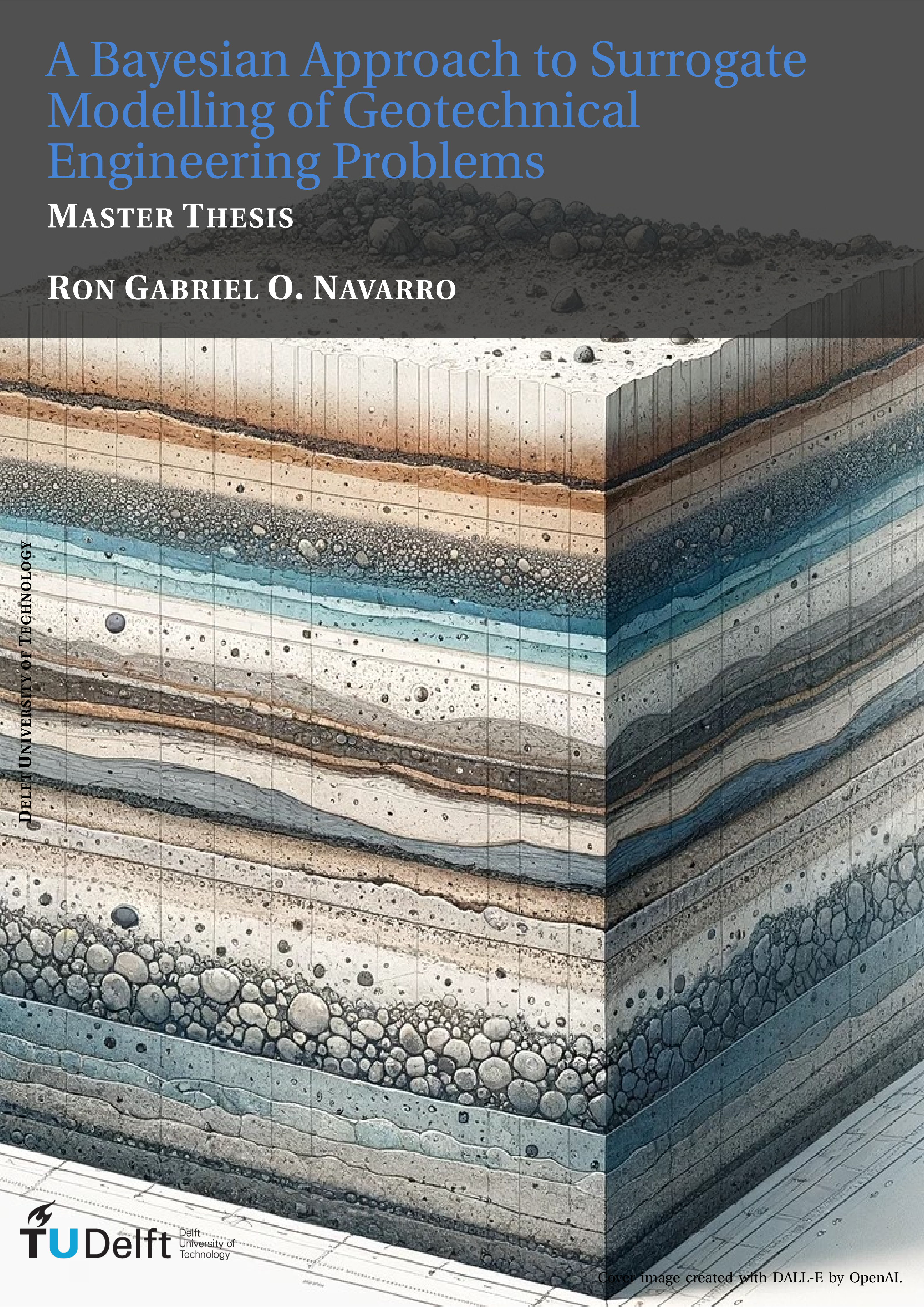
A Bayesian Approach to Surrogate Modelling of Geotechnical Engineering Problems

Master Thesis

Ron Gabriel O. Navarro



Delft University of Technology

Cover image created with DALL-E by OpenAI.

A Bayesian Approach to Surrogate Modelling of Geotechnical Engineering Problems

Thesis report

by

Ron Gabriel O. Navarro

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on July 16, 2024 at 10:00

Thesis committee:

Chair:	Dr. Michael Hicks
Supervisor:	Dr. Bram van den Eijnden
External examiner:	Dr. Iuri Rocha
Place:	Faculty of Civil Engineering and Geosciences, Delft
Project Duration:	February, 2024 - July, 2024
Student number:	5708850

Faculty of Civil Engineering and Geosciences · Delft University of Technology

TUDelft Delft University of Technology

Acknowledgements

First and foremost, I would like to express my deep gratitude to my supervisors. Thank you, Dr. Michael Hicks, for acting as the chair of my supervising committee and for your invaluable guidance in geotechnical reliability analysis. To Dr. Bram van den Eijnden, I am immensely grateful for the extensive time you dedicated to discussing every part of my thesis and guiding me at every step. To Dr. Iuri Rocha, thank you for deepening my interest and knowledge in machine learning, particularly in Gaussian Process regressions, and for your critical insights that have significantly improved this section of my thesis.

I extend my thanks to my home university, the University of the Philippines, for providing me with the opportunity to enhance my academic skills. A special thank you to the OVPAA-FRASDP team for generously funding my Master's program. I am also grateful to the Department of Civil Engineering for their steadfast support in this endeavor and for their help with all administrative matters. I eagerly look forward to my continued involvement with this department.

My sincere thanks go to TU Delft, including all the professors, staff, and teaching assistants under whom I have had the privilege to study, for imparting a wealth of invaluable knowledge in geotechnical engineering and in related fields such as engineering mechanics, project management, stakeholder involvement, data analysis, and data science. I am now eager and more confident to apply these concepts in real-world situations.

I would also like to acknowledge the Geo-engineering section, including all professors and students I have interacted with, for making this journey not just manageable, but far more enjoyable than I had anticipated. I am proud to call you my colleagues and friends.

Of course, I would like to thank my family for always supporting me in every decision that I make and every challenge that I face.

Lastly, I give thanks to the Lord for all the blessings He has bestowed upon me and my loved ones.

SUMMARY

This thesis explores the application of a Bayesian approach to hyperparameter optimization in surrogate modeling for geotechnical engineering problems. Surrogate modeling, particularly employing Gaussian Processes and Kriging, has become an essential tool for accelerating complex numerical simulations in geotechnical engineering. Traditional Maximum Likelihood Estimation (MLE) approaches to hyperparameter optimization, although straightforward, often overlook the inherent uncertainties in model hyperparameters, potentially leading to sub-optimal prediction accuracy and poor generalization.

The research presented in this thesis investigates the feasibility and benefits of applying a Bayesian inference approach to the tuning of hyperparameters in surrogate models. This approach allows for a probabilistic treatment of hyperparameters, providing a comprehensive quantification of uncertainty. The Markov Chain Monte Carlo sampling method, specifically the No-U-Turn Sampler, was employed to sample from the posterior distributions of hyperparameters, addressing the challenges posed by their non-Gaussian nature and non-linear relationship with model outputs.

Three case studies of varying complexity from geotechnical engineering practice are examined to compare the Bayesian approach against traditional MLE in terms of hyperparameter determination, prediction accuracy, uncertainty quantification, and computational efficiency. The findings suggest that the Bayesian approach, while computationally more intensive, could potentially offer more accurate predictions in terms of the mean-squared error and provide a deeper understanding of uncertainty, which is crucial for risk-informed decision-making in geotechnical engineering.

The study concludes that Bayesian hyperparameter optimization in surrogate modelling holds significant potential for improving the robustness and reliability of predictions in geotechnical engineering, particularly in applications involving complex dependencies and where a thorough understanding of uncertainty is crucial. Further research is recommended to enhance the computational efficiency of the Bayesian method and to explore its integration with multi-point enrichment strategies for practical engineering applications.

Contents

List of Tables

List of Figures

List of Symbols and Abbreviations

Symbols

c	Soil cohesion
E	Soil modulus of elasticity
ϵ_{P_f}	Probability of failure convergence criterion
FS	Factor of safety
g	Performance function
$\mathscr{H}$	Metamodel hyperparameters; includes kernel hyperparameters, scaling factor, and noise
k	Soil hydraulic conductivity
κ	Modified Cam-Clay swelling index
K	Kernel of a Gaussian process metamodel
l	Length scale
λ	Modified Cam-Clay compression index
$\mathscr{L}$	Likelihood
$\mathscr{M}$	Metamodel
$\mu_{\hat{y}}$	Predictive mean of metamodel
$\mathbb{P}$	Probability
P_f	Probability of failure
ϕ	Soil friction angle
$\hat{R}$	Statiscical convergence criterion of Markov chain sampling chains as defined by Gelman and Rubin, 1992
R	Correlation function
s	Soil settlement

σ	Noise of a Gaussian process
$\sigma_{\hat{y}}$	Predictive uncertainty (standard deviation) of metamodel
η	Scaling factor between kernel and noise in a Gaussian process metamodel
U	Utility learning function

Abbreviations

DoE	Design of experiments
GP	Gaussian process
HMC	Hamiltonian Monte Carlo
MCMC	Markov Chain Monte Carlo
MLE	Maximum likelihood estimation
MSE	Mean-squared Error
NUTS	No-U-Turn sampler

1

Introduction

This chapter summarizes the current state-of-the-art in surrogate modelling in geotechnical problems and reliability analysis. It gives highlight to the research gaps and the motivation of this study. Lastly, it discusses the study's objectives and scope and limitations.

1.1. Research Context

Surrogate modelling, specifically employing Gaussian processes (GP) and Kriging, has become a powerful approach to accelerate complex numerical simulations in geotechnical engineering. The fundamental premise is to develop an alternative model based on known results in certain input parameter spaces, enabling predictions in unexplored regions with remarkable speed and acceptable accuracy. Before the advent of surrogate modelling, traditional methods such as crude Monte Carlo simulations were predominant but proved inefficient due to their computational demands, particularly when dealing with multiple independent input parameters defining complex structural behaviors and very low probabilities of failure. Therefore, only Level I and Level II methods have been used. These methods provide low to moderate consideration of uncertainties in reliability analysis. Reliability analysis, in the context of civil engineering, is the numerical assessment of an engineering structure's probability of success or, more often, failure. It involves probabilistic analysis of the structure's strength or stability, with a result that typically defines the probability of the structure failing or collapsing under certain loading/environmental conditions. It has a wide use in geotechnical engineering, including assessments of the reliability of footings, slopes, retaining walls, etc.

In recent years, advancements in the use of surrogate models have come through research, ranging from response surfaces to polynomial chaos expansion and support vector machines. Among these, Gaussian processes and Kriging have gained prominence. A Gaussian process, as applied in surrogate modelling, involves fitting joint Gaussian distributions on input and output spaces, with Kriging being the method of prediction using these joint distributions.

The effectiveness of Kriging in conjunction with active learning and Monte Carlo sampling has been explored in reliability analysis, leading to advancements such as Active Learning with Monte Carlo Sampling (Al-Bittar et al., 2018) and its refinement with importance sampling (AK-IS) (Soubra et al., 2019). These techniques demonstrated efficiency gains compared with crude Monte Carlo techniques, particularly in scenarios with smaller probabilities of failure, and have found applications in geotechnical reliability analysis, including bearing capacities, slope stability, and monopile reliability.

However, a commonality among these studies is the reliance on Type II - Maximum Likelihood Estimation (Type-II MLE) for hyperparameter optimization in the Gaussian process. While MLE is a quick and straightforward method, it has inherent limitations, including deterministic results in terms of hyperparameter determination and poor generalization with limited available dataset. Recognizing these drawbacks, the research community has explored the potential of full Bayesian optimization for hyperparameter tuning in general machine learning contexts, but its application specifically to geotechnical models remains unexplored.

1.2. Research Questions

This research aims to answer the question: "What improvements can a full Bayesian optimization in hyperparameter tuning provide in surrogate modelling for geotechnical engineering problems?" As such, the main objective of this research is to determine the effect of applying a full Bayesian optimization in hyperparameter tuning of a surrogate model for a geotechnical engineering problem, particularly when compared to the more common approach of Type-II MLE of hyperparameters. In trying to answer the main research question, additional subquestions arise, including:

1. How can a full Bayesian approach to hyperparameter tuning be implemented in surrogate modelling of geotechnical engineering problems?
2. How can it be implemented in combination with active learning?
3. How does a geotechnical surrogate model that applies full Bayesian optimization in hyperparameter tuning compare with one that performs Type-II MLE to optimize hyperparameters?

1.3. Research Scope and Limitations

This research specifically aims to determine the effect of performing full Bayesian optimization in hyperparameter tuning. Since this method has not been applied to geotechnical engineering models and reliability analysis, it first aims to explore and compare its accuracy and performance when compared with the MLE method. Two simple geotechnical models were considered for this research as will be discussed later. It also aims to see its effectivity in reliability problems, for which one geotechnical problem with a more complex numerical solutions is considered. Random noise are added added to the results of the simple problems, while it was assumed that noise was inherently present in the solutions of the more complex problem, due to possible numerical instabilities. However as this is a preliminary analysis, basic learning functions (i.e. utility) which not fully take into account noise in observations are used. In addition, the use of simple geotechnical engineering problems in this research prevents the chance of having incomplete models, and consequently eliminates the need for more complicated learning functions and multi-step surrogate modelling methods, such as those employed in (Van den Eijnden et al., 2022).

Since this study explores the application of Bayesian optimization of hyperparameters both general geotechincal engineering as well as reliability problems, basic Monte Carlo sampling of the population was used in lieu of Importance sampling. Single point enrichment was performed, and hyperparameter updating was done for every iteration. In addition, while techniques were applied to increase the efficiency of the Bayesian approach, the performance time is not the main focus of this research, and as such more complex algorithms that may decrease computational costs were not looked into.

2

Literature Review

2.1. Surrogate Modelling

Surrogate modelling has been proven to be an effective method in speeding up the calculations of complex numerical models. The basic premise of surrogate modelling is that when results of a numerical model at parts of the input parameter space are available, an alternative model is trained and developed that can predict the results in other regions of the input parameter space at much faster speeds and within an acceptable level of accuracy.

Its application to reliability analysis has been a popular topic of research in recent years. Reliability analysis involves the determination of an engineering structure's probability of failure through stochastic analysis of the inputs and their corresponding function outputs, which generally relate to the safety of the structure. Before the development of surrogate modelling, results for the entire input space were mostly calculated using crude Monte Carlo techniques. This involves calculation of the true performance function's value for every set of input generated through stochastic sampling. The probability of failure is then defined as the fraction of failing points, as defined in Equation 2.1. The performance function as defined such that it is greater than 0 if the structure is safe and less than or equal to 0 if the structure fails.

$$P_{f,MC} = \frac{N_{g(x)\leq 0}}{N_{MC}} \tag{2.1}$$

Where:

$P_{f,MC}$	: Monte Carlo evaluation of the probability of failure
g	: performance function

While this method is effective in estimating the probability of failure, it becomes inefficient when dealing with lower probabilities of failure, for which most structures today are defined. Depending on the structure's importance, the required probabilities of failure can reach values as low as 10^{-6}. For these values, more than 1,000,000 samples and corresponding function values would be needed for a sufficiently accurate prediction of the probability of failure. The large number of required runs of the true model makes this method time consuming. In addition, analysis of the performance of engineering structures require complex time-demanding numerical models (i.e. finite element models), and the computation time increases exponentially with increasing number of input parameters. This led to the development of surrogate models, requiring much less runs of the true model.

Through the years, different types of surrogate models have been developed, each having an advantage over its predecessors. Below is a summary of the most commonly used surrogate models, based on a comprehensive comparison by Teixeira et al., 2021.

1. **Response Surfaces.** Outputs are treated as a sum of polynomial basis functions of the inputs. The use of this type is limited to models exhibiting local reliability, where failure regions are confined in a local region of the input space. It is also limited to weakly non-linear models, where a low order of polynomial basis functions can be used.

2. **Polynomial Chaos Expansion.** The outputs are treated as a combination of multivariate basis functions that are orthogonal to the joint probability density functions of the input variables. It is able to capture the global behavior of the true model and performs well when the model is globally smooth. However, there is not much evidence of the metamodel's performance in complex (i.e. highly non-linear) models, and the implementation is challenging in high-dimensional input spaces.

3. **Support Vector Machines.** This is a kernel-based metamodelling technique initially formulated for classification problems and later extended to regression problems.

4. **Kriging and Gaussian Processes.** The outputs are treated as a combination of a trend and a Gaussian process. The trend is a sum of basis function of the inputs, while the Gaussian process is the covariance of the outputs defined by the autocorrelation of the inputs. It has been shown to be the most robust in application to varying complexity and smoothness of the true model, but is costly for high-dimensional inputs.

Due to its robustness, Kriging has commonly been used in metamodelling highly non-linear problems, particularly in engineering.

2.2. KRIGING AND GAUSSIAN PROCESSES

A Gaussian process is a collection of random variables that have a joint Gaussian distribution (Rasmussen and Williams, 2006). It is used in surrogate modelling by fitting joint Gaussian distributions on the input and output spaces of a parent model. Using properties of these joint distributions, new predictions can be made from new inputs. This prediction method is commonly known as Kriging. Ingredients of a Kriging metamodel include functional basis, kernel, and hyperparameters (Lataniotis et al., 2022).

The functional basis defines the trend of the metamodel. This is used to specify a prior belief regarding the general relationship between the inputs and output of a model. Commonly used trends in Kriging include constant trends, linear trends, and polynomial trends. In the absence of this prior belief, this trend can be removed, thus allowing the kernel to play a more significant role in the Gaussian process metamodel predictions. This is commonly known as a Gaussian process with zero mean, and is utilized in complex geotechnical engineering problems and reliability analysis where the form of the trend is not known a priori.

The kernel defines autocorrelation of the target values as functions of the autocorrelation of the inputs. These are called correlation functions and are defined based on the expected behavior of the model. One of the most common correlation functions used in geotechnical reliability analysis is the squared exponential correlation. This correlation depends on how close two inputs are in the input variable space, such that the output values of closer inputs have higher correlations. Thus, this correlation function is defined by a length scale hyperparameter, describing how far inputs have to be to have no correlation between outputs. Squared exponential kernels create metamodels that are infinitely differentiable, which may not be representative of some models. The Matérn autocorrelation function addresses this by adding a smoothness hyperparameter, which defines how smooth and differentiable the model is (Van den Eijnden et al., 2022).

While the trend and the kernel are the two main ingredients of a Kriging metamodel, it is also heavily affected by a set of metamodel hyperparameters. As mentioned above, the kernel is a function of multiple hyperparameters such as the length scale and smoothness parameter, where different correlation types have its own set of hyperparameter types. In addition to these kernel hyperparameters, the metamodel may have additional hyperparameters, depending on what information is known about the observations. For example, if the uncertainty of observations is not known, then the noise of the metamodel is treated as a hyperparameter.

For a zero-mean Gaussian process where the uncertainty of the observations is unknown, the metamodel is defined by a normal distribution as shown in Equation 2.2.

$$\mathcal{M}(X) = \mathcal{N}(0, \eta^2 K(X,\theta) + \sigma^2) \tag{2.2}$$

Where:

$\mathcal{M}$	: metamodel
η	: scaling factor between kernel and noise
K	: GP kernel
θ	: kernel hyperparameters
σ	: GP noise

Since the metamodel is defined by a Gaussian distribution of functions, the mean and uncertainty of the predictions are a natural results of Kriging, and this is one particular advantage of Kriging over other types of metamodels particularly when used with active learning.

2.3. Active Learning

The computational requirements needed for learning depend heavily on the number of learning points since true values are calculated for each of these learning points. Thus, in surrogate modelling of complex and computationally expensive models, active learning is employed with a small initial design of experiments (DoE). The acceptability of the metamodel is evaluated based on certain accuracy metrics, and the DoE is expanded by adding learning points until the desired accuracy is met. Selection of input points to be added to the DoE depends on learning or acquisition functions evaluated at each point outside the DoE. Multiple learning functions suitable for different types of problems have been developed. Most of these functions derive from the metamodel prediction characteristics at each point, such as the mean and uncertainty. Typical learning functions include:

1. **Variance.** The uncertainty (defined by the variance) of the metamodel prediction of each point is the learning criteria, where the point with the highest uncertainty is chosen as the next learning point. This is typically used in general active learning algorithms where no function is being optimized.

2. **Expected Improvement.** A balance between the uncertainty of the metamodel prediction of each point and the expected improvement of an optimization function caused by the addition of that learning point to the DoE is the learning criteria. This is used when a function is being optimized (i.e. model accuracy).

3. **Utility (*U*).** The probability of misclassification of each point is used as the learning criteria. This is used when the model can be treated as a classification problem. This is commonly employed in reliability analysis of structures, where each input point produces an output that can define whether a structure fails or not.

Since learning functions depend on the mean and variance of metamodel predictions, the use of Kriging over other metamodel structures in combination with active learning is advantageous due to its automatic calculation these results.

The advantages of using Kriging in combination with active learning and Monte Carlo sampling were explored (Echard et al., 2011). This method is now known as AK-MCS. It was found that Monte Carlo sampling can be improved by separating failure and non-failure regions with a failure line, allowing enrichment near the limit state by providing appropriate learning functions. In addition, the advantages of using Kriging over other metamodels were explained. For instance, it naturally obtains the uncertainty at prediction points without the need for further Monte Carlo analysis. This uncertainty is needed in the learning process, thus the computation time of enrichment was lessened. It was shown that the combination of Kriging with Monte Carlo sampling greatly reduced the number of calls to the true model needed to achieve an acceptable accuracy of the probability of failure.

The method was further developed by combining active learning and Kriging with importance sampling (Echard et al., 2013), known as AK-IS. Importance sampling obtains population samples centered around a most probable failure point, whereas Monte Carlo sampling defines the population of samples centered around the mean of input parameters. Particularly for reliability analyses, this allowed for a much smaller size of the population set while maintaining the same level of accuracy of the probability of failure. While more realizations were needed to reach a target accuracy, the faster computation of predictions and learning functions lead to a lower overall computation time.

2.4. Surrogate Modelling in Geotechnical Engineering

The application of surrogate modelling in geotechnical engineering has been a popular topic of research in recent years. The use of polynomial chaos expansions in surrogate modelling of geotechnical problems and reliability analysis was investigated (Sudret, 2014). Soil stochastic inputs involved strength and stiffness parameters. Reliability analysis of a strip footing and settlement analysis of a soil layer with spatial variations were assessed using classical FORM/SORM methods, polynomial chaos expansion, and Monte Carlo estimations. Polymonial chaos expansion was shown to provide accurate estimates of the probability of failure and settlement values, while also being able to represent the model outputs better than FORM or SORM analysis.

More popularly, the use of Kriging in surrogate modelling of geotechnical engineering problems has been a popular topic of research. The use of AK-MCS as a surrogate model for bearing capacities of strip footings on spatially varying soils was explored (Al-Bittar et al., 2018). In this research, the limit state was modelled both using analytical solutions following a general bearing capacity equation and using finite element method. Cohesion and friction angle of soil were treated as stochastic variables that define the soil spatial-variability. Monte Carlo sampling was used to obtain the initial DoE, and the U learning function was used during the enrichment process. The stopping condition was based on a minimum value of 2 for the learning function, which was shown by Echard to correspond to a probability of misclassification equal to 0.023 and provide great accuracy to the predictions. However, it was also shown to be stricter than a simple analysis of the convergence of the probability of failure and its convergence criterion. The Kriging metamodel was able to obtain a sufficiently accurate probability of failure, with low convergence criterion value, at a lower number of added realizations.

Similarly, Al-Bittar refined the method by adopting AK-IS instead of AK-MCS, which permitted a reduction in the population size (Soubra et al., 2019). While a larger number of added realizations where needed for the AK-IS compared to AK-MCS, the faster computation times for predictions and learning functions still resulted in an overall lower computation time for the AK-IS.

Other applications of Gaussian process regression in geotechnical engineering include surrogate modelling for slope stability (Kang et al., 2015) and monopile reliability analysis (El Haj et al., 2019). El Haj and Soubra applied an improved enrichment process, based on the U-learning function, which adds multiple points during each learning step by using a weighted K-means clustering approach in solving for the learning functions.

