} = M(\mathbf{x}^{(0)})$, but in a stochastic model, repeated evaluations with the same input vector $\mathbf{x}^{(0)} \in D$ lead to different output vectors. The present study is restricted to scalar output y and it may either span a continuous space or a set of discrete values. In a black-box surrogate construction, the usually expensive and unknown physics-based function M is replaced by a much simpler function S. This is done by evaluating the output responses at N training points:

$$\mathbf{X} = [\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}]$$
$$\mathbf{y} = [y^{(1)}, \ldots, y^{(N)}]^T = [M(\mathbf{x}^{(1)}), \ldots, M(\mathbf{x}^{(N)})]^T$$

where $\mathbf{X}$ is called the design of experiments (DoE), and $\mathbf{y}$ is the observation vector. A suitable approximation method is then applied to this training data set to estimate the function S relating the inputs and outputs which allows cheap prediction of the system response at any new input point $\mathbf{x}^{(q)}$ such that

$$\mathbf{y} = M(\mathbf{x}^{(q)}) \approx S(\mathbf{x}^{(q)}; \mathbf{X}, \mathbf{y})$$

Sometimes, the model provides the gradients of the output at N training points:

$$\begin{aligned}\mathbf{Y}' &= \left[\left.\frac{dy}{d\mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{(1)}}, \ldots, \left.\frac{dy}{d\mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{(N)}}\right] \\ &= \left[\frac{dy}{d\mathbf{x}^{(1)}}, \ldots, \frac{dy}{d\mathbf{x}^{(N)}}\right]\end{aligned}$$

where

$$\frac{dy}{d\mathbf{x}^{(i)}} = \left[\frac{dy}{dx_1^{(i)}}, \ldots, \frac{dy}{dx_d^{(i)}}\right]^T, \quad \forall i \in \{1, \ldots, N\}$$

As discussed previously in section 1.3.2, there exists simulation models that can provide both the output y and output gradients $\mathbf{Y}'$ at N training points [29] as well as simulation models that provide only the output gradients $\mathbf{Y}'$ and no information about the output y [30]. For both cases, the goal is to build a surrogate of the output and estimate quantities of interest that are functions of the output.

1.5 Sampling designs

An important step in the construction of a black-box surrogate and its subsequent use for prediction, learning or optimization, is the choice of the set of training points in the input sample space. It is worth noting here that the combination of the sampling design and the surrogate construction (approximation) method can be crucial for an efficient surrogate construction. For exam-

ple, let us consider two surrogate methods, non-intrusive PCE and stochastic collocation (SC) interpolation [22]. In PCE, the output is expressed in terms of known orthogonal polynomial basis functions. The unknown coefficients can be computed using either least squares minimization approach or an multidimensional integration approach. Thus both random and structured sampling designs can be used with non-intrusive PCE. On the other hand, in a global setting, SC interpolation expresses the output as a sum of multidimensional Lagrange interpolation polynomials, where the coefficients are essentially known and the multidimensional interpolants need to be formed over structured data sets (tensor or sparse grids) and random sampling designs may not be used in this case.

Sample designs can be broadly divided into three families [59, 60]: classical and space-filling designs, model-oriented (or optimal) designs, and adaptive designs.

1.5.1 Classical And Space-Filling Designs

Classical approaches [3, 61, 62] are based only on geometrical considerations, independent of the choice of the function approximation method (surrogate model) to be used for the training data. In a full-factorial (FF) design, the input samples are placed at the nodes of a regular grid, formed by finite discretizations along each input variable and all possible combinations of the

discrete variables. For example, two-level FF designs is one where each input variable is discretized at the minimum and maximum values, and are typically used for global sensitivity analysis. A few drawbacks of this design include: exponential dependence of the sample number on the number of the input parameters (an FF design with l discretizations in d dimensions results in l^d samples), inadequate space filling properties in high dimensions, and poor subspace projection properties. Other variants of the factorial design include central composite design (circumscribed, inscribed or face-centered type) and fractional factorial design (subsets of FF design) [2]. Smolyak sparse grid design [63] is a hierarchical design whose samples are subsets of the FF design.

The most basic form of random sampling design is Monte Carlo (MC). A major issue with MC samples is the lack of control of the distribution of points in the domain which causes unwanted clustering and scattering of points. Thus it has poor space-filling properties. Latin Hypercube sampling (LHS) [64] is a random design that ensures by construction, good projective properties along each input variable but does not provide good space-filling. Additional space filling criteria can be used to augment the space filling properties of the LHS design; for example, maximin design, maximum-entropy designs [65], etc. Stratified sampling [64] is another random design which tries to achieve good space filling by subdividing the input sample space into multiple strata and generating samples in each of them. However, it suffers from poor projective proper-

ties.

Low discrepancy sequences, also called quasi-random sequences, form another class of sampling design which are sequentially space filling by construction. Faure [66], Niederreiter [67], Halton [68] and Sobol [69] sequences are some common low-discrepancy sequences. For intermediate dimensions, these sequences produce samples with poor projective properties [70]. Various scrambling algorithms [71–73] have been applied to these sequences to improve the uniformity of samples.

1.5.2 Model-Oriented Designs

Samples generated from model-oriented (also called quasi-adaptive) designs are functions of the choice of the approximation method (surrogate model) used to construct the surrogate. When the choice of the approximation method is made a priori, it is possible to choose samples according to the chosen method by optimizing a certain criterion. If, given an approximation method, ϕ is a functional (criterion) which needs to be minimized and depends on the design of experiments $\mathbf{X}$, the design $\mathbf{X}^*$ is called ϕ-optimal if:

$$\mathbf{X}^* = \arg\min[\phi(\mathbf{X})] \tag{1.1}$$

A- and D-optimality [74] attempt to minimize the uncertainty in the parameters of the approximation method (for example, the coefficients of linear regression) due to noisy observations. In linear regression, minimizing the uncertainty in the regression coefficients also minimizes the uncertainty in the prediction which makes these optimality criteria particularly relevant. However, simulation models are mostly deterministic, and hence these criteria are rarely used because of absence of any noise in the data. In case of kriging, there is no simple relation between the uncertainties in covariance parameters and prediction. Instead, two different criteria are available: the integrated mean square error (IMSE) and maximum mean square error (MMSE) [3,4], based on the prediction variance associated with the kriging method. It is to be noted that none of the above criteria used in the designs depend on the response values obtained at the sample points and all the samples of the design can be chosen offline before running the simulation model at any of them.

1.5.3 Adaptive Designs

Adaptive designs have the capability of generating samples, not only using information from the chosen approximation method (surrogate model) but also the output values at evaluated samples. Thus, the designs are built sequentially, by choosing new samples as a function of the other samples and their corresponding output values. New points are typically selected based on

some criterion; for example, sampling in the input space where the prediction variance is maximum, or where the prediction error is beyond a user specified threshold value [25,75–80]. Flexibility of these type of designs have led to their use in different engineering applications [81,82].

1.6 Objective and outline of thesis

In this dissertation, efficient black-box surrogate modeling algorithms have been developed for a range of different application problems in the field of solid mechanics. The simulation black-box models that are used to represent different physical systems in this study vary based on the type of output (discrete or continuous models), extent of model output information (knowledge of output or output gradients or both), and whether the model is stochastic or deterministic in nature. A *priori* qualitative knowledge about the nature of the output variation with respect to the input parameters is also available in some cases. All these variations in information from one problem to the other demand different surrogate modeling algorithms for maximum efficiency, involving different choices about sampling designs and surrogate methods. Even though the surrogate construction approaches implemented for each of the problems follow roughly the black-box surrogate modeling framework shown in figure 1.4, some components in the framework need to be modified to suit the problem at

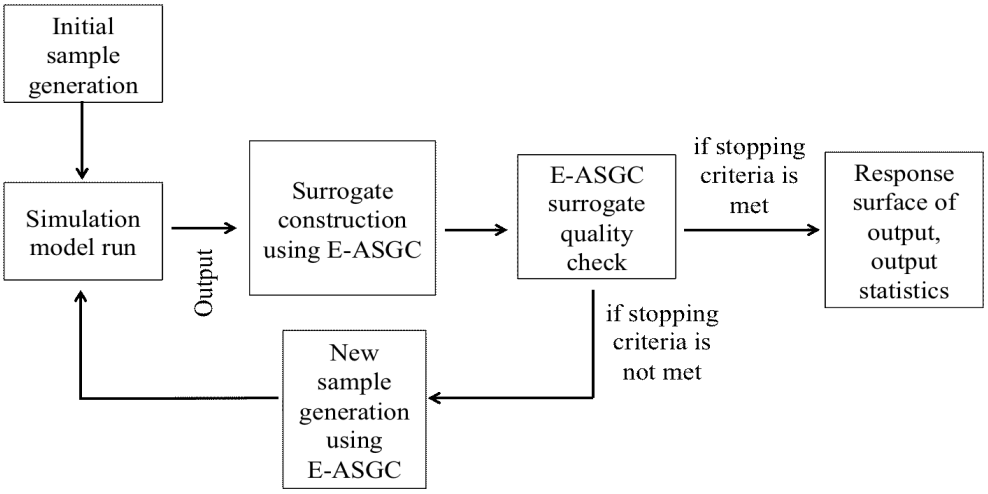


Figure 1.6: Framework of the E-ASGC algorithm in chapter 2

hand and eventually come up with the most efficient surrogate function.

Chapter 2 talks about a proposed surrogate modeling algorithm which is able to track discontinuities. It is an adaptive sparse grid collocation interpolation method based on piece-wise linear basis functions, which avoids unnecessary simulation model evaluations in smoother regions of the input parameter space. This is achieved by using a finite difference based one-dimensional derivative estimation technique along all the input parameters.

Chapter 3 deals with the development of an efficient surrogate modeling algorithm to deal with simulation models with large number of parameters. The algorithm is also designed to deal with discontinuous behavior of the simulation model output with respect to the parameters. It is a multi-element non-intrusive polynomial chaos expansion (PCE) approach which also includes a criterion for checking possible dimensionality reduction of the problem.

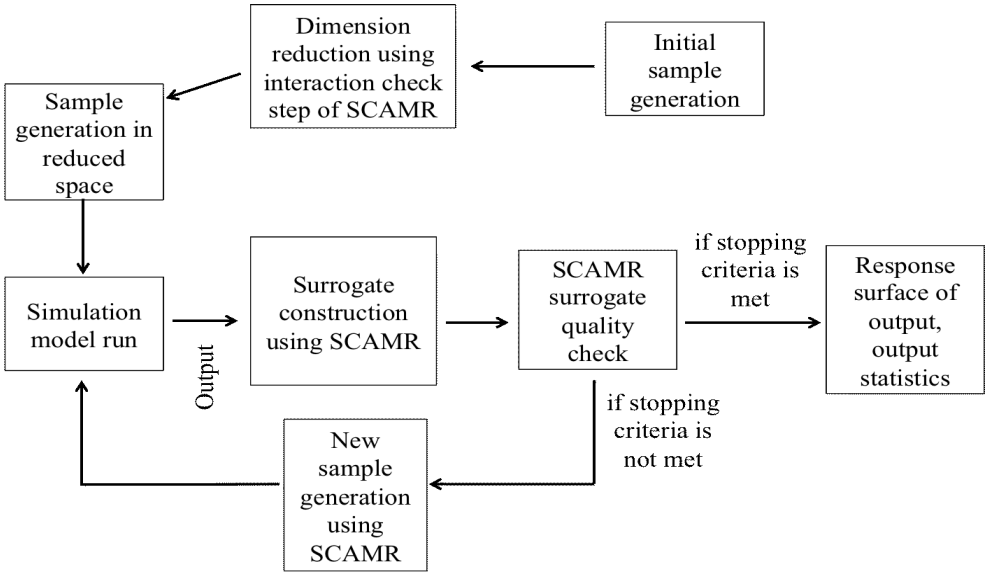


Figure 1.7: Framework of the SCAMR algorithm in chapter 3

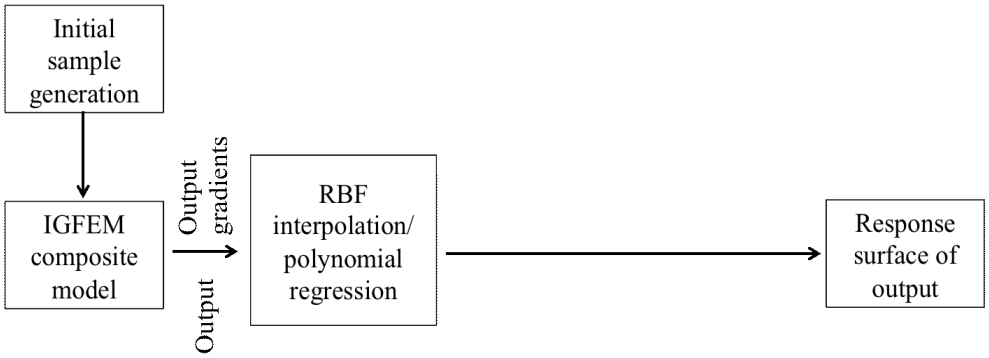


Figure 1.8: Surrogate algorithm framework in chapter 4

Chapter 4 considers a fiber-reinforced composite simulation model based on an Interface-Enriched Generalized Finite Element method (IGFEM). The simulation model can calculate, at any given combination of input parameter values, both the output and the output sensitivities with respect to each parameter. In this study, a number of surrogates are built for the simulation model using different choices of sequential space-filling designs and surrogate

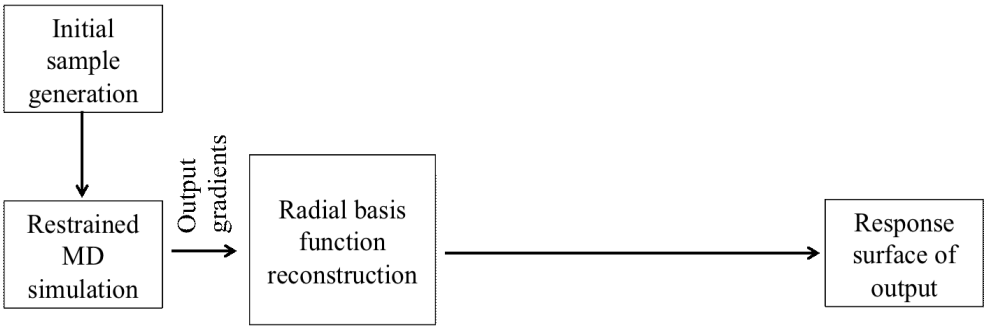


Figure 1.9: Surrogate algorithm framework in chapter 5

models, and their qualities are assessed. The goal here is to verify whether the additional sensitivity information helps in constructing a computationally cheaper but equally (or more) accurate surrogate than the one without any sensitivity information.

Chapter 5 talks about an efficient surrogate modeling methodology for free energy reconstruction. The simulation model considered here is a restrained molecular dynamics (MD) simulation model of solvated alanine dipeptide (AD) molecule. The goal is construct the free energy landscape of the molecule in a efficient. The restrained MD simulation, however, only calculates mean force which is the negative gradient of free energy. Thus, the challenge here is to built a surrogate function (landscape) for the output (free energy) with information of only the sensitivities (mean forces) from the simulation model.

Chapter 6 considers the simulation model of a continuum level plain weave S-2 glass/SC-15 epoxy composite plate under ballistic impact. This study involves construction of classification surrogates with the goal of generating two

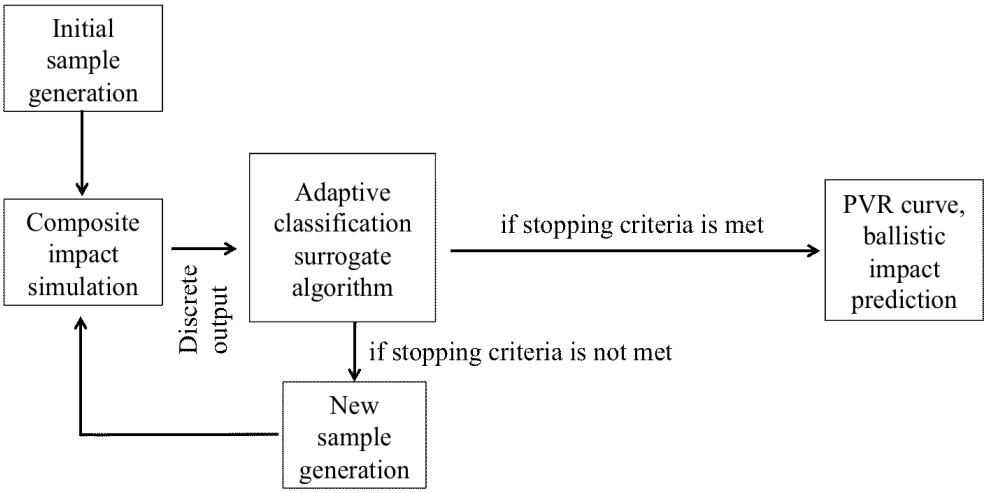


Figure 1.10: Surrogate algorithm framework in chapter 6

important quantities of interest, the probabilistic velocity response (PVR) curve as a function of the impact velocity, and the ballistic limit velocity prediction as a function of the model parameters.

Chapter 7 provides conclusions and possible future directions.

Chapter 2

An efficient adaptive sparse grid collocation method through derivative estimation

For uncertainty propagation of highly complex and/or nonlinear problems, one must resort to sample-based non-intrusive approaches [83]. In such cases, minimizing the number of function evaluations required to evaluate the response surface is of paramount importance. Sparse grid approaches have proven effective in reducing the number of sample evaluations. For example, the discrete projection collocation method has the notable feature of exhibiting fast convergence rates when approximating smooth functions; however, it lacks the ability to accurately and efficiently track response functions that exhibit fluctu-

ations, abrupt changes or discontinuities in very localized regions of the input domain. On the other hand, the piecewise linear collocation interpolation approach can track these localized variations in the response surface efficiently, but it converges slowly in the smooth regions. The proposed methodology, building on an existing work on adaptive hierarchical sparse grid collocation algorithm [84], is able to track localized behavior while also avoiding unnecessary function evaluations in smoother regions of the stochastic space by using a finite difference based one-dimensional derivative evaluation technique in all the dimensions. This derivative evaluation technique leads to faster convergence in the smoother regions than what is achieved in the existing collocation interpolation approaches. Illustrative examples show that this method is well suited to high-dimensional stochastic problems, and that stochastic elliptic problems with stochastic dimension as high as 100 can be dealt with effectively. This chapter is adapted from reference [85].

2.1 Introduction

Simulation models usually contain input parameters that are inherently random. The uncertainty in the inputs naturally leads to an uncertainty in the output. Thus a single solution for the system using a fixed set of input parameters is not sufficient to describe the system completely. Hence, given

the input uncertainties, it is important to understand how these uncertainties propagate through the deterministic system model and result in uncertainties in the output solution. The quantification of the output uncertainties is a much more comprehensive descriptor of the system under study. In this work, the objective is to construct an efficient surrogate of the output.

A polynomial chaos (PC) expansion is a popular method for surrogate modeling which represents the output of interest by the expansion of orthogonal polynomials (with respect to positive weight measure) in the stochastic input domain. It is based on the homogeneous chaos theory by Wiener [86] where a Gaussian random process was essentially expressed by a set of global Hermite polynomials. Ghanem and Spanos [8] have coupled this approach with finite element methods to effectively model uncertainties and this has been subsequently applied to various solid mechanics problems [87–91]. The generalized polynomial chaos (gPC) [9] method makes use of different types of orthogonal polynomials in the Askey scheme [92] as the bases to approximate random functions/processes, and has been applied to different problems [10,11,93]. It is capable of reaching fast convergence for smooth functions when the probability density function (PDF) of the random variables is identical to the weighting function of the orthogonal polynomials from the Askey scheme. This idea has been further extended to arbitrary random distributions [94,95]. In the presence of discontinuities or highly localized variations in the response

surface, this method may fail to converge due to the well-known Gibbs phenomenon. Remedies for this problem have been sought using multielement gPC [15, 96, 97], wavelet basis [12], piecewise polynomial basis [13], and basis enrichment of polynomial chaos expansions [98]. The gPC coefficients in the above works are determined by using the Stochastic Galerkin method [83] and performing Galerkin projection of the model equations onto the space of the orthogonal polynomials. This involves solving a coupled system of deterministic equations which may be non-trivial (difficult and time-consuming) to solve when the original simulation model is very complex in itself. This is the drawback of the intrusive nature of the methods.

By contrast, non-intrusive methods use the deterministic simulation model directly without requiring any modifications, which makes them more applicable to complex systems. Stochastic collocation approaches, for example, solves the simulation model at pre-selected points in the input parameter space, using either interpolation approaches or discrete projection approaches [6]. For example, Xiu [99] proposed a gPC scheme based on the stochastic collocation method, where the gPC coefficients are obtained using the discrete projection approach. Babuska et al. [100] used Gauss quadrature points to sample low-dimensional stochastic input domains and perform tensor product interpolation using 1-D basis functions. Tensor grid approaches suffer from the so-called 'curse of dimensionality' [101] as there is an exponential rise in the required

number of full model evaluations with the increase in dimensionality of the input domain. Sparse grid [63,102] approaches alleviate this problem to some extent as they significantly reduce the number of points in high dimensions while maintaining almost the same level of accuracy. Sparse grids are especially suitable for high dimensional problems involving numerical integration and interpolation. The interpolation approach approximates the stochastic space using multi-dimensional interpolation with the existing data such that the surrogate surface always passes through the pre-determined points. Sparse grid based interpolations [103,104] have been performed with the global Lagrange polynomial basis interpolants. However, these global approaches may not be suitable for tracking local steepness or discontinuities in the output, and the approximation may fail to converge to the true value.

Approaches [12, 100, 105] based on local bases have also been proposed to deal with non-smoothness in the output. Klimke and Wohlmuth [106–108] developed a sparse grid collocation interpolation scheme based on piecewise linear basis functions, which has the ability to resolve discontinuities in the output but suffers from slow convergence rates because of global refinement of the sparse grid. The approach is based on hierarchical sparse grid points where points are added in successive depth levels. The error indicator is known as the hierarchical surplus and acts as a stopping criterion for the algorithm. Ma and Zabaras [84] used a similar approach called adaptive sparse grid collocation

(ASGC) but also incorporated an adaptive strategy that enables a local sparse grid refinement around the discontinuity region, which helps enhance the convergence rate. The ASGC approach checks the hierarchical surplus values at each point in the current depth level and creates new points in the next depth level only in the neighborhood of points whose surplus error exceeds the tolerance value. The approach is restricted to uniform grid points because of the adaptivity criterion. For the purpose of tracking discontinuities, ASGC uses piecewise linear basis functions. This may lead to an unwanted slow convergence for the regions where the approximating response function is smooth. Other works that introduce adaptivity into the sparse grid collocation interpolation approach include dimension-adaptive sparse grid methods [109,110] and multi-element (domain-adaptive) sparse grid interpolation [111,112] methods. A global approach based on Padé-Legendre approximation [113] has also been used to track down strong non-linearities or discontinuities in the response surface.

The algorithm presented in this chapter is based on the work done by Ma and Zabaras [84] on adaptive sparse grid subset interpolation. Similar to that work, the proposed approach uses linear basis functions for the adaptive sparse grid interpolation to capture any localized variations in the response. In addition, it aims to reduce the number of function evaluations by local 1-dimensional cubic spline interpolations [114] in the smoother regions of the

response domain. The smoothness is measured by successive derivative estimation along a straight line of points using finite differences of the output values in any of the input dimensions. Small changes (within a tolerance) in the derivative estimates will indicate sufficient smoothness for cubic spline interpolation along the straight line. This helps to achieve the same accuracy as in [84], but decreases the number of function evaluations, especially when the response function is widely smooth. It is worth mentioning here that the derivative information is extracted approximately from the output values without any exact knowledge about the derivative of the output of interest.

The rest of the chapter is organized as follows: In section 2.2, the general mathematical model for any physical system with uncertainties is described. In section 2.3, the conventional stochastic collocation (CSC) method, the adaptive sparse grid collocation (ASGC) method and then the proposed efficient adaptive sparse grid collocation (E-ASGC) method are discussed in details. Section 2.4 deals with the various numerical examples to compare the performance of the proposed method with a few existing methods. Finally, the concluding remarks are given in section 2.5.

2.2 Problem Definition

Following notations in [84], we represent the complete probability space by the triplet $(\Omega, \mathcal{F}, \mathcal{P})$ where Ω corresponds to the sample space of outcomes, $\mathcal{F} \subset 2^{\Omega}$ is the sigma algebra of measurable events in Ω, and $\mathcal{P} : \mathcal{F} \rightarrow [0, 1]$ is the probability measure. Let $I(\omega) = \{I_1, I_2, I_3, .., I_d\}$ be the multidimensional vector of random input parameters in a problem of interest, where $I : \Omega \rightarrow \in \mathbb{R}^d$

$$Z(\omega) = f(I(\omega)), \quad \forall\, \omega \in \Omega \tag{2.1}$$

The goal is then to find out how the vector valued output $Z(\omega)$ varies with respect to each of the random vector components $I_i(\omega), i \in [1, 2,.., d]$. For a more general case where inputs are stochastic processes (in space or time), see Appendix A.

2.3 Stochastic Collocation Interpolation Method

2.3.1 Conventional Sparse Grid Interpolation

For a function $f : [a, b] \rightarrow \mathbb{R}$, the one-dimensional interpolation formula is given by:

$$U^k(f(x)) = \sum_{x^k \in X^k} a_{x^k}(x) f(x^k) = \sum_{j=1}^{m_k} a_{x_j^k}(x) f(x_j^k), \tag{2.2}$$

where $x \in [a, b]$, $X^k = \{x^k | x^k \in [a, b]\}$, $a_{x_j^k}(x) \in [0, 1] \subset \mathbb{R}^1$, $a_{x_j^k}(x_i^k) = \delta_{ij}$, $\{i, j\} \in [1, 2, ...m_k]$, and m_k = number of points in the set X^k. For multi-dimensional interpolation, the one-dimensional case can be upgraded to obtain a tensor product formulae:

$$(U^{k_1} \otimes ..... \otimes U^{k_d})(f(\mathbf{x})) = \sum_{j_1=1}^{m_1} .... \sum_{j_d=1}^{m_d} (a_{x_{j_1}^{k_1}}(\mathbf{x}) \otimes ..... \otimes a_{x_{j_d}^{k_d}}(\mathbf{x})) f(x_{j_1}^{k_1}, ..., x_{j_d}^{k_d}) \tag{2.3}$$

where d is the total number of dimensions and $\mathbf{x} = \{x_1, x_2, ..., x_d\} \in \mathbb{R}^d$

The major drawback of this tensor product formula is that the total number of points required are $(m_1)(m_2)(m_3).....(m_d)$ which rises exponentially with increase in dimensions, leading to the curse of dimensionality. The sparse grid approach that is used in the current work mitigates this issue to quite an extent by sampling significantly fewer points which are subsets of the tensor

grid structure. Though the accuracy of the algorithm is not totally dimension-independent, it gets weakened down to a logarithmic dependence.

Using similar definitions as in [106], we define $U^0 = 0$ and the incremental interpolant by:

$$\triangle^k(f(x)) = U^k(f(x)) - U^{k-1}(f(x)), \quad \forall k \geq 1 \tag{2.4}$$

where,

$$U^k(f(x)) = \sum_{x^k \in X^k} a_{x^k}(x) f(x^k) \tag{2.5}$$

and

$$U^{k-1}(f(x)) = U^k(U^{k-1}(f(x))) \tag{2.6}$$

By using the above three equations, we thus get,

$$\begin{aligned} \triangle^k(f(x)) &= \sum_{x^k \in X^k} a_{x^k}(x) f(x^k) - \sum_{x^k \in X^k} a_{x^k}(x) g(x^k) \\ &= \sum_{x^k \in X^k} a_{x^k}(x)(f(x^k) - g(x^k)) \end{aligned} \tag{2.7}$$

where $g = U^{k-1}(f(x))$

Now,

$$(f(x^k) - g(x^k)) = 0, \quad \forall x^k \in X^{k-1} \tag{2.8}$$

Thus,

$$\triangle^k(f(x)) = \sum_{x^k \in X_\triangle^{k-1}} a_{x^k}(x)(f(x^k) - g(x^k)) \tag{2.9}$$

where $X_\triangle^k = X^k \setminus X^{k-1}$ denotes the points in set X^k but not in X^{k-1}. Because of the nested property of uniform grids, $X^{k-1} \subset X^k$, the number of elements (points) in $X_\triangle^k = m_k - m_{k-1} = m_\triangle^k$

Rewriting Eq. (2.9), we get,

$$\triangle^k(f(x)) = \sum_{j=1}^{m_\triangle^k} a_{x_j^k}(x)(f(x_j^k) - g(x_j^k)) \tag{2.10}$$

Using the property $\triangle^k(f(x)) = U^k(f(x)) - U^{k-1}(f(x))$,we can write

$$U^k(f(x)) = \sum_{i=1}^{k} \triangle^i(f(x)) \tag{2.11}$$

In the case of a tensor grid, the multivariate interpolant expression is a tensor product extension of Eq. (2.11) and is given by,

$$(U^{k_1} \otimes U^{k_2} \otimes ..... \otimes U^{k_d})(f(\mathbf{x})) = \sum_{i_1=1}^{k_1} ..... \sum_{i_d=1}^{k_d} (\triangle^{i_1} \otimes ..... \otimes \triangle^{i_d})(f(\mathbf{x})) \tag{2.12}$$

On the other hand, Smolyak sparse grids [63] use a much smaller subset of the tensor grid. The sparse grid interpolant only considers points satisfying $|\mathbf{i}| \leq q$

and is defined by,

$$A_{q,d} = \sum_{|\mathbf{i}|\leq q} (\triangle^{i_1} \otimes ..... \otimes \triangle^{i_d})(f(\mathbf{x})) = A_{q-1,d}(f(\mathbf{x})) + \triangle A_{q,d}(f(\mathbf{x})) \tag{2.13}$$

where $|\mathbf{i}| = i_1 + i_2 + ...i_d$, for $\mathbf{i} = (i_1, i_2, ...i_d) \in \mathbb{N}^d$ and $q = k + d - 1$ for $k_1 = k_2 = \cdots = k_d = k \geq 1$.

$$\triangle A_{q,d}(f(\mathbf{x})) = \sum_{|\mathbf{i}|=q} (\triangle^{i_1} \otimes ..... \otimes \triangle^{i_d})(f(\mathbf{x})) \tag{2.14}$$

$$A_{q-1,d}(f(\mathbf{x})) = \sum_{|\mathbf{i}|\leq q-1} (\triangle^{i_1} \otimes ..... \otimes \triangle^{i_d})(f(\mathbf{x})) \tag{2.15}$$

Here i_s $\{\forall s = 1, 2, ..., d\}$ is called the depth of interpolation in the s-th dimension whereas q denotes the global depth of interpolation and $A_{d-1,d} = 0$. Thus, putting $q = d$, we get $A_{d,d}(f) = \triangle A_{d,d}(f(\mathbf{x}))$ and for $q = d + 1$, $A_{d+1,d}(f(\mathbf{x})) = A_{d,d}(f(\mathbf{x})) + \triangle A_{d+1,d}(f(\mathbf{x}))$ and so on. Thus it is seen that there is a hierarchy when it comes to the interpolant at different levels, and the interpolant at a given level contributes to the estimation of the next higher level interpolant.

From Eqs. (2.10) and (2.14), we get,

$$\begin{aligned}\triangle A_{q,d}(f(\mathbf{x})) &= \sum_{|\mathbf{i}|=q}(\triangle^{i_1} \otimes ..... \otimes \triangle^{i_d})(f(\mathbf{x})) \\ &= \sum_{|\mathbf{i}|=q}\sum_{\mathbf{j}}(a_{j_1}^{i_1}(\mathbf{x}) \otimes ..... \otimes a_{j_d}^{i_d}(\mathbf{x}))(f(x_{j_1}^{i_1}, ..., x_{j_d}^{i_d}) \\ &\quad - g_1(x_{j_1}^{i_1}, ..., x_{j_d}^{i_d}))\end{aligned}$$

where $\mathbf{j} = \{(j_1, j_2, ..., j_d) : j_s = 1, ..., m_{\triangle}^{i_s};\ s = 1, ..., d\}$, and g_1 is defined as

$$\begin{aligned}g_1 &= (U^{k_1-1} \otimes ..... \otimes U^{k_d-1})(f(\mathbf{x})) \\ &= \sum_{|\mathbf{i}|\leq q-1}(\triangle^{i_1} \otimes ..... \otimes \triangle^{i_d})(f(\mathbf{x})) \\ &= A_{q-1,d}(f(\mathbf{x}))\end{aligned}$$

Now,

$$\begin{aligned}\triangle A_{q,d}(f(\mathbf{x})) &= \sum_{|\mathbf{i}|=q}\sum_{\mathbf{j}}(a_{j_1}^{i_1}(\mathbf{x}) \otimes ..... \otimes a_{j_d}^{i_d}(\mathbf{x}))(f(x_{j_1}^{i_1}, ..., x_{j_d}^{i_d}) \\ &\quad - g_1(x_{j_1}^{i_1}, ..., x_{j_d}^{i_d})) \\ &= \sum_{|\mathbf{i}|=q}\sum_{\mathbf{j}} a_{\mathbf{j}}^{\mathbf{i}}(\mathbf{x}) w_{\mathbf{j}}^{\mathbf{i}}(\mathbf{y}) \qquad (2.16)\end{aligned}$$

where $a_{\mathbf{j}}^{\mathbf{i}}$ is the d-dimensional basis function, $w_{\mathbf{j}}^{\mathbf{i}}$ is the hierarchical surplus and $\mathbf{y} = \{x_{j_1}^{i_1}, ..., x_{j_d}^{i_d}\}$.

Thus, once the surrogate model has been identified for any given level q, the function value at any point can be calculated as:

$$u(\mathbf{x},\mathbf{y}) = \sum_{|\mathbf{i}|\leq q} \sum_{\mathbf{j}} a_{\mathbf{j}}^{\mathbf{i}}(\mathbf{x}) w_{\mathbf{j}}^{\mathbf{i}}(\mathbf{y}) \tag{2.17}$$

The mean of the solution can be analytically estimated [84] as:

$$\mathbb{E}[u(\mathbf{x},\mathbf{y})] = \bar{u}(\mathbf{y}) = \int_{\Gamma} \sum_{|\mathbf{i}|\leq q} \sum_{\mathbf{j}} w_{\mathbf{j}}^{\mathbf{i}}(\mathbf{y}) a_{\mathbf{j}}^{\mathbf{i}}(\mathbf{x}) \rho(\mathbf{x}) d\mathbf{x} \tag{2.18}$$

Since $\mathbf{x}$ is an uniform random space, the probability density function $\rho(\mathbf{x}) = 1$ for the domain $\Gamma = [0,1]^d$. Substituting this value of $\rho(\mathbf{x})$, rearranging the integral and assuming the random variables are independent of each other,

$$\bar{u}(\mathbf{y}) = \sum_{|\mathbf{i}|\leq q} \sum_{\mathbf{j}} w_{\mathbf{j}}^{\mathbf{i}}(\mathbf{y}) l_{\mathbf{j}}^{\mathbf{i}} \tag{2.19}$$

where

$$l_{\mathbf{j}}^{\mathbf{i}} = \int_{\Gamma} a_{\mathbf{j}}^{\mathbf{i}}(\mathbf{x}) d\mathbf{x} = \prod_{k=1}^{d} \int_0^1 a_{j_k}^{i_k}(x) dx \tag{2.20}$$

and

$$\int_0^1 a_{j_k}^{i_k}(x) dx = \begin{cases} 1, & \text{if } i_k = 1 \\ \frac{1}{4}, & \text{if } i_k = 2 \\ 2^{1-i_k}, & \text{otherwise} \end{cases}$$

To get the variance, we proceed with the square of the function value in Eq. (2.17) which can be approximated as:

$$u^2(\mathbf{x},\mathbf{y}) = \sum_{|\mathbf{i}|\leq q}\sum_{\mathbf{j}} a_{\mathbf{j}}^{\mathbf{i}}(\mathbf{x}) v_{\mathbf{j}}^{\mathbf{i}}(\mathbf{y}) \tag{2.21}$$

where $v_{\mathbf{j}}^{\mathbf{i}}$ is the hierarchical surplus corresponding to the square of the output. Then the expectation of the square of the random solution can be estimated as:

$$\mathbb{E}[u^2(\mathbf{x},\mathbf{y})] = \bar{u^2}(\mathbf{y}) = \sum_{|\mathbf{i}|\leq q}\sum_{\mathbf{j}} v_{\mathbf{j}}^{\mathbf{i}}(\mathbf{y}) \int_{\Gamma} a_{\mathbf{j}}^{\mathbf{i}}(\mathbf{x}) d\mathbf{x} \tag{2.22}$$

Thus the variance of the solution is given by:

$$Var[u(\mathbf{x},\mathbf{y})] = \sigma_u^2(\mathbf{y}) = \bar{u^2}(\mathbf{y}) - \bar{u}(\mathbf{y})^2 \tag{2.23}$$

2.3.2 Adaptive Sparse Grid Interpolation

In conventional sparse grid interpolation methods, the error check is such that if any hierarchical surplus at the current level of interpolation exceeds the tolerance, all points in the next higher level must be evaluated. The algorithm ignores the fact that there may be smooth regions which do not require subsequent refinements. In adaptive sparse grid interpolation, unnecessary higher level samples are avoided by performing selective refinements. This method

makes use of the tree-like data structure of 1-D equidistant sparse grid points. A schematic of the adaptive procedure using the tree-like structure of the grid points is given in figure 2.1. With the exception of the point addition at level 2, two points are added in the neighborhood of each point at the previous level. In [84], a point at the current level has been referred to as a 'father' while points added around it at the next level are referred to as 'sons'. For a d-dimensional random domain, there will be $2d$ sons added for each father if it is not a level-1 point in any of the dimensions. Therefore, if there are m points at the current level at which the hierarchical surplus exceeds the tolerance, then at most $2dm$ points are added at the next level. There may be duplication of next-level points, requiring that a check be performed to avoid redundant sampling. This approach leads to slow convergence in the presence of localized variations or discontinuities, requiring computation up to a very high interpolation level. On the other hand, regardless of the nature of the response surface, the mean and variance converge quickly to a desired level of accuracy because of the sharp drop in the integral weights with rise in interpolation level. So it is reasonable to limit the algorithm to a maximum interpolation level which acts as another termination criterion.

An example of the adaptive procedure in 1-D is given in figure 2.2. A 1-D C^1 discontinuous function shown in figure 2.2a is used to demonstrate the adaptivity of the method. Figure 2.2b shows the conventional approach of adding

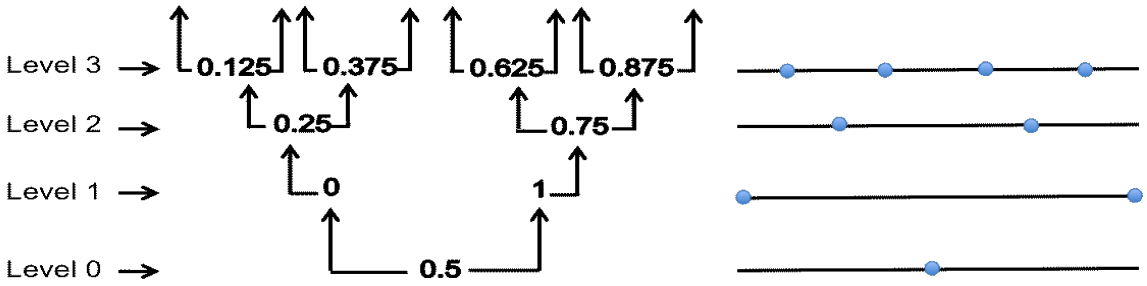
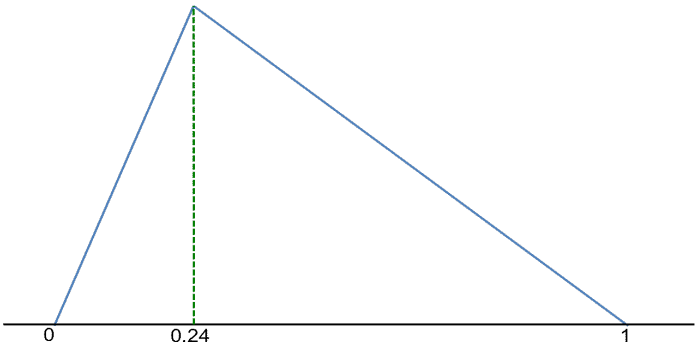


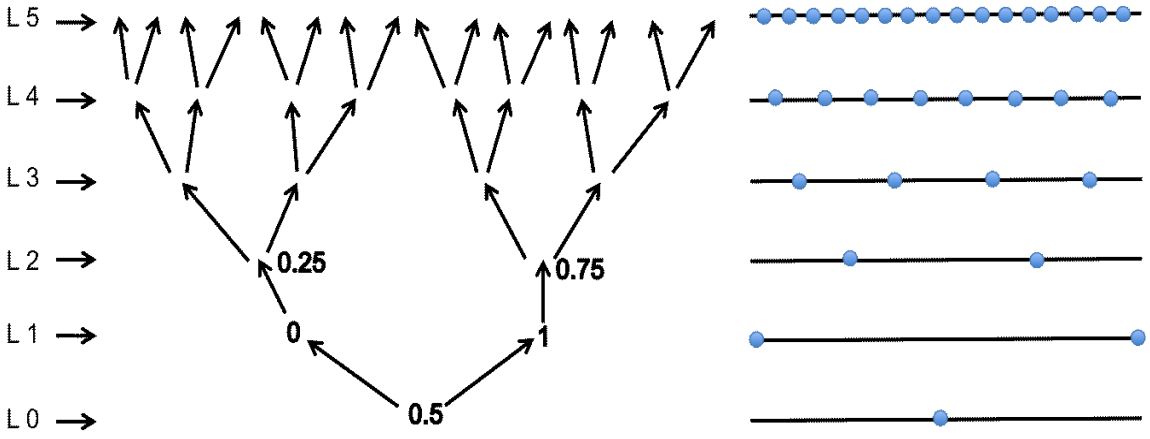
Figure 2.1: Tree-like structure of 1-D grid points

the sparse grid points up to level 5. The absence of any adaptivity leads to a total of 33 point evaluations. In contrast, the adaptive collocation strategy shown in figure 2.2c allows for local refinement of points around the C^1 discontinuous region and results in only 17 function evaluations up to level 5. For example, at level 3, out of the four points, only two of the points at $x = 0.125$ and $x = 0.375$ are 'fathers' to points in the next higher level. This implies that the hierarchical surpluses at those two points are above the tolerance, while the hierarchical surpluses at $x = 0.625$ and $x = 0.875$ are below the tolerance.

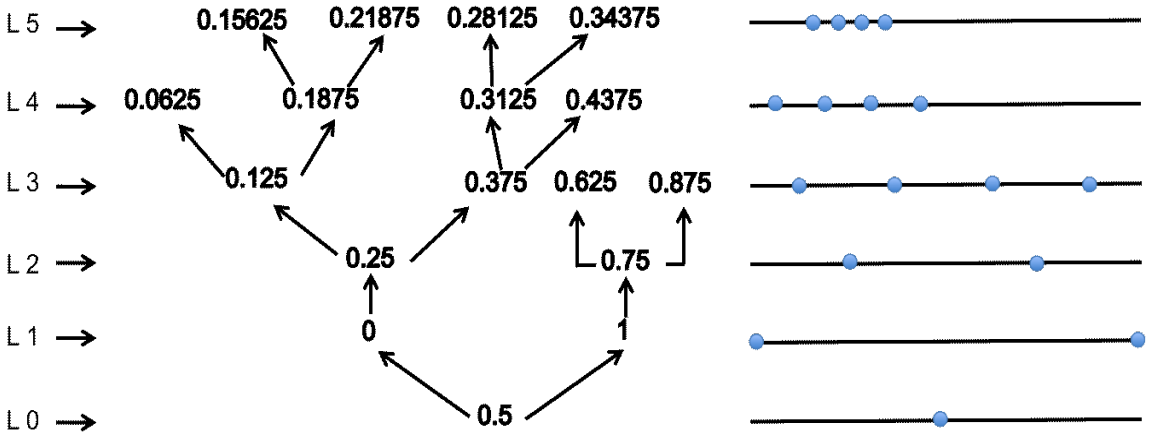
In a high dimensional case with highly localized variations along some dimensions, there may be a significant number of other dimensions along which the response function is smooth without any sharp variations. Thus using a piece-wise linear function leads to slow convergence of the surpluses in the smooth regions. A significant number of full model evaluations can be avoided by handling these smoother regions efficiently. The proposed efficient collocation method is based on this very idea which will be discussed next.



(a) A 1-D function with C^1 discontinuity



(b) Conventional 1-D sparse grid points



(c) Adaptivity on 1-D sparse grid points

Figure 2.2: Comparison between the addition of the conventional and the adaptive sparse grid points in 1-D

2.3.3 Efficient adaptive sparse grid collocation through derivative estimation

One aspect to improve in the adaptive sparse grid subset collocation algorithm [84] is that in allocating more points along discontinuities and important dimensions, there can be a significant number of points also added to smoother regions in the domain. These points that are assigned to the smoother regions unnecessarily increase the computational cost. Thus one way to improve efficiency is to avoid brute force evaluations in the smoother regions as much as possible. The proposed method aims at achieving this by approximating the smoother regions with cubic splines [114]. Therefore when proceeding with the adaptive algorithm [84], if a sparse grid node is generated within any 1-D approximated smooth region, the function value at that point can be approximated by cubic spline interpolation. Cubic splines, being third order polynomials, can achieve sufficiently fast convergence and are robust because of their piece-wise nature. Higher order polynomials were not used to avoid overfitting.

The mechanism works as follows: Let P be the total number of unique sparse grid points and D_i $[i \in 1, 2, ...d]$ be a certain dimension in the d-dimensional stochastic space. A projection is now taken on the plane orthogonal to the D_i dimension to form a D_{d-1} dimensional non-unique data point set. This set is

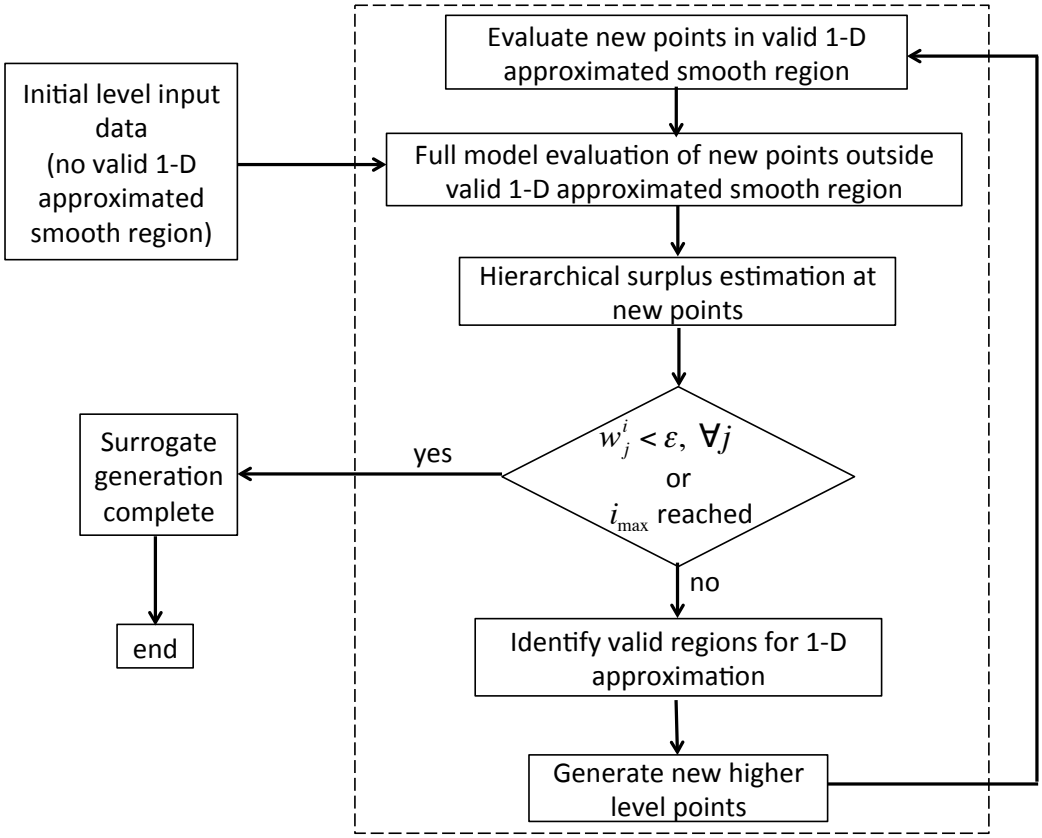


Figure 2.3: Flow chart for the efficient adaptive sparse grid collocation algorithm

non-unique because of the overlap of multiple points due to the projection. The critical control parameter is the minimum number of points in a single straight line along the D_i dimension needed to approximate the linear region. The non-unique set is used to evaluate the frequency of points along every straight line in the D_i dimension. Once the criterion is met, successive finite difference based derivative calculation is done to crudely detect any discontinuity along the line. If the change in the derivative varies gradually enough throughout, the data along that path is stored. Otherwise the line can be split into 1-D sub-intervals and data in those individual sub-intervals are stored. The data storage include the input points, the function value at the points, the midpoint of the linear interval and a parameter L quantifying the extent of the approximation along a straight line. L is defined as half of the total length of any 1-D approximated smooth region. In future calculations, if we come across a point lying in this 1-D region where the function value needs to be known, then instead of performing an expensive brute force evaluation there, we retrieve the associated data and simply approximate the function value by the cubic spline interpolant.

The algorithm is outlined below:

Initialization

(1) Set the maximum interpolation level i_{max}

(2) Set the tolerance parameter ϵ

(3) Set the dimension d of the problem

(4) Initialize the level to $i = 1$

Conventional sparse grid iterations for initial few levels i_1

While $i \leq i_1$,

(5) Perform full model evaluation at level i points

(6) Calculate also the hierarchical surpluses w_j^i at each of the points

(7) Set $i = i + 1$

(8) Form $2d$ points at $(i + 1)$-th level

(9) Check for duplication of previous points

(10) Go to step (5)

Main adaptive loop based on derivative estimation

While $i_1 < i \leq i_{max}$,

(11) Check if any of the points lie in the regions identified as smooth enough for 1-D interpolation by searching the stored database.

(12) For each point lying in any of these regions, perform a cheap interpolation to get the function value.

(13) For the remaining points, perform full model evaluations.

(14) Calculate the hierarchical surpluses w_j^i at each of the points in steps (12) and (13)

(15) Check if $|w_j^i| \geq \epsilon$ for each point. If no points satisfy this criterion, go to step (22).

(16) For each point satisfying criterion in step (15), form $2d$ points at the $(i+1)$-th level

(17) Set $i = i + 1$

(18) Check for duplicity of points

(19) For each dimension, project all points on the hyperplane normal to the dimension, to get M_p number of $(d-1)$-dimensional non-unique projected points. Count the number of unique points N and the number of co-located points at each of the unique points m_i, such that $\sum_i^N m_i = M_p$, where $m_i \geq 1$

(20) For each point $\in M_p$ with m_i greater than a minimum number M_{min},

a) Calculate successive derivative along that straight line using finite differences of the output values.

b) If the change in the gradients are below a tolerance ϕ, the interval is considered smooth, and it is stored in the database for future retrieval. The database for an interval consists of the input data, the output data, the mid-point value of the interval, and half the length of the interval, represented by the quadruplet $(I, O, \bar{I}, L)$.

(21) Go to step (11)

(22) End generation of surrogate model.

The generated surrogate model can be used as the basis for generating moments (e.g., mean and variance) and/or distribution of the response. Also, the output response at any arbitrary query point can be extracted. The flow chart

for the entire algorithm is shown in figure 2.3.

Remark: It is very important to note here that poor choices of parameters M_{min} and ϕ may render the proposed method very inefficient in certain situations. If M_{min} is not large enough, then the finite difference derivatives could be inaccurate, which could lead to inaccurate use of the cubic spline interpolation, particularly if the tolerance ϕ is too high. If the cubic spline interpolated value at a sparse grid point is erroneous as a result of these poorly chosen parameters, then one of two undesirable scenarios could occur. First, the hierarchical surplus error at that point could appear to be larger than the tolerance when in reality it is not, which would direct the algorithm to add new and unnecessary sampling points around that point. Second, the hierarchical surplus error at the point could appear to be smaller than the tolerance when in reality it is not. This would direct the algorithm to add sampling points further away from this point, potentially missing local variations in the response function near this point and reducing the accuracy of the results. If these scenarios are encountered quite often in a particular surrogate modeling procedure, then the performance of the proposed method will be worse than the ASGC method both in terms of the error and the number of function evaluations at a particular sparse grid level.

2.3.4 Convergence and accuracy of the proposed efficient adaptive method

In the proposed adaptive method based on 1-D derivative estimation, the accuracy depends heavily on the minimum number of points M_{min} chosen for the storage and retrieval of points for cubic spline interpolation.

$$\|u^q_{asgc} - u^q_{e-asgc}\|_\infty \leq \epsilon_1 N_a \tag{2.24}$$

where $\epsilon_1 \leq \frac{5}{384}||u^{q(4)}_{asgc}||_\infty h^4$ is the interpolation error [115] which decreases with increase in M_{min}, h is the maximum knot spacing and N_a are the points where full model evaluations are performed in the ASGC [84] sparse grid but are interpolated in the E-ASGC sparse grid. In the interpolation error expression, $(.)^{(4)}$ denotes the fourth order derivative.

Now, the interpolation error [84] between the adaptive sparse grid and the conventional sparse grid is given by:

$$\|u^q_{csc} - u^q_{asgc}\|_\infty \leq \epsilon_2 N_b \tag{2.25}$$

where N_b are the points in the conventional sparse grid but missing in the adaptive sparse grid subset due to the hierarchical surplus based adaptivity.

The interpolation error of the conventional sparse grid [106] is given by:

$$\|u - u^q_{csc}\|_\infty = O(N_t^{-2}|log_2 N_t|^{3(d-1)}) \tag{2.26}$$

where N_t= total number of interpolation points at interpolation depth q in d-dimensional stochastic space.

Thus the approximate bound of the total error using the proposed method is:

$$\begin{aligned}\|u - u^q_{e-asgc}\|_\infty &= \|u - u^q_{csc} + u^q_{csc} - u^q_{asgc} + u^q_{asgc} - u^q_{e-asgc}\|_\infty \\ &\leq \|u - u^q_{csc}\|_\infty + \|u^q_{csc} - u^q_{asgc}\|_\infty + \|u^q_{asgc} - u^q_{e-asgc}\|_\infty \end{aligned} \tag{2.27}$$

It is worth mentioning that M_{min} should be chosen such that $\epsilon_1 < \epsilon_2$.

2.4 Numerical Examples

In this section, results are shown for explicit functions as well as implicit functional variations in different dimensions. All examples compare the adaptive sparse grid interpolation [84] and the proposed efficient adaptive sparse grid collocation method (E-ASGC). The first two examples studies 2-dimensional analytic functions to assess the effectiveness of the proposed method. The third example considers a family of 5-dimensional analytic functions. In the next example, a 1-dimensional Kraichnan-Orszag problem is considered to demon-

strate the performance of the E-ASGC method. After that, a spatial one-dimensional elliptic problem with high-dimensional stochasticity is used to compare the performance of the different methods. The next example deals with an indeterminate truss structure with variable cross-sectional areas of its members. In the last example, a 2-D composite unit cell model with interface damage is considered.

2.4.1 Function with C^1 discontinuity

We consider the function in $[0, 1]^2$ as mentioned in [84]

$$f(x, y) = \frac{1}{|0.3 - x^2 - y^2| + 0.1} \tag{2.28}$$

The exact function is plotted in figure 2.4a and it is seen that there is a line discontinuity which is not along any of the two dimensions. It is also observed that away from the discontinuities, the function is quite smooth which is suited for higher order interpolation. The proposed derivative based approach aims to utilize this feature of the function. The approximate function obtained from the E-ASGC algorithm at the interpolation depth 19 is shown in figure 2.4b. The sampling points for the existing ASGC algorithm are shown in figure 2.4c while those of the E-ASGC method are shown in figure 2.4d. It can seen from the plots that the E-ASGC approach effectively approximates the smooth re-

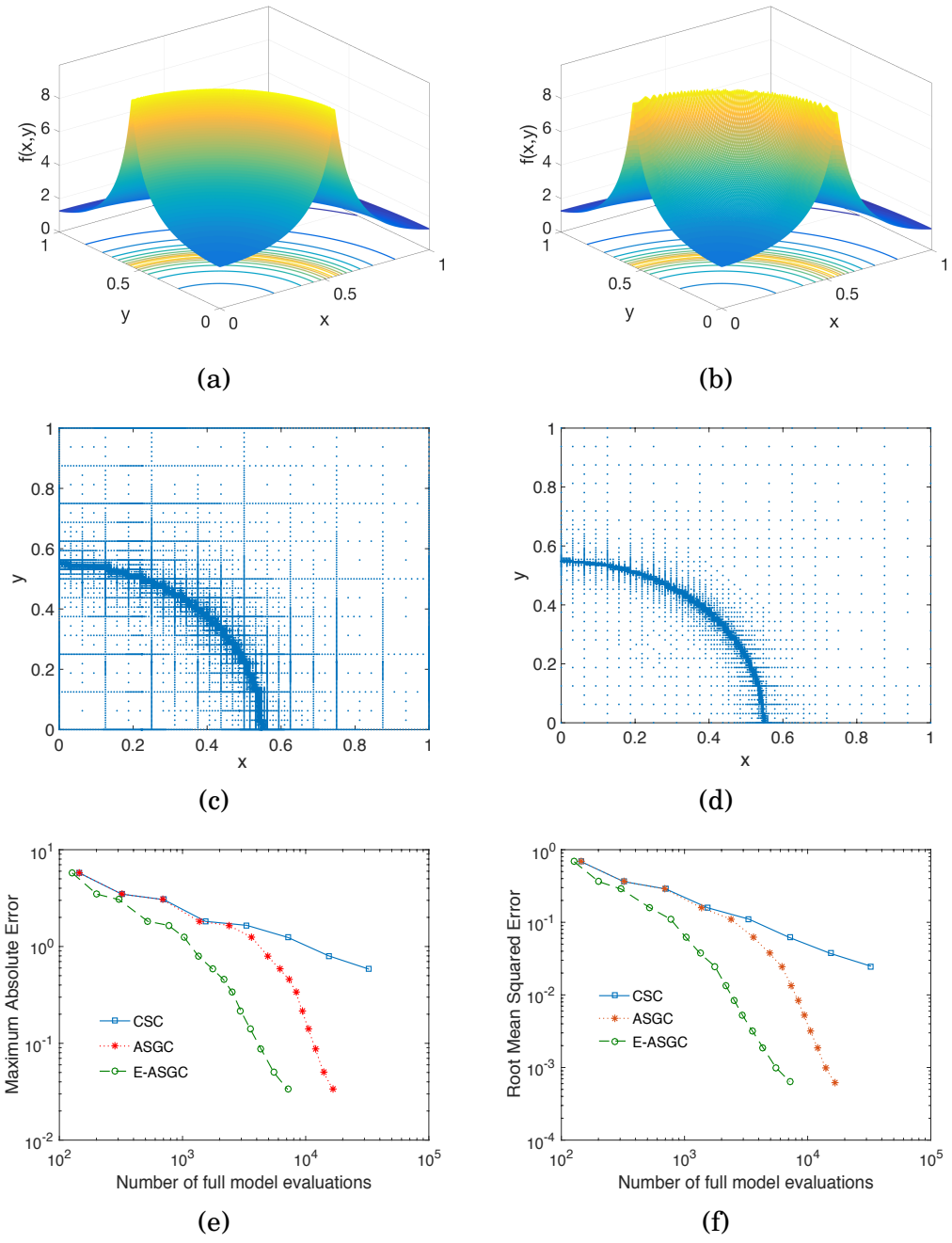


Figure 2.4: Performance comparison of 2-D ine singularity function f(x,y): (a) Exact 2-D line singularity function; (b) Approximate function from E-ASGC; (c) ASGC sampling points (16,659); (d) E-ASGC sampling points (7,149); (e) Maximum Absolute error convergence plot comparing conventional stochastic collocation (CSC), adaptive sparse grid subset collocation (ASGC) and efficient adaptive sparse grid collocation (E-ASGC); (f) Root Mean Squared Error convergence plot

gions of the input domain, with significantly fewer full model evaluations in those regions. Figures 2.4e and 2.4f shows the error convergence plots for the conventional sparse grid collocation (CSC), ASGC and E-ASGC methods. It is seen that the E-ASGC method clearly outperforms both the ASGC and the CSC methods. As a comparison, for a maximum absolute error of 0.0334, the E-ASGC method requires 7,149 function evaluations while the ASGC method requires 16,659 function evaluations. Thus the E-ASGC method reduces sampling by more than a factor of 2 relative to the ASGC method. The CSC method has the worst performance with a total of 32,769 function evaluations required for a maximum absolute error of 0.5824.

2.4.2 Function with C^0 discontinuity

In this section, a 2-dimensional C^0-discontinuous function is considered to study the performance of the E-ASGC method with respect to the ASGC and the CSC approach. We consider the function in $[0,1]^2$ as mentioned in [112]

$$f(x,y) = \begin{cases} 0, & \text{if } x \geq 0.5 \text{ or } y \geq 0.5, \\ \sin(\pi x)\sin(\pi y), & \text{otherwise} \end{cases} \tag{2.29}$$

The exact function is plotted in figure 2.5a and it is seen that there is a jump discontinuity in both dimensions. It is also observed that away from the discon-

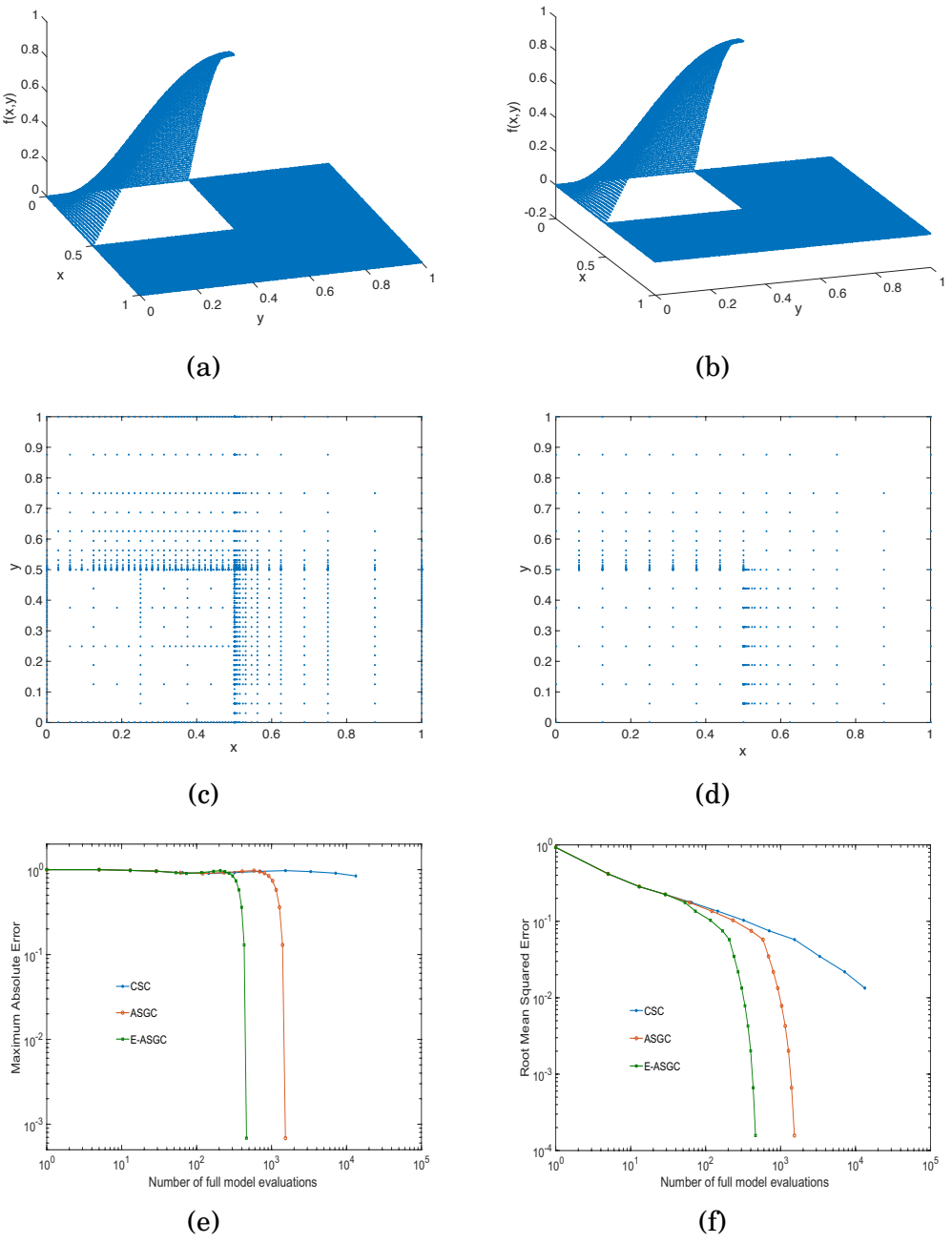


Figure 2.5: Performance comparison of 2-D discontinuous function f(x,y): (a) Exact 2-D discontinuous function; (b) Approximate function from E-ASGC; (c) ASGC sampling points (1,531); (d) E-ASGC sampling points (463); (e) Maximum Absolute error convergence plot comparing conventional stochastic collocation (CSC), adaptive sparse grid subset collocation (ASGC) and efficient adaptive sparse grid collocation (E-ASGC); (f) Root Mean Squared Error convergence plot

tinuities, the function is smooth and in fact has a constant value of 0 in most parts of the domain. The gradient based approach aims to capitalize on these smooth features of the function.

The approximate function obtained from the E-ASGC algorithm at the interpolation depth 16 is shown in figure 2.5b. The input domain for the existing ASGC algorithm and the E-ASGC method are shown in figures 2.5c and 2.5d respectively. From the input domain plots, it is observed that the total number of full model evaluations is effectively reduced by using the E-ASGC method compared to the ASGC method. This is clear from figures 2.5e and 2.5f which shows the maximum absolute error and root mean squared error convergence plots for the conventional sparse grid collocation (CSC), ASGC and E-ASGC methods. The effectiveness of the E-ASGC method relative to the ASGC and the CSC methods is evident. For an interpolation depth of 16, the maximum absolute error from the E-ASGC method is $6.8499\text{x}10^{-4}$ with 463 function evaluations while the ASGC method requires 1,531 function evaluations for the same level of accuracy. Thus the E-ASGC method reduces sampling by more than a factor of 4 relative to the ASGC method. On the other hand, the CSC method does not seem to converge at all with 13,317 function evaluations for a maximum absolute error of 0.8432.

2.4.3 5 dimensional functions

The 5-dimensional family of functions are taken from Genz [116]. They were primarily used to assess the efficiency of numerical integration algorithms. The functions defined on $x \in [0,1]^5$ are described [106] as follows:

Oscillatory function:

$$f_1(\mathbf{x}) = \cos(2\pi w_1 + \sum_{i=1}^{5} c_i x_i) \tag{2.30}$$

where w_1 and $c_i : i = 1, 2, ...5$ are constants.

Corner Peak Function:

$$f_2(\mathbf{x}) = (1 + \sum_{i=1}^{5} c_i x_i)^{-6} \tag{2.31}$$

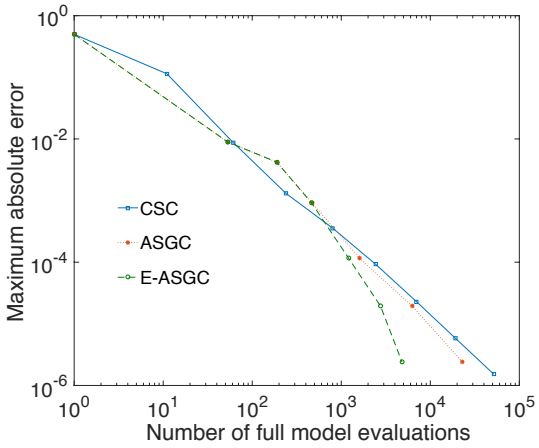
where $c_i : i = 1, 2, ...5$ are constants.

Discontinuous function:

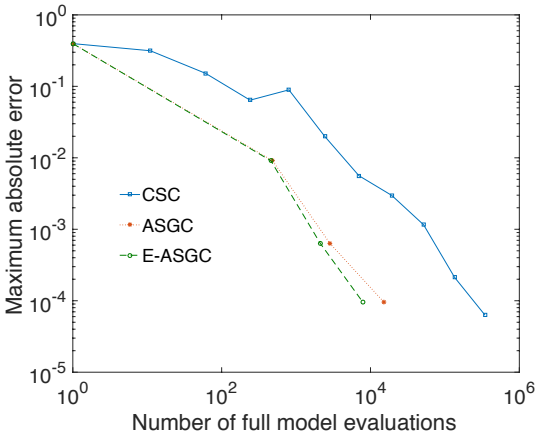
$$f_3(\mathbf{x}) = \begin{cases} 0, & \text{if } x_1 \geq w_1 \text{ or } x_2 \geq w_2, \\ \exp(\sum_{i=1}^{5} c_i x_i), & \text{otherwise} \end{cases}$$

where w_1, w_2 and $c_i : i = 1, 2, ...5$ are constants.

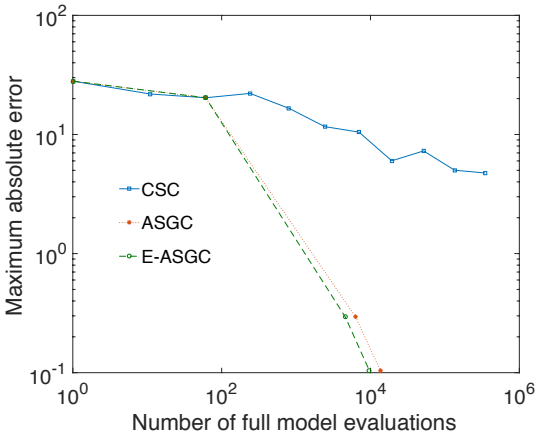
Convergence plots of the maximum absolute error for the E-ASGC method is compared with that of the conventional sparse grid method and the ASGC method for all the three functions, shown in figure 2.6. These results show that the E-ASGC and ASGC are at least as efficient as the CSC method in all the cases and they perform significantly better for the corner peak and discontinu-



(a) Oscillatory function



(b) Corner Peak function



(c) Discontinuous function

Figure 2.6: Error estimation for different 5 dimensional analytic functions

ous functions. E-ASGC reduces the required samples from ASGC somewhat in all the three cases.

