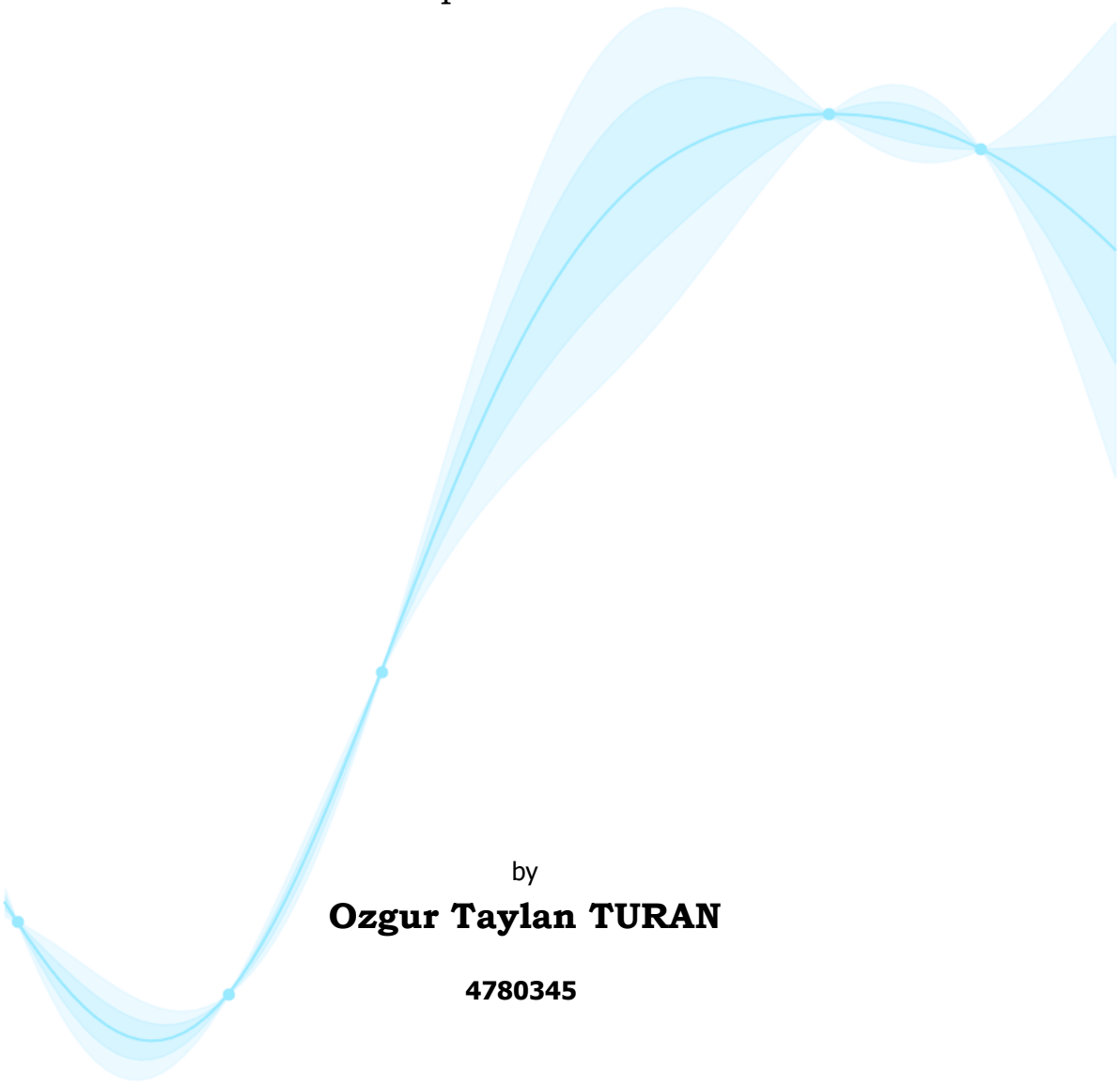
Additional Graduation Work

Reduced Order Model Base Creation with Bayesian Optimization

Reduced Order Model Base Creation with Bayesian Optimization



by

Ozgur Taylan TURAN

4780345

TU Delft Delft University of Technology

Supervisors:

Supervisor	
Dr. M.A. Bessa,	Delft University of Technology
Supervisor	
Dr. ir. F.P. van der Meer,	Delft University of Technology
Daily Supervisor	
Dr. I.B.C.M. Rocha,	Delft University of Technology

Summary

This research considers the *offline* training stage of the Reduced Order Models (ROM), that has been getting attention recently on the endeavor to come up with efficient solutions for the highly complex numerical models. In this work, a simply supported beam problem has been considered, for which a reduced basis creation has been investigated. Reduced basis creation is in utmost importance for the accuracy and reliability of the ROM. Main focus is on the efficient parameter sampling strategies to enrich the reduced basis, which brings forth computational burden. To decrease this burden, a statistical tool Gaussian Processes Regression (GPR) based Bayesian Optimization (BO) is utilized. These tools are used to create a surrogate function of error indicator that is used to select additional training points for ROM.

Results of this work show that randomness in the proposed procedure influences parameter sampling but does not have an impact on the overall accuracy. Finally, this work suggests further work on creation of a stopping criteria and finding a method of storing previous information and combining it with current information regarding training points without losing information. With the help of proposed further research topics, this work intends to be used as a foundation for efficient reduced basis construction.

Contents

List of Figures

List of Algorithms

1

Introduction

This work aims to create a framework for efficient and optimal reduced order basis construction in offline stage for accurate computational behaviour prediction in online stage. This is achieved by utilizing external libraries for Bayesian Optimization (BO), for the work that has been recently presented to construct reduced order basis in Knut Tjensvoll's thesis[2].

1.1. Background

Increasing computational power enabled the very demanding numerical mathematical numerical material models to be used as prediction tools has gained extraordinary growth in last decade [3]. Many recent applications in computational mechanics applications started to rely on these models [4].With increasing amount of computational demand, a need for decreasing that computational burden started to get more attention. Model Order Reduction (MOR) is one of the techniques that is used to decrease the computational demand of complex numerical models. This method replaces the high-fidelity Full Order Model (FOM) with a reasonably accurate Reduced Order Model (ROM). This is achieved by constraining the full order solution space to a lower dimensional space [4]. It should be noted that in the context of this work, full order solution space corresponds to nodal displacements and the lower dimensional space correspond to global displacement modes. Thus, the computational burden of the model in hand channels into finding the appropriate lower order solution space by thoroughly collecting the high-fidelity solutions and acquiring a reduced order basis by means of a data compression technique called Proper Orthogonal Decomposition (POD). This method is employed before the reduced order analysis, and called *offline* training stage and ensures full order solution space to be represented in terms of global modes.

1

In the work presented by Rocha et al. (2020) [1], the importance of sufficient training has been demonstrated on a simply supported beam problem as shown in Figure 1.1. In this demonstration, load has been shifted 5% of the total span and the resulting error in the inelastic regime was observed as 50% lower than the actual solution.