The application of AK-IS to a reliability analysis of sheet pile walls in dykes was explored (Van den Eijnden et al., 2022). In this study, two new challenges were identified and tackled: (1) noisy model and (2) incomplete model. A noisy model is a model that has inherent uncertainty, which can be attributed to the stability of the numerical model itself or to very low tolerance for convergence. An incomplete model is a model that has no solutions at certain regions of the input variable space. In the case of sheet pile analysis, this happens for soil properties that cause failure during initialization of the slope, before the installation of the sheet pile. To address the problem of incomplete models, a two-stage metamodelling procedure was employed, incorporating a classification stage prior to the regression modeling. The classification model was used to classify the space into compatible (regions where solutions exist) and incompatible domains (regions without solutions), and the regression model is only applied on the compatible domain. In addition, an alternative learning function was used, which is based on the difference in the probability of misclassification before and after adding a candidate learning point. This dealt with the problem of noisy models, where the probability of misclassification of a candidate learning point does not reduce to zero after addition of that learning point.

2.5. ADAPTIVE METAMODELLING AND CURRENT RESEARCH DIRECTIONS

Current research efforts in metamodelling focus on altering specific aspects of metamodels to determine if these changes lead to improvements. These can be categorized into several focus areas including (Teixeira et al., 2021):

1. **Initialization of Design of Experiments**. Different strategies for initializing the design of experiments have been studied, including Monte Carlo sampling, importance sampling, Latin hypercube sampling, etc.

2. **Enrichment of Metamodel**. Different learning functions have been developed such as those discussed in 2.3.

3. **Size and Domain of Design of Experiments.** The required dimensionality of the design of experiments for accurate metamodelling, as determined through sensitivity analysis, is a key research direction in each field application.

4. **Assumptions and Estimations of Metamodel Parameters.** Various assumptions and methods for parameter determination have been researched and developed. These include the use of different metamodel types and trends as well as different kernels.

As discussed in 2.4, most of the research in metamodelling in geotechnical engineering have focused on either altering the initialization of the sample population and design of experiments or the enrichment of the metamodel. Different types of metamodels have also been applied. However, a commonality exists in research involving Kriging in geotechnical engineering: the estimation of hyperparameters is done through marginal likelihood estimation, alternatively known as Type-II Maximum Likelihood Estimation (MLE). This involves marginalizing the metamodel prediction over different possible models and determining the hyperparameter values that maximize the marginal likelihood of the observed data. This provides simplicity such that only one set of hyperparameter values is used in building the metamodel, and the uncertainty of the predictions is characacterized only by the uncertainty of the Gaussian process.

However, this approach neglects the effect of hyperparameter uncertainty, resulting in a deterministic hyperparameter determination. In addition, MLE methods are known to cause poor generalization of the model especially when the training dataset is small. Although Kriging generally incorporates a Bayesian approach for model selection, extending this Bayesian framework to hyperparameter selection could enhance the model's generality. Thus, Bayesian approaches to hyperparameter optimization have been attempted.

2.6. Bayesian Approach to Hyperparameter Estimation

A full Bayesian approach to hyperparameter estimation involves applying a prior belief to the hyperparameter distribution, observing data, and building posterior hyperparameter distributions based on the prior distribution and the likelihood of the observed data conditioned on the hyperparameters. This essentially follows Bayesian inference as shown in Equation 2.3.

$$posterior = \frac{likelihood * prior}{marginal} \rightarrow \mathbb{P}(\mathscr{H}|Y) = \frac{\mathscr{L}(Y|\mathscr{H})\mathbb{P}(\mathscr{H})}{\mathbb{P}(Y)} \tag{2.3}$$

Where:

$\mathscr{H}$: concatenation of all hyperparameters of the metamodel

Y : observed values of outputs

When the outputs are linear functions or parameters and the parameters can be fitted with Gaussian distributions, the expansion of Equation 2.3 leads to tractable integrals and an analytical solution for the posterior distribution. However, model hyperparameters are neither Gaussian-distributed nor linear with respect to the model outputs, and as such no analytical solution for the posterior can be obtained. This has hindered the use of a full Bayesian approach to hyperparameter determination.

Alternative approaches to building the posteriors of non-Gaussian priors exist in the form of Markov Chain Monte Carlo (MCMC) sampling (Neal, 1997). This can be done because the marginal in Equation 2.3 is constant for all parameter or hyperparameter values, and the shape of the unnormalized posterior distribution can be calculated. Popular sampling techniques include Metropolis-Hastings MCMC, slice sampling and Hamiltonian Monte Carlo (HMC). Metropolis-Hastings works efficiently in finding local modes of distributions. However, the efficiency of the sampler is highly dependent on a proposal distribution that must be defined a priori. Moreover, the efficiency of its random walk behavior is very limited to simple unimodal distributions and lower-dimensional spaces. On the other hand, slice sampling uses a 'slice' to determine the range from which subsequent samples may be drawn. The size of the slice adjusts adaptively and rapidly, eliminating the need for a predefined proposal distribution (Murray and Adams, 2010 ; Neal, 2000). While it can be used to efficiently sample kernel hyperparameters in low-dimensional spaces, its efficiency decreases as the dimentionality of the sampling space increases. Recently, the incorporation of Hamiltonian dynamics with Metropolis-Hastings algorithm, appropriately named Hamiltonian Monte Carlo, has been shown to be an efficient sampler both in simple and more complex distributions (Neal, 2011; Betancourt, 2018). HMC introduces auxiliary momentum variables, sampled concurrently with each hyperparameter, to guide the sampling process. Following Hamiltonian dynamics, wherein the Hamiltonian remains constant for paired values of hyperparameters and momentum variables, the trajectory of the sampler is determined and the end of this trajectory is taken as the next proposed sample. MCMC acceptance criteria are then applied on the proposed sample. The No-U-Turn sampler (NUTS) (Hoffman and Gelman, 2014) builds on this technique by stopping the trajectory when it starts making a 'U-turn' and goes back to the value of the previous sample, effectively reducing sample redundancy and improving efficiency. Currently, NUTS is the method being employed by coding packages specializing in Bayesian inference, as it has been shown to be effective in building non-Gaussian, high-dimensional, and complex posterior distributions. More detailed information of HMC and NUTS are provided in Section 3.3 of the Theoretical Framework chapter.

3

Theoretical Framework

This section discusses the different theories involved in this thesis. Formulations are presented through equations followed by step-by-step discussions of each algorithm's implementation. There are two different algorithms being compared in this thesis. First is a Type-II MLE approach to hyperparameter determination of Gaussian processes applied to geotechnical engineering problems, and second is a full Bayesian approach to hyperparameter determination of the same metamodel. This section is structured to first discuss the commonly used algorithm for building metamodels with Type-II MLE hyperparameter determination, including its extension to active learning. Subsequently, the full Bayesian algorithm, also in conjunction with active learning, is presented.

The equations below apply for a zero-mean Gaussian process. For a more general formulation that includes a Kriging trend, readers are referred Lataniotis et al., 2022 or Dubourg, 2011.

3.1. Gaussian Processes

3.1.1. Kernel and Autocorrelation Functions

The basic definition of a Gaussian process has been discussed in 2.2, and the equation for a zero-mean Gaussian process is shown in 2.2. The kernel matrix is a function of the autocorrelation of the inputs and the kernel hyperparameters. It is a matrix defining the autocorrelation between each pair of known inputs, as shown in 3.1, where R is the autocorrelation function between two inputs and N is the number of observed data.

$$K(\mathbf{x},\mathbf{x}',\theta) = \begin{pmatrix} R(x_1,x_1,\theta) & R(x_1,x_2,\theta) & \cdots & R(x_1,x_{N-1},\theta) & R(x_1,x_N,\theta) \\ R(x_2,x_1,\theta) & R(x_2,x_2,\theta) & \cdots & R(x_2,x_{N-1},\theta) & R(x_2,x_N,\theta) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ R(x_{N-1},x_1,\theta) & R(x_{N-1},x_2,\theta) & \cdots & R(x_{N-1},x_{N-1},\theta) & R(x_{N-1},x_N,\theta) \\ R(x_N,x_1,\theta) & R(x_N,x_2,\theta) & \cdots & R(x_N,x_{N-1},\theta) & R(x_N,x_N,\theta) \end{pmatrix} \tag{3.1}$$

Different autocorrelation types exist, and listed below are some most commonly used types along with their corresponding equations.

1. Exponential

$$R(x,x',\theta) = \exp\left(-\frac{|x-x'|}{l}\right) \tag{3.2}$$

2. Squared Exponential

$$R(x,x';\theta) = \exp\left(-\frac{1}{2}\left(\frac{|x-x'|}{l}\right)^2\right). \tag{3.3}$$

3. Matérn

$$R(x,x';\theta) = \frac{1}{2^{\nu-1}\Gamma(\nu)}\left(\frac{\sqrt{2\nu}|x-x'|}{l}\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}|x-x'|}{l}\right).. \tag{3.4}$$

Where:

l	: length scale
ν	: smoothness hyperparameter

In the equations above, θ is a concatenation of kernel hyperparameters including the length scale and smoothness hyperparameter. The smoothness hyperparameter used in the Matérn correlation controls the differentiability of the model. When it approaches infinity, the model becomes infinitely differentiable and this Matérn correlation then resembles a squared exponential correlation.

When the input is multi-dimensional, the autocorrelation can be calculated for each dimension and multiplied as shown in 3.5, where M represents the dimensionality of the input.

$$R(x,x',\theta) = \prod_{i=1}^{M} R(x_i,x'_i;\theta_i). \tag{3.5}$$

3.1.2. MARGINAL LIKELIHOOD ESTIMATION

The hyperparameters of the Gaussian process can be estimated by maximizing the marginal likelihood of the observed data conditioned on the hyperparameters. Following a Gaussian assumption of all possible models, the marginal likelihood of the observed outputs is shown in Equation 3.6.

$$\mathscr{L}(Y|X,\mathscr{H}) = \left(\frac{\det(C)^{-1/2}}{(2\pi N)^{N/2}}\right)\exp\left(-\frac{1}{2}Y^T C^{-1} Y\right) \tag{3.6}$$

Where $C = \eta^2 K(x,x',\theta) + \sigma^2$ is the covariance matrix in Equation 2.2. In log-scale, Equation 3.6 can be written as shown in Equation 3.7. The use of log-scale in calculations is beneficial as it provides better computational stability when dealing with very low probability values.

$$\log\left(\mathscr{L}\left(Y|X,\mathscr{H}\right)\right) = -\frac{1}{2}N\log(2\pi\hat{\sigma}^2) - \frac{1}{2}\log(\det(C)) - \frac{N}{2}\log(2\pi) \tag{3.7}$$

Where:

$$\hat{\sigma} = \frac{1}{N}Y^TC^{-1}Y \tag{3.8}$$

Optimization of hyperparameters then becomes a maximization of Equation 3.7.

$$\hat{\mathscr{H}} = \operatorname{argmax}\left(\log\left(\mathscr{L}\left(Y|X,\mathscr{H}\right)\right)\right) \tag{3.9}$$

3.1.3. Prediction

Once an optimal hyperparameter set has been selected, predictions at new points can be made. Equation 3.10 models the joint distribution between observed outputs and new predictions, assuming a multivariate normal distribution.

$$\begin{bmatrix} Y \\ \hat{y} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} C & c^T = \eta^2 k(X,\hat{x},\theta)^T \\ c = \eta^2 k(X,\hat{x},\theta) & \eta^2 I \end{bmatrix}\right) \tag{3.10}$$

In Equation 3.10, k is a vector of covaraince between each of the known inputs and the prediction point. From this joint distribution, the predictive mean and variance are calculated using Equations 3.11 and 3.12, respectively.

$$\mu_{\hat{y}} = c^TC^{-1}Y \tag{3.11}$$

$$\sigma^2_{\hat{y}} = \sigma^2 - c^TC^{-1}c \tag{3.12}$$

3.2. Model Enrichment Through Active Learning

For each DoE set, optimal hyperparameters are selected, enabling predictions at new points. To improve these predictions, new learning points are added based on the learning function values of each point in the population. These learning functions are typically defined by the predictive means and uncertainties, which are natural results of Kriging as demonstrated in Equations 3.11 and 3.12. A simple enrichment approach is to select the point with the highest prediction variance, with the goal of reducing the overall uncertainty of the metamodel. This method allows for efficient exploration of the stochastic variable space and is particularly useful if the goal of the metamodelling is getting the most accurate model. However, in such cases where a particular function value is also desired, such as in optimization or reliability problems, this active learning method does not fully leverage the knowledge that has already been obtained from the available learning points. In these cases, other learning functions such as Expected Improvement and Utility learning functions, which balance exploitation with exploration, are more applicable.

In optimization problems, the goal is to find the maximum value of true function. In this case, a learning point is selected based on the currently believed function value of that point combined with the maximum possible increase of this point's function value, which is defined by its uncertainty. As such, the Expected Improvement learning function is defined as the sum of its mean and its standard deviation.

In reliability problems, points nearer to the limit state surface or points who have higher probabilities of misclassification are selected as the best succeeding learning points. It is common to define the limit state as a performance function equal to zero, where function values less than or equal to zero indicate failure. Thus, the probability of misclassification can be related with the ratio of the absolute value of the predicted mean and standard deviation of a point as shown in Equations 3.13 and 3.14 (Moustapha et al., 2022).

$$U(x) = \frac{|\mu_{\hat{y}}(x)|}{\sigma_{\hat{y}}(x)} \tag{3.13}$$

$$P_m(x) = \Phi(-U(x)) \tag{3.14}$$

Where:

U	: utility learning function
P_m	: probability of misclassification
Φ	: standard Gaussian cumulative density function

Selecting the point with the highest probability of misclassification is the same as selecting the point with the lowest U learning function value. This balances exploitation, by prioritizing points with predictive means closer to the limit state performance function value of zero, and exploration, by giving importance also to points with higher predictive uncertainties.

Each time a learning point is added, a new metamodel is built. This includes rebuilding of the kernel matrix, re-optimization of hyperparameters, and recalculation predictions and prediction uncertainties. Enrichment is stopped once the metamodel has reached a certain level of accuracy. Typically, model accuracy is assessed by comparing predictions to predefined validation points using a loss function, such as mean squared error (MSE). This is shown in 3.15.

$$MSE_{val} = \frac{1}{N_{val}} \sum_{n=1}^{N_{val}} \left(y_{true} - \hat{y}\right)^2 \tag{3.15}$$

Where:

MSE_{val}	: mean squared error at validation points
N_{val}	: number of validation points
y_{true}	: true function values at validation points

Since the predictions are Gaussianly distributed, an MSE calculated with respect to the mean prediction gives a representative accuracy for each metamodel.

In some cases, especially when the true function is complex and expensive to perform, the definition of validation points may not be practical. Instead, the learning can be stopped once the uncertainty or learning function of the model reaches a pre-defined tolerance value. In reliability problems, the training can be stopped upon convergence of the probability of failure. The probability of failure can be defined as the number of points failing, or in this case having predicted performance function values less than zero, divided by the total population of prediction points. This is shown in Equation 3.16, where the probability can simply be obtained as a Monte Carlo estimate (Echard et al., 2011).

Moreover, the uncertainty of the P_f prediction is defined by its coefficient of variation. This can be calculated, for a Monte Carlo analysis, with Equation 3.17. Low values of P_f lead to large values of its coefficient of variation, but this can be lowered by providing a large number of samples.

$$\hat{P}_f = \mathbb{P}(\mu_{\hat{y}} \leq 0) \tag{3.16}$$

$$COV_{\hat{P}_f} = \sqrt{\frac{1-\hat{P}_f}{\hat{P}_f \cdot N_{MC}}} \tag{3.17}$$

3.3. Bayesian Optimization of Hyperparameters

3.3.1. MCMC Sampling of Posterior Distributions

For a full Bayesian optimization of hyperparameters, the posterior distributions of each hyperparameter must be determined in each iteration of metamodel building. As discussed in 2.6, this can be estimated through MCMC sampling methods, where the unnormalized distribution is equal to the product of the likelihood of the observed data and the prior probability of the hyperparameters. This relationship is formalized by Equation 3.18.

$$f(\mathscr{H}) = \mathscr{L}(Y|\mathscr{H})\mathbb{P}(\mathscr{H}) \tag{3.18}$$

Where:

$f(\mathscr{H})$: unnormalized posterior distribution of hyperparameters

As it is more numerically stable to work in the log-scale of probabilities, the logarithm of the unnormalized posterior as shown in Equation 3.19 is conveniently used instead of the unnormalized posterior in sampling.

$$f(\mathscr{H}) = \log(\mathscr{L}(Y|\mathscr{H})) + \log(\mathbb{P}(\mathscr{H})) \tag{3.19}$$

The value of the unnormalized posterior, $f(\mathscr{H})$, can then be used as an acceptance criteria for different sampling methods. Algorithms 1 shows how posterior distributions of hyperparameters can be built using Metropolis-Hastings MCMC. Note that h denotes a set of hyperparameter samples while h_i is used to represent each hyperparameter type.

Algorithm 1 Metropolis-Hastings MCMC Sampling of Hyperparameter Posteriors

```
1: For each hyperparameter, define proposal distributions $q(h_i)$.
2: Define prior distributions for each hyperparameter.
3: Propose initial samples for each hyperparameter. $h_{current}$ is set equal to the initial samples.
4: Calculate $f(h_{current})$.
5: Propose a new set of hyperparameter samples, $h_{proposed}$, from $h_{current}$ and following the proposal distributions $q(h_i)$.
6: Calculate $f(h_{proposed})$.
7: Randomly sample an acceptance factor value, $\alpha$, between 0 and 1.
8: If $\min\left(1, \frac{f(h_{proposed})}{f(h_{current})}\right) \geq \alpha$, accept and add $h_{proposed}$ to the list of samples and set $h_{current} = h_{proposed}$. Else, reject $h_{proposed}$ and re-add $h_{current}$ to the list of samples.
9: Repeat steps 4 to 8 until convergence with true distribution.
```

While Metropolis-Hastings MCMC is a simple and effective method in sampling hyperparameter values, it becomes inefficient in exploring more complex spaces, such as in the case of multidimensional joint distributions or wide and multi-modal posteriors. Particularly, the need for a proposal distribution limits the capability of the sampler in adapting the sampling step sizes based on the current sample of the hyperparameter. Thus, it has the potential of getting stuck in local modes. Hamiltonian Monte Carlo (Neal, 2011; Betancourt, 2018) sampling addresses this by combining the sampling approach of Metropolis-Hastings MCMC with Hamiltonian dynamics. HMC introduces auxiliary momentum variables for each hyperparameter. These variables, sampled concurrently with the hyperparameters, dictate the trajectory of the sampler according to Hamiltonian dynamics as defined with Equations 3.20, 3.21, and 3.22. These equations signify that the Hamiltonian remains constant along the trajectory of the sampler. This mechanism is analogous to the conservation of energy principle in physics, where the total energy of a system remains constant in the absence of external forces. In the case of HMC, the potential energy is the negative log-posterior of the hyperparameter sample as defined in Equation 3.19. Higher values of $f(\mathscr{H})$ correspond to lower potential energy. The kinetic energy on the other hand is a function of the auxiliary momentum, as specified in Equation 3.23.

$$H(\theta, p) = E_u(\theta) + E_k(p) \tag{3.20}$$

$$\frac{d\theta}{dt} = \frac{\partial H}{\partial p} \tag{3.21}$$

$$\frac{dp}{dt} = \frac{\partial H}{\partial \theta} \tag{3.22}$$

$$E_k(p) = \frac{1}{2}mp^2 \tag{3.23}$$

Where:

H	: Hamiltonian energy
E_u	: potential energy
E_k	: kinetic energy
p	: momentum
t	: time (arbitrarily taken as 1)
m	: mass (arbitrarily taken as 1)

Figure 3.1 visualizes how HMC is able to obtain samples from a one-dimensional bimodal distribution composed of Gaussian distributions centered at 10 and 20. The process begins with an initial hyperparameter sample set at 15, and a momentum variable is randomly drawn from a standard unit Gaussian distribution. The next sample is obtained through a deterministic path following the Hamiltonian equations. This path is discretized into a series of 'leapfrog' steps, where both the number of steps and the step size are predetermined. At the end of the trajectory, the hyperparameter sample is accepted following an acceptance criteria, and a new momentum is sampled again to provide the starting point of the new trajectory. The visualization in Figure 3.1 demonstrates how the trajectory varies based on the momentum sampled, illustrating the adaptive exploration capabilities of the HMC method allowing for a more efficient exploration of the hyperparameter space as compared to Metropolis-Hasting's random walk sampling approach. It can also be seen how depending on the sampled momentum, the sampler can stay within a local mode or extend to other possible modes in the space.

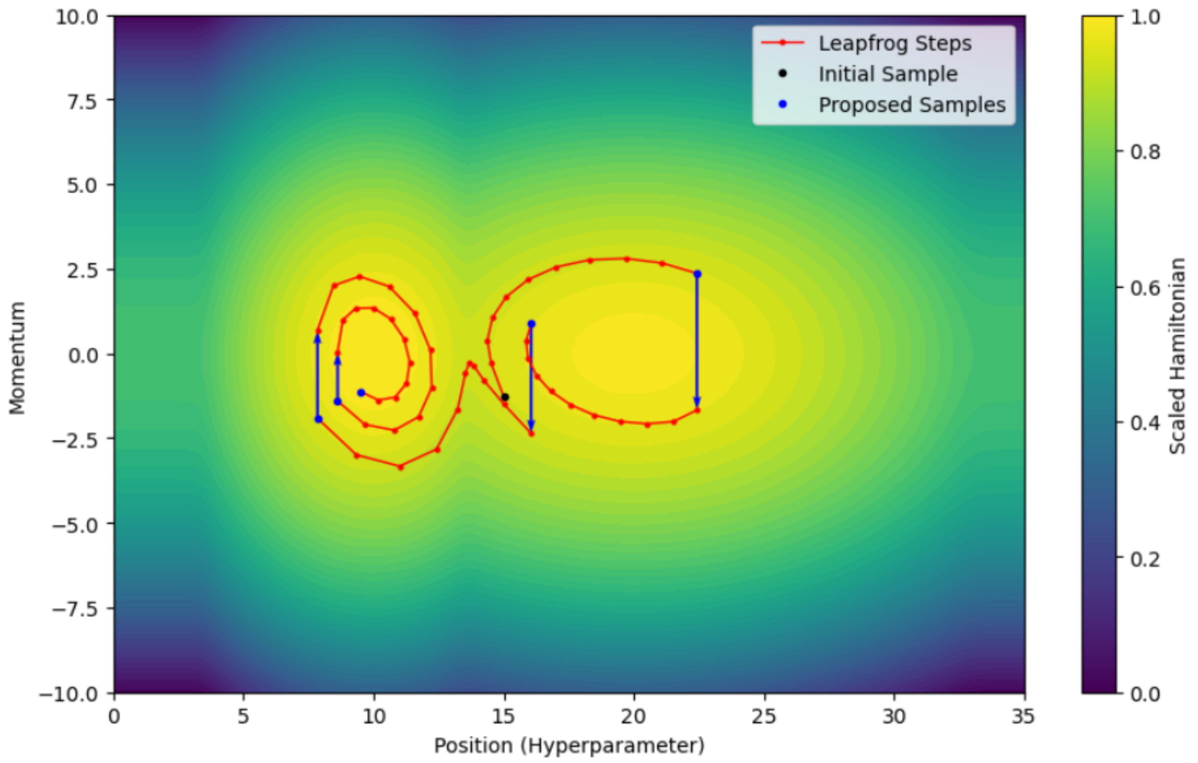


Figure 3.1: HMC Sampling Trajectories

While a one-dimensional example is shown above for a better visualization, it has been demonstrated (Neal, 2011) that HMC also works efficiently in sampling higher-dimensional distributions. However, there are still some limitations to this sampling technique, such as its dependence on pre-defined step sizes and number of leapfrog steps. The No-U-Turn sampler builds on the HMC sampling technique by stopping the leapfrog trajectory once the sampler starts moving back to the original position, effectively making a U-turn. This further decreases the autocorrelation between samples, thereby increasing the efficiency of the sampler (Betancourt, 2018; Hoffman and Gelman, 2014). Moreover, optimization of the leapfrog step sizes can be done by a preliminary sampling phase called the tuning stage, where a target acceptance rate is specified. The step size is adjusted to attain this target acceptance rate. NUTS has been shown to be effective in building posterior distributions in Bayesian problems and is currently being implemented by programming languages and packages specializing in Bayesian analysis.

