ADAPTIVE LOCAL REDUCED BASIS METHOD FOR RISK-AVERSE PDE CONSTRAINED OPTIMIZATION AND INVERSE PROBLEMS

by

Zilong Zou

Department of Civil and Environmental Engineering
Duke University

Date: __________________________

Approved:

______________________________

Wilkins Aquino, Supervisor

______________________________

Johann Guilleminot

______________________________

Jianfeng Lu

______________________________

Sayan Mukherjee

______________________________

Guglielmo Scovazzi

Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Civil and Environmental Engineering in the Graduate School of Duke University

2018
ABSTRACT

ADAPTIVE LOCAL REDUCED BASIS METHOD FOR RISK-averse PDE CONSTRAINED OPTIMIZATION AND INVERSE PROBLEMS

by

Zilong Zou

Department of Civil and Environmental Engineering
Duke University

Date: ______________________

Approved:

__________________________

Wilkins Aquino, Supervisor

__________________________

Johann Guilleminot

__________________________

Jianfeng Lu

__________________________

Sayan Mukherjee

__________________________

Guglielmo Scovazzi

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Civil and Environmental Engineering in the Graduate School of Duke University

2018
Abstract

Many physical systems are modeled using partial differential equations (PDEs) with uncertain or random inputs. For such systems, naively propagating a fixed number of samples of the input probability law (or an approximation thereof) through the PDE is often inadequate to accurately quantify the risk associated with critical system responses. In addition, to manage the risk associated with system response and devise risk-averse controls for such PDEs, one must obtain the numerical solution of a risk-averse PDE-constrained optimization problem, which requires substantial computational efforts resulting from the discretization of the underlying PDE in both the physical and stochastic dimensions.

Bayesian Inverse problem, where unknown system parameters need to be inferred from some noisy data of the system response, is another important class of problems that suffer from excessive computational cost due to the discretization of the underlying PDE. To accurately characterize the inverse solution and quantify its uncertainty, tremendous computational efforts are typically required to sample from the posterior distribution of the system parameters given the data. Surrogate approximation of the PDE model is an important technique to expedite the inference process and tractably solve such problems.

In this thesis, we develop a goal-oriented, adaptive sampling and local reduced basis approximation for PDEs with random inputs. The method, which we denote by local RB, determines a set of samples and an associated (implicit) Voronoi partition of the parameter domain on which we build local reduced basis approximations of the PDE solution. The local basis in a Voronoi cell is composed of the solutions at a fixed number of closest samples as well as the gradient information in that cell. Thanks to the local nature of the method, computational cost of the approximation does not increase as more samples are included in the local RB model. We select the local RB samples in an adaptive and greedy manner using an \textit{a posteriori} error indicator based on the residual of the approximation.

Additionally, we modify our adaptive sampling process using an error indicator that is specifically targeted for the approximation of coherent risk measures evaluated at quantities
of interest depending on PDE solutions. This allow us to tailor our method to efficiently quantify the risk associated with the system responses. We then combine our local RB method with an inexact trust region method to efficiently solve risk-averse optimization problems with PDE constraints. We propose a numerical framework for systematically constructing surrogate models for the trust-region subproblem and the objective function using local RB approximations.

Finally, we extend our local RB method to efficiently approximate the Gibbs posterior distribution for inverse problems under uncertainty. The local RB method is employed to construct a cheap surrogate model for the loss function in the Gibbs posterior formula. To improve the accuracy of the surrogate approximation, we adopt a Sequential Monte Carlo framework to guide the progressive and adaptive construction of the local RB surrogate. The resulted method provides subjective and efficient inference of unknown system parameters under general distribution and noise assumptions.

We provide theoretical error bounds for our proposed local RB method and its extensions, and numerically demonstrate the performance of our methods through various examples.
Acknowledgements

I would like to thank all my advisors, colleagues, collaborators and funding agencies for their support in my Ph.D. study. It would not be possible for me to finish the Ph.D. study without their kindness, help and support.

I would like to express my deepest gratitude to my academic advisor Prof. Wilkins Aquino, who provided me a free, relaxing yet stimulating environment to explore my full potential in research. I appreciate his invaluable guidance and financial support in completing this research. His always inspiring advice and constant encouragement helped me through the difficult times of my Ph.D. study. The lessons and skills I have learned from him will continue helping me in my personal life as well as my professional career.

I am also thankful to my excellent committee members. Prof. Sayan Mukherjee introduced me to Bayesian inference and offered insightful advice on my work with Gibbs posterior. Prof. Guglielmo Scovazzi helped me deepen my understanding about finite element method through his wonderful courses and discussions. Prof. Jianfeng Lu taught me functional analysis and helped me to gain the necessary mathematical skills to complete the proofs in this thesis. Prof. Johann Guilleminot helped me to broaden my knowledge in the field of uncertainty quantification. I feel honored to have such a great committee and deeply appreciate all the guidance and help I have received from them.

I am also lucky to have been working with excellent collaborators during my research. I deeply appreciate Dr. Drew Kouri for his invaluable guidance and help on my work with risk measures and optimization. I am also grateful to Dr. Denis Ridzal for his help on the Rapid Optimization Library (ROL) and the C++ implementation of my work.
In addition, I want to thank the Duke CEE department, who has provided me with a wonderful working environment. My lab mates and close friends have been very helpful and supportive throughout my years at Duke. They have made my Ph.D. life so much easier and enjoyable.

Last but not least, I want to express my deep gratitude to my beloved families, who are the most important individuals in my life. Their persistent love enabled me to reach this far and will motivate me to achieve even more. I dedicate this work to my son, my wife and my parents.
Contents

Abstract iv

Acknowledgements vi

List of Figures xii

List of Tables xvii

1 Introduction 1

1.1 Motivation .......................................................... 1
1.2 Literature review ...................................................... 4
  1.2.1 PDEs with random inputs ...................................... 4
  1.2.2 Reduced basis method ......................................... 8
  1.2.3 PDE-constrained optimization under uncertainty .......... 9
  1.2.4 Bayesian inverse problems .................................... 11
1.3 Contributions and layout of the thesis .......................... 13

2 The Local Reduced Basis Method for PDEs with Random Inputs 17

2.1 Formulations ......................................................... 17
2.2 Deterministic samples and local approximation ................. 20
2.3 The local reduced basis approximation .......................... 23
2.4 A posteriori error estimation and adaptive sampling ............ 27
2.5 Computational details and cost analysis ......................... 31
  2.5.1 The offline-online computing strategy ..................... 31
  2.5.2 Approximating the inf-sup constant ....................... 33
  2.5.3 Computational cost .......................................... 37
2.6 Numerical examples ................................................ 40

viii
2.6.1 1D advection diffusion equation ............................................... 40
2.6.2 1D Helmholtz equation with damped resonance .......................... 43
2.6.3 2D advection diffusion equation ............................................... 48

3 Evaluating Risk Measures 55
3.1 Risk measures ............................................................................ 55
  3.1.1 Coherent risk measures ......................................................... 55
  3.1.2 Optimized certainty equivalent ............................................. 57
3.2 Approximating coherent risk measures ...................................... 59
3.3 Numerical examples .................................................................. 60
  3.3.1 1D advection diffusion equation ........................................... 61
  3.3.2 1D Helmholtz equation with damped resonance ....................... 62
  3.3.3 2D advection diffusion equation ........................................... 64

4 Risk-averse PDE-constrained Optimization 67
4.1 Risk-averse control problem with PDE constraint ....................... 68
4.2 Inexact trust region algorithm with local RB surrogates ............... 70
  4.2.1 Inexact trust region method .................................................. 70
  4.2.2 Inexact trust regions with local RB approximation ................... 72
4.3 Examples .................................................................................... 82
  4.3.1 1D Helmholtz equation with damped resonance ....................... 82
  4.3.2 1D advection diffusion equation ........................................... 88
  4.3.3 2D Advection-diffusion equation ........................................... 93

5 Particle-based Approximation of the Gibbs Posterior for Inverse Problems 98
5.1 Gibbs Posterior for inverse problems ........................................ 99
5.2 Surrogate approximation ......................................................... 101
5.3 Particle based approximation with local RB surrogate . . . . . . . . . . . . 105
  5.3.1 Convergence analysis for fixed particles . . . . . . . . . . . . . . . . 106
5.4 An adaptive Sequential Monte Carlo method for particle evolution . . . . . . 112
  5.4.1 The Sequential Monte Carlo method . . . . . . . . . . . . . . . . . . 113
  5.4.2 Adaptive selection of the SMC step size . . . . . . . . . . . . . . . . 118
5.5 Choosing the weight for the loss function . . . . . . . . . . . . . . . . . . 120
5.6 Numerical examples . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 124
  5.6.1 1D advection diffusion equation . . . . . . . . . . . . . . . . . . . . . 124
  5.6.2 2D advection diffusion equation . . . . . . . . . . . . . . . . . . . . 126
  5.6.3 2D elasticity equation . . . . . . . . . . . . . . . . . . . . . . . . . 131

6 Conclusions 138
  6.1 Summary and Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . 138
  6.2 Possible future works . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 140

A C++ implementation in the Rapid Optimization Library (ROL) 143
  A.1 The local reduced basis method . . . . . . . . . . . . . . . . . . . . . . . 143
    A.1.1 The local reduced basis manager class . . . . . . . . . . . . . . . . 143
    A.1.2 The PDE interface class . . . . . . . . . . . . . . . . . . . . . . . . 146
    A.1.3 The reduced space solver class . . . . . . . . . . . . . . . . . . . . 147
  A.2 Risk-averse optimization . . . . . . . . . . . . . . . . . . . . . . . . . . . 148
    A.2.1 The local reduced basis PDE constraint class . . . . . . . . . . . . 149
    A.2.2 The risk measure tools class . . . . . . . . . . . . . . . . . . . . . 149
    A.2.3 The local reduced basis sampler class . . . . . . . . . . . . . . . . 150
  A.3 An example . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 150
  A.4 A brief comment on future development . . . . . . . . . . . . . . . . . . . 154
List of Figures

<table>
<thead>
<tr>
<th>Figure Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Illustration of the local basis for local RB method.</td>
<td>24</td>
</tr>
<tr>
<td>2.2</td>
<td>The location of 5, 20 and 40 atoms and their corresponding Voronoi cells</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>generated using (a) SROM with local approximation and (b) local RB.</td>
<td></td>
</tr>
<tr>
<td>2.3</td>
<td>Comparison of the SROM with local approximation and the local RB method:</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>moments of the PDE solution.</td>
<td></td>
</tr>
<tr>
<td>2.4</td>
<td>Comparison of the SROM with local approximation and the local RB method:</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>CDFs of the PDE solution.</td>
<td></td>
</tr>
<tr>
<td>2.5</td>
<td>Comparison of the SROM with local approximation and the local RB method:</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>joint distribution of the PDE solution.</td>
<td></td>
</tr>
<tr>
<td>2.6</td>
<td>Comparison of the SROM with local approximation and the local RB method:</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>relative $L^2(D)$-error over test samples.</td>
<td></td>
</tr>
<tr>
<td>2.7</td>
<td>The $L^2(D)$ norm of $u(\cdot, \xi)$ over the parameter domain $\Xi$ with</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>(a) $c = 0.01$ (b) $c = 0.2$. Notice that resonances occur for certain</td>
<td></td>
</tr>
<tr>
<td></td>
<td>realizations of $\xi$.</td>
<td></td>
</tr>
<tr>
<td>2.8</td>
<td>Sample solutions of the stochastic Helmholtz problem.</td>
<td>46</td>
</tr>
<tr>
<td>2.9</td>
<td>(a) Joint density of $\xi$. (b) Background samples of $\xi$.</td>
<td>47</td>
</tr>
<tr>
<td>2.10</td>
<td>Atoms, partition and training set generated by the local RB method when</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>$N_{\text{atom}} = 10, 25, 50$, respectively.</td>
<td></td>
</tr>
<tr>
<td>2.11</td>
<td>Comparison of the adaptive sparse grid methods and the local RB method:</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>relative $L^2(D)$-error over test samples.</td>
<td></td>
</tr>
<tr>
<td>2.12</td>
<td>Comparison of the adaptive sparse grid methods and the local RB method:</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>mean of the PDE solution.</td>
<td></td>
</tr>
<tr>
<td>2.13</td>
<td>Comparison of the adaptive sparse grid methods and the local RB method:</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>variance of the PDE solution.</td>
<td></td>
</tr>
<tr>
<td>2.14</td>
<td>Comparison of the adaptive sparse grid methods and the local RB method:</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>CDFs of the PDE solution.</td>
<td></td>
</tr>
</tbody>
</table>
2.15 Comparison of the adaptive sparse grid methods and the local RB method: relative $L^2(D)$-error over test samples. .................................................. 52

2.16 The approximated mean and variance of $u$ using Monte Carlo method with 50,000 independent parameter samples. .................................................. 53

2.17 Comparison of the adaptive sparse grid methods and the local RB method: CDFs of the PDE solution. .................................................. 54

3.1 First six steps of local RB method tailored for approximating $\text{CVaR}_\beta(G)$. .................................................. 61

3.2 Comparison of the SROM with local approximation and the local RB method: the QoI $G$ and the risk $\text{CVaR}_\beta(G)$. .................................................. 62

3.3 (a) The true QoI, (b) the parameter samples that contribute to $\text{CVaR}_\beta(G)$ are shown in red, (c) the cumulative distribution of $G(\xi)$ and $\text{CVaR}_\beta(G)$ . .................................................. 63

3.4 (a) First six steps of local RB method tailored for approximating $\text{CVaR}_\beta(G)$. (b) The relative error of risk using varying numbers of atoms. .................................................. 64

3.5 Comparison of the adaptive sparse grid methods and the local RB method: the QoI $G$ and the risk $\text{CVaR}_\beta(G)$. .................................................. 65

3.6 The relative error of $\text{CVaR}_{0.9}(\|u(\xi)\|_{L^2(D)})$ obtained by the local RB method with different number of atoms. .................................................. 66

3.7 Comparison of the $\text{CVaR}_{0.9}(\|u(\xi)\|_{L^2(D)})$ obtained using ASG1, ASG2 and local RB surrogate models. .................................................. 66

4.1 Comparison of optimal control and QoI $G(u(\xi), \xi)$ obtained by the full-order Monte Carlo reference and our TR/RB method. .................................................. 84

4.2 The norm of the true gradient evaluated at the iterates produced by our proposed TR/RB approach. .................................................. 85

4.3 The atoms of the local RB surrogate models. .................................................. 85

4.4 A comparison of the total number of full PDE solves for the reference Monte Carlo approach and our proposed TR/RB approach. .................................................. 86

4.5 Comparison of optimal control obtained by the full-order Monte Carlo reference and our TR/RB method with different $\alpha$. .................................................. 86
4.6 Comparison of the total number of full PDE solves for the full-order Monte Carlo reference and our TR/RB method with different $\alpha$. 87

4.7 The CDF and statistics of the QoI $G(u(\xi), \xi)$ under optimal controls with different $\alpha$. 88

4.8 The QoI $G(u(\xi), \xi)$ evaluated under the optimal controls when $\alpha = 0.0$ and $\alpha = 1.0$. 88

4.9 The parameter samples that contribute to the tail risk $\text{CVaR}_{0.9}[G(u(\xi), \xi)]$ under the optimal controls when $\alpha = 0.0$ and $\alpha = 1.0$. 89

4.10 Optimal controls with reference obtained by minimizing expectation, CVaR and entropic risk applied to $G(u(\xi), \xi)$. 90

4.11 The QoI $G(u(\xi), \xi)$ evaluated under the optimal controls. 91

4.12 The CDFs of $G(u(\xi), \xi)$ evaluated under the optimal controls. 91

4.13 The norm of the true gradient evaluated at the iterates produced by our proposed TR/RB approach. 92

4.14 The final atoms of the surrogate models $S_{\text{mod}} \text{(red)}$ and $S_{\text{obj}} \text{(black)}$ which were used to construct $m_k$ and $\hat{J}_k$. 92

4.15 A comparison of the total number of full PDE solves for the reference Monte Carlo approach and our proposed TR/RB approach. 93

4.16 The controls obtained by the TR/RB method with the reference. 96

4.17 The CDFs of $G(u(\xi); \xi)$ under the optimal controls. 96

4.18 The norm of the true gradient evaluated at the iterates produced by our proposed TR/RB approach. 97

4.19 A comparison of the total number of full PDE solves for the reference Monte Carlo approach and our proposed TR/RB approach. 97

5.1 Comparison of the proposed SMC method and the standard MCMC method with Gaussian likelihood function. 126

5.2 SMC iteration 0, with a loss weight $W_0 = 0$. 127

xiv
A.3 Local RB atoms and the adaptive training samples with 5, 10 and 30 atoms respectively. 152

A.4 The mean and standard deviation of the relative errors of surrogate solution over 200 test samples with different number of atoms in the local RB surrogate. 152

A.5 The reference and local RB solutions of the risk-averse control problem. 153

A.6 Number of full PDE solves incurred in reference and local RB solutions. 154

A.7 The reference and local RB solutions of the risk-averse control problem. 154
List of Tables

2.1 Relative $L_2(D)$ error of the predicated moments of $u$ using surrogate models constructed with 1,600 full PDE solves. ................................. 52

3.1 The relative error of CVaR$_{0.9}(G)$ obtained using ASG1, ASG2 and local RB surrogate models with 60 and 600 PDE solves, respectively. ...................... 64

3.2 The relative error of CVaR$_{0.9}(G)$ obtained using ASG1, ASG2 and local RB surrogate models (constructed with 1,600 PDE solves). ......................... 65
Chapter 1

Introduction

1.1 Motivation

Many science and engineering systems can be modeled by partial differential equations (PDE). Numerical simulations of such systems allow engineers to derive insights into the critical system behavior and to make predictions and hypotheses about certain outputs of interest. However, modeling and numerical simulation as such are usually riddled with uncertainty induced by random forces and coefficients, unknown boundary and initial conditions, and unverifiable modeling assumptions. It is usually inadequate to simulate the system without quantifying the effect of the uncertainty on the system response. To this end, PDEs with random inputs (parameters) or stochastic PDEs are employed to account for the uncertainty. Many numerical methods have been proposed in the literature to solve such PDEs with random inputs; see, e.g., [GS03, BTZ04, DBO01, XK03, XH05, MK05, BNR00, NTW08b, BNT07, NTW08a]. However, these methods are usually very expensive when the stochastic dimension is high. This is commonly referred to as the “curse of dimensionality”. That is, the computational cost grows exponentially as a function of the number of random variables.

To further complicate matters, in many high consequence engineering applications, it is critical to accurately approximate the “risk” associated with the statistical tails of the system responses, not just statistical moments [BB90, Rac01, SPK04, CQ13, HKTW17]. In such settings, the standard deviation as a risk measure is not sufficient since it does not accurately characterize the probability or magnitude of rare events. To capture the tail response with sufficient accuracy, typically a large number of samples of the inputs needs to be propagated through the PDE, further exacerbating the problem of computational cost.
A key to achieving computational savings is to recognize that naively propagating the
input probability law (or an approximation thereof) is usually inefficient because the input
law may not accurately capture the behavior of critical system responses that depend
on the PDE solution. Adaptive methods are hence superior due to their ability to take
advantage of the system output to bias their computational efforts towards parameter
regions where solution varies rapidly or that are most important to the computational goal.
Though a plethora of research works have been devoted to the development of adaptive
methods [AO10, WK05, GZ07, NTW08a, MZ09a, Shi18], the need to solve more challenging
and large-scaled problems in practice fuels the continuing efforts to develop novel and
more efficient methods. Therefore, in this thesis, we develop a goal-oriented adaptive local
reduced basis (local RB) method that are specifically targeted at approximating quantity
of interests (QoI) governed by stochastic PDEs. Our approach is very efficient for a wide
class of PDEs with uncertain parameters that allow an affine decomposition with respect
to the parameters and is versatile enough to be tailored for more specific goals such as
quantifying coherent risk measures applied to the QoIs.

Evaluating the response statistics or risk are usually not the only goals for an engineering
problem. In many cases, we want to use the numerical simulation to help with the control or
design of the engineering system. Examples can be found in topology optimization [Ben01],
material design [SAW18], optimal flow control [Gun03], oil field research [SAD+05], etc.
With uncertainties in play, it is critical that we determine optimal solutions that, in some
sense, mitigate the underlying risk associated with large undesirable outcomes. These tasks
can be modeled by risk-averse PDE-constrained optimization problems.

The solution to risk-averse optimization problems has the potential to endow engi-
neering systems with superior performance against uncertain environment than otherwise
possible. Unfortunately, after discretization, these problems become large-scale nonlin-
ear and often nonconvex stochastic programming problems. Solving such fully discretized
problem, in many cases, is extremely expensive if not infeasible. Only a few existing works
have attempted to solve such problems in practice [KS16, APSG17]. The application of
risk-averse PDE-constrained optimization is rather limited due to the inherent computational challenges. To tractably solve these problems, managing the discretization fidelity in the physical and stochastic dimensions throughout the optimization procedure is essential. To this end, we combine our local RB method with an inexact trust region method [KHRvBW13] to efficiently tackle risk-averse optimization problems with PDE constraints. We propose a numerical framework for systematically constructing surrogate models for the trust-region subproblem and the objective function using local RB approximations that guarantees global convergence.

Another field we explore in this thesis is the Bayesian inverse problems where we need to infer the unknown parameters in the governing PDE from noisy measurement of the system response. Such problems are ubiquitous in many application areas including medical imaging [Kou12, FK16], heat conduction [WZ04], geosciences [BTGMS13], atmospheric and oceanic sciences [Ben05]. The Bayesian approach has been a foundation for performing such inference from noisy or incomplete observations while allowing us to quantify the uncertainty of the inverse solution due to the inexactness of data [CDS10, Stu10]. However, the majority of the existing Bayesian methods impose an exact noise model which need not hold in general. Computational cost is another problem for Bayesian inverse problems, since exploring the posterior distribution requires repetitive solutions of the PDE under different parameters. Plenty of efforts have been devoted to expedite such posterior exploration using both more efficient sampling techniques [N+11, MWBG12, EMM12] or surrogate models [FMWvBW10, MNR07, MX09, MN09, MZ09b, LM14, MPL16].

In this thesis, we adopt a general framework based on Gibbs posterior [BHW16] for updating belief distributions for inverse problems governed by PDEs. In contrast with existing Bayesian methods where the distributions of the noise are assumed to be known exactly, the Gibbs posterior update does not require a likelihood function thus no exact model for the noise is needed. Instead, the Gibbs posterior are applicable where the unknown parameters are connected to the data through a loss function. Specifying such a loss function can be a simpler task than an explicit likelihood function that requires the knowledge of the true
data generating mechanism. Furthermore, to manage the computational cost, we employ our local RB method as an surrogate model for parameter evaluation. The accuracy and efficiency of local RB is fully exploited to accelerate the inference through a particle-based approximation to the Gibbs posterior.

1.2 Literature review

The work in this thesis is built on several foundations including numerical solutions for PDEs with random parameters, reduced basis approximation, PDE-constrained optimization under uncertainty as well as Bayesian inference for inverse problems. This section briefly reviews the state-of-art of each topic. The content here is by no means exhaustive. It solely presents the relevant background material for our method.

1.2.1 PDEs with random inputs

In this work, we are primarily interested in applications involving PDEs with random inputs (parameters). Extracting the accurate statistics of the output from such PDEs is challenging because it typically demands high computational cost, especially when the stochastic dimensionality (e.g. number of random inputs) is large.

Significant effort has been devoted to developing efficient methods for solving PDEs with random inputs. Monte Carlo methods [Fis13] are the most general and widely used due to their non-intrusiveness, simplicity as well as the fact that its convergence, though slow, does not depend on the number of stochastic dimensions. However, for many problems, its slow convergence makes the computational cost of Monte Carlo methods prohibitively high. This is because the number of Monte Carlo samples required to obtain accurate statistics is typically very large. When the associated deterministic PDE solution is expensive, the Monte Carlo method quickly becomes infeasible. Other variants of Monte Carlo methods have hence been devised to produce faster convergence rate. In particular, methods based on quasi-Monte Carlo [Nie78], Latin hypercube sampling [Ste87, HD03], importance sampling
[GI89], adaptive Monte Carlo [Shi18] and etc. have been shown to alleviate the slow converge of the Monte Carlo method to some extent.

The stochastic Galerkin and stochastic collocations methods, on the other hand, rely on polynomial interpolation in the stochastic space, which provides exponentially convergent approximation provided the PDE solution is sufficiently regular with respect to the random inputs. In the stochastic Galerkin method, a complete polynomial basis is chosen for approximation in the stochastic space and the expansion coefficients are sought by solving a coupled system of equations, which is derived by projecting the original stochastic equation onto the polynomial basis via a Galerkin projection scheme [GS03, BTZ04, DBO01]. Different polynomial bases can be chosen for different distributions of the underlying random variable in order to achieve optimal convergence rate [XK02b, XK02a]. The biggest drawback of this method is the coupled nature of the system of equations, which tends to become extremely complex as the number of stochastic dimensions or the number of basis functions increases. The system size grows exponentially with the increase of stochastic dimensions.

The stochastic collocation method [XH05, BNT07, NTW08b] combines the convergence property of stochastic Galerkin method with the fully decoupled computing strategy of Monte Carlo method. In the conventional stochastic collocation method, solutions are obtained at a structured tensor grid over the parameter domain and Lagrangian polynomials are employed to interpolate the sample solutions. The number of grid points where sample solutions are needed increase exponentially with the number of stochastic dimensions. In recent years, sparse grid collocation methods [GG98, BNR00, XH05, NTW08b] based on Smolyak algorithm [Smo63] have been proposed to mitigate the curse of dimensionality to some extent. Sparse grid based interpolation can achieve the same level of interpolation accuracy with orders of magnitude fewer collocation points than the conventional collocation approaches. However, the number of collocation points for sparse grid still depends on the number of stochastic dimensions in an exponential manner.

Adaptive collocation methods are promising candidates to mitigate or even overcome
the curse of dimensionality. For many problems, the number of dominant stochastic dimensions that are most influential to the solution is typically much smaller than the total number of random inputs. Being able to identify such dominant stochastic dimensions allows large computational savings by biasing the computation efforts towards those dimensions with only little sacrifice of accuracy. Adaptive sparse grid methods are built to automatically adjust the computational efforts to the dominant stochastic dimensions. In particular, the dimension-adaptive anisotropic sparse grid method based on a generalized sparse grid construction scheme [GG03] has been developed and applied to stochastic problems in [GZ07, NTW08a]. A locally adaptive sparse grid method using multilinear functions with local support has been proposed in [MZ09a]. In these methods, a so-called hierarchical surplus from the sequential construction process of sparse grid is employed as an error indicator to automatically detect the dimensions (or subdomains) that require further refinement.

There also exists a class of methods that utilizes local truncated Taylor expansion in the stochastic dimensions to approximate the solution. The idea was initially explored in [EN06, EN07] based on a geometric partition of the parameter space. In each partition, a first order truncated Taylor expansion is adopted to approximate the quantity of interest. The authors also proposed an error indicator and an adaptive refinement strategy in order to sequentially refine the parameter space where an large error indicator is present. The authors applied the method to several nonlinear stochastic ODE problems and demonstrated the potential of the method to gain orders of magnitude faster convergence than the conventional Monte Carlo method in terms of computing the statistics of interest.

The author in [Gri12] combined such a local approximation approach with a stochastic reduced order model (SROM) [Gri09] based partition of the parameter domain. An SROM is a discrete set of atoms (with an associated probability vector) that can approximate the original random parameters in some prescribed metric. The SROM generates a partition of the parameter space through Voronoi-cells. Such SROM based partition of parameter space is more general because it uses the probability law of the parameters. The author
then employs the same truncated Taylor expansion within each cell to approximate the solution. This method was later applied to study the response of random heterogeneous microstructures in [Gri14] and compared with stochastic Galerkin and collocation methods in [FGE15]. Additionally, because the method is essentially sample-based and utilizes a local approximation scheme, it is, as for the Monte Carlo method, relatively immune to the curse of dimensionality.

Taylor expansion approximations are reasonably accurate as long as the size of the partition cells is small enough. However, for problems in which the gradient of the state variable varies rapidly with the random parameters, the truncated Taylor expansion is likely to yield a large error, which can only be mitigated by refining the local partition with additional atoms. The potential number of atoms required to generate a sufficiently refined partition so that the local approximation is accurate can be quite large and leads to excessive computational cost. In addition, the SROM is only built to approximate the input random parameters. A good representation of the random inputs does not necessarily yield a good approximation to the output of the stochastic PDEs.

More recently, reduced basis (RB) methods [Pet89, RHP08, EPR10, QRM11], which were originally developed for model reduction in parametric problems, have also been applied to the stochastic PDE problems by a few researchers [BLBL+10, BLBM+09, CQR13, CQ13, CQR14, CQ15, CQR17]. Coupled with a greedy sampling procedure and an offline-online computing strategy, the certified RB method [QRM11] has been used to accurately approximate the solution of parametrized PDEs with relatively low computational costs. In particular, the authors in [CQR14] compared the performance of a RB method and stochastic collocation method for affinely parametrized elliptic problems. They concluded that the RB method tends to perform better for large scale and high dimensional problems. However, one problem of these reduced basis methods is that potentially a large number of basis is required when the variability of the PDE solution with respect to the parameters is high, in which case even the reduced problems can become large and expensive to solve repetitively. Because RB methods are a backbone to our work, we will explore in more
depth the literature on these approaches in the next section.

For large scale PDE problems with high dimensional random inputs, it is still a very challenging task to obtain accurate solutions with affordable computational efforts. Therefore, a key motivation for our work is to develop a theoretical and computational framework that allows more efficient approximation of the solution to such PDEs and are scalable to high dimensional stochastic problems.

1.2.2 Reduced basis method

Our proposed local RB method falls into the category of RB methods where an inexpensive projection-based approximation model is used as a proxy to compute the desired QoI or statistics of the PDE solution. We briefly review these approaches in this section.

In essence, the RB method employs a variational framework and constructs a set of basis from solutions of the underlying PDE at different samples of the parameters [RHP08, QRM11, CQR13, CQR17]. These samples are selected from a set of training samples of the parameters in a sequential and greedy manner according to an inexpensive error bound [RHP08]. To this end, residual based error bounds are typically employed to assess the accuracy of the RB approximation over the training set. The RB method then selects its next sample from the training set that maximizes the error bound.

Another key ingredient in the RB method is the offline-online computing strategy. Thanks to the assumption of affine parametrization of the PDE, expensive computations such as constructing the basis and the affine components of the reduced Jacobian and residual can be carried out in an offline stage. Once such information are precomputed and stored, online evaluation of the RB approximation at a new parameter takes negligible computational cost that is independent of the dimension of the full PDE model [RHP08].

However, building an inexpensive error bound is a nontrivial task that requires significant computation itself, which is especially pronounced with noncoercive or nonsymmetric problems [HRSP07, RHP08]. For the latter problem with large parameter dimension, constructing such bound can easily become a major computational bottleneck. In addition,
in standard RB methods, the dimension of the reduced problem grows with the number of samples used to construct the basis. When the size of the basis becomes large, which typically happens when the variability of the PDE solution with respect to the parameters is high, the reduced problems can become large and expensive to solve repetitively.

Finally, for PDEs that do not possess an affine dependence on their parameters, we need an additional level of approximation to recover the computational efficiency of RB methods. Because in such problems, the Jacobian, residual and nonlinear terms in general are not amenable to precomputation, the online RB approximation requires full assembly of such ingredients, incurring a cost similar to solving the full PDE problem. Fortunately, many nonaffine problems can be approximated by affine counterparts using methods like empirical interpolation [BMNP04, CS10, DHO12]. Throughout the thesis, we shall focus on affine problems that allow us to exploit the offline-online computing strategy. A detailed computational cost analysis of our local RB method is given in Section 2.5.

1.2.3 PDE-constrained optimization under uncertainty

PDE-constrained optimization under uncertainty is a relatively new field that combines ideas from both stochastic programming [SDR14] and PDE-constrained optimization. Stochastic programming offers many numerical schemes for solving problems with uncertainty, such as the sample average approximation (SSA), also known as Monte Carlo method, and the stochastic approximation algorithm (SA) [SDR14, Mar05, PAS09]. While there is a rich literature on theoretical and computational aspects of PDE-constrained optimization [HPUU08, BGHvBW03, IIK10], few of existing works have addressed such problems under uncertainty [BSSW10, KS16, APSG17]. The principal difficulty with these problems is the tremendous cost that comes with the discretization in both physical and stochastic dimensions, which when further embedded into an optimization framework, renders the problems computational intractable. Hence managing the discretization fidelity in the physical and stochastic dimensions throughout the optimization procedure is the key to tractably solving these problems.
In this thesis, we shall mainly consider risk-averse optimal control problems for PDEs with random inputs. We focus on PDEs with uncertain coefficients and model risk aversion using the optimized certainty equivalent risk measures [BTT07]. One important optimized certainty equivalent risk measure is the conditional value at risk (CVaR) [RU00]. CVaR is a coherent risk measure [ADEH99] and is commonly used in financial applications to determine risk-averse investment strategies [KPU02]. The difficulty with minimizing CVaR and coherent risk measures in general is that they typically are nonsmooth. Therefore, efficient derivative-based optimization algorithms are often not applicable. Furthermore, accurately computing the risk measure value using, e.g., SSA requires a large number of samples and hence a large number of PDE solves. Recent work [KS16] has shown the feasibility of minimizing CVaR in the context of PDE-constrained optimization problems. However, the computational cost to numerically obtain a solution is still overwhelming.

Our goal is to devise a computational framework to efficiently solve risk-averse PDE-constrained optimization problems using cheap surrogate models for the PDE solutions. Few methods exist for constructing efficient surrogate models in the context of PDE-constrained optimization under uncertainty. For example, the authors in [KHRvBW14] introduce an adaptive sparse-grid approach whereas the author in [Zah16] introduces a global RB method. However, to obtain convergence of the optimization algorithm when using surrogate models, the errors associated with the surrogate approximation need to be properly managed. Otherwise, the bias introduced by the surrogate approximation could lead to nonconvergence or a solution that is not a (local) minimizer of the original problem. To this end, both [KHRvBW14] and [Zah16] employ an inexact trust-region (TR) framework to manage the error in the surrogate approximations throughout the optimization process. Additionally, under standard assumptions, the authors in [KHRvBW14] show that the inexact TR algorithm is guaranteed to converge from any initial guess, provided that errors in the approximation of the objective function and its gradient are adequately bounded.

The inexact TR method can be combined with our local RB approximation to yield
a globally convergent and efficient computational framework for solving risk-averse PDE-constrained optimization problems. We present such a framework in Chapter 4.

### 1.2.4 Bayesian inverse problems

Similar to PDE-constrained optimization problems under uncertainty, Bayesian inverse problems also suffer from excessive computational cost due to the discretization of the underlying PDE. Inverse problems involve estimating unknown model parameters from a set of observations, which is as apposed to the forward problem where parameters are mapped through the model to some quantities that can be measured. Inverse problem permeates all fields of science and engineering ranging from medical imaging [SJHWU+11, BWAB13] to geosciences [FPSIK16, BTGMS13], atmospheric and oceanic sciences [Ben05].

Bayesian methods provide a natural way to integrate noisy or incomplete data into our prior belief about the model parameters and a tool to quantitatively assess the uncertainty in the inverse solution. The solution to Bayesian inverse problem is a probability distribution over the parameter space, which is referred to as the posterior distribution. Except for in very limited settings, e.g., a linear model with Gaussian prior and Gaussian noise, the analytical form of the posterior distribution is rarely tractable. In most cases, one can only hope to approximate the posterior either through sampling or parametrization.

Markov Chain Monte Carlo (MCMC) [GL06] is the mostly well-known and versatile method for Bayesian inverse problems [Stu10]. MCMC requires only pointwise evaluation of the likelihood function to generate a stream of samples from the posterior distribution that can be subsequently used to compute the statistics of the posterior. In MCMC, however, the generation of each new sample requires one or more evaluations for the forward model, which is extremely expensive considering that the forward model is a fully discretized PDE. To accelerate MCMC, various inexpensive surrogate models have been proposed in the literature to approximate the forward evaluation in the sampling procedure, for example, stochastic spectral methods are used in [MNR07], Gaussian process regression is used in [KO01] and projection-based model reduction is employed in [CMW15, MPL16].
The surrogate models are typically constructed to be accurate over the support of the prior distribution [FMWvBW10, MNR07, MX09, MN09, MPL16] and are thought to be “globally accurate”. However, thanks to the information contained in the observations, the posterior distribution typically concentrate on a much smaller portion of the support of the prior, in which case requiring a “globally accurate” surrogate model seems unnecessary and inefficient. To this end, several recent studies have exploited such information (or posterior) and build adaptive and data-driven surrogate models that are more efficient and accurate on the support of the posterior [CMW15, LM14, Con14, CMPS16]. Significant computational savings can be realized through such adaptive methods.

Sequential Monte Carlo (SMC) methods, or particle filters, [DDFG01, DMDJ06] are another class of methods with successful applications for data assimilation and Bayesian filtering [LC98, Cho02, vL10, DGA00, DMDJ12]. In SMC, weighted samples, or particles, are generated and evolved to approximate a sequence of probability measures which interpolate from the prior to the posterior. The weighted samples are eventually used to represent an empirical posterior distribution, which under proper assumptions can be shown to asymptotically converge to the true posterior when the number of particles increases [RVH+15]. A few researchers have applied SMC method in the setting of inverse problems and demonstrated the efficacy of using SMC to solve Bayesian inverse problems [KBJ14, BJMS15]. In [KBJ14] in particular, the authors employed a novel SMC method with a dimension-independent MCMC sampler [CRSW13, Law14] as the mutation kernel to invert the initial conditions for Navier-Stokes equations, and demonstrated significant computational speed-up of the new SMC method for high-dimensional inverse problems. In [BJMS15], the authors enhanced the SMC method in [KBJ14] and provided a proof of the dimension-independent convergence property of the SMC methods for inverse problems.