2.4.4 Kraichnan-Orszag problem

The Kraichnan-Orszag three mode problem [117, 118] is related to the turbulence modeling of three interacting simple shear waves. It is defined by the following coupled system of nonlinear ODEs:

$$\begin{aligned} \frac{dx_1}{dt} &= x_2 x_3 \\ \frac{dx_2}{dt} &= x_1 x_3 \\ \frac{dx_3}{dt} &= -2 x_1 x_3 \end{aligned} \tag{2.32}$$

subjected to initial conditions:

$$x_1(0) = X_1(0; w), \; x_2(0) = X_2(0; w) \text{ and } x_3(0) = X_3(0; w)$$

For fixed initial conditions, the deterministic solutions are generally periodic. However, if the initial conditions are on and around the planes $x_1 = x_2$ and $x_1 = -x_2$, the period goes to infinity giving rise to discontinuity in the response surfaces. Thus the variations in the solutions are very sharp around these planes and need to be tracked down.

To simplify things, the following transformation is performed [15]:

$$\mathbf{y} = T\mathbf{x} \tag{2.33}$$

where

$$T = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

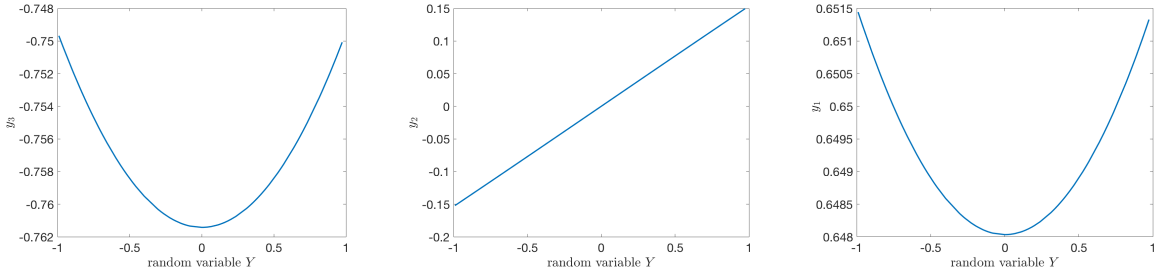
Thus the deterministic solutions rotate by $\frac{\pi}{4}$ around the x_3 axis in the $x_1 - x_2$ plane. The result of this transformation is that the planes defined by $x_1 = x_2$ and $x_1 = -x_2$ are now the $y_1 = 0$ and $y_2 = 0$ planes respectively. This leads to the new system formulation:

$$\begin{aligned} \frac{dy_1}{dt} &= y_1 y_3 \\ \frac{dy_2}{dt} &= -y_2 y_3 \\ \frac{dy_3}{dt} &= -y_1^2 + y_2^2 \end{aligned} \tag{2.34}$$

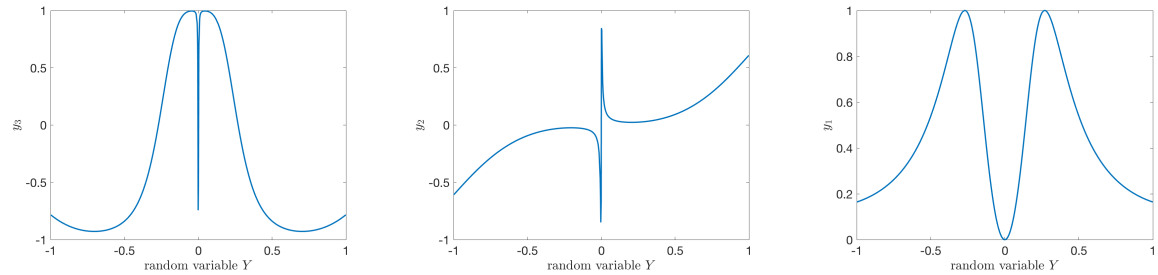
subjected to initial conditions

$$y_1(0) = Y_1(0; w),\ y_2(0) = Y_2(0; w) \text{ and } y_3(0) = Y_3(0; w)$$

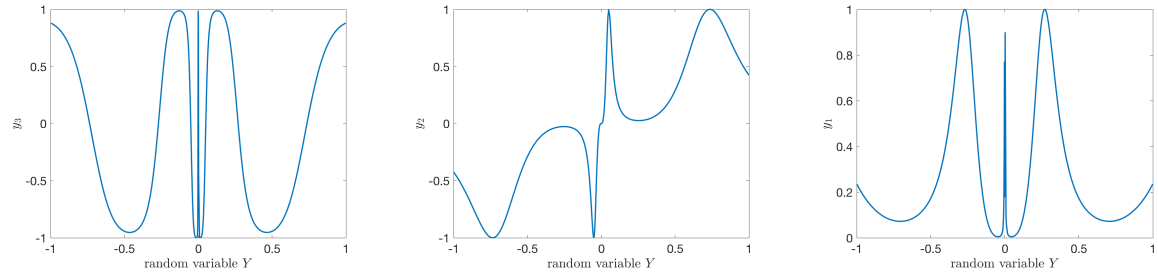
The goal is to assess the variation of the solutions y_1, y_2 and y_3 with different initial conditions and form the corresponding response surfaces. After transformation, discontinuity now occurs around the planes $y_1 = 0$ and $y_2 = 0$. The



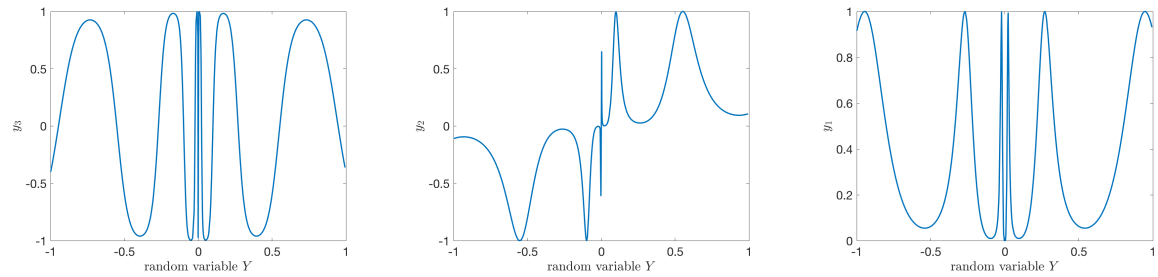
(a) time t=1



(b) time t=10



(c) time t=20



(d) time t=30

Figure 2.7: Variation of y_1, y_2 and y_3 with random variable Y at different times

initial conditions are chosen to be uniformly random in the range $[-1, 1]$. The coupled system of differential equations with a fixed set of initial conditions was solved using the Matlab in-built differential equation solver 'ode15s' using a time step of 0.01.

Table 2.1: Comparison of error values for the 1D K-O problem

Error tolerance	ASGC Samples	E-ASGC Samples	Maximum error of variance
10^{-2}	425	393	2.34×10^{-4}
10^{-3}	1381	1066	2.08×10^{-5}

2.4.4.1 One-dimensional randomness

For this case, the initial conditions are given as:

$$y_1(0) = 1.0,\ y_2(0) = 0.1Y(0; w) \text{ and } y_3(0) = 0$$

The development of variance of y_1, y_2 and y_3 with time is the output we are interested in. The time interval chosen was of the range $[0, 30]$. Quasi Monte Carlo sampling using Sobol sequences were used to generate 10^6 points for Y in $[-1, 1]$. The variance evolution obtained using these points are considered as the 'ground truth' solution. Results using the adaptive sparse grid subset collocation and the gradient based sparse grid collocation are compared against this 'exact' solution for error estimation. With progress in time, the discontinuity around $y_2(0) = 0$ will make the output (y_1, y_2, y_3) variation more and more dras-

tic with variations in $y_2(0)$ as shown in figure 2.7. So the error check at the end of the time interval will ensure that the errors in previous time steps maintain the desired accuracy. The evolution of the variance of the the outputs y_1, y_2 and y_3 within the time interval $[0, 30]$ is shown in figure 2.8. The maximum error of the variance of y_1, y_2 and y_3 defined as $max_{i=1,2,3}|\text{Var}(y_i) - \text{Var}(y_{i,sobol})|$, at $t = 30$ and for a certain error tolerance, is shown in table 2.1, where $\text{Var}(y_i)$ corresponds to the variance obtained from either the ASGC or the E-ASGC method, and $\text{Var}(y_{i,sobol})$ is the variance calculated from the Sobol samples.

2.4.5 Spatial 1-D Poisson Equation

This problem deals with a random process, making it an infinite-dimensional stochastic problem. The model problem is given by:

$$-\nabla(\kappa(x,\omega)\nabla u(x,\omega)) = f(x) \tag{2.35}$$

$$u(0,\omega) = u(1,\omega) = 0 \tag{2.36}$$

$$f(x) = 2x \tag{2.37}$$

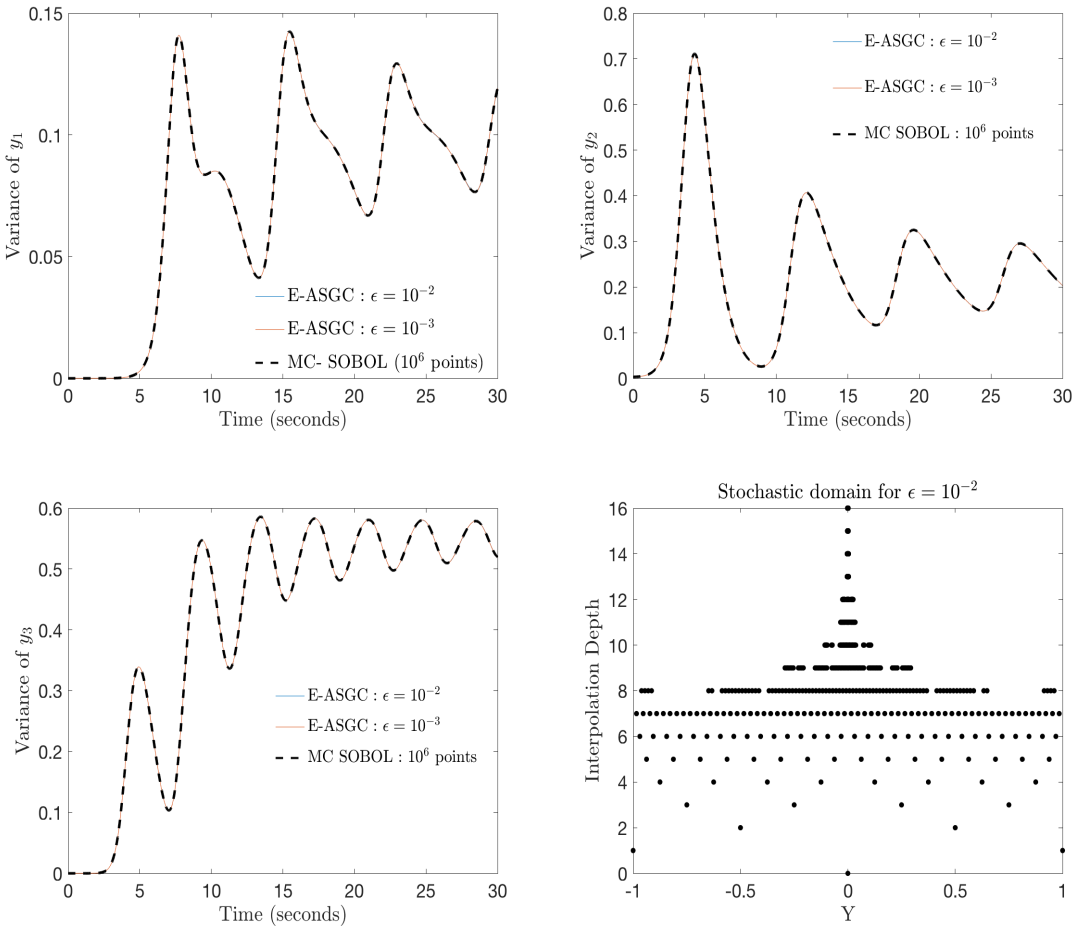


Figure 2.8: Variance of y_1, y_2 and y_3 for 1D K-O Problem

The diffusion coefficient $\kappa(x,\omega)$ is represented by a random process [110] and is approximated as a finite-dimensional random quantity by:

$$\log(\kappa(x,\omega)-0.5) \approx 1 + Y_1(\omega)(\frac{\sqrt{\pi}L}{2})^{1/2} + \sum_{i=2}^{N} \xi_n \phi_n(x) Y_n(\omega) \tag{2.38}$$

where,

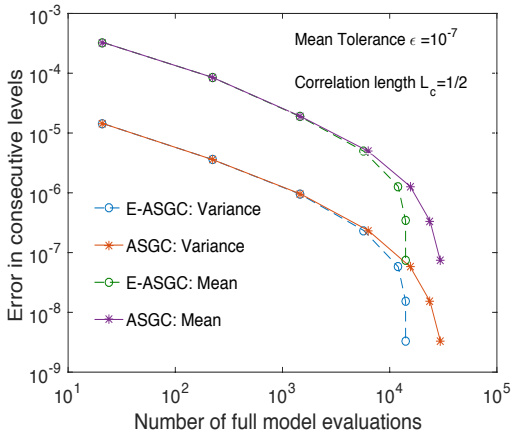
$$\xi_n = (\sqrt{\pi}L)^{1/2} \exp(\frac{-(\lfloor \frac{n}{2} \rfloor \pi L)^2}{8}), \text{ if } n > 1 \tag{2.39}$$

and

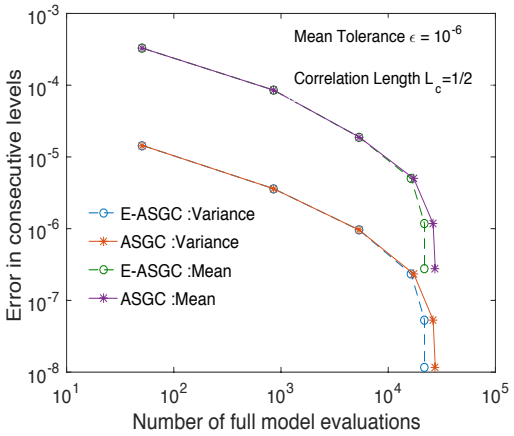
$$\phi_n(x) := \begin{cases} \sin(\frac{\lfloor \frac{n}{2} \rfloor \pi x}{L_p}), & \text{if } n \text{ is even,} \\ \cos(\frac{\lfloor \frac{n}{2} \rfloor \pi x}{L_p}), & \text{if } n \text{ is odd} \end{cases}$$

where $Y_n(\omega)$ $\{n = 1, 2, 3.....N\}$ are independent random variables which are uniformly distributed in $[0,1]$, $L_p = max[1, 2L_c]$, $L = \frac{L_c}{L_p}$, where L_c is the correlation length of the random process.

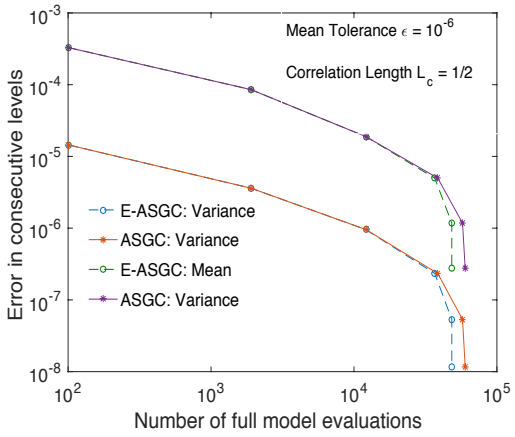
A high value of correlation length implies that the eigenvalues decay fast and the first few dimensions are of more importance than the rest of the dimensions. To make this method self-sufficient for error estimation, the error for the mean between consecutive levels is given by $E(A_{q,N}(u_N)) - E(A_{q,N+1}(u_N))$ which progressively converges to the target precision. Similarly the error in variance is given by $Var(A_{q,N}(u_N)) - Var(A_{q,N+1}(u_N))$. The convergence in the mean and variance for this problem using the efficient adaptive sparse grid collocation



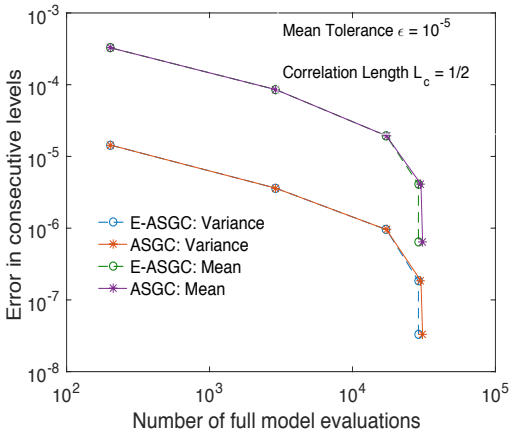
(a) 10 dimensions



(b) 25 dimensions



(c) 50 dimensions



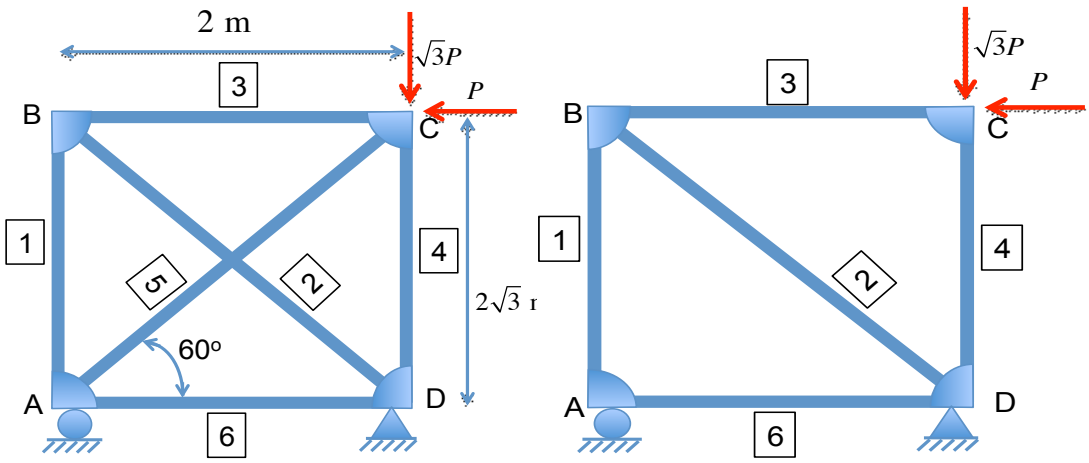
(d) 100 dimensions

Figure 2.9: Error in Mean and Variance for different dimensions in the spatial 1-D stochastic elliptic problem

(E-ASGC) method is shown and compared with the ASGC method in figure 2.9. The results show that for relatively low accuracy, the ASGC and E-ASGC have the same efficiency in all the four cases. With increase in the accuracy, it is seen that the E-ASGC performs more efficiently than the ASGC approach although the efficiency seems to decrease with increase in the dimensionality of the problem from $N = 10$ to $N = 100$.

2.4.6 Truss problem

We consider the 2D truss structure shown in figure 2.10a. It is assumed that all of the elements have the same modulus, E. The lengths of the vertical elements, horizontal elements and the diagonal elements are $\sqrt{3}L$, L and $2L$ respectively. The truss is statically indeterminate to the first degree and is analyzed using the stiffness method. The uncertainty lies in the cross-sectional area of the truss members. The variations in the cross-sectional area of the members are such that for certain combinations of the areas, the force in member 5 exceeds its critical buckling load. It is then considered to have failed and carries no load. In that scenario, the truss effectively converts to that shown in figure 2.10b. The output of interest of this problem is the force in member 4 under the given loads, as a function of the member cross-sectional areas. If member 5 fails, it leads to a discontinuous increase in the force in member 4. The indeterminate truss structure and the determinate structure without the



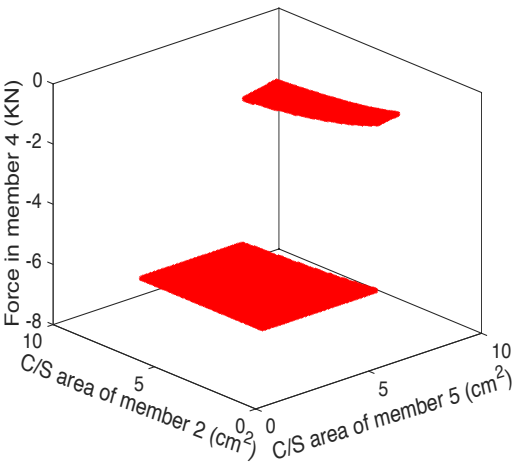
(a) Statically indeterminate (b) Statically determinate

Figure 2.10: Truss structures with and without member 5

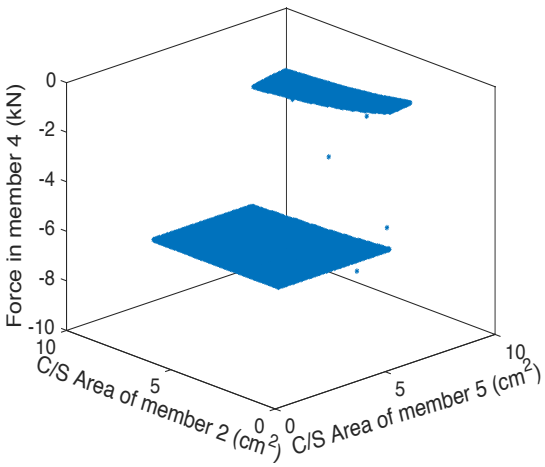
diagonal cross-brace member 5 are shown in figure 2.10. Node C in the truss structure is subjected to a vertical load $\sqrt{3}P$ acting downwards and a horizontal load P to the left.

2.4.6.1 2-dimensional random input

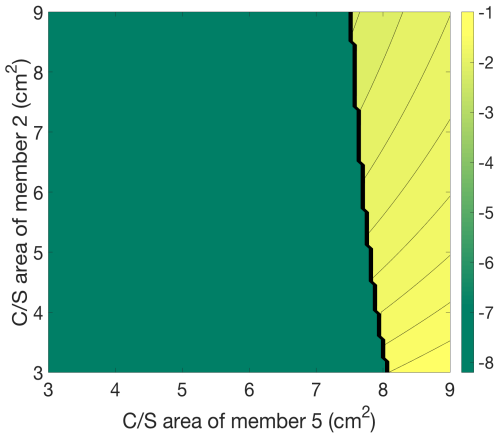
In this section, the diagonal members 2 and 5 are assumed to have random cross-sectional areas, subject to the uniform distributions $A_2 \sim \text{U}(3, 9)\ cm^2$ and $A_5 \sim \text{U}(3, 9)\ cm^2$. The input parameters given in table 2.2. The true variation in the output of interest is shown in figure 2.11a. The surrogate response surface generated by the E-ASGC approach is shown in figure 2.11b. The corresponding contour plots are shown in figures 2.11c and 2.11d respectively. Figure 2.12a shows how the input domain is sampled using the E-ASGC method. The E-ASGC method is compared with the ASGC method in the convergence plot



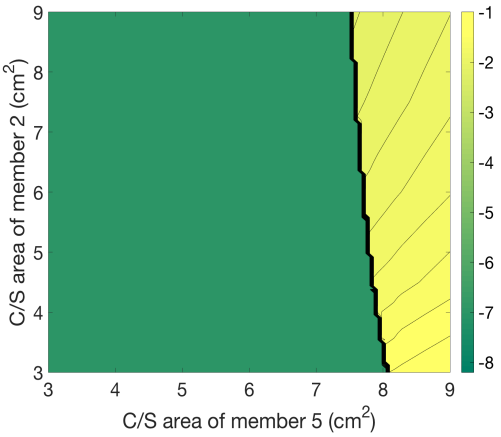
(a) Exact function



(b) Surrogate function



(c) Exact contour plot

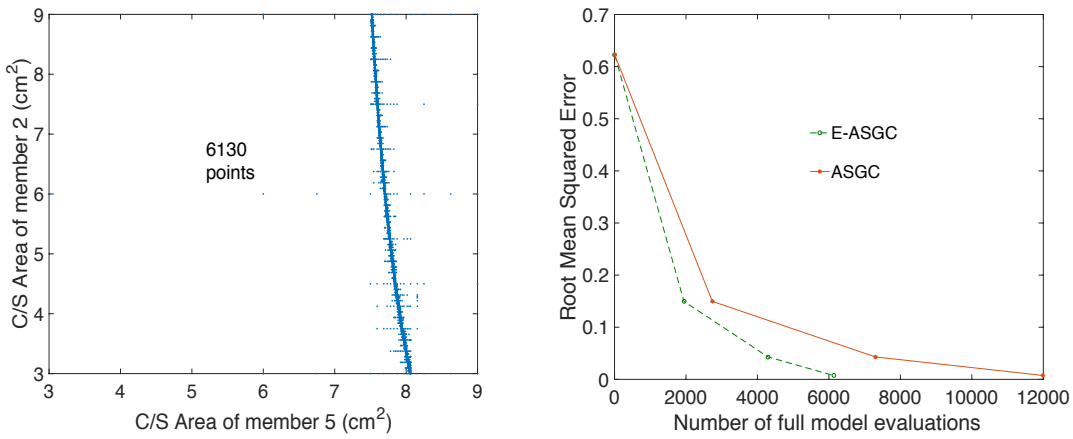


(d) Surrogate contour plot

Figure 2.11: 2 dimensional truss problem output

Table 2.2: Parameter values for the 2D truss problem

Members	Young's Modulus(E,GPa)	Area (A,cm^2)	Length(m)
1	200	6	$2\sqrt{3}$
2	200	U(3,9)	4
3	200	6	2
4	200	6	$2\sqrt{3}$
5	200	U(3,9)	4
6	200	6	2



(a) E-ASGC input domain

(b) Converge plot comparison

Figure 2.12: Input Domain and Convergence plot for 2 dimensional truss problem

of the root mean square error shown in figure 2.12b. The E-ASGC method clearly outperforms the ASGC method especially when the accuracy level is higher. For a root mean square error of 0.0053, the E-ASGC method requires 6,135 function evaluations while the ASGC method requires 12,008 function evaluations, thus reducing the sampling by almost a relative factor of 2.

2.4.6.2 3-dimensional random input

In this section, truss members 1, 3 and 5 are assumed to have random cross-sectional areas. The cross-section of the diagonal elements 5 is subject to the uniform distribution $A_5 \sim$ U(3, 9) cm^2 while the horizontal and vertical members 1 and 3 have cross-sectional areas all subject to uniform distributions A_1 $\sim$ U(5.5,6.5) cm^2 and $A_3 \sim$ U(5.5,6.5) cm^2 respectively. The input parameters are

given in table 2.3.

Table 2.3: Parameter values for the 3D truss problem

Members	Young's Modulus(E,GPa)	Area (A,cm^2)	Length(m)
1	200	U(5.5,6.5)	$2\sqrt{3}$
2	200	6	4
3	200	U(5.5,6.5)	2
4	200	6	$2\sqrt{3}$
5	200	U(3,9)	4
6	200	6	2

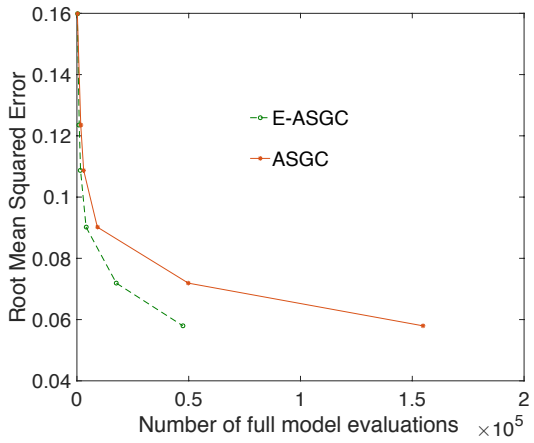


Figure 2.13: Root Mean Squared Error convergence plot for 3 dimensional truss problem

Here, the E-ASGC method is again compared with the ASGC method in the convergence plot of the root mean square error shown in figure 2.13. It is seen that with increase in accuracy, the efficiency of the E-ASGC method relative to the ASGC method also increases. For a root mean square error of 0.0580, the E-ASGC method requires $47,364$ function evaluations while the ASGC method requires $154,677$ function evaluations, thus reducing the sampling by a relative

factor of around 3.

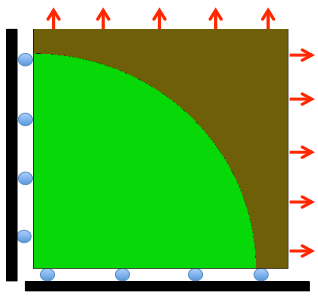
2.4.7 Application to composite materials

After demonstrating the efficiency of the E-ASGC method in the previous section, the method is now applied to a composite material model in this section. A 2-D unidirectional fiber-reinforced composite model with quarter symmetry under bi-axial tension is considered as shown in figure 2.14a. The model consists of a S-2 glass/Epoxy fiber-matrix system where the interface between the fiber and the matrix is modeled using a bi-linear traction separation law. Let the displacement loads in the x and y directions be denoted by δ_x and δ_y respectively. The independent parameters of the interface damage model are the displacement to failure d_f, the displacement to damage initiation d_i, and the interface stiffness K. The cohesive strength is a dependent parameter given by $\sigma_{max} = Kd_i$. The traction-separation curve is shown in figure 2.14b. The output of interest here is the total expended energy under the given input displacement load.

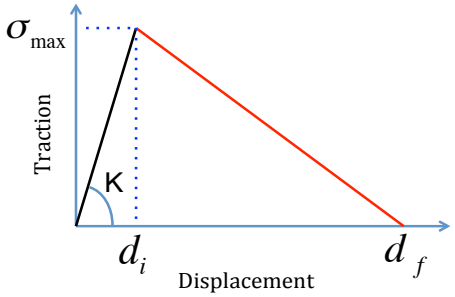
Table 2.4: Parameter values for the 2-parameter composite unit cell problem

Parameter	Values
Fiber Modulus [GPa]	87
Matrix Modulus [GPa]	3.2
Fiber Poisson's ratio	0.2
Matrix Poisson's ratio	0.35
Volume Fraction	0.6

Parameter	Values
K [N/mm^2]	$[10^4, 10^6]$
d_f [mm]	$[10^{-3}, 10^{-2}]$
d_i/d_f	0.01
δ_x=δ_y [mm]	0.005
Edge length [mm]	2.288



(a) Quarter-symmetric unit cell model



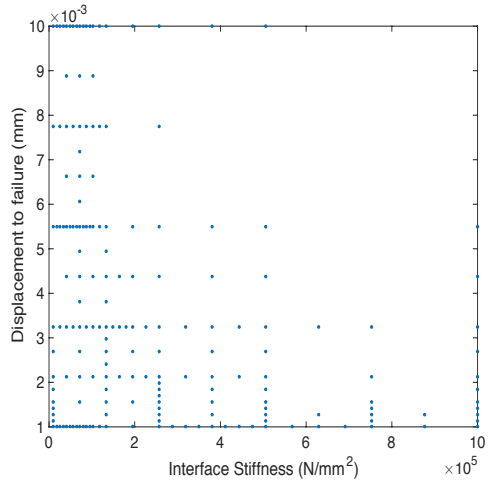
(b) Interface damage law

Figure 2.14: 2-D composite unit cell with interface damage

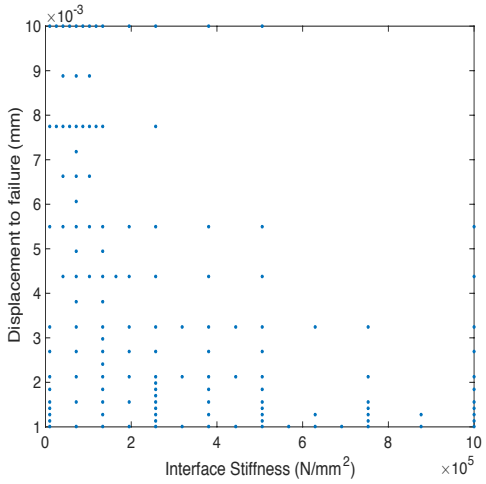
2.4.7.1 Stochastic problem with 2 random parameters

In this study, two of the interface damage law parameters, namely the interface stiffness (K) and the displacement to failure(d_f) are considered random in nature. The aim is to then to capture the variation of the energy response relative to these two parameters with as few full model evaluations as possible. The commercial FEM software ABAQUS [119] is used to solve the model with the in-built interface damage model. The parameters and their corresponding values are listed in table 2.4. The input domain of points obtained from the ASGC and the E-ASGC methods are shown in figures 2.15a and 2.15b respectively. Figure 2.15c shows the maximum absolute error converge plots while figure 2.15d shows the root mean squared error convergence plots. It is evident from the convergence plots that the efficiency of the two methods are almost equal in this case, although the E-ASGC method proves to be slightly more efficient at higher accuracy levels. It is, however, worth mentioning that that

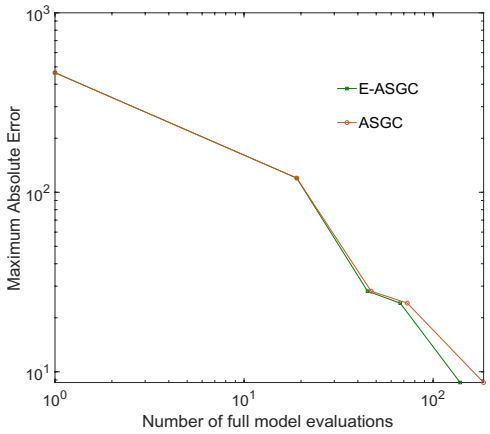
even a small reduction in the number of expensive FE simulations can result in a huge gain in the overall computational cost. The exact energy response surface is shown in figure 2.16a, while the corresponding approximate energy surface generated by the E-ASGC method with only 139 FEM model evaluations is shown in figure 2.16b.



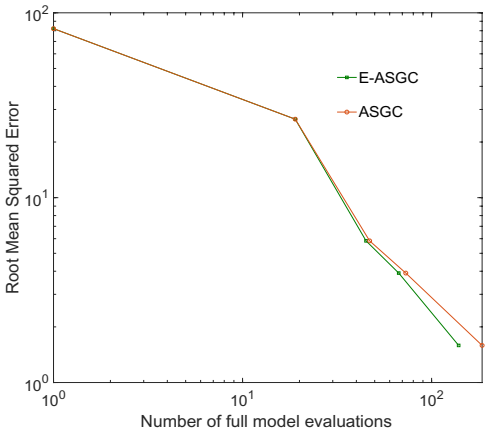
(a) ASGC input domian (185 points)



(b) E-ASGC input domain (139 points)

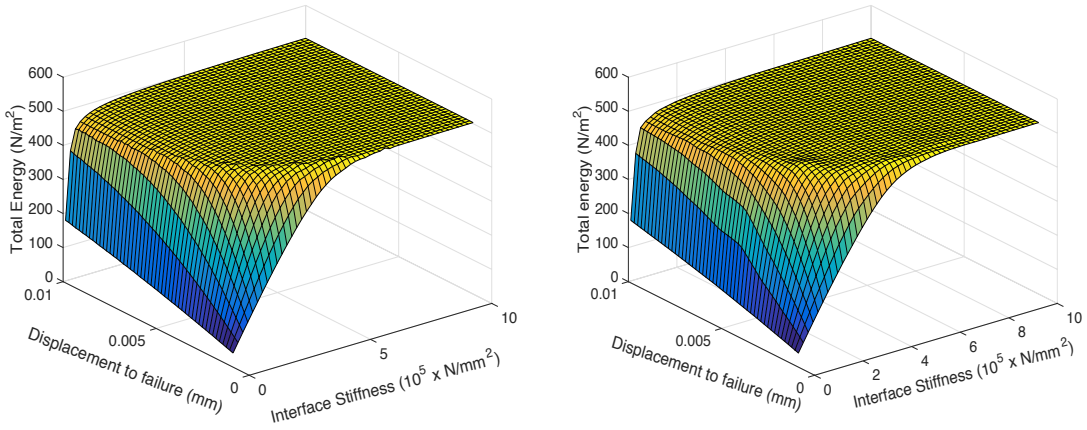


(c) Maximum Absolute Error convergence plot



(d) Root Mean Squared Error convergence plot

Figure 2.15: Performance of E-ASGC method for the composite unit cell model with interface damage



(a) Exact function (b) E-ASGC surrogate function

Figure 2.16: Comparison of the exact energy function and the E-ASGC surrogate energy response surface with 139 points

2.5 Conclusions

An efficient adaptive sparse grid approach through derivative estimation is developed which is based on the adaptive sparse grid subset collocation method (ASGC) [84], which achieves faster convergence in the case of response functions that exhibit highly localized variations (such as discontinuities) in some regions and gradual variations in other regions of the stochastic input domain. The approach is significantly more efficient than the conventional sparse grid approaches. It is at least as efficient as the adaptive sparse grid subset collocation approach (ASGC), with up to two-fold increase in efficiency, depending on the nature of the response surface. If the response surface has many sharp variations, then the 1-D cubic spline interpolation cannot be suitably implemented and the efficiency reduces to that of the adaptive sparse grid subset

approach. Also with increase in dimensions, the effectiveness relative to the ASGC method may decrease as the polynomial interpolations cover less space in the high-dimensional domain, given the control parameters remain the same as in the lower dimensional case. However, it is worth mentioning that given a complex deterministic model, any reduction in full model evaluations can be a significant contribution towards reducing computational cost.

Chapter 3

Stochastic collocation approach with adaptive mesh refinement for parametric uncertainty analysis

The E-ASGC method proposed in the previous chapter uses piece-wise linear basis functions and lacks a dimension reduction scheme. In this chapter, in order to alleviate the shortcomings of the E-ASGC method, we propose a stochastic collocation method with adaptive mesh refinement (SCAMR) to deal with high dimensional stochastic systems with discontinuities. Specifically, SCAMR is a multi-element non-intrusive second-order PCE method, which

uses generalized polynomial chaos (gPC) expansion with Legendre polynomial basis and solves for the gPC coefficients using the least squares method. It also implements an adaptive mesh (element) refinement strategy which checks for abrupt variations in the output based on a low-order gPC approximation error to track discontinuities or non-smoothness. In addition, the proposed method involves a criterion for checking possible dimensionality reduction and consequently, the decomposition of the original high-dimensional problem to a number of lower-dimensional subproblems. Specifically, this criterion checks all the existing interactions between input parameters of a specific problem based on the high-dimensional model representation (HDMR) method, and therefore automatically provides the subproblems which only involve interacting input parameters. The efficiency of the approach is demonstrated using examples of both smooth and non-smooth problems with number of input parameters up to 500, and the approach is compared against other existing algorithms. It is worth noting here that the 'stochastic collocation' term used in the SCAMR algorithm refers to the general class of methods where any suitable surrogate model (interpolation or regression based) is applied to points (structured or unstructured) sampled in the input parameter space to generate a surrogate function [6]. This is different from the usage of the term stochastic collocation in some literature [22, 120] where stochastic collocation specifically refers to an interpolation approach on structured grids using Lagrange polynomials. This

chapter is adapted from reference [121].

3.1 Introduction

One of the main aspects of uncertainty quantification (UQ) is uncertainty propagation, also called forward UQ. It aims to quantify uncertainty in the model outputs that results from uncertainty in the model inputs, which are usually represented using random variables with an associated probability distribution. The goal is therefore to estimate the response function, probability density function (PDF) or statistical moments for the model outputs efficiently.

Several efficient forward UQ methods exist in the current literature. Polynomial chaos expansion (PCE) type approaches are one such class of methods. A brief overview about different intrusive and non-intrusive PCE methods has already been discussed in chapter 2 (section 2.1). In this section, we mostly review existing PCE approaches which employ schemes to tackle high-dimensional non-smooth problems and try to motivate the work presented in this chapter.

To deal with non-smooth problems (functions), multi-element schemes have been proposed for both intrusive and non-intrusive methods. Wan and Karniadakis [15] developed a multi-element generalized polynomial chaos (MEgPC) scheme based on the stochastic Galerkin method to handle the issue of disconti-

nuities in the output and long-term integration of stochastic differential equations. This approach adaptively splits the actual input domain into smaller subdomains by calculating the relative error in variance along each parameter and maintaining a relatively low polynomial order (less than 10) in critical subdomains. However, as an intrusive approach, it requires modification of the deterministic simulation code. Foo et al. [111] introduced the non-intrusive multi-element probabilistic collocation method (MEPCM) with Lagrange polynomial basis to efficiently treat problems (functions) characterized by strong non-linearities or discontinuities and long-term integration. The criterion for adaptively splitting the input domain is similar to that in the MEgPC scheme.

Both the Galerkin and collocation versions of the multi-element gPC scheme are still dimension-dependent, since both the number of subdomains and the number of terms in the gPC expansion increase rapidly with the increase in dimensionality of the stochastic input domain. To mitigate the issue of high computational cost associated with the element decomposition in high-dimensional problems (functions), Foo and Karniadakis [122] developed the MEPCM-A approach, which combines the MEPCM with the high dimensional model representation (HDMR) [123]. The HDMR represents a high-dimensional function as a hierarchical additive combination of lower-dimensional component functions starting from one dimension to full dimension. A way to estimate the component functions is to use the cut-HDMR approach [124]. In the MEPCM-A

approach, a high-dimensional stochastic problem (function) is reduced to a series of lower-dimensional subproblems (subfunctions) by truncating the terms in the HDMR up to a certain dimensionality, ν, followed by the application of the MEPCM approach to each of these subproblems (subfunctions) with maximum dimensionality ν. Parameter ν is generally chosen to be small enough compared to the high dimensionality of the original problem (function) such that mesh (element) refinement is not computationally prohibitive. Another important parameter in the MEPCM implementation is the number of points, μ, in the interpolation rule. Parameters ν and μ are pre-fixed without regard to the actual order of interaction among the input parameters. For problems with high nominal dimensions but low effective dimensions (i.e. only a few input parameters strongly influence the output), the method proves to be efficient. However, the choice of a proper value for ν of the subproblems (subfunctions) needs more exploration. In addition, once ν is prescribed, all the interaction terms up to order ν in the HDMR are considered. Consequently, for complex systems with strong interactions among input parameters, ν may be chosen to be large for satisfactory error estimates and thus the number as well as the dimensionality of the subproblems (subfunctions) could become prohibitively large. Even with a small value of ν, the number of interaction terms can become very large for very high-dimensional problems (functions). Moreover, the model output may not be sensitive to some interaction terms with order upto ν,

and thus a significant number of unnecessary subproblems (subfunctions) are considered which increases the computational cost.

Other efficient multi-element methods have been developed based upon edge detection. For example, Archibald et al. [125] have applied the polynomial annihilation edge detection method to high dimensional functions with discontinuities. Jakeman et al. [126] have combined high-order polynomial annihilation and adaptive sparse grids to decompose the input domain into smaller elements governed by the location of discontinuities. Gorodetsky and Marzouk [127] have used polynomial annihilation to label different regions in the input domain and implemented kernel support vector machines to represent the region separating surface which gets updated by active learning.

Approaches [12, 100, 105, 106] based on local bases have also been proposed to deal with non-smoothness in the output. Ma and Zabaras [84] developed the adaptive sparse grid collocation (ASGC) method (for more details, see section 2.3.2) which incorporates an adaptive strategy enabling a local sparse grid refinement around the discontinuity region. To tackle high-dimensional stochastic problems (functions), Ma and Zabaras [128] combined a dimension-adaptive version of HDMR with ASGC (HDMR-ASGC). Initially, the importance of the component functions in HDMR are estimated through a weight measure which is expressed as the integral value of a component function of certain order with respect to the sum of the integral values of all lower-order component func-

tions. Component functions with weight measures higher than a predefined error threshold are the ones considered important. ASGC is then applied to each of the lower-dimensional subproblems (subfunctions) corresponding to the important component functions. The error indicator used in HDMR-ASGC is a function of the integral value of the basis function as well as the hierarchical surplus. It is different from the original ASGC approach [84] which uses only the surplus value as the error indicator.

In this chapter, we propose a method of stochastic collocation with adaptive mesh refinement (SCAMR). Specifically, the proposed approach uses generalized polynomial chaos (gPC) expansion with Legendre polynomial basis and solves for the gPC coefficients using the least squares method. It also implements an adaptive mesh (element) refinement strategy to track any discontinuities or non-smoothness in the output. The adaptive criteria associated with the mesh refinement strategy check for abrupt variations in the output based on the observed error from a low-order gPC approximation. SCAMR further introduces a criterion for possible dimensionality reduction, allowing for decomposition of the original high-dimensional problem (function) to a number of lower-dimensional subproblems (subfunctions). This criterion checks all the existing interactions between input parameters of a specific problem (function) based on HDMR, and consequently provides the subproblems (subfunctions) which only involve interacting parameters.

This chapter is organized as follows: section 3.2 presents the general framework for a stochastic problem. In section 3.3, we discuss the proposed method of stochastic collocation with adaptive mesh refinement in detail. In section 3.4, we demonstrate the effectiveness and efficiency of the proposed approach using various numerical examples compared to the ASGC, the HDMR-ASGC as well as the MEPCM-A approach. We finally conclude the chapter with a discussion in section 3.5.

3.2 Problem definition

Let the triplet $(\Omega, \mathcal{F}, \mathcal{P})$ represent a complete probability space, where Ω corresponds to the sample space of outcomes, $\mathcal{F} \subset 2^{\Omega}$ is the σ-algebra of measurable events in Ω, and $\mathcal{P} : \mathcal{F} \rightarrow [0, 1]$ is the probability measure. Let $\boldsymbol{\xi} = \big\langle \xi_1(\omega), \xi_2(\omega), \ldots, \xi_n(\omega) \big\rangle : \Omega \rightarrow \Xi \in \mathbb{R}^n$ be a vector of n independent random parameters, which characterize the uncertainty in the system. In the current work, we assume that the random parameters $\boldsymbol{\xi}_i$ follow a uniform distribution with a constant PDF $p(\boldsymbol{\xi}) = \rho_{\boldsymbol{\xi}};\ \boldsymbol{\xi} \in [a_1, b_1] \times [a_2, b_2] \times .... \times [a_n, b_n]$. Let $\boldsymbol{x} \in D \subset \mathcal{R}^d$ $(d \in \{1, 2, 3\})$ be the spatial variable, and $t \in (0, T]$ $(T > 0)$ be the temporal variable.

Consider a general partial differential equation

$$\begin{cases} u_t(x,t,\boldsymbol{\xi}) = \mathcal{L}(u;\boldsymbol{x},t,\boldsymbol{\xi}), & D\times(0,T]\times\Xi \\ \mathcal{B}(u;\boldsymbol{x},t,\boldsymbol{\xi}) = 0, & \partial D\times[0,T]\times\Xi, \\ u = u_0, & \bar{D}\times\{t=0\}\times\Xi, \end{cases} \tag{3.1}$$

where $\mathcal{B}$ is the operator for the boundary conditions, $\mathcal{L}$ is the differential operator, D is the spatial domain, and $u = u_0$ is the initial condition. The problem is assumed to be well-posed in input domain Ξ. The model output $u(\boldsymbol{x},t,\boldsymbol{\xi})$ is the quantity of our interest. For the convenience of notation, we do not consider the dependence of solution on the spatial and time variables $\boldsymbol{x}$ and t, and only discuss the problem for any fixed $\boldsymbol{x}\in D$ and $t\in(0,T]$. As mentioned in [129], this is standard in the UQ literature. Our goal is to quantify the uncertainty in the quantity of interest $u(\cdot,\boldsymbol{\xi}):\Xi\to\mathbb{R}$, due to the uncertainty in the parameter vector $\boldsymbol{\xi}$. Without loss of generality, we consider scalar model output.

3.3 SCAMR approach

In this section, we propose a stochastic collocation method with adaptive mesh refinement (SCAMR). Specifically, SCAMR adopts a mesh refinement scheme with a proposed criteria that checks for discontinuities or abrupt variations in the response function, as well as interactions between different input

parameters. Details are provided in the following subsections.

3.3.1 Non-intrusive gPC based Stochastic Collocation

Let $u(\boldsymbol{\xi}) \in L_2(\Xi)$ be a square-integrable function of the n-dimensional random vector $\boldsymbol{\xi}$ which can be represented using the generalized polynomial chaos expansion as

$$u(\boldsymbol{\xi}(\omega)) = \sum_{i=0}^{\infty} \hat{u}_i \Phi_i(\boldsymbol{\xi}(\omega)), \tag{3.2}$$

where $\hat{u}_i$ are the gPC coefficients and Φ_i are the Legendre polynomials for uniform $\boldsymbol{\xi}$ [9].

For numerical calculations, the series is truncated to $N+1$ terms to approximate the exact output $u(\boldsymbol{\xi}(\omega))$ with polynomial order p and is given by

$$u_p(\boldsymbol{\xi}(\omega)) = \sum_{i=0}^{N} \hat{u}_i \Phi_i(\boldsymbol{\xi}(\omega)), \qquad N+1 = \frac{(n+p)!}{n!p!}, \tag{3.3}$$

where

$$\hat{u}_i = \frac{1}{E[\Phi_i^2]} \int_{\Xi} u(\boldsymbol{\xi}) \Phi_i(\boldsymbol{\xi}) \rho(\boldsymbol{\xi}) d\boldsymbol{\xi}. \tag{3.4}$$

With collocation methods, the gPC coefficients $\hat{u}_i$ can be obtained using discrete

projection as

$$\hat{u}_i = \frac{1}{E[\Phi_i^2]} \sum_{j=1}^{M} u(\boldsymbol{\xi}^j)\Phi_i(\boldsymbol{\xi}^j)\alpha^j, \qquad i = 0, 1, \ldots, N, \tag{3.5}$$

where $\{\boldsymbol{\xi}^j, \alpha^j\}_{j=1}^M$ are sets of quadrature points and their corresponding weights.

Another collocation method for estimating the gPC coefficients utilizes interpolation on the pairs $\{\boldsymbol{\xi}^j, u(\boldsymbol{\xi}^j)\}_{j=1}^{N+1}$. The gPC coefficient vector $\hat{\mathbf{u}} = \langle \hat{u}_0, \ldots, \hat{u}_N \rangle$ is estimated by solving the following linear system:

$$\sum_{i=0}^{N} \hat{u}_i \Phi_i(\boldsymbol{\xi}^j) = u(\boldsymbol{\xi}^j), \forall j = 1, 2, \ldots, N+1.$$

The interpolation method may not produce a proper approximation if $u(\boldsymbol{\xi}^j)$ is corrupted by observational or measurement errors. The projection method, on the other hand, produces the best approximation in the weighted L_2 norm [6]. However, the quadrature nodes used in the discrete projection method have restrictions, such as the structure of the nodes and the number of the nodes.

To allow more flexibility, in terms of the location and the number of nodes, we estimate the vector of gPC coefficients by solving the following least squares problem using a set of M ($> N+1$) points:

$$\hat{\mathbf{u}} = \arg\min_{\tilde{u}} \| \sum_{i=0}^{N} \tilde{u}_i \Phi_i(\boldsymbol{\xi}) - u(\boldsymbol{\xi}) \|_2 \tag{3.6}$$

where $\tilde{\mathbf{u}} = \langle \tilde{u}_0, \tilde{u}_1, \ldots, \tilde{u}_N \rangle$ is an arbitrary gPC coefficient vector which converges to the desired vector $\hat{\mathbf{u}} = \langle \hat{u}_0, \hat{u}_1, \ldots, \hat{u}_N \rangle$ through the minimization in Eq. (3.6). Consequently, the approximated output u_p is estimated using Eq. (3.3). It is to be noted here that the set of M points may have an unstructured arrangement in the input domain.

3.3.2 Decomposition of input domain

In this section, we introduce the standard decomposition method for a stochastic input domain, where the L_2 error of the global approximation has been proven to be bounded by the local L_2 error approximations in the elements [15]. We assume a hypercube input domain in our present work. Without the loss of generality, we consider the original stochastic input domain as $\Xi = [-1, 1]^n$. It is then decomposed into n_e non-overlapping and space-filling elements Ξ_k: $\cup_{k=1}^{n_e} \Xi_k = \Xi$, $\Xi_m \cap \Xi_k = \emptyset$ for $m \neq k$ and $m, k \in \{1, 2, \ldots, n_e\}$. If a_i^k and b_i^k denote the minimum and maximum bounds of element Ξ_k along i-th parameter ($1 \leq i \leq n$), Ξ_k is the tensor product given by

$$\Xi_k = [a_1^k, b_1^k) \times [a_2^k, b_2^k) \times ..... \times [a_n^k, b_n^k). \tag{3.7}$$

Let the local input parameter vector in each element be defined as $\boldsymbol{\xi}^k = \langle \xi_1^k, \xi_2^k, \ldots, \xi_n^k \rangle$. For the purpose of applying the gPC formulation on each ele-

ment locally, the local parameter vector can be transformed to a new parameter vector $\boldsymbol{\eta} \in [-1,1]^n$ such that $\boldsymbol{\eta} = F_k(\boldsymbol{\xi}^k) = \langle \eta_1, \eta_2, \ldots, \eta_n \rangle$. The transformation is a simple scaling relationship between the $[-1,1]^n$ domain and the particular Ξ_k domain:

$$F_k : \eta_i = -1 + \frac{2}{b_i^k - a_i^k}(\xi_i^k - a_i^k), \quad \forall i = 1, 2, ..., n \tag{3.8}$$

3.3.3 Adaptive criteria

The SCAMR algorithm uses adaptive approaches for two purposes: detection of abrupt variations in the response function for non-smoothness and reduction of the high-dimensional input domain to a group of lower-dimensional input domains. Each of these are described in the following subsections.

3.3.3.1 Criterion for detecting abrupt variation in 1-dimension

In the current work, we propose to use low-order Legendre polynomials to efficiently approximate any general response function with local abruptness or discontinuities. In any domain where the function deviates significantly from the polynomial approximation, we decompose the domain further. Specifically, we consider the output variation along the centerline (straight line passing through the center of the domain) along each parameter one at a time with the

rest of the parameters fixed at their midpoints. For example, let Γ be a given n-dimensional domain (element) such that $\Gamma = [a_1, b_1) \times [a_2, b_2) \times ..... \times [a_n, b_n)$. For the i-th parameter, let $\boldsymbol{z} = \langle z_1, \ldots, z_m \rangle$ be a vector of m Chebyshev points of depth level l in the range $[a_i, b_i)$ such that $m = 2^l + 1$. In this study, depth level $l = 2$ is taken and hence $m = 5$. Then the array of input points along the centerline in the i-th parameter is $\boldsymbol{\xi}^{(i)} = \langle \boldsymbol{\xi}_1^{(i)}, \boldsymbol{\xi}_2^{(i)}, \ldots, \boldsymbol{\xi}_m^{(i)} \rangle$, where each n-dimensional point is $\boldsymbol{\xi}_j^{(i)} = \langle \frac{a_1+b_1}{2}, \ldots, \frac{a_{i-1}+b_{i-1}}{2}, z_j, \frac{a_{i+1}+b_{i+1}}{2}, \ldots, \frac{a_n+b_n}{2} \rangle, \forall j \in \{1, 2, \ldots, m\}$. Let $\boldsymbol{u}^{(i)} = \langle u_1^{(i)}, u_2^{(i)}, \ldots, u_m^{(i)} \rangle$ be the corresponding vector of m exact outputs and $\boldsymbol{u}_p^{(i)} = \langle u_{p,1}^{(i)}, u_{p,2}^{(i)}, \ldots, u_{p,m}^{(i)} \rangle$ be the corresponding 1-D gPC approximation along the i-th parameter for the current domain. The model output can then be reasonably approximated as a p-th order polynomial if

$$\|\boldsymbol{u}_p^{(i)} - \boldsymbol{u}^{(i)}\|_\infty < \epsilon_1, \tag{3.9}$$

where ϵ_1 is an error tolerance parameter. Criterion (3.9) is checked in succession with polynomial order p varying from 1 to p_{max} ($p_{max} = 2$ for non-smooth functions and 3 for smooth functions in our present work) and if it is not satisfied for any order, the i-th parameter is considered critical. All the critical parameters are then stored in descending order of the error magnitude obtained from criterion (3.9) and the domain is further decomposed along the center of the two most critical parameters. The domain subdivision is repeated for every

newly formed element until the stopping criteria are satisfied.

3.3.3.2 Criterion for dimensionality reduction

The second criterion helps in achieving dimensionality reduction. It decomposes the original high-dimensional problem (functions) to a number of lower-dimensional subproblems (subfunctions) by identifying the absence of interactions between input parameters with respect to the output of interest. It is worth mentioning that non-interaction between any two input parameter sets means that the parameter sets influence the output independent of each other, i.e., they are additive. This criterion is checked at two levels and takes advantage of the significant gains in computational efficiency by dealing with lower-dimensional subfunctions.

First level criterion. At the first level, i-th parameter is assumed non-interacting with others if

$$||\boldsymbol{u}^{(i)} - u_c||_\infty < \epsilon_1, \tag{3.10}$$

where $\boldsymbol{u}^{(i)}$ is the centerline output vector along the i-th parameter (introduced earlier) and u_c is the exact output value at the center point of the input domain Ξ. By implementing this first level criterion, the n-dimensional problem (function) will be decomposed to one r-dimensional and $n-r$ $(r \leq n)$ one-dimensional problems, where the one-dimensional problems depend on the input parame-

ters which do not interact with others.

Second level criterion. At the second level, we further decompose the r-dimensional problem to a number of lower dimensional subproblems (subfunctions) by verifying $\binom{r}{2}$ pairwise parameter interactions in the r-dimensional domain. All higher order interactions between the input parameters are derived from the pairwise interaction results. This second level criterion is derived from the HDMR representation [124, 130] and the details are provided in the following.

HDMR representation. Let $f(\boldsymbol{Y}) = f(Y_1, Y_2, ..., Y_n)$ be an n-dimensional function. Following the notation in [128], the general expression of the High Dimensional Model Representation (HDMR) for the function is given by

$$f(\boldsymbol{Y}) = f_0 + \sum_{i=1}^{n} f_i(Y_i) + \sum_{1 \le i_1 < i_2 \le n} f_{i_1 i_2}(Y_{i_1}, Y_{i_2}) + \ldots + \sum_{1 \le i_1 < ..i_s \le n} f_{i_1 ....i_s}(Y_{i_1}, ..., Y_{i_s}) + ... + f_{12...n}(Y_1, Y_2, ...Y_n) \tag{3.11}$$

where f_0 is a constant zeroth order function, $f_i()$ denotes a one-dimensional function, $f_{i_1 i_2}()$ is a two-dimensional function and so on.

As seen from Eq. (3.11), the HDMR breaks down the function $f(\boldsymbol{Y})$ into individual contributions from all possible orders of interactions among the parameters. For example, $f_i(Y_i)$ represents how input Y_i influences $f(\boldsymbol{Y})$ keeping the other input parameters fixed. The third term $f_{i_1 i_2}(Y_{i_1}, Y_{i_2})$ represents the

combined contribution of inputs Y_{i_1} and Y_{i_2} towards $f(\boldsymbol{Y})$ after their individual contributions have been accounted for through $f_i(Y_i)$. All parameters except Y_{i_1} and Y_{i_2} are kept fixed in this case. Similarly, $f_{12\ldots n}(Y_1, Y_2, \ldots Y_n)$ denotes the contribution of all input parameters taken together towards $f(\boldsymbol{Y})$ after having accounted for all lower-dimensional subfunction contributions.

Cut-HDMR [131,132] is an efficient technique for estimating the component functions in $f(\boldsymbol{Y})$ which involves evaluating $f(\boldsymbol{Y})$ on lines, planes and hyper-planes (or cuts) passing through a cut center $\boldsymbol{c}$ which is a point in the input domain. The choice of $\boldsymbol{c}$ is important as it influences the convergence of the HDMR expansion. It has been shown [133] that a suitable choice of $\boldsymbol{c}$ can be the mean vector of the input parameter ranges.

The component functions [134] are given by:

$$f_0 = f(\boldsymbol{c}) \tag{3.12}$$

$$f_i(Y_i) = f(Y_i, \boldsymbol{c}^{\{i\}}) - f_0 \quad \forall i \in \{1, 2, \ldots, n\} \tag{3.13}$$

$$f_{i_1 i_2}(Y_{i_1}, Y_{i_2}) = f(Y_{i_1}, Y_{i_2}, \boldsymbol{c}^{\{i_1, i_2\}}) - f_{i_1}(Y_{i_1}) - f_{i_2}(Y_{i_2}) - f_0, \tag{3.14}$$

$$\forall i_1, i_2 \in \{1, 2, \ldots, n\}, \text{ such that } i_1 < i_2$$

$$f_{i_1 i_2 i_3}(Y_{i_1}, Y_{i_2}, Y_{i_3}) = f(Y_{i_1}, Y_{i_2}, Y_{i_3}, \boldsymbol{c}^{\{i_1,i_2,i_3\}}) - f_{i_1 i_2}(Y_{i_1}, Y_{i_2}) - f_{i_1 i_3}(Y_{i_1}, Y_{i_3}) - f_{i_2 i_3}(Y_{i_2}, Y_{i_3}) - f_{i_1}(Y_{i_1}) - f_{i_2}(Y_{i_2}) - f_{i_3}(Y_{i_3}) - f_0, \quad (3.15)$$

$$\forall i_1, i_2, i_3 \in \{1, 2, \ldots, n\}, \text{ such that } i_1 < i_2 < i_3$$

$$\vdots$$

$$f_{12\ldots n}(Y_1, Y_2, \ldots, Y_n) = f(\boldsymbol{Y}) - f_0 - \sum_{i=1}^{n} f_i(Y_{i_1}) - \sum_{1 \le i_1 < i_2 \le n} f_{i_1 i_2}(Y_{i_1}, Y_{i_2}) - \ldots - \sum_{1 \le i_1 < .. i_{n-1} \le n} f_{i_1 \ldots i_{n-1}}(Y_{i_1}, \ldots, Y_{i_{n-1}}) \quad (3.16)$$

where $\boldsymbol{c}^{\{i\}} = \boldsymbol{c} \backslash \{Y_i\}, \boldsymbol{c}^{\{i_1,i_2\}} = \boldsymbol{c} \backslash \{Y_{i_1}, Y_{i_2}\}, \boldsymbol{c}^{\{i_1,i_2,i_3\}} = \boldsymbol{c} \backslash \{Y_{i_1}, Y_{i_2}, Y_{i_3}\}$. For sets A and B, $A \backslash B$ denotes a set with only those elements in A that are not included in B.

Pairwise non-interaction criterion derivation. Using the HDMR representation, we will now derive the non-interaction criterion for dimensionality reduction. In the proposed method, we consider only pairwise interactions of the input parameters. We thus concentrate on the 2-dimensional component function given by Eq. (3.14). Combining Eq. (3.13) with Eq. (3.14), we can

write,

$$f_{i_1 i_2}(Y_{i_1}, Y_{i_2}) = f(Y_{i_1}, Y_{i_2}, \boldsymbol{c}^{\{i_1 i_2\}}) - f(Y_{i_1}, \boldsymbol{c}^{\{i_1\}}) - f(Y_{i_2}, \boldsymbol{c}^{\{i_2\}}) + f_0 \tag{3.17}$$

For a given error tolerance ϵ_2, parameters with indices i_1 and i_2 can be considered non-interacting if the 2-dimensional component function $f_{i_1 i_2}(Y_{i_1}, Y_{i_2})$ is considered negligible, i.e., $f_{i_1 i_2}(Y_{i_1}, Y_{i_2}) \leq \epsilon_2$. This implies,

$$f(Y_{i_1}, Y_{i_2}, \boldsymbol{c}^{\{i_1, i_2\}}) - f(Y_{i_1}, \boldsymbol{c}^{\{i_1\}}) - f(Y_{i_2}, \boldsymbol{c}^{\{i_2\}}) + f_0 \leq \epsilon_2. \tag{3.18}$$

Eq. (3.18) is the pairwise non-interaction criterion.

Let us take a two-dimensional input domain as an example (see figure 3.1),

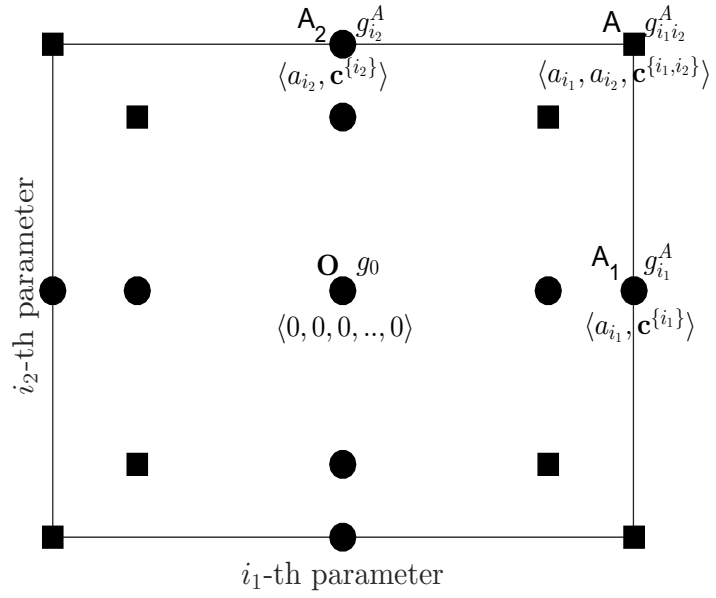


Figure 3.1: Square points denote new points introduced for the interaction check between parameters

where the input domain is projected from a higher n-dimensional domain with all the parameters fixed at the mean of their respective ranges except those two parameters (indexed i_1 and i_2). The cut center is given by $c = \langle 0, 0, \ldots, 0 \rangle$ and is denoted by point O in figure 3.1. All the square points in figure 3.1 are used to test for interaction between the two parameters. The exact values at all the square points are calculated by full model evaluations and compared with the values at those points obtained assuming both parameters are non-interacting. For example, assume the exact value at point A is $g^A_{i_1 i_2}$ and the approximated value at A assuming non-interaction is given by $g^{approx,A}_{i_1 i_2} = g^A_{i_1} + g^A_{i_2} - g_0$. The output values $g^A_{i_1}$ and $g^A_{i_2}$ correspond to input points at A_1 and A_2 which are orthogonal projections of A on axes i_1 and i_2 respectively passing through point O and g_0 is the corresponding output value. Let $\mathbf{g}^{true}_{i_1 i_2}$ be the true output vector corresponding to the square points and $\mathbf{g}^{approx}_{i_1 i_2}$ be the corresponding approximate output vector obtained from the outputs at the circular points such that $\mathbf{g}^{approx}_{i_1 i_2} = \mathbf{g}_{i_1} + \mathbf{g}_{i_2} - g_0$. Then, Eq. (3.18) is considered satisfactory if

$$\|\mathbf{g}^{true}_{i_1 i_2} - \mathbf{g}^{approx}_{i_1 i_2}\|_\infty \leq \epsilon_2 \tag{3.19}$$

As mentioned earlier, using the knowledge about each of the pairwise (2-dimensional) parameter interactions, we derive all possible higher-dimensional parameter interactions. For example, we consider a 5-dimensional response

function where $\{Y_1, Y_2, \ldots, Y_5\}$ are the input parameters. If only pairs $\{Y_1, Y_2\}$, $\{Y_2, Y_3\}$ and $\{Y_3, Y_1\}$ out of total $\binom{5}{2} = 10$ pairs are interacting based on the criterion Eq. (3.18), we decompose the five-dimensional problem (function) into one three-dimensional subproblem (subfunction) in the $\{Y_1, Y_2, Y_3\}$ domain and two one-dimensional subproblems in the Y_4 and Y_5 domains, respectively.

Subproblem representation. As mentioned in the beginning of this section, criteria (3.10) and (3.18) can potentially reduce a high dimensional problem (response function) to a set of lower dimensional subproblems (subfunctions). Next, we discuss, through an example, the effects of applying the two criteria in successive steps and how to generally represent the original response function in terms of a number of lower dimensional subfunctions. As an example, an 8-dimensional (eight input parameters) function is considered and is given by $f(Y) = Y_1Y_2Y_3 + Y_1Y_4 + Y_5 + \delta(Y_6 + Y_7 + Y_8)$ in the input domain $[0, 1]^8$. If $\delta < 2\epsilon_1$, criterion (3.10) will identify the three input parameters Y_6, Y_7 and Y_8 to be non-interacting with each of the other seven parameters. We thus have the following set of non-interacting groups of parameters given by $R = \{R_1, R_2, R_3, R_4\} = \{\{1, 2, 3, 4, 5\}, \{6\}, \{7\}, \{8\}\}$ (with $N_R = 4$) and the func-

tion can now be described by:

$$\begin{aligned} f(\boldsymbol{Y}) &= f(Y_1, Y_2, \ldots, Y_8) \\ &= g_0(Y_{R_1}, \boldsymbol{c}^{R_1}) + h_1(Y_{R_2}, \boldsymbol{c}^{R_2}) + h_2(Y_{R_3}, \boldsymbol{c}^{R_3}) \\ &\quad + h_3(Y_{R_4}, \boldsymbol{c}^{R_4}) - (N_R - 1) f_0 \\ &= g_0(Y_1, Y_2, \ldots, Y_5, \boldsymbol{c}^{\{1,2,\ldots,5\}}) + h_1(Y_6, \boldsymbol{c}^{\{6\}}) \\ &\quad + h_2(Y_7, \boldsymbol{c}^{\{7\}}) + h_3(Y_8, \boldsymbol{c}^{\{8\}}) - 3 f_0. \end{aligned} \tag{3.20}$$

Eq. (3.20) thus shows that the 8-dimensional function has been reduced to a maximum dimensionality of $r = 5$ using the first level check.