To increase the efficiency of sampling, multiple parallel sampling chains can be conducted, each starting from different initial hyperparameter values. The Gelman-Rubin statistic $\hat{R}$, which compares the interchain variance with the intrachain variance, is often used as the stopping criteria for sampling (Gelman and Rubin, 1992). Calculation of $\hat{R}$ is done following Algorithm 2. To ensure the results are unbiased, the variances in the calculation above are normalized with $n_{samples} - 1$. As $\hat{R}$ approaches one, it indicates increasing convergence of the samples toward the true posterior distribution. Traditional acceptable $\hat{R}$ values are below 1.1; however recent research have suggested the use of 1.01 for practical applications (Vehtari et al., 2021).

Algorithm 2 Calculation of $\hat{R}$

```
For each chain, the intra-chain means and variances of the samples are calculated.
The mean of all samples is calculated by taking the mean of the intra-chain means.
The normalized inter-chain variance ($B$) is calculated by taking the variance of the intra-chain means and multiplying it with the number of samples per chain.
The average intra-chain variance ($W$) is calculated by taking the mean of the intra-chain variances.
$\hat{V} = \frac{n_{samples}-1}{n_{samples}} * W + \frac{n_{chains}+1}{n_{chains}*nsamples} * B$
$\hat{R} = \sqrt{\frac{\hat{V}}{W}}$
```

3.3.2. PREDICTION

The samples of hyperparameter sets can be used to build multiple different surrogate models, each one defined by a Gaussian distribution of functions following the same concepts in 3.1 and 3.1.3. Predictive samples of functions can be derived from the posterior means and covariances of the set of prediction points for each hyperparameter sample, leading to a collection of samples that take into account the uncertainties of both the hyperparameter space and Gaussian processes constructed from each hyperparameter set. Since the hyperparameter distribution does not conform to any standard parametric shape, traditional definitions of mean and uncertainty may not be applicable. While Gaussian equations for mean and standard deviation can estimate these metrics, analyzing predictions using the median and confidence intervals based on selected percentiles may be more appropriate. This provides flexibility in analyzing any distribution shape as well as robustness in dealing with outliers.

3.3.3. INCORPORATION WITH ACTIVE LEARNING

When considering a full Bayesian treatment of hyperparameters, learning functions, must be integrated over the hyperparameter space (Snoek et al., 2012). As mentioned, predictive distributions become non-parametric, and the learning functions defined in Section 3.2 must be modified. In this case, the uncertainty can be defined as the larger of the difference between the 97.5th percentile and the median or the difference between the median and the 2.5th percentile. Similarly, the U learning function can be redefined as shown in Equation 3.24, where the denominator takes into account the possibly non-symmetric uncertainty. Appendix Listing

$$U(x) = \frac{|P50(x)|}{max[|P97.5(x) - P50(x)|, |P50(x) - P2.5(x)|]} \tag{3.24}$$

Where:

$P50$	: median of predictions
$P97.5$	: 97.5th percentile of predictions
$P2.5$	: 2.5th percentile of predictions

In addition to the learning functions being integrated, the stopping criteria must also be integrated across the sampled hyperparameter values. How this was implemented is discussed in the Research Methods chapter.

3.4. 1D ILLUSTRATIVE EXAMPLES

To provide a clear and illustrative example of how the Bayesian approach would differ from the MLE approach to hyperparameter determination, a one-dimensional problem of soil consolidation is analyzed below. This problem is specifically chosen as the uncertainty of soil hydraulic conductivity spans several orders of magnitude, and a wider posterior of the length scale can be expected. A relationship between the hydraulic conductivity and one-dimensional consolidation (given other constant soil properties and loading time) is depicted in Figure 3.2. To briefly describe this behavior, at lower magnitudes of hydraulic conductivity, insignificant values of pore pressures are allowed to dissipate within the given time of loading. Between hydraulic conductivity values of $10^{-8} m/s$ to $10^{-5} m/s$, the largest difference in settlement values can be seen, implying that it is within this region that the soil behavior transitions from fully undrained to fully drained within the time duration. Beyond $10^{-5} m/s$, the settlement reaches a plateau, signifying soils that reach their fully drained state within the time duration.

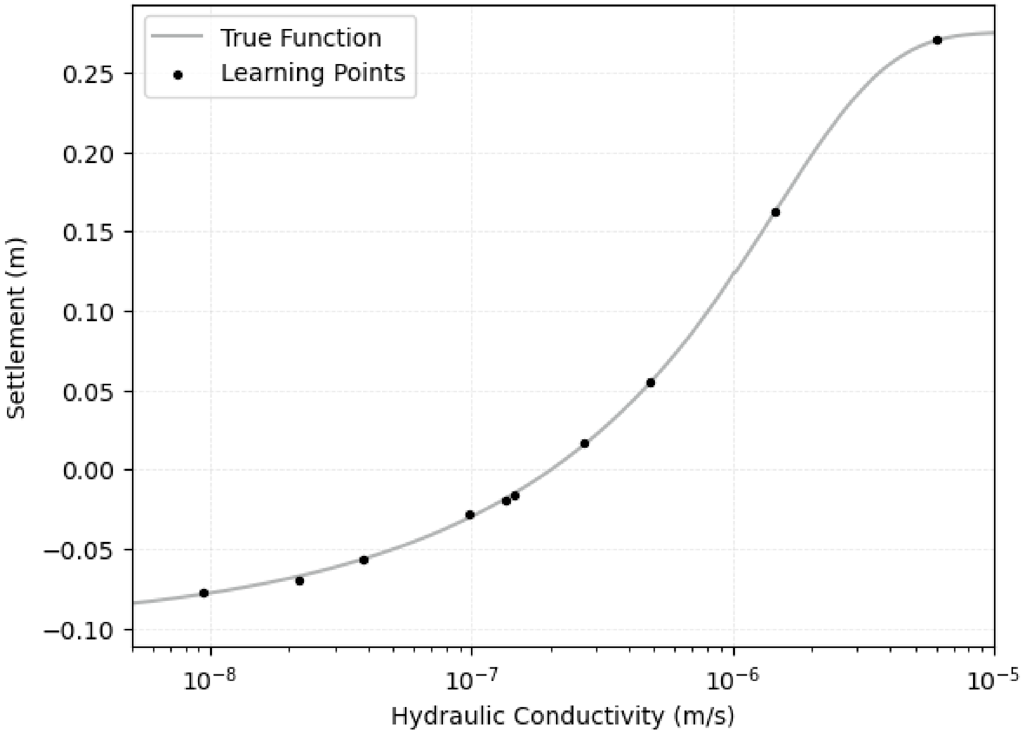


Figure 3.2: Consolidation Settlement vs. Hydraulic Conductivity

Ten training points are randomly selected from a wide log-normal distribution and provided as shown, and learning from these points was performed using MLE and Bayesian optimization of hyperparameters. Figure 3.3 shows the optimized and sampled hyperparameters. L-BFGS, a gradient-based optimizer, was used to obtain the maximum likelihood estimate. Meanwhile, PyMC's implementation of NUTS was used to generate the posterior samples, and it is evident that it can sample through very wide posterior distributions. In the MLE approach, multiple functions were sampled from the Gaussian process defined by the maximum likelihood estimate of the hyperparameter. In the Bayesian approach, one function was sampled from each of the Gaussian processes defined by each hyperparameter sample. Ten random selections of surrogate models are drawn from each bag and plotted in Figure 3.4, where the medians and confidence bounds are also shown. It must be noted that for the MLE approach, the median and 95% confidence bounds correspond to the mean and 1.96 standard deviations away from the mean, since the distribution of functions is Gaussian.

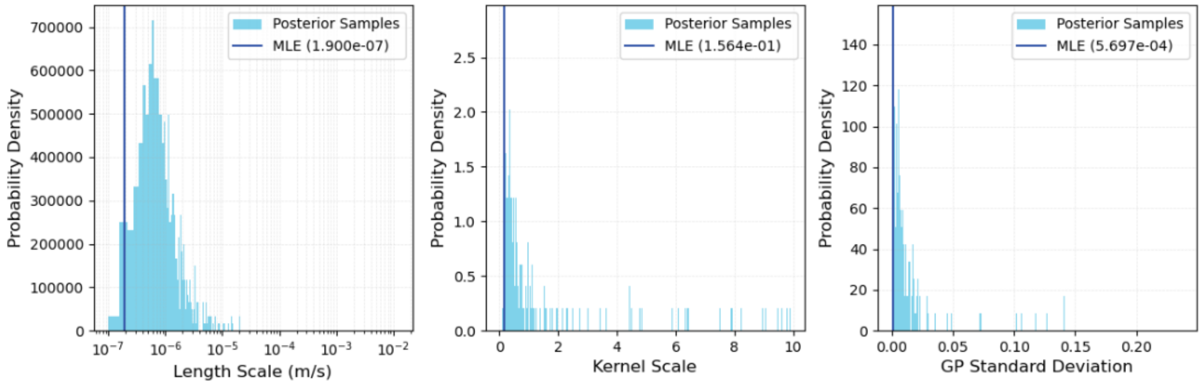


Figure 3.3: Optimized Hyperparameters

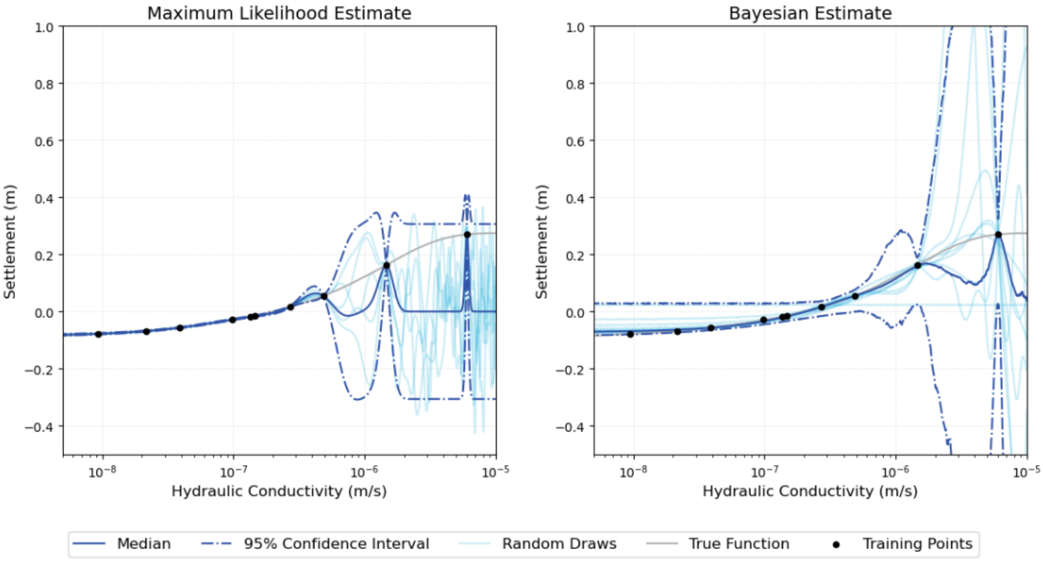


Figure 3.4: Surrogate Model Samples

While the maximum likelihood estimate of the length scale was able to find a value that will fit in some regions of the hydraulic conductivity space, particularly in the regions of similar magnitude with the optimized length scale, it fails to capture the function behavior at higher magnitudes (between $10^{-6} m/s$ and $10^{-5} m/s$) even with the presence of training points in that area. As can bee seen from the sample functions drawn, the relatively low value of the length scale causes short oscillations in higher magnitude orders.

Meanwhile, a Bayesian approach to selecting hyperparameters shows that the posterior distribution of length scales do indeed span across different magnitude orders. While functions derived from length scales of similar magnitude to those of the maximum likelihood estimates perform similarly to the MLE function draws, functions drawn from length scales of higher magnitudes fit better the behavior of the true function at higher hydraulic conductivity values. Integration over functions obtained from the posterior distribution captures better the behavior of the function over the entire parameter space. In addition, the added uncertainty in the hyperparameter space allows for a better encapsulation of the uncertainty of the predictions, signified by the wider confidence interval.

Additionally, one can easily see from Figure 3.4 how poorly defined hyperparameters can affect the learning of the model. With the MLE approach, the uncertainty remains relatively constant at the tail end of the input space. If the learning approach was to select the point with the highest predictive uncertainty, then any point within this region might be selected. In contrast, the Bayesian approach properly scales the uncertainties as the distance from the training points increase.

Figure 3.5 shows how this may affect surrogate modelling of reliability analysis problems. In this case, a performance function was set up based on bearing capacity of a strip footing, where friction angle is considered as the stochastic input variable. Again, the Bayesian approach was able to more closely resemble the true function. On lower values of friction angle, the MLE approach gave false performance function values less than zero. In addition, the MLE gave more confident predictions as shown by the narrower predictive intervals. For friction angles greater than 30 degrees, the entire predictive interval of the MLE method lies above the failure line, indicating a probability of failure misclassification of less than one in twenty. For the Bayesian approach, the lower bound of the predictive interval for friction angles less than 40 degrees is mostly below the failure line. Ultimately, the Bayesian method considers higher probabilities of misclassification due to the added uncertainty in the hyperparameter space.

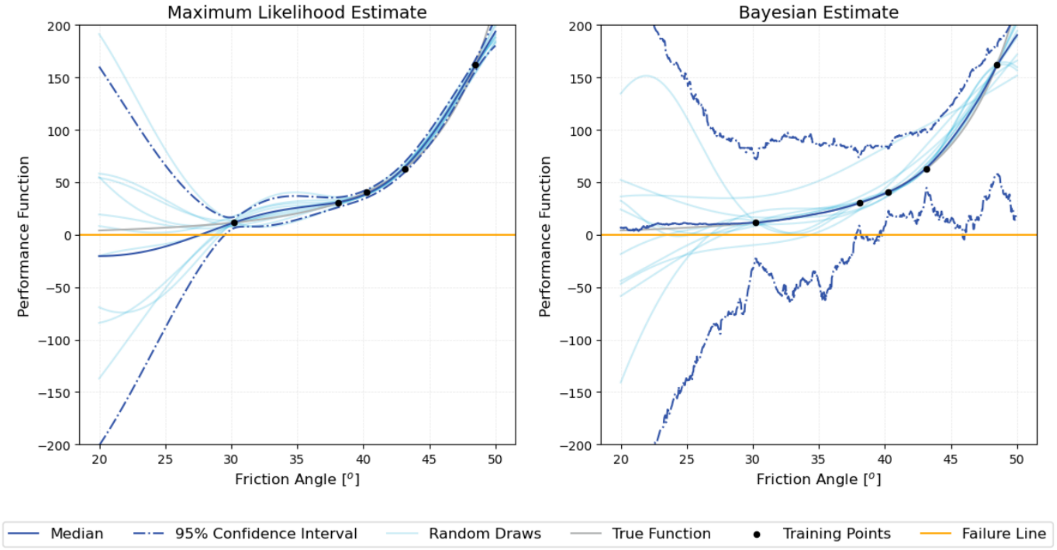


Figure 3.5: Reliability Analysis Illustrative Example

4

Research Methods

This chapter discusses how the concepts and algorithms presented in Chapter 3 were implemented to achieve the target results. It goes through specifics such as types of geotechnical problems considered, assumptions on distributions, sampling method employed, algorithms used, etc.

4.1. Problem Definitions

Three problems were considered for the research project. The first two problems involved two stochastic variables and used simpler analytical solutions for the true function, The third problem involved 9 stochastic variables, and was set up as a reliability analysis such than the true function was defined by a performance function. The last problem utilized a more complex numerical solution in order to obtain the value of the performance function.

4.1.1. Case 1: Retaining Wall

A retaining wall factor of safety problem was chosen for two reasons: (1) this is a textbook problem with well-defined and popular analytical solutions for the factors of safety, and (2) it had multiple modes of failure, which was expected to add non-linearity to the solution. Figure 4.1 shows a simple representation of this problem.

This problem has three failure modes (Das, 2007): sliding, overturning, and bearing capacity failure. The factor of safety for sliding was calculated using Equation 4.1.

$$FS_{\text{slide}} = \frac{\text{resisting force}}{\text{driving force}} \tag{4.1}$$

The driving force is caused by Rankine's active lateral force and calculated using Equation 4.2 in the case of cohesionless soil. While the soil under the wall was cohesive, the soil above the

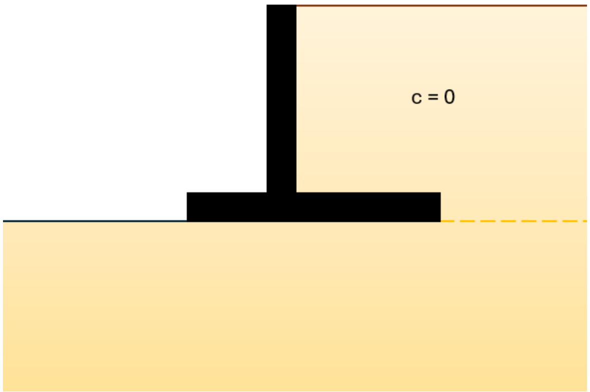


Figure 4.1: Retaining Wall Problem

base of the wall was assumed cohesionless to simplify the calculations by preventing tension cracks between the wall and the soil.

$$\text{driving force} = \frac{1}{2}\gamma K_a H_{wall}^2 = \frac{1}{2}\gamma \frac{1-\sin(\phi)}{1+\sin(\phi)} H_{wall}^2 \tag{4.2}$$

Where:

γ	: soil unit weight
K_a	: Rankine active lateral earth pressure coefficient
ϕ	: soil friction angle
H_{wall}	: wall height

The resisting force is caused by friction and adhesion under the base of the wall. The interface friction was assumed to be 5° less than the soil friction angle and the adhesion was taken as half of the soil cohesion. The resisting force was calculated using Equation 4.3.

$$\text{resisting force} = P_v \tan\phi_{\text{int}} + B_{wall} c_{\text{adh}} \tag{4.3}$$

Where:

P_v	: total weight of wall and soil above wall base
ϕ_{int}	: interface friction between wall and soil
B_{wall}	: width of wall base
c_{adh}	: adhesion between wall and soil

The factor of safety for overturning is equal to the ratio of the righting and overturning moments as shown in Equation 4.4.

$$FS_{\text{overturn}} = \frac{\text{righting moment}}{\text{overturning moment}} \tag{4.4}$$

The overturning moment was calculated by multiplying the driving force by its moment arm from the base of the wall. Similarly, the righting moment was calculated by multiplying the weight of the wall and soil by their respective moment arms from the toe of the wall.

Lastly, the factor of safety for bearing capacity was calculated using Meyerhof's bearing capacity equation for an eccentric footing. The bearing capacity of a shallow footing under an eccentric load was first calculated following Terzaghi's bearing capacity equation with Meyerhof's modifications. The general equation for the bearing capacity is given in Equation 4.5. The surcharge load was conservatively taken as 0, since the toe of the footing lies on ground surface.

$$q_u = c \cdot N_c + q \cdot N_q + \frac{1}{2} \cdot \gamma \cdot B' \cdot N_\gamma \tag{4.5}$$

Where:

q_u	: ultimate bearing capacity
q	: surcharge load at the level of the footing base
N_c, N_q, N_γ	: Meyerhof's bearing capacity factors
B'_{wall}	: effective footing width due to eccentricity = $B_{wall} - 2 \cdot e$
e	eccentricity of load

Empirical equations for Meyerhof's bearing capacity factors were used as shown in Equations 4.6, 4.7, and 4.8.

$$N_q = e^{\pi \tan(\phi)} \tan\left(45^\circ + \frac{\phi}{2}\right)^2 \tag{4.6}$$

$$N_c = (N_q - 1) \cdot \frac{1}{\tan(\phi_)} \tag{4.7}$$

$$N_\gamma = (N_q - 1) \cdot \tan(1.4\phi) \tag{4.8}$$

The load eccentricity was calculated using Equation 4.9.

$$e = \frac{B}{2} - \frac{\text{righting moment} - \text{overturning moment}}{P_v} \tag{4.9}$$

The maximum acting pressure is then calculated. If the resultant vertical load acts within the middle-third of the footing, the maximum pressure, q_{max}, can be calculated using Equation 4.10.

$$q_{max} = \left(\frac{P_v}{B}\right)\left(1 + \frac{6e}{B}\right) \tag{4.10}$$

The factor of safety for bearing capacity can then be calculated by dividing the ultimate bearing capacity with the maximum bearing pressure as shown in Equation 4.11. In the case where the resultant vertical load acts outside the middle-third of the footing, tension forces between the footing and the soil underneath are developed. Since soil has negligible tensile strength, this is generally not allowed and footings are redesigned. For this scenario, the factor of safety is set to 0 when the resultant vertical load acts outside the middle-third of the footing.

$$FS_{\text{bearing pressure}} = \frac{q_u}{q_{max}} \tag{4.11}$$

The factor of safety of the retaining wall was taken as the minimum of the factors of safeties calculated using Equations 4.1, 4.4, and 4.11. Slight artificial noise in the form of a Gaussian distribution with a variance of 0.1 was added to the results of the true analytical solution of the factor of safety.

In the metamodelling of this problem, friction angle and cohesion were treated as the stochastic variables. The distributions of these stochastic inputs were defined as log-normal distributions with moments as shown in Table 4.1. The distributions were given artificially larger uncertainties than usual to provide a wider selection of possible learning points. However, an upper-limit of 50^o was applied to the friction angle to ensure realism of possible sampled values. The sampled population is shown in Figure 4.2.

Table 4.1: Stochastic Properties of Inputs

Stochastic Variable	Distribution	Log-mean	Log-std
Friction Angle	Log-normal	20°	8°
Cohesion	Log-normal	20 kPa	5 kPa

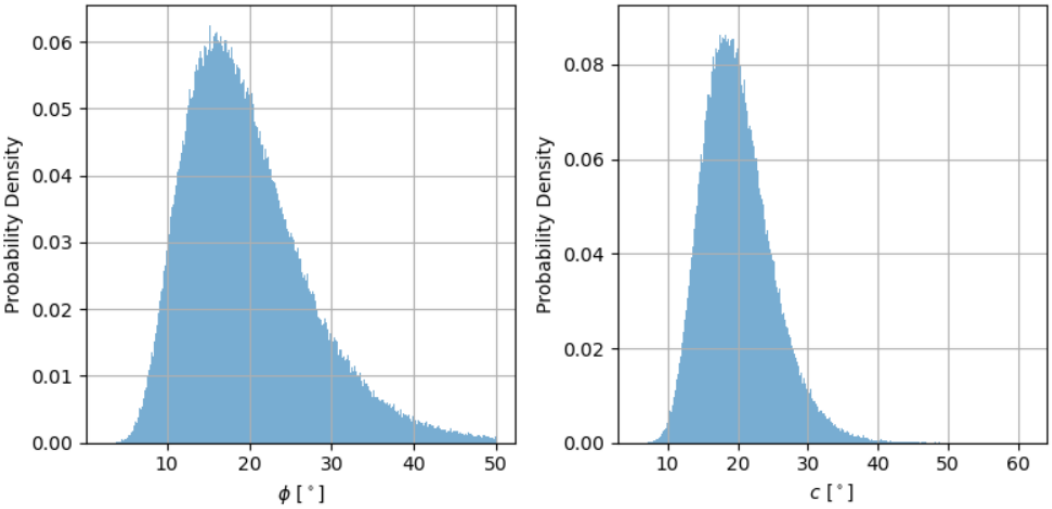


Figure 4.2: Population of Inputs for Retaining Wall Problem

4.1.2. CASE 2: ONE-DIMENSIONAL SOIL CONSOLIDATION

As shown in Section 3.4, maximum likelihood estimate of hyperparameters may lead to inaccurate predictions at certain regions when the stochasticity of the input variables are defined by wide distributions. Therefore, another simple geotechnical problem involving one-dimensional consolidation was also tested. The solution described below for one-dimensional consolidation settlement can be used to approximate the settlement of a consolidating layer under a uniformly distributed surface load applied over a large area. The consolidation settlement of soil depends on hydraulic conductivity, which can vary widely with uncertainties spanning several orders of magnitude.. It also depends on soil stiffness, which, along with the hydraulic conductivity, defines the soil's coefficient of consolidation, as shown in Equation 4.12. In the case of one-dimensional consolidation, the oedometer stiffness is used, which can be calculated from the soil stiffness using Equation 4.13.

$$c_v = \frac{k \cdot E_{oed}}{\gamma_w} \tag{4.12}$$

$$E_{oed} = E\frac{1-\nu}{(1+\nu)(1-2\nu)} \tag{4.13}$$

Where:

c_v	: soil coefficient of consolidation
k	: hydraulic conductivity
E_{oed}	: oedometer stiffness
E	: soil stiffness
ν	: Poisson's ratio

The settlement at any given time t can be calculated as a factor of the end-of-consolidation settlement. This factor is defined by the average degree of consolidation, which is related to the soil's time factor. Figure 4.3 shows Terzaghi's consolidation chart, which provides the relationship between the degree of consolidation and the time factor. Empirical equations that approximate this relationship have been made, such as those shown in Equation 4.14. This was used in lieu of the chart for easier implementation in the coding algorithm.