Another challenging class of Bayesian inverse problems are those where parameter space is high or even infinite-dimensional, which typically happen if the parameters are a representation of a random function or field. For such problems, dimension reduction techniques are typically applied to tractably obtain a solution. For example, truncated Karhunen-Loeve
expansion (KL) \([\text{HQP01, ST06}]\) of the prior distribution can be employed to construct a low dimensional representation of the parameters. This approach exploits the low-dimensional structure of the prior distribution and assumes the posterior distribution possess the same low-dimensional structure, which might not be true in general. More recent works attempt to directly exploit the lower intrinsic dimensionality of the posterior by identifying subspaces where most of the posterior probability mass is contained \([\text{FK16}]\), or exploit the fact that posterior is usually a low-dimensional update of the prior, in the sense that change from prior to posterior is most prominent on a low-dimensional subspace of the parameter space \([\text{BTGMS13, FWA}^{+11, \text{CMM}^{+14, \text{SSC}^{+15, \text{CMW16, SCW}^{+17}}}}]\). Exploiting such low-dimensional structures allows inference tasks to be performed on much lower dimensions, which in turn enables MCMC sampler in such high dimensions \([\text{CMW16}]\).

The majority of Bayesian methods for inverse problems rely on an exact noise model, which is typically assumed to be i.i.d. Gaussian, to perform inference. It is desirable to extend such inference to more general settings where a noise model is unavailable or modeling such data generating mechanism is challenging. The gibbs posterior provides a way to update belief distributions in such general setting without the need of an explicit likelihood function. Instead, the Gibbs posterior are applicable where the unknown parameters are only connected to the data through a loss function \([\text{BHW16, ARC16, SM16}]\). In many inverse problems or inference problems, it can be a simpler task to specify a loss function instead of the true data generating mechanism, i.e., an explicit likelihood function, which is the biggest advantage of using the Gibbs posterior over the usual Bayesian approach. We adopt this general framework and propose a particle-based method with local RB surrogate to efficiently approximate Gibbs posterior in Chapter 5.

1.3 Contributions and layout of the thesis

There are three primary contributions of this thesis. They are: (1) the development a locally adapted reduced basis (local RB) method to efficiently approximate the solution to
PDEs with random inputs, (2) the development of an efficient solver for risk-averse PDE-constrained optimization problems that leverages surrogate models based on the local RB method, (3) the development of an efficient particle-based method for solving inverse problems under uncertainty using Gibbs posterior and local RB surrogates. There are three secondary contributions: (1) the introduction of a residual-based error bounds with an efficient adaptive sparse-grid based approximation for the stability constant, the method is general and can be applied in any RB context, (2) the extension of the local RB method to approximate coherent risk measures applied on the QoIs governed by stochastic PDEs, (3) the extension of the residual-based error bound to various contexts including risk approximation, adjoint equation approximation and trust region error bounds, which showcased the versatility of the residual-based error bound that could enable potential wide application of the local RB method.

In Chapter 2, we present the full details of the proposed local RB method applied to solving PDEs with uncertain inputs. The local RB method can be regarded as a novel extension of the existing local approximation methods based on Taylor expansion [Gri12] and the widely applied global RB methods for parametrized PDEs [RHP08]. Central to the local RB method is a set of adaptively constructed parameter samples which gives rise to an implicit Voronoi partition of the parameter domain. Using this decomposition, we construct local RB approximations of the PDE solution within each Voronoi cell. In contrast to existing RB methods, we form local bases for each Voronoi cell using solutions only at a fixed number of proximal atoms as well as the gradient information at the generating atom. Due to the local nature of our proposed method, the online computational cost does not increase as more atoms are added in the adaptive process, which is a sharp advantage over existing RB methods where the size of reduced problem grows with the number of atoms used to construct the basis. In addition, thanks to the gradient information and the property of Taylor expansion, the solution of our reduced problem is guaranteed to converge almost everywhere over the parameter domain to the solution of the full problem provided certain smoothness assumptions hold. We formally show the assumptions and the
error bound of our local RB method in Section 1 and 2.3.

In addition, we build our atom set on the fly by adapting the greedy approach utilized in [QRM11, RHP08] and guiding the adaptivity using residual-based \textit{a posteriori} error indicators. Our error indicators require the evaluation of the stability (inf-sup) constant of the parametrized PDE operator for each realization of the random input. To circumvent the need to solve an eigenvalue problem for the exact stability constant at each sample, we develop a surrogate model of the constant using dimension-adaptive sparse grids [GZ07, GG03]. This surrogate model for stability constant is universally applicable where residual based error indicator is used, typically in the RB context. The details of the stability surrogate along with the adaptive sampling strategy are presented in Section 2.4.

Our adaptive sampling approach permits a natural extension to approximate coherent risk measures of QoIs that depend on the PDE solution. To this end, we develop a rigorous bound for the error in the risk approximation and subsequently derive an error indicator that is compatible with our adaptive sampling framework. With the risk error indicator, we can easily perform adaptive sampling to specifically target risk approximation. We present details about risk measures and such extensions of local RB in Chapter 3.

In Chapter 4, we leverage the accuracy and efficiency of our local RB approximation to solve risk-averse optimization problems with PDE-constraint. In particular, we combine the inexact trust-region (TR) framework of [KHRvBW14] with the local RB method to efficiently approximate the solutions for risk-averse optimal control problems. We use the local RB method to systematically construct the surrogate models for the TR subproblem and objective function evaluations. To this end, we derive error bounds for the objective and gradient approximation using \textit{a posteriori} error indicators and subsequently introduce adaptive sampling schemes for the sequential construction of the local RB surrogate models. We demonstrate orders of magnitude computational speed-up of the proposed approach in several numerical examples involving different PDEs and risk measures. The extension of local RB method to such contexts and the demonstration of the supreme performance of the proposed approach is a novel contribution.
In Chapter 5, we employ the local RB method to build locally accurate surrogate model to accelerate the approximation of the Gibbs posterior for inverse problems with noisy data. As the Gibbs posterior is a generalization of the conventional Bayesian posterior where a likelihood function is replaced by a general loss function, our computational approach is equally applicable in the conventional Bayesian context. We present a convergence proof of the proposed approach under proper assumptions on the loss function. By using a SMC framework with particle based approximation, our method fully exploits the local accuracy of the local RB method. The emphasis of the local RB surrogate is navigated to a small fraction of the parameter space automatically by the evolving particles that progressively cluster over the support of the posterior. Once the surrogate is accurate enough over the local support of the posterior, further evolution of the particles takes negligible cost. The extension of local RB method to Bayesian inverse problem is novel and highlights the potential applications of the local RB method to problems where accurate and efficient local approximation is required.

Finally, conclusions and ideas for future research are given in Chapter 6.
Chapter 2

The Local Reduced Basis Method for PDEs with Random Inputs

In this chapter, we present the local RB method to solve PDEs with uncertain parameters. We first describe the problem formulation including a discussion of PDEs with uncertain inputs and local Taylor-like approximations. Following this discussion, we introduce our adaptive local RB approach including \textit{a priori} and \textit{a posteriori} error analysis. We present an analysis of the computational cost of the local RB method for affinely parametrized PDEs. We show numerical examples that confirm our results.

2.1 Formulations

Let \((\Omega, \mathcal{F}, \mathbb{P})\) be a probability space, where \(\Omega\) is the sample space, \(\mathcal{F} \subseteq 2^\Omega\) is a \(\sigma\)-algebra of events, and \(\mathbb{P} : \mathcal{F} \to [0, 1]\) is a probability measure. We denote the expectation of a random variable \(X : \Omega \to \mathbb{R}\) by \(E[X] := \int_\Omega X(\omega) \mathrm{d}\mathbb{P}(\omega)\) and by \textit{almost surely} (a.s.) we mean “up to a set of \(\mathbb{P}\)-measure equal to zero.” We denote the space of random variables on \((\Omega, \mathcal{F}, \mathbb{P})\) with \(p\) finite moments, \(p \in [1, \infty)\), by \(L^p(\Omega, \mathcal{F}, \mathbb{P})\) and the space of essentially bounded random variables on \((\Omega, \mathcal{F}, \mathbb{P})\) by \(L^\infty(\Omega, \mathcal{F}, \mathbb{P})\). We denote the extension of \(L^p(\Omega, \mathcal{F}, \mathbb{P})\) for \(p \in [1, \infty]\) to vector-valued random variables by \(L^p(\Omega, \mathcal{F}, P; A)\) where \(A\) is a Banach space. Additionally, for any two Banach spaces \(A\) and \(B\), we denote the space of bounded linear operators that map \(A\) into \(B\) by \(L(A, B)\). We further denote the dual space of \(A\) by \(A^* = L(A, \mathbb{R})\) (i.e., \(A^*\) is the Banach space consisting of all bounded linear functionals acting on elements of \(A\)), and the associated dual pairing by \(\langle a^*, a \rangle_{A^*, A} = a^*(a)\) for \(a^* \in A\) and \(a \in A\). Finally, if \(A\) is a Hilbert space, we denote the inner product on \(A\) by \(\langle a', a \rangle_A\) for all \(a', a \in A\).

Let \(D \subset \mathbb{R}^d\), \(d = 1, 2, 3\), denote the physical domain and let \(U = U(D)\) and \(V = V(D)\) denote Hilbert spaces of (deterministic) functions mapping \(D\) into \(\mathbb{R}\). We consider the
following parametric equation: For fixed \( f \in V^* \), find \( u : \Omega \to U \) such that

\[
\hat{L}(\omega)u(\omega) = f, \quad \text{a.s.}
\] (2.1)

Here \( \hat{L}(\omega) \) is a bounded linear operator from \( U \) into \( V^* \) for \( \mathbb{P} \)-almost all (a.a.) \( \omega \in \Omega \). To facilitate numerical approximation, we assume that the finite-dimensional noise assumption holds [XH05]. That is, we assume there exists a random vector, \( \xi : \Omega \to \Xi \) with \( \Xi := \xi(\Omega) \subseteq \mathbb{R}^M \), and a bounded linear operator \( \mathcal{L}(\xi) \) mapping \( U \) into \( V^* \) a.s. satisfying

\[
\hat{L}(\omega) = \mathcal{L}(\xi(\omega)) \quad \text{a.s.}
\]

We denote the joint distribution of \( \xi \) by \( F_\xi \) and assume that the probability law of \( \xi, \mathbb{P} \circ \xi^{-1} \), is absolutely continuous with respect to the Lebesgue measure with density \( \rho : \Xi \to [0, \infty) \). Here and throughout, we abuse notation and denote the random vector and its realizations by \( \xi \). We note here that uncertainty can also be incorporated in the right-hand side \( f \), but for ease of presentation, we restrict our attention to uncertainty in the operator. Through a change of variables, (2.1) becomes: Find \( u : \Xi \to U \) such that

\[
\mathcal{L}(\xi)u(\xi) = f, \quad \forall \xi \in \Xi.
\] (2.2)

In the subsequent analysis, it will be convenient to define the parametrized bilinear form \( a(\cdot, \cdot; \xi) : U \times V \to \mathbb{R} \) associated with \( \mathcal{L}(\xi) \) for \( \xi \in \Xi \) by

\[
a(u, v; \xi) := \langle \mathcal{L}(\xi)u, v \rangle_{V^*, V} \quad \forall u \in U, \ v \in V.
\]

We can then rewrite (2.2) as the variational problem: Find \( u : \Xi \to U \) such that

\[
a(u(\xi), v; \xi) = \langle f, v \rangle_{V^*, V} \quad \forall v \in V, \ \xi \in \Xi.
\] (2.3)

**Assumption 1.** We assume that the spaces \( U \) and \( V \) are chosen so that problem (2.3) is well-posed. In particular, we assume that there exists positive constants \( m_0, m_1, m_2 > 0 \),
independent of $\xi \in \Xi$, such that for all $\xi \in \Xi$,

$$\langle \mathcal{L}(\xi)u, v \rangle_{V^*, V} \leq m_0\|u\|_U\|v\|_V \quad \forall u \in U, \ v \in V \quad (2.4a)$$

$$\inf_{u \in U \setminus \{0\}} \sup_{v \in V \setminus \{0\}} \frac{\langle \mathcal{L}(\xi)u, v \rangle_{V^*, V}}{\|u\|_U\|v\|_V} =: \gamma_1(\xi) \geq m_1 \quad (2.4b)$$

$$\inf_{v \in V \setminus \{0\}} \sup_{u \in U \setminus \{0\}} \frac{\langle \mathcal{L}(\xi)^*v, u \rangle_{U^*, U}}{\|v\|_V\|u\|_U} =: \gamma_2(\xi) \geq m_2 \quad (2.4c)$$

**Assumption 2.** We assume the bilinear form $a(\cdot, \cdot; \xi)$ allows an affine decomposition with respective to $\xi$, that is,

$$a(u, v; \xi) = \sum_{i=1}^{M} \phi_i(\xi_i) a_i(u, v) \quad (2.5)$$

where $\phi_i : \mathbb{R} \to \mathbb{R}$ and $a_i(\cdot, \cdot) : U \times V \to \mathbb{R}$ are parameter dependent function and parameter independent bilinear forms, respectively.

**Remark 1.** This assumption is typical in RB (and general projection-based model reduction) methods, it allows for an offline-online computing strategy that is critical to control the computational cost of the RB approximation [RHP08]. A wide range of problems satisfy such an assumption, and many nonaffine problems can be approximated by affine counterparts using methods like empirical interpolation [BMNP04, CS10, DHO12]. Therefore, throughout the thesis, we will assume our PDEs allow such an affine parametrization. We provide a detailed computational cost analysis of our method for affine problems in Section 2.5.

Under Assumption 1, the Banach-Nečas-Babuška Theorem [EG13] ensures that a unique solution to (2.3) exists for all $\xi \in \Xi$, and that the solution is bounded independent of $\xi \in \Xi$. Note that when $U$ and $V$ are finite-dimensional, (2.3) represents, e.g., the discretization of a PDE. Our goal in this chapter is to build an approximation $\overline{u} : \Xi \to U$ of $u$ that can be evaluated at $\xi \in \Xi$ efficiently without the need to solve the full problem (2.3). Once we construct $\overline{u}$, we can approximate the statistics of $u$ using the statistics of $\overline{u}$. For example,
given a function $g : U \to \mathbb{R}$, the expectation of $g(u(\xi))$ can be approximated using Monte Carlo as

$$
E[g(u)] \approx E[g(\mathbf{p})] = \int_{\Xi} g(\mathbf{p}(\xi)) \rho(\xi) d\xi \approx \frac{1}{N} \sum_{i=1}^{N} g(\mathbf{p}(\xi_i)) \quad (2.6)
$$

where $\xi_i$ are independent and identically distributed random vectors.

### 2.2 Deterministic samples and local approximation

Our proposed surrogate, $\mathbf{p}$, relies on a set of deterministic samples distributed in $\Xi$. These samples generate an implicit Voronoi partition of the parameter domain on which we construct local RB approximations. This method is inspired by local Taylor approximation on a finite set of (deterministic) samples of $\xi$. In particular, we draw motivation from SROMs [Gri12, Gri14]. We will review some critical facts about Taylor approximation that motivate our approximation method.

Any finite set of (deterministic) samples of $\xi$, denoted $\{\xi_k\}_{k=1}^{m}$ (e.g., computed as a SROM of $\xi$), generates an implicit Voronoi partition of the parameter domain $\Xi = \bigcup_{k=1}^{m} \Xi_k$ where $\{\Xi_k\}$ are the Voronoi cells generated by the atoms $\{\xi_k\}$, i.e.,

$$
\Xi_k := \{\xi \in \Xi : \|\xi - \xi_k\|_2 \leq \|\xi - \xi_j\|_2, \quad j \neq k\}.
$$

Here, $\| \cdot \|_2$ denotes the Euclidean norm on $\mathbb{R}^m$. Using this partition, we can assign each sample $\xi_k$ the probability $p_k := \mathbb{P}(\omega \in \Omega : \xi(\omega) \in \Xi_k)$ for $k = 1, \ldots, m$. Clearly, we have that $\sum_{k=1}^{m} p_k = 1$ since $\Xi = \bigcup \Xi_k$ and $\mathbb{P} \circ \xi^{-1}(\Xi_i \cap \Xi_j) = 0$ for $i \neq j$, which follows from the absolute continuity of $\mathbb{P} \circ \xi^{-1}$ with respect to the Lebesgue measure. Given the samples $\{\xi_k\}$ and assuming sufficient regularity of the random field PDE solution $u$, we can approximate $u$ using a Taylor expansion over $\Xi_k$ anchored at the sample $\xi_k$ in the usual way. For example, the local linear approximation of $u(\xi)$ is

$$
\mathbf{p}_0(\xi) := \sum_{k=1}^{m} \mathbf{1}_{\Xi_k}(\xi) \left[ u(\xi_k) + \nabla_u u(\xi_k) \cdot (\xi - \xi_k) \right]. \quad (2.7)
$$
Here, $1_{\Xi_k}(\xi)$ denotes the characteristic function of the set $\Xi_k$. That is, $1_{\Xi_k}(\xi)$ is one if $\xi \in \Xi_k$ and is zero otherwise. $u(\xi_k)$ is the solution to Equation (2.3) at $\xi_k$, that is
\[
a(u(\xi_k), v; \xi_k) = \langle f, v \rangle_{V^*, V} \quad \forall v \in V,
\]
and
\[
\nabla_{\xi} u(\xi_k) = [\partial_1 u(\xi_k), \partial_2 u(\xi_k), \ldots, \partial_M u(\xi_k)]^T,
\]
where $\partial_i u(\xi_k) := \frac{\partial u(\xi_k)}{\partial \xi_i}$ is the solution to the following problem,
\[
a(\partial_i u(\xi_k), v; \xi_k) = -\partial \phi_i(\xi_k, u(\xi_k), v) \quad \forall v \in V. \tag{2.9}
\]
Here, $\xi_{k,i}$ denote the $i$th components of $\xi_k$.

Remark 2. It is easy to see that a total number of $m(1 + M)$ forward solves are needed to construct the surrogate $\pi_0(\xi)$. However, the only difference between Equation (2.8) and (2.9) is the right-hand-side forcing. The two problems share the same operator. Existing methods can be applied to solve such multiple right-hand-side problems very efficiently [SPM89, SG95]. If a direct solver is used to solve the resulting matrix equation, only one factorization is needed. This reduces the computational costs significantly. Furthermore, Equation (2.9) is automatically well-posed as long as (2.8) is well-posed.

The following theorem provides an error bound for the above Taylor approximation (2.7). It is adapted from [Gri12, Theorem 2] by Drew Kouri.

Theorem 1. Let $\Xi \supseteq \Xi$ be an open set in $\mathbb{R}^M$ and suppose $u : \Xi \to U$ is continuously Fréchet differentiable. Moreover, we assume that there exists $L \geq 0$ such that
\[
\max_{i=1,\ldots,M} \|\partial_i u(\xi) - \partial_i u(\xi')\|_U \leq L\|\xi - \xi'\| \quad \forall \xi, \xi' \in \Xi, \tag{2.10}
\]
then,
\[
\|u(\xi) - \pi_0(\xi)\|_U \leq \frac{L}{2} \sum_{k=1}^{m} 1_{\Xi_k}(\xi) \|\xi - \xi_k\|_2^2 \quad \forall \xi \in \Xi. \tag{2.11}
\]
In particular, if $\Xi$ is bounded then

$$\sup_{\xi \in \Xi} \| u(\xi) - \overline{u}_0(\xi) \|_U \leq \frac{L}{2} \max_{k=1,\ldots,m} \text{diam}(\Xi_k)^2$$

(2.12)

where the diameter of $\Xi_k$ is defined by

$$\text{diam}(\Xi_k) := \sup_{\xi,\xi' \in \Xi_k} \| \xi - \xi' \|_2$$

Proof. Notice that subtracting $\overline{u}_0(\xi)$ from $u(\xi)$ and applying the integral mean value theorem gives

$$u(\xi) - \overline{u}_0(\xi) = \sum_{k=1}^{m} \mathbb{1}_{\Xi_k}(\xi) \int_0^1 (\nabla_\xi u(\overline{x}_k + t(\xi - \overline{x}_k)) - \nabla_\xi u(\overline{x}_k)) \cdot (\xi - \overline{x}_k) dt.$$ 

Computing the norm of both sides and applying Jensen’s and the Cauchy-Schwarz inequalities as well as the Lipschitz continuity bound (2.10) yields (2.11). When $\Xi$ is bounded, (2.12) follows trivially from (2.11) and Hölder’s inequality.

As can be seen, the Taylor approximation is a special discontinuous collocation method. The collocation points, $\{\overline{x}_k\}_{k=1}^m$, are selected according to the probability law of the input random parameters $\xi$. Within some vicinity of each $\overline{x}_k$, the truncated Taylor expansion is accurate. In the context of SROMs, the authors in [FGE15] demonstrate the efficacy of (2.7) to solve stochastic problems and its specific advantages over the traditional stochastic collocation and stochastic Galerkin methods.

Local approximation based on Taylor expansion is only accurate when the size of the cells $\Xi_k$ is sufficiently small, as can be seen in (2.11). Moreover, for problems in which the gradient of $u$ varies rapidly with the random parameters, the truncated Taylor expansion is likely to yield a large error that can only be mitigated by refining the local partition with additional atoms. The number of samples required to generate a sufficiently refined partition can be quite large, which leads to excessive computational cost. On the other hand, SROM samples $\{\overline{x}_k\}$ are constructed to approximate the input random parameters and do not necessarily yield a good approximation of the PDE solution.
In the subsequent sections, we will develop a method that improves upon the local Taylor approach. In particular, we will replace the truncated Taylor expansion with a RB approximation and employ adaptivity to select the samples, and hence to partition the parameter domain. The sample locations are chosen using an \textit{a posteriori} error indicator. For certain problems, these changes yield significant improvement in terms of the approximation quality of the PDE solution.

2.3 The local reduced basis approximation

Our approximation approach is motivated by the observation that the local first-order Taylor approximation (2.7) can be rewritten as

\[ u_0(\xi) = \sum_{k=1}^{m} 1_{\Xi_k}(\xi) \Phi_{k,0} w_k(\xi) \]  

(2.13)

where

\[ \Phi_{k,0} = \left[ u(\bar{\xi}_k), \partial_1 u(\bar{\xi}_k), \partial_2 u(\bar{\xi}_k), \ldots, \partial_M u(\bar{\xi}_k) \right] \]  

(2.14)

\[ w_k(\xi) = \left[ 1, (\xi_1 - \bar{\xi}_{k,1}), (\xi_2 - \bar{\xi}_{k,2}), \ldots, (\xi_M - \bar{\xi}_{k,M}) \right]^T. \]  

(2.15)

Here, \( \xi_i \) and \( \bar{\xi}_{k,i} \) denote the \( i \)th components of \( \xi \) and \( \bar{\xi}_k \), respectively. Thus, for fixed \( \xi \in \Xi_k \), the surrogate in (2.7) lies in the linear span of \( \Phi_{k,0} \) with coefficients \( w_k(\xi) \). As discussed above, this approximation is only accurate when \( \Xi_k \) is “sufficiently small” and is not optimal in terms of fully exploiting the range of \( \Phi_{k,0} \). We address this deficiency using Galerkin projection as a means of computing the coefficient vector \( w_k(\xi) \).

To provide more flexibility and to make the proposed method more robust, we provide the option to extend the range of \( \Phi_{k,0} \) with additional sampled solutions at proximal points. In particular, we append to \( \Phi_{k,0} \) the solutions \( u(\bar{\xi}_j) \) at the closest \( N \) (\( N \leq m - 1 \)) atoms. That is,

\[ \Phi_k = \left[ \Phi_{k,0}, u(\bar{\xi}_{k_1}), u(\bar{\xi}_{k_2}), \ldots, u(\bar{\xi}_{k_N}) \right]. \]  

(2.16)
An example of the proposed basis is shown in Figure 2.1, where a partition of the parameter domain $\Xi$ (with two random parameters) is generated by a group of atoms (red circles and dots). The surrogate solution at the blue dot is computed using a basis consisting of the solution at the red dots as well as the solution and gradient at the black dot.

![Figure 2.1](image)

**Figure 2.1:** Example of the local reduced basis method with two random parameters and $N = 3$. The surrogate solution at the blue dot is computed using a basis consisting of the solution at the red dots as well as the solution and gradient at the black dot. We plot 2,000 Monte Carlo samples of $\xi$ in the background.

Using $\Phi_k$, we define our surrogate, $\ddot{u}$, as

$$
\ddot{u}(\xi) := \sum_{k=1}^{m} 1_{\Xi_k}(\xi)u_k(\xi)
$$

(2.17)

where $u_k(\xi)$ solves the following variational problem: Find $u_k : \Xi_k \to U_k$ such that

$$
a(u_k, v_k; \xi) = \langle f, v_k \rangle_{V^*, V} \quad \forall v_k \in V_k(\xi), \; \xi \in \Xi_k.
$$

(2.18)

Here, $U_k := \text{span}(\Phi_k)$ is the reduced approximation space and $V_k(\xi) \subset V$ is the reduced test space that is possibly dependent on $\xi$.

In general, (2.18) is not automatically well-posed for an arbitrary choice of $V_k(\xi)$. To guarantee the well-posedness of (2.18), we extend the concept of optimal testing. Similar ideas have also been used in [DG14] in the context of discontinuous Petrov-Galerkin
method and in [QRM11, RHP08] for RB applications. To this end, we first introduce the operator \( T(\xi) := R_V^{-1} \circ L(\xi) \) where \( R_V^{-1} : V^* \to V \) is the inverse Riesz isomorphism. It is straightforward to verify that

\[
(T(\xi)u, v)_V = a(u, v; \xi) \quad \forall u \in U, v \in V
\]  

(2.19)

we then define the test space as \( V_k(\xi) := T(\xi)U_k \). Due to the properties of \( a(\cdot, \cdot; \xi) \) (2.4b), \( T(\xi) \) is injective. Obviously, \( V_k(\xi) \subset V \) since \( U_k \subset U \). With this test space, the following theorem demonstrates the well-posedness of (2.18).

**Theorem 2.** If \( V_k(\xi) = T(\xi)U_k \) and (2.4b) holds. Then, we have that

\[
\inf_{0 \neq u \in U_k} \sup_{0 \neq v \in V_k(\xi)} \frac{|a(u, v; \xi)|}{\|u\|_U \|v\|_V} \geq \inf_{0 \neq u \in U_k} \frac{|a(u, T(\xi)u; \xi)|}{\|u\|_U \|T(\xi)u\|_V} = \inf_{0 \neq u \in U_k} \frac{|(T(\xi)u, T(\xi)u)_V|}{\|T(\xi)u\|_V} = \inf_{0 \neq u \in U_k} \frac{|a(u, u; \xi)|}{\|u\|_U \|u\|_V} = \inf_{0 \neq u \in U} \frac{|a(u, u; \xi)|}{\|u\|_U \|u\|_V} = \gamma(\xi) > m_1 \quad \forall \xi \in \Xi. \tag{2.20}
\]

**Proof.** We have that

\[
\inf_{0 \neq u \in U_k} \sup_{0 \neq v \in V_k(\xi)} \frac{|a(u, v; \xi)|}{\|u\|_U \|v\|_V} \geq \inf_{0 \neq u \in U_k} \frac{|a(u, T(\xi)u; \xi)|}{\|u\|_U \|T(\xi)u\|_V} = \inf_{0 \neq u \in U_k} \frac{|(T(\xi)u, T(\xi)u)_V|}{\|T(\xi)u\|_V} = \inf_{0 \neq u \in U_k} \frac{|a(u, u; \xi)|}{\|u\|_U \|u\|_V} = \gamma(\xi) > m_1
\]

where the first and third equalities follow from the definition of \( T(\xi) \) and the second equality follows from

\[
\frac{|(T(\xi)u, T(\xi)u)_V|}{\|T(\xi)u\|_V} = \frac{\|T(\xi)u\|_V^2}{\|T(\xi)u\|_V} = \|T(\xi)u\|_V = \|L(\xi)u\|_{V^*} \leq \sup_{0 \neq v \in V} \frac{|(T(\xi)u, v)_V|}{\|v\|_V}.
\]

\[\square\]

We have the following a priori error bound for our surrogate \( \pi(\xi) \).

**Theorem 3.** Let the assumptions of Theorem 1 hold. Then,

\[
\|\pi(\xi) - u(\xi)\|_U \leq \left( 1 + \frac{m_2}{m_1} \right) \left( \frac{L}{2} \sum_{k=1}^{m} \|\xi_k\|_\Xi \|\xi - \xi_k\|_2^2 \right) \quad \forall \xi \in \Xi. \tag{2.21}
\]
Proof. We have that

\[ \|\pi(\xi) - u(\xi)\|_U \leq \left(1 + \frac{m_2}{m_1}\right) \sum_{k=1}^{m} \inf_{w \in U_k} \|w - u(\xi)\|_U \]

\[ \leq \left(1 + \frac{m_2}{m_1}\right) \sum_{k=1}^{m} \|\pi_0(\xi) - u(\xi)\|_U \]

\[ \leq \left(1 + \frac{m_2}{m_1}\right) \frac{L}{2} \sum_{k=1}^{m} \|\Xi_k(\xi)\|_2 \|\xi - \xi_k\|_2 \]

for all \( \xi \in \Xi \) where the first inequality follows from Theorem 2.2 in [Bab71], the second inequality is due to the fact that \( \pi_0(\xi) \in U_k \) and the final is due to Theorem 1. \( \square \)

Note that \( V_k(\xi) \) is dependent on \( \xi \), which is essential to establish the stability of the reduced problem (2.18) for arbitrary \( \xi \in \Xi \). However, for uniformly coercive elliptic problems, we can set \( V_k = U_k \). Due to the coercivity of the bilinear form \( a(\cdot, \cdot; \xi) \), the restriction argument implies that the problem in (2.18) is automatically well posed. A similar error bound can be derived as in Theorem 3.

Finally, note that since \( N \) is a fixed number that does not grow with the introduction of more atoms, the extended basis \( \Phi_k \) is still local and the Galerkin projection (2.18) has at most \( M + N + 1 \) dimensions which are independent of the number of atoms \( m \). When \( N = 0 \), we recover the Taylor basis \( \Phi_{k,0} \) that is used by the SROM method. On the other hand, when \( N = m - 1 \), we recover a global reduced basis method similar to that used in [RHP08], but with additional local gradient enrichment. We define \( N_k = \text{dim}(\Phi_k) \) as the reduced problem dimension. Obviously, \( N_k \leq M + N + 1 \). We achieve a reduction in the computational cost since the dimension of the full problem (2.3) is much larger than \( N_k \). In many cases, the cost of solving the latter is negligible in comparison to the cost of solving the full problem.

Remark 3. For problems where the stochastic dimension \( M \) so high that even the reduced problem size needs to be controlled, we can apply a basis compression method like Proper Orthogonal Decomposition (POD) [Cha00] to the local basis to maintain the computational
efficiency. By preserving the majority of the variance in $\Phi_k$, we expect to maintain a similar approximation accuracy using the compressed local basis. However, one drawback of such compression is that part of range of $\Phi_k$ is lost, and we lose the error bound in Theorem 3.

2.4 A posteriori error estimation and adaptive sampling

We now describe our adaptive approach for generating the atoms set $\Theta := \{\xi_k\}_{k=1}^m$. Adaptively selecting atoms offers improved accuracy and reduced computational costs to deal with high and heterogeneous parameter dimensions. Our sequential selection of $\Theta$ is guided by the current surrogate solution $\bar{\pi}$. Given a current set of atoms, we select the next atom $\xi_{k+1}$ by identifying the region of $\Xi$ with the largest error, which we estimate using a reliable a posteriori error estimate $\epsilon_u(\xi)$.

In this work, we employ residual-based error estimation as the foundation for adaptivity. Residual-based error estimators have been used with great success in adaptive finite element methods [CEHL12] and in the adaptive construction of reduced bases [QRM11]. We first define the surrogate error as $e(\xi) = u(\xi) - \bar{\pi}(\xi)$ and note that $e$ satisfies

$$a(e(\xi), v; \xi) = r(\bar{\pi}(\xi), v; \xi) \quad \forall v \in V \quad (2.22)$$

where $r(\bar{\pi}(\xi), v; \xi) = \langle f, v \rangle_{V^*, V} - a(\bar{\pi}(\xi), v; \xi)$ is the residual associated with $\bar{\pi}(\xi)$. Note that $r(\bar{\pi}(\xi), \cdot; \xi) \in V^*$ is a bounded linear functional on $V$. Hence, from the inf-sup condition (2.4b), we have

$$\gamma(\xi) ||e(\xi)||_U \leq \sup_{v \in V} \frac{a(e(\xi), v; \xi)}{||v||_V} = ||r(\bar{\pi}(\xi), \cdot; \xi)||_{V^*}. \quad (2.23)$$

That is,

$$||e(\xi)||_U \leq \frac{||r(\bar{\pi}(\xi), \cdot; \xi)||_{V^*}}{\gamma(\xi)} = \epsilon_u^0(\xi). \quad (2.24)$$

The error estimator (2.24) requires explicit knowledge of the inf-sup constant for each $\xi \in \Xi$, which is not readily available and can be computationally expensive to obtain.
In particular, the exact inf-sup constant, $\gamma(\xi)$, for a single $\xi$ requires the solution of a generalized eigenvalue problem with size equal to that of the full problem. To circumvent this issue, we build an efficient surrogate model for $\gamma(\xi)$, denoted by $\tilde{\gamma}(\xi)$. Assuming that $\tilde{\gamma}(\xi)$ satisfies the approximation property

$$ \exists 0 < \delta < 1 \quad \text{such that} \quad \sup_{\xi \in \Xi} \frac{|\gamma(\xi) - \tilde{\gamma}(\xi)|}{\gamma(\xi)} \leq \delta, \quad (2.25) $$

we have that $(1 - \delta)\gamma(\xi) \leq \tilde{\gamma}(\xi) \leq (1 + \delta)\gamma(\xi)$ and we can bound the error as

$$ \|e(\xi)\|_U \leq \epsilon_u(\xi) := \frac{\|r(\pi(\xi), \cdot, \xi)\|_{\mathcal{V}^*}}{\theta \tilde{\gamma}(\xi)} \quad (2.26) $$

where $\theta = (1 + \delta)^{-1}$. Of course, accurately determining $\delta$ is generally a nontrivial task, because evaluating the error associated with $\tilde{\gamma}(\xi)$ requires solving additional eigenvalue problems.

Upon building the error indicator $\epsilon_u(\xi)$, we develop an adaptive algorithm that reduces the expected error $\mathbb{E}[\|e(\xi)\|_U]$ in a greedy manner. Thanks to the Voronoi partition of the parameter domain, we can decompose the error into a local error indicator in each cell as

$$ \mathbb{E}[\|e(\xi)\|_U] = \sum_{k=1}^{m} \mathbb{E}[\|e(\xi)\|_U 1_{\Xi_k}(\xi)] \leq \sum_{k=1}^{m} \mathbb{E}[\epsilon_u(\xi) 1_{\Xi_k}(\xi)] = \sum_{k=1}^{m} \eta_k \quad (2.27) $$

where $\eta_k := \mathbb{E}[\epsilon_u(\xi) 1_{\Xi_k}(\xi)]$ is the local error indicator. In a greedy method, the next atom is selected from the cell with the largest $\eta_k$.

To compute $\eta_k$, an intuitive approach would be to employ Monte Carlo integration via a set of training samples that captures the probability law of $\xi$. However, such an approach has two drawbacks. First, as the dimension of $\xi$ becomes large, a significant number of training samples is required. Second, a predefined, fixed training set may not provide adequate representation of the solution behavior in some regions of the parameter space.

To overcome these limitations, we propose an approach to generate training samples on the fly. To this end, notice that

$$ \eta_k = p_k \mathbb{E}[\epsilon_u(\xi) | \xi \in \Xi_k] \quad (2.28) $$
where \( p_k = \mathbb{P}(\omega \in \Omega : \xi(\omega) \in \Xi_k) \) is the probability of \( \Xi_k \) and \( \mathbb{E}[\epsilon_u(\xi)|\xi \in \Xi_k] \) is the conditional expectation of \( \epsilon_u(\xi) \) given that \( \xi \in \Xi_k \). Hence we can estimate \( \eta_k \) by estimating \( p_k \) and \( \mathbb{E}[\epsilon_u(\xi)|\xi \in \Xi_k] \) individually.

To estimate \( p_k \), we maintain a population (\( N_{\text{bkg}} \)) of background Monte Carlo samples \( \Theta_{\text{bkg}} \). Provided that \( N_{\text{bkg}} \) is sufficiently large, we have

\[
p_k \approx \frac{|\Theta_{\text{bkg}} \cap \Xi_k|}{N_{\text{bkg}}}
\]

where \(|A| \) denotes the number of samples in \( A \). This computation is realized by the implicit Voronoi tessellation. This step is computationally inexpensive since no PDE or reduced problem needs to be solved.

The conditional expectation \( \mathbb{E}[\epsilon_u(\xi)|\xi \in \Xi_k] \) can be estimated in a similar fashion. Let \( \Theta_{\text{tr}} \subset \Xi \) be a dynamic training set that adapts to our need to explore the parameter domain with the largest local error. We can estimate \( \mathbb{E}[\epsilon_u(\xi)|\xi \in \Xi_k] \) as

\[
\mathbb{E}[\epsilon_u(\xi)|\xi \in \Xi_k] \approx \frac{\sum_{\xi \in \Theta_{\text{tr}} \cap \Xi_k} \epsilon_u(\xi)}{|\Theta_{\text{tr}} \cap \Xi_k|}
\]

and we hence have

\[
\eta_k \approx \frac{|\Theta_{\text{bkg}} \cap \Xi_k|}{N_{\text{bkg}}} \times \frac{\sum_{\xi \in \Theta_{\text{tr}} \cap \Xi_k} \epsilon_u(\xi)}{|\Theta_{\text{tr}} \cap \Xi_k|}
\]

which is the computable local error indicator for each cell \( \Xi_k \).

