# UNCERTAINTY AND SENSITIVITY ANALYSIS FOR LONG-RUNNING COMPUTER CODES: A CRITICAL REVIEW

By

Dustin R. Langewisch

B.S. Mechanical Engineering and B.S. Mathematics
University of Missouri-Columbia, 2007

SUBMITTED TO THE DEPARTMENT OF NUCLEAR SCIENCE AND ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN NUCLEAR SCIENCE AND ENGINEERING
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

FEBRUARY 2010

Signature of Author: ______________________
Department of Nuclear Science and Engineering
January 28, 2010

Certified by: ______________________
Dr. George E. Apostolakis, Thesis Supervisor
KEPCO Professor of Nuclear Science and Engineering
Professor of Engineering Systems

Certified by: ______________________
Donald Helton, Thesis Reader
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission

Accepted by: ______________________
Dr. Jacquelyn C. Yanch
Chair, Department Committee on Graduate Students

# UNCERTAINTY AND SENSITIVITY ANALYSIS FOR LONG-RUNNING COMPUTER CODES: A CRITICAL REVIEW

Dustin R. Langewisch

SUBMITTED TO THE DEPARTMENT OF NUCLEAR SCIENCE AND ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN NUCLEAR SCIENCE AND ENGINEERING
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

FEBRUARY 2010

## Abstract

This thesis presents a critical review of existing methods for performing probabilistic uncertainty and sensitivity analysis for complex, computationally expensive simulation models. Uncertainty analysis (UA) methods reviewed include standard Monte Carlo simulation, Latin Hypercube sampling, importance sampling, line sampling, and subset simulation. Sensitivity analysis (SA) methods include scatter plots, Monte Carlo filtering, regression analysis, variance-based methods (Sobol' sensitivity indices and Sobol' Monte Carlo algorithms), and Fourier amplitude sensitivity tests. In addition, this thesis reviews several existing metamodeling techniques that are intended provide quick-running approximations to the computer models being studied. Because stochastic simulation-based UA and SA rely on a large number (e.g., several thousands) of simulations, metamodels are recognized as a necessary compromise when UA and SA must be performed with long-running (i.e., several hours or days per simulation) computational models. This thesis discusses the use of polynomial Response Surfaces (RS), Artificial Neural Networks (ANN), and Kriging/Gaussian Processes (GP) for metamodeling. Moreover, two methods are discussed for estimating the uncertainty introduced by the metamodel. The first of these methods is based on a bootstrap sampling procedure, and can be utilized for any metamodeling technique. The second method is specific to GP models, and is based on a Bayesian interpretation of the underlying stochastic process. Finally, to demonstrate the use of these methods, the results from two case studies involving the reliability assessment of passive nuclear safety systems are presented.

The general conclusions of this work are that polynomial RSs are frequently incapable of adequately representing the complex input/output behavior exhibited by many mechanistic models. In addition, the goodness-of-fit of the RS should not be misinterpreted as a measure of the predictive capability of the metamodel, since RSs are necessarily biased predictors for deterministic computer models. Furthermore, the extent of this bias is not measured by standard goodness-of-fit metrics (e.g., coefficient of determination, $R^2$), so these methods tend to provide overly optimistic indications of the quality of the metamodel. The bootstrap procedure does provide indication as to the extent of this bias, with the bootstrap confidence intervals for the RS estimates generally being significantly wider than those of the alternative metamodeling methods. It has been found that the added flexibility afforded by ANNs and GPs can make these methods superior for approximating complex models. In addition, GPs are exact interpolators, which is an important feature when the underlying computer model is deterministic (i.e., when there is no justification for including a random error component in the metamodel). On the other hand, when the number of observations from the computer model is sufficiently large, all three methods appear to perform comparably, indicating that in such cases, RSs can still provide useful approximations.

| | |
|---|---|
| **Thesis Supervisor:** | George E. Apostolakis<br>KEPCO Professor of Nuclear Science and Engineering<br>Professor of Engineering Systems<br>MIT Department of Nuclear Science and Engineering |
| **Thesis Reader:** | Donald Helton<br>Office of Nuclear Regulatory Research<br>US Nuclear Regulatory Commission |

# Acknowledgments

First of all, I would like to thank my advisor, Professor George Apostolakis, for his guidance and encouragement throughout the duration of this project, not to mention his countless suggested revisions to this thesis. I am grateful to all of the NRC staff who have supported this work. Specifically, I would like to thank Don Helton, who graciously agreed to read this thesis, as well as Nathan Siu, both of whom provided numerous helpful suggestions and comments.

Furthermore, I am deeply indebted to Nicola Pedroni, who has provided numerous contributions to this thesis. The results from the artificial neural network studies were obtained from Nicola. In addition, I was first introduced to several of the advanced Monte Carlo methods through Nicola's research. I am also grateful to Qiao Liu, who provided much-needed assistance on the topic of sensitivity analysis.

I would like to thank Anna Nikiforova for helping me work out numerous issues with the FCRR RELAP model, and C.J. Fong, whose Master's thesis inspired a substantial portion of this research.

Finally, I would like to thank all of the friends and family members who have supported me over the past couple of years. While there are too many to list, I am especially grateful for the support and encouragement that I received from my parents. In addition, I would like to thank my officemate, Tim Lucas, for providing an incessant source of distraction. Without him, this thesis would, undoubtedly, be far more comprehensive, though I probably would not have maintained my sanity.

## Table of Contents

# List of Figures

## List of Tables

## List of Acronyms

| | |
|---|---|
| ACOSSO | Adaptive COmponent Selection and Shrinkage Operator |
| ANN | Artificial Neural Network |
| BBC | Bootstrap Bias-Corrected |
| CART | Classification and Regression Trees |
| CDF | Cumulative Density Function |
| CFD | Computational Fluid Dynamics |
| CI | Confidence Interval |
| DACE | Design and Analysis of Computer Experiments |
| DHR | Decay Heat Removal |
| FAST | Fourier Amplitude Sensitivity Test |
| FCRR | Flexible Conversion Ratio Reactor |
| FORM | First-Order Reliability Method |
| GAM | Generalized Additive Model |
| GBM | Gradient Boosting Machine |
| GFR | Gas-cooled Fast Reactor |
| GP | Gaussian Process |
| I/O | Input/Output |
| IHX | Intermediate Heat Exchanger |
| iid | Independent and Identically Distributed |
| IS | Importance Sampling |
| LHS | Latin Hypercube Sampling |
| LOCA | Loss-of-Coolant Accident |
| LRS | Linear Response Surface |
| LS | Line Sampling |
| MARS | Multivariate Adaptive Regression Splines |
| MCF | Monte Carlo Filtering |
| MCMC | Markov Chain Monte Carlo |
| MCS | Monte Carlo Simulation |
| MLE | Maximum Likelihood Estimate |
| MSPE | Mean Square Prediction Error |
| OAT | One-At-a-Time |
| PAHX | PSACS Heat Exchanger |
| PCT | Peak Clad Temperature |
| PDF | Probability Density Function |
| PRA | Probabilistic Risk Assessment |
| PRESS | PRediction Error Sum-of-Squares |
| PSACS | Passive Secondary Auxiliary Cooling System |
| QRS | Quadratic Response Surface |
| RBF | Radial Basis Function |
| REML | REstricted Maximum Likelihood |
| RF | Random Forest |
| RMS | Root-Mean-Square |
| RMSE | Root-Mean-Square Error |
| RMSPE | Root-Mean-Square Prediction Error |
| RMSRE | Root-Mean-Square Residual Error |
| RPART | Recursive PARTitioning |
| RS | Response Surface |

| | |
|---|---|
| RSM | Response Surface Methodology |
| RVACS | Reactor Vessel Auxiliary Cooling System |
| SA | Sensitivity Analysis |
| SBO | Station Black-Out |
| SORM | Second-Order Reliability Method |
| SRC | Standardized Regression Coefficient |
| SS | Subset Simulation |
| SVM | Support Vector Machine |
| T-H | Thermal-Hydraulic |
| UA | Uncertainty Analysis |

# I INTRODUCTION

Computer modeling and simulation has become an integral component of nuclear safety assessment and regulation. Current regulations demand the consideration of reactor performance under a variety of severe upset conditions, and reactor vendors must demonstrate that adequate safety measures have been taken. However, cost and safety concerns prohibit such assessments based upon integral accident experiments in full-scale prototypes. Consequently, decision makers are forced to rely heavily upon data obtained from accident simulations. Specifically, in the context of Probabilistic Risk Assessment (PRA), such simulations are used for various purposes, including (i) estimating the reliability of passive systems in the absence of operational data, (ii) informing Level 1 accident sequence development and event tree structure, (iii) establishing Level 1 PRA success criteria, (iv) developing Level 2 PRA event tree structure and split fraction values, (v) performing Level 3 PRA offsite consequence analysis, and (vi) providing the simulation capacity in dynamic PRA tools.

These simulations are performed with complex computer codes that implement sophisticated physical models to describe a variety of physical phenomena, including two-phase flow, multi-mode heat transfer, clad oxidation chemistry, stress and strain, and reactor kinetics. Yet, regardless of their level of sophistication, these models are only approximate representations of reality and are therefore subject to uncertainty. Moreover, the application of these models to a specific system (e.g., a nuclear power plant) necessarily introduces additional uncertainty and opportunities for errors into the analysis. This is because, even under the assumption that our understanding of the physics of the phenomena is perfect and that our models are absolutely correct, approximations and/or assumptions will be made, either out of necessity or ignorance, with regards to the environment (i.e., the boundary conditions and initial conditions) within which the phenomena are supposed to be occurring; examples include (i) the use of simplified geometric representations for various plant components, (ii) the inability to precisely account for manufacturing variability that causes the actual system to differ from the 'paper' system being analyzed, and (iii) the inability to predict environmental conditions such as ambient temperature, wind speed and direction, and rainfall, that may influence the phenomena in question. Hence, in order for regulators to make sound and defensible decisions regarding matters of public safety, it is important that these inaccuracies be understood and quantified. The science of quantifying these inaccuracies and the collective effect that the various assumptions and approximations has on the predictions obtained from simulations is known generally as uncertainty analysis.

Uncertainty analysis (UA) has been recognized as an integral component of safety and risk assessment for decades [1-6]. Indeed, the word 'risk' is meaningless without the notion of uncertainty. Generally speaking, UA refers to any attempt to quantify the uncertainty in any quantitative statement, such as the output from some simulation model. In particular, UA is usually focused on assessing the uncertainty in the model output that results from uncertainty in the model inputs [6-8]. This is not as restrictive as it seems, since the term 'model' can be taken to mean an ensemble of computational submodels, each of which might represent the same phenomenon, but under differing assumptions; for example, such an ensemble could be a collection of different Computational Fluid Dynamics (CFD) codes that all attempt to simulate the same flow conditions, or each submodel could represent an alternative nodalization scheme for the same physical system. In such cases, the inputs to the model include not only initial conditions and boundary conditions, but also the assumptions regarding which of the submodels is most appropriate.

In any case, there are at least two tasks that must be performed in any uncertainty analysis. First, it is necessary to identify and quantify all of the sources of uncertainty that are relevant to a particular model. This task is usually carried out through an expert elicitation process, as summarized by Helton [9]. Although this first step is, without a doubt, crucial to the remainder of the uncertainty analysis, it will be assumed throughout this work that such preliminary measures have been taken and the expert elicitation process will not be considered further; interested readers should refer to Helton [9] who provides an extensive list of references on the topic. After the input uncertainties have been appropriately quantified, the task remains to quantify the influence of this uncertainty on the model's output. This task is referred to as uncertainty propagation and will be the subject of more detailed discussion in Chapter II of this thesis.

Uncertainty analysis, as it has been defined thus far, is only part of the story, as the equally important subject of sensitivity analysis remains to be discussed. It should be noted, however, that the distinction between sensitivity analysis (SA) and UA is somewhat confusing; this is because no UA is complete without an appropriate SA, thereby suggesting that the distinction is irrelevant. On the other hand, there are various analyses that fall within the umbrella of SA, many of which are outside the scope of an analysis of uncertainty. The confusion likely spawns from the use of the term 'uncertainty analysis,' which, while seeming to denote a complete analysis of uncertainty, is usually defined so as to include only a small piece of the actual analysis. Regardless, we will adhere to the conventional definitions and state that the objective of uncertainty analysis is to quantify the uncertainty in the model output, whereas a sensitivity analysis aims to identify the contributions of each model input to this uncertainty [10,11]. Specifically, Saltelli defines sensitivity analysis as "*the study of how the uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources of uncertainty in the model input*" [11]. Thus, we see that while UA proceeds in the forward direction, mapping uncertain inputs to uncertain outputs and quantifying the uncertainty in this output, SA proceeds in opposite direction, starting with the premise that the output of a model is unknown or uncertain and then attempting to probe the model in an effort to better understand the behavior of the model for different input settings. More concisely, UA attempts to determine *what* effect the inputs and their uncertainties have on the model output, while SA attempts to determine *how* these inputs and uncertainties affect the model's output.

Although SA encompasses a broad range of analyses, one would hardly be wrong in stating that, in all cases, the objective of SA is to quantify the 'importance' of the model inputs, either individually or as collective groups. However, it is the definition of importance that is variable, depending on the question that is asked, and different tools exist for answering these various questions. For example, when an analyst is attempting to develop a model to describe some physical phenomenon, she may be interested in identifying how the measured response of an experiment correlates with various control parameters. Moreover, many of the control parameters may be irrelevant to the phenomena, and she may wish to identify the subset of parameters that correlates most highly with the response so that the remaining parameters can be neglected. This is one example of a sensitivity analysis. As another example, an uncertainty analysis may be performed to assess the reliability of, say, a passive emergency core cooling system. If the analysis indicates that the system does not meet certain safety requirements, the engineers may wish to know what input uncertainties contribute the most to the variability in the system response, and hence the failure probability of the system. In this case, SA can be used to rank the inputs based on their contribution to the output uncertainty, thereby allowing for optimal resource allocation by indicating where research efforts should be focused to most effectively reduce the output uncertainty. On the other hand, if the UA results indicated that the system did meet all safety requirements, regulators may be interested in whether the analysis, itself, is acceptable. The regulators may then perform a SA to determine the sensitivity of the UA results to the various assumptions that were made during the preliminary UA. Any assumptions that are deemed 'important' to the results of the UA can then be more vigorously scrutinized. Each of these examples demonstrates the diverse class of questions that can be answered with SA, while also illuminating the common underlying theme in each – that of quantifying importance. In Chapter III of this thesis, we discuss many of the commonly used metrics for quantifying this importance, and discuss some of the popular techniques for computing these metrics.

Although the methods for UA and SA are backed by a sound theoretical foundation, the implementation of these methods has often proven quite troublesome in practice. This is because many of the most useful metrics in UA and SA must be computed with stochastic simulation methods, such as Monte Carlo Simulation (MCS) and its variants. Although these methods will be discussed in detail in following sections, we simply note here that these methods typically require several thousand evaluations of the computer model being studied. Even recently developed advanced simulation methods require several hundred model evaluations, and, in cases where each evaluation of the model requires several minutes or hours, this clearly poses an enormous computational burden. Consequently, there has been much recent interest in the use of so-called metamodels, which are simplified models that are intended to approximate the output from the computer model of interest [12-15]. The metamodel serves as a surrogate to the actual computational model and has the advantage of requiring almost negligible evaluation time. Thus, the UA and SA can be performed through the metamodel with minimal computational burden. In Chapter IV we discuss, in brief, many of the existing metamodeling techniques, and give appropriate references where additional details can be found. Moreover, we provide a detailed discussion of three such methods: namely, polynomial response surfaces, artificial neural networks, and Gaussian process modeling. As should be expected, when an approximation to the actual computer model is adopted, additional uncertainty is introduced to the analysis. Therefore, we conclude Chapter IV with a discussion regarding how this uncertainty can be accounted for.

To summarize, this thesis is intended to provide a critical review of many of the available methods for performing uncertainty and sensitivity analysis with complex computational models of physical phenomena. Chapter II provides a detailed overview of uncertainty analysis, beginning with a classification of some of the most important sources of uncertainty in modeling physical phenomena, followed by a summary of commonly used measures for quantifying uncertainty. We conclude Chapter II by presenting various methods for computing these measures. In Chapter III we discuss many of the methods for performing sensitivity analysis, focusing on the sensitivity measures that each method delivers and the information provided by each of these measures. Chapter IV comprises the bulk of this thesis, describing various techniques for metamodeling and focusing on the relative advantages and disadvantages of each. Furthermore, Chapter IV concludes with a discussion regarding how to account for the additional uncertainty that is introduced by the use of an approximating model. Finally, in Chapter V, we present the results of two case studies that were selected to provide a comparative basis for many of the tools discussed in Chapters II-IV.

# II UNCERTAINTY ANALYSIS

From the discussion given in the introduction, it is apparent that there exist numerous sources of, and even various types of, uncertainty. Before proceeding to a detailed classification of the sources of uncertainty, it is helpful to recognize that two broad classes of uncertainty are frequently distinguished in practice; these classes are aleatory uncertainty and epistemic uncertainty [9,16]. Aleatory uncertainty refers to the randomness, or stochastic variability, that is inherent to a given phenomenon; examples include the outcome of a coin toss or the roll of a die. On the other hand, epistemic uncertainty, also known as subjective or state-of-knowledge uncertainty, refers to one's degree of belief that a claim is true or false, or that a particular model is appropriate. For instance, a flipped coin will land heads up with some probability, $p$, but we may not know precisely what that probability is; rather, we have only a degree-of-belief that the coin is fair ($p = 0.5$) or otherwise. This is an example of epistemic uncertainty concerning the appropriateness of a model (the value of $p$ to use) that describes a random (aleatory) event.

Aleatory and epistemic uncertainties can also be distinguished by the notion of reducibility. In particular, epistemic uncertainty is reducible in the sense that with additional information one's state of knowledge is altered, so that it is, in principle, possible to reduce this uncertainty by learning more about the phenomenon. By contrast, aleatory uncertainty is considered to be irreducible since it is inherent to the physics of the process. Note, however, that the distinction is not always clear since it may be that our description of the physics is insufficient, and that by using a more elaborate model, we could attribute the seemingly random variability to other observable parameters (i.e., the point of contact between the coin and the thumb, and the speed with which it leaves the hand). Despite these nuances, the distinction between aleatory and epistemic uncertainties is still a practical convenience for communicating the results of an uncertainty analysis. It should be pointed out that fundamentally there is only one kind of uncertainty and a unique concept of probability [16].

Due to the qualitative differences between aleatory and epistemic uncertainties, there has been some disagreement as to how they should be represented, and in particular, whether it is appropriate to use a common framework for their representation. While it is almost unanimously agreed that probability theory is the appropriate framework with which to represent aleatory uncertainty, a variety of alternative frameworks have been proposed for representing epistemic uncertainty, including evidence theory, interval analysis, and possibility theory [9,17]. Regardless, probability theory (specifically, the Bayesian interpretation) remains the most popular method for representing epistemic uncertainty, and O'Hagan and Oakley argue that probability theory is "uniquely appropriate" for this task [18]. In this work, we shall adhere to the probabilistic framework and use probability density functions (PDFs) to represent both aleatory and epistemic uncertainty.

Throughout this thesis we shall denote by, $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$, a set of $m$ model parameters (or, inputs), and we let $p_{\mathbf{X}}(\mathbf{x})$ denote the joint PDF for the set, $\mathbf{x}$. Furthermore, we note that a simulation model is simply an implicitly defined function, $\mathbf{y} = \mathbf{g}(\mathbf{x})$, mapping the inputs, $\mathbf{x}$, to some set of model outputs, $\mathbf{y}$. However, to simplify notation throughout this thesis, we shall only consider the case where a single output is desired; that is, the simulation model can be represented as $y = g(\mathbf{x})$, where $y$ is a scalar quantity. The following ideas can be extended to the more general case of a vector output by considering each component of $\mathbf{y}$ separately. Moreover, for any arbitrary quantity, $Q$, we let $\hat{Q}$ denote an estimate of that quantity. With this notation, we can proceed in classifying some of the important sources of uncertainty that we must consider.

## II.1 Classification of Uncertainty

The classification provided here is by no means complete or even unique, but is intended only to illustrate some of the dominant sources of uncertainty in simulation models. Excepting some changes in terminology, the following classification scheme that we have adopted is consistent with that provided by Kennedy and O'Hagan [19].

*i. Aleatory Parametric Uncertainty:*

Because models are idealizations of some real phenomena, they frequently allow for greater control over the input parameter values than can actually be realized. However, it may not be possible to specify, or control, the values of some of these inputs in the real system. Moreover, the system may interact with its environment in a complicated and dynamical manner so that some of the inputs are effectively random. The distinguishing feature of these parameters is that their values could conceivably change each time the code is run. For instance, if the computational model must repeatedly call a subroutine that contains an aleatory variable, a new value must be randomly selected from its distribution for each call to this subroutine. Consequently, repeated runs of the code under identical input configurations will lead to different outputs, and the output of the code will be a random variable.

*ii. Epistemic Parametric Uncertainty:*

Not all of the inputs to the model will be random in the sense described above. Yet, when attempting to simulate some phenomena, it is necessary to specify these remaining parameters in a manner that is consistent with the actual system being simulated. In practice, however, the appropriate values of many of these parameters may not be precisely known due to a lack of data concerning the phenomena under consideration. As described previously, this type of uncertainty is one instance of epistemic uncertainty. More specifically, we refer to this uncertainty as epistemic parametric uncertainty. Epistemic parametric uncertainty is distinguished from aleatory parametric variability in that, in the case of the former, the input parameter takes some fixed, albeit unknown, value. Thus, the value does not change each time the model is run.

*iii. Model Inadequacy:*

All models are idealized and approximate representations of reality. More specifically, suppose we let $\tilde{y}$ denote the observed response from some experiment. For the sake of illustration, suppose that $\tilde{y} = R(\xi)$, where $\xi$ represents a set, not necessarily finite, of predictor variables (analogous to model inputs), and $R(\,\cdot\,)$ represents the operation of nature on these predictors. The observation $\tilde{y}$ may be either deterministic or inherently stochastic. Either way, in the ideal scenario, we would know $R(\,\cdot\,)$ exactly, so that we could predict without error the outcome (deterministically or statistically) of any experiment. However, in reality, we are never so lucky. Rather, we are forced to approximate the relation, $R(\,\cdot\,)$, by observing a subset, $\mathbf{x} \subset \xi$, of predictor variables that correlate most strongly with $\tilde{y}$ and then developing a simulation model, $y = g(\mathbf{x})$, that approximates the experimental results. Consequently, any simulation will not, in general, provide an exact replication of the actual scenario as $y \neq \tilde{y}$. We define model inadequacy to be the difference between the true response, $\tilde{y}$, and the model response, $y$. More specifically, if the true response, $\tilde{y}$, is random, model inadequacy is a measure of the difference between $y$ and the average of $\tilde{y}$.

iv. *Residual Variability:*

Roughly speaking, residual variability refers the uncertainty that remains after all the other sources of uncertainty have been accounted for, and can be either aleatory or epistemic (or both). In one case, the physical process may be inherently random so that residual variability represents the random variability that remains after parametric variability has been considered. In other words, this is the variability that was averaged out when computing the model inadequacy. On the other hand, the variability could be reduced if more conditions were specified. In this case, our knowledge of the process lacks sufficient detail to distinguish between different experimental conditions that yield different outcomes. For instance, the variability may actually be a type of parametric variability, but we are simply unable to observe or measure the appropriate parameter.

v. *Metamodel Uncertainty:*

This type of uncertainty will be important when we discuss metamodeling in Chapter IV. The basic idea of metamodeling is to evaluate the output from the simulation code for a specified set of input configurations, and then to build an approximating model based on this data. For any input configuration that is not included in this data set, the output of the simulation code will be unknown. While the metamodel will be able to provide an estimate of this unknown output, there will be some amount of uncertainty regarding the true value. This uncertainty is called metamodel uncertainty. Note, Kennedy and O'Hagan [19] refer to this as code uncertainty, but we feel that metamodel uncertainty is a more natural name.

We cautioned previously that this classification of uncertainty is not complete. For instance, in some applications, it may be necessary to further distinguish between model inadequacies resulting from an inadequate physical representation of the system and those that are simply the result of discretization and round-off error. For the present purposes, however, we shall restrict our attention to parametric uncertainty and parametric variability. This decision is made out of practical necessity, as the experimental data required for quantifying model inadequacy is unavailable for the case studies that we will consider in Chapter V. That being said, it is important to recognize that the methods to be discussed herein are generally applicable for handling any of the aforementioned uncertainties.

## II.2 Measures of Uncertainty

In the probabilistic framework for representing uncertainty, all of the relevant information regarding the uncertainty of the model output, $y$, is contained in its PDF. Note that we are assuming that $y$ is continuous-valued; the discrete-valued case is not considered in this work. However, the very fact that we are considering computer models implies that the relation $y = g(\mathbf{x})$ cannot be represented by an analytical expression; consequently, we cannot, in general, obtain an expression for the PDF of $y$. However, decision makers generally do not require all of the information contained in the entire PDF, but instead rely on a set of summary measures that provides a sufficient description of the uncertainty in $y$. The summary measures are integral quantities that include the various moments of the distribution for $y$, the most common of which are the expected value (or mean):

$$\mathrm{E}[y] = \int g(\mathbf{x}) p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}, \tag{II.1}$$

and the variance:

$$\mathrm{var}[y] = \mathrm{E}[y - \mathrm{E}(y)]^2 = \int [g(\mathbf{x}) - \mathrm{E}(y)]^2 p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = \mathrm{E}[y^2] - \mathrm{E}^2[y]. \tag{II.2}$$

The lack of integration bounds in Eqs. (II.1) and (II.2) is intended to imply that the integration is taken over all possible values of $x_i$ for each of the $m$ components of $\mathbf{x}$ (more formally, the integration is performed over the support of the PDF of $\mathbf{x}$).

Additionally, one may require the percentiles of $y$, where the $\alpha$-percentile, $y_\alpha$, is defined as the value of $y \in \mathbb{R}$ which the probability that $y \leq y_\alpha$ is $100 \times \alpha$. Mathematically, this quantity is determined as the solution to:

$$\int_{-\infty}^{y_\alpha} p_{\mathrm{Y}}(y) dy = \alpha. \tag{II.3}$$

Equation (II.3) is of little use, however, since we are unable to provide an expression for the PDF, $p_Y(y)$. By introducing the indicator function, $\mathbf{I}_{g(\mathbf{x})\geq C}(\mathbf{x})$, which satisfies the following properties:

$$\mathbf{I}_{g(\mathbf{x})\geq C}(\mathbf{x}) = \begin{cases} 1 & \text{if } g(\mathbf{x}) \geq C \\ 0 & \text{otherwise} \end{cases}, \tag{II.4}$$

for some number, $C$, we can compute the $\alpha$-percentile as the solution to:

$$\alpha = G(y_\alpha) = 1 - \int \mathbf{I}_{g(\mathbf{x})\geq y_\alpha}(\mathbf{x}) p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}. \tag{II.5}$$

In Eq. (II.5), we have taken $C = y_\alpha$ and the expression, $G(\,\cdot\,)$, is merely intended to illustrate that the last term in Eq. (II.5) is to be viewed as a function of $y_\alpha$.

Although not technically a measure of uncertainty, another quantity that is frequently of interest is the threshold exceedance probability, $P_E$, which is the probability that the computer model's output exceeds some a priori defined threshold, $y_{max}$. Making use of the indicator function given in Eq. (II.4), we can compute the threshold exceedance probability as:

$$P_E = \int \mathbf{I}_{g(\mathbf{x})\geq y_{\max}}(\mathbf{x}) p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}. \tag{II.6}$$

The threshold exceedance probability is the primary quantity of interest in structural reliability problems where a structure is considered to fail when stress (load) in a structural member exceeds the material's yield stress (capacity) [20-24]. More recently, this measure has been of interest in the reliability assessment for passive nuclear safety systems [25-31]. In these cases, an extension of the structural load-capacity framework is adopted to describe the performance of the system by defining, say, the peak cladding temperature to be a surrogate for the load and representing the capacity as some predetermined temperature above which the structural integrity of the cladding rapidly deteriorates. Computation of the threshold exceedance probability is of particular interest for this work because the case studies that we discuss in Chapter V are examples taken from the passive system reliability assessment literature.

## II.3 Computation of Uncertainty Measures

Having discussed the primary metrics of interest in UA, it remains to actually compute these measures. Since each of the metrics defined above is expressed as an integral, the problem is one of numerical integration; as such, a variety of methods exist for their computation. However, due to the use of the indicator function in Eqs. (II.5) and (II.6), the integrand in these expressions is not a smooth function so that standard numerical integration methods, such as Simpson's rule, are inefficient. Furthermore, the computer model could depend on hundreds of inputs, so that the integration must be performed over the high-dimensional domain of **x**. The accuracy of lattice-based numerical integration methods scales exponentially with the dimensionality of the problem, often making such methods infeasible for practical UA problems with high dimensionality; this is sometimes referred to as the "curse of dimensionality" [32]. Nevertheless, Kuo and Sloan [32] present a brief discussion of some highly efficient lattice-based methods that can be used for integrating over high-dimensional inputs defined on a bounded domain (i.e., the unit hypercube). An alternative to the lattice-based methods is offered by stochastic simulation methods, including standard Monte Carlo simulation (MCS) and its many variants. Because current practice in UA relies almost exclusively on MCS integration, we will limit our discussion to methods of this type. In the following sections, we provide a brief overview of the MCS method, followed by a description of a few of the more popular variants, such as Importance Sampling (IS) and Latin Hypercube Sampling (LHS). We shall also discuss, in brief, some of the more recently developed sampling-based strategies, such as Line Sampling (LS) and Subset Simulation (SS). In addition, we will discuss many of the relative advantages and drawbacks of each of these techniques. For convenience, many of the important points from this section have been summarized in Appendix A.

### *II.3.A Standard Monte Carlo Simulation (MCS)*

Each of the uncertainty measures listed above can be expressed as an integral of the form:

$$\int h(\mathbf{x})p_{\mathrm{X}}(\mathbf{x})d\mathbf{x} \tag{II.7}$$

where the function, $h(\mathbf{x})$ , depends on the uncertainty measure being computed and $p_{\mathrm{X}}(\mathbf{x})$ is the joint distribution for the model inputs. Notice that Eq. (II.7) is simply the expression for the expectation of the function $h(\mathbf{X})$:

$$\mathrm{E}[h(\mathbf{x})] = \int h(\mathbf{x}) p_{\mathrm{X}}(\mathbf{x}) d\mathbf{x} = \int h\, p_H(h)\, dh. \tag{II.8}$$

Although the distribution function of $h(\mathbf{x})$, $p_H(h)$, is generally unknown, MCS provides a way for us to draw samples from this distribution. This is accomplished by randomly sampling a set of input configurations, $\{\mathbf{x}^{(j)}\}$, $j = 1,2\ldots N_S$, of size $N_S$ from the joint distribution, $p_{\mathrm{X}}(\mathbf{x})$, and then evaluating $h^{(j)} = h(\mathbf{x}^{(j)})$, $j = 1,2,\ldots N_S$ for each of the $N_S$ samples. Each of the $h^{(j)}$ is a random variable distributed as $p_H(h)$. Then, we can estimate the expected value given in Eq. (II.8) by computing the sample average:

$$\mathrm{E}[h(\mathbf{x})] \approx \bar{h} = \frac{1}{N_S} \sum_{j=1}^{N_S} h^{(j)}. \tag{II.9}$$

We can obtain an expression for the accuracy of our estimate by computing the standard deviation of Eq. (II.9):

$$s_E = \sqrt{\mathrm{var}[\bar{h}]} = \frac{\sigma_H}{\sqrt{N_S}} \tag{II.10}$$

where $\sigma_H$ is the standard deviation of $h$, which can be estimated by the well-known expression for the unbiased sample variance [33]:

$$\sigma_H \approx \sqrt{\hat{\sigma}_H^2} = \frac{1}{\sqrt{N_S - 1}} \left[ \sum_{j=1}^{N_S} \left(h^{(j)} - \bar{h}\right)^2 \right]^{1/2}. \tag{II.11}$$

The expression in Eq. (II.10) is called the standard error. Several observations are worth noting at this point. First, inspection of Eq. (II.10) reveals that the accuracy of our estimate of the integral in Eq.(II.8) scales with $N_S^{-1/2}$, so that we can always obtain a better estimate by taking more samples. Moreover, the standard error does not depend on the dimension of **x**, thereby allowing MCS to overcome the "curse of dimensionality" mentioned previously. It is for this reason that MCS integration is so popular in practical UA studies. Additional desirable features of MCS include its straightforward implementation, regardless of the complexity of the function to be integrated, as well as the fact that it is quite intuitive and easy to understand.

As discussed previously, problems requiring the computation of the threshold exceedance probability, such as in Eq. (II.6), are very common in reliability assessment settings, so we will spend a moment to elaborate. In such cases, we take $h(\mathbf{x}) = \mathbf{I}_{g(\mathbf{x}) \geq y_{\max}}(\mathbf{x})$ to be the indicator function. Then, for each sample, $\mathbf{x}^{(j)}$, the quantity $h^{(j)}$ will equal unity with probability $P_E$, and will equal zero with probability 1-$P_E$. Thus, each $h^{(j)}$ is a Bernoulli random variable whose sum is a random variable drawn from a Binomial distribution with variance $N_S P_E (1 - P_E)$. Using this result, we can express the standard error of our estimate as:

$$s_E = \sqrt{\frac{P_E(1-P_E)}{N_S}} \approx \sqrt{\frac{\hat{P}_E(1-\hat{P}_E)}{N_S}} \tag{II.12}$$

where,

$$\hat{P}_E = \frac{1}{N_S} \sum_{j=1}^{N_S} I_{g(\mathbf{x}) \geq y_{\max}}\left(\mathbf{x}^{(j)}\right) \tag{II.13}$$

When dealing with highly reliable systems with small failure probabilities (i.e., <$10^{-4}$), the standard error in Eq. (II.12) can be misleading. For instance, the standard error may be $10^{-2}$, but if the failure probability is ~$10^{-4}$, then our estimate is clearly not accurate. It is convenient, therefore, to normalize the standard error in Eq. (II.12) by the estimated expected value in Eq. (II.13). This quantity is called the coefficient of variation, $\delta$, and is given as:

$$\delta = \frac{1}{\sqrt{N_S}} \sqrt{\frac{1-\hat{P}_E}{\hat{P}_E}}. \tag{II.14}$$

The percent error of the estimation is then simply $100 \times \delta$. If an approximate estimate of $P_E$ is available a priori, one can use Eq. (II.14) to estimate the required number of samples to obtain a desired accuracy. For instance, suppose we believe that $P_E \sim 10^{-4}$, and we wish to obtain an estimate accurate to within 10% of this estimate, which corresponds to letting $\delta = 0.1$. Solving for the number of samples, we obtain, $N_S \sim 10^6$. Hence, a rather large number of samples is required.

This simple example illustrates that the major drawback of using standard MCS to estimate the failure probability of highly reliable systems is the exceedingly large number of samples required, an observation that has been confirmed in the literature, e.g., in [21]. The computer model must be evaluated for each of the sampled input configurations. If we assume that the model takes one second to run a single configuration (a highly optimistic assumption), then one million samples will require approximately 11 days of continuous CPU time. For more realistic models that are commonly encountered in practice, each model evaluation may require hours or days. Consequently, the necessary computational time rapidly escalates to possibly unmanageable levels.

Various approaches to circumvent this computational burden have been presented in the literature. All of these methods can be roughly classified as either (i) attempts to decrease the number of samples required for an accurate estimate, or (ii) attempts to reduce the time required to evaluate the model by developing simplified approximations to the model. Methods falling under the former category include Latin Hypercube Sampling, importance sampling, and other advanced sampling schemes. These will be discussed in the next three sections of this thesis. Methods of the second category are referred to as metamodeling (also surrogate modeling) methods. These techniques are discussed in Chapter IV.

### *II.3.B Latin Hypercube Sampling (LHS)*

When sampling input configurations using MCS, the samples will tend to cluster about the mode (i.e., the region of highest probability) of the joint distribution, $p_{\mathbf{X}}(\mathbf{x})$, and very few points will be sampled from the low probability regions (i.e. the tails) of this distribution. LHS, on the other hand, was developed to generate samples with a more uniform coverage of the entire input domain [7]. The technique was first described by McKay et al. [34] in the 1970's, and has since been further developed for a variety of purposes by numerous researchers [35-38]. We present here a brief description of the method; the references can be consulted for additional details.

Suppose that all of the $m$ input variables, $x_i$, are independent (for correlated variables, see [38]), and we wish to draw a sample of size $N_S$. We first partition the domain of each $x_i$ into $N_S$

disjoint intervals of equal probability. Starting from $x_1$, we randomly sample one value in each of the $N_S$ intervals according to the distribution for $x_1$ in that interval. For each of the $N_S$ values for $x_1$, we randomly select one interval for variable $x_2$ from which to sample. This random pairing is done without replacement so that only one sample of $x_2$ is taken in each of the $N_S$ intervals. We continue this process, randomly pairing, without replacement, each of the $N_S$ values of $(x_1,x_2)$ with a value of $x_3$ from one of the $N_S$ intervals. This process is repeated until all $m$ variables have been sampled. The result is a sample set wherein each of the $m$ inputs has been sampled exactly once in each of its $N_S$ intervals.

LHS provides more uniform coverage of the domain for each input. Furthermore, the pairing of intervals retains the random nature of the sampling scheme, so that the MCS estimate for the integral given in Eq. (II.9) is still applicable. However, the samples are no longer independent since the intervals are paired without replacement. Hence, the standard error estimate given by Eq. (II.10) is no longer applicable. In fact, the standard error of the LHS estimate cannot be easily estimated. Nevertheless, LHS has been extremely popular in the literature because numerical experiments have demonstrated that, compared to standard MCS, LHS is capable of providing more accurate estimates of means with smaller sample sizes [36]. However, numerical experiments have also suggested that LHS is only slightly more efficient than standard MCS for estimating small failure probabilities [39].

### ***II.3.C Importance Sampling (IS)***

Importance sampling is a modification of MCS that seeks to reduce the standard error of estimation, and, hence, increase the efficiency, by biasing the samples [33]. In particular, for reliability problems, analysts are often interested in extreme behavior of the system where safety systems are severely challenged. Generally, this type of behavior is only expected when the inputs, themselves, take values in the tails (i.e., the low probability regions) of their respective distributions. As stated previously, samples drawn with MCS will tend to be clustered about the mode, so that very few samples will be drawn from the tail regions. Moreover, if no failures are observed for a set of samples, the estimated failure probability will be zero, but the coefficient of variation, given in Eq. (II.14), will be infinite. This is not desirable since, in such a case, we cannot obtain quantitative bounds on the failure probability. IS can overcome this issue by forcing more samples to be drawn from the important (extreme) regions.

As stated, the basic idea behind IS is to bias the samples so that they are drawn from important input regions. This is done by defining, a priori, a sampling distribution, $q_{\mathrm{X}}(\mathbf{x})$, from which to draw input samples. From Eq. (II.8), it follows that [33]:

$$\mathrm{E}_p[h(\mathbf{x})] = \int h(\mathbf{x}) p_{\mathrm{X}}(\mathbf{x}) d\mathbf{x} = \int \left( \frac{h(\mathbf{x}) p_{\mathrm{X}}(\mathbf{x})}{q_{\mathrm{X}}(\mathbf{x})} \right) q_{\mathrm{X}}(\mathbf{x}) d\mathbf{x} = \int \tilde{h}(\mathbf{x}) q_{\mathrm{X}}(\mathbf{x}) d\mathbf{x} = \mathrm{E}_q\left[\tilde{h}(\mathbf{x})\right] \quad \text{(II.15)}$$

where we have defined $\tilde{h}(\mathbf{x}) = h(\mathbf{x}) p_{\mathrm{X}}(\mathbf{x}) / q_{\mathrm{X}}(\mathbf{x})$, and the subscripts $p$ and $q$ denote that the expectation is taken with respect to the distributions, $p_{\mathrm{X}}(\mathbf{x})$ and $q_{\mathrm{X}}(\mathbf{x})$, respectively. We shall next demonstrate that one can reduce the standard error of the estimate, $\bar{h} = \mathrm{E}_p[h(\mathbf{x})] = \mathrm{E}_q[\tilde{h}(\mathbf{x})]$, by judiciously choosing the sampling distribution, $q_{\mathrm{X}}(\mathbf{x})$.

We consider the threshold exceedance problem from Eq. (II.6), so that $h(\mathbf{x}) = \mathbf{I}_{g(\mathbf{x}) \geq y_{\max}}(\mathbf{x})$. Then, from Eq. (II.10), the squared standard error of our estimate for the exceedance probability is given by $s_E^2 = \mathrm{var}[\tilde{h}(\mathbf{x})] / N_S$, where:

$$\mathrm{var}\left[\tilde{h}(\mathbf{x})\right] = \int \left( \frac{\mathbf{I}_{g(\mathbf{x}) \geq y_{\max}} p_{\mathrm{X}}(\mathbf{x})}{q_{\mathrm{X}}(\mathbf{x})} - P_E \right)^2 q_{\mathrm{X}}(\mathbf{x}) d\mathbf{x}. \quad \text{(II.16)}$$

Now, if we consider the idealized case where we know exactly the region where $g(\mathbf{x}) \geq y_{\max}$, then we can let $q_{\mathrm{X}}(\mathbf{x}) = \frac{1}{c} p_{\mathrm{X}}(\mathbf{x}) \mathbf{I}_{g(\mathbf{x}) \geq y_{\max}}(\mathbf{x})$ for some constant, $c$. Then, Eq. (II.16) reduces to $\mathrm{var}\left[\tilde{h}(\mathbf{x})\right] = (c - P_E)^2$. Furthermore, from Eq. (II.15) we have:

$$P_E = \mathrm{E}_q[\tilde{h}(\mathbf{x})] = \int \frac{h(\mathbf{x}) p_{\mathrm{X}}(\mathbf{x})}{q_{\mathrm{X}}(\mathbf{x})} q_{\mathrm{X}}(\mathbf{x}) d\mathbf{x} = c \int q_{\mathrm{X}}(\mathbf{x}) d\mathbf{x} = c \quad \text{(II.17)}$$

which implies $s_E^2 = 0$, the standard error of our estimate is zero. This is not unexpected since our choice of the sampling distribution required us to know, a priori, the region where $g(\mathbf{x}) \geq y_{\max}$, and hence, $P_E$.

In reality, such perfect information would certainly not be available; or, if it were, the problem would already be solved. Still, even a suboptimal selection of $q_{\mathrm{X}}(\mathbf{x})$ could yield a reduced variance. On the other hand, a poor choice for $q_{\mathrm{X}}(\mathbf{x})$ could lead to a large increase in the variance estimate. For instance, suppose for the sake of argument that for most of the points satisfying $\mathbf{I}_{g(\mathbf{x})\geq y_{\max}}(\mathbf{x})=1$, the distributions, $q_{\mathbf{X}}(\mathbf{x})$ and $p_{\mathbf{X}}(\mathbf{x})$, are proportional (i.e., the ratio $q_{\mathbf{X}}(\mathbf{x})/p_{\mathbf{X}}(\mathbf{x})=\kappa$, for some constant, $\kappa$). For these points, $\tilde{h}(\mathbf{x})=h(\mathbf{x})/\kappa$. However, suppose that for some point, $\mathbf{x}_o$, $q_{\mathbf{X}}(\mathbf{x}_o)/p_{\mathbf{X}}(\mathbf{x}_o)<<\kappa$. At this point, we will have $\tilde{h}(\mathbf{x}_o)=h(\mathbf{x}_o)p_{\mathbf{X}}(\mathbf{x}_o)/q_{\mathbf{X}}(\mathbf{x}_o)>>h(\mathbf{x}_o)/\kappa$. Consequently, the variance computed from Eq. (II.16) will be high due to $\tilde{h}(\mathbf{x})$ being disproportionately larger for $x_o$ than for the other $x$. This illustrates the biggest drawback of the IS method – that poor choices of the sampling distribution can actually lead to worse estimates. This is particularly troublesome for complex problems where it is impossible to know, a priori, where the important regimes are located in the input domain, and this is precisely the reason that standard IS methods have not received much attention in the UA and SA literature.

### *II.3.D Line Sampling & Subset Simulation*

With the exception of importance sampling, all of the aforementioned methods require an excessively large sample size to make reasonable estimates of small failure probabilities – a very common problem in safety-critical industries, such as the nuclear power industry, where quantitative safety assessments must be carried out for highly reliable systems. However, IS requires a significant amount of prior information regarding the model's behavior, which is often unavailable. As a consequence, there has been a great deal of effort to develop sampling strategies specifically tailored to problems of this type. Two such methods that have resulted from these efforts are Line Sampling (LS) [40-45] and Subset Simulation (SS) [45-48]. Both of these methods are somewhat more involved than the previous methods, and will not be discussed in detail here. Rather, we shall simply provide a brief description to convey the general substance behind each. Of particular relevance are references [44] and [48], which demonstrate, respectively, the application of LS and SS to the Gas-Cooled Fast Reactor (GFR) case study discussed in Chapter V of this thesis.

Line Sampling can be described as a stochastic extension of the classical First-Order Reliability Method (FORM) and Second-Order Reliability Method (SORM) familiar to structural reliability analysis [45]. Haldar and Mahadevan [49] provide an excellent textbook introduction to the FORM/SORM techniques. As a technical detail, the LS implementations presented in the literature that we reviewed require all of the model inputs to be independently and identically (iid) normally distributed; however, Koutsourelakis et al. [40] claim in their original presentation that the LS method can be utilized for any joint distribution, provided at least one of the inputs is independent of all of the others, and that the Gaussian distribution is adopted only for notational simplicity. In any case, while methods do exist for transforming random variables [50-52] to be iid standard normal, these are only approximate and some errors are introduced into the analysis [53]. Regardless, some approximation error is likely tolerable if standard sampling techniques are incapable of providing reasonable results.

Line Sampling requires the analyst to specify, a priori, the direction in the input space that the failure domain (i.e., the set of $\mathbf{x}$ for which $g(\mathbf{x}) \geq y_{\max}$) or other region of interest is expected to lie. This is accomplished through the specification of a unit vector, $\boldsymbol{\alpha}$, which points in the desired direction. Starting from a randomly sampled input configuration, a search is performed in the direction specified by $\boldsymbol{\alpha}$ for a point, $\mathbf{x}$, satisfying $g(\mathbf{x}) = y_{\max}$ (these points comprise a hypersurface in the input space). In effect, the length of the line connecting this point with the origin (in standard normal space) provides a measure of the failure probability. While numerous technical details have been neglected in this description, an algorithmic description of this procedure can be found in [41,42,45].

Much of the computational effort rests in the root finding process to locate the points on the hypersurface where $g(\mathbf{x}) = y_{\max}$, which may require numerous evaluations of the model. However, LS is most effective for estimating small failure probabilities, and in such cases, it may be unnecessary to locate these points with high accuracy, sufficing instead to identify bounds that contain such a point. It should also be apparent that the selection of the direction, $\boldsymbol{\alpha}$, is very important. Indeed, if $\boldsymbol{\alpha}$ were chosen perpendicular to the hypersurface defined by, $g(\mathbf{x}) = y_{\max}$, then no solutions to this equation would be found during the root finding procedure. However, Koutsourelakis et al. state that in such a worst-case scenario, LS would perform equally as well as MCS [40]. Although this statement is true on a per-sample basis, when one accounts for the model evaluations that are effectively wasted while searching for failures in a direction where no failures occur, we find that LS will require more model evaluations than MCS for the same accuracy. Nevertheless, in practical situations, the analyst often has some qualitative idea of how the input parameters affect the model and should be able to roughly specify an appropriate direction, $\boldsymbol{\alpha}$, in which to search. Therefore, this worst-case scenario is not likely to be realized in practice, and LS should always outperform standard MCS for estimating small failure probabilities. More quantitative approaches to appropriately selecting $\boldsymbol{\alpha}$ are also discussed in the literature [40,41].

Subset Simulation is an alternative advanced sampling procedure that was proposed by Au and Beck for structural reliability problems [46,47]. SS is essentially a variant of importance sampling, with the advantage being that the sampling distribution does not need to be specified a priori. The process works by expressing the failure event of interest, which we denote by $F$, as the conjunction of a sequence of $k$ intermediate events:

$$F = \bigcap_{i=1}^{k} F_i \tag{II.18}$$

where $F_i$ is the $i^{\text{th}}$ intermediate event. By defining the intermediate events such that $F_1 \supset F_2 \supset \cdots \supset F_k = F$ , the failure probability can be expressed as:

$$P_F = P(F_k) = P(F_1)\prod_{i=1}^{k-1} P(F_{i+1} \mid F_i) . \tag{II.19}$$

Furthermore, the intermediate events, $F_i$, are judiciously chosen such that each of the probabilities in Eq. (II.19) is relatively large (i.e. ~0.1), so that each can be estimated with relatively few samples; more information on how one can define the intermediate events is given in the references. Samples are drawn from the conditional distributions in Eq. (II.19) are using a Markov Chain Monte Carlo (MCMC) algorithm, as discussed in [46]. Since each subsequent $F_i$ approaches the final failure event, $F$, the procedure works by continually biasing the samples to the region of interest, $F$. Consequently, more samples are drawn from the failure domain, resulting in a decreased variance in the failure probability estimate.

Perhaps the greatest advantage of SS is that it requires very little input from the user a priori. That is, no sampling distribution needs to be specified, and it is not necessary to identify the direction in which failure is expected. Hence, the method is quite robust. However, as with any method that utilizes MCMC, care must be taken with regards to issues such as serial correlation between samples and convergence to the stationary distribution. Furthermore, although SS has been demonstrated to yield impressive increases in efficiency compared to MCS and LHS, it seems to be slightly less efficient than LS [45-47]. This may be justified, however, since it requires less information from the user. Moreover, SS is capable of identifying multiple, disconnected failure regions.

The choice of whether to use LS or SS for a particular problem depends on the amount of information that is available prior to performing the analysis. If it is known that the failure region is isolated in a particular region of the input space, and if some information is available regarding the location of this region, then LS is expected to be superior. However, if no such information is available, or if it is expected that there exist multiple disconnected failure regions scattered more or less randomly throughout the input space, then SS should be implemented. Finally, while both methods have been quite successful based on the numerous studies in the literature, there does not seem to exist any indication that it is possible to reduce the required number of model evaluations to anything below a few hundred (i.e., 250-500) [42-45,48]. If the model requires many hours, or days, to perform a single evaluation, even this increase in efficiency is insufficient. As a consequence, analysts are forced to rely on metamodeling techniques, as discussed in Chapter IV.