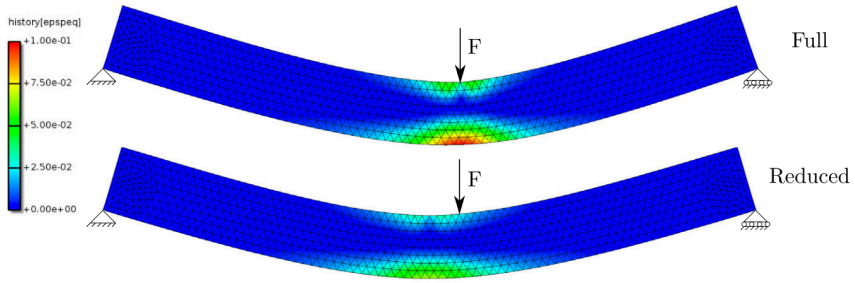


Figure 1.1: Importance of the Training Phase[1].

In the recently presented work of Tjensvoll (2019) [2], two methods, namely Surrogate Parameter Space (adopted from Goury [5]) and Gaussian Process Regression (GPR), have been proposed for sampling the parameter space to create a sufficient reduced order basis for the same simply supported beam problem. This work will focus on training the same simply supported beam problem presented, by using external GPR libraries.

1.2. Aim of the Research

The purpose of this research is to optimally develop the reduced order basis for the ROM to perform accurately. This work will utilize the information gathered from Tjensvoll (2019) [2] to come up with an efficient procedure to construct the basis for ROM, which could be utilized to decrease computational burden of multi-scale model implemented by Rocha [1]. The main focus will be on the GPR sampling procedure, which will be done through usage of BO operation with the help of GPR libraries of "scikit-learn" written in Python, while trying to overcome the shortcomings of the prior method presented by Tjensvoll (2019).

1.3. Outline

This research report is constructed as follows;

- Chapter 2 will introduce the background information that is crucial for the understanding of implementation, which are MOR and BO with GPR.

- Chapter 3 will focus on integration and implementation of presented information in Chapter 2.
- Chapter 4 the results of implementation will be presented and discussed.
- Chapter 5 conclusions of this research will be made with the recommendations for future research work.

2

Literature review

This research brings together two different topics, computational mechanics and machine learning. In this chapter, a brief background information will be provided for both of the topics, that is necessary to follow this research. All the detailed information can be traced back from the given references.

2.1. Model Order Reduction

In simpler terms model order reduction refers to the pursue of constructing a simpler model from the complex model in hand, where a simpler model is named ROM and the complex model named FOM. In this research endeavour, focus is mainly on projection based reduced order models as in the case of work published by Hernandez et al. (2017) [6]. In this work, unlike the published framework of Rocha et al. (2020) [1] only first reduction stage POD is used. This enables the collected snapshots, which are the full order model solutions for nodal displacement, to be compressed and projected to the reduced basis, which are the global modes of deformation.

2.1.1. Reduced Order Modeling

The following formulations are meant to depict a road map behind the concept, thus kept as compact as possible for the sake of brevity and can be followed thoroughly from [1] and [6].

Full Order Problem

Utilizing the Finite Element Method (FEM), a numerical approximation (weak form) of equilibrium of a volume Ω of the given problem with Dirichlet Γ_u and Neumann

Γ_f boundary conditions can be expressed as:

$$\boldsymbol{r} = \boldsymbol{f}^{\Gamma} - \boldsymbol{f}^{\Omega}(\boldsymbol{u}) = \boldsymbol{0} \tag{2.1}$$

where, $\boldsymbol{f}^{\Gamma}$ is the external force vector and $\boldsymbol{f}^{\Omega}(\boldsymbol{u})$ is the internal force vector depending on the nodal displacement field, that is coming from the conventional FEM formulations. The difference between internal and external force vectors has been defined as residual, $\boldsymbol{r}$. For the $\boldsymbol{f}^{\Omega}(\boldsymbol{u})$ calculation, numerical integration has been utilized. Since material response involves inelastic response an iterative solution procedure has been utilized to solve Equation 2.1 and the equilibrium is stated to be reached after the following expression has been satisfied:

$$\frac{||\boldsymbol{r}||}{\lambda} < tolerance \tag{2.2}$$

where λ is the scale factor computed from external and internal force vectors.

Dimensionality Reduction

As stated earlier the result of FEM is nodal displacements at every so-called time step (load displacement steps). These displacements are referred to as snapshots. Due to the fact that snapshots are correlated with each other, dimensionality reduction is possible and practical. The linear relationship between the snapshots brings forth the unnecessary information storage in the collection of snapshots. One of the most convenient and applied procedure for excessive information elimination from huge data sets is Proper Orthogonal Decomposition (POD) [7].

After collection of full order snapshots in the snapshot matrix $\boldsymbol{X}$, Singular Value Decomposition (SVD) could be performed on this matrix. It should be noted that SVD is in the broad sense finding the eigenvalues of a non-square matrix , since in the case of a square matrix this procedure results in eigenvalues and eigenvectors. Decomposition of a matrix by SVD is shown in Equation 2.3:

$$\boldsymbol{X} = \boldsymbol{\Phi S V}^T \tag{2.3}$$

where $\boldsymbol{S}$ representing singular values sorted from highest to lowest, $\boldsymbol{\Phi}$ representing reduced basis matrix that stores the aforementioned modes of displacement, and $\boldsymbol{V}$ named right singular matrix. It should be noted that singular values give out information regarding the energy content of the corresponding mode [8]. This is utilized to get rid of the aforementioned redundant information in the snapshot matrix. Thus, if full order equilibrium problem is of size N, after POD we can just store the $n < N$ modes in reduced basis. Note that, from now on reduced order basis will be represented by $\boldsymbol{\Phi} \in \mathbb{R}^{Nxn}$, which means the low energy modes

will not be stored. Then, displacement field reduced space could be expressed as follows:

$$\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{\alpha} \tag{2.4}$$

where $\boldsymbol{\alpha}$ is a set of latent variables. Thus, the full order equilibrium equation could be projected to reduced space as follows, by assuring orthogonality between $\boldsymbol{r}$ and $\boldsymbol{\Phi}$ (Galerkin projection):

$$\boldsymbol{\Phi}^T\boldsymbol{r} = 0 \tag{2.5}$$

After, imposing Equation 2.4 to 2.1, the residual defined in Equation 2.1 could be represented as:

$$\boldsymbol{r} = \boldsymbol{f}^{\Gamma} - \boldsymbol{f}^{\Omega}(\boldsymbol{\Phi}\boldsymbol{\alpha}) = \boldsymbol{0} \tag{2.6}$$

Remembering the main idea behind adaptively training ROM, which is minimizing the maximum projection error with a least squares like approach, ROM is fed with information where the a measure of projection error is maximum [5]. In this work that measure of error (error indicator), noting the fact that $\boldsymbol{r}$ is a function of $\boldsymbol{\Phi}$, is defined as:

$$\mathcal{R} = \sqrt{\frac{\sum_{i=0}^{N_t} ||r(t_i)||^2}{N_t + 1}} \tag{2.7}$$

where N_t is the number of time steps in the nonlinear solution procedure and $r(t_i)$ is the residual in the i^{th} time step.

It should be noted that the error indicator does not represent the real (exact) error. Exact error at nodes can be obtained by,

$$\mathcal{E} = ||\boldsymbol{u} - \boldsymbol{\Phi}\boldsymbol{\alpha}|| \tag{2.8}$$

2.2. Bayesian Optimization (BO)

BO have two main components namely surrogate model creation, which is in this work sachieved by GPR, and acquisition functions, which are used to select the the next point to be evaluated. In this section both of these components will be brifly explained. To find more in depth infromation, reader is recommended to refer to [9] and [10].