Upon computing \( \eta_k \), the next atom \( \xi_{m+1} \) is selected from \( \Theta_{\text{tr}} \cap \Xi_k \) as the one that maximizes \( \epsilon_u(\xi) \) over this set, i.e., \( \xi_{m+1} = \arg \max_{\xi \in \Theta_{\text{tr}} \cap \Xi_k} \epsilon_u(\xi) \) and \( \Theta \) is updated to \( \Theta \cup \{\xi_{m+1}\} \). A new partition is then implicitly formed using the updated \( \Theta \).

The next step is to update the training set \( \Theta_{\text{tr}} \) so that there are enough training samples for further exploration. In particular, we require at least \( N^{\text{min}}_{\text{tr}} \) training samples inside each Voronoi-cell of the new partition. To this end, we verify the cell identity of the training samples by implicit Voronoi tessellation again. We identify cells with insufficient number of training samples and draw additional training samples in those cells using rejection.
sampling to ensure that \( \min_{k=1,\ldots,m+1} |\Xi_{tr} \cap \Xi_k| \geq N_{tr}^{\min} \). To initialize the algorithm, we start with a small \( (N_{tr}^{\text{init}}) \) of training samples in \( \Theta_{tr} \) and only one atom \( \Theta = \{ \mathbb{E}[\xi] \} \).

**Algorithm 1: Adaptive sequential construction of \( \Theta \) and \( \overline{u}(\xi) \)**

**Initialization:**

- Specify the maximum number of atoms \( m \) and the error tolerance \( \text{Err}_{tol} \).
- Form a background set \( \Theta_{bkg} := \{ \xi \}_{i=1}^{N_{bkg}} \) consisting of a sufficiently large number of Monte Carlo samples of \( \xi \).
- Form an initial training set \( \Theta_{tr} := \{ \xi \}_{i=1}^{N_{tr}^{\text{init}}} \) consisting of a few random samples of \( \xi \).
- Select an initial atom \( \Theta = \{ \mathbb{E}[\xi] \} \) and build the initial surrogate model \( \overline{u}(\xi) \) based on the solution \( u(\mathbb{E}[\xi]) \) and gradient \( \nabla_{\xi} u(\mathbb{E}[\xi]) \).
- Set \( N_{atom} = 1 \) and \( \text{Err}_{\max} = +\infty \).

**while** \( N_{atom} < m \) \( \text{and} \) \( \text{Err}_{\max} > \text{Err}_{tol} \) **do**

- Evaluate the error indicator \( \epsilon_u(\xi) \) of \( \overline{u}(\xi) \) at each \( \xi \in \Theta_{tr} \);
- Compute \( \eta_k, k = 1, 2, \ldots, N_{atom} \) via implicit Voronoi tessellation of \( \Xi \);
- Set \( \bar{k} = \arg \max_{k=1,\ldots,N_{atom}} \eta_k \), \( \text{Err}_{\max} = \max_{k=1,2,\ldots,N_{atom}} \eta_k \);
- Set \( \xi_{\max} = \arg \max_{\xi \in \Theta_{tr} \cap \Xi_k} \epsilon_u(\xi) \), \( \Theta = \Theta \cup \{ \xi_{\max} \} \) and \( N_{atom} = N_{atom} + 1 \);
- Compute \( u(\xi_{\max}) \) and gradient \( \nabla_{\xi} u(\xi_{\max}) \);
- Incorporate the new information at \( \{ \xi_{\max} \} \) into the surrogate \( \overline{u}(\xi) \);
- Draw additional training samples to ensure \( \min_{k=1,\ldots,N_{atom}} |\Xi_{tr} \cap \Xi_k| \geq N_{tr}^{\min} \).

**end**

Store the atoms \( \Theta \) and surrogate model \( \overline{u}(\xi) \) for future computations.
2.5 Computational details and cost analysis

We now present some relevant computational details to efficiently implement the local RB method as well as a cost analysis of the method.

2.5.1 The offline-online computing strategy

The offline-online computing strategy is the key to achieve computational efficiency for RB methods and projection based model reduction methods in general. In such a strategy, we exploit the affine decomposition in Assumption 2 and use storage to trade for improved computational efficiency. In particular, expensive computations such as constructing the basis and the affine components of the reduced Jacobian and residual are carried out in an offline stage. Such information are precomputed and stored in memory. Doing so, the online evaluation using the reduced basis at each new parameter only incurs a negligible cost independent of the dimension of the fully discretized PDE.

In the Petrov-Galerkin method in Theorem 2, $V_k(\xi)$ depends on $\xi$. To avoid computing the test basis at each parameter $\xi$, the affine decomposition of $a(\cdot;\cdot;\xi)$ is exploited. For each basis function $\Phi_{k,i} \in \Phi_k$, the corresponding test function $\Phi_{k,i}^{\text{test}}$ is defined as $\Phi_{k,i}^{\text{test}} = T(\xi)\Phi_{k,i}$. Thanks to the affine decomposition of $a(\cdot;\cdot;\xi)$ (Assumption 2), we have that

$$T(\xi) = \sum_{i=1}^{M} \phi_i(\xi)T_i,$$

where

$$(T_i u, v)_V = a_i(u, v) \quad \forall u \in U, v \in V$$

hence $\Phi_{k,i}^{\text{test}} = \sum_{j=1}^{d} \phi_j(\xi_j)T_j\Phi_{k,i}$ and $T_j\Phi_{k,i}$ can be precomputed by solving the following problem,

$$(T_j\Phi_{k,i}, v)_V = a_j(\Phi_{k,i}, v) \quad \forall v \in V$$

In addition, Equation (2.18) yields a low dimensional matrix equation as follows,

$$A(\xi)w(\xi) = g$$

(2.32)

where $A(\xi)_{ij} = a(\Phi_{k,i}^{\text{test}}, \Phi_{k,j}; \xi)$, $g_i = (f, \Phi_{k,i}^{\text{test}})_V$, and $\Phi_{k}w(\xi)$ is the surrogate solution. Again, by exploiting the affine parametrization $a(\Phi_{k,i}^{\text{test}}, \Phi_{k,j}; \xi) = \sum_{p=1}^{M} \phi_p(\xi_p)\alpha_p(\Phi_{k,i}^{\text{test}}, \Phi_{k,j})$, 

31
hence $a_p(\Phi_{k,i}, \Phi_{k,j})$ can be precomputed and stored in an offline stage, while the online assembly of $A(\xi)$ induces a cost that is independent of the dimension of the full PDE.

Now, we turn to the online evaluation of the error indicator $\epsilon_u(\xi)$ in (2.26). As the indicator needs to be evaluated repetitively for adaptive sampling, it is crucial that we can compute it in an efficient manner. To this end, we first address the evaluation of $\|r(\pi, \cdot; \xi)\|_{V^*}$, i.e., the dual-norm of the residual. Once again, the affine parametrization of $a(\cdot, \cdot; \xi)$ is exploited.

We assume $\pi(\xi) = \Phi_k \psi(\xi)$ is the surrogate solution at a sample $\xi \in \Xi_k$. Using the affine decomposition of $a(\cdot, \cdot; \xi)$, we have

$$r(\pi(\xi), v; \xi) = \langle f, v \rangle_{V^*, V} - \sum_{i=1}^{M} \phi_i(\xi) a_i(\pi(\xi), v)$$

$$= \langle f, v \rangle_{V^*, V} - \sum_{i=1}^{M} \sum_{j=1}^{N_k} \phi_i(\xi) w_j(\xi) a_i(\Phi_{k,j}, v). \quad (2.33)$$

By Riesz representation, we have that $r(\pi(\xi), v; \xi) = (\tilde{e}, v)_V$ for some $\tilde{e} \in V$ and that $\|r(\pi, \cdot; \xi)\|_{V^*} = \|\tilde{e}\|_V$, hence

$$(\tilde{e}, v)_V = \langle f, v \rangle_{V^*, V} - \sum_{i=1}^{M} \sum_{j=1}^{N_k} \phi_i(\xi) w_j(\xi) a_i(\Phi_{k,j}, v)$$

$$= (L, v)_V - \sum_{i=1}^{M} \sum_{j=1}^{N_k} \phi_i(\xi) w_j(\xi) (L_{k,j}, v)_V$$

where $L$ and $L_{k,j}$ are defined by the relations $(L, v)_V = \langle f, v \rangle_{V^*, V}$ and $(L_{k,j}, v)_V = a_i(\Phi_{k,j}, v)$ for $\forall v \in V$. Their existence is a consequence of the Riesz representation theorem. Hence,

$$\tilde{e} = L - \sum_{i=1}^{M} \sum_{j=1}^{N_k} \phi_i(\xi) w_j(\xi) L_{k,j}$$
\[\|\tilde{e}\|^2_V = (L, L)_V - \sum_{i=1}^{M} \sum_{j=1}^{N_k} \phi_i(\xi_i)w_j(\xi) [(L, L_{k,j})_V + (L_{k,j}, L)_V] \]
\[+ \sum_{i=1}^{M} \sum_{j=1}^{N_k} \sum_{p=1}^{M} \sum_{q=1}^{N_k} \phi_i(\xi_i)w_j(\xi)\phi_p(\xi_p)w_q(\xi)(L_{k,j}, L_{k,q})_V\]

(2.34)

which allows \((L, L)_V, (L, L_{k,j})_V\) and \((L_{k,j}, L_{k,q})_V\) to be precomputed in an offline stage. The online evaluation of the residual only calls for a linear combination of the inner products.

### 2.5.2 Approximating the inf-sup constant

Another part of computing the error indicator \(\epsilon_u(\xi)\) in (2.26) is an approximation of the stability constant \(\gamma(\xi)\), which we address by building a cheap surrogate model \(\tilde{\gamma}(\xi)\). To this end, we employ the dimension-adaptive sparse grid interpolation [GG03] due to their well understood convergence behavior and relatively low cost to achieve satisfactory accuracy.

Essentially, the dimension-adaptive sparse grid method attempts to find the important stochastic dimensions along which the stability constant varies most rapidly and puts more interpolation grid points along such important dimensions using a generalized sparse grid interpolation scheme. This feature fits well with the need to interpolate the stability constant over the parameter domain as the stability constant is typically smooth with respect to the parameters and usually only sensitive to a small fraction of stochastic dimensions that can be well captured by the dimension-adaptive sparse grid method. We shall give a brief explanation of the method. For more details, refer to [GG03, GZ07].

**Generalized sparse grid interpolation**

The conventional sparse grid interpolation method is typically built on the Smolyak’s algorithm [Smo63]. The construction of Smolyak and tensor product rules begins with the one-dimensional (along \(\xi\)) interpolation formula,

\[\mathcal{U}^I(\gamma) = \sum_{i=0}^{I} \tilde{\gamma}(\xi^{(i)})L_i(\xi)\]
based on the nodal set
\[ \Theta^I = (\xi^{(1)}, \xi^{(2)}, \ldots, \xi^{(I)}) \]
where \( I \) is the interpolation order and \( L_i(\xi) \) are the Lagrangian interpolation polynomial defined as follows,
\[ L_i(\xi) = \prod_{k=0, k \neq i}^{I} \frac{\xi - \xi^{(k)}}{\xi^{(i)} - \xi^{(k)}} \]

For \( M \)-dimensional interpolations, given a fixed interpolation order \( I \) along each dimension, the full tensor product interpolation formula is given as
\[ F^I_M(\gamma) = \left( U^I_1 \otimes \cdots \otimes U^I_M \right)(\gamma) = \sum_{k_1=0}^{I} \cdots \sum_{k_M=0}^{I} \gamma(\xi^{(k_1)}_1, \ldots, \xi^{(k_M)}_M)(L^1_{k_1} \otimes \cdots \otimes L^M_{k_M}) \quad (2.35) \]

based on the nodal set
\[ N^I_{F,M} = \Theta^I_1 \times \cdots \times \Theta^I_M. \]
It is clear that the interpolation requires solving the stability constant \( \gamma \) (i.e., a generalized eigenvalue problem) at \( I^M \) interpolation points, which is a tremendous cost even with a slightly large \( M \).

To introduce the Smolyak’s algorithm, we first introduce the following one-dimensional incremental interpolation and incremental nodal set, \( \Delta U^i \) and \( \Delta \Theta^i \), as given by
\[ U^0 = 0, \quad \Delta U^i = U^i - U^{i-1} \]
\[ \Theta^0 = \emptyset, \quad \Delta \Theta^i = \Theta^i \setminus \Theta^{i-1} \]

Obviously, we have that
\[ U^I(\gamma) = \sum_{i=1}^{I} \Delta U^i(\gamma) \]
\[ \Theta^I = \bigcup_{i=1}^{I} \Delta \Theta^i \]
Using the incremental interpolation, we can rewrite the multidimensional interpolation in (2.35) as

\[ F^I_M(\gamma) = \sum_{\max_k i_k \leq I} (\Delta U_1^{i_1} \otimes \Delta U_2^{i_2} \otimes \cdots \otimes \Delta U_M^{i_M})(\gamma) \]  

(2.36)

and the full node set as

\[ N^I_{F,M} = \bigcup_{\max_k i_k \leq I} \Delta \Theta_1^{i_1} \times \Delta \Theta_2^{i_2} \times \cdots \times \Delta \Theta_M^{i_M} \]

In Smolyak’s algorithm, the full tensor product interpolation in (2.36) is approximated by

\[ S^I_M(\gamma) = \sum_{\sum_k i_k \leq I + M - 1} (\Delta U_1^{i_1} \otimes \Delta U_2^{i_2} \otimes \cdots \otimes \Delta U_M^{i_M})(\gamma) \]  

(2.37)

based on a sparse node set

\[ N^I_{S,d} = \bigcup_{\sum_k i_k \leq I + M - 1} \Delta \Theta_1^{i_1} \times \Delta \Theta_2^{i_2} \times \cdots \times \Delta \Theta_M^{i_M} \]

The approximation is accurate if the function \( \gamma \) is smooth.

To generalize the above sparse grid interpolation formula, we define \( i = \{i_1, i_2, \ldots, i_M\} \) as an index and let \( I \subseteq \mathbb{N}_+^M \) be an admissible index set, where \( \mathbb{N}_+ = \{1, 2, \ldots\} \). A index set \( I \) is only admissible if for all \( i \in I \), we have that \( j = (j_1, j_2, \ldots, j_M) \in \mathbb{N}_+^M \) with \( j_k \leq i_k \) for \( \forall k = 1, 2, \ldots, M \) implies that \( j \in I \). The generalized interpolation formula can hence be written as

\[ G^I_M(\gamma) = \sum_{i \in I} (\Delta U_1^{i_1} \otimes \Delta U_2^{i_2} \otimes \cdots \otimes \Delta U_M^{i_M})(\gamma) \]  

(2.38)

with a general interpolation node set

\[ N^I_{G,M} = \bigcup_{i \in I} \Delta \Theta_1^{i_1} \times \Delta \Theta_2^{i_2} \times \cdots \times \Delta \Theta_M^{i_M} \]

With this definition, the full tensor product interpolation (2.36) can be expressed as a special case of (2.38) with \( I = \{(i_1, i_2, \ldots, i_M) : \max_k i_k \leq I\} \). In addition, the Smolyak’s algorithm (2.37) can be defined by the index set \( I = \{(i_1, i_2, \ldots, i_M) : \sum_k i_k \leq I + M - 1\} \).
Dimension adaptivity by adaptive selection of index sets

To obtain higher interpolation accuracy with minimal computational cost, an adaptive procedure can be used to sequentially select an admissible index set that places more emphasis on the important dimensions that are more influential to the target function $\gamma(\xi)$. To this end, the algorithm begins with the one element index set $\{1\}$, $1 = (1, 1, \ldots, 1)$ and adds indices successively. In the process, the following two conditions always need to be satisfied: the updated index set remains admissible and a large reduction in the interpolation error is achieved after adding each index into the index set. To achieve the second goal, an error indicator needs to be defined to quantify the error reduction.

In the first step, the algorithm adds the indices $(2, 1, \ldots, 1), \ldots, (1, 1, \ldots, 2)$ and obtains an initial set of error indicators for the added indices. To proceed, the index set $I$ is partition into two sets, the active index set and the old index set. The active index set includes all indices whose error indicator has been computed but the error indicator of their admissible forward neighbors (the forward neighbors of an index $j$ are defined as the M-indices $\{j + e_k, 1 \leq k \leq M\}$) has not been yet computed. All the other indices in the index set are in the old index set. In each step, the indices in the active index set with the largest error indicator is moved into the old index set, meanwhile all its admissible forward neighbors are added into $I$ and put into the active index set.

The error indicator of an index $j$ is chosen to measure the deviation of the true value of $\gamma$ at index $j$ from its current interpolated value $\hat{\gamma}$ (using the interpolation built with all the old indices). This deviation is nothing but the incremental interpolation at $j$: $\Delta_j = (\Delta U_1^{j_1} \otimes \cdots \otimes \Delta U_M^{j_M})(\gamma)$. The error indicator function $\text{Err}_{\gamma}(\Delta_j)$ simply measures the average absolute value of $\Delta_j$ at the set of new nodes associated with $j$, i.e., $\Delta \Theta_1^{j_1} \times \cdots \times \Delta \Theta_M^{j_M}$.

In this way, the algorithm adaptively add the important indices whose error is maximal into the admissible index set. For a more detailed explanation of the procedure, see [GG03]. Our implementation of this dimension adaptive sparse grid approach is based on the sparse grid interpolation toolbox developed in [Kli07]. A complete process is given in Algorithm 2.
Algorithm 2: Dimension adaptive sparse grid to construct \( \hat{\gamma}(\xi) \)

**Initialization:**
- Specify the maximum number of interpolation point \( N_{\gamma} \) and the error tolerance \( \text{Err}_{tol} \).
- Set \( i = (1, \ldots, 1), \mathcal{O} = \emptyset, \mathcal{A} = \{i\} \).
- Set \( G_M(\gamma) = (\Delta U_1^1 \otimes \cdots \otimes \Delta U_M^1)(\gamma), \eta_i = \eta = \text{Err}_{\gamma}(G_M) \).
- Set \( N_{G,M} = \Delta \Theta_1^1 \times \cdots \times \Delta \Theta_M^1, n = |N_{G,M}| \).

while \( \eta > \text{Err}_{tol} \) or \( n < N_{\gamma} \) do
  Select \( i \in \mathcal{A} \) with the largest \( \eta_i \), set \( \mathcal{A} = \mathcal{A} \setminus \{i\}, \mathcal{O} = \mathcal{O} \cup \{i\}, \eta = \eta_i \)
  for \( k = 1, \ldots, M \)
    Set \( j = i + e_k \)
    if \( \mathcal{O} \cup \{j\} \) is admissible
      Set \( u_j = (\Delta U_1^j \otimes \cdots \otimes \Delta U_M^j)(\gamma), \theta_j = \Delta \Theta_1^j \times \cdots \times \Delta \Theta_M^j \)
      Set \( \mathcal{A} = \mathcal{A} \cup \{j\}, \eta_j = \text{Err}_{\gamma}(\Delta_j), n = n + |\theta_j| \)
      Update \( G_M(\gamma) = G_M(\gamma) + \Delta_j, N_{G,M} = N_{G,M} \cup \theta_j \)
    end if
  end for
end

Finally, set \( \hat{\gamma}(\xi) = G_M(\gamma)(\xi) \).

### 2.5.3 Computational cost

Now we are ready to present a coarse estimate of the computational complexity associated with each stage of our method. To this end, we assume that \( U \) and \( V \) are finite dimensional with dimension \( N_{\text{space}} \) and that \( M + N \ll N_{\text{space}} \). In our estimate, we omit constants that are neither an algorithmic parameter nor a dimension of the discretization. When adding estimates, we omit terms that are clearly comparable to or dominated by another term.
In practice, the indicator $\epsilon_u(\xi)$ is a conservative upper bound of the true error $\|e(\xi)\|_U$ and only serves as a guide for the adaptive algorithm to select the next atom. Therefore, a somewhat loose approximation $\epsilon_u(\xi)$ may be sufficient to construct an accurate surrogate $\bar{u}$. To this end, we employ dimension-adaptive sparse grid interpolation [GZ07, GG03, MZ10] as shown in Section 2.5.2. Of course, error bounds of the form (2.25) are not readily available for adaptive sparse grid interpolation and thus our error indicator (2.26) is heuristic. We limit the number of collocation points to be $N_\gamma$. Given that we only need moderate accuracy of the surrogate model, $N_\gamma$ can be chosen to be a relatively small number. For instance, in all our examples, we chose $N_\gamma = \max(100, 10M)$. We assume a conservative cost of the solution of the smallest generalized eigenvalue as $O(N_{\text{space}}^3)$ [PC99], although more refined estimates are available for sparse systems. Hence $\hat{\gamma}(\xi)$ is built with $O(N_\gamma N_{\text{space}}^3)$ operations.

We would like to point out that a similar or higher computational cost is required in existing RB methods to construct a lower bound for the stability constant [QRM11]. Once $\hat{\gamma}(\xi)$ is constructed, evaluating it at a new parameter takes negligible cost.

Our adaptive procedure is similar to the greedy sampling of conventional RB methods [QRM11]. For $m$ atoms, the number of high fidelity solutions is $m(M + 1)$. Hence, the operation count is $O(m(M + 1)N_{\text{space}}^3)$. The key to RB methods is the assumption of an (approximate) affine parametrization of the weak formulation.

Given the assumption of an (approximate) affine parametrization as in Assumption 2, the online cost to solve a reduced problem and evaluate the a posteriori error estimator (2.26) can be made independent of $N_{\text{space}}$ by precomputing and storing offline the affine components of the reduced system and those of the dual norm of the residual, see Section 2.5.1. The most expensive part in the preprocessing steps is the repeated application of an inverse Riesz map $R_V^{-1} : V^* \rightarrow V$. However, note that $R_V$ is $\xi$-independent, which means that we only need to factorize $R_V$ once. After the factorization, the cost of applying $R_V^{-1}$ is $O(N_{\text{space}}^2)$. Overall, an additional cost of $O(N_{\text{space}}^3) + O(mM(M + N + 1)N_{\text{space}}^2)$ is required for the offline preprocessing. The last term can be absorbed into the former cost estimate given that $N_{\text{space}}$ is the dominant dimension, yielding a combined offline construction cost.
of \(O((1 + mM + m)N_{\text{space}}^3)\).

Under the assumption of an affine parametrization, the reduced problem and error indicator \(\epsilon_u(\xi)\) can be computed with \(O(N_k^3 + M^2N_k^2)\) operations at each sample, where \(N_k \leq M + N + 1\) is the size of the reduced problem and is by assumption much smaller than \(N_{\text{space}}\). If a training set \(\Theta_{\text{tr}}\), as in the existing RB methods, is used in the adaptive sampling process, an overhead of \(O(m|\Theta_{\text{tr}}| (N_k^3 + M^2N_k^2))\) is added to the construction cost of the surrogate model. The latter cost is not negligible if the cardinality of \(\Theta_{\text{tr}}\), which is assumed to be an exhaustive set to capture the input uncertainty, is large. In contrast, our approach uses a dynamic training set of much smaller cardinality. The total number of our training samples is bounded by \(N_{\text{tr}}^{\text{init}} + mN_{\text{tr}}^{\text{min}}\), which is substantially smaller than the cardinality of an exhaustive training set \(|\Theta_{\text{tr}}|\). Hence the construction cost for our method is \(O \left( (1 + mM + m)N_{\text{space}}^3 \right) + O \left( m(N_{\text{tr}}^{\text{init}} + mN_{\text{tr}}^{\text{min}})(N_k^3 + M^2N_k^2) \right)\).

In addition, a potential saving for our method can be realized by exploiting the gradient computations. At each atom, we perform one high fidelity solve for the state \(u\) and \(M\) high fidelity solves for the partial derivatives. However, these \(1 + M\) solutions share the same operator. If a direct solver were to be used, only one factorization per atom would be required. The latter would reduce our construction cost to \(O \left( (1 + m)N_{\text{space}}^3 \right) + O \left( m(N_{\text{tr}}^{\text{init}} + m^2N_{\text{tr}}^{\text{min}})(N_k^3 + M^2N_k^2) \right)\). However, we do not exploit this computational advantage in our numerical examples when comparing to other collocation based methods.

In summary, the overall computational cost to build a surrogate model using our adaptive approach is \(O \left( (N_\gamma + 1 + mM + m)N_{\text{space}}^3 \right) + O \left( m(N_{\text{tr}}^{\text{init}} + m^2N_{\text{tr}}^{\text{min}})(N_k^3 + M^2N_k^2) \right)\). Given that our choices of the algorithmic parameters typically results in a very small training set, the last term is negligible in comparison to the first one in large scale problems. Therefore, the overall cost can be approximated as \(O \left( (N_\gamma + 1 + mM + m)N_{\text{space}}^3 \right)\).

Once a surrogate model is constructed, the online computational cost to obtain a surrogate solution at a new sample is simply \(O(N_k^3 + M^2N_k^2)\). Given a fixed \(N\), this number is independent of \(N_{\text{space}}\) as well as of the number of atoms used to construct the surrogate model \(m\).
2.6 Numerical examples

In this section, we present several numerical examples to showcase the capability of the proposed local reduced basis method (local RB) in terms of accurately approximating the solution to PDEs with random parameters.

We compare the performance of our local RB method with the SROM based local Taylor approximation [Gri12] as well as two popular adaptive sparse grid approaches [GZ07, MZ09a]. The first method, denoted as ASG1 [MZ09a] from here on, uses locally supported piecewise multi-linear functions for interpolation. The second approach, denoted as ASG2 [GZ07], uses global Lagrange polynomials for interpolation and a dimension adaptive strategy.

In all the comparisons presented herein, we fix the number of high-fidelity solves for all methods and compare the accuracy of their approximation. The comparison is particularly relevant in solving large-scale stochastic PDEs with limited computational budget. That is, we want to achieve a relatively low approximation error with only a small number of full PDE solves.

It is important to point out that our reported number of full PDE solves does not include the $N_\gamma = \max\{100, 10M\}$ eigenvalue solves to build the surrogate model for the stability constants. Also, we did not exploit the fact that using a direct solver we could reduce the number of high-dimension matrix factorizations by a factor of $M + 1$. However, the advantage of local RB method is apparent from our results.

2.6.1 1D advection diffusion equation

We first consider one-dimensional advection-diffusion equation with random advection. In particular, for this example, we carry out a detailed comparison between SROM with local
RB. Let $D = (0, 1)$. The stochastic boundary value problem is:

$$-
u \frac{\partial^2 u}{\partial x^2}(x, \omega) + b(x, \omega) \frac{\partial u}{\partial x}(x, \omega) = f(x), \quad x \in D, \text{ a.s.} \quad (2.39a)$$

$$u(0, \omega) = u(1, \omega) = 0, \quad \text{a.s.} \quad (2.39b)$$

The diffusivity, $\nu$, and source, $f$, are deterministic whereas the advection field, $b$, is a piecewise constant random field given by

$$b(x, \omega) = [b_1 + \xi_1(\omega)] \mathbb{1}(x \in [0, 0.5)) + [b_2 + \xi_2(\omega)] \mathbb{1}(x \in [0.5, 1)). \quad (2.40)$$

Here, $\xi_1$ and $\xi_2$ are independent random variables that are uniformly distributed on the interval $[-1, 1]$. In our numerical results, we set $b_1 = 0.5$ and $b_2 = 0.8$. This leads to four unique characteristics for the advection. With probability 0.675, the advection field, $b$, will be positive on $D$ (i.e., $\xi_1 > -0.5$ and $\xi_2 > -0.8$). With probability 0.225, $b$ will be negative on $(0, 0.5)$ and positive of $(0.5, 1)$ (i.e., $\xi_1 < -0.5$ and $\xi_2 > -0.8$). With probability 0.075, $b$ will be positive on $(0, 0.5)$ and negative on $(0.5, 1)$ (i.e., $\xi_1 > -0.5$ and $\xi_2 < -0.8$). Finally, with probability 0.025, $b$ will be negative on $D$ (i.e., $\xi_1 < -0.5$ and $\xi_2 < -0.8$).

The other algorithmic parameters for this example are: $N_{\text{space}} = 200$, $N = 3$, $N_{\text{tr}}^{\text{init}} = 50$ and $N_{\text{tr}}^{\text{min}} = 5$. We compare local RB with the SROM approach in [Gri12] (denoted “SROM”). In Figure 2.2, we plot the atoms and corresponding Voronoi cells constructed using the two methods. We compare the atoms at three stages of the algorithm. For SROM, the atoms are chosen to match the distribution of $\xi_1$ and $\xi_2$, and do not exploit the sensitivity of the PDE solution with respect to $\xi$. This leads to atoms that appear uniformly distributed. On the other hand, the local RB approach employs our error indicator, $\epsilon_u(\xi)$, to determine atom location. This permits the adaptive method to locate atoms in the regions of $\Xi$ that are most influential to the PDE solution.

In Figures 2.3-2.6, we compare the performance of the two surrogates (SROM and local RB) for approximating the statistical moments and distributions associated with the PDE solution. In Figure 2.3, we compare the first six moments of $u$ computed using the two surrogate models. As a reference, we use 50,000 Monte Carlo samples of the high-fidelity
Figure 2.2: The location of 5, 20 and 40 atoms and their corresponding Voronoi cells generated using (a) SROM with local approximation and (b) local RB.

PDE solution. We compute the moments using only 5 atoms in (a) and using 20 atoms in (b). The local RB accurately captures the first 6 moments of the PDE solution with only 5 atoms. In comparison, the SROM solution produces large errors even when we use 20 atoms.

In Figure 2.4, we compare the cumulative distribution functions of the PDE solution at \(x = 0.1, 0.5, 0.9\) computed using SROM and local RB. Similarly, in Figure 2.5, we compare the joint probability density function of \(u(0.1)\) and \(u(0.9)\) computed with using SROM and local RB. As before, we compare with the high fidelity solution approximated using 50,000 Monte Carlo samples. In both figures, we computed the distributions and densities in (a) using 5 atoms and in (b) using 20 atoms. The local RB method generates accurate distributions and joint density of \(u\) with only 5 atoms, whereas noticeable errors are still present with 20 atoms using SROM with local approximation.

In Figure 2.6, we compare the relative \(L^2(D)\)-error of the SROM and local RB surrogates. That is,

\[
e_u(\xi) = \frac{\|\bar{u}(\xi) - u(\xi)\|_{L^2(D)}}{\|u(\xi)\|_{L^2(D)}}.
\]

We compute the error, \(e_u(\xi)\), at 1,000 Monte Carlo samples of the parameters \(\xi\). We plot the mean and standard deviation of \(e_u(\xi)\) as a function of the number of atoms used to construct the surrogates in (a). In (b), we plot the distribution of \(e_u(\xi)\) over these 1,000
Figure 2.3: Comparison of the moments of the PDE solution using the stochastic reduced order models with local approximation (blue) and local adaptive reduced basis (red). The black line corresponds to the high fidelity PDE solution using 50,000 Monte Carlo samples. Image (a) depicts the moments computed using 5 atoms while image (b) depicts the moments computed using 20 atoms.

samples. For the distribution, we use 100 atoms to construct the surrogates. As is seen in Figure 2.6 (b), the error committed by the local RB method is more than 2 orders of magnitude smaller than the error committed by the SROM method.

2.6.2 1D Helmholtz equation with damped resonance

In the second example, we consider a stochastic Helmholtz equation and compare local RB with ASG1 and ASG2. Let $D = (0,1)$. The damped Helmholtz problem we consider is defined as

$$-rac{\partial}{\partial x} \left( \nu(x, \omega) \frac{\partial u(x, \omega)}{\partial x} \right) - i \tau u(x, \omega) - \tau^2 u(x, \omega) = f(x), \quad x \in D, \text{ a.s.} \quad (2.42a)$$

$$u(0, \omega) = u(1, \omega) = 0, \quad \text{a.s.} \quad (2.42b)$$
Figure 2.4: Comparison of the cumulative distribution function of the PDE solution at $x = 0.1, 0.5, 0.9$ using SROM with local approximation (blue) and local RB (red). The black line corresponds to the distribution of the high fidelity PDE solution approximated using 50,000 Monte Carlo samples. The (a) plots correspond to approximation with 5 atoms and the (b) plots correspond to approximation with 20 atoms.

Figure 2.5: Comparison of the joint probability density of the PDE solution at $x = 0.1$ and $x = 0.9$ using SROM with local approximation and local RB. The reference was computed using the high-fidelity PDE solution with 50,000 Monte Carlo samples. The (a) plots correspond to approximation with 5 atoms and the (b) plots correspond to approximation with 20 atoms.

where $\nu(x, \omega)$ is a piecewise constant random field, $c$ is a deterministic damping factor, $\tau$ is a deterministic angular frequency, and $f$ is a deterministic source. Stability results for
Figure 2.6: Comparison of the relative $L^2(D)$-error for the stochastic reduced order model with local approximation and local adaptive reduced basis methods. (a) The relative error decreases by increasing the number of atoms used to construct the surrogate models. (b) The distribution of the relative error when using 100 atoms to construct the surrogate models.

this type of problem can be found in [Dem96].

We model the random field $\nu(x, \omega)$ as

$$
\nu(x, \omega) = [0.3 + 1.2\xi_1(\omega)] \mathbb{1}_{[0,0.5]}(x) + [0.5 + 3.5\xi_2(\omega)] \mathbb{1}_{[0.5,1]}(x)
$$

(2.43)

where $\xi_1 \sim \text{Beta}(1,3)$ and $\xi_2 \sim \text{Beta}(3,2)$ with correlation equal to 0.5. We set the angular frequency to $\tau = 4\pi$. These parameters are chosen so that damped resonances occur for certain realizations of the random field $\nu(x, \omega)$. To see this, we plot the $L^2(D)$ norm of the solution $\|u(\cdot, \xi)\|_{L^2(D)}$ over the entire parameter domain $\Xi$ with two different values of damping constants $c = 0.01$ and $c = 0.2$ in Figure 2.7.

From Figure 2.7, it is clear that resonance occurs in three different regions of $\Xi$. It is well known that as $c$ decreases towards zero, the solution norm at resonances increases unboundedly. This behavior makes Helmholtz-type problems particularly challenging for constructing reduced-order models in the parameter space. For this example, we fix $c = 0.2$, which is shown in Figure 2.7 (b).

We plot the real and imaginary components of $u(x, \cdot)$ for some samples and $c = 0.2$ in Figure 2.8. Notice that the solutions vary widely depending on the choice of $\xi$. 

45
Figure 2.7: The $L^2(D)$ norm of $u(\cdot, \xi)$ over the parameter domain $\Xi$ with (a) $c = 0.01$ (b) $c = 0.2$. Notice that resonances occur for certain realizations of $\xi$.

Figure 2.8: Sample solutions of the stochastic Helmholtz problem.

In Figure 2.9, we show the joint density of $\xi$ as well as $N_{\text{bkg}} = 2,000$ background samples of these parameters used in our local RB construction. The samples clearly concentrate on the left part of the domain. The other algorithmic parameters for this example are: $N_{\text{space}} = 200$, $N = 3$, $N_{\text{init}}^\text{tr} = 50$ and $N_{\text{min}}^\text{tr} = 5$. The resulting reduced basis dimension in this example is equal to 6, which is significantly smaller than $N_{\text{space}}$.

In Figure 2.10, we show the samples and corresponding Voronoi cells constructed using the local RB method in three different stages of Algorithm 1. We show the atoms with the corresponding training set $\Xi_{\text{tr}}$ in the background. Notice how the adaptively selected atoms are placed close to resonance regions (Figure 2.7), where we expect the error $\epsilon_u(\xi)$ to be large. It is interesting to notice that the atoms and training samples also cluster around the resonance area on the top right of the domain, even though this area is in the region of low probability density (Figure 2.9). The latter indicates how important it is to accurately
Figure 2.9: (a) Joint density of $\xi$. (b) Background samples of $\xi$.

capture the solution behavior around resonance regions in stochastic Helmholtz problems.

Figure 2.10: Atoms, partition and training set generated by the local RB method when $N_{\text{atom}} = 10, 25, 50$, respectively.

In Figure 2.11, we compare the relative $L_2(D)$ error (2.41) computed for the background samples of Figure 2.9. As can be observed, the mean relative $L_2(D)$ error of the local RB method is almost three orders of magnitude smaller than that of the two adaptive sparse grid methods. The bad performance for the dimension adaptive method ASG2 is anticipated due to the localized nature of resonance regions in parameter space.

We show the mean and variance of the solutions obtained using the different algorithms in Figure 2.12 and Figure 2.13, respectively. In addition, in Figure 2.14, we compare the cumulative distribution function of the PDE solution at $x = 0.1, 0.5, 0.9$. From all comparisons, we see the local RB method has better accuracy than the sparse grid algorithms given the same number of full PDE solves. The statistics of the solution can be accurately
Figure 2.11: Comparison of the relative $L_2(D)$ error of the adaptive sparse grid methods and the local RB method. (a) Mean and standard deviation for different number of full PDE solves, (b) Distribution of the relative error when 600 full PDE solves are used to construct the surrogate models.

captured using the local RB surrogate constructed with only 60 full PDE solves (20 atoms).

2.6.3 2D advection diffusion equation

Finally, we consider the 2D advection-diffusion problem. Let $D = (0,1)^2$, 

$$-\nabla \cdot (\kappa(x,\omega) \nabla u(x,\omega)) + v(x,\omega) \cdot \nabla u(x,\omega) = f(x,\omega) \quad x \in D \text{ a.s.}$$  \hspace{0.5cm} \text{(2.44a)}

$$u(x,\omega) = 0 \quad x \in \Gamma_d \text{ a.s.}$$  \hspace{0.5cm} \text{(2.44b)}

$$\kappa(x,\omega) \nabla u(x,\omega) \cdot n = 0 \quad x \in \Gamma_n \text{ a.s.}$$  \hspace{0.5cm} \text{(2.44c)}

where $\Gamma_d := [0,1] \times \{0\}$ and $\Gamma_n := \partial D \setminus \Gamma_d$. The uncertainty in this example arises from diffusivity, advection and the forcing term.