# III SENSITIVITY ANALYSIS

Sensitivity analysis (SA) refers to a collection of tools whose aim is to elucidate the dependency of (some function of) the model output, $y$, on (some function of) the set of model inputs, $\mathbf{x}$. We include the qualifier 'some function of' to clarify that such efforts include not only direct attempts to determine how $y$ depends on $\mathbf{x}$, but also any attempt to assess how the uncertainty in $y$ depends on the uncertainty in $\mathbf{x}$. Sensitivity analysis methods can be categorized as being either deterministic or statistical [54,55]. Deterministic methods use derivative information to propagate deviations (or errors) through the simulation, and then use this information for uncertainty quantification purposes. These methods are useful when an intimate knowledge of the inner-workings of the simulation code is available, and can potentially be used to develop simulation models with built-in UA capabilities. Zhao and Mousseau provide a detailed discussion of one class of deterministic SA methods, the so-called Forward SA methods, in a recent report [56]. An alternative class of deterministic SA techniques, known as the Adjoint SA methods, is based on the development of an adjoint model for the system. These methods are highly complex, relying on some rather abstract mathematical constructs, and will not be discussed herein; interested readers should consult references [54,57-59] for more details.

In addition to the distinction between deterministic and statistical methods, SA methods can be classified as either local or global. As the name suggests, local methods consider the variation in the model output that results from a local perturbation about some nominal input value, whereas global methods account for output variability that results when the input parameters are varied throughout their entire domain. Deterministic methods, being derivative-based (i.e., using Taylor series expansions), are almost exclusively local; the only exception seems to be the Global Adjoint Sensitivity Analysis Procedure (GASAP) [54]. For the purposes of this thesis, we are most interested in global SA methods, and in particular the statistical SA methods. In the following sections, we provide a more detailed explanation of many of the more useful statistical SA methods. Additional details, as well as details concerning any methods that we only mention in passing, can be found by consulting the appropriate references. In particular, references [60-65] provide examples of practical applications of SA and UA. Turányi [66] and Hamby [67] provide an excellent review of existing SA techniques, in addition to the review papers by Cacuci and Ionescu-Bujor cited above [54,55]. A thorough introduction to SA can be found in the textbook by Saltelli, Chan, and Scott [68], and a more summary introduction is given by Saltelli et al. [69]. Appendix B of this thesis summarizes many of the main conclusions taken from the following sections.

## III.1 Scatter Plots

Scatter plots are the simplest tool for performing sensitivity analysis, and provide a visual indication of possible dependencies. Scatter plots are constructed by first performing a Monte Carlo simulation, sampling a set of input configurations, $\{\mathbf{x}^{(j)}\}$, $j = 1,2 \ldots N_S$, of size $N_S$ and evaluating, for each $\mathbf{x}^{(j)}$, the model output, $y^{(j)}$. It should be noted that the sample size need not be as large as that required for the full uncertainty analysis (see Section II.3.A) since the scatter plots are only used to qualitatively investigate relationships and no quantitative statements are derived. From the MCS, one obtains a set, $[\mathbf{x}^{(j)}, y^{(j)}]$, of input-output mappings. Then, for each of the $m$ model inputs, $\mathbf{x}_i$, a plot of $\mathbf{x}_i^{(j)}$ vs. $y^{(j)}$ is created. Visual inspection of each of these plots can provide a variety of insights regarding the behavior the model being studied. Figure 1 illustrates a set of scatter plots produced from the GFR case study described in Chapter 0. For this example, there are nine input parameters, as labeled, and the output of interest is the hot channel core outlet temperature.

Inspection of Fig. 1 reveals a strong negative correlation (trend) between the model output and the system pressure. On the other hand, the model output appears to depend only weakly, if at all, on the remaining eight parameters. As a consequence, an analyst may decide to neglect some of the other parameters during the subsequent quantitative UA to reduce the computational demand. We note, however, that in such cases, alternative measures must be taken to account for these neglected uncertainties at a later stage in the analysis. Moreover, if one is interested in the possibility of the model output exceeding some critical limit, then it may be possible to identify threshold effects from the scatter plots [30]. For example, in the GFR study, the maximum allowable outlet temperature was selected as 1200°C. By observing those input configuration that result in this limit being exceeded, we can obtain additional information regarding potential system improvements. Specifically, in Fig. 1, we see that no failures occurred for pressures greater than 1500 kPa, and engineers may then wish to make design alterations or implement operational regulations that further minimize the possibility of the pressure dropping below this level.

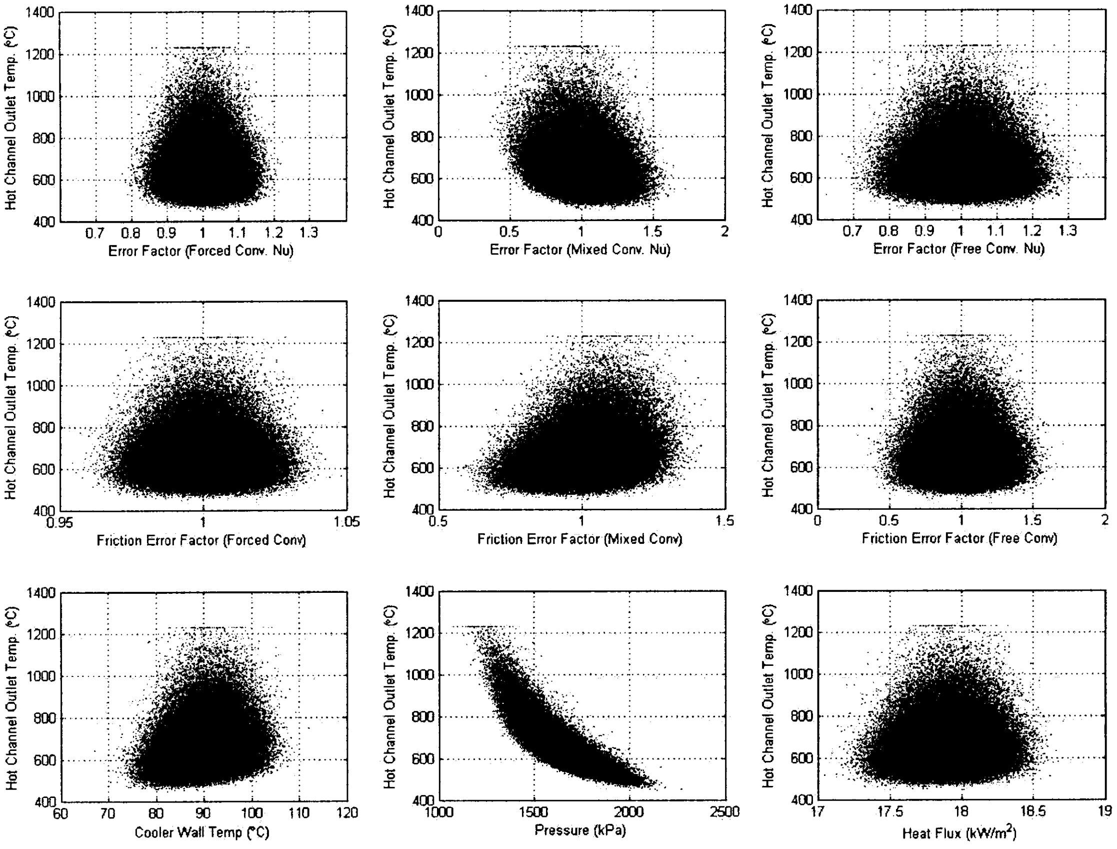

Figure 1: Scatter Plots for GFR Case Study.

An additional insight that can be gained from scatter plots regards modeling errors. If the scatter plots reveal trends that disagree with the engineers' intuition, it is possible that a mistake was made during the modeling process. In addition, anomalous data can reveal coding errors or problems with the numerical routine that is implemented. As an example, a close inspection of Fig. 1 reveals a possible saturation effect for outputs near 1225°C. Indeed, the model used in this study terminates the iteration process when the temperature falls outside the range for which fluid property data is available. If this were more than just a case study for demonstration purposes, such an effect could be cause for concern.

While scatter plots are quite versatile and can be extremely revealing about model behavior and system performance, they are limited in that they can only provide qualitative information. Furthermore, if the model output is only moderately dependent on each of the model inputs, it may not be possible to discern which parameters are actually most important from visual inspection alone. In addition, visual inspection of scatter plots may be impractical in cases where there are a large number (i.e., ~100) model inputs. Nevertheless, in many instances, scatter plots can provide an excellent starting point for a more quantitative SA, provided that the MCS data are available or can be easily generated. In cases where each model evaluation takes several hours, this may not be feasible and more structured approaches, such as experimental design procedures, should be adopted; these methods will be discussed in more detail in subsequent sections.

## III.2 Monte Carlo Filtering (MCF)

In the previous section, it was noted that threshold effects could be identified by observing only those inputs that led to the model output exceeding some critical value. Suppose that we took each of these samples and put it in a bin labeled ***B*** (behavioral), and put the remaining samples in a bin labeled ***NB*** (non-behavioral). Then, we could create two sets of scatter plots with one set consisting of only the samples in bin ***B***, and the other consisting of the samples in bin ***NB***. By comparing these two sets of scatter plots, it might be possible to make inferences on the influences of certain parameters on the model output; for instance, substantial differences between the two scatter plots for a particular parameter would indicate that the output is significantly more likely to exceed the threshold value if that parameter takes a high (low) value than otherwise. Alternatively, instead of making two sets of scatter plots, we could plot the empirical cumulative distribution function (CDF) of the samples belonging to bin ***B*** and of those belonging to bin ***NB***. As with the scatter plots, large differences in these distributions for any parameter indicate that the parameter has an appreciable influence on the model output. To illustrate, Fig. 2 provides plots of the empirical CDFs for the parameters from the GFR study. Again, it is overwhelmingly clear that the pressure is a highly influential parameter. Moreover, there is more evidence suggesting that the error factor for the mixed convection friction factor is also an influential parameter; this dependency was less obvious based on the scatter plots given in Fig. 1.

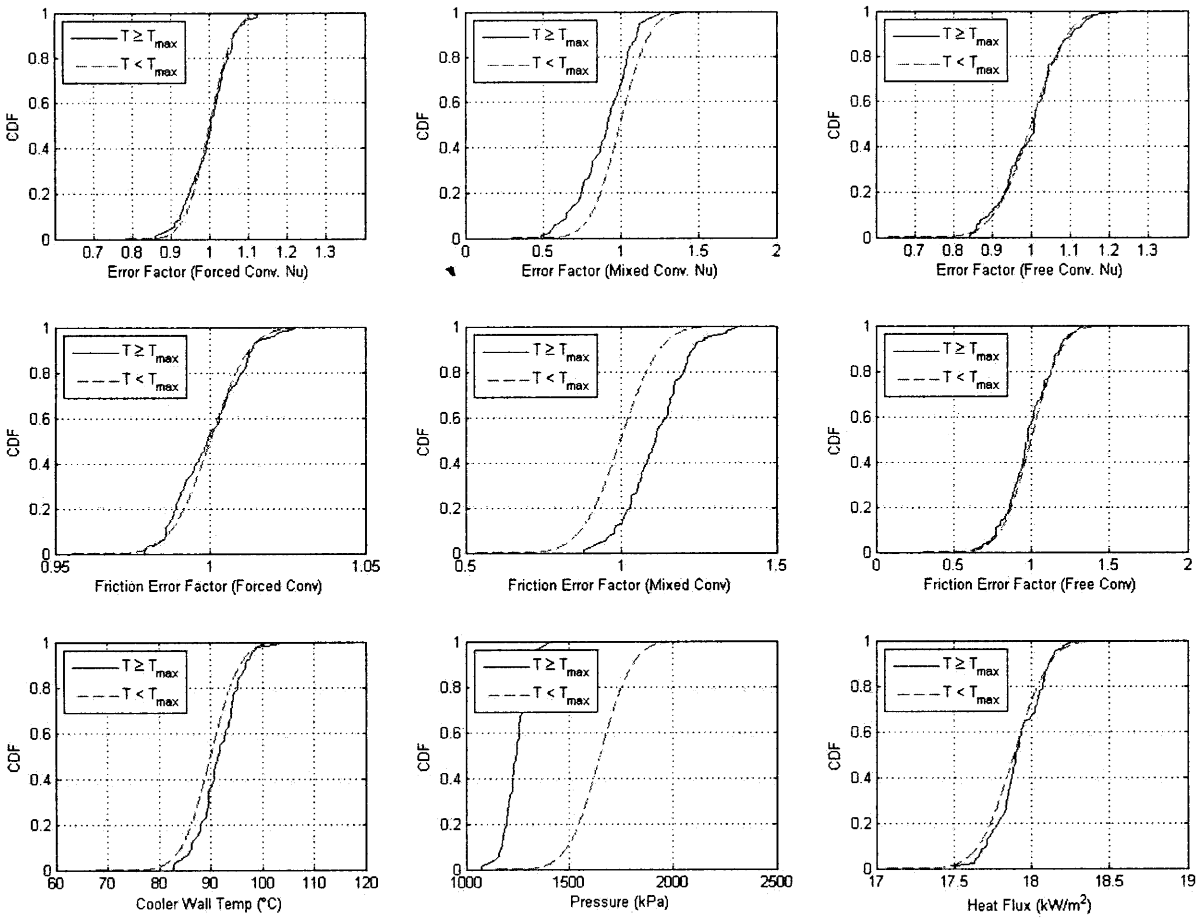

Figure 2: Empirical Cumulative Distribution Functions for Filtered Samples in GFR Study

While it may occasionally be possible to identify the influential parameters by visual inspection, in many cases this can be misleading as an apparent discrepancy between two empirical CDFs can simply be an artifact of the sampling process. Monte Carlo Filtering overcomes this limitation by utilizing various statistical goodness-of-fit tests, such as the Kolmogorov-Smirnov test, to determine whether the difference between the two empirical distributions is statistically significant [69]. It should be noted that essentially the same analysis has been referred to as Generalized Sensitivity Analysis in the literature [70].

Besides being able to determine the individual parameters that influence the model's output, additional MCF tests can be performed to extract information regarding two-way interaction effects. The details of this process are provided by Saltelli et al. and will not be described here [69]. Rather, it suffices to say that the drawback of MCF is its limited statistical power, especially for identifying the parameters that influence rare events (e.g., small failure probability). In these cases, unless a large number of simulations is performed, the number of simulations that will result in the occurrence of such an event will be quite small, and as a result, goodness-of-fit tests will be incapable of determining, with a reasonable degree of statistical significance, whether a particular parameter is influential.

## III.3 Regression Analysis

Regression analysis refers to a diverse collection of tools that are designed to make inferences concerning the structure of the model under consideration. These methods are generally classified as either parametric or nonparametric and are distinguished by the assumptions required regarding the structure of the regression models. In particular, parametric regression requires the analyst to specify, a priori, the functional form of the regression model (i.e., linear, quadratic, etc.), whereas nonparametric regression requires no such assumptions. To be more specific, nonparametric regressions generally require a specified model form on a local scale, but the global (i.e., over the entire input domain) model is constructed in such a way as to not be limited by these assumptions; examples include cubic spline interpolants, which assume that the regression model is a cubic polynomial on a local scale connecting three data points, but the global model need not be a cubic polynomial. Storlie and Helton [71] provide an excellent discussion of a variety of regression methods, both parametric and nonparametric, and demonstrate the application of these methods to several case studies in a sister article [72]. A review of these articles demonstrates the vast array of existing regression techniques, and a review of comparable breadth is beyond the scope of this thesis. Thus, we shall limit our discussion to some of the simplest regression methods. It should be noted, however, that the metamodeling methods discussed in Chapter IV are actually regression methods, and these will be discussed in more detail later. In particular, polynomial response surfaces are a particular example of linear parametric regression models, while kriging and artificial neural networks can be considered special cases of, respectively, nonparametric and nonlinear regression models.

Although it is true that regression analysis and metamodeling are related, they are often performed under differing contexts. For instance, many metamodels are regression models that are built specifically with the intent of predicting the output from some process (e.g., a computer simulation) at input configurations that have not been observed. On the other hand, regression analysis is frequently employed only as a method for identifying trends in the process response data, such as identifying whether the output varies in a more or less linear manner with one of the input parameters. In such cases, the regression model need not be as accurate as when attempting to make predictions. This distinction is important for understanding how some of the simpler regression models (i.e., linear or quadratic models), which are often sufficient for identifying the input parameters to which the model output is most sensitive, can fail to provide reasonable predictions when used as a metamodel for a complex simulation code. This issue is discussed in greater detail in Chapter IV. At present, we shall proceed with an overview of some of the basic regression techniques that are effective for input parameter ranking.

The simplest, and most commonly used, parametric regression methods are the linear regression models. These methods attempt to approximate the relation, $y = g(\mathbf{x})$, which represents the simulation model, with a simplified expression, $\hat{y}$, where:

$$y = \hat{y} + \varepsilon = \beta_0 + \sum_{i=1}^{q-1} f_i(\mathbf{x})\beta_i + \varepsilon . \tag{III.1}$$

In Eq. (III.1), $\varepsilon$ is a zero-mean random error term representing the regression approximation error, the $\beta_i$ represent a set of $q$ unknown regression coefficients to be determined from the data, and $f_i(\mathbf{x})$ are a set of $q$-1 functions that are assumed to be known. That is, $f_i(\cdot)$ can, in general, represent any function, but must be specified a priori. Consequently, we see that the linear regression model does not assume that the model output, $y$, depends linearly on the inputs, $\mathbf{x}$. Rather, the only linearity restriction is that the approximator, $\hat{y}$, be a linear combination of the unknown regression coefficients, $\beta_i$. Although Eq. (III.1) is quite general, in practice it is often difficult to specify the functions, $f_i(\mathbf{x})$, that are most appropriate for the model in question. Therefore, one often begins the analysis with the assumption that $y$ is, in fact, a linear function of $\mathbf{x}$, so that $f_i(\mathbf{x}) = x_i$ for each of the $m$ input parameters and $q = m + 1$ is the total number of unknown regression coefficients. The result is a regression model of the form:

$$\hat{y} = \beta_0 + \sum_{i=1}^{m} x_i\beta_i = \beta_0 + \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta} , \tag{III.2}$$

where $\boldsymbol{\beta}$ is an $m \times 1$ vector of regression coefficients. We refer to models of this form as standard linear models to distinguish between the general linear models given by Eq. (III.1). To estimate the coefficients $\beta_i$, we must obtain a set of observations, or input/output mappings,

$[y^{(j)}, \mathbf{x}^{(j)}], j = 1, 2, \ldots, N_S$ of the model output; these can be obtained from a standard Monte Carlo simulation or from more sophisticated experimental design procedures, as discussed briefly in Chapter IV. With this data, the regression coefficients are typically estimated by a least-squares procedure [71], giving a set of coefficient estimates $\hat{\beta}_i$.

Using the fact that $\varepsilon$ is a zero-mean random variable, we see from Eq. (III.1) that $\mathrm{E}(y) = \mathrm{E}(\hat{y})$. Thus, by assuming $\beta_i$ to be fixed constants and taking the expectation of Eq. (III.2), we have:

$$\mathrm{E}(y) = \beta_0 + \sum_{i=1}^{m} \mathrm{E}(x_i)\beta_i \tag{III.3}$$

Note that, in the Bayesian interpretation, $\beta_i$ are assumed to be random variables; this is discussed in more detail in Chapter IV. Subtracting Eq. (III.3) from Eq. (III.2) yields:

$$y - \bar{y} = \sum_{i=1}^{m} \hat{\beta}_i (x_i - \bar{x}_i) \tag{III.4}$$

where we have replaced the terms in Eq. (III.3) with its corresponding estimate obtained from our data sample; that is, $\bar{y}$ and $\bar{x}_i$ represent the sample means of $y$ and $x_i$, respectively, and each of the $\beta_i$ have been replaced by their corresponding estimate, $\hat{\beta}_i$. It is often convenient to somehow normalize Eq. (III.4). One possibility is to divide Eq. (III.4) by the sample mean, $\bar{y}$, giving:

$$\frac{y - \bar{y}}{\bar{y}} = \sum_{i=1}^{m} \hat{\beta}_i \frac{(x_i - \bar{x}_i)}{\bar{y}} = \sum_{i=1}^{m} \left( \frac{\hat{\beta}_i \bar{x}_i}{\bar{y}} \right) \frac{(x_i - \bar{x}_i)}{\bar{x}_i}, \tag{III.5}$$

where the terms given by $\hat{\beta}_i \bar{x}_i / \bar{y}$ provide a measure of the effect of varying the parameter, $x_i$, by a fraction of its nominal, or mean, value. Alternatively, from the definition of the sample standard deviation of $y$ and $x_i$, respectively:

$$\hat{s}_y = \left[ \sum_{j=1}^{N_S} \frac{(y^{(j)} - \bar{y})^2}{N_S - 1} \right]^{1/2} \text{ and } \hat{s}_i = \left[ \sum_{j=1}^{N_S} \frac{(x_i^{(j)} - \bar{x}_i)^2}{N_S - 1} \right]^{1/2} \tag{III.6}$$

we can obtain an expression analogous to Eq. (III.5) as:

$$\frac{y-\bar{y}}{\hat{s}_y}=\sum_{i=1}^{m}\left(\frac{\hat{\beta}_i\hat{s}_i}{\hat{s}_y}\right)\frac{(x_i-\bar{x}_i)}{\hat{s}_i}. \tag{III.7}$$

In this latter case, the coefficients $\hat{\beta}_i\hat{s}_i/\hat{s}_y$ are called the Standardized Regression Coefficients (SRC), and provide an indication as to the effect of varying parameter, $x_i$, by a fraction of its standard deviation. While the coefficients in both Eqs. (III.5) and (III.7) can be used to assess parameter importance, it is important to recognize that if any of the $x_i$ are correlated, a ranking based solely on these coefficients can be misleading [71]. If, however, all of the $x_i$ are uncorrelated, then a large absolute value for these coefficients, $|\hat{\beta}_i\hat{s}_i/\hat{s}_y|$ or $|\hat{\beta}_i\bar{x}_i/\bar{y}|$, indicates that the corresponding parameter is highly influential. Conversely, a small absolute value does not necessarily indicate that a parameter is not influential, as the assumption of linearity in Eq. (III.2) could be poor. Using the Standardized Regression Coefficients, and assuming that the supposed model form (linear, in this case) is adequate, then a variety of hypothesis tests can be exploited to assess whether any of the SRCs are significantly, in the statistical sense, different from zero [73]. As Storlie and Helton warn, however, these tests should only be used for general guidance and their results should not be misconstrued as absolute [71]. This is because the results of these tests depend critically upon assumptions regarding the distribution of the error terms, $\varepsilon$, and these assumptions (namely, that $\varepsilon$ is Gaussian) are often invalid. This is particularly true when building a regression model to describe the output from a deterministic simulation model; more will be said regarding this in Chapter IV.

The effectiveness of the regression model for ranking the input parameters based on their influence of the model output is clearly dependent on how well the regression model fits the observations. One method for quantifying this goodness-of-fit is to compute the coefficient of determination, $R^2$, defined as:

$$R^2=\frac{SS_{reg}}{SS_{tot}}=1-\frac{SS_{res}}{SS_{tot}} \tag{III.8}$$

where $SS_{reg}$, $SS_{res}$, $SS_{tot}$ are, respectively, the regression, residual, and total sum-of-squares, and can be computed with the following formulae:

$$SS_{reg} = \sum_{j=1}^{N_S} (\hat{y}^{(j)} - \bar{y})^2 \tag{III.9a}$$

$$SS_{res} = \sum_{j=1}^{N_S} (\hat{y}^{(j)} - y^{(j)})^2 \tag{III.9b}$$

$$SS_{tot} = \sum_{j=1}^{N_S} (y^{(j)} - \bar{y})^2 \tag{III.9c}$$

The coefficient of determination takes values between zero and unity, with values near zero indicating that the regression model provides a poor fit to the data and values near unity indicating a good fit. Specifically, $R^2$, can be interpreted as the fraction of the variance of $y$ that is explained, or accounted for, by the regression model $\hat{y}$. Clearly, the coefficient of determination accounts only for the data that were available when it was computed, and, as we shall discuss in Chapter IV, $R^2$ does not provide a measure of the predictive capability of the regression model; that is, despite the regression model's fidelity to the data $(R^2 \sim 1)$, there is no guarantee that an additional set of observations of $y$ will lie even remotely close to the values predicted by the regression model. On the other hand, if the regression model is only to be used for SA and is not intended for prediction purposes, then $R^2$ can still be a valuable measure of the quality of the regression model.

As Storlie and Helton [71] note, linear regression models for SA are usually developed in a stepwise manner. For example, at the initial stage, the regression model would only include the constant $\beta_0$. Then, $m$ different regression models would be developed with each consisting of the constant, $\beta_0$, plus one of $m$ the linear terms, $\beta_i x_i$. For each of the $m$ models, the coefficient of determination would be computed, and the one with the highest value for $R^2$ would be selected as the second stage regression model. This process would be repeated for the remaining $m$-1 parameters, thereby producing the third stage model and so on. The order in which each parameter is introduced to the regression model then provides one indication of variable importance, with those variables introduced in the first few stages of the stepwise process seeming to be the most influential. Additionally, the stepwise increase in $R^2$ at each stage of the process is also a useful indicator of variable importance [71]. As always, however, if the input variables are correlated, then caution must be exercised when implementing this procedure; for example, two correlated input variables may individually appear to be noninfluential, but if introduced to the model simultaneously, they may appear to have a much higher influence on the model output.

For many problems of interest, the standard linear regression model given by Eq. (III.2) may fail to provide an adequate fit to the observations. Consequently, a more complex regression model must be sought. The most natural extension of the standard linear model would be to introduce higher order terms, such as quadratic terms $x_i^2$. In addition, one could relax the assumption that parameter effects are additive by introducing cross-product terms for the form, $x_i x_j$ for $i \neq j$, representing first-order interaction effects. This leads to the quadratic regression model:

$$\hat{y} = \beta_0 + \sum_{i=1}^{m} x_i \beta_i + \sum_{j,k=1}^{m} x_j x_k \beta_{jk} \,. \tag{III.10}$$

The coefficients in Eq. (III.10) are usually estimated by the method of least-squares, and the remainder of the sensitivity analysis proceeds in exactly the same way as with the standard linear regression analysis. Although the quadratic regression model is capable of representing more complex model behavior as compared to the standard linear model, certain phenomena, such as asymptotes and discontinuities in the model output, cannot be accurately represented [71]. Even more generally, one can appeal to the use of nonlinear regression models. Unlike the general linear regression models given by Eq. (III.1), the nonlinear models relax the assumption that the regression model output, $\hat{y}$, be a linear combination of the unknown regression coefficients, $\beta_i$. Although the possibilities are limitless, one example of a nonlinear regression model would be of the form:

$$\hat{y} = \left[\beta_0 + (\beta_1 x_1 + \beta_2 x_2)^2 + \sin(\beta_3 x_3)\right] e^{-\beta_4 x_4^2} \,. \tag{III.11}$$

For nonlinear regression models, simple linear least-squares estimation cannot be used to estimate the coefficients. Consequently, much of the statistical analysis that was applicable for linear regression must be modified appropriately. The greatest limitation of nonlinear regression is that the form of the model must be specified a priori. Although this was also a limitation of the linear regression models, it is far more difficult to justify the selection of a model such as that given in Eq. (III.11) as opposed to a simple linear model, unless of course, there is some physical motivation to suppose such a model. For instance, if an analytical solution exists for some simplified analogue to the simulation model, it might be reasonable to assume a nonlinear model form based on some modification of the analytical solution. For our purposes, however, we do not assume such information to be available, and therefore we do not consider nonlinear regression methods further; for more information, a number of standard textbooks can be consulted [74,75].

Yet another regression method that is commonly used makes use of rank transformations. Rather than building a regression model to express the relationship between the input parameters, **x**, and the output, $y$, rank regression uses linear regression to approximate the relationship between the ranks of **x** and $y$. That is, after generating a sample of size $N_S$ of input/output mappings, $[y^{(j)}, \mathbf{x}^{(j)}], j = 1, 2, \ldots, N_S$, each of the $y^{(j)}$ are assigned an integer value between 1 and $N_S$ corresponding to the position of $y^{(j)}$ in a vector consisting of all the $y^{(j)}$ sorted in ascending order. Hence, the smallest $y^{(j)}$ is assigned a rank of 1, and the largest is assigned a rank of $N_S$. Similarly, each of $x_i^{(j)}$ is replaced by its rank. Finally, a linear regression model, such as in Eq. (III.2), is built with the **x** and $y$ replaced by their corresponding ranks. If $y$ depends monotonically on each of the $x_i$, the rank transformation results in a linear relationship [71]; this is true even if the underlying relationship between $y$ and $\mathrm{x_i}$ is nonlinear. Hence, rank regression can provide a more accurate fit than standard linear regression if the output is nonlinear and monotonic. However, if the output is non-monotonic, the performance of rank regression is generally quite poor [71]. Furthermore, rank regression models are not appropriate as predictive models; this is because the rank regression model only provides information regarding the relative ranks of observations of $y$ and provides no information regarding the actual values of $y$.

## III.4 Variance-Based Methods

The SA methods described thus far have been geared towards the identification of behavioral trends in the input/output data; to be more specific, the aforementioned methods were developed with the intention of studying how **x** maps to $y$. Variance-based methods, on the other hand, were designed to study the relationship between the input uncertainty and output uncertainty; that is, these methods focus on the mapping of $p_{\mathrm{X}}(\mathbf{x})$ to $p_{\mathrm{Y}}(y)$, or more precisely, the mapping of $p_{\mathrm{X}}(\mathbf{x})$ to $\mathrm{var}(y)$. Consequently, the variance-based SA methods are more aligned with Saltelli's definition of SA given in Chapter I [11]. The variance-based SA methods are most useful when the model output uncertainty results from epistemic uncertainty in the model inputs; in such cases, it may be useful to know what parameters are driving the uncertainty in the output. Variance-based sensitivity measures provide a ranking of parameters based upon their contribution to the output variance; hence, these measures can be useful for research allocation, by providing indication as to where research efforts should be focused (i.e., what parameters need to be better understood) so as to most effectively reduce the model output variance. Furthermore, variance-based SA methods have an important advantage over many other SA methods in that they are model independent; by this, we mean that the variance-based methods make no assumptions regarding the structure or form of the model. This is in stark contrast to regression methods. Consequently, they are not limited by model complexity and can, in principle, be used for any type of problem. However, as we shall see, these methods are highly demanding computationally, often requiring many evaluations of the model. Although a variety of variance-based importance measures have been proposed [76], the Sobol' indices seem to be the most popular, and these will be discussed in the following section.

### ***III.4.A The Sobol' Sensitivity Indices***

To motivate the discussion of the Sobol' indices, we begin by considering some quantity $y = y(\mathbf{x})$, where $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$. To determine the effect of some input, $x_i$, one seemingly reasonable approach is to simply fix $x_i$ to an arbitrary value and average $y$ over the remaining $m-1$ inputs. Mathematically, we express this as:

$$\mathrm{E}_{-i}(y \mid x_i), \tag{III.12}$$

where the subscript, - $i$, denotes the set of all $\mathbf{x}$ excluding $x_i$. Hence, $\mathrm{E}_{-i}(\,\cdot\,)$ is to be interpreted as the expectation taken over all $\mathbf{x}$ except $x_i$. Oakley and O'Hagan [77] refer to the difference between this quantity and $\mathrm{E}(y)$ as the Main Effect of variable $x_i$. Notice that the main effect is a function of $x_i$. Plots of the main effects for each of the inputs can provide another useful visual indication of the dependencies in the model. To remove the functional dependence on $x_i$ of Eq. (III.12), one approach is to simply take its variance:

$$V_i = \mathrm{var}_i[\mathrm{E}_{-i}(y \mid x_i)]. \tag{III.13}$$

By making use of the following conditional variance identity:

$$\mathrm{var}(y) = \mathrm{var}_i[\mathrm{E}_{-i}(y \mid x_i)] + \mathrm{E}_i[\mathrm{var}_{-i}(y \mid x_i)] \tag{III.14a}$$

or, equivalently,

$$\mathrm{var}(y) = \mathrm{var}_{-i}[\mathrm{E}_i(y \mid \mathbf{x}_{-i})] + \mathrm{E}_{-i}[\mathrm{var}_i(y \mid \mathbf{x}_{-i})]. \tag{III.14b}$$

we can rewrite Eq. (III.13) as:

$$V_i = \mathrm{var}_i[\mathrm{E}_{-i}(y \mid x_i)] = \mathrm{var}(y) - \mathrm{E}_i[\mathrm{var}_{-i}(y \mid x_i)]. \tag{III.15}$$

Notice that $\mathbf{x}_{-i}$ is a vector consisting of the $m$-1 elements of $\mathbf{x}$ excluding $x_i$. From Eq. (III.15), we see that $V_i$ represents the expected reduction in variance that results when $x_i$ is perfectly known [60]. Thus, the ratio of $V_i$ to the total variance of $y$, given by:

$$S_i = \frac{V_i}{V} = \frac{\mathrm{var}_i[\mathrm{E}_{-i}(y \mid x_i)]}{\mathrm{var}(y)}, \tag{III.16}$$

represents the expected fraction of total variance that is reduced when $x_i$ is known. The quantity $S_i$ is variously referred to as the first-order Sobol' index [78], the first-order sensitivity index [69], the main effect index [77], and the correlation ratio [55]. Notice that $0 \le V_i \le V$, so that $S_i$ can take on values between zero and unity. In particular, $S_i = 1$ implies that all of the variance of $\mathbf{y}$ is expected to be reduced if parameter $x_i$ is fixed. On the other hand, $S_i = 0$ indicates that fixing $x_i$, is expected to have no effect on the variance of $y$. This does not necessarily imply that $x_i$ is noninfluential, however, since it may be the case that $x_i$ has a large influence on the variance $y$ through its interactions with another parameter, $x_j$. That is, the expression for $y$ could include interaction terms such as $x_ix_j$ for $j \neq i$. Thus, while a large value of $S_i$ is sufficient to identify a parameter as influential, a small value of $S_i$ is *not* a sufficient indication that a parameter can be neglected.

As an alternative, suppose we fixed every parameter except $x_i$ to some arbitrary value and computed the expectation over $x_i$. The variance of this resulting quantity, $\mathrm{var}_{-i}[\mathrm{E}_i(y \mid \mathbf{x}_{-i})]$, can be interpreted as the expected reduction in variance that results when every parameter except $x_i$ is known, which follows from Eq. (III.14b). Consequently, the difference, $Var(y)-\text{var}_{-i}[\text{E}_i(y\,|\,\mathbf{x}_{-i})]$, is the expected variance that remains after all of the parameters except $x_i$ are known or fixed. This is the variance of $y$ that is attributable to the total variation (including interactions) of parameter $x_i$. With this, we can define the total effect index, $S_i^T$, as [76]:

$$S_i^T = 1 - \frac{\text{var}_{-i}[\text{E}_i(y\,|\,\mathbf{x}_{-i})]}{\text{var}(y)} = \frac{\text{E}_{-i}[\text{var}_i(y\,|\,\mathbf{x}_{-i})]}{\text{var}(y)} \tag{III.17}$$

which provides a measure of the total contribution of $x_i$ to the variance of $y$. Since $S_i^T$ includes variance contributions from interactions between $x_i$ and all other parameters, while $S_i$ only accounts for first-order variance contributions, it should be clear that $S_i^T \geq S_i$ with equality if the model is additive (i.e., no parameter interactions). Furthermore, a small value of $S_i^T$ is sufficient to identify a parameter as noninfluential, at least in terms of variance.

The above arguments can be generalized by appealing to the variance decomposition formula given by Sobol' [76]:

$$V = \sum_{i}^{m} V_i + \sum_{i<j}^{m} V_{i,j} + \sum_{i<j<k}^{m} V_{i,j,k} + \cdots + V_{1,2,\ldots,m}, \tag{III.18}$$

where, $V_i$ is given by Eq. (III.13), $V_{i,j} = \text{var}_{i,j}[\text{E}_{-i,j}(y\,|\,x_i,x_j)] - V_i - V_j$, and so on. Defining the general Sobol' index to be:

$$S_{i_1,i_2,\ldots i_s} = \frac{V_{i_1,i_2,\ldots i_s}}{V}, \tag{III.19}$$

we can rewrite Eq. (III.17) as:

$$1 = \sum_{i}^{m} S_i + \sum_{i<j}^{m} S_{i,j} + \sum_{i<j<k}^{m} S_{i,j,k} + \cdots + S_{1,2,\ldots,m}\,. \tag{III.20}$$

From Eq. (III.19), we see that each of the sensitivity indices, $S_{i_1,i_2,\ldots i_s}$, represents the fraction of total variance that is attributable to interactions between the set of inputs $\{x_{i_1}, x_{i_2}, \ldots x_{i_s}\}$. Furthermore, by summing over all of the sensitivity indices with a particular subscript (e.g. 2), one obtains the total effect index for the corresponding input (e.g., $x_2$).

Ideally, one would compute all of the sensitivity indices in Eq. (III.19) to obtain the most complete representation of the variance sensitivity of the model. However, practical considerations often prohibit this, as there can be as many as $2^m - 1$ different sensitivity indices [79]. Furthermore, each index requires the computation of an expected value embedded within a variance computation, and a naïve approach to computing these indices would require multiple nested Monte Carlo simulations. Fortunately, in practice, it often suffices to calculate the full set of first-order Sobol' indices, $S_i$, together with the full set of total effect indices, $S_i^T$. This is because the quantities $S_i$ and $S_i^T$ are sufficient to determine whether a parameter is influential. Moreover, detailed information regarding the interactions between parameters is often unnecessary. It should be noted, however, that by comparing $S_i^T$ and $S_i$, it is possible to obtain some limited information regarding interactions; specifically, a large difference between the two quantities is an indication that the parameter is interacting strongly with other parameters, but it is not possible to determine, from $S_i^T$ and $S_i$ alone, with what parameters it is interacting. If this is the case, and if it suspected that such interactions are important, additional data can be obtained to compute, say, the second-order indices.

### ***III.4.B The Sobol' Monte Carlo Algorithm***

In the preceding discussion it was noted that the computation of the Sobol' indices requires the estimation of a conditional expectation embedded within a variance estimation. Consequently, a naïve application of MCS with nested Monte Carlo calculations would clearly be infeasible. Fortunately, Sobol' presented an efficient Monte Carlo-based algorithm for approximating the Sobol' indices that has been modified by Saltelli to also provide estimates of the two-way interaction indices, $V_{i,j}$ [79]. We will refer to the modified algorithm as the Sobol'-Saltelli algorithm. Here, we present an overview of the original algorithm and provide formulas for the first-order indices and the total effect indices. Additional details regarding the computation of the two-way interaction indices can be found in [79].

The first step of the Sobol' method is to draw two sets of samples of model input configurations, both of size $N_S$, and to store these samples as matrices **A** and **B**, where:

$$\mathbf{A} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N_S,1} & x_{N_S,2} & \cdots & x_{N_S,m} \end{bmatrix} \text{ and } \mathbf{B} = \begin{bmatrix} x'_{1,1} & x'_{1,2} & \cdots & x'_{1,m} \\ x'_{2,1} & x'_{2,2} & \cdots & x'_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{N_S,1} & x'_{N_S,2} & \cdots & x'_{N_S,m} \end{bmatrix}$$

and $x_{i,j}$ denotes the $i^{\text{th}}$ sample of parameter $x_j$; the primes above each of the elements in matrix **B** are intended to distinguish the two sets of samples. Next, we define a set of $m$ matrices, $\mathbf{C}_j$ , where for each $j = \{1,2,\ldots,m\}$, $\mathbf{C}_j$ is given by the matrix **B** with the $j^{\text{th}}$ column replaced by the $j^{\text{th}}$ column of **A**. The model, $y = g(\mathbf{x})$, is then evaluated for each of $N_S(m+2)$ samples stored in the matrices, **A**, **B**, and $\mathbf{C}_j$. By letting $\mathbf{Y}_\mathbf{A} = g(\mathbf{A})$, $\mathbf{Y}_\mathbf{B} = g(\mathbf{B})$, and $\mathbf{Y}_{\mathbf{C}_j} = g(\mathbf{C}_j)$ denote the $N_S \times 1$ vectors corresponding to the model output for the input samples contained in **A**, **B**, and $\mathbf{C}_j$, respectively, the first-order Sobol' indices can then be estimated as:

$$S_j = \frac{N_S \mathbf{Y}_\mathbf{A}^\mathbf{T} \mathbf{Y}_{\mathbf{C}_j} - (\mathbf{Y}_\mathbf{A}^\mathbf{T}\mathbf{1})^2}{N_S \mathbf{Y}_\mathbf{A}^\mathbf{T} \mathbf{Y}_\mathbf{A} - (\mathbf{Y}_\mathbf{A}^\mathbf{T}\mathbf{1})^2} \tag{III.21}$$

where $\mathbf{1}$ denotes the $N_S \times 1$ vector of ones and superscript $^\mathbf{T}$ denotes the transpose. Furthermore, the total sensitivity indices can be computed as:

$$S_j^T = 1 - \frac{N_S \mathbf{Y}_\mathbf{B}^\mathbf{T} \mathbf{Y}_{\mathbf{C}_j} - (\mathbf{Y}_\mathbf{A}^\mathbf{T}\mathbf{1})^2}{N_S \mathbf{Y}_\mathbf{A}^\mathbf{T} \mathbf{Y}_\mathbf{A} - (\mathbf{Y}_\mathbf{A}^\mathbf{T}\mathbf{1})^2} \tag{III.22}$$

As compared to a brute-force nested MC approach, the Sobol' method, requiring $N_S(m+2)$ model evaluations, enjoys far greater computational efficiency. However, in this expression, the quantity, $N_S$, is somewhat arbitrary and we are not aware of any definitive guidelines for choosing its value. In his presentation of the algorithm discussed above, Saltelli uses a value of approximately 1000 for the case studies that he considers [79]. While it may be possible to reduce this number to a degree, it seems unlikely that $N_S$ can be reduced below a few hundred for reasonable results. Consequently, we are once again led to the conclusion that a fast-running surrogate model is necessary when each simulation requires several hours.

## III.5 Fourier Amplitude Sensitivity Test (FAST)

As its name suggest, FAST makes use of Fourier analysis to express variations in the model output in terms of harmonics on the model input parameters. The importance of the different input parameters can then be determined by analyzing the strength of the various frequency components that are present in the output 'signal.' The mathematics of this process can be somewhat complex, so we shall forego a formal presentation and simply note that the details of the FAST method are given by Saltelli et al. [80]. Fortunately, the basic idea underlying the FAST methodology is rather intuitive and easily understood. That being said, however, most of the literature regarding FAST tends to delve into the mathematics without giving much insight into what FAST is actually doing, so we shall spend a moment to elaborate.

Suppose that, rather than randomly sampling input configurations as with MCS, we were to draw inputs in a structured fashion; specifically, we sequentially draw inputs in such way that each parameter oscillates between its respective high and low values with a specified frequency that is unique to that parameter. In practice, this is done by parameterizing the input variables by a scalar, $s \in (-\infty, \infty)$, so that the curve traced by the vector, $\mathbf{x}(s) = \{x_1(s), x_2(s), \ldots x_m(s)\}$ fills the entire input space. Such curves are very similar to multidimensional analogues of Lissajous curves, and the space filling property is satisfied if the frequencies assigned to each of the inputs are incommensurate (a technical term meaning that they are linearly independent). If the frequencies were linearly dependent, then the curve would be periodic and would therefore not fill the entire input space. After properly drawing a set of inputs by following this curve throughout space, we should expect that the corresponding model outputs will exhibit some type of oscillatory behavior. Thus, it seems natural to conclude that the frequency of this oscillation should provide an indication as to which parameter(s) is (are) responsible. In particular, if the model output oscillates at a frequency corresponding to a harmonic of one of the input parameters, then it seems reasonable to deduce that this parameter is causing the oscillatory behavior. Indeed, this is where Fourier analysis is helpful; by decomposing the output signal into its frequency spectrum, we can assess the importance of, say, the input $x_i$, by observing the strength (spectral amplitude) of the spectrum at the frequency for $x_i$.

Saltelli and Bolado [81] showed that the FAST sensitivity indices are equivalent to the first-order Sobol' indices, and Saltelli et al. [80] extended the classic FAST methodology to compute the total-order sensitivity indices. Hence, we view FAST as an alternative to the Sobol' algorithm discussed in the previous section. As far as we know, neither method is clearly superior to the other. It seems that in cases where there are few input parameters, FAST can be made to perform more efficiently, whereas in cases with many input parameters the Sobol' algorithm would outperform FAST. Furthermore, FAST can suffer from bias problems [80], and Saltelli's extension to the Sobol' algorithm allows one to compute second-order interaction effects using essentially the same samples used to compute the first-order and total-effects sensitivity indices [79]. In any case, for our purposes, the question of which method is superior is largely irrelevant; this is because neither method seems capable of reducing the number of required model evaluations to below approximately 100. Hence, we shall be forced to consider the use of metamodels to reduce the computational demand, in which case the efficiency of simulation procedure for SA and UA will be a non-issue.

## III.6 Additional Methods and Factor Screening

It would be impossible to provide a complete review of SA methods in anything short of a multivolume encyclopedia, and the methods that we have presented thus far fall short of representing the vastness and diversity of the field of sensitivity analysis. We have already noted that there exists an entirely different class of SA tools, namely the deterministic SA methods, that are neglected in this review. Additional methods that deserve to be mentioned include grid-based analyses, which utilize entropy measures for identifying patterns in scatter plots in a more or less autonomous fashion [8,30]. Furthermore, Zio and Pedroni demonstrate how the results from Line Sampling [44] and Subset Simulation [48] can be directly used for SA and parameter ranking. Finally, in some applications, the number of input parameters for the model of interest can be so large (i.e., ~100 or more) that the development of regression models or metamodels is impractical without first reducing the number of parameters to more tractable levels. In these cases, it is necessary to determine, using as few model evaluations as possible, the set of input parameters that can be removed from further consideration. This problem is referred to as screening, and a variety of techniques have been proposed for effective and efficient parameter screening. The simplest of these techniques are classical One-At-a-Time (OAT) experiments [82,83]. Morris [84] presented an alternative OAT-type procedure, called the Elementary Effects method that was developed specifically for computer experiments and was recently revised by

Campolongo et al. [85]. In addition, a technique known as Controlled Sequential Bifurcation has recently been proposed by Wan et al. [86] as an advancement on the classical Sequential Bifurcation method [87,88], and even more recently, Shen and Wan have proposed the Controlled Sequential Factorial Design method [89]. Clearly, factor screening is currently an area of active research, particularly in the field of operations research. However, we have restricted our attention to models that, although extremely computationally time consuming, depend on relatively few input parameters, so that factor screening is not necessary.

# IV METAMODELING

We have stated time and again that probabilistic UA and SA of complex simulation models is particularly troublesome due to the large number of model evaluations required coupled with the often excessive run-time of these models. It is not uncommon for these computer models to require several hours, or even days, to perform a single simulation (see Table 1 for an approximate range of run-times to be expected for various codes that are commonly used for nuclear reactor safety analyses). In these cases, it is clearly impractical, if not impossible, to perform a standard Monte Carlo uncertainty analysis that requires, say, 10,000 model evaluations. Moreover, performing reliability assessments of safety-critical systems often requires upwards of a million model evaluations to estimate failure probabilities on the order of $10^{-4}$ with high accuracy. Although some advanced sampling methods were discussed in Chapter II that can reduce the required sample size to more tractable levels on the order of a few hundred samples, there are instances where even this approach presents an infeasible computational burden. Consequently, the only alternative seems to be the use of metamodels, or surrogate models.

**Table 1. Approximate Range of Runtimes for Typical Codes Used for Reactor Safety Assessment**

| Code | Application | Approximate range of typical simulation run-time |
|---|---|---|
| MACCS | Offsite Consequence Analysis | Tens of minutes to a few hours |
| MELCOR | Severe Accident Analysis | Tens of hours to a few days |
| TRACE | Design-Basis Accidents | One day to a few days |
| Fluent | Computational Fluid Dynamics | A few days to a few weeks |

So, what is a metamodel? To answer this question, it is first necessary to consider what a model is. To put it simply, a model is an inference tool designed to address the following question: given a finite set of data, how can one use this information to make general predictions regarding the behavior of similar, though not identical, systems? In the same vein, metamodels are tools for predicting the output from a complex simulation model based on an often limited set of observations from this model in the form of input/output mappings (alternatively referred to as I/O patterns). Hence, the term 'metamodel' is aptly chosen to denote that such a construction is a model of a model. Alternatively, the term surrogate model is occasionally used, the reason being that in practical applications, the metamodel serves as a quick-running surrogate to the actual model being analyzed so that multiple approximate simulations can be obtained with negligible computational demand.

As Storlie et al. point out, the use of metamodels for performing UA and SA of complex simulation models is advantageous due to their efficient use of computational resources [15]. In particular, every observation from the simulation model brings new information to the table, and metamodels are designed to utilize this information to make predictions. This is in stark contrast to standard MCS methods, which assume that each realization from the model is an independent event. Thus, by assumption, no data already obtained from the model can be used for making inferences regarding future realizations.

Although numerous metamodeling methods have been used for decades in a variety of engineering disciplines, as we shall see in the following, a review of the literature reveals a recent surge in the popularity of these methods. While each field surely has its own reasons, the growing interest in metamodeling from the UA and SA communities seems to be predicated on the belief that it is better to perform complete, albeit approximate, uncertainty and sensitivity analyses with a metamodel than to perform incomplete analyses. For instance, if standard MCS is used to estimate a failure probability of, say $10^{-4}$, but only 1000 samples can be afforded, the results of this analysis will be of little use to the analyst. That is, of course, unless the information provided by these samples is somehow incorporated into a subsequent analysis (i.e., by being used to construct a metamodel).