2.2.1. Gaussian Process Regression (GPR)

An assembly of random variables for which any finite number of have Gaussian Distribution represents a Gaussian Process and could be defined completely by its mean and covaraince function (kernel) as follows [9]:

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')) \tag{2.9}$$

where $\boldsymbol{x}$ is input vector. In simpler terms, when in function space setting, Gaussian Process is a distribution over functions with the given mean and covariance. In general, the mean of most of the Gaussian Processes can be safely defined as 0, but kernels need a more detailed look. It should be noted that from now on the mean will be considered as zero for all the GPR related subjects.

Kernels

In Gaussian Processes kernels define the similarity between two adjacent points, meaning points that are in the vicinity of each other posses information regarding each other . The most widely used kernel is the Squared Exponential kernel, expresed as [11]:

$$k_{SE}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 exp\left(-\frac{\boldsymbol{x} - \boldsymbol{x}'}{2l^2} \right) \tag{2.10}$$

where σ_f is the signal variance and represents the average distance of the function away from its mean which scales the kernel and l is length scale that determines the smoothness of the function. These free parameters are called hyperparameters and influence the behaviour of kernel enourmously [9].

Despite of its attractive properties, some argue that the strong smoothness of this kernel is not representative of many physical processes [9]. The Matérn kernel is one of the kernels that is being utilized to circumvent that bottleneck. It is expressed as:

$$k_M(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \left[\frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}(\boldsymbol{x} - \boldsymbol{x}')}{l} \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}(\boldsymbol{x} - \boldsymbol{x}')}{l} \right) \right] \tag{2.11}$$

where, ν and l are positive parameters and K_ν is a modified Basel function. The difference between the two given kernels could be observed in Figure 2.1.

As it can be seen from Figure 2.1, as ν increases the Matérn kernel becomes similar to Squared Exponential kernel and in theory when $\nu \to \infty$ it exactly becomes the SE kernel. Moreover, the increase in l is results in a more smoother function to be observed for SE kernel. For more information about kernels and their combinations Chapter 4 of [9] and Chapter 2 of [11] are detailed sources.

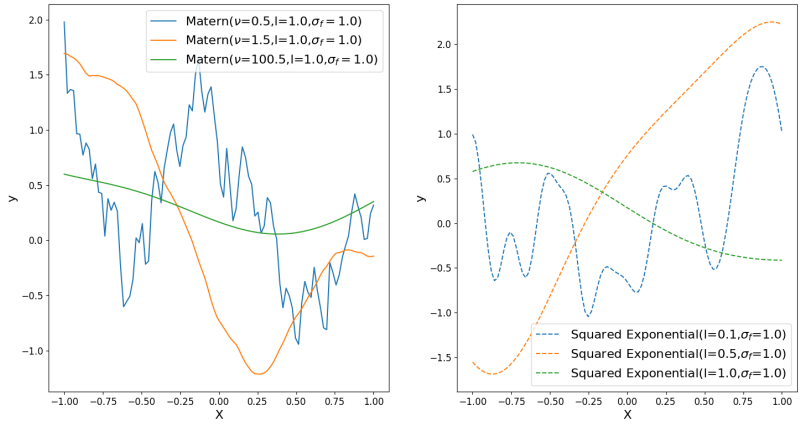


Figure 2.1: Brief comparison of kernels with changing parameters.

Prediction

The functions in Figure 2.1 are sampled randomly from a prior in which no underlying knowledge lies. Noting that $\boldsymbol{f}$ is training outputs and $\boldsymbol{f}_*$ is test outputs, the prior of the joint distribution can be expressed as follows:

$$p(\boldsymbol{f},\boldsymbol{f}_*)\sim\mathcal{N}\left(0,\ \begin{bmatrix} K(\boldsymbol{x},\boldsymbol{x}) & K(\boldsymbol{x},\boldsymbol{x}_*) \\ K(\boldsymbol{x}_*,\boldsymbol{x}) & K(\boldsymbol{x}_*,\boldsymbol{x}_*) \end{bmatrix}\right) \tag{2.12}$$

where $K(\boldsymbol{x},\boldsymbol{x})$ is the matrix of covariances evaluated for all pairs of training points.

To impose the prior knowledge to and get a posterior distribution over the functions it should be conditioned to contain only functions that pass through the observed points. Thus, the posterior predictive distribution can be expressed as,

$$p(\boldsymbol{f}_*|\boldsymbol{x}_*,\boldsymbol{x},\boldsymbol{f})\sim\mathcal{N}(K(\boldsymbol{x}_*,\boldsymbol{x})K(\boldsymbol{x},\boldsymbol{x})^{-1}\boldsymbol{f}, K(\boldsymbol{x}_*,\boldsymbol{x}_*)-K(\boldsymbol{x}_*,\boldsymbol{x})K(\boldsymbol{x},\boldsymbol{x})^{-1}K(\boldsymbol{x},\boldsymbol{x}_*))) \tag{2.13}$$

In a more realistic scenerio, the observed data can include some noise which can be expressed as $\epsilon\sim\mathcal{N}(0,\sigma_n^2)$, which leads to the addition of σ_n^2 to the diagonal of the $K(\boldsymbol{x},\boldsymbol{x})$ matrix [9]. An illustration of functions from conditioned data could be observed in Figure 2.2, where 3 sample functions have been drawn and possibilities of functions with 95% confidence are plotted from the given data without noise.

It should be noted that from this point on only the GPR representation on the right end side of Figure 2.2 plot will be utilized, in which mean($\boldsymbol{m}$) is represented with boldest blue and standard deviations $\boldsymbol{\sigma}$(70% confidence), $2\boldsymbol{\sigma}$(95% confidence) are

represented with light blue and lighter blue fills.

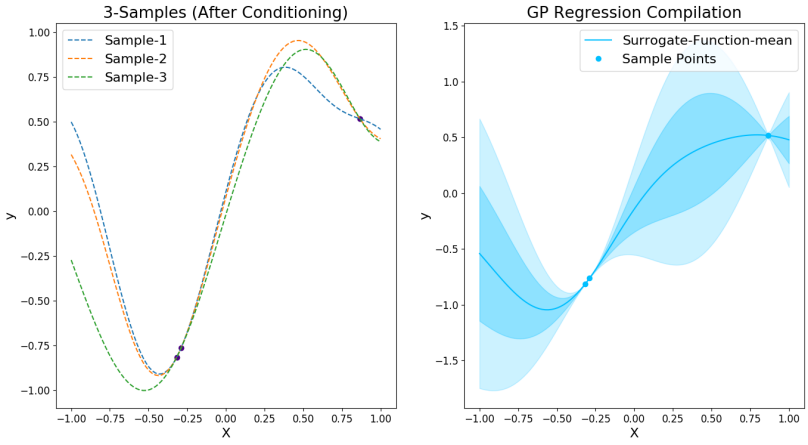


Figure 2.2: Sampling After Conditioning.