When criterion (3.18) is now checked with the 5-dimensional subfunction $g_0()$, a set $I = \{\{1,2\},\{1,3\},\{2,3\},\{1,4\}\}$ is obtained whose elements consist of pairs of parameters which are interacting. Using information from the set I, $R_1 = \{1,2,3,4,5\}$ is reduced to the following set Q:

$$Q = \{Q_1, Q_2, Q_3\} = \{\{1,2,3\},\{1,4\},\{5\}\}$$

where elements Q_j consists of indices for input parameters such that the parameter sets with indices Q_j and Q_k ($\forall j,k \in \{1,2,3\}$ and $j \neq k$) are additive. We note that the presence of the 3-dimensional interaction $\{1,2,3\}$ have been derived from the interacting pairs $\{1,2\},\{1,3\}$ and $\{2,3\}$. This is how higher

level interactions are derived from pairwise interaction results. Putting it all together, a set S is formed and is given by

$$\begin{aligned} S &= \{S_1, S_2, S_3, S_4, S_5, S_6\} \\ &= \{\{6\}, \{7\}, \{8\}, \{1,2,3\}, \{1,4\}, \{5\}\} \end{aligned}$$

where the number of elements $N_S = 6$, and elements S_i consists of indices for input parameters such that the parameter sets with indices S_i and S_j ($\forall i, j \in \{1, 2, \ldots, 6\}$ and $i \neq j$) are additive. Here, the set T involving the overlapping parameters in S (i.e., $T = \{S_i \cap S_l\} \setminus \emptyset, \quad \forall i < l \in \{1, ..., N_S\}$) is given by $T = \{T_1\} = \{\{1\}\}$ with $N_T = 1$. The function $g_0()$ will now be given by:

$$\begin{aligned} g_0(Y_1, Y_2, ...Y_5) &= h_4(\boldsymbol{Y}_{S_4}, \boldsymbol{c}^{S_4}) + h_5(\boldsymbol{Y}_{S_5}, \boldsymbol{c}^{S_5}) \\ &\quad + h_6(\boldsymbol{Y}_{S_6}, \boldsymbol{c}^{S_6}) - U_1 p_1(\boldsymbol{Y}_{T_1}, \boldsymbol{c}^{T_1}) - f_0 \\ &= h_4(Y_1, Y_2, Y_3, \boldsymbol{c}^{\{1,2,3\}}) + h_5(Y_1, Y_4, \boldsymbol{c}^{\{1,4\}}) \\ &\quad + h_6(Y_5, \boldsymbol{c}^{\{5\}}) - p_1(Y_1, \boldsymbol{c}^{\{1\}}) - f_0 \end{aligned} \tag{3.21}$$

where the coefficient U_j ($\forall j = 1, 2, ..., N_T$) associated with each set T_j equals the difference between frequency of its occurrence in S and the frequency of its occurrence in T. In this example, $j = 1$. Note that the frequency of occurrence of an index set in S or T is the number of times an index set features in S or T

by itself or as a subset in a larger index set. If the elements in T are subsets of each other, the factors need to be adjusted accordingly. Thus, the function $f(\boldsymbol{Y})$ is given by:

$$\begin{aligned} f(\boldsymbol{Y}) &= h_4(\boldsymbol{Y}_{S_4}, \boldsymbol{c}^{S_4}) + h_5(\boldsymbol{Y}_{S_5}, \boldsymbol{c}^{S_5}) + h_6(\boldsymbol{Y}_{S_6}, \boldsymbol{c}^{S_6}) - p_1(\boldsymbol{Y}_{T_1}, \boldsymbol{c}^{T_1}) - f_0 \\ &\quad + h_1(Y_6, \boldsymbol{c}^{\{6\}}) + h_2(Y_7, \boldsymbol{c}^{\{7\}}) + h_3(Y_8, \boldsymbol{c}^{\{8\}}) - 3f_0 \\ &= h_1(\boldsymbol{Y}_{S_1}, \boldsymbol{c}^{S_1}) + h_2(\boldsymbol{Y}_{S_2}, \boldsymbol{c}^{S_2}) + h_3(\boldsymbol{Y}_{S_3}, \boldsymbol{c}^{S_3}) + h_4(\boldsymbol{Y}_{S_4}, \boldsymbol{c}^{S_4}) \\ &\quad + h_5(\boldsymbol{Y}_{S_5}, \boldsymbol{c}^{S_5}) + h_6(\boldsymbol{Y}_{S_6}, \boldsymbol{c}^{S_6}) - p_1(\boldsymbol{Y}_{T_1}, \boldsymbol{c}^{T_1}) - 4f_0 \\ &= \sum_{i=1}^{6} h_i(\boldsymbol{Y}_{S_i}, \boldsymbol{c}^{S_i}) - \sum_{j=1}^{1} p_j(\boldsymbol{Y}_{T_j}, \boldsymbol{c}^{T_j}) - 4f_0. \end{aligned} \tag{3.22}$$

The example can be easily generalized to any arbitrary function of n input parameters using the following two equations corresponding to Eqs. (3.20) and (3.22) respectively, given by,

$$f(\boldsymbol{Y}) = g_0(Y_{R_1}, \boldsymbol{c}^{R_1}) + \sum_{i=1}^{N_R-1} h_i(Y_{R_{i+1}}, \boldsymbol{c}^{R_{i+1}}) - (N_R - 1)f_0 \tag{3.23}$$

and the HDMR-like representation

$$f(Y_1, Y_2, ...Y_n) = \sum_{i=1}^{N_S} h_i(\boldsymbol{Y}_{S_i}, \boldsymbol{c}^{S_i}) - \sum_{j=1}^{N_T} U_j p_j(\boldsymbol{Y}_{T_j}, \boldsymbol{c}^{T_j}) - V f_0, \tag{3.24}$$

where $h_i()$ is an $|S_i|$-dimensional function, $p_j()$ is a $|T_j|$-dimensional function, and $V = N_S - \sum_{j=1}^{N_T} U_j - 1$ is an integer constant. In case of no overlapping of

elements in S, i.e., $\cap_{i=1}^{N_S} S_i = \emptyset$, then $T = \{\emptyset\}$ and $N_T = 0$.

3.3.3.3 gPC approximation error

Let us consider a d-dimensional input domain where $1 \leq d \leq n$. Let $\boldsymbol{\xi_a} = \left\langle \boldsymbol{\xi}_{\boldsymbol{a},1}, \boldsymbol{\xi}_{\boldsymbol{a},2}, \ldots, \boldsymbol{\xi}_{\boldsymbol{a},m} \right\rangle$ be a d-dimensional array of m Clenshaw-Curtis sparse grid points of depth level 2. There may also exist an additional array of q unstructured points $\boldsymbol{\xi_b} = \left\langle \boldsymbol{\xi}_{\boldsymbol{b},1}, \boldsymbol{\xi}_{\boldsymbol{b},2}, \ldots, \boldsymbol{\xi}_{\boldsymbol{b},q} \right\rangle$ which have been previously evaluated. They correspond to sparse grid points in all "predecessor" elements that are contained in the current domain. Let $\boldsymbol{u}_p$ be the low-order gPC approximation for the current domain corresponding to input points $\boldsymbol{\xi}$ where the gPC coefficients are calculated by solving a least squares problem given by Eq. (3.6) such that $\boldsymbol{\xi} = \left\langle \boldsymbol{\xi_a}, \boldsymbol{\xi_b} \right\rangle$ and $q + m = M$. Assuming $\boldsymbol{u}$ is the corresponding exact solution vector, the domain can be suitably approximated by a low-order gPC approximation if

$$\|\boldsymbol{u}_p - \boldsymbol{u}\|_\infty < \epsilon_1 \tag{3.25}$$

Criterion (3.25) is checked in succession with polynomial order p varying from 1 to 3 (depending on the number of points in that domain) and if it is not satisfied for any order, the domain is further subdivided into smaller elements along the center of its most critical parameter.

3.3.4 Numerical implementation

The proposed algorithm is discussed below:

Initialization and stopping criteria. The dimension n of the problem is first determined by the number of input random parameters considered in the model problem. N_{iter} is the maximum number of iterations in the adaptive mesh refinement algorithm. V_{min} is a minimum hyper-volume fraction of the non-converged elements below which the subdivision into smaller elements is stopped. When N_{iter} is reached or the total hyper-volume fraction of the non-converged elements is less than V_{min}, the remaining non-converged elements are approximated by a first order gPC expansion and the algorithm terminates. Error tolerance parameters ϵ_1 and ϵ_2 are related to criteria (3.9), (3.10), (3.18) and (3.25). With decrease in the values of the chosen tolerance parameters, the approximation error also has a decreasing trend but with an increase in the computational cost because of more number of full model evaluations.

Checking global smoothness and possible dimensionality reduction. This step initiates with the implementation of a first order gPC approximation in the original n-dimensional input domain. The gPC coefficients are evaluated using the discrete projection method given by Eq. (3.5) using Clenshaw-Curtis sparse grid points of depth level 1. The accuracy of the approximation is tested using criterion (3.25). If the criterion is not satisfied, we go to the step of performing a one-dimensional (1-D) abrupt variation check. Otherwise, the first

order gPC approximation is considered satisfactory and the algorithm skips to the surrogate value extraction step.

The 1-D abrupt variation check is now performed on the input domain to identify the influence of each parameter towards the output of interest. Criterion (3.9) is used to identify the critical parameters while criterion (3.10) helps to reduce the n-dimensional problem to a number of subproblems with a maximum dimensionality of r where $r < n$. The interaction check is performed next, on the r-dimensional input domain using criterion (3.18) to further reduce the maximum dimensionality to $w(< r)$ where $w = \max(|S_i|), \forall S_i \in S$.

If any of the parameters are found to be critical based on the criterion of global abrupt variation, we directly go to the step of adaptive mesh refinement. Otherwise, a second-order gPC approximation is now performed in the original n-dimensional input domain using the discrete projection method. The function at the Clenshaw-Curtis sparse grid points of depth level 2 used for this approximation has already been evaluated in previous step of interaction check. Therefore, there is no extra computational cost involved for function evaluations in this step. The accuracy of the approximation is tested using criterion (3.25). If the criterion is satisfied, the second-order gPC approximation is considered satisfactory and the algorithm skips to the surrogate value generation step. Otherwise, we go to the next step.

Adaptive mesh refinement. This part of the algorithm in general deals with

Algorithm 1 : **Summarized steps**

Initialization
Set n, N_{iter}, V_{min}, ϵ_1, and ϵ_2.
Global checks and dimensionality reduction
perform first-order gPC on the original input domain using Eq. (3.5)
if $||\boldsymbol{u}^p - \boldsymbol{u}||_\infty < \epsilon_1$ (see Eq. (3.25)) **then**
  go to the Surrogate value extraction step
else
  perform abrupt variation check using criterion (3.9)
  perform dimensionality reduction using criteria (3.10) and (3.18) to reduce the original problem into a number of lower dimensional subproblems.
  if $||\boldsymbol{u}_p^{(i)} - \boldsymbol{u}^{(i)}||_\infty < \epsilon_1$ (see Eq. (3.9)) for all parameters **then**
    perform second-order gPC on the original input domain using Eq. (3.5)
    if $||\boldsymbol{u}^p - \boldsymbol{u}||_\infty < \epsilon_1$ (see Eq. (3.25)) **then**
      go to the Surrogate value extraction step
    else
      go to the Adaptive mesh refinement step
    end if
  else
    go to the Adaptive mesh refinement step
  end if
end if
Adaptive mesh refinement
for all lower-dimensional subproblems **do**
  check abrupt variations using criterion (3.9)
  if criterion (3.9) is satisfied **then**
    check gPC approximation using criterion (3.25)
    if criterion (3.25) is not satisfied **then**
      subdivide the element along the center of its most critical parameter
    end if
  else
    subdivide the element along the center of its two most critical parameters
  end if
end for
Surrogate value extraction
extract output values corresponding to query inputs from the approximate model obtained.

$(N_S + N_T)$ low dimensional subproblems as mentioned in section 3.3.3.2. For a subproblem P_i $(1 \leq i \leq N_S + N_T)\}$), the algorithm initiates with the subdivision of the subproblem domain into elements along its two most critical parameters. The iteration count $Iter$ starts here. For each of the E_{P_i} elements formed in P_i in a certain iteration, an abrupt variation check is performed as was done on the original n-dimensional domain. If the low-order approximation criterion (3.9) is not met, the element $E_{P_i}^j$ $(j \in \{1, 2, ..., E_{P_i}\})$ is again subdivided into smaller elements along its two most critical parameters. Satisfaction of criterion (3.9) implies there are no abrupt variations in the current element. This leads to checking criterion (3.25) for low-order gPC approximation in the whole element. If that criterion is met, the element $E_{P_i}^j$ is said to have converged for the given tolerance ϵ_1 and can be suitably approximated by the gPC approximation. The polynomial order, the coefficient vector and the range of the converged element is then stored for future surrogate retrieval. If criterion (3.25) is not satisfied, then the element is also subdivided into smaller elements. This procedure is performed for all E_{P_i} elements and all the new subelements formed undergo similar operations at the next iteration $Iter = Iter + 1$. At the end of each iteration, the hyper volume V of the subelements created and the number of iterations $Iter$ are compared with the corresponding critical values V_{min} and N_{iter} respectively to check if either of the two stopping criteria is met. If the stopping condition gets satisified, then all the remaining subelements are

approximated by a first order gPC approximation. After meeting the stopping criteria, the next subproblem is taken up and we repeat the process of characterizing it.

Surrogate value extraction. After having characterized the n-dimensional problem through the various steps mentioned, the final step is to generate output values corresponding to arbitrary query input points in the n-dimensional domain and also output statistics, such as, mean. Output value estimation corresponding to a query input involves locating the element in which the query point lies in each subproblem. The stored information for that element is then retrieved to generate the local surrogate output values in each subproblem, which are then combined together to get the global output value. Mean value estimation is performed by evaluating the integration in each of the elements in each subproblem. For each subproblem, the global mean is calculated by the weighted average of local means corresponding to each element, and the weight is based on the ratio of the hyper-volume of the elements and the hyper-volume of the whole domain.

A summary of the all the above steps is given in algorithm 1.

3.3.5 Comparison with other algorithms

In this section, we discuss the difference between our approach and the MEPCM-A, ASGC and HDMR-ASGC methods.

The main difference between MEPCM-A and SCAMR lies in the value of the parameter ν, the number of terms given ν, and the method of calculating the gPC coefficients. First, the parameter ν in MEPCM-A is prefixed at the start of the algorithm and is independent of the original problem. On the other hand, SCAMR approach learns from the given problem all the interactions among the input parameters and then decomposes the problem into subproblems of appropriate dimensions. For example, consider an eight-dimensional function. If there exists interaction among five input parameters and the other three parameters are each non-interacting with the rest, SCAMR will check criteria (3.10) and (3.19) and will be able to figure out (for an appropriate tolerance value) that the maximum dimensionality of the subproblems is 5. There will be convergence in the approximation error with increase in number of samples. On the other hand, MEPCM-A will usually prefix ν to be 2 or 3 and thus may incur an error even with the increase of number of samples because of neglecting higher-dimensional interactions. Secondly, all interaction terms up to order ν in the HDMR are considered in MEPCM-A while SCAMR only considers the parameter sets that are actually interacting with each other. For example, considering the same eight-dimensional function mentioned above, if $\nu = 3$ for MEPCM-A, all $\binom{8}{3}+\binom{8}{2}+\binom{8}{1}$ terms are considered. This means 56 3-dimensional functions, 28 2-dimensional functions and 8 1-dimensional functions need to be approximated. But in SCAMR, only one 5-dimensional function and three

1-dimensional functions need to be approximated. However, it is noted that samples in all 28 2-dimensional functions need to be considered to perform the interaction checks. Thirdly, MEPCM-A uses interpolation to calculate gPC coefficients while SCAMR adopts the least squares method. In addition, the criterion for mesh refinement in MEPCM-A is based on the relative error in variance along each parameter, while SCAMR performs the centerline abrupt variation and the gPC approximation error checks.

SCMAR differs from ASGC in the following aspects. ASGC is based on adaptive sampling in the original domain and does not involve any domain decomposition. It calculates error hierarchically and adds samples in the critical regions. SCAMR involves a domain decomposition technique. It uses one-dimensional gPC approximation technique to detect critical dimensions in a certain domain. In addition, ASGC performs linear interpolation to build the surrogate while SCAMR performs low order gPC approximations on the individual domains using least squares minimization.

HDMR-ASGC combines HDMR and ASGC where the component functions (subproblems) of certain order in HDMR are identified as important based on a weight measure involving lower order component functions. In SCAMR, important component functions of all order are chosen based on an interaction criterion using only 2-dimensional component functions.

3.4 Numerical results

In this section, SCAMR is applied to a variety of functions with smoothness as well as discontinuities and number of input parameters as high as 500. Through these examples, its performance is tested against existing efficient algorithms, like, ASGC [84], HDMR-ASGC [128] and MEPCM-A [122]. For each of the following examples, a simulation of the SCAMR algorithm was run with fixed tolerance $\epsilon_1 = \epsilon_2 = \epsilon$. The value of ϵ is varied over a range and for each value of ϵ, the overall error and the number of function evaluations are estimated. It is worth mentioning that the range of value chosen for ϵ may be different for different examples since the error calculation is done using absolute values of the output. For all the examples in this chapter, $V_{min} = 10^{-6}$ and $N_{iter} = 25$.

3.4.1 Demonstration of SCAMR performance

We first demonstrate the effectiveness and efficiency of the proposed SCAMR method using simple smooth functions with stochastic input domains of varying dimensions. Then, we will focus on functions with non-smoothness or discontinuities in the stochastic input domain, as well as a high-dimensional stochastic elliptic problem. Our results are compared to those from ASGC method since both approaches use low-order polynomials as a basis and both

use adaptivity to track discontinuities. Specifically, we compare the root mean squared error calculated using $N = 10^5$ randomly generated samples, given by

$$\epsilon = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f(\mathbf{x_i}) - \tilde{f}(\mathbf{x_i}))^2}, \tag{3.26}$$

where f is the exact function and $\tilde{f}$ is the numerical approximation using ASGC or SCAMR.

3.4.1.1 Performance of SCAMR on smooth functions

We first implement the proposed method on a few simple smooth functions of varying dimensions. The two-dimensional test functions are quadratic and sine functions defined as follows.

$$f_1(x_1, x_2) = x_1^2 + x_2^2, \tag{3.27}$$

$$f_2(x_1, x_2) = \sin(4x_1)\sin(4x_2), \tag{3.28}$$

where x_i are i.i.d. uniform random variables in $[0, 1]$ ($i = 1, 2$). The tolerance vector $\boldsymbol{\epsilon}$ is given by $[1, 3e\text{-}1, 1e\text{-}1, 3e\text{-}2, 1e\text{-}2, 3e\text{-}3, 1e\text{-}3, 3e\text{-}4, 1e\text{-}4]$. The exact functions are provided in figures 3.2a and 3.2b for f_1 and f_2 respectively. Clearly, the product of sine functions f_2 exhibits more abrupt variations than the summation of quadratic functions f_1 in the $[0, 1]^2$ domain; therefore, one would expect slower convergence of the numerical approximation for f_2. The numerical

errors of SCAMR method are provided in figures 3.2c and 3.2d, and compared to those from ASGC method. From the results, one can observe that i) both SCAMR and ASGC methods have slower convergence for f_2 compared to f_1 as we expected, and ii) our proposed SCAMR approach converges faster than ASGC for both the functions.

We extend two-dimensional quadratic and sine functions to four and ten dimensions as follows.

$$f_3(x_1, x_2, x_3, x_4) = \sum_{i=1}^{4} x_i^2, \tag{3.29}$$

$$f_4(x_1, x_2, x_3, x_4) = \sum_{i=1}^{4} \sin(4x_i), \tag{3.30}$$

$$f_5(x_1, x_2, x_3, x_4) = \sin(4x_1)\sin(4x_2) + \sin(4x_3)\sin(4x_4), \tag{3.31}$$

$$f_6(x_1, x_2, \ldots, x_{10}) = \sum_{i=1}^{10} \sin(4x_i), \tag{3.32}$$

where x_i are i.i.d. uniform random variables in $[0, 1]$ ($i = 1, 2, \ldots, 10$). The tolerance vector ϵ for the 4D functions is given by [2, 1, 3e-1, 1e-1, 3e-2, 1e-2, 3e-3, 1e-3, 3e-4, 1e-4] and for the 10D function is given by [5, 1, 3e-1, 1e-1, 3e-2, 1e-2, 3e-3, 1e-3, 3e-4, 1e-4]. The functions f_3, f_4 and f_6 are independent of the interaction terms between the inputs, while f_5 depends on some interaction terms between the inputs. The numerical errors of both the SCAMR and the ASGC methods are provided in figure 3.3 with respect to number of function evaluations. The numerical approximation from both methods converges slower as

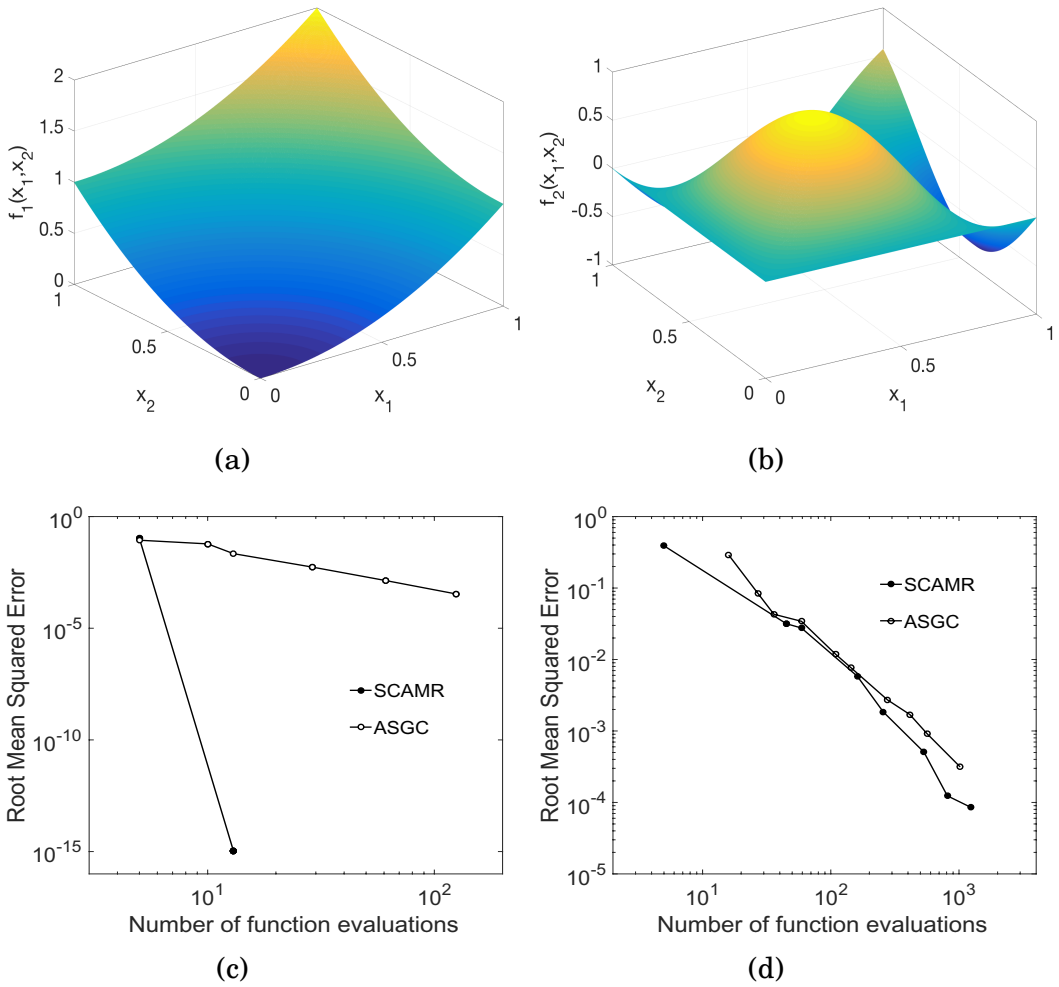


Figure 3.2: Results for 2D smooth functions: (a) surface plot of function f_1; (b) surface plot of function f_2; (c) error of estimated f_1 using SCAMR and the ASGC method; and (d) error of estimated f_2 using SCAMR and the ASGC method.

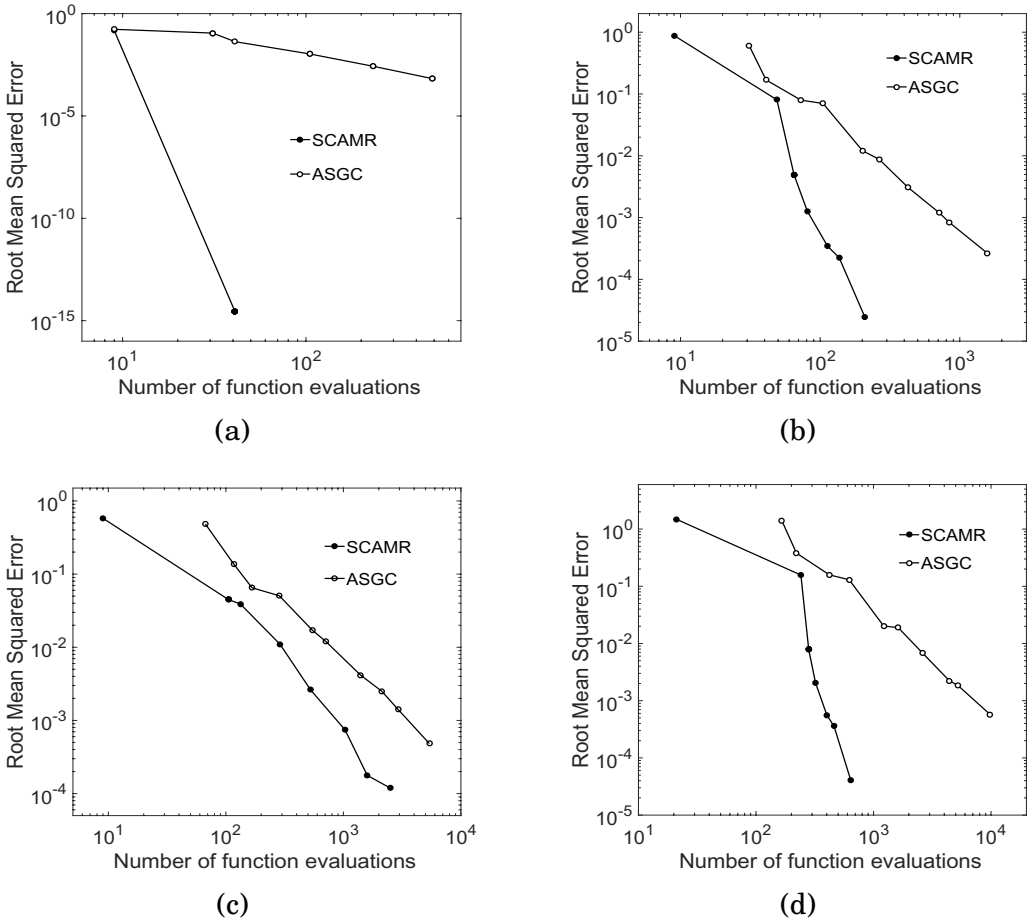


Figure 3.3: Error analysis of SCAMR and ASGC methods for 4D and 10D smooth functions: (a) 4D f_3, (b) 4D f_4, (c) 4D f_5, and (d) 10D f_6.

the complexity of the function increases, such as, from a polynomial function to a sine function, from an additive function to a multiplicative function or from a lower dimensional (4-D) function to a higher dimensional (10-D) function. Figure 3.3 shows that SCAMR converges faster than ASGC for all four smooth functions.

Having tested the SCAMR approach on smooth functions with random inputs in different dimensions, we will next discuss its performance on non-

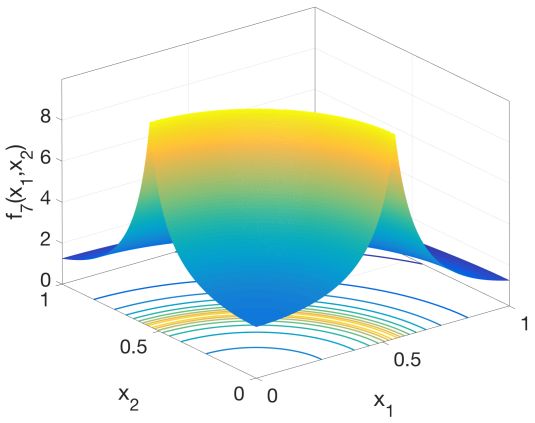


Figure 3.4: Surface plot of function $f_7(x_1, x_2)$.

smooth functions.

3.4.1.2 Performance of SCAMR on functions with line singularity

Here we adopt the same 2-D function with line singularity as in [84].

$$f_7(x_1, x_2) = \frac{1}{|0.3 - x_1^2 - x_2^2| + 0.1}. \tag{3.33}$$

The function is plotted in figure 3.4. Clearly, the function has a C^1 discontinuity going across both x_1 and x_2 directions. The 4-D and 10-D extensions of the above

function are defined as

$$f_8(x_1, x_2, x_3, x_4) = \frac{1}{|0.3 - x_1^2 - x_2^2| + 0.1} + \sum_{i=3}^{4} x_i, \tag{3.34}$$

$$f_9(x_1, x_2, \ldots, x_{10}) = \frac{1}{|0.3 - x_1^2 - x_2^2| + 0.1} + \sum_{i=3}^{10} x_i \tag{3.35}$$

where x_i are i.i.d. uniform random variables in $[0, 1]$ ($i = 1, 2, \ldots, 10$). The tolerance vector ϵ is given by [1, 3e-1, 1e-1, 3e-2, 1e-2, 3e-3]. Notice that the added dimensions in f_8 and f_9 are not interactive with x_1 and x_2. Therefore one would expect that the computational cost will not increase dramatically as the number of parameters increases.

The proposed SCAMR approach is implemented for the above 2-D, 4-D and 10-D functions. The locations of function evaluations for the 2-D function f_7 are plotted in figure 3.5a. The plot shows that the line singularity is well captured by the approach and more function evaluations are required in the area of line singularity as expected. The error analysis of the numerical approximations are provided in figures 3.5b, 3.5c and 3.5d for functions f_7, f_8 and f_9, respectively. From the figure, one can observe that the convergence rates of SCAMR are similar for the three functions with different dimensions as expected. The SCAMR approach converges faster than ASGC for all three functions.

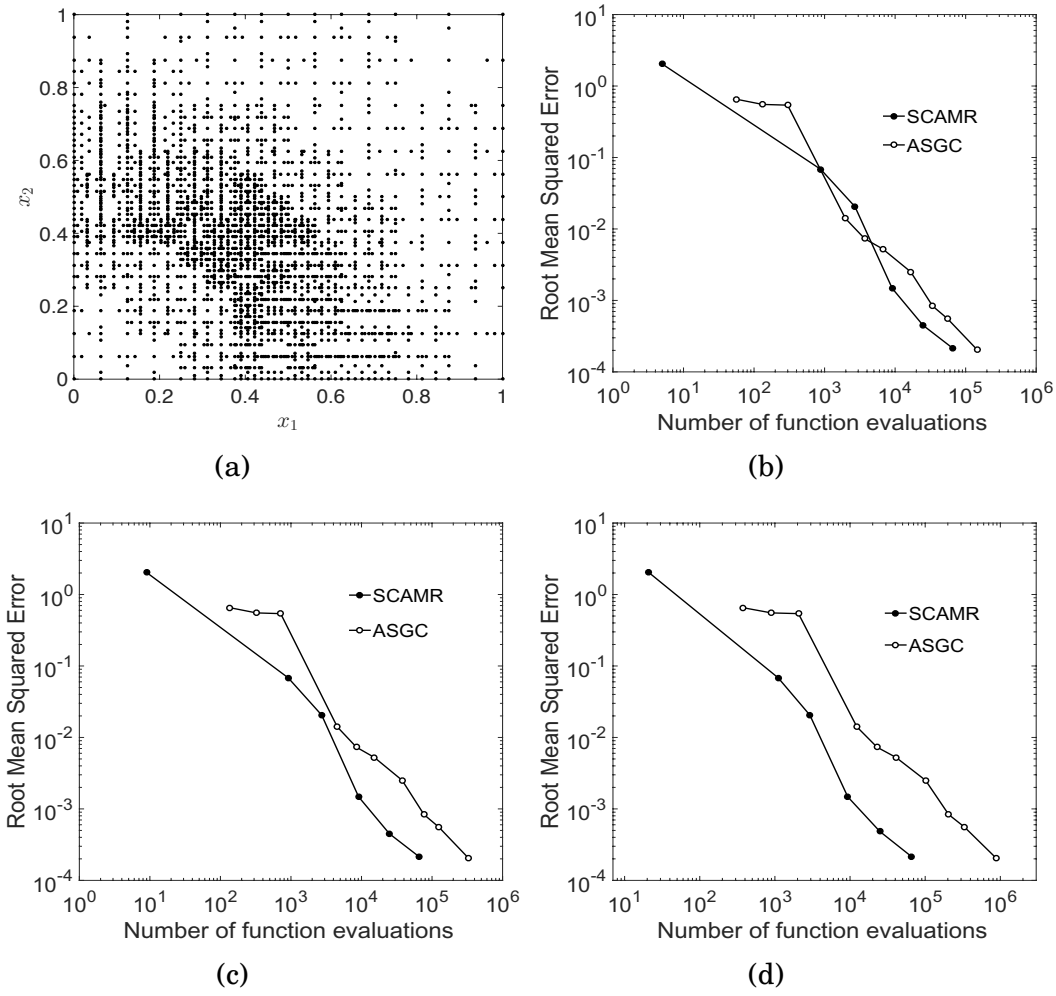


Figure 3.5: Input domain and error analysis for functions with line singularity: (a) input domain for function f_7, (b) numerical error as a function of the number of samples for 2D f_7, (c) numerical error as a function of the number of samples for 4D f_8, and (d) numerical error as a function of the number of samples for 10D f_9.

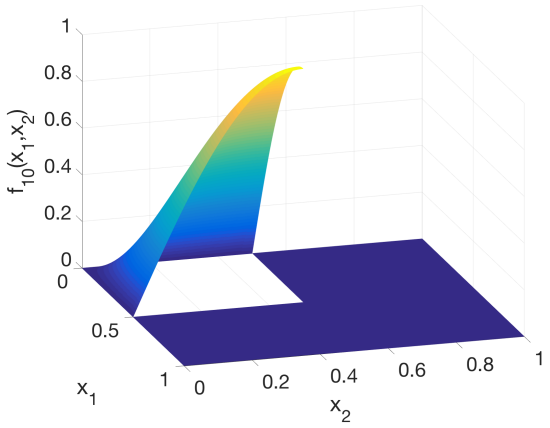


Figure 3.6: Surface plot of function $f_{10}(x_1, x_2)$.

3.4.1.3 Performance of SCAMR on functions with C^0 discontinuity

SCAMR is tested on another 2-D function, this one with a C^0 discontinuity as in [112]:

$$f_{10}(x_1, x_2) = \begin{cases} 0, & \text{if } x_1 > 0.5 \text{ or } x_2 > 0.5, \\ \sin(\pi x_1)\sin(\pi x_2), & \text{otherwise} \end{cases}$$

The function is plotted in figure 3.6.

Similarly, we extend it to 4-D and 10-D functions with discontinuity as

$$f_{11}(x_1, x_2, x_3, x_4) = \begin{cases} \sum_{i=3}^{4} x_i, & \text{if } x_1 > 0.5 \text{ or } x_2 > 0.5, \\ \sin(\pi x_1)\sin(\pi x_2) + \sum_{i=3}^{4} x_i, & \text{otherwise} \end{cases}$$

and

$$f_{12}(\mathbf{x}) = \begin{cases} \sum_{i=3}^{10} x_i, & \text{if } x_1 > 0.5 \text{ or } x_2 > 0.5, \\ \sin(\pi x_1)\sin(\pi x_2) + \sum_{i=3}^{10} x_i, & \text{otherwise} \end{cases}$$

where $\mathbf{x} = \{x_1, x_2, \ldots, x_{10}\}$.

The proposed SCAMR approach is implemented for these 2-D, 4-D and 10-D functions. The tolerance vector $\boldsymbol{\epsilon}$ is given by [1, 3e-1, 1e-1, 3e-2, 1e-2, 3e-3, 1e-3, 3e-4, 1e-4]. The function evaluation locations for 2-D function f_{10} are plotted in figure 3.7a, and the error analysis of the numerical approximation from SCAMR for f_{10}, f_{11} and f_{12} are provided in figures 3.7b, 3.7c and 3.7d. The numerical approximations are compared to those from ASGC method. From the results, similar conclusions to the previous example can be drawn.

3.4.1.4 SCAMR in a stochastic elliptic problem

Finally, we apply the SCAMR approach to a stochastic elliptic problem as in [84, 104]. The model problem is given as

$$\begin{aligned} -\nabla(a_n(\omega, x)\nabla u(\omega, x, y)) &= f(x, y), \text{in } D \times \Gamma \\ u(\omega, x, y) &= 0, \text{ on } \partial D \times \Gamma \end{aligned} \tag{3.36}$$

where spatial variable $(x, y) \in D = [0, 1]^2$, random variable $\omega \in \Gamma$, $f(x, y) = \cos(x)\sin(y)$.

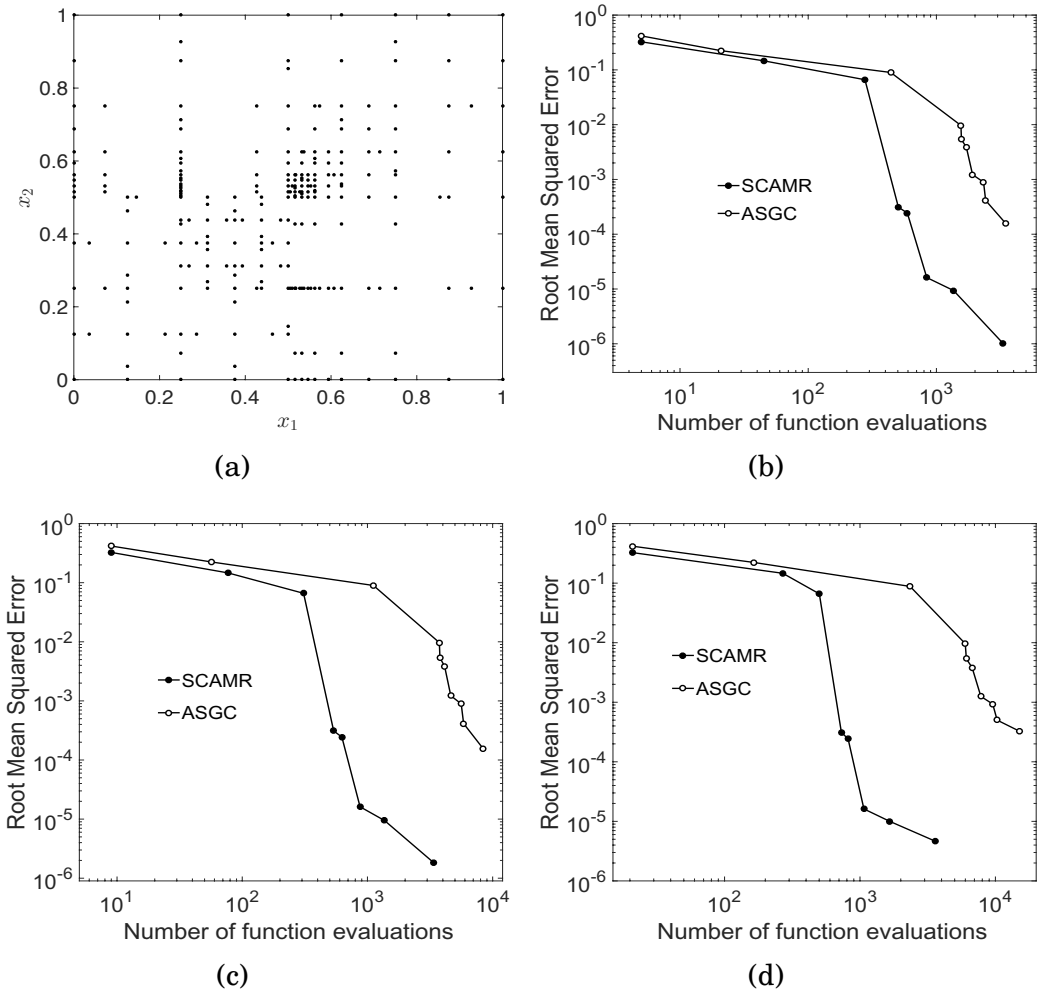


Figure 3.7: Input domain and error analysis for functions with discontinuty: (a) input domain for function f_{10}, (b) numerical error for 2D f_{10}, (c) numerical error for 4D f_{11}, and (d) numerical error for 10D f_{12}.

The diffusion coefficient $a_n(\omega, x)$ is assumed to be a random field that can be approximated in a finite n-dimensional stochastic domain as:

$$\log(a_n(\omega, x) - 0.5) = 1 + Y_1(\omega)(\frac{\sqrt{\pi}L}{2})^{1/2} + \sum_{i=2}^{n} \xi_i \phi_i(x) Y_i(\omega), \tag{3.37}$$

where $Y_i(\omega)$ $[i = 1, 2, \ldots, n]$ are independent random parameters which are uniformly distributed in $[-\sqrt{3}, \sqrt{3}]$, and

$$\xi_i = (\sqrt{\pi}L)^{1/2} \exp(\frac{-(\lfloor \frac{i}{2} \rfloor \pi L)^2}{8}), \text{ if } i > 1 \tag{3.38}$$

and

$$\phi_i(x) := \begin{cases} \sin(\frac{\lfloor \frac{i}{2} \rfloor \pi x}{L_p}), & \text{if i is even,} \\ \cos(\frac{\lfloor \frac{i}{2} \rfloor \pi x}{L_p}), & \text{if i is odd} \end{cases}$$

where $L_p = \max\{1, 2L_c\}$, and $L = \frac{L_c}{L_p}$ where $L_c = 0.5$ is the correlation length.

Without loss of generality, we consider the uncertainty in the output at a fixed point in space $x = y = 0.5$, which is the center of the spatial domain. Figure 3.8 displays two realizations of the output contour in the spatial domain for $n = 50$ using the deterministic code of the elliptic problem. The proposed SCAMR approach is implemented for the elliptic problem with varying dimensions n in the stochastic input domain. The error analysis of the numerical

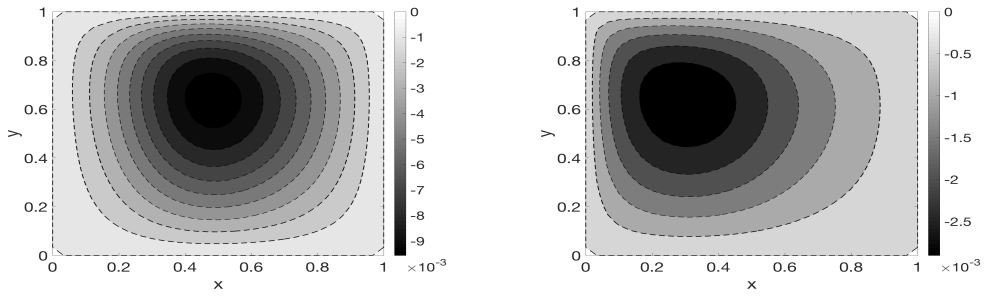


Figure 3.8: Two realizations of the output u for $n = 50$ and correlation length $L_c = 0.5$.

approximations are provided in figures 3.9(a-f) for $n = 2, 11, 25, 50, 75, 100$ respectively. The tolerance vector ϵ for $n = 2$ is given by [3e-3, 1e-3, 3e-4, 1e-4, 3e-5, 1e-5, 3e-6] while for the rest, it is given by [3e-3, 1e-3, 3e-4, 1e-4, 3e-5]. The numerical approximations from SCAMR are compared to those from the ASGC method. From the figure, one can observe that the numerical approximation from SCAMR converges faster for very low dimensions such as $n = 2$, achieves similar convergence for $n = 11$, and for large dimensions such as $n = 25, 50, 75, 100$, its efficiency increases with increase in dimensionality. The reason is that the tail terms of Eq. (3.37) for $n > 25$ could be negligible due to the fast decay of the eigenvalues ξ_i. As with the previous examples, SCAMR converges faster or at a similar rate as ASGC for this problem.

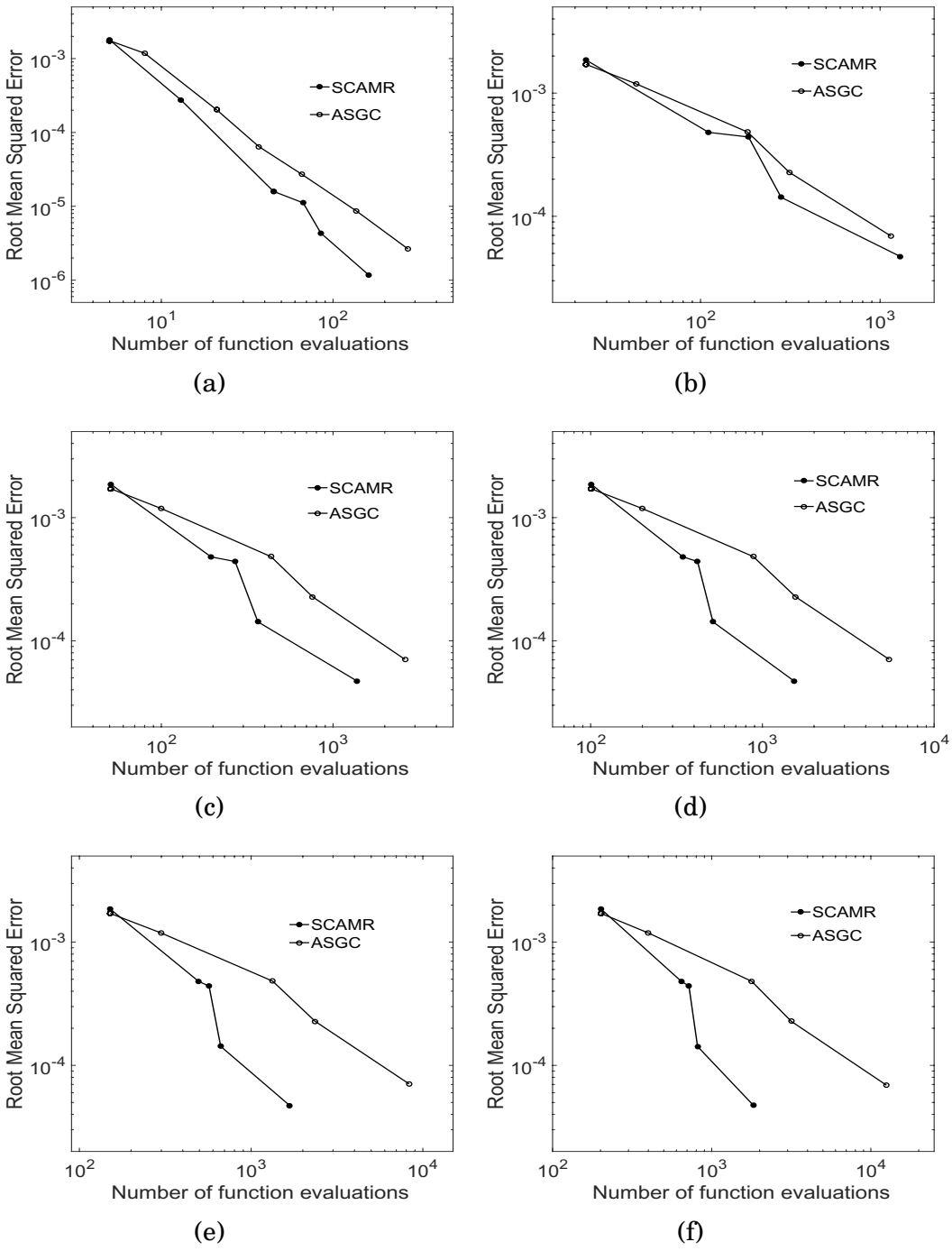


Figure 3.9: Error analysis of the stochastic elliptic problem with (a) $n = 2$, (b) $n = 11$, (c) $n = 25$, (d) $n = 50$, (e) $n = 75$, (f) $n = 100$ dimensions for correlation length $L_c = 0.5$.

3.4.2 Comparison to HDMR-guided algorithms for high dimensional problems

To further illustrate the efficiency of SCAMR regarding the model reduction criterion, we implement our proposed approach for more high-dimensional problems and compare the results to those from HDMR-ASGC and MEPCM-A methods.

A 10-dimensional function is considered to compare the efficiency of SCAMR and HDMR-ASGC [128]. The error estimate used here is the normalized L_2 interpolation error given by

$$\epsilon = \frac{\sqrt{\sum_{i=1}^{N}(f(\mathbf{x_i}) - \tilde{f}(\mathbf{x_i}))^2}}{\sqrt{\sum_{i=1}^{N} f(\mathbf{x_i})^2}}, \tag{3.39}$$

where f is the exact function, $\tilde{f}$ is the numerical approximation using HDMR-ASGC or SCAMR and $N = 10^6$ randomly generated samples.

A high dimensional integration problem is then used as an example to compare the SCAMR and the MEPCM-A methods. The error estimate used here is the mean relative error [122] given by

$$\epsilon = \frac{|I_{exact} - I_{approx}|}{I_{exact}} \tag{3.40}$$

where I_{exact} is the true mean of the problem and I_{approx} is the numerical approx-

imation of the mean using either MEPCM-A or SCAMR.

3.4.2.1 Comparison to HDMR-ASGC

We consider a 10-D function [128] given by

$$f_{13}(\boldsymbol{x}) \quad = \frac{1}{1+\sum_{i=1}^{10} \alpha_i x_i} \tag{3.41}$$

where $\alpha_i = 0.1/2^{i-1}$, random input $x_i = \sigma y_i$ and y_i are i.i.d. uniform random parameters in $\left[-\sqrt{3}, \sqrt{3} \right]$, $i \in \{1, 2, \ldots, 10\}$. σ is related to the standard deviation of the input and for this example, $\sigma = 2$. The weights drop drastically with increase in dimensions and hence the number of effective dimensions is low compared to 10 nominal dimensions. Table 5.1 shows a comparison of the normalized L_2 interpolation error and the number of points needed for the HDMR-ASGC and the SCAMR approach. It can be seen from the results that SCAMR proves to be more efficient than HDMR-ASGC in approximating f_{13}. The HDMR-ASGC results are read directly from figure 8 (right) in [128]. Identification of the low effective dimensions using the interaction check in the SCAMR approach is achieved at a lower computational cost compared to the corresponding check in HDMR-ASGC [128]. The subsequent surrogate construction of the lower-dimensional subproblems also requires lesser number of samples when using the low-order gPC approximation in SCAMR compared to

the linear basis interpolation in the HDMR-ASGC approach. For example, the number of points needed for an L_2 error of approximately 6×10^{-5} is around 1575 points in the case of HDMR-ASGC while the number of points needed for an L_2 error of 2.4038×10^{-5} using SCAMR is 293.

Table 3.1: HDMR-ASGC and SCAMR error and cost comparison for function f_{13}

HDMR-ASGC		SCAMR		
L_2 error	Points	ϵ	L_2 Error	Points
$\approx 9 \times 10^{-3}$	≈ 200	3×10^{-2}	1.2081×10^{-3}	85
$\approx 1 \times 10^{-3}$	≈ 700	3×10^{-3}	3.6996×10^{-4}	133
$\approx 1 \times 10^{-4}$	≈ 1144	1×10^{-4}	8.9142×10^{-5}	149
$\approx 6 \times 10^{-5}$	≈ 1575	3×10^{-5}	2.4038×10^{-5}	293

3.4.2.2 Comparison to MEPCM-A

We consider a discontinuous Genz function given by:

$$f_{14}(\mathbf{x}) = \begin{cases} 0, & \text{if } x_1 \geq 0.5 \text{ or } x_2 \geq 0.5, \\ \exp(\sum_{i=1}^{n} c_i x_i), & \text{otherwise} \end{cases}$$

where $c_i = e^{-35i/(n-1)}$. Using SCAMR, we evaluate the high dimensional integration $I_{approx} = \int \tilde{f}_{14}(\mathbf{x}) d\mathbf{x}$ where $\tilde{f}_{14}$ is the numerical approximation to f_{14}. The relative mean error is then calculated and compared with MEPCM-A results in table 6 given in [122] with different dimensions $n = 100$, 200, 300, 400, and 500. It can be seen from the form of function f_{14} that the importance of the param-

Table 3.2: MEPCM-A and SCAMR error and cost comparison for function f_{14}

	MEPCM-A			SCAMR		
n	(ν, μ)	Error	Points	ϵ	Error	Points
100	(1,1)	$O(1)$	103	5	2.1308	201
	(2,2)	0.0197	20,801	0.2	2.7×10^{-3}	2645
	(3,3)	0.0098	4,677,148	0.1	1.1×10^{-3}	3111
200	(1,1)	$O(1)$	203	8	2.3705	401
	(2,2)	0.067	81,601	2	0.0742	3459
	(3,3)	0.047	36,714,298	0.3	1.3×10^{-4}	9911
300	(1,1)	$O(1)$	303	25	2.5435	601
	(2,2)	0.12	182,401	10	0.1209	4905
	(3,3)	0.09	123,111,448	5	0.0180	8623
400	(1,1)	$O(1)$	403	120	2.6806	801
	(2,2)	0.22	323,201	45	0.1716	6173
	(3,3)	0.07	290,868,598	30	0.0509	9681
500	(1,1)	$O(1)$	503	500	2.7891	1001
	(2,2)	0.43	504,001	200	0.1887	8391
	(3,3)	0.21	566,985,748	150	0.0940	10,713

eters decrease exponentially with increase in dimensions. Thus this is an example where the function has a high nominal dimensionality but low effective dimensionality depending on the error tolerance. Table 3.2 shows a comparison of the mean relative error and the number of points needed for the MEPCM-A approach and the SCAMR approach. For the SCAMR approach, mean value extraction is performed by generating weighted Clenshaw-Curtis sparse grid points in each of the elements in each subproblem. Then local means are calculated for each subproblem by assigning weights to each element according to their hypervolume. Local means are finally combined together to get the global mean. It can be seen from the results that SCAMR proves to be very

efficient in identifying the low effective dimensions. In MEPCM-A, the effective dimensions depend on the parameter ν. Even though ν is chosen to be small ($\nu = 2\ or\ 3$), the number of terms in HDMR becomes very large for high nominal dimensions. SCAMR thus achieves much better precision with less number of points compared to the MEPCM-A approach. For example, for the 300-dimensional case, the number of points needed for a relative error of 0.09 is around 123 million points in the case of MEPCM-A while the number of points needed in SCAMR for a relative mean error of 0.018 is only 8623.

3.5 Conclusions

In this chapter, an efficient stochastic collocation method with adaptive mesh refinement has been proposed. Specifically, this approach utilizes generalized polynomial chaos as the basis and solves the gPC coefficient using the least squares method, which provides more flexibility on the number and locations of function evaluations. It also implements adaptive mesh refinement to track the discontinuities, and the adaptive criteria of the mesh refinement to check for abrupt variations in the output based on error measured from a low-order gPC. In addition, this approach uses a criterion to check possible dimensionality reduction and decomposes the original high-dimensional problem to a number of lower-dimensional subproblems, based on the HDMR method.

Therefore, for a specific problem, the highest dimensionality of subproblems which involve interacting parameters, can be obtained. The effectiveness of this method has been shown using different low and high dimensional, smooth and non-smooth examples. It is noticeable that this approach is particularly efficient for high nominal dimensional problems, like the stochastic elliptic problem with a large number of terms for the diffusivity coefficient, where a significant number of parameters can be less important (low effective dimensions) and thus non-interacting with other more important parameters. However, if the parameters are all coupled in their contribution towards the output of interest, then the efficiency of this method decreases with the increase in the dimensionality of the problem, especially when the response function is highly non-linear. This is because of the generation of a large number of high dimensional subdomains (subelements), where new input points are to be generated according to the sparse grid quadrature. When there is significant non-linearity, the subdomains (subelements) generally do not converge with the low-order gPC approximation and hence split up into further smaller domains.

Chapter 4

On the usefulness of gradient information in surrogate modeling: Application to uncertainty propagation in composite material models

In this chapter, the performance of non-gradient as well as gradient-enhanced versions of two different classes of surrogate modeling methods, polynomial least squares regression and kernel based radial basis function interpolation, are compared in the context of a composite mechanics problem. Sequential

space filling random designs are used for selecting the training points. The primary goal is to investigate whether additional gradient information obtained from the simulation model at a relatively small cost helps in generating surrogates of better quality compared to those obtained without any gradient information. It is found from the study that if the output and/or the output gradients are noisy, then the quality of the surrogate construction is similar for both the gradient enhanced and the non-gradient based surrogate models. However, if the output and the output gradients are accurate, the gradient-enhanced surrogate models consistently perform better than the non-gradient based surrogate models, indicating that the gradient information enhances the quality of the surrogates. Low dimensional analytical test functions are used to demonstrate this behavior. As an application problem, we consider a multi-fiber reinforced matrix composite simulation model with a different interfacial damage parameter assigned to each fiber/matrix interface. The focus here is to build a surrogate which efficiently describes the variation of the homogenized stress at a given input strain as a function of the interface damage parameters. The Interface-Enriched Generalized Finite Element Method (IGFEM) is used in this case to solve for the stress as well as the gradients of the stress with respect to the damage parameters. Thus the goal of this study is two-fold: 1) to compare the error convergence properties in surrogate modeling using different sequential random space-filled designs, with and without gradient informa-

tion; 2) to identify the circumstances in which additional gradient information is beneficial for surrogate modeling.

4.1 Introduction

Standard surrogate models make use of output values evaluated at a set of sampling points in the parameter space. For example, there are surrogate models which use structured sampling designs [85, 100, 112, 121] while others use unstructured designs [135–137]. The current work focuses on surrogate modeling for simulation models that provide both the outputs as well as the output sensitivities (gradients) with respect to the input parameters. Gradient information, when available at a relatively low cost, can help in making surrogate modeling very efficient [138], [139].

The first step in developing a surrogate function is the design of sampling points in the parameter space, at which the output and the output gradients are calculated. Some popular designs used in the literature are simple random samples [140], Latin Hypercube Sampling (LHS) [29] and Improved Hypercube Sampling (IHS) [141]. It is typically desirable to have a good space filling design, such that the samples cover most of the input space. For example, in a kernel-based surrogate modeling method like radial basis function interpolation, a good sampling design may help alleviate the ill-conditioning issue of the

kernel matrix that can result from highly irregular spacing (local scattering and clustering) of sampling points. Sobol quasi random sequences [142], [69], [143], Halton quasi random sequences [68], and Latin Hypercube Sampling (LHS) [64, 144] are a few examples of such designs. The current work focuses on sequential space filling designs, which are natural candidates for systematic surrogate modeling error convergence studies. These include Hierarchical Latin Hypercube Sampling (HLHS) [145, 146], Refined Latinized Stratified Sampling (RLSS) [147] and Scrambled Sobol (ScSo) [148] sequences.

There are a few gradient-informed surrogate methods based on unstructured sampling points, namely gradient-enhanced kriging [149], gradient-assisted radial basis function [29, 140], and gradient-enhanced radial basis function [150] among others, which have been used for efficient optimization in computational fluid dynamics (CFD) applications. An overview of the different existing surrogate models that use output gradients in addition to output values can be found in [141]. Very little, if any, work is available extending these approaches to solid mechanics applications.

In this chapter, an Interface-Enriched Generalized Finite Element Method (IGFEM) based fiber-reinforced composite model is considered in which mechanical responses and their gradients with respect to the input parameters are calculated efficiently [151, 152]. It is of interest to see how this gradient information affects the accuracy and efficiency of surrogate modeling. The study

considers three different space-filling designs (HLHS, RLSS and ScSo) and four surrogate modeling methods: least squares (LS) regression, gradient-enhanced least squares (GLS) regression, radial basis function (RBF) interpolation and gradient-enhanced radial basis function (GRBF) interpolation. In LS regression and RBF interpolation, only output values are used to build the surrogate. In GLS regression and GRBF interpolation, both output and output gradients are used for surrogate construction. The chapter is divided into the following sections. The HLHS, RLSS and ScSo designs are described briefly in section 4.2. Section 4.3 discusses the surrogate modeling approaches. In section 4.4, 1-dimensional and 2-dimensional analytical test functions are considered and the relative performance of the different sampling designs and surrogate methods are compared in order to draw some general conclusions. In section 4.5, damage models of fiber reinforced composites with variation in spatial arrangement as well as the number of fibers are considered for a similar performance comparison as in section 4.4. Section 4.6 concludes with a summary of the overall observations and related discussion.

4.2 Sequential space filled designs

Optimal sequential sampling designs should maintain space-filling and non-collapsible characteristics as each individual sampling point is added to the

population. Space-filling suggests that the parameter space is covered in a relatively even fashion, and non-collapsibility indicates that lower dimensional projections of the samples do not show duplication. Three classes of sampling designs are considered here: latin hypercube sampling (LHS), stratified sampling (SS), and Sobol sequences.

Latin Hypercube Sampling (LHS) [64] is a popular design used for sample generation in various application because of its ease of implementation. LHS ensures that the samples do not collapse on each other when projected to lower dimensions; however it does not guarantee good space filling properties. Subsequently, a lot of work has been done on the addition of secondary criterion to the LHS design in order to improve its space filling properties [153–156]. Improved Latin Hypercube Sampling (IHS) design [157] improves upon the LHS design by choosing a more optimal uniform spacing of points using a duplication factor.

Stratified sampling (SS) design, on the other hand, has good space filling properties but can generate collapsive samples [64]. In SS design, equisized strata are formed all at once and samples are added randomly in each of them. Latinized stratified sampling (LSS) [158] design attempts to combine the advantages of the LHS and SS designs in order to achieve good non-collapsive as well as space filling properties. In LSS, dimension-wise stratifications (as in LHS design) as well as full-dimensional stratifications (as in SS design) are

performed simultaneously.

It is noted that the traditional LHS and SS algorithms are not designed for addition of new sample points to an existing set of sampled points, already generated from the design algorithms. Therefore, a very attractive property of some designs is to be sequentially space filling, where the designs try to maintain space-filling with each new addition of a single sampling point to the population.

Hierarchical Latin Hypercube Sampling (HLHS) design [145, 146] is a sequential version of LHS design, in which new samples are added by gradual refinements of the dimension-wise stratifications. HLHS is not a purely sequential design and at least doubles the current sample size at each extension because the minimum refinement that can be performed is subdividing each dimension-wise stratum into two sub-strata. The Refined stratified sampling (RSS) design [159] enables sequential sampling in the SS design, by selecting a stratum for each new sample one at a time. The sample size extension version of LSS design, Hierarchical Latinized Stratified Sampling (HLSS) design [147], involves dimension-wise stratification refinements, as in HLHS, as well as subdivision of the SS-type strata in an existing LSS design and then sampling in the permissible regions. In Refined Latinized Stratified Sampling (RLSS) [147], a sequential version of HLSS, strata subdivision is performed one at a time and a new sample is added in the permissible region.

Sobol sequences are deterministic sequences with low discrepancy measures. A Sobol sequence generation is dependent on the choice of a set of pre-determined initialization numbers. For intermediate dimensions, Sobol sequences produce wide gaps in some of the dimensions resulting in poor projective properties [70]. Scrambled Sobol (ScSo) design algorithms alleviate this issue to some extent by using scrambling algorithms, like Owens method [71], thus improving uniformity of samples. The scrambling procedure makes the scrambled Sobol sequences random in nature.

Considering the benefits and drawbacks of each approach, this chapter focuses on 3 sequential space-filling designs: HLHS, RLSS, and ScSo.

4.3 Surrogate modeling methods

Given a set of samples based on the sampling designs discussed in the previous section, there are a number of methods for building continuous surrogates. For the randomly spaced sample points considered here, two of the most common methods are radial basis function interpolation and least squares regression. Both methods are capable of building a surrogate based on only the outputs calculated at the sampling points, or based on the outputs as well as the output gradients calculated at the sampling points. The two methods are discussed in more details in this section.

4.3.1 Radial basis function interpolation

Radial basis function (RBF) interpolation [160, 161] is a kernel-based approximation technique for scattered data and takes the form:

$$\hat{f}(\boldsymbol{z}) = \sum_{i=1}^{N} c_i \phi(||\boldsymbol{z} - \boldsymbol{z}^i||) \tag{4.1}$$

where $\phi(||\boldsymbol{z} - \boldsymbol{z}^i||)$ is a radial basis function, the norm $||.||$ is Euclidean, N is the number of training points, $\boldsymbol{z}^i$ $[i = 1, \ldots, N]$ are the training points and c_i are the set of coefficients. Enforcing the following interpolation conditions,

$$\hat{f}(\boldsymbol{z}^i) = f(\boldsymbol{z}^i), \qquad i = 1, 2, ..., N \tag{4.2}$$

produces the system of linear equations:

$$\boldsymbol{K}\boldsymbol{c} = \boldsymbol{f} \tag{4.3}$$

where $\boldsymbol{K}$ is a $N \times N$ kernel matrix, $\boldsymbol{c}$ is a $N \times 1$ coefficient vector and $\boldsymbol{f}$ is a $N \times 1$ vector containing the output values at the N training points.

The gradient-enhanced radial basis function (GRBF) interpolation [29] takes the form:

$$\hat{f}(\boldsymbol{z}) = \sum_{i=1}^{N} \beta_i \phi(||\boldsymbol{z} - \boldsymbol{z}^i||) + \sum_{i=1}^{N} \sum_{j=1}^{d} \tilde{\beta}_{ij} \frac{\partial \phi}{\partial z_j}(||\boldsymbol{z} - \boldsymbol{z}^i||) \tag{4.4}$$

where $\phi(||\boldsymbol{z} - \boldsymbol{z}^i||)$ is a radial basis function, d is the number of input variables and β_i and $\tilde{\beta}_{ij}$ are a set of coefficients to be determined. Using the function values and the derivative values at the training samples, the following conditions are imposed to calculate the coefficients:

$$\hat{f}(\boldsymbol{z}^i) = f(\boldsymbol{z}^i), \qquad i = 1, 2, ..., N \tag{4.5}$$

$$\nabla\hat{f}(\boldsymbol{z}^i) = \nabla f(\boldsymbol{z}^i), \qquad i = 1, 2, ..., N \tag{4.6}$$

To use Eq. (4.6), Eq. (4.4) is differentiated with respect to z_k to obtain

$$\begin{aligned}\frac{\partial\hat{f}(\boldsymbol{z})}{\partial z_k} = &\sum_{i=1}^{N}\beta_i\frac{\partial}{\partial z_k}\phi(||\boldsymbol{z} - \boldsymbol{z}^i||) \\ &+ \sum_{i=1}^{N}\sum_{j=1}^{d}\tilde{\beta}_{ij}\frac{\partial}{\partial z_k}\frac{\partial\phi}{\partial z_j}(||\boldsymbol{z} - \boldsymbol{z}^i||)\end{aligned} \tag{4.7}$$

Using Eq. (4.5) and Eq. (4.6), the following linear system is obtained:

$$\boldsymbol{A}\boldsymbol{\beta} = \boldsymbol{y} \tag{4.8}$$

where $\boldsymbol{A}$ is a $N(d+1) \times N(d+1)$ kernel matrix, $\boldsymbol{\beta}$ is the $N(d+1) \times 1$ coefficient vector and $\boldsymbol{y}$ is the $N(d+1) \times 1$ vector containing the function values and the gradient values at the training points. It is noted that the radial basis

function ϕ should be at least twice differentiable for the above formulation to work. Unlike the RBF interpolation, the number of linear equations depends on the number of training points N as well as the number of input parameters d. Thus the size of the gradient informed kernel matrix can become very large and hence computationally expensive to invert with a large d and even with a moderate N. This study uses isotropic Gaussian radial basis functions (motivated by the study in appendix B) which are infinitely differentiable and given by:

$$\phi(||\boldsymbol{z}^k - \boldsymbol{z}^j||) = exp(-\frac{||\boldsymbol{z}^k - \boldsymbol{z}^j||_2^2}{2\sigma^2}) \tag{4.9}$$

where $||.||_2$ is the Euclidean norm and σ is a hyperparameter to be estimated.

In RBF interpolation, the hyperparameter σ of the chosen kernel is an unknown internal parameter, which is estimated by minimizing the leave one out cross validation (LOOCV) error [162, 163] over the training samples, given by:

$$LOOCV(\sigma) = \frac{1}{N}\sum_{i=1}^{N}(\hat{f}_{-i}(\boldsymbol{z}^i) - \hat{f}(\boldsymbol{z}^i))^2 \tag{4.10}$$

where $\hat{f}_{-i}(\boldsymbol{z}^i)$ is the RBF approximation at point $\boldsymbol{z}^i$ using responses at $N-1$ training points and not considering the response at that sample point $\boldsymbol{z}^i$. The criterion used for GRBF hyperparameter estimation is an extended version of

the above criterion [164] and is given by:

$$LOOCV_g(\sigma) = \frac{1}{N(d+1)} \sum_{i=1}^{N} \Bigg[(\hat{f}_{-i}(\boldsymbol{z}^i) - \hat{f}(\boldsymbol{z}^i))^2 + \sum_{j=1}^{d} \Big(\frac{\partial \hat{f}_{-i,j}(\boldsymbol{z}^i)}{\partial z_k} - \frac{\partial f(\boldsymbol{z}^i)}{\partial z_k} \Big)^2 \Bigg] \tag{4.11}$$

where $\frac{\partial \hat{f}_{-i,j}(\boldsymbol{z}^i)}{\partial z_k}$ is the GRBF approximation of the gradient along the k-th dimension at point $\boldsymbol{z}^i$ provided by the metamodel built without considering the true k-th component of the gradient at point $\boldsymbol{z}^i$. A particle swarm optimization (PSO) algorithm [165] was used for the hyperparameter estimation.

4.3.2 Polynomial regression metamodels

Least Squares (LS) regression is a popular function approximation method where a linear relationship is assumed between the output and the unknown regression coefficients:

$$\boldsymbol{F}\boldsymbol{\beta} = \boldsymbol{y} \tag{4.12}$$

where $\boldsymbol{y}$ is the vector of the output, $\boldsymbol{\beta}$ is the vector of regression coefficients and the matrix $\boldsymbol{F}$ is a function of the input variables and essentially defines the relationship (linear/non-linear) between the output and the input variables which depends on the polynomial order p. $\boldsymbol{F}$ is a $N \times N_t$ matrix, $\boldsymbol{\beta}$ is a $N_t \times 1$ vector and $\boldsymbol{y}$ is a $N \times 1$ vector containing the output values, where $N_t = \binom{p+d}{p}$,

d is the number of input variables, p is the polynomial order and N is the number of training samples. The unknown coefficient vector $\boldsymbol{\beta}$ is estimated using the ordinary least squares (OLS) approach by minimizing the following mean squared error function:

$$MSE(\beta) = ||(\boldsymbol{F}\boldsymbol{\beta} - \boldsymbol{y})||_2^2 \tag{4.13}$$

which gives,

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{F}^T\boldsymbol{F})^{-1}\boldsymbol{F}^T\boldsymbol{y} \tag{4.14}$$

In the gradient-enhanced version of least squares (GLS) regression, the overall model is also assumed to be linear between the output and the regression coefficients:

$$\boldsymbol{F}_g\boldsymbol{\beta}_g = \boldsymbol{y}_g \tag{4.15}$$

where $\boldsymbol{F}_g$ is a $N(d+1) \times N_t$ matrix, $\boldsymbol{\beta}_g$ is a $N_t \times 1$ vector and $\boldsymbol{y}_g$ is a $N(d+1) \times 1$ vector containing the output and the gradient values. The above model now contains additional gradient information which increases the size of the vectors and matrix depending on the number of input variables d. Similar to the non-gradient version, the regression coefficient vector $\boldsymbol{\beta}_g$ is estimated as:

$$\hat{\boldsymbol{\beta}}_g = (\boldsymbol{F}_g^T\boldsymbol{F}_g)^{-1}\boldsymbol{F}_g^T\boldsymbol{y}_g \tag{4.16}$$

In LS regression, to avoid ill conditioning of matrix $\boldsymbol{F}^T\boldsymbol{F}$ in Eq. (4.14), $N \geq N_t$. Thus for sufficiently small number of samples, the polynomial degree should be low enough to get a stable solution. On the other hand, in GLS regression, ill conditioning of matrix $\boldsymbol{F}_g^T\boldsymbol{F}_g$ in Eq. (4.16) can be avoided if $N(d+1) \geq N_t$. Thus the additional gradient information, especially with higher number of input variables d, can help in achieving stable solutions with higher polynomial order and smaller number of samples compared to the non-gradient LS regression. For example, in a 2-dimensional example $(d = 2)$, for sample size $N = 5$ with polynomial order $p = 2$ $(N_t = 6)$, the LS method becomes unstable due to lack of samples because $N < N_t$. On the other hand, the additional gradient information in the GLS regression makes the solution stable because $N(d+1) > N_t$.