Some of the earliest applications of metamodels include the estimation of radiation dose from the ingestion of $^{90}$Sr [90], the calculation of ignition delay times in methane combustion [91,92], as well as the simulation of the compression molding of an automobile hood [92]. In addition, Brandyberry and Apostolakis used response surfaces as an approximation for a fire risk analysis code [93]. More recently, response surfaces and Gaussian Process (GP) models have been used extensively for UA and SA of radionuclide transport models [13,78,94-97], as well as for design optimization problems [98,99]. Some specific examples from the aerospace industry include the use of quadratic RSs and GPs for an Earth-Mars transfer orbit design and a satellite constellation design [100], and the use of Treed-GPs for the simulation and design of a rocket booster under atmospheric re-entry conditions [101]. Additional applications of metamodeling have included hydrogeological modeling [102], oil reservoir simulation [103], forecasting for an active hydrocarbon reservoir [104], the design of a sewage overflow system [105], and simulations of nuclear fuel irradiation [106], to name a few. In addition, Kennedy et al. [107] describe several case studies where GP models were used for approximating atmospheric and climate simulation codes. Perhaps most relevant to this work have been the countless applications to reliability assessment. In particular, Fong et al. [31] utilize quadratic response surfaces to estimate the reliability of a passively cooled nuclear safety system. Similarly, Burrati et al. [108], Gavin and Yau [109], and Liel et al., [110] have applied response surfaces of various forms to numerous problems in structural reliability, including seismic reliability assessment. Finally, Deng [111], Cardoso et al. [112], Cheng et al. [113] and Hurtado [114] have applied various statistical learning models, such as Artificial Neural Networks (ANN), Radial Basis Functions (RBF), and Support Vector Machines (SVM) to multiple structural reliability problems, and Bucher and Most [115] provide a comparative evaluation of polynomial response surfaces and ANNs with an application to structural reliability.

The discussion above should provide some indication regarding the diversity of metamodeling techniques. We have already mentioned tools such as polynomial RSs, GP models, ANNs, RBFs, and SVMs. A recent review by Storlie et al. [15] provides a comprehensive comparison of quadratic RS (QRS)[†] and GP models, as well as various alternative methods, including Multivariate Adaptive Regression Splines (MARS), Random Forests (RF), Generalized Additive Models (GAM), Gradient Boosting Machines (GBM), Recursive Partitioning (RPART) regression, and Adaptive COmponent Selection and Shrinkage Operator (ACOSSO). For their analysis, Storlie et al. [15] applied each of the aforementioned methods to a series of three analytical test functions with the objective of estimating the total sensitivity indices for the input parameters. While a thorough review of each of these methods is beyond the scope of this thesis, we can summarize some of the main conclusions of Storlie et al. In particular, the authors found that, despite their good performance in certain cases, the overall performance of RPART, RF, and GBM was rather inconsistent, and consequently, these methods were not recommended for general metamodel use [15]. On the other hand, the authors note that

[†] Note that Storlie et al. actually refer to quadratic regression, abbreviate as QREG in their paper, rather than quadratic RSs. However, as we discuss later, QREG and QRS are mechanically (i.e., mathematically) identical. We use the term QRS to be consistent with subsequent discussions.

QRS and MARS look to be very attractive practical options for reasons including the automatic variable screening capability of MARS and the ease of interpretation of QRSAdditional desirable features of MARS and QRS are that both are easily computed and provide an extremely computationally efficient metamodel. ACOSSO and GP were found to perform consistently well in all cases considered, but suffer from a larger computational overhead as compared to MARS and RS. Furthermore, the lack of an inherent variable selection capability with GP models was noted as one of the primary disadvantages of this method; however, the authors acknowledge and provide references for some recent work regarding variable selection for GP models. As final suggestion, Storlie et al. [15] recommend the use of quadratic RS, MARS, ACOSSO and GP models for practical SA applications.

In the following sections, we present a general framework for the metamodeling problem, followed by a detailed discussion regarding two of the metamodeling techniques recommended by Storlie et al. [15]: namely, polynomial Response Surfaces and Gaussian process models. In addition, we have used ANNs for one of the case studies in Chapter V, so we provide a brief discussion of this metamodeling method; this discussion is primarily intended to provide some context for those who are unfamiliar with ANNs and we do not discuss most of the extensive theory underlying the mechanics of ANNs. The interested reader is encouraged to consult any of a number of excellent textbooks on the topic, including that by Bishop [116]. Alternatively, another excellent resource is the user's manual for the MATLAB® Neural Network Toolbox™, freely available online from The MathWorks™, Inc. [117]. Finally, we conclude this chapter by discussing some approaches to accounting for the uncertainty in the predictions obtained from the metamodels - the metamodel uncertainty, as defined in Chapter II. Once again, the several of the main conclusions from this chapter have been summarized in Appendix C.

## IV.1 A General Metamodeling Framework

As mentioned in the opening paragraphs of this chapter, metamodels are intended to make predictions of the output of a complex simulation model using the information obtained from a limited set of observations. We shall denote this data as the set $\{y_D^{(i)}, \mathbf{x}_D^{(i)}\}$, $i=1,2,\ldots,N_D$, where $N_D$ is the number of available data and each of the $\mathbf{x}_D^{(i)}$ is an $m$-vector, with $m$ being the number of inputs to the model. The subscript $D$ is intended to emphasize that this data is the design data used for building the metamodel. For UA and SA purposes, we are generally interested in using the metamodel to make predictions of the output, $y_S^{(i)}$, for a set of sampled input configurations, $\mathbf{x}_S^{(i)}$, $i=1,2,\ldots,N_S$.

Although we have assumed throughout this thesis that the output from the simulation model is a single scalar quantity ($y = g(\mathbf{x})$), we must reconsider this assumption in the context of metamodeling. In the general case of a vector output, each of the components of **y** are likely to be correlated, and strictly speaking, this behavior should be taken into account when making simultaneous predictions for multiple components of **y**. However, this added complexity is usually neglected in practice, presumably because most metamodeling methods are not equipped to deal with multiple correlated outputs, and one simply builds a separate metamodel for each output. We should note that ANNs are an exception to this rule, as they are inherently well suited for handling multiple outputs. Most other metamodels require some augmentation to treat multiple outputs, and for the case of GP metamodels, Le and Zidek [118] provide the relevant details for this extension.

In the following sections, we shall assume that the simulation model is deterministic. That is, if the model is run multiple times for a particular input configuration, each simulation will yield the same output. Since most simulation models satisfy this criterion, the assumption of determinism is not too restrictive. For cases where a stochastic model needs to be considered, it may be possible to enforce determinism by adding as an additional input the seed used for the random number generation.

For our purposes, we can regard the metamodel as a function of the form:

$$\hat{y} = r(\mathbf{x},\boldsymbol{\beta}) + \delta(\mathbf{x},\boldsymbol{\theta}) \tag{IV.1}$$

where, as before, $\hat{y}$ denotes an approximation to the true model output, $y$, and $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ are vectors of metamodel parameters that are estimated from the data. The reason for expressing the metamodel as a sum of two functions in Eq. (IV.1) instead of a single function is that each of these functions is often given a different interpretation; namely, $r(\mathbf{x},\boldsymbol{\beta})$ is interpreted as a regression-type model that describes the global trend of the model and $\delta(\mathbf{x},\boldsymbol{\theta})$ represents local variations, or deviations, from this global trend.

Both the RS and GP models that we shall consider in this chapter are linear regression models, which allows us to simplify the regression term as $r(\mathbf{x},\boldsymbol{\beta}) = \mathbf{f}^{\mathrm{T}}(\mathbf{x})\boldsymbol{\beta}$. However, each method differs in its treatment and interpretation of the second term in Eq. (IV.1), as will be discussed in the following sections. At this point, we simply note that for a RS, $\delta(\mathbf{x},\boldsymbol{\theta})$ is replaced by a error term, $\varepsilon$, to account for differences between the regression model and the true output; consequently, the form of the RS is completely specified by the function **f(x)**, making this method a type of parametric regression. On the other hand, GP models are nonparametric regressors since the functional form of $\delta(\mathbf{x},\boldsymbol{\theta})$ is not specified a priori. ANNs can be considered a type of nonlinear regression, and therefore do not admit the aforementioned simplification.

## IV.2 Polynomial Response Surfaces (RS)

The Response Surface Methodology (RSM) was developed in the 1950's as a tool for identifying optimal experimental configurations for physical experiments by providing a convenient method for understanding the approximate relationship between a response variable (i.e., the output from the experiment) and set of predictor variables (i.e., the parameters that are controlled by the experimenter) [120]. Having proven itself to be a valuable resource for experimenters in all fields, RSM has more recently been adopted by the computational experimental community as an efficient and simple, yet often effective, tool for metamodeling. Currently, linear (LRS) and quadratic (QRS) response surfaces are two of the most commonly used metamodels in practice. Below, we provide the relevant mathematical details for both LRS and QRS, and more generally the polynomial RS.

### *IV.2.A Mathematical Formulation*

As discussed in the previous section, RSs are an example of a parametric linear regression model, thereby making them mechanically equivalent to the regression models discussed in Section III.3; that is, these models are based on the assumption that the model output, $y$, can be expressed as:

$$y = \hat{y} + \varepsilon = \mathbf{f}^{\mathrm{T}}(\mathbf{x})\boldsymbol{\beta} + \varepsilon \tag{IV.2}$$

where, as before, $\hat{y}$ represents the approximation (i.e., the RS metamodel), $\varepsilon$ represents a zero-mean random error, $\boldsymbol{\beta}$ is a $q \times 1$ vector of unknown coefficients, and $\mathbf{f}^{\mathrm{T}}(\mathbf{x})$ is a $1 \times q$ vector representing the polynomial form for **x**. Hence, for a linear RS, $\mathbf{f(x)} = \{1, x_1, x_2, \ldots, x_m\}^{\mathrm{T}}$, whereas for a quadratic RS, $\mathbf{f(x)} = \{1, x_1, x_2, \ldots, x_1 x_2, x_1 x_3, \ldots x_m^2\}^{\mathrm{T}}$. Notice that we are using a slightly modified notation from that in Section III.3; this is to maintain a degree of consistency with the RS literature. With the assumed model given by Eq. (IV.2), we can represent this relationship for each of the $N_D$ design points in the following convenient matrix form:

$$\mathbf{y}_D = \mathbf{F}_D \boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{IV.3}$$

where $\mathbf{y}_D = \left\{ g(\mathbf{x}^{(1)}), g(\mathbf{x}^{(2)}), \ldots, g(\mathbf{x}^{(N_D)}) \right\}^{\mathrm{T}}$ and the vector $\boldsymbol{\varepsilon} = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{N_D}\}^{\mathrm{T}}$ is the vector of residuals. The matrix $\mathbf{F}_D$ is called the design matrix, and for a QRS is given by:

$$\mathbf{F}_D = \begin{Bmatrix} \mathbf{f}^{\mathrm{T}}(\mathbf{x}_D^{(1)}) \\ \vdots \\ \mathbf{f}^{\mathrm{T}}(\mathbf{x}_D^{(N_D)}) \end{Bmatrix} = \begin{bmatrix} 1 & x_{D,1}^{(1)} & x_{D,2}^{(1)} & \cdots & x_{D,k}^{(1)} x_{D,l}^{(1)} & \cdots & (x_{D,m}^{(1)})^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & x_{D,1}^{(N_D)} & x_{D,2}^{(N_D)} & \cdots & x_{D,k}^{(N_D)} x_{D,l}^{(N_D)} & \cdots & (x_{D,m}^{(N_D)})^2 \end{bmatrix}. \tag{IV.4}$$

Analogously, we can define the sample matrix, $\mathbf{F}_S$. However, to make predictions for $\mathbf{y}_S$, it is first necessary to use Eq. (IV.3) to estimate the unknown regression coefficients, $\boldsymbol{\beta}$. The typical approach is to use linear least-squares to minimize the residual sum-of-squares, $SS_{res} = \boldsymbol{\varepsilon}^{\mathrm{T}} \boldsymbol{\varepsilon}$, with the resulting estimate given by [121]:

$$\hat{\boldsymbol{\beta}} = (\mathbf{F}_D^{\mathrm{T}} \mathbf{F}_D)^{-1} \mathbf{F}_D^{\mathrm{T}} \mathbf{y}_D \tag{IV.5}$$

Using this result, the predictor for the sampled input configurations is:

$$\hat{\mathbf{y}}_S = \mathbf{F}_S \hat{\boldsymbol{\beta}} = \mathbf{F}_S (\mathbf{F}_D^{\mathrm{T}} \mathbf{F}_D)^{-1} \mathbf{F}_D^{\mathrm{T}} \mathbf{y}_D . \tag{IV.6}$$

We provide an analogous expression for the predictor for the design points as:

$$\hat{\mathbf{y}}_D = \mathbf{F}_D \hat{\boldsymbol{\beta}} = \mathbf{F}_D (\mathbf{F}_D^{\mathrm{T}} \mathbf{F}_D)^{-1} \mathbf{F}_D^{\mathrm{T}} \mathbf{y}_D = \hat{\mathbf{H}} \mathbf{y}_D , \tag{IV.7}$$

where the $N_D \times N_D$ matrix, $\hat{\mathbf{H}} = \mathbf{F}_D(\mathbf{F}_D^{\mathrm{T}}\mathbf{F}_D)^{-1}\mathbf{F}_D^{\mathrm{T}}$, is called the hat matrix [121]. Notice that, in general, $\hat{\mathbf{H}} \neq \mathbf{I}$, were $\mathbf{I}$ is the $N_D \times N_D$ identity matrix; consequently, we have that $\mathbf{y}_D \neq \hat{\mathbf{y}}_D$, and the RS does not exactly interpolate the data. If $\mathbf{F}_D$ is invertible, however, then one can show that $\hat{\mathbf{H}} = \mathbf{I}$ and the RS will exactly interpolate the data $(\mathbf{y}_D = \hat{\mathbf{y}}_D)$. This corresponds to the case when there are exactly as many data points as there are unknown coefficients (i.e., $\mathbf{F}_D$ is square) so that exactly one RS of the assumed form (i.e., linear or quadratic) interpolates the data. More generally, however, Eq. (IV.3) represents an overdetermined system so that the above inequality holds.

Combining Eqs. (IV. 3) and (IV.7), we obtain the following expression for the vector of residuals:

$$\boldsymbol{\varepsilon} = \mathbf{y}_D - \hat{\mathbf{y}}_D = \mathbf{y}_D - \hat{\mathbf{H}}\mathbf{y}_D = (\mathbf{I} - \hat{\mathbf{H}})\mathbf{y}_D \,. \tag{IV.8}$$

Notice that the inner product of the residuals, $\boldsymbol{\varepsilon}$, and the RS predictions, $\hat{\mathbf{y}}_D$, can be expressed as:

$$\hat{\mathbf{y}}_D^{\mathrm{T}}\boldsymbol{\varepsilon} = (\hat{\mathbf{H}}\mathbf{y}_D)^{\mathrm{T}}(\mathbf{I} - \hat{\mathbf{H}})\mathbf{y}_D = \mathbf{y}_D^{\mathrm{T}}\hat{\mathbf{H}}^{\mathrm{T}}(\mathbf{I} - \hat{\mathbf{H}})\mathbf{y}_D = \mathbf{y}_D^{\mathrm{T}}(\hat{\mathbf{H}}^{\mathrm{T}} - \hat{\mathbf{H}}^{\mathrm{T}}\hat{\mathbf{H}})\mathbf{y}_D. \tag{IV.9}$$

Additional insight can be gained by considering the properties of the hat matrix in more detail. In particular, notice that $\hat{\mathbf{H}}$ is symmetric idempotent, meaning that $\hat{\mathbf{H}} = \hat{\mathbf{H}}^{\mathrm{T}}$ and $\hat{\mathbf{H}} = \hat{\mathbf{H}}\hat{\mathbf{H}}$. Consequently, Eq. (IV.9) simplifies to:

$$\hat{\mathbf{y}}_D^{\mathrm{T}}\boldsymbol{\varepsilon} = \mathbf{y}_D^{\mathrm{T}}(\hat{\mathbf{H}}^{\mathrm{T}} - \hat{\mathbf{H}}^{\mathrm{T}}\hat{\mathbf{H}})\mathbf{y}_D = \mathbf{y}_D^{\mathrm{T}}(\hat{\mathbf{H}} - \hat{\mathbf{H}})\mathbf{y}_D = 0\,, \tag{IV.10}$$

demonstrating the important fact that the residuals are simply the components of $\mathbf{y}_D$ that are perpendicular to the RS. More specifically, the residuals are orthogonal to the space spanned by the design matrix, $\mathbf{F}_D$. In fact, we can recognize the matrices $\hat{\mathbf{H}}$ and $\mathbf{I} - \hat{\mathbf{H}}$ as orthogonal projection matrices, with the operations $\hat{\mathbf{H}}\mathbf{y}_D$ and $(\mathbf{I} - \hat{\mathbf{H}})\mathbf{y}_D$ projecting the vector, $\mathbf{y}_D$, onto two orthogonal spaces.

### *IV.2.B Goodness-of-Fit and Predictive Power*

Once the RS model has been built, one must consider various diagnostic questions, such as how well does the model fit the data and how well can the model predict new outputs. The first of these two questions is easier to answer, so we shall address it first. The property of the RS that we wish to ascertain is referred to as goodness-of-fit.

The first goodness-of-fit measure that we will consider is the Root Mean Square Error (RMSE), whose meaning is clear from its name and can be computed from:

$$\text{RMSE} = \left[ \sum_{i=1}^{N_D} \frac{(y_D^{(i)} - \hat{y}_D^{(i)})^2}{N_D} \right]^{1/2} = \sqrt{\frac{SS_{res}}{N_D}} \tag{IV.11}$$

where $SS_{res}$ is the residual sum of squares defined in Chapter III. Clearly, the better the model fits the data the lower the RMSE, and so this quantity provides some indication as to the quality of the RS. Alternative quantities generally require some additional assumptions to be made regarding the distribution of the residuals. Having previously stated that $\varepsilon$ is a zero-mean random variable, we shall now impose the additional assumption that each $\varepsilon_i$ in the vector $\boldsymbol{\varepsilon}$ be uncorrelated random variables with constant variance, $\sigma^2$.

We can obtain an estimate for the residual variance, $\sigma^2$, as well as the output variance, $\text{var}[\mathbf{y}_D]$, from:

$$\sigma^2 \approx \hat{\sigma}_B^2 = \frac{SS_{res}}{N_D - 1} \tag{IV.12a}$$

$$\text{var}[\mathbf{y}_D] \approx \frac{SS_{tot}}{N_D - 1} \tag{IV.12b}$$

where $SS_{tot}$ is the total sum-of-squares defined in Eq. (III.9c). The subscript $B$ appearing in Eq. (IV.12a) is intended to indicate that the residual variance estimated from this equation is biased [121]; this will be discussed momentarily. The ratio of Eqs. (IV.12a) and (IV.12b) represents the fraction of the total response variance that is not accounted for by the response surface:

$$\frac{\hat{\sigma}_B^2}{\text{var}[y]} = \frac{SS_{res}/(N_D - 1)}{SS_{tot}/(N_D - 1)} = \frac{SS_{res}}{SS_{tot}} = 1 - R^2 \tag{IV.13}$$

where $R^2$ is the coefficient of determination defined in Chapter III.

We mentioned previously that $\hat{\sigma}_B^2$ is a biased estimate for the residual variance. Since there are $q$ RS coefficients to be estimated from the data, there remain only $N_D - q$ degrees of freedom for estimating the residual variance. Hence, an unbiased estimate for the residual variance is given by:

$$\hat{\sigma}^2 = \frac{SS_{res}}{N_D - q} \tag{IV.14}$$

By analogy with the definition of $R^2$ from Eq. (III.8), we can use the unbiased estimate to define the so-called Adjusted-$R^2$ [121]:

$$R^2_{adj} = 1 - \frac{SS_{res}/(N_D - q)}{SS_{tot}/(N_D - 1)} = 1 - \frac{N_D - 1}{N_D - q}\frac{SS_{res}}{SS_{tot}} = 1 - \frac{N_D - 1}{N_D - q}(1 - R^2) \tag{IV.15}$$

As discussed in Chapter III, $R^2$ provides a measure of the goodness-of-fit of a regression model, or the RS in the present case. One arguably undesirable feature of $R^2$ is that it always increases as more parameters are introduced to the model, or as the order of the RS is increased [121]. Consequently, if a RS model were selected solely on the basis of $R^2$, one would always be led to choose the most complex RS that contains the most terms. On the contrary, $R^2_{adj}$ penalizes the introduction of extraneous parameters that, if added to the RS, would not decrease the response variance by a statistically significant amount. Nevertheless, $R^2$ seems to remain the measure of choice for assessing goodness-of-fit.

Although the measures just given provide useful diagnostic information regarding the degree to which the model fits the data, they turn out to be far less useful for assessing RS quality overall. In particular, they do not, at least as we have defined them thus far, provide any indication of how well the RS can predict the outputs for input configurations not included in the design data. This should not be surprising since the aforementioned measures were based on the residual sum-of-squares. This is precisely the quantity that was minimized when building the RS, so it is only natural that these measures provide an overly optimistic indication of the RS's quality. This brings us to the topics of overfitting and underfitting, which, as the names suggest, refer to two phenomena that occur when a model either fits the data too well, so to speak, or not well enough. These are very important concepts so we shall elaborate further.

It seems natural to believe that, if model A fits the data better than model B, then A must be better than B. Yet, this overfitting phenomenon suggests otherwise. Overfitting can be easily understood in the context of building regression models for data that include random variations.

For example, the true underlying model could be linear, but the data may appear to include higher order variations due to the effect of the random variability, or noise. Hence, a high-order model fit to the data would, in fact, fit the data too well because the high-order effects are simply artifacts of the noise. The results of this would be two-fold; first, the estimated variance of the noise would be smaller than it actually is because some of this variability has been captured by the regression model. Secondly, if the regression model were used to make predictions, these predictions would be biased since they would be based on the incorrect high-order model; that is, a quadratic model would be used for predicting responses from an underlying linear model. The use of $R^2_{adj}$ provides a partial remedy to this problem by penalizing the introduction of insignificant, high-order terms.

Figure 3 attempts to illustrate the concept of overfitting. In this illustration, the solid line represents the true mean of the underlying model, $y = 2x + \varepsilon$, where $\varepsilon$ is a random error term with standard deviation of unity. The five circles are a sample of observations from the model, corrupted by noise, and the dot-dashed and dashed lines represent, respectively, the fitted linear and quadratic regression models. Furthermore, the shaded regions correspond to ±2 residual standard deviations. The sampled data seem to exhibit a clear quadratic tendency, which is captured by the quadratic regression model. However, since the underlying model is actually linear, the quadratic model is biased near the endpoints ($x = \pm 1$) and the center ($x = 0$). In addition, because the quadratic model overfits the data, the variance of the random error is underestimated (the shaded region surrounding the dashed line is too narrow). On the other hand, the linear regression model (dot-dashed line) agrees well with the true mean (solid line), and the estimated variance more closely represents the true variance (it is still an underestimate, but less so than that given by the quadratic model).

RSM attempts to prevent overfitting by starting with a low-order (i.e., linear) regression model, and only adding higher order terms (i.e., interaction and/or quadratic terms) if there is sufficient evidence, in the statistical significance sense, to suggest that the apparent high-order effects are unlikely to be the effect of statistical noise. Consequently, we can see how the use of low-order regression models is justified when dealing with noisy experimental data since one does not often have enough data to adequately indicate that the apparent high-order effects are not the result of noise. On the contrary, the data from deterministic simulation models do not contain random noise, so the best metamodel should be that which best fits the data; that is, there is no fear of overfitting the model to spurious random data. However, the failure of a RS to exactly interpolate the data from a deterministic simulation is conclusive proof that the true model form is different from that assumed by the RS; in other words, in metamodeling one encounters the opposite problem – that of underfitting the data. Regardless, the effects of underfitting and overfitting for prediction are identical; in both instances, one is attempting to predict responses from some model (say model A), using another model (model B) that is inconsistent with model A. As a result, the predictions are biased estimates for the true responses. These issues will be addressed further in Section IV.2.D when we discuss some of the criticisms of using RSs for metamodeling that have been raised in the literature.

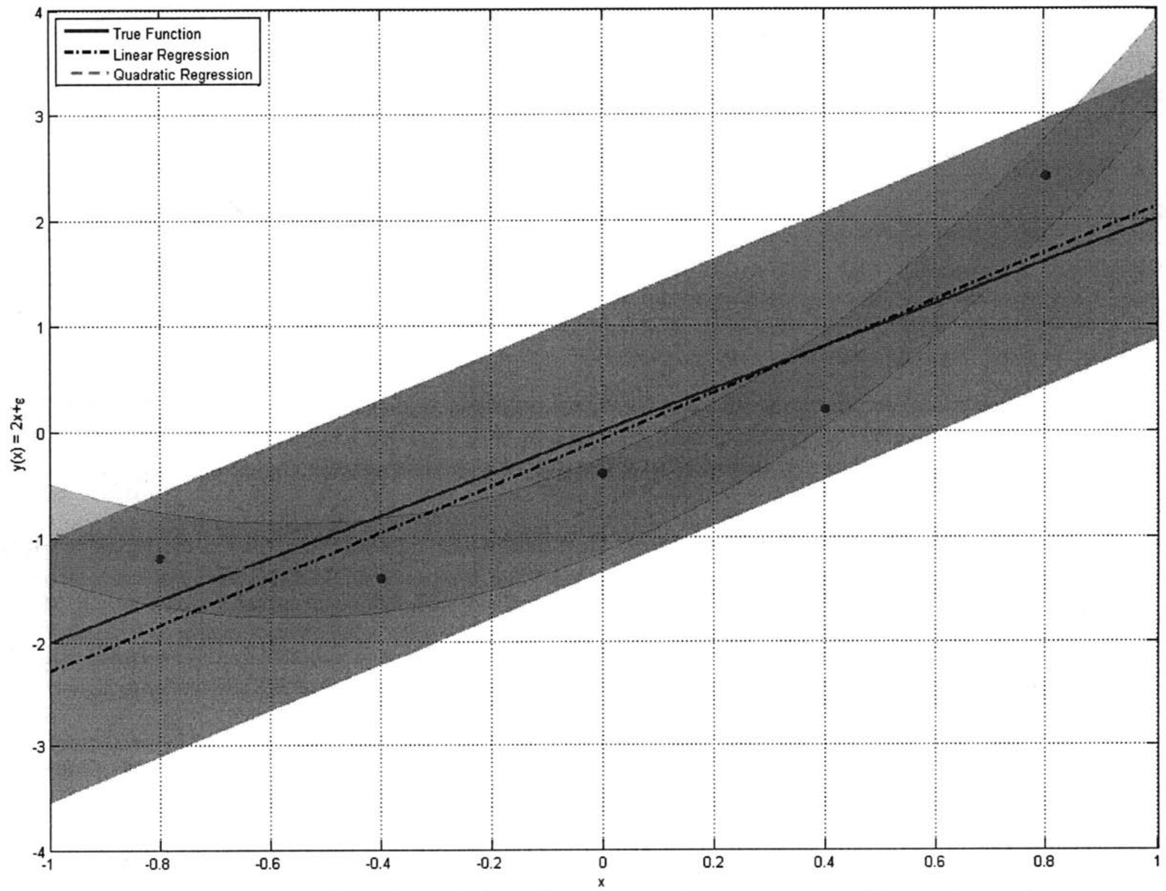

Figure 3: Illustration of overfitting when observations are corrupted by random noise.

Because good predictive capability is what we desire of the RS for UA and SA, we must consider alternative measures for assessing this quality. Perhaps the most straightforward for assessing predictive power is to compute the same measures given above (e.g., RMSE, $R^2$, and $R^2_{adj}$), except using data different from those used to build the response surface. We refer to such data as the validation set since its sole purpose is for validating the metamodel and cannot be used during the construction stage. In the following sections, it will be clear from the context whether the above measures have been computed using the design data or the validation data.

An additional measure that is useful for determining how well a RS can predict new data is the PRediction Error Sum of Squares (PRESS), or the Prediction REsidual Sum of Squares, defined as [121]:

$$\text{PRESS} = \sum_{i=1}^{N_D} \left( \frac{\varepsilon_i}{1 - h_{ii}} \right)^2 \tag{IV.16}$$

where $h_{ii}$ is the $i^{\text{th}}$ diagonal term in the hat matrix. To understand this quantity, consider each of the $N_D$ different data sets that can be formed by selecting $N_D$-1 points from a set of $N_D$ data without replacement. We can designate these design sets as $D_i$, $i$=1,2,...$N_D$. Next, consider constructing $N_D$ different RSs, where each is constructed from one of the design sets $D_i$, and use each of these RSs to predict the response for the data point that was not included in the design set $D_i$. We designate the difference between the predicted response and the true response as $\rho_i$, which we call the prediction residual. If $\varepsilon_i$ is the $i^{\text{th}}$ residual for the RS built from all $N_D$ data, and if $h_{ii}$ is the $i^{\text{th}}$ diagonal term in the corresponding hat matrix, it can be shown that $\rho_i = \varepsilon_i /(1 - h_{ii})$. Hence, PRESS given by Eq. (IV.16) is simply the sum of squares of the prediction residuals, $\rho_i$. It should be clear from this discussion that the best predictor is the model that minimizes PRESS; as we shall see in Chapter V, the best predictor is not necessarily the model that provides the best fit to the data.

The strategy just described is an example of a leave-one-out cross validation test that is useful for selecting the most appropriate model from a set of possible alternatives. Unfortunately, as discussed by Shao [122], leave-one-out cross validation is asymptotically inconsistent in the sense that as the number of available data tends to infinity, the probability of selecting the best predictor using these measures does not converge to unity. Thus, alternative methods, such as leave-*n*-out cross validation have been proposed [122]. Although we will not discuss such methods here, later in this chapter we will describe a bootstrap method for estimating the prediction uncertainty for metamodels. The bootstrap method allows one to estimate confidence intervals for the relevant statistics computed from the simulation model.

### *IV.2.C Experimental Designs for RSs*

Although we have assumed thus far that the design points were given, in most applications we have the freedom to choose the design points; this provides substantial flexibility in terms of selecting the design points to optimize the amount of information available for constructing the metamodel. Returning to the assumption that the residuals are uncorrelated with constant variance, $\sigma^2$, it can be shown that the covariance matrix for the estimated regression coefficients, $\hat{\boldsymbol{\beta}}$, is given by [121]:

$$\mathrm{cov}[\hat{\boldsymbol{\beta}}] = \sigma^2 (\mathbf{F}_D^{\mathrm{T}} \mathbf{F}_D)^{-1} . \qquad \text{(IV.17)}$$

From this expression, it follows that the structure of the design matrix, $\mathbf{F}_D$, and hence the input configurations used to build the RS, directly affects our ability to provide accurate estimates of the RS coefficients. Consequently, many standard textbooks on RSM focus on the proper selection of input configurations so as to minimize some function of $\mathrm{cov}[\hat{\boldsymbol{\beta}}]$ (e.g., the determinant or trace) [73,121]. This process is referred to as Design of Experiments, or experimental design, and the basic objective is to select the input configurations that are expected to yield the most information regarding the underlying model while requiring as few model evaluations as possible. Numerous experimental designs have been proposed, to the extent that entire textbooks are dedicated to the subject, so that a detailed discussion of each of these methods is beyond the current scope. Many of the existing designs are quite sophisticated, having been optimized to elucidate various specific features of the underlying model [73,121]. Other designs, such as the 2-level factorial designs, are simpler and more easily understood, and we will discuss these briefly.

The basic approach to the 2-level factorial design is to discretize the range of each of the $m$ input parameters to a high setting, designated as +1, and a low setting, designated as -1, resulting in a total of $2^m$ possible design configurations. By evaluating the response at each of these configurations, it is possible to identify all of the main effects (i.e., the linear terms in the RS, such as $x_1$) as well as all of the interaction effects (i.e., the cross-product terms, such as $x_1 x_3$ or $x_1 x_2 x_3$) [121]. However, 2-level factorial designs cannot be used to identify higher order (i.e., quadratic) effects. Moreover, the number of necessary model evaluations increases exponentially with the number of model inputs, $m$, quickly becoming unreasonable for large $m$. In these cases, fractional factorial designs are often implemented. As the name suggests, fractional factorial designs consist of various fractions (i.e., ½-fraction, ¼-fraction, etc.) of a full factorial design, with the space of each fractional design being orthogonal to the other fractions. Hence, a fractional factorial design allows for a reduction in the number of required model evaluations, but at the cost of reduced experimental resolution. Specifically, a ½-fractional factorial design requires half as many model evaluations as a full factorial design, but only half (i.e., $2^{m-1}$) as many effects can be resolved; the remaining $2^{m-1}$ parameter effects will be indistinguishable from those that are estimated. This is known as confounding in the experimental design literature, and is analogous to the phenomena of aliasing in signal processing [73]. Nonetheless, fractional designs can still be quite useful if it is known that some of the high-order interaction effects are small or negligible; in these cases, one can design the experiment so as to purposefully confound these high-order effects with the low-order effects that are desired with minimal loss of accuracy [73]. These ideas are naturally extended to 3-level factorial designs and so on. However, alternative designs, such as Central Composite Designs and Box-Behnken Designs, are generally considered more efficient than simple 3-level Factorial designs for the development of QRSs [121].

One notable feature of many of these experimental design methodologies is that they tend to concentrate the design configurations near the periphery of the input space; for instance, the determinant of the covariance matrix is minimized when $\det(\mathbf{F}_D^{\mathrm{T}}\mathbf{F}_D)$ is maximized, and it can be shown that this occurs when the input parameters are set to their respective high and low extreme values (i.e., ±1) [121]. One can understand this graphically by considering the problem of estimating the slope of a linear regression model when the observations are corrupted by random noise. The solid line in Fig. 4 illustrates a linear model of the form $y = ax$, where $x$ takes values between +1 and -1. If observations are taken at $x = \pm 1$ with the distribution of random error indicated by the two Gaussian distributions at $x = \pm 1$, the range of possible models that would explain these hypothetical observations is given by the two dash-dotted lines. If, instead, we were to make observations $x = \pm 0.2$ with the same error distribution (illustrated at $x = \pm 0.2$), the resulting range of models would be given by the dashed lines. It is clear from this illustration that the variance in the estimated slope is minimized by making observations at $x = \pm 1$: that is, by setting the input parameter to its high and low extreme values. Clearly, when evaluating a deterministic model, such considerations are unnecessary, and consequently, several concerns have been raised regarding appropriateness of these design methodologies for deterministic simulation experiments [123,124]. This issue will be discussed further in the next section.

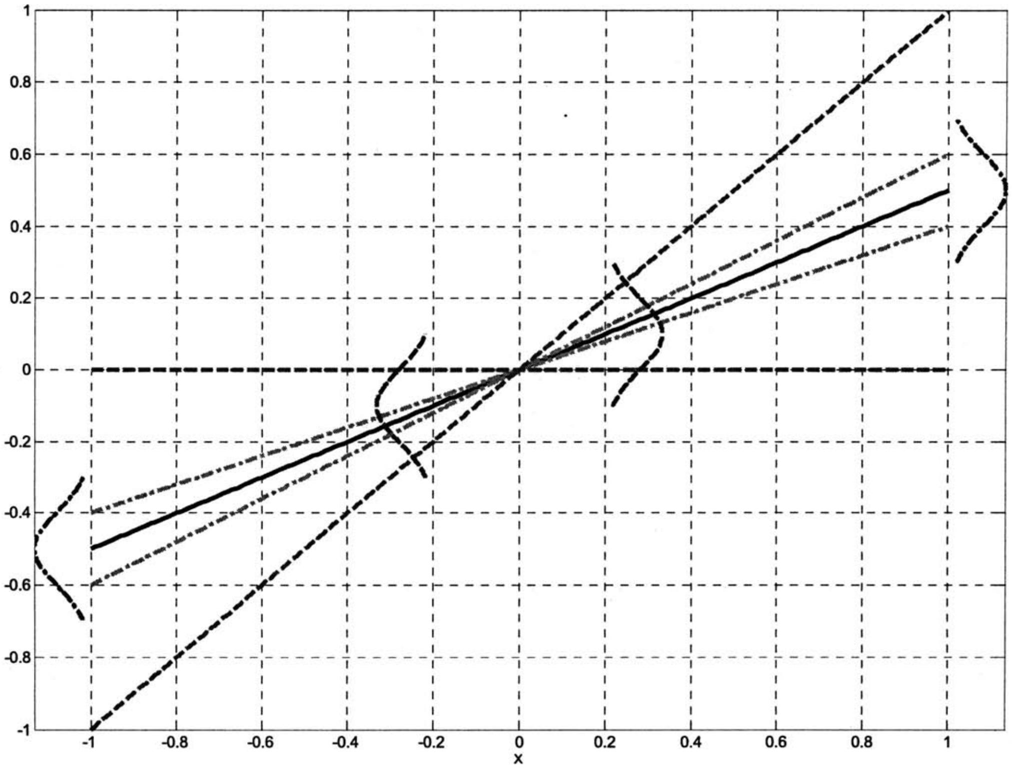

Figure 4: Illustration of the Effect of Random Error on Slope Estimation

### *IV.2.D Criticisms of RS Metamodeling*

The application of RSs to metamodeling of deterministic simulation models has not been met with unanimous approval, and some authors have voiced strong criticisms [123,124]. Much of this criticism stems from the lack of random variability in deterministic simulations. In particular, Sacks et al. [125] note that "*in the absence of independent random errors, the rationale for least-squares fitting of a response surface is not clear*," and go on to state that the "*usual measures of uncertainty derived from least-squares residuals have no obvious statistical meaning*." That is, although least-squares fitting can result in residuals that appear to be randomly distributed, these residuals are certainly not random and any attempt to interpret these residuals as such seems questionable. Consequently, unless numerous I/O data are available and these data are densely distributed throughout the entire input space, there is no statistical basis for supposing that such a fabricated residual distribution will be applicable across the entire problem domain. In other words, as Mitchell and Morris [92] put it, when the simulation model is deterministic, "*there is no basis for the 'response = signal + noise' model common to response surface methods*."

We should note, however, that RSs may be useful for identifying the overall trends that are present in the simulation model, and in such instances, the use of least-squares fitting seems justified since a precise description of the behavior of the residuals is unnecessary. To quote van Beers and Kleijnen [128], "*regression* [equivalently, RSs] *may be attractive when looking for an explanation – not a prediction – of the simulation's I/O behavior; for example, which inputs are most important*." Indeed, the results from our case studies presented in Chapter V indicate this to be true. However, one should bear in mind that because most statistical tests for parameter importance are based on the assumption that the residuals are iid (independent and identically distributed) random variables, an assumption which is clearly violated, the applicability of these tests is questionable.

Another criticism of RS metamodeling is due to the fact that many deterministic simulation models exhibit complex I/O behavior that cannot be adequately represented by a low-order polynomial RS. Specifically, Iman and Helton [129] have concluded that "*the use of response surface replacements for the computer models is not recommended as the underlying models are often too complex to be adequately represented by a simple function*." This is to be contrasted with the apparent success of RS approximations for physical experiments. However, as Saltelli et al. [80] note, "*in physical experimental design, the variation in the factors is often moderate (due to cost, for instance)*" so that the RS is being constructed over a localized region of the parameter space. Consequently, a low-order polynomial approximation is often sufficient. On the other hand, "*in numerical experiments ... factors are* [often] *varied generously over orders of magnitude*" [80]. In these cases, such a low-order approximation may not be sufficient. As a related matter, it was mentioned in Section IV.2.B that the failure of a RS to exactly interpolate the data from a deterministic simulation is conclusive proof that the assumed model form is incorrect. Thus, rather than being faced with the challenge of selecting the correct model from a set of competing metamodels, each of which could conceivably be correct based on the available data, we are forced to select, from a set of incorrect models, the one that is least in error. Moreover, as Sacks et al. [125] claim, "*The adequacy of a response surface model fitted to the observed data is determined solely by systematic bias*." In other words, if the assumed form of the RS is incorrect, any prediction made with this RS is necessarily biased since the predicted response is nowhere (i.e., for any **x**) expected to equal the true response. Unfortunately, the extent and direction of this bias can be very difficult to quantify, although methods such as bootstrapping (see Section IV.5) can be of some assistance.

Finally, Sacks et al. [125] have criticized the use of the experimental design procedures discussed in the previous section, claiming that "*it is unclear that current methodologies for the design and analysis of physical experiments are ideal for complex deterministic computer models.*" As an example, these authors note that the "*classical notions of experimental unit, blocking, replication and randomization are irrelevant,*" due to the lack of random variability in deterministic simulations. In addition, we previously illustrated how optimal experimental designs tend to concentrate the design points near the periphery of the input space whenever the output is corrupted by random noise; recall that this was a consequence of using $\det(\mathbf{F}_D^{\mathrm{T}}\mathbf{F}_D)$ as the objective function for optimizing the experimental design. However, when the simulation model is deterministic, such a design configuration may not be optimal, and as a result, various alternative objective functions have been proposed for optimizing experimental designs for computational experiments [91,92,123,125,130,131]. Koehler and Owen [132] provide a concise summary of many of these modern experimental design strategies, and the book by Santner et al. [126] contains multiple chapters devoted to experimental designs for deterministic computer models. A description of each of these methods is beyond the scope of this thesis. It should be noted, however, that most of these designs seek a more uniform sampling across the entire input space so as to provide a better indication of the global model behavior.

## IV.3 Gaussian Process (GP) Models and the Kriging Predictor

Gaussian Process models derive from a spatial prediction technique known as kriging to the geostatistics community. Kriging is named after the South African mining engineer Daniel G. Krige. The technique was formalized as a statistical prediction methodology in the 1960's by Matheron [133]. Sacks et al. [125] first proposed the use of kriging for predicting the output from computer simulations in the 1980's and the technique has since evolved into the modern GP models that have been quite popular for metamodeling. The precise distinction between kriging and GPs is somewhat subtle and the two terms seem to be used interchangeably on occasion in the literature; one notable distinction is that the kriging literature occasionally refers to something called a covariogram, which is similar to, yet still distinct from, the more familiar concept of covariance used by GP practitioners. Although both approaches lead to the same expression for the metamodel, kriging technically relies on fewer assumptions so we shall begin our development by discussing the kriging predictor. This will be followed by a discussion of GP models, primarily because this formulation will be useful when we consider Bayesian prediction in Section IV.5.B.

### *IV.3.A The Kriging Predictor*

Suppose we wish to determine the output from the model, $y_0 = g(\mathbf{x}_0)$, for some arbitrary input configuration, $\mathbf{x}_0$. Before actually running the model, we do not know the precise value that $y$ will take, so it seems natural to assume that, a priori, $y_0$ is a random variable. More precisely, we say that an output from the computer model is a realization of a stochastic process of the form:

$$y_0 = y(\mathbf{x}_0) = \mathbf{f}_0^{\mathbf{T}}\boldsymbol{\beta} + \delta_0 \tag{IV.18}$$

where, as before, $\boldsymbol{\beta}$ is a $q\times 1$ vector of unknown regression coefficients, $\mathbf{f}_0^{\mathbf{T}} = \mathbf{f}^{\mathbf{T}}(\mathbf{x}_0)$ represents some known transformation of the inputs (e.g., a linear or quadratic model, as for the RS), and $\delta_0 = \delta(\mathbf{x}_0)$ is a zero-mean random function. At this point, we will not make any assumptions regarding the distribution of $\delta(\mathbf{x})$. Now, suppose that we have observed the outputs, $\mathbf{y}_D$, for a set of N$_D$ design points, $\mathbf{x}_D^{(i)}$, $i$=1,2,...$N_D$. Expressing each of these data in terms of the function given by Eq. (IV.18) gives the following system of equations:

$$\mathbf{y}_D = \mathbf{F}_D\boldsymbol{\beta} + \boldsymbol{\delta}_D \tag{IV.19}$$

where $\mathbf{F}_D$ is the design matrix defined identically to Eq. (IV.4), and $\boldsymbol{\delta}_D = \{\delta(\mathbf{x}_D^{(1)}),\ldots\delta(\mathbf{x}_D^{(N_D)})\}^{\mathbf{T}}$.

Our goal is to use the information contained in the I/O mappings of the design points to predict the output, say $y_0$, for some input configuration, $\mathbf{x}_0$, that has not yet been observed. To do so, we consider a linear predictor of the form:

$$\hat{y}_0 = \boldsymbol{\lambda}^{\mathbf{T}}\mathbf{y}_D \tag{IV.20}$$

where $\boldsymbol{\lambda}$ is some unknown vector that we will determine. Notice that Eq. (IV.20) states that the estimated output for any input, $\mathbf{x}_o$, is some linear combination (i.e., a weighted average) of the observed responses, $\mathbf{y}_D$. One natural constraint that we can impose on the estimate is that it be unbiased; that is, we require that:

$$\mathrm{E}(\hat{y}_o) = \mathrm{E}(y_o) = \mathrm{E}[\mathbf{f}_o^{\mathrm{T}}\boldsymbol{\beta} + \delta_o] = \mathbf{f}_o^{\mathrm{T}}\boldsymbol{\beta} \tag{IV.21}$$

where the middle equality follows from Eq. (IV.18) and the last inequality results from the assumption that $\delta_o$ is a zero-mean random function. On the other hand, from Eqs. (IV.19) and (IV.20), we have:

$$\mathrm{E}(\hat{y}_o) = \mathrm{E}(\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{y}_D) = \boldsymbol{\lambda}^{\mathrm{T}}\mathrm{E}(\mathbf{y}_D) = \boldsymbol{\lambda}^{\mathrm{T}}\mathrm{E}(\mathbf{F}_D\boldsymbol{\beta} + \boldsymbol{\delta}_D) = \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{F}_D\boldsymbol{\beta}\,. \tag{IV.22}$$

Finally, equating Eqs. (IV.21) and (IV.22), gives the unbiasedness constraint:

$$\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{F}_D\boldsymbol{\beta} - \mathbf{f}_o^{\mathrm{T}}\boldsymbol{\beta} = (\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{F}_D - \mathbf{f}_o^{\mathrm{T}})\boldsymbol{\beta} = 0 \quad \Rightarrow \quad \mathbf{F}_D^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{f}_o \tag{IV.23}$$

since, by assumption, $\boldsymbol{\beta} \neq 0$.

Any predictor of the form given by Eq. (IV.20) that satisfies the unbiasedness constraint given by Eq. (IV.23) is called a Linear Unbiased Predictor (LUP) [126]. Notice that Eq. (IV.23) represents an underdetermined system of equations, since $\mathbf{F}_D^{\mathrm{T}}$ is a $q \times N_D$ matrix ($q < N_D$). Hence, there is no unique LUP. However, by imposing an additional constraint, it is possible to identify the LUP that is, in some sense, optimal. Specifically, we would like to find an expression for the LUP that minimizes the Mean Square Prediction Error (MSPE), defined as:

$$\mathrm{MSPE}(\hat{y}_o) = \mathrm{E}\left[(\hat{y}_o - y_o)^2\right] \tag{IV.24}$$

where the difference, $\hat{y}_o - y_o$, is the prediction error. Combining Eqs. (IV.18)-(IV.20), we have:

$$\begin{aligned} \hat{y}_o - y_o &= \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{y}_D - \mathbf{f}_o^{\mathrm{T}}\boldsymbol{\beta} - \delta_o = \boldsymbol{\lambda}^{\mathrm{T}}(\mathbf{F}_D\boldsymbol{\beta} + \boldsymbol{\delta}_D) - \mathbf{f}_o^{\mathrm{T}}\boldsymbol{\beta} - \delta_o \\ &= (\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{F}_D - \mathbf{f}_o^{\mathrm{T}})\boldsymbol{\beta} + \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{\delta}_D - \delta_o \end{aligned}\,. \tag{IV.25}$$

The second quantity in parenthesis in Eq. (IV.25) is zero by virtue of Eq. (IV.23), so the MSPE simplifies to:

$$\begin{aligned}\mathrm{MSPE}(\hat{y}_{\mathrm{o}}) &= \mathrm{E}\left[(\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{\delta}_D - \delta_{\mathrm{o}})^2\right] = \mathrm{E}\left[\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{\delta}_D\boldsymbol{\delta}_D^{\mathrm{T}}\boldsymbol{\lambda} - 2\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{\delta}_D\delta_{\mathrm{o}} + \delta_{\mathrm{o}}^2\right] \\ &= \boldsymbol{\lambda}^{\mathrm{T}}\mathrm{E}(\boldsymbol{\delta}_D\boldsymbol{\delta}_D^{\mathrm{T}})\boldsymbol{\lambda} - 2\boldsymbol{\lambda}^{\mathrm{T}}\mathrm{E}(\boldsymbol{\delta}_D\delta_{\mathrm{o}}) + \mathrm{E}(\delta_{\mathrm{o}}^2) \qquad . \\ &= \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{\Sigma}_{DD}\boldsymbol{\lambda} - 2\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{\sigma}_{\mathrm{o}} + \sigma^2\end{aligned} \tag{IV.26}$$

In the last equality in Eq. (IV.26), we have made use of the following definitions:

$$\boldsymbol{\Sigma}_{DD} = \mathrm{E}(\boldsymbol{\delta}_D\boldsymbol{\delta}_D^{\mathrm{T}}) = \begin{bmatrix} \mathrm{cov}\left(\boldsymbol{\delta}_D^{(1)},\boldsymbol{\delta}_D^{(1)}\right) & \mathrm{cov}\left(\boldsymbol{\delta}_D^{(1)},\boldsymbol{\delta}_D^{(2)}\right) & \cdots & \mathrm{cov}\left(\boldsymbol{\delta}_D^{(1)},\boldsymbol{\delta}_D^{(N_D)}\right) \\ \mathrm{cov}\left(\boldsymbol{\delta}_D^{(2)},\boldsymbol{\delta}_D^{(1)}\right) & \mathrm{cov}\left(\boldsymbol{\delta}_D^{(2)},\boldsymbol{\delta}_D^{(2)}\right) & \cdots & \mathrm{cov}\left(\boldsymbol{\delta}_D^{(2)},\boldsymbol{\delta}_D^{(N_D)}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{cov}\left(\boldsymbol{\delta}_D^{(N_D)},\boldsymbol{\delta}_D^{(1)}\right) & \mathrm{cov}\left(\boldsymbol{\delta}_D^{(N_D)},\boldsymbol{\delta}_D^{(2)}\right) & \cdots & \mathrm{cov}\left(\boldsymbol{\delta}_D^{(N_D)},\boldsymbol{\delta}_D^{(N_D)}\right) \end{bmatrix} \tag{IV.27a}$$

$$\boldsymbol{\sigma}_{\mathrm{o}} = \mathrm{E}(\boldsymbol{\delta}_D\delta_{\mathrm{o}}) = \left\{\mathrm{cov}\left[\boldsymbol{\delta}_D^{(1)},\delta_{\mathrm{o}}\right],\ldots,\mathrm{cov}\left[\boldsymbol{\delta}_D^{(N_D)},\delta_{\mathrm{o}}\right]\right\}^{\mathrm{T}} \tag{IV.27b}$$

$$\sigma^2 = \mathrm{var}(\delta_{\mathrm{o}}) = \mathrm{E}(\delta_{\mathrm{o}}^2) \tag{IV.27c}$$

where $\boldsymbol{\delta}_D^{(i)} = \delta(\mathbf{x}_D^{(i)})$. The matrix, $\boldsymbol{\Sigma}_{DD}$ is called the covariance matrix, and $\sigma^2$ is referred to as the process variance as it is the variance of the stochastic process, $\delta(\mathbf{x})$. We shall defer a detailed discussion of the covariance structure until Section IV.3.B and assume, for the present, that all of the quantities in Eq. (IV.27) are known.