The diffusion coefficient $\kappa$ is modeled as a random field using the approach in [Gri14].
Figure 2.12: Comparison of the mean of the PDE solution using ASG1 (blue), ASG2 (green) and local RB (red). The black line corresponds to the full PDE solution using 100,000 Monte Carlo samples. (a) depicts the real and imaginary parts of the mean generated by surrogate models constructed using 60 full PDE solves, (b) depicts the real and imaginary parts of the mean generated by surrogate models constructed using 600 full PDE solves.

In particular, we characterize the diffusion coefficient as

$$\kappa(x, \omega) = \sum_{i=1}^{5} \sum_{j=1}^{5} \phi_i(x_1) \phi_j(x_2) \xi_5(i-1)+j(\omega)$$

$$(2.45a)$$

$$\xi_k = 0.02 + 0.48\beta_k \quad k = 1, 2, \ldots, 25$$

$$(2.45b)$$

where $\phi_i(x) = B_{i,5}(x)$ are Bernstein's polynomials of order 5 and $\beta_k \sim \text{Beta}(2, 5)$.

The advection field is divergence free and is parametrized by two uniform random variables as

$$v(x, \omega) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \xi_{26} + \begin{pmatrix} -x_1 \\ x_2 \end{pmatrix} \xi_{27}$$

$$(2.46a)$$

$$\xi_{26} \sim U[10, 20] \quad \xi_{27} \sim U[0, 10]$$

$$(2.46b)$$
Figure 2.13: Comparison of the variance of the PDE solution using ASG1 (blue), ASG2 (green) and local RB (red). The black line corresponds to the full PDE solution using 100,000 Monte Carlo samples. (a) depicts the real and imaginary parts of the variance generated by surrogate models constructed using 60 full PDE solves, (b) depicts the real and imaginary parts of the variance generated by surrogate models constructed using 600 full PDE solves.

Finally, the forcing term $f$ is modeled by four Gaussians functions with uncertain magnitudes and located at $[0.25, 0.25]$, $[0.25, 0.75]$, $[0.75, 0.25]$ and $[0.75, 0.75]$:

$$f(x, \omega) = \exp\left(\frac{-(x_1 - c_i)^2 - (x_2 - d_i)^2}{\sigma^2}\right) \xi_{27+i}$$

where $c_i \in \{0.25, 0.25, 0.75, 0.75, 0.5\}$, $d_i \in \{0.25, 0.75, 0.25, 0.75, 0.5\}$ and $\sigma = 1/3$. The forcing magnitudes are chosen to be uniformly distributed,

$$\xi_{28}, \xi_{31} \sim U[0, 5] \quad \xi_{29}, \xi_{30} \sim U[0, 10].$$

For this example, we use the following algorithmic parameters: $N_{\text{bkg}} = 5,000$, $N_{\text{space}} = 1,089$, $N = 20$, $N^{\text{init}} = 100$ and $N^{\text{min}} = 5$. Hence, the reduced basis dimension is equal to 52 in this case. We again compare the adaptive local RB method with ASG1 and ASG2. We compare the relative $L_2(D)$ error (2.41) over 5,000 test samples of the parameters. As seen from Figure 2.15, the mean relative $L_2(D)$ error of the local RB method is one order of magnitude smaller than that of the two adaptive sparse grid methods. Notice that the
**Figure 2.14:** Comparison of the CDF of the PDE solution at $x = 0.1, 0.5, 0.9$ using ASG1 (blue), ASG2 (green) and local RB (red). The black line corresponds to the distribution of the full PDE solution approximated using 100,000 Monte Carlo samples. The (a) plots correspond to approximation generated by surrogate models constructed using 60 full PDE solves, the (b) plots correspond to approximation generated by surrogate models constructed using 600 full PDE solves.

The sharp computational advantage of the local RB method persists as more full PDE solves are included for all methods.

In Figure 2.16 we show estimates of the mean and variance of the solution for the different algorithms with 1,600 full PDE solves. All fields are computed using 50,000 Monte Carlo samples of the uncertain parameters. Since the surrogate approximation at each sample point is cheap to compute given that the surrogate model has already been constructed, statistics of the solution such as the mean and variance fields are much cheaper to obtain via a surrogate model.

In Figure 2.16, it can be seen that the mean field seems to be well approximated by all
Figure 2.15: Comparison of the relative $L_2(D)$ error of the adaptive sparse grid methods and the local RB method (a) Relative error decreases slowly as more full PDE solves are used to construct the surrogate models. The local RB method is one order of magnitude more accurate than the adaptive sparse grid methods. (b) Distribution of the relative error when 1,600 full PDE solves are used to construct the surrogate models.

surrogate models. However, the variance field is only accurately reproduced by the local RB method. The two adaptive sparse grid methods fail to deliver good approximations to the variance field for the number of full PDE solves used. We also computed the relative $L_2(D)$ error of the mean and variance of the solution. These errors are shown in Table 2.1. We can see from the table that the mean and variance are approximated with much higher accuracy by the local RB surrogate.

<table>
<thead>
<tr>
<th></th>
<th>ASG1</th>
<th>ASG2</th>
<th>Local RB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>$9.40 \times 10^{-2}$</td>
<td>$1.44 \times 10^{-2}$</td>
<td>$1.56 \times 10^{-4}$</td>
</tr>
<tr>
<td>Variance</td>
<td>$1.66 \times 10^{-1}$</td>
<td>$1.29 \times 10^{-1}$</td>
<td>$1.59 \times 10^{-3}$</td>
</tr>
</tbody>
</table>

Table 2.1: Relative $L_2(D)$ error of the predicted moments of $u$ using surrogate models constructed with 1,600 full PDE solves.

We then show the cumulative distribution of $u$ at $x = [0.1, 0.1], [0.5, 0.5]$ and $[0.9, 0.9]$.
Figure 2.16: The approximated mean and variance of $u$ using Monte Carlo method with 50,000 independent parameter samples.

as well as the distribution of the maximum value of $u(x)$ within the spatial domain in Figure 2.17. The distributions (dashed lines) are obtained by approximating $u$ with the three different surrogate models at 50,000 Monte Carlo samples. The reference distributions in black are obtained using full PDE solves at the same samples. In all cases, the surrogate model based on the local RB method is the most accurate in reproducing the various distributions of $u$. 
Figure 2.17: Comparison of the distribution of $u$ using ASG1 (blue), ASG2 (green) and local RB (red). (a) Cumulative distribution of $u$ at $x = [0.1, 0.1], [0.5, 0.5]$ and $[0.9, 0.9]$. (b) Cumulative distribution of the maximum value of $u(x)$ within the spatial domain.
Chapter 3

Evaluating Risk Measures

In this chapter, we extend the local RB method and the adaptive sampling procedure to approximate risk evaluated at a quantity of interest (QoI) that depends on the PDE solutions. As mentioned, many applications require the accurate approximation of the “risk” associated with critical system responses. We will introduce an axiomatic definition of a measure of risk and describe how our local RB approximation can be extended to approximate risk. In general, we are interested in quantifying the risk of some QoI $g : U \rightarrow \mathbb{R}$ evaluated at the PDE solution $u$. We define $G(\xi)$ to be the random variable induced by $G(\xi) = g(u(\xi)) \in \mathcal{X}$ where $\mathcal{X} := L^p(\Omega, \mathcal{F}, \mathbb{P})$ with $1 \leq p < \infty$.

3.1 Risk measures

A risk measure $\mathcal{R} : \mathcal{X} \rightarrow (-\infty, +\infty]$ is a functional that quantifies the overall hazard or risk associated with $G(\xi)$.

3.1.1 Coherent risk measures

The risk measure $\mathcal{R}$ is said to be coherent [ADEH99] if it satisfies the following axioms:

(C1) **Convexity**: $\mathcal{R}(tX + (1 - t)X') \leq t\mathcal{R}(X) + (1 - t)\mathcal{R}(X')$ for all $X, X' \in \mathcal{X}$ and $t \in [0, 1]$;

(C2) **Monotonicity**: $\mathcal{R}(X) \leq \mathcal{R}(X')$ whenever $X, X' \in \mathcal{X}$ with $X \leq X'$ almost surely;

(C3) **Translation Equivariance**: $\mathcal{R}(X + t) = \mathcal{R}(X) + t$ for all $X \in \mathcal{X}$ and $t \in \mathbb{R}$;

(C4) **Positive Homogeneity**: $\mathcal{R}(tX) = t\mathcal{R}(X)$ for all $X \in \mathcal{X}$ and $t \geq 0$. 

55
In the context of engineering applications, (C4) ensures that $R(X)$ inherits, e.g., any change of units in $X$ while (C3) and (C4) ensure that deterministic quantities are “riskless,” i.e., $R(t) = t$ for all $t \in \mathbb{R}$. In addition, (C1) and (C4) are equivalent to (C4) and $R$ being subadditive, i.e., $R(X + X') \leq R(X) + R(X')$ for all $X, X' \in \mathcal{X}$. It is worth pointing out that the common mean-plus-deviation risk measure of order $q$ with $q \approx 1$, 

$$R(X) = \mathbb{E}[X] + c\mathbb{E}[(X - \mathbb{E}[X])^q]^{\frac{1}{q}}, \quad c > 0,$$

is not coherent since it does not satisfy the monotonicity property (C2).

Moreover, if a risk measure $R$ is coherent, Theorem 6.7 in [SDR14] ensures that

$$R(X) = \sup_{\varrho \in \mathfrak{A}} \mathbb{E}[\varrho X]$$

(3.1)

where the risk envelope, $\mathfrak{A}$, is a convex bounded and weak* closed subset of $\mathcal{X}^*$ consisting of probability density functions. That is,

$$\mathfrak{A} \subseteq \{ \varrho \in \mathcal{X}^* : \mathbb{E}[\varrho] = 1, \ \varrho \geq 0 \ a.s. \}.$$

In particular, property (C2) implies $\varrho \geq 0$ a.s. for all $\varrho \in \mathfrak{A}$ and property (C3) ensures if $\varrho \in \mathfrak{A}$ then $\mathbb{E}[\varrho] = 1$.

One popular coherent measure of risk is the conditional value-at-risk (CVaR)

$$R(X) = \text{CVaR}_\beta(X) := \frac{1}{1 - \beta} \int_0^1 q_\alpha(X) \, d\alpha = \inf_{t \in \mathbb{R}} \left\{ t + \frac{1}{1 - \beta} \mathbb{E}[(X - t)_+] \right\}$$

(3.2)

for $0 \leq \beta < 1$, where $q_\alpha(X) := \inf\{\eta \in \mathbb{R} : \mathbb{P}(X \leq \eta) \geq \alpha\}$ is the upper $\alpha$-quantile of $X$ and $(\cdot)_+ := \max\{\cdot, 0\}$. When the distribution function of the random variable $X$ is continuous, $\text{CVaR}_\beta(X)$ is the expected value of $X$ conditioned on the event that $X$ is larger than its $\beta$-quantile [RU00]. Additionally, the risk envelope $\mathfrak{A}$ for $\text{CVaR}_\beta$ is given by

$$\mathfrak{A} = \left\{ \varrho \in \mathcal{X}^* : \mathbb{E}[\varrho] = 1, \ 0 \leq \varrho \leq \frac{1}{1 - \beta} \ a.s. \right\}.$$

Another popular coherent measure of risk is the mean-plus-upper-semideviation of order $q \in [1, \infty),$

$$R(X) = \mathbb{E}[X] + c\mathbb{E}[(X - \mathbb{E}[X])^q]^{\frac{1}{q}}, \quad 0 \leq c \leq 1.$$
The risk envelope $\mathfrak{A}$ for this risk measure is given by

$$
\mathfrak{A} = \{ \theta \in \mathcal{X}^* : \theta = 1 + \theta' - \mathbb{E}[\theta'], \ \|\theta'\|_{\mathcal{X}^*} \leq c, \ \theta' \geq 0 \text{ a.s.} \}.
$$

See [SDR14, § 6.3] for more examples of coherent risk measures.

### 3.1.2 Optimized certainty equivalent

A more general class of risk measures are defined by

$$
\mathcal{R}(X) = \inf_{t \in \mathbb{R}} \{ t + \mathbb{E}[v(X - t)] \}, \quad (3.3)
$$

where $v : \mathbb{R} \to \mathbb{R}$ is convex and satisfies $v(0) = 0$ and $v(x) > x$ for $x \neq 0$.

The risk measure $\mathcal{R}$ defined by (3.3) is called an optimized certainty equivalent [BTT07]. Any such $\mathcal{R}$ satisfies (C1) and (C3). Furthermore, $\mathcal{R}$ satisfies (C2) if and only if $v(x) \leq 0$ whenever $x < 0$ and satisfy (C4) if and only if $v$ is positive homogeneous [RU13]. Three common risk measures arising from (3.3) are the mean-plus-variance

$$
\mathcal{R}(X) = \mathbb{E}[X] + c\mathbb{E}[|X - \mathbb{E}[X]|^2], \quad c > 0,
$$
in which case $v(x) = x + cx^2$ and $p = 2$, the conditional value-at-risk (3.2) in which case $v(x) = (1 - \beta)^{-1}[x^+]$ and $p = 1$, and the entropic risk

$$
\mathcal{R}(X) = \log(\mathbb{E}[^{\exp(cX)}]) / c, \quad c > 0,
$$
in which case $v(x) = (\exp(cx) - 1)/c$ and $p = \infty$. Note that the mean-plus-variance risk measure does not satisfy (C2) and (C4) while the entropic risk measure fails to satisfy (C4).

Another important risk measure we consider in this work is the convex combination of expectation and CVaR, i.e.,

$$
\mathcal{R}(X) = (1 - \alpha)\mathbb{E}[X] + \alpha\text{CVaR}_{\beta}[X], \quad \alpha \in [0, 1], \quad (3.5)
$$

where $\alpha$ is a parameter that tunes the weights between the two. The above risk measure has an associated $v(x) = (1 - \alpha)x + \alpha(1 - \beta)^{-1}[x^+]$ and is coherent satisfying all (C1)-(C4).

Since $v$ is finite valued and convex, it is continuous. Additionally, for $\mathcal{R}$ to be finite valued, we require that one of the following two properties to holds.
Assumption 3. The function $v : \mathbb{R} \to \mathbb{R}$ satisfies one of the following conditions:

(Vp) $\mathcal{X} = L^p(\Omega, \mathcal{F}, P)$ with $p \in [1, \infty)$ and there exists $\gamma_0, \gamma_1 \in \mathbb{R}$ with $\gamma_1 \geq 0$ such that

$$|v(x)| \leq \gamma_0 + \gamma_1 |x|^p, \quad \forall x \in \mathbb{R};$$

(Vx) $\mathcal{X} = L^\infty(\Omega, \mathcal{F}, P)$ and for all $c \geq 0$ there exists $\gamma = \gamma(c) \geq 0$ such that

$$|v(x)| \leq \gamma \quad \forall x \in \mathbb{R}, \quad |x| \leq c.$$

Under Assumption 3, $v : \mathcal{X} \to L^1(\Omega, \mathcal{F}, P)$ is continuous [GKT92, Thm. 4] and $R(X)$ is finite for all $X \in \mathcal{X}$ since $\mathbb{E}[v(X)]$ is.

As we will apply derivative-based optimization algorithms to solve risk-averse optimization problems in Chapter 4 based on the optimized certainty equivalents. We further assume that $v$ is continuously differentiable and its derivative $v'$ satisfies one of the two following conditions.

Assumption 4. The derivative $v' : \mathbb{R} \to \mathbb{R}$ satisfies one of the following conditions:

(Dp) $\mathcal{X} = L^p(\Omega, \mathcal{F}, P)$ with $p \in (1, \infty)$ and there exists $\delta_0, \delta_1 \in \mathbb{R}$ with $\delta_1 \geq 0$ such that

$$|v'(x)| \leq \delta_0 + \delta_1 |x|^{p-1}, \quad \forall x \in \mathbb{R};$$

(Dx) $\mathcal{X} = L^\infty(\Omega, \mathcal{F}, P)$ and for all $c \geq 0$ there exists $\delta = \delta(c) \geq 0$ such that

$$|v'(x)| \leq \delta \quad \forall x \in \mathbb{R}, \quad |x| \leq c.$$

Under Assumption 4, $v : \mathcal{X} \to L^1(\Omega, \mathcal{F}, P)$ is continuously Fréchet differentiable and its derivative at $X \in \mathcal{X}$ is given by $v'(X)$ [GKT92, Thm. 7]. Therefore, $\mathbb{E}[v(\cdot)]$ is also continuously Fréchet differentiable. Note that we require a norm gap between the domain and range of $v$, i.e., since the range is $L^1(\Omega, \mathcal{F}, P)$, the domain must be $L^p(\Omega, \mathcal{F}, P)$ with $p > 1$ in order to obtain differentiability.
3.2 Approximating coherent risk measures

Now, returning to our original goal of approximating \( R(G) \) where \( R \) is a coherent risk measure. The author would like to acknowledge Drew Kouri for contributing the error bound in (3.7) which directly leads to our a posteriori error indicator in (3.8). We first note that for general coherent risk measures, using the convexity (C1), the monotonicity (C2) and the biconjugate representation (3.1), we have that for any \( X, X' \in X' \),

\[
R(X) = R(X' + (X - X')) \leq R(X') + R(X - X') \\
\leq R(X') + R(|X - X'|) = R(X') + \sup_{\theta \in A} E[|\theta|X - X'|].
\]

The same bound holds if we exchange \( X \) with \( X' \) and thus we have the error bound

\[
|R(X) - R(X')| \leq \sup_{\theta \in A} E[|\theta|X - X'|]. \tag{3.6}
\]

We have that (3.6) holds with \( X = G \) and \( X' = \overline{G} = g(\bar{\pi}(\xi)) \). To bound the right-hand side of (3.6), we employ our a posteriori error indicator. Given the current atom set \( \Theta \), the surrogate, \( \bar{u} \), is locally defined over the implicit Voronoi cells \( \Xi_k, k = 1, \ldots, m \). Using the explicit representation (2.17), we define \( G_k = g(u_k(\xi)) \) and note that

\[
|R(G) - R(\overline{G})| \leq \sup_{\theta \in A} \sum_{k=1}^{m} E[|\theta|1_{\Xi_k}(\xi)|G - G_k|] \leq \sum_{k=1}^{m} \sup_{\theta \in A} E[|\theta|1_{\Xi_k}(\xi)|G - G_k|]. \tag{3.7}
\]

Hence, the error in the risk of our surrogate model is bounded by the expected worst case error over the Voronoi tessellation. This bound provides a natural means to balance the error. Intuitively, we should refine cells whose local error is large.

To arrive at a practical local error estimate, we assume the functional \( g \) is Hölder continuous. That is, there exists \( K > 0 \) and \( \alpha > 0 \) such that

\[
|g(w) - g(w')| \leq K\|w - w'||^\alpha_U \quad \forall w, w' \in U.
\]

Under this assumption, we arrive at

\[
E[|\theta|1_{\Xi_k}(\xi)|G - G_k|] \leq K E[|\theta|1_{\Xi_k}(\xi)\|u(\xi) - u_k(\xi)|]s
\]
for all $\theta \in \mathcal{A}$ and all $k = 1, \ldots, m$. Therefore, we can bound the right-hand side of (3.6) using our a posteriori error indicator (2.24) as

$$|\mathcal{R}(G) - \mathcal{R}(\tilde{G})| \leq K \sum_{k=1}^{m} \sup_{\mathcal{A}} \mathbb{E}[\partial \mathbb{1}_{\Xi_k}(\xi)\|e(\xi)\|^2] \leq K \sum_{k=1}^{m} \sup_{\mathcal{A}} \mathbb{E}[\partial \mathbb{1}_{\Xi_k}(\xi)e_u^0(\xi)^\alpha].$$

Our local cell error estimate is thus $\delta_k^0 := \sup_{\mathcal{A}} \mathbb{E}[\partial \mathbb{1}_{\Xi_k}(\xi)e_u^0(\xi)^\alpha] = \mathcal{R}(\mathbb{1}_{\Xi_k}(\xi)e_u^0(\xi)^\alpha)$ whereas our computable (heuristic) local error indicator is

$$\delta_k := \sup_{\mathcal{A}} \mathbb{E}[\partial \mathbb{1}_{\Xi_k}(\xi)e_u(\xi)^\alpha] = \mathcal{R}(\mathbb{1}_{\Xi_k}(\xi)e_u(\xi)^\alpha). \quad (3.8)$$

We note that if we are only interested in approximating the QoI and the associated risk measures, an alternative error indicator for $G$ that is possibly more efficient can be derived using a dual-weighted-residual method in [CS15, DC15]. The benefit of using dual-weighted-residual is that we don’t need an approximation for the inf-sup constant, an additional reduced adjoint solve, however, needs to be carried out at each sample to evaluate the indicator. We refer the readers to [DC15] for more details.

Recall that in Section 2.4 we have a cell error indicator $\eta_k$ which is used in the greedy adaptive sampling approach in Algorithm 1. It is straight forward to modify Algorithm 1 to account for the error associated with the risk computation by replacing $\eta_k$ with $\delta_k$. This substitution provides an adaptive reduced basis approach for efficiently evaluating the risk associated with QoIs that depends on the PDE solution.

### 3.3 Numerical examples

We present three numerical examples using the same PDEs from Section 2.6, but with a different goal to accurately approximate CVaR applied to some QoI’s of the PDE solution.
3.3.1 1D advection diffusion equation

First, we consider the PDE in Section 2.6.1. We investigate the performance of our adaptive algorithm for estimating the CVaR of the following QoI,

\[ G(\xi) = g(u(\xi)) = \|u(\xi) - w\|_{L^2(D)} \]  \hspace{1cm} (3.9)

where \( w(x) = 2\sin(\pi x) \) is some desired system output. By the reverse triangle inequality, we have that \( g \) is Lipschitz continuous with constant \( K = 1 \), i.e., \( g \) is Hölder continuous with \( K = 1 \) and \( \alpha = 1 \). We compare the approximation of the CVaR of \( G \) using SROM with local approximation and with our local RB method.

In Figure 3.1, we plot the Voronoi tessellations from the first six steps of Algorithm 1 modified to approximate the risk CVaR \( G \) with \( \beta = 0.8 \). The color of each cell corresponds to the logarithm of the local error indicator \( \delta_k \) as defined in (3.8). At each iteration of Algorithm 1, the cell with the largest error, \( \delta_k \), is selected and a new atom is found within the cell, which are depicted as the red dots. These atoms are depicted as the red dots in Figure 3.1. As more atoms are added, the local errors, \( \delta_k \), decrease resulting in an accurate approximation of CVaR \( G \).

Figure 3.1: Voronoi tessellation from the first six iterations of Algorithm 1 for approximation CVaR \( G \). The color of each cell corresponds the logarithm of the local error indicator, \( \log_{10}(\delta_k) \). The red dots correspond to the selected atoms.
In Figure 3.2 we compare the estimates of $\text{CVaR}_\beta(G)$ computed using the SROM and local RB methods with only 5 atoms. We use Monte Carlo applied to the high-fidelity solution as reference. Figure 3.2 clearly demonstrates that $\text{CVaR}_\beta(G)$ is well approximated using the local RB method, resulting in a relative error of approximately $2.2 \times 10^{-3}$. In comparison, the SROM approximation results in a relative error of approximately $6.8 \times 10^{-2}$. That is, the local RB method produces an estimate of CVaR that is 30 times more accurate than that of the SROM method.

![Figure 3.2](image)

**Figure 3.2:** Comparison of the SROM and the local RB for estimating the CVaR of the QOI $G$ using Monte Carlo as reference. (a) The quantity of interest $G(\xi)$. (b) The cumulative distribution function of $G$ computed using the three methods. Included are the VaR and the CVaR with confidence level $\beta = 0.8$ for each approximation.

### 3.3.2 1D Helmholtz equation with damped resonance

In the second example, we consider the stochastic Helmholtz equation in Section 2.6.2. We investigate the performance of our adaptive algorithm for estimating the CVaR of the QoI

$$G(\xi) = g(u(\xi)) = \|u(\cdot, \xi)\|_{L^2(D)}.$$

(3.10)

The functional $g$ is Hölder continuous with $\alpha = 1$ and $K = 1$. We aim to approximate $\text{CVaR}_\beta(G)$ with $\beta = 0.9$. We refer to the QoI and $\text{CVaR}_\beta$ computed from full PDE solutions the reference values and compare these values with the local RB method as well as the two sparse grid methods.
Figure 3.3(a) shows the true QoI over the parameter domain $\Xi$, Figure 3.3(b) shows the 100,000 Monte Carlo samples that we use to compute the risk, the red dots indicate the samples that contribute to the risk, i.e., the 10% largest samples of $G$. The cumulative distribution of $G(\xi)$ as well as CVaR$_{\beta}(G)$ are shown in Figure 3.3(c).

![Figure 3.3](image)

**Figure 3.3:** (a) The true QoI, (b) the parameter samples that contribute to CVaR$_{\beta}(G)$ are shown in red, (c) the cumulative distribution of $G(\xi)$ and CVaR$_{\beta}(G)$.

In Figure 3.4(a), we plot the Voronoi tessellations from the first six steps of Algorithm 1 modified to approximate the risk CVaR$_{0.9}(G)$. The color of each cell corresponds to the logarithm of the local error indicator $\delta_k$ as defined in (3.8). At each iteration of Algorithm 1, the cell with the largest error, $\delta_k$, is selected and a new atom is found within the cell, which are depicted as the red dots. As more atoms are added, the local errors, $\delta_k$, decrease resulting in an accurate approximation of CVaR$_{\beta}(G)$. In Figure 3.4(b), we plot the relative error of the risk,

$$e_R = \frac{|\text{CVaR}_\beta(G) - \text{CVaR}_\beta(G)|}{\text{CVaR}_\beta(G)}$$

(3.11)

(corresponding to the local RB method with different numbers of atoms. Clearly, as more atoms are added, the relative error drops.

In Figure 3.5(a) and (c), we show $G$ computed over the parameter space using local RB, ASG1 and ASG2 with 60 and 600 PDE solves, respectively. The comparisons of the
Figure 3.4: (a) Voronoi tessellation from the first six iterations of Algorithm 1. The color of each cell corresponds the logarithm of the local error indicator, $\log_{10}(\delta_k)$. The red dots correspond to the selected atoms. (b) shows the relative error of risk obtained with the local RB method using varying numbers of atoms.

corresponding CVaR$_\beta(G)$ are shown in Figure 3.5(b) and (d) with the relative error $e_R$ shown in Table 3.1. The comparison clearly demonstrates that the CVaR$_\beta(G)$ is captured much more accurately using the local RB method.

<table>
<thead>
<tr>
<th>PDE Solves</th>
<th>ASG1</th>
<th>ASG2</th>
<th>Local RB</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>$4.04 \times 10^{-1}$</td>
<td>$4.35 \times 10^{-1}$</td>
<td>$3.78 \times 10^{-4}$</td>
</tr>
<tr>
<td>600</td>
<td>$2.84 \times 10^{-2}$</td>
<td>$1.70 \times 10^{-2}$</td>
<td>$6.94 \times 10^{-8}$</td>
</tr>
</tbody>
</table>

Table 3.1: The relative error of CVaR$_{0.9}(G)$ obtained using ASG1, ASG2 and local RB surrogate models with 60 and 600 PDE solves, respectively.

3.3.3 2D advection diffusion equation

Finally, we consider the 2D advection diffusion equation in Section 2.6.3. We use the local RB method and the two sparse grid methods to approximate the CVaR$_\beta(\|u(\xi)\|_{L^2(D)})$ with $\beta = 0.9$. Figure 3.6 shows the relative error of the risk obtained by the local RB method
Figure 3.5: (a) and (c): Comparison of the $G(\xi)$ obtained using ASG1, ASG2 and local RB surrogate models with 60 and 600 PDE solves, respectively. The (b) and (d) plot the corresponding risk generated by the surrogate models with 60 and 600 PDE solves, respectively.

with varying numbers of atoms. Figure 3.7 shows the CDF of the QoI obtained using the different methods when 1,600 PDE solves (50 atoms for local RB) are used to construct the surrogate models. The risk is accurately captured by the local RB method with a relative error of less than 0.1%, whereas the relative error is 3.56% for ASG1 and 3.97% for ASG2, as shown in Table 3.2.

<table>
<thead>
<tr>
<th></th>
<th>ASG1</th>
<th>ASG2</th>
<th>Local RB</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_R$</td>
<td>$3.56 \times 10^{-2}$</td>
<td>$3.97 \times 10^{-2}$</td>
<td>$3.90 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

Table 3.2: The relative error of $\text{CVaR}_{0.9}(G)$ obtained using ASG1, ASG2 and local RB surrogate models (constructed with 1,600 PDE solves).
Figure 3.6: The relative error of $\text{CVaR}_{0.9}(\|u(\xi)\|_{L_2(D)})$ obtained by the local RB method with different number of atoms.

Figure 3.7: Comparison of the $\text{CVaR}_{0.9}(\|u(\xi)\|_{L_2(D)})$ obtained using ASG1, ASG2 and local RB surrogate models.
Chapter 4

Risk-averse PDE-constrained Optimization

In this chapter, our goal is to devise a computational framework to efficiently solve risk-averse PDE-constrained optimization problems using local RB surrogate models for the PDE solutions. Few methods exist for constructing efficient surrogate models in the context of PDE-constrained optimization under uncertainty. For example, the authors in [KHRvBW14] introduce an adaptive sparse-grid approach whereas the author in [Zah16] introduces a global reduced basis method. However, to obtain convergence of the optimization algorithm when using surrogate models, the errors associated with the surrogate approximation need to be properly managed. To this end, both [KHRvBW14] and [Zah16] employ an inexact trust-region (TR) framework to manage the error in the surrogate approximations throughout the optimization process. Additionally, under standard assumptions, the authors in [KHRvBW14] show that the inexact TR algorithm is guaranteed to converge from any initial guess, provided that errors in the approximation of the objective function and its gradient are adequately bounded.

In this chapter, we combine the inexact TR framework of [KHRvBW14] with the adaptive local RB method. We use the local RB method to systematically construct the surrogate models for the TR subproblem and objective function evaluations. To this end, we derive error bounds for the objective and gradient approximation using \textit{a posteriori} error indicators and subsequently introduce adaptive sampling schemes for the sequential construction of the surrogate models. Throughout the TR algorithm, we maintain two separate surrogate models: one to approximate the objective function gradient and one to approximate the objective function value. These two surrogate models employ different sets of basis functions and different error indicators to efficiently achieve the required error bounds of the TR method. We demonstrate the performance of our approach through several nu-
merical examples, which show that our method can solve risk-averse control problems with PDE constraints using far fewer full PDE PDE solves than those required with Monte Carlo methods.

4.1 Risk-averse control problem with PDE constraint

We follow the general formulations in Section 2.1. In addition, to formulate the optimization problem, we restate equation (2.2) with an additional control term. Now, let $U, V$ and $Z$ be real Hilbert spaces. Here, $U$ is the deterministic state space (i.e., the space of PDE solutions) and $Z$ is the space of optimization variables. We will refer to $z \in Z$ as a control variable. Since controls typically must be implemented prior to observing the uncertainty in the PDE solution, we require that $Z$ is a space of deterministic functions or vectors. We consider the following parametrized, linear PDE: for fixed $z \in Z$, find $u : \Xi \rightarrow U$ such that

$$M(u(\xi), z; \xi) := \mathcal{L}(\xi)u(\xi) + \mathcal{B}(\xi)z + \ell(\xi) = 0 \quad \forall \xi \in \Xi$$

(4.1)

where $\mathcal{L}(\xi) \in L(U, V^*)$ for all $\xi \in \Xi$ is the parametrized PDE operator, $\mathcal{B}(\xi) \in L(Z, V^*)$ for all $\xi \in \Xi$ is the parametrized control operator and $\ell(\xi) \in V^*$ for all $\xi \in \Xi$ is the parametrized load or force. If $U, V$ and $Z$ are finite-dimensional, $\mathcal{L}(\xi)$ and $\mathcal{B}(\xi)$ are matrices and $\ell(\xi)$ is a vector representing, e.g., the discretization of a PDE. Again, we define the parametrized bilinear form $a(\cdot, \cdot; \xi) : U \times V \rightarrow \mathbb{R}$ associated with $\mathcal{L}(\xi)$ for $\xi \in \Xi$ by

$$a(u, v; \xi) := \langle \mathcal{L}(\xi)u, v \rangle_{V^*, V} \quad \forall u \in U, \ v \in V.$$

The variational problem corresponding to (4.1) is: Find $u : \Xi \rightarrow U$ such that

$$a(u(\xi), v; \xi) + \langle \mathcal{B}(\xi)z, v \rangle_{V^*, V} + \langle \ell(\xi), v \rangle_{V^*, V} = 0 \quad \forall v \in V, \ \xi \in \Xi.$$

(4.2)

In addition to Assumption 1, we impose the following additional assumptions regarding (4.2).

**Assumption 5.** We assume that the weak form (4.2) satisfies Assumption 1, in addition, we assume that
1. The control operator satisfies: \( \exists \kappa_1 \geq 0 \) independent of \( \xi \in \Xi \) such that \( \| B(\xi) \|_{L(Z,V^*)} \leq \kappa_1 \) for all \( \xi \in \Xi \);

2. The force satisfies: \( \exists \kappa_2 \geq 0 \) independent of \( \xi \in \Xi \) such that \( \| \ell(\xi) \|_{V^*} \leq \kappa_2 \) for all \( \xi \in \Xi \).

The well-posedness of problem is ensured under Assumption 1 for all \( \xi \in \Xi \) and \( z \in Z \), and that the solution is bounded independent of \( \xi \in \Xi \) due to Assumption 1 and 5. We denote the solution of (4.1) by \( S(\xi; z) \in U \) and assume that \( S(\xi(\cdot); z) \) is strongly measurable, i.e., \( S(\xi(\cdot); z) \in L^\infty(\Omega, \mathcal{F}, P; U) \) for all \( z \in Z \).

Turning our attention to the target optimization problem, let \( \wp : Z \to \mathbb{R} \) and \( G : U \times \Xi \to \mathbb{R} \), and note that \( G(S(\xi(\omega); z), \xi(\omega)) \) is a function of \( \omega \) and hence is viewed as a random variable. We assume that \( G(S(\xi; z), \xi) \in \mathcal{X} := L^p(\Omega, \mathcal{F}, P) \) for some \( p \in [1, \infty] \) and seek to minimize

\[
 J(z) = \wp\left(G(S(\xi; z), \xi)\right) + \wp(z) \tag{4.3}
\]

where \( \mathcal{R} : \mathcal{X} \to \mathbb{R} \) is a risk measure introduced in Section 3.1.2, i.e., an optimized certainty equivalent [BTT07].

It is important to point out that the objective function in (4.3) may be nonsmooth in \( z \) depending on the function \( v \). Therefore, to take advantage of gradient-based optimization algorithms, we only consider differentiable \( v \). In the context of CVaR, the associated \( v \) is not differentiable at \( x = 0 \). As such, one could consider a smooth approximation similar to those studied in [KS16].

To minimize (4.3), we incorporate \( t \) from the definition of \( \mathcal{R} \), see (3.3), as an additional optimization variable and solve

\[
 \min_{t \in \mathbb{R}, z \in Z} \tilde{J}(t, z) \quad \text{where} \quad \tilde{J}(t, z) := t + \mathbb{E}\left[v\left(G(S(\xi; z), \xi) - t\right)\right] + \wp(z). \tag{4.4}
\]

Under appropriate smoothness assumptions on \( M, G, \wp \) and \( v \), see Assumption 2.3 in [KS16] for example, the objective function \( \tilde{J} \) is continuously Fréchet differentiable and the
partial derivatives \( \hat{J} \) are

\[
\nabla_z \hat{J}(t, z) = \mathbb{E} \left[ v'(G(S(\xi; z), \xi) - t)B(\xi)^*\lambda(\xi) \right] + \nabla \psi(z) \tag{4.5a}
\]

\[
\nabla_t \hat{J}(t, z) = 1 - \mathbb{E} \left[ v'(G(S(\xi; z), \xi) - t) \right] \tag{4.5b}
\]

where \( \lambda : \Xi \rightarrow V \) solves the adjoint equation

\[
\mathcal{L}(\xi)^* \lambda(\xi) = -\nabla_u G(S(\xi; z), \xi) \quad \forall \xi \in \Xi. \tag{4.6}
\]

Notice that the computation of the gradient requires the solution of the state equation (4.1) and the solution of the adjoint equation (4.6) for all \( \xi \in \Xi \). We denote the adjoint solution for a fixed state variable \( u \) by \( \Lambda(\xi; u) \). For example, in (4.6), we can write \( \lambda(\xi) = \Lambda(\xi; S(\xi; z)) \) for \( \xi \in \Xi \). To simplify notation in the subsequent sections, we denote \( W := \mathbb{R} \times Z, \ w = (t, z) \) and \( \hat{J}(w) = \hat{J}(t, z) \).

### 4.2 Inexact trust region algorithm with local RB surrogates

To overcome the potentially enormous computational cost of minimizing \( \hat{J} \), we develop an adaptive optimization framework based on two key components. The first component is local RB method to build computationally inexpensive approximations of \( S \). Using these low-cost surrogate models, we can approximate the objective function value and its gradient in a computationally tractable manner. The second key component of our method is the inexact TR algorithm developed in [KHRvBW14], which prescribes accuracy requirements on the surrogate model approximations that guarantee global convergence to local minimizers of \( \hat{J} \).