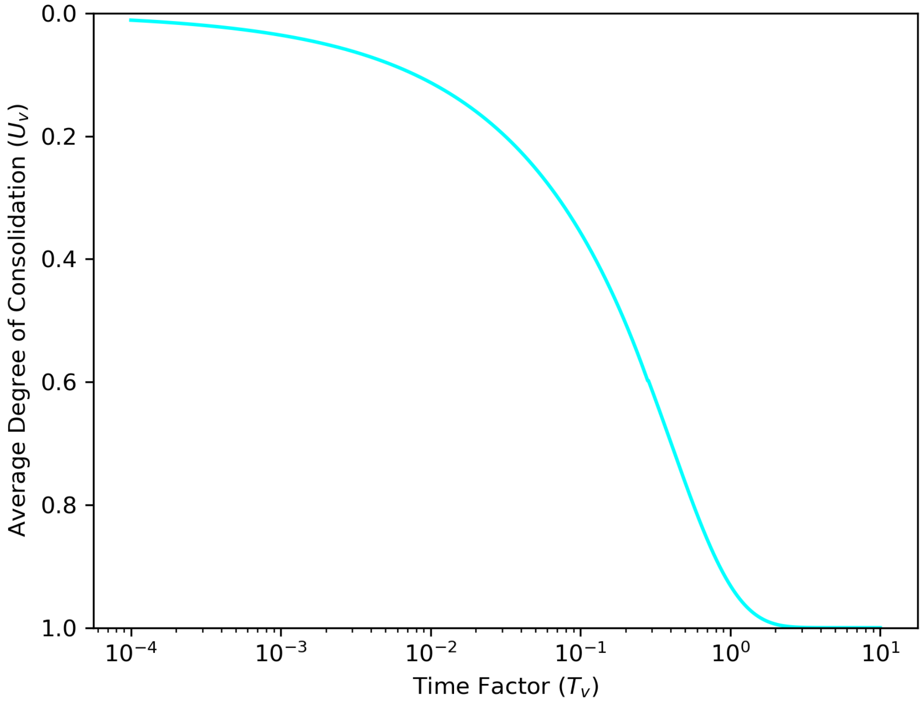


Figure 4.3: Terzaghi's Consolidation Curve

$$T_v = \begin{cases} \frac{\pi}{4} \cdot U_v^2 & \text{if } U_v < 0.6 \\ -0.933 \cdot \log_{10}(1 - U_v) - 0.085 & \text{if } U_v \geq 0.6 \end{cases} \tag{4.14}$$

Where:

T_v : time factor for soil consolidation
U_v : degree of consolidation

The time factor is a function of the coefficient of consolidation, the drainage path length, and the time allotted for consolidation. This is shown in Equation 4.15.

$$T_v = \frac{c_v \cdot t}{H_{dr}^2} \tag{4.15}$$

Where:

t : times
H_{dr} : length of drainage path

The settlement was calculated with Equation 4.16.

$$s_t = \frac{q \cdot H}{E_{oed}} U_v \tag{4.16}$$

Where:

s_t : settlement at time t
q : uniform surface load

Hydraulic conductivity and soil stiffness were considered as the stochastic variables for this problem. Table 4.2 shows the moments of the distribution used to define the stochasticities of these variables. The resulting sampled population is shown in Figure 4.4.

Table 4.2: Stochastic Properties of Inputs for Consolidation Problem

Stochastic Variable	Distribution	Log-mean	Log-std
Hydraulic Conductivity	Log-normal	$1x10^{-8}$ m/s	$1x10^{-8}$ m/s
Soil Stiffness	Log-normal	15000kPa	1000kPa

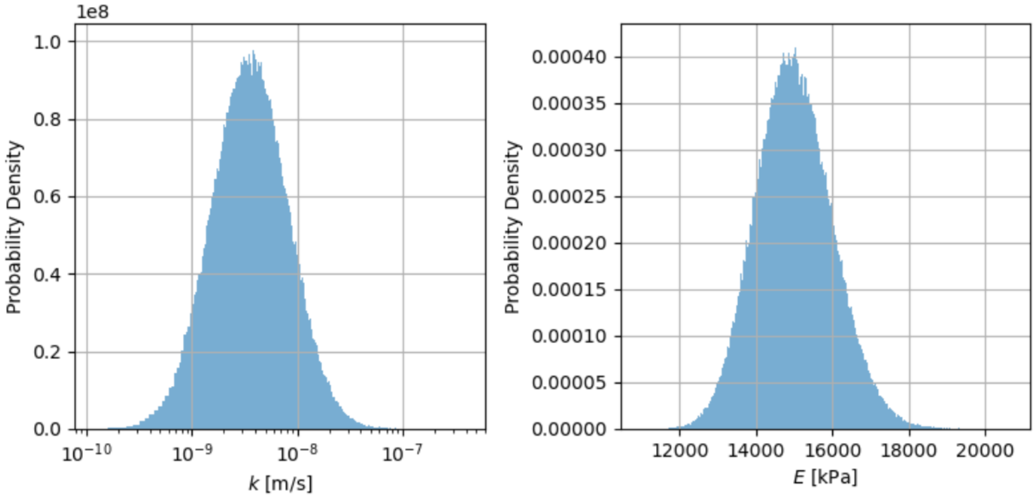


Figure 4.4: Population of Inputs for Consolidation Problem

4.1.3. CASE 3: PILED EMBANKMENT

The performance of the methods when applied to more complex numerical problems was also investigated. In addition, the application of the method to geotechnical reliability analysis was explored. The case of a piled embankment built on top of a layered soil was considered. An illustration of this problem is shown in Figure 4.5. The soil stratigraphy is composed of a 2-meter sand layer overlying a 13-meter clay layer, with the groundwater lying at the interface of these two layers. A 20-m wide and 4-m high embankment with side-slopes of 3:1 was then built on the ground surface. The construction of the embankment was divided into four 1-meter intervals, with each construction stage lasting two days. The long-term settlement of the piled embankment was used to define the performance function as shown in 4.17. In this case, the failure is defined as the long-term settlement being greater than the allowable settlement. Note that allowable settlements of embankments are typically discussed with the client of a project; therefore, no actual allowable value is listed in geotechnical codes. A common guideline allows for a settlement equal to 10% of the embankment height. However, a stricter allowable settlement of 150 mm was used for this problem.

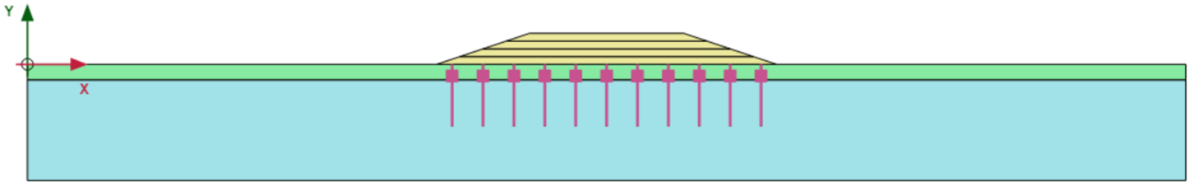


Figure 4.5: Piled Embankment in Plaxis

$$g(x) = \frac{s_{allow}}{s(x)} - 1 \tag{4.17}$$

Where:

s_{allow} : allowable long-term settlement

$s(x)$: settlement of embankment after 50 years

The clay layer was modelled with a Modified Cam-Clay (MCC) constitutive model, while the sand and embankment layers were modelled with a Hardening Soil model. Nine stochastic soil parameters were defined as shown in Table 4.3. These properties were particularly chosen as those that may have the most effect on soil settlement.

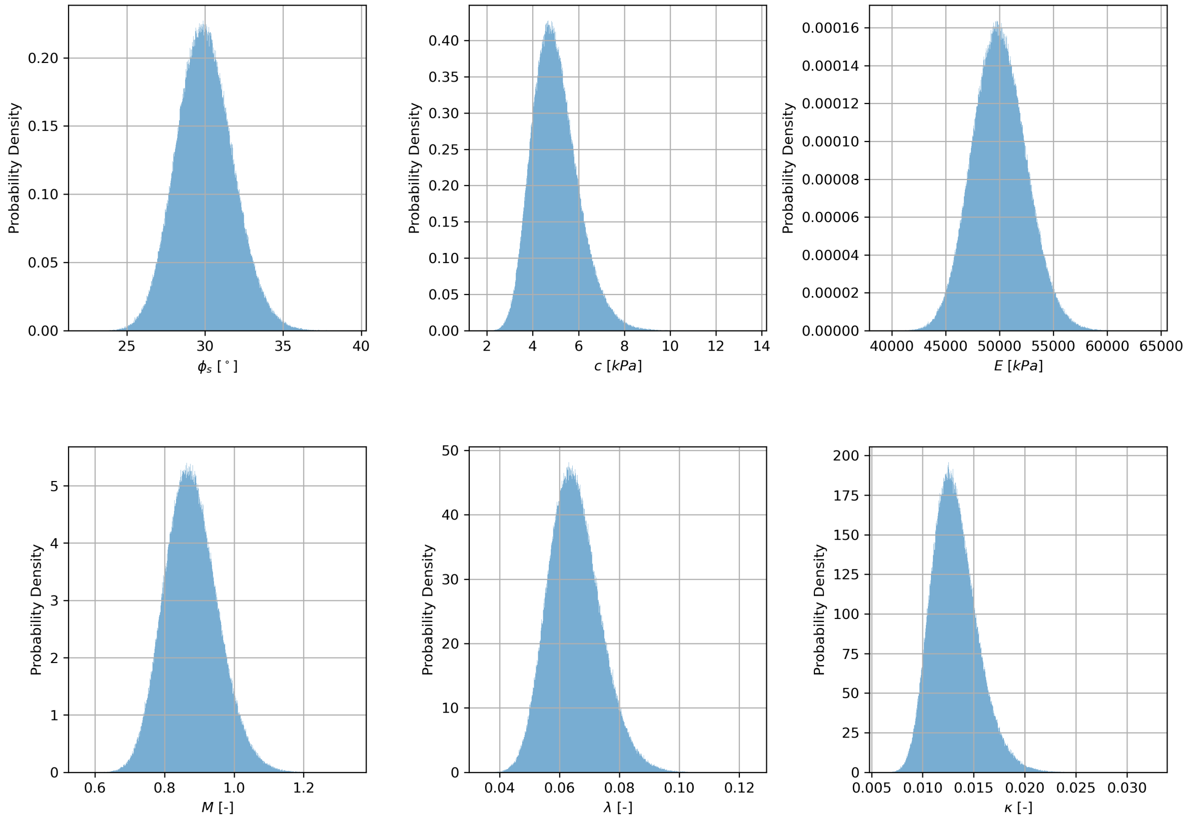
The mean values for friction angle, cohesion, and stiffness were taken from "NEN 9997-1", 2016. The slope of the critical state line used for the Modified Cam-Clay model of the clay layer was calculated from the clay friction angle using Equation 4.18. To define the uncertainties around the means, the general range of values specified in "NEN 9997-1", 2016 and the distributions used in previous geotechnical engineering reliability research (ex. Al-Bittar et al., 2018) were used as guidelines. The means and uncertainties of the compression index λ represented the range of values for clay with medium deformability as specified in Kulhawy and Mayne, 1990. The distribution of swelling index was then calculated from the values of compression index, given the ratio shown in the table. Hydraulic conductivity values and uncertainties were defined to resemble the range of values provided in Das, 2007. Lastly, the uncertainty for the overconsolidation ratio of the clay layer was defined to represent lightly to moderately overconsolidated soils. Other soil properties were automatically defined based on default values or relationships with these stochastic variables (*PLAXIS 2D 2024.1 Material Models Manual*, 2023). In addition to these, the distributions were also adjusted such that there is a low probability of defining an invalid material and the probability of failure is between 10^{-5} and 10^{-3}. The latter was done through short preliminary runs.

Table 4.3: Stochastic Properties of Inputs for Piled Embankment Problem

Stochastic Variable	Distribution	Mean	Standard Deviation
Sand Friction Angle (ϕ_s)	Log-normal	30°	1.8°
Sand Cohesion (c)	Log-normal	5 kPa	1 kPa
Sand Stiffness (E)	Log-normal	50000 kPa	2500 kPa
Clay Friction Angle (ϕ_c)	Log-normal	22.5°	1.8°
Clay Compression Index (λ)	Log-normal	0.15	0.02
Index Ratio ($\frac{\lambda}{\kappa}$)	Normal	5	0.5
Sand Hydraulic Conductivity (k_s)	Log-normal	2e-4 m/s	1e-4 m/s
Clay Hydraulic Conductivity (k_c)	Log-normal	1.5e-8 m/s	2e-8 m/s
Clay OCR	Log-normal	2	0.3

$$M = \frac{6\sin(\phi_c)}{3 - \sin(\phi_c)} \tag{4.18}$$

Figure 4.6 shows the histogram of the population of each stochastic variable used as input for the piled embankment problem.



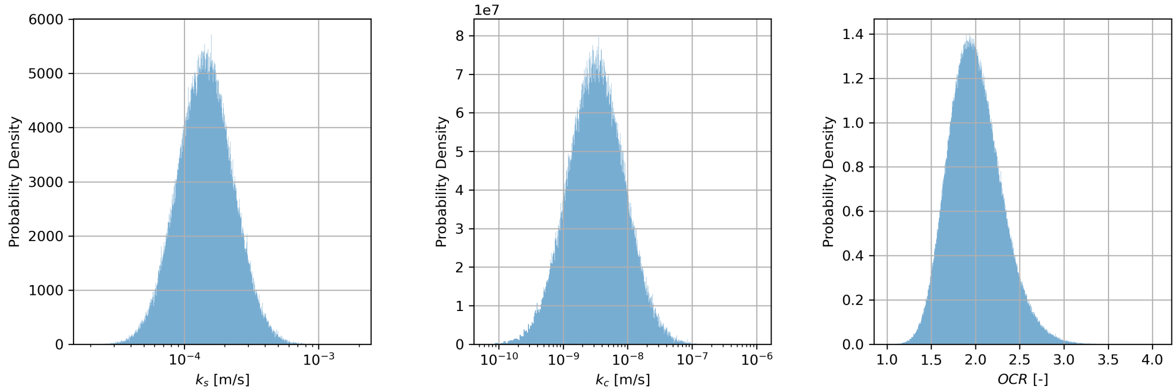


Figure 4.6: Population of Inputs for Piled Embankment Problem

4.2. SURROGATE MODELLING WITH MLE

For each input variable, 1,000,000 samples were randomly taken from the corresponding distributions and concatenated. This represents the total population of input points from which training points can then be selected and added to the design of experiments. For the first two case problems, an additional 1,000 sets of inputs were generated, whose true functions were evaluated. These were used as the validation set in the calculation of the MSE.

From the 1,000,000 possible training points, 5 were randomly selected as the initial design of experiments. A small number of samples were taken initially since the comparison of early performances between MLE and Bayesian approaches was also of interest. The same set of input data was used for both MLE and Bayesian optimization methods, such that the only difference was the optimization methods.

The true function values for the initial DoE were calculated. The initial DoE was then used to build the first metamodel following the equations presented in Section 3.1. A Matérn-$\frac{5}{2}$ autocorrelation function was used. For each problem, hyperparameters include the length scales for each of the stochastic variables, a kernel scaling factor, and a Gaussian process constant noise. Thus, the simple analytical problems involved 4 hyperparameters, while the numerical problem involved 11 hyperparameters.

PyMC does not have a built-in maximum likelihood estimator. Instead, a maximum a posteriori estimator was used, where uniform prior distributions are defined. This approach yields the maximum likelihood estimate, provided the bounds of these distributions are well-defined. Bounds of these distributions were defined as shown in Table 4.4. L-BFGS was used to find the maximum likelihood estimate. The initial guesses for the length scales were specified as the means of the corresponding populations of variables, and the initial values for both kernel scale and Gaussian process noise were set to 1. A single gradient-descent path was used.

Table 4.4: Bounds of Prior Distributions

Hyperparameter	Bounds
Length Scale of x	[min(x)/100, max(x)*100]
Kernel Scale	[10^{-6}, 10]
GP Standard Deviation	[10^{-6}, 10]

Once the optimal hyperparameters had been determined, predictive means and variances were calculated for the entire population set. These results were then used to calculate the learning function values of each point, and the point with the highest learning function value was chosen and added to the existing DoE. For the analytical problems, where the interest of the investigation was the accuracy of the models, the learning function used was simply the variance of the predictions, such that the point with the highest predictive uncertainty was added to the DoE. For the piled embankment reliability analysis problem, the U learning function was used as defined in Equation 3.13. When a point was in a sparse location of the input space, it may happen that it has the highest uncertainty even if it was already in the DoE. To prevent re-adding a point already in the DoE, the learning function values were sorted in descending order, and the point with the highest learning function value that was not currently in the DoE was added. In this case, the negative of the U learning function was used instead. The true function value of the newly added learning point was calculated. Predictive values for the validation sets were also calculated and the MSEs between the predictions and true values of the validation sets were calculated.

From the new DoE, a new metamodel is built. Predictions, learning function values, and MSE's of the validation points were recalculated. For the analytical problems, one hundred iterations were performed and the evolution of the MSE values was observed.

For the reliability analysis problem, the accuracy of the calculated probability of failure is defined by a convergence criterion related to the bounds of the limit state surface, as shown in Equation 4.19. The bounds $\hat{P}_f^+$ and $\hat{P}_f^-$ can be conservatively defined with Equation 4.20 (Schöbi et al., 2017). The enrichment was done until the convergence criterion reached a value of 0.05, as suggested by Schöbi, with a minimum of 100 iterations to better see and compare the behavior of the two methods.

$$\frac{\hat{P}_f^+ - \hat{P}_f^-}{\hat{P}_f} = \epsilon_{\hat{P}_f} \tag{4.19}$$

$$\hat{P}_f^{\pm} = \mathbb{P}\left(\mu_{\hat{y}} \mp 1.96\sigma_{\hat{y}}\right) \leq 0 \tag{4.20}$$

Where:

$\epsilon_{\hat{P}_f}$: convergence criterion for probability of failure

4.3. Surrogate Modelling with Bayesian Optimization

The same sets of inputs from the MLE metamodels were used as the starting points for the metamodels using a Bayesian approach. As discussed earlier, one main advantage of the Bayesian approach over a simple MLE method is the ability to define prior assumptions regarding the distribution of hyperparameters. Due to the lack of previous studies in this area, informative prior distributions cannot yet be provided and non-informative priors were used. Since hyperparameters of Gaussian processes can vary across different magnitude orders depending on the uncertainty of the input variables as well as the behavior of the function with respect to the inputs, log-uniform distributions were defined as priors. This suggests that length scale values across different orders of magnitude have equal probabilities of defining the observed data. While not strongly informative, the use of log-uniform priors is expected to perform better and provide more information than uniform priors; the normalization of different scale magnitudes should prevent the posterior from being highly peaked at the maximum likelihood estimate. It should also aid the sampler in finding the correct scale magnitudes of the hyperparameters. The same bounds used to define the prior uniform distributions in the MLE method were used as the bounds of the prior log-uniform distributions for the Bayesian method. PyMC's implementation of NUTS was used to build the distributions, where 4 chains of 1000 samples each were run. Prior to the sampling process, PyMC performs a tuning stage, in which it runs a pre-sampling mechanism while also adjusting the step size of the NUTS sampler to meet a user defined acceptance rate. For all problems, an acceptance rate of 95% was set after preliminary runs showed that lower acceptance rates resulted in multiple divergences within the sampling chains. In addition, the number of tuning steps was set to 1000 for the analytical problems but was increased to 2000 for the complex numerical problem. The samples obtained during the tuning stage are discarded, thus also acting as the burn-in period of the sampler.

During preliminary runs of the Bayesian method, it was found that usage of all the sampled sets of hyperparameters for prediction led to enormous and impractical computation times. To alleviate this, the chains were greatly thinned, with the idea that hyperparameter samples with high autocorrelations provide similar information regarding predictions and predictive uncertainties. Chain thinning reduces the autocorrelation between samples and greatly reduces computation time, at the expense of providing less predictive samples of hyperparameters. Consequently, more predictive samples of functions were made for each hyperparameter sample in the thinned posterior. In this case, the hyperparameter sample chains were thinned by a factor of 40 (i.e. every 40th accepted sample in each chain is kept), and 10 predictive samples of functions were made each hyperparameter. This led to 1000 predictions for each set of stochastic variable inputs. The use of more predictive samples is expected to provide a better representation of the uncertainty in each Gaussian process; however, memory limitations prevented this.

To properly draw functions for each hyperparameter value, the posterior mean and covariance matrices are needed. Functions are then drawn from the multivariate normal distribution defined be these matrices. Here lies a problem when dealing with multiple predictions points, as the full posterior covariance matrix is a square matrix with the dimension equal to the number of prediction points and training points combined. The memory requirement needed to store this matrix greatly surpasses that of standard computing machines when the number of prediction points is in the range of 1,000,000.

Thus, instead of sampling full functions spanning the entire prediction set, predictive samples of the function value for each prediction point were instead made individually. This incorrectly implies that the prediction points are uncorrelated and jumps between function values become present. This may have a consequence of an increase in the spread of possible function values for each point. The results of this simplified method was compared to a less simplified approaches, involving providing more predictive samples as well as lessening the sparsity of the posterior covariance matrix. Due to memory and computation time limitations, the comparison was done on a smaller prediction dataset.

The entire process of generating Bayesian predictive samples is detailed in Algorithm 3. A code snippet is provided in Appendix Listing 7.2; the code used for MLE metamodelling is also provided in Appendix Listing 7.1 for reference. The final output of the Bayesian metamodel is a two-dimensional array, containing predictive function values for each point in the prediction set.

Algorithm 3 Generating Bayesian Predictive Samples

```
1: Prior distributions for each hyperparameter type are defined with appropriate bounds.
2: A Gaussian process with a zero mean and a Matérn-$\frac{5}{2}$ kernel is defined and conditioned
   on the DoE.
3: From the prior distributions and likelihood of currently known data, a joint posterior
   distribution is built with NUTS with 4 chains and 1000 samples per chain.
4: Chains are thinned by a factor of 40, leading to 100 hyperparameter sample sets.
5: For each hyperparameter sample set, the predictive means and variances at all prediction
   points are calculated individually following Equations 3.11 and 3.12.
6: For each hyperparameter sample set, 10 predictive function values are sampled from the
   calculated means and variances for each prediction point.
```

Predictions were made again for the population and validation sets. The medians of the predictive function values of the points in the validation set were used in calculating the MSE. Similar to the MLE approach, enrichment continued until no further discernible improvement in the MSE values of the validation set was observed, with a minimum of 100 iterations. For each iteration, the traces of hyperparameter samples as well as the MSE of the validation set were stored in lists.

For the reliability analysis of the piled embankment, modifications were made to Equation 4.20 to account for the non-Gaussian nature of the predictive samples. Percentiles were used instead of the mean and standard deviation to define the boundaries of the limit state, as shown in Equations 4.21, 4.22, and Equation 4.23. Recall from 3.24 that $P50$, $P2.5$, and $P97.5$ are the median, 2.5^{th}, and 97.5^{th} percentiles of the predictive function values for each point. The use of percentiles instead of the mean and standard deviation allows for greater flexibility in analyzing distribution shapes and provides robustness in dealing with outliers.