The Best Linear Unbiased Predictor (BLUP) is defined as the LUP given by the vector $\boldsymbol{\lambda}$ that minimizes Eq. (IV.26) subject to the unbiasedness constraint given by Eq. (IV.23). The details of this calculation can be found in [126], and the result is as follows:

$$\boldsymbol{\lambda}^{\mathrm{T}} = \boldsymbol{\sigma}_{\mathrm{o}}^{\mathrm{T}}\boldsymbol{\Sigma}_{DD}^{-1} + (\mathbf{f}_{\mathrm{o}}^{\mathrm{T}} - \boldsymbol{\sigma}_{\mathrm{o}}^{\mathrm{T}}\boldsymbol{\Sigma}_{DD}^{-1}\mathbf{F}_D)(\mathbf{F}_D^{\mathrm{T}}\boldsymbol{\Sigma}_{DD}^{-1}\mathbf{F}_D)^{-1}\mathbf{F}_D^{\mathrm{T}}\boldsymbol{\Sigma}_{DD}^{-1}. \tag{IV.28}$$

Using this expression, the BLUP can be expressed as:

$$\hat{y}_{\mathrm{o}} = \mathbf{f}_{\mathrm{o}}^{\mathrm{T}}\hat{\boldsymbol{\beta}} + \boldsymbol{\sigma}_{\mathrm{o}}^{\mathrm{T}}\boldsymbol{\Sigma}_{DD}^{-1}(\mathbf{y}_D - \mathbf{F}_D\hat{\boldsymbol{\beta}}) \tag{IV.29}$$

where $\hat{\boldsymbol{\beta}}$ is defined as:

$$\hat{\boldsymbol{\beta}} = (\mathbf{F}_D^{\mathrm{T}}\boldsymbol{\Sigma}_{DD}^{-1}\mathbf{F}_D)^{-1}\mathbf{F}_D^{\mathrm{T}}\boldsymbol{\Sigma}_{DD}^{-1}\mathbf{y}_D. \tag{IV.30}$$

Equation (IV.29) defines the kriging predictor of the model output for the input, $\mathbf{x}_{\mathrm{o}}$, and Eq. (IV.30) is the generalized (or weighted) least-squares estimate for the regression coefficients. An estimate for the prediction error is given by the MSPE, which, after substituting the expression for $\boldsymbol{\lambda}$ in Eq. (IV.28), becomes [126]:

$$\mathrm{MSPE}(\hat{y}_o) = \sigma^2 - \boldsymbol{\sigma}_o^{\mathrm{T}} \boldsymbol{\Sigma}_{DD}^{-1} \boldsymbol{\sigma}_o + (\mathbf{f}_o^{\mathrm{T}} - \boldsymbol{\sigma}_o^{\mathrm{T}} \boldsymbol{\Sigma}_{DD}^{-1} \mathbf{F}_D)(\mathbf{F}_D^{\mathrm{T}} \boldsymbol{\Sigma}_{DD}^{-1} \mathbf{F}_D)^{-1} (\mathbf{f}_o^{\mathrm{T}} - \boldsymbol{\sigma}_o^{\mathrm{T}} \boldsymbol{\Sigma}_{DD}^{-1} \mathbf{F}_D)^{\mathrm{T}} \tag{IV.31}$$

Notice that if $\mathbf{x}_o = \mathbf{x}_D^{(i)}$ for some $i \in \{1,2,...,N_D\}$, then $\hat{y}_o = \hat{y}(\mathbf{x}_D^{(i)}) = \hat{y}_D^{(i)}$, $\mathbf{f}_o^{\mathrm{T}} = \mathbf{f}^{\mathrm{T}}(\mathbf{x}_D^{(i)})$ and $\boldsymbol{\sigma}_o^{\mathrm{T}} \boldsymbol{\Sigma}_{DD}^{-1} = \mathbf{e}_i^{\mathrm{T}}$, where $\mathbf{e}_i$ is a vector with a one as its $i^{\text{th}}$ element and zeros for every other element (i.e., a Euclidean basis vector). Furthermore, it is readily verified that $\mathbf{e}_i^{\mathrm{T}} \mathbf{F}_{\mathbf{D}} = \mathbf{f}^{\mathrm{T}}(\mathbf{x}_D^{(i)}) = \mathbf{f}_o^{\mathrm{T}}$ and $\mathbf{e}_i^{\mathrm{T}} \boldsymbol{\sigma}_o = \mathrm{cov}(\delta_o, \delta_o) = \sigma^2$. Consequently, for any $i \in \{1,2,...,N_D\}$, we have:

$$\mathrm{MSPE}(\hat{y}_D^{(i)}) = 0 \,. \tag{IV.32}$$

That is, the prediction error is zero for any input in the design set. Similarly, one can easily show that $\hat{y}_D^{(i)} = y_D^{(i)}$. In other words, unlike the standard polynomial RS, the kriging predictor exactly interpolates the design data. To demonstrate these results, Fig. 5 illustrates a simple kriging predictor and the ±2×MSPE prediction error bands for the one-dimensional nonlinear function, $y = [3x + \frac{1}{2}\cos(4\pi x)]e^{-x}$. A total of eight data points were used to construct the predictor and it is apparent that the kriging predictor provides an exceptionally good fit to the true function.

Recall that because the RS fails, in general, to interpolate the design data, the probability that the RS represents the true model is zero; the polynomial RS is, therefore, necessarily biased, since for any of the design points, $\hat{y}_D^{(i)} \neq y_D^{(i)} = \mathrm{E}\left[y\left(x_D^{(i)}\right) | \mathbf{y}_D\right]$. In other words, since the output from the RS does not match the known outputs, there is no reason to expect that the RS predictions for other input configurations will match the true output. On the other hand, since the kriging predictor does interpolate the design data, there is a nonzero probability, however small, that the kriging model will correctly predict, to an arbitrary degree of accuracy, the model output for all possible inputs. Hence, the kriging predictor is not necessarily biased. Kriging can be considered a generalization of the simple polynomial RS. Indeed, the first term in Eq. (IV.29) is simply a RS model, although the expression for the regression coefficients is slightly different compared to Eq. (IV.5). It is the second term in Eq. (IV.29), which we call the covariance term, that gives rise to the unique features of the kriging metamodel. In particular, while the regression model describes the global variation of the predictor, the covariance term forces the kriging predictor to locally interpolate the known data points. This feature is what makes kriging a type of nonparametric regression. Because of the significance of the covariance structure in the kriging model, the following section is dedicated to a discussion of how to appropriately model the covariance.

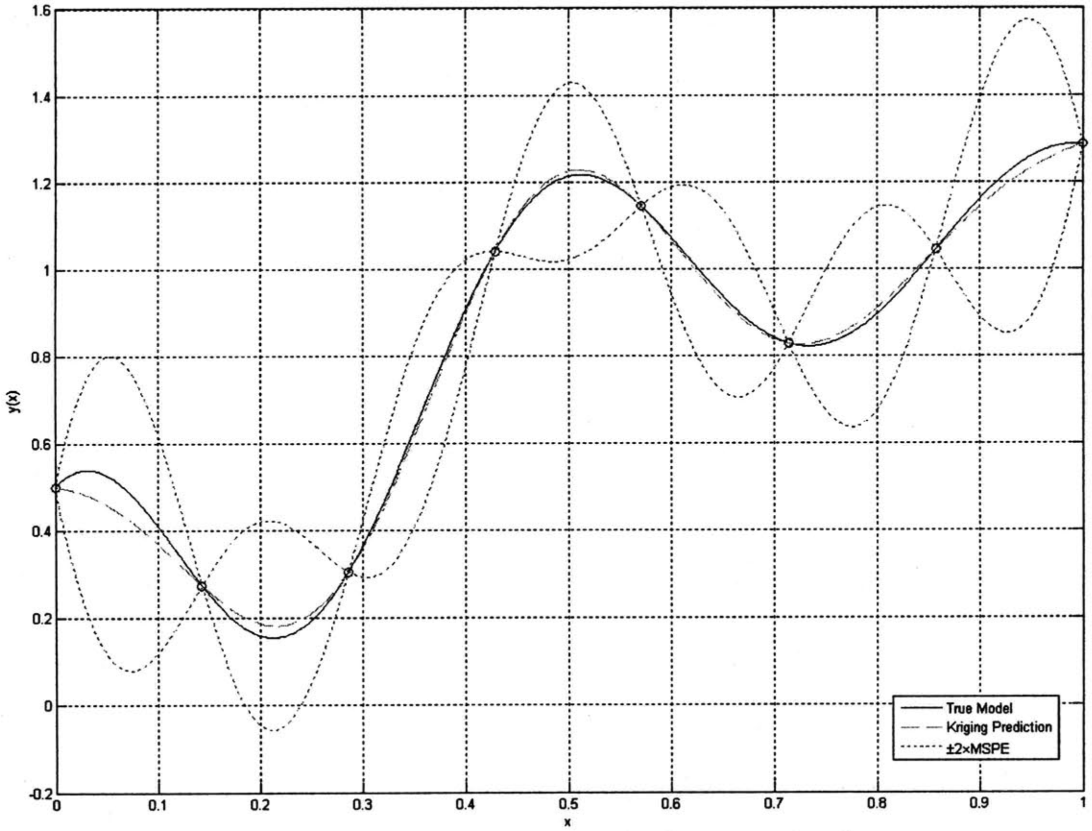

Figure 5: Illustration of kriging for a simple 1-D test function

### ***IV.3.B Modeling the Covariance***

The capability of the kriging estimator to make accurate predictions is, to a large extent, determined by the assumed covariance structure of the random process, $\delta(\mathbf{x})$. As a result, it often suffices to simply assume $\mathbf{f}(\mathbf{x}) = \beta_o$, where $\beta_o$ is some constant representing the mean output from the computer model; this approach is called ordinary kriging [134]. Generally, one considers a covariance model of the form:

$$\operatorname{cov}[\delta(\mathbf{u}), \delta(\mathbf{v})] = \sigma^2 K(\mathbf{u}, \mathbf{v} \mid \boldsymbol{\psi}) \tag{IV.33}$$

where $\mathbf{u}$ and $\mathbf{v}$ are two arbitrary input configurations, $K(\mathbf{u},\mathbf{v}\mid\boldsymbol{\psi})$ represents a family of correlation functions parameterized by the vector $\boldsymbol{\psi}$, and $\sigma^2 = \mathrm{cov}[\delta(\mathbf{u}),\delta(\mathbf{u})] = \mathrm{var}[\delta(\mathbf{u})]$. Consequently, the correlation functions satisfy $K(\mathbf{u},\mathbf{u}\mid\boldsymbol{\psi}) = 1$ for any input $\mathbf{u}$. From the covariance model given by Eq. (IV.33), we have the following:

$$\boldsymbol{\Sigma}_{DD} = \sigma^2\mathbf{K}_{DD} = \sigma^2\begin{bmatrix} K\left(\mathbf{x}_D^{(1)},\mathbf{x}_D^{(1)}\mid\boldsymbol{\psi}\right) & K\left(\mathbf{x}_D^{(1)},\mathbf{x}_D^{(2)}\mid\boldsymbol{\psi}\right) & \cdots & K\left(\mathbf{x}_D^{(1)},\mathbf{x}_D^{(N_D)}\mid\boldsymbol{\psi}\right) \\ K\left(\mathbf{x}_D^{(2)},\mathbf{x}_D^{(1)}\mid\boldsymbol{\psi}\right) & K\left(\mathbf{x}_D^{(2)},\mathbf{x}_D^{(2)}\mid\boldsymbol{\psi}\right) & \cdots & K\left(\mathbf{x}_D^{(2)},\mathbf{x}_D^{(N_D)}\mid\boldsymbol{\psi}\right) \\ \vdots & \vdots & \ddots & \vdots \\ K\left(\mathbf{x}_D^{(N_D)},\mathbf{x}_D^{(1)}\mid\boldsymbol{\psi}\right) & K\left(\mathbf{x}_D^{(N_D)},\mathbf{x}_D^{(2)}\mid\boldsymbol{\psi}\right) & \cdots & K\left(\mathbf{x}_D^{(N_D)},\mathbf{x}_D^{(N_D)}\mid\boldsymbol{\psi}\right) \end{bmatrix} \tag{IV.34a}$$

$$\boldsymbol{\sigma}_o = \sigma^2\boldsymbol{\kappa}_o = \sigma^2\left\{K\left(x_D^{(1)},x_o\mid\boldsymbol{\psi}\right),\ldots,K\left(x_D^{(N_D)},x_o\mid\boldsymbol{\psi}\right)\right\}^{\mathrm{T}} \tag{IV.34b}$$

where, we refer to $\mathbf{K}_{DD}$ as the correlation matrix. To simplify notation, we do not explicitly denote the dependence of $\mathbf{K}_{DD}$ and $\boldsymbol{\kappa}_o$ on the parameters $\boldsymbol{\psi}$. In the following section, we discuss some of the basic assumptions regarding the general form of the correlation function. We note that while the parametric covariance model just described is the most commonly used method, Le and Zidek describe a fully nonparametric approach for modeling the covariance [118]. However, their approach is rather complex and will not be discussed herein.

(i) Some General Assumptions on Correlation Functions:

Santner et al. [126] and Cressie [134] discuss the necessary conditions for $K(\cdot,\cdot\mid\boldsymbol{\psi})$ to be a valid correlation function, the most important being that the resulting correlation matrix, $\mathbf{K}_{DD}$, must be positive semidefinite. Frequently, one imposes additional restrictions on $K(\cdot,\cdot\,\boldsymbol{\psi})$ to simplify the analysis. For example, one such assumption is that the stochastic process is *isotropic*, so that:

$$K(\mathbf{u},\mathbf{v}\mid\boldsymbol{\psi}) = K(d\mid\boldsymbol{\psi}) \tag{IV.35}$$

where $d = \|\mathbf{u}-\mathbf{v}\|$ denotes the standard Euclidean distance between $\mathbf{u}$ and $\mathbf{v}$. This correlation function is invariant under spatial translations and orthogonal transformations (e.g., rotations). In words, Eq. (IV.35) assumes that the correlation between two observations from the computer model, $g(\mathbf{x})$, is only a function of the distance between the corresponding inputs. This assumption is motivated by the idea that if $g(\mathbf{x})$ is a continuous function, then two inputs, **u** and **v**, that are close together (i.e., $d \approx 0$) should yield highly correlated outputs, and the degree of correlation should decrease as the distance between these points increases. Consequently, an isotropic correlation model will result in a kriging estimator that predicts $g(\mathbf{x})$ by weighting the observed design points, $\mathbf{x}_D$, in proportion to their distance from **x**.

Although the isotropic correlation model can be quite effective for making spatial predictions (e.g., predicting geological conditions), in the context of computer simulation, each component of the input, **x**, generally has a different physical interpretation. Thus, assuming that the correlation depends only on the Euclidean distance between two inputs makes little sense. For instance, suppose that $\mathbf{x} = \{x_1, x_2\}$, where $x_1$ represents a dimensionless pressure and $x_2$ is a dimensionless temperature. An isotropic correlation model would suggest that the correlation between $\{x_1, x_2\}$ and $\{x_1+\Delta, x_2\}$, for some quantity $\Delta$, is identical to the correlation between $\{x_1, x_2\}$ and $\{x_1, x_2 + \Delta\}$. Even though the inputs are dimensionless, it might be reasonable to suppose that a change in $x_1$ will have an entirely different effect than a change in $x_2$. Consequently, the assumption that the stochastic process is isotropic is frequently relaxed by assuming, instead, that the process is *stationary* in each dimension, separately; that is, we assume a product correlation model of the form:

$$K(\mathbf{u},\mathbf{v} \mid \psi) = \prod_{i=1}^{m} K_i(d_i \mid \psi_i) \tag{IV.36}$$

where $d_i = | u_i - v_i |$ is the absolute difference between the $i^{\text{th}}$ components of **u** and **v**, and each of the $m$ functions, $K_i(\cdot \mid \psi_i)$, is a correlation function. Notice that, although Eq. (IV.36) is invariant under componentwise translations, it is not invariant with respect to orthogonal transformations. The advantage of the stationary correlation model is that we can specify the degree of correlation for each input parameter separately. In the remainder of this thesis, we will be considering stationary correlation models of the form given by Eq. (IV.36). The assumption of stationarity is generally not too restrictive since, in many cases, any nonstationary effects can be handled through proper specification of the regression model [125]. However, there are some instances where the stationarity assumption is clearly not valid, and these issues will be discussed in Section IV.3.D.

Although we have restricted our consideration to deterministic computer models, there may be occasions where one must deal with random variability in the model outputs. Alternatively, if some of the input parameters are screened from consideration and are not included in the prediction metamodel, then, in general, the metamodel cannot be expected to exactly interpolate all of the design data points. For instance, suppose $\mathbf{x} = \{x_1, x_2, x_3\}$ and the simulation model has been run for each of the nine configurations for which $x_i = \pm 1$. If $x_3$ is removed during the screening process, then the metamodel will be of the form: $\hat{y} = \hat{y}(x_1, x_2)$. However, if $g(x_1, x_2, +1) \neq g(x_1, x_2, -1)$, then the metamodel will not be able to exactly interpolate the data since, from the perspective of the metamodel, the computer model is returning two different outputs for the same input configuration, $\{x_1, x_2\}$. Thus, screening introduces apparent random variability. This can be dealt with by modifying the correlation model as such:

$$K(\mathbf{u}, \mathbf{v} \mid \psi, \tau) = K(\mathbf{u}, \mathbf{v} \mid \psi) + \frac{\tau^2}{\sigma^2} \tilde{\delta}\left( \left\| \mathbf{u} - \mathbf{v} \right\| \right) \tag{IV.37}$$

where $\tau$ is some constant and $\tilde{\delta}(\cdot)$ is the Dirac delta function, not to be confused with the random process, $\delta(\mathbf{x})$. The second term in Eq. (IV.37) is called the 'nugget' [101,135]. The resulting covariance matrix will then take the form, $\mathbf{\Sigma}_{DD} = \sigma^2 \mathbf{K}_{DD} + \tau^2 \mathbf{I}$, where $\mathbf{I}$ is the $N_D \times N_D$ identity matrix. Notice that when $\sigma^2 / \tau^2 << 1$, the nugget dominates, and the covariance matrix can be approximated by $\tau^2 \mathbf{I}$. In this case, the deviations from the underlying regression model are essentially uncorrelated, and the resulting kriging model will resemble a typical RS model; this is called a limiting linear model and this effect can be inferred from Fig. 5 [135].

In addition to accounting for random variability, the nugget effect is useful for conditioning the covariance matrix. This will be discussed further in later sections, but for now, it suffices to recognize that when two inputs, say $\mathbf{x}_D^{(i)}$ and $\mathbf{x}_D^{(j)}$, are very close together, the $i$ and $j$ columns in the covariance matrix will be very similar. Consequently, the correlation matrix (as well as the covariance matrix) will be nearly singular and ill-conditioned. This is troublesome because the kriging predictor requires the inversion of the covariance matrix, and if this matrix is ill-conditioned, large numerical errors can result. This problem can be alleviated by adding a nugget. Mathematically, this is equivalent to adding a multiple of the identity matrix to the correlation matrix, with the result being that the covariance matrix is no longer singular and is better conditioned. In practice, this conditioning process is often necessary, particularly when many design data are available so that some inputs are inevitably close together. Fortunately, a relatively small nugget (i.e., on the order of $N_D$ times the machine accuracy) often suffices [136]. A consequence of this is that the kriging estimator will not exactly interpolate the design points, but will still be so close as to be inconsequential for most applications. In the next section we discuss the most popular families of correlation functions that have been used in the literature.

(ii) Common Families of Correlation Functions:

Numerous families of correlation functions have been proposed in the literature; for example, Le and Zidek [118] describe at least 13 different models, including the exponential, Gaussian, spherical, Cauchy, and Matèrn families (see also Chapter 2 of [119]). While each of these models possesses its own advantages, we will restrict our discussion to two families of correlation functions: namely, the generalized power exponential and Matèrn families. We note that both the exponential and Gaussian families are special cases of these two families. Both the generalized power exponential and Matèrn families are parameterized by $\psi_i = \{\theta_i, \upsilon_i\}$, where $\theta_i$ is called the range parameter and $\upsilon_i$ is a smoothness parameter. In all of the following equations, we take $d_i = | u_i - v_i |$.

*The Generalized Power Exponential Family:*

The generalized power exponential family consists of correlation functions of the following form:

$$K_i(d_i \mid \psi_i) = \exp\left[ -\left( \frac{d_i}{\theta_i} \right)^{\upsilon_i} \right] \qquad \text{(IV.38)}$$

where $\theta_i \in (0, \infty)$ and $\upsilon_i \in (0,2]$. Notice that, as its name would suggest, the range parameter, $\theta_i$, controls the extent to which the correlation extends; for small $\theta_i$, a small increase in $d_i$ causes the correlation to rapidly decay, whereas a large $\theta_i$ indicates that the correlation extends over larger distances. The smoothness parameter, $\upsilon_i$, is so named because it determines the degree to which the random process is mean-square differentiable; in particular, when $\upsilon_i < 2$, the process is not mean-square differentiable, and the resulting predictor is not smooth [132]. On the other hand, when $\upsilon_i = 2$, the process is infinitely mean-square differentiable [132]. For this case, the predictor will be an analytic function. When $\upsilon_i = 1$, Eq. (IV.38) is the exponential correlation model, and when $\upsilon_i = 2$ it is called the Gaussian correlation model. Fig. 6 illustrates the qualitative differences between the kriging predictors constructed with both of these correlation models and with various range parameters; we have used the same demonstration function used for Figure 5. As expected, the exponential correlation model results in a non-smooth predictor, with discontinuous derivatives at each of the design points, whereas the Gaussian correlation model yields smooth predictors. Moreover, as $\theta_i$ is increased, the range of correlation is extended and, for the exponential case, the predictor approaches a piecewise linear interpolator. On the contrary, as $\theta_i$ is decreased, one can clearly see the exponential kriging predictor approaching the underlying linear regression model (i.e., the limiting linear model) between each of the design points. To a lesser degree, the same features are evident in the accuracy, sufficing instead to??identify bounds that co6: Comparison of Kriging Predictors with

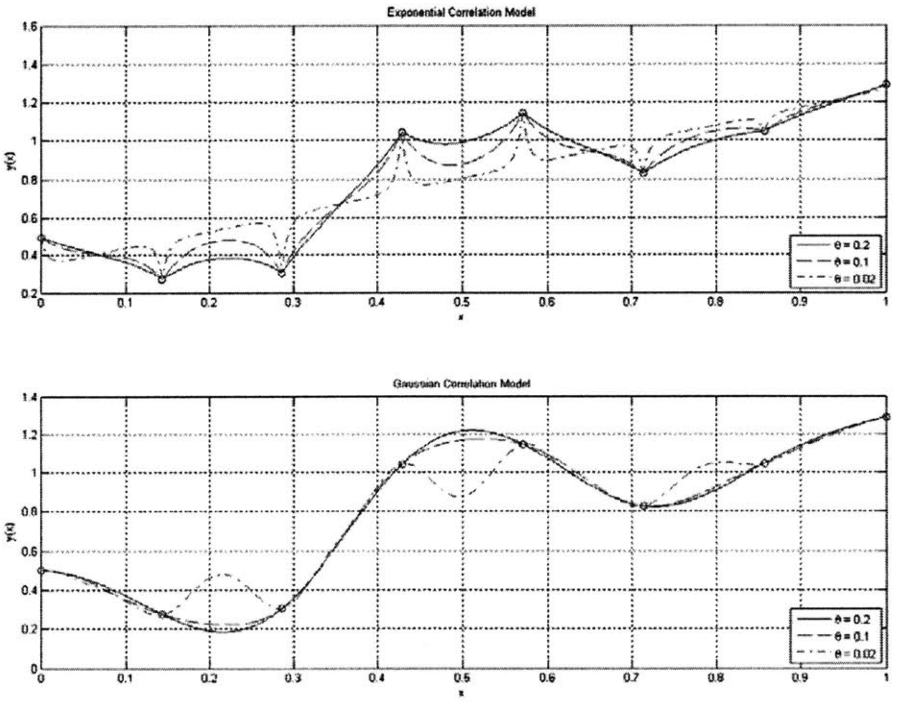

Figure 6: Comparison of Kriging Predictors with Exponential and Gaussian Correlation Models

The generalized power exponential correlation family has remained one of the most commonly used correlation models for computer simulation metamodeling. The reason for this is seems to be its simplicity; the correlation matrices can be computed quickly, and, if necessary, the functional form of Eq. (IV.38) makes it easy to analytically compute derivatives of the correlation matrices. This latter feature is useful for estimating the parameters, $\boldsymbol{\psi}_i$, through an optimization process (i.e., to maximize the likelihood function). In Section IV.3.D, we will return to the important issue of estimating the correlation parameters. The only shortcoming of the generalized power exponential family seems to be the lack of control over the differentiability of the kriging estimator. We noted previously that for $\upsilon_i < 2$, the kriging predictor will be continuous but will have discontinuous first derivatives, whereas, for $\upsilon_i = 2$, the kriging predictor will be infinitely differentiable. Thus, it is not possible to specify any particular order of differentiability that the predictor should have (i.e., that the predictor should have continuous 2nd derivatives, but not 3rd-order). The significance of this limitation is clearly problem specific, and in several cases it might be safe to assume that the output from the computer model is infinitely differentiable: for instance, if the computer model is solving a system of partial differential equations whose solutions are known to be analytic. Nevertheless, it is for this lack of control that Stein [137] recommends the Matèrn model.

*The Matèrn Family:*

The Matèrn family consists of correlation functions of the form [138]:

$$K_i(d_i \mid \boldsymbol{\psi}_i) = \frac{(\widetilde{d}_i)^{\upsilon_i}}{\Gamma(\upsilon_i)2^{\upsilon_i - 1}} B_{\upsilon_i}(\widetilde{d}_i) \tag{IV.39}$$

where $\widetilde{d}_i = d_i / \theta_i$, with $\theta_i \in (0, \infty)$ and $\upsilon_i > 0$. The function, $\Gamma(\cdot)$ is the Gamma function, and $B_{v_i}(\cdot)$ is the modified Bessel function[‡] of the second kind of order $\upsilon_i$ [138]. The interesting feature of the Matèrn family is that the random process will be $n$ times differentiable if and only

[‡] The standard notation for this function is $K_v(\cdot)$ (see, e.g., [139]), and in MATLAB® it is called with the function 'BESSELK'. Clearly, this would be confusing since we are using $K_i(\cdot)$ to denote the correlation function.

if $\upsilon_i > n$ [132]. Hence, the Matèrn family allows for more control over the differentiability of the kriging predictor. Furthermore, when $\upsilon_i = \frac{1}{2}$ Eq. (IV.39) reduces to the exponential model, and for $\upsilon_i \to \infty$, Eq. (IV.39) gives the Gaussian model. As compared to the power exponential model, the Matèrn is more complex, and it can be difficult to give analytical expressions for the derivatives of the correlation matrix. Furthermore, the Matèrn model requires more computational time due to the need to evaluate the modified Bessel functions, and this can be disadvantageous in some applications.

Regardless of whether one chooses to use the generalized power exponential family or the Matèrn family for modeling the kriging covariance structure, it should be clear that proper specification of the correlation parameters, $\psi_i$, is critical for successful prediction. These parameters are usually estimated from the available data, and this task has been the focus of much of the geostatistical kriging literature. For metamodeling purposes, however, the approaches for estimating these parameters seem to be somewhat different. We will discuss several of these methods in Section IV.3.D after we first cover some necessary results from Gaussian Process modeling in the next section.

### *IV.3.C The Gaussian Process (GP) Model*

GP models follow a similar development as the kriging predictor described in the previous section. Once again, we regard the output from the computer model realization from the stochastic process given by Eq. (IV.18). The distinction lies in the assumptions regarding the distribution for the stochastic process, $\delta(\mathbf{x})$. In particular, we assume that $\delta(\mathbf{x})$ is Gaussian, with zero mean and covariance given by Eq. (IV.33). Consequently, the model outputs, $\mathbf{y}_D$, for the design configurations, $\mathbf{x}_D^{(i)}$ for $i = \{1,2,\cdots N_D\}$, will be multivariate normally (MVN) distributed with mean, $\mathbf{F}_D\boldsymbol{\beta}$, and covariance, $\boldsymbol{\Sigma}_{DD}$. To be precise, we should say that $\mathbf{y}_D$ is MVN conditional on the parameters, $\boldsymbol{\beta}$, $\sigma^2$, and $\boldsymbol{\psi}$, being known, and we express this as:

$$\mathbf{y}_D \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi} \sim N(\mathbf{F}_D\boldsymbol{\beta}, \boldsymbol{\Sigma}_{DD}) \,. \qquad \text{(IV.40)}$$

We shall explicitly denote all conditional dependencies, such as in Eq. (IV.40), so that the development of the Bayesian prediction methodology in Section IV.5 will, hopefully, be easier to follow. Subsequently, we will refer to the set of parameters, $\boldsymbol{\beta}$, $\sigma^2$, and $\boldsymbol{\psi}$, simply as the GP parameters.

We can extend Eq. (IV.40) to describe the joint distribution of the design points, $\mathbf{y}_D$, and the simulation points, $\mathbf{y}_S$, by making the following definitions:

$$\mathbf{y}=\begin{pmatrix}\mathbf{y}_D\\ \mathbf{y}_S\end{pmatrix}\quad,\quad \mathbf{F}=\begin{bmatrix}\mathbf{F}_D\\ \mathbf{F}_S\end{bmatrix}\quad,\text{and}\quad \boldsymbol{\Sigma}=\begin{bmatrix}\boldsymbol{\Sigma}_{DD} & \boldsymbol{\Sigma}_{SD}^{\mathrm{T}}\\ \boldsymbol{\Sigma}_{SD} & \boldsymbol{\Sigma}_{SS}\end{bmatrix}=\sigma^2\begin{bmatrix}\mathbf{K}_{DD} & \mathbf{K}_{SD}^{\mathrm{T}}\\ \mathbf{K}_{SD} & \mathbf{K}_{SS}\end{bmatrix} \tag{IV.41}$$

where $\mathbf{K}_{DD}$ was defined in Eq. (IV.34a) and $\mathbf{K}_{SS}$ is similarly defined. The $N_S \times N_D$ cross-correlation matrix, $\mathbf{K}_{SD}$, is given by:

$$\mathbf{K}_{SD}=\begin{bmatrix} K\left(\mathbf{x}_S^{(1)},\mathbf{x}_D^{(1)}\mid\boldsymbol{\psi}\right) & K\left(\mathbf{x}_S^{(1)},\mathbf{x}_D^{(2)}\mid\boldsymbol{\psi}\right) & \cdots & K\left(\mathbf{x}_S^{(1)},\mathbf{x}_D^{(N_D)}\mid\boldsymbol{\psi}\right)\\ K\left(\mathbf{x}_S^{(2)},\mathbf{x}_D^{(1)}\mid\boldsymbol{\psi}\right) & K\left(\mathbf{x}_S^{(2)},\mathbf{x}_D^{(2)}\mid\boldsymbol{\psi}\right) & \cdots & K\left(\mathbf{x}_S^{(2)},\mathbf{x}_D^{(N_D)}\mid\boldsymbol{\psi}\right)\\ \vdots & \vdots & \ddots & \vdots\\ K\left(\mathbf{x}_S^{(N_S)},\mathbf{x}_D^{(1)}\mid\boldsymbol{\psi}\right) & K\left(\mathbf{x}_S^{(N_S)},\mathbf{x}_D^{(2)}\mid\boldsymbol{\psi}\right) & \cdots & K\left(\mathbf{x}_S^{(N_S)},\mathbf{x}_D^{(N_D)}\mid\boldsymbol{\psi}\right)\end{bmatrix}. \tag{IV.42}$$

Then, the conditional distribution of $\mathbf{y}$ is also multivariate normal with mean, $\mathbf{F}\boldsymbol{\beta}$, and covariance matrix, $\boldsymbol{\Sigma}$.

Naturally, since $\mathbf{y}_D$ are known, we should be interested in the distribution for $\mathbf{y}_S$ conditional on the known values for $\mathbf{y}_D$. Such a distribution is called the predictive distribution since it is the PDF used to make predictions for new outputs. It is a standard result from multivariate statistics that this conditional distribution is also MVN, with mean and covariance given by (see, e.g., [140]):

$$\mathrm{E}(\mathbf{y}_S\mid\mathbf{y}_D,\boldsymbol{\beta},\sigma^2,\boldsymbol{\psi})=\mathbf{F}_S\boldsymbol{\beta}+\boldsymbol{\Sigma}_{SD}\boldsymbol{\Sigma}_{DD}^{-1}(\mathbf{y}_D-\mathbf{F}_D\boldsymbol{\beta}) \tag{IV.43}$$

$$\mathrm{cov}(\mathbf{y}_S\mid\mathbf{y}_D,\boldsymbol{\beta},\sigma^2,\boldsymbol{\psi})=\boldsymbol{\Sigma}_{S|D}=\boldsymbol{\Sigma}_{SS}-\boldsymbol{\Sigma}_{SD}\boldsymbol{\Sigma}_{DD}^{-1}\boldsymbol{\Sigma}_{SD}^{\mathrm{T}} \tag{IV.44}$$

where $\boldsymbol{\Sigma}_{S|D}$ can be recognized as the Schur complement of $\boldsymbol{\Sigma}_{SS}$ in $\boldsymbol{\Sigma}$. When we discuss Bayesian kriging in Section IV.5, we will make use of the entire predictive distribution; however, for now, it makes sense to use the conditional expectation given by Eq. (IV.43) as the predictor, $\hat{\mathbf{y}}_S$ for the unknown simulation outputs, $\mathbf{y}_S$. We can think of Eq. (IV.43) as a point-estimate for $\mathbf{y}_S$. A comparison with the kriging predictor in Eq. (IV.29) reveals two distinctions.

First, whereas the kriging predictor was formulated for predicting a single output, $y_{\text{o}}$, Eq. (IV.43) can be used for predicting multiple outputs simultaneously. We should note, however, that the kriging predictor can be readily extended to account for multiple simultaneous predictions (see, e.g., [141]). The second distinction is that kriging predictor uses the generalized least-squares estimate for the regression coefficients, $\hat{\boldsymbol{\beta}}$, given by Eq. (IV.30), while Eq. (IV.43) assumes that $\boldsymbol{\beta}$ is known. However, since $\boldsymbol{\beta}$ is generally unknown, a priori, it is necessary to somehow estimate this quantity. One obvious choice would be to use the generalized least-squares estimate, $\hat{\boldsymbol{\beta}}$. Alternatively, one could use the maximum likelihood estimate (MLE). From Eq. (IV.40), the conditional distribution for $\mathbf{y}_D$ is given by:

$$p\left(\mathbf{y}_D \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi}\right) = (2\pi)^{-N_D/2} \det(\boldsymbol{\Sigma}_{DD})^{-\frac{1}{2}} \exp\left\{-\tfrac{1}{2}(\mathbf{y}_D - \mathbf{F}_D\boldsymbol{\beta})^{\mathrm{T}} \boldsymbol{\Sigma}_{DD}^{-1} (\mathbf{y}_D - \mathbf{F}_D\boldsymbol{\beta})\right\}. \tag{IV.45}$$

Recognizing that $\det(\boldsymbol{\Sigma}_{DD}) = \det(\sigma^2 \mathbf{K}_{DD}) = \left(\sigma^2\right)^{N_D} \det(\mathbf{K}_{DD})$, the log likelihood for the GP parameters is:

$$\ell\left(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi}\right) = -\tfrac{1}{2}\left[N_D \ln\left(\sigma^2\right) + \ln\left(\det\left(\mathbf{K}_{DD}\right)\right) + \tfrac{1}{\sigma^2}(\mathbf{y}_D - \mathbf{F}_D\boldsymbol{\beta})^{\mathrm{T}} \mathbf{K}_{DD}^{-1} (\mathbf{y}_D - \mathbf{F}_D\boldsymbol{\beta})\right]. \tag{IV.46}$$

As it turns out, the MLE of $\boldsymbol{\beta}$ is simply the generalized least-squares estimate, $\hat{\boldsymbol{\beta}}$. Furthermore, from Eq. (IV.46), we find that the MLE estimate of $\sigma^2$ is:

$$\hat{\sigma}^2 = \tfrac{1}{N_D}\left(\mathbf{y}_D - \mathbf{F}_D\hat{\boldsymbol{\beta}}\right)^{\mathrm{T}} \mathbf{K}_{DD}^{-1} \left(\mathbf{y}_D - \mathbf{F}_D\hat{\boldsymbol{\beta}}\right). \tag{IV.47}$$

Strictly speaking, both of the MLEs, $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$, are functions of $\boldsymbol{\psi}$. The estimation of $\boldsymbol{\psi}$ is the subject of Section IV.3.D, and will not be considered here.

One advantage of the GP model over the kriging predictor is that, rather than obtaining only a point estimate, $\hat{\mathbf{y}}_S$, of the unknown responses, we obtain the full probability distribution over the plausible values for $\mathbf{y}_S$:

$$\mathbf{y}_S \mid \mathbf{y}_D, \boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi} \sim N\left(\mathbf{F}_S\boldsymbol{\beta} + \mathbf{K}_{SD}\mathbf{K}_{DD}^{-1}(\mathbf{y}_D - \mathbf{F}_D\boldsymbol{\beta})\,,\ \sigma^2\left(\mathbf{K}_{SS} - \mathbf{K}_{SD}\mathbf{K}_{DD}^{-1}\mathbf{K}_{SD}^{\mathrm{T}}\right)\right). \tag{IV.48}$$

Because of this, GP models lend themselves naturally to a Bayesian interpretation. In particular, by specifying prior distributions for the GP parameters one can obtain (in some cases) an expression for the unconditional (i.e., not conditional on the GP parameters) distribution of $\mathbf{y}_S \mid \mathbf{y}_D$. We will discuss this approach in more detail in Section IV.5, but note at this point that, unless there is prior information to suppose otherwise, the expected value of $\mathbf{y}_S \mid \mathbf{y}_D$ is identical to that in Eq. (IV.43) with $\hat{\boldsymbol{\beta}}$ substituted for $\boldsymbol{\beta}$. However, as one might expect, the covariance in Eq. (IV.48) underestimates the true covariance since imperfect knowledge regarding any of the GP parameters will lead to more uncertainty in the predicted responses. In fact, it can be shown that (e.g., see [126,141]) the expression for the MSPE of the kriging estimate in Eq. (IV.31) can be generalized for multiple predictions as:

$$\mathrm{MSPE}(\hat{\mathbf{y}}_S) = \sigma^2 \left\{ \mathbf{K}_{SS} - \mathbf{K}_{SD}\mathbf{K}_{DD}^{-1}\mathbf{K}_{SD}^{\mathrm{T}} + (\mathbf{F}_S - \mathbf{K}_{SD}\mathbf{K}_{DD}^{-1}\mathbf{F}_D)(\mathbf{F}_D^{\mathrm{T}}\mathbf{K}_{DD}^{-1}\mathbf{F}_D)^{-1}(\mathbf{F}_S - \mathbf{K}_{SD}\mathbf{K}_{DD}^{-1}\mathbf{F}_D)^{\mathrm{T}} \right\}. \qquad \text{(IV.49)}$$

From a Bayesian perspective, this expression can be regarded as the covariance of $\mathbf{y}_S \mid \mathbf{y}_D, \sigma^2, \boldsymbol{\psi}$ when no prior information for $\boldsymbol{\beta}$ is available. Comparison with the covariance in Eq. (IV.48) indicates that the last term in Eq. (IV.49) is the increased variance resulting from our uncertainty in $\boldsymbol{\beta}$. Notice that Eq. (IV.49) does not depend on the observed outputs, $\mathbf{y}_D$; in other words, the prediction error for any simulation point is determined by the relative location (in the input space) of that point to each of the design points. This is ultimately a consequence of our use of a stationary correlation function.

### *IV.3.D Estimation of the Correlation Parameters*

Proper estimation of the correlation parameters, $\boldsymbol{\psi}$, is perhaps the biggest challenge in constructing a GP model or kriging predictor, and although several approaches have been suggested (see, e.g., Santner et al. [126]), none are clearly superior. It seems the majority of these approaches are based on some variant of maximum likelihood estimation, although Santner et al. [126] describe one alternative based on cross-validation, as well as a Bayesian alternative that uses the posterior mode of $\boldsymbol{\psi} \mid \mathbf{y}_D$ to estimate $\boldsymbol{\psi}$. Based on some results from a simple study, Santner et al. [126] recommend using either standard MLE or restricted (or marginal) MLE (REML), and we discuss both of these approaches below. Subsequently, Li and Sudjianto [142] proposed a penalized MLE approach which we also summarize.

(i) Standard MLE:

In Eq. (IV.46) we provided the log likelihood function for the GP parameters and discussed the MLEs, $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$, for the regression coefficients and process variance. Substituting these expressions into Eq. (IV.46) gives the following:

$$\ell\left(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2, \boldsymbol{\psi}\right) = -\tfrac{1}{2}\left[N_D \ln\left(\hat{\sigma}^2\right) + \ln\left(\det\left(\mathbf{K}_{DD}\right)\right) + N_D\right] \qquad \text{(IV.50)}$$

which, we can regard as the log likelihood function for $\boldsymbol{\psi}$ provided we recognize that $\hat{\sigma}^2 = \hat{\sigma}^2(\boldsymbol{\psi})$ and $\mathbf{K}_{DD} = \mathbf{K}_{DD}(\boldsymbol{\psi})$ are both functions of $\boldsymbol{\psi}$. The only remaining problem, then, is to find the appropriate $\boldsymbol{\psi}$ that maximizes Eq. (IV.50). Unfortunately, this is easier said than done. In Section IV.3.B, we defined $\boldsymbol{\psi}_i = \{\theta_i, \upsilon_i\}$ for each of the $m$ model inputs. Thus, we are faced with a $2m$-dimensional optimization problem where each iteration requires the inversion of the $N_D \times N_D$ matrix, $\boldsymbol{\Sigma}_{DD}$, and the computation of its determinant; needless to say, if $N_D$ is large, this can be problematic, and it is possible that the computational burden to estimate $\boldsymbol{\psi}$ could exceed that of simply running the computer model that we are attempting to approximate. Clearly, whether this is the case depends on the model being approximated; for instance, if the computer model takes days to run, it is unlikely that computing the MLE of $\boldsymbol{\psi}$ will ever exceed this computational demand, not to mention that in such cases $N_D$ is likely to be quite small.

An additional challenge is that the likelihood functions tend to be quite flat [13,142,143], particularly when the data are limited, and often consist of multiple local optima. Hence, many conventional optimization algorithms, such as those based on Newton's method (see, e.g., Mardia and Marshall [144]), have difficulties locating the global maximum, converging, instead, to one of the numerous local maxima. Consequently, stochastic optimization algorithms have been recommended [14]. Santner et al. [126] and Marrel et al. [14] mention some existing software packages for performing these calculations, and Marrel et al. briefly discuss the relative pros and cons of these algorithms. For this work, we have used the DACE Toolbox for MATLAB®, which is freely available from the developers online [145].

### (ii)Restricted MLE (REML)

We noted in Section IV.2.B that the standard expression for the coefficient of determination, $R^2$, uses a biased formula for the residual variance. Similarly, the MLE for the process variance given by Eq. (IV.46) is biased for exactly the same reason; that is, if $\mathbf{F}_D$ is of full column rank (i.e., all of its $q$ columns are linearly independent), then, after the $N_D$ data in $\mathbf{y}_D$ have been used for estimating the $q$ regression parameters, $\boldsymbol{\beta}$, there remain at most $N_D$-$q$ independent data (i.e., degrees of freedom) that can be used for estimating the process variance. At this point, however, we would like to be a bit more formal in our development.

Recall that $\mathbf{y}_D$ is MVN with mean, $\mathbf{F}_D\boldsymbol{\beta}$, and covariance matrix, $\boldsymbol{\Sigma}_{DD}$. Then, for some matrix, $\mathbf{A}$, which we momentarily consider to be arbitrary, the linear transformation given by $\mathbf{A}\mathbf{y}_D$ is MVN with mean $\mathbf{A}\mathbf{F}_D\boldsymbol{\beta}$ and covariance matrix $\mathbf{A}\boldsymbol{\Sigma}_{DD}\mathbf{A}^{\mathrm{T}}$. In particular, let $\mathbf{A}$ be the $N_D$×$N_D$ matrix given by:

$$\mathbf{A}=\begin{bmatrix}\mathbf{A}_1\\ \cdots\\ \left(\mathbf{F}_D^{\mathrm{T}}\boldsymbol{\Sigma}_{DD}^{-1}\mathbf{F}_D\right)^{-1}\mathbf{F}_D^{\mathrm{T}}\boldsymbol{\Sigma}_{DD}^{-1}\end{bmatrix} \tag{IV.51}$$

where $\mathbf{A}_1$ is some ($N_D$-$q$)×$N_D$ matrix satisfying $\mathbf{A}_1\mathbf{F}_D=\mathbf{0}$. Next, consider the $N_D$×$N_D$ matrix, $\mathbf{B}$, defined as:

$$\mathbf{B}=\left[\boldsymbol{\Sigma}_{DD}\mathbf{A}_1^{\mathrm{T}}(\mathbf{A}_1\boldsymbol{\Sigma}_{DD}^{-1}\mathbf{A}_1^{\mathrm{T}})^{-1}\ \vdots\ \mathbf{F}_D\right]. \tag{IV.52}$$

The product, **AB**, of these matrices is:

$$\mathbf{AB}=\begin{bmatrix}\mathbf{A}_1\boldsymbol{\Sigma}_{DD}\mathbf{A}_1^{\mathrm{T}}(\mathbf{A}_1\boldsymbol{\Sigma}_{DD}^{-1}\mathbf{A}_1^{\mathrm{T}})^{-1} & \mathbf{A}_1\mathbf{F}_{\mathbf{D}}\\ \left(\mathbf{F}_D^{\mathrm{T}}\boldsymbol{\Sigma}_{DD}^{-1}\mathbf{F}_D\right)^{-1}\mathbf{F}_D^{\mathrm{T}}\boldsymbol{\Sigma}_{DD}^{-1}\boldsymbol{\Sigma}_{DD}\mathbf{A}_1^{\mathrm{T}}(\mathbf{A}_1\boldsymbol{\Sigma}_{DD}^{-1}\mathbf{A}_1^{\mathrm{T}})^{-1} & \left(\mathbf{F}_D^{\mathrm{T}}\boldsymbol{\Sigma}_{DD}^{-1}\mathbf{F}_D\right)^{-1}\mathbf{F}_D^{\mathrm{T}}\boldsymbol{\Sigma}_{DD}^{-1}\mathbf{F}_D\end{bmatrix} \tag{IV.53}$$

which, after simplifying using the fact that $\mathbf{A}_1\mathbf{F}_D=\mathbf{0}$, becomes:

$$\mathbf{AB}=\begin{bmatrix}\mathbf{I}_{N_D-q} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}_q\end{bmatrix}=\mathbf{I}_{N_D} \tag{IV.54}$$

where $\mathbf{I}_M$ denotes the $M \times M$ identity matrix to keep track of the sizes of the various block matrices. Since **A** and **B** are both square matrices, **B** is the unique inverse of **A**, so that **AB=BA=I**. Thus, we have the following:

$$\mathbf{BA} = \boldsymbol{\Sigma}_{DD}\mathbf{A}_1^{\mathrm{T}}(\mathbf{A}_1\boldsymbol{\Sigma}_{DD}^{-1}\mathbf{A}_1^{\mathrm{T}})^{-1}\mathbf{A}_1 + \mathbf{F}_D\left(\mathbf{F}_D^{\mathrm{T}}\boldsymbol{\Sigma}_{DD}^{-1}\mathbf{F}_D\right)^{-1}\mathbf{F}_D^{\mathrm{T}}\boldsymbol{\Sigma}_{DD}^{-1} = \mathbf{I} \qquad \text{(IV.55)}$$

which is readily manipulated to give the following useful identity which will be needed momentarily [141]:

$$\mathbf{A}_1^{\mathrm{T}}(\mathbf{A}_1\boldsymbol{\Sigma}_{DD}^{-1}\mathbf{A}_1^{\mathrm{T}})^{-1}\mathbf{A}_1 = \boldsymbol{\Sigma}_{DD}^{-1} - \boldsymbol{\Sigma}_{DD}^{-1}\mathbf{F}_D\left(\mathbf{F}_D^{\mathrm{T}}\boldsymbol{\Sigma}_{DD}^{-1}\mathbf{F}_D\right)^{-1}\mathbf{F}_D^{\mathrm{T}}\boldsymbol{\Sigma}_{DD}^{-1} \qquad \text{(IV.56)}$$

Next, consider the transformation defined by $\mathbf{z} = \mathbf{A}_1\mathbf{y}_D$. Then, **z** will be MVN with mean, $\mathbf{A}_1\mathbf{F}_D\boldsymbol{\beta} = \mathbf{0}$ and $(N_D\text{-}q)\times(N_D\text{-}q)$ covariance matrix, $\mathbf{A}_1\boldsymbol{\Sigma}_{DD}\mathbf{A}_1^{\mathrm{T}}$. We call the log likelihood function based on the distribution of **z** the restricted log likelihood function. Following Eq. (IV.46), the restricted log likelihood function for the GP parameters is:

$$\ell_{\mathrm{R}}\left(\boldsymbol{\beta},\sigma^2,\boldsymbol{\psi}\right) = -\tfrac{1}{2}\left[(N_D - q)\ln\left(\sigma^2\right) + \ln\left(\det\left(\mathbf{A}_1\mathbf{K}_{DD}\mathbf{A}_1^{\mathrm{T}}\right)\right) + \tfrac{1}{\sigma^2}\mathbf{z}^{\mathrm{T}}\left(\mathbf{A}_1\mathbf{K}_{DD}\mathbf{A}_1^{\mathrm{T}}\right)^{-1}\mathbf{z}\right] \qquad \text{(IV.57)}$$

Consequently, the REML for the process variance is given by:

$$\tilde{\sigma}^2 = \frac{\mathbf{z}^{\mathrm{T}}\left(\mathbf{A}_1\mathbf{K}_{DD}\mathbf{A}_1^{\mathrm{T}}\right)^{-1}\mathbf{z}}{N_D - q}\,. \qquad \text{(IV.58)}$$

Since, $\mathbf{A}_1\mathbf{F}_D = \mathbf{0}$, it follows that $\mathbf{A}_1\left(\mathbf{y}_D - \mathbf{F}_D\boldsymbol{\beta}\right) = \mathbf{A}_1\mathbf{y}_D = \mathbf{z}$. Using this result, the numerator in Eq. (IV.58) becomes:

$$\left(\mathbf{A}_1\left(\mathbf{y}_D - \mathbf{F}_D\boldsymbol{\beta}\right)\right)^{\mathrm{T}}\left(\mathbf{A}_1\mathbf{K}_{DD}\mathbf{A}_1^{\mathrm{T}}\right)^{-1}\left(\mathbf{A}_1\left(\mathbf{y}_D - \mathbf{F}_D\boldsymbol{\beta}\right)\right) = \left(\mathbf{y}_D - \mathbf{F}_D\boldsymbol{\beta}\right)^{\mathrm{T}}\left\{\mathbf{A}_1^{\mathrm{T}}\left(\mathbf{A}_1\mathbf{K}_{DD}\mathbf{A}_1^{\mathrm{T}}\right)^{-1}\mathbf{A}_1\right\}\left(\mathbf{y}_D - \mathbf{F}_D\boldsymbol{\beta}\right) \qquad \text{(IV.59)}$$

where we can recognize the quantity in braces on the right-hand side as the left-hand side of Eq. (IV.59). Using this result, and simplifying, we arrive at the following expression for the REML for the process variance [126]:

$$\tilde{\sigma}^2 = \frac{N_D}{N_D - q}\hat{\sigma}^2 = \frac{\left(\mathbf{y}_D - \mathbf{F}_D\boldsymbol{\beta}\right)^{\mathrm{T}}\mathbf{K}_{DD}^{-1}\left(\mathbf{y}_D - \mathbf{F}_D\boldsymbol{\beta}\right)}{N_D - q} \qquad \text{(IV.60)}$$

where $\hat{\sigma}^2$ is the standard MLE from Eq. (IV.47). Finally, we define the REML for the correlation parameters as the values for $\boldsymbol{\psi}$ that maximize:

$$\ell_{\mathrm{R}}\left(\hat{\boldsymbol{\beta}},\tilde{\sigma}^2,\boldsymbol{\psi}\right)=-\tfrac{1}{2}\left[(N_D-q)\ln\left(\tilde{\sigma}^2\right)+\ln\left(\det\left(\mathbf{K}_{DD}\right)\right)+N_D-q\right]. \qquad \text{(IV.61)}$$

After all that, it is worthwhile to pause and consider what we have just done. Basically, since $\mathbf{A}_1\mathbf{F}_D=\mathbf{0}$, the vector given by $\mathbf{z}=\mathbf{A}_1\mathbf{y}_D$ represents the components of $\mathbf{y}_D$ that are orthogonal to, and therefore independent of, the regression model, $\mathbf{F}_D\boldsymbol{\beta}$. These orthogonal components of the data are called generalized increments [141]. Consequently, by restricting ourselves to only using the generalized increments, the estimate of the process variance is increased by a factor of $N_D$ / ($N_D$-$q$) because there are $q$ fewer being used for its estimation. The use of Eq. (IV.61) is, arguably, technically more correct than using Eq. (IV.50) to estimate $\boldsymbol{\psi}$. However, the optimization process is still plagued by the same the same difficulties; in fact, Eq. (IV.61) may present more difficulties since it includes fewer data.