#### 4.2.1 Inexact trust region method

We now briefly describe the inexact TR algorithm. For a concrete statement of the algorithm as well as the rigorous convergence theory, we refer to [KHRvBW14]. Given the current iterate \( w_k = (t_k, z_k) \in W \), we construct a local model \( m_k \) of the objective function
s \rightarrow \hat{J}(w_k + s) \text{ in the region } \{s \in W : \|s\|_W \leq \Delta_k\}. \text{ Here } \Delta_k > 0 \text{ is the current TR radius and } \| \cdot \|_W \text{ denotes a norm on } W, \text{ e.g., } \|w\|_W^2 = t^2 + \|z\|_Z^2. \text{ We determine the trial step } s_k \text{ by solving or approximately solving the TR subproblem}

\min_{s \in W} m_k(s) \text{ subject to } \|s\|_W \leq \Delta_k. \tag{4.7}

We require that any (approximate) solution to (4.7) satisfies the so-called fraction of Cauchy decrease condition, that is

\begin{equation}
m_k(0) - m_k(s_k) \geq \kappa_0 \|\nabla m_k(0)\| \min \left\{ \Delta_k, \frac{\|\nabla m_k(0)\|_W}{\beta_k} \right\}, \tag{4.8}
\end{equation}

where \(\kappa_0 > 0\) and \(\beta_k = 1 + \sup_{\{s : \|s\|_W \leq \Delta_k\}} \|\nabla^2 m_k(s)\|_{L(W,W^*)}\).

Once a step \(s_k\) is computed, we decide whether to accept or reject \(s_k\) based on the ratio of the actual and predicted reduction

\(\rho_k := \frac{\text{ared}_k}{\text{pred}_k}\) \text{ where } \text{ared}_k := \hat{J}(w_k) - \hat{J}(w_k + s_k) \text{ and } \text{pred}_k := m_k(0) - m_k(s_k).

Here, \text{ared}_k \text{ denotes the actual reduction obtained by the step } s_k \text{ while } \text{pred}_k \text{ denotes the predicted reduction based on the model } m_k.

Now, to ensure convergence of the TR algorithm from arbitrary initial guesses, we require that the model in (4.7) satisfies the inexact gradient condition

\(\|\nabla m_k(0) - \nabla \hat{J}(w_k)\|_W \leq \kappa \min \{\|\nabla m_k(0)\|_W, \Delta_k\}\)

for some \(\kappa > 0\) independent of \(k\). Additionally, since the computation of \text{ared}_k requires the evaluation of the true objective \(\hat{J}\), which is computationally prohibitive or even impossible, we evaluate \text{ared}_k inexactiy. To this end, we introduce a model \(\hat{J}_k\) that approximates \(\hat{J}\) and allows us to approximate \text{ared}_k as

\(\text{ared}_k \approx \text{cred}_k := \hat{J}_k(w_k) - \hat{J}_k(w_k + s_k).\)

Here, \text{cred}_k is the computed reduction. To guarantee convergence of the TR algorithm, we require that

\(|\text{ared}_k - \text{cred}_k| \leq K \theta_k \text{ with } \theta_k^{\omega} \leq \eta \min \{\text{pred}_k, r_k\}\) \tag{4.11}
for some $K > 0$, fixed $\omega \in (0, 1)$ and

$$\eta < \min\{\eta_1, 1 - \eta_2\} \quad \text{and} \quad r_k \geq 0 \quad \text{with} \quad \lim_{k \to \infty} r_k = 0$$

where $0 < \eta_1 < \eta_2 < 1$ are algorithmic constants and $r_k$ is a forcing sequence. The authors in [KHRvBW14] prove that as long as one can control the surrogate approximation error for the objective function value and gradient to satisfy conditions (4.9) and (4.11), then the TR algorithm converges to a local minimizer for any starting point $w_0 = (t_0, z_0)$.

### 4.2.2 Inexact trust regions with local RB approximation

To construct approximations that satisfy (4.9) and (4.11), we employ the local RB method. Recall in Section 2.4, we rely on a greedy adaptive sampling procedure to select the local RB atom set $\Theta := \{\xi_k\}_{k=1}^n$. The adaptive selection of $\Theta$ is guided by reliable *a posteriori* error indicators, denoted by $\epsilon_u(\xi; z)$, i.e.,

$$\|\hat{S}(\xi; z) - S(\xi; z)\|_U \leq \epsilon_u(\xi; z)$$

where $x \leq y$ denotes “$x$ is less than or equal to a constant times $y$.” Here, we make explicit the dependence of $\epsilon_u$ on $z$ because the residual computation in (2.26) for (4.1) depends on the control $z$. In local RB, given $k$ atoms, the next atom $\xi_{k+1}$ is selected from the region of $\Xi$ where the current surrogate error $\epsilon_u(\xi; z)$ is the largest. In addition, we have shown in Chapter 3 that the error indicator $\epsilon_u(\xi; z)$ can be further used to build more complex error indicators that are specifically targeted for the approximation of coherent risk measures evaluated at quantities of interest depending on PDE solutions. The versatility provided by the adaptive sampling procedure tailored to individual tasks is of fundamental importance for the algorithmic development in this chapter.

To simplify the subsequent analysis, we assume that the random inputs $\xi$ are discretely distributed. Let $\{\xi_j\}_{j=1}^N$ denote the atoms of $\xi$ with the associated probabilities $\{p_j\}_{j=1}^N$ with $p_1 + \ldots + p_N = 1$ and $p_j > 0$ for $j = 1, \ldots, N$. In this setting, we can then rewrite
(3.3) as
\[
\mathcal{R}(G(S(\xi; z), \xi)) = \inf_{t \in \mathbb{R}} \left\{ t + \sum_{j=1}^{N} p_j v(G(S(\xi_j; z), \xi_j) - t) \right\}.
\] (4.12)

For example, if \( \{\xi_j\} \) are Monte Carlo samples of \( \xi \), then \( p_j = 1/N \) for \( j = 1, \ldots, N \).

The assumption that \( \xi \) is discretely distributed can be relaxed if we are given a discrete approximation to \( \xi \) that can be refined to control its approximation error (e.g., quadrature approximation or Monte Carlo).

As mentioned above, to obtain surrogate models that meet the accuracy requirements of the TR algorithm, we employ the adaptive local RB method. To this end, we construct two surrogate models of \( S(\xi; z) \), denoted by \( S_{\text{mod}}(\xi; z) \) and \( S_{\text{obj}}(\xi; z) \), which we then use to construct \( m_k \) and \( \tilde{J}_k \), respectively. Moreover, since (4.9) depends on gradient information, we construct a surrogate model for the adjoint variable \( \Lambda(\xi; u) \), denoted \( \Lambda_{\text{mod}}(\xi; u) \). By constructing \( S_{\text{mod}}(\xi; z) \), \( \Lambda_{\text{mod}}(\xi; u) \) and \( S_{\text{obj}}(\xi; z) \) adaptively, we aim to achieve the desired error levels with as few full PDE solves as possible.

Using the local RB approximation \( S_{\text{mod}} \), we can choose, e.g., the quadratic model for the subproblem
\[
m_k(s) = \frac{1}{2} \langle H_k s, s \rangle_W + \langle g_k, s \rangle_W
\] (4.13)
where \( s \in W, H_k \in L(W, W) \) is an approximation of the Hessian of \( \tilde{J}(w_k) \) and \( g_k \in W \) is an approximation of the gradient of \( \tilde{J}(w_k) \). Since our focus is on satisfying (4.9), we assume \( H_k \) is provided and is bounded for all \( k \). We can then choose \( g_k \) by approximating the gradients in Equation (4.5) by
\[
g_k = \left[ \frac{1 - \sum_{j=1}^{N} p_j [v'(G(S_{\text{mod}}(\xi_j; z_k), \xi_j) - t_k)]}{\sum_{j=1}^{N} p_j [v'(G(S_{\text{mod}}(\xi_j; z_k), \xi_j - t_k)B(\xi)^*\Lambda_{\text{mod}}(\xi_j; S_{\text{mod}}(\xi_j; z_k))] + \nabla \psi(z_k)} \right].
\] (4.14)

For affinely parametrized PDEs satisfying Assumption 2, this approximation based on the local RB surrogates can be evaluated very efficiently with negligible cost that is independent
of the dimension of the full PDE, see Section 2.5 for more details. In addition, efficient methods such as the truncated conjugate gradient method [NW06, CGT00] can be applied to obtain an approximate solution to the TR subproblem (4.7) that is known to satisfy the fraction of Cauchy decrease condition.

We can also choose the following trust-region subproblem model

\[ m_k(s) = (t_k + \tau) + \sum_{j=1}^{N} p_j \nu(G(S_{\text{mod}}(\xi_j; z_k + \zeta); \xi_j) - (t_k + \tau)) + \varphi(z_k + \zeta) \quad (4.15) \]

where \( \tau \in \mathbb{R}, \zeta \in \mathbb{Z} \) and \( s = (\tau, \zeta) \). Though this \( m_k \) is often highly nonlinear, its value and gradient evaluation solely requires evaluations of the local RB surrogates \( S_{\text{mod}}(\xi_j; z_k + \zeta) \) and \( A_{\text{mod}}(\xi_j; S_{\text{mod}}(\xi_j; z_k + \zeta)) \) as well. This TR subproblem can be solved accurately using a variety of nonlinear optimization solvers. For example, we employ an interior point method [NW06] with a Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [Fle13] to solve the above subproblem.

A large benefit of employing the above nonlinear subproblem and solving it exactly is that the subproblem model fully exploits the accuracy of local RB surrogates and always yields high-quality proposals that are accepted by the TR algorithm. This in turn results in significant computational savings by reducing the total number of TR steps and hence full PDE evaluations required to maintain/refine the surrogates for each TR step. In addition, one can easily verify the fraction of Cauchy decrease condition once a minimizer of the subproblem is obtained. For example, a simple check to verify the condition is by checking

\[ m_k(0) - m_k(s_k) \geq \kappa_0 \| \nabla m_k(0) \| \min \{ \Delta_k, \| \nabla m_k(0) \|_W \}, \quad (4.16) \]

for some preset constant \( \kappa_0 \), proposals satisfying (4.16) automatically satisfies the fraction of Cauchy decrease condition (4.8) as well.

For affinely parametrized linear PDEs (see Assumption 2) where the cost of solving the subproblem is negligible, we always adopt this nonlinear subproblem and solve it exactly. However, the benefit of using this model decreases for general PDEs, where affine parametrization is not present and hence solving the above nonlinear subproblem is costly as well, in which case one can employ the quadratic model instead.

74
As shown in previous chapters, a key requirement to apply local RB method is the availability of a residual based error bound so that we can use an \textit{a posteriori} error indicator in an adaptive sampling process to construct surrogate models with efficiency. To this end, we show that residual based error bound holds for both the forward problem (4.1) and the adjoint problem (4.6). We require the following assumptions on $G$ and $v$.

**Assumption 6.** We require the following Lipschitz continuity conditions to hold:

(L1) $G(\cdot, \xi)$ is continuously Fréchet differentiable for all $\xi \in \Xi$ and there exists $K_G \geq 0$ independent of $\xi$ such that

$$\|\nabla_u G(u, \xi) - \nabla_u G(u', \xi)\|_{U^*} \leq K_G \|u - u'\|_U \quad \forall u, u' \in U \quad \forall \xi \in \Xi;$$

in addition, there exists $C_G \geq 0$ independent of $\xi$ such that

$$\|\nabla_u G(0, \xi)\|_{U^*} \leq C_G$$

for $\forall \xi \in \Xi$.

(L2) There exists $K_v \geq 0$ such that

$$|v'(x) - v'(x')| \leq K_v |x - x'| \quad \forall x, x' \in \mathbb{R}.$$

Under Assumption 1 and a fixed $z \in Z$, the error $e^u_{\text{mod}}(\xi; z) = S_{\text{mod}}(\xi; z) - S(\xi; z)$ between the local RB surrogate and the true solution satisfies the residual-based error bound

$$\|e^u_{\text{mod}}(\xi; z)\|_U \leq \frac{r(S_{\text{mod}}(\xi; z), \cdot; \xi)}{\gamma_1(\xi)}$$

which is the bound we have derived in (2.24). The same bound holds for $e^u_{\text{obj}}(\xi; z) = S_{\text{obj}}(\xi; z) - S(\xi; z)$, i.e.,

$$\|e^u_{\text{obj}}(\xi; z)\|_U \leq \frac{r(S_{\text{obj}}(\xi; z), \cdot; \xi)}{\gamma_1(\xi)}.$$

For the adjoint problem (4.6), we need to address two approximation errors, the first error $e^\lambda_1(\xi; z) = \Lambda_{\text{mod}}(\xi; S_{\text{mod}}(\xi; z)) - \Lambda(\xi; S_{\text{mod}}(\xi; z))$ is related to the surrogate approximation under the same state $S_{\text{mod}}(\xi; z)$. The second error $e^\lambda_2(\xi; z) = \Lambda(\xi; S_{\text{mod}}(\xi; z)) - \Lambda(\xi; S_{\text{mod}}(\xi; z))$. 

75
\( \Lambda(\xi; S(\xi; z)) \) is due to the surrogate approximation of the state \( S(\xi; z) \) that enters the right-hand-side of (4.6). By Assumption 1 (in particular Equation (2.4c)), it is easy to show that, the residual based error bound holds for \( \epsilon^\lambda_1(\xi; z) \), i.e.,

\[
\| \epsilon^\lambda_1(\xi; z) \|_V \leq \frac{\| \mathcal{L}(\xi)^* \Lambda_{\text{mod}}(\xi; S_{\text{mod}}(\xi; z)) + \nabla_u G(S_{\text{mod}}(\xi; z), \xi)\|_{U^*}}{\gamma_2(\xi)}
\]  

(4.19)

For \( \epsilon^\lambda_2(\xi; z) \), we have that

\[
\gamma_2(\xi) \| \epsilon^\lambda_2(\xi; z) \|_V \leq \sup_{u \in U \setminus \{0\}} \frac{\langle \mathcal{L}(\xi)^* \epsilon^\lambda_2(\xi; z), u \rangle_{U^*}}{\|u\|_U}
\]

\[
= \sup_{u \in U \setminus \{0\}} \frac{\| \nabla_u G(S(\xi; z), \xi) - \nabla_u G(S_{\text{mod}}(\xi; z), \xi), u \rangle_{U^*}}{\|u\|_U}
\]

\[
= \| \nabla_u G(S(\xi; z), \xi) - \nabla_u G(S_{\text{mod}}(\xi; z), \xi)\|_{U^*}
\]

\[
\leq K_G \| S_{\text{mod}}(\xi; z) - S(\xi; z) \|_U = K_G \| \epsilon^u_{\text{mod}}(\xi; z) \|_U \quad \forall \xi \in \Xi.
\]

where the first inequality is by Assumption 1 and the second inequality is by Assumption 6. Hence

\[
\| \epsilon^\lambda_2(\xi; z) \|_V \leq \frac{K_G \| \epsilon^u_{\text{mod}}(\xi; z) \|_U}{\gamma_2(\xi)},
\]

(4.20)

which again, by Equation (4.17), allows us to use a residual based error bound for \( \epsilon^\lambda_2(\xi; z) \).

Computable \textit{a posteriori} error indicators can be easily built based on Equation (4.17), (4.18), (4.19) and (4.20) using the same idea as in Section 2.4, for example, by building independent surrogate models for \( \gamma_1(\xi) \) and \( \gamma_2(\xi) \). We denote the computable \textit{a posteriori} error indicators for \( S_{\text{mod}}(\xi; z) \), \( \Lambda_{\text{mod}}(\xi; u) \) and \( S_{\text{obj}}(\xi; z) \) by \( \epsilon^u_{\text{mod}}(\xi; z) \), \( \epsilon^\lambda_{\text{mod}}(\xi; u) \) and \( \epsilon^u_{\text{obj}}(\xi; z) \), respectively; that is,
\[ \|e_{\text{mod}}^u(\xi; z)\|_V \lesssim e_{\text{mod}}^u(\xi; z), \]
\[ \|e_1^\lambda(\xi; z)\|_V \lesssim e_{\text{mod}}^\lambda(\xi; u), \]
\[ \|e_2^\lambda(\xi; z)\|_V \lesssim e_{\text{mod}}^u(\xi; z), \]
\[ \|e_{\text{obj}}^u(\xi; z)\|_V \lesssim e_{\text{obj}}^u(\xi; z). \]

Now, we address the key accuracy requirements in the inexact TR framework that enables the usage of local RB surrogates. To this end, let \( u(\xi) = S(\xi; z_k), u_j = u(\xi_j), \)
\( \hat{u}(\xi) = S_{\text{mod}}(\xi; z_k), \hat{u}_j = \hat{u}(\xi_j), \lambda(\xi) = \Lambda(\xi; u(\xi)), \lambda_j = \lambda(\xi_j), \hat{\lambda}(\xi) = \Lambda_{\text{mod}}(\xi; \hat{u}(\xi)) \) and
\( \hat{\lambda}_j = \hat{\lambda}(\xi_j) \). We assume that the optimization variable, i.e., the control \( z_k \), satisfies \( \|z_k\|_Z \leq C_Z \) for some \( C_Z > 0 \) for all iterations \( k \). This condition translates into the boundedness of \( u_j, \nabla_u G(u_j, \xi_j) \) and \( \lambda_j \) in their respective norms for all \( \xi_j \in \Xi \) and for all iterations \( k \) by Assumption 1, 5 and 6.

Using the explicit form of the gradient (4.5), the approximate gradient (4.14) and the triangle inequality, we arrive at the bounds
\[
\|\nabla_z \hat{J}(w_k) - \nabla_z m_k(0)\|_Z \leq \sum_{j=1}^{N} p_j \|v'(G(u_j, \xi_j) - t_k)B(\xi_j)^*\lambda_j - v'(G(\hat{u}_j, \xi_j) - t_k)B(\xi_j)^*\hat{\lambda}_j\|_Z 
\]

and
\[
|\nabla_t \hat{J}(w_k) - \nabla_t m_k(0)| \leq \sum_{j=1}^{N} p_j |v'(G(u_j, \xi_j) - t_k) - v'(G(\hat{u}_j, \xi_j) - t_k)|. \]

We then bound the individual samples on the right hand side of (4.21) by
\[
\|B(\xi_j)^*(v'(G(u_j, \xi_j) - t_k)\lambda_j - v'(G(\hat{u}_j, \xi_j) - t_k)\hat{\lambda}_j)\|_Z 
\]
\[
\leq \kappa_3 \|v'(G(u_j, \xi_j) - t_k) - v'(G(\hat{u}_j, \xi_j) - t_k)\| \|\lambda_j\|_V + \kappa_3 \|v'(G(\hat{u}_j, \xi_j) - t_k)\| \|\hat{\lambda}_j - \lambda_j\|_V. \]
Using Assumption 6 and the integral mean value theorem, we bound the first term on the right-hand-side of (4.23) and the right-hand-side of (4.22) as (the bound is contributed by Drew Kouri)

\[
\left| v'(G(u_j, \xi_j) - t_k) - v'(G(\hat{u}_j, \xi_j) - t_k) \right| \leq K_v |G(u_j, \xi_j) - G(\hat{u}_j, \xi_j)|
\]

\[
= K_v \left| \int_0^1 \langle \nabla_u G(\hat{u}_j + t(u_j - \hat{u}_j), \xi_j), (u_j - \hat{u}_j) \rangle_U \, dt \right|
\]

\[
\leq \frac{1}{2} K_v K_G \| u_j - \hat{u}_j \|_U^2 + K_v \| \nabla_u G(\hat{u}_j, \xi_j) \|_U \| u_j - \hat{u}_j \|_U
\]

\[
= \frac{1}{2} K_v K_G \| e_{mod}^u(\xi; z) \|_U^2 + K_v \| \nabla_u G(\hat{u}_j, \xi_j) \|_U \| e_{mod}^u(\xi; z) \|_U
\]

\[
\leq \epsilon_{mod}^u(\xi_j, z_k)^2 + \| \nabla_u G(\hat{u}_j, \xi_j) \|_U \epsilon_{mod}^u(\xi_j; z_k).
\]

Now, let \( \lambda_j^0 = \Lambda(\xi_j; \hat{u}_j) \), then we can bound the second term on the right-hand-side of (4.23) using

\[
\| \tilde{\lambda}_j - \lambda_j \|_V \leq \| \tilde{\lambda}_j - \lambda_j^0 \|_V + \| \lambda_j^0 - \lambda_j \|_V = \| e_1^\lambda(\xi; z) \|_V + \| e_2^\lambda(\xi; z) \|_V \leq \epsilon_{mod}^\lambda(\xi_j, \hat{u}_j) + \epsilon_{mod}^u(\xi_j, z_k).
\]

Combining these bounds, we obtain the gradient error bound

\[
\| \nabla J(w_k) - \nabla m_k(0) \|_W \leq E_{mod}(w_k) := \sum_{j=1}^N p_j \delta_{mod}(\xi_j; w_k),
\]

where

\[
\delta_{mod}(\xi_j; w_k) := \left| v'(G(\hat{u}_j, \xi_j) - t_k) \right| (\epsilon_{mod}^\lambda(\xi_j, \hat{u}_j) + \epsilon_{mod}^u(\xi_j, z_k))
\]

\[
+ \epsilon_{mod}^u(\xi_j, z_k)^2 + \| \nabla_u G(\hat{u}_j, \xi_j) \|_U \epsilon_{mod}^u(\xi_j; z_k).
\]

Hence as long as \( |v'(G(\hat{u}_j, \xi_j) - t_k)| \) and \( \| \nabla_u G(\hat{u}_j, \xi_j) \|_U \) are bounded for all iterations \( k \), we can satisfy (4.9) by refining our local reduced basis models \( S_{mod} \) and \( \Lambda_{mod} \).

The error indicator \( E_{mod}(z_k) \) is only small provided we approximate both forward and adjoint solutions accurately. We therefore modify the local RB method using a new set of local bases that is composed of both forward and adjoint solutions as well as their gradients.
The mixed bases are shared by $S_{\text{mod}}$ and $\Lambda_{\text{mod}}$. In fact, only one surrogate model with such mixed bases needs to be built for $m_k$. This surrogate model, which we denote by $S_{\text{mod}}\Lambda_{\text{mod}}$, is interpreted as $S_{\text{mod}}$ when used for forward solves and as $\Lambda_{\text{mod}}$ for adjoint solves. Note that even though $m_k$ changes at each TR step, we can employ the same $S_{\text{mod}}\Lambda_{\text{mod}}$ to obtain $m_k$ for all steps, which means that $S_{\text{mod}}\Lambda_{\text{mod}}$ can have a set of bases composed of solutions under different $z_k$. The error indicator will guide the refinement of $S_{\text{mod}}\Lambda_{\text{mod}}$ at each step so that (4.9) is always satisfied. The recycling of $S_{\text{mod}}\Lambda_{\text{mod}}$ from previous TR steps reduces the computational cost substantially since we do not need to rebuild an $S_{\text{mod}}\Lambda_{\text{mod}}$ whenever $z_k$ is updated in the optimization process. The complete algorithm for constructing $S_{\text{mod}}\Lambda_{\text{mod}}$ is listed in Algorithm 3.

Similar to the TR subproblem model $m_k$, we approximate the objective function by

$$\hat{J}_k(w) = \left(t + \sum_{j=1}^{N} p_j v(G(S_{\text{obj}}(\xi_j;z),\xi_j) - t)\right) + \psi(z)$$  \hspace{1cm} (4.26)$$

By similar arguments as above, i.e., using Assumption 6 and applications of the integral mean value theorem, we arrive at the bound

$$|\hat{J}_k(w) - \hat{J}(w)| \leq E_{\text{obj}}(w) := \sum_{j=1}^{N} p_j \delta_{\text{obj}}(\xi_j;w)$$ \hspace{1cm} (4.27)$$

where

$$\delta_{\text{obj}}(\xi_j;w) := \epsilon_{\text{obj}}^u(\xi_j;z)^2 + |v'(G(S_{\text{obj}}(\xi_j;z))) - t)|\epsilon_{\text{obj}}^u(\xi_j;z).$$ \hspace{1cm} (4.28)$$

Therefore, we can bound the difference between the actual and computed reduction by

$$|\text{ared}_k - \text{cred}_k| \leq |\hat{J}_k(w_k) - \hat{J}(w_k)| + |\hat{J}_k(w_k + s_k) - \hat{J}(w_k + s_k)|$$

$$\leq E_{\text{obj}}(w_k) + E_{\text{obj}}(w_k + s_k).$$ \hspace{1cm} (4.29)$$

Again, as long as $|v'(G(S_{\text{obj}}(\xi_j;z_k),\xi_j) - t_k)|$ and $|v'(G(S_{\text{obj}}(\xi_j;z_k + \xi_k),\xi_j) - (t_k + \tau_k))|$ are bounded for all iterates $w_k = (t_k, z_k)$ and steps $s_k = (\tau_k, \xi_k)$, we can satisfy (4.11) by refining our local RB model $S_{\text{obj}}$. 79
Algorithm 3: Adaptive algorithm to build $S_{\Lambda_{\text{mod}}}$ for $m_k$

If $k = 0$, model initialization:

- Let $\xi_0 = \mathbb{E}[^1\xi]$ and build an initial surrogate model $S_{\Lambda_{\text{mod}}}$ based on the solution $S(\xi_0; z_0)$, its gradient $\nabla_\xi S(\xi_0; z_0)$, the adjoint solution $\Lambda(\xi_0; S(\xi_0; z_0))$ and its gradient $\nabla_\xi \Lambda(\xi_0; S(\xi_0; z_0))$.

Model refinement:

Given $w_k$, $\Delta_k$, and $S_{\Lambda_{\text{mod}}}$, which is recycled from step $k - 1$,

- Build $m_k$ with $S_{\Lambda_{\text{mod}}}$, evaluate $\delta_{\text{mod}}(\xi_j; w_k)$ for all $j = 1, \ldots, N$, compute $E_{\text{mod}}(w_k)$ and $\|\nabla m_k(0)\|_Z$.

- While $E_{\text{mod}}(w_k) > \kappa \min\{\|\nabla m_k(0)\|_Z, \Delta_k\}$, do
  - Select $\xi_{\text{max}} = \arg\max_{j=1,\ldots,N} p_j \delta_{\text{mod}}(\xi_j; w_k)$.
  - Compute $S(\xi_{\text{max}}; z_k)$, $\Lambda(\xi_{\text{max}}; S(\xi_{\text{max}}; z_k))$, $\nabla_\xi S(\xi_{\text{max}}; z_k)$ and $\nabla_\xi \Lambda(\xi_{\text{max}}; S(\xi_{\text{max}}; z_k))$.
  - Incorporate the new information at $\xi_{\text{max}}$ into $S_{\Lambda_{\text{mod}}}$ using the local RB method.
  - Update $m_k$ with $S_{\Lambda_{\text{mod}}}$.
  - Update $\delta_{\text{mod}}(\xi_j; w_k)$ for $j = 1, \ldots, N$ and $E_{\text{mod}}(w_k)$, and recompute $\|\nabla m_k(0)\|_Z$.

End

Return $S_{\Lambda_{\text{mod}}}$.

The objective error bound (4.29) requires us to accurately approximate the solution to (4.1) at both $w_k$ and $w_k + s_k$. This motivates us to use a local basis that is composed of solutions and gradients at both $w_k$ and $w_k + s_k$. We employ a similar recycling scheme as
above to save computations while refining $S_{obj}$. The complete algorithm for constructing $S_{obj}$ is listed in Algorithm 4.

Algorithm 4: Adaptive algorithm to build $S_{obj}$ for $\hat{J}_k$

If $k = 0$, model initialization:

- Let $\xi_0 = \mathbb{E}[\xi]$ and build the initial surrogate model $S_{mod}$ based on the solution $S(\xi_0, z_0)$, the gradient $\nabla_\xi S(\xi_0, z_0)$.

Else, model refinement:

Given $w_k, s_k, \text{pred}_k, r_k$ and $S_{obj}$, which is recycled from step $k - 1$,

- Compute $\gamma_k = K (\eta \min \{\text{pred}_k, r_k\})^{\frac{1}{2}}$, evaluate $\delta_{obj}(\xi_j; w_k), \delta_{obj}(\xi_j; w_k + s_k)$ for $j = 1, \ldots, N$ and compute $E_{obj}(w_k)$ and $E_{obj}(w_k + s_k)$.

- While $E_{obj}(w_k) + E_{obj}(w_k + s_k) > \gamma_k$, do
  - Select $\xi_{max} = \arg \max_{j=1,\ldots,N} p_j (\delta_{obj}(\xi_j; w_k) + \delta_{obj}(\xi_j; w_k + s_k))$.
  - Compute $S(\xi_{max}, z_k), \nabla_\xi S(\xi_{max}, z_k), S(\xi_{max}, z_k + \zeta_k)$, and $\nabla_\xi S(\xi_{max}, z_k + \zeta_k)$.
  - Incorporate the new information at $\xi_{max}$ into $S_{obj}$ using the local RB method.
  - Update $\delta_{mod}(\xi_j; w_k)$ and $\delta_{obj}(\xi_j; w_k + s_k)$ for $j = 1, \ldots, N$, update $E_{obj}(w_k)$ and $E_{obj}(w_k + s_k)$.

End

End

Return $S_{obj}$. 81
4.3 Examples

In this section, we present three numerical examples to showcase the capability of the proposed method in terms of computational efficiency to solve risk-averse PDE constrained optimization problems. In particular, we use the Monte Carlo method with full PDE solves as reference and show the reduction of computational cost that we can achieve by using our method. All reference solutions are obtained using a quasi-newton method with BFGS Hessian update [NW06]. We show that we can achieve risk-averse solutions of the same quality with tremendous amount of computational savings.

4.3.1 1D Helmholtz equation with damped resonance

We first consider the stochastic Helmholtz problem in Section 2.6.2 with controls. We present the problem here again for the sake of completeness. Let $D = (0, 1)$, the problem is defined as

$$-rac{\partial}{\partial x} \left( \nu(x, \omega) \frac{\partial u(x, \omega)}{\partial x} \right) - ic\tau u(x, \omega) - \tau^2 u(x, \omega) = z(x), \quad x \in D, \text{ a.s.} \quad (4.30a)$$

$$u(0, \omega) = u(1, \omega) = 0, \quad \text{a.s.} \quad (4.30b)$$

where $\nu(x, \omega)$ is a piecewise constant random field, $c$ is a deterministic damping factor, $\tau$ is a deterministic angular frequency, and $z$ is a real-valued deterministic control.

We model the random field $\nu(x, \omega)$ as

$$\nu(x, \omega) = [0.3 + 1.2\xi_1(\omega)]1_{[0,0.5]}(x) + [0.5 + 3.5\xi_2(\omega)]1_{[0.5,1]}(x) \quad (4.31)$$

where $\xi_1 \sim \text{Beta}(1,3)$ and $\xi_2 \sim \text{Beta}(3,2)$ with correlation equal to 0.5. We set the angular frequency to $\tau = 4\pi$. These parameters are chosen so that damped resonances occur for certain realizations of the random field $\nu(x, \omega)$ if the control is trivially prescribed. The $L_2(D)$ norm of the solution $\|u(\cdot, \xi)\|_{L_2(D)}$ over the entire parameter domain $\Xi$ when $z(x) = 1$ with two different values of damping constants $c = 0.01$ and $c = 0.2$ are shown in Figure 2.7. From Figure 2.7, it is clear that resonance occurs in three different regions of $\Xi$. 

82
It is well known that as $c$ decreases towards zero, the solution norm at resonances increases unboundedly. This behavior makes Helmholtz-type problems particularly challenging for constructing surrogate models in the parameter space. For this example, we fix $c = 0.2$, which is shown in Figure 2.7(b).

We define the general objective function as

$$G(u; \xi) = \|Du - d\|_Y^2$$

(4.32)

where $Y$ is a Hilbert space, $D \in L(U, Y)$ is an observation operator, and $d \in Y$ is a desired profile. For this problem, we set $Y = \mathbb{R}^5$ and $D$ to evaluate the PDE solution $u$ at 5 equally spaced points $x \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. Note that since $D \subset \mathbb{R}$, $U$ is continuously embedded into $C(D)$ [AF03] (i.e., $u \in U$ is a continuous function) and thus $D$ is a bounded linear operator. The target vector $d$ is simply a vector of 1’s, that is, we want to apply a control so that the solution to (4.30a) is close to a deterministic constant 1.

The control is assumed to be piecewise constants, i.e., $z(x) = \sum_{i=1}^{10} z_i \mathbb{1}_{I_i}(x)$ where $I_i = (0.1(i-1), 0.1i)$, $i = 1, \ldots, 10$. and the control penalty is

$$\varphi(z) = w_p \|z\|_{L_2}^2$$

with $w_p = 1e - 4$.

For this problem, we consider the family of risk measures in (3.5) which are convex combinations of the expectation and the smoothed CVaR at confidence level $\beta = 0.9$ [KS16]. We use the following algorithmic parameters: $\eta_1 = 0.05$, $\eta_2 = 0.75$, $\gamma = 0.5$, $\omega = 0.75$, $\mu = 10$, $K = 2$ and $\Delta_0 = 3$. We employed $N = 20,000$ Monte Carlo samples to approximate $\xi$. We used 6 neighbors to construct the local bases $\Phi_k$.

We first consider the case where $\alpha = 1.0$ in (3.5). In this case, $\mathcal{R}(X) = \text{CVaR}_\beta[X]$. In Figure 4.1 we depict the optimal controls (left image) and the cumulative distribution functions (CDFs) with the 0.9-quantile and 0.9-CVaR of $G(u(\xi), \xi)$ (right image) obtained by the TR method (red) and by solving the full optimization problem using Monte Carlo (blue).
Figure 4.1: The left image provides a comparison of the optimal controls computed by the proposed TR/RB method (red) and those computed by solving the full-order Monte Carlo reference problem (blue). The right image depicts the CDF of $G(u(\xi), \xi)$ evaluated under optimal controls from the TR/RB method (red) and the full-order Monte Carlo problem (blue). The dashed vertical lines correspond to the 0.9-quantile whereas the solid vertical lines correspond to the 0.9-CVaR. These lines (blue for the reference solution and red for the TR/RB solution) are virtually indistinguishable.

We solved the nonlinear subproblem model exactly in (4.15) for each TR step. The algorithm terminated due to negligible objective reduction after only 4 successful TR steps. In Figure 4.2, we plot the norm of the true gradient at each iteration of the TR algorithm. Notice that the norm of true gradient drops almost three orders of magnitude, indicating that the algorithm makes significant progress towards satisfying the necessary optimality conditions.

In Figure 4.3, we show the atoms of the surrogate models $S_{\text{mod}}$ (red) and $S_{\text{obj}}$ (black) which we used to construct $m_k$ and $\hat{J}_k$ respectively during each TR step with the 20,000 Monte Carlo samples in the background. At each atom, the surrogate model construction required six full PDE solves for the entire optimization process. Due to the recycling of $S_{\text{mod}}$ and $S_{\text{obj}}$, same atoms can be used across different TR steps without the need to rebuild the models from scratch, resulting in additional computational savings. Also note that atoms tend to cluster around the resonance regions shown in Figure 2.7, which is due

84
Figure 4.2: The norm of the true gradient evaluated at the iterates produced by our proposed TR/RB approach.

to the large error associated with resonant solutions that required more local refinement to satisfy the accuracy requirements of the TR algorithm.

Figure 4.3: The atoms of the surrogate models $S_{mod}$ (black) and $S_{obj}$ (red) that are used to build the models $m_k$ and $\hat{J}_k$ respectively during each TR step. The blue dots are the 20,000 Monte Carlo samples used to approximate $\xi$.

We compare the total number of full PDE solves required for the reference Monte Carlo solution and the total number of full PDE solves required for our TR/RB method in Figure 4.4. Solving the full optimization problem with Monte Carlo results in over 5,000,000 full PDE evaluations. In comparison, the total number of full PDE evaluations for our TR/RB approach is less than 3,000.

We next perform a parametric study using a range of values of $\alpha$ in the risk measure (3.5). $\alpha$ is set to take five different values in $\{0.0, 0.2, 0.5, 0.8, 1.0\}$ where we go from minimizing the expectation to minimizing the full CVaR. The optimal controls with different
Figure 4.4: A comparison of the total number of full PDE solves for the reference Monte Carlo approach and our proposed TR/RB approach.

α computed by both reference Monte Carlo method and our method are shown in Figure 4.5. The number of full PDE solves induced in all cases are shown in Figure 4.6. For all values of α, the control obtained by our TR/RB approach is very close to the reference control. However, significant computational savings exceeding three orders of magnitude can be achieved using TR/RB approach.

Figure 4.5: The optimal controls with different α computed by the reference Monte Carlo method and our proposed TR/RB approach. For all values of α, the control obtained by TR/RB approach is very close to the reference control.

In Figure 4.7, we show the CDFs and statistics, i.e., mean and standard deviation, of $G(u(\xi), \xi)$ under the optimal controls with different values of α. As can be seen, when
\section*{Figure 4.6:} A comparison of the total number of full PDE solves for the reference Monte Carlo approach and our proposed TR/RB approach, with different values of $\alpha$.

$\alpha = 0.0$, that is when we are risk-neutral, though the expected loss $\mathbb{E}[G(u(\xi), \xi)]$ is minimal, the CDF of $G(u(\xi), \xi)$ has a heavy tail extending to values over 8. As we increase $\alpha$, though the $\mathbb{E}[G(u(\xi), \xi)]$ is increasing, we achieve more and more compact CDFs with smaller deviations, indicating less uncertainties associated with the controlled QoI.