Equations 3.17 and 4.19 were still used to calculate the coefficient of variation and the convergence criterion for the probability of failure.

$$\hat{P}_f = \mathbb{P}(P50 \leq 0) \tag{4.21}$$

$$\hat{P}_f^{+} = \mathbb{P}(P2.5 \leq 0) \tag{4.22}$$

$$\hat{P}_f^{-} = \mathbb{P}(P97.5 \leq 0) \tag{4.23}$$

Code snippets of the implementations of the learning functions and calculation of P_f and its convergence criterion are provided in Appendix Listings 7.3 and 7.4, respectively.

5
Results and Discussion

5.1. Case 1: Retaining Wall Factor of Safety

Figure 5.1 shows the evolution of the *MSE* with every iteration of the active learning process for both MLE and Bayesian approaches. Recall that both methods started with the same initial design of experiments with 5 training points. The initial runs of the MLE approach produce higher magnitudes of the *MSE*, while the Bayesian approach has a faster decrease in MSE, with a steep decline in the first 5 iterations. As more realizations are added, the *MSE* for the MLE method gradually becomes slightly lower than that of the Bayesian method. After 100 iterations, the *MSE*'s for both method seem to have negligible decrease in value. The final *MSE* was 0.0256 for the MLE method and 0.0287 for the Bayesian method. To further understand how the two methods differ, a comparison of the optimized hyperparameters and the added learning points is discussed below.

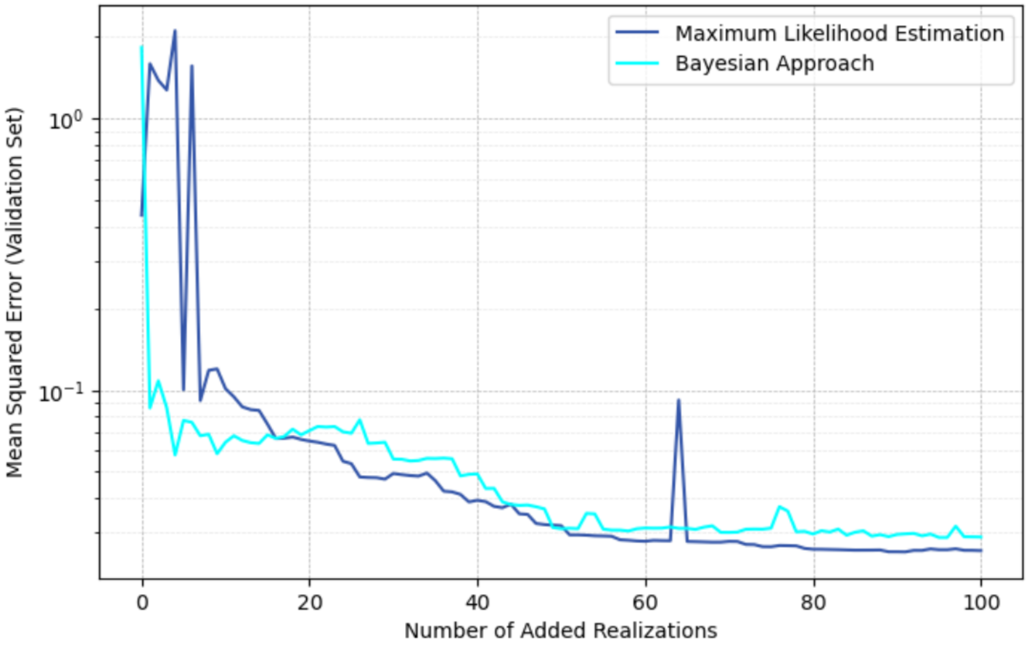


Figure 5.1: Mean Squared Error for Retaining Wall Problem

Figure 5.2 shows the optimized hyperparameters of every iteration of the active learning process of both the MLE and Bayesian approaches. In each iteration, the maximum likelihood estimates are plotted, whereas the Bayesian posterior samples are displayed in a scattered format to resemble a histogram. Generally, the MLE approach required more learning points to find a good estimate of the hyperparameter values. Specifically, in the case of the friction angle length scale and Gaussian process noise, the first 10 iterations resulted in large maximum likelihood estimates, indicating that the metamodel predominantly interpreted the true behavior as noise. This contributed to the higher initial MSE values. The Bayesian approach obtained better samples of friction angle length scales at very early iterations, with less than 5 learning points.

In later iterations, the hyperparameter posterior distributions appear to be bounded between two peak locations, as evidenced by the maximum likelihood estimates across most iterations and the notable jump at iteration 65. It also becomes thin in the log scale, which is likely due to the higher number of learning points, lower leap-frog step size caused by the higher acceptance rate in the tuning stage, and weakly informative prior.

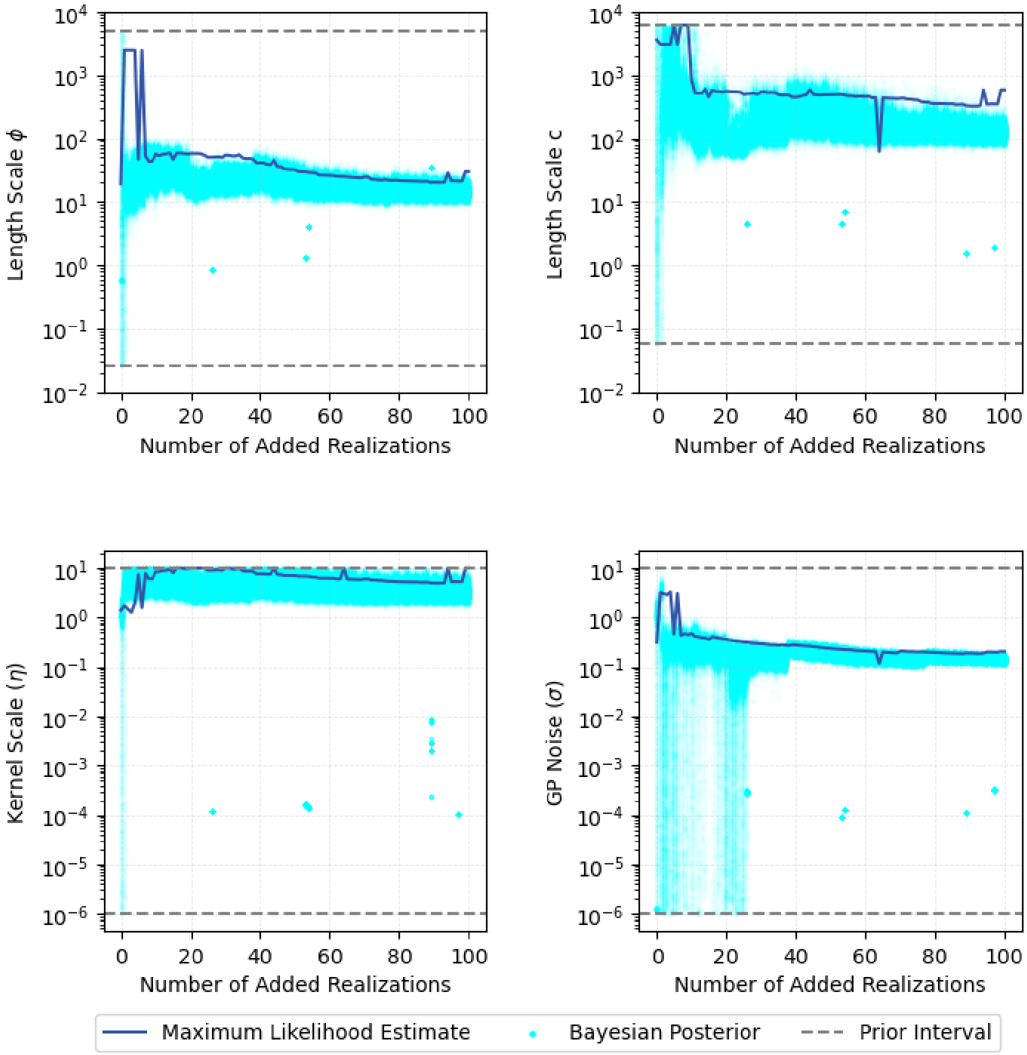


Figure 5.2: Optimized Hyperparameters for Retaining Wall Problem

There is a slight deviation between the length scales optimized by the two methods in the final iterations. Figures 5.3 and 5.4 show log-likelihood values after 100 iterations calculated for different combinations of the length scales of friction angle and cohesion, where the kernel scale and GP noise were fixed at the maximum likelihood estimates. Note that there is a difference in the log-likelihood values between the two methods, attributed to differences in the design of experiments. Also shown are the maximum likelihood estimates and the samples taken by the NUTS sampler. The MLE method failed to find the true peak; however, it still found a point with a log-likelihood value almost equal to that of the true peak. It can be seen visually that Bayesian sampling with NUTS was effective as it was able to generate samples that are concentrated in the mode of the joint distribution. However, there is a low spread of hyperparameter samples in the Bayesian posterior, which again is maybe due to the high acceptance rate in the tuning stage and the weakly informative prior. There are some outliers in some iterations. These were still accepted as part of the sampling process due to the possibility of having multiple modes.

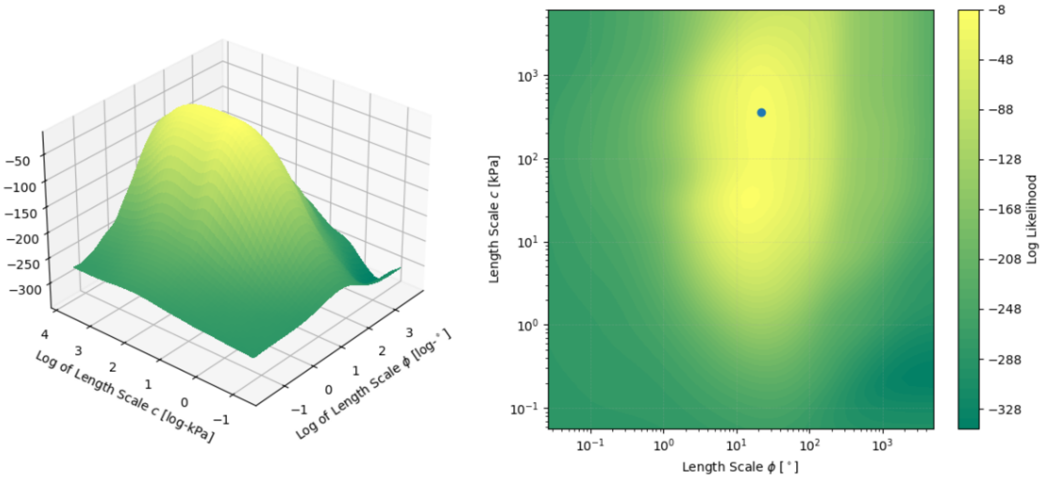


Figure 5.3: Log-likelihoods for MLE Iteration 100

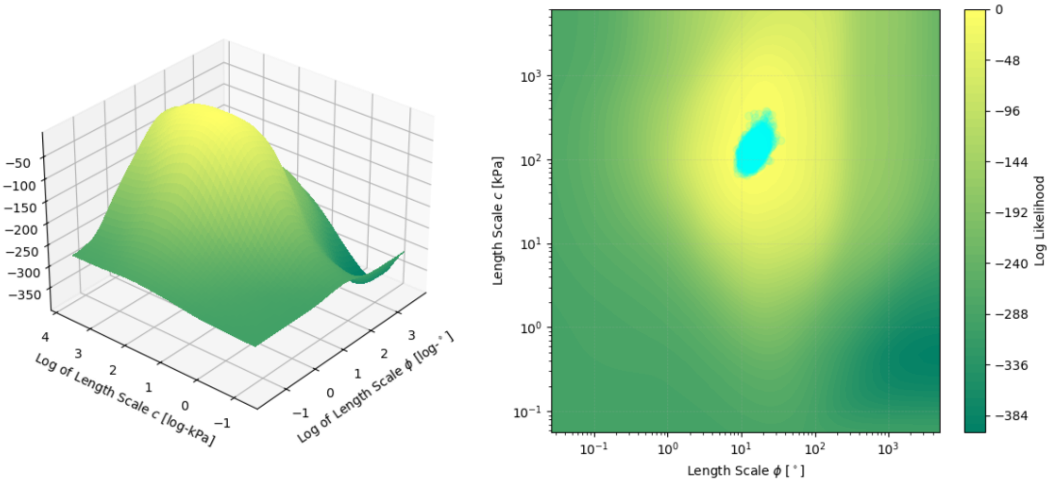


Figure 5.4: Log-likelihoods for Bayesian Iteration 100

Figure 5.5 shows the learning points that were added after selected numbers of iterations. In the initial iterations, the Bayesian approach took points that have less correlations compared to the MLE method, which tended to choose points clustered in specific areas. This may have caused the faster determination of the appropriate range of length scale values for friction angle and reduction of the *MSE*. In later iterations, the MLE method acquired learning points mostly at the bounds of the population distribution, while the Bayesian method included points near the center, leading to a more evenly spread design of experiments.

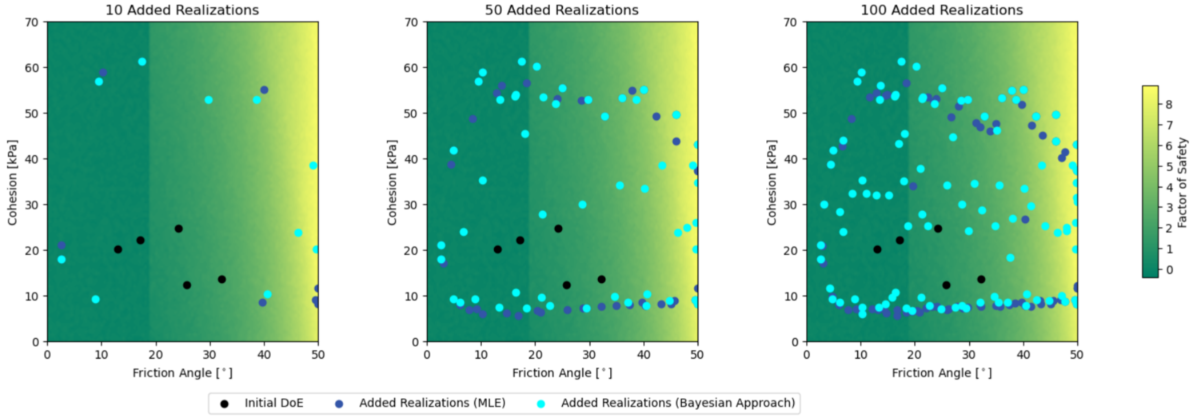


Figure 5.5: Enrichment for Retaining Wall Problem

The predictions made from the final models for the retaining wall problem are shown in Figure 5.6. While both methods were able to capture the general behavior of the factor of safety, neither was able to fully replicate the steep drop to a factor of safety of zero at friction angle of around 20^o. Visually, there is negligible difference between the predictions of the two surrogate model, as evidenced by the minor difference in MSE values.

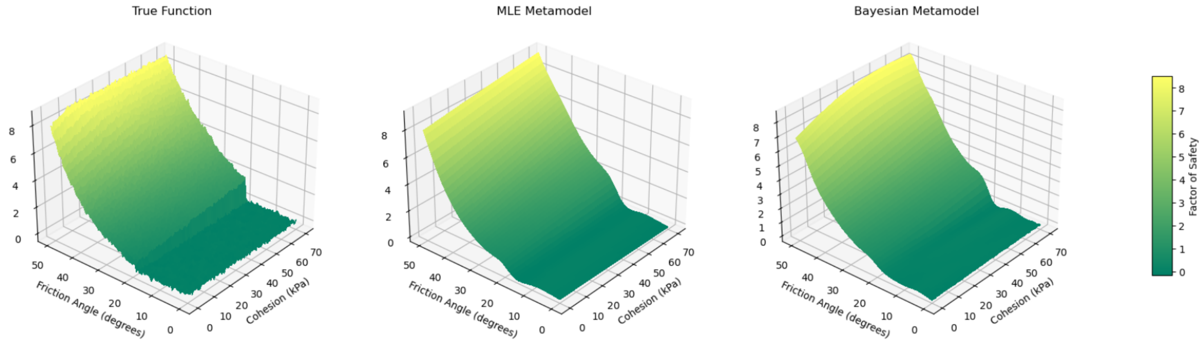


Figure 5.6: Metamodel Predictions for Retaining Wall Problem

Figure 5.7 shows the credible and predictive intervals of the two methods after 20 iterations. The credible interval defines the uncertainty in predictions due solely to the hyperparameter uncertainty, while the predictive interval combines both hyperparameter uncertainty and the uncertainty inherent in each Gaussian process. When, comparing the uncertainty of the MLE and Bayesian predictions for this problem, the variability in the hyperparameter space contributes only a slight increase in the overall uncertainty of the predictions.

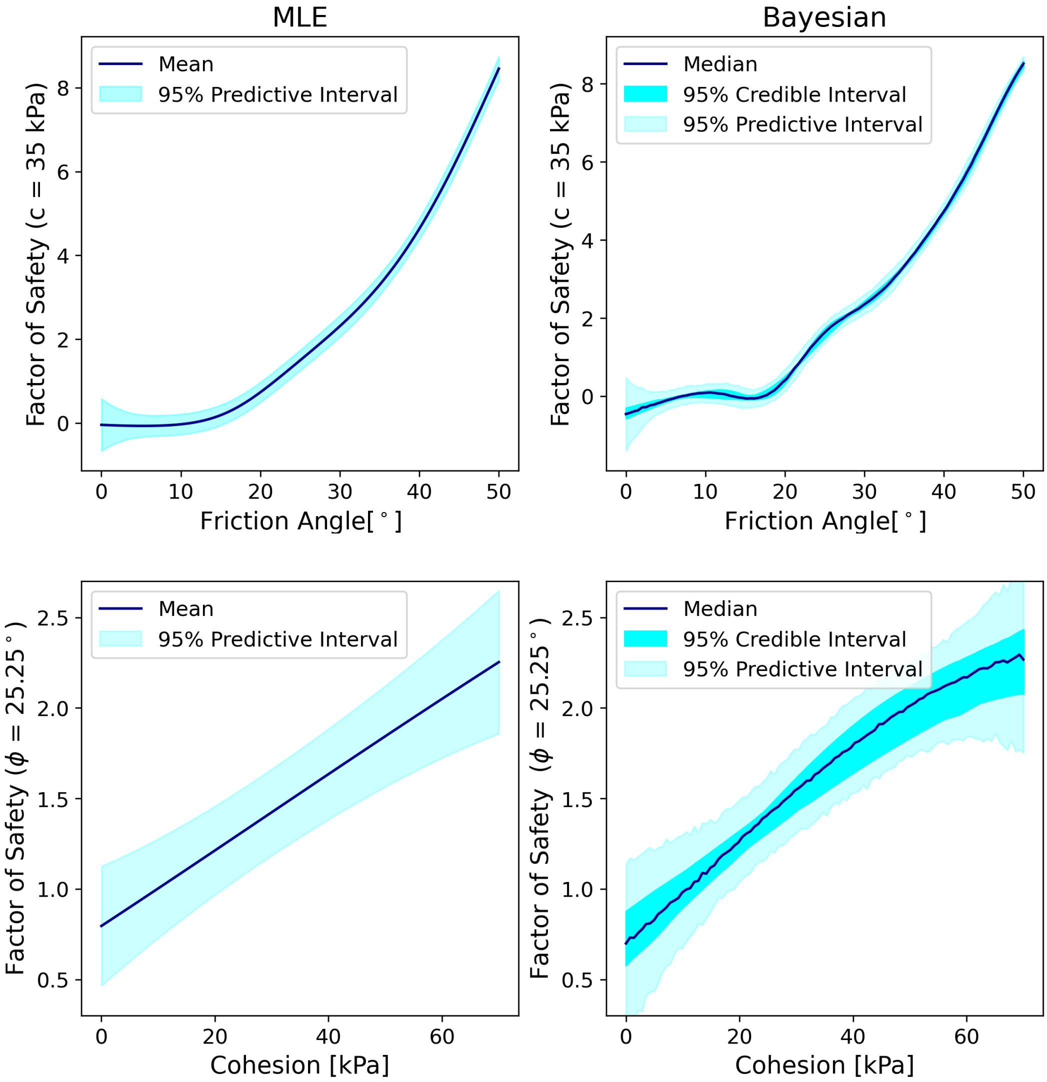


Figure 5.7: Predictive Intervals for Retaining Wall Problem

5.2. CASE 2: CONSOLIDATION SETTLEMENT

Figure 5.8 shows a comparison between the evolution of the *MSE* of the MLE and Bayesian methods for the consolidation problem. In this case, the differences between the *MSE* values and the number of iterations to reach a low value are more pronounced, with the Bayesian method achieving an *MSE* of $5.34 \cdot 10^{-7} m^2$ after 52 added realizations, whereas the MLE method reached its lowest *MSE* of $8.96 \cdot 10^{-7} m^2$ after 88 added realizations. Jumps in *MSE* values are observed in the MLE approach after 60 iterations, while the Bayesian values are consistently in the lower ranges of *MSE* values. To better understand the behavior of this *MSE* evolution, analyses of the sampled hyperparameters are presented in Figure 5.9.