In this study, the open source matlab/octave GRENAT Toolbox [166] was used to implement the surrogate modeling methods.

4.4 Numerical results

In this section, 1-dimensional and 2-dimensional analytical test functions are used to demonstrate the performance of surrogate modeling using different combinations of the training design and the surrogate modeling method. The performance of each case was measured using the Root Mean Squared error

(RMSE) given by:

$$RMSE = \sqrt{\frac{1}{n_t}\sum_{i=1}^{n_t}\left(y_{true}^{(i)} - y_{predicted}^{(i)}\right)^2} \tag{4.17}$$

where n_t is the total number of test samples, $\boldsymbol{y}_{true} = [y_{true}^{(1)}, \ldots, y_{true}^{(n_t)}]$ is the vector of the true function values at the n_t points and $\boldsymbol{y}_{predicted} = [y_{predicted}^{(1)}, \ldots, y_{predicted}^{(n_t)}]$ is the vector of predicted values from the surrogate construction at the same n_t points.

4.4.1 1-dimensional results

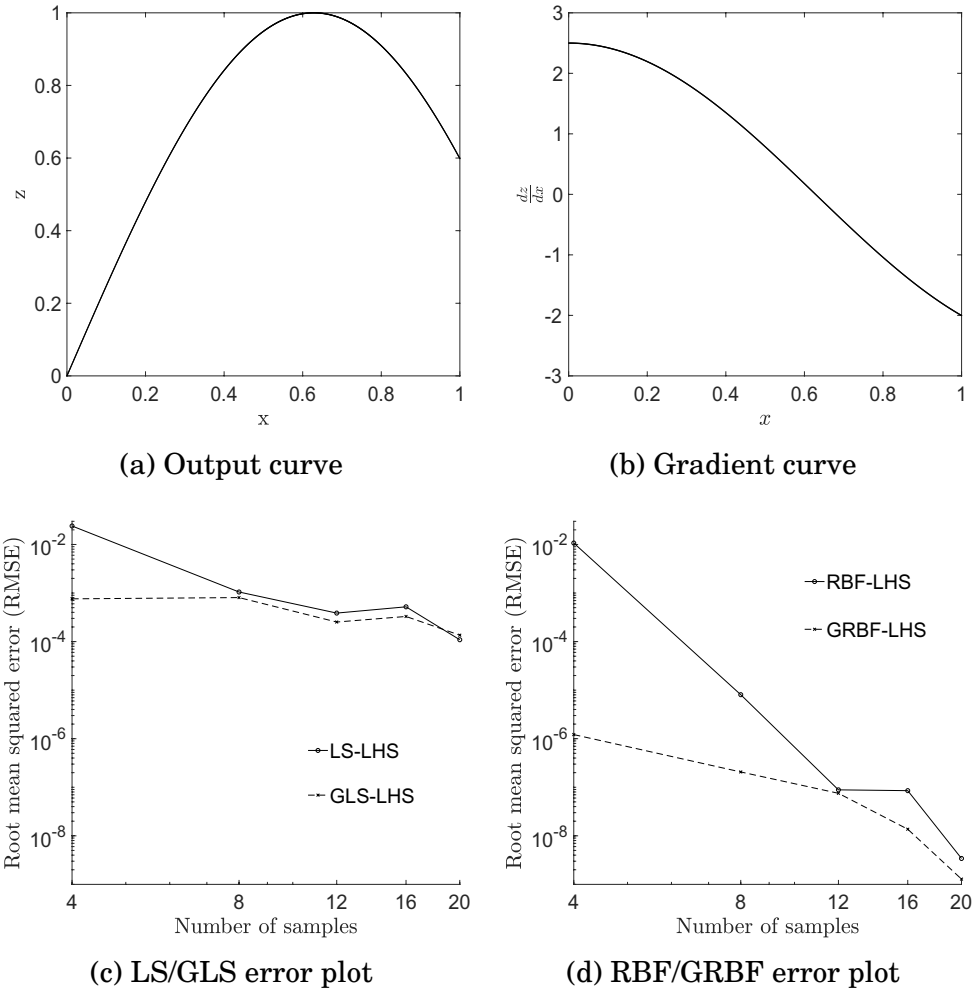
In this section, we consider a 1-dimensional smooth sine function as shown in figure 4.1a given by:

$$z = \sin(5x/2) \tag{4.18}$$

and its gradient (shown in figure 4.1b) is:

$$\frac{dz}{dx} = \frac{5}{2}\cos(5x/2) \tag{4.19}$$

Plots of the RMSE as a function of sample size (figures 4.1c and 4.1d) show that incorporating gradient information helps to achieve faster convergence when there is no noise in either the output or the gradient values. These plots also indicate that for this particular function the RBF approaches converge faster



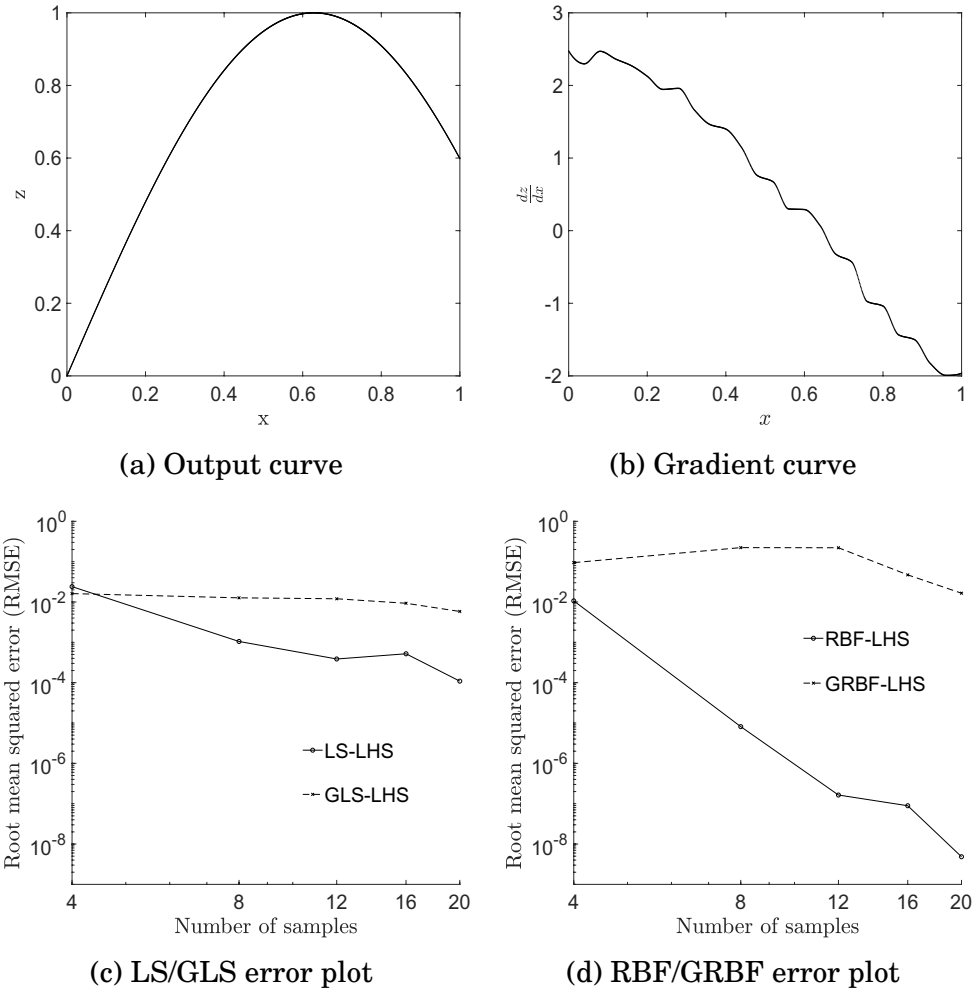
(a) Output curve

(b) Gradient curve

(c) LS/GLS error plot

(d) RBF/GRBF error plot

Figure 4.1: 1-dimensional smooth sine function (a) and gradient (b). The corresponding RMSE as a function of sample size for LS/GLS surrogates (c) and for RBF/GRBF surrogates (d) show improved convergence when incorporating gradient information.



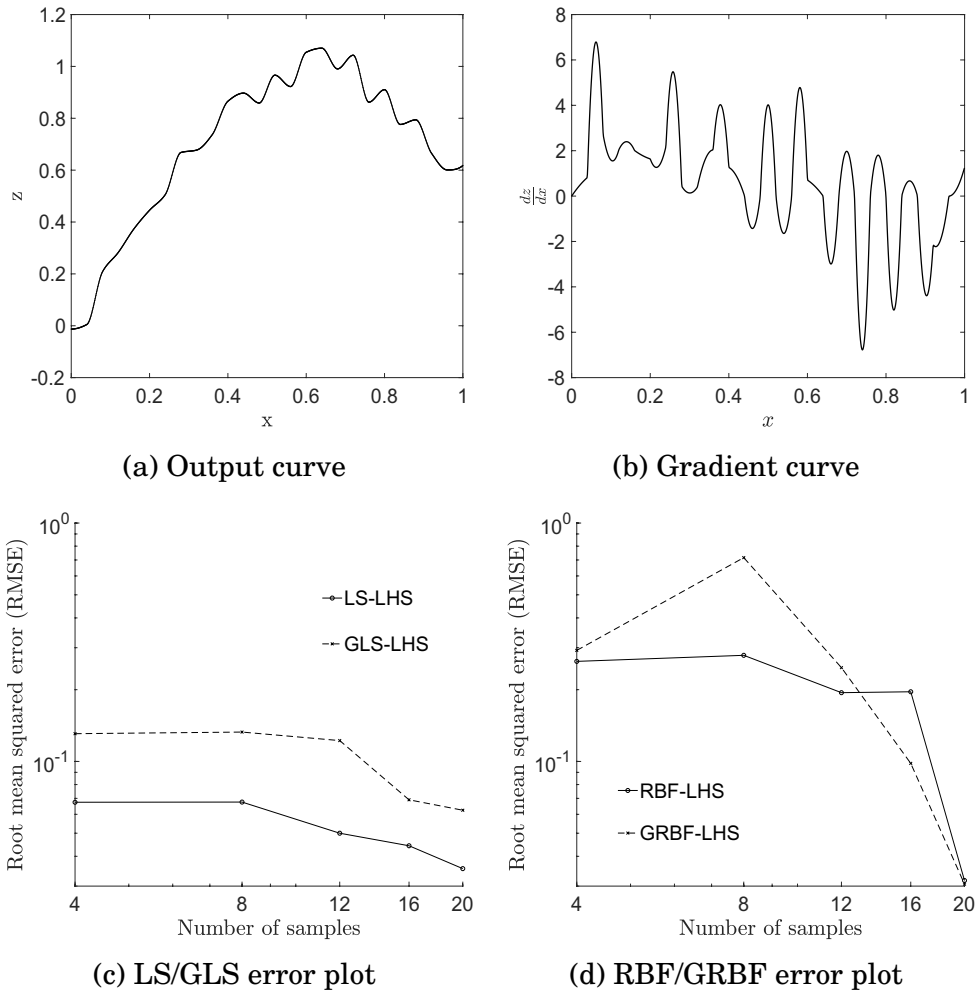
(a) Output curve

(b) Gradient curve

(c) LS/GLS error plot

(d) RBF/GRBF error plot

Figure 4.2: 1-dimensional smooth sine function (a) with noisy gradient (b). The corresponding RMSE as a function of sample size for LS/GLS surrogates (c) and for RBF/GRBF surrogates (d) show worse convergence when incorporating inaccurate gradient information.



(a) Output curve

(b) Gradient curve

(c) LS/GLS error plot

(d) RBF/GRBF error plot

Figure 4.3: 1-dimensional noisy sine function (a) with associated noisy gradient (b). The corresponding RMSE as a function of sample size for LS/GLS surrogates (c) and for RBF/GRBF surrogates (d) show poorer convergence in all surrogates than the cases without noise. Incorporating gradients based on inaccurate function/gradient calculations usually worsens the convergence.

than the LS approaches. Note that all results were generated using the LHS sampling design. Comparison to other sampling designs will be performed in subsequent examples.

The effect of random noise on these results is addressed by considering two cases: the output values are smooth but the output gradients are noisy as shown in figure 4.2(a,b) and in the other case, the output is noisy, leading to a noisy gradient, as shown in figure 4.3(a,b). Figure 4.2(c,d) shows RMSE as a function of sample size for the case in which there is noise in the gradient only. When the gradient is noisy, the gradient based surrogate degrades in quality and performs much worse than the non-gradient surrogate as seen in figure 4.2c. Figure 4.3(c,d) shows RMSE vs. sample size for the case in which there is noise in both the output and the gradient. When both the output and the gradient are noisy, the gradient-based approaches still exhibit worse performance than the non gradient-based approaches.

4.4.2 2-dimensional results

In this section, two 2-dimensional test functions, the quadratic function and the Michalewicz function [167], are considered and the performance of the designs and the surrogate models on these functions are compared. The quadratic function is given by:

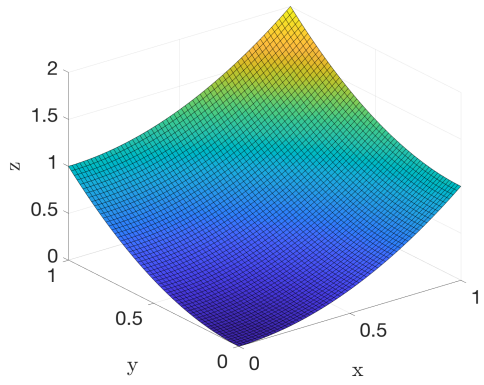
$$z = x^2 + y^2, \quad 0 \leq x, y \leq 1 \tag{4.20}$$

and the Michalewicz function is given by:

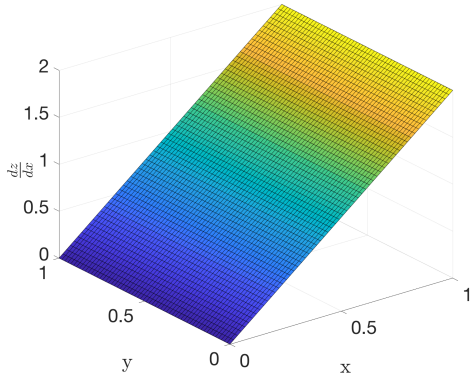
$$z = -\sin(x)\Big[\sin\Big(\frac{x^2}{\pi}\Big)\Big]^{20} - \sin(y)\Big[\sin\Big(\frac{2y^2}{\pi}\Big)\Big]^{20},$$
$$1.5 \leq x \leq 2.5, \quad 1 \leq y \leq 2 \tag{4.21}$$

where x and y are the input variables. The function was evaluated at the sample points generated by the HLHS, RLSS and ScSo designs and then the four surrogate modeling methods (LS, GLS, RBF and GRBF) were applied to the data set to predict the function values at 60×60 $(= 3600)$ test grid points in the problem domain. Thus, for this case, $n_t = 3600$ in calculating the RMSE from Equation (4.17). For each of the 3 sampling designs, 20 independent sets of samples are generated, in order to account for statistical variations from one sample set to another. For each set, the metamodeling procedure is conducted and the RMSE value is calculated. The estimated median RMSE values from the set of 20 values are reported here.

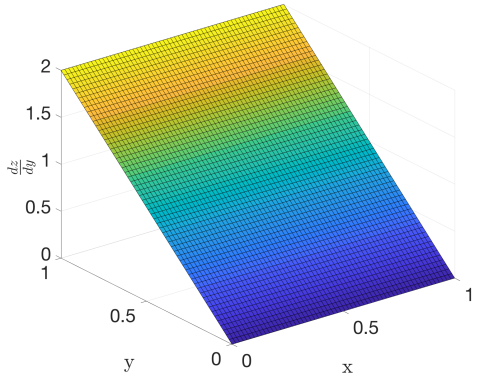
Figure 4.4a shows the quadratic function and figures 4.4b and 4.4c show the gradient surfaces w.r.t x and y respectively. Figure 4.4d shows the RMSE convergence plot comparison between the LS and GLS regression models using the three different designs. A polynomial of order 2 is used and although the GLS methods in general seem to perform slightly better than the LS methods especially in the smaller sample size regime, the RMSE values for all the cases



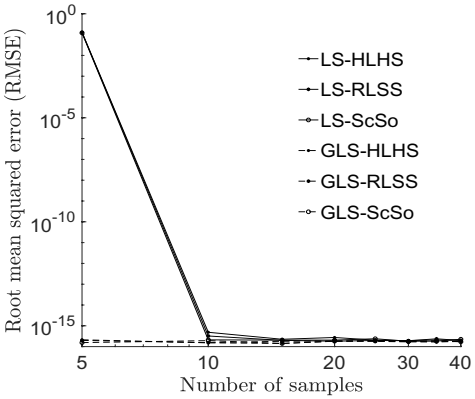
(a) True output surface



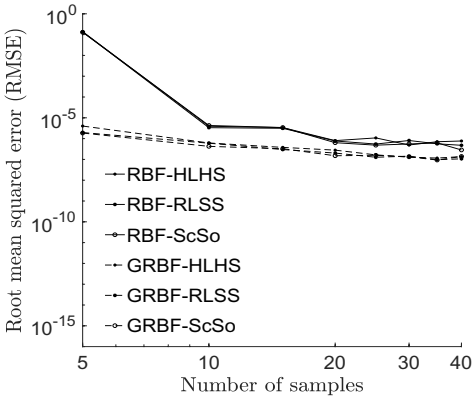
(b) True gradient (w.r.t x) surface



(c) True gradient (w.r.t y) surface

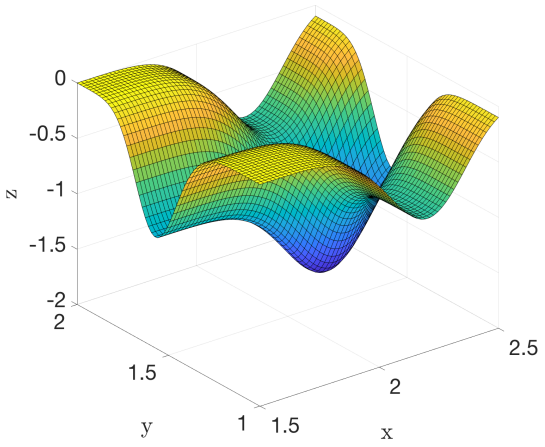


(d) LS/GLS error plot

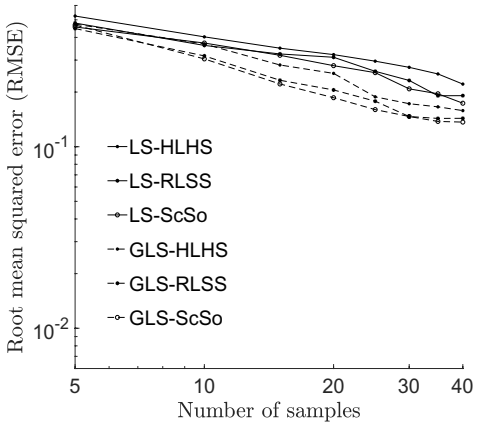


(e) RBF/GRBF error plot

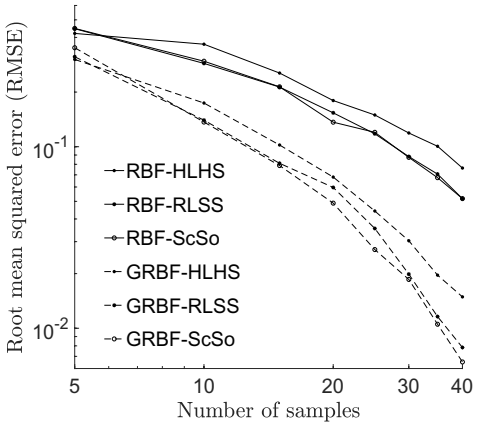
Figure 4.4: (a) 2-dimensional smooth quadratic function; (b) function gradient w.r.t x; (c) function gradient w.r.t y; RMSE plots as a function of sample size for (d) LS/GLS and (e) RBF/GRBF surrogate constructions; Incorporation of gradient information improves convergence in both cases.



(a) True output surface



(b) LS/GLS error plot



(c) RBF/GRBF error plot

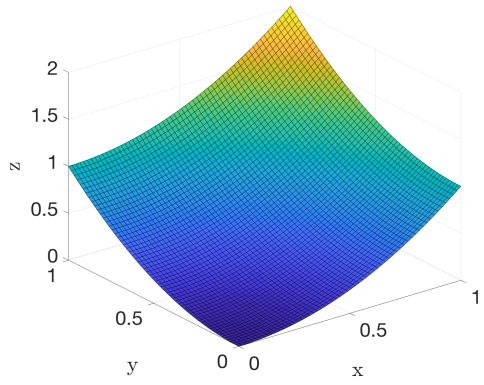
Figure 4.5: (a) 2-dimensional Michalewicz function, with RMSE as a function of sample size for (b) LS/GLS and (c) RBF/GRBF surrogate constructions. Incorporation of gradient information improves convergence in both cases.

are very close to the machine double precision value of $1e-16$ (on a 32-bit computer) which is expected, given that we are using a quadratic fit to a quadratic function. Figure 4.4e shows the RMSE convergence plot comparison between the RBF and GRBF methods using the three different designs. The advantage of having gradient information is clear from the faster convergence in RMSE of the GRBF method with increase in number of training samples. In this particular case, the regression methods produce much more accurate results than the RBF methods.

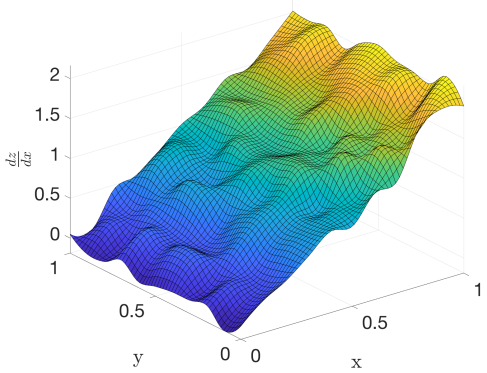
Figure 4.5a shows the Michalewicz function while figure 4.5b shows the RMSE convergence plot comparison between the LS and GLS methods using the three different sampling designs. Figure 4.5c shows the RMSE convergence plot comparison between the RBF and GRBF methods. From figures 4.5b and 4.5c, it is seen that the gradient-enhanced methods are found to converge faster than the corresponding non-gradient approaches for each of the three designs. Also, the RBF methods converge faster than the LS regression methods.

Comparing results from the two different functions, the improvement in RMSE convergence provided by the gradients for the Michalewicz function is more significant. This might be attributed to the fact that the quadratic function is much smoother than the Michalewicz function, so that there is little advantage to knowing gradients at multiple points in the parameter space.

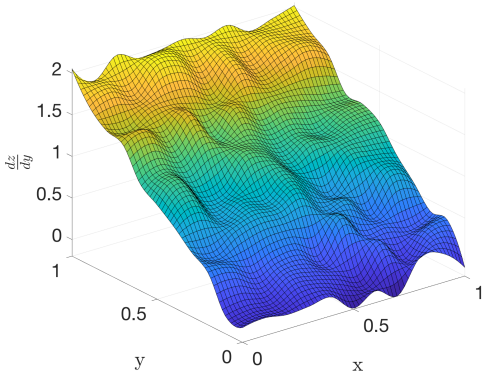
Two more variants of the quadratic function are also considered. In one



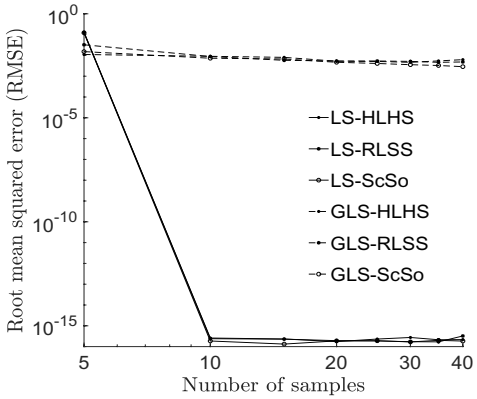
(a) True output surface



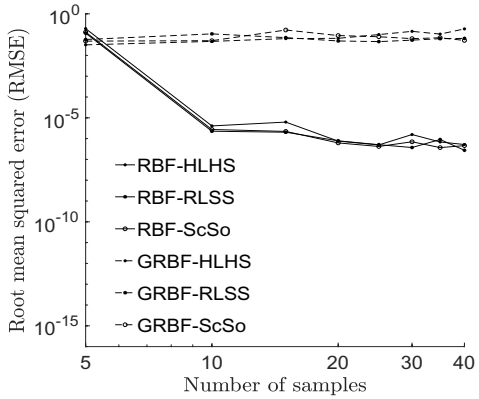
(b) True gradient (w.r.t x) surface



(c) True gradient (w.r.t y) surface

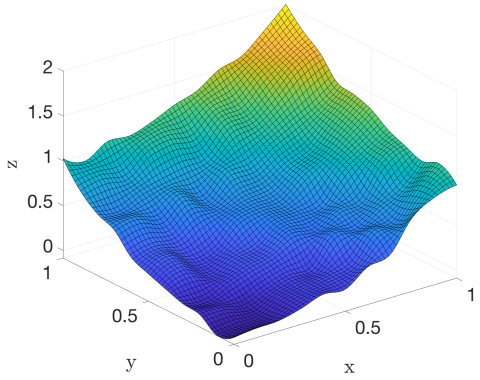


(d) LS/GLS error plot

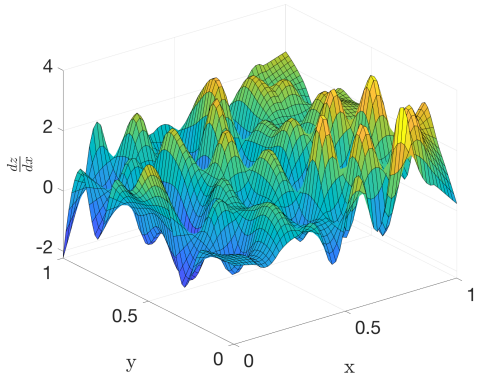


(e) RBF/GRBF error plot

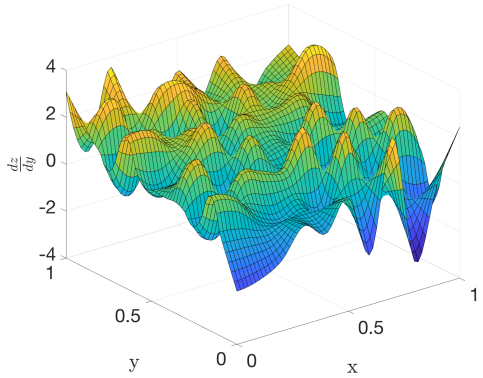
Figure 4.6: (a) 2-dimensional smooth quadratic function; (b) noisy function gradient w.r.t x; (c) noisy function gradient w.r.t y; RMSE plots as a function of sample size for (d) LS/GLS and (e) RBF/GRBF surrogate constructions; Incorporation of gradient information degrades convergence in both cases.



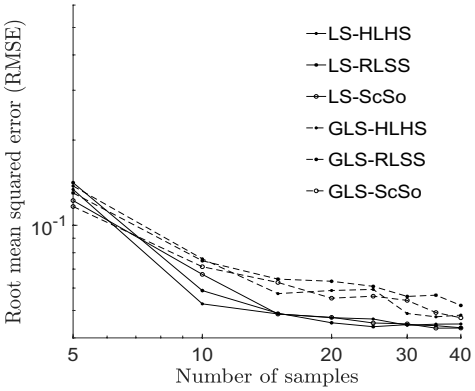
(a) True output surface



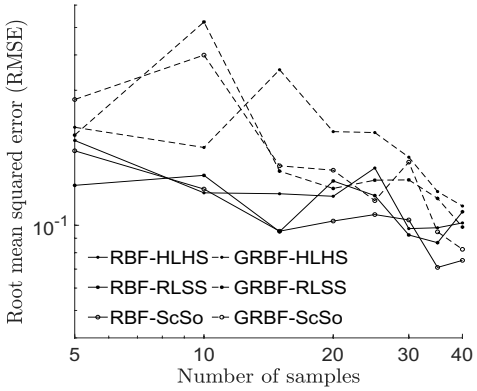
(b) True gradient (w.r.t x) surface



(c) True gradient (w.r.t y) surface



(d) LS/GLS error plot



(e) RBF/GRBF error plot

Figure 4.7: (a) 2-dimensional noisy quadratic function; (b) noisy function gradient w.r.t x; (c) noisy function gradient w.r.t y; RMSE plots as a function of sample size for (d) LS/GLS and (e) RBF/GRBF surrogate constructions.

case, the function is smooth but the gradient is noisy as shown in figure 4.6 and in the other case, the function is itself noisy as shown in figure 4.7. Figures 4.6d and 4.6e shows the RMSE convergence plot comparison between the LS/GLS methods and RBF/GRBF methods respectively for the noisy gradient case while figures 4.7d and 4.7e shows the RMSE convergence plot comparison between the LS/GLS methods and RBF/GRBF methods respectively for the noisy output case. It is found that when the function is smooth but the gradient information is noisy, the gradient based surrogates in both cases degrade in quality and performs much worse than the non-gradient surrogates as seen in Figs. 4.6d and 4.6e. It is seen from Figs. 4.7d and 4.7e where the function is itself noisy, the gradient and the non-gradient versions both perform poorly. However, the regression models seems to better handle the noise in the function and produce smaller RMSE values compared to the kernel based RBF methods. From the above results (ignoring the noisy cases), it is also found that the RLSS and ScSo designs when used with any of the surrogate methods (LS/GLS/RBF/GRBF) lead to similar or slightly more accurate results than using HLHS design. Thus for all future examples (except 1-dimensional cases), only the RLSS and ScSo designs are considered.

4.5 Composite model application

The application considered here is a Representative Volume Element (RVE) simulation model of a transverse section of a fiber-reinforced composite with varying numbers of fibers. The carbon fibers $(E = 19.5\ GPa,\ \nu = 0.45)$ are embedded in an epoxy matrix $(E = 2.38\ GPa,\ \nu = 0.43)$ and all bulk material is assumed to be linear elastic. Along each fiber/matrix interface, debonding is captured using an exponential cohesive zone model (CZM) [168] where the maximum cohesive traction σ_c and the characteristic opening displacement δ_c are the associated model parameters. Linear unloading toward the origin is added to the cohesive model to avoid unrealistic healing of the interfaces. The RVE is subjected to a prescribed uniaxial transverse strain of 2% in the x-direction by applying Dirichlet boundary conditions fixing the x-displacement of the left and right sides as shown in Figure 4.8a. The simulation model output is the homogenized Cauchy stress $\bar{\sigma}_{xx}$ over the entire RVE at the end of the strain loading.

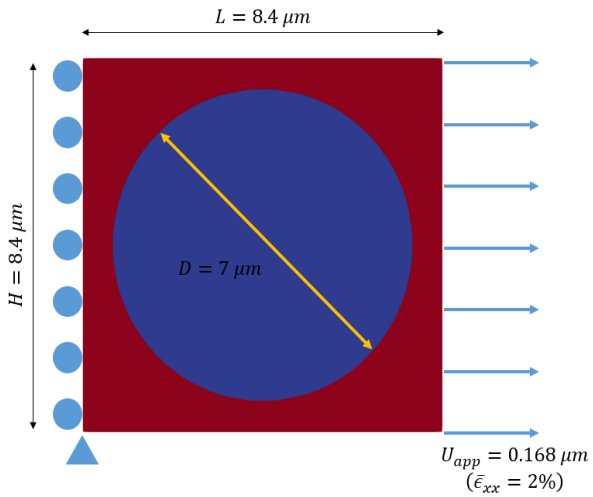
An Interface-Enriched Generalized Finite Element Method (IGFEM) [151, 152] is used to solve the structural RVE problem and obtain the homogenized transverse stress. Instead of using meshes that conform to the fiber/matrix interfaces as in conventional FEM [169], IGFEM uses a nonconformal mesh and captures the discontinuities of the solution field at these interfaces using

C^{-1} continuous enrichment functions in the finite elements that span them. An analytic material sensitivity analysis developed in [170] is used to obtain the sensitivities of the output with respect to the cohesive model parameters of each interface, concurrently with the nonlinear simulation and at very little extra computational cost.

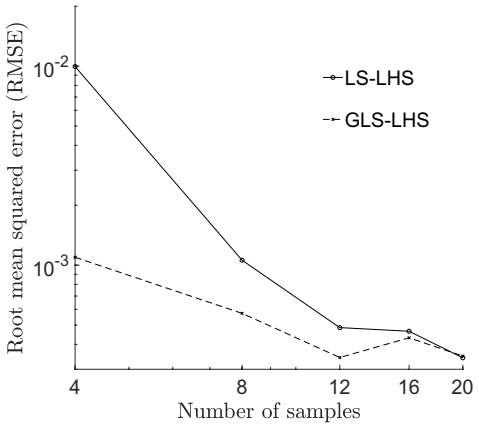
The goal is to efficiently build a response surface of the average stress at the final applied strain with respect to the interfacial strength parameters σ_{c_i} of each of the fibers. Evaluation of the simulation model for a certain combination of the input parameters involves a complex nonlinear finite element solution of an RVE. Thus an efficient production of the response surface with as few model evaluations as possible would be highly advantageous. In this study, RVE models with $1,\ 2,\ 5,\ 9,\ 20$ and 50 fibers are considered. For each of these models, the fiber/matrix interfaces have characteristic opening displacements of $\delta_{c_i} = 100\ nm$. In all models, a 2% strain is applied in the horizontal direction and the average stress corresponding at the 2% strain level is the model output in our study.

4.5.1 1-fiber model

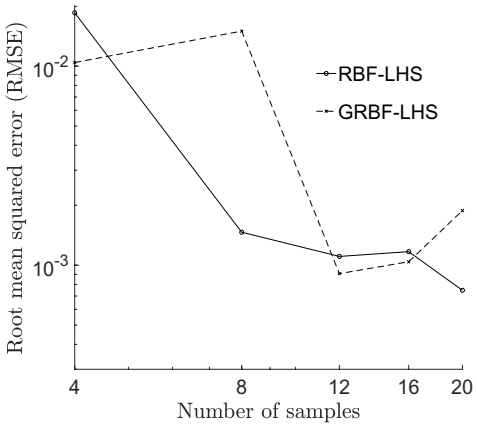
An RVE with a single fiber at the center is considered as shown in Figure 4.8a which details the loading and dimensions of the RVE. The RVE has a volume fraction of 0.55. The maximum cohesive traction σ_c at the fiber/matrix in-



(a) 1-fiber composite model RVE



(b) LS/GLS error plot



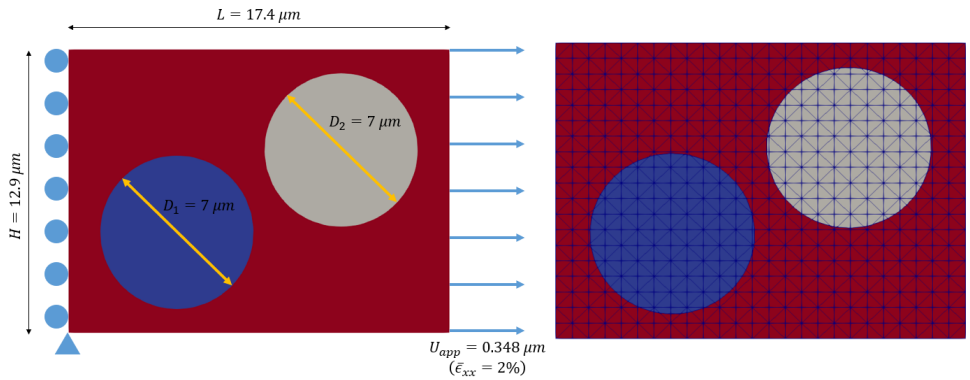
(c) RBF/GRBF error plot

Figure 4.8: (a) 1-fiber composite RVE model with applied boundary conditions, with RMSE as a function of sample size for (b) LS/GLS and (c) RBF/GRBF surrogate constructions. In most cases, incorporating the gradient information speeds convergence.

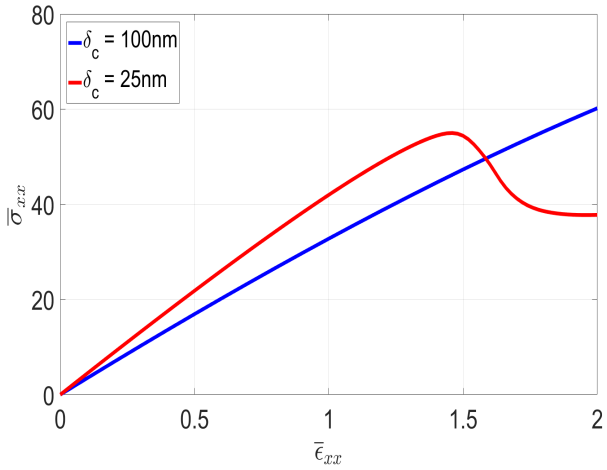
terface is varied between 10 and 60 MPa and $\delta_c = 100\ nm$. Figures 4.8b and 4.8c shows the RMSE plot comparisons between LS/GLS methods and RBF/GRBF methods respectively. The LHS sampling design is used for training the surrogate models by gradually increasing the number of σ_c samples from 4 to 20 in steps of 4. The RMSE values of the surrogate models for these 5 sample sets are estimated using a test data set of 1001 equispaced samples in the range $[10\ MPa,\ 60\ MPa]$. For the LS/GLS case, it is seen that GLS error convergence is faster than the LS error convergence showing the usefulness of gradient information. For the RBF/GRBF case, the RMSE values are very similar to each other indicating that the gradient information is not very useful in this case. It is also observed that the LS regression methods produce more accurate results than the RBF methods.

4.5.2 2-fiber model

An RVE with 2 fibers of equal radii with a volume fraction of 0.34 is considered. As shown in figure 4.9a, σ_{c_1} is the maximum cohesive traction parameter for the fiber/matrix interface corresponding to the fiber on the left with diameter D_1 and σ_{c_2} corresponds to the fiber on the right with diameter D_2. Both σ_{c_1} and σ_{c_2} vary independently between 10 and 60 MPa. Figure 4.9b shows the non-conformed RVE mesh used for all the simulation cases. Figure 4.9c shows the stress-strain response curves for two cases run at a particular case

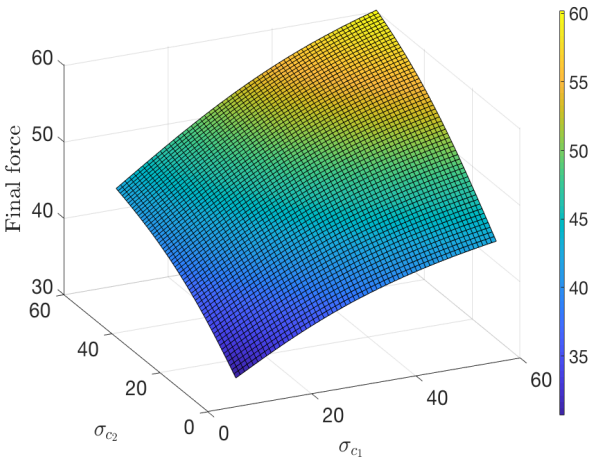


(a) 2-fiber composite model RVE (b) RVE nonconformal mesh

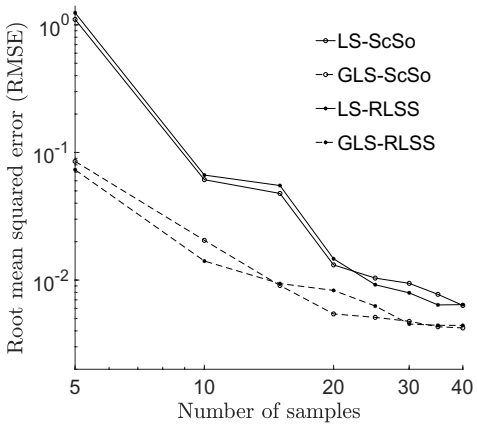


(c) Stress-strain curves

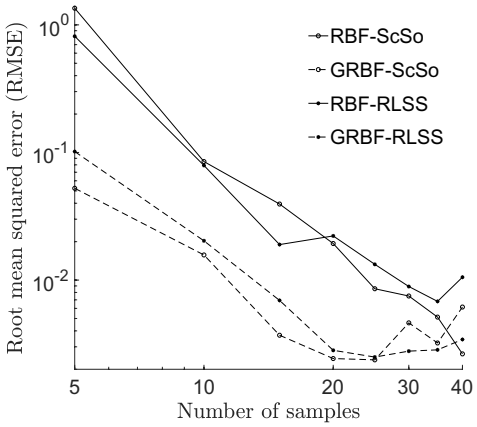
Figure 4.9: (a) 2-fiber microstructure RVE geometries, boundary conditions, (b) mesh, and (c) computed effective stress vs. strain for two different characteristic crack opening displacements, $\delta_c = 25\ nm$ and $\delta_c = 100\ nm$, assuming both fibers have the same maximum cohesive tractions $\sigma_{c_1} = \sigma_{c_2} = 60$ MPa.



(a) True output surface



(b) LS/GLS error plot



(c) RBF/GRBF error plot

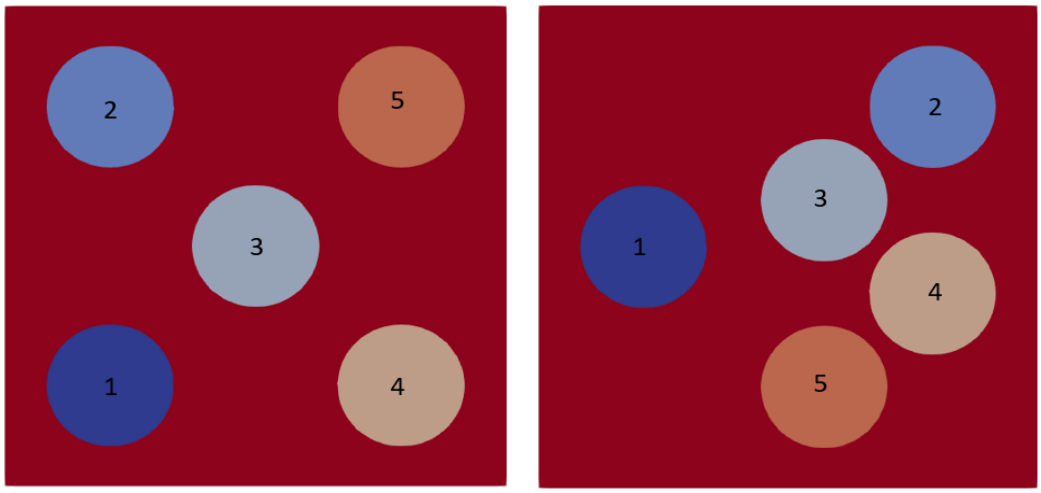
Figure 4.10: (a) True output surface from a 2-fiber composite model with characteristic crack opening displacement $\delta_c = 100\ nm$, with the RMSE error as a function of sample size using (b) LS/GLS) and (c) RBF/GRBF surrogate constructions. Gradient information leads to faster convergence of the results.

of $\sigma_{c_1} = \sigma_{c_2} = 60$ MPa but two different δ_c values of 25 nm and 100 nm. It is again noted that the stress corresponding to the final applied strain is the model output in this study. For each case, ScSo and RLSS samples are used for training the surrogate models by gradually increasing the number of σ_c samples from 5 to 40 in steps of 5 samples. The RMSE values of the surrogate models are estimated using a set of $60 \times 60 (= 3600)$ test grid points in the range $[10,\ 60]\ MPa$.

Figure 4.10a shows the smooth nature of the true response surface of the average stress with respect to the cohesive traction parameters for the 2-fiber composite model at a characteristic opening displacement of $\delta_c = 100\ nm$. Figures 4.10b and 4.10c show that the RMSE converges faster when the gradient-based approaches (GLS and GRBF) are used, for both sampling designs (ScSo and RLSS).

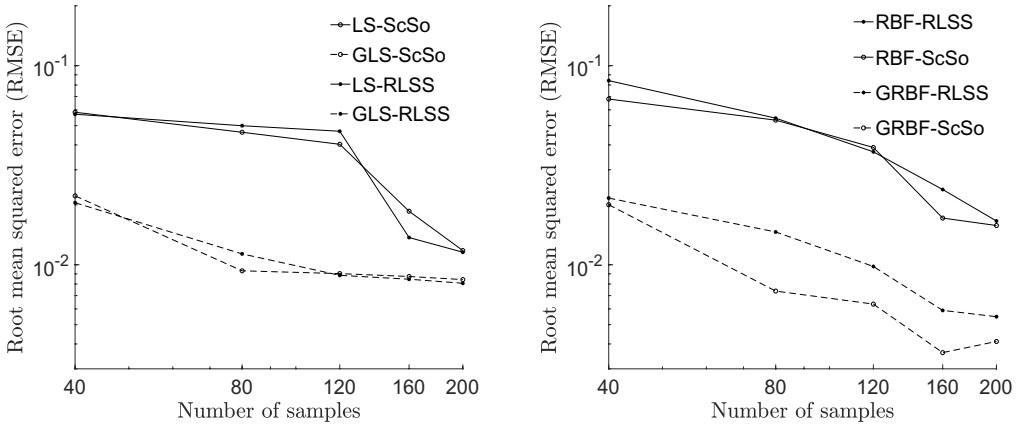
4.5.3 5-fiber model

In this section, two RVEs with 5 fibers are considered where one has a structured arrangement of the fibers as shown in figure 4.11a while the other has a more random and clustered arrangement as shown in figure 4.11b. For each of the 5 fibers, the cohesive traction parameters σ_{c_i} $(i = 1, 2, \ldots, 5)$ are assumed to vary between 10 MPa and 60 MPa, and $\delta_c = 100nm$. 500 ScSo and 500 RLSS design samples are generated for training and testing of the surrogate mod-



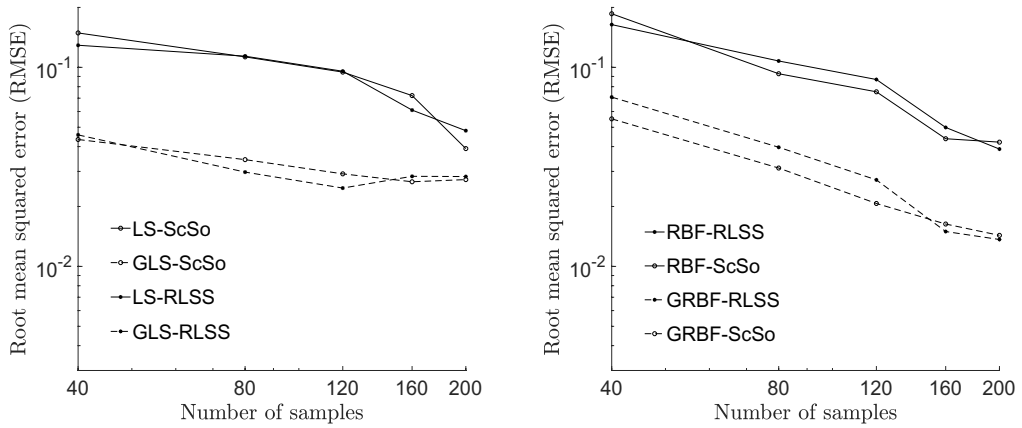
(a) Structured 5-fiber model (b) Clustered 5-fiber model

Figure 4.11: 5-fiber composite RVE models with (a) structured and (b) clustered arrangement of fibers



(a) LS/GLS error plot (b) RBF/GRBF error plot

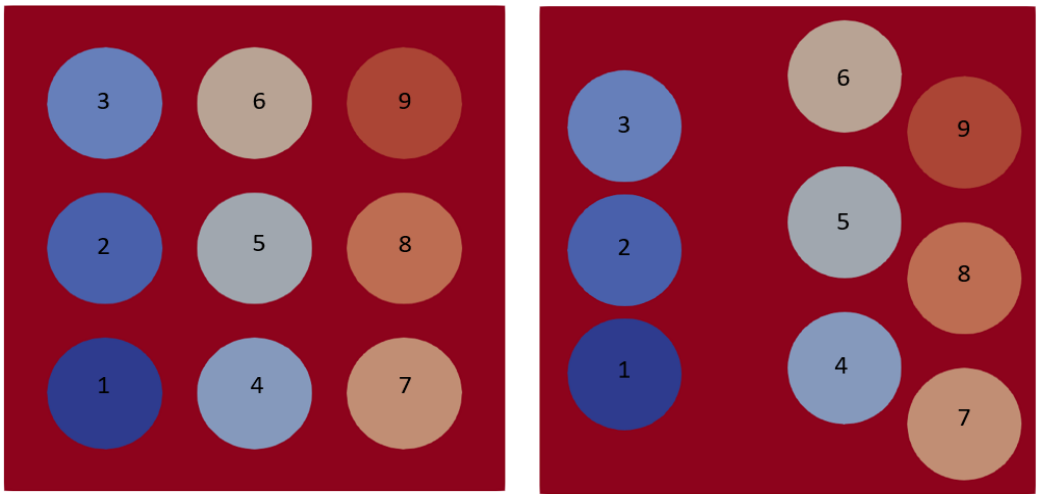
Figure 4.12: RMSE as a function of sample size for the structured 5-fiber composite model with characteristic crack opening displacement $\delta_c = 100nm$, for (a) LS/GLS surrogate construction and (b) RBF/GRBF surrogate construction.



(a) LS/GLS error plot (b) RBF/GRBF error plot

Figure 4.13: RMSE as a function of sample size for the clustered 5-fiber composite model with characteristic crack opening displacement $\delta_c = 100nm$, for (a) LS/GLS surrogate construction and (b) RBF/GRBF surrogate construction.

els. For each design, the first 200 samples are used for training by gradually increasing the number of σ_c samples from 40 to 200 in steps of 40 samples. After training the surrogate models with either design, RMSE calculations for measuring surrogate quality are performed on a common 600 sample test set using the remaining 300 samples from both designs. Figure 4.12 shows the error comparison for the structured 5-fiber models using the LS/GLS methods and the RBF/GRBF methods while Figure 4.13 shows the error comparison for the clustered 5-fiber models using the LS/GLS methods and the RBF/GRBF methods. It is seen from figures 4.12a and 4.13a that the GLS method performs better than the LS method especially for small samples sizes and their performance curve seems to converge with increase in sample sizes. From figures 4.12b and 4.13b, it is observed that the difference in performance of the



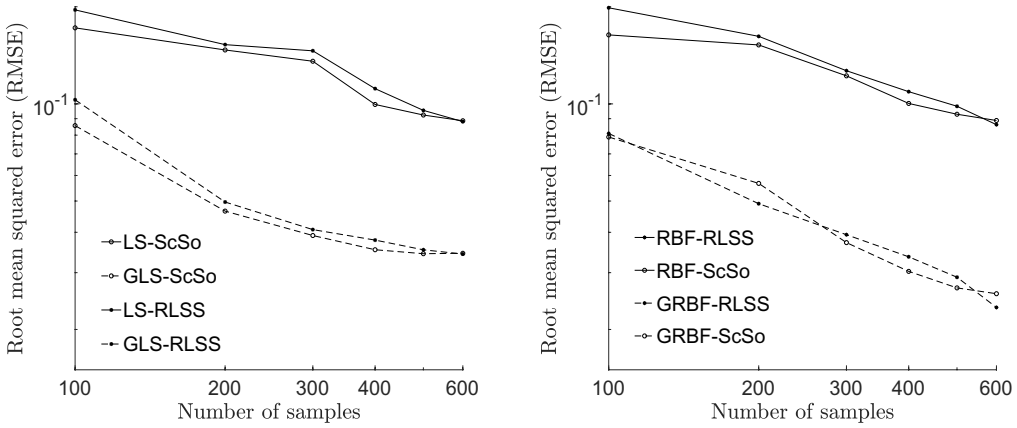
(a) Structured 9-fiber model (b) Clustered 9-fiber model

Figure 4.14: 9-fiber composite RVE models with (a) structured and (b) clustered arrangement of fibers

RBF and the GRBF methods with either of the designs tend to be the same over the range of sample sizes considered. While there are some differences in the RMSE between the structured and clustered arrangements, this does not appear to be a significant trend.

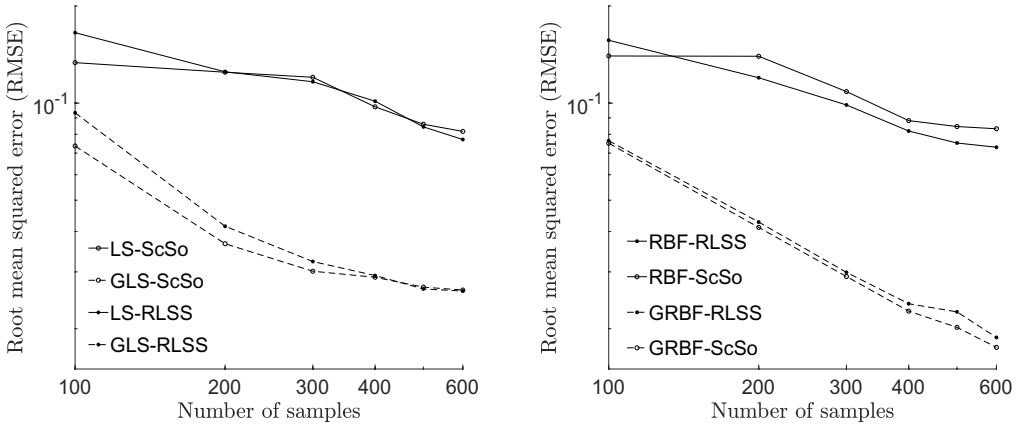
4.5.4 9-fiber model

This section considers two 9-fiber RVEs where one has a structured arrangement of the fibers as shown in figure 4.14a while the other has a clustered arrangement as shown in figure 4.14b. Similar to studies in the previous section, for each of the 9 fibers, the cohesive traction parameters σ_{c_i} $(i = 1, 2, \ldots, 9)$ are assumed to vary between 10 MPa and 60 MPa, and $\delta_c = 100\ nm$. 1000 ScSo and 1000 RLSS design samples are generated for training and testing of the



(a) LS/GLS error plot (b) RBF/GRBF error plot

Figure 4.15: RMSE as a function of sample size for the structured 9-fiber composite model with characteristic crack opening displacement $\delta_c = 100nm$, for (a) LS/GLS surrogate construction and (b) RBF/GRBF surrogate construction.



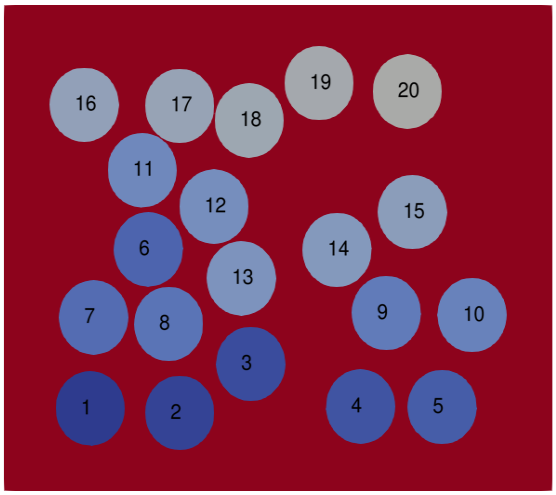
(a) LS/GLS error plot (b) RBF/GRBF error plot

Figure 4.16: RMSE as a function of sample size for the clustered 9-fiber composite model with characteristic crack opening displacement $\delta_c = 100nm$, for (a) LS/GLS surrogate construction and (b) RBF/GRBF surrogate construction.

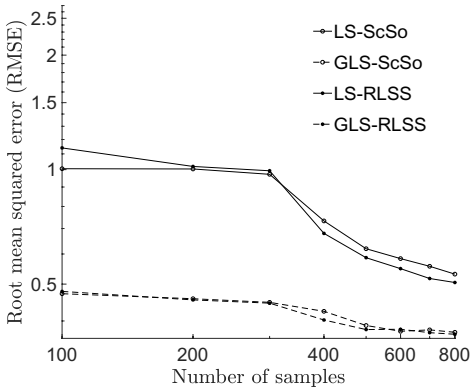
surrogate models. For each design, the first 600 samples are used for training by gradually increasing the number of σ_c samples from 100 to 600 in steps of 100 samples. After training the surrogate models with either design, RMSE calculations for measuring surrogate quality are performed on a common 800 sample test set using the remaining 400 samples from both designs. Figure 4.15 shows the error comparison plots for the structured 9-fiber models using the LS/GLS and RBF/GRBF methods while figure 4.16 shows similar error comparison plots for the clustered 9-fiber models. It is again observed that the gradient-based surrogate methods perform better than the corresponding non-gradient surrogate methods with both the sampling designs. Similar to the 5-fiber case, the fiber arrangement does not seem to have an influence on the quality of the surrogate approximation in the sense that the corresponding RMSE values are similar in order for both the clustered and structured arrangements. Therefore, in subsequent models, only a random arrangement of fibers is considered.

4.5.5 20-fiber model

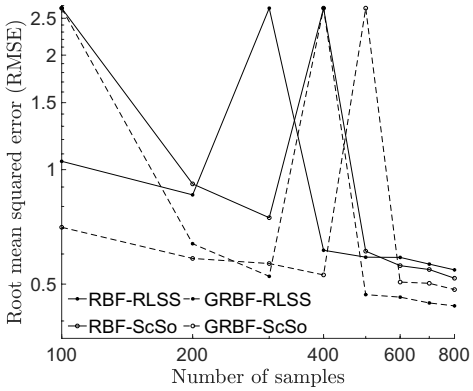
This section considers a 20-fiber RVE with a random arrangement of the fibers as shown in figure 4.17a. The cohesive traction parameters σ_{c_i} $(i = 1, 2, \ldots, 20)$ corresponding to each of the 20 fiber/matrix interfaces are assumed to vary between 10 MPa and 60 MPa, and $\delta_c = 100\ nm$. 1000 ScSo and 1000



(a) Random 20-fiber model

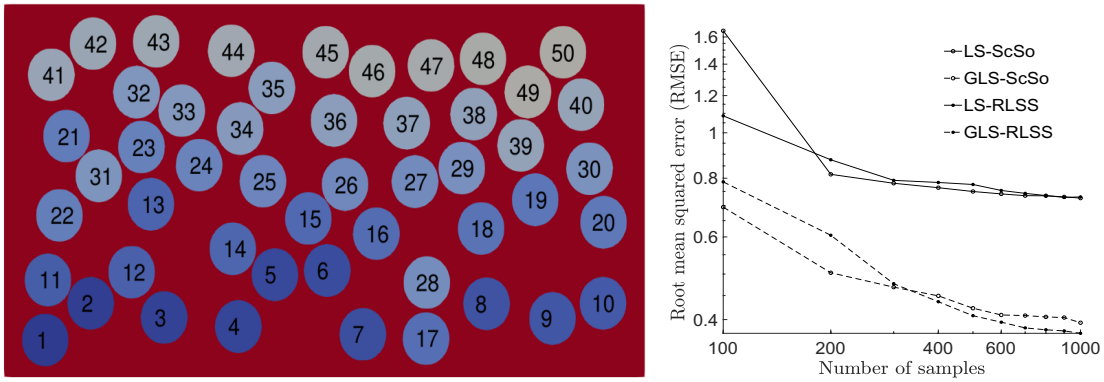


(b) LS/GLS error plot



(c) RBF/GRBF error plot

Figure 4.17: (a) 20-fiber composite model with a random arrangement of fibers. RMSE as a function of sample size for (b) LS/GLS and (c) RBF/GRBF construction show that gradients improve convergence.



(a) Random 50-fiber model

(b) LS/GLS error plot

Figure 4.18: (a) 50-fiber composite model with a random arrangement of fibers; (b) RMSE as a function of sample size for LS/GLS construction shows that gradients improve convergence. RBF/GRBF results not included as these constructions do not converge well in high dimensions, as observed in the 20-fiber case.

RLSS design samples are generated for training and testing of the surrogate models. For each design, the first 800 samples are used for training by gradually increasing the number of σ_c samples from 100 to 800 in steps of 100 samples. After training the surrogate models with either design, RMSE calculations for measuring surrogate quality are performed on a common 400 sample test set using the remaining 200 samples from both designs. Figures 4.17b and 4.17c show the error comparison plots for the random 20-fiber model using the LS/GLS and RBF/GRBF methods respectively. It is again observed that the gradient-based GLS surrogate method performs better than the corresponding non-gradient LS surrogate methods with both the sampling designs. The RBF and GRBF methods tend to produce noisy RMSE plots and the RMSE values are similar to one another. The LS/GLS results seem to be more consistently

accurate in this case than the RBF/GRBF results. This can be attributed to the use of isotropic Gaussian kernels which might not be an ideal choice for higher dimensional RBF interpolation.

4.5.6 50-fiber model

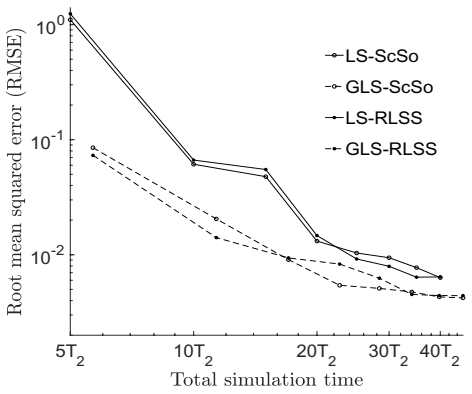
This section considers a 50-fiber RVE with a random arrangement of the fibers as shown in figure 4.18a. The cohesive traction parameters σ_{c_i} $(i = 1, 2, \ldots, 50)$ corresponding to each of the 50 matrix/fiber interfaces are assumed to vary between 10 MPa and 60 MPa, and $\delta_c = 100\ nm$. 1500 ScSo and 1500 RLSS design samples are generated for training and testing of the surrogate models. For each design, the first 1000 samples are used for training by gradually increasing the number of σ_c samples from 100 to 1000 in steps of 100 samples. After training the surrogate models with either design, RMSE calculations for measuring surrogate quality are performed on a common 1000 sample test set using the remaining 500 samples from both designs. Figure 4.18b shows the error comparison plots for the random 50-fiber model using the LS/GLS methods. It is again observed that the additional gradient information helps in achieving faster error convergence for the GLS method compared to the LS methods with both the sampling designs. Similar to the 20-fiber model, the RBF and GRBF methods perform inconsistently for the 50-fiber case with increase in number of samples and the results are not shown here.

Table 4.1: Time for calculating the sensitivities expressed as percentage of the simulation time of solving the IGFEM model without calculating the sensitivities for the different IGFEM multi-fiber models

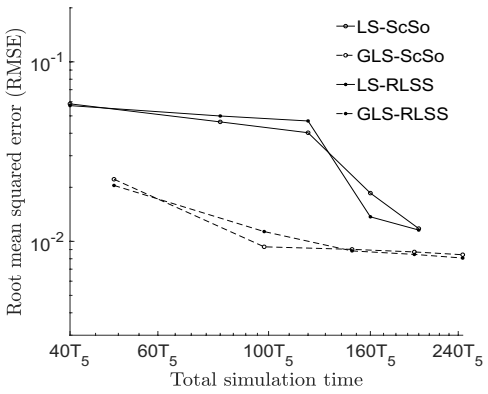
IGFEM model	Sensitivity calculation time (in $\%$)
2-fiber	13.6
5-fiber	22.5
9-fiber	25.2
20-fiber	44
50-fiber	70

4.5.7 Analysis of total computational cost

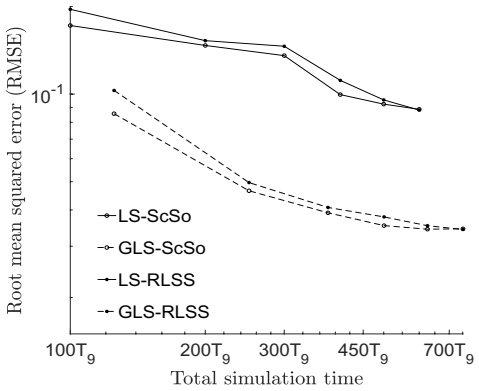
The gradient based surrogates were consistently found to be more efficient than the corresponding non-gradient based surrogates for the same number of IGFEM model simulations, but the additional cost of calculating the gradients were not taken into account. In this section, the total computational cost of the gradient and the non-gradient based surrogates are taken into account to generate a fair comparison between the two cases. For example, for the 2-fiber model, the simulation takes 13.6% longer when calculating the model output and its sensitivities as compared to only calculating the model output. If the simulation time for solving the 2-fiber IGFEM model without calculating the sensitivities is given by T_2, the total simulation time for solving the model with sensitivities is $1.136T_2$. The calculation times for the sensitivities of the different multi-fiber models is given in Table 5.1 which shows that the total time for calculating the sensitivities increases with increase in the number of fibers



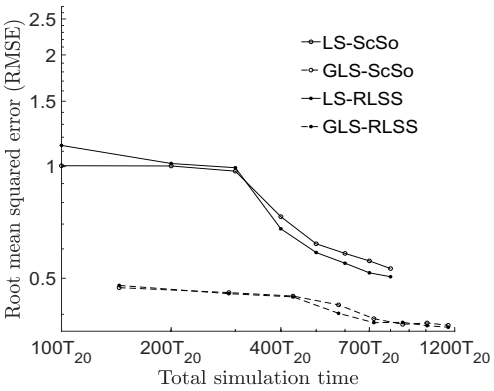
(a) 2-fiber model



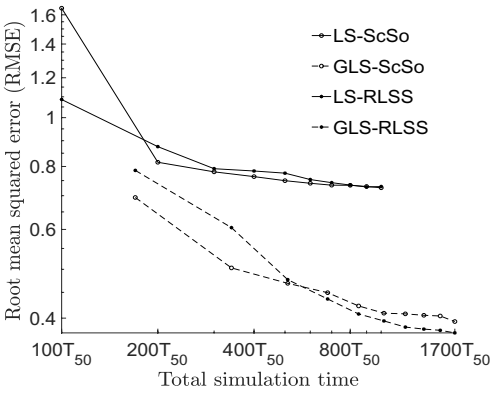
(b) Structured 5-fiber model



(c) Structured 9-fiber model



(d) 20-fiber model



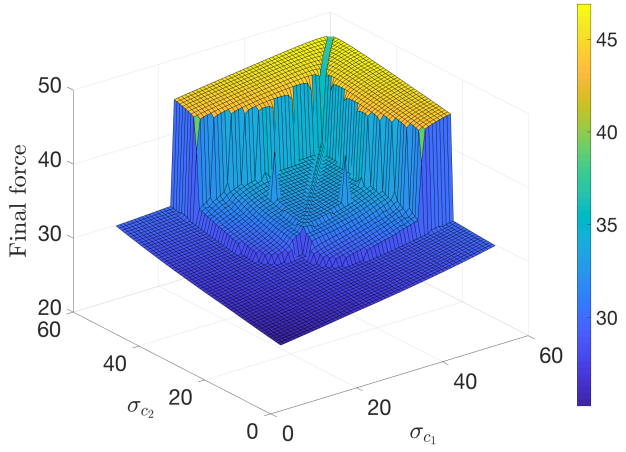
(e) 50-fiber model

Figure 4.19: Computational cost comparison for different multi-fiber composite RVE models where T_i is the total time taken to complete a simulation without the sensitivity calculation for an i-fiber model

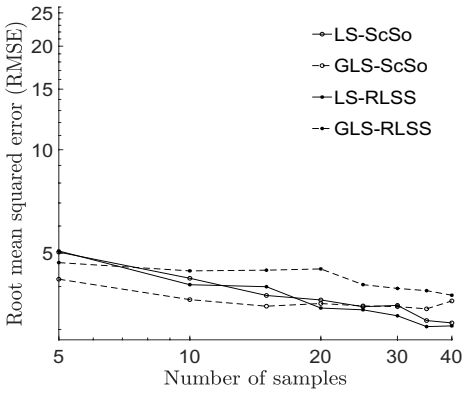
in the model. It is noted that the total simulation time without the sensitivity calculations is different for different multi-fiber models and increases with increase in number of fibers in the model. Figure 4.19 shows the variation of RMSE with increase in total simulation time for the 2-fiber, structured 5-fiber, structured 9-fiber, 20-fiber and 50-fiber models when LS/GLS methods are used as the surrogate methods. From the figures, it can be concluded that the enhancement in the accuracy of the surrogate approximation using the sensitivity information clearly outweighs the computational cost of obtaining the sensitivities. Similar trends (data not shown) are observed for the clustered 5-fiber and 9-fiber models, and also in most cases where RBF/GRBF methods are used.

4.5.8 2-fiber model with $\delta_c = 25nm$

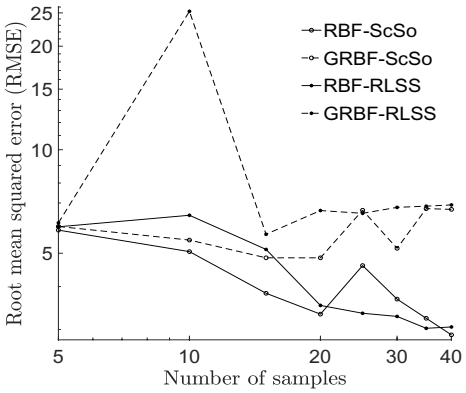
In this section, a 2-fiber RVE model is again considered with the same range of interface strengths but with a different characteristic opening displacement of $\delta_c = 25\ nm$. Figure 4.9c shows the non-linear stress-strain response curve run at $\sigma_{c_1} = \sigma_{c_2} = 60$ MPa and $\delta_c = 25\ nm$. Similar to the $100\ nm$ case, ScSo and RLSS samples are used for training the surrogate models by gradually increasing the number of σ_c samples from 5 to 40 in steps of 5 samples. The RMSE values of the surrogate models are estimated using a set of $60 \times 60 (= 3600)$ test grid points in the range $[10,\ 60]\ MPa$. Figure 4.20a shows the discontinuous true response surface of the average stress with respect to the cohesive



(a) True output surface



(b) LS/GLS error plot



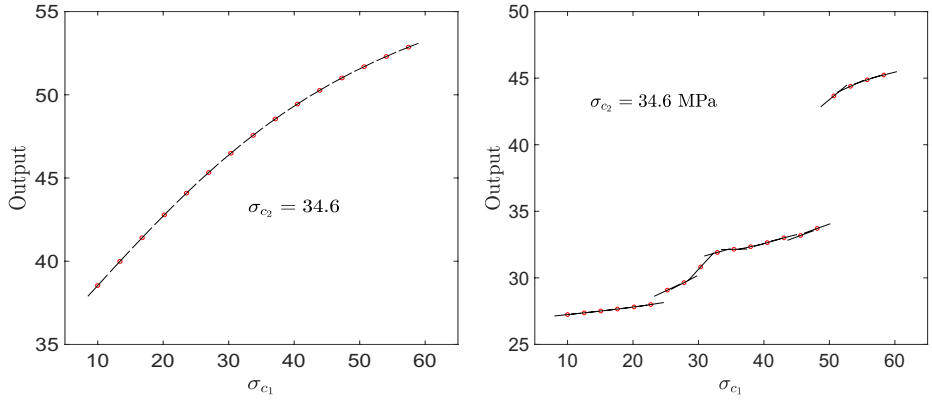
(c) RBF/GRBF error plot

Figure 4.20: (a) True output surface plot for 2-fiber composite model with characteristic crack opening displacement $\delta_c = 25\ nm$, with RMSE as a function of sample size for (b) LS/GLS and (c) RBF/GRBF construction. The discontinuity in the response surface leads to poor convergence of the global surrogate models.

traction parameters for the 2-fiber composite model at a characteristic opening displacement of $\delta_c = 25\ nm$ while figures 4.20b and 4.20c shows the RMSE convergence plot comparison of the output test values using the LS/GLS methods and RBF/GRBF methods using the two different designs. The error plots show that all the surrogate methods perform equally poorly in predicting the output values although the LS/GLS regression methods tend to perform slightly better and the error convergence is smoother than the RBF/GRBF interpolation methods.