Finally, we show the profiles of $G(u(\xi), \xi)$ under the optimal controls when $\alpha = 0.0$ and $\alpha = 1.0$ over the parameter space in Figure 4.8. We also show the parameter samples that contribute to the tail risk CVaR$_{0.9}$ in Figure 4.9. In Figure 4.8, it can be seen that when $\alpha = 0.0$, i.e., risk-neutral, $G(u(\xi), \xi)$ has large values over the resonance region in the top-right corner. The samples at the same location, see Figure 4.9, are the main contributors to the tail risk. However, under risk-averse control with $\alpha = 1.0$, the resonance in the top-right corner is properly mitigated. The tail risk now comes from the samples in the bottom-left corner of the parameter domain with smaller values of $G(u(\xi), \xi)$, leading to smaller tail risk CVaR$_{0.9}$. Though stripes of resonance can still be seen in the bottom-left corner of the parameter domain under the risk-averse control, these areas, in comparison to resonance region in the top-right corner, have extremely low probability measure and their effect for CVaR$_{0.9}$ is negligible and is not captured by our sample approximation (see the samples in Figure 4.3.)
Figure 4.7: The left image provides a comparison of the CDFs of $G(u(\xi), \xi)$ under the optimal controls computed by the proposed TR/RB method with different values of $\alpha$. The right image shows the mean and standard deviation of $G(u(\xi), \xi)$ under the optimal controls computed by the proposed TR/RB method with different values of $\alpha$. As $\alpha$ increases, $E[G(u(\xi), \xi)]$ increases as well, however, the CDF becomes more compact with decreasing standard deviation, indicating less uncertainties associated with the controlled QoI.

Figure 4.8: The QoI $G(u(\xi), \xi)$ evaluated under the optimal controls when $\alpha = 0.0$ and $\alpha = 1.0$.

4.3.2 1D advection diffusion equation

In the second example, we study the same advection-diffusion problem in Section 2.6.1 with controls. We consider different risk measures including CVaR and entropic risk (see (3.4)). For the purpose of comparison, we also consider the risk-neutral optimization problem with $E$ in place of $R$ in (4.3).
Figure 4.9: The parameter samples that contribute to the tail risk $\text{CVaR}_{0.9}[G(u(\xi), \xi)]$ under the optimal controls when $\alpha = 0.0$ and $\alpha = 1.0$.

Let $D = (0, 1)$ and the problem of interest is defined as

$$
-\nu \frac{\partial^2 u(x, \xi)}{\partial x^2} + b(x, \xi) \frac{\partial u(x, \xi)}{\partial x} = z(x), \quad x \in X, \text{ a.s.} \tag{4.33a}
$$

$$
u \frac{\partial^2 u(x, \xi)}{\partial x^2} + b(x, \xi) \frac{\partial u(x, \xi)}{\partial x} = z(x), \quad x \in X, \text{ a.s.} \tag{4.33b}
$$

The diffusivity, $\nu$, and control, $z$, are deterministic whereas the advection velocity, $b$, is a piecewise constant random field given by

$$
b(x, \xi) = [b_1 + \xi_1] \mathbb{1}_{[0,0.5)}(x) + [b_2 + \xi_2] \mathbb{1}_{[0.5,1]}(x) \tag{4.34}
$$

Here, $b_1 = 0.5$, $b_2 = 0.8$, $\xi_1$ and $\xi_2$ are independent random variables that are uniformly distributed on the interval $[-1, 1]$.

We consider the following QoI,

$$
G(u(\xi); \xi) = \|u(\xi) - \sin(\pi x)\|_{L^2(D)}^2, \tag{4.35}
$$

which means we want make the solution to (4.33a) match a target sinusoidal function in $L^2(D)$. We try to minimize $\mathcal{R}(G)$ for two different risk measure: (1) CVaR with $\beta = 0.9$, and (2) the entropic risk with $c = 100$. The control is described as piecewise constants, i.e.,

$$
z(x) = \sum_{i=1}^{10} z_i \mathbb{1}_{I_i}(x) \text{ where } I_i = (0.1(i - 1), 0.1i), \quad i = 1, \ldots, 10, \text{ and we apply the following control penalty}
$$

$$
\varphi(z) = w_p \|z\|_{L^2}^2
$$
with \( w_p = 1e-4 \). We used the following TR parameters: \( \eta_1 = 0.05, \eta_2 = 0.75, \gamma = 0.5, \omega = 0.75, \mu = 3, K = 3 \) and \( \Delta_0 = 3 \). We employed \( N = 10,000 \) Monte Carlo samples to approximate \( \xi \). We used 3 neighbors to construct the local bases \( \Phi_k \).

We first show the optimal controls obtained by minimizing expectation, CVaR and the entropic risk in Figure 4.10, and compare them with the reference solutions obtained using Monte Carlo. For all three different objectives, the controls are virtually indistinguishable to the reference. In addition, for the parameters used here, minimizing CVaR and the entropic risk yield similar controls that have larger values close to the left boundary of \( D \). Due to the setup of the PDE, controls applied close to the left boundary are most influential to the PDE solution because the advection velocity \( b \) will be positive on \( D \) with the majority of the probability mass. Risk-averse controls take advantage of this underlying property and exert higher control efforts on the left boundary.

![Figure 4.10: Optimal controls with reference obtained by minimizing expectation, CVaR and entropic risk applied to \( G(u(\xi), \xi) \).](image)

In Figure 4.11 and 4.12, we show the profile of \( G(u(\xi), \xi) \) over the parameter space and CDFs of \( G(u(\xi), \xi) \) under the optimal controls. From the figures, it can be seen that risk-averse controls obtained by minimizing CVaR and entropic risk have much better effects against large values of \( G(u(\xi), \xi) \). The \( G(u(\xi), \xi) \) under the the risk-neutral control, on the other hand, suffers from a heavy tail with large values occurring in the top-left corner.
of the parameter space. Minimizing CVaR and entropic risk, in this case, leads to similar performance.

Figure 4.11: The QoI $G(u(\xi), \xi)$ evaluated under the optimal controls.

Figure 4.12: The CDFs of $G(u(\xi), \xi)$ evaluated under the optimal controls.

We now examine in more detail the performance of the TR/RB algorithm in this problem. For all three different objective, the algorithm converges in 5 or fewer TR steps. The norms of the true gradients after each TR iteration are shown in Figure 4.13. In all cases, the norm of the true gradient drops by 4 orders of magnitude at the end of the algorithm, indicating good progress towards the true minimizer.

The final atoms of the surrogate models $S_{\text{mod}}$ (red) and $S_{\text{obj}}$ (black) which were used to construct $m_k$ and $\hat{J}_k$ are shown in Figure 4.14 for all three different objectives. It can be seen that in risk-averse cases (CVaR and entropic risk), the atoms tend to cluster more on the top-left corner of the parameter space where the large values of $G(u(\xi), \xi)$ occur. This is
Figure 4.13: The norm of the true gradient evaluated at the iterates produced by our proposed TR/RB approach.

especially true for CVaR which only required a few atoms to construct the surrogates that meet the accuracy requirements of the TR algorithm. The reason is that for CVaR where $v(x) = (1 - \beta)^{-1}[x]^{+}$, the gradient $v'(G - t_k)$, which is part of the error indicators $\delta_{\text{mod}} (4.25)$ and $\delta_{\text{obj}} (4.28)$, vanishes for majority (90% in this case) of the samples of $G$. These samples hence contribute less to the objective and gradient errors. Therefore, the local RB models tend to choose atoms in locations where $G$ is large, leading to more efficient approximation of the objective.

Figure 4.14: The final atoms of the surrogate models $S_{\text{mod}}$ (red) and $S_{\text{obj}}$ (black) which were used to construct $m_k$ and $\hat{J}_k$.

Finally, we compare the number of full PDE PDE solves induced by the reference solutions and our TR/RB solutions in Figure 4.15. Again, significant computational savings, up to 3 orders of magnitudes, are realized by the TR/RB method, which yields optimal
controls similar to those obtained by the Monte Carlo for all three different objectives.

**Figure 4.15:** A comparison of the total number of full PDE solves for the reference Monte Carlo approach and our proposed TR/RB approach.

### 4.3.3 2D Advection-diffusion equation

In the last example, we consider a more complicated 2D advection-diffusion equation with higher stochastic dimensions. Let $D = (0, 1)^2$, and consider the following problem,

\[-\nabla \cdot (\kappa(x, \omega) \nabla u(x, \omega)) + v(x, \omega) \cdot \nabla u(x, \omega) = f(x, \omega) - z(x) \quad x \in D \text{ a.s.} \quad (4.36a)\]

\[u(x, \omega) = 0 \quad x \in \Gamma_d \text{ a.s.} \quad (4.36b)\]

\[\kappa(x, \omega) \nabla u(x, \omega) \cdot n = 0 \quad x \in \Gamma_n \text{ a.s.} \quad (4.36c)\]

where $\Gamma_d := [0, 1] \times \{0\}$ and $\Gamma_n := \partial D \setminus \Gamma_d$. The uncertainty in this example arises from diffusivity, advection and the source term $f(x, \omega)$. This problem models, for example, the transport of a contaminant in some random medium. The goal is to determine how to inject chemicals that dissolve the contaminant and thus minimize its total concentration.

The diffusion coefficient $\kappa$ is modeled as a random field using the approach in [Gri14]. In particular, we characterize the diffusion coefficient as

\[\kappa(x, \omega) = \sum_{i=1}^{2} \sum_{j=1}^{2} \phi_i(x_1) \phi_j(x_2) \xi_{2(i-1)+j}(\omega)\]

\[\xi_k = 0.1 + 0.4\beta_k \quad k = 1, 2, 3, 4\]

93
where $\phi_i(x) = B_{i,2}(x)$ are Bernstein’s polynomials of order 2 and $\beta_k \sim \text{Beta}(2, 5)$.

The advection field is divergence free and is parametrized by two uniform random variables as
\[
v(x, \omega) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \xi_5 + \begin{pmatrix} -x_1 \\ x_2 \end{pmatrix} \xi_6
\]
\[
\xi_5 \sim U[10, 20] \quad \xi_6 \sim U[0, 10].
\]

Finally, the forcing term $f$ is modeled by a Gaussian function with uncertain magnitude and location
\[
f(x, \omega) = \xi_7 \exp \left( -\frac{(x_1 - \xi_8)^2}{\sigma_1^2} + \frac{-(x_2 - \xi_9)^2}{\sigma_2^2} \right),
\]
where $\sigma_1 = 0.3, \sigma_2 = 0.2$. The forcing magnitude and location are chosen to be uniformly distributed as
\[
\xi_7 \sim U[0, 5] \quad \xi_8 \sim U[0.2, 0.6] \quad \xi_9 \sim U[0.1, 0.8].
\]

The control is modeled by five Gaussian functions located at $[0.25, 0.25], [0.25, 0.75], [0.75, 0.25], [0.75, 0.75]$ and $[0.5, 0.5]$, i.e.,
\[
z(x) = \sum_{i=1}^{5} z_i \exp \left( -\frac{(x_1 - c_i)^2 - (x_2 - d_i)^2}{\sigma^2} \right). \tag{4.39}
\]
where $c_i \in \{0.25, 0.25, 0.75, 0.75, 0.5\}, d_i \in \{0.25, 0.75, 0.25, 0.75, 0.5\}$ and $\sigma = 0.1$.

The goal is to control the the following quantity
\[
G(u(\xi); \xi) = \|u(x, \xi)\|_{L_2(D)}^2 \tag{4.40}
\]
and to minimize the associated risk measure $R(G)$ without control penalty.

For this problem, we only consider CVaR with $\beta = 0.95$ as the risk measure and compare the effect of risk-averse control with the risk-neutral control. The algorithmic parameters for this problem are: $\eta_1 = 0.05, \eta_2 = 0.75, \gamma = 0.5, \omega = 0.75, \mu = 10, K = 10$ and $\Delta_0 = 3$. We employed $N = 20,000$ Monte Carlo samples to approximate $\xi$. Finally, we used 20
neighbors to construct the local bases $\Phi_k$, due to the high dimensionality of the parameter space.

We first compare the risk-neutral control and the risk-averse control obtained by the TR/RB method with the reference. The results are shown in Figure 4.16. The relative error of the controls measured by $\|\hat{z} - z\|_{L_2}/\|z\|_{L_2}$ is 2.01% for the risk-neutral case and 0.87% for the risk-averse case, respectively. It is interesting to note that the risk-averse control at the location $[0.25, 0.25]$ has a much higher intensity than the risk-neutral control at the same location. In addition, to compensate for the strong intensity at $[0.25, 0.25]$, a slight negative control is applied at location $[0.75, 0.25]$. Due to the setup of the PDE, particularly the directions of the advection velocities (rightwards and upwards), the control at location $[0.25, 0.25]$ is most influential to the PDE solution over $D$. Risk averse control exerts higher control effort at this location, which is similar to what we have observed for the previous example.

The CDFs of $G(u(\xi); \xi)$ under the optimal controls are shown in Figure 4.17. The two controls result in notable difference of the tail risk of $G(u(\xi); \xi)$, which is measured by the CVaR with $\beta = 0.95$. In particular, the risk-neutral control has a CVaR$_\beta$ of 0.0303 and the risk-averse control has a CVaR$_\beta$ of 0.0248.

Finally, we show the norm of the true gradient throughout the TR iterations in Figure 4.18 and the number of full PDE solves in Figure 4.19. Again, we see rapid convergence with significant computational savings by the proposed TR/RB method. The number of full PDE solves incurred by the TR/RB method is more than 2 orders of magnitude smaller than Monte Carlo. Such efficiency of our proposed TR/RB method allows us to obtain near-optimal risk-averse solutions with a much lower computational cost.
Figure 4.16: The controls obtained by the TR/RB method with the reference.

Figure 4.17: The CDFs of $G(u(\xi); \xi)$ under the optimal controls.
Figure 4.18: The norm of the true gradient evaluated at the iterates produced by our proposed TR/RB approach.

Figure 4.19: A comparison of the total number of full PDE solves for the reference Monte Carlo approach and our proposed TR/RB approach.
Chapter 5

Particle-based Approximation of the Gibbs Posterior for Inverse Problems

In this chapter, we present a particle-based method with local RB surrogate models to approximate the Gibbs posterior. We intent to develop a general method for updating belief distributions for inverse problems governed by PDEs by exploiting the accuracy and computational efficiency of local RB surrogate models. Different from the usual Bayesian methods where the distributions of the noise are assumed to be known exactly (typically assumed as i.i.d. Gaussian), the Gibbs posterior update does not require a likelihood function thus no exact model for the noise is needed. Instead, the Gibbs posterior are applicable where the unknown parameters are connected to the data through a loss function [BHW16]. It provides a more general framework for updating belief distributions where the true data generating mechanism is unknown or difficult to specify.

Though an explicit likelihood function is not required, an important aspect of Gibbs posterior is the choice of a scaling parameter (or weight) for the loss. Given a loss function and a prior distribution, the weight uniquely determines the Gibbs posterior distribution. We will describe possible methods to assign such a weight. In addition, we employ a particle based discretization to approximate the continuous prior distribution, which after applying the Gibbs update results in an explicit formula for the posterior weights associated with each particle. We evolve the population of the particles following a Sequential Monte Carlo (SMC) approach recently developed in [KBJ14]. We propose a simple adaptive method to control the progress of the evolution of the particles based on a criteria of the sample degeneracy. To manage the cost of propagating increasing number of particles through the loss function, we employ the local RB method to build an efficient surrogate model that is used in the Gibbs update formula in place of the true loss. We show asymptotic convergence of our approximation to the true posterior distribution under appropriate assumptions.
Our method coincides with Bayesian updating strategy when a true likelihood function is known and the loss function corresponds to the negative log-likelihood, yet provides subjective inference in more general settings.

5.1 Gibbs Posterior for inverse problems

We assume we have a system (governed by a PDE) with unknown deterministic parameters \( \xi^* \in \Xi \) and that we can measure some noisy output \( d \in \mathbb{R}^D \) from the system, that is

\[
d = \mathcal{F}(\xi^*) + \epsilon,
\]

where \( \mathcal{F} : \Xi \to \mathbb{R}^D \) is a PDE model representing the system that maps each parameter to an observation, which means each evaluation of the map requires one PDE solve. We assume the PDE model satisfies Assumption 2 and allows an affine decomposition with respect the parameters \( \xi \). Finally, \( \epsilon \in \mathbb{R}^D \) is the random noise. We define \( d^* = \mathcal{F}(\xi^*) \) as the true data without the corruption due to noise.

We further assume we have a prior belief about \( \xi^* \) which can be expressed as a prior distribution \( \rho_0(\cdot) : \Xi \to \mathbb{R} \), and that we possibly have access to multiple measurements with independent realization of the noise, i.e., \( d_i = \mathcal{F}(\xi^*) + \epsilon_i \), where \( i \) is the index of the measurement. Without imposing strong assumptions or an explicit distribution model on \( \epsilon \), we would like to integrate the information contained in the data into our belief about \( \xi^* \).

In Gibbs posterior formulation, we do not need a likelihood function, instead, we have access to a given loss function \( l(\cdot, \cdot) : \Xi \times \mathbb{R}^D \to \mathbb{R} \). A loss function measures the discrepancy between the predicted output at a parameter and the true measurement, for example, one can use

\[
l(\xi, d) = \|\mathcal{F}(\xi) - d\|_{l_2}^2
\]

as the loss function. Unlike a likelihood function that requires exact knowledge of the data generating mechanism (or noise model), loss functions are typically easier to specify for inverse problems, which is the biggest advantage of Gibbs posterior. When a set of data points \( d_i \), \( i = 1, 2, \ldots, n \) are observed, we update our belief according to the following
optimization problem

$$\rho(\xi) = \arg\min_{\rho(\xi) \in P} \int_\Xi W \sum_{i=1}^n l(\xi, d_i) \rho(\xi) d\xi + D_{KL}(\rho\|\rho_0).$$  \hspace{1cm} (5.1)$$

where $W$ is a weight, or scaling factor, for the loss that is yet to be specified. For now, we assume $W$ is a fixed positive constant and will describe methods to prescribe $W$ later on. $D_{KL}(\rho\|\rho_0)$ is the Kullback-Leibler (KL) divergence between the posterior and the prior distributions. $P$ is the space of distributions of $\xi$, within which we search for the posterior. If we allow $P$ to contain all distributions over $\Xi$, we have an explicit update formula for $\rho(\xi)$ [BHW16], that is

$$\rho(\xi) = \frac{\exp\left(-W \sum_{i=1}^n l(\xi, d_i)\right) \rho_0(\xi)}{\int_\Xi \exp\left(-W \sum_{i=1}^n l(\xi, d_i)\right) \rho_0(\xi) d\xi}.$$  \hspace{1cm} (5.2)$$

This is a coherent update formula, in the sense that if we use the data one-by-one in a sequential manner results in the same posterior as when we use them all at once, as in (5.2). In addition, one can see from (5.1) that the Gibbs posterior minimizes the expected loss with an additional regularization term enforcing the posterior to be close to the prior in the sense of KL divergence. This suggest a way to pick the weight $W$ according to some balancing principle. We explore one possible choice of $W$ in Section 5.5.

Also, we can see that the usual Bayes rule based on likelihood function is a special case of Gibbs posterior using negative log-likelihood as the loss function with $W = 1$. Indeed, if we use $l(\xi, d_i) = -\log(\pi(d_i|\xi))$ where $\pi(d_i|\xi)$ is the likelihood function, we get

$$\rho(\xi) = \frac{\prod_{i=1}^n \pi(d_i|\xi) \rho_0(\xi)}{\int_\Xi \prod_{i=1}^n \pi(d_i|\xi) \rho_0(\xi) d\xi}$$

which is the conventional Bayes rule.

As integrating more data points into the Gibbs update requires nothing but a summation of the individual losses to form an accumulative loss $l(\xi, \{d_i\}_{i=1}^n) := \sum_{i=1}^n l(\xi, d_i)$, and the dependence of loss function on data is tangential to our presentation, in the following
content, we hence assume a generic loss function $l(\xi)$ for simplicity. In this case, the Gibbs update formula becomes

$$
\rho(\xi) = \frac{\exp(-Wl(\xi))\rho_0(\xi)}{\int_{\Xi} \exp(-Wl(\xi))\rho_0(\xi)d\xi}.
$$

(5.3)

### 5.2 Surrogate approximation

In the Gibbs update formula (5.3), evaluating $l(\xi)$ at each parameter $\xi$ requires one evaluation of the PDE model. The integral on the denominator, i.e., $\int_{\Xi} \exp(-Wl(\xi))\rho_0(\xi)d\xi$, is computationally intractable in general. To this end, Markov Chain Monte Carlo (MCMC) methods can be used to draw samples from the posterior without computing the integral. MCMC requires only pointwise evaluation of the loss function to generate a stream of samples from the posterior distribution that can be subsequently used to compute the statistics of the posterior. In MCMC, however, the generation of each new sample requires one or more evaluations of the PDE model. In addition, the mixing time of MCMC is typically unknown a priori, one needs to discard a large amount of samples in a so-called “burn-in” phase to be able to start sampling from the posterior, leading to additional computational cost. More recently, Sequential Monte Carlo (SMC) methods, or particle filters have also been applied in the setting of inverse problems by a few researchers [KBJ14, BJMS15]. The computational advantage of SMC for large-scale inverse problems has been demonstrated in [KBJ14]. However, the number of PDE evaluations can still be tremendous.

In this work, we intend to utilize the local RB method to build a cheap surrogate model $\tilde{l}(\xi)$ for the loss function $l(\xi)$. Using $\tilde{l}(\xi)$ for the Gibbs update in (5.3) results in an approximate Gibbs posterior that can be sampled (approximated) much more efficiently, as one can evaluate $\tilde{l}(\xi)$ as many times as needed with negligible cost. However, it is important to understand the error in such an approximation in order to build an efficient surrogate model with accuracy. To this end, we first define the approximate Gibbs posterior $\tilde{\rho}(\xi)$ as

$$
\tilde{\rho}(\xi) = \frac{\exp(-W\tilde{l}(\xi))\rho_0(\xi)}{\int_{\Xi} \exp(-W\tilde{l}(\xi))\rho_0(\xi)d\xi}.
$$

(5.4)
To quantify the error introduced by using the surrogate $\overline{l}(\xi)$ in (5.4), we derive a bound for the discrepancy between the approximate posterior $\overline{\rho}(\xi)$ and $\rho(\xi)$. To this end, we first state the following boundedness assumption on the loss function $l(\xi)$ and the surrogate $\overline{l}(\xi)$.

**Assumption 7.** We assume that

- $l(\xi)$ is nonnegative and is uniformly bounded from above: $\exists C_l > 0$ independent of $\xi \in \Xi$ such that for all $\xi \in \Xi$

  $$0 \leq l(\xi) \leq C_l. \quad (5.5)$$

- $\overline{l}(\xi)$ is nonnegative and is uniformly bounded from above: $\exists C_{\overline{l}} > 0$ independent of $\xi \in \Xi$ such that for all $\xi \in \Xi$

  $$0 \leq \overline{l}(\xi) \leq C_{\overline{l}}. \quad (5.6)$$

To measure the distance between probability distributions, we use the following metric

$$h(\rho_1, \rho_2) = \sup_{|f| \leq 1} \sqrt{\mathbb{E}|\rho_1[f] - \rho_2[f]|^2}$$

where $\rho_1, \rho_2 \in P$ are two possibly random elements in $P$, the supreme is over all $f : \Xi \to \mathbb{R}$ such that $\sup_{\xi \in \Xi} |f(\xi)| \leq 1$, $\rho[f] = \int_{\Xi} f(\xi) \rho(\xi) d\xi$. The expectation is with respect to the randomness of $\rho_1, \rho_2$. In case where $\rho_1$ is determined, and $\rho_2$ is an approximation to $\rho_1$ through a randomized algorithm, e.g., Monte Carlo, the expectation is with respect to the randomness of the algorithm. Note that $h$ is indeed a metric on $P$, in particular, it satisfies triangle inequality [RVH+15, BJMS15].

In addition, we define an $e$-feasible set, i.e.,

$$\Xi_e := \{\xi \in \Xi : |l(\xi) - \overline{l}(\xi)| \leq e\} \quad (5.7)$$

where $e > 0$ is some constant indicating the accuracy of the surrogate model $\overline{l}(\xi)$. We always assume that $e$ is small, e.g., $We \ll 1$. The set $\Xi_e$ contains all the parameters where the surrogate is accurate in the sense that the absolute difference between $l(\xi)$ and the $\overline{l}(\xi)$ is bounded by $e$. The complement of $\Xi_e$ is denoted by $\Xi_e^c := \Xi \setminus \Xi_e$. Now, we state the following theorem regarding the approximation accuracy of $\overline{\rho}(\xi)$. 


Theorem 4. Under Assumption 7, the following bound holds:

\[ h(\rho, \overline{\rho}) \leq 2 \exp(W C_l) C_e W e + 2 \exp(W C_l + W \max\{C_l, C_l^*\}) \rho[1_{\Xi_e}], \]

for some constants \( C_e > 0 \).

Proof. First, note that for \( \forall \xi \in \Xi_e \), we have that \( -e \leq l(\xi) - \overline{l}(\xi) \leq e \), for \( e \) sufficiently small, e.g., \( W e \ll 1 \), we have

\[ |\exp(-W l(\xi)) - \exp(-W \overline{l}(\xi))| \leq \exp(-W l(\xi))|1 - \exp(W l(\xi) - W \overline{l}(\xi))| \leq C_e W e \]

for some \( C_e > 0 \). Let

\[ Z_1 = \int_{\Xi} \exp(-W l(\xi)) \rho_0(\xi) d\xi, \quad Z_2 = \int_{\Xi} \exp(-W \overline{l}(\xi)) \rho_0(\xi) d\xi. \tag{5.8} \]

Using Assumption 7 and the fact that \( |f|_\infty \leq 1 \), we have

\[ Z_1 = \int_{\Xi} \exp(-W l(\xi)) \rho_1(\xi) d\xi \geq \exp(-W C_l), \]

\[ Z_2 \geq \int_{\Xi} \exp(-W \overline{l}(\xi)) \rho_2(\xi) |f(\xi)| d\xi. \]

In addition, we have

\[ |Z_1 - Z_2| \leq \int_{\Xi_e} |\exp(-W l(\xi)) - \exp(-W \overline{l}(\xi))| \rho_0(\xi) d\xi \]

\[ + \int_{\Xi_e^\pm} |1 - \exp(W l(\xi) - W \overline{l}(\xi))| \exp(-W l(\xi)) \rho_0(\xi) d\xi \]

\[ \leq C_e W e + \int_{\Xi_e^\pm} |1 - \exp(W l(\xi) - W \overline{l}(\xi))| Z_1 \rho(\xi) d\xi \]

\[ \leq C_e W e + \exp(W \max\{C_l, C_l^*\}) \rho[1_{\Xi_e^\pm}]. \]
Hence

\[
|\rho[f] - \overline{\rho}[f]| = \left| \frac{\int_{\Xi} \exp(-Wl(\xi))\rho_0(\xi)f(\xi)d\xi}{\int_{\Xi} \exp(-Wl(\xi))\rho_0(\xi)d\xi} - \frac{\int_{\Xi} \exp(-W\tilde{l}(\xi))\rho_0(\xi)f(\xi)d\xi}{\int_{\Xi} \exp(-W\tilde{l}(\xi))\rho_0(\xi)d\xi} \right|
\]

\[
\leq \frac{\int_{\Xi} |\exp(-Wl(\xi)) - \exp(-W\tilde{l}(\xi))|\rho_0(\xi)|f(\xi)|d\xi}{Z_1}
\]

\[
+ \frac{|Z_2 - Z_1|\int_{\Xi} \exp(-W\tilde{l}(\xi))\rho_2(\xi)|f(\xi)|d\xi}{Z_1 Z_2}
\]

\[
\leq \frac{C_e W e + \exp(W \max\{C_l, C_l\})\rho[1_{\Xi^\bot}] + |Z_2 - Z_1|}{Z_1} + 2 \exp(W C_l + W \max\{C_l, C_l\})\rho[1_{\Xi^\bot}].
\]

This completes the proof. \(\square\)

Note that \(\rho[1_{\Xi^\bot}] = \int_{\Xi^\bot} \rho(\xi)d\xi\) is exactly the posterior measure of \(\Xi^\bot\). Theorem 4 says that, given a prescribed \(e\), if the posterior measure of the region where the surrogate model \(\tilde{l}(\xi)\) is inaccurate, i.e., \(\rho[1_{\Xi^\bot}]\), is small, the approximate posterior \(\overline{\rho}\) is close to the true posterior \(\rho\) (depending on the prescribed accuracy \(e\)). This indicates that the local RB surrogate model only needs to be accurate over the “important region” where the majority of the posterior mass is contained.

Indeed, thanks to the information contained in the data, the posterior distribution typically concentrate on a much smaller portion of the support of the prior, in which case requiring a “globally accurate” surrogate model over the support of the prior seems unnecessary and inefficient. To this end, several recent studies have exploited such information (or posterior) to build adaptive and data-driven surrogate models that are more efficient and accurate on the support of the posterior [CMW15, LM14, Con14, CMPS16]. Significant computational savings can be realized through such adaptive methods. The local RB method is naturally tailored to provide locally accurate approximations as shown in the previous chapters. We will show that the local RB indeed provides efficient approximations that allow us to approximate Gibbs posterior with a rather limited number of PDE solves.
5.3 Particle based approximation with local RB surrogate

A particle based approximation depends on a set of weighted samples \( \{\xi_i, w_i\}_{i=1}^m \), which defines the following empirical measure

\[
\rho^E(\xi) = \sum_{i=1}^m w_i \delta(\xi - \xi_i)
\]

(5.9)

where \( \delta(\cdot) \) is the Dirac’s delta function and the weights \( w_i \) satisfies that \( w_i \in [0, 1] \) and \( \sum_{i=1}^m w_i = 1 \). This form of approximation has been used extensively in sequential Monte Carlo (SMC) methods [DDFG01, DMDJ06]. Here, we use such an approximation for the Gibbs posterior.

To approximate a given distribution \( \rho(\xi) \), \( \xi_i \) and \( w_i \) need to be selected in some principled manner, e.g., by minimizing some distance between \( \rho^E(\xi) \) and \( \rho(\xi) \). A set of Monte Carlo (MC) samples drawn from \( \rho(\xi) \), with equal weights is a particle approximation to \( \rho(\xi) \). However, MC approximation typically has slow convergence and high variance. One can obtain better particle approximation using, e.g. stochastic reduced order models (SROM) proposed in [Gri09, Gri12]. In SROM, the samples \( \xi_i \) are determined by solving an optimization problem. Here \( \xi_i \) form a Voronoi-cell based partition of \( \Xi \), i.e., \( \Xi = \bigcup_{i=1}^m \Xi_i \) where \( \Xi_i \) is centered at \( \xi_i \). The weights \( w_i \) are the probability measure associated with each \( \Xi_i \), i.e., \( w_i = \int_{\Xi_i} \rho(\xi) d\xi \).

Now, suppose we have a particle based approximation for the prior \( \rho_0(\xi) \) based on the particle set \( \{\xi_i^0, w_i^0\}_{i=1}^m \). Using the empirical measure \( \rho^E_0(\xi) := \sum_{i=1}^m w_i^0 \delta(\xi - \xi_i) \) in (5.3), we have an update formula for the particle weights as

\[
w_i = \frac{\exp(-Wl(\xi_i)) w_i^0}{\sum_{k=1}^m w_k^0 \exp(-Wl(\xi_k))},
\]

(5.10)

where \( w_i \) is the posterior weight associated with \( \xi_i \). By (5.9), \( \{\xi_i, w_i\}_{i=1}^m \) defines a particle based approximation \( \rho^E(\xi) \) to the Gibbs posterior distribution \( \rho(\xi) \) in (5.3).
When a surrogate model $\tilde{I}(\xi)$, e.g., the local RB surrogate, is employed for the loss function, we have an approximated update as

$$w_i = \frac{\exp(-W\tilde{I}(\xi_i))w_i^0}{\sum_{k=1}^m w_k^0 \exp(-W\tilde{I}(\xi_k))}.$$  

(5.11)

The above particles $\{\xi_i, w_i\}_{i=1}^m$ defines a surrogate empirical measure

$$\rho^E(\xi) = \sum_{i=1}^m w_i \delta(\xi - \xi_i)$$

(5.12)

that is an approximation to $\rho^E(\xi)$.

### 5.3.1 Convergence analysis for fixed particles

We now address the convergence of the above approximations based on a fixed set of particles. The evolution of particles will be addressed in the following section using the framework of SMC methods. Some of the results here are cornerstones for the convergence of SMC method we describe later. We first state a lemma regarding a randomized particle based approximation.

**Lemma 1.** Given a distribution $\rho_0(\xi)$, let $\{\xi_i\}_{i=1}^m$ be drawn independently from $\rho_0(\xi)$, $w_i^0$ be the associated probability weight, and $\rho^E_0(\xi)$ be the empirical distribution in (5.9), then we have that

$$h(\rho^E_0, \rho_0) \leq \sqrt{\sum_{i=1}^m (w_i^0)^2}$$

(5.13)

in particular, if $w_i^0 = \frac{1}{m}$ as for the MC weights, we have that

$$h(\rho^E_0, \rho_0) \leq \frac{1}{\sqrt{m}}$$

(5.14)

**Proof.** For $\forall f$, we have

$$\rho^E_0[f] = \sum_{i=1}^m w_i^0 f(\xi_i),$$

106
hence
\[ \rho_0^E[f] - \rho_0[f] = \sum_{i=1}^{m} w_i^0 (f(\xi_i) - \rho_0[f]). \]

Note that since \( \xi_i \) are i.i.d. with distribution \( \rho_0(\xi) \), we have \( \mathbb{E}[f(\xi_i)] = \rho_0[f] \) \( \forall \ i = 1, \ldots, m \).
Hence
\[ \mathbb{E}[(f(\xi_i) - \rho_0[f])(f(\xi_i) - \rho_0[f])] = \delta_{ij}\mathbb{E}[|f(\xi_i) - \rho_0[f]|^2]. \]

Additionally, since \( |f|_\infty \leq 1 \), we have
\[ \mathbb{E}[(f(\xi_i) - \rho_0[f])^2] = \mathbb{E}[|f(\xi_i)|^2] - \rho_0[f]^2 \leq 1. \]

Therefore
\[ \mathbb{E}[|\rho_0^E[f] - \rho_0[f]|^2] = \sum_{i=1}^{m} (w_i^0)^2 \mathbb{E}[|(f(\xi_i) - \rho_0[f])|^2] \leq \sum_{i=1}^{m} (w_i^0)^2, \]

which completes the proof. \( \square \)

**Lemma 2.** Denote the transformation of Gibbs posterior formula in (5.3) as \( G_W : P \to P \), such that \( \rho(\xi) = G_W \rho_0(\xi) \). Under Assumption 7, we have
\[ h(G_W \rho_1, G_W \rho_2) \leq 2 \exp(WCl)h(\rho_1, \rho_2), \]  
for \( \forall \ \rho_1, \rho_2 \in P \).

**Proof.** We have
\[
G_W \rho_1[f] - G_W \rho_2[f] = \frac{\int_{\Xi} \exp(-Wl(\xi)) \rho_1(\xi)f(\xi)d\xi}{\int_{\Xi} \exp(-Wl(\xi)) \rho_1(\xi)d\xi} - \frac{\int_{\Xi} \exp(-Wl(\xi)) \rho_2(\xi)f(\xi)d\xi}{\int_{\Xi} \exp(-Wl(\xi)) \rho_2(\xi)d\xi}
\]
\[
= \frac{\int_{\Xi} \exp(-Wl(\xi)) (\rho_1(\xi) - \rho_2(\xi)) f(\xi) d\xi}{Z_3} + \frac{(Z_4 - Z_3) \int_{\Xi} \exp(-Wl(\xi)) \rho_2(\xi)f(\xi)d\xi}{Z_3 Z_4},
\]

where
\[
Z_3 = \int_{\Xi} \exp(-Wl(\xi)) \rho_1(\xi)d\xi, \quad Z_4 = \int_{\Xi} \exp(-Wl(\xi)) \rho_2(\xi)d\xi.
\]
Hence

\[ |G_W \rho_1[f] - G_W \rho_2[f]| \leq \left| \frac{\int_{\Xi} \exp(-Wl(\xi))(\rho_1(\xi) - \rho_2(\xi))f(\xi)d\xi}{Z_3} \right| + \frac{|Z_4 - Z_3|}{Z_3 Z_4} \left| \int_{\Xi} \exp(-Wl(\xi))\rho_2(\xi)f(\xi)d\xi \right|.

Note that since \(|f|_\infty \leq 1|,\)

\[ \left| \int_{\Xi} \exp(-Wl(\xi))\rho_2(\xi)f(\xi)d\xi \right| \leq \int_{\Xi} \exp(-Wl(\xi))\rho_2(\xi)d\xi = Z_4 \]

Using Assumption 7, we have

\[ Z_3 = \int_{\Xi} \exp(-Wl(\xi))\rho_1(\xi)d\xi \geq \exp(-WC_l) \]

hence we have

\[ |G_W \rho_1[f] - G_W \rho_2[f]| \leq \frac{\int_{\Xi} \exp(-Wl(\xi))(\rho_1(\xi) - \rho_2(\xi))f(\xi)d\xi}{\exp(-WC_l)} + \frac{|Z_4 - Z_3|}{\exp(-WC_l)}. \]

Note that \(|\exp(-Wl(\xi))|_\infty \leq 1| and \(|\exp(-Wl(\xi))f(\xi)|_\infty \leq 1\) for \(\forall f\) such that \(|f|_\infty \leq 1|, hence

\[ \left| \int_{\Xi} \exp(-Wl(\xi))(\rho_1(\xi) - \rho_2(\xi))f(\xi)d\xi \right| \leq \sup_{|g|_\infty \leq 1} |\rho_1[g] - \rho_2[g]| \]

and

\[ |Z_4 - Z_3| = \left| \int_{\Xi} (\rho_1(\xi) - \rho_2(\xi)) \exp(-Wl(\xi))d\xi \right| \leq \sup_{|g|_\infty \leq 1} |\rho_1[g] - \rho_2[g]|. \]

Therefore,

\[ |G_W \rho_1[f] - G_W \rho_2[f]| \leq 2 \exp(WC_l) \sup_{|g|_\infty \leq 1} |\rho_1[g] - \rho_2[g]| \]

for \(\forall f\) such that \(|f|_\infty \leq 1|\). The lemma follows easily from the above inequality. \(\square\)

Next we consider the surrogate approximation. We first state a counterpart of Lemma 2 when the surrogate model (5.4) is used instead of (5.3):
Lemma 3. Denote the transformation of Gibbs posterior formula in (5.4) as $G_W : P \to P$, such that $\bar{p}(\xi) = G_W p_0(\xi)$. Under Assumption 7, we have

$$h(G_W p_1, G_W p_2) \leq 2 \exp(WC_1) h(p_1, p_2),$$

(5.17)

for $\forall p_1, p_2 \in P$.