(iii) Penalized MLE:

Penalized MLE is a more recent alternative to the aforementioned methods for estimating the correlation parameters. It was previously mentioned that one of the difficulties encountered in estimating the correlation parameters is that the log likelihood function is often flat near the maximum. Consequently, the MLE for $\boldsymbol{\psi}$ will have a large variance. As a result, the predictors tend to exhibit erratic behavior away from the design points [142]. The penalized MLE approach attempts to alleviate this by adding a penalty function, $\varsigma(\boldsymbol{\psi})$, to the standard log likelihood function in Eq. (IV.46). The resulting penalized likelihood function can be expressed as:

$$\ell\left(\boldsymbol{\beta},\sigma^2,\boldsymbol{\psi}\right)=-\tfrac{1}{2}\left[N_D\ln\left(\sigma^2\right)+\ln\left(\det\left(\mathbf{K}_{DD}\right)\right)+\tfrac{1}{\sigma^2}\left(\mathbf{y}_D-\mathbf{F}_D\boldsymbol{\beta}\right)^{\mathrm{T}}\mathbf{K}_{DD}^{-1}\left(\mathbf{y}_D-\mathbf{F}_D\boldsymbol{\beta}\right)+\varsigma(\boldsymbol{\psi})\right]. \qquad \text{(IV.62)}$$

The details of selecting the penalty function are discussed by Li and Sudjianto [142], and will not be repeated here. Basically, as Li and Sudjianto describe it, the penalized likelihood function reduces the variance in the estimate of $\boldsymbol{\psi}$ by introducing a (hopefully) small bias. They contrast this with the REML approach where the objective is to reduce the bias at the expense of increasing the variance in the estimate of the process variance. Although Li and Sudjianto have reported promising results using the penalized MLE, the method does not currently appear to be widely used; the reason for this is not clear. This method was not used in this work, but is mentioned only for completeness. We note that in their development, Li and Sudjianto have considered only ordinary kriging models (i.e., a constant regression model), but this assumption does not appear to be a necessary restriction.

### *IV.3.E Summary and Discussion*

We have already noted many of the advantages of kriging and GP models over polynomial RSs for metamodeling deterministic computer models. Most notably was the fact that the kriging and GP predictors exactly interpolate the design data, and we stated that one consequence of this feature is that there is a nonzero probability that the metamodel will correctly predict, within an arbitrary degree of accuracy, the simulation outputs for every input, $\mathbf{x}$. Furthermore, the nonparametric nature of kriging and GP models makes them capable of representing more complex behavior than polynomial RSs. For this reason, kriging and GPs are often regarded as superior to polynomial RSs as global predictors [128]; that is, while a low-order polynomial may be sufficient for representing the model output locally (i.e., where a second-order Taylor expansion is adequate), they are incapable of representing higher order effects that may be evident in the data. In particular, a single quadratic RS (QRS) cannot account for multiple optima. We demonstrate this in Fig. 7 where we have illustrated a surface plot of a simple 2-D function, $f(x, y) = -xy\exp\left(-(x^2 + y^2)\right)$. The second pane in Fig. 7 is a contour plot of this function, and the third and fourth panes illustrate the contours approximated using a GP model with a Gaussian covariance function and a QRS, respectively. Both of these metamodels were constructed using the same data, a 25-point uniform Latin hypercube sample, illustrated by the white triangles. From Fig. 7, we see that the GP model predicts the correct contours remarkably well, while the QRS is clearly incapable of replicating the multiple optima. This is likely one of the reasons that kriging and GP models have been so popular to the optimization community, while QRSs have essentially been ignored.

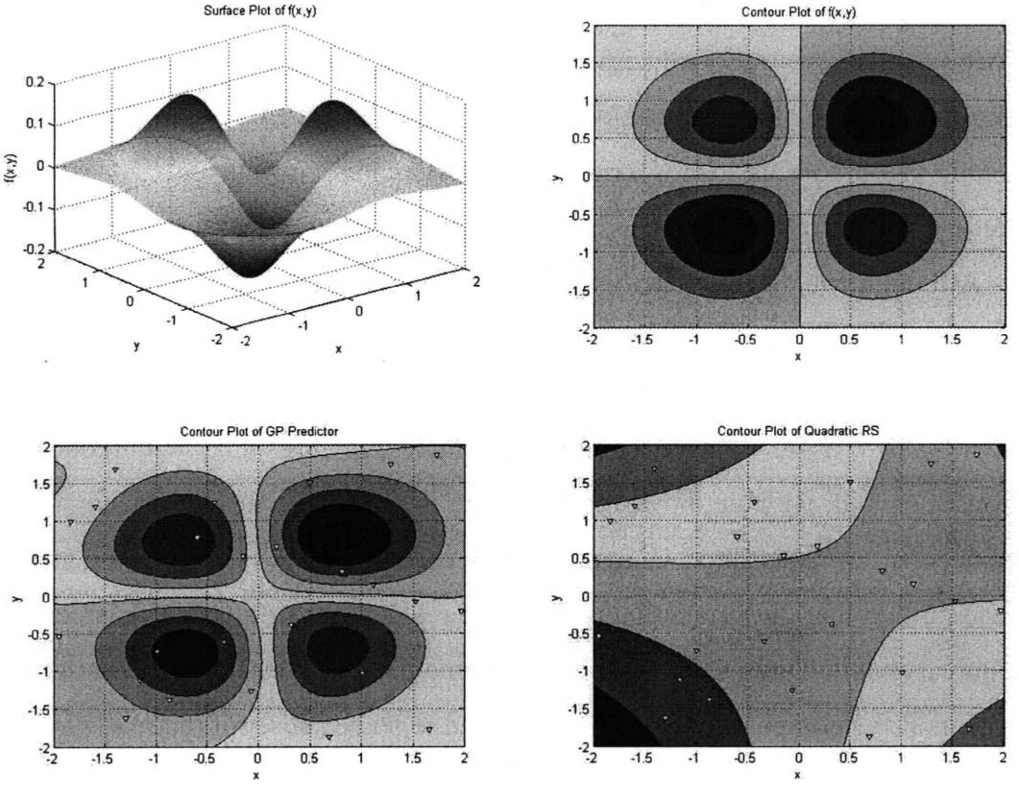

Figure 7: Comparison of GP and QRS for predicting contours of a multimodal function

Despite the advantages just noted, kriging and GP models do suffer from their share of problems. Perhaps the biggest issue is the computational demand required of these metamodels. We noted previously that the construction of the kriging and GP models requires the inversion of the design covariance matrix, $\mathbf{\Sigma}_{DD}$. The number of computations for performing this operation scales with $N_D^3$, so if many design data are available, this computation can become problematic. On the other hand, constructing a polynomial RS only requires the inversion of $\mathbf{F}_D^{\mathbf{T}}\mathbf{F}_D$, which is a $q\times q$ matrix, independent of the number of data; this is clearly one advantage that RSs have over kriging and GPs. In addition to the computational demand, kriging and GPs are susceptible to numerous numerical complications, most of which result from ill conditioning of the covariance matrix. We mentioned that if two design points are in close proximity, then the corresponding columns of the correlation matrix will be very similar; this is one consequence of using of stationary correlation function, and additional consequences will be discussed momentarily. As a result, the covariance matrix will be ill-conditioned and its inversion can lead to devastatingly large numerical errors. In these cases, the numerical uncertainty (which we are not considering) will dominate the prediction uncertainty. Various numerical strategies exist for alleviating this, and Lophaven et al. [136,145] discuss the techniques that they have implemented in their DACE Toolbox. Moreover, these authors discuss some additional tricks to accelerate the computations, such as taking advantage of matrix symmetry and sparsity [136].

Recall, in our discussion of kriging and GP models, we made two assumptions that demand some justification and/or discussion. The first assumption that we will consider is that of normality. Although we noted that this assumption is not strictly necessary for the kriging predictor to be valid, this assumption will be of enormous value when we consider Bayesian prediction in Section IV.5. The reason is that the Normal distribution is extremely easy to deal with and will allow us to develop some analytical expressions that will reduce the overall computational demand; we note, however, that even in the Bayesian case, this assumption is not absolutely necessary. Regardless, in many instances, if there is no prior evidence to support otherwise, assuming that the simulation outputs are realizations of a Gaussian stochastic process is not too unreasonable. Unfortunately, there are some instances where this assumption is clearly invalid, such as if the computer outputs are restricted to some subset of the real numbers (i.e., the outputs must be positive). For instance, if the simulation output is absolute temperature, then clearly this quantity cannot be negative. However, a GP metamodel for such a simulation would not be confined by such restrictions, and it is possible that the metamodel could predict negative temperatures as a result of the normality assumption. Granted, as long as the probability that the metamodel returns a negative temperature is sufficiently low so as to be negligible, the normality assumption may still be acceptable. In some cases, it might be possible to choose the regression model (i.e., the mean of the GP) so that the deviations are approximately normally distributed. If this cannot be done, then we must resort to alternatives. One alternative is to assume that the random process, $\delta(\mathbf{x})$, can be treated as some nonlinear transformation, such as a Box-Cox transformation, of a Gaussian random field. This technique is called Trans-Gaussian kriging and will not be treated in this thesis. However, Christensen et al. [146] provide a brief overview of the method and De Oliviera et al. [147] describe Trans-Gaussian in a Bayesian context.

The second assumption that was made, and possibly the more restrictive of the two, is that the random process be stationary. Recall, this assumption was manifest through our use of correlation functions of the form given by Eq. (IV.36). One consequence of this assumption is that the predictor will be continuous, although not necessarily smooth. On one hand, some degree of continuity is necessary for any metamodeling approach. If the computer model is allowed to be everywhere discontinuous, then every observation would, indeed, be independent. Then, it would not be possible to use the information from some set of observations to infer the values of unobserved responses. On the other hand, if there is a discontinuity in the response that is evident in the design data, then, by virtue of the stationarity assumption, the metamodel will make predictions under the assumption that this discontinuity is present throughout the entire input space. In other words, the metamodel 'sees' the discontinuity, and subsequently assumes that the responses are only very weakly correlated by assigning a small range coefficient in the correlation function. Since this correlation function is independent of location (only differences in inputs), this weak correlation will pervade the entire design space, and the metamodel will predict erratic behavior. Alternatively, even if the model outputs are continuous, there may still be instances where the stationarity assumption is violated. For instance, consider the simple function, $\sin(x^2)$. As $x$ increases, this function oscillates at an increasing 'frequency.' Thus, the correlation between, say $x_1$ and $x_2$, will not be location-independent; that is, if both $x_1$ and $x_2$ are increased while keeping $|x_1-x_2|$ constant, we might expect the correlation to decrease since the function will be oscillating more erratically.

Several approaches have been proposed for handling potential nonstationarity (see, e.g., [148,149]). One of the more promising methods is the use of Treed-GPs, as proposed by Gramacy and Lee [101]. This method was motivated by a problem involving CFD simulations of a spacecraft on reentry. The simulation had to account for flow conditions at a wide range of Mach numbers, and because of the discontinuities that resulted from the transition from subsonic to supersonic flow, the stationarity assumption was clearly invalid. Moreover, Gramacy and Lee [101] noted that, although the model was deterministic, the solver was iterative and used random restarts to improve convergence. As they explain, Treed-GPs are a generalization of the Classification and Regression Tree (CART) model. Basically, the input parameter space is partitioned into multiple subdomains, with each subdomain being represented as a separate branch on a tree. For each branch, a unique GP model is constructed using the design data that correspond to that branch. The details of the branching and partitioning are somewhat involved, and will not be given here. Suffice it to say that Treed-GPs can allow for a different correlation structure on each branch, thereby providing a means for handling nonstationarity. As an added benefit, since the design data is divided between multiple different branches, each GP is constructed from a smaller data set, which greatly reduces the overall computational burden; for example, if there are $b$ different branches, and each GP is constructed from $n/b$ data, the total computational demand will be on the order of $b \times (n/b)^3 = n^3/b^2$. Hence, if $b$ is large, the computational demand will be much less than $n^3$. On the other hand, since each GP is constructed from fewer data, the estimated variance for each may be larger.

## IV.4 Artificial Neural Networks (ANNs)

ANNs are statistical learning devices whose structure is inspired by the functioning of the brain [116]. They turn out to be incredibly versatile, and have been successfully implemented for a vast array of purposes, including data processing, system identification, and pattern recognition. ANNs are composed of many parallel computing units (called neurons or nodes), each of which performs a few simple, nonlinear operations and communicates the results to its neighboring units through a network of weighted connections (called synapses). Given a set of data in the form of I/O mappings (or patterns), the weights are adjusted through a process called training so that the ANN replicates this I/O pattern. Although each neuron is only tasked with a very simple operation, they can be connected in such a way as to be capable of reproducing quite complex I/O patterns; the structure of these connections is called the network topology, or architecture.

In its most basic form, an ANN is composed of the three layers – an input layer, a hidden layer, and an output layer – each consisting of $n_i$, $n_h$, and $n_o$ neurons, respectively. We denote the topology of such a network by ($n_i$ - $n_h$ - $n_o$). To illustrate, Fig. 8 shows a simple (3-4-2) architecture. The ANN in this illustration could represent, for instance, a regression model for the function $\mathbf{y} = g(x_1,x_2,x_3)$, where $\mathbf{y} = \{y_1,y_2\}$. At the input layer, each of the three neurons receives as its signal the value for one of the inputs and subsequently feeds that signal to each of the four neurons in the hidden layer. In addition, there is a node labeled 'bias' whose output is always unity; one can think of these nodes as providing the mean signal (it is similar to the first column of the RS design matrix, $\mathbf{F}_D$, consisting of all ones). Although not indicated in Fig. 8, each of the connections between the various neurons (including the bias) is weighted; that is, the signal that is transmitted from node $1^i$ of the input layer to, say, node $3^h$ of the hidden layer is multiplied by a weight factor, $w^i_{1\rightarrow 3}$. Note that in the following, a superscript $i$, $h$, or $o$ refers, respectively, to the input, hidden, or output layer of the ANN.

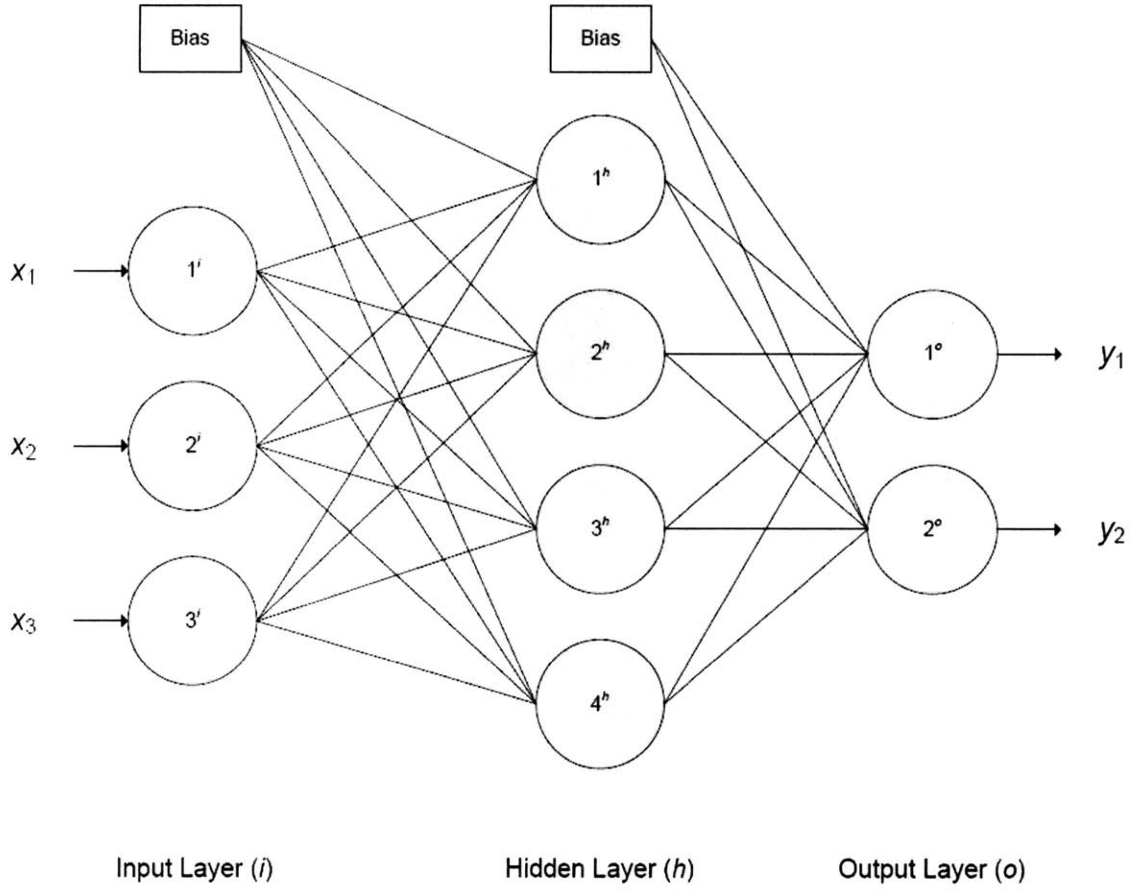

Figure 8: Illustration of a (3-4-2) ANN topology

Figure 9 provides an expanded view of neuron $1^h$ from Fig. 8. The signals that are received by neuron $1^h$ are denoted by $u_{j\to 1} = x_j w^i_{j\to 1}$, with $u_{0\to 1} = w^i_{0\to 1}$ being the signal from the bias neuron. The first operation, denoted by the box marked '+', is to sum all of the incoming signals. The intermediate signal, $z = \sum_{j=0}^{3} u_{j\to 1}$, is taken as the input to the function $S(\cdot)$, whose output is then passed to the two nodes in the output layer. The function, $S(\cdot)$, is a nonlinear function called a sigmoid function, and examples include functions such as $S(z) = \left(1+e^{-z}\right)^{-1}$. The motivation for using these functions comes from a theorem by Cybenko [150] which states, more or less, that any continuous function, $f:[0,1]^n \to \mathbb{R}$, can be approximated, with arbitrary accuracy, by a linear combination of sigmoid functions (see the cited reference for a rigorous statement). The neurons in the output layer can be viewed similarly as in Fig. 9, although generally these nodes do not include the sigmoid transformation.

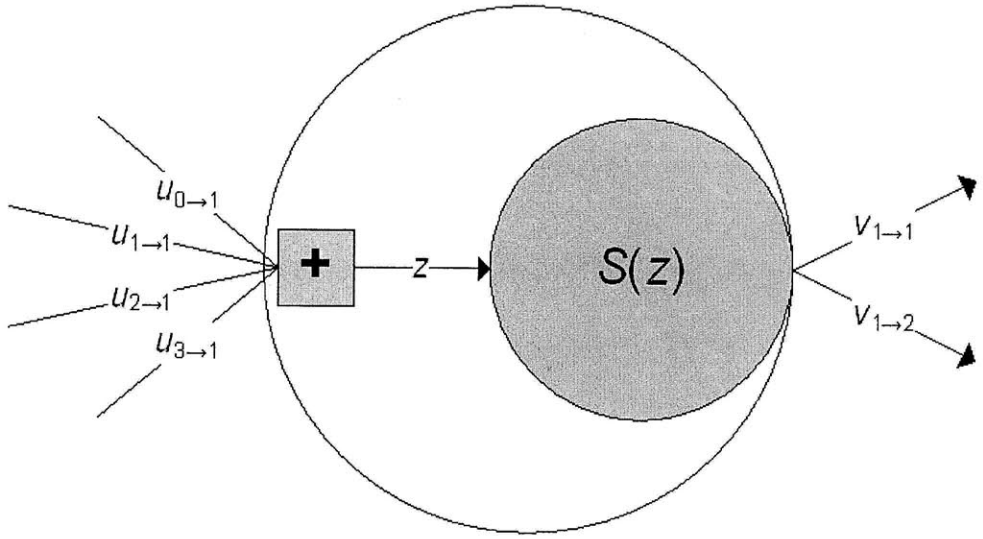

Figure 9: Expanded view of neuron $1^h$ in the hidden layer.

Clearly, the number of neurons in the input and output layers is determined by the function being approximated. However, the number of neurons in the hidden layer is essentially arbitrary. Furthermore, one has the freedom to use multiple hidden layers. It is because of this flexibility that ANNs have been used so successfully for diverse applications. Unfortunately, the choice of a proper ANN architecture is somewhat ambiguous, and it turns out that the architecture is absolutely critical for prediction. If too few hidden neurons are used, the ANN will not be sufficiently flexible to reproduce the training data (i.e., the design data). On the other hand, if too many hidden neurons are included in the architecture, the ANN will be prone to overfitting the training data and will be a poor predictor. Consequently, special precautions are taken during the training phase to prevent overfitting, as described next.

The ANN training process can be performed with any of a variety of nonlinear optimization routines. Often, the RMSE is chosen as the error measure and the training process proceeds by adjusting the various weights to minimize the RMSE. Contrary to RSs, ANNs are usually sufficiently flexible that at some point in the training process, the ANN I/O pattern will exactly match the training data; it will exactly interpolate the data, and the RMSE will be zero. However, unlike the case for kriging and GP models whose formulation provided an expression for the prediction error (i.e., the MSPE), ANNs do not provide such a measure, and their approximation error is estimated from the RMSE. Thus, if the RMSE is zero, there is no way to quantify the uncertainty in making predictions; it is in this context that we say the ANN is overfit to the training data. To prevent this, one usually obtains a secondary set of data, called the validation set, from the computer model. This data is not directly used for training the ANN; rather, it is used to monitor the predictive capability of the ANN during the training process. As the ANN is trained using the design/training data, one also computes the RMSE using the validation set (this is essentially the same as the square of the MSPE). This process is illustrated in Fig. 10, which shows the RMSE of the training data continuously decreasing. Meanwhile, the RMSE for the validation set decreases initially until reaching a minimum value, after which the RMSE begins to increase as the ANN is no longer able to predict the validation data. To prevent this phenomenon, one common strategy for ANN training is to employ an early stopping criterion whereby the training process is terminated once the RMSE for the validation set begins increasing [116]; this is illustrated by the dashed line in Fig. 10. Thus, the early stopping criterion presents a tradeoff between fidelity to the training data and predictive capability.

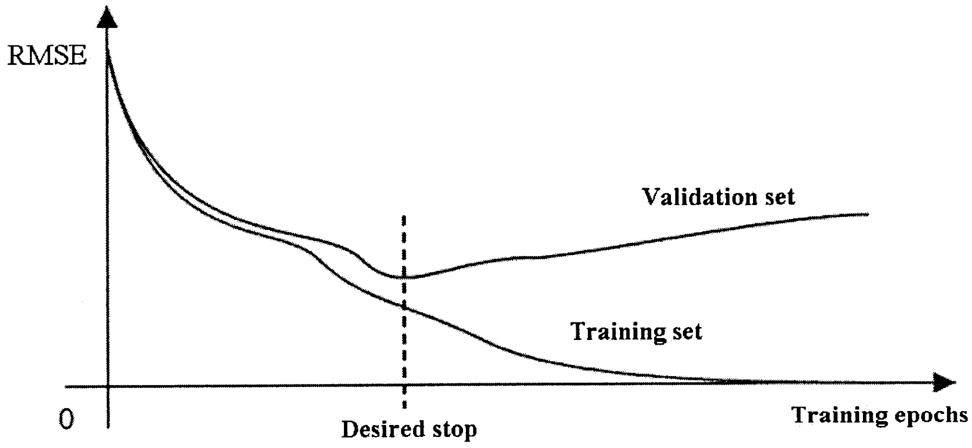

Figure 10: Illustration of the early stopping criterion to prevent overfitting (adapted from [116]).

In summary, ANNs comprise a remarkably flexible class of nonlinear regression models that can be used for modeling highly complex I/O patterns. They are inherently well-suited for dealing with multiple outputs, an advantage over the alternative methods such as RSs and GPs; recall that a separate RS must be constructed for each output, and although the same strategy could be used for GPs, the more correct approach would account for correlations between the different outputs. However, while it should always be possible, in principle, to construct an ANN metamodel with sufficiently complex architecture to exactly interpolate the design data, this is avoided in practice so as to prevent overfitting; in this regard, ANNs are similar to RSs. Moreover, due to their nonlinear nature, ANNs must be trained through a nonlinear optimization algorithm, and this process can be time consuming, depending on the complexity of the ANN topology (i.e., the number of weights to be determined). Finally, ANNs are black-box predictors; they do not admit an analytical formula for the predictor, and the resultant lack of interpretability has been noted as a disadvantage of these methods [14].

## IV.5 Metamodel Uncertainty

Metamodels are, by their very definition, only approximate representations of the computer models they are intended to emulate. Thus, what is gained in computational efficiency is lost, to some degree, in accuracy, and it is necessary to quantify the resultant uncertainty; this is particularly true when the overall goal is uncertainty and sensitivity analysis. In this section, we describe two approaches for quantifying the uncertainty in any estimate obtained with a metamodel. The first approach is based on a statistical technique known as bootstrapping, a distribution-free, brute-force sampling scheme for making statistical inferences from limited data [151-153]. The bootstrap method is used to generate an ensemble of metamodels, each of which is constructed from a different data set bootstrapped from the original design data [15,154]. This ensemble of metamodels is used to correct for any existing bias in the original metamodel, thereby obtaining an improved estimate, called the Bootstrap Bias-Corrected (BBC) estimate, for the quantity of interest (i.e., failure probability, sensitivity coefficients, etc.); we refer to this method as BBC Metamodeling. The second approach is based on a Bayesian extension of the GP models discussed in Section IV.3; no consistent terminology seems to be in use as the technique has been called various names, including Bayesian kriging and hierarchical Gaussian random field modeling. Occasionally, these models are simply called GPs. We will refer to this method as Bayesian kriging so as not to confuse it with the GP models discussed previously.

### *IV.5.A Bootstrap Bias-Corrected (BBC) Metamodeling*

The bootstrap method is a nonparametric statistical inference method that requires no prior knowledge regarding the distribution function of the underlying population being sampled from [153]. To understand how this method works, consider the standard problem of inferring the value of some quantity, $Q$, that describes some population. Standard practice would require one to draw a random sample, say of size $N$, from this population, and then use these data to obtain an estimate, $\hat{Q}$, of the quantity of interest. If the underlying distribution is unknown but $N$ is large, then various asymptotic properties of random samples, such as the central limit theorem, can be utilized to obtain confidence bounds for the estimate, $\hat{Q}$. However, if $N$ is not large, and this is commonly the case, then these theorems are not applicable. In this case, one strategy would be to repeatedly (say, $B$ times) draw samples of size $N$ from the population, obtaining a new estimate, $\hat{Q}$, for each sample; the distribution of these estimates could then be used to construct confidence bounds for the estimate. Clearly, such a strategy would not be effective since, if we have drawn $B$ samples of size $N$, we may as well consider this to be a single sample of size $B$×$N$ for which, presumably, the law of large numbers does apply. Moreover, in many cases it is simply infeasible, or even impossible, to draw more samples than the $N$ data that we started with.

The bootstrap method attempts to sidestep this issue by 'simulating' the process just described. That is, rather than actually drawing $B$ samples from the population, the samples are drawn from the empirical distribution function of the original $N$ samples which serves as an approximation (i.e., a surrogate) to the true distribution function. This procedure amounts to drawing random samples of size $N$ from the original data set, with replacement. In the context of metamodeling, data are bootstrapped from the original design data to construct an ensemble of metamodels, each of which is used to provide a separate estimate for $Q$ [15,154]. From the theory and practice of ensemble empirical models, it can be shown that the estimates given by bootstrapped metamodels is, in general, more accurate than the estimate obtained from a single metamodel constructed from all of the available data [154,155]. A step-by-step description of the bootstrap method as it can be applied to metamodels is as follows [154]:

1. Beginning with a set of design data, denoted as $\mathbf{D}_0 = \{(\mathbf{x}_j, y_j),\ j = 1,2,\ldots N_D\}$, construct a metamodel, $M_0(\mathbf{x}, \boldsymbol{\omega}_0)$, using all of these data. The vector, $\boldsymbol{\omega}_0$, is used to denote the set of all parameters used to define the metamodel.
2. Obtain a point estimate, $\hat{Q}_0$, for the quantity of interest (e.g., a percentile, failure probability, or a sensitivity index) using the metamodel, $M_0(\mathbf{x}, \boldsymbol{\omega})$, as a surrogate for the computer model.

3. Generate $B$ (~500-1000) data sets, $\mathbf{D}_b = \{(\mathbf{x}_j, y_j),\ j = 1,2,\ldots N_D\}$, $b$ = 1,2,...$B$, by bootstrapping the design data $\mathbf{D}_o$. In other words, randomly sample $N_D$ pairs, ($\mathbf{x}$, $y$), from the data in $\mathbf{D}_o$ with replacement to construct $\mathbf{D}_1$. Repeat this $B$ times to obtain $\mathbf{D}_2$, $\mathbf{D}_3$, ...$\mathbf{D}_B$. Notice that because the sampling is performed with replacement, for any given data set, $\mathbf{D}_b$, some of the original design data will be duplicated and others will be left out.
4. Build an ensemble of $B$ metamodels, $\{M_b(\mathbf{x}, \boldsymbol{\omega}_b),\ b = 1,2,\ldots B\}$, by constructing a separate metamodel for each of the $B$ bootstrapped data sets, $\mathbf{D}_b$, $b$ = 1,2,...$B$.
5. Using each of the $B$ metamodels as a surrogate to the computer model, obtain a set of $B$ point estimates, $\hat{Q}_b$, $b$ = 1,2,...$B$, for the quantity of interest; by so doing, a bootstrap-based empirical PDF for the quantity $Q$ is generated, which is the basis for constructing the corresponding confidence intervals.
6. Compute the bootstrap sample average, $\hat{Q}_{boot} = \frac{1}{B}\sum_{b=1}^{B} \hat{Q}_b$, from the estimates obtained in Step 5.
7. Using the bootstrap average, calculate the so-called Bootstrap Bias-Corrected (BBC) point estimate, $\hat{Q}_{BBC}$, from:

$$\hat{Q}_{BBC} = 2\hat{Q}_o - \hat{Q}_{boot} \qquad \text{(IV.63)}$$

where $\hat{Q}_o$ is the point estimate obtained with the metamodel, $M_o(\mathbf{x}, \boldsymbol{\omega}_o)$, constructed from the full design data set, $\mathbf{D}_o$ (Steps 1 and 2 above). The BBC estimate $\hat{Q}_{BBC}$ is taken as the final point estimate for $Q$, and its motivation is as follows: It can be demonstrated that if the bootstrap average estimate $\hat{Q}_{boot}$ is biased with respect to the estimate, $\hat{Q}_o$, then $\hat{Q}_o$ will be similarly biased with respect to the true value $Q$ [156]. Thus, in order to obtain a bias-corrected estimate, $\hat{Q}_{BBC}$, of the quantity of interest, $Q$, the estimate, $\hat{Q}_o$, must be adjusted by subtracting the corresponding bias ($\hat{Q}_{boot}$- $\hat{Q}_o$). Therefore, the appropriate expression for the bias-corrected estimate is $\hat{Q}_{BBC}$ = $\hat{Q}_o$ - ($\hat{Q}_{boot}$ - $\hat{Q}_o$) = $2\hat{Q}_o$ - $\hat{Q}_{boot}$.

8. Calculate the two-sided BBC 100·(1 - $\alpha$)% Confidence Interval (CI) for the BBC point estimate in Eq. (IV.63) as follows:

   a. Sort the bootstrap estimates $\hat{Q}_b$, $b$ = 1, 2, ..., $B$, in ascending order so that $\hat{Q}_{(i)} = \hat{Q}_b$ for some $b \in \{1,2,\ldots B\}$, and $\hat{Q}_{(1)} < \hat{Q}_{(2)} < \ldots < \hat{Q}_{(b)} < \ldots < \hat{Q}_{(B)}$.

   b. Then, the $[B \times \alpha / 2]^{\text{th}}$ and $[B \times (1-\alpha/2)]^{\text{th}}$ elements in the ordered list, $\hat{Q}_{(1)} < \hat{Q}_{(2)} < \ldots < \hat{Q}_{(b)} < \ldots < \hat{Q}_{(B)}$, correspond to the 100·$\alpha$/2th and 100·(1 – $\alpha$/2)th quantiles of the bootstrapped empirical PDF of $Q$, and we denote these elements by $\hat{Q}_{([B\cdot\alpha/2])}$ and $\hat{Q}_{([B\cdot(1-\alpha/2)])}$, respectively. Here, [·] stands for "closest integer".

   c. Calculate the two-sided BBC 100·(1 - $\alpha$)% CI for $\hat{Q}_{BBC}$ as:

$$\left[\, \hat{Q}_{BBC} - (\hat{Q}_{boot} - \hat{Q}_{([B\cdot\alpha/2])}),\ \hat{Q}_{BBC} + (\hat{Q}_{([B\cdot(1-\alpha/2)])} - \hat{Q}_{boot}) \,\right]. \qquad \text{(IV.64)}$$

The advantages of using the bootstrap method to supplement a metamodel are two-fold: first, the bootstrap procedure provides an estimate of the existent bias in the metamodel predictions, which can then be readily corrected for. Secondly, through generating an empirical distribution function of the point estimates, $\hat{Q}_b$, the bootstrap method allows for a straightforward estimation of confidence intervals. Furthermore, all of this is done without resorting to any assumptions regarding the distribution of model outputs or the distribution of residuals. On the other hand, since multiple metamodels have to be constructed, the computational cost can be quite high if the metamodel construction process (i.e., the training procedure) is computationally intensive. Once again, it seems we have been had by the free-lunch principle; although metamodels are intended to reduce the computational burden by serving as a quick-running surrogate to more complex computer models, to use them effectively, we are forced to resort to computationally expensive procedures to quantify their accuracy. Fortunately, for many practical problems, it seems unlikely that the added computational expense presented by BBC metamodeling will exceed the cost of brute-force MCS using the original computer model.

### *IV.5.B Bayesian Kriging*

In our discussion of GP models in Section IV.3.C, we obtained an expression for the predictive distribution of $\mathbf{y}_S$, conditional on the observations, $\mathbf{y}_D$, assuming that the GP parameters, $\boldsymbol{\beta}$, $\sigma^2$, and $\boldsymbol{\psi}$ were perfectly known. In practice, however, these parameters are rarely known a priori, so we must somehow estimate them. In Sections IV.3.C and IV.3.D, we discussed various point estimation techniques based on maximum likelihood estimation to approximate these parameters. However, since the GP parameters are themselves uncertain, the prediction uncertainty of these metamodels will be underestimated. In this section, we will frame the problem in a fully Bayesian perspective to account for this additional uncertainty. While we cannot say who was the first to consider a Bayesian interpretation of kriging, one of the earliest articles that we have found is that by Kitanidis [141]; we recommend this article as an excellent starting point for anyone investigating Bayesian kriging.

The objective of Bayesian kriging is to obtain the predictive distribution for $\mathbf{y}_S \mid \mathbf{y}_D$ that is not conditional on any of the GP parameters. To achieve this goal, the first step is to assign an appropriate joint prior distribution to these parameters. Generally speaking, the GP parameters can be assigned any legitimate prior probability distribution, provided that it is consistent with the information that is available. However, the integration required to remove the conditioning can easily become intractable so that an analytical expression for the predictive distribution cannot be obtained. For many practical problems, there will be very little prior information regarding the values of the GP parameters (or, at least, there will be no clear way to translate the available information into a statement regarding the values of these parameters). Hence, a good starting point for Bayesian kriging is to assign a so-called diffuse, or noninformative, prior, such as the Jeffreys prior [157,158]:

$$\pi(\boldsymbol{\beta},\sigma^2,\boldsymbol{\psi}) \propto \frac{\pi(\boldsymbol{\psi})}{\sigma^2} \tag{IV.65}$$

where $\pi(\boldsymbol{\psi})$ denotes the prior distribution for the correlation parameters, which we shall currently leave unspecified. Notice that we have denoted the relationship in Eq. (IV.65) as a proportionality, rather than an equality, since this distribution does not integrate to unity. In fact, this PDF is improper since its integral is divergent.

Recall from Section IV.3.C that the fundamental assumption for GP metamodels is that the model outputs, $\mathbf{y}_D$, evaluated at the design points are multivariate normally distributed, conditional on the GP parameters:

$$\mathbf{y}_D \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi} \sim N(\mathbf{F}_D\boldsymbol{\beta}, \, \boldsymbol{\Sigma}_{DD}) \, , \tag{IV.66}$$

where $\mathbf{F}_D$ and $\boldsymbol{\Sigma}_{DD}$ have been defined, respectively, in Eqs. (IV.4) and (IV.27). Furthermore, it was determined that the distribution of the simulation outputs, $\mathbf{y}_S$, (i.e., the outputs that we wish to simulate with MCS), conditional on the observed design points, $\mathbf{y}_D$, is also multivariate normal:

$$\mathbf{y}_S \mid \mathbf{y}_D, \boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi} \sim N\left(\mathbf{F}_S\boldsymbol{\beta} + \boldsymbol{\Sigma}_{SD}\boldsymbol{\Sigma}_{DD}^{-1}(\mathbf{y}_D - \mathbf{F}_D\boldsymbol{\beta}) \, , \, \boldsymbol{\Sigma}_{SS} - \boldsymbol{\Sigma}_{SD}\boldsymbol{\Sigma}_{DD}^{-1}\boldsymbol{\Sigma}_{SD}^{\mathbf{T}}\right). \tag{IV.67}$$

The expression that we seek is the predictive distribution, $p(\mathbf{y}_S \mid \mathbf{y}_D)$; that is, the PDF for $\mathbf{y}_S$, conditional on the design points, $\mathbf{y}_D$, but marginalized with respect to the GP parameters. From the rules of conditional probability, this distribution is obtained from:

$$p(\mathbf{y}_S \mid \mathbf{y}_D) = \int p(\mathbf{y}_S \mid \mathbf{y}_D, \boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi}) \, \pi(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi} \mid \mathbf{y}_D) \, d\boldsymbol{\beta} \, d\sigma^2 \, d\boldsymbol{\psi} \, , \tag{IV.68}$$

where the integration is taken over the entire domain of the GP parameters. The first term on the right-hand side of Eq. (IV.68), $p(\mathbf{y}_S \mid \mathbf{y}_D, \boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi})$, is known from Eq. (IV.67). From Bayes' Theorem, the posterior distribution of the GP parameters is given by:

$$\pi(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi} \mid \mathbf{y}_D) = \frac{p(\mathbf{y}_D \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi}) \, \pi(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi})}{\int p(\mathbf{y}_D \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi}) \, \pi(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi}) d\boldsymbol{\beta} \, d\sigma^2 \, d\boldsymbol{\psi}} \tag{IV.69}$$

Recognizing that the denominator in Eq. (IV.69) is simply a normalizing constant, we can simplify this expression as:

$$\pi(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi} \mid \mathbf{y}_D) \propto p(\mathbf{y}_D \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi}) \, \pi(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi}) \propto p(\mathbf{y}_D \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi}) \frac{\pi(\boldsymbol{\psi})}{\sigma^2} \, , \tag{IV.70}$$

where we have used the prior distribution given in Eq. (IV.65). Combining Eqs. (IV.68) and (IV.70), the desired predictive distribution can be obtained from:

$$p(\mathbf{y}_S \mid \mathbf{y}_D) \propto \int p(\mathbf{y}_S \mid \mathbf{y}_D, \boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi})\, p(\mathbf{y}_D \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi}) \frac{\pi(\boldsymbol{\psi})}{\sigma^2} d\boldsymbol{\beta}\, d\sigma^2\, d\boldsymbol{\psi} \ . \tag{IV.71}$$

Thus far, we have not specified the prior distribution, $\pi(\boldsymbol{\psi})$, for the correlation parameters. We shall return to this momentarily, but for now we shall take one step backwards and retain the conditioning on $\boldsymbol{\psi}$. Then, from Eq. (IV.65) we have:

$$\pi(\boldsymbol{\beta}, \sigma^2 \mid \boldsymbol{\psi}) \propto \frac{1}{\sigma^2} \ . \tag{IV.72}$$

Repeating the above arguments, one finds that $p(\mathbf{y}_S \mid \mathbf{y}_D, \boldsymbol{\psi})$ is obtained from:

$$p(\mathbf{y}_S \mid \mathbf{y}_D, \boldsymbol{\psi}) \propto \int p(\mathbf{y}_S \mid \mathbf{y}_D, \boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi})\, p(\mathbf{y}_D \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{\psi}) \frac{1}{\sigma^2} d\boldsymbol{\beta}\, d\sigma^2 \ . \tag{IV.73}$$

It turns out (see, e.g., [126,158]) that the integration in Eq. (IV.73) can be carried out analytically, and in doing so, it is found that the predictive distribution is a shifted multivariate student distribution with $N_D - q$ degrees of freedom, where $q$ is number of unknown regression coefficients (i.e., the length of $\boldsymbol{\beta}$). That is, if $\boldsymbol{\mu}_{S|D}$ represents the posterior mean of $\mathbf{y}_S$, then:

$$\mathbf{y}_S - \boldsymbol{\mu}_{S|D} \ \sim \ t_{N_S}\left(N_D - q, \tilde{\boldsymbol{\Sigma}}_{S|D}\right) \tag{IV.74}$$

where the posterior mean and covariance matrix are, respectively, given by:

$$\boldsymbol{\mu}_{S|D} = \mathbf{F}_S \hat{\boldsymbol{\beta}} + \mathbf{K}_{SD}\mathbf{K}_{DD}^{-1}(\mathbf{y}_D - \mathbf{F}_D\hat{\boldsymbol{\beta}}) \tag{IV.75}$$

$$\tilde{\boldsymbol{\Sigma}}_{S|D} = \tilde{\sigma}^2 \left\{ \mathbf{K}_{SS} - \mathbf{K}_{SD}\mathbf{K}_{DD}^{-1}\mathbf{K}_{SD}^{\mathrm{T}} + (\mathbf{F}_S - \mathbf{K}_{SD}\mathbf{K}_{DD}^{-1}\mathbf{F}_D)(\mathbf{F}_D^{\mathrm{T}}\mathbf{K}_{DD}^{-1}\mathbf{F}_D)^{-1}(\mathbf{F}_S - \mathbf{K}_{SD}\mathbf{K}_{DD}^{-1}\mathbf{F}_D)^{\mathrm{T}} \right\} \tag{IV.76}$$

and with:

$$\hat{\boldsymbol{\beta}} = (\mathbf{F}_D^{\mathrm{T}}\mathbf{K}_{DD}^{-1}\mathbf{F}_D)^{-1}\mathbf{F}_D^{\mathrm{T}}\mathbf{K}_{DD}^{-1}\mathbf{y}_D \tag{IV.77}$$

$$\tilde{\sigma}^2 = \tfrac{1}{N_D - q}\, \mathbf{y}_D^{\mathrm{T}} \left[ \mathbf{K}_{DD}^{-1} - \mathbf{K}_{DD}^{-1}\mathbf{F}_D(\mathbf{F}_D^{\mathrm{T}}\mathbf{K}_{DD}^{-1}\mathbf{F}_D)^{-1}\mathbf{F}_D^{\mathrm{T}}\mathbf{K}_{DD}^{-1} \right] \mathbf{y}_D \, . \tag{IV.78}$$

Notice that the posterior mean in Eq. (IV.75) is identical to the kriging predictor given by Eq. (IV.29), but with $\mathbf{f}_o^{\mathrm{T}}$ replaced by $\mathbf{F}_S$ and the product, $\boldsymbol{\sigma}_o^{\mathrm{T}}\boldsymbol{\Sigma}_{DD}^{-1}$ replaced by $\hat{\sigma}^2\mathbf{K}_{SD}\mathbf{K}_{DD}^{-1}$. Moreover, using Eq. (IV.77), we arrive at the following equivalent expression for Eq. (IV.78):

$$\tilde{\sigma}^2 = \frac{\left(\mathbf{y}_D - \mathbf{F}_D\hat{\boldsymbol{\beta}}\right)^{\mathrm{T}} \mathbf{K}_{DD}^{-1} \left(\mathbf{y}_D - \mathbf{F}_D\hat{\boldsymbol{\beta}}\right)}{N_D - q}, \tag{IV.79}$$

which, upon comparing with Eq. (IV.60), we see is identical to the REML estimate for $\sigma^2$.

From the above observations, we see that, with the diffuse Jeffreys prior assigned to $\boldsymbol{\beta}$ and $\sigma^2$, the expected value of the predictive distribution is equivalent to the kriging predictor with the regression coefficients, $\boldsymbol{\beta}$, given by the generalized least-squares estimate (equivalently, the MLE) and the process variance, $\sigma^2$, given by the restricted MLE. That is to say, the diffuse Jeffreys prior has no influence on the expected predictions (i.e., $\boldsymbol{\mu}_{S|D}$) as one would hope. In other words, $\boldsymbol{\mu}_{S|D}$ is not influenced by prior beliefs, demonstrating that the Jeffreys prior is, indeed, noninformative. Thus, the expected predictions are solely determined by the available data. Occasionally, however, prior information regarding $\boldsymbol{\beta}$ and $\sigma^2$ may be available to the analyst. In these cases, the above analysis would not be applicable, strictly speaking, since the Jeffreys prior neglects this information. Through a similar, albeit more complicated, analysis based on conjugate priors, one can obtain a distribution analogous to Eq. (IV.74) that accounts for any available prior information. It turns out that the predictive distribution is still a shifted multivariate student distribution, but the expressions for the mean, covariance matrix, and degrees of freedom differ from those given above. We shall not discuss this here, and the interested reader should refer to [126,141,158] for more details.

Notice that the predictive distribution in Eq. (IV.74) is an expression for the joint PDF of the simulation outputs, $\mathbf{y}_S$, for the input configurations, $\mathbf{x}_S^{(i)}$ for $i = \{1,2,\cdots N_S\}$, and suppose that this set of inputs consists of every possible input configuration (in general, this would require $N_S$ to be infinite). That is, if $D_X$ represents the entire input space, then the set $\{\mathbf{x}_S^{(i)}\} = D_X$. Then, Eq. (IV.74) gives a distribution over a restricted[§] set of all possible functions that interpolate the observed data, $\mathbf{y}_D$. Furthermore, the computer model, itself, can be represented as some function, $\mathbf{y} = \mathbf{g}(\mathbf{x})$, that interpolates these data. Thus, if we were to somehow sample functions from Eq. (IV.74) (see Oakley and O'Hagan [13] for one approach), it

[§] The class of functions supported by the distribution in Eq. (IV.74) is determined by the correlation function and its parameters. For instance, assuming a Gaussian correlation function restricts this class to the set of all real-valued analytic functions (i.e., functions that are infinitely differentiable). In other words, choosing a Gaussian correlation function assigns all of the prior probability to this class of functions.

is possible that one such sample, say $\tilde{\mathbf{g}}(\mathbf{x})$, would, in fact, be $\mathbf{g}(\mathbf{x})$. More precisely, $\tilde{\mathbf{g}}(\mathbf{x})$ would approximate $\mathbf{g}(\mathbf{x})$ to arbitrary accuracy such that, for any $\delta > 0$, $|\mathbf{g}(\mathbf{x}) - \tilde{\mathbf{g}}(\mathbf{x})| < \delta$ for all $\mathbf{x} \in D_X$. Eq. (IV.74) expresses our current state of knowledge regarding those functions that are plausibly representative of the actual computer model (i.e., $\mathbf{g}(\mathbf{x})$); that is, Eq. (IV.74) is a distribution over a set of plausible alternative models. This provides a very intuitive representation of metamodel uncertainty since we now have a set of functions/models, of which one is (hopefully) the model $\mathbf{g}(\mathbf{x})$ that we seek. Then, by performing more simulations with the computer model, we are updating our state of knowledge so as to refine this set of possible models. If this process were continued indefinitely, we should eventually obtain a distribution with all of the probability concentrated on $\mathbf{g}(\mathbf{x})$.