4.5.9 Observation and hypothesis

From the RMSE convergence plots of all the multi-fiber composite models for $\delta_c = 100\ nm$, it is seen that the gradient-based GLS and GRBF methods consistently perform better than the non-gradient LS and RBF methods, indicating that the gradient information helped in achieving better quality surrogates. The only exception is for the 20-fiber and 50-fiber model cases where the RBF and GRBF methods show inconsistencies in the error measures and the advantage of using gradients is not clear. On the other hand, in a 2-fiber model with $\delta_c = 25\ nm$, all the surrogate methods perform equally poorly and the behavior is found to be similar to that of the 1-dimensional noisy sine function shown in figure 4.3. The discontinuity in the output surface has a similar effect as noise for these global surrogate methods. The additional gradient in-



(a) (b)

Figure 4.21: 2-fiber composite model stresses and stress sensitivities for crack opening displacement (a) $\delta_c = 100\ nm$ and (b) $\delta_c = 25\ nm$ with respect to fiber 1 cohesive strength, and the cohesive strengths of all other fibers in the models fixed at $\sigma_c = 35$ MPa. Discontinuities make it difficult to construct surrogates using the global methods presented here.

formation in this case provides no apparent advantage in building surrogates.

A rough idea about the nature of the output surfaces for $\delta_c = 25\ nm$ for models with higher numbers of fibers can be obtained by checking the 1-dimensional sensitivities of the stress output with respect to the cohesive traction parameters σ_{c_i} $(i = 1, \ldots, n_f)$ of each fiber/matrix interface in a n_f-fiber model where n_f denotes the number of fibers in the model. Figures 4.21a and 4.21b show the variation as well as the local gradients of the average stress (output) with respect to the cohesive traction parameter σ_{c_1} of the fiber/matrix interface corresponding to the fiber with diameter D_1 in a 2-fiber model [see figure 4.9a] for $\delta_c = 100\ nm$ and $\delta_c = 25\ nm$ respectively. The cohesive traction parameter value of the fiber/matrix interface corresponding to the other fiber

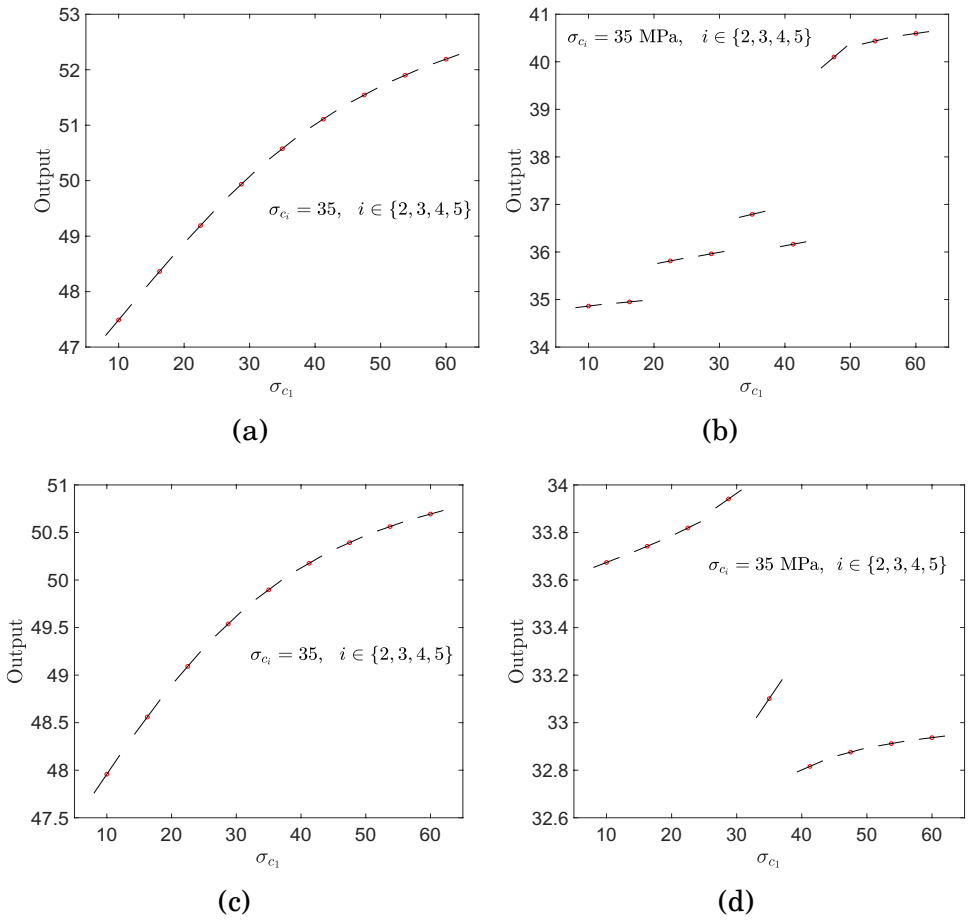


Figure 4.22: Structured 5-fiber composite model stresses and stress sensitivities for crack opening displacements (a) $\delta_c = 100\ nm$ and (b) $\delta_c = 25\ nm$ with respect to fiber 1 cohesive strength, and the cohesive strengths of all other fibers in the models fixed at $\sigma_c = 35$ MPa; Clustered 5-fiber composite model stresses and stress sensitivities for crack opening displacements (c) $\delta_c = 100\ nm$ and (d) $\delta_c = 25\ nm$.

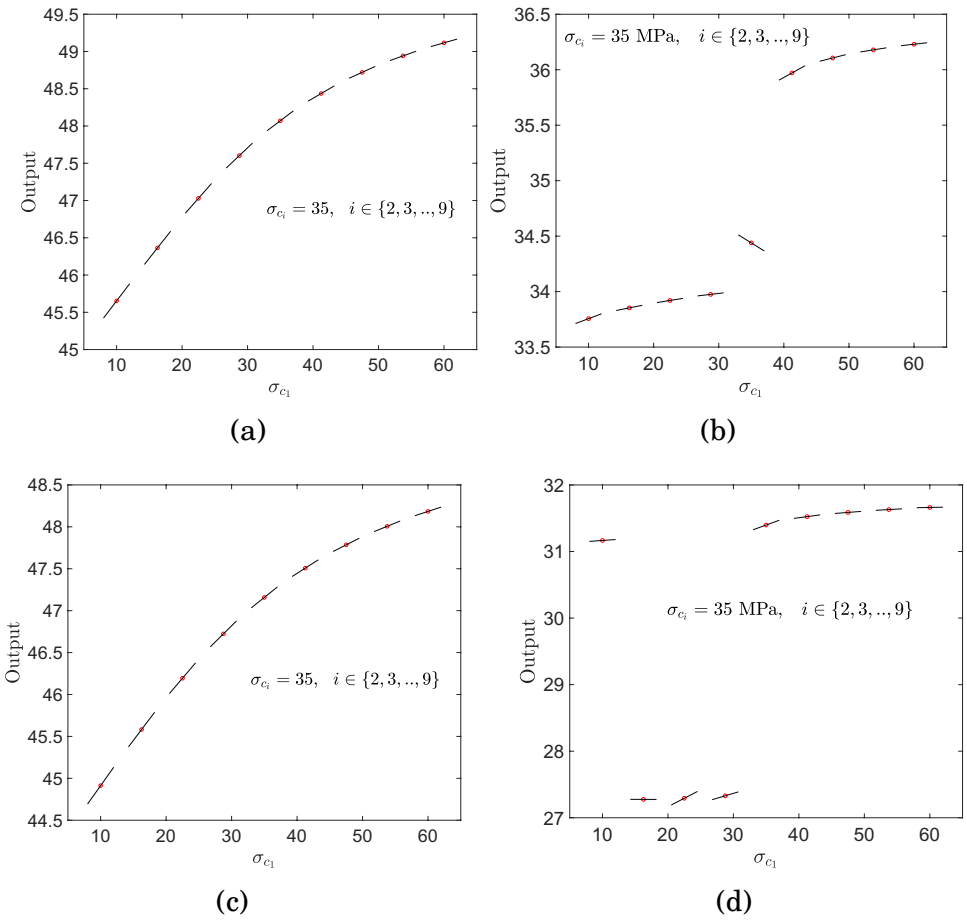


Figure 4.23: Structured 9-fiber composite model stresses and stress sensitivities for crack opening displacements (a) $\delta_c = 100\ nm$ and (b) $\delta_c = 25\ nm$ with respect to fiber 1 cohesive strength, and the cohesive strengths of all other fibers in the models fixed at $\sigma_c = 35$ MPa; Clustered 9-fiber composite model stresses and stress sensitivities for crack opening displacements (c) $\delta_c = 100\ nm$ and (d) $\delta_c = 25\ nm$.

is kept fixed at a value of 35 MPa. For $\delta_c = 100\ nm$, the variation is found to be smooth with conforming local gradients while for $\delta_c = 25\ nm$, the variation itself was found to be quite noisy. This noisy behavior seems to be the reason for the poor approximation quality of the surrogates as seen in figures 4.20b and 4.20c which is also obvious from figure 4.20a for the $\delta_c = 25\ nm$ case. Figures 4.22a and 4.22b shows the variation as well as the local gradients of the average stress (output) with respect to the cohesive traction parameter σ_{c_1} of the fiber/matrix interface corresponding to fiber 1 in a structured 5-fiber model [see figure 4.11a], for characteristic crack opening displacements of $\delta_c = 100\ nm$ and $\delta_c = 25\ nm$ respectively while figures 4.22c and 4.22d report the same for a randomly arranged clustered 5-fiber model [see figure 4.11b] for $\delta_c = 100\ nm$ and $\delta_c = 25\ nm$ respectively. The cohesive traction parameter value of the fiber/matrix interfaces corresponding to the 4 other fibers is kept fixed at a value of 35 MPa. For $\delta_c = 100\ nm$, the variation is found to be smooth with conforming local gradients while for $\delta_c = 25\ nm$, the variation itself was found to be quite noisy. Figures 4.23a and 4.23b show the variation as well as the local gradients of the average stress (output) with respect to the cohesive traction parameter σ_{c_1} of the fiber/matrix interface corresponding to fiber 1 in a structured 9-fiber model [see figure 4.14a] for $\delta_c = 100\ nm$ and $\delta_c = 25\ nm$ respectively while figures 4.23c and 4.23d report the same for a clustered 9-fiber model [see figure 4.14b] for $\delta_c = 100\ nm$ and $\delta_c = 25\ nm$ respectively. The

cohesive traction parameter value of the fiber/matrix interfaces corresponding to the 8 other fibers is kept fixed at a value of 35 MPa. For $\delta_c = 100\ nm$, the variation is found to be smooth with conforming local gradients while for $\delta_c = 25\ nm$, the variation itself was found to be quite noisy. Thus, at $\delta_c = 25\ nm$, all multi-fiber models are expected to produce a noisy output surface and the surrogate quality will be poor with or without the gradient-enhanced global surrogate models used in this study. To deal with such discontinuities, local surrogate methods should be implemented to get better error estimates. Such gradient-enhanced local surrogates are left to future work.

4.6 Conclusions

In this study, the usefulness of additional gradient information in building surrogates of low to intermediate dimensions is assessed. Using 3 different sampling point designs, both least squares (LS) and radial basis function approaches (RBF) are used in the gradient-enhanced and non-gradient-enhanced surrogate construction. The different sampling designs and surrogate models were first tested for simple 1-dimensional and 2-dimensional functions, which showed that the gradient-enhanced approaches produce more accurate results than the corresponding non-gradient-enhanced approaches if the gradients do not contain any noise. However, if the gradients are noisy,

the gradient-enhanced methods degrade in performance and the non-gradient-enhanced methods produce much more accurate results. There is relatively little distinction between the performance of the three sampling designs, though the RLSS and ScSo designs perform slightly better than the HLHS design for these examples. The RBF surrogate constructions lead to faster convergance than the LS surrogate constructions in the cases of a fluctuating function. The exception is that the LS approaches converged more rapidly when building a surrogate for a quadratic function.

As an application problem, fiber-reinforced matrix composite models with fiber/matrix interface damage are considered in this study. RVE models with 1, 2, 5, 9, 20 and 50 fibers are used. The homogenized stress (model output for this application problem) surrogates were generated as a function of the interfacial strengths of the fiber/matrix interfaces corresponding to each fiber in the RVE model. When a value of $100\ nm$ is used for the characteristic opening displacement in this particular example, the variation of the homogeneous stress is smooth and the gradient information used in GLS regression method results in better quality surrogates than the LS regression method. Similar trends were observed for the GRBF and the RBF methods except for the 20-fiber and 50-fiber model cases, in which cases the two techniques show similar results. When the value of the characteristic opening displacement is set at $25\ nm$, the variation of the output is found to be discontinuous (with small noise) for all

multi-fiber models and the quality of these global surrogate models is poor. Local surrogate methods might be an alternative for tackling discontinuous surfaces.

In summary, this study shows that the gradient-enhanced techniques presented here can lead to more efficient construction of surrogates, if the gradients are highly accurate, if calculating them does not lead to significant additional computational cost for each sampling point, and if the function to be approximated does not exhibit significant discontinuities or other localizations.

Chapter 5

Free energy calculation using space filled design and weighted reconstruction: A modified single sweep approach

A modified single sweep approach is proposed for generating free energy landscapes. The approach replaces the use of temperature-accelerated molecular dynamics (TAMD) [171] to generate centers in collective variable (CV) space at which mean forces are computed using restrained molecular dynamics (MD) simulations with a sequential space-filling design. This approach also modifies the radial basis function reconstruction step of the traditional single sweep

approach [30] and proposes a weighted reconstruction of the free energy surface using the previously generated mean forces. The weighted reconstruction strategy attempts to improve the approximation of the free energy landscape in the lower energy regions in the CV space and has been found to produce desired results for the free energy reconstruction of the analytical Müller potential [172]. The modified approach is compared to the traditional single sweep (SS) approach [30] on the (ϕ,ψ) dihedral free-energy map of solvated alanine dipeptide (AD). It is found that the new approach results in a more accurate reconstructed free energy than does the traditional approach when compared to the directly-computed reference free energy landscape. It is shown that the increased accuracy of the overall map stems from the improved 1-dimensional space filling (projective) property of the proposed design compared to that of the TAMD generated centers. A further enhancement in the accuracy of the crucial lower energy regions is enabled by the introduction of weights in the reconstruction step that give more importance to lower energy-valued regions. The increase in accuracy, observed even with a small number of centers, suggests that the new approach might be advantageous in computing free-energy landscapes in higher-dimensional CV spaces where it is not affordable to run restrained MD simulations at more than a certain number of centers.

5.1 Introduction

On molecular timescales, many biomolecular interactions and functions rely on relatively rare transitions between metastable states. All-atom molecular dynamics (MD) simulations offer the potential to view such events with atomic precision, provided one can overcome its computational timescale limitations. A standard way to do this is to invoke some meaningful coarse-grained or collective variables and statistically characterize their equilibrium distributions; because of the one-to-one correspondence between coarse-grained equilibrium distributions and free energy, methods to characterize such variables are termed free-energy methods. Many such methods focus on reconstructing a free-energy landscape via enhanced sampling, such as the weighted histogram analysis method (WHAM) [173] and metadynamics [174, 175], and the single-sweep approach [30].

Here, we revisit the single-sweep approach. In its original formulation, single-sweep first uses trajectories from temperature-accelerated MD (TAMD) [171] to obtain as broad as possible a sampling of the CV-space. Restrained simulations at each sample are then performed to compute mean forces, which in turn are used in a gradient-based reconstruction of the free energy. The single-sweep method has been used to study various processes including ion recruitment by proteins [176], gas diffusion in myoglobin and MSOX [177,178], ligand

binding [179] and other biologically relevant rare events [180, 181]. Though TAMD performs an enhanced sampling of CV space, the probability distribution it samples retains the structure of the physical surface, and thus it has relatively lower probabilities of visiting high-energy saddles as opposed to minima. Here we hypothesize that an efficient non-collapsive space-filling algorithm for assigning points in CV space at which gradients are computed via expensive MD simulations may lead to improved accuracy in the free-energy landscape relative to TAMD-based single-sweep.

Efficient space filling algorithms are heavily studied in the realm of response surface modeling/surrogate modeling. Standard surrogate models make use of model output values evaluated at a set of sampling points (structured sampling [85, 100, 112, 121] or unstructured sampling [135–137]) in the parameter space to build a response surface of the output. From the surrogate modeling framework, the current work focuses on building a response surface of the free energy (output) from MD simulation models that provide only the mean forces (output gradients) with respect to the collective variables (input parameters) at certain locations in the collective variable space.

In this work, we propose a sequential maximin space filling design for center selection where new candidate samples are placed at midpoints of existing samples along each dimension and selected by an L_1-distance maximization. The space-filling algorithm will effectively cover the landscape space in evenly

placed points minimizing error associated with biased distribution. The process will be further refined through a secondary step of weighted radial basis function reconstruction in an attempt to improve the quality of approximation in the crucial lower energy regions.

This chapter is organized as follows: section 5.2 discusses the single sweep method in brief. In section 5.3, the modified single sweep method is discussed in details. Section 5.4 includes two examples, the Müller potential and alanine dipeptide in aqueous solution with dihedral collective variables. Section 5.5 discusses the results in more details. Section 5.6 provides conclusions.

5.2 Single-sweep method

In the single-sweep method, the centers are at first identified in the CV space where the mean forces are calculated from restrained MD simulations. They are selected by quickly sweeping through low to moderately high energy regions in the free energy landscape using TAMD. Once the mean forces have been calculated at the chosen centers, the free energy is expressed as a linear combination of radial basis functions at the centers and the landscape is obtained by a global reconstruction from the evaluated mean forces. Thus the approach essentially consists of two sequential steps - 1) mean force evaluation at centers identified using TAMD sweeping, and 2) free energy reconstruction

using the mean forces. Each of the steps are described in brief below.

5.2.1 Generation of centers using TAMD

TAMD is a CV-based method to allow more extensive access to the free energy surface in a particular CV space. The method as used in this work has been fully described previously [171] and will only be briefly reiterated. A $3n$-dimensional molecular system configuration $\underset{\sim}{x}$ of n atoms is considered. A set of d collective variables (CVs) which are functions of the configuration $\underset{\sim}{x}$ is chosen and is denoted by $\underset{\sim}{\theta}(\underset{\sim}{x}) = (\theta_1(\underset{\sim}{x}), \theta_2(\underset{\sim}{x}), \ldots, \theta_d(\underset{\sim}{x}))$ such that $d \ll 3n$. In TAMD, normal molecular dynamics are extended to include restraints to CVs which are slowing evolving diffusively along the free energy landscape. In practice, evolution of the fundamental variables, $\underset{\sim}{x}(t)$, will occur as standard, all-atom MD simulations with a thermostat at the physical temperature, β^{-1}. The dynamics are extended to include forces which restrain each CV $\theta_j(\underset{\sim}{x})$ to a particular value z_j. Assuming Langevin dynamics, the dynamics of the fundamental variables $\underset{\sim}{x}$ in this system are:

$$M\ddot{\underset{\sim}{x}} = -\nabla_x V(\underset{\sim}{x}) - \kappa \sum_{j=1}^{d} (\theta_j(\underset{\sim}{x}) - z_j) \nabla_x \theta_j(\underset{\sim}{x}) - \gamma M \dot{\underset{\sim}{x}} + \underset{\sim}{\eta}(t; \beta) \tag{5.1}$$

In this equation, M is the mass matrix, $V(\underset{\sim}{x})$ is the interatomic potential, κ is the restrain spring constant, γ is the Langevin friction coefficient, and

$\underset{\sim}{\eta}$ is random fluctuation as described by the fluctuation-dissipation theory at physical temperature β^{-1}:

$$\langle \eta_i(t;\beta)\eta_j(t';\beta)\rangle = \beta^{-1}\gamma\mu_i\delta_{ij}\delta(t-t'). \tag{5.2}$$

In TAMD, CVs are not fixed but rather evolve as a slow moving variables according to their own equations of motion: in this case diffusive motion, over the free energy surface as:

$$\bar{\gamma}\bar{\mu}_j\dot{z}_j = \kappa[\theta_j(\underset{\sim}{x}) - z_j] + \zeta_j(t;\bar{\beta}) \tag{5.3}$$

Here, $\bar{\gamma}$ is the friction acting on the fictitious particle and $\bar{\mu}_j$ is the mass of z_j. On the right hand side, the first term represents the instantaneous force on z_j and the second term is the thermal noise at fictitious temperature $\bar{\beta}^{-1} > \beta^{-1}$:

$$\langle \zeta_i(t;\bar{\beta})\zeta_j(t';\bar{\beta})\rangle = \bar{\beta}^{-1}\bar{\gamma}\mu_j\delta_{ij}\delta(t-t'). \tag{5.4}$$

If we choose $\bar{\gamma}$ such that the movement of the CV is indeed slow in comparison to the fundamental variables and κ such that $\underset{\sim}{z}(\underset{\sim}{x}(t)) \approx \underset{\sim}{\theta}(t)$, then the dynamics of $\underset{\sim}{\theta}(t)$ become:

$$\bar{\gamma}\bar{\mu}_j\dot{z}_j = -\frac{\partial F(\underset{\sim}{z})}{\partial z_j} + \zeta(t;\bar{\beta}) \tag{5.5}$$

and the force acting on $\underset{\sim}{z}$ becomes approximately equal and opposite to the

gradient of the free energy. Under these conditions, the slow variable exhibits dynamics at an increased fictitious temperature $\bar{\beta}^{-1}$ while moving according to the free energy as evaluated at the physical temperature β^{-1}. The enhanced sampling enabled by $\bar{\beta}^{-1}$ allows one to visit higher energy states in CV space without altering the actual free energy landscape.

To find suitable centers for free energy reconstruction, TAMD is used to sweep the free energy landscape and center placement is determined as described in [30]. We use the resulting TAMD trajectory $\underset{\sim}{z}(t)$ to generate the relevant centers $\underset{\sim}{z}_1, \ldots, \underset{\sim}{z}_N$. A cutoff distance, s_{min}, is specified as the minimum distance in CV space between centers and each trajectory snapshot is probed to determine the distance from all other chosen centers. In practice, $\underset{\sim}{z}(0) = \underset{\sim}{z}_1$ for each distance, and a new center is determined at each snapshot when $\underset{\sim}{z}(t)$ is more than s_{min} away from all other predetermined centers. The distance s_{min} indirectly determines the number of centers where a larger s_{min} will result in less centers over the space and a smaller s_{min} will result in closer centers and thus a greater quantity of centers.

This method of center placement results in a random and robust set of centers which should be adequately evenly placed. The method gives no preference towards the free energy of any given region and assumes the given TAMD trajectory visits all relevant phase transitions. The center generation procedure is followed by restrained MD simulations at each center to obtain mean forces.

5.2.2 Radial basis function reconstruction

Once the mean forces have been obtained from restrained MD simulations at the centers, a 2-dimensional thermodynamic integration approach can then be used to recover the free energy which is discussed in this section. Free energy $P(\underset{\sim}{z})$ in the CV space can be expressed as

$$\begin{aligned} P(\underset{\sim}{z}) &= -k_B T \; ln \int d\underset{\sim}{x} \; \rho(\underset{\sim}{x}) \; \delta[\underset{\sim}{\theta}(\underset{\sim}{x}) - \underset{\sim}{z}] \\ &= -k_B T \; ln \frac{\int d\underset{\sim}{x} \; \delta[\underset{\sim}{\theta}(\underset{\sim}{x}) - \underset{\sim}{z}] \; e^{-\frac{V(\underset{\sim}{x})}{k_B T}}}{\int d\underset{\sim}{x} \; e^{-\frac{V(\underset{\sim}{x})}{k_B T}}} \\ &= -k_B T \ln \langle \delta[\underset{\sim}{\theta}(\underset{\sim}{x}) - \underset{\sim}{z}] \rangle \end{aligned} \tag{5.6}$$

where $V(\underset{\sim}{x})$ is the interatomic potential energy, k_B is Boltzmann's constant, T is temperature and $\langle . \rangle$ is the ensemble average of its argument over the equilibrium probability distribution $\rho(\underset{\sim}{x})$. However, this definition of free energy necessitates visiting all possible conformations in space which would require infinite MD simulation time and therefore is not a realistic approach especially in real systems which must overcome large free energy barriers to transition between metastable states. As such, alternative methods have been studied which allow for computation of the free energy profile. Thermodynamic integration is an alternative way to efficiently compute the free energies. This method requires computation of mean forces $\underset{\sim}{f}_{z}$, the negative gradient of free

energy $\left[\underset{\sim}{f}_{z} = -\nabla_{\underset{\sim}{z}} P(\underset{\sim}{z})\right]$, locally at centers $\underset{\sim}{z}$, and the goal is to mathematically reconstruct the free energy landscape from the mean force values. A radial-basis function representation [182] for the free energy P_z is given by

$$\tilde{P}(\underset{\sim}{z}) = \sum_{j=1}^{N} a_j \phi_\sigma(|\underset{\sim}{z} - \underset{\sim}{z}_j|) + C \tag{5.7}$$

and the gradient is given by

$$\nabla_{\underset{\sim}{z}} \tilde{P}(\underset{\sim}{z}) = \sum_{j=1}^{N} a_j \nabla_{\underset{\sim}{z}} \phi_\sigma(|\underset{\sim}{z} - \underset{\sim}{z}_j|) \tag{5.8}$$

where C is a global constant used to change the origin of $\tilde{P}(\underset{\sim}{z})$, $|.|$ denotes the euclidean norm in $\mathbb{R}^d$, N denotes the number of centers, $\phi_\sigma(.)$ is a radial basis function with adjustable shape parameter σ, and $\underset{\sim}{a} = [a_1, \ldots, a_N]$ is the unknown coefficient vector. The radial basis function $\phi(.)$ is chosen to be Gaussian and is given by:

$$\phi_\sigma(|\underset{\sim}{z} - \underset{\sim}{z}_j|) = e^{-|\underset{\sim}{z} - \underset{\sim}{z}_j|^2 / 2\sigma^2} \tag{5.9}$$

Free energy calculation involves the estimation of the coefficient vector $\underset{\sim}{a}$ and parameter σ. This is done by fixing σ and minimizing a residual function to obtain the coefficient vector $\underset{\sim}{a}$. Then σ is varied to get the combination of $(\underset{\sim}{a}, \sigma)$ which achieves the lowest value of the residual function over all possible combinations. The residual function measures the error between the negative of

the free energy gradient in Eq. (5.8) and the mean forces calculated at the centers and is given by:

$$R(\underset{\sim}{a},\sigma)=\sum_{i=1}^{N}\left|\sum_{j=1}^{N}a_j\nabla_{\underset{\sim}{z}}\phi_\sigma(|\underset{\sim}{z}_i-\underset{\sim}{z}_j|)+\underset{\sim}{f}_i\right|^2 \tag{5.10}$$

where $\underset{\sim}{f}_i$ is the mean force vector at center i. Minimizing Eq. (5.10) leads to the following linear system of equation which is solved to obtain the coefficient vector $\underset{\sim}{a}$:

$$\underset{\approx}{G}\underset{\sim}{a}=\underset{\sim}{F} \tag{5.11}$$

which can also be expressed component-wise as

$$\sum_{j=1}^{N}G_{k,j}(\sigma)a_j(\sigma)=F_k(\sigma) \tag{5.12}$$

where $G_{k,j}$ and $F_k(\sigma)$ are given by

$$G_{k,j}(\sigma)=\sum_{i=1}^{N}\nabla_{\underset{\sim}{z}}\phi_\sigma(|\underset{\sim}{z}_i-\underset{\sim}{z}_k|).\nabla_{\underset{\sim}{z}}\phi_\sigma(|\underset{\sim}{z}_i-\underset{\sim}{z}_j|) \tag{5.13}$$

$$F_k(\sigma)=-\sum_{i=1}^{N}\nabla_{\underset{\sim}{z}}\phi_\sigma(|\underset{\sim}{z}_i-\underset{\sim}{z}_k|).\underset{\sim}{f}_i \tag{5.14}$$

Derivation of the linear system of equations. In this section, it is shown how the standard minimization procedure of the residual function expression leads

to the linear system of equations. The residual function R in Eq. (5.10) can be expanded in terms of the collective variables as follows:

$$R(\underset{\sim}{a},\sigma)=\sum_{i=1}^{N}\sum_{m=1}^{d}\left[\sum_{j=1}^{N}a_j\nabla_{z^{(m)}}\phi_\sigma(|\underset{\sim}{z}_i-\underset{\sim}{z}_j|)+f_i^{(m)}\right]^2 \tag{5.15}$$

Taking the derivative with respect to the coefficient vector components leads to:

$$\frac{\partial R(\underset{\sim}{a},\sigma)}{\partial a_k}=\sum_{i=1}^{N}\sum_{m=1}^{d}2\nabla_{z^{(m)}}\phi_\sigma(|\underset{\sim}{z}_i-\underset{\sim}{z}_k|)\left[\sum_{j=1}^{N}a_j\nabla_{z^{(m)}}\phi_\sigma(|\underset{\sim}{z}_i-\underset{\sim}{z}_j|)+f_i^{(m)}\right] \tag{5.16}$$

In order to minimize R, the derivative is set to zero such that,

$$\frac{\partial R(\underset{\sim}{a},\sigma)}{\partial a_k}=0 \tag{5.17}$$

$$\begin{aligned}\Rightarrow\quad \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{m=1}^{d}\left[\nabla_{z^{(m)}}\phi_\sigma(|\underset{\sim}{z}_i-\underset{\sim}{z}_k|)\right]\left[\nabla_{z^{(m)}}\phi_\sigma(|\underset{\sim}{z}_i-\underset{\sim}{z}_j|)\right]a_j \\ =-\sum_{i=1}^{N}\sum_{m=1}^{d}\nabla_{z^{(m)}}\phi_\sigma(|\underset{\sim}{z}_i-\underset{\sim}{z}_k|)f_i^{(m)}\end{aligned} \tag{5.18}$$

$$\begin{aligned}\Rightarrow\quad \sum_{i=1}^{N}\sum_{j=1}^{N}\left[\nabla_{\underset{\sim}{z}}\phi_\sigma(|\underset{\sim}{z}_i-\underset{\sim}{z}_k|).\nabla_{\underset{\sim}{z}}\phi_\sigma(|\underset{\sim}{z}_i-\underset{\sim}{z}_j|)\right]a_j \\ =-\sum_{i=1}^{N}\nabla_{\underset{\sim}{z}}\phi_\sigma(|\underset{\sim}{z}_i-\underset{\sim}{z}_k|).\underset{\sim}{f}_i\end{aligned} \tag{5.19}$$

$$\Rightarrow \quad \underset{\sim}{\underset{\sim}{G}}\,\underset{\sim}{a} = \underset{\sim}{F} \tag{5.20}$$

Eq. (5.20) can also be written component-wise as:

$$\sum_{j=1}^{N} G_{k,j}(\sigma) a_j(\sigma) = F_k(\sigma) \tag{5.21}$$

where

$$G_{k,j}(\sigma) = \sum_{i=1}^{N} \nabla_{\underset{\sim}{z}} \phi_\sigma(|\underset{\sim}{z}_i - \underset{\sim}{z}_k|).\nabla_{\underset{\sim}{z}} \phi_\sigma(|\underset{\sim}{z}_i - \underset{\sim}{z}_j|) \tag{5.22}$$

$$F_k(\sigma) = -\sum_{i=1}^{N} \nabla_{\underset{\sim}{z}} \phi_\sigma(|\underset{\sim}{z}_i - \underset{\sim}{z}_k|).\underset{\sim}{f}_i \tag{5.23}$$

It is noted that the following property:

$$\nabla_{\underset{\sim}{z}} \phi_\sigma(|\underset{\sim}{z} - \underset{\sim}{z}_k|) = \nabla_{\underset{\sim}{z}} \phi_\sigma(|\underset{\sim}{z}_k - \underset{\sim}{z}|) \tag{5.24}$$

has been used to obtain Eq. (5.20).

5.3 Proposed modified single sweep approach

The single-sweep method proposed in [30] uses TAMD to sweep through the free energy landscape in the CV space and implements a distance threshold based approach to choose centers on the TAMD trajectory path. The free energy values are then generated by a radial basis function reconstruction approach using the mean forces values at the centers. In this study, a modified single sweep approach is proposed where a sequential space filled design has been proposed to choose the centers instead of the TAMD generated centers. In addition, a weighted reconstruction scheme is also proposed to improve the quality of the free energy landscape in the low energy regions. Each of the modified steps is described next.

5.3.1 Space filled design for choosing centers

A good space filling design ensures that for any given number of points sampled, the input sample space is covered in a relatively even fashion. A brief overview of three classes of sampling designs, latin hypercube sampling (LHS), stratified sampling (SS) and low discrepancy sequences, has already been given in chapter 4 (section 4.2) and to some extent in chapter 1 (section 1.5.1).

In this study, a hierarchical space filled design has been adopted to select centers in the CV space. This method ensures that the centers are not too close to each other which is crucial for an efficient reconstruction using radial basis functions. It is expected that the free energy landscape can potentially have many local minima, local maxima and saddle points. Since the radial basis function reconstruction is global, it is intuitive to choose centers that try to cover the entire CV space.

The proposed design is a sequential maximin distance design where design points are added one at a time. Let d be the number of input parameters/collective variables/input dimensions. The lower bound, centroid, and upper bound of the domain along each dimension act as anchor coordinates. The candidate coordinates along any dimension are generated by successive bisection of ordered anchor coordinates along each dimension. Once sampled, the coordinates of that center also play the role of anchor coordinates along respective dimensions for generating future candidates. If N centers have been generated, the number of candidate coordinates is $(N-1)$. Thus, the total number of candidate centers for dimension d is $(N-1)^d$. The candidate center which has the maximum nearest neighbour distance from the existing centers is chosen as the new center and obtained by the following optimization problem:

$$\max_{\forall \underset{\sim}{x}_c \in C} \min_{\forall \underset{\sim}{x}_s \in S} D(\underset{\sim}{x}_c, \underset{\sim}{x}_s) \tag{5.25}$$

where $D(\underset{\sim}{a}, \underset{\sim}{b}) = \left(\sum_{i=1}^{d} |a_i - b_i|^p \right)^{1/p}$, S is the set of all existing centers and C is the set of all candidate centers. When d or N is large, the number of candidate centers $(N-1)^d$ can get quite large. For that reason, a feasible number of 1-dimensional candidate coordinates are randomly sampled with replacement along each dimension and combined to get the candidate centers. Thus the design is random in nature. A simple 2-dimensional demonstration of sample addition in an aperiodic domain using the proposed design is shown in figure 5.1.

Algorithm steps. The sequential maximin design algorithm needs the following inputs for sample generation: number of collective variables (input dimensions) d, coordinates of initial centers $\underset{\sim}{x}_{old}$, multiplicative factor associated with candidate sample generation f and number of samples to be added N_{new}.

Let the input domain be an unit hypercube $H = [0,1]^d$ and let N_{old} be the total number of already existing (initial) centers in the domain. The algorithm proceeds as follows:

1. Consider the extreme and the centroidal coordinates along each dimension. The initial set of anchor coordinates (shown as black-bordered grey squares in figure 5.1) are thus given by the vector: $[0, 1, 0.5]$.

2. Set $N = N_{old}$. If $N_{old} = 0$, go to step 3. Otherwise, go to step 5.

3. If the construction is periodic, coordinates $(0, 0, \ldots, 0)$ and $(0.5, 0.5, \ldots, 0.5)$

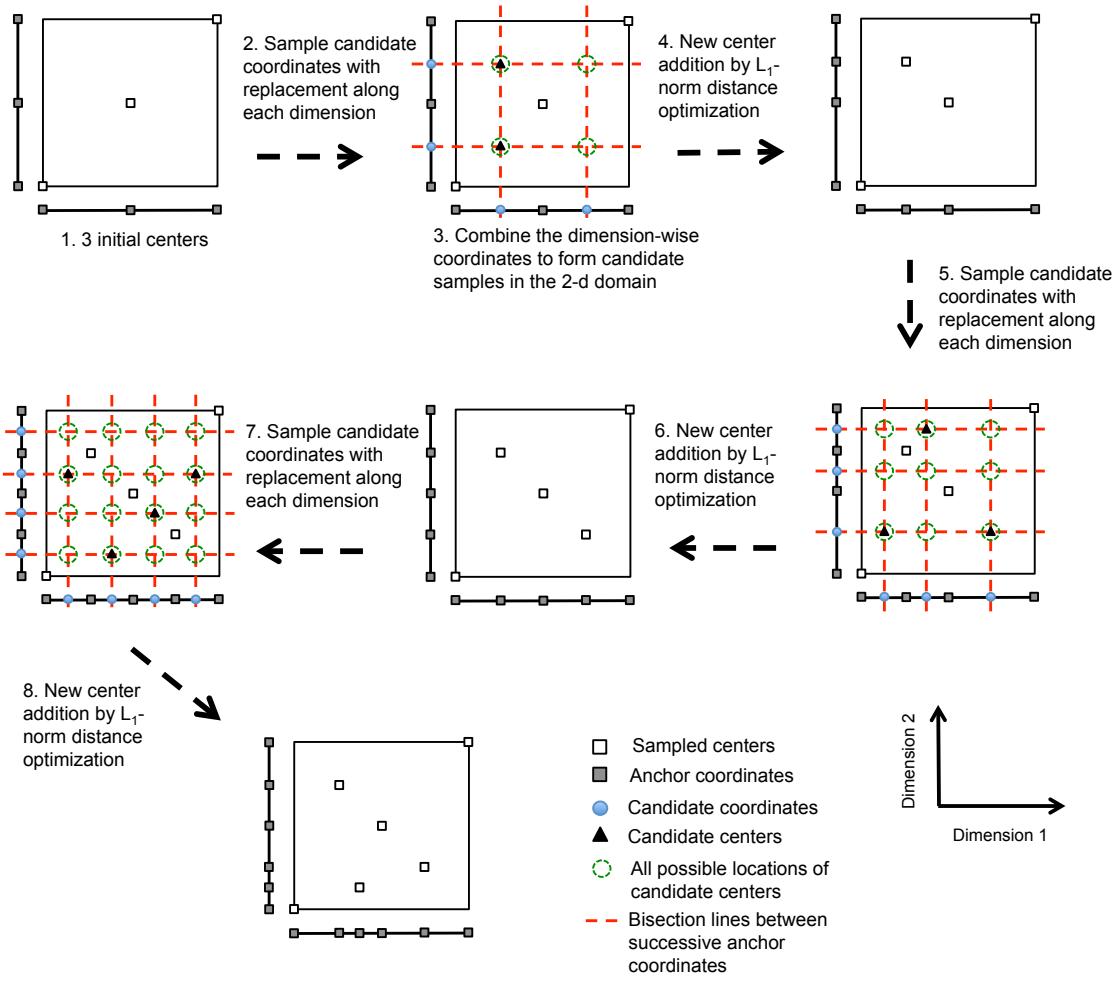


Figure 5.1: Sequential sampling procedure using the proposed space filling design for an aperiodic 2-dimensional domain, starting with 3 initial samples and adding 3 new samples in succession. (note: multiplicative factor $f = 1$ assumed here for demonstrative purposes)

are chosen as the 2 initial d-dimensional centers. Set $N = 2$. If the construction is aperiodic, coordinates $(0, \ldots, 0)$, $(0.5, \ldots, 0.5)$ and $(1, \ldots, 1)$ are chosen as the 3 initial sampled centers (shown as black-bordered white squares in figure 5.1). Set $N = 3$. Store them in the centers matrix X.

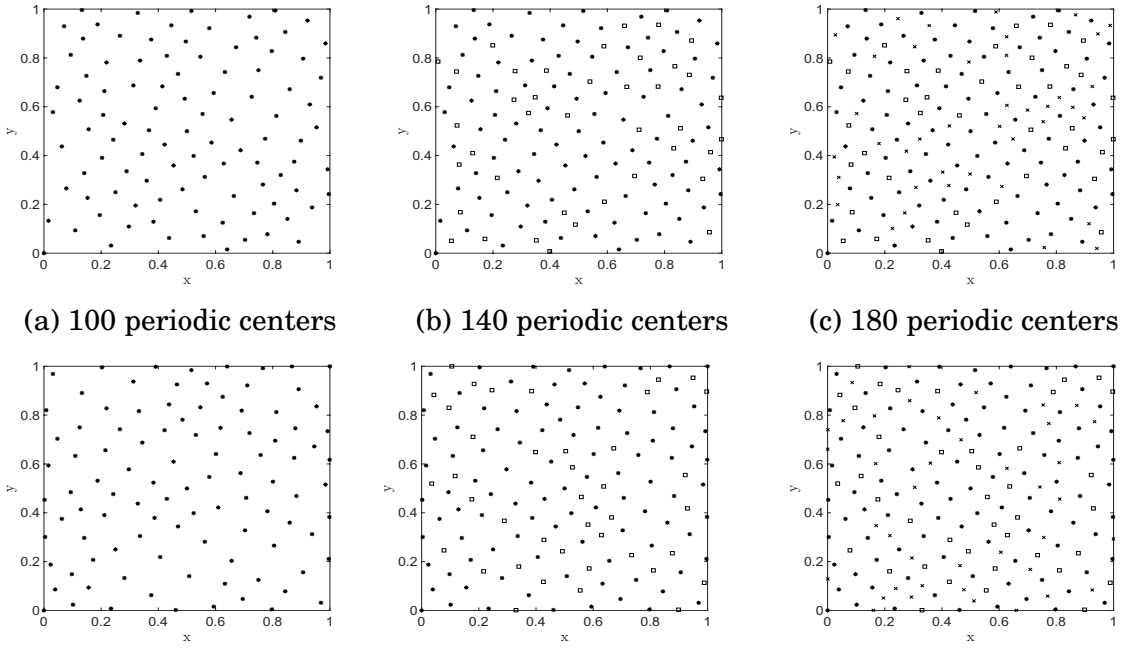
4. If $N < N_{new} + N_{old}$, go to step 5. Otherwise, terminate the loop.

5. Incorporate the coordinates of all the centers sampled thus far into the set of anchor coordinates and sort them along each dimension. Remove any duplication of coordinates. New candidate coordinates (N for periodic and $N - 1$ for aperiodic) are selected at the midpoints between successive anchor coordinates along each dimension (shown as blue circles in figure 5.1).

6. Randomly generate samples (Nf for periodic and $(N - 1)f$ for aperiodic) from the candidate coordinates with replacement for each dimension. In this study, $f = 100$. It is noted that there will be duplication of the generated samples because of the 'sampling with replacement' procedure.

7. Combine the 1-dimensional samples to get Nf periodic ($(N-1)f$ aperiodic) d-dimensional candidate centers (shown as black triangles in figure 5.1). Remove any duplication of these candidates, and the one that satisfies (5.25) is chosen. Store the center in X. Set $N = N + 1$.

8. If $N < N_{new} + N_{old}$, go to step 5. Otherwise, terminate the loop.



(a) 100 periodic centers (b) 140 periodic centers (c) 180 periodic centers

(d) 100 aperiodic centers (e) 140 aperiodic centers (f) 180 aperiodic centers

Figure 5.2: Sequential addition of the space filled design centers: (a-c) addition of periodic centers from 100 to 180 in steps of 40; (d-f) addition of aperiodic centers from 100 to 180 in steps of 40

It is clear from the description above that the design is sequential and tries to maintain its space filling nature with the addition of each new center. Thus, for a reconstruction procedure, the number of centers generated from this design can be increased in steps to progressively increase the accuracy of the free energy landscape. Figure 5.2(a-c) shows the sequential addition of centers from 100 to 180 in steps of 40 in case of a periodic construction while figure 5.2(d-f) shows the same for an aperiodic construction.

Mean nearest neighbor is the distance metric considered here to assess the space filling properties of the different sampling designs. The nearest neighbor

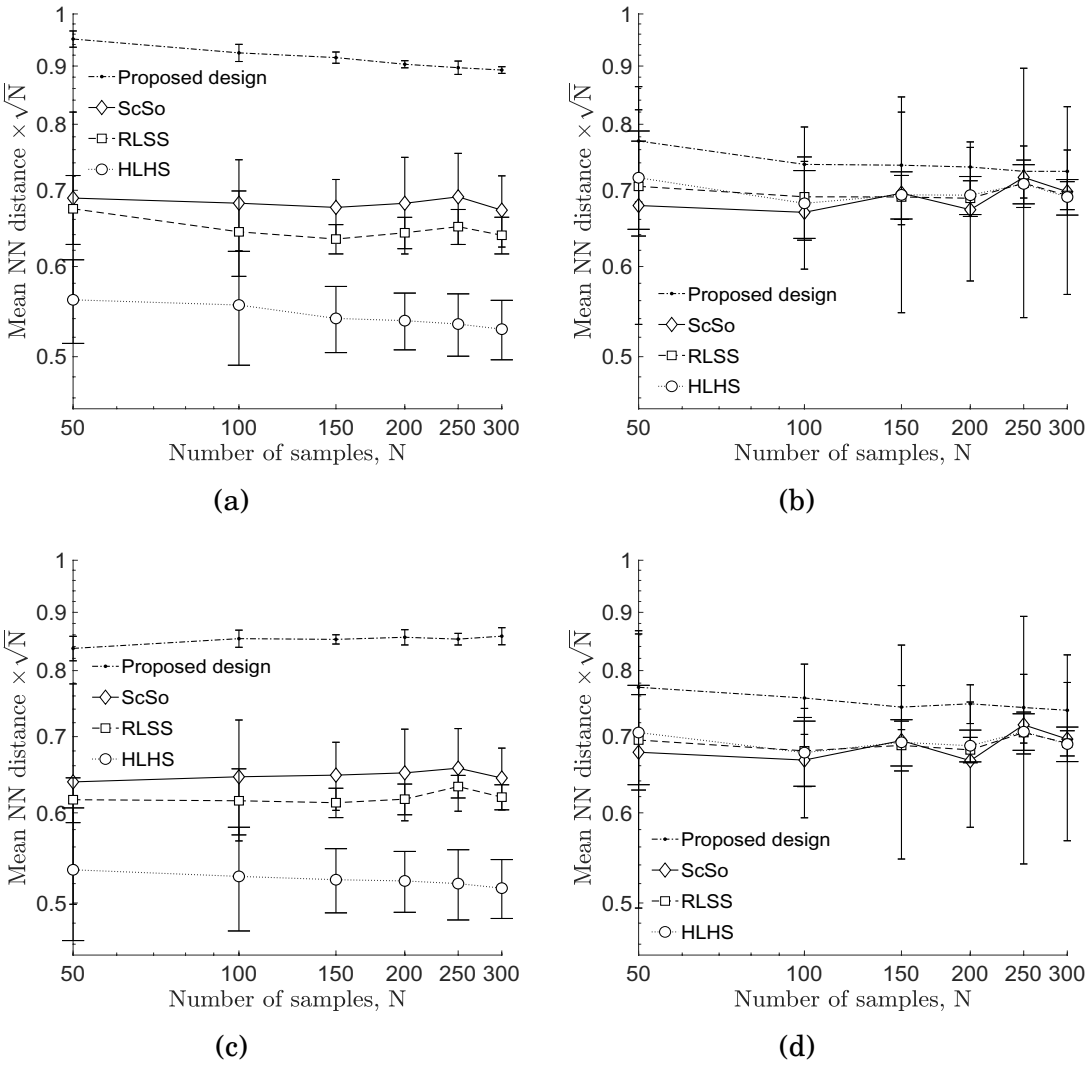


Figure 5.3: Performance comparison of space filling design for (a) aperiodic centers in 2 dimensions; (b) aperiodic centers projected along dimension 1; (c) periodic centers in 2 dimensions; (d) periodic centers projected along dimension 1

(NN) distances are obtained by calculating the Euclidean distances between each sample and its nearest neighbor and the mean NN distance is calculated by taking the mean of all the NN distances. From the above definition, it is clear that for a certain sample size N, the higher the values of the mean NN distance, the more the separation between the samples, which implies that the samples are more space filling. 20 sample sets are generated for each design. Figure 5.3a shows the mean NN distance comparison between the proposed design and the three common sequential space filled designs, HLHS, RLSS and Scrambled Sobol (ScSo), for the 2-dimensional samples when the domain is aperiodic. Figure 5.3c shows a similar comparison when the domain is periodic. Figures 5.3b and 5.3d show the mean NN distance comparison between the sampling designs for the 1-dimensional projections of the samples along the 1st dimension, when the domain is aperiodic and periodic, respectively. A similar trend as in figures 5.3(b,d) is observed when samples are projected along the 2nd dimension. It is thus seen that the proposed design produces samples with better space filling properties in both 1 and 2 dimensions. This improved space filling property in the lower dimensions suggests that the proposed design will have better projective properties. Also, the variation of the distance metric is smallest for the proposed design in 2 dimensions, suggesting that the approach is less sensitive to random sampling error.

5.3.2 Weighted free energy reconstruction

The important regions in the free energy landscape correspond to regions around local minima which are essentially low free energy regions. A weighted reconstruction scheme is hence proposed here to improve the quality of the free energy landscape in the low energy regions. This is achieved by introducing weights in the residual function from Eq. (5.10) and the modified expression is given by:

$$R^w(\underset{\sim}{a}, \sigma) = \sum_{i=1}^{N} w_i \left| \sum_{j=1}^{N} a_j \nabla_{\underset{\sim}{z}} \phi_\sigma(|\underset{\sim}{z}_i - \underset{\sim}{z}_j|) + \underset{\sim}{f}_i \right|^2 \tag{5.26}$$

where $w_i\{i = 1, 2, \ldots, N\}$ are weight coefficients corresponding to each center $z_i\{i = 1, 2, \ldots, N\}$.

Let's say P_i are the approximate free energy values at the centers z_i calculated from a previous reconstruction. Then the weights are calculated at the centers as follows:

$$w_i = e^{-(P_i - min\{\underset{\sim}{P}\})^2 / 2\bar{\underset{\sim}{P}}^2} \tag{5.27}$$

where $\bar{\underset{\sim}{P}}$ is the mean of free energy values estimated by minimizing the unweighted residual function given in Eq. (5.10). Minimizing the weighted residual function in Eq. (5.26) similar to the unweighted case leads to the following linear system of equations:

$$\underset{\sim}{\underset{\sim}{G}}^w \; \underset{\sim}{a}^w = \underset{\sim}{F}^w \tag{5.28}$$

where,

$$F_k^w(\sigma) = -\sum_{i=1}^{N} w_i \nabla_{\underset{\sim}{z}} \phi_\sigma(|\underset{\sim}{z}_i - \underset{\sim}{z}_k|).\underset{\sim}{f}_i \tag{5.29}$$

$$G_{k,j}^w = \sum_{i=1}^{N} w_i \nabla_{\underset{\sim}{z}} \phi_\sigma(|\underset{\sim}{z}_i - \underset{\sim}{z}_k|).\nabla_{\underset{\sim}{z}} \phi_\sigma(|\underset{\sim}{z}_i - \underset{\sim}{z}_j|) \tag{5.30}$$

5.3.3 Numerical implementation

The algorithm consists of the following main steps:

Initialization. The number of collective variables is denoted by d. N_{total} is the total number of centers where the mean forces would be calculated along each collective variable. N_{step} is the number of centers to be added at the start of each iteration for mean force evaluations. When all of the N_{total} centers have been used for mean force evaluation, the algorithm is terminated.

Choice of centers. The centers are selected from the space filled sequential design in a d-dimensional CV space described in section 5.3.1. Bounds of the CV space need to be known to implement such a design. In case of an unbounded CV space, the TAMD trajectory can be used only to get a rough estimate of the bounds of the relevant CV space and then the space filled design can be used to generate centers within those bounds.

Mean force evaluations. In this step, the mean forces are evaluated along each collective variable using restrained molecular dynamics (MD) simulations

Algorithm 2 : **Summarized steps**

1) *Initialization.* Set d, N_{total}, N_{step}, $n = N_{initial}$ and $iter = 1$.
2) Generate N_{total} centers and select the first n out of N_{total} centers.
3) Compute mean forces at the n centers.
4) Using the mean force values, the reconstruction step is executed using radial basis functions.
if reconstruction is unweighted, **then**
  estimate $\underset{\sim}{a}$ and σ from Eq. (5.11) and generate free energy values from Eq. (5.7)
end if
if reconstruction is weighted, **then**
  if iter = 1, **then**
    estimate $\underset{\sim}{a}$ and σ from Eq. (5.11) and generate free energy values from Eq. (5.7)
    Generate the weight vector $\underset{\sim}{w}$ from the energy values using Eq. (5.27) and then estimate $\underset{\sim}{a}$ and σ from Eq. (5.28)
  else
    estimate $\underset{\sim}{a}$ and σ from Eq. (5.28) using $\underset{\sim}{w}$ generated from the previous iteration.
  end if
end if
5) Construct the free energy landscape from the reconstruction parameters using Eq. (5.7).
if N_{total} centers have been used up, **then**
  terminate the algorithm
else
  consider N_{step} new centers as candidates for further mean force calculations.
  if reconstruction is weighted, **then**
    estimate free energy values at the $(n + N_{step})$ centers and estimate $\underset{\sim}{w}$
  end if
  set $n = (n + N_{step})$ and go to step 3.
end if

at each center.

Reconstruction. All the mean force data obtained thus far from restrained MD simulations in the CV space are used to estimate the free energy landscape globally using a radial basis function reconstruction approach. The reconstruction can be either weighted as proposed in this study, or unweighted [30]. The unweighted case [30] corresponds to solving Eq. (5.11) for the reconstruction parameters $\underset{\sim}{a}$ and σ while the weighted case involves solving Eq. (5.28) for the reconstruction parameters. It is worth stating here that the initial weights for the weighted reconstruction are obtained from an unweighted reconstruction in the first step.

A summary of the all the above steps is given in Algorithm 2.

5.4 Numerical Results

In this section, the two proposed concepts of selecting the centers using the maximin design as well as the weighted reconstruction is tested with two popular benchmark systems used in the literature for free energy calculations. One is a model potential, the Müller potential which has closed form expressions of the mean forces as well as the free energy. The other system is a solvated alanine dipeptide (AD) molecule with its configuration expressed in terms of two torsion angles at 300 K.

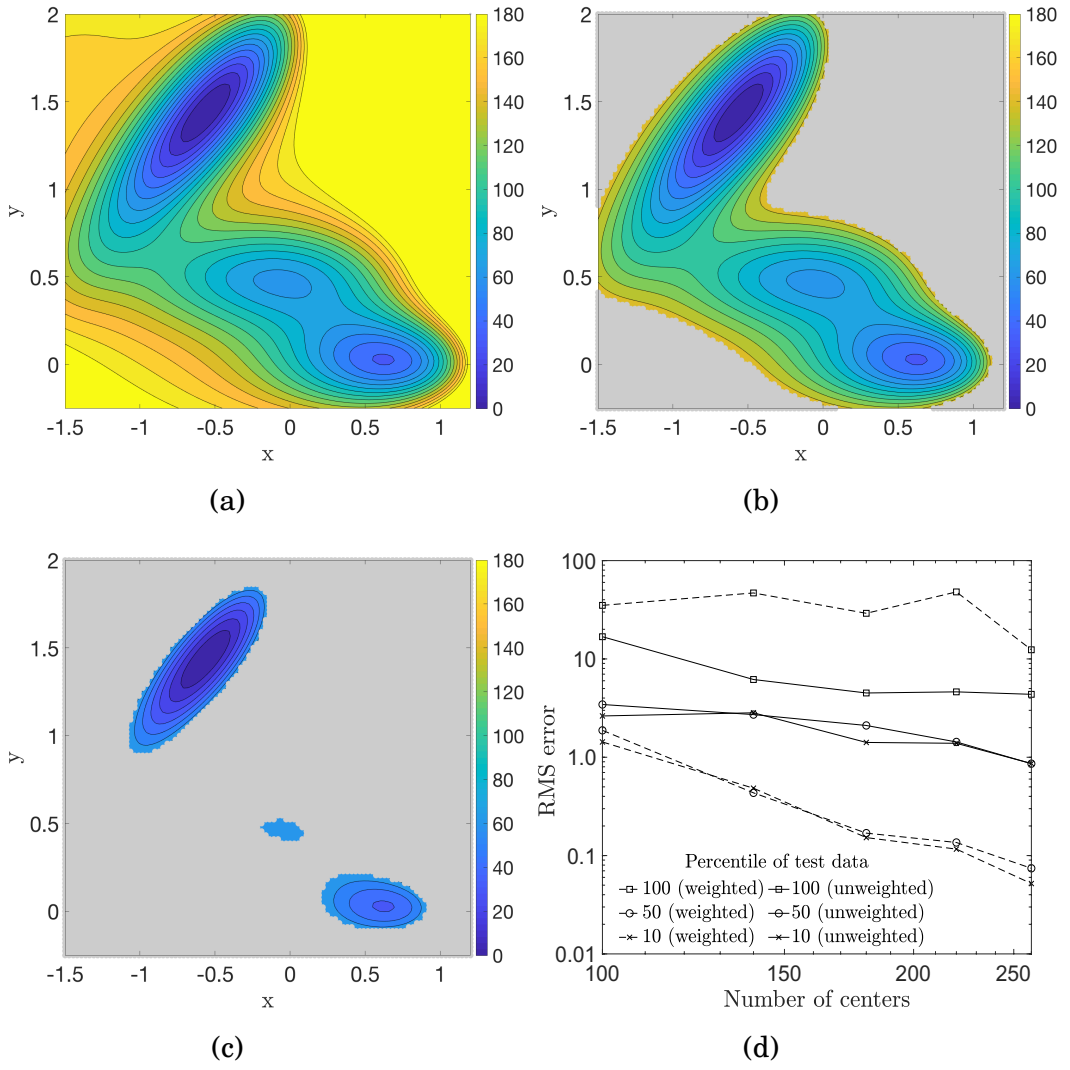


Figure 5.4: Müller potential contour with colored non-grey regions corresponding to values below (a) 100-th percentile, (b) 50-th percentile and (c) 10-th percentile of the test data set; (d) RMSE comparison plot between weighted (dashed line) and unweighted (solid line) reconstruction with the proposed space filled design using 10-th percentile (cross marker), 50-th percentile (circle marker) and 100th percentile (square marker) test data.

5.4.1 Müller Potential

The Müller Potential is a 2-dimensional model potential whose surface has 3 minima and 2 saddle points, and is given by [172]:

$$P = \sum_{i=1}^{4} D_i \, e^{[a_i(x-x_i^0)^2 + b_i(x-x_i^0)(y-y_i^0) + c_i(y-y_i^0)^2]},$$

$$-1.5 \leq x \leq 1.2, \quad -0.25 \leq y \leq 2 \tag{5.31}$$

where $\underset{\sim}{D} = [-200, -100, -170, 15]$, $\underset{\sim}{a} = [-1, -1, -6.5, 0.7]$, $\underset{\sim}{b} = [0, 0, 11, 0.6]$, $\underset{\sim}{c} = [-10, -10, -6.5, 0.7]$, $\underset{\sim}{x}^0 = [1, 0, -0.5, -1]$, $\underset{\sim}{y}^0 = [0, 0.5, 1.5, 1]$. The gradients can be obtained analytically at any location in the $x - y$ space by differentiating P in Eq. (5.31). Figure 5.4a shows the contour plot of the Müller potential with contours ranging from 0 to 180. True Müller potential values are estimated at a meshgrid of 100×100 $(= 10000)$ test points in the x-y space. 3 test data sets are used to assess the quality of the free energy construction. The first test data set is the 100-percentile data set, which constitutes all of the true data set and corresponds to the entire input x-y space as shown in figure 5.4a. The second test data set is the 50-percentile test data set, which constitutes all free energy values below the 50-th percentile and corresponds to the colored region (non-grey) in the input x-y space as shown in figure 5.4b. The third test data set is the 10-percentile test data set, which constitutes all free energy values below the 10-th percentile and corresponds to the colored region (non-grey) in

the input x-y space as shown in figure 5.4c. Test data in the regions marked in grey in the x-y space are ignored. Samples from the proposed design are used as centers for the reconstruction by increasing the number of samples from 100 to 260 in steps of 40 samples and the approximation quality of the potential surface is assessed for each of the 3 sample set cases described above. Since the input domain is aperiodic, the sampling design uses the aperiodic construction algorithm to generate the centers. It is noted that since the sampling design is sequential by construction, when moving from one sample set with M samples to the next sample set with $M + m$ centers, the M centers are common for both the sets. Figure 5.4d shows the root mean squared error (RMSE) convergence plot of the reconstructed Müller potential values for each of the 3 test data sets using the unweighted and the proposed weighted reconstruction scheme. As expected, there is a vertically downward shift in the error curve corresponding to the 10-percentile and 50-percentile test data set when the weighted reconstruction was used as opposed to the unweighted reconstruction. This implies that the weighted reconstruction helps improve the quality of approximation of the potential values in the lower energy regions. On the other hand, it is seen that the RMSE values for the 100-percentile test data set is worsened by the weighted reconstruction. This is expected because less weight is given to centers with approximate high potential values in the reconstruction leading to poorer approximation in those regions.

5.4.2 Alanine Dipeptide in solution

In this section, the benchmark example of the equilibrated solvated AD molecular system in two torsion angles at 300 K is considered. Though this system is not particularly interesting from a physical perspective, the extensive work into understanding and obtaining the 2-dimensional free energy profile proves useful in determining the efficacy of new methods. The goal here is to compare the performance of the single sweep (SS) method [30] with the proposed modified single sweep method used for the reconstruction of the free energy of the AD molecule. It is noted that the single sweep method uses TAMD sweeping simulations to obtain centers by a distance threshold criterion and then reconstruct the free energy by an unweighted reconstruction scheme. On the other hand, the proposed modified single sweep method uses a sequential space filling design to obtain centers for mean force calculations and then uses a weighted reconstruction scheme to obtain the free energy values. In the present work, TAMD sweeping simulations employed a time step of 1 fs. Temperature was controlled through Langevin dynamics [183] with a temperature of 300K and damping constant of 5 ps^{-1}. Long-range electrostatics are employed through particle-mesh Ewald summation with grid spacing of 1 . A cutoff distance of 9 was employed for nonbonded interactions. CHARMM force field [184, 185] was used with TIP3P waters [186]. In the solvated form, the simulation included 488 waters in a $26.4 \times 22.8 \times 24.1$ simulation box. To

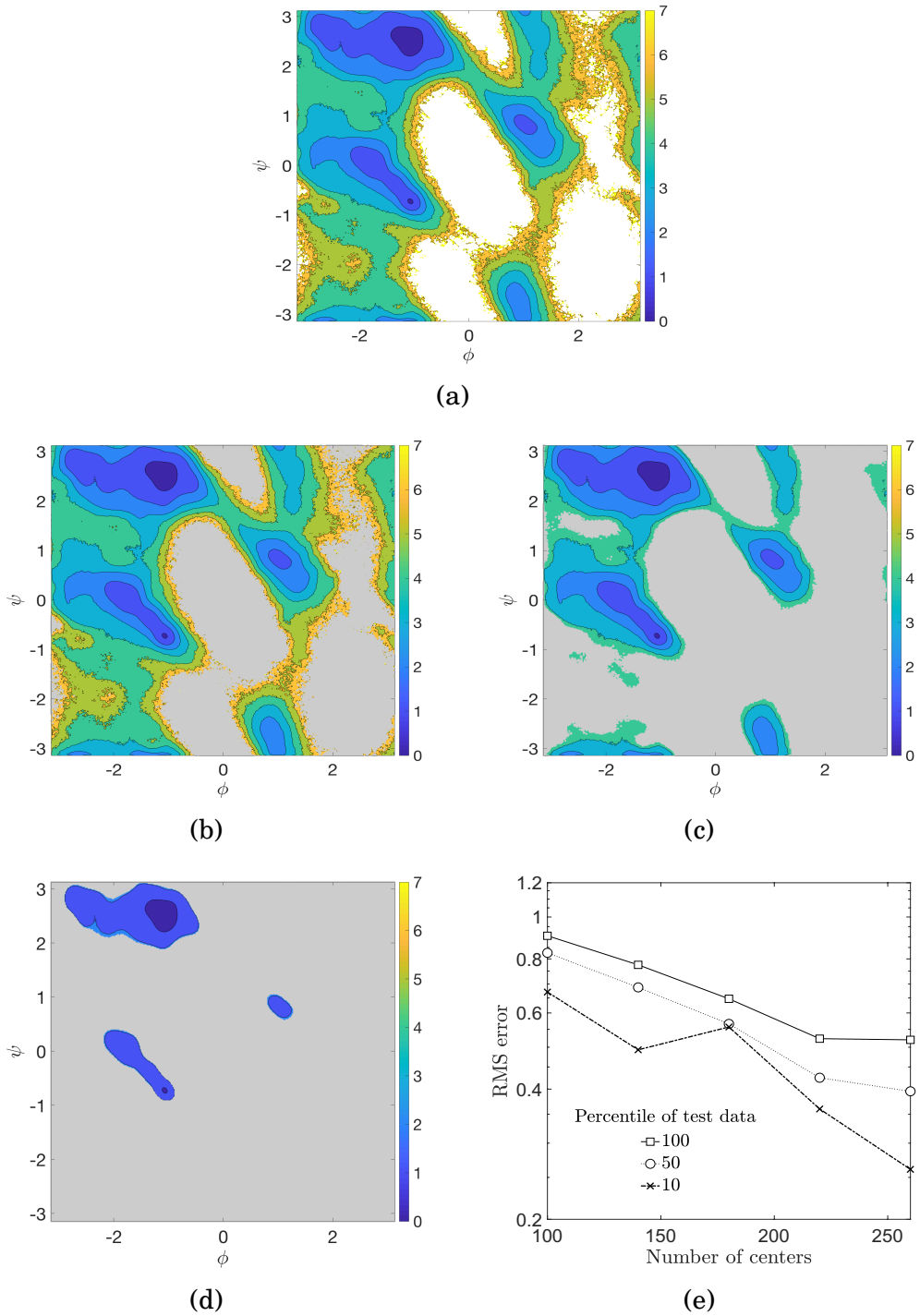


Figure 5.5: (a) Free energy contour of AD molecule from DMDS; Free energy contour with colored non-grey regions corresponding to values below (b) 100-th percentile, (c) 50-th percentile and (d) 10-th percentile of the test data set; (e) RMS error plot for 100 ps restrained MD simulations using 10-th percentile (cross marker), 50-th percentile (circle marker) and 100th percentile (square marker) test data

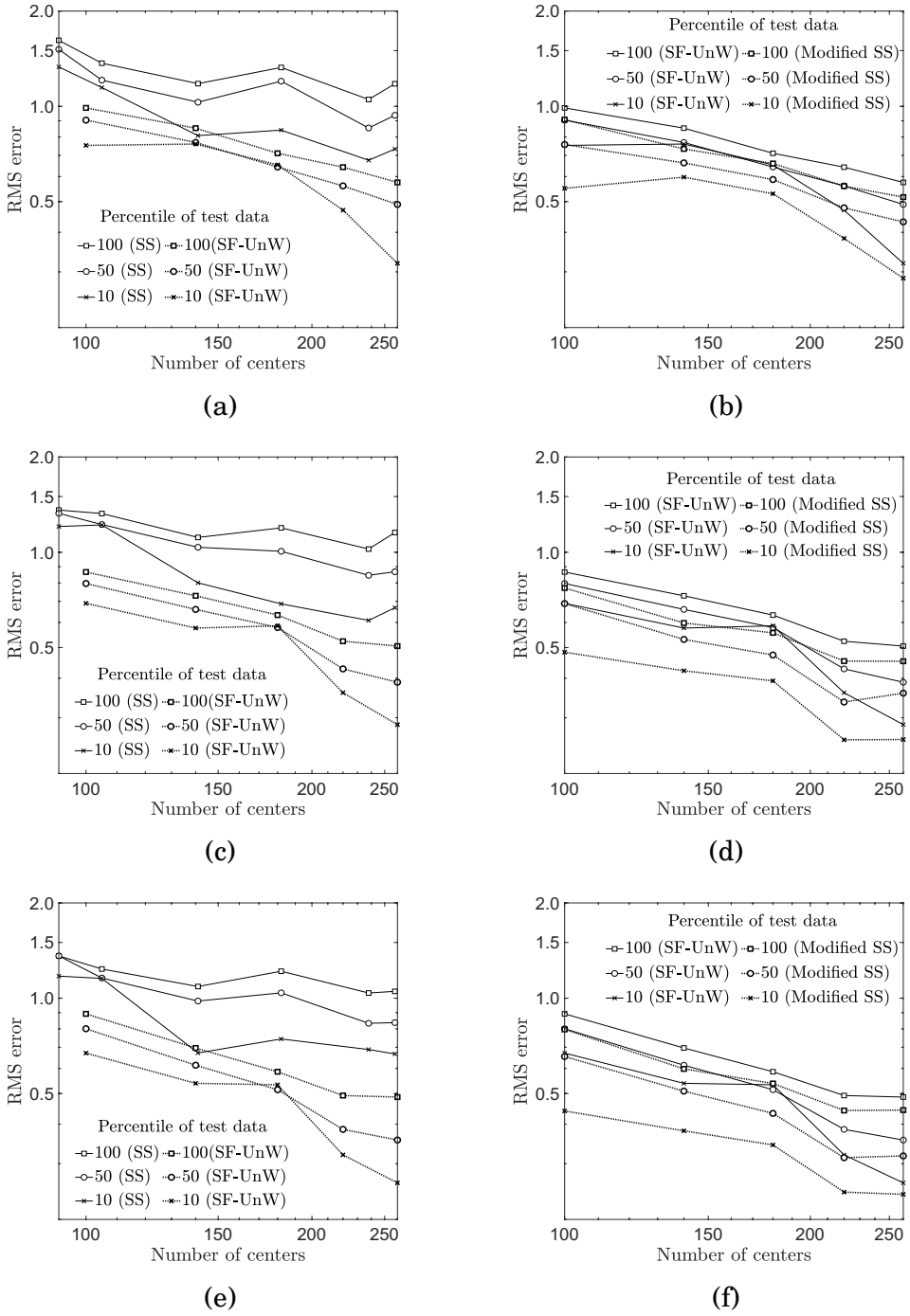


Figure 5.6: RMSE comparison between single sweep (SS) (solid lines) and partially modified single sweep (SF-UnW) (dashed lines) for (a) 10 ps, (c) 50 ps, (e) 100 ps MD simulations (on the left); RMSE comparison between partially modified single sweep (SF-UnW) (solid lines) and modified single-sweep (Modified SS) (dashed lines) for (b) 10 ps, (d) 50 ps, (f) 100 ps MD simulations (on the right)

maintain a straight peptide, the omega dihedrals were restrained to 180 using NAMD's Colvars package [187] with a force constant of 0.003 kcal/mol/deg^2. All MD simulations were performed with NAMD v2.12 [188] and visualized with VMD [189].