During the conditioning hyperparameters of the kernel has to be optimized for the provided data set. This is done by maximizing the log-marginal likelihoood (LML) function, expressed as:

$$logp(\boldsymbol{f}|\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{f}^T K^{-1}\boldsymbol{f} - \frac{1}{2}log|K| - \frac{n}{2}log2\pi \tag{2.14}$$

2.2.2. Acquisition Functions

As it can be seen from Figure 2.2 that GPs are able to offer an estimate of uncertainty over the estimated function. This can be utilized to balance between exploration and exploitation in the optimization procedure. Acquisition functions are the mechanicsm that implement this trade-off. Main purpose of the acquisition function is to guide searching procedure to the optimum of the surrugate function [10]. In this context exploration means directing search towards unknown regions of the surrogate function whereas exploitation means directing the search for the maximum chance to improve the current solution.

Probability of Improvement (PI)

PI is one of the first acquisition functions to be proposed and can expressed as:

$$PI(x) = p(f(x) \leq f(x^+) + \xi) = \mathbf{\Phi}\left(\frac{f(x^+) - m(x) - \xi}{\sigma(x)}\right) \tag{2.15}$$

where $\mathbf{\Phi}(\cdot)$ is the normal cumulative function, $m(x)$ and $\sigma(x)$ are the mean and the standard deviation of the surrogate function, $f(x^+)$ is the best observation so far and finally ξ is the term that is used to mitigate the exploitive nature of this acquisition function [10]. It should be noted that lower values of ξ elevate exploitation.

Expected Improvement (EI)

Because of the inability of PI to incorporate the potential magnitude of improvement, the following acquisition function has been proposed, which is expressed as:

$$EI(x) = \begin{cases} (f(x^+) - m(x) - \xi)\mathbf{\Phi}(Z) + \sigma(x)\phi(Z), & \text{if} \quad \sigma(x) > 0 \\ 0, & \text{if} \quad \sigma(x) = 0 \end{cases} \tag{2.16}$$

where,

$$Z = \begin{cases} \frac{f(x^+) - m(x) - \xi}{\sigma(x)}, & \text{if} \quad \sigma(x) > 0 \\ 0, & \text{if} \quad \sigma(x) = 0; \end{cases} \tag{2.17}$$

with the only extra term needed to be introduced being $\phi(\cdot)$ as the probability density function of the normal distribution [10]. EI automatically balances exploitation and exploration, however when tuning exploitation and exploration by hand we need ξ term.

Figures 2.3 and 2.4 illustrate how the optimum of a surrogate function is found by statistical tools with EI as acquisition function. As it can be observed after 2 iterations (7 observations) optimum of the objective function is confidently represented with the surrogate function as well.

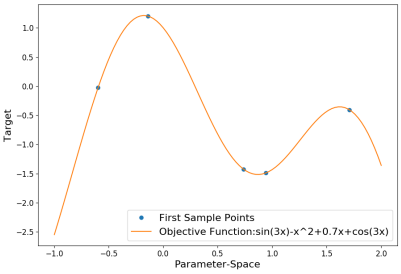


Figure 2.3: Objective function with initial samples.

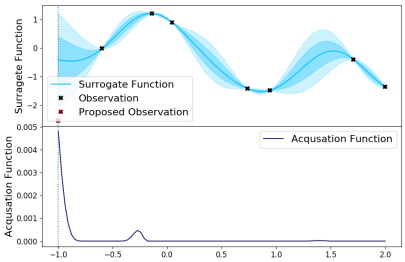


Figure 2.4: Optimum search 3rd iteration

3

Implementation of BO for ROM Training

The main objective of this research is to find a way to efficiently construct a reliable reduced basis for the simply supported beam problem shown in Figure 3.1. To achieve this objective main road map includes finding reliable starting points and stopping criteria for BO. Moreover, a trustworthy way to add the new training point information to the reduced basis is necessary, but not in the reach of this research. This research is constrained with the first two namely finding reliable starting points in parameter space and finding a reliable stopping criteria for the optimization procedure.

It should be noted that the reason for selecting a small parameter space is to avoid increased computation time due to SVD computation, since in this research truncated SVD is not utilized due to the possibility of observing high values of error indicator near supports as mentioned in [2]. Another reason is to be able to check the exact error ($\mathcal{E}(\mu)$) in a reasonable amount of time without GPR which is again implemented in the same research. Finally it should be noted that error indicator ($\mathcal{R}(\mu)$) need not to be calculated everywhere, since GPR will be utilized to get a surrogate version of it . However, for academic purposes it will be computed and shown in Chapter 4.

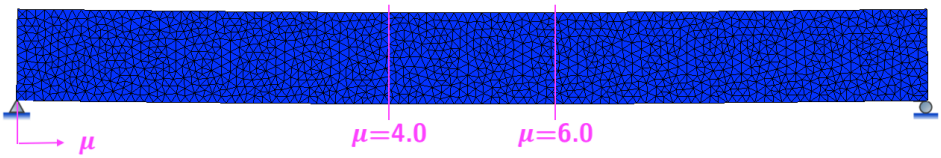


Figure 3.1: Simply supported beam problem (length=10 mm) with the constrained parameter space.

The MOR framework utilized in this implementation presented by Rocha et al. [3] and is written in C++ language using Jem/Jive toolbox. Moreover, the GPR tools of scikit-learn library developed in Python is utilized to develop the BO algorithms needed for surrogate function optimization.

The general framework for the overall training procedure is as follows:

Algorithm 1: Training Procedure

Prescribe Initial Training Points
while $\mathcal{E}(\mu = 4.0 - 6.0)$ > *tolerance* **do**
  Find $max(\mathcal{R}(\mu = 4.0 - 6.0))$ with BO
  Update the Reduced Basis($\mathbf{\Phi}$) with $max(\mathcal{R}(\mu = 4.0 - 6.0))$
  Run ROM for the Updated Reduced Basis($\mathbf{\Phi}_{\text{updated}}$)
  Calculate $\mathcal{E}(\mu = 4.0 - 6.0)$
end

Error Calculations have been conducted utilizing Python and the whole training procedure has been automatized using bash scripting language.

3.1. Bayesian Optimization (BO)

In this section, an explanation and verification of the BO procedure will be done for different types of functions without noise, since the error indicator is directly obtained in the ROM and does not involve noise. It should be noted that due to the mentioned limitation of PI, EI will be utilized as acquisition function in this implementation.

The Expected Improvement Algorithm is conducted as follows:

Algorithm 2: Optimization Procedure

Prescribe Initial Sampling Points
while *Stopping Criteria is not met* **do**
    Get the Surrogate Model for Error Indicator $\mathcal{R}$
    Search over parameter space for maximum EI and choose it as next point
    Compute Stopping Criteria
end

3.1.1. Initial Sampling

In the reasearch of [2], the initial sampling points are randomly selected and when looked in depth, it can be seen that some of the initial sampling points are really close to each other. It is evident the that initial sampling points should have space filling properties [12]. This is especially important if the evaluation of the sampled points is computationally expensive. Thus, in this work a simple idea based on the Latin hypercube sampling scheme will be utilized. The utilized scheme for this reasearch on one dimension is dividing the parameter space to the required number of equal domains and pick out random points from that subdomain. This procedure ensures the paramter space to be filled similar to the proposed scheme of [2].