The results we have presented thus far are conditional on the values of the correlation parameters, $\boldsymbol{\psi}$. In principle, we could simply choose some distribution, $\pi(\boldsymbol{\psi})$, and carry out the integration in Eq. (IV.71) to obtain an expression for the predictive distribution, $p(\mathbf{y}_S \mid \mathbf{y}_D)$. Unfortunately, this integration is apparently intractable, since it seems that nobody has been able to develop an analytical expression for the predictive distribution that accounts for correlation parameter uncertainty. One alternative is to use the distribution in Eq. (IV.74), but with $\boldsymbol{\psi}$ replaced with some point-estimate, $\hat{\boldsymbol{\psi}}$ (e.g., the MLE); this approach is referred to as empirical Bayes estimation, or empirical Bayesian kriging [137,157]. While this removes many of the difficulties associated with a fully Bayesian approach, it is recognized that empirical Bayes estimation underestimates the prediction uncertainty. On the other hand, some authors, such as Kennedy and O'Hagan [19], have suggested that the uncertainty in the correlation parameters may not be that important. We do not necessarily endorse this view, as the effect of this uncertainty on predictions seems specific to a given problem.

As an added difficulty, Berger et al. [138] note that when little prior information is available regarding the correlation parameters, many of the common choices for diffuse priors, such as the Jeffreys prior, can yield an improper posterior distribution. Although improper distributions are frequently used as priors, an improper posterior distribution is not acceptable since, to make predictions with the posterior distribution, it must be a legitimate (i.e., proper) PDF; the prior PDF only appears indirectly in the analysis, and therefore need not be proper. Due to this finding, Berger et al. [138] developed an alternative prior, which they call the reference prior, that always yields a proper posterior distribution. We note, however, that these considerations seem to be applicable only if $\psi$ is a scalar (i.e., if there is only one uncertain correlation parameter) since, according to Paulo [159], the problem of posterior impropriety does not extend to higher dimensions (i.e., for $\psi$ a vector).

For the purpose of making predictions, we require a method for drawing samples from the predictive distribution, $p(\mathbf{y}_S \mid \mathbf{y}_D)$. However, without an explicit expression for this distribution, the only sampling approach that seems applicable is Markov Chain Monte Carlo (MCMC) simulation. Some details regarding the application of MCMC to Bayesian kriging are given by Banerjee et al. [119], and we shall not elaborate here. In addition, Paolo [159] discusses an MCMC algorithm that is applicable for the diffuse priors that he suggests, and Agarwal and Gelfand [160] describe an advanced MCMC technique called Slice Sampling; for further details on Slice Sampling, Prof. Radford Neal provides several technical reports available from his University of Toronto webpage, http://www.cs.toronto.edu/~radford/. As one might suspect, MCMC greatly increases the computational overhead required by Bayesian kriging. However, this added cost seems to be on par with the bootstrapping method discussed in the previous section. On the other hand, the implementation of the MCMC techniques is far more sophisticated, and it seems that there are no existing and/or readily available software packages for performing fully Bayesian kriging. For these reasons, this approach was not pursued for this research. We note, however, that conceptually, Bayesian kriging seems to be a very promising approach to metamodeling and is certainly an important subject for future research.

# V THE CASE STUDIES

In this chapter, we present results from two case studies that were carried out to compare the relative performance of the metamodeling methods discussed in Chapter IV. Both of these case studies involve the reliability assessment of a passive nuclear safety system and, although the two studies focus on different systems for different types of reactors, they are similar in that both are natural circulation cooling systems. The reason for focusing on passive system reliability assessment is that such studies often require the simulation of complex thermal-hydraulic (T-H) phenomena using sophisticated computer models that can require several hours, or even days, to run. Furthermore, these simulations must often be performed hundreds, if not thousands, of times for uncertainty propagation. Thus, passive system reliability assessment is a good arena for metamodeling.

The first case study that we consider involves a natural convection core cooling system in a Gas-cooled Fast Reactor under post-LOCA (Loss of Coolant Accident) conditions [29]. The problem consists of propagating epistemic uncertainties through a thermal-hydraulic (T-H) model to estimate the probability that the passive safety system will fail to perform its intended function (e.g., that the natural convection cooling will be insufficient to prevent core melting or other structural failures). Such a failure is referred to as a 'functional failure' in the literature [25-31,161]. The primary motivation for selecting this cases study was that the model is of sufficient simplicity to present a minimal computational burden while still maintaining the essential features of any passive system reliability study. As a result, it was possible to perform a standard MCS to obtain 'true' estimates of the relevant quantities (e.g., percentiles, failure probability, and sensitivity indices) with which to compare the performance of the metamodels. That being said, a MCS with $2.5\times10^5$ samples still required just over one week (approximately 8 days) of continuous CPU time on a standard desktop computer (the run time for a single simulation was 2-3 seconds). Section V.1 presents the results from this study.

The second case study that we discuss considers the performance of two passive decay heat removal systems, the Reactor Vessel Auxiliary Cooling System (RVACS) and the Passive Secondary Auxiliary Cooling System (PSACS), from a lead-cooled Flexible Conversion Ratio Reactor (FCRR). This case study, which is based on the reference study of Fong et al. [31], investigates the capability of RVACS and PSACS to limit the peak clad temperature (PCT) during a simulated station black-out (SBO) event. A model of the system was previously developed using RELAP5-3D [168]. Due to the high thermal capacity of the lead coolant, the simulation needed to be performed for the entire 72-hour duration of the SBO transient; consequently, a single simulation required approximately 30 hours of CPU time. Thus, a standard MCS presents an exceedingly large computational burden and alternative approaches to UA and SA are, therefore, necessary. In fact, it was this system that provided the initial motivation to investigate the use of metamodels for uncertainty and sensitivity analysis. The results from this study are presented in Section V.2.

## V.1 The Gas-Cooled Fast Reactor (GFR) Case Study

In the following sections, we discuss the results from a comparative evaluation of Bootstrap Bias-Corrected (BBC) RSs, ANNs, and kriging/GPs[**] for estimating the outputs from a thermal-hydraulic (T-H) simulation model of a natural circulation decay heat removal (DHR) system for a 600-MW Gas-cooled Fast Reactor (GFR). Specifically, these metamodels are compared based upon their ability to estimate (i) percentiles of the simulation output, (ii) the functional failure probability of the DHR system, and (iii) the Sobol' sensitivity indices[††] for each of the model inputs; these results are discussed in Section V.1.D. In Sections V.1.A-V.1.C, we present a brief description of the DHR system for the GFR and give an overview of the relevant model uncertainties and system failure criteria. Finally, Section V.1.E summarizes the main conclusions from this case study.

### *V.1.A Description of the System*

The system under consideration is a 600-MW Gas-cooled Fast Reactor (GFR) cooled by helium flowing through separate channels in a silicon carbide matrix core. The design of the GFR has been the subject of study for several years at the Massachusetts Institute of Technology (MIT) [162-165]. In these studies, the possibility of using natural circulation to remove the decay heat in case of an accident is investigated. In particular, in case of a LOCA, long-term heat

[**] In the following, we will not be distinguishing between GPs and kriging predictors. We will simply refer to this type of metamodel as a GP.

[††] GPs were excluded from this portion of the study due to time constraints.

removal is ensured by natural circulation in a given number, $N_{loops}$, of identical and parallel heat removal loops. Figure 11 provides a schematic of a single loop from the GFR decay heat removal system. The flow path of the helium coolant is indicated by the black arrows. As shown in the figure, the loop has been divided into $N_{sections}$ = 18 sections; technical details about the geometrical and structural properties of these sections are given in [29] and are not reported here for brevity.

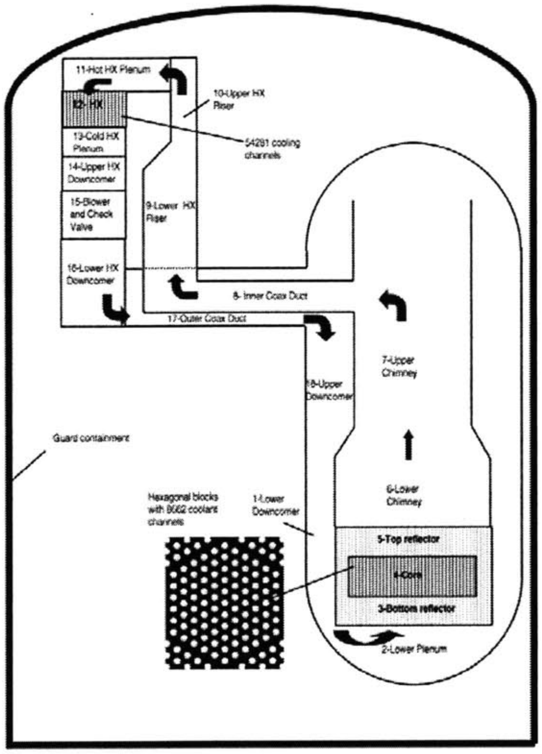

Figure 11: Schematic representation of one loop of the 600-MW GFR passive decay heat removal system [29]

In the present analysis, the average core power to be removed is assumed to be 12 MW, equivalent to about 2% of full reactor power (600 MW). To guarantee natural circulation cooling at this power level, it is necessary to maintain an elevated pressure following the LOCA. This is accomplished by a guard containment (see Fig.11) that surrounds the reactor vessel and power conversion unit and maintains the requisite pressure following the depressurization of the primary system. It has been determined that a nominal pressure of 1650 kPa is required to maintain sufficient heat removal capacity for this system [29]. Furthermore, the secondary side of the heat exchanger (labeled as item 12 in Fig. 11) is assumed to have a nominal wall temperature of 90 °C [29].

It should be noted that, as in [29], the subject of the present analysis is the quasi-steady-state natural circulation cooling that takes place after the LOCA has occurred. Thus, the analyses reported hereafter refer to this steady-state period and are conditional on the successful inception of natural circulation. To simulate the steady-state behavior of the system, a one-dimensional thermal-hydraulic model, implemented in MATLAB®, has been developed at MIT; specific details regarding this model are given by Pagani, et al. [29]. In brief, the code balances the pressure losses around the loops that friction and form losses are exactly balanced by the buoyancy force. The code treats all the $N_{loops}$ loops as identical, with each loop being divided in $N_{sections} = 18$ sections, as stated above. Furthermore, the sections corresponding to the heater (i.e., the reactor core, labeled as item 4 in Fig. 11) and the cooler (i.e., the heat exchanger, labeled as item 12 in Fig. 11) are divided into $N_{nodes} = 40$ axial nodes to compute the temperature and flow gradients with sufficient detail. Both the average and hot channels are modeled in the core so that the increase in temperature in the hot channel due to the radial peaking factor can be calculated [29]. We note that in the subsequent analyses we have assumed $N_{loops} = 3$.

### *V.1.B Input Uncertainties for the GFR Model*

Since a deterministic T-H model is a mathematical representation of the behavior of the passive system, predictions of the system response to given accident conditions are accurate to the extent that the hypotheses made in the mathematical representation, and for its numerical solution, are true. Indeed, uncertainties affect the actual operation of a passive system and its modeling. On the one hand, there are phenomena, like the occurrence of unexpected events and accident scenarios, e.g., the failure of a component or the variation of the geometrical dimensions and material properties, which are random (i.e., aleatory) in nature. On the other hand, an additional contribution to uncertainty comes from our incomplete knowledge of the properties of the system and the conditions in which the phenomena occur (i.e., natural circulation). Recall that this uncertainty is termed epistemic and results from the hypotheses assumed by the model (i.e., *model* uncertainty), as well as the assumed values for the parameters of the model (i.e., *parameter* uncertainty) [30]. In this work, as well as in the reference work of Pagani, et al. [29], aleatory uncertainties are not considered for the estimation of the functional failure probability of the T-H passive system. Instead, we shall consider only uncertainties that are epistemic in nature, including both model and parameter uncertainty.

Parameter uncertainty is, perhaps, the simplest type of epistemic uncertainty to understand. As discussed in Chapter II, parameter uncertainty refers to the uncertainty regarding the exact numerical value to assign to various parameters used in the model (e.g., reactor power level, system pressure, etc.). Parameter uncertainty is readily accounted for by assigning a probability distribution to each parameter that represents our degree of belief that the parameter takes some specified value.

Model uncertainty, on the other hand, arises because mathematical models are simplified representations of real systems and, therefore, their outcomes may be affected by errors or bias. For instance, when attempting to model the flow of a fluid through a pipe, it is uncommon to model, in full resolution, the momentum transfer occurring at the wall by accounting for boundary layer effects; rather, these effects are approximated with the use of a friction factor. Similarly, Newton's law of convective cooling requires the specification of a heat transfer coefficient, and it is equally uncommon for this quantity to be calculated from a model that fully accounts for heat conduction into an advecting flow field. In both of these cases, engineers frequently rely on correlations[‡‡] developed from experimental data, and, because of their approximate nature, these correlations introduce model uncertainty into the analysis. One approach to account for this uncertainty is to parameterize it by introducing an error factor; for example, one could use a multiplicative model [30,166]:

$$z = m(\boldsymbol{x}) \cdot \zeta, \tag{V.1}$$

where $z$ is the real value of the parameter to be determined (e.g., heat transfer coefficients, friction factors, etc.), $m(\cdot)$ is the mathematical model of the correlation, $\boldsymbol{x}$ is the vector of correlating variables and $\zeta$ is a multiplicative error factor. Hence, the uncertainty in the output quantity $z$ is translated into an uncertainty in the error factor, $\zeta$, thereby parameterizing the model uncertainty. In other words, the model given by Eq. (V.1) permits us to treat model uncertainty as a parameter uncertainty by letting $\zeta$ represent an additional model parameter.

[‡‡] The term 'correlation' here is not be confused with the distinct, yet similar, concept of statistical correlation. Recall that statistical correlation refers to the strength of a linear relationship between two or more random variables, whereas engineers use correlation to refer to an empirically determined, (usually) nonlinear relationship between two or more quantities (e.g., friction factor and Reynolds number). This relationship is used as a surrogate to the true, yet unknown, relationship between the various quantities, and is not, itself, interpreted as a measure of statistical covariation between these quantities. One can think of an engineering correlation as the nonlinear transformation of, say, the Reynolds number that (statistically) correlates most strongly with the friction factor.

Additional model uncertainty might arise because the model is too simplified and therefore neglects some important phenomena which significantly influence the model output. This particular manifestation of uncertainty is sometimes distinguished from model uncertainty as *completeness* uncertainty [167].

For the GFR analysis, three parameters were identified whose uncertainty was deemed to be a likely contributor to the overall uncertainty in the system's performance – the reactor power level, the containment pressure following primary system depressurization, and the cooler wall temperature [29]. In addition, Pagani, et al. [29] noted two sources of model uncertainty resulting from the use of correlations for the Nusselt number and friction factor; because multiple flow regimes (forced convection, mixed convection, and free convection) were possible, and because the correlation errors are often highly dependent on the flow regime, a total of six error factors (three for the Nusselt number and three for the friction factor) were assigned to treat model uncertainty. Table 2 provides a summary of the relevant uncertainties and lists the mean ($\mu$) and standard deviation ($\sigma$) for each of the nine model inputs, $\{x_i : i = 1, 2, ..., 9\}$. Note that each input has been assumed to be normally distributed. Additional discussion regarding the normality assumption, as well as some justification for the values in Table 1, is provided by Pagani, et al. [29].

**Table 2. Summary of GFR model input uncertainties from Pagani, et al. [29].**

| | Name | Mean, $\mu$ | Standard deviation, $\sigma$ (% of $\mu$) |
|---|---|---|---|
| **Parameter Uncertainty** | Power (MW), $x_1$ | 18.7 | 1% |
| | Pressure (kPa), $x_2$ | 1650 | 7.5% |
| | Cooler wall temperature (°C), $x_3$ | 90 | 5% |
| **Model Uncertainty (Error Factor, $\zeta$)** | Nusselt number in forced convection, $x_4$ | 1 | 5% |
| | Nusselt number in mixed convection, $x_5$ | 1 | 15% |
| | Nusselt number in free convection, $x_6$ | 1 | 7.5% |
| | Friction factor in forced convection, $x_7$ | 1 | 1% |
| | Friction factor in mixed convection, $x_8$ | 1 | 10% |
| | Friction factor in free convection, $x_9$ | 1 | 1.5% |

### *V.1.C Failure Criteria for the GFR Passive Decay Heat Removal System*

The GFR passive decay heat removal system is considered failed whenever the temperature of the helium coolant leaving the core exceeds either 1200°C in the hot channel or 850°C in the average channel. These values are expected to limit the fuel temperature to levels which prevent excessive release of fission gases and high thermal stresses in the cooler (item 12 in Fig. 11) and in the stainless steel cross ducts connecting the reactor vessel and the cooler (items from 6 to 11 in Fig. 11) [29].

Letting **x** be the vector of the nine uncertain model inputs listed in Table 2, the failure region, *F*, for this system can be expressed as:

$$F = \{\mathbf{x} : T_{hot}(\mathbf{x}) > 1200\} \cup \{\mathbf{x} : T_{avg}(\mathbf{x}) > 850\},, \tag{V.1}$$

where $T_{hot}(\mathbf{x})$ and $T_{avg}(\mathbf{x})$ represent, respectively, the coolant outlet temperatures in the hot and average channels. Note that the quantity, $\{\mathbf{x} : T_{hot}(\mathbf{x}) > 1200\}$, refers to the set of values, **x**, such that $T_{hot}(x) > 1200$, and similarly for the second quantity in Eq. (V.1). Thus, Eq. (V.1) states, mathematically, that the failure domain is the set of inputs, **x**, such that either the hot-channel core outlet temperature exceeds 1200°C or the average-channel outlet temperature exceed 850°C.

By defining the performance indicator, *y*(*x*), as:

$$y(\mathbf{x}) = max\left\{\frac{T_{hot}(\mathbf{x})}{1200}, \frac{T_{avg}(\mathbf{x})}{850}\right\}, \tag{V.2}$$

the failure region given by Eq. (V.1) simplifies to:

$$F = \{\mathbf{x} : y(\mathbf{x}) > 1\}. \tag{V.3}$$

Hence, for the purpose of estimating the failure probability, *P*(*F*), of the system, we can regard the performance indicator defined in Eq. (V.2) as the single output of interest.

### ***V.1.D Comparative Evaluation of Bootstrapped RSs, ANNs, and GPs***

In the following sections, we compare the performance of bootstrapped quadratic RSs, ANNs, and GPs as metamodels for the GFR T-H simulation model. Each type of metamodel was constructed using design, or training, data sets, $\mathbf{D}_o = \{(\mathbf{x}_j, \mathbf{y}_j),\ j = 1,2,\ldots N_D\}$, where **y** is the 2-dimensional vector consisting of both model outputs, $T_{hot}$ and $T_{avg}$. In addition, a validation set consisting of 20 additional I/O pairs was obtained for monitoring the predictive capability of the ANNs to prevent overfit (see the discussion on the early stopping criterion in Section IV.4). The size of the design data set was varied (i.e., $N_D$ = 20, 30, 50, 70, and 100) so that we could investigate the effect of the number of available data on the predictive capability of each metamodel type. For each case, a Latin Hypercube Sample (LHS) of size $N_D$ was used to select the input configurations for the design data; this was done to forego the use of sophisticated experimental design procedures (see Sections IV.2.C and IV.2.D for a brief discussion on experimental designs with references). We note, however, that although Latin Hypercube designs are very simple to generate and therefore have been relatively popular, more sophisticated designs will often allow for more efficient metamodeling by optimizing the amount of available information in a given design set. On the other hand, an experimental design that is optimal for one type of metamodel will not necessarily be optimal for other metamodels.

After constructing the metamodels, a set of test data, $\mathbf{D}_{test}$, consisting of 20 I/O pairs (in addition to the design and validation data) was obtained. These test data were used to compute, for each type of metamodel, the coefficient of determination, $R^2$, and the RMSE for each model output (i.e., $T_{hot}$ and $T_{avg}$), thereby providing a general assessment of the metamodel accuracy. Notice that by computing these measures using a set of test data that is distinct from the design data used to construct the metamodels, we are avoiding the criticisms noted in Section IV.2.B. Table 3 summarizes these results for all three metamodels. In addition, the number of adjustable parameters for each of the metamodels is listed for comparison.

For $N_D$ = 20, the coefficient of determination, $R^2$, for the RS has not been reported because it was computed to be negative. Although $R^2$ is, theoretically, always positive, negative values can arise whenever it is computed using a data set that is different from the data used to create the regression model. Furthermore, whenever $N_D < 55$, the least-squares problem for estimating the RS coefficients is underdetermined, so one should expect large errors in its predictions. It is evident that both the ANN and GP outperform the RS in all the cases considered, both in terms of higher $R^2$ and lower RMSEs. This is most evident when the size of the data set is small (e.g., $N_D = 20$ or 30). This is explained by the higher flexibility of ANNs and GPs compared to RSs. Furthermore, the ANN and GP perform comparably, with the ANN being slightly superior. Recall, however, that a validation set consisting of 20 additional I/O pairs was needed for constructing the ANN. If this additional data is taken into consideration, the GP appears to be the superior predictor. Finally, we note that when the data set becomes sufficiently large (e.g., $N_D$ = 100), the RS performs very well, although still not quite as well the other metamodels. The reason for this is apparent from the scatter plot in Fig. 1 (page 25), which indicates that the model output is dominated by a single parameter (e.g., the pressure), and the relationship is roughly quadratic.

**Table 3. Summary of $R^2$ and RMSE for ANN, RS, and GP Predictions of GFR Model Outputs**

| **Artificial Neural Network (ANN)** | | | | | |
|---|---|---|---|---|---|
| | | $R^2$ | | RMSE [°C] | |
| $N_D$ | Number of parameters ($\omega_o$) | $T_{hot}$ | $T_{avg}$ | $T_{hot}$ | $T_{avg}$ |
| 20 | 50 | 0.8937 | 0.8956 | 38.5 | 18.8 |
| 30 | 50 | 0.9140 | 0.8982 | 34.7 | 18.6 |
| 50 | 62 | 0.9822 | 0.9779 | 15.8 | 8.7 |
| 70 | 50 | 0.9891 | 0.9833 | 12.4 | 6.8 |
| 100 | 50 | 0.9897 | 0.9866 | 12.0 | 6.3 |
| **Response Surface (RS)** | | | | | |
| | | $R^2$ | | RMSE [°C] | |
| $N_D$ | Number of parameters ($\beta$) | $T_{hot}$ | $T_{avg}$ | $T_{hot}$ | $T_{avg}$ |
| 20 | 55 | -- | -- | 247.5 | 130.9 |
| 30 | 55 | 0.4703 | 0.1000 | 59.7 | 37.2 |
| 50 | 55 | 0.7725 | 0.9127 | 39.1 | 11.6 |
| 70 | 55 | 0.9257 | 0.9592 | 22.4 | 7.9 |
| 100 | 55 | 0.9566 | 0.9789 | 17.1 | 6.5 |
| **Gaussian Process (GP)** | | | | | |
| | | $R^2$ | | RMSE [°C] | |
| $N_D$ | Number of parameters* | $T_{hot}$ | $T_{avg}$ | $T_{hot}$ | $T_{avg}$ |
| 20 | 11 | 0.7546 | 0.7213 | 40.6 | 20.7 |
| 30 | 11 | 0.9168 | 0.8942 | 23.7 | 12.7 |
| 50 | 11 | 0.9675 | 0.9802 | 14.8 | 5.5 |
| 70 | 11 | 0.9727 | 0.9838 | 13.6 | 5.0 |
| 100 | 11 | 0.9780 | 0.9935 | 12.2 | 3.2 |

* This GP model consists of a constant regression model (one parameter, $\beta_o$) and a Gaussian covariance model with 10 unknown parameters – one for the process variance ($\sigma^2$), and 9 range parameters ($\boldsymbol{\theta}$). Note, the smoothness parameter is fixed to 2.

For further illustration, Fig. 12 shows the empirical PDF and CDF of the hot-channel coolant outlet temperature, $T_{hot}(\mathbf{x})$, obtained with $N_T$ = 250000 simulations of the original T-H code (solid lines), together with the PDFs and CDFs estimated with the RS (dot-dashed lines) and ANN (dashed lines) constructed from $N_D$ = 100 design data. Similarly, Fig. 13 illustrates the same comparison for RSs and GPs. We note that Figs. 12 and 13 were constructed using different Monte Carlo simulation data, which explains the slight differences between the empirical PDFs and CDFs computed with the original code and the RS.

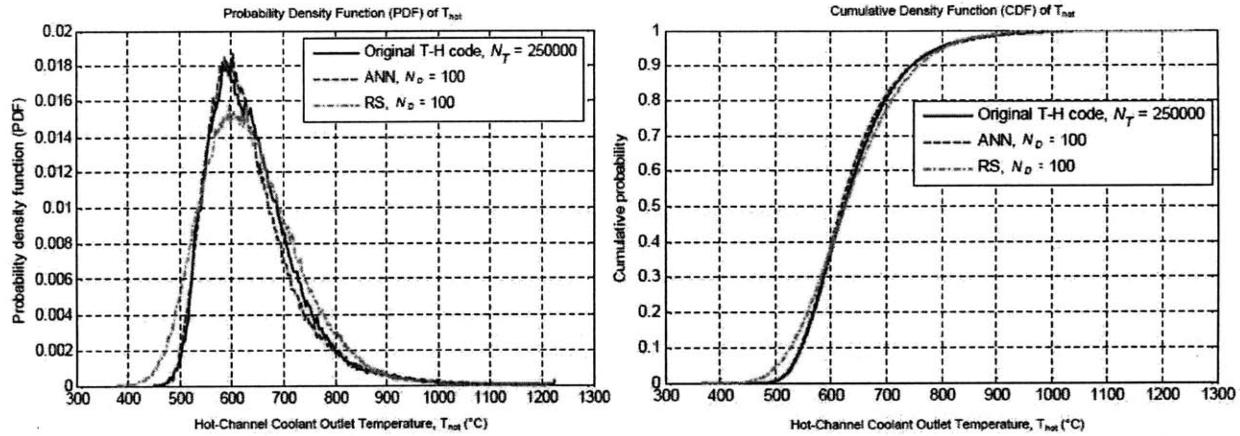

Figure 12. Comparison of empirical PDFs (left) and CDFs (right) for $T_{hot}$ estimated with ANNs and RSs

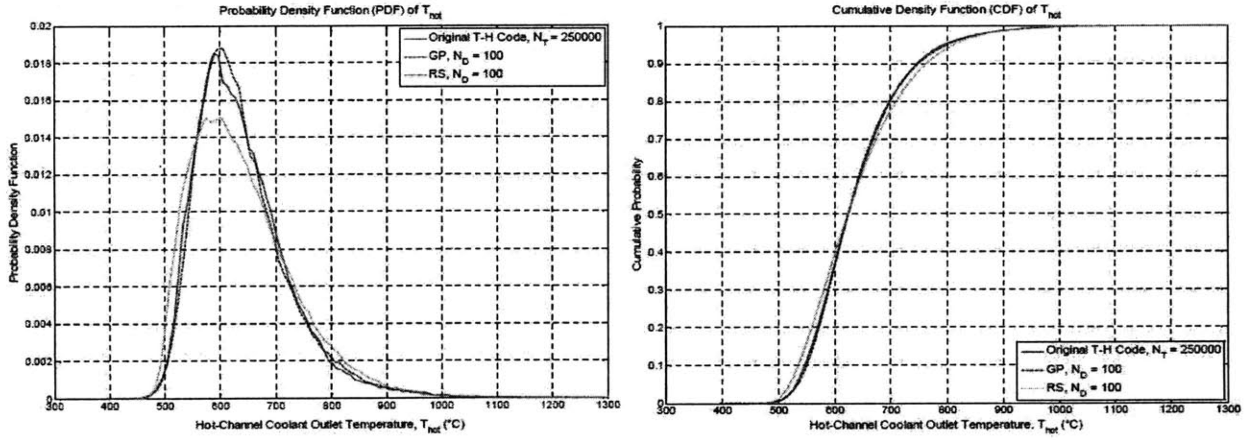

Figure 13. Comparison of empirical PDFs (left) and CDFs (right) for $T_{hot}$ estimated with GPs and RSs

It can be seen from Figs. 12 and 13 that both the ANN and GP estimates of the PDF and CDF are in much closer agreement to the reference results than the estimates given by the RS. This agrees with our discussion above based on the preliminary goodness-of-fit measures ($R^2$ and RMSE). However, it is generally known that a simple visual comparison of PDFs and CDFs can be misleading, so in the following sections we present a more quantitative assessment by computing the BBC estimates of (i) the 95th percentiles of the simulation output (i.e., $T_{hot}$ and $T_{avg}$), (ii) the functional failure probability of the DHR system, and (iii) the Sobol' sensitivity indices for each of the model inputs. These studies were carried out for each type of metamodel (i.e., RSs, ANNs, and GPs), with the exception of (iii) where GPs were excluded due to time constraints.

(i)BBC Estimates of the 95th Percentiles of the Coolant Outlet Temperatures

The $100{\cdot}\alpha^{th}$ percentiles of the hot- and average-channel coolant outlet temperatures are defined, respectively, as the values, $T_{hot,\alpha}$ and $T_{avg,\alpha}$, which satisfy:

$$P(T_{hot} \leq T_{hot,\alpha}) = \alpha \tag{V.2}$$

and

$$P(T_{avg} \leq T_{avg,\alpha}) = \alpha\,. \tag{V.3}$$

Figure 14 illustrates the Bootstrap Bias-Corrected (BBC) point-estimates (dots) and BBC 95% confidence intervals (CIs) (bars) for the 95th percentiles, $T_{hot,0.95}$ and $T_{avg,0.95}$, of the hot- (left) and average- (right) channel coolant outlet temperatures, respectively, obtained by bootstrapped ANNs (top), RSs (middle), and GPs (bottom), constructed with $N_D$ = 20, 30, 50, 70 and 100 design data. The "true" (i.e., reference) values, indicated by the dashed lines ($T_{hot,0.95} = 796°\text{C}$ and $T_{avg,0.95} = 570°\text{C}$), have been obtained by propagating the epistemic uncertainties through the original T-H code by standard Monte Carlo Simulation with $N_T = 250000$ samples. For this study, $B$ = 1000 bootstrap samples were used.

It can be seen that the point estimates provided by the bootstrapped ANNs and GPs are very close to the real values in all the cases considered (i.e., for $N_D$ = 20, 30, 50, 70 and 100). In contrast, bootstrapped RSs provide accurate point-estimates only for $N_p$ = 70 and 100. The metamodel uncertainty associated with the RS estimates is much higher than that of the ANN, as demonstrated by the wider confidence intervals. This is a consequence of overfitting. Recall that the bootstrap procedure requires that $B$ bootstrap samples $\mathbf{D}_b$, $b$ = 1, 2, ..., $B$, be drawn at random with replacement from the original set $\mathbf{D}_o$ of input/output patterns. Consequently, some of the I/O patterns in $\mathbf{D}_o$ will appear more than once in $\mathbf{D}_b$, and some will not appear at all. Thus, the number of *unique* data in each bootstrap sample $\mathbf{D}_b$ will typically be lower than the number of the original data in $\mathbf{D}_o$. This is particularly true if the number of data in $\mathbf{D}_o$ is very low (e.g., $N_D$ = 20 or 30). Since, during the bootstrap procedure (Section IV.5.A) the number of adjustable parameters (i.e., $\boldsymbol{\omega}_o$ for ANNs, $\boldsymbol{\beta}$ for RSs, etc.) in the metamodel is fixed, it is possible that the number of these parameters becomes larger than the number of data in the bootstrap sample $\mathbf{D}_b$; in the case of RSs, the least-squares problem for estimating the coefficients becomes underdetermined. This causes the metamodel to overfit the bootstrap training data $\mathbf{D}_b$, resulting in degraded estimation performance. As a result, each individual datum in the set, $\mathbf{D}_b$, will have a large influence on the estimated coefficients, so that each of the $B$ data sets will yield very different estimates for the model coefficients. This, in turn, will lead to large variations between the predictions from these metamodels.

Figure 14 indicates that GPs also from this effect when $N_D \leq 30$. For these cases, although the point-estimates are very close to the true percentiles, the confidence intervals are very wide. In practice, this would be a clear indication that more data are needed to reduce the metamodel uncertainty. ANNs suffer much less from overfitting due to the use of the early stopping method, described in Section IV.5.A. Essentially, by adding additional data (i.e., the validation set) to the problem, the early stopping criterion results in more consistent estimates of the metamodel coefficients, thereby reducing the variability in the ANN predictions. For $N_D \geq 50$, ANNs and GPs perform comparably, with both providing highly accurate estimates of the true percentiles.

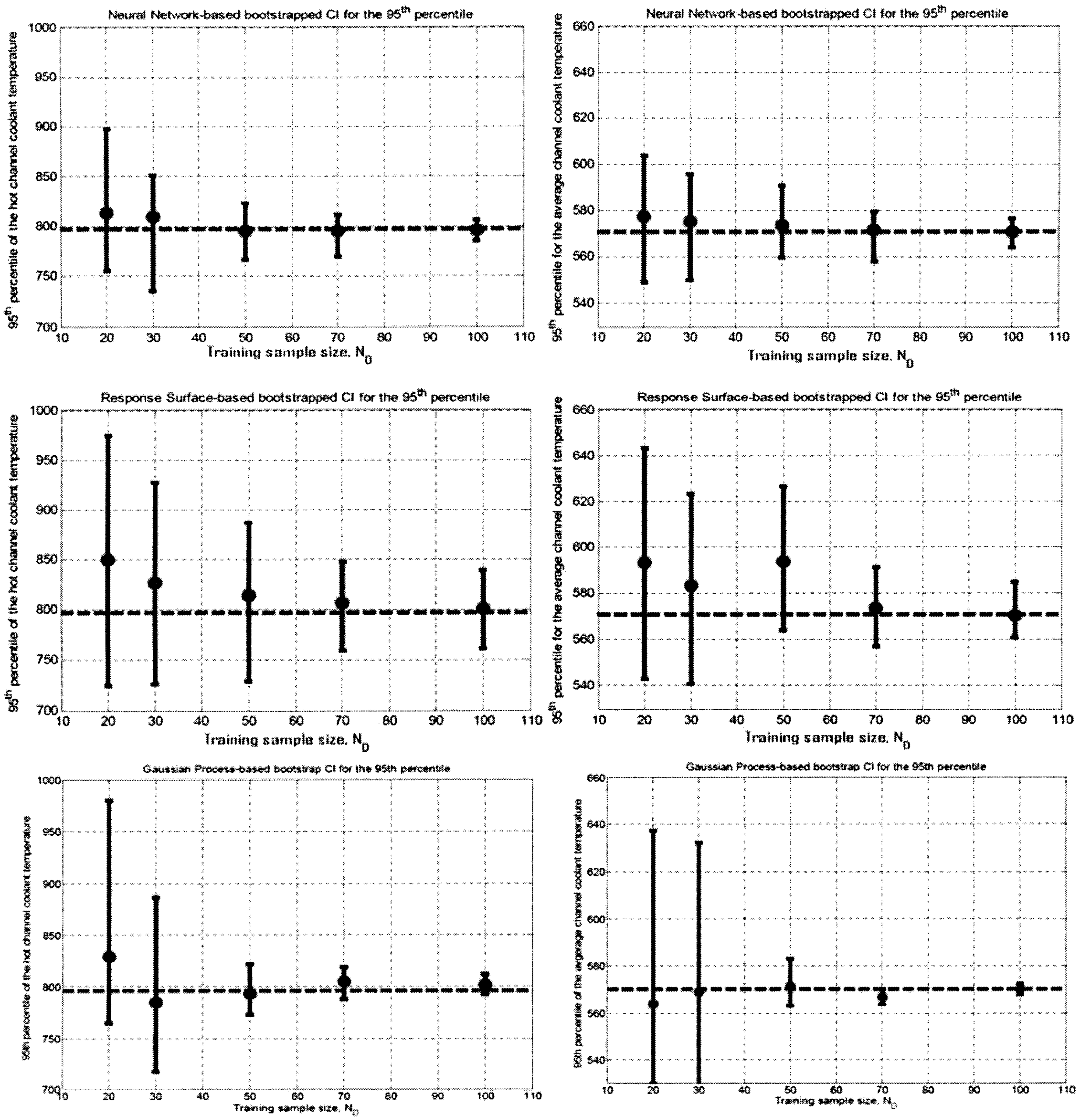

Figure 14. Comparison of Bootstrap Bias-Corrected point-estimates and 95% Confidence Intervals for the 95th percentiles of the hot- and average- channel coolant outlet temperatures obtained with ANNs, RSs, and GPs.

(ii)BBC Estimates of the Functional Failure Probability

This section presents a comparison of RSs, ANNs, and GPs for predicting the functional failure probability (per demand) of the passive DHR system for the GFR. As before, standard MCS with $N_T$ = 250000 simulations was performed with the original T-H code to provide a reference estimate of the failure probability. The reference MCS gave a failure probability estimate of $P(F) = 3.34 \times 10^{-4}$, which we take as the "true" value.

Table 4 summarizes the BBC point-estimates, $\hat{P}(F)_{BBC}$, and the 95% confidence intervals (CIs) of the function failure probability estimated with bootstrapped ANNs, RSs, and GPs constructed with $N_D$ = 20, 30, 50, 70, and 100 design data. These results are also illustrated in Fig. 15, with the reference estimate (i.e., $P(F) = 3.34 \times 10^{-4}$) indicated by the dashed line.

**Table 4. BBC Point-Estimates and 95% Confidence Intervals for the Functional Failure Probability Estimated with ANNs, RSs, and GPs**

| **Failure probability, $P(F) = 3.34 \cdot 10^{-4}$** | | |
|---|---|---|
| **Artificial Neural Network (ANN)** | | |
| **Training sample size, $N_p$** | **BBC estimate** | **BBC 95% CI** |
| 20 | $1.01 \times 10^{-4}$ | $[0, 7.91 \times 10^{-4}]$ |
| 30 | $1.53 \times 10^{-4}$ | $[0, 6.70 \times 10^{-4}]$ |
| 50 | $2.45 \times 10^{-4}$ | $[8.03 \times 10^{-5}, 4.27 \times 10^{-4}]$ |
| 70 | $3.01 \times 10^{-4}$ | $[2.00 \times 10^{-4}, 4.20 \times 10^{-4}]$ |
| 100 | $3.59 \times 10^{-4}$ | $[2.55 \times 10^{-4}, 4.12 \times 10^{-4}]$ |

| **Response Surface (RS)** | | |
|---|---|---|
| **Training sample size, $N_p$** | **BBC estimate** | **BBC 95% CI** |
| 20 | $9.81 \times 10^{-5}$ | $[0, 8.39 \times 10^{-4}]$ |
| 30 | $1.00 \times 10^{-4}$ | $[0, 7.77 \times 10^{-4}]$ |
| 50 | $2.15 \times 10^{-4}$ | $[7.43 \times 10^{-5}, 5.07 \times 10^{-4}]$ |
| 70 | $2.39 \times 10^{-4}$ | $[1.16 \times 10^{-4}, 4.61 \times 10^{-4}]$ |
| 100 | $3.17 \times 10^{-4}$ | $[2.20 \times 10^{-4}, 4.40 \times 10^{-4}]$ |

| **Gaussian Process (GP)** | | |
|---|---|---|
| **Training sample size, $N_p$** | **BBC estimate** | **BBC 95% CI** |
| 20 | $7.47 \times 10^{-4}$ | $[0, 2.98 \times 10^{-3}]$ |
| 30 | $8.36 \times 10^{-4}$ | $[0, 6.51 \times 10^{-3}]$ |
| 50 | $3.34 \times 10^{-4}$ | $[0, 1.22 \times 10^{-3}]$ |
| 70 | $2.76 \times 10^{-4}$ | $[2.34 \times 10^{-4}, 3.86 \times 10^{-4}]$ |
| 100 | $3.24 \times 10^{-4}$ | $[2.51 \times 10^{-4}, 4.45 \times 10^{-4}]$ |

It can be seen that as the size $N_D$ of the training sample increases, both the ANN and the RS provide increasingly accurate estimates of the true functional failure probability $P(F)$, as one would expect. It is also evident that in some of the cases considered (e.g., $N_D$ = 20, 30 or 50) the functional failure probabilities are significantly underestimated by both the ANN and the RS (e.g., the BBC estimates for $P(F)$ lie between $9.81 \times 10^{-5}$ and $2.45 \times 10^{-4}$) and the associated uncertainties are quite large (e.g., the lengths of the corresponding BBC 95% CIs are between $4 \times 10^{-4}$ and $8 \times 10^{-4}$). On the other hand, the GPs tend to overestimate the failure probability for

$N_D \leq 30$, with very large confidence intervals. Even for $N_D$ = 50, although the BBC point-estimate from the GPs agrees perfectly with the true failure probability, the large confidence interval suggests that this is only a matter a chance. It is not until $N_D \geq 70$ that the bootstrapped GPs begin providing consistent estimates. For these larger design sets, all of the metamodels seem to perform comparably.

Notice that, for the purposes of estimating the failure probability, the discrepancy in the performance of the metamodels is less evident than in the case of percentile estimation, especially for the larger design sets. In particular, RSs seem to perform almost as well as GPs and ANNs for estimating the failure probability, whereas for estimating the percentiles, RSs were clearly inferior. One explanation of this is due to the binary nature of the indicator function. For example, suppose that the true value of the hot channel coolant temperature is 1250 °C and the corresponding estimate by the metamodel is 1500 °C. In such a case, the estimate is *inaccurate* in itself, but *exact* for the purpose of functional failure probability estimation. That is, both the true temperature and the estimated temperature are counted as failures. Hence, even if the RS is unable to accurately predict the exact temperature, it is often able to predict whether a failure will occur. On the other hand, GPs appear to perform worse for failure probability estimation, particularly for small design sets. In these cases, it is very likely that, during the bootstrap procedure, none of the design points will be near the failure region. Since the GP makes predictions based on its nearest neighbors, if none of the design points are near the failure region, the GP will be a very poor predictor in this region. Conversely, if many points were clustered near the failure region, one should expect the GP to give very accurate predictions for the failure probability. Finally, we note that the performance of the metamodels can be sensitive to the experimental design used in its construction, particularly when the data are limited. If the design points in this study had been selected based on some optimal experimental design, we might expect that the metamodels would provide more accurate estimates of the failure probability for the cases with $N_D \leq 20$.

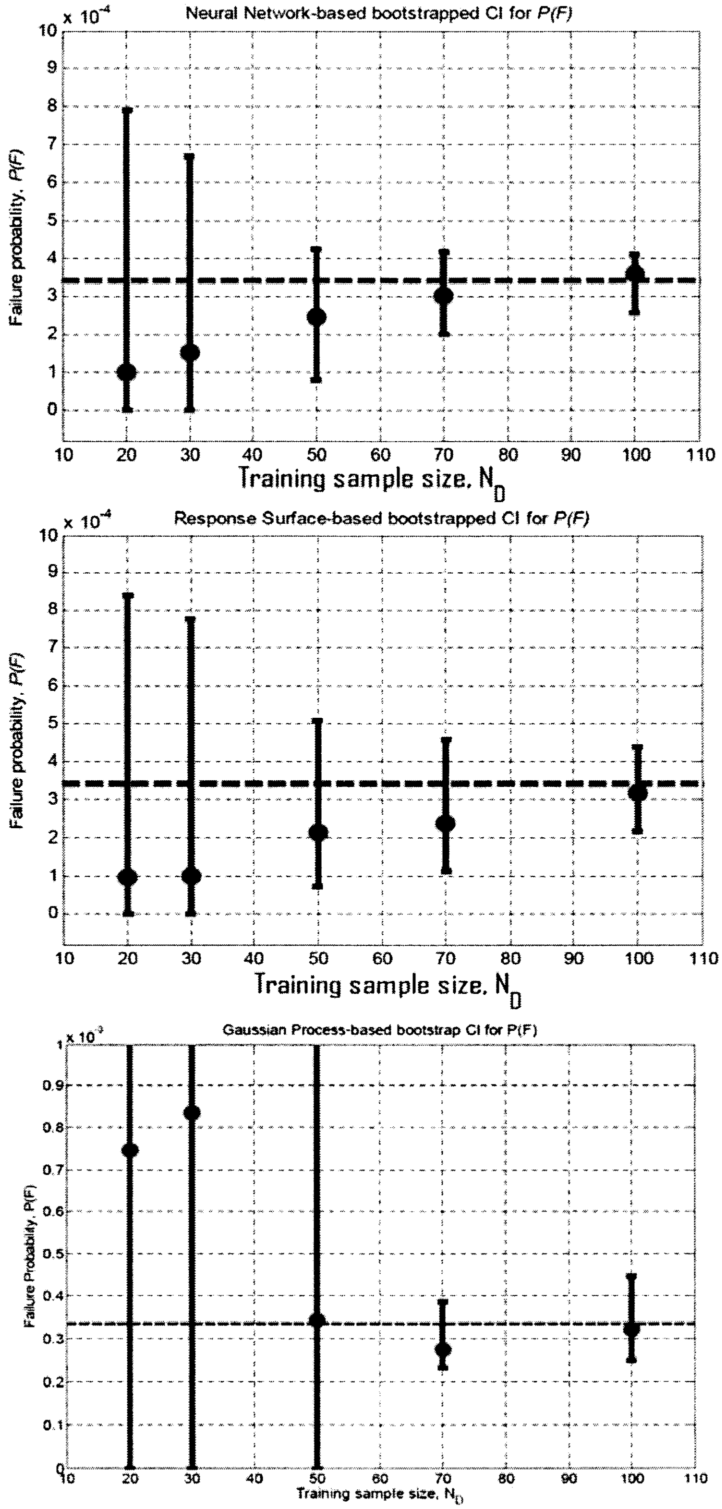

Figure 15. Comparison of BBC point-estimates (dots) and 95% Confidence Intervals (bars) for $P(F)$ estimated with ANNs (top), RSs (middle), and GPs (bottom).The reference value of $P(F) = 3.34\times10^{-4}$ is given by the dashed line.

### (iii)BBC Estimates of First-Order Sobol' Indices and Input Parameter Ranking

In this section, first-order Sobol sensitivity indices (see Eq. (III.16)) are computed for each of the nine model input parameters using bootstrapped RSs and ANNs. The sensitivity indices are computed with respect to the hot-channel coolant outlet temperature, $T_{hot}$; that is, for this study, $T_{hot}$ was the only model output that was considered. The sensitivity indices were computed via the Sobol'-Saltelli MC algorithm discussed in Section III.4.B using $N$ = 10000 samples. The total number of samples required is then $N_T = N(m+2) = 110000$, since $m = 9$ for this problem. Standard MCS with $N_T = 110000$ was performed with the original T-H code to provide reference estimates of the Sobol' indices, $S_i$ for $i$=1, 2,..., 9. The left column of Table 5 gives the ranking of the input parameters based on the reference estimates of the sensitivity indices (listed in parenthesis). Table 5 also provides the parameter rankings and BBC estimates, $\hat{S}_{i,BBC}$, obtained with ANNs (middle column) and RSs (right column) constructed from $N_D$ = 100 design data. It can be seen that the ranking provided by bootstrapped ANNs is exactly the same as the reference ranking (i.e., the ranking obtained by the original T-H code), whereas the bootstrapped RS correctly ranks only the first five uncertain variables. It is worth noting that these five parameters account for about 96% of the variance of the hot channel coolant outlet temperature $T_{hot}$; hence, the failure of the RS to exactly rank the remaining four parameters is not likely to be significant since the dominant parameters have been correctly identified.

**Table 5. Comparison of input parameter rankings from first-order Sobol' indices estimated with ANNs and RSs with reference estimates**

| **Uncertain variables ranking – Output considered: hot-channel coolant outlet temperature, $T_{hot}$** | | |
|---|---|---|
| **Original T-H code, $N_T$ = 110000 ($S_i$)** | **ANN, $N_D$ = 100 ($\hat{S}_{i,BBC}$)** | **RS, $N_D$ = 100 ($\hat{S}_{i,BBC}$)** |
| Pressure (0.8105) | Pressure (0.8098) | Pressure (0.8219) |
| Friction mixed (0.0594) | Friction mixed (0.0605) | Friction mixed (0.0715) |
| Nusselt mixed (0.0583) | Nusselt mixed (0.0591) | Nusselt mixed (0.0665) |
| Cooler wall temperature (0.0303) | Cooler wall temperature (0.0368) | Cooler wall temperature (0.0240) |
| Power ($5.950\times10^{-3}$) | Power ($6.345\times10^{-3}$) | Power ($5.523\times10^{-3}$) |
| Nusselt free ($5.211\times10^{-4}$) | Nusselt free ($5.199\times10^{-4}$) | Friction forced ($4.030\times10^{-3}$) |
| Friction free ($2.139\times10^{-4}$) | Friction free ($1.676\times10^{-4}$) | Nusselt free ($6.790\times10^{-4}$) |
| Nusselt forced ($4.214\times10^{-5}$) | Nusselt forced ($6.430\times10^{-5}$) | Nusselt forced ($3.700\times10^{-4}$) |
| Friction forced ($1.533\times10^{-5}$) | Friction forced ($1.634\times10^{-5}$) | Friction free ($2.153\times10^{-5}$) |

For illustration purposes, Fig. 16 shows the BBC point-estimates (dots) and 95% confidence intervals (bars) for the first-order Sobol sensitivity indices of parameters $x_1$ (power) and $x_2$ (pressure) computed with bootstrapped ANNs and RSs constructed from $N_D$ = 20, 30, 50, 70 and 100 design data. The reference (i.e., "true") estimates for the Sobol' indices are indicated by the dashed lines. The BBC point-estimate and 95% CI data for each of the nine inputs and for each design data set can be found in Appendix D.