Figure 5.5a shows the contour plot of the solvated Alanine Dipeptide free energy landscape obtained by the inverse histogram method after running direct MD simulation for 1 μs. The blank region in white correspond to high energy configurations of the AD molecule which is never visited during the MD simulation. A meshgrid of 360×360 $(= 129600)$ input test points are generated in the ϕ-ψ CV space. Out of these generated points, free energy values at 90806 locations could only be obtained because of missing data in the high energy region which forms the true data set. 3 test data sets are derived from the true data set. The first test data set is the 100-percentile test data set which constitutes all of the true data set and corresponds to the colored region (non-grey) in the CV space as shown in figure 5.5b. The second test data set is the 50-percentile test data set which constitutes true values below the 50-th percentile and corresponds to the colored region (non-grey) in the CV space as shown in figure 5.5c. The third test data set is the 10-percentile test data set which constitutes true values below the 10-th percentile and corresponds to the colored region (non-grey) in the CV space as shown in figure 5.5d. Test data in the regions marked in grey in the CV space are ignored. TAMD generated

centers are used for the single sweep (SS) approach and reconstruction is performed on sample (center) sets of varying sizes obtained by varying the cutoff distance s_{min}. For the modified approaches, samples from the proposed design are used as centers for the reconstruction by increasing the number of samples from 100 to 260 in steps of 40 samples and the approximation quality of the potential surface is assessed for each of the sample set cases. Keeping in mind the stochasticity of the mean force values obtained from the MD simulations at each center, the entire procedure is repeated 5 times and the median root mean square errors (RMSE) at each sample set case over these 5 runs are reported here. Since the input CV domain $[-\pi, \pi]^2$ is periodic in this case with periodicity of 2π along each CV, the sampling design uses the periodic construction algorithm to generate the centers. It is again noted here that since the sampling design is sequential by construction, when moving from one sample set with M samples to the next sample set with $M + m$ centers, the M centers are common for both the sets. Conversely, with TAMD generated centers, overlap between point sets with varying numbers of centers may exist but is not guaranteed. Figure 5.5e shows the root mean squared error (RMSE) convergence plot of the reconstructed Alanine Dipeptide free energy values for each of the 3 test data sets using the unweighted and the proposed weighted reconstruction scheme when restrained MD simulations of duration 100 ps are performed at each center to get the mean force values.

Figure 5.6 shows the RMSE comparison between the single sweep (SS) method and the partially modified proposed method (SF-UnW) as well as the RMSE comparison between the partially modified proposed method (SF-UnW) and the proposed modified single sweep method. In SF-UnW, the space filled design is used instead of TAMD sweeping for center selection but the reconstruction is unweighted. In modified SS method, the space filled design is used along with a weighted reconstruction approach. Restrained MD simulations of duration 10 ps, 50 ps and 100 ps were performed at each center to obtain the mean forces. Figure 5.6a shows that the partially modified proposed method (SF-UnW) results are more accurate than the corresponding single sweep (SS) method results for all the 3 test data sets when the MD simulation duration is 10 ps. Similar trends are observed in figures 5.6c and 5.6e when the MD simulation durations are 50 ps and 100 ps respectively. Figure 5.6b shows that the RMSE values for the modified single sweep method are lower than the corresponding RMSE values for the partially modified proposed method (SF-UnW) results for all the 3 test data sets when the MD simulation duration is 10 ps. Similar trends are observed in figures 5.6d and 5.6f when the MD simulation durations are 50 ps and 100 ps respectively. Thus, from the figures, it can be concluded that the usage of space filled centers instead of TAMD swept centers improves the quality of the reconstructed free energy surface. In addition to that, if a weighted reconstruction is used, the results improve further for all

the 3 test data sets. Intuitively, it is expected (similar to the Müller potential example) that the results with the 100-th percentile test data set would be worse for the modified SS method compared to the SF-UnW method while the results improve for the 10-th and 50-th percentile test data sets. This is not the case with the AD example as we are basing the error off the true free energy values from a direct MD simulation which only samples relatively low energy regions. Thus the 100-th percentile data set is comprised of data which belong to relatively low energy regions, due to the absence of true values in the high energy regions from the direct MD simulation. Therefore, the weighted reconstruction produces more accurate results even with the 100-th percentile test data set in this case.

5.5 Discussion

In computing any free energy landscape, the major considerations are accuracy and computational cost. In the context of single-sweep reconstruction, this work proposes a secondary weighting reconstruction scheme which favors lower energy regions. These regions are typically the areas of interest where accuracy would be most important. In addition, due to the nature of high energy states, it is difficult to accurately measure the mean forces and error increases in these regions. The Müller potential example has showed that the

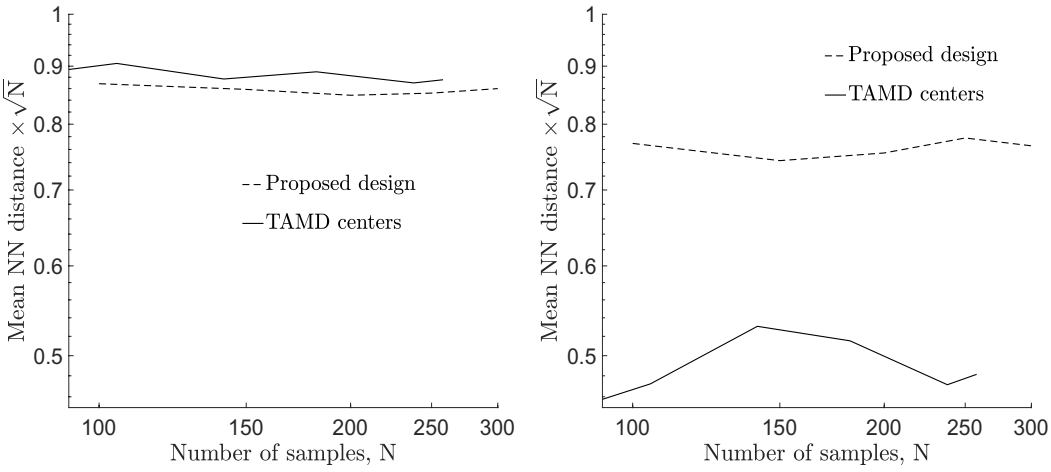


Figure 5.7: Space filling metric comparison of the proposed design and the TAMD centers in (a) the 2-dimensional domain and (b) the projected 1-dimensional domain along dimension 1

accuracy of the crucial low energy regions in the reconstructed free energy map increases if a weighted reconstruction is used instead of an unweighted reconstruction [30]. Centers are selected from the proposed space filling algorithm for the study and the comparison is essentially made between the partially modified single sweep (SF-UnW) approach and the fully modified single sweep (SS) approach. This is done to test the performance of the proposed weighted reconstruction approach.

In the solvated alanine dipeptide example, the quality of free energy reconstruction using TAMD swept centers is checked and compared with that using space-filled centers. In using the space-filling algorithm for point selection, the accuracy of the reconstructed free energy profile may be increased,

most notably at higher center counts. When 250 centers are used, the accuracy of the partially modified single-sweep with space-filling centers (SF-UnW) is much greater than TAMD single-sweep. In point selection through TAMD sweep, points are selected only from the frames in the trajectory. Thus, even though there is space filling in the 2-dimensional space, there is no control over the 1-dimensional space filling of the centers. This is evident from figure 5.7a where it is seen that the 2-dimensional space filling metric is slightly better for the TAMD center set. However, figure 5.7b reveals that the projected 1-dimensional space filling metric is much worse for the TAMD centers. The kernel matrix G [Eq. (5.13)] or G^w [Eq. (5.30)] is a function of the absolute difference between dimension-wise coordinates of the samples and the condition number of the matrix (for a fixed shape parameter) would be lower for a sample set with better projective properties leading to more stable solutions. The improvement in the reconstruction quality with the space filling design can thus be attributed to this enhanced projective property compared to the TAMD centers. In the calculated errors between TAMD single-sweep and partially modified single-sweep seen in figures 5.6a, 5.6c, and 5.6f, the accuracy of TAMD single-sweep increases disproportionately in the energy wells (lowest 10% of the free energy map) when 150 or more centers are probed. Error associated with partially modified single-sweep more proportionately scales with number of centers in the lower energy regions and this might be a byproduct

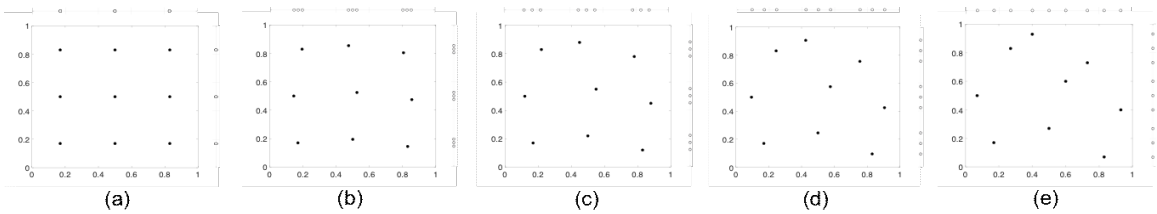


Figure 5.8: Sample point (black dots) sets with different values of (a) δ = 0, (b) δ = 0.025, (c) δ = 0.05, (d) δ = 0.075, and (e) δ = 0.1; X and Y projections of the samples for each case are on the right and top (white dots with black edges) of each figure respectively.

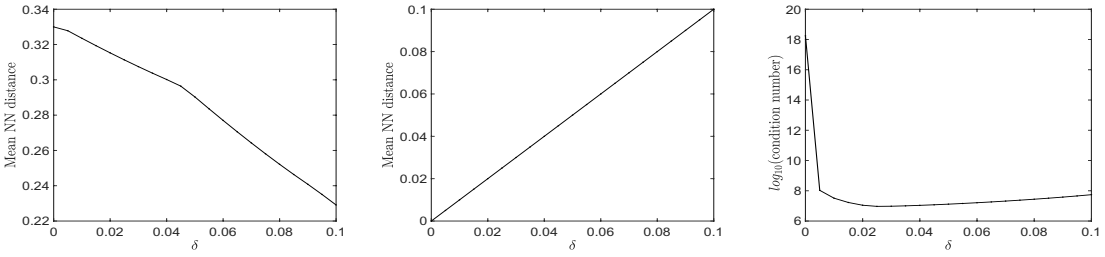
of the sequential nature of the design. With higher numbers of centers, we can ensure higher accuracy with the space-filling center algorithm as compared to TAMD trajectory sweep.

Table 5.1: Point coordinates as a function of δ

X:	0.17	0.17-δ	0.17+δ	0.5	0.5+δ	0.5-δ	0.83	0.83+δ	0.83-δ
Y:	0.17	0.5	0.83	0.17+δ	0.5+δ	0.83+δ	0.17-δ	0.5-δ	0.83-δ

The importance of good collapsive properties in a sampling design can be justified using the following example. Let us consider a set of 9 sample points in $[0, 1]^2$. The coordinates of the points are assumed to be a function of a scalar variable δ and is shown in Table 5.1. δ is varied between 0 and 0.1 in steps of 0.005 and for each case, a sampling arrangement is obtained. When $\delta = 0$, a 3×3 mesh grid of samples is obtained and corresponds to the sample set with the best space filling in 2-dimensions compared to sets obtained from any other δ value. However, it has the worst projective property.

Figure 5.8 shows sample sets for 5 different values of δ $(0, 0.025, 0.05, 0.075, 1)$ along with the 1-dimensional projections of each set on the the top (along di-



(a) (b) (c)

Figure 5.9: (a) Variation of the mean NN distance measure of the 2-dimensional sample sets with δ; (b) Variation of the mean NN distance measure of the 1-dimensional projections sample sets with δ; (c) Variation of the condition number of the kernel matrix with δ when the shape parameter $\sigma = 1$

mension 1 and right (along dimension 2) of the $[0, 1]^2$ domain. Increasing δ from 0 to 0.1 leads to consistent degradation in the mean NN distance in the 2-dimensional domain but consistent improvement in the mean NN distance in the projected 1-dimensional space which is seen in figures 5.9a and 5.9b respectively. Figure 5.9c shows that the condition number of G matrix [Eq. (5.13)] (for shape parameter equal to 1) decreases in the beginning and then starts to increase with increase in δ. Thus an ideal design should be one which has decent space filling property in the full domain as well as decent projective property in the lower dimensions. The proposed design seems to match that criteria better than the TAMD swept centers.

The RMSE comparison plots between partially modified single-sweep and modified single-sweep seen in figures 5.6b, 5.6d, and 5.6f, show how the space filled centers and the weighted reconstruction combine together to produce highly accurate lower energy spaces. The modified single sweep algorithm

shows increased performance most notably with fewer centers. Accurate results with as few as 100 centers can be obtained using the weighted reconstruction. The ability to use significantly less centers is desirable as it leads to higher computational efficiency.

It is important to ensure computational efficiency in these algorithms. TAMD single-sweep requires no a *priori* information about the system as it obtains information regarding CV restraints from the initial TAMD sweep. When using the space-filling algorithm, CV bounds must be selected prior to point selection. In order to obtain this information, the TAMD sweep may still be necessary to find the appropriate CV space if these bounds are not known, especially for non-periodic domains. The weighted reconstruction requires a *priori* information about the free energy landscape in order to obtain weighting factors. This algorithm would necessitate performing the reconstruction twice, once to obtain initial weights from the unweighted free energy landscape and the second to perform the weighting procedure. The computationally costly mean force calculations would not need to be rerun but the reconstruction phase would require extra computation. The weighted reconstruction can obtain similar accuracy as the unweighted reconstruction with significantly lesser number of centers, thus offsetting the cost associated with performing the reconstruction twice.

5.6 Conclusions

In this work, a more accurate version of the single sweep method for free energy reconstruction has been proposed. Modifications are proposed in both steps of the single sweep method. The use of a proposed sequential space filled design for center selection instead of the TAMD swept centers has led to an increase in accuracy of the free energy landscape. In addition to that, the implementation of a weighted reconstruction approach as an alternative to the unweighted reconstruction increase the accuracy of the free energy values in the crucial lower energy regions. The performance of the weighted reconstruction was tested using the Müller potential as an example and it was found that the accuracy of the crucial lower energy regions can be improved by such a scheme at the expense of the quality of the highest energy regions. The proposed space filled design is used for generating centers in the comparison above. The performance of the proposed concepts are then assessed during reconstruction of the free energy of alanine dipeptide (AD) in solution in two of its dihedral angles. Space filled centers generated from the proposed sampling design when combined with unweighted reconstruction (SF-UnW) is found to produce a more accurate reconstruction compared to the single sweep method. In addition to that, a weighted reconstruction combined with the space filled samples (modified SS) further improves the quality of the reconstruction in the

crucial lower energy regions, compared to the SF-UnW method. The increased efficiency observed in 2-dimensional CV space with relatively small number of centers shows promise for usage in higher dimensional CV spaces.

Chapter 6

A probabilistic modeling framework for composite plate models under projectile impact

This chapter considers an application of surrogate construction in the field of armor design and modeling, and deals with the development of a computational framework for generation of probabilistic penetration response of S-2 glass/SC-15 epoxy composite plates under ballistic impact. This involves the computationally feasible generation of the probabilistic velocity response (PVR) curve or the $V_0 - V_{100}$ curve as a function of the impact velocity, and the ballistic limit velocity prediction as a function of the model parameters. The PVR curve incorporates the variability of all the model parameters and

describes the probability of penetration of the plate as a function of impact velocity. The simulation model system considered here is that of a continuum single-layer plain weave S2/SC15 composite plate impacted by a 0.22 inch diameter steel right circular cylinder (RCC) projectile. The key goal here is to locate the surface of separation between rebound and penetration in an input space of many parameters. An adaptive domain-based decomposition and classification method, combined with sparse grid sampling is used to locate the critical region of separation. An assumption of monotonic behaviour of the output with respect to the model parameters, based on the physics of the problem, helps in reducing the number of model evaluations and makes the algorithm more efficient. The longitudinal tensile strength (LTS) and the punch shear strength (PSS) are considered as variable strength parameters to the model and the proposed methodology is found to efficiently capture the variability of the ballistic limit velocity with respect to the two strengths and also generate the PVR curve.

6.1 Introduction

In the field of armors design, performance evaluation of composite plates under ballistic impact in the presence of various sources of statistical variability is an ongoing topic of research. There is a need to explore high-dimensional

conceptual design space comprising material properties and weave architecture but the cost of prototyping such designs and experimentally characterizing the ballistic impact response is prohibitively high. Thus the alternative way is virtual armor design and testing which requires efficient computational frameworks capable of predicting the impact performance. The probabilistic velocity response (PVR) curve or $V_0 - V_{100}$ curve is of particular interest in this regard. V_X is defined by the velocity at which the probability of penetration is $X\%$. In simulations as well as experiments, the current state-of-the-art approach to generate the PVR curve is to use the Neyer D-Optimal sensitivity test method [190, 191]. In this method, the PVR curve is assumed to be the cumulative density function (cdf) of a normal distribution with mean and variance of the V_{50} impact velocity. The method requires as input, initial guesses for V_{50} variance and bounds of the V_{50} mean. The method is then used to guide the selection of the projectile impact velocities, such that the outcome of the previous impact velocity test (penetration or rebound) is used to determine the next impact velocity. Once all the impact velocities are obtained, maximum likelihood estimation (MLE) method is used to obtain the V_{50} mean and variance estimates, which finally generates the PVR curve. In essence, the whole curve is generated based on V_{50} mean and variance estimates, which might not be realistic. A lot of work has been done previously in generating the virtual PVR curve by mapping in different intrinsic (for example, yarn tensile strength)

and extrinsic (for example, projectile impact location) sources of experimentally characterized variability into fabric finite element (FE) models [192–196]. In some of these studies [195, 196], the numerical PVR curves have been experimentally validated and are shown to be in good agreement with the experimental PVR curve. It is, however, noted that the number of simulations or experiments performed to generate the PVR curves were only around 40. From the experimental perspective, considering the huge cost of performing each ballistic experiment, such an approach is understandable. But it might not be the true experimental PVR curve and we might see a change in the PVR curve if more experiments are performed. As far as the numerical PVR curves are concerned, in [196], the yarn tensile strength and yarn tensile modulus were sampled from a 274-dimensional sampling domain as there were 137 warp and 137 fill yarns. This might have worked in this special case, but the method might not be the ideal choice in general.

The current work proposes a more systematic approach for generating numerical PVR curve given a composite plate model. It tries to account for the statistical variability of the material properties by explicitly sampling in the design space of the variable parameters. It uses an adaptive domain-based decomposition and classification approach to converge on the surface of separation of the penetration and rebound impact behaviour and simultaneously running the impact model simulations in regions of the input parameter space

dictated by the algorithm. An upper and lower bound of the PVR curve is estimated analytically using the output label information of samples in the decomposed elements. A similar approach is also adopted for ballistic limit velocity predictions. With increase in each level of domain decomposition, it is found that the upper and lower bounds of the PVR curve as well as the ballistic limit estimates converge in increments.

This chapter is organized as follows: section 6.2 discusses the proposed methodology in details. In section 6.3, the method is applied to a single layer continuum plain weave S-2 glass/SC-15 epoxy composite plate model under ballistic impact by a steel projectile. Section 6.4 provides conclusions.

6.2 Methodology

The proposed methodology is a domain decomposition based adaptive classification approach. It involves sparse grid [63] sampling to generate samples in an element where the simulation model is run to obtain the corresponding outputs. A classification based criterion is then used to decide if the element needs to be further decomposed and if so, choose dimensions along which it should decompose.

6.2.1 Domain decomposition

Let us consider a d-dimensional hypercube input domain, assuming the simulation model is valid everywhere in that domain. If A_i and B_i denote the minimum and maximum bounds of the domain along dimension i $[i \in \{1, 2, \ldots, d\}]$, the domain can be denoted by $\Xi = [A_1, B_1] \times [A_2, B_2] \times \cdots \times [A_d, B_d]$. We further assume the decomposition of the domain takes place orthogonally along the dimensions. In a general orthogonal decomposition, if the domain Ξ is divided along the first d' $(1 \leq d' \leq d)$ dimensions and there are n_j divisions along dimension j $[j \in \{1, 2, \ldots, d'\}]$, then N non-overlapping and space-filling elements are formed where, $N = (n_1)(n_2)\ldots(n_{d'})$ and each element is denoted by Ξ_k: $\cup_{k=1}^{N} \Xi_k = \Xi$, $\Xi_m \cap \Xi_k = \emptyset$ for $m \neq k$ and $m, k \in [1, 2, \ldots, N]$. If a_i^k and b_i^k denote the minimum and maximum bounds of element Ξ_k along dimension i $[i \in \{1, 2, \ldots, d\}]$, Ξ_k is the tensor product given by

$$\Xi_k = [a_1^k, b_1^k) \times [a_2^k, b_2^k) \times \cdots \times [a_d^k, b_d^k). \tag{6.1}$$

For example, if a 3-dimensional hypercube is decomposed along dimensions 1 and 2 with 3 divisions for each dimensions, then $(3)(3) = 9$ elements are formed.

In this study, an element is divided into 2 sub-elements along only one of the dimensions. If subdivisions are performed along all dimensions, the number of sub-elements formed from each subdivision increases exponentially with in-

crease in dimensionality of the problem and the situation becomes intractable. This is called the 'curse of dimensionality' [197, 198]. A tessellation based approach of subdividing the domain is also not used here because of the same reason. The criterion to choose the dimension along which the element is subdivided, is discussed in section 6.2.3. A similar domain decomposition procedure has also been implemented in chapter 3 for the purpose of element level gPC approximation.

6.2.2 Domain-based classification

Classification is a supervised learning method for predicting labels for unseen new samples, based on the class labels of the training samples. Most of the popular classification methods, like, k-nearest neighbors (k-NN), support vector machines (SVM), artificial neural networks (ANN), etc., are designed to work with preassigned training data. In this study, data is generated by a sampling design of choice in the input parameter space and running the simulation model at those parameter values. The domain-based classification approach tries to take advantage of the structured sparse grid sample data in each element in order to classify the entire element space with the same label. Sparse grid samples are designed to be biased towards the edges of a domain. This, somewhat, helps in the 'same label' assumption for an entire element in the case where the existing sparse grid samples all have the same label. It is

precisely the combination of the sparse grid sampling and the domain decomposition which enables this type of classification. The domain-based classification is only used to classify regions with same type of label and is not capable of generating decision surfaces. It is used to basically converge to the region containing the decision surface.

6.2.3 Adaptive procedure

The core objective of the proposed methodology is to direct the sampling towards the region of the boundary of separation between penetration and rebound of the projectile. Sparse grid samples of level 1 are used for all the examples in this study. Before explaining the adaptive procedure, few terminologies are mentioned first. An element also implies the original domain. A sampled element refers to an element where sparse grid samples have been explicitly generated to obtain output labels from model evaluations, in addition to having evaluated samples present in that element from previous iterations. If a sampled element contains samples with only one type of output label (only rebound or only penetration), the entire element is classified with that label. This is the domain-based classification step and thus it is assumed that the surface of separation of the penetration and rebound will not pass through that element. On the other hand, when a sampled element contains samples with both types of output label (rebound and penetration), the surface of separation

of the penetration and rebound passes through that element and thus it needs to be resolved further. The unresolved element is split up equally into two sub-elements along a dimension which corresponds to the maximum edge length of the element. This is done to avoid biasing the samples along particularly sensitive dimensions. If the element edge length is found to be same along all dimensions, then the subdivision is done along a dimension such that one of the sub-elements have samples with least number (ideally none) of dissimilar output labels.

6.2.4 PVR curve estimation and ballistic limit predictions

For computing the PVR curve, the impact velocity is fixed at a particular value and the goal is to find out the fraction of penetration at that velocity. This is equivalent to calculating the fraction of penetration in the projected $(d-1)$-dimensional parameter space fixed at the velocity. An approach to generate the PVR curve is to estimate, at first, the surface of separation using a suitable classification algorithm with the binary training data. Then, $(d-1)$-dimensional uniformly distributed test samples can be generated at each impact velocity, and the fraction of the 'penetration' labels predicted over the total sample set denote the probability of penetration at that velocity. The process

can be repeated for all values in the impact velocity range to obtain the PVR curve. However, there is classification error as well as sampling error involved in this approach for estimating the PVR curve. An alternative approach uses domain-based classification method to analytically compute lower and upper bounds of the probability penetration values for each impact velocity of interest. It requires the decomposition of each unresolved element (element with dissimilar output labels) along all its dimensions. This leads to a possibility of generating sub-elements with same output labels (new resolved elements) such that the domain-based classification method can be applied there. Next, for each impact velocity value, all the d-dimensional elements that include the value are selected. Let us assume that the total hypervolume of the selected elements in the projected $(d-1)$-dimensional space (fixed at an impact velocity value) be denoted by V_{total}. Then, the total hypervolume (say, V_1) of the projected $(d-1)$-dimensional resolved elements with 'penetration' labels are calculated. In case of the unresolved elements, the entire element is given a 'rebound' label to estimate the lower bound and a 'penetration' label to estimate the upper bound. If the total hypervolume of the projected $(d-1)$-dimensional unresolved elements is denoted by V_2, then the probability of penetration at that impact velocity has a lower bound value of V_1/V_{total} and a upper bound value of $(V_1+V_2)/V_{total}$. This approach thus avoids the error in sampling in a potentially high-dimensional parameter space. However, an uncertain region

exists between the upper and lower PVR curves whose area reduces with further refinement of the parameter space.

A similar approach is used to obtain lower and upper bound estimates of the ballistic limit velocity predictions as a function of the material parameters. For each $(d-1)$-dimensional input parameter value, the d-dimensional unresolved elements that include the value is selected. The minimum lower bound value of the impact velocity among all the unresolved elements corresponds to the lower bound of the ballistic limit velocity and the maximum upper bound value of the impact velocity among those elements corresponds to the upper bound of the ballistic limit velocity. Thus, the uncertainty of the actual ballistic limit value is bounded by the minimum lower bound and maximum upper bound of the impact velocity values corresponding to the candidate unresolved elements.

6.2.5 Monotonicity constraints of parameters

The advantage of dealing with physics-based simulation models is that it is sometimes possible to obtain some initial knowledege about the behavior of the system under study. For the models used in this work, the likelihood of a 'penetration' outcome can be assumed to have a monotonicity constraint with respect to some parameters of the model. For example, with increase in impact velocity of the projectile, keeping all other input parameters constant, the likelihood of 'penetration' outcome increases. This is a case of monotonicity

constraint of increasing type.

Monotonic constraint knowledge is expressed in terms of dominance relation between ordered samples in the input space [199]. A sample is said to dominate another when each coordinate of the former is equal or greater than the corresponding coordinate of the latter. The class labels are also assumed to be ordered. For example, a student in an exam can get a grade of 'A', 'B' or 'C'. In a binary label output case, either of the label can be considered of lower value and the other of higher value. Thus, binary label outputs are ordinal by nature. Monotonicity constraints can be either increasing type or decreasing type. In the case of an increasing monotonicity constraint, the class label assigned to input samples should be equal or higher than the class labels assigned to the samples it dominates. In the case of a decreasing monotonicity constraint, the class label assigned to input samples should be the same or lower than the class labels assigned to the samples it dominates. As an example, consider a monotonicity constraint relating one input attribute and the target class. Keeping other attributes of the sample fixed, if the constraint is of increasing type, a sample with a higher value of the input attribute (dominant sample) should not be associated to a lower class value, and if the constraint is of decreasing type, a dominant sample should not be associated to a higher class value.

Following notations and definitions similar to [199], we consider d input

parameters with N ordered samples $\mathbf{x}^{(i)} \subseteq R^d$ and an associated class label $y^{(i)}$ $[i = 1, \ldots, N]$. C ordered labels are considered here such that $y^{(i)} \in \{1, \ldots, C\}$. The data set is denoted by $D = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})\}$. A dominance relation, $\succeq$, is defined as follows:

$$\mathbf{x} \succeq \mathbf{x}' \iff x_j \geq x_j^{'}, \ \ \forall j \in \{1, \ldots, d\}, \tag{6.2}$$

where x_j and $x_j^{'}$ are the j-th coordinates of samples x and x′, respectively. This means, x dominates x′, if each coordinate of x is not smaller than the corresponding coordinate of x′. Since the samples as well as the corresponding class labels are ordered, the elements in data set D are comparable among each other. Two examples z and z′ are identical if $z_j = z'_j, \forall j \in \{1, ..., d\}$, and they are non-identical if $\exists j$, such that $z_j \neq z_j^{'}$. A pair of elements in D, $(\mathbf{x}, y)$ and $(\mathbf{x}', y')$ is said to have a monotonic relation of increasing type if,

$$\mathbf{x} \succeq \mathbf{x}' \ \ \wedge \ \ \mathbf{x} \neq \mathbf{x}' \ \ \wedge \ \ y \geq y'$$

or

$$\mathbf{x} = \mathbf{x}' \ \ \wedge \ \ y = y'. \tag{6.3}$$

and a monotonic relation of decreasing type if,

$$\mathbf{x} \succeq \mathbf{x}' \quad \wedge \quad \mathbf{x} \neq \mathbf{x}' \quad \wedge \quad y \leq y'$$

or

$$\mathbf{x} = \mathbf{x}' \quad \wedge \quad y = y'. \tag{6.4}$$

If all possible pairs of elements in D are either monotone (increasing or decreasing) or incomparable, then data set D is also considered to be monotone. In a general case, a data set can be monotonic of increasing type for a subset of the input parameters, monotonic of decreasing type for another subset of parameters, and non-monotonic in the rest of the parameters. Let there are d_1 monotonically increasing parameters, d_2 monotonically decreasing parameters, and d_3 non-monotonic parameters, such that $d = d_1 + d_2 + d_3$. Then for a pair of non-identical elements in D, $(\mathbf{x}, y)$ and $(\mathbf{x}', y')$, condition for a general monotonic relation is given by,

$$\begin{aligned}
&\mathbf{x}_{(d_1)} \succeq \mathbf{x}'_{(d_1)} \quad \wedge \quad \mathbf{x}_{(d_1)} \neq \mathbf{x}'_{(d_1)} \quad \wedge \quad y \geq y' \\
&\quad \wedge \\
&\mathbf{x}_{(d_2)} \succeq \mathbf{x}'_{(d_2)} \quad \wedge \quad \mathbf{x}_{(d_2)} \neq \mathbf{x}'_{(d_2)} \quad \wedge \quad y \leq y' \\
&\quad \wedge \\
&\mathbf{x}_{(d_3)} = \mathbf{x}'_{(d_3)}
\end{aligned} \tag{6.5}$$

where, $\mathbf{x}_{(d_1)}$ is the d_1-dimensional subspace containing the increasing monotone parameters, $\mathbf{x}_{(d_2)}$ is the d_2-dimensional subspace containing the decreasing monotone parameters, and $\mathbf{x}_{(d_3)}$ is the d_3-dimensional subspace containing the non-monotone parameters.

In the examples considered in this study, the class labels are binary and hence ordered. The 'rebound' label is assumed to be the lower class label and 'penetration' label is considered to be the higher class label. There is assumption of monotonicity constraint of increasing type for the impact velocity parameter and monotonicity constraint of decreasing type for the strength parameters (longitudinal tensile strength and punch shear strength). Thus, given all other parameters are fixed, a sample with a higher value of impact velocity should not have a 'rebound' label, if the sample which it dominates has a 'penetration' label. Similarly, given all other parameters are fixed, a sample with a lower value of longitudinal tensile strength (or punch shear strength) should not have a 'penetration' label, if the sample which dominates it has a 'rebound' label. When samples are generated in new elements using the adaptive algorithm, the monotonicity criterion in Eq. (6.5) is implemented to save expensive simulations by cheaper approximation.

6.2.6 Numerical implementation

The algorithm has the following steps:

1. Select the number of variable parameters or input dimensions d and their corresponding ranges of interest.

2. Set the total number of iterations N_{iter}, and the minimum hyper-volume fraction of the non-converged elements V_{min} below which the subdivision into smaller elements is stopped.

3. Initialize $iter = 1$. Use sparse grid samples of level 1 to generate samples in the original input domain (element). Run simulations at the samples and note their output. The output considered here is binary: rebound or penetration of the projectile. If the final residual impact velocity is in the same direction as the impact velocity, the output is penetration. If they are in opposite directions, the output is rebound. If all the simulations have same output labels (all 'rebound' or all 'penetration'), the element is assumed to be resolved, and go to step 6. Otherwise, if the output label outcomes are dissimilar ('rebound' and 'penetration'), the element is said to be unresolved.

4. Subdivide each unresolved element into two sub-elements along the dimension which corresponds to the maximum element edge length. If the element edge lengths are same along all dimensions, then the subdivision is done along a dimension such that one of the sub-elements have samples with minimum number of dissimilar output labels.

5. Generate new samples in each of the new elements using sparse grid sampling of level 1. Remove duplicity of samples. Check if any of these new samples can be assigned a label based on the monotonicity constraint in Eq. (6.5). Run simulations with the parameter values corresponding to the newly created unlabeled samples. If an element now contains samples with the same type of output labels, it is assumed to have been resolved (domain-based classification).

6. At iteration '$iter$', generate upper and lower bounds of PVR curve as a function of the impact velocity as well as the ballistic limit velocity prediction as a function of the strength parameters as described in section 6.2.4.

7. Set $iter = iter + 1$. Calculate the cumulative hyper-volume V of all the unresolved sub-elements created and the number of iterations $iter$. Compare with the corresponding critical values V_{min} and N_{iter} respectively to check if either of the two stopping criteria is met. If met, terminate the algorithm. If either of the criteria is not met, go to step 4.

The flow chart for the entire algorithm is shown in figure 6.1. It is noted here that an optional monotonicity enforcement step can be implemented at the end of the algorithm before estimating the quantities of interest. The need of this implementation may arise because the simulation model behavior may

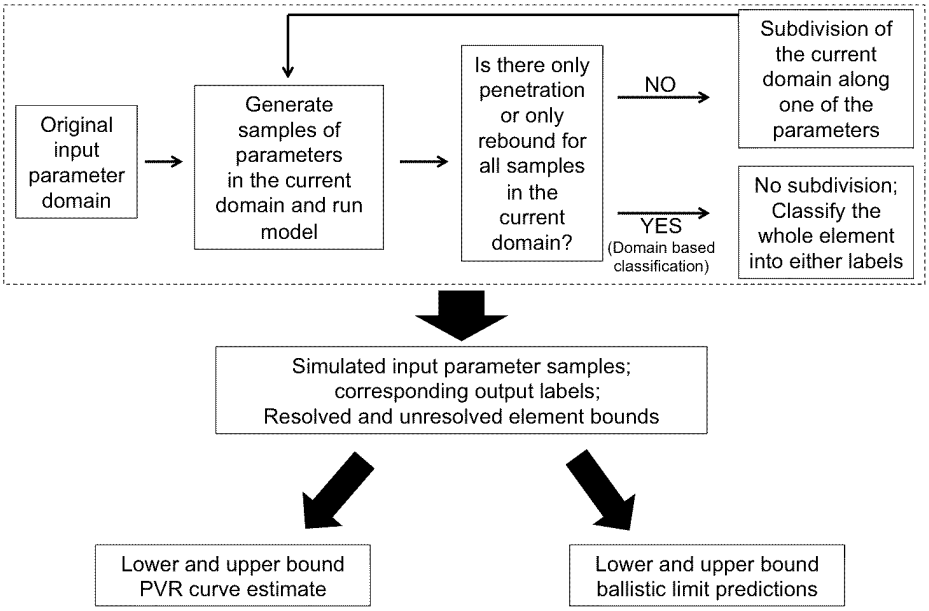


Figure 6.1: Flow chart of the algorithm steps for generating the PVR curve as a function of the impact velocity and predicting ballistic limit velocities as functions of the mode parameters.

,

become noisy near the decision boundary (surface of separation) separating the rebound and the penetration region. This means, if the model is run at input parameter combinations which correspond to regions in input parameter space close to the surface of separation, the outcome of the model may be such that it violates the general monotonic behavior and has the wrong label. A way to enforce monotonicity is to check the label of each sample and see if it violates the general monotonic behavior with respect to the other sample labels. If there is violation, the label is changed. This procedure may be repeated for multiple iterations over the entire sample set to resolve the monotonicity violation issue.

6.3 Results

In this section, the proposed framework is applied to a single layer plain weave S-2 glass/SC-15 epoxy continuum composite model impacted by a steel projectile. Impact velocity of the projectile is the extrinsic source of variability and the longitudinal tensile strength and punch shear strength of the plate are the two intrinsic sources of variability considered in this study. The first example is a 1-dimensional case which considers only the variation in the impact velocity of the projectile is considered and the strength parameters are fixed at their baseline values. The next example is a 2-dimensional problem which considers the additional variation of the longitudinal tensile strength. Finally a 3-dimensional example is also shown where the impact velocity, the longitudinal tensile strength, and the punch shear strength are all variable parameters for the simulation model. Two types of input distributions for the strength parameters in this study: uniform and normal.

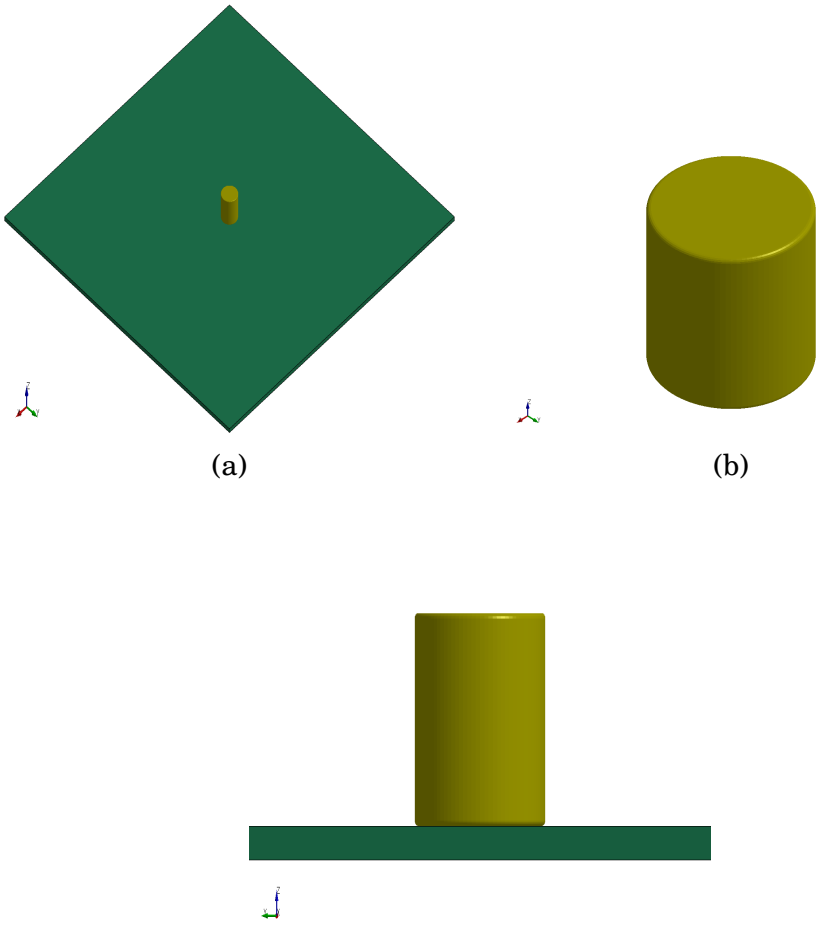
6.3.1 Simulation Model description

The simulation model used in this study is a single layer, continuum plain weave composite plate target as shown in figure 6.2. It is modeled in LS-DYNA [200] using hexahedral elements. The plate dimensions are 101.6 mm $\times$ 101.6 mm $\times$ 0.887 mm and it is subjected to clamped boundary conditions. The

plate model is impacted at the center by a rigid, right-circular, cylindrical elastic steel projectile having a diameter of 5.6 mm and height 6.1 mm. A slight fillet ($r \approx 0.15$ mm) is applied to projectile edges to reduce stress concentration. The damage in the plate is governed by the MAT-162 material model [201, 202]. It is a continuum damage model that models progressive failure in unidirectional and plain weave composite materials subjected to large strain rates and pressures (i.e., impact). The model uses a generalized Hashin failure criteria [203] and the damage progression is characterized by damage evolution laws proposed by Matzenmiller et al. [204]. The failure model simulates fiber failure, matrix damage, and delamination under modes I, II, and III loading. The damage model simulates material softening (degradation of stiffness matrix) following damage initiation (governed by failure model) when loaded beyond the failure strength parameters such as longitudinal tensile strength and punch-shear strength. Thus, varying these strength parameters controls initiation of material softening as well as the total energy dissipation prior to failure under impact loading conditions.

6.3.2 1-dimensional example

This section considers the impact velocity variation over the range $[80, 200]$ m/s, with the effective longitudinal tensile strength (LTS) and the punch shear strength (PSS) of the continuum level composite plate set at baseline strengths



(a) (b)

(c)

Figure 6.2: (a) Isometric view of the $4'' \times 4''$ composite plate model under impact by a RCC projectile, (b) RCC steel projectile, (c) Magnified front view of the continuum plate model

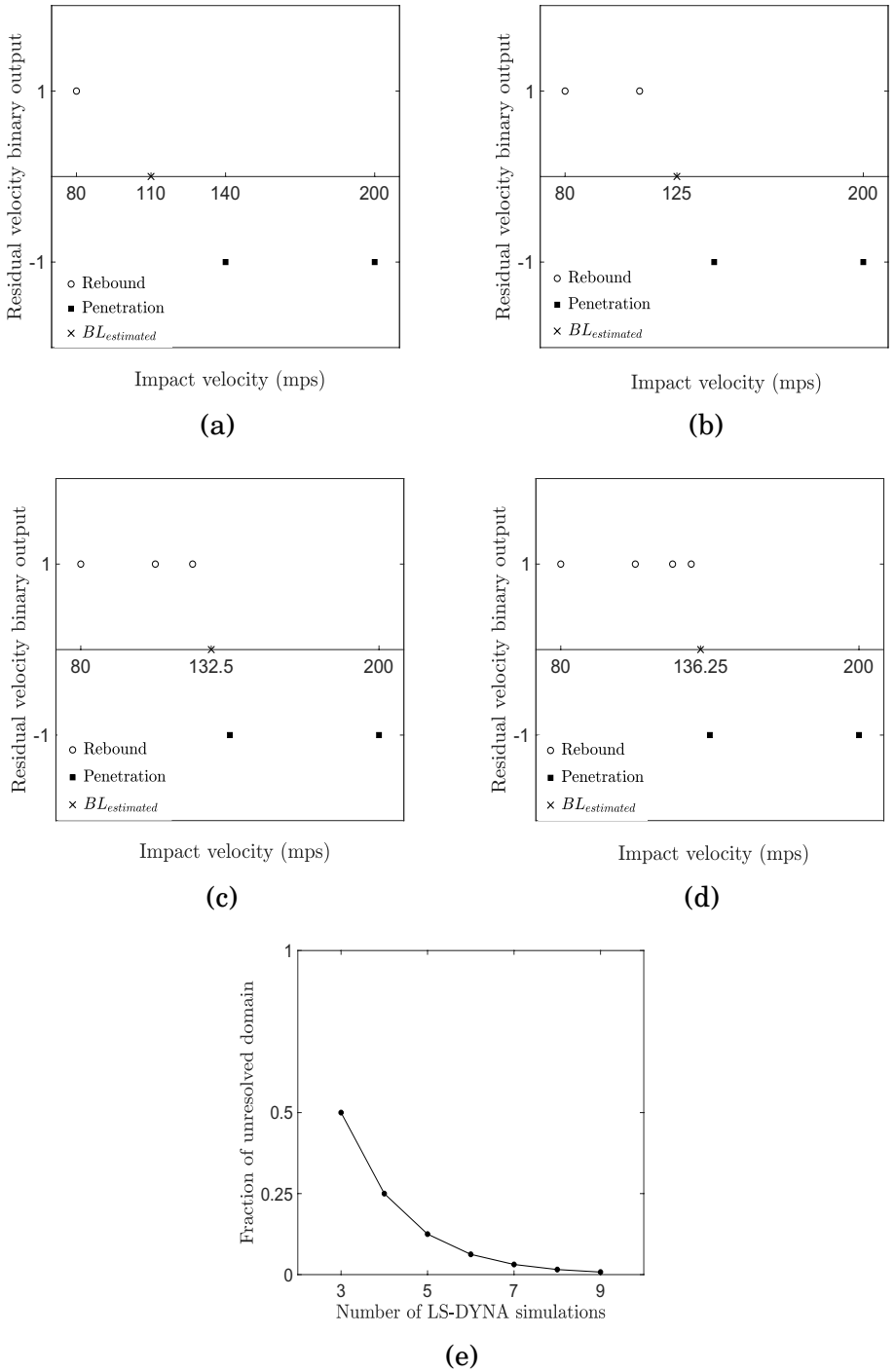


Figure 6.3: (a-d) Ballistic limit prediction with adaptive sampling of impact velocities: samples with 'rebound' outcomes are denoted by 'circle' markers, samples with 'penetration' outcomes are denoted by 'square' markers, ballistic limit mean estimate is denoted by 'cross' markers; (e) Reduction in the fraction of the unresolved space with increase in the number of LS-DYNA simulations.

of 1100 MPa and 300 MPa respectively. Since there is no variability in the strength parameters, it is not possible to generate a PVR curve in this case. The only relevant quantity of interest that can be extracted from this scenario is the ballistic limit velocity at the baseline strengths. This case is more like a demonstration of the domain-based decomposition and classification method in 1-dimensions to obtain the ballistic limit velocity at the given strengths. Figures 6.3(a-d) show the evolution of the mean ballistic limit velocity estimate with increase in iterations of the algorithm. After starting with 3 initial samples as seen in figure 6.3a, 1 sample is added at each subsequent iteration of the algorithm as seen in 6.3(b-d). In this 1-dimensional case, the domain decomposition is very similar to the bisection method. The unresolved domain is the region in the 1-d impact velocity space bounded by samples with dissimilar labels ('rebound' label on one side and 'penetration' label on the other), and the ballistic limit is considered to lie within the unresolved domain. With each new simulation, the fraction of unresolved 1-d domain gets reduced by 50% as seen in figure 6.3e. For example, in 6.3a, the unresolved 1-d space is between the impact velocity values of 80 and 140 m/s, which is a half of the entire impact velocity domain. With addition of the 1 more sample with 'rebound' label at 110 m/s, the unresolved space shifts to values of 110 and 140 m/s, which is a quarter of the entire impact velocity domain. It is to be noted here that the any value of the impact velocity in the unresolved 1-dimensional space could

be a potential candidate for the ballistic limit estimation for a given number of samples. For this particular case, candidate of choice is the mean.

6.3.3 2-dimensional example

In this section, the effective longitudinal tensile strength (LTS) of the continuum level composite plate is considered as a variable parameter over the range $[600, 1600]$ MPa. The impact velocity variation is again considered over the range $[80, 200]$ m/s. From the methodology perspective, this is now a 2-dimensional problem with impact velocity and the longitudinal tensile strength as the two parameters. It is of interest to generate the PVR curve by taking into account the variability of the tensile strength parameter, and assess the variation of the ballistic limit with respect to the strength parameter variation. Figures 6.4a, 6.4c and 6.4e show the evolution of sample in the input space using the domain decomposition method with increase in the number of iterations of the algorithm. The input domain can be roughly divided into three regions: rebound region, penetration region, and transition region separating the penetration and rebound region. It is of primary interest to resolve the different regions in the entire input domain in an efficient manner in order to estimate the relevant quantities of interest. The key to resolving the entire domain is to accurately locate the transition region. It is seen that the adaptive algorithm biases sampling towards the transition region. Figures 6.4b, 6.4d and

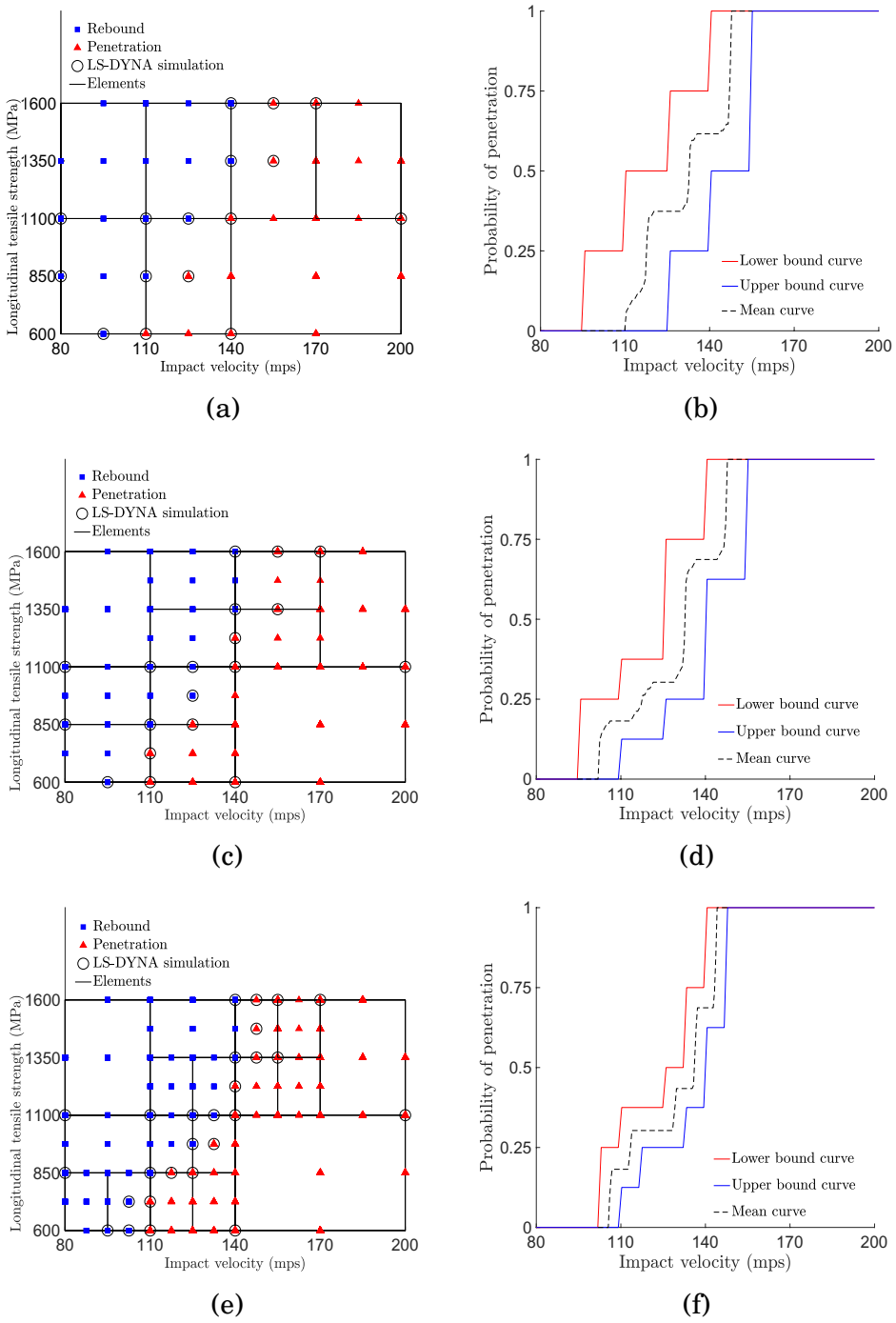


Figure 6.4: (a, c, e): Adaptive sampling and domain decomposition in the 2-dimensional input domain of impact velocity and longitudinal tensile strength with increase in the number of LS-DYNA simulations from (a) 16 to (c) 19 to (e) 27; (b, d, f): Evolution in the lower, upper and mean PVR curves using (b) 16, (d) 19 and (f) 27 LS-DYNA simulations; longitudinal tensile strength is assumed follow a uniform distribution.

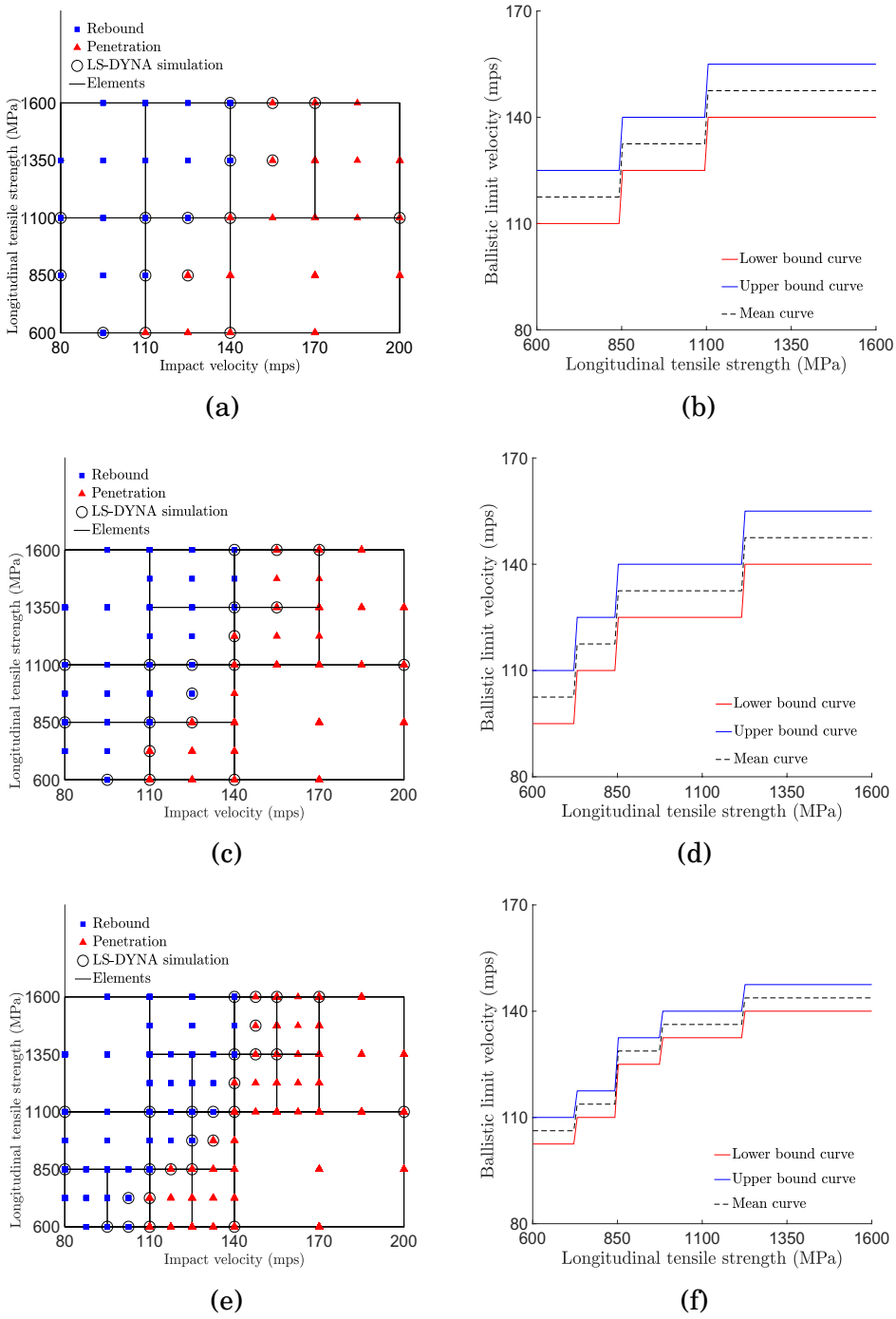


Figure 6.5: (a, c, e): Adaptive sampling and domain decomposition in the 2-dimensional input domain of impact velocity and longitudinal tensile strength with increase in the number of LS-DYNA simulations from (a) 16 to (c) 19 to (e) 27; (b, d, f): Evolution in the lower, upper and mean ballistic limit velocity curves using (b) 16, (d) 19 and (f) 27 LS-DYNA simulations; longitudinal tensile strength is assumed follow a uniform distribution.

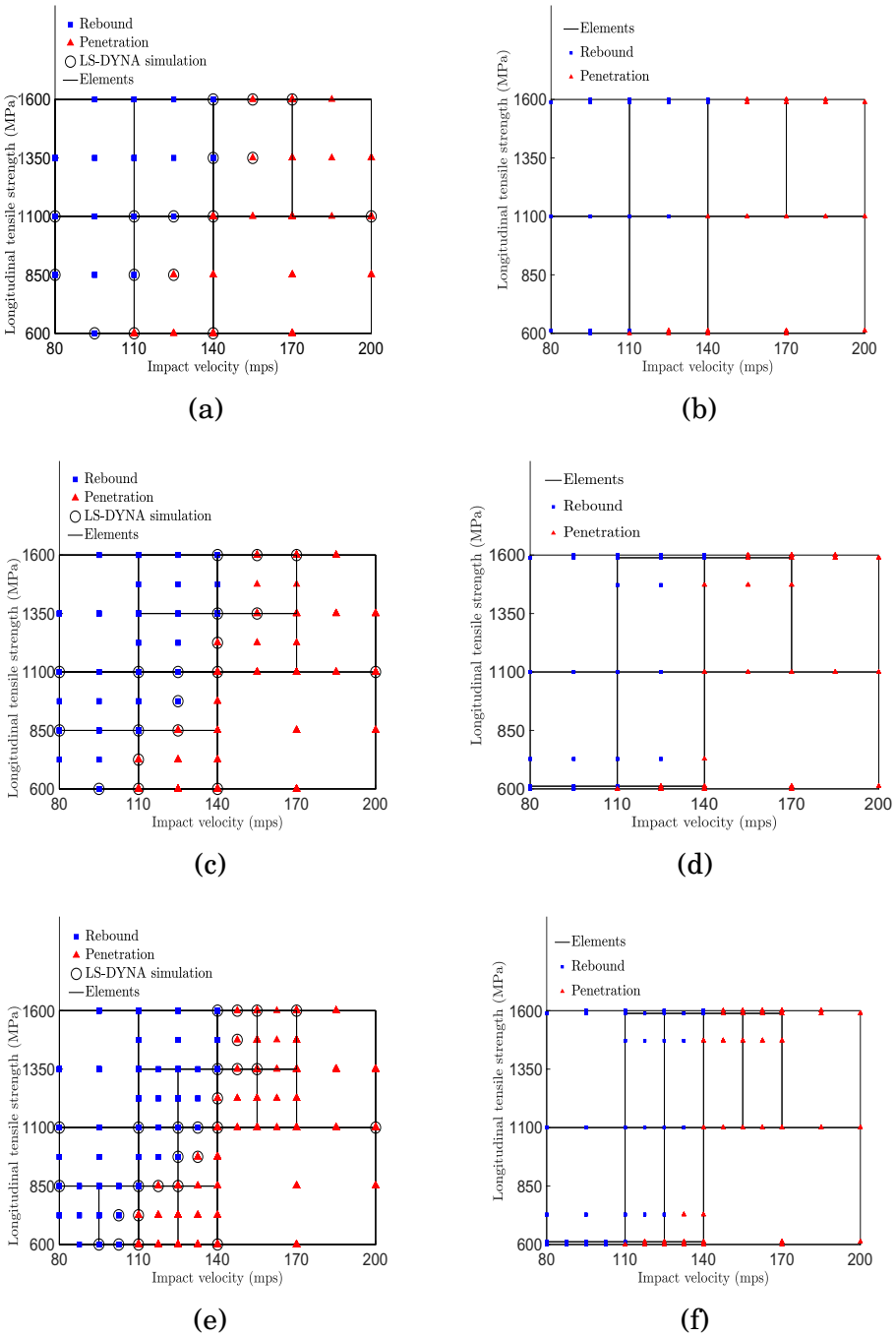


Figure 6.6: (a, c, e): Adaptive sampling and domain decomposition in the 2-dimensional input domain of impact velocity and normally distributed longitudinal tensile strength with increase in the number of LS-DYNA simulations from (a) 16 to (c) 19 to (e) 27; longitudinal tensile strength is assumed to follow a normal distribution with $\mu = 1100$ MPa and $\sigma = 110$ MPa; (b, d, f): Transformation of samples and elements to an uniform domain involving (b) 16, (d) 19 and (f) 27 LS-DYNA simulations.

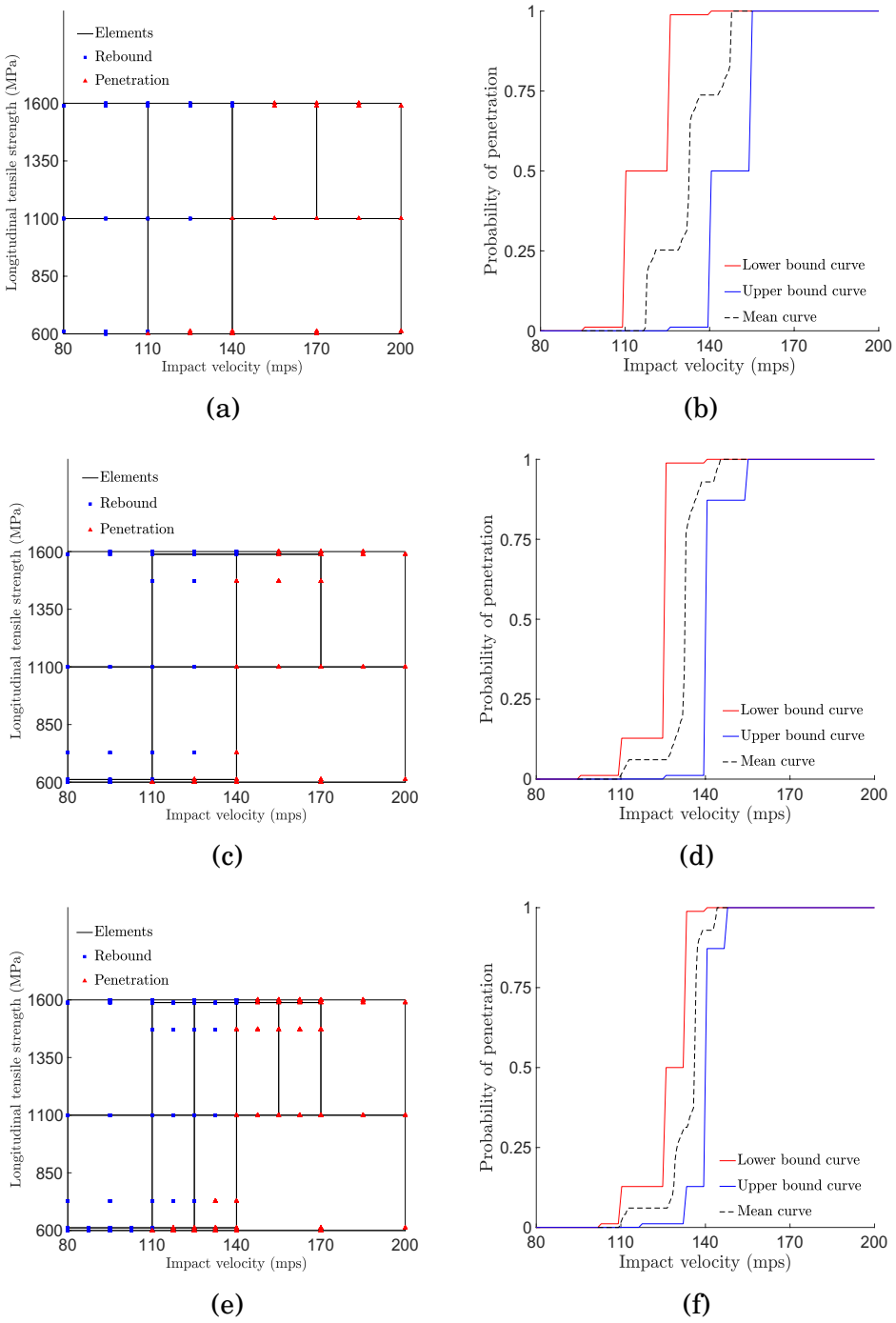


Figure 6.7: (a, c, e): Adaptive samples and elements in a transformed uniform 2-dimensional domain involving (b) 16, (d) 19 and (f) 27 LS-DYNA simulations, where longitudinal tensile strength is normally distributed; (b, d, f): Evolution in the lower, upper and mean PVR curves using (b) 16, (d) 19 and (f) 27 LS-DYNA simulations where longitudinal tensile strength is normally distributed.

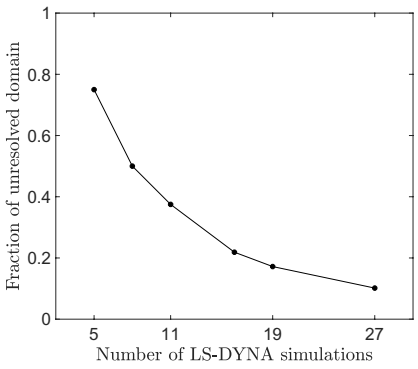


Figure 6.8: Evolution of the fraction of unresolved domain for the 2-dimensional problem where the impact velocity of the projectile and the longitudinal tensile strength of the plate are the variable parameters

6.4f show the corresponding lower bound, upper bound and mean PVR curves assuming that the longitudinal tensile strength parameter follows a uniform distribution. It is observed that the area between the lower bound and upper bound PVR curves decreases with increase in the number of sample evaluations from 16 in figure 6.4b to 27 in figure 6.4f. The space between the upper bound and lower bound PVR curve gives a measure of the overall uncertainty of the PVR curve estimate and it is a function of the unresolved space in the input domain. Addition of more samples using the adaptive algorithm helps in resolving more space in the input domain which helps in closing down the gap between the two PVR curves and reducing the overall uncertainty.

Figures 6.5a, 6.5c and 6.5e, same as in figures 6.4a, 6.4c and 6.4e, show the evolution of sample in the input space using the domain decomposition method with increase in the number of iterations of the algorithm. Figures 6.5b, 6.5d and 6.5f show the corresponding lower bound, upper bound and mean ballistic

limit velocity curves. It is observed that the area between the lower bound and upper bound ballistic limit curves decreases with increase in the number of sample evaluations from 16 in figure 6.5b to 27 in figure 6.5f. As observed with the PVR curves, addition of more samples using the adaptive algorithm helps in resolving more space in the input domain which helps in closing down the gap between the two ballistic limit curves.

Figures 6.6a, 6.6c and 6.6e, same as in figures 6.5a, 6.5c and 6.5e, show the evolution of sample in the input space using the domain decomposition method with increase in the number of iterations of the algorithm, but in this case, the longitudinal tensile strength parameter is assumed to be normally distributed with mean $\mu = 1100$ MPa and standard deviation $\sigma = 110$ MPa. Figures 6.6b, 6.6d and 6.6f show the corresponding samples and elements after transformation of the tensile strength coordinates from the gaussian (normal) domain to the uniform domain. The transformation is performed using the 'normcdf()' matlab [205] function which transforms the original sample coordinates in the gaussian space to $[0, 1]$-probability space using the mean and standard deviation values, and it is then linearly transformed back to the original range.

Figures 6.7a, 6.7c and 6.7e, same as in figures 6.6b, 6.6d and 6.6f, show the samples and elements in a transformed uniform domain when the longitudinal tensile strength is normally distributed. Figures 6.7b, 6.7d and 6.7f show the corresponding lower bound, upper bound and mean PVR curves assuming that

the longitudinal tensile strength parameter follows a gaussian (normal) distribution. As observed in the uniform distribution case, addition of more samples using the adaptive algorithm helps in resolving more space in the input domain which helps in closing down the gap between the two PVR curves.

Comparing the PVR curves in the uniformly distributed tensile strength case (uniform PVR curves) and the normally distributed tensile strength case (normal PVR curves), it is seen that the normal distribution has a stretching effect on the curves. It looks as if the probability of penetration (PoP) values greater than 0.5 get magnified to values close to 1, while PoP values less than 0.5 get reduced to values close to 0, when the underlying distribution of the strength changes from uniform to gaussian. This happens because the strength coordinates of the samples as well as the elements under a transformation from the gaussian domain to the uniform domain are biased towards the upper and lower bounds. Figure 6.8 shows the reduction in the fraction of the unresolved domain with increase in the number of LS-DYNA simulations. The fraction of the unresolved domain is measured by the total hypervolume of the unresolved (unclassified) elements of the finest resolution obtained after subdivision of the larger unresolved elements (remaining at the end of each iteration) along all dimensions.