3.1.2. Stopping Criteria

In most works on BO, the problems considered are generallly solved with fixed iteration numbers [13]. However, this method could be introducing unnecessary computations or under computation both of which will not be desired in this case of training procedure since the aim is to try to get the best training point with least amount of computational power. Lorenz et al. [14] has proposed 2 stopping critera, the first one being the Euclidean distance between consecutive point proposals and the other one being the hybrid stopping criteria which utilizes PI information, whiile EI was being utilized as acquisition function. In this research for, accuracy both of the approaches introduced and tested in [14] will be combined and utilized as stopping criteria in this work. Thus, when PI is below the prescribed limits and the search is directed to a point which is away from the previous prescribed point by a certain margin the optimum of objective function is considered to be found.

4

Results and Discussion

This chapter will introduce the results and discussion regarding the implementation of ROM training procedure with Bayesian Optimization. It should be noted that all the given model information has been introduced regarding the beam problem and the optimization procedure of a surrogate function, in Chapter 3. Moreover, the parameter space $\mu = 4.0 - 6.0$ has been represented on the x-axis of the resulting plots and numbered from $0 - 20$, where 0 corresponds to 4.0 and 20 corresponds to 6.0. Finally, it should be noted that the beam is only loaded at one point at all times, and for the optimization procedure, the same PI and Euclidean distance thresholds have been utilized as in presented cases of tests presented in this chapter

4.1. Testing the BO Procedure

In this section proposed BO procedure will be tested on two different types of functions. It should be noted that for the test all smooth functions have been utilized. This is the reason for SE kernel to be utilized with length scale bounds for the log-marginal likelihood maximization being 0.01 and 1 respectively. Moreover, for the stopping criteria the Probability of Improvement limit has been fixed at 0.01 and the Euclidean Distance limit is $\frac{1}{8}^{\text{th}}$ of the parameter space. Furthermore, it should be noted again that the acquisition function utilized is EI, but PI is utilized as a stopping criteria and in both of them $\xi = 0.1$ has been utilized to enable the BO to work in explorative behavior.

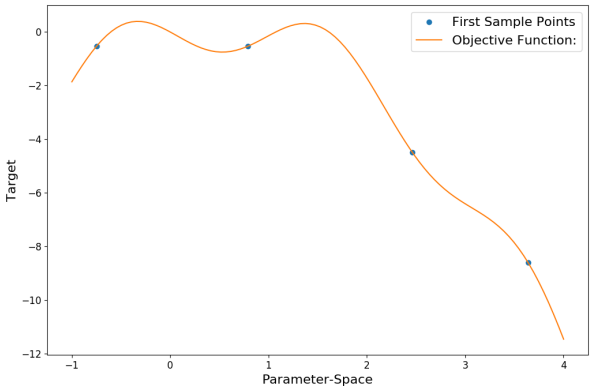


Figure 4.1: Objective-1 $f(x) = -sin(3x) - x^2 + x$ in $[-1, 4]$ interval, with two local maxima

4.1.1. Smooth Function with two Close Local Maxima:

The objective function has been selected as $f(x) = -sin(3x) - x^2 + x$ in $[-1, 4]$ interval, due to the two close local maxima as shown in Figure 4.1. Moreover, the space filling initial points can be observed from the same figure as well.

The Specified Bayesian Optimization procedure is able to come up with the global maximum with $8.87 * 10^{-8}$ error in just 4 iterations which means only 8 points have been evaluated in this function. Results after every iteration can be followed from Figures 4.2,4.3,4.4 and 4.5.

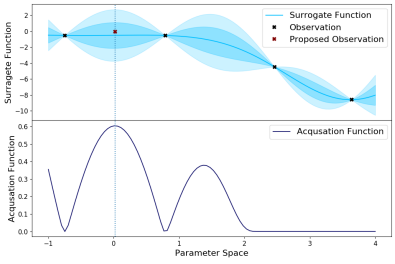


Figure 4.2: Surrogate Function for Objective-1 in 1^{st} Iteration

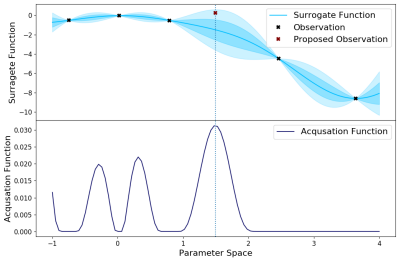


Figure 4.3: Surrogate Function for Objective-1 2nd Iteration

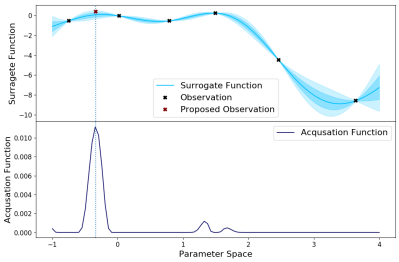


Figure 4.4: Surrogate Function for Objective-1 3rd Iteration

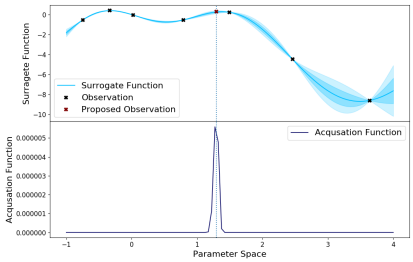


Figure 4.5: Surrogate Function for Objective-1 4th Iteration

4.1.2. Multiple Local Maxima with one Distinctive Global Maximum

To examine the reliability of the procedure it has been tested with the $f(x) = sinc(x)$ function, which involves multiple local maxima and one distinctive global maxima. The f unction in the interval of $[-5, 5]$ with initial sampling points can be observed in Figure 4.6.

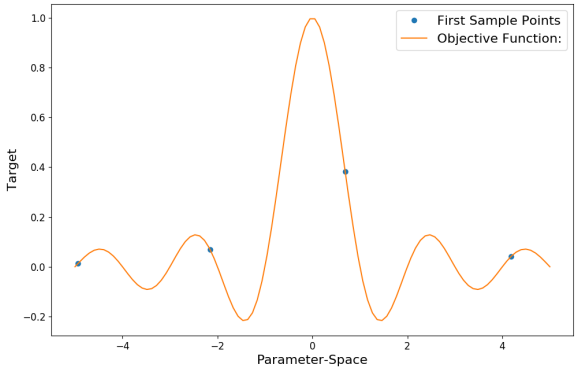


Figure 4.6: Objective-2 function $f(x) = -sin(3x) - x^2 + x$ in $[-1, 4]$ interval, with two local maxima

The importance of the space filling initial sampling and explorative search can be observed in this example. If there is no initial sample in the right arm of the global maximum search it would have been difficult for surrogate function to search that area. Again, with this test function optimization procedure is able to find the maximum within 4 iterations with an error of $1.16 * 10^{-6}$. Results for all iterations can be seen in Figures 4.7, 4.8, 4.9 and 4.10.

4.1.3. Conclusion of Tests

Besides the given test results multiple runs have been done for the same functions, because of the fact that initial samples are randomly selected from their respective subdomains. As a result of these runs it has been deduced that:

- Although both test functions found the maximum values in 4 iterations, this is not a general rule and for same function for different runs different iteration numbers can be observed.
- Each run was able to find the maximum with reasonable error values and number of iterations.
- There were some cases, but very rare, where unrepresentative versions of the GPR results were obtained for only one iteration. It has been found that reason behind this was hyperparameter tuning by log-marginal likelihood maximization. It should be noted that this one iteration does not cause any disturbance on the overall functioning of the optimization procedure.