Proof. Same as Lemma 2. \hfill \Box

We impose the following additional assumption on the local RB surrogate loss function $l(\xi)$.

Assumption 8. We assume that the surrogate loss function $l(\xi)$ is accurate over the particle set, that is,

$$\sup_{i=1, \ldots, m} |l(\xi_i) - \bar{l}(\xi_i)| \leq e$$

(5.18)

for some $e > 0$ indicating the error of the local RB approximation.

This bound (5.18) can be satisfied by running the local RB Algorithm 1 using $\{\xi_i\}_{i=1}^m$ as training samples. A local RB error indicator can be built for $l(\xi)$ in the same fashion that we build error indicators in Chapter 3 for QoIs. We have the following two lemmas quantifying the difference between $\rho^E$ with weights computed by (5.10) and $\bar{\rho}^E$ in (5.12).

Lemma 4. Under Assumptions 7 and 8, the following bound holds:

$$D_{KL}(\bar{\rho}^E | \rho^E) \leq 2We.$$

Proof. Based on Assumption 8, we have

$$\frac{\exp(-Wl(\xi_i))}{\exp(-W\bar{l}(\xi_i))} = \frac{\exp(Wl(\xi_i) - W\bar{l}(\xi_i))}{\exp(W\bar{l}(\xi_i) - Wl(\xi_i))} \leq \exp(We) \quad \forall i = 1, \ldots, m$$

and

$$\frac{\exp(-W\bar{l}(\xi_i))}{\exp(-W\bar{l}(\xi_i))} = \frac{\exp(W\bar{l}(\xi_i) - W\bar{l}(\xi_i))}{\exp(W\bar{l}(\xi_i) - Wl(\xi_i))} \leq \exp(We) \quad \forall i = 1, \ldots, m.$$

We first define that

$$Z_5 = \sum_{k=1}^m w_k^0 \exp(-Wl(\xi_k)), \quad Z_6 = \sum_{k=1}^m w_k^0 \exp(-W\bar{l}(\xi_k)).$$

(5.19)
It is clear that
\[
\log \left( \frac{Z_5}{Z_6} \right) \leq \log \left( \frac{\exp(We) \sum_{k=1}^m w_k^0 \exp(-W\tilde{\epsilon}(\xi_k))}{\sum_{k=1}^m w_k^0 \exp(-W\tilde{\epsilon}(\xi_k))} \right) = We.
\]

We hence have that
\[
D_{KL}(\rho^E|\rho^E) = \sum_{k=1}^m \frac{\exp(-W\tilde{\epsilon}(\xi_k))w_k^0}{Z_6} \log \left( \frac{\exp(-W\tilde{\epsilon}(\xi_k))w_k^0 Z_5}{\exp(-W\tilde{\epsilon}(\xi_k))w_k^0 Z_6} \right)
\]
\[
\leq \sum_{k=1}^m \frac{\exp(-W\tilde{\epsilon}(\xi_k))w_k^0}{Z_6} \left[ \log \left( \frac{Z_5}{Z_6} \right) + We \right]
\]
\[
= \log \left( \frac{Z_5}{Z_6} \right) + We = 2We.
\]

Lemma 5. Under Assumptions 7 and 8, the following bound holds:
\[
h(\rho^E,\bar{\rho}^E) \leq 2 \exp(WC_l)C_e We,
\]
for some constant $C_e > 0$.

Proof. Recall the definition of $Z_5$ and $Z_6$ in (5.19), we have
\[
\rho^E[f] - \bar{\rho}^E[f] = \frac{\sum_{j=1}^m \exp(-W\tilde{\epsilon}(\xi_j))w_j^0 f(\xi_j)}{\sum_{k=1}^m \exp(-W\tilde{\epsilon}(\xi_k))w_k^0} - \frac{\sum_{j=1}^m \exp(-W\tilde{\epsilon}(\xi_j))w_j^0 f(\xi_j)}{\sum_{k=1}^m \exp(-W\tilde{\epsilon}(\xi_k))w_k^0}
\]
\[
= \frac{\sum_{j=1}^m (\exp(-W\tilde{\epsilon}(\xi_j)) - \exp(-W\tilde{\epsilon}(\xi_j)))w_j^0 f(\xi_j)}{Z_5} + \frac{(Z_6 - Z_5) \sum_{j=1}^m \exp(-W\tilde{\epsilon}(\xi_j))w_j^0 f(\xi_j)}{Z_5 Z_6}.
\]

Note that
\[
Z_5 = \sum_{k=1}^m \exp(-W\tilde{\epsilon}(\xi_k))w_k^0 \geq \exp(-WC_l),
\]
and that
\[
\left| \sum_{j=1}^m \exp(-W\tilde{\epsilon}(\xi_j))w_j^0 f(\xi_j) \right| \leq Z_6,
\]

110
since \(|f|_\infty \leq 1\). Also, since \(-e \leq l(\xi_j) - l(\xi_j) \leq e\), for \(e\) sufficiently small, e.g., \(We \ll 1\), we have

\[|\exp(-Wl(\xi_j)) - \exp(-W\tilde{l}(\xi_j))| \leq \exp(-Wl(\xi_j))[1 - \exp(Wl(\xi_j) - W\tilde{l}(\xi_j))] \leq C_e We\]

for some \(C_e > 0\). Hence

\[|Z_5 - Z_6| \leq C_e We.\]

Finally, we have

\[|\rho^E[f] - \rho^{E*}[f]| \leq 2 \exp(WC_t)C_e We,

which implies the Lemma.

Now, we state a theorem concerning the particle-based approximation to the Gibbs posterior using local RB surrogate model.

**Theorem 5.** For a empirical measure \(\rho_0^E\) based on \({\xi_i, w_i^0}\)\(i=1, m\), which is an approximation to the prior \(\rho_0\), the approximate posterior \(\rho^{E*}\) defined by (5.12) satisfies

\[h(\rho^{E*}, \rho) \leq 2 \exp(WC_t)C_e We + 2 \exp(WC_t) \sqrt{\sum_{i=1}^{m} (w_i^0)^2}\]

(5.20)

for the same constants \(C_e > 0\) as in Lemma 5. In particular, if \(w_i^0 = \frac{1}{m}\), i.e., the particles are MC samples of the prior, we have

\[h(\rho^{E*}, \rho) \leq 2 \exp(WC_t)C_e We + 2 \exp(WC_t) \frac{1}{\sqrt{m}}.\]

**Proof.** By triangle inequality and Lemma 1, 2 and 5, we have that

\[h(\rho^{E*}, \rho) \leq h(\rho^{E*}, \rho^E) + h(\rho^E, \rho)\]

\[\leq h(\rho^{E*}, \rho^E) + 2 \exp(WC_t)h(\rho^E_0, \rho)\]

\[\leq 2 \exp(WC_t)C_e We + 2 \exp(WC_t) \sqrt{\sum_{i=1}^{m} (w_i^0)^2}.\]

This completes the proof. \(\Box\)
The above theorem says that if we use MC approximation for the prior and drive $e \to 0$ and $m \to \infty$, we have $h(\rho^*, \rho) \to 0$. In practice, however, it is nontrivial to do both at the same time, as when the number of particles $m$ increases, we require stronger global accuracy on our surrogate model $\tilde{l}(\xi)$, which can only be achieved by globally refining the surrogate with more PDE solves. To efficiently represent the posterior distribution with a limited number of particles, we rely on a SMC method to progressively evolve the particles through a sequence of interpolating distributions from $\rho_0$ to $\rho$. The focus of the local RB surrogate is automatically navigated to the support of the posterior in the process.

5.4 An adaptive Sequential Monte Carlo method for particle evolution

The above discussion deals with the asymptotic convergence of the particle approximation based on a fixed set of particles. In practice, using (5.11) as an approximation to the posterior weights with a fixed set of particles drawn from $\rho_0$ leads to rather poor approximation to $\rho$, especially when $W$ is large. The reason is due to the loss of sample diversity, i.e., the posterior weights become very unbalanced over the particle set that majority of the total weight is assigned to only one or a few particles. This means that the effective sample size of the particles is diminished and the statistical information of the ensemble is lost.

In SMC, instead of computing the posterior weights in one shot, particles are evolved to approximate a sequence of intermediate distributions interpolating from the prior to posterior. The particles are resampled and mutated after each iteration to prevent degeneracy. We adopt such an SMC framework to approximate the Gibbs posterior distribution. In particular, our SMC method for Gibbs posterior follows closely the recent work in [KBJ14] where the authors proposed an SMC method for high dimensional inverse problems.
5.4.1 The Sequential Monte Carlo method

In the context of Gibbs posterior, the sequence of the interpolating distributions are defined by

$$\rho_t = G_{W_t} \rho_0, \ 0 \leq t \leq N$$  \hfill (5.21)

where $0 = W_0 < W_1 < W_2 \cdots < W_t < \cdots < W_N = W$, and recall the definition of $G_W$ as the Gibbs update formula defined in (5.3). Apparently $\rho_N = \rho$ that is the posterior distribution we want to approximate. Also, it is easy to show that we have the following property

$$\rho_t = G_{W_t - W_s} \rho_s, \ 0 \leq s \leq t \leq N$$  \hfill (5.22)

by the Gibbs update formula (5.3). This property allows us to apply SMC methods and progressively approximate $\rho$ (i.e., $\rho_N$).

As mentioned, the key idea of SMC is to start from a particle based approximation of $\rho_0$, i.e., $\rho_0^E$, which is easy to obtain, and gradually increase the weight $W_t$ until it reaches $W$, adjusting the particles along the way. To this end, we denote the particle approximation to $\rho_t$ as $\rho_t^E$,

$$\rho_t^E = \sum_{i=1}^{m} w_t^i \delta(\xi - \xi_t^i)$$

based on the particle set $\{\xi_t^i, w_t^i\}_{i=1}^m$.

The iteration $t+1$ of the SMC involves three steps: (i) update the weights of the current particle set $\{\xi_t^i\}_{i=1}^m$ by

$$w_{i}^{t+1} = \frac{\exp(-(W_{t+1} - W_t)l(\xi_t^i))w_t^i}{\sum_{k=1}^{m} w_k^i \exp(-(W_{t+1} - W_t)l(\xi_k^i))}.$$  \hfill (5.23)

The distribution based on $\{\xi_t^i, w_{t+1}^i\}_{i=1}^m$ is denoted by

$$\rho_{t+1,t}^E = \sum_{i=1}^{m} w_{t+1}^i \delta(\xi - \xi_t^i),$$

that is $\rho_{t+1,t}^E = G_{W_{t+1} - W_t} \rho_t^E$. By Lemma 2, we have that

$$h(\rho_{t+1,t}^E, \rho_{t+1}) \leq 2 \exp((W_{t+1} - W_t)C_1) h(\rho_t^E, \rho_t).$$  \hfill (5.24)
(ii) Resample the particles \( \{\xi^{t+1}_{i} \}_{i=1}^{m} \) with replacement according to the weights \( \{w^{t+1}_{i} \}_{i=1}^{m} \), i.e., resample according to \( \rho_{t+1,t}^{E} \). This step effectively eliminates the particles with negligible weights and duplicate the particles with large weights. All resampled particles, including the duplicates, are denoted by \( \{\xi^{t+1}_{i} \}_{i=1}^{m} \) and are assigned equal weights \( \frac{1}{m} \). The resampled distribution is denoted by

\[
\rho_{t+1,t}^{E,S} = \frac{1}{m} \delta(\xi - \xi^{t+1}_{i})
\]

By Lemma 1, we have that

\[
h(\rho_{t+1,t}^{E,S}, \rho_{t+1,t}^{E}) \leq \frac{1}{\sqrt{m}}.
\]

This resampling step alone does not prevent sample degeneracy, as only a few particles will survive and copy themselves. To preserve population diversity, a third step is required.

(iii) Apply a \( \rho_{t+1} \)-invariant mutation to the resampled set \( \{\xi^{t+1}_{i} \}_{i=1}^{m} \) in step (ii). This can be achieved by evolving the particles \( \{\xi^{t+1}_{i} \}_{i=1}^{m} \) independently by one or more steps using a \( \rho_{t+1} \)-invariant Markov kernel \( K_{t+1} \) (i.e., \( \rho_{t+1} = \rho_{t+1} K_{t+1} \)), e.g., an MCMC kernel with \( \rho_{t+1} \) as the stationary distribution. Note that the property of \( K_{t+1} \) implies that [RVH15, BJMS15],

\[
h(pK_{t+1}, qK_{t+1}) \leq h(p, q), \ \forall \ p, q \in P, \ \forall \ 0 \leq t \leq N - 1.
\]

The resulted particles from step (iii), denoted by \( \{\xi^{t+1}_{i} \}_{i=1}^{m} \), with weights \( \frac{1}{m} \), define the distribution

\[
\rho_{t+1}^{E} = \frac{1}{m} \delta(\xi - \xi^{t+1}_{i})
\]

that is used to approximate \( \rho_{t+1} \) and is used for the next iteration of the SMC. We adopt the same MCMC mutation kernel proposed in [KB14] for this step, which has been shown to be efficient for high dimensional inverse problems.

We have the following theorem regarding the SMC method for the Gibbs posterior.
Theorem 6. Assuming that the initial particles are a set of MC samples with equal weights $1/m$, then following the above outlined SMC method with the exact loss function $l(\xi)$, we have that for all iterations $t : 0 \leq t \leq N - 1,$

$$h(\rho_{t+1}^E, \rho_{t+1}) \leq \frac{1}{\sqrt{m}} \sum_{s=0}^{t+1} 6^{t+1-s} \exp((W_{t+1} - W_s)C_t)$$

(5.27)

in particular, the we have a posterior error bound

$$h(\rho^E, \rho) \leq \frac{1}{\sqrt{m}} \sum_{s=0}^{N} 6^{N-s} \exp((W - W_s)C_t),$$

(5.28)

where we have $W = W_N$.

Proof. First, by Equation (5.25), (5.26) and the fact that $\rho_{t+1} = \rho_{t+1}K_{t+1}$, we have

$$h(\rho_{t+1}^E, \rho_{t+1}^E) = h(\rho_{t+1}^ES_{t+1}K_{t+1}, \rho_{t+1}^E) \leq h(\rho_{t+1}^ES_{t+1}K_{t+1}, \rho_{t+1}^EK_{t+1}) + h(\rho_{t+1}^E, \rho_{t+1}^E)$$

$$\leq h(\rho_{t+1}^ES_{t+1}, \rho_{t+1}^E) + h(\rho_{t+1}^E, \rho_{t+1}^E) + h(\rho_{t+1}, \rho_{t+1}^E)$$

$$\leq \frac{1}{\sqrt{m}} + 2h(\rho_{t+1}, \rho_{t+1}^E).$$

Hence

$$h(\rho_{t+1}^E, \rho_{t+1}) \leq h(\rho_{t+1}^E, \rho_{t+1}^E) + h(\rho_{t+1}, \rho_{t+1}^E) \leq \frac{1}{\sqrt{m}} + 3h(\rho_{t+1}, \rho_{t+1}^E)$$

$$\leq \frac{1}{\sqrt{m}} + 6 \exp((W_{t+1} - W_t)C_t)h(\rho_t^E, \rho_t)$$

by Equation (5.24). Iterating gives

$$h(\rho_{t+1}^E, \rho_{t+1}) \leq \frac{1}{\sqrt{m}} \sum_{s=0}^{t+1} 6^{t+1-s} \exp((W_{t+1} - W_s)C_t),$$

which completes the proof. \qed
When a local RB surrogate loss function $\overline{l}(\xi)$ is used, the sequence of distributions are defined by $\overline{\rho^E}$. The update in step (i) is replaced by

$$w_{i}^{t+1} = \frac{\exp(-(W_{t+1} - W_t)\overline{l}(\xi_i^t))w_i^t}{\sum_{k=1}^{m}w_k^t \exp(-(W_{t+1} - W_t)\overline{l}(\xi_k^t))},$$  \hspace{1cm} (5.29)$$

which defines $\overline{\rho_{t+1}^E} = \sum_{i=1}^{m}w_{i}^{t+1}\delta(\xi - \xi_i^t)$. That is, $\overline{\rho_{t+1}^E} = \overline{C_{W_{t+1} - W_t}\overline{\rho^E}}$. By Lemma 3, we have

$$h(\overline{\rho_{t+1}^E}, \overline{\rho^{t+1}}) \leq 2 \exp((W_{t+1} - W_t)C_t)h(\overline{\rho^E}, \overline{\rho^t}),$$  \hspace{1cm} (5.30)$$

where

$$\overline{\rho^t}(\xi) = \frac{\exp(-(W_{t}\overline{l})(\xi))\rho_0(\xi)}{\int_\Xi \exp(-(W_{t}\overline{l})(\xi))\rho_0(\xi)d\xi}.$$  \hspace{1cm} (5.31)$$

In addition, the kernel mutation step requires evaluation of the loss function $l(\xi)$ at new parameters which can be accelerated by $\overline{l}(\xi)$ as well. To this end, we use a surrogate kernel $\overline{K_{t+1}}$ that is invariant with respect to $\overline{\rho_{t+1}}$. The mutation with respect to $\overline{K_{t+1}}$ only requires evaluation of $\overline{l}(\xi)$. We have the following theorem regarding the SMC method using $\overline{l}(\xi)$:

**Theorem 7.** Assuming that the initial particles are a set of MC samples with equal weights $1/m$, then following the above outlined SMC method using the local RB surrogate $\overline{l}(\xi)$ for step (i) and (iii), we have that for all iterations $t : 0 \leq t \leq N - 1,$

$$h(\overline{\rho_{t+1}^E}, \overline{\rho^{t+1}}) \leq \frac{1}{\sqrt{m}} \sum_{s=0}^{t+1} 6^{t+1-s} \exp((W_{t+1} - W_s)C_t) + 2 \exp(W_{t+1}C_t)C_{eW_{t+1}}e$$

$$+ 2 \exp(W_{t+1} + W_{t+1} \max\{C_t, C_t\})\rho_{t+1}[1_{\Xi^c}]$$  \hspace{1cm} (5.32)$$

In particular, the we have a posterior error bound

$$h(\overline{\rho^E}, \rho) \leq \frac{1}{\sqrt{m}} \sum_{s=0}^{N} 6^{N-s} \exp((W - W_s)C_t) + 2 \exp(WC_t)C_{eW}e$$

$$+ 2 \exp(WC_t + W \max\{C_t, C_t\})\rho[1_{\Xi^c}],$$  \hspace{1cm} (5.33)$$

where we have $W = W_N$. 

116
**Proof.** The first term on the right-hand-side comes from a simple restatement of Theorem 6 for \( h(\rho_{t+1,1}, \rho_{t+1}) \). The remainders of the right-hand-side is due to the error bound in Theorem 4. 

From the Theorem 7, we see how we should construct the surrogate model \( \tilde{l}(\xi) \). Given the number of particles \( m \) and the prescribed surrogate accuracy \( e \), we need to build the surrogate \( \tilde{l}(\xi) \) so that \( \rho[1_{\Xi^K,e}] \), i.e., the posterior measure of the “unfeasible set” \( \Xi^K,e \) is minimized. In terms of local RB surrogate, this requires us to concentrate local RB atoms and accurately approximate \( l(\xi) \) over the support of the posterior. To this end, we progressively train the local RB surrogate using the sequence of particles \( \{\xi_i^t\}_{i=1}^m \). As the particles gradually cluster over the support for the posterior through the SMC iterations, the focus of the local RB surrogate is automatically navigated to the support of the posterior as well, resulting in the reduction of \( \rho[1_{\Xi^K,e}] \). In addition, because the support of the posterior is typically a small and local region of the support of the prior, once the local RB model becomes sufficiently accurate over that region, further evolving the particles no longer triggers expensive update of the local RB model, leading to computational savings.

Note that the surrogate loss function \( \tilde{l}(\xi) \) is changing throughout the SMC iterations, one can recover consistency in (5.31) for all \( t \) by rerunning the SMC from beginning up to the current \( W_t \) using the latest \( \tilde{l}(\xi) \) before the next SMC iteration. This procedure is cheap since the surrogate model needs not evolve during the rerun and hence no full PDE solves are required. Therefore, we always assume the update (5.31) is consistent for all iterations with the latest surrogate model \( \tilde{l}(\xi) \).

We now present the MCMC algorithm for the mutation step with local RB surrogate \( \tilde{l}(\xi) \). To this end, we first define the following mean and variance based on \( \rho^{E}_{t+1,t} \), which is the particle distribution after SMC step (i) and before resampling step (ii), for each parameter dimension \( j \in \{1,2,\ldots,M\} \) as

\[
\begin{align*}
\bar{m}_{t+1,t,j}^E &= \sum_{i=1}^m w_i^{t+1} \xi_{i,j}^t, \\
\Sigma_{t+1,t,j}^E &= \sum_{i=1}^m w_i^{t+1} (\xi_{i,j}^t - \bar{m}_{t+1,t,j}^E)^2.
\end{align*}
\] (5.34)

The above quantities provide estimates of the mean and variance of \( \rho_{t+1} \) along each indi-
individual parameter dimension at SMC iteration $t + 1$, and will be used to facilitate the design of a proposal distribution for the MCMC kernel $\tilde{K}_{t+1}$.

Based on the above definition, a proposal $\xi_{i,j}^{t+1,t}$ for a particle $\xi_{i}^{t+1,t} \in \{\xi_{i}^{t+1,t}\}_{i=1}^{m}$ can be obtained by the following mutation

$$
\xi_{i,j}^{t+1,t} = m_{t+1,t,j} + \gamma (\xi_{i,j}^{t+1,t} - m_{t+1,t,j}) + \sqrt{1 - \gamma^2} N(0, \Sigma_{t+1,t,j}) \quad 1 \leq j \leq M
$$

where $\gamma$ is an algorithmic constant. Note that the scaling of the proposal distribution is tailored for each individual parameter dimension by the variance estimates $\Sigma_{t+1,t,j}$ to improve mixing. And different from random walk proposals, the above proposal scales to high dimensional problems as shown in [KBJ14]. The transition probability associated with the proposal in (5.35) is given by

$$
Q(\xi_{i}^{t+1,t} | \xi_{i}^{t+1,t}) = \exp \left( -\frac{1}{2(1 - \gamma^2)} \sum_{j=1}^{M} \frac{(\xi_{i,j}^{t+1,t} - m_{t+1,t,j} - \gamma (\xi_{i,j}^{t+1,t} - m_{t+1,t,j}))^2}{\Sigma_{t+1,t,j}} \right). \quad (5.36)
$$

Algorithm 5 shows the $\tilde{p}_{t+1}$-invariant mutation MCMC sampler.

### 5.4.2 Adaptive selection of the SMC step size

We now describe how the sequence of step size $0 = W_{0} < W_{1} < W_{2} \cdots < W_{t} < \cdots < W_{N} = W$ can be selected adaptively. For each SMC iteration $t + 1$, we would like to greedily apply all the residual weight $\Delta W = W_{N} - W_{t}$ to the particle distribution $\tilde{p}_{t}^{E}$ from the previous iteration, so that we can directly approximate the posterior $\rho$. After applying the SMC step (i), i.e., updating the weights by Equation (5.29) using $\Delta W$, we check a simple criteria called the effective sample size (ESS), which is used to measure the sample degeneracy of the current weights,

$$
\text{ESS} = \left( \sum_{k=1}^{m} \left( w_{i}^{t+1} \right)^2 \right)^{-1}.
$$

Note that ESS is small if the majority of the probability weights are pivoted on only a few particles, which indicates the loss of sample diversity. In this case, we reduce the incremental
Algorithm 5: The MCMC algorithm for $\rho_{t+1}$-invariant mutation

For each $\xi(0) \in \{\xi^{t+1,1}_i\}_{i=1}^m$, evolve $\xi(0)$ independently for $I$ steps with the following procedure

- For $i = 1, 2, \ldots, I$, do
  - Draw a proposal $\hat{\xi}(i)$ using the proposal distribution in (5.35) based on $\xi(i-1)$.
  - Use $l_p(\xi)$ to evaluate $\alpha = 1 \wedge \exp\left(\frac{-W_{t+1}\xi(i)) \rho_0(\xi(i))Q(\xi(i-1))\xi(i)}{\exp(-W_{t+1}\xi(i-1)) \rho_0(\xi(i-1))Q(\xi(i))\xi(i-1)}\right)$
  - With probability $1 - \alpha$, reject $\hat{\xi}(i)$ and set $\xi(i) = \xi(i-1)$, $i = i + 1$. Go back to the proposal step.
  - If $\hat{\xi}(i)$ not rejected, set $\xi(i) = \hat{\xi}(i)$ and $i = i + 1$. Go back to the proposal step.

End

- Finally, return $\xi(I)$ as a $\rho_{t+1}$-invariant mutation of $\xi(0)$.

Weights by a constant factor $\Delta W = \theta \Delta W$ with $\theta \in (0, 1)$ and repeatedly backtrack and reevaluate Equation (5.29) and ESS until the ESS is above some preset threshold, in which case we accept the $\Delta W$, set $W_{t+1} = W_t + \Delta W$, move on to the step (ii) and (iii) and finish the current SMC iteration $t + 1$. If the residual weight is not zero after iteration $t + 1$, we set $t = t + 1$ and move to the next SMC iteration. Otherwise, we have applied the total weight to the prior and obtained an approximation to the posterior.

Of course, the local RB surrogate model $\tilde{l}(\xi)$ evolves as well by the adaptive training on the particles before applying the incremental weight in each SMC iteration. We require $\tilde{l}(\xi)$ to satisfy Assumption 8 for all iteration $t$. As the particles gradually cluster over a small region in the parameter space, i.e., the support for the posterior, $\tilde{l}(\xi)$ becomes accurate in that region as well, reducing $\rho_{t+1,\Xi^K_{t+1}}$ as a result. In addition, as the particles become more compact in a local region, expensive refinements of the local RB model are less often triggered due to the local accuracy of $\tilde{l}(\xi)$.
We first present the adaptive refinement of the local RB surrogate over a given particle set \( \{\xi^i\}^m_{i=1} \) in Algorithm 6. We then show the complete adaptive SMC method in Algorithm 7. In Algorithm 6, we note that the accuracy parameter \( \epsilon_{\text{thre}} \) is prescribed beforehand and can be made adaptive as well. For example, we can further improve computational efficiency by setting a larger \( \epsilon_{\text{thre}} \) in the beginning stage of the SMC algorithm and gradually reduce \( \epsilon_{\text{thre}} \) throughout the iterations. This strategy leads to computational savings in the beginning stage when the particles are far from the support of the posterior and are less relevant for characterizing the posterior distribution. However, it is essential to set \( \epsilon_{\text{thre}} \) small enough such that each SMC iteration still leads to the particles moving towards the posterior.

One possible approach is to choose \( \epsilon_{\text{thre}} \) based on the range of variation of \( \bar{l}(\xi) \) over the current particle set \( \{\xi^i\}^m_{i=1} \), e.g., we can set \( \epsilon_{\text{thre}} \) to be a small fraction of the standard deviation of \( \{\bar{l}(\xi^i)\}^m_{i=1} \) and compute \( \epsilon_{\text{thre}} \) automatically for each SMC iteration. With this approach, \( \epsilon_{\text{thre}} \) is large in the beginning stage of the SMC algorithm where particles are diverse and the range of variation of \( \bar{l}(\xi) \) is large. In the latter stages where the particles are more compact, \( \bar{l}(\xi) \) has a smaller range over the particles, which leads to a smaller \( \epsilon_{\text{thre}} \).

### 5.5 Choosing the weight for the loss function

Till now, the above discussion assumes a given and fixed weight \( W \) in front of the loss function. This section describes one possible method to pick \( W \). We want to mention that prescribing \( W \) is an important aspect of the Gibbs posterior formulation and requires further validation and investigation.

Methods to select \( W \) are generally subjective in context of Gibbs posterior, and they can be problem dependent as well [BHW16, SM16]. When a likelihood function is available as in the conventional Bayesian inverse problem, \( W \) is built into the distribution model of the noise. For example, when a Gaussian noise model with a standard deviation \( \sigma \) is employed, one can interpret the \( \frac{1}{2\pi\sigma^2} \) in the likelihood function as a weight. The weight makes intuitive sense in that if the noise level is high, we should weight the loss function lightly against
Algorithm 6: Adaptive refinement of local RB surrogate

Given the current particle set $\Xi_P := \{\xi_i^t\}_{i=1}^m$, the current local RB surrogate model $\tilde{l}(\xi)$, and a desired accuracy threshold $e_{\text{thre}}$, 

- Compute the local RB error indicator for each particle in $\Xi_P$ and find the maximum error $e_{\text{max}}$.

- While $e_{\text{max}} > e_{\text{thre}}$, do
  - Select the particle $\xi_{\text{max}}$ in $\Xi_P$ that achieves $e_{\text{max}}$.
  - Update the surrogate model $\tilde{l}(\xi)$ by the new PDE information at $\xi_{\text{max}}$ via local RB method.
  - Update the local RB error indicator for each particle in $\Xi_P$ and recompute $e_{\text{max}}$.

End

- Return the updated surrogate $\tilde{l}(\xi)$.

The prior belief. On the contrary, when the noise level is low, the data should be enforced more heavily since it contains more useful information about the unknown parameters. The work [BHW16] introduced several subjective ways to select $W$. In particular, one proposed method is to select $W$ by balancing two isolated loss terms from the objective function (5.1), under the assumption of a simple $l_2$ loss and a Gaussian prior, the method results in a similar weight $\frac{1}{2\sigma^2}$ based on the variance $\sigma^2$ of the data generating distribution. Inherently, the weight $W$ should be tailored to both the loss function as well as the data generating mechanism. In settings where the number of available data is large, one can also select $W$ using methods like cross-validation to tune the predictive performance of the posterior.

For inverse problems, however, one typically has access to a rather limited number of observations. Without any knowledge about the noise, it is hard to make sense of the
**Algorithm 7: The adaptive SMC method**

Given initial particle approximation $\rho^E_0 := \sum_{i=1}^{m} w_i^0 \delta(\xi - \xi_i^0)$ (and $\overline{w}_i^0 := w_i^0$), the total loss weight $W$ to be applied, set $W_{\text{current}} = 0$, set $t = 0$.

- **While** $W_{\text{current}} < W$, **do**
  - Run Algorithm 6 to possibly refine the local RB surrogate over the current particle set $\{\xi_i\}_{i=1}^m$.
  - Set $\Delta W = W - W_{\text{current}}$.
  - **While** TRUE, **do**
    - Compute the new weight $\overline{w}_i^{t+1}$ by Equation (5.29) using $\Delta W$ as incremental weight.
    - Compute ESS of $\left\{\overline{w}_i^{t+1}\right\}_{i=1}^m$.
    - If ESS > ESS$_{\text{thre}}$, **break**.
    - Backtrack $\Delta W = \theta \Delta W$.
  - End
  - Resample particles $\{\xi_i\}_{i=1}^m$ according to $\left\{\overline{w}_i^{t+1}\right\}_{i=1}^m$ to obtain $\{\xi_i^{t+1,t}\}_{i=1}^m$ with weights $1/m$.
  - Mutate each particle in $\{\xi_i^{t+1,t}\}_{i=1}^m$ with Algorithm 5 to obtain a new set of evolved particles $\{\xi_i^{t+1}\}_{i=1}^m$, set $\overline{w}_i^{t+1} = 1/m$, obtain the current particle approximation $\overline{\rho}^E_{t+1} = \sum_{i=1}^{m} \frac{1}{m} \delta(\xi - \xi_i^{t+1})$.
  - Set $W_{\text{current}} = W_{\text{current}} + \Delta W$, $t = t + 1$.

- **End**

- **Report** $\overline{\rho}^E_{t+1}$ as an approximation to the Gibbs posterior $\rho$.

uncertainty about the inverse solution. Therefore, to pick the value for $W$, we impose some assumptions on the noise. Though we do not require an exact or explicit distribution.
model, we assume the noise are i.i.d with known mean $\mathbb{E}[\epsilon] = \epsilon^M$ and known standard deviation $\sqrt{\mathbb{E}[\epsilon^2] - \mathbb{E}[\epsilon]^2} = \epsilon^D$, which is a weaker assumption than an explicit distribution model. Our approach is in same spirit as Morozov’s discrepancy principle [CPP97, Sch93], in the sense that we would like to select the weight $W$ so that our posterior prediction of the system output, when compared with the actual observed data, reproduces the given noise statistics. More specifically, we would like to select $W = W_{\text{opt}}$ such that

$$W_{\text{opt}} = \arg\min_{W \in \mathcal{W}} \frac{1}{n} \sum_{i=1}^{n} \hat{\epsilon}_i(W) - \epsilon^M + \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (\hat{\epsilon}_i(W) - \epsilon^M)^2 - \epsilon^D}}{\epsilon^D} \tag{5.37}$$

where we assume we have $n$ observations indexed by $i$, and $\hat{\epsilon}_i(W) = \mathcal{F}(\mathbb{E}_\rho(\xi)) - d_i$ is the predicted noise at the expected parameters of the posterior obtained using a weight $W$. Note that, even with only one observation, if multiple channels are available, one can still estimate the statistics of noise and use (5.37) to select $W$. $\mathcal{W}$ is a set that contains the potential candidates for $W$. Of course, designing such a set that allows efficient determination of $W$ is nontrivial as well.

We describe one possible choice here. Since $\frac{1}{2(\epsilon^D)^2}$ is an intuitive and recognized weight which allows us to recover Bayesian update when a $l_2$ loss and Gaussian noise model are used, we pivot the set $\mathcal{W}$ around $\frac{1}{2(\epsilon^D)^2}$, in the sense that $\mathcal{W} := [\frac{1}{2(\epsilon^D)^2} s, T \frac{1}{2(\epsilon^D)^2}]$ with some $T > 1$ (e.g., $T = 50$). However, the optimization problem above is both intractable and unnecessary to be solved exactly, as the objective function is inherently stochastic and inexact, it is of no value to obtain an exact minimizer. We instead use a discrete set with candidate weights spread through $[\frac{1}{2(\epsilon^D)^2}, T \frac{1}{2(\epsilon^D)^2}]$ and pick a weight the minimizes the objective over the set. However, the predicted noise statistics is troublesome when the number of data points $n$ is small. When the number of data is too small, say $n = 1$, there is no way to obtain a reasonable $W$ using the method above. To this end, we stabilize the weight $W$ using a simple convex combination of $\frac{1}{2(\epsilon^D)^2}$ and $W_{\text{opt}}$, e.g.,

$$W = \frac{S}{S + n - 1} \frac{1}{2(\epsilon^D)^2} + \frac{n - 1}{S + n - 1} W_{\text{opt}} \tag{5.38}$$
for some $S \geq 1$ (e.g., $S = 10$). When $n$ is small, we favors the empirical weight $\frac{1}{2(n)^2}$, when $n$ is large and the noise statistics can be computed with good accuracy, we favors the optimized weight $W_{opt}$.

One shortcoming of the above approach is the optimal weight $W_{opt}$ is only based on the posterior mean $E_p(\xi)$; it would be more desirable to integrate more information of $\rho(\xi)$ into the choice of $W_{opt}$. In addition, it would be significant if one can choose the weight $W$ purely based on the data, instead of imposing additional assumptions of the noise. Further investigations are required in this respect.

Note that since our SMC method sequentially increase the weight in an iterative process, hence the above weight selection can be done by running the SMC once up to the maximal weight in $W$ with intermediate steps containing the set $W$. The intermediate distributions at every entry in $W$ are employed to evaluate the objective in (5.37), $W_{opt}$ can be easily recorded upon finishing the SMC run. The final weight is computed by (5.38), after which we run the SMC again to obtain the final approximation to the posterior. Note that the latter step reuses the same local RB surrogate model $\tilde{l}(\xi)$ that has already been constructed in the weight selection step, hence incurs little additional cost to further refine the surrogate through the process.

5.6 Numerical examples

Now, we present three numerical examples to show the behavior and computational efficiency of our SMC method.

5.6.1 1D advection diffusion equation

In the first example, we consider a 1D advection-diffusion problem. We show that our method recovers the usual Bayesian approach when a likelihood function is available and that we use the negative log-likelihood as the weighted loss function.
Let $D = (0,1)$ and consider the following boundary value problem

$$-\nu \frac{\partial^2 u}{\partial x^2}(x,\xi^*) + b(x,\xi^*) \frac{\partial u}{\partial x}(x,\xi^*) = f(x), \quad x \in D \tag{5.39a}$$

$$u(0,\xi^*) = u(1,\xi^*) = 0 \tag{5.39b}$$

The diffusivity, $\nu$, and source, $f$, are known whereas the advection field, $b$, is a piecewise constant field parametrized by two unknown parameters $\xi_1^*$ and $\xi_2^*$:

$$b(x,\xi^*) = [b_1 + 2\xi_1^*] \mathbb{1}_{[0,0.5]}(x) + [b_2 + 2\xi_2^*] \mathbb{1}_{[0.5,1]}(x) \tag{5.40}$$

where $\mathbb{1}_S(x)$ is one if $x$ is in the set $S$ and is zero otherwise.