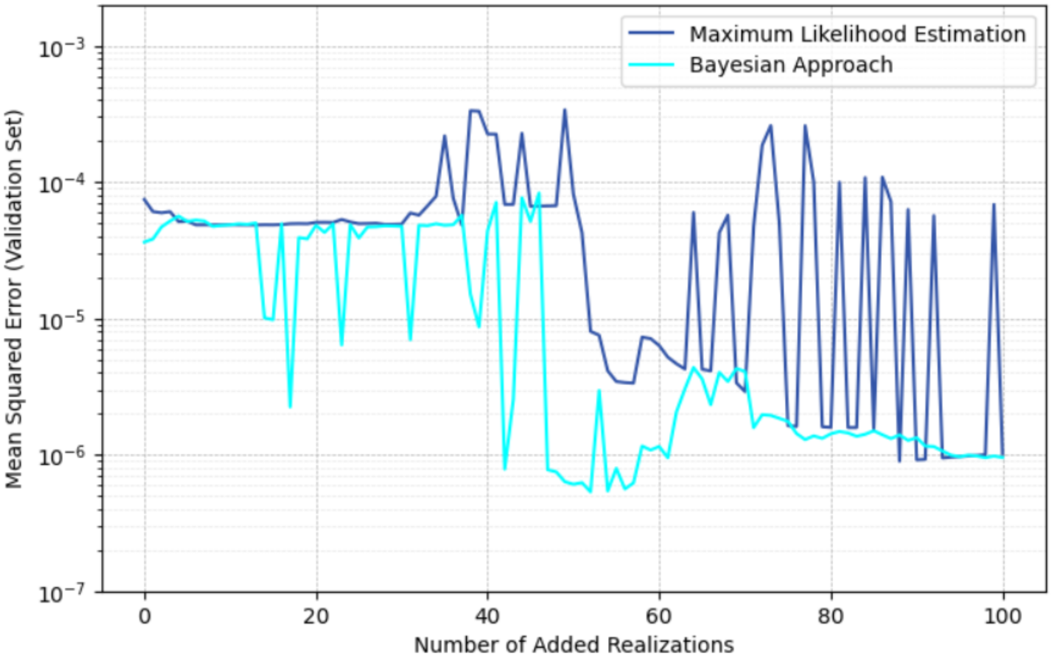
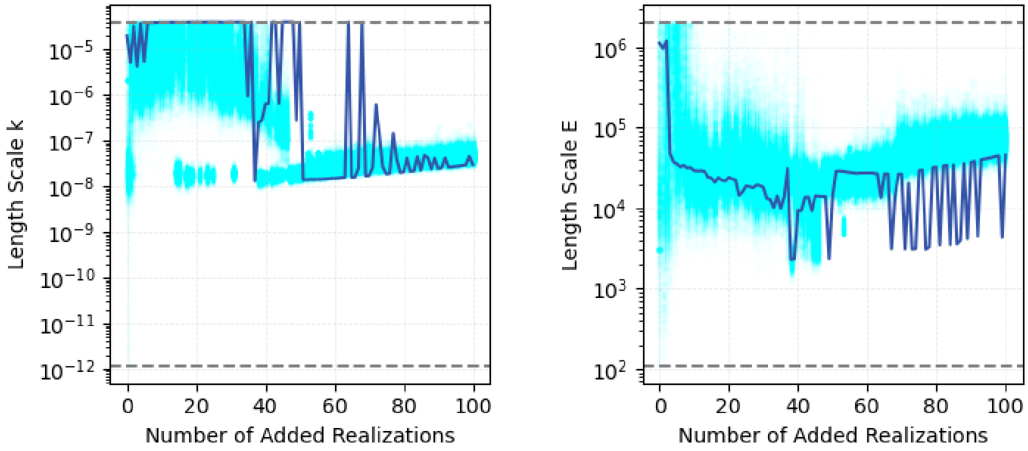


Figure 5.8: Mean Squared Error for Consolidation Problem



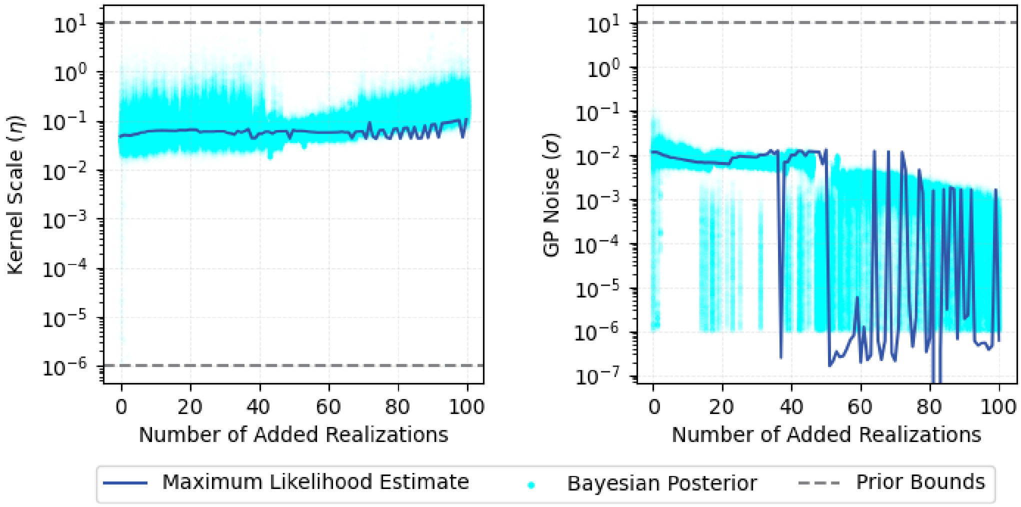


Figure 5.9: Optimized Hyperparameters for Consolidation Problem

Early iterations show that both methods struggled to find reasonable values for the length scale of the hydraulic conductivity, with optimized values and samples taken at the upper limits of the prior distributions. It's interesting to note however that at around 20 iterations, the Bayesian method was able to find modes at two different locations of the hydraulic conductivity length scale posterior distribution, signifying its potential to spot multiple modes. This can also be directly correlated with the jumps to lower *MSE* values around this time. On the other hand, wide posteriors of stiffness were obtained until 48 added realizations, after which they narrowed down. It was at this moment when the *MSE* value of the Bayesian method drops to its lowest values. At this point, the MLE finds hyperparameter estimates within the same range of the Bayesian method, yet the *MSE* values attained by the methods differ by approximately a factor of 10. It is also noticeable how the changes and jumps in hyperparameter values are more sudden and often for the MLE approach. The Bayesian approach on the other hand has a smoother and more gradual shift of its hyperparameter posteriors. It should be noted that both methods did not use prior information from optimized hayperparameter values of previous iterations.

Figures 5.10 and 5.11 show the joint log-likelihoods of the length scales of k and E after 21 added realizations, which is when the Bayesian approach already seems to find samples that give better likelihoods while the MLE is still stuck in the bounds. It can be seen that the MLE fails to find the true optimum of the log-likelihood, getting stuck in a mode that has a wider spread than the true peak. The Bayesian approach on the other hand was able to take samples from both regions of high likelihood densities, with the spread of samples relatively proportional with the width of the peaks. This demonstrates the NUTS sampler's effectiveness in approximating more complex posterior distributions, particularly in capturing samples from the relevant peaks. The localization of samples in the modes can again be attributed to the higher acceptance rate of the tuning stage of the sampler.

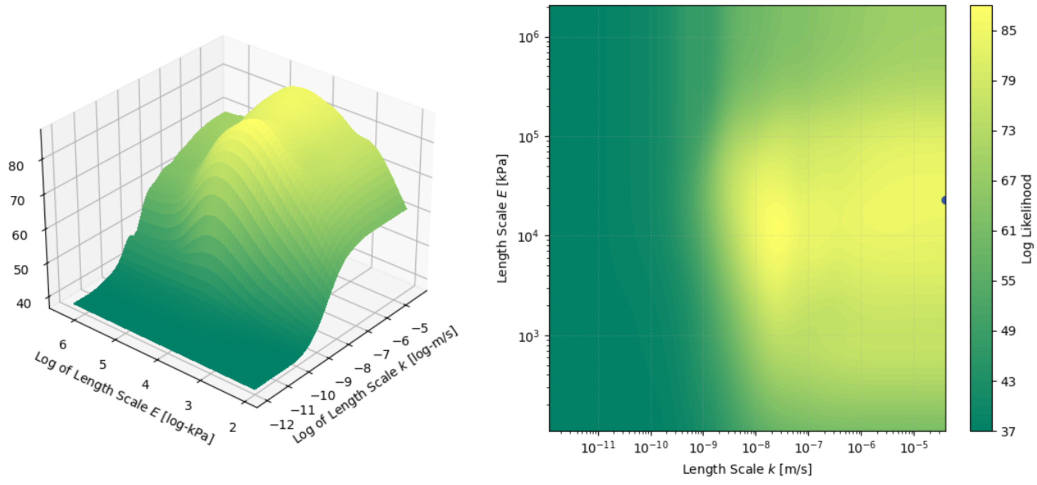


Figure 5.10: Log-likelihood for MLE Iteration 21

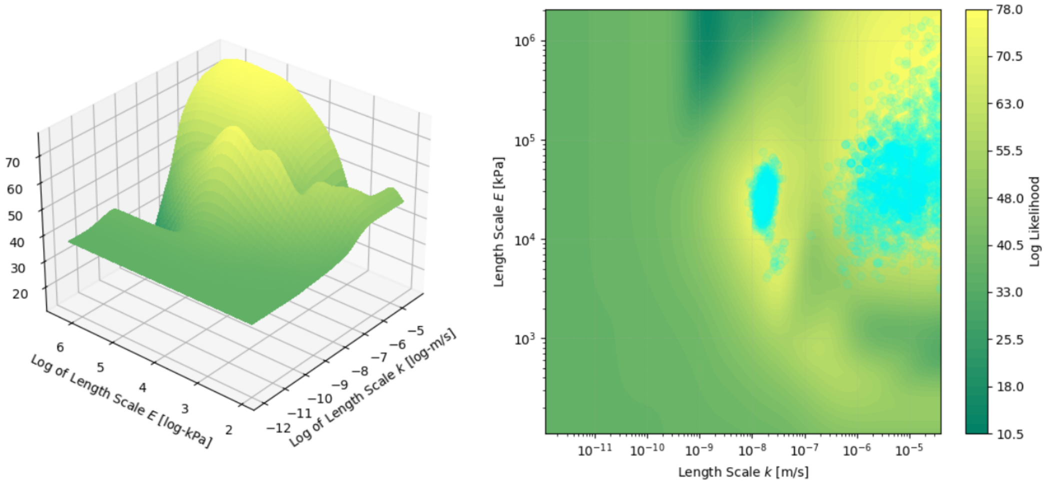


Figure 5.11: Log-likelihood for Bayesian Iteration 21

Jumps in maximum likelihood estimates of hyperparameters are also observed from round 70 iterations onwards. Figures 5.12 and 5.13 show the joint log-likelihood of the length scales of k and E at iteration 70. Although the MLE method properly locates the peak for this iteration, there are noticeable jumps in the optimized noise values, leading to different peak locations. The Bayesian method on the other hand considers different possible peak locations and obtains a wide range of samples of cohesion length scale and noise.

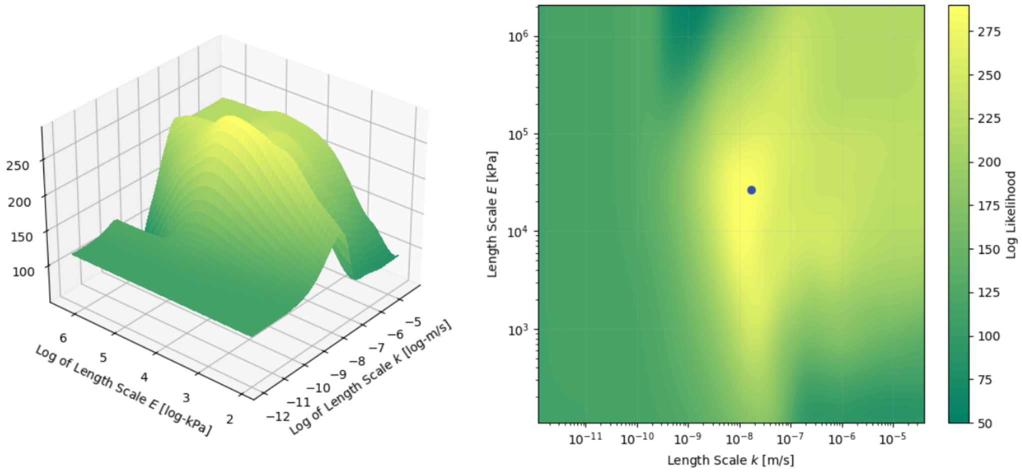


Figure 5.12: Log-likelihood for MLE Iteration 70

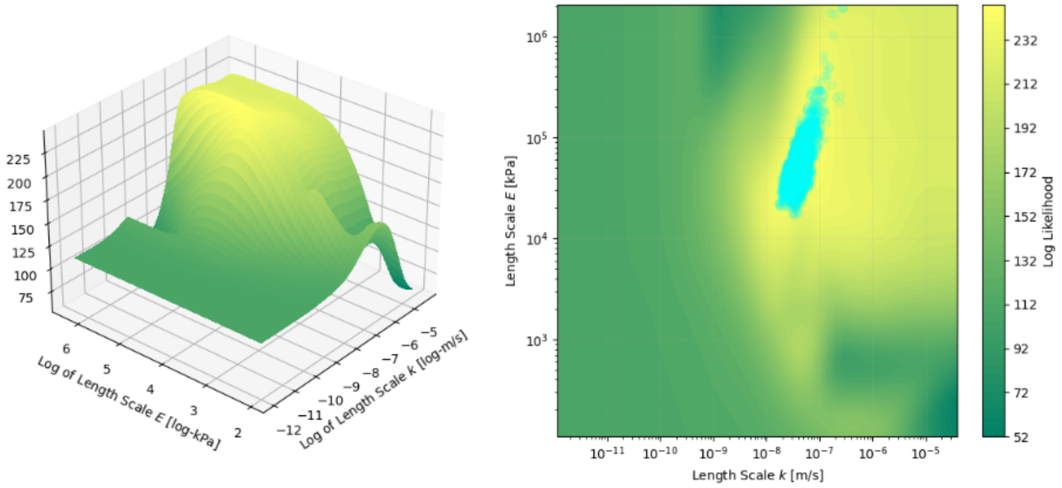


Figure 5.13: Log-likelihood for Bayesian Iteration 70

Figure 5.14 shows how learning points were selected during the enrichment of both models. In initial runs, the MLE method selected points with high correlations. Both methods prioritized taking learning points from the right tail of the hydraulic conductivity length scale distribution, which is expected as points there are spaced wider apart. After 50 and 100 added realizations, the spread of the design of experiments becomes similar for both methods. Therefore, the difference in the *MSE* values is most likely explained by the difference in the sampled hyperparameter values

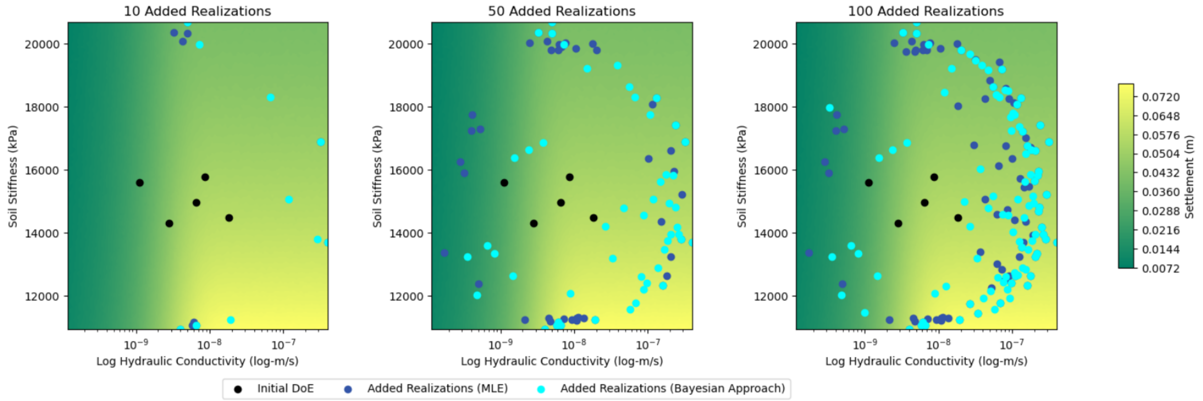


Figure 5.14: Enrichment for Consolidation Problem

The predictions made from the best models of each method are shown in Figure 5.15. Visually, there is no noticeable difference in predictive performance. However, the Bayesian method needed 36 fewer added realizations and still achieved lower *MSE* value. It should be noted however that there are still oscillations of predicted settlements at higher values of hydraulic conductivity, which can be attributed to relatively narrower posteriors of hydraulic conductivity length scales.

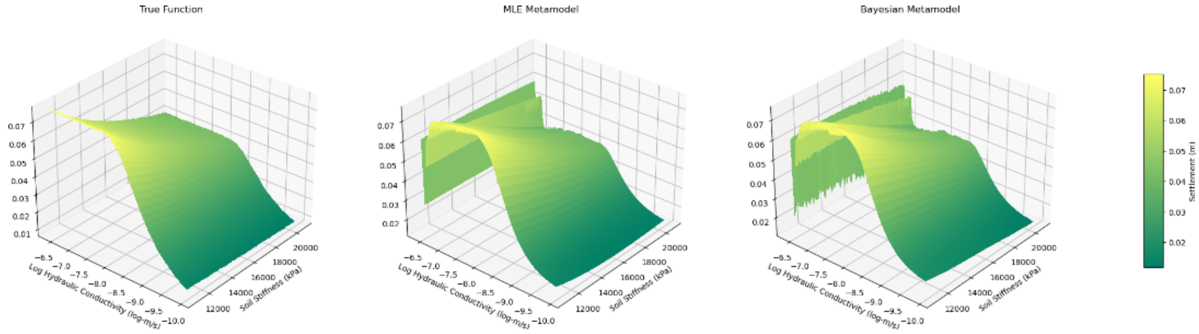


Figure 5.15: Metamodel Predictions for Consolidation Problem

Figure 5.16 shows the predictive intervals for both methods in the consolidation problem after 52 added realizations, when the Bayesian method attained its lowest *MSE*. In this case, the effect of the uncertainty in the hyperparameter space is significant. There are larger uncertainties in the higher values of hydraulic conductivity for the MLE approach, likely due to a sub-optimal fit of the obtained length scale in this region. Aside from this, the hyperparameter uncertainty of the Bayesian approach greatly increased the predictive intervals compared to the MLE approach. A large skew in the both hyperparameter and predictive uncertainty can also be seen.

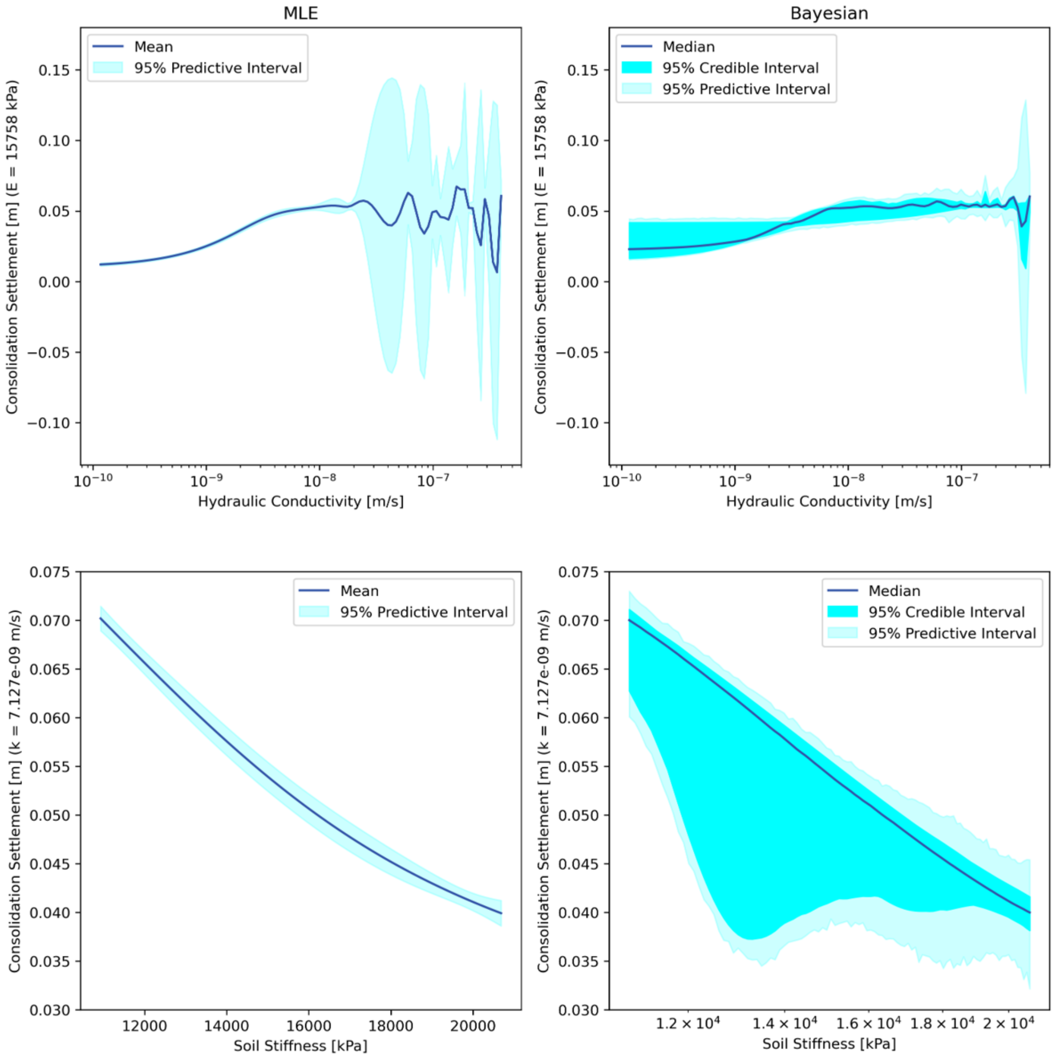


Figure 5.16: Predictive Intervals for Consolidation Problem

5.3. Case 3: Piled Embankment

Figure 5.17 shows the evolution of the probability of failure and its convergence criterion value as the number of added realizations increased for both MLE and Bayesian approaches. One hundred realizations were added for both methods; however, the required convergence criterion value of 0.05 was reached at iteration 50 for the MLE approach and 46 for the Bayesian approach. Higher initial values of the convergence criterion are observed for the Bayesian approach, which can be attributed to the increase of the uncertainty of the Bayesian predictions caused by the additional uncertainty of hyperparameter samples. This lead to wider intervals of the probability of failure for the Bayesian approach. The uncertainty caused solely by the hyperparameter distribution diminished at around 30 to 40 iterations. Afterwards, a sharp reduction in overall predictive uncertainty at around 50 iterations for both methods, though a larger change is seen in the Bayesian method.

The stabilization of the $\hat{P}_f$ and $COV_{\hat{P}_f}$ values happens after around 35 to 40 added realization for both methods, less than the number of iterations needed to reach the convergence criteria. The probability of failure converges at $4.94 * 10^{-4}$, with a corresponding coefficient of variation of 4.5% following Equation 3.17.

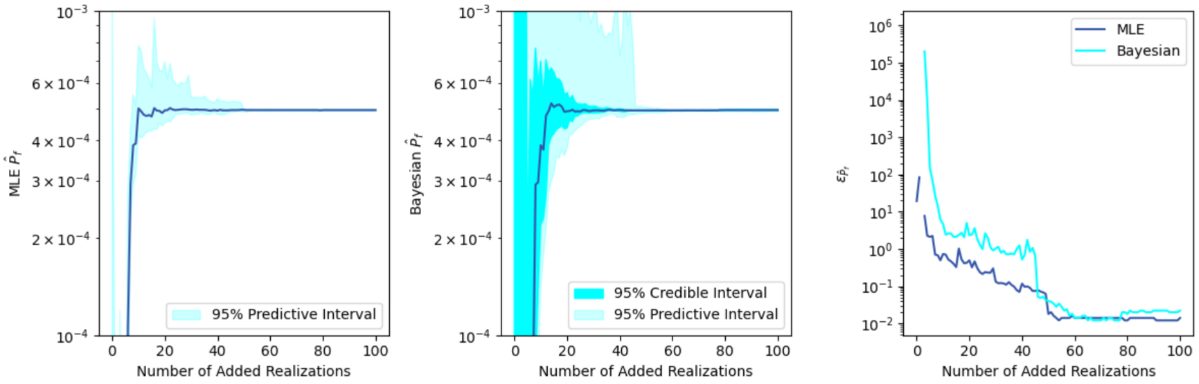


Figure 5.17: $\hat{P}_f$, $COV_{\hat{P}_f}$, and $\epsilon_{\hat{P}_f}$ for Piled Embankment

Figure 5.18 shows the optimized and sampled hyperparameter values for each iteration of the enrichment process. There seems to be a general agreement between the obtained values for both methods, with the MLE values lying within the distributions obtained by the Bayesian approach. Sudden changes in optimized values for the length scales of friction angle and stiffness of sand can be observed in the MLE method. There is also a difference in the Gaussian process noise obtained by the two methods at later iterations, where the maximum likelihood estimate obtains values at the lower bound, while the Bayesian method took samples at around 10^{-3} m. These noise are low compared to the performance function values and only translate to a small increase in the convergence criterion values for the Bayesian approach.

It is also observed that some of the hyperparameters obatained by both methods are situated at the upper bounds of the prior distribution. For Gaussian processes metamodels, this usually signifies one of two things: either the input variable does not affect the output and the relationship between the two is explained by noise (which may be the case for the hydraulic conductivity of sand), or the input variable and the output have a linear relationship (which may be the case for stiffness). Furthermore, the posterior distributions have a wider uncertainty for these hyperparameters. However, varying draws from these distributions likely resulted in negligible differences in function values, as the magnitudes of these length scales are all high compared to the scale of the corresponding input variable.

The observed drop in the uncertainty at around 50 iterations does not seem to be caused by hyperparameter selection, as the hyperparameter plots for both MLE and Bayersian methods show no significant difference in the sampled hyperparameters at this time.