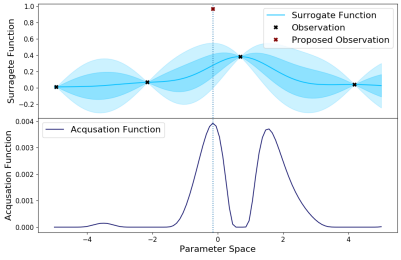


Figure 4.7: Surrogate Function Objective-2 in 1^{st} Iteration

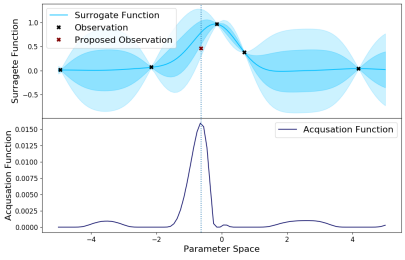


Figure 4.8: Surrogate Function Objective-2 2nd Iteration

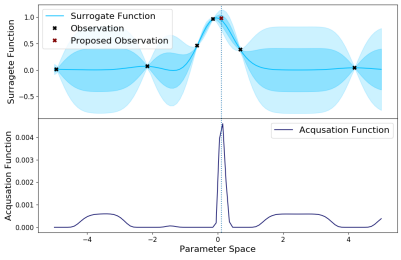


Figure 4.9: Surrogate Function Objective-2 3rd Iteration

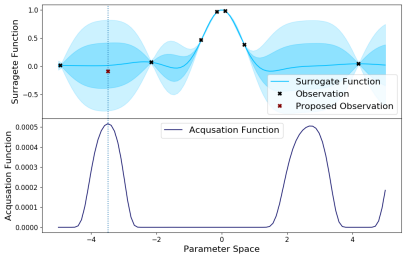


Figure 4.10: Surrogate Function Objective-2 4th Iteration

4.2. Automized Training

Before starting the automized training it is important to have an idea about exact error and error indicator, which can be used as an educated foundation to start training and choose a kernel that represents the points in parameters space better. For this reason, $\mathcal{R}$ and $\mathcal{E}$ are found by just training the ROM at $\mu = 5.0$.

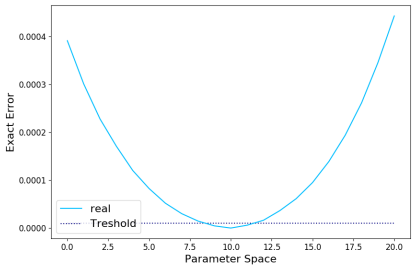


Figure 4.11: Exact Error ($\mathcal{E}$) for Training at $\mu = 5.0$

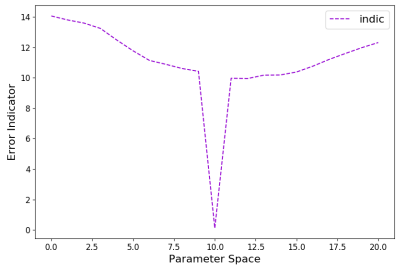


Figure 4.12: Error Indicator ($\mathcal{R}$) for Training at $\mu = 5.0$

As it can be seen from Figures 4.11 and 4.12, errors reach their max values at both ends of parameter space due to the nature of bending problem.

It should be noted that the exact error threshold is chosen by assuming the ROM is sufficiently accurate to represent the FOM on the training point. As it can be seen from Figure 4.11, the threshold is selected such that ROM trained at $\mu = 5.0$ is accurate enough compared to the FOM.

From this point on only the surrogate representation of $\mathcal{R}$ will be utilized to find the next training point and the procedure will end when the exact error goes under the prescribed threshold of 0.000012. Due to the sharp dropping nature of $\mathcal{R}$, the Matérn kernel with constant $\nu = 1.5$ will be utilized in GPR without noise. Results of surrogate $\mathcal{R}$, to find the maximum and add to the training space, can be observed in below figures.

As can be followed from Figures 4.13 to 4.22 , in the first iterations the proposed procedure has successfully found the maximum of surrogate $\mathcal{R}$. However, due to the randomness involved in the procedure during initial sampling, in some of the iterations the maximum is not found with the desired accuracy. The most clear example of this can be observed in Figure 4.18, in which because of the sampling in the far end of the parameter space the GPR gets over confident in that area and does not search that specific area, but the global maximum lies in that region for the given error indicator.

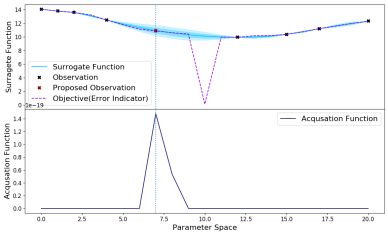


Figure 4.13: Iteration-1 ($\mu = [5.0]$)

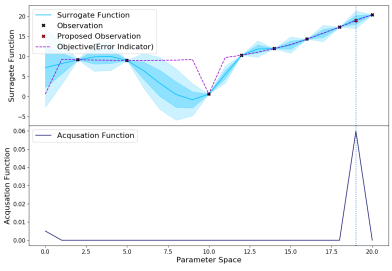


Figure 4.14: Iteration-2 ($\mu = [5.0, 4.0]$)

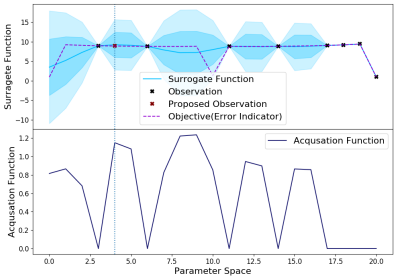


Figure 4.15: Iteration-3 ($\mu = [5.0, 4.0, 6.0]$)

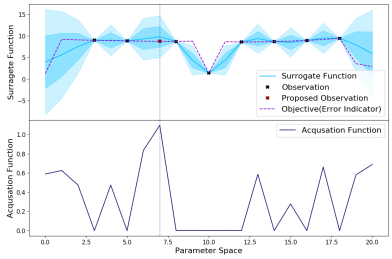


Figure 4.16: Iteration-4 ($\mu = [5.0, 4.0, 6.0, 5.9]$)

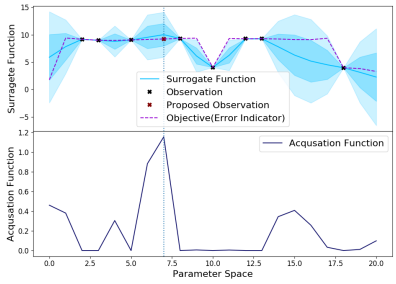


Figure 4.17: Iteration-5 ($\mu = [5.0, 4.0, 6.0, 5.9, 5.8]$)

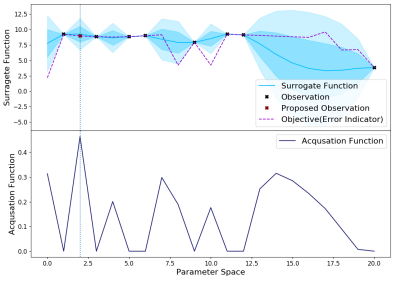


Figure 4.18: Iteration-6 ($\mu = [5.0, 4.0, 6.0, 5.9, 5.8, 4.8]$)