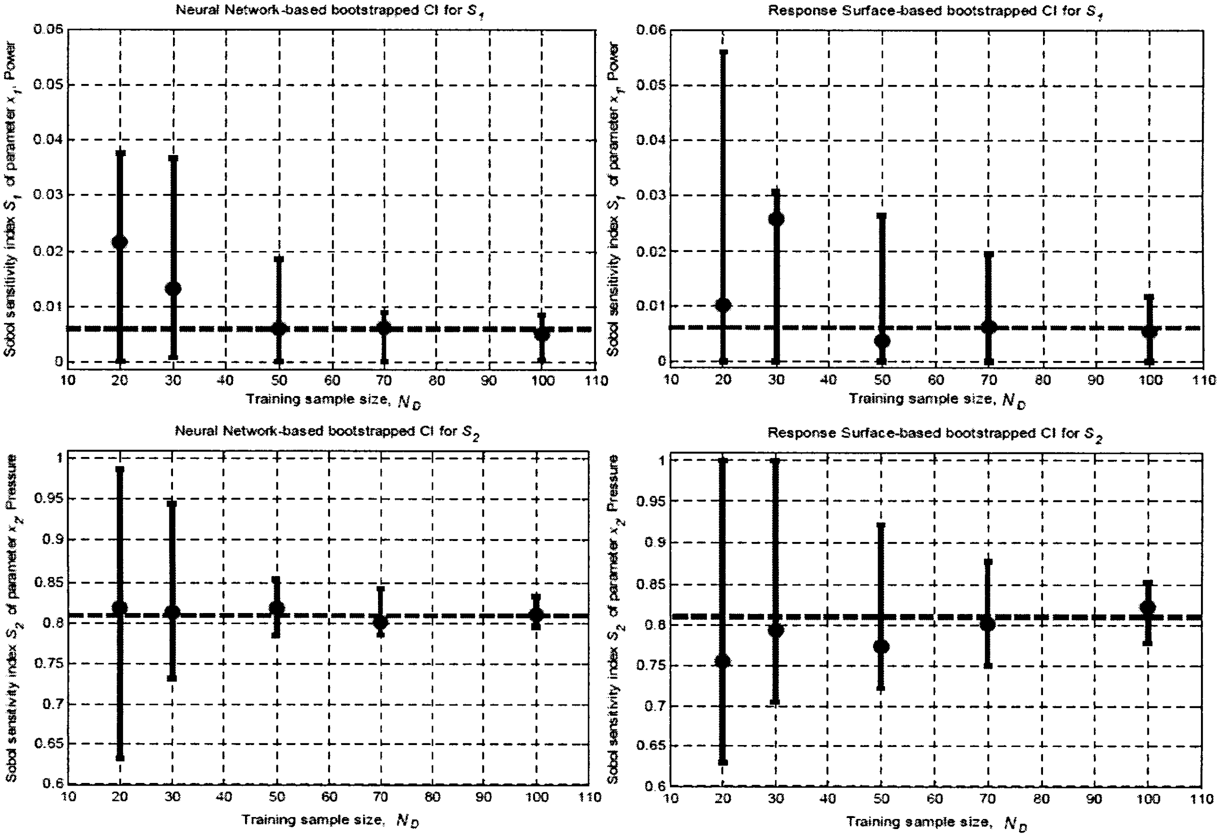

Figure 16. Comparison of BBC point-estimates (dots) and 95% Confidence Intervals (bars) for the first-order Sobol' indices of power (top) and pressure (botton) estimated with ANNs (left) and RSs (right).

From Fig.16, it can be seen that, with one exception, the BBC point-estimates from the ANNs are closer to the reference values than those given by the RSs, regardless of the size of the design data set, $N_D$; the exceptional case can be seen in the top-right panel, where, for $N_D$ = 20, the RS provides a better point-estimate for the Sobol' index of $x_1$ (power). A comparison of the confidence intervals, however, indicates that this is likely to be mere chance. Across the board, the ANN estimates have smaller confidence intervals, indicating that bootstrapped ANNs provide superior (i.e., less uncertain) estimates.

Finally, it is interesting to note that for very important variables (i.e., those with Sobol' indices close to one), such as $x_2$ (pressure), the estimates provided by the regression models converge much faster to the true values as compared to noninfluential parameters (i.e., those with Sobol' indices close to zero) such as $x_1$ (i.e., power); for example, bootstrapped ANNs provide an almost correct estimate of $S_2$ even for $N_D = 20$ (see the bottom-left panel of Fig. 16), whereas the point estimates for $S_1$ do not appear to converge until $N_D \geq 50$.

### *V.1.E Summary and Conclusions from GFR Case Study*

This section has presented a comparative evaluation of RSs, ANNs, and GPs for metamodeling a T-H model of a passive (i.e., natural circulation) decay heat removal (DHR) system for the GFR. The T-H model consists of nine input parameters and two relevant outputs (i.e., the hot- and average-channel coolant outlet temperatures). Metamodels were constructed using various sizes of design data sets to assess the predictive capability of each when data are limited. Three studies were carried out to predict (i) the 95th percentiles of the hot-channel and average-channel coolant outlet temperatures, (ii) the functional failure probability of the DHR system, and (iii) the first-order Sobol' sensitivity indices for parameter ranking. For the first two studies, all three metamodels (i.e., RSs, ANNs, and GPs) were compared, whereas, for the last study, GPs were excluded due to time constraints.

The results of these studies demonstrate that, due to the added flexibility afforded by ANNs and GPs, these metamodels are capable of making more accurate predictions than quadratic RSs. In most of the cases considered, the confidence intervals provided by bootstrapped ANNs and GPs were narrower than those of the bootstrapped RSs; this demonstrates that ANNs and GPs tend to provide more consistent predictions. On the other hand, there seems to be a threshold effect when the data are limited; specifically, none of the metamodels gave consistent predictions when the design data were limited to fewer than 30. When the data were increased to somewhere around 50 data, the predictions from ANNs and GPs became far more consistent, with very tight confidence intervals. The RS estimates were also more consistent for $N_D > 50$, but to a lesser degree. It was also found that the GPs gave more inconsistent predictions of failure probability. We noted that this was likely a result of poor data placement. The failure probability of this system is quite low, with the failure domain existing in the periphery of the design space. If design data are not located near this region, the

GPs will be unable to accurately predict the response; in other words, GPs are poor extrapolators. Because the system response is roughly quadratic, a second order RS can provide a relatively good fit to the data. Consequently, the extrapolation error is less severe when a RS is being used for prediction.

Perhaps most importantly, these results suggest that bootstrapping may be useful to assess metamodel uncertainty. In all of the cases, the confidence intervals computed from bootstrapping provided useful insight as to whether the point-estimates were accurate. This information can be very valuable to the analyst who must determine whether enough data have been obtained with the model. In the cases above, it is clear that more than 30 data, and perhaps more than 50 data, are necessary to make accurate predictions. Clearly, the question of how much data is enough is problem-specific; that is, the "goodness" of the estimate must be assessed in the context of the problem. For instance, even for $N_D$ = 20 or 30, the failure probability confidence bounds for ANNs and RSs indicate that these methods could provide at least an estimate of the order of magnitude. If this is sufficient to demonstrate that the system satisfies the target safety goals, then more data might not be necessary. On the other hand, if a more accurate estimate is needed, then the bootstrap confidence intervals make it possible to know whether more data should be obtained.

## V.2 The Flexible Conversion Ratio Reactor (FCRR) Case Study

The second case study that was carried out considers the functional failure probability of a passive decay removal (DHR) system consisting of two systems, the Reactor Vessel Auxiliary Cooling System (RVACS) and the Passive Secondary Auxiliary Cooling System (PSACS), operating in tandem to provide emergency core cooling for a lead-cooled FCRR during a simulated station black-out (SBO) transient. More details regarding the FCRR are presented in Section V.2.A.

Due to the prohibitive computational demand presented by the RELAP5-3D thermal-hydraulic (T-H) system model, metamodels were investigated for performing the requisite uncertainty and sensitivity analyses. Fong et al. [31] first considered the use of quadratic RSs as a surrogate to the T-H model for UA and SA. In Section V.2.B, we elaborate on their results by performing a similar analysis with quadratic RSs, but account for the metamodel uncertainty using the bootstrapping technique discussed in Section IV.5.A. In addition, we compare these results with those obtained using GP metamodels. Finally, Section V.2.C discusses the conclusions from this case study.

### *V.2.A Description of the System*

The FCRR is a 2400 $MW_{th}$ lead-cooled fast reactor whose design has been the subject of recent research at MIT [168,169]. The design includes two passive DHR systems, the RVACS and PSACS, whose joint operation is intended to prevent fuel failure during unexpected events. RVACS can be seen in Fig. 17, which illustrates the primary system for the FCRR, and a schematic of PSACS is given in Fig.18. Both of these systems are designed for decay heat removal during normal and emergency conditions. In brief, RVACS is a natural circulation system designed to directly cool the reactor guard vessel with ambient air. During shutdown, the lead coolant conducts heat from the core to the reactor vessel wall. This heat is then radiated across the gap between the reactor vessel wall and the guard vessel which is in contact with the air circulating through the RVACS chimney.

A secondary passive core cooling function is provided by four PSACS trains in parallel, one of which is illustrated in Fig. 18. During a transient, the supercritical-$CO_2$ (S-$CO_2$) working fluid is diverted from the turbines into the PSACS heat exchanger (PAHX), which is submerged in a pool of water. Upon cooling, the S-$CO_2$ returns to the intermediate heat exchanger (IHX, illustrated in both Figs. 17 and 18) where it is reheated by the lead coolant in the primary system. The PAHX is situated two meters above the top of the IHX to provide the necessary gravity head to drive the natural circulation. Fong et al. [31] considered a variety of scenarios where the PSACS isolation valves failed to open, which will not be repeated here. They identified the risk-limiting case (i.e., in terms of both severity and likelihood) as consisting of the failure of two-out-of-four PSACS isolation valve trains so that only two PSACS trains are available for decay heat removal. This is the scenario that we consider in the subsequent section.

The DHR systems for the FCRR are considered to have failed if the peak clad temperature (PCT) exceeds 725°C at any time, and for any duration, during a 72-hour window following reactor shutdown [168]. In their reference study, Fong et al. [31] identified five input parameters whose uncertainty was deemed likely to have a significant effect on the PCT; these parameters are (1) the fraction of tubes in the PAHX that are plugged, (2) the water temperature in the PSACS heat sink at the start of the transient, (3) the emissivity of the reactor vessel wall (this quantity affects the efficacy of the radiative heat transfer between the reactor vessel and the guard vessel), (4) the fraction of the total cross-sectional area of the RVACS chimney that is blocked, and (5) the temperature of the air entering the RVACS chimney (i.e., the ambient air temperature). Table 6 summarizes these parameters and lists their respective medians (i.e., central level) and upper and lower levels; the levels refer to the discretization of the parameter ranges for the experimental design that is used in constructing the metamodels. Furthermore, Table 7 summarizes the distributions assigned to each parameter. Fong et al. [31] discuss the motivation for choosing these distributions. We note that both the RVACS blockage ($x_4$) and the fraction of plugged PSACS tubes ($x_1$) are assigned exponential distributions, and since this distribution is parameterized by a single parameter, $\lambda$, the specification of their respective 80th percentiles in Table 7 is redundant (although, consistent). The remaining parameters were assumed normally distributed, with the distributions truncated to prevent unrealistic scenarios from being simulated (i.e., negative emissivity, etc.); more on this is can be found in [31].

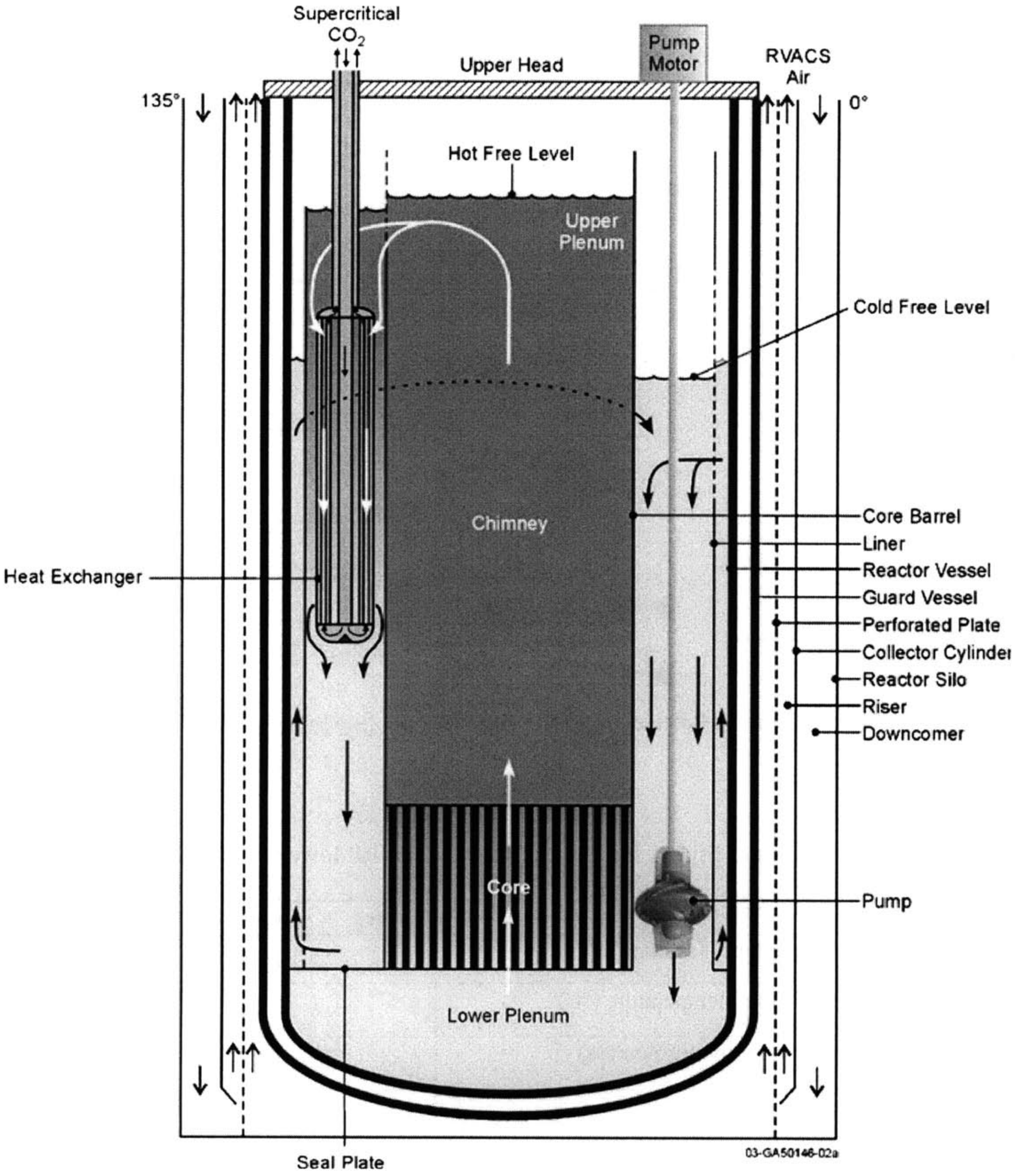

Figure 17. Schematic of the FCRR primary system, including the Reactor Vessel Auxiliary Cooling System (RVACS) (from [169])

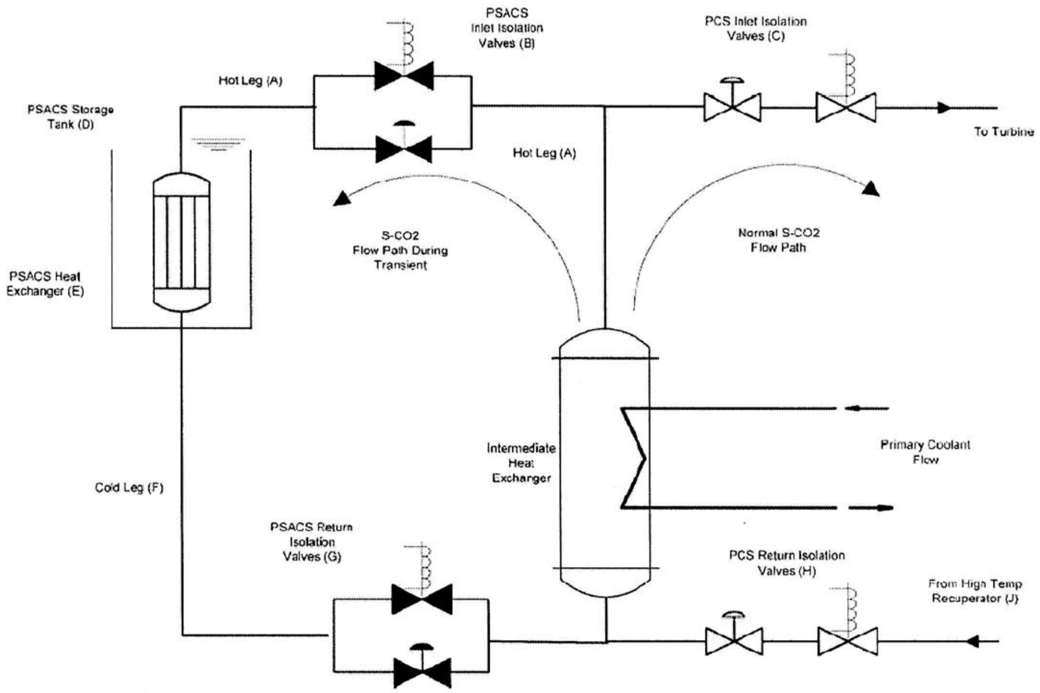

Figure 18. Schematic of the Passive Secondary Auxiliary Cooling System (PSACS) (from [31])

**Table 6. Summary of model inputs for the FCRR model with lower, central, and upper levels**

| Factor | Lower | Central | Upper |
|---|---|---|---|
| $x_1$, PSACS plugged tubes (fraction) | 0 | 0.075 | 0.15 |
| $x_2$, PSACS initial water temperature (°C) | 7 | 27 | 47 |
| $x_3$, RVACS emissivity (dimensionless) | 0.65 | 0.75 | 0.85 |
| $x_4$, RVACS blockage (fraction) | 0 | 0.075 | 0.15 |
| $x_5$, RVACS inlet temperature (°C) | 7 | 27 | 47 |

**Table 7. Summary of input uncertainty distributions for the FCRR model**

| Variable | Distribution | $X_{20}$ | $X_{80}$ | $\Lambda$ |
|---|---|---|---|---|
| $x_1$ | Exponential | - | 0.15 | 10.7 |
| $x_2$ | Truncated Normal | 7 | 47 | - |
| $x_3$ | Truncated Normal | 0.65 | 0.85 | - |
| $x_4$ | Exponential | - | 0.15 | 10.7 |
| $x_5$ | Truncated Normal | 7 | 47 | - |

### *V.2.B Comparison of RS and GP Metamodels for Estimating the Failure Probability of the FCRR*

In this section, we present a comparison of RS and GP metamodels for estimating the failure probability of the DHR systems for the FCRR. In the reference study by Fong, et al. [31], an experimental design consisting of 27 runs was used for constructing a quadratic RS. The data for this experiment are provided in Table E.1 in Appendix E. In Table 8, we summarize the goodness-of-fit statistics for the quadratic RS (QRS) built from these data and provide the estimated failure probability. It can be seen that the QRS provides a very good fit to the data, with $R^2 = 0.99$ and RMSRE of 1.93. Note that $R^2$ has not been computed using test data, as was done for the GFR case study. Furthermore, the RMSRE is the RMSE computed using the RS residuals, whereas the RMSPE is a convenient rescaling of PRESS (i.e., $\text{RMSPE} = \sqrt{\text{PRESS}/N_D}$). A comparison of the two reveals that the RMSPE is approximately a factor of 5 larger than the RMSRE, indicating that, although the QRS fits the design data very well, it is a poor predictor; in other words, the QRS is overfit to the data and the bias error is large.

**Table 8. Goodness-of-fit summary of 27-point quadratic RS for FCRR model**

| Quadratic Response Surface ($N_D$=27) | |
|---:|---:|
| **RMS Residual Error (RMSRE):** | 1.93 |
| **RMS Prediction Error (RMSPE):** | 10.02 |
| **Coefficient of Determination ($R^2$):** | 0.99 |
| **Adjusted $R^2$:** | 0.95 |
| **Failure Probability:** | 0.11 |

Fong et al. [31] performed eight additional simulations with the RELAP5-3D model to validate the QRS, and these data have been provided in Table E.2 in Appendix E. Table 9 summarizes the results from the QRS constructed with the resulting set of 35 design data. In this case, we see that, although the RMSPE has slightly decreased, the RMSRE has increased by a factor of two. Hence, the observed responses (i.e., the RELAP outputs) are deviating from the assumed quadratic form and, consequently, the goodness-of-fit of the QRS has depreciated. Furthermore, and perhaps most worrisome, is that the estimated failure probability has increased nearly 50%. Consequently, it was decided that an additional 27 simulations would be performed with the RELAP5-3D model to assess whether the failure probability estimate converges. These data are provided in Table E.3 of Appendix E, and the results from the QRS constructed from the total set of 62 data are summarized in Table 10.

**Table 9. Goodness-of-fit summary of 35-point quadratic RS for FCRR model**

| Quadratic Response Surface ($N_D$=35) | |
|---:|---|
| RMS Residual Error (RMSRE): | 3.89 |
| RMS Prediction Error (RMSPE): | 9.57 |
| Coefficient of Determination ($R^2$): | 0.96 |
| Adjusted $R^2$: | 0.90 |
| Failure Probability: | 0.16 |

**Table 10. Goodness-of-fit summary of 62-point quadratic RS for FCRR model**

| Quadratic Response Surface ($N_D$=62) | |
|---:|---|
| RMS Residual Error (RMSRE): | 4.32 |
| RMS Prediction Error (RMSPE): | 7.13 |
| Coefficient of Determination ($R^2$): | 0.95 |
| Adjusted $R^2$: | 0.92 |
| Failure Probability: | 0.21 |

A comparison of Tables 9 and 10 indicates a trend similar to what was observed when comparing Tables 8 and 9. Specifically, the RMSRE has increased while the RMSPE has decreased. Hence, we see that each subsequent QRS sacrifices goodness-of-fit to reduce RMSPE. Statistically, this is equivalent to reducing bias error (i.e., poor predictive capability) at the expense of increased variance error (i.e., larger residuals). This further demonstrates one of the key points emphasized in Section IV.2.D – the "goodness" of the RS should not be determined on $R^2$ alone. That is, a high $R^2$ (or, equivalently, a low RMSRE) is not indicative of the predictive capability of the RS. Finally, inspection of Tables 8-10 reveals that each subsequent estimate of the failure probability has increased, with no indication that the estimate is converging to some definite value. Consequently, from the results presented thus far, there is no clear way to utilize this information to estimate the failure probability. Each subsequent QRS is less accurate, in terms of residual variance, but less biased, and each provides a significantly different estimate of the failure probability. Fortunately, the bootstrap method, discussed in IV.5.A, can be used to compute confidence bounds for these estimates, thereby giving some indication of which of the three estimates given above is closer to the truth.

Figure 19 and Table 11 summarize the results from using bootstrapped QRSs and GPs with $B = 1000$ bootstrap samples. Once again, for simplicity, a Gaussian correlation model has been assumed for the GP. These results indicate that for $N_D$ = 27 and 35, the BBC estimates from the QRS are quite low and likely underestimate the true failure probability. Moreover, the confidence intervals for these estimates are rather large, indicating that these results are not likely to be accurate. The BBC estimates from the GPs are more consistent, with each being near 0.2. Furthermore, the confidence intervals for the GP estimates are significantly smaller that those for the QRS estimates, particularly for $N_D$ = 27 and 35. This provides further indication that the added flexibility offered by GPs can potentially make these metamodels more accurate for prediction. We note, however, that for $N_D$ = 62, both the GP and QRS estimates are similar, and their corresponding confidence intervals do not differ significantly. This indicates that when sufficient data are available, the QRS can still provide relatively accurate predictions.

From Fig. 19, we see that when $N_D$ = 35, the confidence interval for the QRS estimate is much larger than for the case when $N_D$ = 27. Part of the reason for this is that the additional 8 points, which were originally used for a preliminary validation measure by Fong et al. [31], were not selected based on any experimental design considerations; they were chosen more or less arbitrarily, but slightly biased toward the expected failure region. In fact, from Table E.2 of Appendix E, we see that the RVACS inlet temperature ($x_5$) was fixed at 36.85°C (310 K) for each of the simulations. Consequently, during the bootstrap sampling process, there were occasions when the design matrix was ill-conditioned and nearly singular. Hence, it is likely that a portion of the additional uncertainty for this case is due to numerical errors. This further illustrates the importance of a good experimental design for constructing accurate metamodels.

**Table 11. Summary of bootstrapped estimates for FCRR failure probability**

| **Quadratic Response Surface (QRS)** | | |
|---|---|---|
| **Training sample size, $N_D$** | **BBC estimate** | **BBC 95% CI** |
| 27 | 0.0737 | [0.024, 0.284] |
| 35 | 0.0913 | [0.037, 0.395] |
| 62 | 0.2351 | [0.172, 0.296] |

| **Gaussian Process (GP)** | | |
|---|---|---|
| **Training sample size, $N_D$** | **BBC estimate** | **BBC 95% CI** |
| 27 | 0.171 | [0.092, 0.262] |
| 35 | 0.216 | [0.166, 0.257] |
| 62 | 0.193 | [0.170, 0.257] |

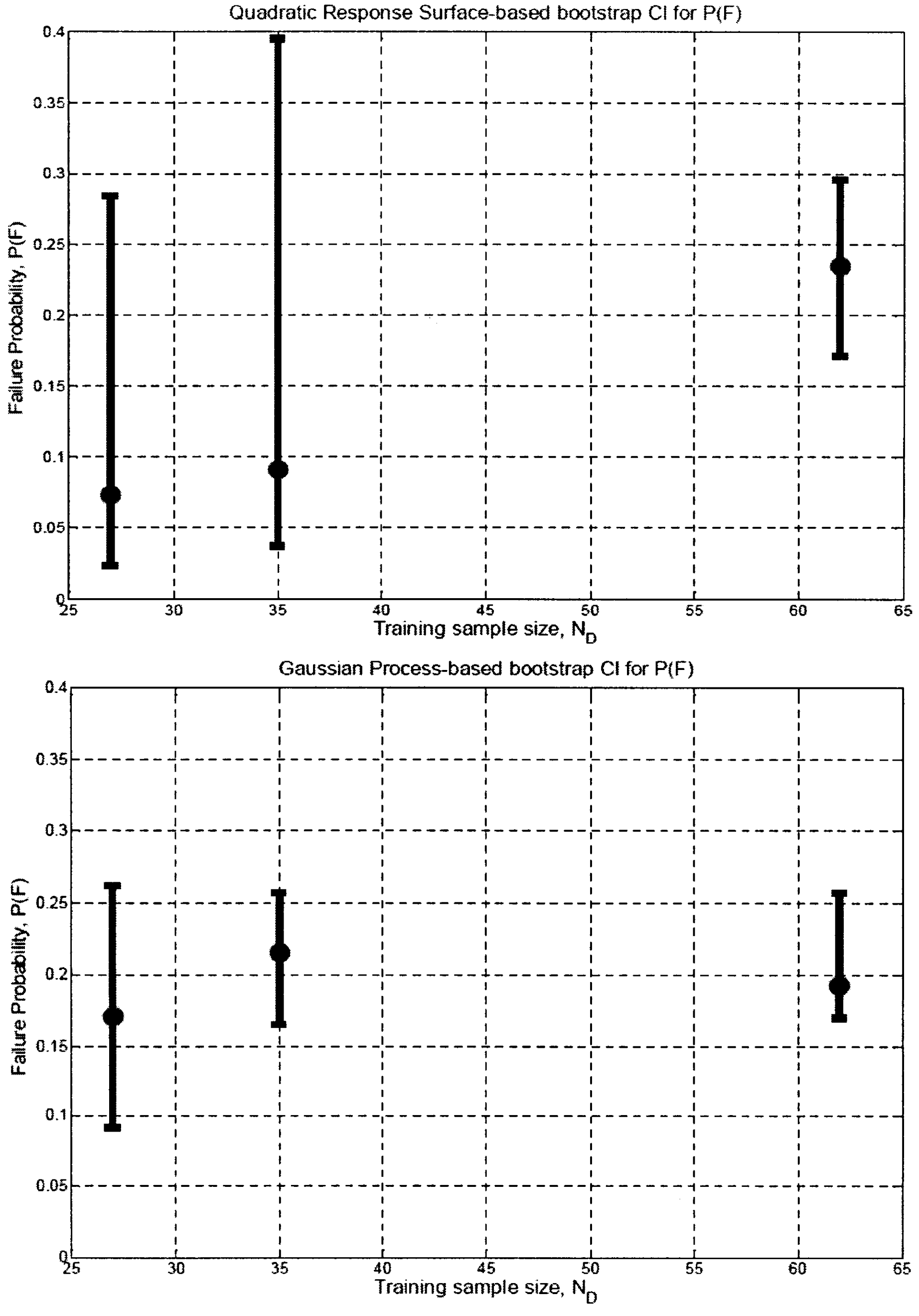

Figure 19. BBC Confidence Intervals for FCRR Failure Probability Estimated with QRS (top) and GP (bottom)

### ***V.2.C Summary and Conclusions from FCRR Case Study***

This section has presented the results from a simple comparative evaluation of QRS and GP metamodels for predicting the outputs from a complex RELAP5-3D T-H model. Two notable differences between this case study and the GFR study are that (i) in the present study, it was not possible to obtain the true estimates of the failure probability due to the prohibitive computational burden posed by the model, and (ii) the failure probability of the FCRR system is much higher than that from the GFR model. Consequently, even for very small sets of design data (i.e., small $N_D$), the GP metamodel was capable of providing relatively accurate estimates of the failure probability; this is to be contrasted with the results from Fig. 15 which illustrate the failure of the GPs to accurately estimate a very small failure probability when few data are available near the failure domain.

The data presented in Tables 8-10 supports our previous claim that neither the coefficient of determination ($R^2$), nor the residual RMSE (RMSRE), is a sufficient measure of the predictive capability of the RS. Clearly, this should serve as a warning against simply choosing the RS that best fits the data, since such a RS may be highly biased. Fortunately, the bootstrapping procedure helps to quantify this bias and can provide confidence intervals for any quantity estimated from the RS (or any metamodel, for that matter). This is good news since it is possible to know whether the estimates from the metamodel are more or less accurate without the need for additional expensive simulations from the model.

# VI CONCLUSIONS

This thesis has presented a critical review of existing methods for performing probabilistic uncertainty and sensitivity analysis (UA and SA, respectively) for complex, computationally expensive simulation models. Chapter I begins by providing definitions and motivating factors for uncertainty and sensitivity analysis (UA and SA, respectively). The focus of this thesis has been on deterministic computer models; thus, any uncertainty in the model output is solely the result of uncertainty in the model outputs. In this context, UA can be regarded as an attempt to determine *what* effect the inputs and their uncertainties have on the model output, while SA is an attempt to determine *how* these inputs and uncertainties affect the model's output. In other words, UA is focused on propagating the input uncertainties through the model, while SA studies the relationship between the inputs and the outputs.

Chapter II presented a detailed discussion of many of the existing methods for UA, including standard Monte Carlo simulation, Latin Hypercube sampling, importance sampling, line sampling, and subset simulation. Many of the relative advantages and drawbacks of these methods have been summarized in Appendix A. In particular, while standard MCS is the most robust method for uncertainty propagation, for many problems of practical interest it presents an unmanageable computational burden. This is because standard MCS often requires an excessive number (i.e., several thousands) of evaluations from the model being studied, and each evaluation may take several hours or days. LHS, IS, LS, and SS are alternative sampling methods that attempt to reduce the number of required samples; however, as outlined in Appendix A, the efficacy of these methods is problem specific and depends on the amount of prior information available to the analyst.

Chapter III followed with a detailed description of various techniques for performing SA, including scatter plots, Monte Carlo filtering, regression analysis, and Sobol' indices. These methods are summarized in Appendix B. Deterministic SA methods, such as Adjoint-based SA method (e.g., GASAP), were not considered in this thesis, but relevant references can be found in Chapter III. Qualitative methods, such as scatter plots, can be useful for revealing important and/or anomalous model behavior, but may not be feasible for large numbers of inputs. Moreover, it may be difficult to discern the effects of parameters that only weakly affect the model output. Regression-based methods, as well as the Sobol' indices, provide more quantitative measures of parameter effects and importance. A distinguishing feature is that regression methods generally study the relationship between the *values* of the model inputs and outputs, while the Sobol' decomposition studies the relationship between the *variances* of the model inputs and outputs. The former may be more useful for design purposes by providing a better understanding of how various parameters affect the response of a system so that design changes can be made. On the other hand, the latter provides information regarding which uncertain inputs should be better understood to most effectively reduce the uncertainty in the output.

The predominant limiting factor in most of the UA and SA methods discussed is the enormous computational burden. As a result, there has been a recent shift in research efforts towards developing better methods for approximating, or emulating, complex computer models. As discussed in Chapter IV, the objective of these methods is to construct a simplified, probabilistic model, called a metamodel, that is capable of approximating the output from the computer model. Once constructed, the metamodel serves as a fast-running surrogate to the computer model and is used to predict outputs from a Monte Carlo simulation. While numerous different approaches to metamodeling have been proposed in the literature, each with unique advantages and disadvantages, in this thesis, we have restricted our attention to Polynomial Response Surfaces, Artificial Neural Networks, and kriging/Gaussian Processes.

Response Surfaces (RS) have been quite popular due to their ease of interpretation and their high computational efficiency. They are easily constructed and can be used to make predictions almost instantaneously. However, RSs have been criticized by numerous authors who have questioned the applicability of these methods for metamodeling deterministic computer models. In particular, these authors have questioned the practice of treating the RS residuals as random variables when they are, in fact, not random. Moreover, the a priori assumption that the computer model's outputs behave as a low-order polynomial is often excessively restrictive and, as a result, the RS metamodel can be highly biased. On the other hand, if this assumption is accurate, one could argue that there is no better method for approximating the model. In addition, even when the RS fails to provide accurate predictions over the entire input domain, they have been found to be useful for identifying overall trends in the outputs, as well as the input parameters that most influence the model response. Nevertheless, when practical considerations require more accurate predictions, RSs are often not capable of living up to these demands due to their inability to adequately represent the complex input/output (I/O) behavior exhibited by many mechanistic models.

Artificial Neural Networks (ANN) are one alternative metamodeling method whose flexibility makes them well-suited for predicting complex I/O behavior, particularly when one must account for multiple outputs. An added benefit is that many software packages currently exist for ANN modeling (e.g., MATLAB® Neural Network Toolbox™). However, as is generally the case, the added flexibility offered by ANNs comes at a higher computational cost; ANNs are "trained" through a nonlinear optimization routine that can be time-consuming, depending on the number of unknown parameters (i.e., weights) that must be determined. In addition, ANNs have been criticized due to their lack of interpretability; they are black-box predictors that admit no analytical expression. Perhaps the biggest disadvantage of ANNs is that a supplementary data set (i.e., the validation set) is necessary to prevent overfitting during the training. This can present difficulties if the available data are limited or if it is not possible to obtain additional data. In the latter case, it will be necessary divide the existing data into a training set and a validation set. Consequently, the available information will not be utilized in its entirety.

Kriging, or Gaussian Processes (GP), possess the unique advantage that they interpolate exactly all of the available data while still providing prediction error estimates for input configurations not contained in the design data. Hence, one could argue that, compared to ANNs and RSs, GPs make more effective use of the available information. This is because, as just noted, ANNs require some of the data to be cast aside (i.e., not directly used for training the ANN) in order to prevent overfitting. On the other hand, RSs smooth the data, and as a result, interesting features in the output may be lost; that is, any deviations from the RS are randomized and treated as statistical noise, even if these deviations are systematic and represent important model behavior. Despite these advantages, however, GPs do present various challenges. Constructing the GP requires the inversion of the design covariance matrix and this can be computationally demanding, depending on its size (i.e., the number of available data). Moreover, it was noted that the covariance matrices are prone to ill conditioning, particularly when design data are in close proximity. In these cases, the GP predictions may be unreliable due to the significant numerical error that is introduced when these matrices are inverted. To prevent this, various numerical techniques must be employed to improve the matrix conditioning.

In addition to the challenges just noted, some applications bring into question the assumptions underlying GP metamodels. Specifically, if the model outputs are bounded (e.g., if the output can only be a positive value), the assumption of normality is strictly invalid since it would give a nonzero probability to negative outputs. Granted, for some applications, the probability of these negative outputs is negligibly small (e.g., if the mean is very large, say 500°C, with a standard deviation of 100°C), so that the normality assumption is acceptable. In other cases, Trans-Gaussian kriging may be used, which requires one to transform the model outputs so that they are approximately normal. An additional assumption that is occasionally invalid is that the random process be stationary, a consequence of which is that the GP outputs are continuous. If this is not the case, then alternative techniques, such as Treed-GPs, must be sought.

In Section IV.5, we discussed the important topic of metamodel uncertainty. Recall that metamodels are, by definition, only approximate representations of the computer models they are intended to emulate. Thus, the predictions given by the metamodel will, in general, differ from the true, yet unknown, model output. The difference between these two quantities represents the metamodel uncertainty, and two approaches were discussed for quantifying this uncertainty. The first approach, which we referred to as Bootstrap Bias-Corrected (BBC) metamodeling, is a distribution-free, brute force sampling scheme that requires the construction of multiple metamodels built from data that are bootstrapped (i.e., sampled with replacement) from the original design data. The results from this bootstrapping procedure can be used to correct for any bias in the metamodel predictions. Furthermore, this procedure provides confidence intervals for any quantity estimated from the metamodel. The primary drawback of this method is the increased computational demand resulting from the need to construct multiple (e.g., 500-1000) metamodels. On the other hand, the procedure is relatively straightforward and can be easily implemented.

The second technique that was discussed is based on a Bayesian interpretation of the GP metamodels. By assigning appropriate prior distributions to the parameters of the GP model (e.g., the regression coefficients and process variance), one can obtain an expression for the posterior probability distribution of the model outputs, conditional on the observed data (i.e., the design data). Conceptually, this approach is very intriguing since the resulting PDF is a distribution over a set of possible surfaces, each of which interpolates the known data. Hence, it is reasonable to suppose that one of these surfaces is, in fact, representative of the true model being approximated. It can be argued that this approach leads to a more intuitive representation of the metamodel uncertainty; specifically, the prediction problem becomes one of updating one's state of knowledge (e.g., by performing more simulations with the computer model) in order to identify which of the possible surfaces is most likely the correct metamodel. Similarly with the BBC approach, however, the Bayesian approach requires additional computational demand. In particular, it seems that analytical expressions for the posterior PDF of the model outputs can only be obtained when priors are assigned to the regression coefficients and the process variance. More generally, one should also account for prior uncertainty in the correlation parameters. Unfortunately, this greatly increases the complexity of the metamodel, and advanced Markov Chain Monte Carlo (MCMC) algorithms are required to sample from the posterior distribution. In addition, choosing an appropriate prior distribution to the correlation parameters seems to be an ongoing challenge. It has been found that, in some cases, standard noninformative (i.e., diffuse) priors on the correlation parameters lead to an improper posterior distribution for the model outputs. Although there have been some recent developments in identifying diffuse priors that yield proper posterior distributions, these distributions are rather complicated and their implementation in MCMC algorithms seems challenging. Currently, it seems that there are no existing and/or readily available software packages for performing fully Bayesian GP metamodeling; hence, this technique was not utilized in this work. Nevertheless, this approach is certainly an important subject for future research.

Chapter V presented the results from two case studies that were carried out to compare the performance of several metamodeling methods (RSs, ANNs, and GPs) for various UA and SA objectives. In both of these studies, it was found that GPs were capable of better predicting the outputs of the computer models compared to RSs, and, although not considered in the second case study (i.e., the FCRR study), similar conclusions could be made regarding ANNs. These results are particularly true when the design data are limited. In the GFR study, it was found that ANNs and GPs provide more consistent, and more accurate (i.e., with smaller confidence intervals), predictions of the 95th percentiles of the model outputs. We note that for the GFR case study, it was possible to obtain "exact" estimates of the relevant quantities using standard MCS with the original model. As the number of data was increased, the point estimates for the 95th percentiles given by the ANNs and GPs converged to the "true" percentiles, and confidence intervals for these estimates became increasingly narrow. This indicates the possibility for these methods to make more or less exact predictions provided sufficient data are available. On the contrary, because of their assumed functional form, RS metamodels cannot, in general, provide exact estimates, regardless of the number of available data; in other words, the confidence interval width will not decrease indefinitely as additional data are obtained, unless, of course, the model output is truly quadratic (or whatever the order of the RS).

For estimating the failure probability of the GFR, RSs were found to perform favorably, only slightly worse than ANNs. This is because accurate predictions of the model outputs are necessarily needed to accurately predict the failure probability. For instance, in the case of the GFR study, the metamodel need not accurately predict the core outlet temperature so long as it can correctly classify these outputs as either successes or failures. On the other hand, GPs were found to perform rather poorly when few design data were available. This is because GPs make predictions based on the nearest data, and for a system whose failure probability is very small, it is likely that no data points will be near the failure domain. Consequently, the GP will be a poor predictor in this region of the input space. As expected, however, when the number of data is increased, the GPs were better able to estimate the failure probability. In these cases, all of the metamodels performed comparably.

ANNs and RSs were also compared based on their ability to estimate the first-order Sobol' sensitivity coefficients for the GFR model inputs. Once again, ANNs were found to be superior, consistently providing more accurate BBC point-estimates as well as narrower confidence intervals for these estimates. Nevertheless, RSs performed reasonably well, and given that RSs are considerably simpler than ANNs, a strong case could be made for using RSs for SA. It is generally recognized that RSs are effective for identifying the overall interactions, or trends, between the input parameters and the model outputs.

Finally, the FCRR study presented a comparison of RSs and GPs for predicting the outputs of a complex RELAP5-3D thermal-hydraulic (T-H) model. For this study, each simulation of the RELAP model required approximately 30+ hours, so it was not possible to perform direct MCS with the RELAP model to obtain "true" estimates. An evaluation of the predictive performance of RSs built from three different sized data sets revealed many interesting insights. Namely, it was found that the RS that best fits the data is not necessarily the most accurate metamodel. This is a consequence of overfitting, which results in the RS being a biased predictor. The PRESS statistic was found to provide reasonable indication as to whether the RS is overly biased. Moreover, the bootstrapping procedure provided further indication that the RSs constructed from the two smallest data sets were highly biased and, therefore, poor predictive metamodels. On the other hand, GPs were found to perform significantly better for these small data sets. For these cases, the BBC point-estimates for the failure probability computed with GPs were more consistent and closer to what is expected to be the true failure probability. For the largest of the design data sets, GPs still outperformed RSs, but the discrepancy between the two models was less apparent.

All of these results suggest that metamodels can be effective tools for performing UA and SA when the model under study is computationally prohibitive. Specifically, it was found that reasonable estimates for various quantities (e.g., failure probabilities, percentiles, and sensitivity indices) could be estimated while requiring only a relatively small number (i.e., less than 100) of evaluations from the model. Thus, metamodels can provide an enormous increase in computational efficiency compared to direct MCS (i.e., MCS performed directly with the computer model). Even in comparison with advanced sampling techniques, metamodels seem more efficient. Note that joint applications of metamodeling and advanced sampling methods would likely lead to further increases in efficiency. Still, it is important to recognize that metamodels are only an approximation to the underlying computer model, and their use introduces an additional source of uncertainty that must be accounted for. The results from the case studies indicate that bootstrapping may be an effective technique for quantifying this uncertainty. Most importantly, perhaps, is that bootstrapping provides a clear indication as to when the metamodel is not a reliable surrogate for the computer model.

# VII References

1. Kaplan, S., Garrick, B.J., (1981), "On the Quantitative Definition of Risk," *Risk Analysis*, Vol. 1, No. 1, pp. 11-27.
2. Vesely, W.E., Rasmuson, D.M., (1984), "Uncertainties in Nuclear Probabilistic Risk Analyses," *Risk Analysis*, Vol. 4, 4, pp. 313-322.
3. Campbell, J.E., Cranwell, R.M., (1988), "Performance Assessment of Radioactive Waste Repositories," *Science*, Vol. 239, pp. 1389-1392.
4. Apostolakis, G.E., (1988), "The Interpretation of Probability in Probabilistic Safety Assessments," *Reliability Engineering & System Safety*, Vol. 23, 4, pp. 247-252.
5. Apostolakis, G.E., (1989), "Uncertainty in Probabilistic Safety Assessment," *Nuclear Engineering and Design*, Vol. 115, 1, pp. 173-179.
6. Helton, J.C., (1993), "Uncertainty and Sensitivity Analysis Techniques for Use in Performance Assessment for Radioactive Waste Disposal," *Reliability Engineering & System Safety*, Vol. 42, 2-3, pp. 327-367.
7. Helton, J.C., Davis, F.J., Johnson, J.D., (2005), "A Comparison of Uncertainty and Sensitivity Analysis Results Obtained with Random and Latin Hypercube Sampling," *Reliability Engineering & System Safety*, Vol. 89, 3, pp. 305-330.
8. Helton, J.C., Johnson, J.D., Sallaberry, C.J., Storlie, C.B., (2006), "Survey of Sampling-Based Methods for Uncertainty and Sensitivity Analysis," *Reliability Engineering & System Safety*, Vol. 91, 10-11, pp. 1175-1209.
9. Helton, J.C., (2004), "Alternative Representations of Epistemic Uncertainties," *Reliability Engineering & System Safety*, Vol. 85, 1-3, pp. 23-69.
10. Saltelli, A., Tarantola, S., Campolongo, F., (2000), "Sensitivity Analysis as an Ingredient of Modeling," *Statistical Science*, Vol. 15, 4, pp. 377-395.
11. Saltelli, A., (2002), "Sensitivity Analysis for Importance Assessment," *Risk Analysis*, Vol. 22, 3, pp. 579-590.
12. Simpson, T.W., Peplinski, J., Koch, P.N., Allen, J.K., (2001), "Metamodels for Computer-Based Engineering Design: Survey and Recommendations," *Engineering with Computers*, 17, pp. 129-150.
13. Oakley, J., O'Hagan, A., (2002), "Bayesian Inference for the Uncertainty Distribution of Computer Model Outputs," *Biometrika*, Vol. 84, 4, pp. 769-784.
14. Marrel, A., Iooss, B., Dorpe, F.V., Volkova, E., (2008), "An Efficient Methodology for Modeling Complex Computer Codes with Gaussian Processes," *Computational Statistics & Data Analysis*, 52, pp. 4731-4744.
15. Storlie, C.B., Swiler, L.P., Helton, J.C., Sallaberry, C.J., (2009), "Implementation and Evaluation of Nonparametric Regression Procedures for Sensitivity Analysis of Computationally Demanding Models," *Reliability Engineering & System Safety*, 94, pp. 1735-1763.
16. Apostolakis, G.E., (1990), "The Concept of Probability in Safety Assessment of Technological Systems," *Science*, Vol. 250, 4986, pp. 1359-1394.
17. Helton, J.C., Johnson, J.D., Oberkampf, W.L., (2004), "An Exploration of Alternative Approaches to the Representation of Uncertainty in Model Predictions," *Reliability Engineering & System Safety*, Vol. 85, 1-3, pp. 39-71.

18. O'Hagan, A., Oakley, J.E., (2004), "Probability is Perfect, But We Can't Elicit It Perfectly," *Reliability Engineering & System Safety*, Vol. 85, 1-3, pp. 239-248.
19. Kennedy, M.C., O'Hagan, A., (2001), "Bayesian Calibration of Computer Models," *J. of the Royal Statistical Society B*, 63, Part 3, pp. 425-464.
20. Liel, A.B., Haselton, C.B., Deierlein, G.G., Baker, J.W., (2009), "Incorporating Modeling Uncertainties in the Assessment of Seismic Collapse Risk of Buildings," *Structural Safety*, Vol. 31, 2, pp. 197-211.
21. Schuëller, G.I., (2007), "On the Treatment of Uncertainties in Structural Mechanics and Analysis," *Computers and Structures*, 85, pp. 235-243.
22. Ditlevsen, O., Madsen, H.O., (1996), Structural Reliability Methods, John Wiley & Sons.
23. Melchers, R.E., (1999), Structural Reliability Analysis and Prediction, 2nd Ed., Wiley.
24. Choi, S.K., Grandhi, R.V., Canfield, R.A., (2007), Reliability-Based Structural Design, Springer.
25. Jafari, J., D'Auria, F., Kazeminejad, H., Davilu, H., (2003), "Reliability Evaluation of a Natural Circulation System," *Nuclear Engineering and Design*, 224, pp. 79-104.
26. Marquès, M., Pignatel, J.F., Saignes, P., D'Auria, F., Burgazzi, L., Müller, C., Bolado-Lavin, R., Kirchsteiger, C., La Lumia, V., Ivanov, I., (2005), "Methodology for the Reliability Evaluation of a Passive System and Its Integration Into a Probabilistic Safety Assessment," *Nuclear Engineering and Design*, 235, pp. 2612-2631.
27. Mackay, F.J., Apostolakis, G.E., Hejzlar, P., (2008), "Incorporating Reliability Analysis Into the Design of Passive Cooling Systems with an Application to a Gas-Cooled Reactor," *Nuclear Engineering and Design*, 238, pp. 217-228.
28. Mathews, T.S., Ramakrishnan, M., Parthasarathy, U., John Arul, A., Senthil Kumar, C., (2008), "Functional Reliability Analysis of Safety Grade Decay Heat Removal System of Indian 500 MWe PFBR," *Nuclear Engineering and Design*, 238, pp. 2369-2376.
29. Pagani, L., Apostolakis, G.E., and Hejzlar, P., (2005), "The Impact of Uncertainties on the Performance of Passive Systems," *Nuclear Technology*, 149, pp. 129-140.
30. Patalano, G., Apostolakis, G.E., Hejzlar, P., (2008), "Risk-Informed Design Changes in a Passive Decay Heat Removal Systems," *Nuclear Technology*, 163, pp. 191-208.
31. Fong, C.J., Apostolakis, G.E., Langewisch, D.R., Hejzlar, P., Todreas, N.E., Driscoll, M.J., (2009), "Reliability Analysis of a Passive Cooling System Using a Response Surface with an Application to the Flexible Conversion Ratio Reactor," *Nuclear Engineering and Design*, doi:10.1016/j.nucengdes.2009.07.008.
32. Kuo, F., Sloan, I.H., (2005), "Lifting the Curse of Dimensionality," *Notices of the American Mathematical Society*, Vol. 52, 11, pp. 1320-1328.
33. Hammersley, J.M., Handscomb, D.C., (1965), Monte Carlo Methods, John Wiley & Sons, New York.
34. McKay, M.D., Beckman, R.J., Conover, W.J., (1979), "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, 21, pp. 239-245.
35. Iman, R.L., Helton, J.C., Campbell, J.E., (1981), "An Approach to Sensitivity Analysis of Computer Models, Part 1: Introduction, Input Variable Selection and Preliminary Variable Assessment," *Journal of Quality Technology*, Vol. 13, 3, pp. 174-183.
36. Olsson, A., Sabdberg, G., Dahlblom, O., (2003), "On Latin Hypercube Sampling for Structural Reliability Analysis," *Structural Safety*, 25, pp. 47-68.