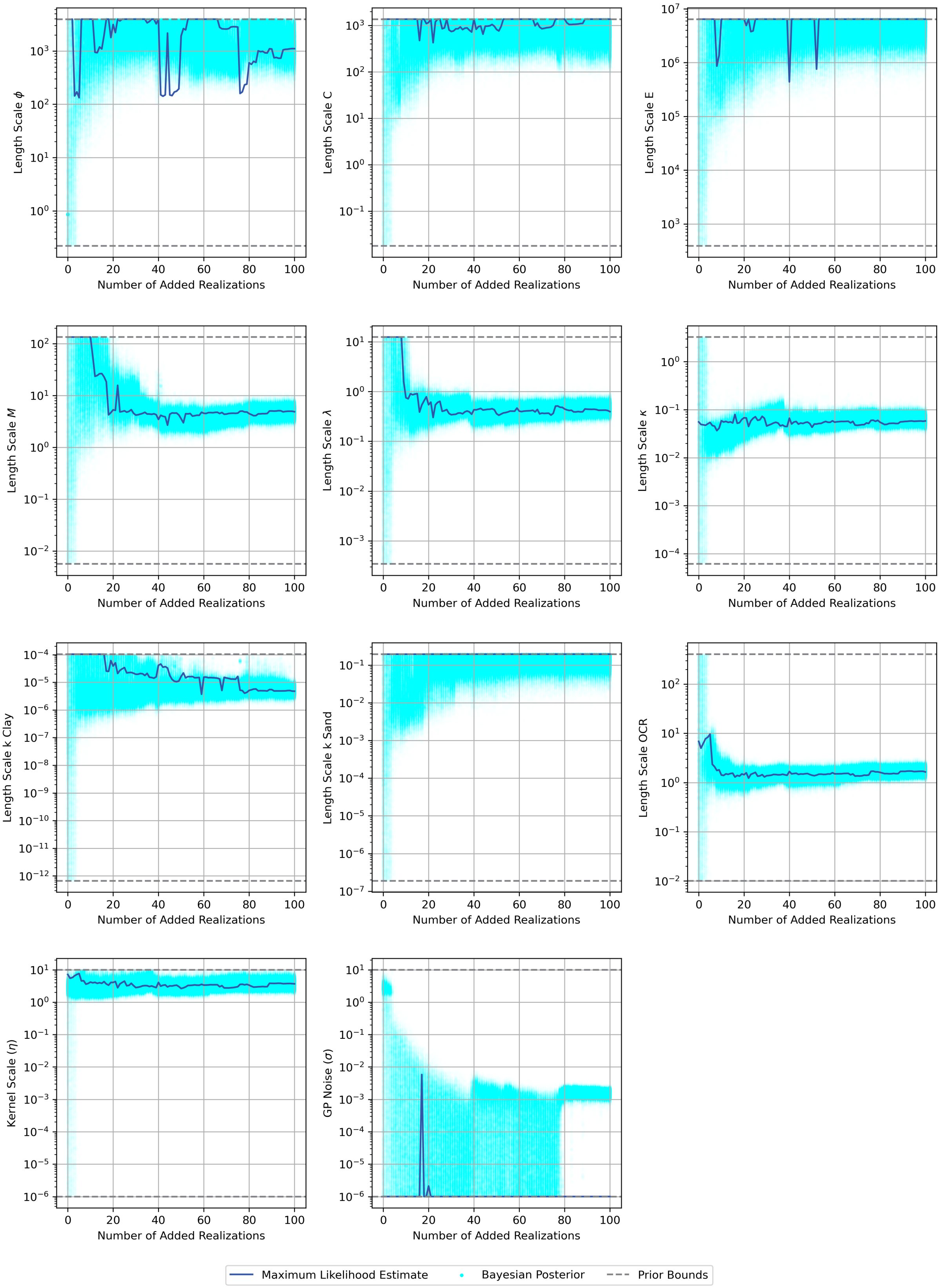


Figure 5.18: Optimized Hyperparameters for Piled Embankment

Figure 5.19 shows the predictive values of the performance function for a small sample of the population after 100 added realizations. Predictive samples were directly obtained from the Bayesian metamodel, while the Gaussian uncertainty of the MLE predictive samples was generated from its predictive mean and variance. In most cases, the true values are visually closer to the Bayesian predictive samples than the MLE predictive uncertainty. A comparison between *MSE* values after 100 iterations was also performed, where a small selection of 100 random points was selected as the validation set. Note that the *U* learning function was used during enrichment, leading to points closer to the failure criteria being added to the DoE in both methods, while the validation set was randomly sampled from the entire population. The MLE method achieved an *MSE* of 0.0309 while the Bayesian method achieved an *MSE* of 0.0141. While this comparison was done on a relatively small validation set size, it again shows the potential of the Bayesian method to give predictive estimates closer to the true function values.

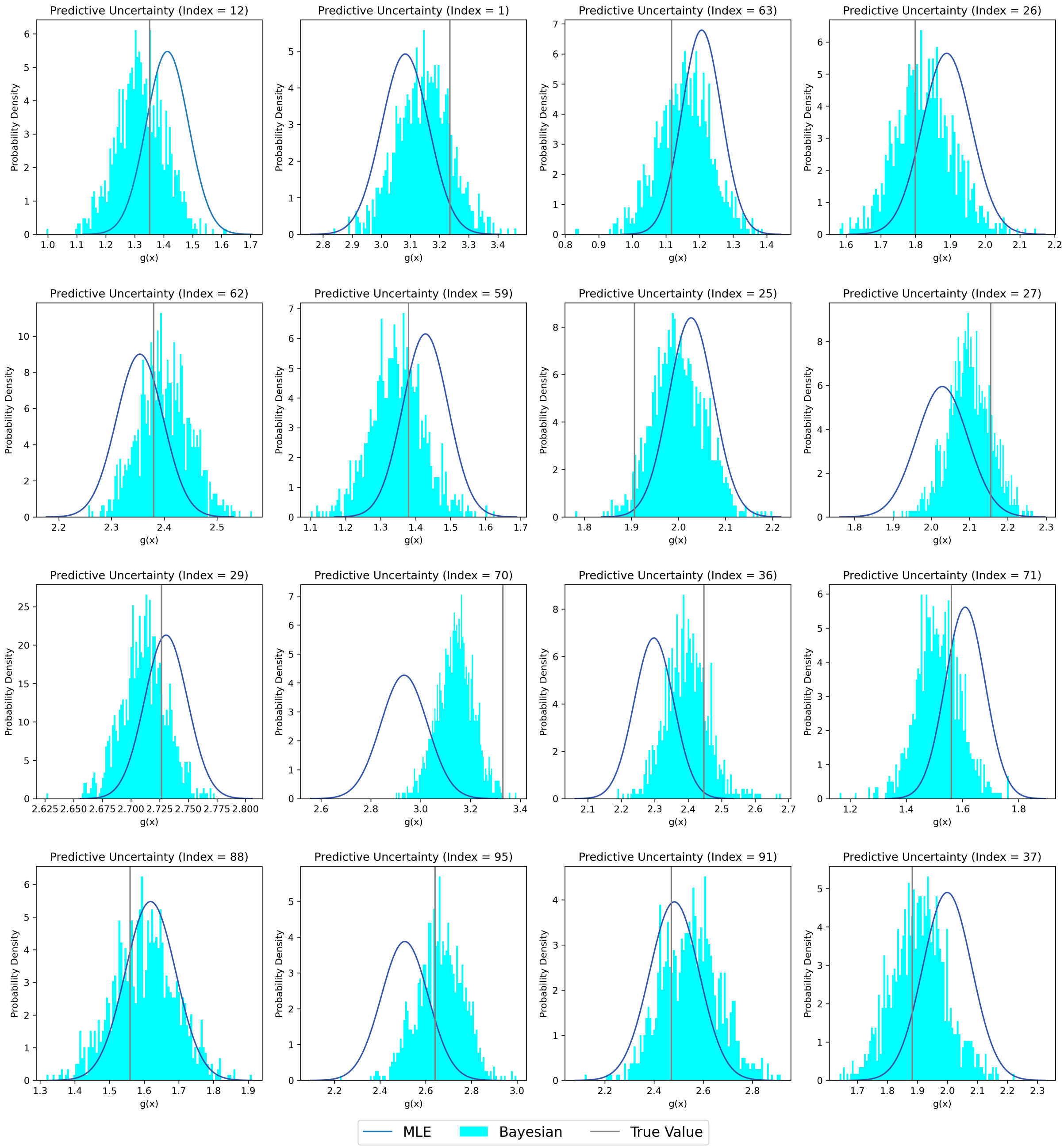


Figure 5.19: Piled Embankment Predictions

5.4. Comparative Summary and Additional Remarks

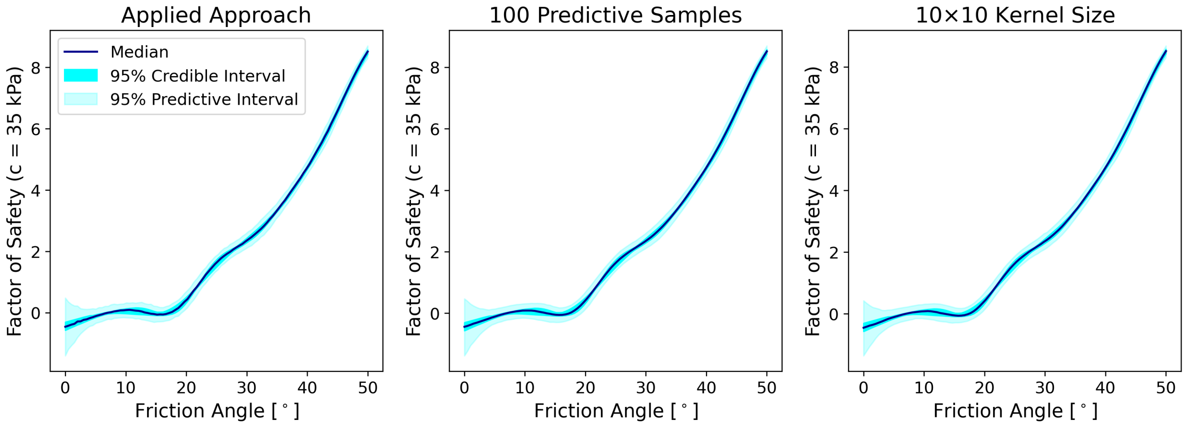
In general, the NUTS approach in obtaining the distribution of hyperparameter samples was either in accordance with the maximum likelihood estimate or fit better the likelihood of the design of experiments. It showed better capability than the MLE's gradient-descent based search algorithm in finding the true optima of the likelihood space. It must be noted that the MLE method's robustness in finding the true optimum can be improved by utilizing multiple restarts of the L-BFGS search algorithm at different locations of the hyperparameter space. Similary, the Bayesian method may also further be improved by providing different starting points of the sampling chains. The effect and comparison of these improvements were not investigated.

The full stochastic analysis of hyperparameters led to an increased understanding of the uncertainty in metamodel predictions, where the extent of the increase is problem dependent. In terms of MSE scores, the Bayesian method provides better generalization at earlier iterations, reaching lower MSE values sooner than the MLE approach. After addition of multiple points, while it is not definitive that the Bayesian method gives better predictive accuracy, it gave better MSE scores for Cases 2 and 3 and performed only slightly poorer in Case 1. For all cases, it gave a more comprehensive quantification of the uncertainty, but the increase in the predictive uncertainty is problem dependent. For Case 1, only a slight increase in predictive uncertainty was observed. In contrast, Case 2 experienced a significant increase, while Case 3 initially saw a substantial rise in uncertainty that quickly diminished after just one additional learning point.

The Bayesian method involved calculation of 1000 metamodels per iteration, and this significantly increased the required computational costs and time. In addition, the predictive mean and uncertainties are not automatically calculated and percentiles had to be obtained from the predictive samples. The sorting process needed for each of the prediction point also increased the computation time. As a comparison, one iteration ,including hyperparameter optimization, metamodel building, prediction, and enrichment, of the MLE method takes a few seconds to complete. On the other hand, this takes at least 8 minutes for the Bayesian method, and took as long as around 30 minutes for the latter iterations of the more complex piled embankment problem, where a bulk of the computation time was spent on generating predictive samples and quantifying the uncertainties based on the percentiles. All analysis were performed on a Lenovo Thinkpad E14 Gen2, a consumer-grade laptop.

During sampling, there were occasional warnings reported by PyMC regarding divergences, Gelman-Rubin value, and the effective sample size. As stated in the methodology, measures were taken to reduce the occurrence of these warnings, including increasing the tuning steps and the acceptance rate during the tuning stage. This did not however completely resolve the issue, and while some iterations had divergences or larger $\hat{R}$ values, further measures were no longer taken. The acceptance ratio was already high at 0.95, and increasing the number of samples would also further increase the computation time. These warnings typically occur in more complex posteriors, where different step sizes might be suitable for different areas of the posterior space. In addition, chains might take samples from different modes of the space, leading to different sampling chains and larger $\hat{R}$ values. A full table of number of divergences and $\hat{R}$ values are presented in Appendix. Higher $\hat{R}$ values occur at instances when the sampler takes samples from different mode locations, leading to bad mixing. However, spotting of different modes was a goal for this research, and this was accepted. Ideally, all four chains will spot all the different mode locations, which leads to lower $\hat{R}$ values. This requires more generated samples and was not done for this research.

As mentioned in the methodology, some simplifications to the process of making predictions were performed, which resulted in uncertainty quantification through uncorrelated predictive sampling. Recall that the method applied in this thesis involves ten predictive samples per hyperparameter sample and uncorrelated prediction points, such that only the diagonal of the predictive covariance matrix was used. Theoretically, an infinite number of samples and a full predictive posterior covariance matrix would lead to the correct values of the predictive median and intervals. While this is not possible to perform, the smoothness of the predictions and predictive uncertainty can be improved by providing more predictive samples per hyperparameter sample and lessening the sparsity of the predictive covariance matrix. Figure 5.20 shows how the results of the method used in this research compare to less simplified approaches. The first involves providing 100 predictive samples per hyperparameter instead of ten. This led to smaller jumps in the values of the median and bounds of the predictive samples. The second involves a less sparse predictive covariance matrix, where prediction points were manually clustered into 100 points and fully correlated Gaussian predictions were made for each of these clusters using 10 by 10 covariance matrices. This led to fewer jumps, located between smooth segments. These jump locations are potentially the boundaries between clusters. The less simplified methods can only be performed with a smaller number of prediction points due to memory allocation and computation time constraints; thus, the predictive performance was tested only for one iteration and the methods were not fully integrated with the active learning approach. While the approach applied in this study resulted in less smooth predictive medians and uncertainties, the magnitudes are similar to those of the less simplified approaches. This may have led to slight misselection of learning points. However, the locations of these learning points are expected to be near the correct learning points.



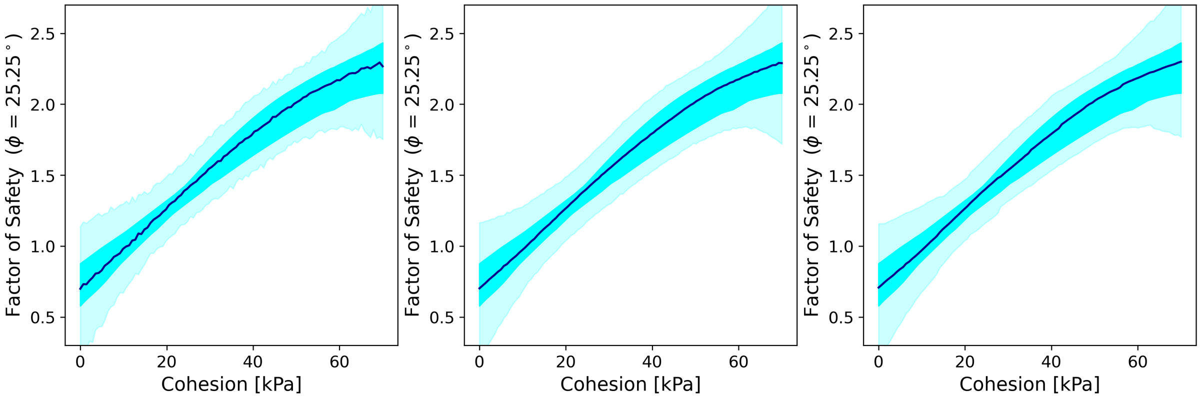


Figure 5.20: Effect of Predictive Simplifications

6

Conclusion

This research aimed to explore the application of a Bayesian approach to surrogate modelling in geotechnical engineering problems. First, it explored how a Bayesian method can be applied to hyperparameter determination, whose posterior distributions could not be directly obtained due to their non-Gaussian nature and their non-linear relationship with the function values. A Markov Chain sampling approach using No-U-Turn sampling was shown to be reliable both in low-dimensional and high-dimensional geotechnical problems. To improve sampling diagnostics, larger tuning steps and acceptance ratios were used, causing more localized sampling of hyperparameters. Measures and simplifications also had to be taken to reduce the needed computational resources and time to a level that is manageable and practical, including large thinning of samples and uncorrelated predictions. Increasing the number of samples or the correlation between the prediction points provided smoother medians and predictive intervals but did not significantly affect the magnitudes of these values. Log-uniform priors appeared to be effective in finding samples at more reasonable values for the length scales, requiring fewer added realizations compared to the MLE method in the early iterations. The suitability of other prior distributions for these hyperparameters remains unexplored; providing informative priors for surrogate model hyperparameters may be challenging, as the length scale can vary depending on the model. For example, the optimized length scales for cohesion, friction angle, hydraulic conductivity and stiffness obtained in Cases 1 and 2 differ greatly from the same length scales obtained in Case 3.

This Bayesian sampling approach was able to obtain hyperparameter samples in accordance with log-likelihood plots as well as maximum likelihood estimates of the hyperparameters. It showed better ability than the MLE in finding true optima of the likelihood space. It also showed good ability in spotting multiple modes for more complex joint distributions of hyperparameters, especially in early iterations where the learning points are still few, finding more reasonable values of hyperparameters earlier than the MLE method.

When combined with active learning, the benefits of already having the predictive means and variance in Kriging with one hyperparameter value disappear, and the predictive uncertainty has to be defined from generated predictive samples. Using percentiles instead of the standard mean and variance is ideal, given the non-Gaussian nature of the hyperparameter distribution and its non-linear relationship with predictive function values. This potentially causes greatly skewed predictive samples, as portrayed in the predictive intervals of Cases 2 and 3, and the use of percentiles provides better robustness over this skew. This increases the computational requirements, especially when dealing with a large set of prediction points.

The use of Bayesian inference in hyperparameter determination allowed for a more comprehensive quantification of the model uncertainty, as evidenced by the increased predictive intervals of both functions and probabilities of failure. This increase in the understanding of the uncertainty is vital in decision making in geotechnical engineering, where the stakes involve high monetary and safety consequences. Extensive quantification of uncertainty helps engineers make more informed choices, balancing risks and benefits effectively. The Bayesian approach also showed potential in giving more accurate function values as indicated by the lower MSE's in Cases 2 and 3. This however is limited mostly in the prediction of the function values. It didn't show any improvement in the predictive accuracy in terms of the MSE's of the probability of failure in a reliability analysis problem, and needed more added realization to reach an acceptable convergence criterion. Thus, it is more useful in cases where the actual function values as well as its uncertainty is desired. For example, it can be applied to geotechnical design optimization problems or extensive sensitivity analysis, where the effect of the joint uncertainty of different input geotechnical parameters on the uncertainty of the output is desired.

While not much difference was shown in the P_f values of MLE and Bayesian methods, the uncertainty quantification is more comprehensive for the Bayesian method. Thus, if there is greater interest in uncertainty analysis in geotechnical reliability problems, then the application of the Bayesian method can be advantageous over the MLE method. This provides stricter requirements for the size of the DoE needed to conclude a sufficiently accurate P_f value.

Further research need to be done with regards to the improvement of the efficiency of the Bayesian method. Due to the long computation times for each iteration, the Bayesian method could greatly benefit from multi-point enrichment, which is suggested as a direction for future research. In this scenario, additional samples of hyperparameters can be added to ensure improved mixing of the chains and enhanced sampling diagnostics.

7

Appendices

Table 7.1: Sampling Statistics for Retaining Wall Problem

Iteration	Divergences	$\hat{R}$			
		l_ϕ	l_c	η	σ
1	1	1.01	1.00	1.01	1.01
2	0	1.00	1.00	1.01	1.01
3	3	1.00	1.00	1.00	1.00
4	0	1.00	1.00	1.01	1.00
5	0	1.01	1.00	1.00	1.01
6	1	1.01	1.01	1.01	1.01
7	0	1.00	1.00	1.00	1.01
8	16	1.01	1.01	1.00	1.01
9	0	1.02	1.01	1.01	1.03
10	0	1.00	1.00	1.00	1.01
11	18	1.00	1.02	1.01	1.01
12	0	1.01	1.01	1.01	1.01
13	1	1.00	1.01	1.00	1.01
14	0	1.01	1.00	1.00	1.01
15	0	1.01	1.01	1.00	1.01
16	0	1.00	1.01	1.01	1.00
17	0	1.01	1.01	1.01	1.01
18	31	1.00	1.01	1.01	1.01
19	14	1.01	1.02	1.02	1.01
20	10	1.01	1.01	1.00	1.02
21	39	1.02	1.01	1.02	1.01
22	10	1.02	1.02	1.01	1.02
23	8	1.02	1.02	1.01	1.01

Continued on next page

Iteration	Divergences	$\hat{R}$			
		l_ϕ	l_c	η	σ
24	8	1.06	1.07	1.02	1.11
25	12	1.00	1.00	1.00	1.00
26	10	1.01	1.01	1.00	1.01
27	4	1.60	1.56	1.54	1.19
28	0	1.01	1.01	1.01	1.00
29	0	1.00	1.00	1.00	1.00
30	0	1.00	1.00	1.00	1.00
31	0	1.00	1.01	1.00	1.01
32	0	1.00	1.00	1.00	1.00
33	0	1.00	1.00	1.01	1.00
34	1	1.01	1.01	1.01	1.01
35	0	1.01	1.00	1.01	1.00
36	0	1.00	1.00	1.00	1.01
37	0	1.00	1.01	1.00	1.00
38	0	1.00	1.00	1.00	1.00
39	0	1.00	1.00	1.00	1.00
40	0	1.00	1.00	1.00	1.00
41	0	1.00	1.00	1.00	1.00
42	0	1.00	1.00	1.00	1.00
43	0	1.00	1.00	1.01	1.00
44	0	1.01	1.01	1.01	1.00
45	0	1.00	1.00	1.00	1.00
46	0	1.00	1.00	1.00	1.00
47	0	1.00	1.00	1.00	1.00
48	0	1.00	1.00	1.00	1.00
49	0	1.00	1.00	1.00	1.00
50	1	1.00	1.00	1.00	1.00
51	0	1.01	1.00	1.01	1.00
52	0	1.00	1.00	1.00	1.00
53	2	1.00	1.00	1.00	1.00
54	1	1.59	1.58	1.60	1.55
55	4	1.59	1.59	1.60	1.59
56	0	1.00	1.00	1.00	1.00
57	0	1.00	1.00	1.00	1.00
58	0	1.00	1.00	1.00	1.00
59	0	1.00	1.00	1.00	1.00
60	0	1.01	1.01	1.01	1.00
61	0	1.00	1.01	1.01	1.00
62	0	1.00	1.00	1.00	1.00
63	0	1.00	1.00	1.00	1.00
64	0	1.01	1.00	1.01	1.01

Continued on next page

Iteration	Divergences	$\hat{R}$			
		l_ϕ	l_c	η	σ
65	0	1.00	1.00	1.01	1.00
66	0	1.01	1.01	1.01	1.00
67	0	1.01	1.01	1.01	1.01
68	0	1.00	1.01	1.00	1.01
69	0	1.00	1.00	1.00	1.00
70	169	1.01	1.01	1.01	1.01
71	0	1.00	1.00	1.00	1.00
72	0	1.01	1.01	1.00	1.00
73	0	1.00	1.00	1.00	1.00
74	0	1.00	1.00	1.00	1.00
75	0	1.00	1.00	1.01	1.00
76	0	1.00	1.00	1.00	1.00
77	0	1.00	1.00	1.00	1.00
78	0	1.00	1.00	1.00	1.00
79	0	1.00	1.00	1.00	1.00
80	0	1.00	1.00	1.00	1.00
81	0	1.00	1.00	1.00	1.00
82	0	1.00	1.00	1.00	1.00
83	0	1.00	1.00	1.00	1.00
84	0	1.00	1.00	1.00	1.00
85	0	1.00	1.00	1.00	1.00
86	0	1.00	1.00	1.00	1.00
87	0	1.01	1.00	1.01	1.00
88	0	1.00	1.00	1.00	1.00
89	0	1.00	1.00	1.00	1.00
90	0	1.58	1.59	1.60	1.59
91	0	1.00	1.00	1.00	1.00
92	0	1.00	1.00	1.00	1.00
93	0	1.00	1.00	1.00	1.00
94	0	1.00	1.00	1.01	1.00
95	0	1.00	1.00	1.00	1.00
96	0	1.00	1.00	1.00	1.00
97	0	1.00	1.00	1.00	1.00
98	4	1.53	1.56	1.59	1.60
99	0	1.00	1.00	1.00	1.00
100	0	1.00	1.00	1.00	1.00