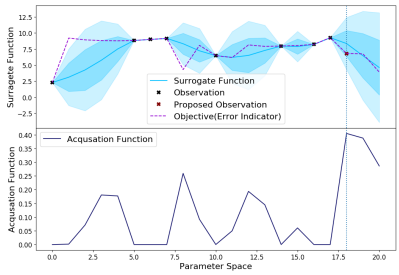


Figure 4.19: Iteration-7
($\mu = [5.0, 4.0, 6.0, 5.9, 5.8, 4.8, 5.1]$)

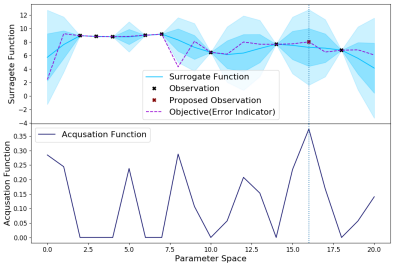


Figure 4.20: Iteration-8
($\mu = [5.0, 4.0, 6.0, 5.9, 5.8, 4.8, 5.1, 5.7]$)

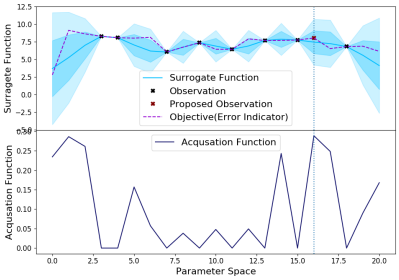


Figure 4.21: Iteration-9
($\mu = [5.0, 4.0, 6.0, 5.9, 5.8, 4.8, 5.1, 5.7, 4.7]$)

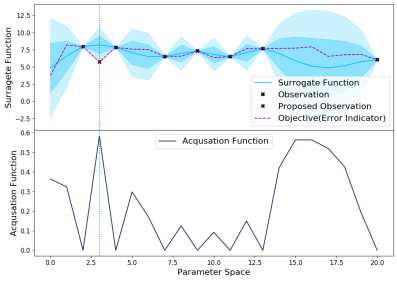


Figure 4.22: Iteration-10
($\mu = [5.0, 4.0, 6.0, 5.9, 5.8, 4.8, 5.1, 5.7, 4.7, 4.3]$)

Another important result that can be observed is the importance of the minimization technique that has been utilized. In this reasearch all the optimizations (maximizations) have been conducted using "L-BFGS-B" method has been utilized. Although very homogenous starting points has been given to the optimizer with enough restarts, there were some cases where this procedure has not come up with the global maximum. Best example of this could be observed in Figure 4.15, where the real maximum has not been found in the acquisition function. This has some potential consequences, if this type of small errors occur in an iteration leading to bad decision making. However, in this case there is no observable effects of this error.

Three drawbacks of the selected minimization technique is mentioned in [15] which are expressed as:

- Not being rapidly convergent,
- Inability to give accurate results on ill conditioned problems and

- No acceleration being possible with the information given the structure of the problem to be minimized.

Moreover, observing the wrong maxima can be caused by the fact that the acquisition function is not smooth for the given grid spacing as well.

Again as it can be observed from Figure 4.22 the focus of training has been focused mostly on the ends due to the nature of the $\mathcal{R}$ distribution on parameter space, with some training point selection coming near the middle of the parameter space.

The exact error progression for the given run is as shown in Figures 4.23 to 4.32.

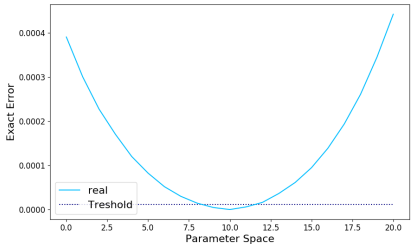


Figure 4.23: Iteration-1 ($\mu = [5.0]$)

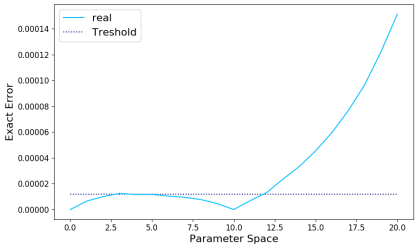


Figure 4.24: Iteration-2 ($\mu = [5.0, 4.0]$)

The most observable feature of the exact error propogasion through 10 iterations is the fact that $\mathcal{R}$ and $\mathcal{E}$ are similar in the first iterations, where both of them get their maximum at both ends of the parameter space. The right choice of the maximum of surrogate $\mathcal{R}$ in the first 2 consecutive iterations enables the $\mathcal{E}$ to get in to a reasonable scale for the selected parameter space, as it can be seen in Figure 4.25. However, after this point the corroletion between $\mathcal{R}$ and $\mathcal{E}$ decreases. It is again evident that addition of another training point results in decreasing exact error in the near neighbourhood of the training point. This result has been mentioned in

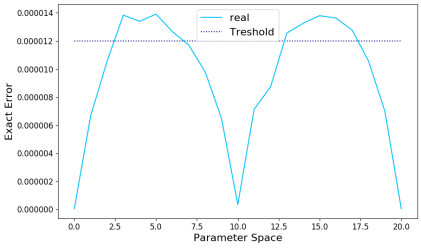


Figure 4.25: Iteration-3 ($\mu = [5.0, 4.0, 6.0]$)

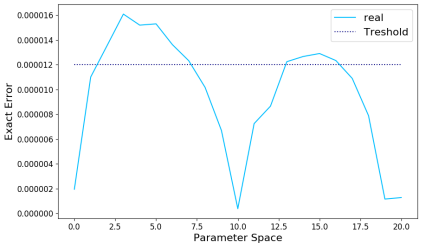


Figure 4.26: Iteration-4 ($\mu = [5.0, 4.0, 6.0, 5.9]$)

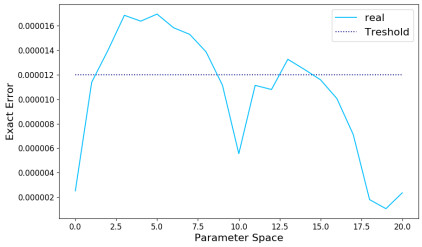


Figure 4.27: Iteration-5
($\mu = [5.0, 4.0, 6.0, 5.9, 5.8]$)

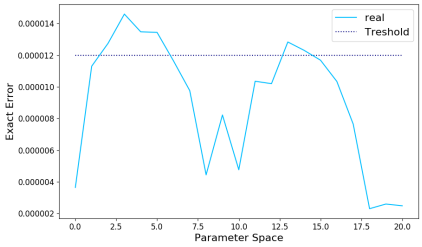


Figure 4.28: Iteration-6
($\mu = [5.0, 4.0, 6.0, 5.9, 5.8, 4.8]$)

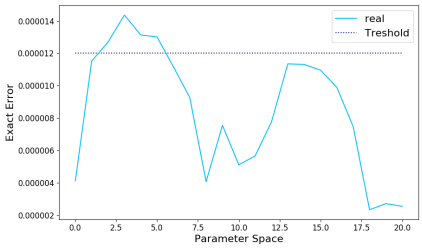


Figure 4.29: Iteration-7
($\mu = [5.0, 4.0, 6.0, 5.9, 5.8, 4.8, 5.1]$)