We are able to measure the solution at three different locations of $x = [0.1, 0.5, 0.9]$. Our noisy data is hence given by

$$d = D u + \epsilon$$

where $D$ is an operator that maps the solution $u(x,\xi^*)$ to the measurement and $\epsilon$ is a noise vector that contains i.i.d entries. We assume the noise is drawn from a Gaussian distribution with a deviation equal to 10% of the magnitude of the true data. In particular, we have $\epsilon^D = 0.173$. To match the Gaussian likelihood, we use $W = \frac{1}{2(\epsilon^D)^2} = 16.70$ and the following loss,

$$l(\xi, d) = \|Du(x,\xi) - d\|_2^2.$$

The values of the known parameters are $\nu = 0.1$, $b_1 = -0.5$, $b_2 = -0.2$, $f(x) = 1$ and that of the unknown parameters are $\xi_1^* = 0.2$, $\xi_2^* = 0.7$. The prior distribution we assume are $\xi_1 \sim U[0,1]$, $\xi_2 \sim U[0,1]$. We use Algorithm 7 to compute the Gibbs posterior with $m = 100$ evolving particles, with local RB approximation accuracy set to be $1e-3$. In addition, as reference, we perform the standard Random Walk Metropolis-Hastings algorithm with Gaussian likelihood to obtain 5,000 samples from the posterior with 1,000 burn-in steps.

We show the comparison of our SMC result with the MCMC reference in Figure 5.1. Clearly, the SMC method perform similarly to the reference to approximate the posterior.
distribution. In particular, the SMC method took 3 iterations to reach the posterior, the evolution of the particles, the atoms of the local RB surrogate and the intermediate distributions are shown in Figure 5.2 to 5.5. As can be seen, as the weight $W$ is progressively increased, the particles cluster to the support of the posterior, and simultaneously leading the local RB surrogate to concentrate on the local support of the posterior as well.

We report the accumulative number of PDE solves at each SMC iteration in Figure 5.6. The local RB surrogate spend most of its computational effort in the first iteration as the particles move the most towards the posterior support. And once the local RB becomes accurate over the posterior region, further evolution of the particles rarely triggers the refinement of surrogate. The total number of PDE solves to obtain the posterior is around 200, representing a significant computational savings from the MCMC method.

![CDF plots for $\xi_1$ and $\xi_2$](a) CDF plot for $\xi_1$  \hspace{1cm} (b) CDF plot for $\xi_2$

**Figure 5.1:** Comparison of the posterior distribution of the parameters computed by Algorithm 7 and the standard MCMC method with Gaussian likelihood function.

### 5.6.2 2D advection diffusion equation

In the second example, we consider the simultaneous identification of the diffusivity constant and unknown source for a 2D advection-diffusion problem. Let $D = (0,1)^2$. We consider
(a) The true parameters (black), the particles (red) and the atoms for local RB (blue)

Figure 5.2: SMC iteration 0, with a loss weight $W_0 = 0$.

Figure 5.3: SMC iteration 1, with a loss weight $W_1 = 0.0782$.

the following problem,

$$-\nabla \cdot (\kappa(\xi^*) \nabla u(x, \xi^*)) + v(x) \cdot \nabla u(x, \xi^*) = f(x, \xi^*) \quad x \in D \quad (5.41a)$$

$$u(x, \xi^*) = 0 \quad x \in \Gamma_d \quad (5.41b)$$

$$\kappa(\xi^*) \nabla u(x, \xi^*) \cdot n = 0 \quad x \in \Gamma_n \quad (5.41c)$$
where $\Gamma_d := [0, 1] \times \{0\}$ and $\Gamma_n := \partial D \setminus \Gamma_d$. The unknown parameters $\xi^*$ are included in the diffusivity constant $\kappa(\xi^*)$ and the source term $f(x, \xi^*)$.

In particular, the diffusivity is modeled by

$$\kappa(\xi^*) = 0.02 + 0.98\xi_1^* \quad (5.42)$$
The advection field is divergence free and is defined by
\[
  v(x) = 13 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + 9 \begin{pmatrix} -x_1 \\ x_2 \end{pmatrix}.
\] (5.43a)

Finally, the forcing term \( f \) is modeled by two Gaussians function with unknown magnitudes, i.e.,
\[
  f(x, \xi) = 10 \exp \left( -\frac{(x_1 - 0.25)^2 - (x_2 - 0.5)^2}{0.25^2} \right) \xi_2^* + 5 \exp \left( -\frac{(x_1 - 0.75)^2 - (x_2 - 0.75)^2}{0.333^2} \right) \xi_3^*
\] (5.44)

The goal is to identify \( \xi^* \) from noisy measurement of the PDE solution \( u(x, \xi^*) \). Again, we are able to measure the solution over a uniform grid in the domain. Our noisy data is hence given by
\[
d = Du + \epsilon
\]
where \( D \) is an operator that maps the solution \( u(x, \xi^*) \) to the measurement and \( \epsilon \) is a noise vector that contains i.i.d entries. We assume the noise is drawn from a Gaussian distribution with a deviation equal to 20\% of the magnitude of the true data. In particular, we have \( \epsilon^D = 0.0197 \). For this problem, we use the following \( l_1 \) loss:
\[
l(\xi, d) = \|D u(x, \xi) - d\|_{l_1}.
\]
and selects the weight $W$ using the approach outlined in Section 5.5. After the determination of the $W$, we compare the SMC method in Algorithm 7 to a Random Walk Metropolis-Hastings method using $\exp(-Wl(\xi))$ as the likelihood (the Gibbs posterior is invariant with respect to the MC transitions).

The true values of the unknown parameters are $\xi_1^* = 0.1$, $\xi_2^* = 0.7$, $\xi_3^* = 0.5$. The prior distribution we assume are $\xi_1 \sim \beta(1, 2)$, $\xi_2 \sim \beta(3, 1)$, $\xi_3 \sim \beta(3, 1)$ and that they are independent. The final weight selected is $W = 25.8$, representing approximately $1/50$ of $\frac{1}{2(e(D))}$. We use Algorithm 7 to compute the Gibbs posterior with $m = 100$ evolving particles.

In addition, when training the local RB surrogate model at each SMC step $t$, we employ an adaptive accuracy $\epsilon_{\text{thre}}$ that is equal to $2\%$ of the standard deviation of $\{l(\xi_1^t)\}_{t=1}^m$. We run the reference MCMC method to obtain 5,000 samples from the posterior with 1,000 burn-in steps.

In Figure 5.7, we show the true diffusivity, advection and source fields. In Figure 5.8 we show the true PDE solution, the corrupted solution and the measurement points. We show the comparison of our SMC result with the MCMC reference in Figure 5.9. Again, the SMC method perform similarly to the reference to approximate the posterior distribution. $\xi_3$ has the most posterior uncertainty due to the fact that the solution, hence the data, has least sensitivity with respect to change in $\xi_3$, as the source associated with $\xi_3$ is located at the down-flow of the domain and has limited impact over $D$.

The SMC method took 3 iterations to reach the posterior, in Figure 5.10, we show the evolution of particles as well as the local RB atoms throughout the SMC iterations. Clearly, the particles and local RB atoms simultaneously evolve towards the support of the posterior, leading to improved approximation of the posterior distribution. In addition, as the particles become more clustered, the variation of $l(\xi)$ over the particles becomes lower, leading to smaller $\epsilon_{\text{thre}}$ (higher accuracy requirement on the surrogate).

Finally, we show the accumulative number of PDE solves at each iteration in Figure 5.11. Note that we did not include the PDE solves in the preprocessing step to select $W$, and we reinitialized the local RB surrogate model before computing the posterior under
the final weight. We do this to demonstrate how the computational efforts of local RB are distributed in the SMC iterations. The number of PDE solves (local RB atoms) depends critically on $e_{\text{thre}}$ in that iteration. In the first iteration, because $e_{\text{thre}}$ is relatively large, only about 100 PDE solves are incurred. In the latter iterations, $e_{\text{thre}}$ becomes lower and the accuracy requirement imposed on $\tilde{f}(\xi)$ becomes tighter as well. On the other hand, the particles become more compact in the latter iterations. Though the decreased $e_{\text{thre}}$ demands more PDE solves to refine the surrogate model, the increased compactness of the particles makes it easier for the local RB surrogate to reach the accuracy requirement. These two competing factors jointly determine the number of additional refinements on the surrogates. Overall, only less than 400 PDE solves are incurred in the SMC method to obtain the approximate posterior, representing a significant computational saving from the MCMC reference.

![Image](image1.png)

(a) Constant diffusivity field  (b) Advection field  (c) Source

**Figure 5.7**: The diffusivity, advection and source fields of the 2D advection-diffusion equation.

### 5.6.3 2D elasticity equation

In the last example, we consider two simple elastography problems where we need to infer the mechanical properties of materials from their deformation under given loads. These problems possess higher parameter dimensions and hence are more expensive to solve. Let
Figure 5.8: The solution and the noisy measurement.

Figure 5.9: Comparison of the posterior distribution of the parameters computed by Algorithm 7 and the standard MCMC method.

\[ D = (0, 1)^2. \] We consider the following linear elasticity problem,

\[-\nabla \cdot \sigma(x, \xi^*) + f = 0, \quad x \in D \quad (5.45a)\]

\[ \epsilon(x, \xi^*) = \frac{1}{2}(\nabla u(x, \xi^*) + \nabla u(x, \xi^*)^T), \quad x \in D \quad (5.45b)\]

\[ \sigma(x, \xi^*) = C(\xi^*) : \epsilon(x, \xi^*), \quad x \in D \quad (5.45c)\]

\[ u(x, \xi^*) = 0 \quad x \in \Gamma_d \quad (5.45d)\]

\[ \sigma(x, \xi^*) \cdot n = \tau \quad x \in \Gamma_n \quad (5.45e)\]

where \( \Gamma_d := [0, 1] \times \{0\} \) and \( \Gamma_n := \partial D \setminus \Gamma_d \). The unknown parameters \( \xi^* \) are included in the modulus of the material, which is part of the elasticity tensor \( C(\xi^*) \). We consider isotropic
Figure 5.10: The evolution of the particles and the local RB atoms at each iteration of SMC algorithm. The true parameter is in black, the particles are in red and the local RB atoms are in blue.

plane stress problems where we know the Poisson’s ratio $\nu = 0.3$ and try to identify the unknown Young’s modulus $E(\xi^*)$ from the noisy measurement of $u(x, \xi^*)$. The setup of the problem as well as two unknown modulus models we use are shown in Figure 5.12. The two problems have parameter dimensions of 5 and 9, respectively.

The displacement fields and the noisy measurements of the two models are shown in Figure 5.13 and 5.14, respectively. Note that we only use the noisy displacement data in the vertical, i.e., $x_2$, direction. We perturb the solution with 5%, 10% and 20% Gaussian
Figure 5.11: The accumulative number of PDE solves at each iteration of the SMC Algorithm 7 for the 2D advection diffusion problem.

Figure 5.12: The boundary condition, and true material properties for the simple elastography problems.

noise and investigate the posterior mean and deviation computed by the SCM method. For the prior distribution, we assume the parameters are independent and follow $\beta(1,3)$ distribution scaled to the range of $[0.1, 10]$. We use a simple $l_2$ loss function and weights $W = \frac{1}{2(e^{0.5})}$, which corresponds to the Gaussian noise model exactly.

We plot the inversion for the two models under different level of noise in Figure 5.15 and 5.16, respectively. The posterior mean gives reasonable approximations to the true modulus, and as the level of noise in the data increases, we have higher uncertainty about
our inverse solution, which is as expected. Finally, we present the number of local RB atoms used in each of the models with each level of noise in Figure 5.17. When the noise level is low, i.e., the weight for the loss is large, the posterior becomes increasingly concentrated in a small region within the support of the prior, and more SMC steps are needed to approximate the Gibbs posterior, leading to a larger number of refinements (atoms) for the local RB. Also noticed from the comparison is that when the dimension of the parameter space becomes high, as for the inclusion problem, higher computational cost is required to approximate the Gibbs posterior.

![Diagram](attachment:displacement_fields_and_noisy_measurements.png)

(a) Horizontal displacement  (b) Vertical displacement  (c) Measurement of the vertical displacement with 10% Gaussian noise

**Figure 5.13**: The displacement fields and the noisy measurements for the layered material.
(a) Horizontal displacement  (b) Vertical displacement  (c) Measurement of the vertical displacement with 10% Gaussian noise

**Figure 5.14:** The displacement fields and the noisy measurements for the material with a hard inclusion.

(a) 5% noise  (b) 10% noise  (c) 20% noise

**Figure 5.15:** The mean and standard deviation of Gibbs posterior computed using data with different levels of noise for the layered material.

(a) 5% noise  (b) 10% noise  (c) 20% noise

**Figure 5.16:** The mean and standard deviation of Gibbs posterior computed using data with different levels of noise for the material with a hard inclusion.
Figure 5.17: The total number of local RB atoms upon solving the Gibbs posterior for both material models and with different levels of noise (Case NO.).
Chapter 6

Conclusions

6.1 Summary and Conclusion

In Chapter 2, we have presented a new adaptive sampling approach for solving PDEs with random parameters based on a local reduced basis approximation. We denote the approach as the local RB method. It is the cornerstone of the entire thesis. The local RB method relies on a set of parameter atoms to generate an implicit Voronoi partition of the parameter domain and employs a local reduced basis projection to approximate the solution within each cell. The local basis in each cell is composed of the PDE solutions at proximal atoms as well as the local gradients of the PDE solution with respect to the parameters. The atoms in local RB are sequentially and adaptively selected based on an effective \textit{a posteriori error} indicator with a dynamic training set. We derived a rigorous error bound and carefully analyzed the computational cost for our local RB method. In addition, we compared our local RB method with two state-of-art adaptive sparse grid methods through numerical examples. We demonstrated that our method outperforms the others in terms of accurately approximating the PDE solution as well as the statistics of the solution when the PDE solution is highly sensitive to the random inputs.

We extended our local RB method to accurately approximate coherent risk measures evaluated at a quantity of interest (QoI) depending on the PDE solution in Chapter 3. The key to the extension is a modified error indicator that makes use of the dual representation of the coherent risk measures (see Equation (3.8)). The error indicator can be embedded into our local RB adaptive sampling framework and allows us to specifically tailor the local RB approximation to evaluate risks. With several examples, we showed that the proposed method indeed can accurately quantify risks associated with the tail distribution of the QoI.
with much lower computational cost than existing methods.

In Chapter 4, we have developed a computational framework based on local RB method to solve extremely computational challenging risk-averse optimization problems with PDE constraints. We leverage two essential components to achieve computational efficiency: (i) we employed the local RB method to build surrogate models to approximate the expensive objective value and gradient computations; and (ii) we employed an inexact trust region (TR) algorithm to control the surrogate approximation accuracy and achieve convergence. We derived error indicators to guide the adaptive refinement of the surrogate models in an efficient manner. Also, to ensure that our surrogate models rapidly satisfy the prescribed accuracy of the TR method, we employed local bases that were specifically tailored for these error requirements. In our numerical examples, we applied the TR/RB method to three risk-averse optimal control problems constrained by Helmholtz and advection-diffusion equations. We demonstrated the efficiency of our method in terms of number of full PDE solves required to achieve a highly accurate solution. We also demonstrated the effectiveness of our method for different risk measures in the objective. In all cases, our TR/RB approach obtained similar optimal controls with 1,000 times smaller number of full PDE solves than the Monte Carlo method. The cost reduction achieved by our method is significant.

Finally, we have proposed a particle-based approach with local RB surrogate model to approximate the Gibbs posterior for inverse problems in Chapter 5. Gibbs posterior has a particular advantage over the usual Bayesian approach, in the sense that it does not require an explicit model of the data generating mechanism (i.e., a likelihood function.) Instead, the Gibbs posterior are applicable where the unknown parameters are connected to the data through a loss function. It provides a more general framework for updating belief distributions where the true data generating mechanism is unknown or difficult to specify. We employed our local RB method to approximate the loss function in the Gibbs update formula. Based on a Sequential Monte Carlo (SMC) framework, we presented a method to progressive approximate the Gibbs posterior by simultaneously evolving the particles
and adapting the local RB surrogate model in a sequential manner. The emphasis of the local RB surrogate is navigated to a small fraction of the parameter space automatically by the evolving particles that progressively cluster over the support of the posterior. Computational savings are achieved thanks to the local accuracy and the efficiency of our local RB method. Indeed, once the local RB surrogate becomes accurate enough (specified by a parameter representing the approximation accuracy) over the local support of the posterior, further evolution of the particles takes minimal cost. Through several numerical examples that include advection-diffusion problems and elasticity imaging problems, we demonstrated the consistency of our method with the state-of-art Markov Chain Monte Carlo (MCMC) method. Furthermore, we showed that significant computational savings can be achieved to approximate the Gibbs posterior using our proposed method.

6.2 Possible future works

The work in this thesis is based on the key assumption of an affine decomposition of the parametrized bilinear form, i.e., Assumption 2. It is of great value to extend the current work to more general problems, including nonaffine, nonlinear and time-dependent PDEs. Methods exist in the literature to approximate nonaffine problems with affine counterparts, e.g., empirical interpolation method [BMNP04, CS10, DHO12]. To apply these methods in practice, however, it will be important to understand different sources of approximation errors and how to manage them to yield a converging approximation. In the optimization context, for example, we need to properly manage the approximation errors in order to satisfy the accuracy requirements imposed by the inexact trust region method (see Section 4.2.1). Such extension of the local RB method to general PDEs has yet to be studied and is the natural next step of the current work.

Another challenging extension of the current work is to apply the local RB method to problems with higher parameter dimensions. We have shown in our numerical example that, for mildly high dimensional problem (e.g., Section 2.6.3), the local RB method outperforms
other adaptive collocation methods by a large extent. The potential applicability of the local RB to high dimensional problems is warranted because the local RB atoms, which are the pivot of the local RB approximation, are samples of the parameters, and it is known that sample based approach, e.g., Monte Carlo, SROM [Gri09, Gri12], are relatively less affected by the curse of dimensionality. However, computational challenges do arise in high dimensional problems. For example, we note that the size of the local basis is proportional to the parameter dimension. For high dimensional problems, the reduced problems can be expensive to solve as well. To this end, one needs to apply a basis compression method, e.g., proper orthogonal decomposition (POD), to maintain the size of the reduced problems. This additional procedure inevitably introduces another level of approximation error that needs to be addressed. Another challenge that arises in high dimension is the implicit Voronoi partition we rely on to define the locality of the basis can become both expensive and ill-behaved. The simple $l_2$ metric we employ to define the partition of the parameter space become flat and ineffective in high dimensions. More effective and meaningful metric is hence required to measure the similarity between samples and to redefine the locality of the basis. The extension of the local RB method to high dimensional problems will be a significant contribution that can be pursued in the future.

Finally, the advantage of Gibbs posterior over the usual Bayesian approach for inverse problem requires further clarification through more concrete examples. It was shown in Chapter 5 that the Gibbs posterior is a generalization of the Bayesian approach that only requires a loss function instead of an exact data generating likelihood function. The need to specify an exact noise model is hence replaced by another supposedly simpler task to specify the weight for the given loss function. To this end, we proposed a simple approach to select such weight based on weaker assumptions of the noise. However, further investigations and examples are required to reveal the advantages of this new approach in the context of inverse problems, e.g., to demonstrate the robustness of Gibbs posterior to the potential bias that could be suffered by the Bayesian approach with a misspecified noise model. In addition, it would be significant if one can choose the weight purely based on the data,
instead of imposing additional assumptions of the noise. Finally, it would be interesting to apply the proposed SMC method to higher dimensional inverse problems, as the SMC framework we adopted are well suited to scale to higher parameter dimensions as shown in [KBJ14, BJMS15]. To this end, the local RB method needs to be modified/enhanced, same as the previous point, to retain computational efficiency in higher dimensions.
Appendix A

C++ implementation in the Rapid Optimization Library (ROL)

All examples in the previous chapters of this thesis are obtained with an efficient Matlab implementation of the local RB method that exploits the offline-online computing strategy for affinely parametrized PDEs. In addition, the proposed local RB method and its extension to solve risk-averse optimization problems have also been implemented by the author into the Rapid Optimization Library (ROL) [KRVBWV14], which is part of the Trilinos project [HBH+05] developed and maintained by the Sandia National Laboratories. This chapter briefly introduces the design structure and some implementation details of the local RB method. The implementations are based on the foundational components of the PDE-OPT application development kit in ROL that is specifically designed to solve large-scale PDE-constrained optimization problems.

We want to mention that the current implementation of the local RB method in ROL is for generic PDEs that can be nonlinear and nonaffine (as long as the PDE allows a residual-based error bound). Hence the implementation is not optimized for computational cost and does not take advantage of the affine decomposition even if a PDE does possess affinity. Future works are required in this aspect to optimize the implementation for computational efficiency and applicability to large-scale problems.

A.1 The local reduced basis method

A.1.1 The local reduced basis manager class

LocalRB_Manager is the main driver class for the local RB method in ROL. It stores and manages the local RB atoms and the associated local basis. It also performs adaptive sampling to construct the local RB surrogate. The key interface functions the LocalRB_Manager
provides are:

- **LocalRB_Manager():** this function is the constructor where all the algorithmic parameters for the local RB method are passed in from an input file; in addition, the constructor requires pointers to an object of the PDE interface (**PDE_Constraint_Extended**) and an object of the reduced space solver (**PDE_Constraint_RBCoeff**) in order to perform the computations required for the surrogate construction.

- **BuildModelWithNAtoms():** calling this function after initializing the **LocalRB_Manager** object will build the local RB surrogate using Algorithm 1 with a certain number of atoms specified by the user. All the computed atoms and local basis are stored as private data members of the **LocalRB_Manager**.

- **UpdateModelByNAtoms():** this function allows the user to sequentially build and maintain the local RB surrogate. User can update an existing surrogate by a specified number of new atoms that are adaptively selected using Algorithm 1.

- **RefineModelGivenTolerance():** this function builds/refines an existing local RB surrogate by adaptively adding more atoms till some user implemented model error indicator drops below a user specified threshold. This function is particularly useful, e.g., in the inexact trust region framework where the optimization algorithm dictates the accuracy that needs to be satisfied by the local RB surrogate.

- **ComputeSurroSolForward():** this is the function where the local RB surrogate model is employed to compute an approximate solution to the forward PDE at a user-input sample of the parameters. Essentially, this function uses the precomputed and stored data members of **LocalRB_Manager**, i.e., the atoms and basis, to form and solve a reduced problem at the given sample. Similarly, there is a function called **ComputeSurroSolAdjoint()** that uses the local RB surrogate to compute an approximate adjoint solution, which is relevant for optimization problems.

Some important private functions are also listed here:
• **GetAtomSolGrad()**: this function is used to obtain the PDE solution and its gradient at a given sample of the parameters from the PDE interface \texttt{PDE\_Constraint\_Extended}. The computed solution and gradient are the building blocks to form the local basis.

• **CreateBasisFromSolGrad()**: this function takes the solutions and gradients at the local RB atoms, and uses the neighbor information to form the local basis at each atom.

• **UpdateNeibStructure()**: this function computes/maintains the neighbor structure for the local RB atoms. Whenever a new atom is added to the local RB surrogate, this function needs to be called to update the neighbor structure.

• **UpdateTrSampIDandErr()**: this function evaluates the performance of the current surrogate model over the training set and computes the error indicators at each training sample, which are then used to select the next local RB atom to refine the model (see Algorithm 1).

• **LocalRBAdaptiveSamplingOneStep()**: this function performs one step of Algorithm 1 to select the next local RB atom and update the local RB model properly.

Finally, users can build a new class by inheriting from the \texttt{LocalRB\_Manager} and implement their own local basis and error criteria for adaptive sampling in order to tailor the local RB method to a more specific computational goal. Using different local basis can be achieved by overwriting the function \texttt{GetAtomSolGrad()}, e.g., one can include both forward solutions and adjoint solutions as in Algorithm 3. Using a different error criterion can be achieved by overwriting the function \texttt{UpdateTrSampIDandErr()} to implement a different residual-based error indicator for the training samples. The adaptive sampling procedure in local RB automatically calls \texttt{UpdateTrSampIDandErr()} and relies on these training error indicators for refinement.
A.1.2 The PDE interface class

PDE\textunderscore Constraint\_Extended, which inherits from the existing PDE\textunderscore Constraint class in PDE\text\(-\)OPT, is the PDE interface class that provides all the functions and information that LocalRB\_Manager requires from the underlying PDE model. Such information includes the Jacobian, the right-hand-side vector, the residual, the sensitivity of the PDE solution with respect to the parameters and etc..

When constructing the local RB surrogate, LocalRB\_Manager repetitively sets the value of the parameters for PDE\textunderscore Constraint\_Extended and requires PDE\textunderscore Constraint\_Extended to perform the full dimensional computations including assembling the full Jacobian, solving the PDE and its gradients, evaluating the norm of the residual for a given surrogate solution at the set values of the parameters. These are the most costly operations because all the computations happen in the full space. In addition, for optimization problems where solving the adjoint problem is also required by the LocalRB\_Manager, PDE\textunderscore Constraint\_Extended also provides the full dimensional adjoint computations based on a given quantity of interest (QoI). A list of the key interface functions of PDE\textunderscore Constraint\_Extended is given below:

- \texttt{setParameter}(): this function is used to set the values of the parameters for the underlying PDE model.
- \texttt{SetStateQoI}(): this function sets up a QoI for the adjoint problem. It is only used when adjoint computations are involved, as in the case of optimization.
- \texttt{SolveForwardProblem}(): this function solves the full forward problem.
- \texttt{SolveForwardGivenRHS}(): this function solves a linearized problem with a given right-hand-side vector at a given state.
- \texttt{SolveAdjointProblem}(): this function solves the adjoint problem at a given state. The right-hand-side vector for the adjoint problem is automatically derived from the state and the QoI.
- \texttt{SolveBasisAtSamp}(): this is the key function that computes the solution and its
derivative with respect to the parameters for the purpose of constructing the local basis in LocalRB_Manager. If a QoI is available for the use of mixed basis as in Algorithm 3, this function also computes the adjoint solution and its gradient. One can check the consistency of the computed derivatives by the finite difference method in the helper function CheckParamDerivative().

- **ComputeForwardResidualNorm()**: this is the function that evaluates the norm of the residual for a given surrogate solution. Similarly, there is a function that evaluates the residual for the adjoint problem **ComputeAdjointResidualNorm()**. The norm of the residual is the key component for the error indicators used in the LocalRB_Manager for adaptive sampling.

### A.1.3 The reduced space solver class

**PDE_Constraint_RBCoeff** is the class that contains the reduced space tools including forming the reduced Jacobian, solving the reduced problem and performing the bidirectional transformation of ROL vectors between full and reduced spaces. It also inherits from the **PDE_Constraint** class so that it has access to the underlying PDE model and the existing linear/nonlinear solvers. A list of the key interface functions of **PDE_Constraint_Extended** is given below:

- **setBasis()**: this function allows the LocalRB_Manager to set the local basis in PDE_Constraint_RBCoeff for reduced space computations.

- **TransformHighDimToReduced()** and **TransformReducedToHighDim()**: these two helper functions provide the bidirectional transformation of ROL vectors between full and reduced spaces.

- **RBSurrogateSolution()**: this function solves the reduced space problem and returns the surrogate solution at a given sample of the parameters using the local RB method.

- **RBAdjointSurrogateSolutionGivenRHS()**: this function computes the surrogate solution for the linearized problem with a given right-hand-side vector at a given state.
• RBAdjointSurrogateSolutionGivenRHS(): this function computes the surrogate solution for the adjoint PDE with a given right-hand-side vector at a given state.

Finally, we outline the inheritance and dependence structure for the relevant classes in Figure A.1.

Figure A.1: The inheritance and dependence structure of the local RB method in ROL. The white blocks are the ROL classes, the gray blocks are the developed classes.

A.2 Risk-averse optimization

To use the local RB method for risk-averse optimization problems, we follow Algorithm 3 and 4 in Chapter 4 closely. Algorithm 3 and 4 require their own versions of the local RB surrogate with different forms of local basis and goal-oriented error indicators. To this end, we inherit from the LocalRB_Manager to create two new derived classes called LocalRB_Manager_MixedBases_ForwardAdjoint and LocalRB_Manager_MixedBases_DoubleControls. The former is used to approximate the trust region subproblem model (i.e., $S_{\Lambda\text{mod}}$ in Algorithm 3), whereas the latter is used to approximate the objective function at two (possibly) different controls (i.e., $S_{\text{obj}}$ in Algorithm 4). As discussed in Chapter 4, these models use their own forms of local basis and error indicators that are specifically tailored to their own computational goals. These are realized by overwriting the functions GetAtomSolGrad() and UpdateTrSampIDandErr() in LocalRB_Manager, as discussed earlier. In the following content, we describe other important classes required for solving risk-averse optimization
problems with local RB surrogates and show the inheritance and dependence structure of the classes at the end of the section.

A.2.1 The local reduced basis PDE constraint class

PDE\_Constraint\_LocalRB is the PDE constraint class that interfaces with the SimOpt optimization algorithms provided by ROL. To this end, PDE\_Constraint\_LocalRB inherits from PDE\_Constraint and provides the same interface functions as PDE\_Constraint. The only difference is that the most expensive functions in PDE\_Constraint\_LocalRB class, e.g., solving the forward PDE, applying the inverse Jacobian and applying the inverse adjoint Jacobian, are carried out through a PDE\_Constraint\_RBCoeff object. That is, these functions use the local RB surrogate model to approximate the computations instead of using the full PDE model, leading to computational savings.

A.2.2 The risk measure tools class

The LocalRB\_RiskTools class provides the tools to compute the risk-related quantities in the error indicators (4.25) and (4.28). Both LocalRB\_Manager\_MixedBases\_ForwardAdjoint and LocalRB\_Manager\_MixedBases\_DoubleControls classes have a pointer to an object of LocalRB\_RiskTools. The functions in LocalRB\_RiskTools are hence directly accessible to the two classes for the local RB adaptive sampling. Different risk measures need to inherit from LocalRB\_RiskTools to specify their own implementations of the interface functions. The two most important interface functions in this class are

- **GetValAtQoI()**: recall the definition of optimized certainty equivalent in (3.3), this function simply evaluates the $v(G - t)$ term where $G$ is some given QoI and $t$ is a data member of LocalRB\_RiskTools that can be set according to the value of the current optimization variables.

- **GetDerivativeAtQoI()**: this is the function that computes $v'(G - t)$.  

149
A.2.3 The local reduced basis sampler class

To interface with the \texttt{StochasticObjective} class in ROL, samplers are required to provide sample average approximations (SSA) of the stochastic objective and gradient. In addition, sampler class also provides the interface to the inexact TR algorithm in ROL that allows the sampler to perform adaptive refinement. To this end, we build a \texttt{LocalRB\_Sampler} class that inherits from the \texttt{SampleGenerator} in ROL as a wrapper for the \texttt{LocalRB\_Manager} to interface with the \texttt{StochasticObjective} and the inexact TR method. In addition, for the TR/RB framework we proposed in Chapter 4, two sampler objects are required: one for the \texttt{LocalRB\_Manager\_MixedBases\_ForwardAdjoint} for gradient sampling, the other for the \texttt{LocalRB\_Manager\_MixedBases\_DoubleControls} for objective sampling. The key interface functions of the \texttt{LocalRB\_Sampler} are:

- \texttt{LocalRB\_Sampler()}: the constructor that defines the parameter samples to be used for the SSA of the stochastic objective (or gradient).
- \texttt{refine()}: this function, when called by the inexact TR algorithm with a prescribed accuracy, triggers the adaptive refinement function \texttt{RefineModelGivenTolerance()} in \texttt{LocalRB\_Manager} to update the local RB surrogate model till the model error indicator drops below the desired threshold. This function is the key for the TR algorithm to communicate with the local RB surrogate models.

The inheritance and dependence structure for the relevant classes for risk-averse optimization is outlined in Figure A.2.

A.3 An example

Finally, we show an example of the local RB method in ROL applied to a nonlinear advection diffusion equation. This nonlinear PDE does not possess affinity but it does satisfy a residual-based error bound due to the monotonicity of the nonlinear term. This example
Figure A.2: The inheritance and dependence structure of the local RB method for risk-averse optimization with PDE constraint in ROL. The white blocks are the ROL classes, the gray blocks are the developed classes.

is implemented in the `localrb-adv-diff-react-nonlinear` folder under the PDE-OPT application development kit.

We consider the following 1D nonlinear advection-diffusion equation with two parameters $\xi = [\xi_1, \xi_2]$:

$$
-\frac{\partial}{\partial x} \left( \mu(\xi) \frac{\partial u(x, \xi)}{\partial x} \right) + v \frac{\partial u(x, \xi)}{\partial x} + u(x, \xi)^3 = f(\xi) + z(x) \quad x \in D, \ \forall \ \xi \quad (A.1a)
$$

$$
u(0, \xi) = u(1, \xi) = 0 \quad \forall \ \xi \quad (A.1b)$$

where the diffusivity field $\mu(\xi)$ and the source term $f(\xi)$ are parametrized as:

$$\mu(\xi) = 10^{2\xi_1 - 3}, \quad f(\xi) = 2\xi_2 - 1,$$

and the control $z(x)$ is a continuous scalar field. The parameters are independent and follow uniform distribution in $[0, 1]$, that is $\xi_1, \xi_2 \sim U[0, 1]$. We first show the result of using local RB method to approximate the PDE solution with Algorithm 1. To this end, we show the training samples and atoms adaptively selected by the local RB method in Figure A.3, and the relative error of the local RB surrogate solution over 200 test samples.
using different number of atoms in Figure A.4. It is clear that with more refinement in the local RB surrogate, we are able to approximate the PDE solutions more accurately.

**Figure A.3:** Local RB atoms and the adaptive training samples with 5, 10 and 30 atoms respectively.

**Figure A.4:** The mean and standard deviation of the relative errors of surrogate solution over 200 test samples with different number of atoms in the local RB surrogate.

Next, we consider the QoI in (A.2) and solve the risk-averse control problem using CVaR$\beta$ as the risk measure as in Equation (A.3).

$$G(\xi; z) = \frac{1}{2}\|S(\xi; z) - 1\|_{L^2(X)}^2.$$  \hspace{1cm} (A.2)

$$J(z) = \text{CVaR}_\beta[G(S(\xi; z))] + \phi(z).$$  \hspace{1cm} (A.3)
where \( \varphi(z) = \frac{\alpha}{2} z^2 \) is the regularization term. We use sample average approximation based on 300 Monte Carlo samples to approximate the expectation in the CVaR. In addition, we also solve the same problem using a trust region method with full PDE solves as reference. The optimal control obtained in the reference solution and with the local RB method as well as the CDF and CVaR of the QoI under optimal control are shown in Figure A.5. The number of full PDE solves incurred in both methods are compared in Figure A.6. It can be seen that the local RB method achieves the same optimal control with an order of magnitude less full PDE solves than that is required for the reference solution. However, due to the fact the PDE is nonlinear and nonaffine and that our current implementation in ROL is not optimized for computational cost, solving a reduced problem in this case is almost as expensive as solving a full problem. The performance of local RB can be improved by a proper optimization of the current implementation.

![Optimal control](image1)

(a) Optimal control

![CDF of G(ξ) under optimal control](image2)

(b) CDF of G(ξ) under optimal control

**Figure A.5:** The reference and local RB solutions of the risk-averse control problem.

Finally, we use the local RB method to investigate the effect of risk-aversion for the above control problem. To this end, we compare the risk-averse control with a control obtained in a risk-neutral formulation (that is using \( \mathbb{E} \) instead of CVaR in the objective). The result is shown in Figure A.7. As expected, the risk-averse control provides much better control over the tail of the \( G(ξ) \) where the “loss” is large.
Figure A.6: Number of full PDE solves incurred in reference and local RB solutions.

Figure A.7: The reference and local RB solutions of the risk-averse control problem.

A.4 A brief comment on future development

The key problem to be addressed in the future development is an optimization and re-structuring of the current implementation to exploit the offline-online computing strategy for affinely decomposable PDEs. In addition, to make the implementation applicable to large-scale nonlinear nonaffine PDEs, methods like empirical interpolation [BMNP04, CS10, DHO12] need to be implemented as well to enable the offline-online computing strategy for such problems. Finally, an efficient QR solver can help with the conditioning of the reduced system by orthonormalizing the local basis in the offline stage.
Bibliography


<table>
<thead>
<tr>
<th>Reference</th>
<th>Authors</th>
<th>Title</th>
<th>Details</th>
</tr>
</thead>
</table>


164


Biography

Zilong Zou obtained his B.S. degree in Engineering Mechanics from Shanghai Jiao Tong University in 2010. After receiving the B.S. degree, he studied at the University of Tokyo, Japan, and earned his M.S. degree in Civil Engineering in 2012 under the guidance of Prof. Tomonori Nagayama and Prof. Yozo Fujino. After that, he was an intern at Kozo Keikaku Engineering, Tokyo and a Research Assistant with the Structural Monitoring and Control Center, Harbin Institute of Technology, China, for two years. He moved to North Carolina in August 2014 and started his Ph.D. study under Prof. Wilkins Aquino’s guidance at Duke University. He received his Ph.D. degree in Computational Mechanics along with a concurrent M.S. degree in Computer Science in December, 2018.

Zilong’s biggest hobby is soccer, which he has been playing since 1998. He feels very fortunate to have met a lot of great friends and teammates on the soccer field. In 2015, he was a key player on the Duke Chinese soccer team and helped the team to win the championship in the Duke University Intramural soccer league.

Zilong has published twelve journal and conference publications on various topics including Computational Mechanics, Uncertainty Quantification, Structural Health Monitoring and Wireless Sensors. A list of his publications is given below:


11. Yuequan Bao, Yan Yu, Hui Li, Xingquan Mao, Wenfeng Jiao, Zilong Zou, Jinping Ou. Compressive sensing-based lost data recovery of fast-moving wireless sensing for