Table 7.2: Sampling Statistics for Consolidation Problem

Iteration	Divergences	$\hat{R}$			
		l_k	l_E	η	σ
1	0	1.01	1.01	1.01	1.01
2	0	1.03	1.01	1.00	1.07
3	4	1.07	1.02	1.01	1.09
4	0	1.00	1.00	1.00	1.00
5	0	1.01	1.00	1.00	1.00
6	0	1.00	1.00	1.00	1.00
7	0	1.00	1.00	1.00	1.00
8	0	1.01	1.00	1.00	1.00
9	0	1.00	1.00	1.00	1.00
10	0	1.00	1.00	1.00	1.00
11	0	1.00	1.00	1.00	1.00
12	0	1.00	1.00	1.00	1.00
13	0	1.00	1.00	1.00	1.00
14	0	1.00	1.00	1.00	1.00
15	0	1.74	1.08	1.10	1.74
16	0	1.74	1.10	1.14	1.73
17	0	1.00	1.00	1.01	1.00
18	0	1.53	1.11	1.11	1.53
19	0	1.53	1.10	1.11	1.53
20	0	1.54	1.09	1.11	1.53
21	0	1.00	1.00	1.00	1.00
22	277	1.55	1.13	1.12	1.53
23	0	1.00	1.00	1.00	1.00
24	0	1.74	1.16	1.18	1.73
25	0	1.01	1.00	1.00	1.00
26	0	1.54	1.10	1.10	1.53
27	0	1.01	1.00	1.00	1.00
28	0	1.00	1.00	1.00	1.00
29	0	1.00	1.00	1.00	1.00
30	0	1.00	1.00	1.00	1.00
31	0	1.00	1.01	1.01	1.00
32	0	1.74	1.12	1.18	1.73
33	0	1.00	1.00	1.00	1.00
34	0	1.01	1.00	1.00	1.00
35	0	1.01	1.01	1.01	1.00
36	0	1.01	1.01	1.00	1.00
37	0	1.01	1.01	1.01	1.00
38	0	1.01	1.00	1.01	1.00
39	0	1.74	1.48	1.13	1.74

Continued on next page

Iteration	Divergences	$\hat{R}$			
		l_k	l_E	η	σ
40	0	1.74	1.15	1.17	1.74
41	0	1.54	1.06	1.13	1.53
42	0	1.00	1.00	1.00	1.00
43	2	1.47	1.18	1.09	1.53
44	295	1.44	1.40	1.40	1.46
45	0	1.53	1.20	1.05	1.53
46	54	1.46	1.19	1.05	1.48
47	1	1.02	1.01	1.00	1.02
48	0	1.00	1.00	1.00	1.01
49	0	1.00	1.00	1.00	1.01
50	0	1.00	1.00	1.00	1.01
51	0	1.00	1.01	1.00	1.00
52	0	1.01	1.01	1.01	1.01
53	0	1.05	1.03	1.01	1.05
54	156	1.84	1.59	1.26	1.70
55	0	1.02	1.01	1.02	1.02
56	0	1.14	1.13	1.09	1.10
57	0	1.02	1.01	1.00	1.02
58	0	1.01	1.01	1.01	1.01
59	0	1.01	1.00	1.00	1.00
60	0	1.01	1.00	1.00	1.01
61	0	1.01	1.01	1.01	1.01
62	0	1.01	1.01	1.00	1.01
63	0	1.01	1.00	1.00	1.01
64	0	1.00	1.00	1.00	1.02
65	0	1.01	1.01	1.01	1.02
66	0	1.01	1.00	1.01	1.01
67	0	1.01	1.01	1.00	1.01
68	0	1.01	1.00	1.01	1.00
69	0	1.01	1.00	1.01	1.00
70	0	1.01	1.01	1.01	1.01
71	0	1.01	1.01	1.01	1.01
72	0	1.01	1.00	1.00	1.02
73	0	1.00	1.00	1.00	1.00
74	0	1.00	1.00	1.01	1.01
75	0	1.00	1.00	1.00	1.02
76	0	1.00	1.01	1.01	1.01
77	0	1.01	1.00	1.01	1.01
78	0	1.01	1.00	1.01	1.00
79	0	1.01	1.00	1.01	1.00
80	0	1.00	1.00	1.00	1.01

Continued on next page

Iteration	Divergences	$\hat{R}$			
		l_k	l_E	η	σ
81	0	1.00	1.00	1.00	1.00
82	0	1.01	1.01	1.01	1.00
83	0	1.00	1.00	1.00	1.01
84	0	1.00	1.00	1.00	1.02
85	0	1.01	1.00	1.00	1.01
86	0	1.00	1.00	1.00	1.01
87	0	1.01	1.01	1.01	1.02
88	0	1.00	1.00	1.00	1.01
89	0	1.01	1.01	1.01	1.01
90	0	1.01	1.01	1.01	1.01
91	0	1.00	1.00	1.01	1.00
92	0	1.01	1.01	1.01	1.01
93	0	1.01	1.01	1.01	1.01
94	0	1.00	1.00	1.00	1.00
95	0	1.01	1.01	1.01	1.01
96	0	1.00	1.00	1.00	1.00
97	0	1.00	1.00	1.00	1.00
98	0	1.00	1.00	1.00	1.01
99	0	1.01	1.01	1.01	1.00
100	0	1.00	1.00	1.00	1.00

Table 7.3: Sampling Statistics for Piled Embankment Problem

Iteration	Divergences	$\hat{R}$										
		l_ϕ	l_c	l_E	l_M	l_λ	l_κ	$l_{k_{clay}}$	$l_{k_{sand}}$	l_{OCR}	η	σ
1	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00
2	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01
3	0	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00	1.01	1.01	1.01
4	0	1.00	1.00	1.01	1.00	1.00	1.01	1.00	1.00	1.00	1.00	1.01
5	0	1.01	1.00	1.00	1.00	1.01	1.02	1.01	1.01	1.03	1.01	1.01
6	1	1.00	1.00	1.00	1.00	1.01	1.00	1.00	1.00	1.00	1.00	1.00
7	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
8	0	1.00	1.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
9	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00	1.00	1.00
10	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
11	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
12	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
13	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
14	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
15	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
16	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
17	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
18	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
19	0	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00	1.00	1.00	1.00
20	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
21	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00
22	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
23	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
24	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Continued on next page

Iteration	Divergences	$\hat{R}$										
		l_ϕ	l_c	l_E	l_M	l_λ	l_κ	$l_{k_{clay}}$	$l_{k_{sand}}$	l_{OCR}	η	σ
25	0	1.00	1.00	1.00	1.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00
26	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
27	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
28	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
29	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
30	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00
31	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
32	0	1.00	1.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
33	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
34	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
35	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
36	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
37	14	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
38	7	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
39	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
40	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
41	0	1.00	1.00	1.00	1.01	1.00	1.00	1.01	1.00	1.00	1.00	1.00
42	0	1.00	1.01	1.00	1.02	1.00	1.00	1.01	1.00	1.00	1.00	1.01
43	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
44	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01
45	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
46	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
47	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
48	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
49	0	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00	1.00	1.01	1.00
50	0	1.00	1.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Continued on next page

Iteration	Divergences	$\hat{R}$										
		l_ϕ	l_c	l_E	l_M	l_λ	l_κ	$l_{k_{clay}}$	$l_{k_{sand}}$	l_{OCR}	η	σ
51	0	1.00	1.00	1.00	1.01	1.00	1.00	1.00	1.00	1.00	1.01	1.00
52	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00
53	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
54	0	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00	1.00	1.00	1.01
55	0	1.00	1.00	1.00	1.00	1.00	1.01	1.00	1.00	1.01	1.01	1.00
56	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01
57	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01
58	0	1.01	1.00	1.00	1.01	1.00	1.01	1.01	1.00	1.01	1.01	1.00
59	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01
60	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
61	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
62	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
63	0	1.00	1.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01
64	0	1.00	1.00	1.00	1.00	1.00	1.01	1.00	1.00	1.00	1.01	1.00
65	0	1.01	1.00	1.00	1.00	1.01	1.00	1.00	1.00	1.00	1.01	1.01
66	0	1.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01
67	0	1.00	1.01	1.01	1.00	1.01	1.00	1.00	1.01	1.00	1.00	1.00
68	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00
69	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00
70	0	1.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
71	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00
72	0	1.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
73	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
74	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.01	1.00
75	0	1.01	1.00	1.00	1.00	1.01	1.00	1.00	1.00	1.01	1.01	1.00
76	0	1.01	1.00	1.00	1.00	1.01	1.01	1.00	1.00	1.01	1.01	1.01

Continued on next page

Iteration	Divergences	$\hat{R}$										
		l_ϕ	l_c	l_E	l_M	l_λ	l_κ	$l_{k_{clay}}$	$l_{k_{sand}}$	l_{OCR}	η	σ
77	0	1.02	1.00	1.01	1.01	1.00	1.00	1.03	1.00	1.01	1.01	1.03
78	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
79	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01
80	0	1.01	1.01	1.00	1.00	1.00	1.01	1.00	1.00	1.00	1.00	1.09
81	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
82	0	1.00	1.00	1.00	1.00	1.00	1.01	1.00	1.00	1.00	1.01	1.00
83	0	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00	1.00	1.00	1.00
84	0	1.00	1.02	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01
85	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
86	0	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00	1.00	1.00	1.00
87	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00
88	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
89	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01
90	0	1.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00
91	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
92	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
93	0	1.00	1.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
94	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00	1.00
95	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00
96	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
97	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
99	4	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
100	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00

Listing 7.1: Python Code for MLE GP Modelling with PyMC

```
with pm.Model() as GP_model:
    # Hyperparameters for the Gaussian Process
    ls_phi = pm.Uniform("ls_phi", lower=np.min(phi_pop)/100, upper=np.max(phi_pop)
    *100)
    ls_c = pm.Uniform("ls_c", lower=np.min(c_pop)/100, upper=np.max(c_pop)*100)
    cov_scale = pm.Uniform("cov_scale", lower=0.000001, upper=10)
    sigma = pm.Uniform("sigma", lower=0.000001, upper=10)

    # Mean functiom
    mean_func = pm.gp.mean.Zero()

    # Covariance function
    cov_func = cov_scale ** 2 * pm.gp.cov.Matern52(input_dim=2, ls=[ls_phi, ls_c])

    # GP prior with zero mean
    gp = pm.gp.Marginal(mean_func = mean_func, cov_func = cov_func)

    # GP likelihood
    y_ = gp.marginal_likelihood("y_", X_DoE, Y_DoE, sigma)

    start = {'ls_phi': np.mean(phi_pop), 'ls_c': np.mean(c_pop), 'cov_scale': 1, '
    sigma': 1}
    map_estimate = pm.find_MAP()

    mean_pop, var_pop = gp.predict(X_pop, map_estimate, diag=True)
    mean_val, var_val = gp.predict(X_val, map_estimate, diag=True)
```

Listing 7.2: Python Code for Bayesian GP Modelling with PyMC

```
n_chain = 4
n_trace = 1000
thinning_factor = 40
n_thin = int(n_trace/thinning_factor)
n_pred_samples = 10
N_pred_samples = int(n_chain*n_thin*n_pred_samples)

y_hat_pop = np.zeros((N_pred_samples, len(X_pop)))
y_hat_val = np.zeros((N_pred_samples, len(X_val)))

with pm.Model() as GP_model:
    # Hyperparameters for the Gaussian Process
    log_ls_phi = pm.Uniform("log_ls_phi", lower=np.log(np.min(phi_pop)/100), upper=np.
    log(np.max(phi_pop)*100))
    log_ls_c = pm.Uniform("log_ls_c", lower=np.log(np.min(c_pop)/100), upper=np.log(np
    .max(c_pop)*100))
    log_cov_scale = pm.Uniform("log_cov_scale", lower=np.log(0.000001), upper=np.log
    (10))
    log_sigma = pm.Uniform("log_sigma", lower=np.log(0.000001), upper=np.log(10))

    ls_phi = pm.Deterministic('ls_phi', pm.math.exp(log_ls_phi))
    ls_c = pm.Deterministic('ls_c', pm.math.exp(log_ls_c))
    cov_scale = pm.Deterministic('cov_scale', pm.math.exp(log_cov_scale))
    sigma = pm.Deterministic('sigma', pm.math.exp(log_sigma))

    # Mean function
    mean_func = pm.gp.mean.Zero()
    # Covariance function
    cov_func = cov_scale ** 2 * pm.gp.cov.Matern52(input_dim=2, ls=[ls_phi, ls_c])

    # GP prior with zero mean
    gp = pm.gp.Marginal(mean_func = mean_func, cov_func = cov_func)

    # GP likelihood
    y_ = gp.marginal_likelihood("y_", X_DoE, Y_DoE, sigma)

    trace = pm.sample(n_trace, return_inferencedata=True, chains = 4, cores = 1, tune
    =1000, target_accept=0.95)
    idata = trace.sel(draw=slice(None, None, thinning_factor))

    count = 0

    for i in range(len(idata.posterior.chain)):
        for j in range(len(idata.posterior.draw)):

            hyperparam = {
                "log_ls_phi": idata.posterior['log_ls_phi'][i][j],
                "log_ls_c": idata.posterior['log_ls_c'][i][j],
                "log_cov_scale": idata.posterior['log_cov_scale'][i][j],
                "log_sigma": idata.posterior['log_sigma'][i][j],

                "ls_phi": idata.posterior['ls_phi'][i][j],
                "ls_c": idata.posterior['ls_c'][i][j],
                "cov_scale": idata.posterior['cov_scale'][i][j],
                "sigma": idata.posterior['sigma'][i][j]
            }

            mean_pop, var_pop = gp.predict(X_pop, point = hyperparam, diag=True)
            y_hat_pop[count:count+n_pred_samples] = np.random.normal(mean_pop, np.sqrt
    (var_pop), (n_pred_samples, len(X_pop)))

            mean_val, var_val = gp.predict(X_val, point = hyperparam, diag=True)
            y_hat_val[count:count+n_pred_samples] = np.random.normal(mean_val, np.sqrt
    (var_val), (n_pred_samples, len(X_val)))

            count += n_pred_samples
```

Listing 7.3: Python Code for Learning Functions

```
def learning_function_MLE (mean_hat, var_hat, learning_function_type):
    '''
    Solves and returns learning function values.

    Parameters
    ----------
    mean_hat : predictive mean of metamodel in each point
        element type: array ; float

    var_hat : predictive variance of metamodel in each point
        element type: array ; float

    learning_function_type : type of learning function employed (Options: U, var)
        element type: string
    Returns
    -------
    LFV : learning function values at population points
        element type: array ; float
    '''

    N = len(mean_hat)
    LFV = np.zeros(N)

    sigma_hat = np.sqrt(var_hat)

    if learning_function_type == 'U':
        for i in range(N):
            U = np.abs(mean_hat[i])/sigma_hat[i]
            LFV[i] = -U

    elif learning_function_type == 'var':
        LFV = sigma_hat

    return LFV

def learning_function_bayesian(p2_5, p50, p97_5, learning_function_type):
    '''
    Solves and returns learning function values, Bayesian approach.

    Parameters
    ----------
    p2_5 : 2.5th percentile of predictions in each point
        element type: array ; float

    p50 : 50th percentile of predictions in each point
        element type: array ; float

    p97_5 : 97.5th percentile of predictions in each point
        element type: array ; float

    learning_function_type : type of learning function employed (Options: U, var)
        element type: string

    Returns
    -------
    LFV : learning function values at population points
        element type: array ; float
    '''
    N = len(p50)

    LFV = np.zeros(N)

    if learning_function_type == 'U':
        for i in range(N):
            # phi = min(np.sum(preds[:,i] > 0), np.sum(preds[:,i] <= 0))/H
            U = np.abs(p50[i])/max(np.abs(p50[i] - p2_5[i]), np.abs(p97_5[i] - p50[i])
    )
            LFV[i] = -U

    elif learning_function_type == 'var':
        for i in range(N):
            LFV[i] = max(np.abs(p50[i] - p2_5[i]), np.abs(p97_5[i] - p50[i]))

    return LFV
```

Listing 7.4: Python Code for Calculation of Probability of Failure and Bounds

```
def calculate_Pf_MLE(mean_hat, var_hat):
    '''
    Calculates probability of failure (including bounds) and convergence (for
    reliability problems).

    Parameters
    ----------
    mean_hat : mean of metamodel predictions
        element type: array ; float

    var_hat : variance of metamodel predictions
        element type: array ; float

    Returns
    -------
    Pf : predicted probability of failure
        element type: float
    Pf_plus : upper bound of predicted probability of failure
        element type: float
    Pf_minus : lower bound of predicted probability of failure
        element type: float
    conv_criterion : convergence criteriion value of predicted probability of failure
        element type: float
    '''

    N = len(mean_hat)
    sigma_hat = np.sqrt(var_hat)

    Pf_plus = np.sum(mean_hat-1.96*sigma_hat <= 0)/N
    Pf = np.sum(mean_hat <= 0)/N
    Pf_minus = np.sum(mean_hat+1.96*sigma_hat <= 0)/N

    conv_criterion = (Pf_plus - Pf_minus)/Pf

    return Pf, Pf_plus, Pf_minus, conv_criterion

def calculate_Pf_bayesian(p2_5, p50, p97_5):
    '''
    Calculates probability of failure (including bounds) and convergence (for
    reliability problems), Bayesian approach.

    Parameters
    ----------
    p2_5 : 2.5th percentile of predictions in each location
        element type: array ; float

    p50 : 50th percentile of predictions in each location
        element type: array ; float

    p97_5 : 97.5th percentile of predictions in each location
```

```
        element type: array ; float

    Returns
    -------
    Pf : predicted probability of failure
        element type: float
    Pf_plus : upper bound of predicted probability of failure
        element type: float
    Pf_minus : lower bound of predicted probability of failure
        element type: float
    conv_criterion : convergence criteriion value of predicted probability of failure
        element type: float
    '''

    N = len(p50)

    Pf_plus = np.sum(p2_5 <= 0)/N
    Pf = np.sum(p50 <= 0)/N
    Pf_minus = np.sum(p97_5 <= 0)/N

    conv_criterion = (Pf_plus - Pf_minus)/Pf

    return Pf, Pf_plus, Pf_minus, conv_criterion
```

References

Al-Bittar, T., Soubra, A.-H., & Thajeel, J. (2018). Kriging-Based Reliability Analysis of Strip Footings Resting on Spatially Varying Soils. *Journal of Geotechnical and Geoenvironmental Engineering*, *144*(10), 04018071. https://doi.org/10.1061/(ASCE)GT.1943-5606.0001958

Betancourt, M. (2018, July). A Conceptual Introduction to Hamiltonian Monte Carlo [arXiv:1701.02434 [stat]]. Retrieved July 2, 2024, from http://arxiv.org/abs/1701.02434

Das, B. M. (2007). *Principles of foundation engineering* (6th ed.). Thomson.

Dubourg, V. (2011). *Méta-modèles adaptatifs pour l'analyse de fiabilité et l'optimisation sous contrainte fiabiliste* [Doctoral dissertation, Université Blaise Pascal – Clermont II].

Echard, B., Gayton, N., & Lemaire, M. (2011). AK-MCS: An active learning reliability method combining Kriging and Monte Carlo Simulation. *Structural Safety*, *33*(2), 145–154. https://doi.org/10.1016/j.strusafe.2011.01.002

Echard, B., Gayton, N., Lemaire, M., & Relun, N. (2013). A combined Importance Sampling and Kriging reliability method for small failure probabilities with time-demanding numerical models. *Reliability Engineering & System Safety*, *111*, 232–240. https://doi.org/10.1016/j.ress.2012.10.008

El Haj, A.-K., Soubra, A.-H., & Al-Bittar, T. (2019). Probabilistic analysis of strip footings based on enhanced Kriging metamodeling. *International Journal for Numerical and Analytical Methods in Geomechanics*, *43*(17), 2667–2686. https://doi.org/10.1002/nag.2995

Gelman, A., & Rubin, D. B. (1992). Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, *7*(4). https://doi.org/10.1214/ss/1177011136

Hoffman, M., & Gelman, A. (2014). The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo (A. Kottas, Ed.) [Submitted 11/11; Revised 10/13; Published 4/14]. *Journal of Machine Learning Research*, *15*, 1593–1623. http://jmlr.org/papers/v15/hoffman14a.html

Kang, F., Han, S., Salgado, R., & Li, J. (2015). System probabilistic stability analysis of soil slopes using Gaussian process regression with Latin hypercube sampling. *Computers and Geotechnics*, *63*, 13–25. https://doi.org/10.1016/j.compgeo.2014.08.010

Kulhawy, F., & Mayne, P. (1990, August). *Manual on estimating soil properties for foundation design* [Electric Power Research Institute (EPRI) Report, Document EL-6800]. Ithaca, New York.

Lataniotis, C., Wicaksono, D., Marelli, S., & Sudret, B. (2022). *Uqlab user manual gaussian process modeling*. ETH Zurich. Zurich.

Moustapha, M., Marelli, S., & Sudret., B. (2022). *Uqlab user manual active learning reliability*. ETH Zurich. Zurich.

Murray, I., & Adams, R. P. (2010). Slice sampling covariance hyperparameters of latent Gaussian models [arXiv:1006.0868 [stat]]. Retrieved July 2, 2024, from http://arxiv.org/abs/1006.0868

Neal, R. M. (1997, January). *Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification* (tech. rep.) (arXiv:physics/9701026). arXiv. Retrieved July 2, 2024, from http://arxiv.org/abs/physics/9701026

Neal, R. M. (2000). Slice Sampling [arXiv:physics/0009028]. Retrieved July 2, 2024, from http://arxiv.org/abs/physics/0009028

Neal, R. M. (2011, May). MCMC using Hamiltonian dynamics [arXiv:1206.1901 [physics, stat]]. Retrieved July 2, 2024, from http://arxiv.org/abs/1206.1901

Nen 9997-1: Geotechnical design of structures – part 1: General rules (2016th ed.). (2016, June). *National Standards of the Netherlands.*

Plaxis 2d 2024.1 material models manual [Last Updated: December 29, 2023]. (2023). PLAXIS.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning* [www.GaussianProcess.org/gpml]. The MIT Press.

Schöbi, R., Sudret, B., & Marelli, S. (2017). Rare Event Estimation Using Polynomial-Chaos Kriging. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, *3*(2), D4016002. https://doi.org/10.1061/AJRUA6.0000870

Snoek, J., Larochelle, H., & Adams, R. P. (2012, August). Practical Bayesian Optimization of Machine Learning Algorithms [arXiv:1206.2944 [cs, stat]]. Retrieved July 2, 2024, from http://arxiv.org/abs/1206.2944

Soubra, A.-H., Al-Bittar, T., Thajeel, J., & Ahmed, A. (2019). Probabilistic analysis of strip footings resting on spatially varying soils using kriging metamodeling and importance sampling. *Computers and Geotechnics*, *114*, 103107. https://doi.org/10.1016/j.compgeo.2019.103107

Sudret, B. (2014). Polynomial chaos expansions and stochastic finite element methods. In K.-K. Phoon & J. Ching (Eds.), *Risk and reliability in geotechnical engineering* (pp. 265–300). CRC Press.

Teixeira, R., Nogal, M., & O'Connor, A. (2021). Adaptive approaches in metamodel-based reliability analysis: A review. *Structural Safety*, *89*, 102019. https://doi.org/10.1016/j.strusafe.2020.102019

Van den Eijnden, A. P., Schweckendiek, T., & Hicks, M. A. (2022). Metamodelling for geotechnical reliability analysis with noisy and incomplete models. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, *16*(3), 518–535. https://doi.org/10.1080/17499518.2021.1952611

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., & Bürkner, P.-C. (2021). Rank-normalization, folding, and localization: An improved $\hat{R}$ for assessing convergence of MCMC [arXiv:1903.08008 [stat]]. *Bayesian Analysis, 16*(2). https://doi.org/10.1214/20-BA1221