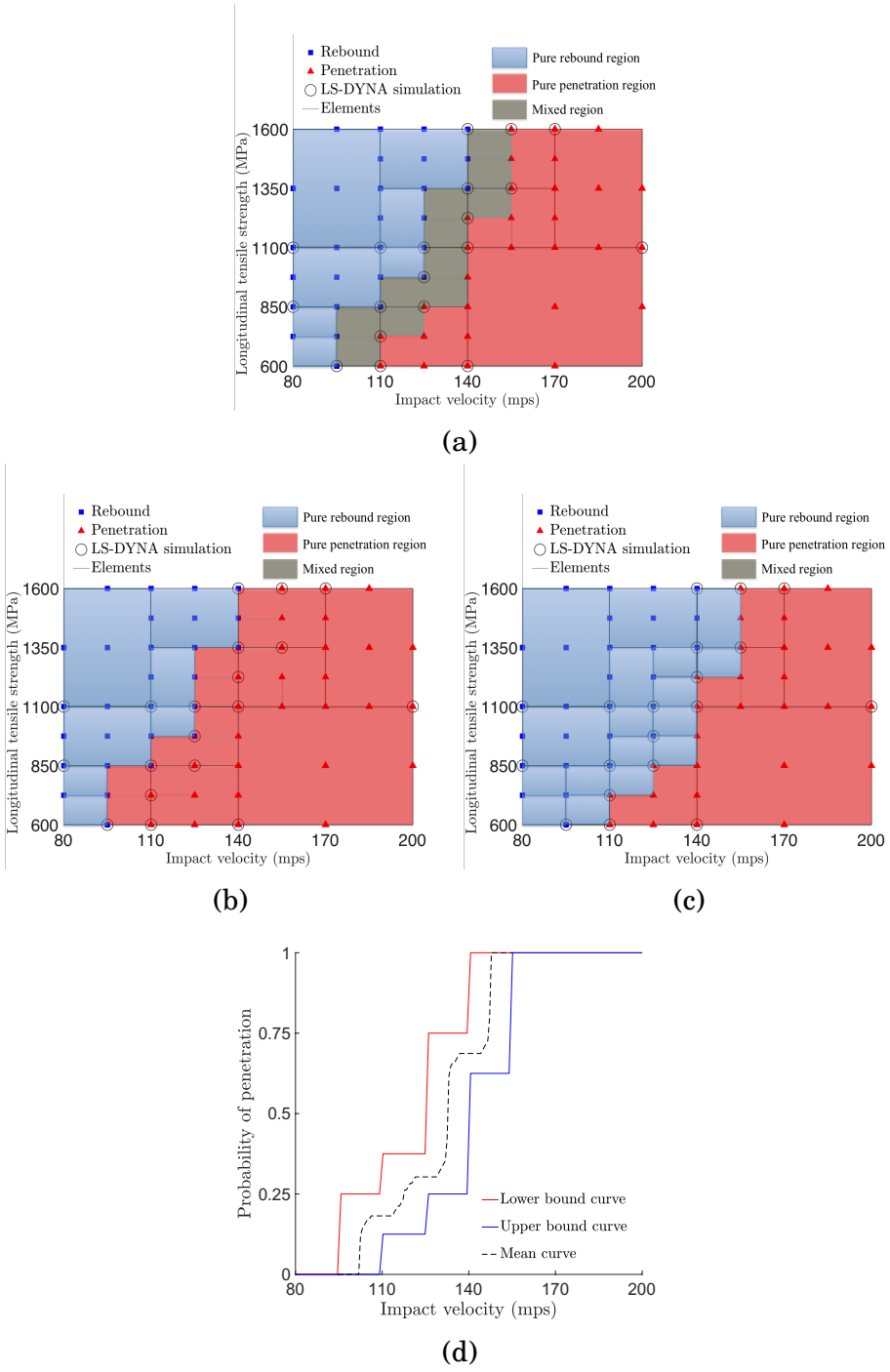


Figure 6.9: Schematic demonstration of lower bound and upper bound PVR curve generation; (a) shows the pure rebound region in blue, the pure penetration region in red and the mixed region in grey; (b) shows the mixed region as part of the penetration region which leads to the red lower bound PVR curve in (d); (c) shows the mixed region as part of the rebound region which leads to the blue upper bound PVR curve in (d).

6.3.4 Illustration of procedure for PVR curve estimation and ballistic limit prediction

Figures 6.9(a-d) illustrates the procedure for generation of the lower bound and upper bound PVR curve for the 2-dimensional case. Figure 6.9a shows the classification of the 2-dimensional input parameter space into 3 regions: pure rebound region, the pure penetration region and the mixed region. This classification is obtained by the adaptive domain-based decomposition and classification capability of the proposed algorithm. Based on the results from 19 LS-DYNA simulations, the true surface (curve) of separation between the rebound and penetration phenomenon lies somewhere in the grey mixed region. The bounds of the mixed region correspond to edges of decomposed elements and hence known. The mixed region bounds can thus be used to estimate the lower and upper bounds of the PVR curve which will then bound the true PVR curve. Figure 6.9b explains the procedure to generate the lower bound PVR curve. In this case, the entire mixed region is approximated as the penetration region. Thus, all the unresolved elements is resolved by assigning a 'penetration' label to the existing samples in the elements and using domain-based classification to classify all the elements into the penetrating region. Once that is done, the exact boundary between the rebound and penetration region is now exactly known and the lower bound PVR curve (shown as a red line in figure 6.9d) can

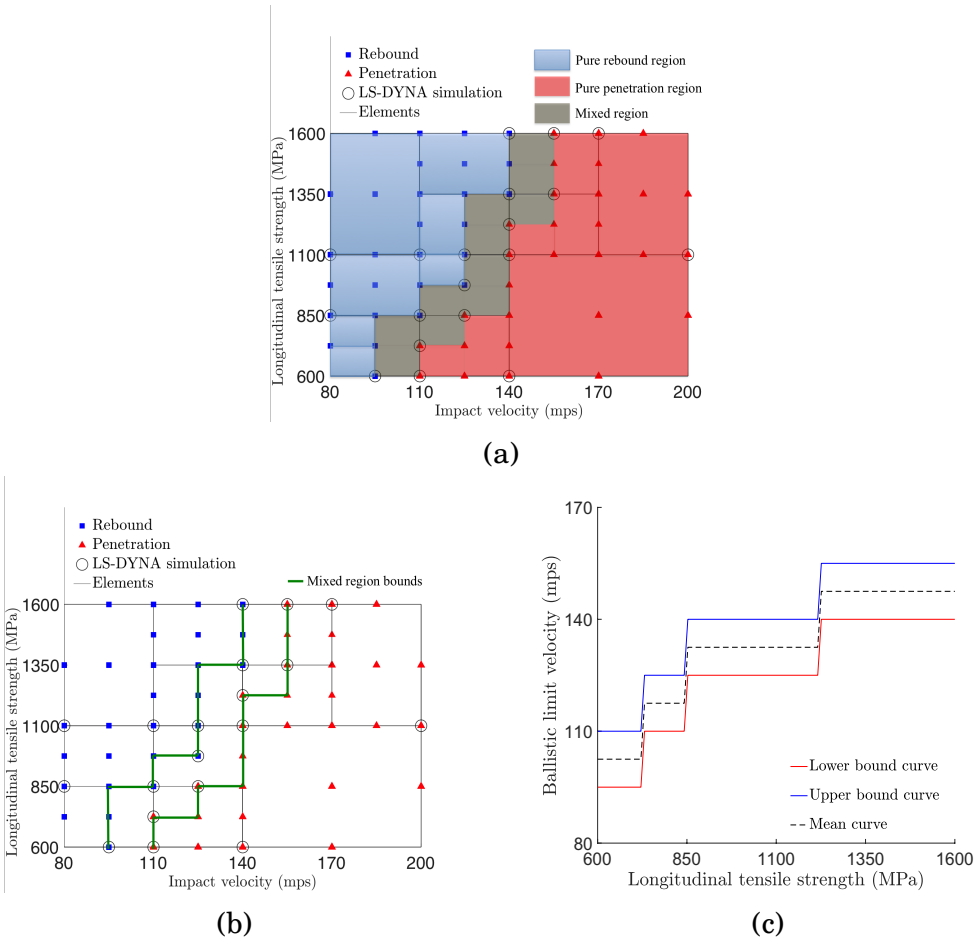


Figure 6.10: Schematic description of the ballistic limit velocity prediction as a function of the tensile strength parameter; (a) shows the pure rebound region in blue, the pure penetration region in red and the mixed region in grey; (b) shows the lower bound (on the left) and upper bound (on the right) of the mixed region in solid green lines; (c) shows the lower bound prediction of the ballistic limit as a function of the strength parameter (in red solid lines), the upper bound prediction of the ballistic limit (in blue solid lines), and the mean prediction (in black dotted lines).

be analytically calculated as explained in section 6.2.4. Figure 6.9c explains the procedure to generate the upper bound PVR curve. In this case, the entire mixed region is approximated as part of the rebound region. The unresolved elements are thus resolved by assigning a 'rebound' label to the existing samples in the elements and classified into the rebound region. Then, the exact boundary between the rebound and penetration region is again exactly known and the upper bound PVR curve (shown as a blue line in figure 6.9d) can be analytically calculated as explained in section 6.2.4. The mean PVR curve shown as a black dotted line in figure 6.9d is obtained by implementing the k-nearest neighbor classification using Chebychev distance, assuming that the red PVR curve and blue PVR curve are data with two different class labels. The mean curve is basically the k-NN decision surface.

Figure 6.10 explains schematically the generation of the lower bound and upper values of the ballistic limit velocity for the 2-dimensional case. Figure 6.10a shows the 3 classified regions: pure rebound region, the pure penetration region and the mixed region, in the 2-dimensional input parameter space. Ballistic limit velocity is the velocity at or around which the outcome of the impact changes from 'rebound' to 'penetration', or 'penetration' to 'rebound', for a fixed value of other material parameters. As can be seen from figure 6.10a, the transition regions spans at atleast one element for any particular strength value. Thus, it is intuitive to calculate the lower and upper bounds of the ballistic

limit velocity. Figure 6.10b shows the upper and lower bounds of the mixed region (in solid green lines). These bounds can be easily located as they coincide with the decomposed elements in the domain obtained from the algorithm. After obtaining the mixed region bounds, a value of the strength parameter is chosen and the impact velocity values at which it intersects the mixed region bounds, correspond to the ballistic limit lower and upper bounds. 6.10c shows the corresponding curve obtained for the entire range of strength parameter values. The mean ballistic limit curve is obtained by taking the average of the lower and upper bound curve.

6.3.5 3-dimensional example

In this section, in addition to the effective longitudinal tensile strength (LTS), the effective punch shear strength (PSS) of the continuum level composite plate is also considered as a variable parameter over the range $[100, 500]$ MPa. The effective longitudinal tensile strength (LTS) is again assumed to vary over the range $[600, 1600]$ MPa, and the impact velocity over the range $[80, 200]$ m/s. This is now a 3-dimensional problem with impact velocity, the longitudinal tensile strength and the punch shear strength as the three variable parameters. The objective here is to generate the PVR curve by taking into account the variability of the tensile strength parameter as well as the punch shear strength, and assess the variation of the ballistic limit with re-

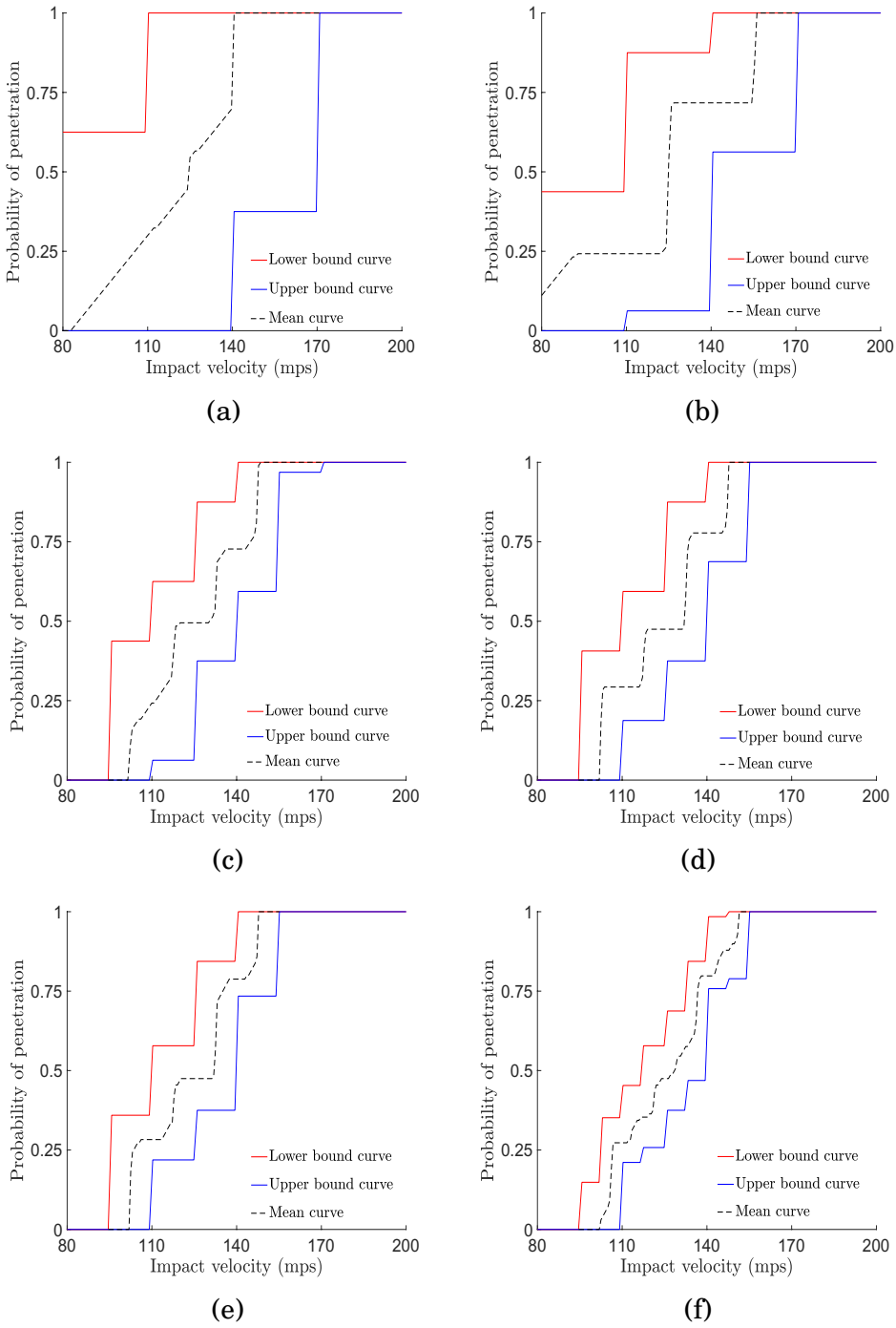


Figure 6.11: Generation of PVR curves as a function of the impact velocity of the projectile using (a) 20, (b) 27, (c) 43, (d) 56, (e) 67 and (f) 104 LS-DYNA simulations; the longitudinal tensile strength and the punch shear strength parameters are assumed to follow an uniform distribution in their respective ranges.

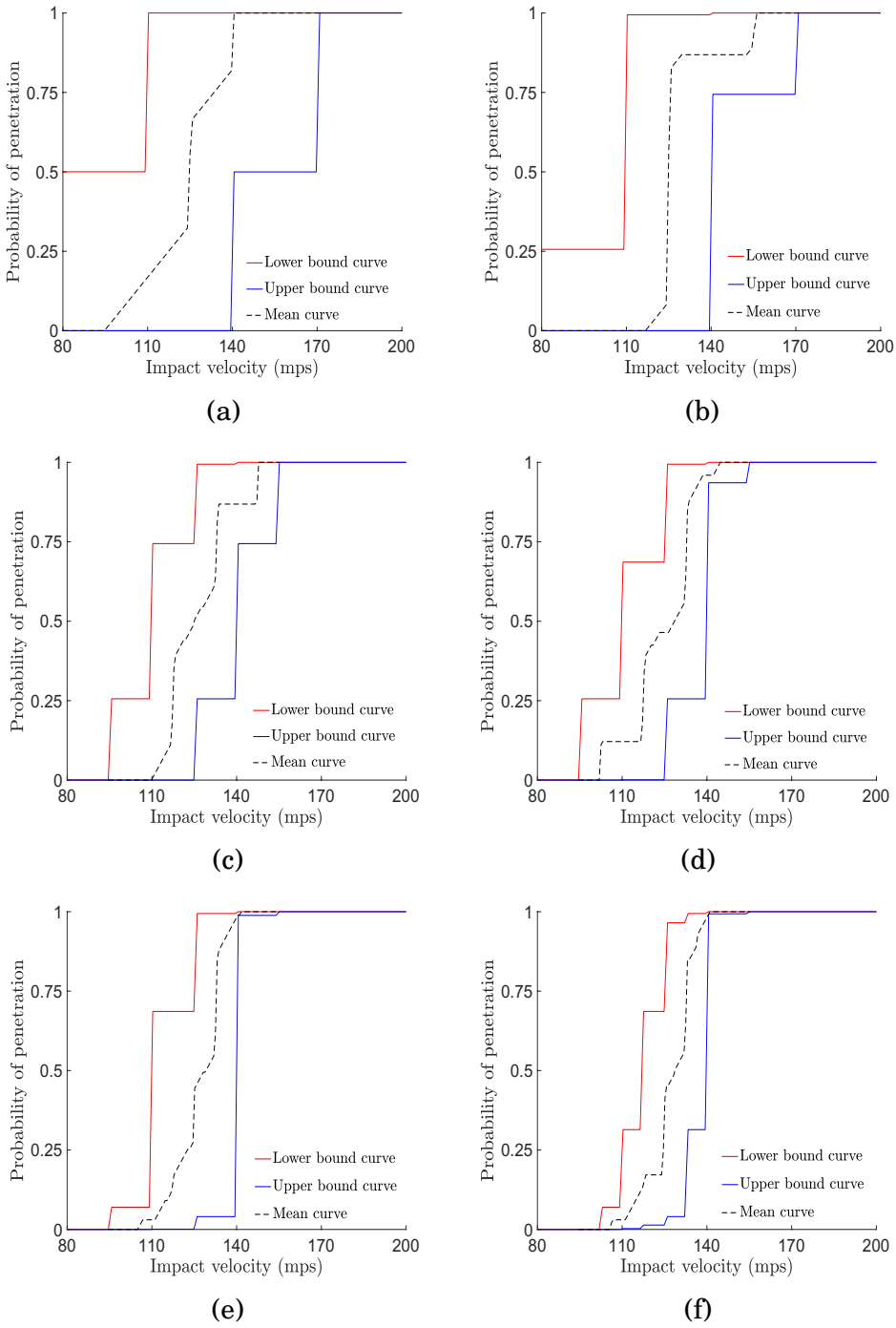


Figure 6.12: Generation of PVR curves as a function of the impact velocity of the projectile using (a) 20, (b) 27, (c) 43, (d) 56, (e) 67 and (f) 104 LS-DYNA simulations; the longitudinal tensile strength and the punch shear strength parameters are assumed to follow independent normal distributions with $\mu = 1100$ MPa, $\sigma = 110$ MPa and $\mu = 300$ MPa, $\sigma = 30$ MPa respectively.

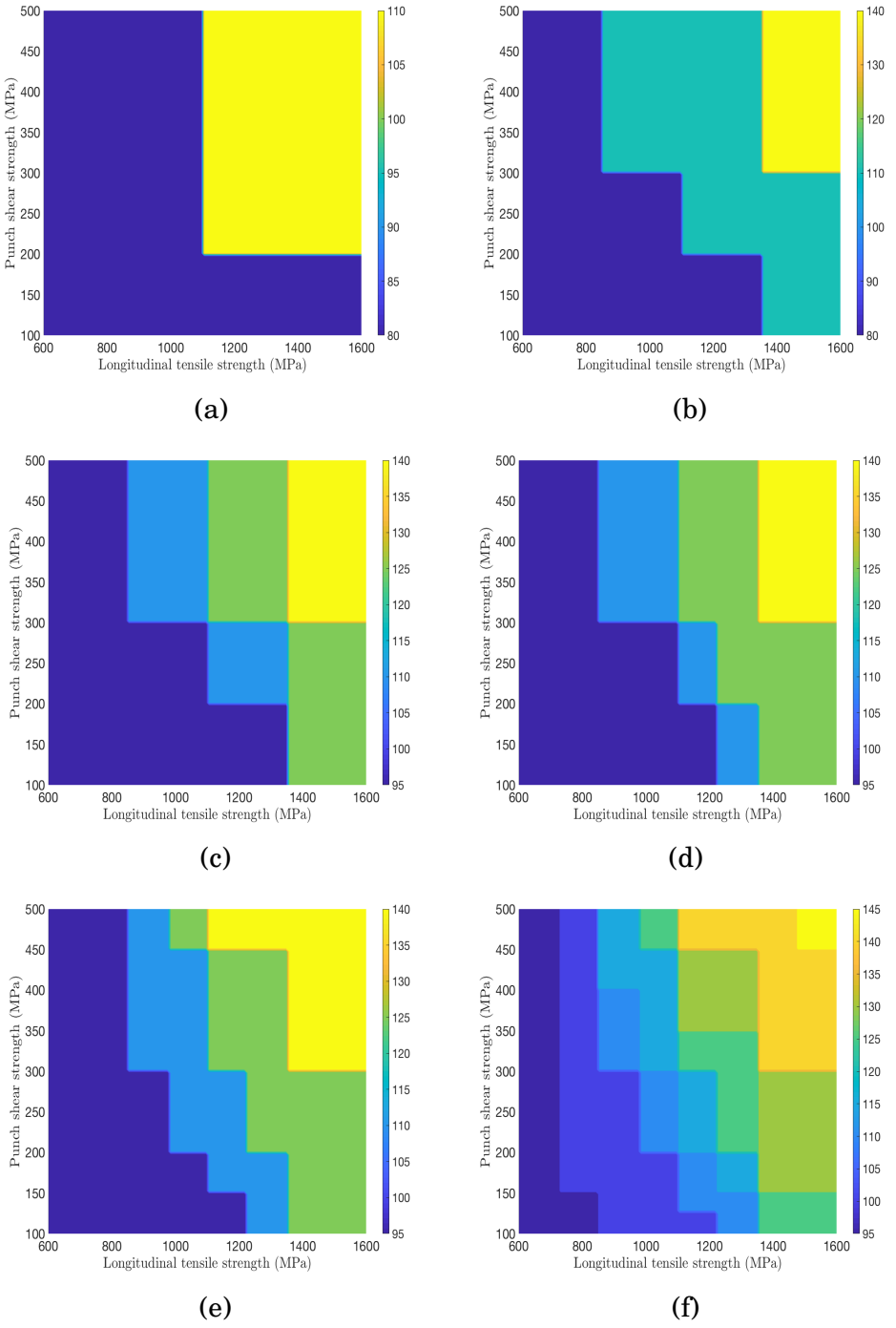
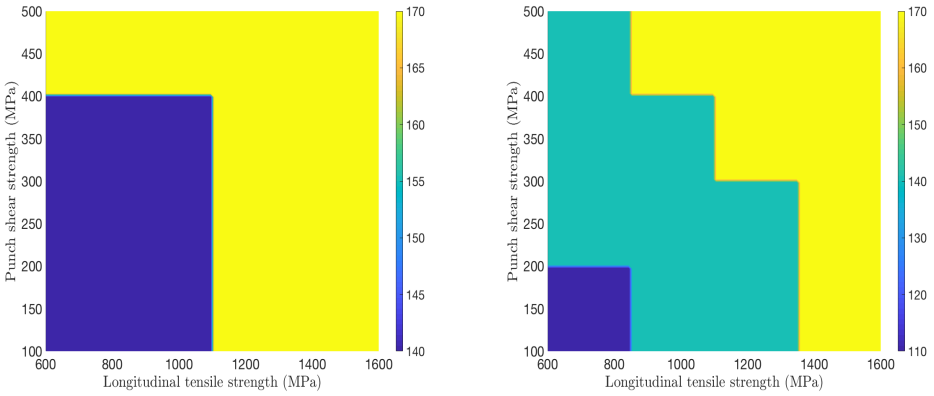
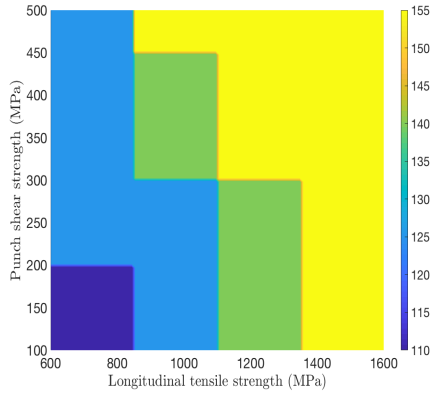


Figure 6.13: Evolution of the lower bound contour of the ballistic limit velocity as a function of the longitudinal tensile strength and punch shear strength using (a) 20, (b) 27, (c) 43, (d) 56, (e) 67 and (f) 104 LS-DYNA simulations.

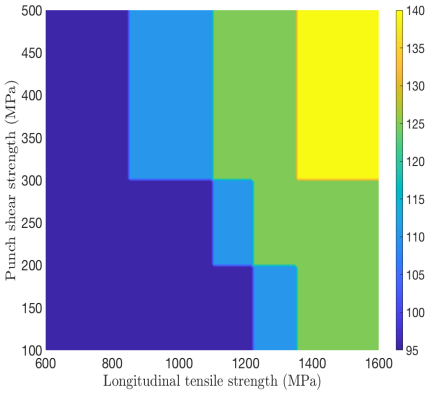


(a)

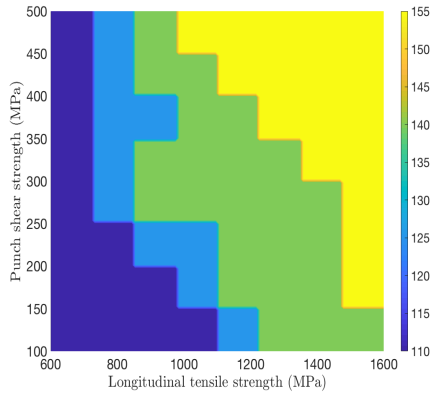
(b)



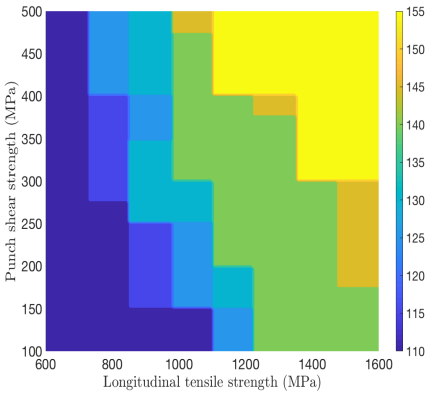
(c)



(d)



(e)



(f)

Figure 6.14: Evolution of the upper bound contour of the ballistic limit velocity as a function of the longitudinal tensile strength and punch shear strength using (a) 20, (b) 27, (c) 43, (d) 56, (e) 67 and (f) 104 LS-DYNA simulations.

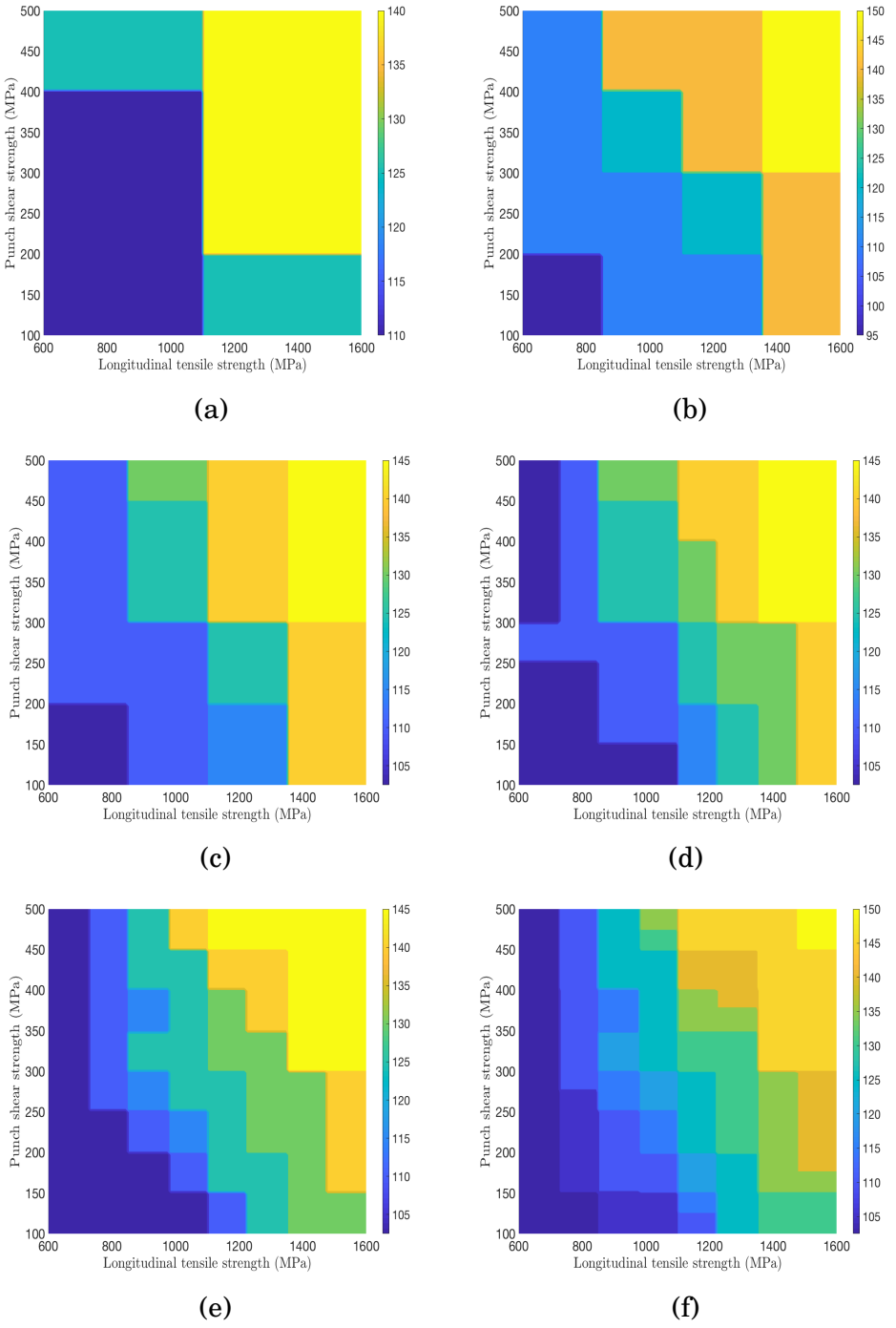


Figure 6.15: Evolution of the mean contour of the ballistic limit velocity as a function of the longitudinal tensile strength and punch shear strength using (a) 20, (b) 27, (c) 43, (d) 56, (e) 67 and (f) 104 LS-DYNA simulations.

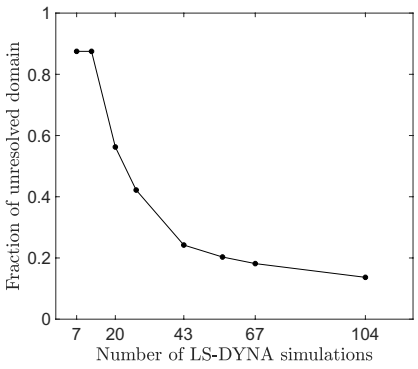


Figure 6.16: Evolution of the fraction of unresolved domain for the 3-dimensional problem where the impact velocity of the projectile, and the longitudinal tensile strength and punch shear strength of the plate are the variable parameters

spect to the variation in the strength parameters. Figure 6.11 shows the evolution of the PVR curves with increase in the number of LS-DYNA simulations. As observed in the 2-dimensional case, addition of more samples using the adaptive algorithm helps in reducing the width between the lower bound and upper bound PVR curves. In this case, the longitudinal tensile strength and the punch shear strength parameters are assumed to follow an uniform distribution in $[600, 1100]$ MPa and $[300, 500]$ MPa respectively. Figure 6.12 shows the corresponding evolution of the PVR curves when the longitudinal tensile strength and the punch shear strength parameters are now assumed to follow independent normal distributions with mean $\mu = 1100$ MPa and standard deviation $\sigma = 110$ MPa for the longitudinal tensile strength and mean $\mu = 300$ MPa and standard deviation $\sigma = 30$ MPa for the punch shear strength. Next, results describing the variation of the ballistic limit velocity with respect to

changes in both the strength parameters, the longitudinal tensile strength and the punch shear strength, are shown. Figure 6.13(a-f) shows the evolution of the lower bound (LB) contour plot of the ballistic limit velocity with increase in the number of LS-DYNA simulations and figure 6.14(a-f) shows the evolution of the upper bound (UB) contour plot of the ballistic limit velocity with increase in the number of LS-DYNA simulations. The evolution of the mean ballistic limit contour is shown in figure 6.15(a-f) for increasing number of LS-DYNA simulations. As seen in all the three figures, more and more finer details about the ballistic limit variation get captured with increase in the number of simulations. It is also seen that the ballistic limit values (in the mean, LB or UB contours) has an increasing trend with respect to both the strength parameters. A closer look at the ballistic limit values of the lower and upper bound contours reveals that the two contours converge closer to each other with increase in the number of LS-DYNA simulations, which signifies reduction in the uncertainty in the ballistic limit estimates. It is noted here that the new simulations should essentially be run at sample (parameter) values which correspond to important regions in the parameter space. This will help in extracting maximum information out of a new simulation.

6.4 Conclusions

The state-of-the-art method for estimating the PVR curve works in the 1-dimensional impact velocity space and all the different sources variability are implicitly included in it. The proposed methodology, on the other hand, explicitly accounts for the different variability and works in a high-dimensional space which includes parameters associated with different sources of variability. Although this involves more computational cost, it is much more robust approach with the capability of extracting much more reliable and useful information from a computational impact model, for example, ballistic limit velocity bound predictions at different model parameter values. The computational framework in this study efficiently characterizes the probabilistic response of a continuum plain weave composite plate with different sources of variability under projectile impact. As seen from the results, the methodology allows for an efficient determination of the lower and upper bounds of the PVR curve. It also captures the variation of the ballistic limit velocity bounds as a function of each of the different sources of variability which also helps in assessing the sensitivity of the ballistic limit towards these parameters. The current state-of-the-art method, on the other hand, estimates the PVR curve based on V_{50} mean and variance estimates by assuming a Gaussian distribution. The methodology is also shown to efficiently handle non-uniform marginal distributions of

strength parameters, like, gaussian, without any sampling based approach.

Chapter 7

Conclusions and future work

7.1 Summary

The work presented here proposes different surrogate modeling strategies to deal with a range of black box simulation models related to applications in the fields of mechanics, along with the demonstration of the algorithm performances. The developed algorithms, because of their non-intrusive nature, can be easily extended to simulation models pertaining to other fields of application. For example, if a certain simulation model is stochastic/deterministic in nature providing only the output gradients and the goal is to build a surrogate function of the output, then the modified single sweep approach presented in chapter 5 can be the surrogate modeling algorithm of choice.

The two adaptive surrogate modeling algorithms developed in chapters 2

and 3 are applicable to problems where the simulation models provide only the information about the output corresponding to inputs (and no information about output gradients) and the outputs are continuous (or real valued). In chapter 2, an efficient algorithm E-ASGC is developed which improves upon the existing adaptive sparse grid collocation (ASGC) interpolation method in tracking down discontinuities. It uses cubic spline based approximations of input samples in smoother regions to reduce the number of model evaluations and has been successfully applied to a stochastic elliptic problem, truss problem, composite model with interface damage, etc. The E-ASGC method, being an interpolation approach, is applied to deterministic black-box models. Chapter 3 presents a multi-element non-intrusive second-order PCE method, SCAMR, which is basically a regression approach. An efficient criterion for checking higher dimensional interactions has been proposed which can help in achieving dimensionality reduction. Problems with dimensionality as high as 500 has been successfully tackled using this approach. The SCAMR, being a regression approach, can be applied to both deterministic and stochastic black-box models.

Chapter 4 then considers the surrogate modeling of a type of simulation model which provides both output as well output sensitivity information. The simulation model is an IGFEM-based fiber-reinforced matrix composite model under uniaxial strain loading. This work is a performance study of surrogates

built using different combinations of existing sampling designs and surrogate methods. It is found that the efficiency of the surrogate construction using the sensitivity information outperforms the additional cost of extracting the sensitivity infromation from the model.

Chapter 5 considers a scenario where the simulation model provides only the output sensitivity information and no output information at all, and the goal is to estimate the quantity of interest, which is a function of the output. The simulation model considered here is stochastic in nature. It is a restrained MD simulation model of solvated alanine dipeptide molecule which provides the mean forces (negative gradient of free energy) in the collective variable (input) space. The objective is to efficiently construct the free energy landscape. The surrogate algorithm developed is a modified single sweep approach using a weighted reconstruction scheme and a sequential space filling design with improved space filling properties.

Finally, in chapter 6, a composite model under ballistic impact is considered where a classification surrogate is built for an efficient calculation of important probabilistic quantities like the probabilistic velocity response (PVR) curve and ballistic limit velocities. This is a case where the simulation model is deterministic and provides outputs that are discrete.

7.2 Future work

The work presented here provides several potential opportunities for future research in each of the topics discussed.

In chapter 2, the input parameters are assumed to follow uniform distribution as evident from the mean and variance analytical expressions in Eqs. (2.18) and (2.23). The way to tackle non-uniform input distribution in the current state is to construct the surrogate assuming a uniform distribution of the parameters, then sample the input parameter space according to the non-uniform distributions they follow and finally extract the corresponding approximate surrogate outputs. The outputs can then be used to estimate the relevant quantity of interest. A future work can consider the analytical estimation of the mean and variance for non-uniform distributions with correlation information. It has also been shown [206, 207] that selection of input points by considering the probability structure of the input domain can lead to efficient sampling. Similar efforts can be made to further increase the efficiency of the E-ASGC method by incorporating the non-uniform distribution information of the parameters in its formulation.

In chapter 3, the input parameters were assumed to be uniform. An extension can be made to incorporate arbitrary distributions for the input parameters and thus generalize the approach. Also, in the subdivision step of

the algorithm, an element is split into two equal parts along each critical dimension. An increase in the convergence of the algorithm can be potentially achieved by coming up with a criterion that enables the subdivision at an arbitrary location along a critical dimension and not necessarily at the midpoint. The interaction check step is performed at the beginning of the algorithm on the original domain to reduce the dimensionality of the problem. It would be interesting to study the performance of the SCAMR algorithm when the interaction check is applied to the new elements formed by the decomposition with a goal to achieve further dimension reduction in them.

The work in chapter 4 can also be extended to a large scale problem with a large (of the order of 1000) number of fibers if an efficient dimension reduction scheme (like an active subspace) can be introduced. It may also be of interest to consider the use of local surrogate models to tackle the discontinuous behaviour of the response surface for $\delta_c = 25$ nm.

The restrained MD simulation models considered in chapter 5, when run for different simulation durations, provide solutions of different fidelity (longer it runs for, higher the fidelity of the solution). A potential extension of this work can be to implement a multi-fidelity approach with a optimal combination of high-fidelity and low-fidelity models to generate free energy landscapes with similar accuracy but of much lower computational cost.

A direct extension of the current work with the continuum level composite

plate model in chapter 6 is to implement the algorithm by introducing more number of variable input parameters in the study. Another future goal is to replace the continuum plate model with a meso-scale plate model with explicit woven architecture of the fiber and the matrix. The meso-scale model is more realistic but more expensive to solve which can a potential challenge for efficient surrogate construction.

Appendix A

Chapter 2 appendix

For a valid probability space represented by $(\Omega, \mathcal{F}, \mathcal{P})$, following three axioms should always be satisfied:

1. $0 \leq \mathcal{P}(E) \leq 1, \forall E \in \mathcal{F}$ implying any event E in the probability space has non-negative probability of occurrence not exceeding 1.

2. $\mathcal{P}(\Omega) = 1$, means that any elementary event occurring will always lie in the sample space Ω and never outside it.

3. For a countable number of events $E_1, E_2, E_3, ... \in \mathcal{F}$ and $E_i \cap E_j = \phi$, $\forall i \neq j$, $\mathcal{P}(\cup_{i=1}^{\infty} E_i) = \sum_{i=1}^{\infty} \mathcal{P}(E_i)$

We can represent any physical system in the form of differential equation as:

$$L(u; x, \omega) = f(x, \omega), \forall x \in D \text{ and } \forall \omega \in \Omega \tag{A.1}$$

Theoretically, Ω is infinitely dimensional. Thus the first step in numerically solving such stochastic differential equations is to make the stochastic space finite dimensional. This is done by representing the inherent randomness in the input parameters by a finite number of random variables. For example, if a certain random input is a stochastic process in space or time, then the index domain (spatial or temporal) needs to be discretized into a finite number of independent random variables. This obviously is an approximation but to make the stochastic system feasible for numerical computation, it is necessary. If the random process is Gaussian, K-L expansion can be suitably used to represent the process by a set of uncorrelated Gaussian random variables. This scenario works fine because of the property that uncorrelated Gaussian random variables are also independent. This is not usually the situation while dealing with a non-Gaussian process where lack of correlation does not necessarily imply independence. Representation of non-Gaussian processes is thus a challenging issue and details on the ongoing research work in that field can be found in [208–210]. A more favorable scenario is when the random inputs are system parameters where the inputs are already parameterized.

Let ψ denote the finite dimensional stochastic space Ψ. Thus the original equation reduces to:

$$L(u; x, \psi) = f(x, \psi), \forall x \in D \text{ and } \forall \psi \in \Psi \tag{A.2}$$

A.1 Representation of stochastic processes

Let $\mu_Y(t)$ be the mean of the input process and let $C(x_1, x_2)$ be its covariance function. The Karhunen-Loeve expansion of the process denoted by $f(x, \omega)$ is given by:

$$f(x, \omega) = \mu_Y(t) + \sum_{i=1}^{\infty} \sqrt{\lambda_i}\phi_i(x)Y_i(\omega) \tag{A.3}$$

where ϕ_i are the orthogonal eigenvectors and λ_i are the corresponding eigenvalues of $C(x_1, x_2)$ which is by definition bounded, symmetric and positive definite. The deterministic eigenvectors $\phi_i(t)$ are obtained by solving the following homogeneous Fredholm integral equation of the second kind,

$$\int_D C(x_1, x_2)\phi_i(x_2)dx_2 = \lambda_i\phi_i(x_1), \quad x_1 \in D \tag{A.4}$$

It is to be noted that the eigenfunctions form a complete orthogonal set satisfying,

$$\int_D \phi_i(x)\phi_j(x)dx = \delta_{ij} \tag{A.5}$$

$Y_i(\omega)$ are mutually uncorrelated random variables satisfying

$$E[Y_i(\omega)] = 0, \quad E[Y_i(\omega)Y_j(\omega)] = \delta_{ij} \tag{A.6}$$

and defined by,

$$Y_i(\omega) = \frac{1}{\sqrt{\lambda_i}} \int_D (f(x,\omega) - \mu_Y(x))\phi(t)dx, \quad \forall i \tag{A.7}$$

For practical purposes, the K-L infinite series is truncated to a finite series expansion with N terms,

$$f(x,\omega) \approx \mu_Y(t) + \sum_{i=1}^{N} \sqrt{\lambda_i}\phi_i(x)Y_i(\omega). \tag{A.8}$$

The number of terms to be considered depends on the rate of decay of the eigenvalues which is related to the correlation length parameter of the covariance function.

Appendix B

Chapter 4 appendix: Surrogate modeling and model selection in irreducible high dimensions with small sample size

In this study, the performance of three hierarchical space filled designs, namely Refined Latinized Stratified Sampling (RLSS), Hierarchical Latin Hypercube Sampling (HLHS) and Sobol quasi-random sequence, are compared using the Rosenbrock function in different dimensions. An IGFEM based fiber-reinforced matrix composite model is also considered as an application problem. Ordinary kriging interpolation is chosen as the surrogate modeling method

with different choices of correlation functions. The AIC criterion is used for model selection and the accuracy of selection is cross-verified using the root mean squared (RMS) error values.

B.1 Kriging interpolation

Kriging [211, 212], also known as Gaussian process modeling, is an interpolation algorithm which tries to build a metamodel $\hat{y}(\boldsymbol{x})$ corresponding to an unknown true function $y(\boldsymbol{x})$ by assuming it to be a realization of a Gaussian process $Y(\boldsymbol{x})$. $Y(\boldsymbol{x})$ is given by:

$$Y(\boldsymbol{x}) = \boldsymbol{\beta}^T \boldsymbol{f}(\boldsymbol{x}) + \sigma^2 Z(\boldsymbol{x}, \omega) \tag{B.1}$$

where term $\boldsymbol{\beta}^T \boldsymbol{f}(\boldsymbol{x})$ represents the mean value of the Gaussian process and $\boldsymbol{\beta}$ is the regression vector and $\boldsymbol{f}(\boldsymbol{x})$ is the basis function vector. The term $\sigma^2 Z(\boldsymbol{x}, \omega)$ represent the local variations of the function about the mean $\boldsymbol{\beta}^T \boldsymbol{f}(\boldsymbol{x})$ where σ^2 is the process variance and $Z(\boldsymbol{x}, \omega)$ is a stationary Gaussian process with zero mean $[\mathbb{E}(Z(\boldsymbol{x})) = 0]$ and a correlation function given by:

$$Cov[Z(\boldsymbol{x}), Z(\boldsymbol{x}')] = K(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta}) \tag{B.2}$$

where $K = K(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta})$ is a measure of the similarity between two samples of the input space, e.g. $\boldsymbol{x}$ and $\boldsymbol{x}'$ and depends on the hyperparameters $\boldsymbol{\theta}$. We consider ordinary kriging in our study where $\boldsymbol{f}(\boldsymbol{x}) = 1$ and $\boldsymbol{\beta}^T \boldsymbol{f}(\boldsymbol{x}) = \beta_0$. Then the expression of the mean estimate of the kriging predictor is given by:

$$\mu_{\hat{Y}}(\boldsymbol{x}) = \hat{\beta}_0 + \boldsymbol{r}(\boldsymbol{x})^T \boldsymbol{R}^{-1}(\boldsymbol{y} - \mathbf{1}\hat{\beta}_0) \tag{B.3}$$

and the prediction variance estimate is given by:

$$s^2_{\hat{Y}}(\boldsymbol{x}) = \hat{\sigma}^2(1 - \boldsymbol{r}^T(\boldsymbol{x})\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x})) \tag{B.4}$$

where,

$$\hat{\beta}_0 = (\mathbf{1}^T \boldsymbol{R}^{-1} \mathbf{1})^{-1} \mathbf{1}^T \boldsymbol{R}^{-1} \boldsymbol{y}$$

$$\hat{\sigma}^2 = \frac{1}{n}(\boldsymbol{y} - \mathbf{1}\hat{\beta}_0)^T \boldsymbol{R}^{-1}(\boldsymbol{y} - \mathbf{1}\hat{\beta}_0) \tag{B.5}$$

In kriging prediction, the choice of covariance function K is important. In this study, four families of ellipsoidal type covariance functions are used: Gaussian, Exponential, Matérn 3/2 and Matérn 5/2, whose expressions are given in table B.1. The Kriging module in UQlab [213,214] was used to perform Kriging surrogate modeling.

Table B.1: Stationary covariance functions

Covariance functions	Expression
Gaussian	$exp\big(-\sum_{i=1}^{d}\theta_i p_i^2\big)$
Exponential	$exp\big(-\sum_{i=1}^{d}\theta_i p_i\big)$
Matérn 3/2	$(1+\sqrt{3}\sum_{i=1}^{d}\theta_i p_i)exp\big(-\sum_{i=1}^{d}\theta_i p_i\big)$
Matérn 5/2	$(1+\sqrt{5}\sum_{i=1}^{d}\theta_i p_i + ...$ $\frac{5}{3}\sum_{i=1}^{d}\theta_i^2 p_i^2)exp\big(-\sum_{i=1}^{d}\theta_i p_i\big)$

Note: $p_i = |x_i - x_i^{'}|$; for isotropic case, $\theta_i = \theta$

B.2 Numerical results

In this section, a commonly used benchmark function for high-dimensional applications, the Rosenbrock function, is used as an example problem given by:

$$f(x) = \sum_{i=1}^{d-1} 100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2 \tag{B.6}$$

where **x** is a d-dimensional vector. 4 Rosenbrock functions of dimensions 2, 5, 10 and 20 are considered. Sample points are generated according to three designs: Sobol, HLHS and RLSS. For a given dimension case and a given design, the function was evaluated at the generated sample points and kriging was applied to the data set to predict the function values at 10^6 Monte Carlo (MC) test samples in the problem domain. The performance of each case was measured

using Root Mean Squared Error (RMSE) given by:

$$RMSE = \sqrt{\frac{1}{N_t}\sum_{i=1}^{N_t}\left(y_{true}^{(i)} - y_{predicted}^{(i)}\right)^2} \tag{B.7}$$

where N_t is the total number of test samples, $\mathbf{y}_{true}$ is the vector of true values of the function at the N_t points and $\mathbf{y}_{predicted}$ is the vector of kriging-predicted values at the same N_t points. Here, $N_t = 10^6$. The results from the different designs are combined together to compare their performance with different choice of kriging covariance functions. The covariance functions are considered to be isotropic. Maximum Likelihood (ML) was chosen as the estimation method [135] and an interior point gradient-based optimization method with L-BFGS Hessian approximation [215, 216] was used to obtain the optimized parameter $\hat{\theta}$. The Akaike information criterion (AIC) [217, 218] was used for model selection from candidate covariance functions and the best models from each design were compared with each other. AIC for small datasets [219] is given by:

$$AIC_c = -2log(L(\hat{\theta}|D, M)) + 2n + \frac{2n(n+1)}{N-n-1} \tag{B.8}$$

where L is the likelihood function, $\hat{\theta}$ is the maximum likelihood estimate, D is the data, M is the model, n denotes the number of model parameters and N

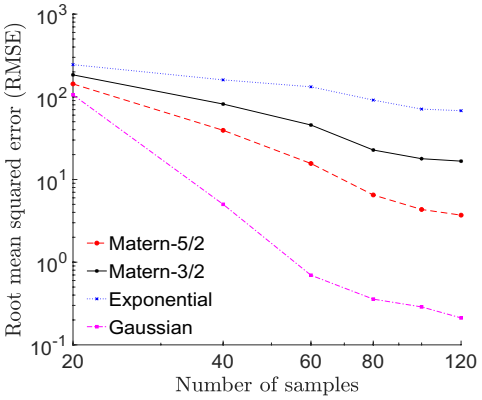
denotes the sample size. The model selection probability is then given by:

$$p_i = p(M_i|D) = \frac{exp(-\frac{\triangle_A^{(i)}}{2})}{\sum_{i=1}^{K} exp(-\frac{\triangle_A^{(i)}}{2})} \tag{B.9}$$

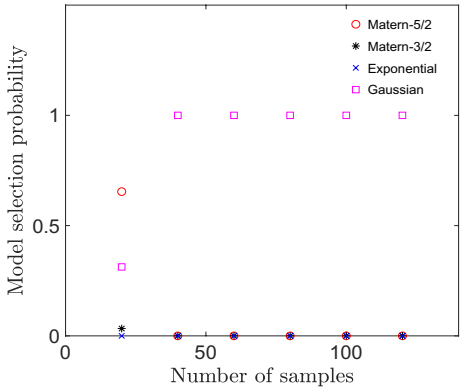
where, $\triangle_A^{(i)} = AIC_c^{(i)} - AIC_c{}^{min}$. In this study, $n = 1$ since isotropic covariance functions are considered as candidate models, and the model performance is compared at a fixed N. Thus, essentially, from the AIC criterion, the most suitable model is the one with the highest maximum likehood function value.

For HLHS and RLSS designs, 30 independent set of samples are generated. For each set, the metamodeling procedure is conducted and the RMSE value is calculated. From these values, an errorbar plot is generated using the minimum, maximum and median RMSE values. On the other hand, Sobol sequence is deterministic and generation of only a single set of samples is sufficient for performance comparisons.

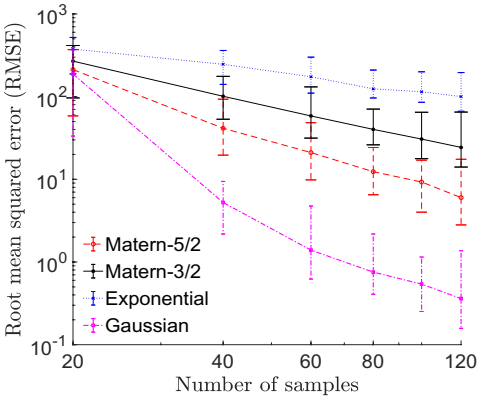
Figure B.1a shows the RMS error convergence plot comparison between different types of covariance function models with Sobol design points for 2-dimensional Rosenbrock function, and the surrogate model with the Gaussian covariance function is found to be most accurate for all the sample cases. Figure B.1b shows the probabilities of model selection with Sobol design. Except the 20-sample case, the AIC criterion chooses the Gaussian model as the most appropriate model which matches with the true RMSE error estimates in fig-



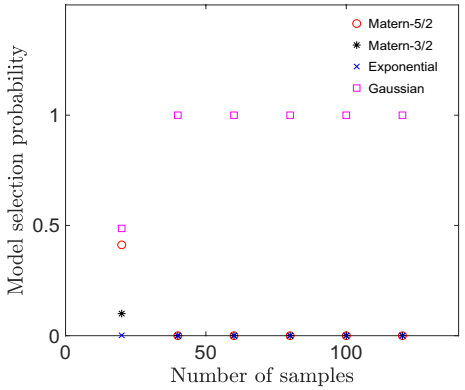
(a) Error convergence plot with Sobol design



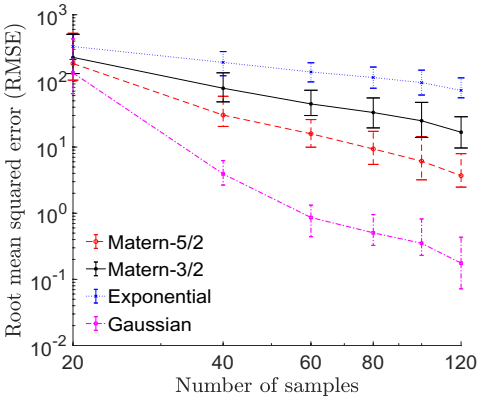
(b) Model selection probability plot with Sobol design



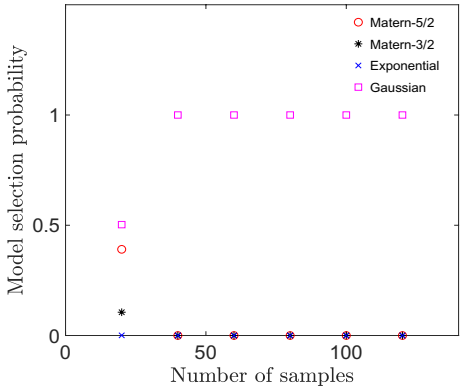
(c) Error convergence plot with HLHS design



(d) Model selection probability plot with HLHS design



(e) Error convergence plot with RLSS design



(f) Model selection probability plot with RLSS design

Figure B.1: 2-dimensional Rosenbrock function

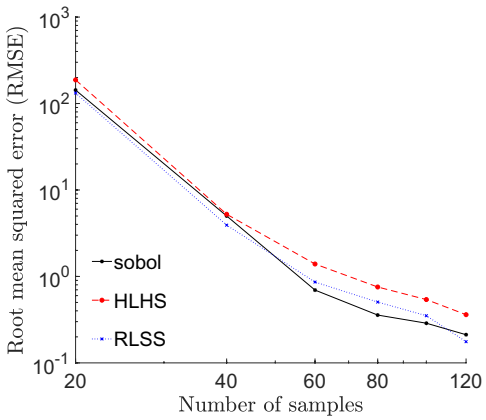
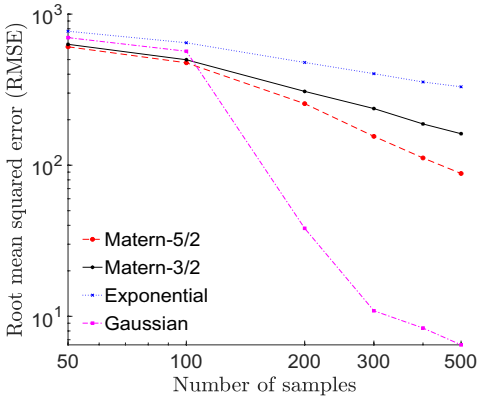


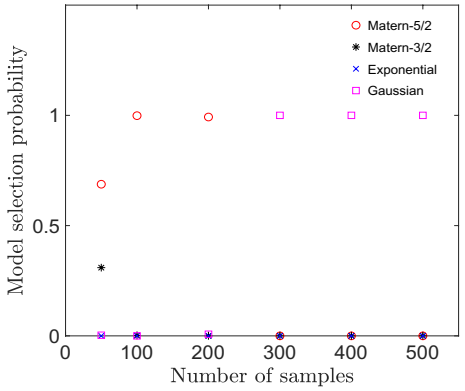
Figure B.2: 2-dimensional Rosenbrock function

ure B.1a. For the 20-sample case, the Matern-5/2 model is the most probable model but it has a higher RMS error than the Gaussian model, although the Gaussian model also has a finite probability of being selected. Figures B.1c and B.1e show the RMS error convergence plot comparisons between different types of covariance function models with HLHS design points and RLSS design points, respectively, where the Gaussian model again has the minimum error. As shown in figures B.1d and B.1f, the AIC criterion selects the Gaussian model as the most suitable model which is in agreement with the RMS error values. Figure B.2 shows the RMS error values of the most probable models selected from the AIC criterion using each of the 3 different designs for 2-dimensional Rosenbrock function.

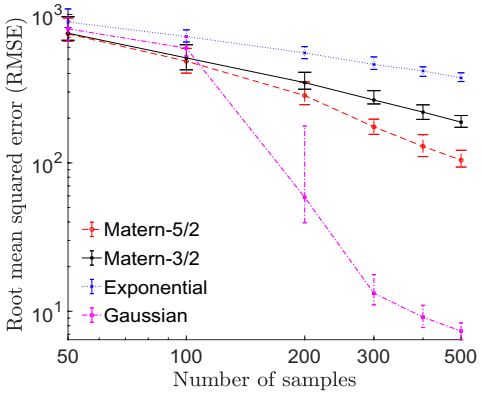
Figures B.3a, B.3c and B.3e shows the RMS error convergence plot comparison between different types of covariance function models with Sobol, HLHS and RLSS design points respectively for 5-dimensional Rosenbrock function.



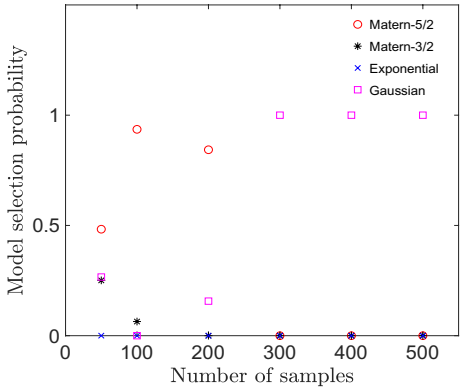
(a) Error convergence plot with Sobol design



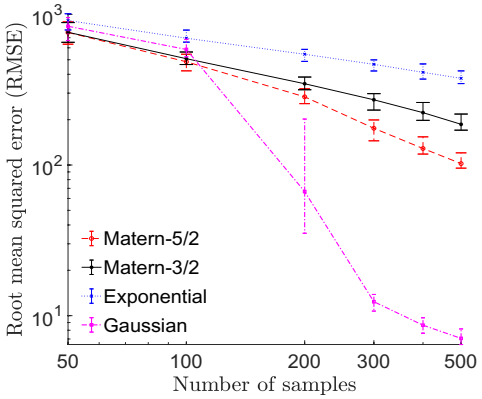
(b) Model selection probability plot with Sobol design



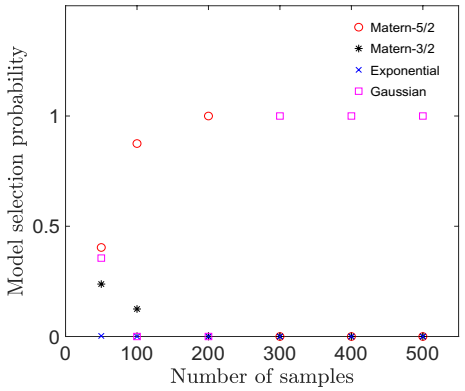
(c) Error convergence plot with HLHS design



(d) Model selection probability plot with HLHS design



(e) Error convergence plot with RLSS design



(f) Model selection probability plot with RLSS design

Figure B.3: 5-dimensional Rosenbrock function

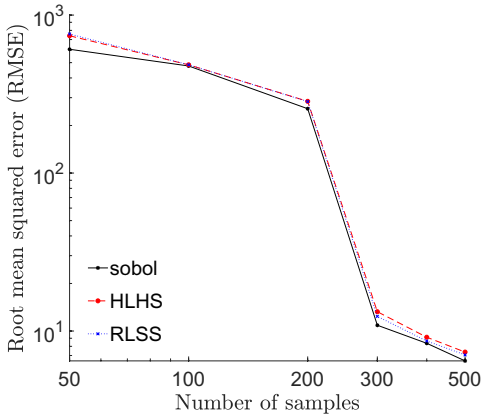
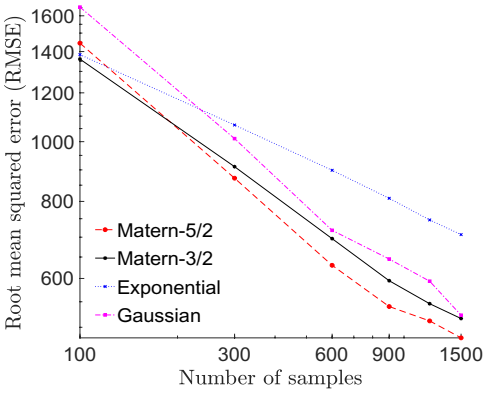


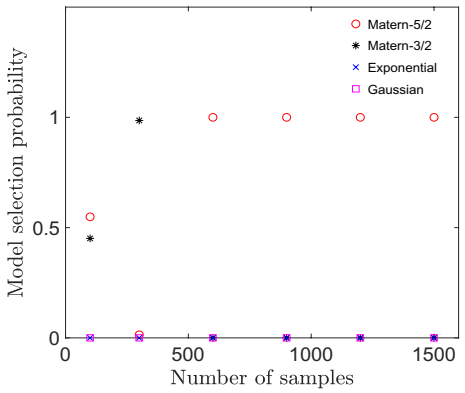
Figure B.4: 5-dimensional Rosenbrock function

The corresponding model selection probability plots in figures B.3b, B.3d and B.3f behave similarly in the sense that the selected (most probable) models are the ones with the minimum RMS error among the candidates. The only exception is the 200-sample case where the Matern-$5/2$ model is the selected model for all the 3 designs even though the RMS error is lowest for the Gaussian model for each design case. Figure B.4 shows the RMS error convergence plot comparison between the best models from the AIC criterion using the 3 different designs for 5-dimensional Rosenbrock function.

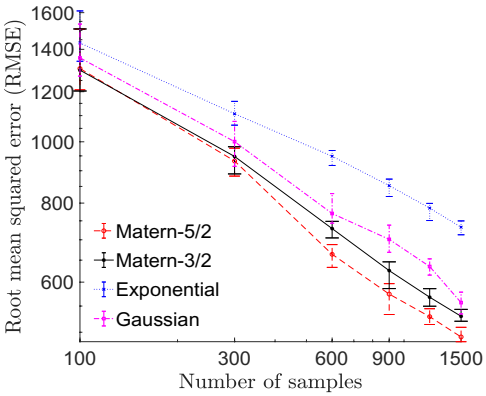
Figures B.5a, B.5c and B.5e shows the RMS error convergence plot comparison between different types of covariance function models with the 3 designs for 10-dimensional Rosenbrock function. In figures B.5b, B.5d and B.5f, it is seen that there is some discrepancy in the model selection and the corresponding RMS error for 100 and 300 sample cases, but the difference in error between the selected model and the most accurate (minimum RMS error) model



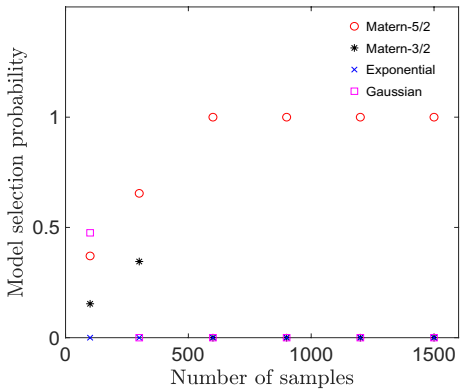
(a) Error convergence plot with Sobol design



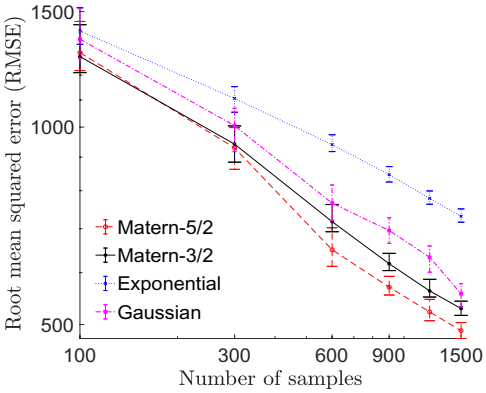
(b) Model selection probability plot with Sobol design



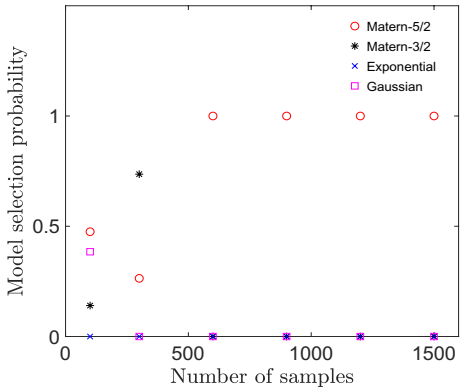
(c) Error convergence plot with HLHS design



(d) Model selection probability plot with HLHS design



(e) Error convergence plot with RLSS design



(f) Model selection probability plot with RLSS design

Figure B.5: 10-dimensional Rosenbrock function

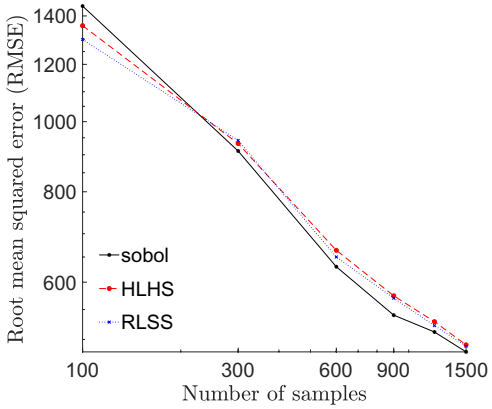
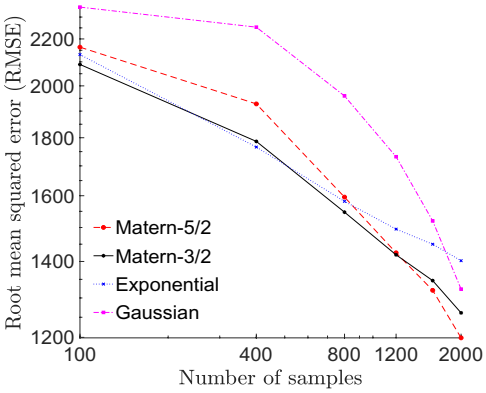


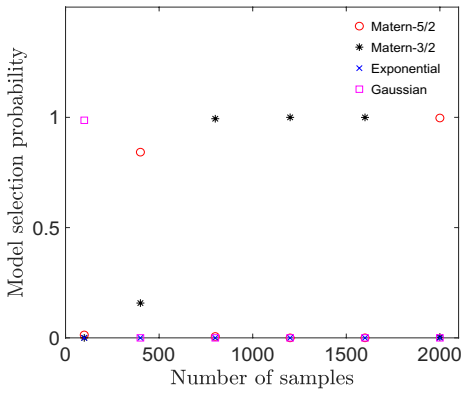
Figure B.6: 10-dimensional Rosenbrock function

is very small. However, there is good agreement between the maximum likelihood function value and the RMS error for the higher sample cases. Figure B.6 shows the RMS error convergence plot comparison between the best models from the AIC criterion using the 3 different designs for 10-dimensional Rosenbrock function.

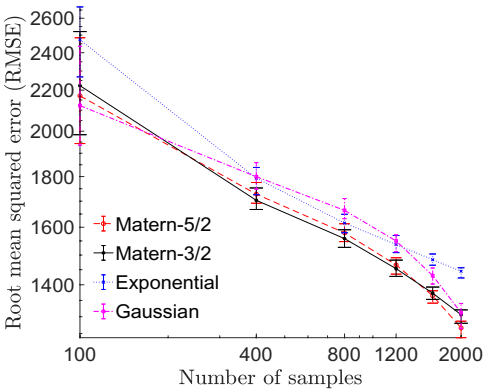
Figure B.7 shows the RMS error convergence plot comparison between different types of covariance function models with the 3 designs and the corresponding model selection probabilities for 20-dimensional Rosenbrock function. Apart from the 100-sample case with the Sobol design, there is a general agreement between the maximum likelihood function value and the RMS error and in cases of disagreement, the difference in the RMS error is fairly small. Figure B.8 shows the RMS error convergence plot comparison between the best models from the AIC criterion using the 3 different designs for 20-dimensional Rosenbrock function.



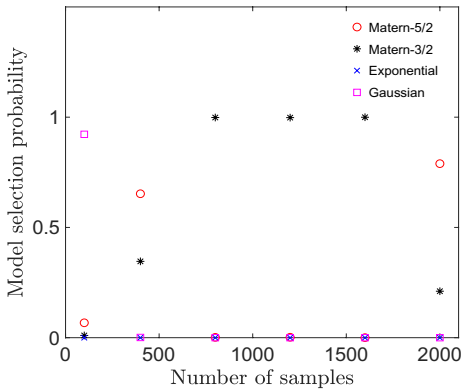
(a) Error convergence plot with Sobol design



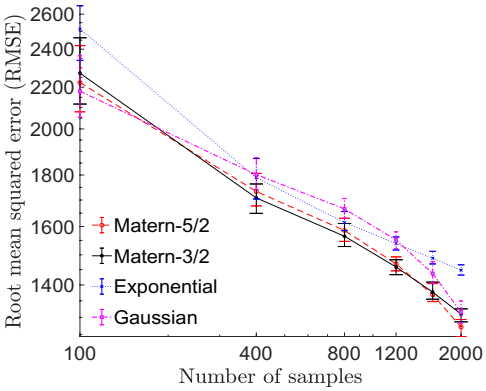
(b) Model selection probability plot with Sobol design



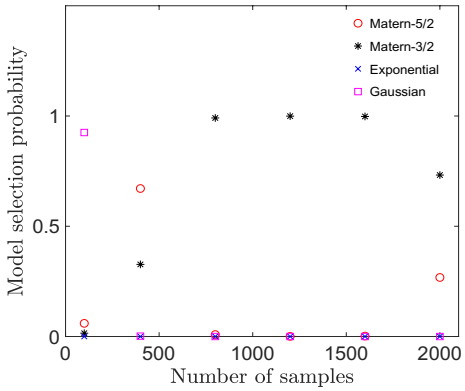
(c) Error convergence plot with HLHS design



(d) Model selection probability plot with HLHS design



(e) Error convergence plot with RLSS design



(f) Model selection probability plot with RLSS design

Figure B.7: 20-dimensional Rosenbrock function

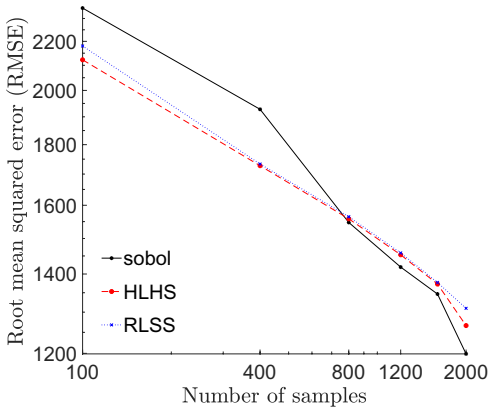
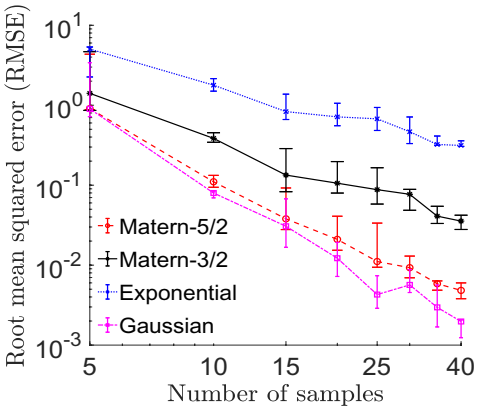


Figure B.8: 20-dimensional Rosenbrock function

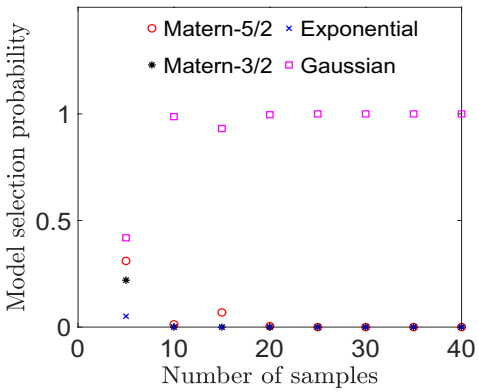
B.3 Application to composite models

In this section, an IGFEM-based RVE simulation model of a transverse section of a fiber-reinforced composite with varying numbers of fibers is considered similar to the one in section 4.5. The only difference in the models is that in this section, the IGFEM model only provides the homogenized stresses (outputs) with no sensitivity information (output gradients) with respect to the damage parameters, whereas in section 4.5, both the outputs and the output sensitivities are provided by the IGFEM model. The goal of this study is to assess the performance of the different combinations of sampling designs and kriging methods (different kernel) in the surrogate construction of this non-gradient IGFEM model. Models with 2, 5, 9, 20 and 50 fibers are considered for the study, the details of which can be found in section 4.5.