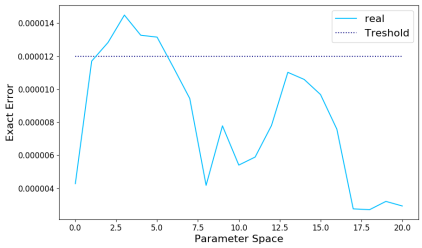


Figure 4.30: Iteration-8
($\mu = [5.0, 4.0, 6.0, 5.9, 5.8, 4.8, 5.1, 5.7]$)

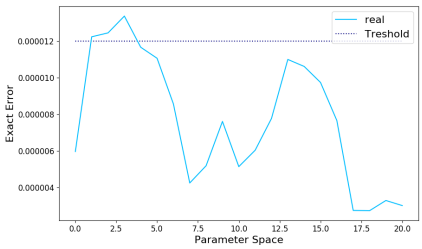


Figure 4.31: Iteration-9
($\mu = [5.0, 4.0, 6.0, 5.9, 5.8, 4.8, 5.1, 5.7, 4.7]$)

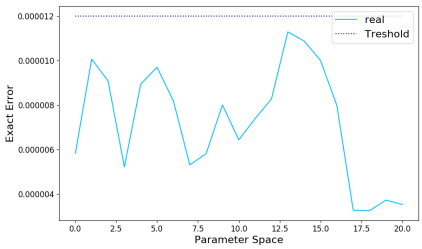


Figure 4.32: Iteration-10
($\mu = [5.0, 4.0, 6.0, 5.9, 5.8, 4.8, 5.1, 5.7, 4.7, 4.3]$)

the findings of [2] as well.

As it can be seen from Figure 4.32 after training with 10 points, the exact error goes below the prescribed threshold, successfully, although there were some problems with the BO procedure. This is a strong indication for the applicability of proposed procedure.

Due to the random nature of the procedure, multiple runs have been conducted and in all of them the exact error was able to be lowered below the prescribed threshold. However, it should be noted that selected points and their selection order changed between runs.

5

Conclusions and Recommendations

The main focus of this research was the implimention of a BO procedure utilizing existing GPR libraries into the MOR technique procedure developed in [1] and [2]. Implementation of this procedure has been used to sample the parameter space efficiently to construct the reduced basis. The main reason for utilizing BO for this procedure was to keep the evaluation of the ROM at minimum while having a good idea about error indicator that exist in the whole parameter space. Upon computational simulation, better understanding of the behavior of proposed method and its implementation in the reduced basis construction has been developed. The conclusions reached with this research are as follows:

- The randomness introduced in the proposed procedure affects the direction of parameter sampling. This in turn, might give rise to increased computational time or inefficient training point selection. However, the obtained accuracy is not affected by this, since the training point decreases the exact error not only at the point, but also in its near vicinity.
- Although, space filling initial sampling is utilized, due to the nature of the error indicator in the given parameter space, it has been observed that the BO accuracy is highly determined by the initial sampling points. Again, this does not influence the outcome. However, more representative initial sampling of the parameter space could be more efficient.
- Since the proposed truncation method in [2] has not been utilized, disruptions in the error indicator was not observed, which makes this method applicable to a wider range in this problem. However, increasing the size of the snapshot matrix, makes it time consuming to perform SVD procedure, which makes it impractical to be utilized in more dense parameter spaces.

- Compared to [2], different acquisition function and stopping criteria for BO procedure have been utilized. It has been seen that the proposed methods can be utilized in the same sense to get the desired reduced basis.
- It has been observed that correlation between error indicator and the exact error is changing as the training points for the reduced basis increases.

For implementation of this work to the Model Order Reduction Framework further improvements are needed. These improvements and some other recommendations are as follows:

- Initial sampling dependency should be addressed and other types of space filling random points selection techniques could be tested.
- A stopping criteria should be implemented that does not involve the FOM solutions that have been utilized in this research, since getting FOM solutions is against the prime objective of MOR.
- Different than the case of [2], a way to compress, store and combine the information gathered from each step in an efficient way without loosing information should be sought for general applicability of this automatized procedure to be valid.
- Different types of kernels or combinations of kernels could be tested for capturing the relations between the data points of error indicator in the parameter space more accurately.
- The effect of initial sampling number to the efficiency could be sought.
- For this research no information regarding the exact error has been utilized other than for stopping criteria. Since FOM solutions are known from the training points, this information could be utilized for the next training point selection.
- Since the parameter space have been selected small, a bigger parameter space can be investigated for the same proposed framework.

References

[1] I. Rocha, F. van der Meer, and L. Sluys, *An adaptive domain-based POD/ECM hyper-reduced modeling framework without offline training,* Computer Methods in Applied Mechanics and Engineering **358**, 112650 (2020).

[2] K. Tjensvoll, *Optimizing the Reduced Basis Construction for Reduced-Order Mechanical Models(Msc. Thesis)* (2019).

[3] I. Rocha, *Numerical and Experimental Investigation of Hygrothermal Aging in Laminated Composites*, Ph.D. thesis, Delft University of Technology (2018).

[4] I. B. Rocha, F. P. van der Meer, S. Raijmaekers, F. Lahuerta, R. P. Nijssen, L. P. Mikkelsen, and L. J. Sluys, *A combined experimental/numerical investigation on hygrothermal aging of fiber-reinforced composites,* European Journal of Mechanics, A/Solids **73**, 407 (2019).

[5] O. Goury, D. Amsallem, S. P. A. Bordas, W. K. Liu, and P. Kerfriden, *Automatised selection of load paths to construct reduced-order models in computational damage micromechanics: from dissipation-driven random selection to Bayesian optimization,* Computational Mechanics **58**, 213 (2016).

[6] J. A. Hernández, M. A. Caicedo, and A. Ferrer, *Dimensional hyper-reduction of nonlinear finite element models via empirical cubature,* Computer Methods in Applied Mechanics and Engineering **313**, 687 (2017).

[7] J. A. Hernández, J. Oliver, A. E. Huespe, M. A. Caicedo, and J. C. Cante, *High-performance model reduction techniques in computational multiscale homogenization,* Computer Methods in Applied Mechanics and Engineering **276**, 149 (2014).

[8] R. A. van Tuijl, J. J. Remmers, and M. G. Geers, *Integration efficiency for model reduction in micro-mechanical analyses,* Computational Mechanics **62**, 151 (2018).

[9] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning series)* (2006).

[10] F. Archetti and A. Candelieri, *Bayesian Optimization and Data Science* (2019).

[11] D. K. Duvenaud, *Automatic Model Construction with Gaussian Processes,* (2014).

[12] M. Schonlau, *Computer Experiments and Global Optimization* (1997).

[13] M. McLeod, M. A. Osborne, and S. J. Roberts, *Optimization, fast and slow: Optimally switching between local and Bayesian optimization,* 35th International Conference on Machine Learning, ICML 2018 **8**, 5521 (2018), arXiv:1805.08610 .

[14] R. Lorenz, R. P. Monti, I. R. Violante, A. A. Faisal, C. Anagnostopoulos, R. Leech, and G. Montana, *Stopping criteria for boosting automatic experimental design using real-time fMRI with Bayesian optimization,* , 1 (2015), arXiv:1511.07827 .

[15] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, *L-BFGS-B - Fortran Subroutines for Large-Scale Bound Constrained Optimization,* ACM Trans Math Softw **23**, 550 (1994).