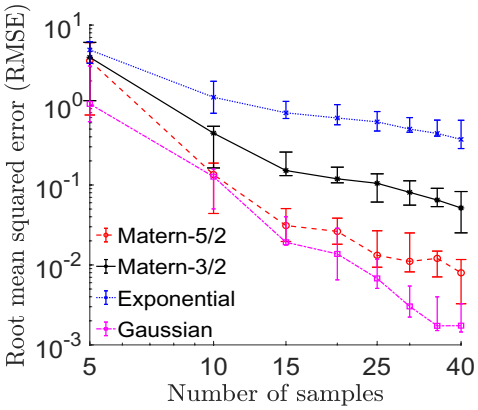
Figures B.9a and B.9c shows the RMS error convergence plot comparison



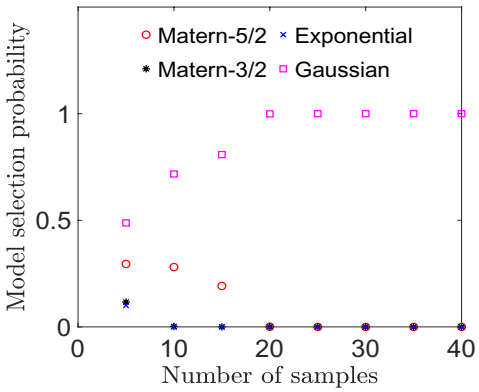
(a) Error convergence plot with scrambled sobol design



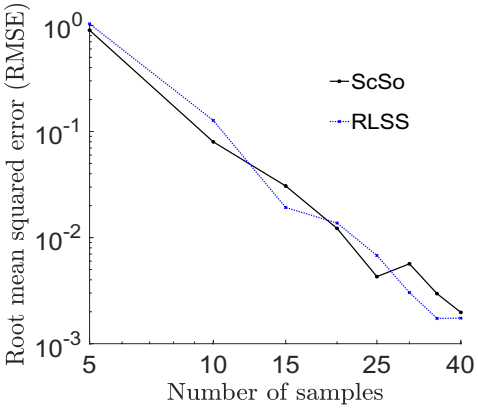
(b) Model selection probability plot with ScSo design



(c) Error convergence plot with RLSS design



(d) Model selection probability plot with RLSS design



(e) Best model

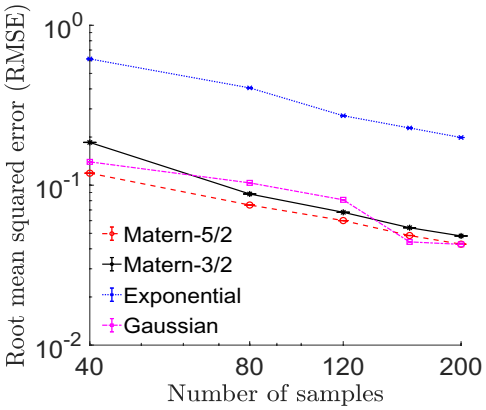
Figure B.9: 2-fiber model

between different types of covariance function models with Scrambled Sobol (ScSo) and RLSS design points respectively for the 2-fiber model. The corresponding model selection probability plots are shown in figures B.9b and B.9d. Figure B.9e shows the RMS error convergence plot comparison between the best models from the AIC criterion using the 2 different designs for 2-fiber model.

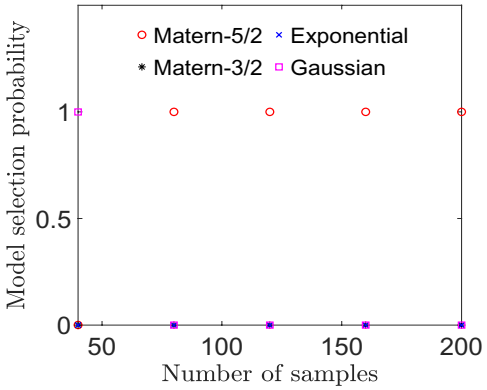
Figures B.10a and B.10c shows the RMS error convergence plot comparison between different types of covariance function models with ScSo and RLSS design points respectively for the clustered 5-fiber model. The corresponding model selection probability plots are shown in figures B.10b and B.10d. Figure B.10e shows the RMS error convergence plot comparison between the best models from the AIC criterion using the 2 different designs for the clustered 5-fiber model.

Figures B.11a and B.11c shows the RMS error convergence plot comparison between different types of covariance function models with ScSo and RLSS design points respectively for the structured 5-fiber model. The corresponding model selection probability plots are shown in figures B.11b and B.11d. Figure B.11e shows the RMS error convergence plot comparison between the best models from the AIC criterion using the 2 different designs for the structured 5-fiber model.

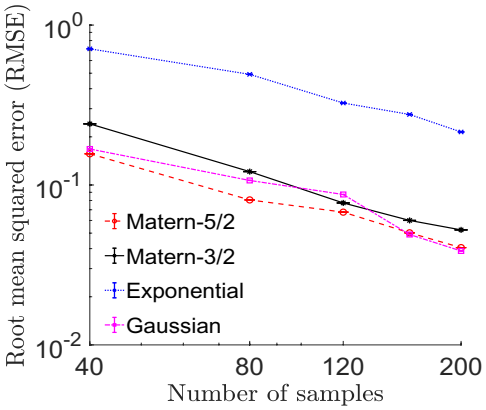
Figures B.12a and B.12c shows the RMS error convergence plot compari-



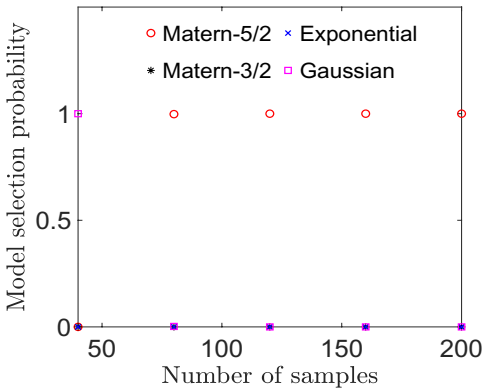
(a) Error convergence plot with scrambled sobol design



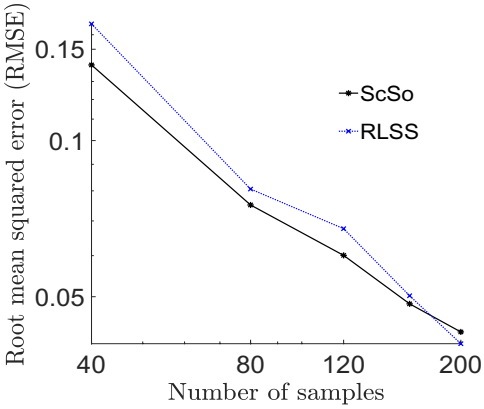
(b) Model selection probability plot with ScSo design



(c) Error convergence plot with RLSS design

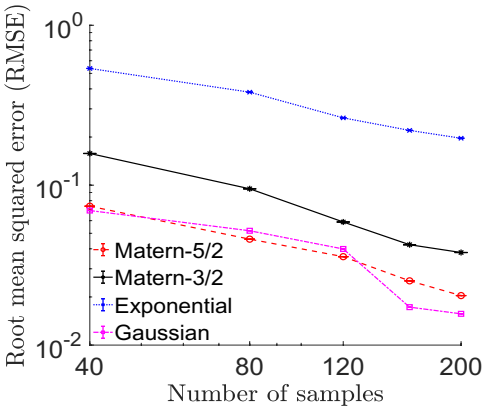


(d) Model selection probability plot with RLSS design

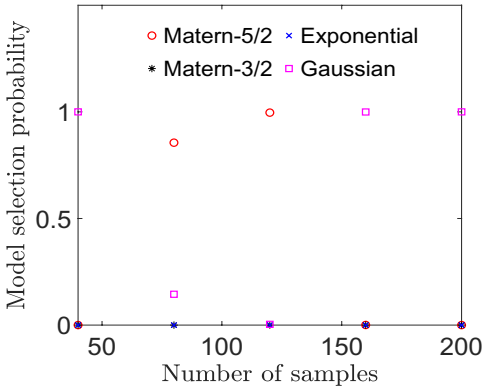


(e) best model

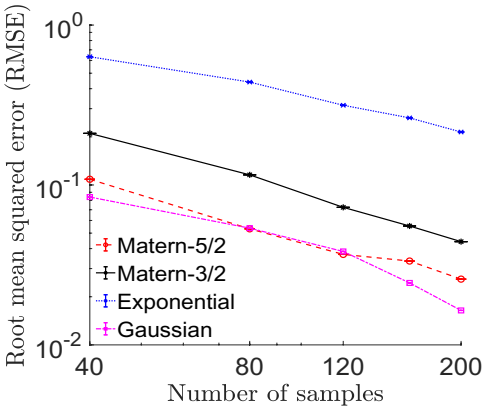
Figure B.10: Clustered 5-fiber model



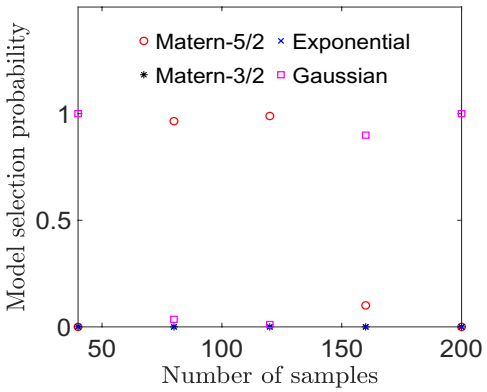
(a) Error convergence plot with scrambled sobol design



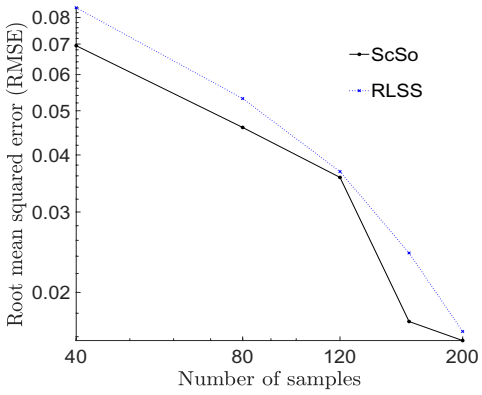
(b) Model selection probability plot with ScSo design



(c) Error convergence plot with RLSS design



(d) Model selection probability plot with RLSS design



(e) best model

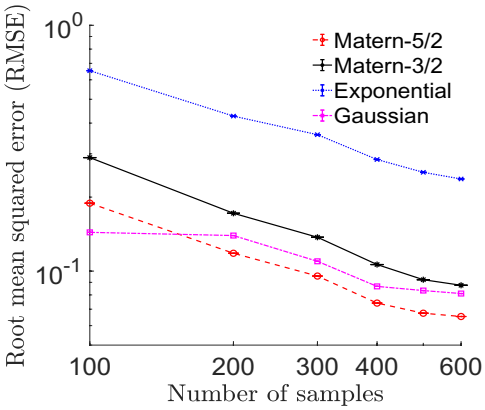
Figure B.11: Structured 5-fiber model

son between different types of covariance function models with ScSo and RLSS design points respectively for the clustered 9-fiber model. The corresponding model selection probability plots are shown in figures B.12b and B.12d. Figure B.12e shows the RMS error convergence plot comparison between the best models from the AIC criterion using the 2 different designs for the clustered 9-fiber model.

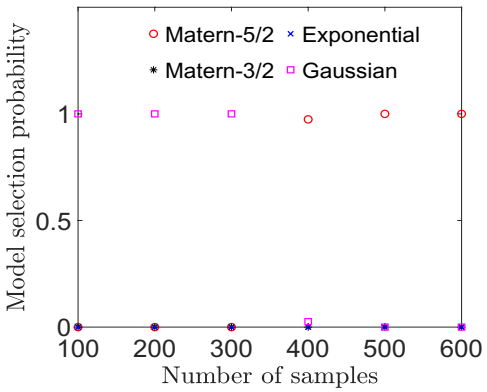
Figures B.13a and B.13c shows the RMS error convergence plot comparison between different types of covariance function models with ScSo and RLSS design points respectively for the structured 9-fiber model. The corresponding model selection probability plots are shown in figures B.13b and B.13d. Figure B.13e shows the RMS error convergence plot comparison between the best models from the AIC criterion using the 2 different designs for the structured 9-fiber model.

Figures B.14a and B.14c shows the RMS error convergence plot comparison between different types of covariance function models with ScSo and RLSS design points respectively for the randomly arranged 20-fiber model. The corresponding model selection probability plots are shown in figures B.14b and B.14d. Figure B.14e shows the RMS error convergence plot comparison between the best models from the AIC criterion using the 2 different designs for the 20-fiber model.

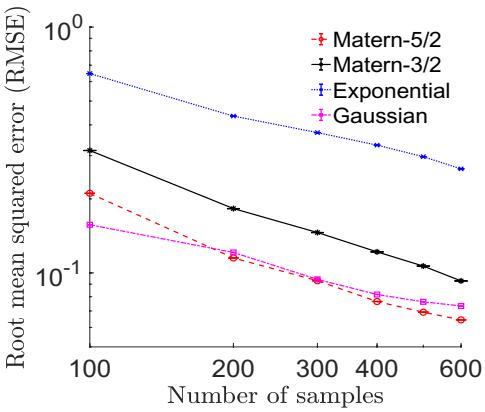
Figures B.15a and B.15c shows the RMS error convergence plot compari-



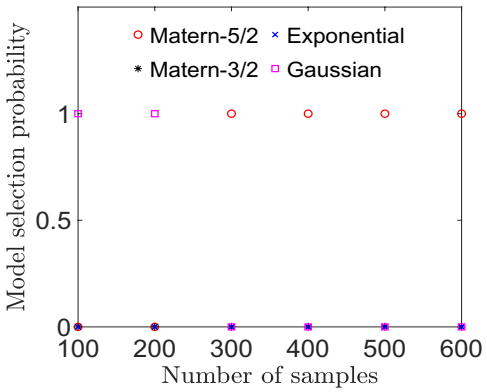
(a) Error convergence plot with scrambled sobol design



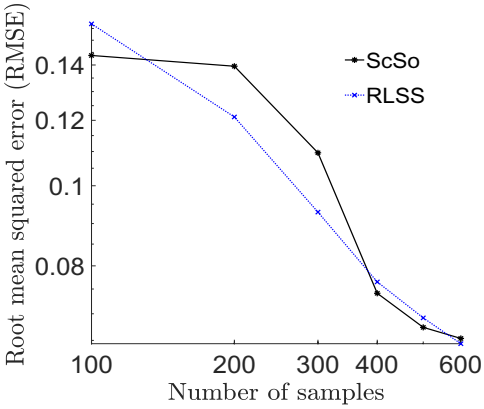
(b) Model selection probability plot with ScSo design



(c) Error convergence plot with RLSS design

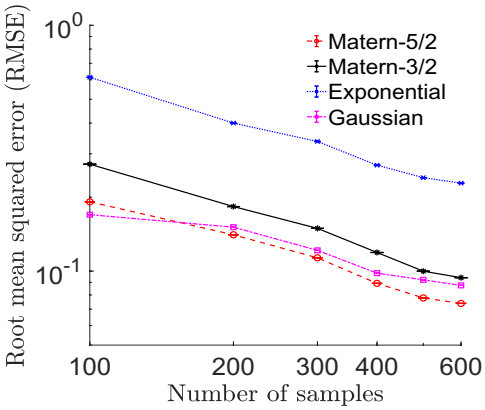


(d) Model selection probability plot with RLSS design

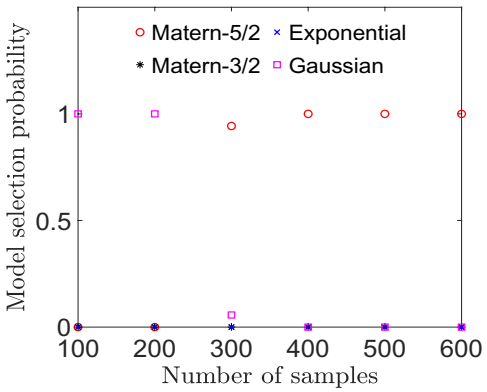


(e) best model

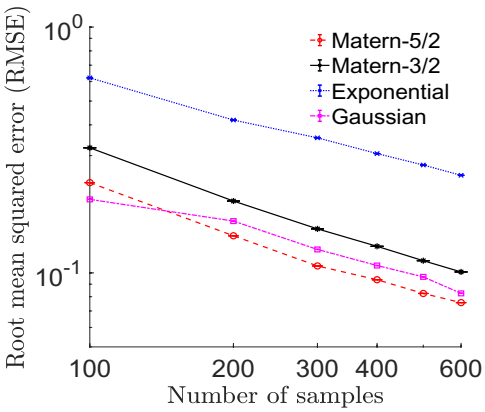
Figure B.12: Clustered 9-fiber model



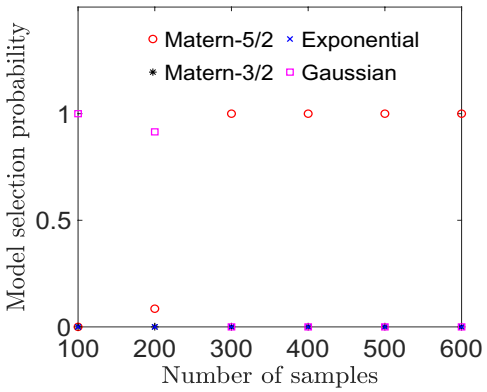
(a) Error convergence plot with scrambled sobol design



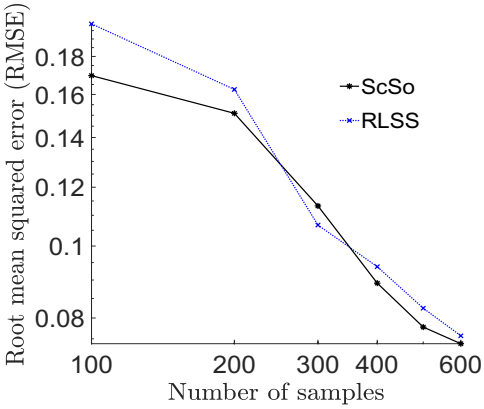
(b) Model selection probability plot with ScSo design



(c) Error convergence plot with RLSS design

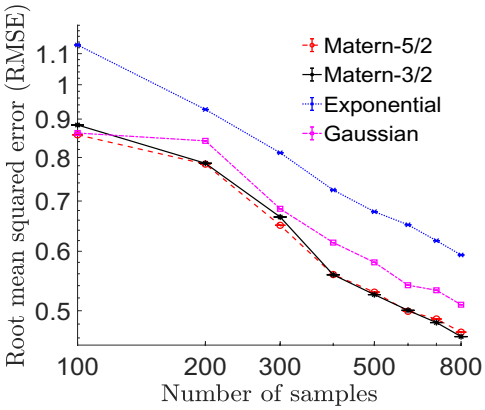


(d) Model selection probability plot with RLSS design

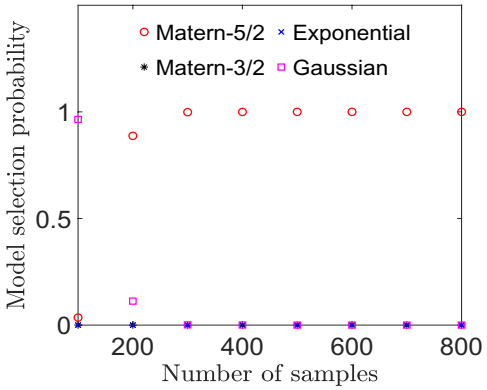


(e) best model

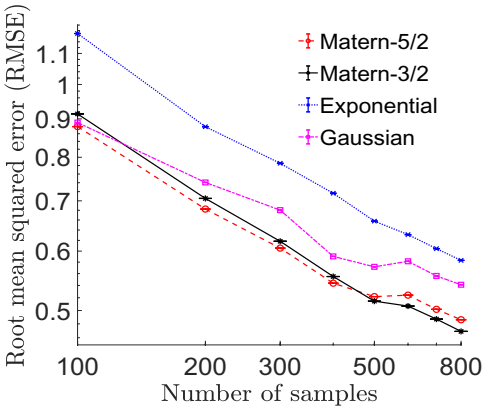
Figure B.13: Structured 9-fiber model



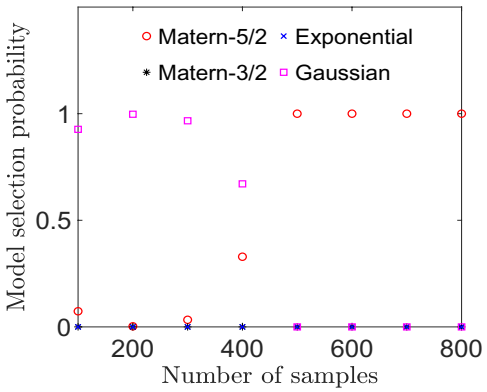
(a) Error convergence plot with scrambled sobol design



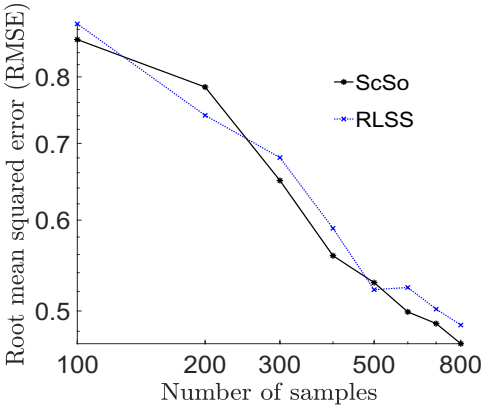
(b) Model selection probability plot with ScSo design



(c) Error convergence plot with RLSS design

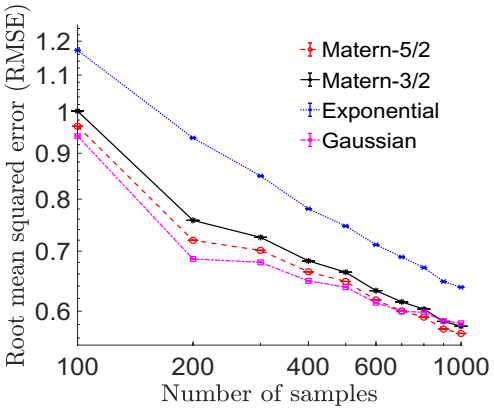


(d) Model selection probability plot with RLSS design

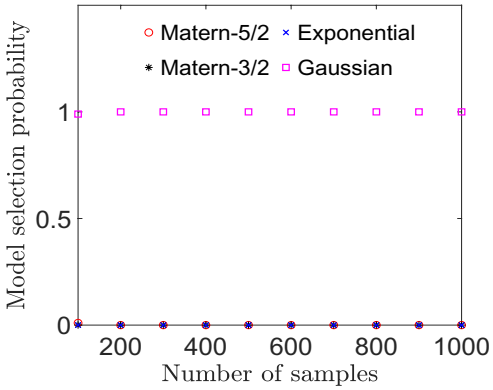


(e) best model

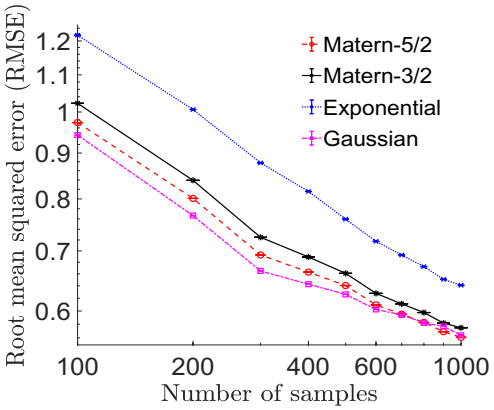
Figure B.14: 20-fiber model



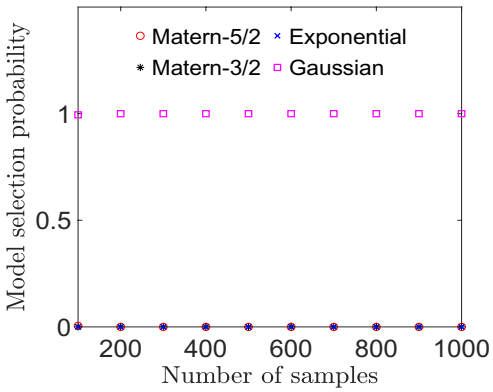
(a) Error convergence plot with scrambled sobol design



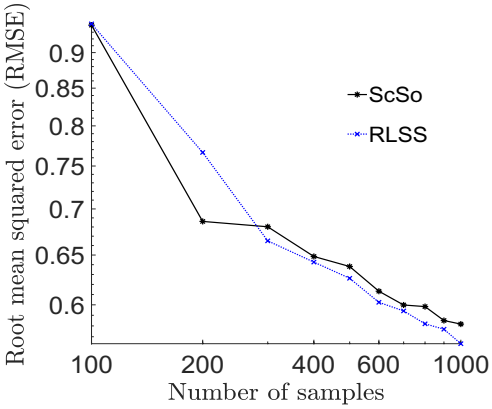
(b) Model selection probability plot with ScSo design



(c) Error convergence plot with RLSS design



(d) Model selection probability plot with RLSS design



(e) best model

Figure B.15: 50-fiber model

son between different types of covariance function models with ScSo and RLSS design points respectively for the randomly arranged 50-fiber model. The corresponding model selection probability plots are shown in figures B.15b and B.15d. Figure B.15e shows the RMS error convergence plot comparison between the best models from the AIC criterion using the 2 different designs for the 50-fiber model.

B.4 Conclusions

From the Rosenbrock function results, we can conclude that the performance of all three designs, Sobol, HLHS and RLSS, in most of the cases are very close to each other and none of them have a distinct advantage over the other. The only notable difference in performance is in the small sample 20-dimensional case where the selected model for the Sobol design has a significantly higher RMS error than the selected models with the other two designs. In general, for the 2 and 5-dimensional case, the Gaussian models seems to perform the best but its performance degrades with increase in dimensions while Matérn-$5/2$ and Matérn-$3/2$ models shows better performance in higher dimensions. These results motivate the use of isotropic Gaussian kernels for the RBF interpolation (gradient based and non-gradient) applied to the 2-dimensional

analytical functions in chapter 4. It is noted here that when isotropic kernels are used, kriging mean predictions and RBF predictions become identical.

For the non-gradient IGFEM model, the performance of the Scrambled Sobol and RLSS designs in most of the cases are again found to be similar to one other. As far as the choice of kernels in kriging interpolation is concerned, the Gaussian kernels seemed to perform the best for the 2-fiber model. For the 5-fiber, 9-fiber and 20-fiber cases, the Matérn kernels seem to slightly outperform the Gaussian kernels in performance in some situations. But for the 50-fiber case, the Gaussian kernel based kriging outperforms the rest. Thus, given the IGFEM model, the Gaussian kernel seems to perform satisfactorily across the range of different RVE models. This leads to the use of isotropic Gaussian kernels for the RBF interpolation (gradient based and non-gradient) applied to the IGFEM model in chapter 4.

It is to be noted that these observations are specific to the Rosenbrock function and the IGFEM composite model problem and might change for any other arbitrary function/problem.

Bibliography

[1] G. E. Box and N. R. Draper, *Empirical model-building and response surfaces*. John Wiley & Sons, 1987.

[2] R. H. Myers, D. C. Montgomery, and C. M. Anderson-Cook, *Response surface methodology: process and product optimization using designed experiments*. John Wiley & Sons, 2016.

[3] T. J. Santner, B. J. Williams, W. Notz, and B. J. Williams, *The design and analysis of computer experiments*. Springer, 2003, vol. 1.

[4] J. Sacks, S. B. Schiller, and W. J. Welch, "Designs for computer experiments," *Technometrics*, vol. 31, no. 1, pp. 41–47, 1989.

[5] K. Crombecq, "Surrogate modeling of computer experiments with sequential experimental design," Ph.D. dissertation, Ghent University, 2011.

[6] D. Xiu, *Numerical methods for stochastic computations: a spectral method approach*. Princeton university press, 2010.

[7] J. D. Jakeman and S. G. Roberts, "Stochastic galerkin and collocation methods for quantifying uncertainty in differential equations: a review," *ANZIAM Journal*, vol. 50, pp. 815–830, 2008.

[8] R. G. Ghanem and P. D. Spanos, *Stochastic finite elements: a spectral approach*. Courier Corporation, 2003.

[9] D. Xiu and G. E. Karniadakis, "The wiener–askey polynomial chaos for stochastic differential equations," *SIAM journal on scientific computing*, vol. 24, no. 2, pp. 619–644, 2002.

[10] ——, "Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos," *Computer methods in applied mechanics and engineering*, vol. 191, no. 43, pp. 4927–4948, 2002.

[11] ——, "Modeling uncertainty in flow simulations via generalized polynomial chaos," *Journal of computational physics*, vol. 187, no. 1, pp. 137–167, 2003.

[12] O. Le Maıtre, O. Knio, H. Najm, and R. Ghanem, "Uncertainty propagation using wiener–haar expansions," *Journal of computational Physics*, vol. 197, no. 1, pp. 28–57, 2004.

[13] I. Babuska, R. Tempone, and G. E. Zouraris, "Galerkin finite element approximations of stochastic elliptic partial differential equations," *SIAM Journal on Numerical Analysis*, vol. 42, no. 2, pp. 800–825, 2004.

[14] M. K. Deb, I. M. Babuška, and J. T. Oden, "Solution of stochastic partial differential equations using galerkin finite element techniques," *Computer Methods in Applied Mechanics and Engineering*, vol. 190, no. 48, pp. 6359–6372, 2001.

[15] X. Wan and G. E. Karniadakis, "An adaptive multi-element generalized polynomial chaos method for stochastic differential equations," *Journal of Computational Physics*, vol. 209, no. 2, pp. 617–642, 2005.

[16] A. Nouy, "A generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 45-48, pp. 4521–4537, 2007.

[17] G. Venter, R. T. Haftka, and J. H. Starnes, "Construction of response surface approximations for design optimization," *AIAA journal*, vol. 36, no. 12, pp. 2242–2249, 1998.

[18] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker, "Surrogate-based analysis and optimization," *Progress in aerospace sciences*, vol. 41, no. 1, pp. 1–28, 2005.

[19] R. P. Liem, *Multimission Fuel-Burn Minimization in Aircraft Design: A Surrogate-Modeling Approach*. University of Toronto (Canada), 2015.

[20] T. W. Simpson, D. K. Lin, and W. Chen, "Sampling strategies for computer experiments: design and analysis," *International Journal of Reliability and Applications*, vol. 2, no. 3, pp. 209–240, 2001.

[21] S. M. Stigler, "Gergonne's 1815 paper on the design and analysis of polynomial regression experiments," *Historia Mathematica*, vol. 1, no. 4, pp. 431–439, 1974.

[22] M. Eldred, "Recent advances in non-intrusive polynomial chaos and stochastic collocation methods for uncertainty analysis and design," in *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 17th AIAA/ASME/AHS Adaptive Structures Conference 11th AIAA No*, 2009, p. 2274.

[23] N. Cressie, "Statistics for spatial data," *Terra Nova*, vol. 4, no. 5, pp. 613–617, 1992.

[24] J. Koehler and A. Owen, "Computer experiments. handbook of statistics," *Elsevier Science*, pp. 261–308, 1996.

[25] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and analysis of computer experiments," *Statistical science*, pp. 409–423, 1989.

[26] J. H. Friedman and W. Stuetzle, "Projection pursuit regression," *Journal of the American statistical Association*, vol. 76, no. 376, pp. 817–823, 1981.

[27] A. I. Forrester, N. W. Bressloff, and A. J. Keane, "Optimization using surrogate models and partially converged computational fluid dynamics simulations," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 462, no. 2071, pp. 2177–2204, 2006.

[28] D. McDonald, W. Grantham, W. Tabor, and M. Murphy, "Response surface model development for global/local optimization using radial basis functions," in *8th Symposium on Multidisciplinary Analysis and Optimization*, 2000, p. 4776.

[29] Y.-S. Ong, K. Y. Lum, and P. B. Nair, "Hybrid evolutionary algorithm with hermite radial basis function interpolants for computationally expensive adjoint solvers," *Computational Optimization and Applications*, vol. 39, no. 1, pp. 97–119, 2008.

[30] L. Maragliano and E. Vanden-Eijnden, "Single-sweep methods for free energy calculations," *The Journal of chemical physics*, vol. 128, no. 18, p. 184110, 2008.

[31] A. J. Keane and P. B. Nair, "Computational approaches for aerospace design," *John Wiley&Sons, Ltd, West Sussex*, vol. 582, 2005.

[32] J. H. Friedman *et al.*, "Multivariate adaptive regression splines," *The annals of statistics*, vol. 19, no. 1, pp. 1–67, 1991.

[33] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[34] J. Hoskins and D. Himmelblau, "Process control via artificial neural networks and reinforcement learning," *Computers & chemical engineering*, vol. 16, no. 4, pp. 241–251, 1992.

[35] G. Fadel, M. Riley, and J. Barthelemy, "Two point exponential approximation method for structural optimization," *Structural optimization*, vol. 2, no. 2, pp. 117–124, 1990.

[36] L. Wang and R. V. Grandhi, "Improved two-point function approximations for design optimization," *AIAA journal*, vol. 33, no. 9, pp. 1720–1727, 1995.

[37] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *Proceedings of the 1968 23rd ACM national conference*. ACM, 1968, pp. 517–524.

[38] P. Chen, A. Quarteroni, and G. Rozza, "Reduced order methods for uncertainty quantification problems," *ETH Zurich, SAM Report*, vol. 3, 2015.

BIBLIOGRAPHY

[39] A. C. Antoulas, *Approximation of large-scale dynamical systems*. Siam, 2005, vol. 6.

[40] K. Gallivan, E. Grimme, and P. Van Dooren, "Padé approximation of large-scale dynamic systems with lanczos methods," in *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, vol. 1. IEEE, 1994, pp. 443–448.

[41] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by padé approximation via the lanczos process," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 5, pp. 639–649, 1995.

[42] E. Grimme, "Krylov projection methods for model reduction," Ph.D. dissertation, 1997.

[43] M. S. Tombs and I. Postlethwaite, "Truncated balanced realization of a stable non-minimal state-space system," *International Journal of Control*, vol. 46, no. 4, pp. 1319–1330, 1987.

[44] M. G. Safonov and R. Chiang, "A schur method for balanced-truncation model reduction," *IEEE Transactions on Automatic Control*, vol. 34, no. 7, pp. 729–733, 1989.

[45] T. Penzl, "Algorithms for model reduction of large dynamical systems," *Linear algebra and its applications*, vol. 415, no. 2-3, pp. 322–343, 2006.

[46] D. C. Sorensen and A. Antoulas, "The sylvester equation and approximate balanced reduction," *Linear algebra and its applications*, vol. 351, pp. 671–700, 2002.

[47] S. Gugercin and A. C. Antoulas, "A survey of model reduction by balanced truncation and some new results," *International Journal of Control*, vol. 77, no. 8, pp. 748–766, 2004.

[48] T. Bui-Thanh, M. Damodaran, and K. E. Willcox, "Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition," *AIAA journal*, vol. 42, no. 8, pp. 1505–1516, 2004.

[49] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge university press, 2012.

[50] L. Sirovich, "Turbulence and the dynamics of coherent structures. part 1: Coherent structures," *Quart Appl Math*, vol. 45, pp. 561–571, 1987.

[51] K. Willcox and J. Peraire, "Balanced model reduction via the proper orthogonal decomposition," *AIAA journal*, vol. 40, no. 11, pp. 2323–2330, 2002.

[52] A. Deane, I. Kevrekidis, G. E. Karniadakis, and S. Orszag, "Low-dimensional models for complex geometry flows: application to grooved channels and circular cylinders," *Physics of Fluids A: Fluid Dynamics*, vol. 3, no. 10, pp. 2337–2354, 1991.

[53] T. Robinson, M. Eldred, K. Willcox, and R. Haimes, "Strategies for multifidelity optimization with variable dimensional hierarchical models," in *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 14th AIAA/ASME/AHS Adaptive Structures Conference 7th*, 2006, p. 1819.

[54] T. D. Robinson, "Surrogate-based optimization using multifidelity models with variable parameterization," Ph.D. dissertation, Massachusetts Institute of Technology, 2007.

[55] N. Alexandrov, E. Nielsen, R. Lewis, and W. Anderson, "First-order model management with variable-fidelity physics applied to multi-element airfoil optimization," in *8th Symposium on Multidisciplinary Analysis and Optimization*, 2000, p. 4886.

[56] P. Thokala and J. R. Martins, "Variable-complexity optimization applied to airfoil design," *Engineering optimization*, vol. 39, no. 3, pp. 271–286, 2007.

[57] W. L. Briggs, S. F. McCormick *et al.*, *A multigrid tutorial*. Siam, 2000, vol. 72.

[58] R. Lewis and S. Nash, "A multigrid approach to the optimization of systems governed by differential equations," in *8th Symposium on Multidisciplinary Analysis and Optimization*, 2000, p. 4890.

[59] V. Picheny, "Improving accuracy and compensating for uncertainty in surrogate modeling," Ph.D. dissertation, University of Florida Gainesville, 2009.

[60] S. Bilicz, "Application of design-of-experiment methods and surrogate models in electromagnetic nondestructive evaluation," Ph.D. dissertation, Paris 11, 2011.

[61] K.-T. Fang, R. Li, and A. Sudjianto, *Design and modeling for computer experiments*. Chapman and Hall/CRC, 2005.

[62] T. P. Ryan and J. Morgan, "Modern experimental design," *Journal of Statistical Theory and Practice*, vol. 1, no. 3-4, pp. 501–506, 2007.

[63] S. A. Smolyak, "Quadrature and interpolation formulas for tensor products of certain classes of functions," in *Doklady Akademii Nauk*, vol. 148, no. 5. Russian Academy of Sciences, 1963, pp. 1042–1045.

[64] M. D. McKay, R. J. Beckman, and W. J. Conover, "Comparison of three

methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.

[65] M. C. Shewry and H. P. Wynn, "Maximum entropy sampling," *Journal of applied statistics*, vol. 14, no. 2, pp. 165–170, 1987.

[66] H. Faure, "Discrépance de suites associées à un système de numération (en dimension s)," *Acta Arith.*, vol. 41, no. 4, pp. 337–351, 1982.

[67] H. Niederreiter, "Point sets and sequences with small discrepancy," *Monatsh. Math.*, vol. 104, no. 4, pp. 273–337, 1987.

[68] J. H. Halton, "On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals," *Numer. Math.*, vol. 2, no. 1, pp. 84–90, 1960.

[69] I. M. Sobol, "Uniformly distributed sequences with an additional uniform property," *USSR Comput. Math. Math. Phys.*, vol. 16, no. 5, pp. 236–242, 1976.

[70] W. J. Morokoff and R. E. Caflisch, "Quasi-random sequences and their discrepancies," *SIAM J. Sci. Comput.*, vol. 15, no. 6, pp. 1251–1279, 1994.

[71] J. Matoušek, "On the L_2-discrepancy for anchored boxes," *J. Complex.*, vol. 14, no. 4, pp. 527–556, 1998.

[72] L. Kocis and W. J. Whiten, "Computational investigations of low-discrepancy sequences," *ACM Trans. Math. Softw.*, vol. 23, no. 2, pp. 266–294, 1997.

[73] S. Tezuka and H. Faure, "I-binomial scrambling of digital nets and sequences," *J. Complex.*, vol. 19, no. 6, pp. 744–757, 2003.

[74] A. I. Khuri and J. A. Cornell, *Response surfaces: designs and analyses*. Routledge, 2018.

[75] B. J. Williams, T. J. Santner, and W. I. Notz, "Sequential design of computer experiments to minimize integrated response functions," *Statistica Sinica*, pp. 1133–1152, 2000.

[76] C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker, "Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments," *Journal of the American Statistical Association*, vol. 86, no. 416, pp. 953–963, 1991.

[77] I. G. Osio and C. H. Amon, "An engineering design methodology with multistage bayesian surrogates and optimal sampling," *Research in Engineering Design*, vol. 8, no. 4, pp. 189–206, 1996.

[78] J. Pavo and S. Gyimothy, "Adaptive inversion database for electromag-

netic nondestructive evaluation," *NDT & e International*, vol. 40, no. 3, pp. 192–202, 2007.

[79] S. Gyimóthy, I. Kiss, and J. Pávó, "Adaptive sampling technique based on moving meshes for building data-equidistant inversion databases for ndt," *International Journal of Applied Electromagnetics and Mechanics*, vol. 30, no. 3-4, pp. 309–319, 2009.

[80] J. P. Kleijnen and W. v. Beers, "Application-driven sequential designs for simulation experiments: Kriging metamodelling," *Journal of the operational research society*, vol. 55, no. 8, pp. 876–883, 2004.

[81] C. J. Turner, M. I. Campbell, and R. H. Crawford, "Generic sequential sampling for metamodel approximations," in *ASME 2003 international design engineering technical conferences and computers and information in engineering conference*. American Society of Mechanical Engineers, 2003, pp. 555–564.

[82] Y. Lin, D. Luo, T. Bailey, R. Khire, J. Wang, and T. W. Simpson, "Model validation and error modeling to support sequential sampling," in *ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2008, pp. 1255–1264.

[83] O. Le Maître and O. M. Knio, *Spectral methods for uncertainty quantifi-*

cation: with applications to computational fluid dynamics. Springer Science & Business Media, 2010.

[84] X. Ma and N. Zabaras, "An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations," *Journal of Computational Physics*, vol. 228, no. 8, pp. 3084–3113, 2009.

[85] A. Bhaduri and L. Graham-Brady, "An efficient adaptive sparse grid collocation method through derivative estimation," *Probabilistic Engineering Mechanics*, vol. 51, pp. 11–22, 2018.

[86] N. Wiener, "The homogeneous chaos," *American Journal of Mathematics*, vol. 60, no. 4, pp. 897–936, 1938.

[87] R. Ghanem, "Scales of fluctuation and the propagation of uncertainty in random porous media," *Water Resources Research*, vol. 34, no. 9, pp. 2123–2136, 1998.

[88] ——, "Stochastic finite elements with multiple random non-gaussian properties," *Journal of Engineering Mechanics*, vol. 125, no. 1, pp. 26–40, 1999.

[89] D. Ghosh and C. Farhat, "Strain and stress computations in stochastic finite element methods," *International Journal for Numerical Methods in Engineering*, vol. 74, no. 8, pp. 1219–1239, 2008.

[90] A. Doostan and G. Iaccarino, "A least-squares approximation of partial differential equations with high-dimensional random inputs," *Journal of Computational Physics*, vol. 228, no. 12, pp. 4332–4345, 2009.

[91] R. Ghanem and J. Red-Horse, "Propagation of probabilistic uncertainty in complex physical systems using a stochastic finite element approach," *Physica D: Nonlinear Phenomena*, vol. 133, no. 1, pp. 137–144, 1999.

[92] R. Askey and J. Wilson, "Some basic hypergeometric polynomials that generalize jacobi polynomials memoirs amer," *Math. Soc. AMS Providence RI*, vol. 319, 1985.

[93] D. Xiu and G. E. Karniadakis, "A new stochastic approach to transient heat conduction modeling with uncertainty," *International Journal of Heat and Mass Transfer*, vol. 46, no. 24, pp. 4681–4693, 2003.

[94] J. A. Witteveen and H. Bijl, "Modeling arbitrary uncertainties using gram-schmidt polynomial chaos," in *44th AIAA aerospace sciences meeting and exhibit*, 2006, p. 896.

[95] X. Wan and G. E. Karniadakis, "Beyond wiener–askey expansions: handling arbitrary pdfs," *Journal of Scientific Computing*, vol. 27, no. 1, pp. 455–464, 2006.

[96] G. Kewlani and K. Iagnemma, "A multi-element generalized polynomial

chaos approach to analysis of mobile robot dynamics under uncertainty," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 1177–1182.

[97] X. Wan and G. E. Karniadakis, "Multi-element generalized polynomial chaos for arbitrary probability measures," *SIAM Journal on Scientific Computing*, vol. 28, no. 3, pp. 901–928, 2006.

[98] D. Ghosh and R. Ghanem, "Stochastic convergence acceleration through basis enrichment of polynomial chaos expansions," *International journal for numerical methods in engineering*, vol. 73, no. 2, pp. 162–184, 2008.

[99] D. Xiu, "Efficient collocational approach for parametric uncertainty analysis," *Commun. Comput. Phys*, vol. 2, no. 2, pp. 293–309, 2007.

[100] I. Babuška, F. Nobile, and R. Tempone, "A stochastic collocation method for elliptic partial differential equations with random input data," *SIAM Journal on Numerical Analysis*, vol. 45, no. 3, pp. 1005–1034, 2007.

[101] R. Cools, "Advances in multidimensional integration," *Journal of Computational and Applied Mathematics*, vol. 149, no. 1, pp. 1–12, 2002.

[102] H.-J. Bungartz and M. Griebel, "Sparse grids," *Acta numerica*, vol. 13, pp. 147–269, 2004.

[103] D. Xiu and J. S. Hesthaven, "High-order collocation methods for differential equations with random inputs," *SIAM Journal on Scientific Computing*, vol. 27, no. 3, pp. 1118–1139, 2005.

[104] F. Nobile, R. Tempone, and C. G. Webster, "A sparse grid stochastic collocation method for partial differential equations with random input data," *SIAM Journal on Numerical Analysis*, vol. 46, no. 5, pp. 2309–2345, 2008.

[105] L. Mathelin, M. Y. Hussaini, and T. A. Zang, "A stochastic collocation algorithm for uncertainty analysis," 2003.

[106] A. Klimke and B. Wohlmuth, "Algorithm 847: spinterp: Piecewise multilinear hierarchical sparse grid interpolation in matlab," *ACM Transactions on Mathematical Software (TOMS)*, vol. 31, no. 4, pp. 561–579, 2005.

[107] W. A. Klimke, *Uncertainty modeling using fuzzy arithmetic and sparse grids*. Citeseer, 2006.

[108] A. Klimke, "Sparse grid interpolation toolbox–users guide," *IANS report*, vol. 17, 2007.

[109] B. Ganapathysubramanian and N. Zabaras, "Sparse grid collocation

schemes for stochastic natural convection problems," *Journal of Computational Physics*, vol. 225, no. 1, pp. 652–685, 2007.

[110] F. Nobile, R. Tempone, and C. G. Webster, "An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data," *SIAM Journal on Numerical Analysis*, vol. 46, no. 5, pp. 2411–2442, 2008.

[111] J. Foo, X. Wan, and G. E. Karniadakis, "The multi-element probabilistic collocation method (me-pcm): Error analysis and applications," *Journal of Computational Physics*, vol. 227, no. 22, pp. 9572–9595, 2008.

[112] N. Agarwal and N. R. Aluru, "A domain adaptive stochastic collocation approach for analysis of mems under uncertainties," *Journal of Computational Physics*, vol. 228, no. 20, pp. 7662–7688, 2009.

[113] T. Chantrasmi, A. Doostan, and G. Iaccarino, "Padé–legendre approximants for uncertainty analysis with discontinuous response surfaces," *Journal of Computational Physics*, vol. 228, no. 19, pp. 7159–7180, 2009.

[114] D. R. Kincaid and E. W. Cheney, *Numerical analysis: mathematics of scientific computing*. American Mathematical Soc., 2002, vol. 2.

[115] C. A. Hall and W. W. Meyer, "Optimal error bounds for cubic spline inter-

polation," *Journal of Approximation Theory*, vol. 16, no. 2, pp. 105–122, 1976.

[116] A. Genz, "A package for testing multiple integration subroutines," in *Numerical Integration*. Springer, 1987, pp. 337–340.

[117] R. H. Kraichnan, "Direct-interaction approximation for a system of several interacting simple shear waves," *The Physics of Fluids*, vol. 6, no. 11, pp. 1603–1609, 1963.

[118] S. A. Orszag and L. Bissonnette, "Dynamical properties of truncated wiener-hermite expansions," *The Physics of Fluids*, vol. 10, no. 12, pp. 2603–2613, 1967.

[119] Hibbett, Karlsson, and Sorensen, *ABAQUS/standard: User's Manual*. Hibbitt, Karlsson & Sorensen, 1998, vol. 1.

[120] M. Eldred and J. Burkardt, "Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification," in *47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition*, 2009, p. 976.

[121] A. Bhaduri, Y. He, M. D. Shields, L. Graham-Brady, and R. M. Kirby, "Stochastic collocation approach with adaptive mesh refinement for

parametric uncertainty analysis," *Journal of Computational Physics*, vol. 371, pp. 732–750, 2018.

[122] J. Foo and G. E. Karniadakis, "Multi-element probabilistic collocation method in high dimensions," *Journal of Computational Physics*, vol. 229, no. 5, pp. 1536–1557, 2010.

[123] I. M. Sobol, "Theorems and examples on high dimensional model representation," *Reliability Engineering & System Safety*, vol. 79, no. 2, pp. 187–193, 2003.

[124] H. Rabitz, Ö. F. Aliş, J. Shorter, and K. Shim, "Efficient input-output model representations," *Computer Physics Communications*, vol. 117, no. 1-2, pp. 11–20, 1999.

[125] R. Archibald, A. Gelb, R. Saxena, and D. Xiu, "Discontinuity detection in multivariate space for stochastic simulations," *Journal of Computational Physics*, vol. 228, no. 7, pp. 2676–2689, 2009.

[126] J. D. Jakeman, A. Narayan, and D. Xiu, "Minimal multi-element stochastic collocation for uncertainty quantification of discontinuous functions," *Journal of Computational Physics*, vol. 242, pp. 790–808, 2013.

[127] A. Gorodetsky and Y. Marzouk, "Efficient localization of discontinuities

in complex computational simulations," *SIAM Journal on Scientific Computing*, vol. 36, no. 6, pp. A2584–A2610, 2014.

[128] X. Ma and N. Zabaras, "An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations," *Journal of Computational Physics*, vol. 229, no. 10, pp. 3884–3915, 2010.

[129] X. Chen, Y. He, and D. Xiu, "An efficient method for uncertainty propagation using fuzzy sets," *SIAM Journal on Scientific Computing*, vol. 37, no. 6, pp. A2488–A2507, 2015.

[130] H. Rabitz and Ö. F. Aliş, "General foundations of high-dimensional model representations," *Journal of Mathematical Chemistry*, vol. 25, no. 2-3, pp. 197–233, 1999.

[131] G. Li, S.-W. Wang, C. Rosenthal, and H. Rabitz, "High dimensional model representations generated from low dimensional data samples. i. mp-cut-hdmr," *Journal of Mathematical Chemistry*, vol. 30, no. 1, pp. 1–30, 2001.

[132] G. Li, C. Rosenthal, and H. Rabitz, "High dimensional model representations," *The Journal of Physical Chemistry A*, vol. 105, no. 33, pp. 7765–7777, 2001.

[133] H. Xu and S. Rahman, "A generalized dimension-reduction method for

multidimensional integration in stochastic mechanics," *International Journal for Numerical Methods in Engineering*, vol. 61, no. 12, pp. 1992–2019, 2004.

[134] S. Shan and G. G. Wang, "Metamodeling for high dimensional simulation-based design problems," *Journal of Mechanical Design*, vol. 132, no. 5, p. 051009, 2010.

[135] T. J. Santner, B. J. Williams, and W. I. Notz, *The design and analysis of computer experiments*. Springer Science & Business Media, 2013.

[136] J. Zhang, S. Chowdhury, J. Zhang, A. Messac, and L. Castillo, "Adaptive hybrid surrogate modeling for complex systems," *AIAA journal*, vol. 51, no. 3, pp. 643–656, 2013.

[137] M. D. Shields, "Adaptive monte carlo analysis for strongly nonlinear stochastic systems," *Reliability Engineering & System Safety*, vol. 175, pp. 207–224, 2018.

[138] N. H. Kim, H. Wang, and N. V. Queipo, "Efficient shape optimization under uncertainty using polynomial chaos expansions and local sensitivities," *AIAA journal*, vol. 44, no. 5, pp. 1112–1116, 2006.

[139] J. Brezillon and R. Dwight, "Discrete adjoint of the navier-stokes equa-

tions for aerodynamic shape optimization," *Evolutionary and Deterministic Methods for Design, EUROGEN*, 2005.

[140] K. Giannakoglou, D. Papadimitriou, and I. Kampolis, "Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels," *Computer methods in applied mechanics and engineering*, vol. 195, no. 44-47, pp. 6312–6329, 2006.

[141] L. Laurent, R. Le Riche, B. Soulier, and P.-A. Boucard, "An overview of gradient-enhanced metamodels with applications," *Archives of Computational Methods in Engineering*, vol. 26, no. 1, pp. 61–106, 2019.

[142] I. M. Soboĺ, "On the distribution of points in a cube and the approximate evaluation of integrals," *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, vol. 7, no. 4, pp. 784–802, 1967.

[143] P. Bratley and B. L. Fox, "Algorithm 659: Implementing sobol's quasirandom sequence generator," *ACM Transactions on Mathematical Software (TOMS)*, vol. 14, no. 1, pp. 88–100, 1988.

[144] J. C. Helton and F. J. Davis, "Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems," *Reliability Engineering & System Safety*, vol. 81, no. 1, pp. 23–69, 2003.

[145] C. J. Sallaberry, J. C. Helton, and S. C. Hora, "Extension of latin hy-

percube samples with correlated variables," *Reliability Engineering & System Safety*, vol. 93, no. 7, pp. 1047–1059, 2008.

[146] M. Vořechovskỳ, "Hierarchical refinement of latin hypercube samples," *Computer-Aided Civil and Infrastructure Engineering*, vol. 30, no. 5, pp. 394–411, 2015.

[147] M. D. Shields, "Refined latinized stratified sampling: A robust sequential sample size extension methodology for high-dimensional latin hypercube and stratified designs," *International Journal for Uncertainty Quantification*, vol. 6, no. 1, 2016.

[148] H. Chi, P. Beerli, D. W. Evans, and M. Mascagni, "On the scrambled soboĺ sequence," in *International Conference on Computational Science*. Springer, 2005, pp. 775–782.

[149] H.-S. Chung and J. Alonso, "Using gradients to construct cokriging approximation models for high-dimensional design optimization problems," in *40th AIAA Aerospace Sciences Meeting & Exhibit*, 2002, p. 317.

[150] D. Liu, A. Litvinenko, C. Schillings, and V. Schulz, "Quantification of airfoil geometry-induced aerodynamic uncertainties—comparison of approaches," *SIAM/ASA Journal on Uncertainty Quantification*, vol. 5, no. 1, pp. 334–352, 2017.

[151] S. Soghrati, A. M. Aragón, C. Armando Duarte, and P. H. Geubelle, "An interface-enriched generalized FEM for problems with discontinuous gradient fields," *International Journal for Numerical Methods in Engineering*, vol. 89, no. 8, pp. 991–1008, 2012.

[152] S. Soghrati and P. H. Geubelle, "A 3D interface-enriched generalized finite element method for weakly discontinuous problems with complex internal geometries," *Computer Methods in Applied Mechanics and Engineering*, vol. 217-220, pp. 46–57, 2012.

[153] E. R. van Dam, "Two-dimensional minimax latin hypercube designs," *Discrete Appl. Math.*, vol. 156, no. 18, pp. 3483–3493, 2008.

[154] E. R. Van Dam, B. Husslage, D. Den Hertog, and H. Melissen, "Maximin latin hypercube designs in two dimensions," *Oper. Res.*, vol. 55, no. 1, pp. 158–169, 2007.

[155] S. Leary, A. Bhaskar, and A. Keane, "Optimal orthogonal-array-based latin hypercubes," *J. Appl. Stat.*, vol. 30, no. 5, pp. 585–598, 2003.

[156] A. Jourdan and J. Franco, "Optimal latin hypercube designs for the kullback–leibler criterion," *AStA Adv. Stat. Anal.*, vol. 94, no. 4, pp. 341–351, 2010.

[157] B. Beachkofski and R. Grandhi, "Improved distributed hypercube sam-

pling," in *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2002, p. 1274.

[158] M. D. Shields and J. Zhang, "The generalization of latin hypercube sampling," *Reliability Engineering & System Safety*, vol. 148, pp. 96–108, 2016.

[159] M. D. Shields, K. Teferra, A. Hapij, and R. P. Daddazio, "Refined stratified sampling for efficient monte carlo based uncertainty quantification," *Reliability Engineering & System Safety*, vol. 142, pp. 310–325, 2015.

[160] M. D. Buhmann, *Radial basis functions: theory and implementations*. Cambridge university press, 2003, vol. 12.

[161] R. Schaback, "Error estimates and condition numbers for radial basis function interpolation," *Advances in Computational Mathematics*, vol. 3, no. 3, pp. 251–264, 1995.

[162] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 36, no. 2, pp. 111–133, 1974.

[163] S. Rippa, "An algorithm for selecting a good value for the parameter c in radial basis function interpolation," *Advances in Computational Mathematics*, vol. 11, no. 2-3, pp. 193–210, 1999.

[164] M. Bompard, J. Peter, and J.-A. Desideri, "Surrogate models based on function and derivative values for aerodynamic global optimization," in *V European Conference on Computational Fluid Dynamics ECCOMAS CFD 2010*, 2010.

[165] B. Birge, "Psot-a particle swarm optimization toolbox for use with matlab," in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No. 03EX706)*. IEEE, 2003, pp. 182–186.

[166] L. Laurent, "Grenat(matlab/octavetoolbox) https://bitbucket.org/luclaurent/grenat," ., 2016.

[167] H. Pohlheim, "Examples of objective functions," *Retrieved*, vol. 4, no. 10, p. 2012, 2007.

[168] M. Ortiz and A. Pandolfi, "Finite-deformation irreversible cohesive elements for three-dimensional crack-propagation analysis," *International Journal for Numerical Methods in Engineering*, vol. 44, no. 9, pp. 1267–1282, 1999.

[169] E. J. Barbero, *Finite element analysis of composite materials using AbaqusTM*. CRC press, 2013.

[170] D. Brandyberry and P. H. Geubelle, "Multiscale sensitivity analysis of failure in composite materials using an interface-enriched generalized

fem," *32nd Technical Conference of the American Society for Composites 2017*, vol. 2, pp. 1044–1056, 2017.

[171] L. Maragliano and E. Vanden-Eijnden, "A temperature accelerated method for sampling free energy and determining reaction pathways in rare events simulations," *Chem. Phys. Lett.*, vol. 426, no. 1, pp. 168–175, 2006.

[172] K. Müller and L. D. Brown, "Location of saddle points and minimum energy paths by a constrained simplex optimization procedure," *Theor. Chimica Acta*, vol. 53, no. 1, pp. 75–93, 1979.

[173] S. Kumar, J. M. Rosenberg, D. Bouzida, R. H. Swendsen, and P. A. Kollman, "The weighted histogram analysis method for free-energy calculations on biomolecules. I. The method," *J. Comput. Chem.*, vol. 13, no. 8, pp. 1011–1021, 1992.

[174] A. Laio and F. L. Gervasio, "Metadynamics: a method to simulate rare events and reconstruct the free energy in biophysics, chemistry and material science," *Rep. Prog. Phys.*, vol. 71, no. 12, p. 126601, 2008.

[175] A. Barducci, G. Bussi, and M. Parrinello, "Well-tempered metadynamics: a smoothly converging and tunable free-energy method," *Phys. Rev. Lett.*, vol. 100, no. 2, p. 020603, 2008.

[176] S. Baday, E. A. Orabi, S. Wang, G. Lamoureux, and S. Bernèche, "Mechanism of NH_4^+ recruitment and NH_3 transport in Rh proteins," *Structure.*, vol. 23, no. 8, pp. 1550–1557, 2015.

[177] T.-Q. Yu, M. Lapelosa, E. Vanden-Eijnden, and C. F. Abrams, "Full kinetics of CO entry, internal diffusion, and exit in myoglobin from transition-path theory simulations," *J. Am. Chem. Soc.*, vol. 137, no. 8, pp. 3041–3050, 2015.

[178] A. Bucci and C. F. Abrams, "Oxygen pathways and allostery in monomeric sarcosine oxidase via single-sweep free-energy reconstruction," *J. Chem. Theory Comput.*, vol. 10, no. 7, pp. 2668–2676, 2014.

[179] S. Decherchi, A. Berteotti, G. Bottegoni, W. Rocchia, and A. Cavalli, "The ligand binding mechanism to purine nucleoside phosphorylase elucidated via molecular dynamics and machine learning," *Nat. Commun.*, vol. 6, p. 6155, 2015.

[180] R. Spezia, Y. Jeanvoine, C. Beuchat, L. Gagliardi, and R. Vuilleumier, "Hydration properties of Cm (III) and Th (IV) combining coordination free energy profiles with electronic structure analysis," *Phys. Chem. Chem. Phys.*, vol. 16, no. 12, pp. 5824–5832, 2014.

[181] F. Sterpone, S. Bonella, and S. Meloni, "Early stage of the dehydrogena-

tion of $NaAlH_4$ by ab initio rare event simulations," *J. Phys. Chem. C.*, vol. 116, no. 37, pp. 19 636–19 643, 2012.

[182] M. D. Buhmann, "Radial basis functions," *Acta Numer.*, vol. 9, pp. 1–38, 2000.

[183] G. S. Grest and K. Kremer, "Molecular dynamics simulation for polymers in the presence of a heat bath," *Phys. Rev. A.*, vol. 33, no. 5, p. 3628, 1986.

[184] A. D. MacKerell Jr, D. Bashford, M. Bellott, R. L. Dunbrack Jr, J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha *et al.*, "All-atom empirical potential for molecular modeling and dynamics studies of proteins," *J. Phys. Chem. B.*, vol. 102, no. 18, pp. 3586–3616, 1998.

[185] A. D. MacKerell Jr and N. K. Banavali, "All-atom empirical force field for nucleic acids: II. Application to molecular dynamics simulations of DNA and RNA in solution," *J. Comput. Chem.*, vol. 21, no. 2, pp. 105–120, 2000.

[186] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein, "Comparison of simple potential functions for simulating liquid water," *J. Chem. Phys.*, vol. 79, no. 2, pp. 926–935, 1983.

[187] G. Fiorin, M. L. Klein, and J. Hénin, "Using collective variables to drive

molecular dynamics simulations," *Mol. Phys.*, vol. 111, no. 22-23, pp. 3345–3362, 2013.

[188] J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kale, and K. Schulten, "Scalable molecular dynamics with NAMD," *J. Comput. Chem.*, vol. 26, no. 16, pp. 1781–1802, 2005.

[189] W. Humphrey, A. Dalke, and K. Schulten, "VMD: Visual molecular dynamics," *J. Mol. Graph.*, vol. 14, no. 1, pp. 33–38, 1996.

[190] B. T. Neyer, "Ad-optimality-based sensitivity test," *Technometrics*, vol. 36, no. 1, pp. 61–70, 1994.

[191] M. Standard, "V50 ballistic test for armor," MIL-STD-662F, Tech. Rep., 1997.

[192] G. Nilakantan, M. Keefe, E. D. Wetzel, T. A. Bogetti, R. Adkinson, and J. Gillespie, "Using ls-dyna® to computationally assess the v0-v100 impact response of flexible fabrics through probabilistic methods," in *11th International LS-DYNA® Users Conference: Proceedings of the International Users Conference (06-08 June 2010, Detroit)*, vol. 5, 2010, pp. 1–12.

[193] G. Nilakantan and S. Nutt, "State of the art in the deterministic and probabilistic ballistic impact modeling of soft body armor: filaments to

fabrics," in *American Society for Composites 29th Technical Conference, San Diego, CA, USA September*, 2014, pp. 8–10.

[194] Y. Wang, Y. Miao, L. Huang, D. Swenson, C.-F. Yen, J. Yu, and J. Q. Zheng, "Effect of the inter-fiber friction on fiber damage propagation and ballistic limit of 2-d woven fabrics under a fully confined boundary condition," *International Journal of Impact Engineering*, vol. 97, pp. 66–78, 2016.

[195] G. Nilakantan, S. Horner, V. Halls, and J. Zheng, "Virtual ballistic impact testing of kevlar soft armor: Predictive and validated finite element modeling of the v0-v100 probabilistic penetration response," *Defence technology*, vol. 14, no. 3, pp. 213–225, 2018.

[196] G. Nilakantan, "Experimentally validated predictive finite element modeling of the v0-v100 probabilistic penetration response of a kevlar fabric against a spherical projectile," *International Journal of Protective Structures*, vol. 9, no. 4, pp. 504–524, 2018.

[197] B. R. Ernest, "Dynamic programming," 2003.

[198] R. Ernest and E. Bellman, "Adaptive control processes: a guided tour," 1961.

[199] J.-R. Cano, P. A. Gutiérrez, B. Krawczyk, M. Woźniak, and S. García,

"Monotonic classification: An overview on algorithms, performance measures and data sets," *Neurocomputing*, vol. 341, pp. 168–182, 2019.

[200] J. O. Hallquist, "Ls-dyna® keyword users manual: volumes i, ii, and iii lsdyna r7. 1," *Livermore Software Technology Corporation, Livermore (LSTC), Livermore, California*, vol. 1265, 2014.

[201] C.-F. Yen, "A ballistic material model for continuous-fiber reinforced composites," *International Journal of Impact Engineering*, vol. 46, pp. 11–22, 2012.

[202] B. Z. G. Haque, "A progressive composite damage model for unidirectional and woven fabric composites," *User manual*, vol. 10, 2015.

[203] Z. Hashin, "Failure criteria for unidirectional fiber composites," *Journal of applied mechanics*, vol. 47, no. 2, pp. 329–334, 1980.

[204] A. Matzenmiller, J. Lubliner, and R. Taylor, "A constitutive model for anisotropic damage in fiber-composites," *Mechanics of materials*, vol. 20, no. 2, pp. 125–152, 1995.

[205] MATLAB, *2018a*. Natick, Massachusetts, USA: The MathWorks Inc., 2018.

[206] H. C. Elman and C. W. Miller, "Stochastic collocation with kernel density

estimation," *Computer Methods in Applied Mechanics and Engineering*, vol. 245, pp. 36–46, 2012.

[207] M. Grigoriu, "Response statistics for random heterogeneous microstructures," *SIAM/ASA Journal on Uncertainty Quantification*, vol. 2, no. 1, pp. 252–275, 2014.

[208] G. Deodatis and R. C. Micaletti, "Simulation of highly skewed non-gaussian stochastic processes," *Journal of engineering mechanics*, vol. 127, no. 12, pp. 1284–1295, 2001.

[209] S. Sakamoto and R. Ghanem, "Simulation of multi-dimensional non-gaussian non-stationary random fields," *Probabilistic Engineering Mechanics*, vol. 17, no. 2, pp. 167–176, 2002.

[210] H. Kim and M. D. Shields, "Modeling strongly non-gaussian non-stationary stochastic processes using the iterative translation approximation method and karhunen–loève expansion," *Computers & Structures*, vol. 161, pp. 31–42, 2015.

[211] D. G. Krige, "A statistical approach to some basic mine valuation problems on the witwatersrand," *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 52, no. 6, pp. 119–139, 1951.

[212] G. Matheron, "Principles of geostatistics," *Economic geology*, vol. 58, no. 8, pp. 1246–1266, 1963.

[213] S. Marelli and B. Sudret, "Uqlab: A framework for uncertainty quantification in matlab," in *Vulnerability, Uncertainty, and Risk: Quantification, Mitigation, and Management*, 2014, pp. 2554–2563.

[214] C. Lataniotis, S. Marelli, and B. Sudret, "Uqlab user manual–kriging (gaussian process modelling)," *Report UQLab-V0*, pp. 9–105, 2015.

[215] J. Nocedal, "Updating quasi-newton matrices with limited storage," *Mathematics of computation*, vol. 35, no. 151, pp. 773–782, 1980.

[216] R. H. Byrd, M. E. Hribar, and J. Nocedal, "An interior point algorithm for large-scale nonlinear programming," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 877–900, 1999.

[217] H. Akaike, "A new look at the statistical model identification," *IEEE transactions on automatic control*, vol. 19, no. 6, pp. 716–723, 1974.

[218] K. P. Burnham and D. R. Anderson, *Model selection and multimodel inference: a practical information-theoretic approach*. Springer Science & Business Media, 2003.

[219] J. E. Cavanaugh *et al.*, "Unifying the derivations for the akaike and

corrected akaike information criteria," *Statistics & Probability Letters*, vol. 33, no. 2, pp. 201–208, 1997.

Vita

Anindya Bhaduri was born in Kolkata, India on January 4, 1991. He attended Jadavpur University where he earned a Bachelor of Engineering (B.E.) in Mechanical Engineering in 2012. After that, he spent two years at the Indian Institute of Technology (IIT) Delhi, where he earned his Masters degree in Applied Mechanics with a focus on Design Engineering in 2014. Anindya joined the department of Civil Engineering at The Johns Hopkins University in Fall 2014 as a Ph.D. student under the supervision of Lori Graham-Brady. He received the Meyerhoff fellowship in 2014 to pursue his Ph.D.education. He was supported by the Hopkins Extreme Materials Institute and Army Research Lab for the rest of his Ph.D. career. His research focused on developing different surrogate modeling algorithms applied to problems in the field of computational mechanics. He completed the requirements for the degree of Doctor of Philosophy in September 2019.