37. Helton, J.D., Davis, F.J., (2003), "Latin Hypercube Sampling and the Propagation of Uncertainty in Analyses of Complex Systems," *Reliability Engineering & System Safety*, 81, pp. 23-69.
38. Sallaberry, C.J., Helton, J.C., Hora, S.C., (2008), "Extension of Latin Hypercube Samples with Correlated Variables," *Reliability Engineering & System Safety*, Vol. 93, 7, pp. 1047-1059.
39. Pebesma, E.J., Heuvelink, G.B.M., (1999), "Latin Hypercube Sampling of Gaussian Random Fields," *Technometrics*, Vol. 41, 4, pp. 203-212.
40. Koutsourelakis, P.S., Pradlwarter, H.J., Schueller, G.I., (2004), "Reliability of Structures in High Dimensions, Part 1: Algorithms and Applications," *Probabilistic Engineering Mechanics*, 19, pp. 409-417.
41. Pradlwarter, H.J., Pellissetti, M.F., Schenk, C.A., Schuëller, G.I., Kreis, A., Fransen, S., Calvi, A., Klein, M., (2005), "Realistic and Efficient Reliability Estimation for Aerospace Structures," *Computer Methods in Applied Mechanics and Engineering*, Vol. 194, 12-16, pp. 1597-1617.
42. Pradlwarter, H.J., Schuëller, G.I., Koutsourelakis, P.S., Charmpis, D.C., (2007), "Application of the Line Sampling Method to Reliability Benchmark Problems," *Structural Safety*, 29, pp. 208-221.
43. Schuëller, G.I., Pradlwarter, H.J., (2007), "Benchmark Study on Reliability Estimation in Higher Dimensions of Structural Systems – An Overview," *Structural Safety*, 29, pp. 167-182.
44. Zio, E., Pedroni, N., (2009), "Functional Failure Analysis of a Thermal-Hydraulic Passive System by means of Line Sampling," *Reliability Engineering & System Safety*, 94, pp. 1764-1781.
45. Schuëller, G.I., Pradlwarter, H.J., Koutsourelakis, P.S., (2004), "A Critical Appraisal of Reliability Estimation Procedures for High Dimensions," *Probabilistic Engineering Mechanics*, 19, pp. 463-474.
46. Au, S.K., Beck, J.L., (2001), "Estimation of Small Failure Probabilities in High Dimensions by Subset Simulation," *Probabilistic Engineering Mechanics*, Vol. 16, 4, pp. 263-277.
47. Au, S.K., Beck, J.L., (2003), "Subset Simulation and its Application to Seismic Risk Based on Dynamic Analysis," *Journal of Engineering Mechanics*, Vol. 129, 8, pp. 1-17.
48. Zio, E., Pedroni, N., (2009), "Estimation of the Functional Failure Probability of a Thermal-Hydraulic Passive System by Subset Simulation," *Nuclear Engineering and Design*, Vol. 239, 3, pp. 580-599.
49. Haldar, A., Mahadevan, S., (2000), Reliability Assessment Using Stochastic Finite Element Analysis, Wiley, New York.
50. Rosenblatt, M., (1952), "Remarks on Multivariate Transformations," *Annals of Mathematical Statistics*, Vol. 23, 3, pp. 470-472.
51. Lebrun, R., Dutfoy, A., (2009), "An Innovating Analysis of the Nataf Transformation from the Copula Viewpoint," *Probabilistic Engineering Mechanics*, Vol. 24, 3, pp. 312-320.
52. Huang, B., Du, X., (2006), "A Robust Design Method Using Variable Transformation and Gauss-Hermite Integration," *Int. J. for Numerical Methods in Engineering*, 66, pp. 1841-1858.
53. Qin, Q., Lin, D., Mei, G., Chen, H., (2006), "Effects of Variable Transformations on Errors in FORM Results," *Reliability Engineering & System Safety*, 91, pp. 112-118.

54. Cacuci, D.G., Ionescu-Bujor, M., (2004), "A Comparative Review of Sensitivity and Uncertainty Analysis of Large-Scale Systems – I: Deterministic Methods," *Nuclear Science & Engineering*, Vol. 147, 3, pp. 189-203.
55. Cacuci, D.G., Ionescu-Bujor, M., (2004), "A Comparative Review of Sensitivity and Uncertainty Analysis of Large-Scale Systems – II: Statistical Methods," *Nuclear Science & Engineering*, Vol. 147, 3, pp. 204-217.
56. Zhao, H., Mousseau, V.A., (2008), "Extended Forward Sensitivity Analysis for Uncertainty Quantification," Idaho National Laboratory Report INL/EXT-08-14847.
57. Cacuci, D.G., Ionescu-Bujor, M., (2000), "Adjoint Sensitivity Analysis of the RELAP5/MOD3.2 Two-Fluid Thermal-Hydraulic Code System – I: Theory," *Nuclear Science & Engineering*, Vol. 136, 1, pp. 59-84.
58. Ionescu-Bujor, M., Cacuci, D.G., (2000), "Adjoint Sensitivity Analysis of the RELAP5/MOD3.2 Two-Fluid Thermal-Hydraulic Code System – II: Applications," *Nuclear Science & Engineering*, Vol. 136, 1, pp. 85-121.
59. Cacuci, D.G., (2003), Sensitivity & Uncertainty Analysis, Volume 1: Theory, Chapman & Hall/CRC.
60. Iman, R.L., (1987), "A Matrix-Based Approach to Uncertainty and Sensitivity Analysis for Fault Trees," *Risk Analysis*, Vol. 7, 1, pp. 21-33.
61. Kocher, D.C., Ward, R.C., Killough, G.G., Bunning, D.E., Hicks, B.B., Hosker, R.P., Ku, J.Y., Rao, K.S., (1987), "Sensitivity and Uncertainty Studies of the CRAC2 Computer Code," *Risk Analysis*, Vol. 7, 4, pp. 497-507.
62. Modarres, M., Cadman, T.W., Lois, E., Gardner, A.R., (1985), "Sensitivity and Uncertainty Analyses for the Accident Sequence Precursor Study," *Nuclear Technology*, 69, pp. 27-35.
63. Andsten, R.S., Vaurio, J.K., (1992), "Sensitivity, Uncertainty, and Importance Analysis of a Risk Assessment," *Nuclear Technology*, 98, pp. 160-170.
64. Iooss, B., Dorpe, F.V., Devictor, N., (2006), "Response Surfaces and Sensitivity Analyses for an Environmental Model of Dose Calculations," *Reliability Engineering & System Safety*, Vol. 91, 10-11, pp. 1241-1251.
65. Homma, T., Tomita, K., Hato, S., (2005), "Uncertainty and Sensitivity Studies with the Probabilistic Accident Consequence Assessment Code OSCAAR," *Nuclear Engineering and Technology*, Vol. 37, 3, pp. 245-257.
66. Turányi, T., (1990), "Sensitivity Analysis of Complex Kinetic Systems – Tools and Applications," *Journal of Mathematical Chemistry*, 5, pp. 203-248.
67. Hamby, D.M., (1994) "A Review of Techniques for Parameter Sensitivity Analysis of Environmental Models," *Environmental Monitoring and Assessment*, 32, pp. 135-154.
68. Saltelli, A., Chan, K., Scott, E.M., (2000), Sensitivity Analysis, John Wiley & Sons, Chichester.
69. Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., Tarantola, S., (2008), Global Sensitivity Analysis: The Primer, John Wiley & Sons.
70. Ghosh, S.T., Apostolakis, G.E., (2006), "Extracting Risk Insights from Performance Assessments for High-Level Radioactive Waste Repositories," *Nuclear Technology*, Vol. 153, pp. 70-88.
71. Storlie, C.B., Helton, J.C., (2008), "Multiple Predictor Smoothing Methods for Sensitivity Analysis: Description of Techniques," *Reliability Engineering & System Safety*, Vol. 93, pp. 28-54.

72. Storlie, C.B., Helton, J.C., (2008), "Multiple Predictor Smoothing Methods for Sensitivity Analysis: Example Results," *Reliability Engineering & System Safety*, Vol. 93, pp. 55-77.
73. Box., G.E.P., Draper, N.R., (2007), Response Surfaces, Mixtures, and Ridge Analysis, 2nd Ed., Wiley Series in Probability and Statistics.
74. Seber, G.A.F., Wild, C.J., (2003), Nonlinear Regression, Wiley Series in Probability & Statistics.
75. Bates, D.M., Watts, D.G., (2007), Nonlinear Regression Analysis and Its Applications, Wiley Series in Probability & Statistics.
76. Homma, T., Saltelli, A., (1996), "Importance Measures in Global Sensitivity Analysis on Nonlinear Models," *Reliability Engineering & System Safety*, Vol. 52, 1, pp. 1-17.
77. Oakley, J.E., O'Hagan, A., (2004), "Probabilistic Sensitivity Analysis of Complex Models: A Bayesian Approach," *J. of the Royal Statistical Society B*, Vol. 66, 3, pp. 751-769.
78. Marrel, A., Iooss, B., Laurent, B., Roustant, O., (2009), "Calculations of Sobol' Indices for the Gaussian Process Metamodel," *Reliability Engineering & System Safety*, Vol. 94, pp. 742-751.
79. Saltelli, A., (2002), "Making Best Use of Model Evaluations to Compute Sensitivity Indices," *Computer Physics Communications*, Vol. 145, pp. 280-297.
80. Saltelli, A., Tarantola, S., Chan, K., (1999), "A Quantitative Model-Independent Method for Global Sensitivity Analysis of Model Output," *Technometrics*, Vol. 41, 1, pp. 39-56.
81. Saltelli, A., Bolado, R., (1998), "An Alternative Way to Compute Fourier Amplitude Sensitivity Test (FAST)," *Computational Statistics & Data Analysis*, Vol. 26, pp. 445-460.
82. Daniel, C., (1973), "One-at-a-Time Plans," *J. of the American Statistical Association*, Vol. 68, 342, pp. 353-360.
83. Daniel, C., (1994), "Factorial One-Factor-at-a-Time Experiments," *The American Statistician*, Vol. 48, 2, pp. 132-135.
84. Morris, M.D., (1991), "Factorial Sampling Plans for Preliminary Computational Experiments," *Technometrics*, Vol. 33, 2, pp. 161-174.
85. Campolongo, F., Cariboni, J., Saltelli, A., (2007), "An Effective Screening Design for Sensitivity Analysis of Large Models," *Environmental Modelling & Software*, Vol. 22, 10, pp. 1509-1518.
86. Wan., H., Ankenman, B.E., Nelson, B.L., (2006), "Controlled Sequential Bifurcation: A New Factor-Screening Method for Discrete Event Simulation," *Operations Research*, Vol. 54, 4, pp. 743-755.
87. Bettonvil, B., Kleijnen, J.P.C., (1996), "Searching for Important Factors in Simulation Models with Many Factors: Sequential Bifurcation," *European Journal of Operations Research*, Vol. 96, pp. 180-194.
88. Kleijnen, J.P.C., (2009), "Factor Screening in Simulation Experiments: A Review of Sequential Bifurcation," from Advancing the Frontiers of Simulation, International Series in Operations Research & Management Science, SpringerLink.
89. Shen, H., Wan, H., (2009), "Controlled Sequential Factorial Design for Simulation Factor Screening," *European Journal of Operations Research*, Vol. 198, 2, pp. 511-519.
90. Downing, D.J., Gardner, R.H., Hoffman, F.O., (1985), "An Examination of Response-Surface Methodologies for Uncertainty Analysis in Assessment Models," *Technometrics*, Vol. 27, 2, pp. 151-163.
91. Sacks, J., Schiller, S.B., Welch, W.J., (1989), "Designs for Computer Experiments," *Technometrics*, Vol. 31, pp. 41-47.

92. Mitchell, T.J., Morris, M.D., (1992), "Bayesian Design and Analysis of Computer Experiments: Two Examples," *Statistica Sinica 2*, pp. 359-379.
93. Brandyberry, M., Apostolakis, G., (1990), "Response Surface Approximation of a Fire Risk Analysis Computer Code," *Reliability Engineering & System Safety*, Vol. 29, pp. 153-184.
94. Haylock, R., O'Hagan, A., (1996), "On Inference for Outputs of Computationally Expensive Algorithms with Uncertainty on the Inputs," Bayesian Statistics 5 (J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith, eds.), Oxford University Press, pp. 629-637.
95. Haylock, R., O'Hagan, A., (1997), "Bayesian Uncertainty Analysis and Radiological Protection," Statistics for the Environment 3: Pollution Assessment and Control (V. Barnett and K.F. Turkman, eds.), John Wiley & Sons Ltd.
96. Iooss, B., Van Dorpe, F., Devictor, N., (2006), "Response Surfaces and Sensitivity Analyses for an Environmental Model of Dose Calculations," *Reliability Engineering & System Safety*, Vol. 91, pp. 1241-1251.
97. Volkova, E., Iooss, B., Van Dorpe, F., (2008), "Global Sensitivity Analysis for a Numerical Model of Radionuclide Migration from the RRD 'Kurchatov Institute' Radwaste Disposal Site," in Stochastic Environmental Research and Risk Assessment, Vol. 22, 1, Springer Berlin.
98. Jones, D.R., Schonlau, M., Welch, W.J., (1998), "Efficient Global Optimization of Expensive Black-Box Functions," *J. of Global Optimization*, Vol. 13, pp. 455-492.
99. Jin, R., Du, X., Chen, W., (2003), "The Use of Metamodeling Techniques for Optimization Under Uncertainty," *Structural & Multidisciplinary Optimization*, Vol. 25, pp. 99-116.
100. Gano, S.E., Kim, H., Brown II, D.E., (2006), "Comparison of Three Surrogate Modeling Techniques: Datascape® , Kriging, and Second Order Regression," *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference.*
101. Gramacy, R.B., Lee, H.K.H., (2008), "Bayesian Treed Gaussian Process Models with an Application to Computer Modeling," *J. of the American Statistical Association*, Vol. 103, 483, pp. 1119-1130.
102. Shahsavani, D., Grimvall, A., (2009), "An Adaptive Design and Interpolation Technique for Extracting Highly Nonlinear Response Surfaces from Deterministic Models," *Reliability Engineering & System Safety*, Vol. 94, pp. 1173-1182.
103. O'Hagan, A., Kennedy, M.C., Oakley, J.E., (1998), "Uncertainty Analysis and Other Inference Tools for Complex Computer Codes," Bayesian Statistics 6 (J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith, etd.), Oxford University Press, pp. 503-524.
104. Craig, P.S., Goldstein, M., Rougier, J.C., Seheult, A.H., (2001), "Bayesian Forecasting for Complex Systems Using Computer Simulators," *J. of the American Statistical Association*, Vol. 96, 454, pp. 717-729.
105. Oakley, J., (2004), "Estimating Percentiles of Uncertain Computer Code Outputs," *J. of the Royal Statistical Society C (Applied Statistics)*, Vol. 53, 1, pp. 83-93.
106. Iooss, B., Ribatet, M., (2009), "Global Sensitivity Analysis of Computer Models with Functional Inputs," *Reliability Engineering & System Safety*, Vol. 94, pp. 1194-1204.
107. Kennedy, M.C., Anderson, C.W., Conti, S., O'Hagan, A., (2006), "Case Studies in Gaussian Process Modelling of Computer Codes," *Reliability Engineering & System Safety*, Vol. 91, pp. 1301-1309.

108. Buratti, N., Ferracuti, B., Savoia, M., (2009), "Response Surface with Random Factors for Seismic Fragility of Reinforced Concrete Frames," *Structural Safety*, doi:10.1016/j.strusafe.2009.06.003.
109. Gavin, H.P., Yau, S.C., (2008), "High-Order Limit State Functions in the Response Surface Method for Structural Reliability Analysis," *Structural Safety*, Vol. 30, pp. 162-179.
110. Liel, A.B., Haselton, C.B., Deierlein, G.G., Baker, J.W., (2009), "Incorporating Modeling Uncertainties in the Assessment of Seismic Collapse Risk of Buildings," *Structural Safety*, Vol. 31, 2, pp. 197-211.
111. Deng, J., (2006), "Structural Reliability Analysis for Implicit Performance Function Using Radial Basis Function Network," *Int. J. of Solids & Structures*, Vol. 43, pp. 3255-3291.
112. Cardoso, J.B., De Almeida, J.R., Dias, J.M., Coelho, P.G., (2008), "Structural Reliability Analysis Using Monte Carlo Simulation and Neural Networks," *Advances in Engineering Software*, Vol. 39, pp. 505-513.
113. Cheng, J., Li, Q.S., Xiao, R.C., (2008), "A New Artificial Neural Network-Based Response Surface Method for Structural Reliability Analysis," *Probabilistic Engineering Mechanics*, Vol. 23, pp. 51-63.
114. Hurtado, J.E., (2007), "Filtered Importance Sampling with Support Vector Margin: A Powerful Method for Structural Reliability Analysis," *Structural Safety*, Vol. 29, pp. 2-15.
115. Bucher, C., Most, T., (2008), "A Comparison of Approximate Response Functions in Structural Reliability Analysis," *Probabilistic Engineering Mechanics*, Vol. 23, pp. 154-163.
116. Bishop, C.M., (1996), Neural Networks for Pattern Recognition, Oxford University Press.
117. Demuth, H., Beale, M., Hagan, M., (2009), MATLAB® Neural Network Toolbox™ User's Manual, available online at http://www.mathworks.com/access/helpdesk/help/pdf_doc/nnet/nnet.pdf
118. Le, N.D., Zidek, J.V., (2006), Statistical Analysis of Environmental Space-Time Processes, Springer Series in Statistics, New York.
119. Banerjee, S., Carlin, B.P., Gelfand, A.E., (2004), Hierarchical Modeling and Analysis for Spatial Data, Chapman & Hall/CRC.
120. Box, G.E.P., Wilson, K.B., (1951), "On the Experimental Attainment of Optimal Conditions," *J. of the Royal Statistical Society B*, Vol. 13, 1, pp. 1-45.
121. Myers, R.H., Montgomery, D.C., Anderson-Cook, C.M., (2009), Response Surface Methodology: Process and Product Optimization Using Design Experiments, 3rd Ed., Wiley.
122. Shao, J., (1993), "Linear Model Selection by Cross-Validation," *J. of the American Statistical Society*, Vol. 88, 422, pp. 486-494.
123. Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P., (1989), "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, 4, pp. 409-435.
124. Simpson, T.W., Peplinski, J.D., Koch, P.N., Allen, J.K., (1997), "On the Use of Statistics in Design and the Implications for Deterministic Computer Experiments," *Proc. of DETC'97, ASME Design Engineering Technical Conferences*, No. DETC97DTM3881, Sept. 14-17, Sacramento, CA.
125. Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P., (1989), "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, 4, pp. 409-435.
126. Santner, T.J., Williams, B.J., Notz, W.I., (2003), The Design and Analysis of Computer Experiments, Springer Series in Statistics, New York.

127. Simpson, T.W., Peplinski, J.D., Koch, P.N., Allen, J.K., (1997), "On the Use of Statistics in Design and the Implications for Deterministic Computer Experiments," *Proc. of DETC'97, ASME Design Engineering Technical Conferences*, No. DETC97DTM3881, Sept. 14-17, Sacramento, CA.

128. van Beers, W.C.M., Kleijnen, J.P.C., (2004), "Kriging Interpolation in Simulation: A Survey," *Proc. of the 36th Conference on Winter Simulation*, Dec. 05-08, Washington, D.C.

129. Iman, R.L., Helton, J.C., (1988), "An Investigation of Uncertainty and Sensitivity Analysis Techniques for Computer Models," *Risk Analysis*, Vol. 8, pp. 71-90.

130. Currin, C., Mitchell, T., Morris, M., Ylvisaker, D., (1991), "Bayesian Prediction of Deterministic Functions, with Applications to the Design and Analysis of Computer Experiments," *J. of the American Statistical Association*, Vol. 86, 416, pp. 953-963.

131. Welch, W.J., Buck, R.J., Sacks, J., Wynn, H.P., Mitchell, T.J., Morris, M.D., (1992), "Screening, Predicting and Computer Experiments," *Technometrics*, Vol. 34, 1, pp. 15-25.

132. Koehler, J.R., Owen, A.B., (1996), "Computer Experiments," Handbook of Statistics 13, (S.Ghosh and C.R.Rao, eds.), Elsevier, pp. 261-308.

133. Matheron, G., (1963), "Principles of Geostatistics," *Economic Geology*, Vol. 58, pp. 1246-1266.

134. Cressie, N., (1993), Statistics for Spatial Data, Wiley, New York.

135. Gramacy, R., Lee, H.K.H., (2008), "Gaussian Processes and Limiting Linear Models," *Computational Statistics & Data Analysis*, Vol. 53, pp. 123-136.

136. Lophaven, S.N., Nielsen, H.B., Søndergaard, J., (2002), "Aspects of the MATLAB® Toolbox DACE," Technical Report No. IMM-REP-2002-13, Technical University of Denmark, available online at http://www2.imm.dtu.dk/~hbn/dace/.

137. Stein, M.L., (1999), Interpolation of Spatial Data: Some Theory for Kriging, Springer Series in Statistics, New York.

138. Berger, J.O., De Oliveira, V., Sansó, B., (2001), "Objective Bayesian Analysis of Spatially Correlated Data," *J. of the American Statistical Association*, Vol. 96, 456, pp. 1361-1374.

139. Abramowitz, M., Stegun, I.A., (1965), Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, Dover.

140. Anderson, T.W., (2003), An Introduction to Multivariate Statistical Analysis, 3rd Ed., Wiley Series in Probability and Statistics.

141. Kitanidis, P.K., (1986), "Parameter Uncertainty in Estimation of Spatial Functions: Bayesian Analysis," *Water Resources Research*, Vol. 22, 4, pp. 499-507.

142. Li, R., Sudjianto, A., (2005), "Analysis of Computer Experiments Using Penalized Likelihood in Gaussian Kriging Models," *Technometrics*, Vol. 47, 2, pp. 111-120.

143. Paulo, R., (2005), "Default Priors for Gaussian Processes," *The Annals of Statistics*, Vol. 33, 2, pp. 556-582.

144. Mardia, K.V., Marshall, R.J., (1984), "Maximum Likelihood Estimation of Models for Residual Covariance in Spatial Regression," *Biometrika*, Vol. 71, 1, pp. 135-146.

145. Lophaven, S.N., Nielsen, H.B., Søndergaard, J., (2002), "DACE: A MATLAB® Kriging Toolbox," Technical Report No. IMM-TR-2002-12, available online at http://www2.imm.dtu.dk/~hbn/dace/.

146. Christensen, O.F., Diggle, P.J., Ribeiro, P.J., (2001), "Analysing Positive-Valued Spatial Data: The Transformed Gaussian Model," *geoENV III – Geostatistics for Environmental Applications* (P.Monestiez et al., eds.), Kluwer Academic Publishers.

147. De Oliveira, V., Kedem, B., Short, D.A., (1997), “Bayesian Prediction of Transformed Gaussian Random Fields,” *J. of the American Statistical Association*, Vol. 92, 440, pp. 1422-1433.

148. Higdon, D., Swall, J., Kern, J., (1999), “Non-Stationary Spatial Modeling,” Bayesian Statistics 6, (J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith, eds.), Oxford University Press, pp. 761-768.

149. Schmidt, A.M., O’Hagan, A., (2003), “Bayesian Inference for Nonstationary Spatial Covariance Structure via Spatial Deformations,” *J. of the Royal Statistical Society B*, Vol. 65, pp. 745-758.

150. Cybenko, G., (1989), “Approximation by Superposition of Sigmoidal Function,” *Mathematics of Control, Signals, and Systems*, Vol. 2, pp. 303-314.

151. Efron, B., (1979), “Bootstrap Methods: Another Look at the Jackknife,” *The Annals of Statistics*, Vol. 7, 1, pp. 1-26.

152. Efron, B., (1981), “Nonparametric Estimates of Standard Error: The Jackknife, the Bootstrap, and Other Methods,” *Biometrika*, Vol. 68, 3, pp. 589-599.

153. Efron, B., Tibshirani, R.J., (1994), An Introduction to the Bootstrap, Monographs on Statistics and Applied Probability, Chapman & Hall/CRC.

154. Zio, E., (2006), “A Study of the Bootstrap Method for Estimating the Accuracy of Artificial Neural Networks in Predicting Nuclear Transient Processes,” *IEEE Transactions on Nuclear Safety*, Vol. 53, 3, pp. 1460-1478.

155. Cadini, F., Zio, E., Kopustinskas, V., Urbonas, R., (2008), “A Model Based on Bootstrapped Neural Networks for Computing the Maximum Fuel Cladding Temperature in an RBMK-1500 Nuclear Reactor Accident,” *Nuclear Engineering & Design*, Vol. 238, 9, pp. 2165-2172.

156. Baxt, W.G., White, H., (1995), “Bootstrapping Confidence Intervals for Clinical Input Variable Effects in a Network Trained to Identify the Presence of Acute Myocardial Infarction,” *Neural Computation*, Vol. 7, 3, pp. 624-638.

157. Pilz, J., Spöck, G., (2008), “Why Do We Need and How Should We Implement Bayesian Kriging Methods,” *Stochastic Environmental Research and Risk Assessment*, Vol. 22, 5, Springer Berlin, doi.10.1007/s00477-007-0165-7.

158. O’Hagan, A., (1994), Kendall’s Advanced Theory of Statistics, Vol. 2B: Bayesian Inference, Wiley, New York.

159. Paulo, R., (2005), “Default Priors for Gaussian Processes,” *The Annals of Statistics*, Vol. 33, 2, pp. 556-582.

160. Agarwal, D.K., Gelfand, A.E., (2005), “Slice Sampling for Simulation Based Fitting of Spatial Data Models,” *Statistics and Computing*, Vol. 15, pp. 61-69.

161. Burgazzi, L., (2003), “Reliability Evaluation of Passive Systems through Functional Reliability Analysis,” *Nuclear Technology*, Vol. 144, pp. 145-151.

162. Hejzlar, P., Pope, M.J., Williams, W.C., Driscoll, M.J., (2005), "Gas Cooled Fast Reactor for Generation IV Service", *Progress in Nuclear Energy*, Vol. 47, 1-4, pp. 271-282.

163. Driscoll, M.J., Hejzlar, P., (2003), “Active or Passive Post-LOCA Cooling of GFRs?,” *Trans. Of the American Nuclear Society*, Vol. **88,** pp. 58.

164. Williams, W.C., Hejzlar, P., Driscoll, M.J., (2004), “Decay Heat Removal from GFR Core by Natural Convection,” *Proc. Int. Congress on Advances in Nuclear Power Plants ICAPP '04, Paper 4166* Pittsburgh, PA, USA.

165. Williams, W.C., Hejzlar, P., Saha, P., (2004), "Analysis of a Convection Loop for GFR Post-LOCA Decay Heat Removal," *Proc. of ICONE12, 12th International Conference on Nuclear Engineering, Paper ICONE12-49360* Arlington, VA, USA.
166. Zio, E., Apostolakis, G.E., (1996), "Two Methods for the Structured Assessment of Model Uncertainty by Experts in Performance Assessments of Radioactive Waste Repositories," *Reliability Engineering & System Safety*, Vol. 54, 2-3, pp. 225-241.
167. US Nuclear Regulatory Commission, (1998), "An Approach for Using Probabilistic Risk Assessment in Risk-Informed Decisions on Plant-Specific Changes to the Licensing Basis," Regulatory Guide 1.174, Washington, DC. Available at www.nrc.gov
168. Nikiforova, A., Hejzlar, P., Todreas, N.E., (2009), "Lead-Cooled Flexible Conversion Ratio Fast Reactor," *Nuclear Engineering & Design*, in press: doi:10.1016/ j.nucengdes.2009.07.013.
169. Todreas, N.E., Hejzlar, P., Nikiforova, A., Petroski, R., Shwageraus, E., Fong, C.J., Driscoll, M.J., Elliott, M.A., Apostolakis, G.E., (2009), "Flexible Conversion Ratio Fast Reactors: Overview," *Nuclear Engineering & Design*, in press: dol:10.1016/ j.nucengdes.2009.07.014.

# Appendix A – Summary Comparison of Sampling-Based Uncertainty Analysis Methods

| Method | Advantages | Drawbacks |
|---|---|---|
| Monte Carlo Simulation (MCS) | • Conceptually straightforward<br>• Estimation error is independent of the dimensionality of the problem<br>• Requires the fewest assumptions compared to alternative methods listed below | • Can be inefficient for estimating small probabilities due to slow convergence rate |
| Latin Hypercube Sampling (LHS) | • Provides more uniform coverage of the input domain (i.e., less clustering about the mode)<br>• Provides more efficient (i.e., requiring fewer samples) estimates of the mean compared to standard MCS | • Estimation error can be difficult to estimate due to correlations induced amongst samples<br>• Numerical studies in the literature suggest LCS is only slightly more efficient than MCS for estimating small probabilities |
| Importance Sampling (IS) | • Can yield very efficient estimates of small probabilities provided sufficient prior information is available to choose an appropriate sampling distribution | • Poor choice of a sampling distribution can lead to worse estimates (i.e., higher estimation error) as compared to standard MCS |
| Line Sampling (LS) | • Extremely efficient if the region of interest (e.g., the failure domain) lies in an isolated region of the input space and the general location of this region is known<br>• Even if the location of the region of interest is unknown, LS can still outperform standard MCS and LHS | • The simplest formulation of LS requires all inputs to be normally distributed<br>• The most general formulation of LS requires at least one input to be independent of all other inputs<br>• Search direction must be specified a priori |
| Subset Simulation (SS) | • Requires minimal prior information from the analyst<br>• More efficient than LS if the region of interest consists of multiple, disconnected domains in the input space | • Less efficient than LS if the location of the region of interest is known<br>• Because SS relies on Markov Chain Monte Carlo, there exists the potential for serial correlation between samples to degrade performance |

# Appendix B – Summary Comparison of Sensitivity Analysis Methods

| Method | Advantages | Drawbacks |
|---|---|---|
| Scatter Plots | • Easy to construct and interpret<br>• Extremely versatile<br>• Provide a visual representation of model input-output (I/O) dependencies<br>• Can easily reveal important and/or anomalous model behavior | • Quantitative measures of sensitivity and/or importance are not provided<br>• Can be difficult to discern the effects of parameters that only weakly affect the model output<br>• Visual inspection of scatter plots can be infeasible if the number of input parameters is very large |
| Monte Carlo Filtering (MCF) | • Useful for identifying threshold effects (i.e., whether a particular parameter influences the propensity of the output to exceed a specified value)<br>• Provides a quantitative measure of I/O dependencies that can be used for parameter ranking | • Unless the number of samples is large, in cases where the I/O dependencies are very weak or the output threshold represents a rare event, identification of important (i.e., influential) parameters is hampered by the low statistical power of the goodness-of-fit tests |
| Parametric Regression Analysis | • Useful for identifying overall trends in the I/O behavior of a model<br>• Standardized regression coefficients provide a convenient means for input parameter ranking | • Functional form of the global regression model must be specified a priori<br>• Special precautions must be taken if the regressors (i.e., the model inputs) are statistically correlated<br>• Overfitting of the regression model to the data can result in incorrect conclusions being drawn from the available data, particularly when few data are available |
| Nonparametric Regression Analysis | • Functional form of the regression model need only be specified on a local scale (no assumptions need to be made regarding the global regression model)<br>• As compared to parametric regression, nonparametric regression allows more complex I/O behavior to be represented | • Excepting the need for prior specification of the global regression model, Nonparametric regression analysis is subject to the same limitations listed above for parametric regression |
| Rank Regression Analysis | • Provides a better ranking of input parameters than linear parametric regression analysis when the model is highly nonlinear but monotonic | • Generally performs quite poorly when model is non-monotonic |

| | | |
|---|---|---|
| Sobol' Sensitivity Indices | • Because parameter ranking is done on a variance basis (i.e., by measuring the contribution of input parameter variances to the model output variance), Sobol' indices may be more useful than regression analysis when the model output is highly sensitive to a particular parameter whose variance is small. Such a parameter might be flagged as important in a regression analysis while not appreciably contributing to the uncertainty/variability in the model output due to the limited variability of the parameter | • Input parameter rankings depend on their assumed probability distribution functions<br>• Assumes that variance is an appropriate measure of uncertainty<br>• Computation of the Sobol' indices is more computationally demanding than regression-based ranking, despite the development of efficient computational techniques, such as the Sobol' MC algorithm and FAST |

# Appendix C – Summary Comparison of Metamodeling Methods

| Method | Advantages | Drawbacks |
|---|---|---|
| Response Surfaces (RS) | • Conceptually straightforward<br>• Can be constructed with minimal computational effort<br>• Extremely efficient and accurate metamodels provided the assumptions regarding the form of the RS are consistent with the underlying model being approximated<br>• Despite their limitations (see right), RS metamodels can be extremely effective when large numbers of data are available | • While standard goodness-of-fit metrics are useful for quantifying the degree of fidelity between the RS and the available data, they are far less useful for assessing the predictive capability of the RS<br>• Overfitting/underfitting can severely degrade the predictive performance of RS metamodels<br>• Because RSs do not, in general, exactly interpolate the available data, many authors have questioned their use for approximating deterministic computer models (see Section IV.2.D beginning on page 58 for a detailed discussion of this and other related criticisms that have been presented in the literature) |
| Gaussian Process (GP) Models | • Requires no prior assumptions on the functional form of the metamodel<br>• GP models can represent complex global I/O behavior that cannot be adequately represented by a polynomial RS<br>• Readily admits a Bayesian interpretation of the metamodel prediction uncertainty<br>• GP models exactly interpolate the available data, increasing their attractiveness for approximating deterministic computer models | • Computational demand increases exponentially with the number of available data, making GP models inefficient if many data are available<br>• Bayesian estimation of covariance function parameters can present a large computational burden<br>• Predictive performance can be sensitive to various assumptions on the correlation model<br>• Ill-conditioning of the covariance matrix can introduce large numerical errors into the analysis<br>• Normality assumption may not be applicable in some cases. Although methods exist to circumvent this issue, they result in an increased complexity |
| Artificial Neural Networks (ANN) | • Highly flexible, allowing for modeling of highly complex I/O behaviors<br>• Inherently well-suited for cases when multiple model outputs are of interest | • ANNs are trained to match a given data set through a complex, nonlinear optimization process that can be computationally demanding<br>• Special precautions must be taken during the training procedure to prevent overfitting<br>• Because of their black-box nature, ANNs are less readily interpretable than alternative metamodeling methods |

# Appendix D - Sensitivity Analysis Data from GFR Case Study

**Table D.1. Bootstrap Bias-Corrected (BBC) estimates and BBC 95% Confidence Intervals (CIs) for the first-order Sobol sensitivity indices $S_i^1$ of parameters $x_i$, i = 1, 2, ..., 9, obtained by bootstrapped ANNs build on $N_D$ = 20, 30, 50, 70, 100 training samples**

| Output considered: hot channel coolant outlet temperature, $T^{hot}_{out,core}$ | | | |
|---|---|---|---|
| Artificial Neural Network (ANN) | | | |
| Training sample size, $N_D = 20$ | | | |
| Variable, $x_i$ | Index, $S_i^1$ | BBC estimate | BBC 95% CI |
| Power, $x_1$ | $5.95\cdot10^{-3}$ | 0.0217 | $[2.558\cdot10^{-4}, 0.0376]$ |
| Pressure, $x_2$ | 0.8105 | 0.8178 | [0.6327, 0.9873] |
| Cooler wall T, $x_3$ | 0.0303 | 0.0506 | $[5.528\cdot10^{-3}, 0.1173]$ |
| Nusselt forced, $x_4$ | $4.214\cdot10^{-5}$ | $2.262\cdot10^{-3}$ | $[0,3.4\cdot10^{-2}]$ |
| Nusselt mixed, $x_5$ | 0.0583 | 0.0535 | $[1.595\cdot10^{-4}, 0.1133]$ |
| Nusselt free, $x_6$ | $5.211\cdot10^{-4}$ | $3.620\cdot10^{-3}$ | [0, 0.0591] |
| Friction forced, $x_7$ | $1.533\cdot10^{-5}$ | $6.059\cdot10^{-3}$ | $[0, 2.645\cdot10^{-2}]$ |
| Friction mixed, $x_8$ | 0.0594 | 0.0612 | $[7.184\cdot10^{-3}, 0.1542]$ |
| Friction free, $x_9$ | $2.139\cdot10^{-4}$ | 0.0101 | [0, 0.0799] |

| Training sample size, $N_D = 30$ | | | |
|---|---|---|---|
| Variable, $x_i$ | Index, $S_i^1$ | BBC estimate | BBC 95% CI |
| Power, $x_1$ | $5.95\cdot10^{-3}$ | 0.0134 | $[9.283\cdot10^{-4}, 0.0368]$ |
| Pressure, $x_2$ | 0.8105 | 0.8130 | [0.7313, 0.9447] |
| Cooler wall T, $x_3$ | 0.0303 | 0.0373 | [0.0100, 0.1033] |
| Nusselt forced, $x_4$ | $4.214\cdot10^{-5}$ | $5.077\cdot10^{-5}$ | $[0, 2.781\cdot10^{-3}]$ |
| Nusselt mixed, $x_5$ | 0.0583 | 0.0398 | $[5.051\cdot10^{-3}, 0.0919]$ |
| Nusselt free, $x_6$ | $5.211\cdot10^{-4}$ | $5.897\cdot10^{-4}$ | $[0, 8.711\cdot10^{-3}]$ |
| Friction forced, $x_7$ | $1.533\cdot10^{-5}$ | $2.033\cdot10^{-4}$ | $[0, 1.014\cdot10^{-3}]$ |
| Friction mixed, $x_8$ | 0.0594 | 0.0708 | [0.0328, 0.1069] |
| Friction free, $x_9$ | $2.139\cdot10^{-4}$ | 0.0118 | [0, 0.0260] |

| Training sample size, $N_D = 50$ | | | |
|---|---|---|---|
| Variable, $x_i$ | Index, $S_i^1$ | BBC estimate | BBC 95% CI |
| Power, $x_1$ | $5.95\cdot10^{-3}$ | $6.043\cdot10^{-3}$ | $[2.732\cdot10^{-4}, 0.0185]$ |
| Pressure, $x_2$ | 0.8105 | 0.8185 | [0.7846, 0.8541] |
| Cooler wall T, $x_3$ | 0.0303 | 0.0302 | [0.0259, 0.0613] |
| Nusselt forced, $x_4$ | $4.214\cdot10^{-5}$ | $1.573\cdot10^{-4}$ | $[0, 5.485\cdot10^{-4}]$ |
| Nusselt mixed, $x_5$ | 0.0583 | 0.0579 | [0.0222, 0.0801] |
| Nusselt free, $x_6$ | $5.211\cdot10^{-4}$ | $2.330\cdot10^{-4}$ | $[0, 4.560\cdot10^{-3}]$ |
| Friction forced, $x_7$ | $1.533\cdot10^{-5}$ | $1.228\cdot10^{-4}$ | $[0, 4.217\cdot10^{-4}]$ |
| Friction mixed, $x_8$ | 0.0594 | 0.0591 | [0.0373, 0.0824] |
| Friction free, $x_9$ | $2.139\cdot10^{-4}$ | $2.259\cdot10^{-4}$ | $[0, 1.034\cdot10^{-3}]$ |

| Training sample size, $N_D = 70$ | | | |
|---|---|---|---|
| Variable, $x_i$ | Index, $S_i^1$ | BBC estimate | BBC 95% CI |
| Power, $x_1$ | $5.95\cdot10^{-3}$ | $6.345\cdot10^{-3}$ | $[3.614\cdot10^{-4}, 9.084\cdot10^{-3}]$ |
| Pressure, $x_2$ | 0.8105 | 0.8015 | [0.7856, 0.8420] |
| Cooler wall T, $x_3$ | 0.0303 | 0.0428 | [0.0329, 0.0588] |
| Nusselt forced, $x_4$ | $4.214\cdot10^{-5}$ | $8.523\cdot10^{-5}$ | $[0, 2.302\cdot10^{-4}]$ |
| Nusselt mixed, $x_5$ | 0.0583 | 0.0561 | [0.0431, 0.0731] |

| | | | |
|---|---|---|---|
| Nusselt free, $x_6$ | $5.211\cdot10^{-4}$ | $3.307\cdot10^{-4}$ | $[0, 1.204\cdot10^{-3}]$ |
| Friction forced, $x_7$ | $1.533\cdot10^{-5}$ | $5.449\cdot10^{-5}$ | $[0, 2.522\cdot10^{-4}]$ |
| Friction mixed, $x_8$ | 0.0594 | 0.0611 | [0.0427, 0.0774] |
| Friction free, $x_9$ | $2.139\cdot10^{-4}$ | $1.202\cdot10^{-4}$ | $[0, 6.149\cdot10^{-4}]$ |

| **Training sample size, $N_D$ = 100** | | | |
|---|---|---|---|
| **Variable, $x_i$** | **Index, $S_i^I$** | **BBC estimate** | **BBC 95% CI** |
| Power, $x_1$ | $5.95\cdot10^{-3}$ | $5.199\cdot10^{-3}$ | $[4.137\cdot10^{-4}, 8.563\cdot10^{-3}]$ |
| Pressure, $x_2$ | 0.8105 | 0.8098 | [0.7949, 0.8324] |
| Cooler wall T, $x_3$ | 0.0303 | 0.0368 | [0.0352, 0.04791] |
| Nusselt forced, $x_4$ | $4.214\cdot10^{-5}$ | $6.430\cdot10^{-5}$ | $[0, 9.523\cdot10^{-5}]$ |
| Nusselt mixed, $x_5$ | 0.0583 | 0.0591 | [0.0491, 0.0649] |
| Nusselt free, $x_6$ | $5.211\cdot10^{-4}$ | $6.338\cdot10^{-4}$ | $[0, 8.413\cdot10^{-4}]$ |
| Friction forced, $x_7$ | $1.533\cdot10^{-5}$ | $1.634\cdot10^{-5}$ | $[0, 4.393\cdot10^{-5}]$ |
| Friction mixed, $x_8$ | 0.0594 | 0.0605 | [0.0536, 0.0711] |
| Friction free, $x_9$ | $2.139\cdot10^{-4}$ | $1.676\cdot10^{-4}$ | $[0, 3.231\cdot10^{-4}]$ |

**Table D.2. Bootstrap Bias-Corrected (BBC) estimates and BBC 95% Confidence Intervals (CIs) for the first-order Sobol sensitivity indices $S_i^1$ of parameters $x_i$, i = 1, 2, ..., 9, obtained by bootstrapped RSs build on $N_D$ = 20, 30, 50, 70, 100 training samples**

**Output considered: hot channel coolant outlet temperature, $T^{hot}_{out,core}$**

**Response Surface (RS)**

**Training sample size, $N_D = 20$**

| Variable, $x_i$ | Index, $S_i^1$ | BBC estimate | BBC 95% CI |
|---|---|---|---|
| Power, $x_1$ | $5.95\cdot10^{-3}$ | 0.0103 | [0, 0.0561] |
| Pressure, $x_2$ | 0.8105 | 0.7550 | [0.6300, 1] |
| Cooler wall T, $x_3$ | 0.0303 | 0.1600 | [0, 0.3091] |
| Nusselt forced, $x_4$ | $4.214\cdot10^{-5}$ | 0.0344 | [0, 0.0702] |
| Nusselt mixed, $x_5$ | 0.0583 | 0.0389 | [0, 0.0957] |
| Nusselt free, $x_6$ | $5.211\cdot10^{-4}$ | $9.684\cdot10^{-3}$ | [0, 0.0172] |
| Friction forced, $x_7$ | $1.533\cdot10^{-5}$ | $6.380\cdot10^{-3}$ | [0, 0.0155] |
| Friction mixed, $x_8$ | 0.0594 | 0.0886 | [0, 0.1734] |
| Friction free, $x_9$ | $2.139\cdot10^{-4}$ | $9.808\cdot10^{-3}$ | [0, 0.0296] |

**Training sample size, $N_D = 30$**

| Variable, $x_i$ | Index, $S_i^1$ | BBC estimate | BBC 95% CI |
|---|---|---|---|
| Power, $x_1$ | $5.95\cdot10^{-3}$ | 0.0257 | [0, 0.0306] |
| Pressure, $x_2$ | 0.8105 | 0.7932 | [0.7049, 1] |
| Cooler wall T, $x_3$ | 0.0303 | 0.0148 | [0, 0.0241] |
| Nusselt forced, $x_4$ | $4.214\cdot10^{-5}$ | $1.506\cdot10^{-3}$ | [0, $2.350\cdot10^{-3}$] |
| Nusselt mixed, $x_5$ | 0.0583 | 0.0493 | [0, 0.0884] |
| Nusselt free, $x_6$ | $5.211\cdot10^{-4}$ | $7.436\cdot10^{-3}$ | [0, 0.0141] |
| Friction forced, $x_7$ | $1.533\cdot10^{-5}$ | $8.526\cdot10^{-3}$ | [0, $9.871\cdot10^{-3}$] |
| Friction mixed, $x_8$ | 0.0594 | 0.0832 | [0, 0.1579] |
| Friction free, $x_9$ | $2.139\cdot10^{-4}$ | 0.0159 | [0, 0.0176] |

**Training sample size, $N_D = 50$**

| Variable, $x_i$ | Index, $S_i^1$ | BBC estimate | BBC 95% CI |
|---|---|---|---|
| Power, $x_1$ | $5.95\cdot10^{-3}$ | $3.793\cdot10^{-3}$ | [0, 0.0263] |
| Pressure, $x_2$ | 0.8105 | 0.7730 | [0.7219, 0.9218] |
| Cooler wall T, $x_3$ | 0.0303 | 0.0395 | [0, 0.0666] |
| Nusselt forced, $x_4$ | $4.214\cdot10^{-5}$ | $2.295\cdot10^{-3}$ | [0, $4.930\cdot10^{-3}$] |
| Nusselt mixed, $x_5$ | 0.0583 | 0.0541 | [0.0170, 0.0936] |
| Nusselt free, $x_6$ | $5.211\cdot10^{-4}$ | $2.011\cdot10^{-3}$ | [0, $3.910\cdot10^{-3}$] |
| Friction forced, $x_7$ | $1.533\cdot10^{-5}$ | $9.258\cdot10^{-5}$ | [0, $7.041\cdot10^{-3}$] |
| Friction mixed, $x_8$ | 0.0594 | 0.0808 | [0.0153, 0.1325] |
| Friction free, $x_9$ | $2.139\cdot10^{-4}$ | $3.620\cdot10^{-5}$ | [0, $9.231\cdot10^{-3}$] |

**Training sample size, $N_D = 70$**

| Variable, $x_i$ | Index, $S_i^1$ | BBC estimate | BBC 95% CI |
|---|---|---|---|
| Power, $x_1$ | $5.95\cdot10^{-3}$ | $6.300\cdot10^{-3}$ | [0, 0.0196] |
| Pressure, $x_2$ | 0.8105 | 0.8017 | [0.7506, 0.8773] |
| Cooler wall T, $x_3$ | 0.0303 | 0.0464 | [0.0139, 0.0733] |
| Nusselt forced, $x_4$ | $4.214\cdot10^{-5}$ | $2.082\cdot10^{-4}$ | [0, $6.411\cdot10^{-4}$] |
| Nusselt mixed, $x_5$ | 0.0583 | 0.0613 | [0.0287, 0.0881] |
| Nusselt free, $x_6$ | $5.211\cdot10^{-4}$ | $1.369\cdot10^{-4}$ | [0, $1.890\cdot10^{-3}$] |
| Friction forced, $x_7$ | $1.533\cdot10^{-5}$ | $3.826\cdot10^{-3}$ | [0, $7.920\cdot10^{-3}$] |
| Friction mixed, $x_8$ | 0.0594 | 0.0750 | [0.0275, 0.1105] |

| | | | |
|---|---|---|---|
| Friction free, $x_9$ | $2.139\cdot10^{-4}$ | $6.673\cdot10^{-4}$ | $[0, 1.410\cdot10^{-3}]$ |

| **Training sample size, $N_D$= 100** | | | |
|---|---|---|---|
| **Variable, $x_i$** | **Index, $S_i'$** | **BBC estimate** | **BBC 95% CI** |
| Power, $x_1$ | $5.95\cdot10^{-3}$ | $5.523\cdot10^{-3}$ | [0, 0.0118] |
| Pressure, $x_2$ | 0.8105 | 0.8219 | [0.7769, 0.8516] |
| Cooler wall T, $x_3$ | 0.0303 | 0.0240 | [0.0189, 0.0356] |
| Nusselt forced, $x_4$ | $4.214\cdot10^{-5}$ | $3.700\cdot10^{-4}$ | $[0, 5.990\cdot10^{-4}]$ |
| Nusselt mixed, $x_5$ | 0.0583 | 0.0665 | [0.0366, 0.0727] |
| Nusselt free, $x_6$ | $5.211\cdot10^{-4}$ | $6.790\cdot10^{-4}$ | $[0, 1.061\cdot10^{-3}]$ |
| Friction forced, $x_7$ | $1.533\cdot10^{-5}$ | $4.030\cdot10^{-3}$ | $[0, 4.645\cdot10^{-3}]$ |
| Friction mixed, $x_8$ | 0.0594 | 0.0715 | [0.0435, 0.0961] |
| Friction free, $x_9$ | $2.139\cdot10^{-4}$ | $2.153\cdot10^{-5}$ | $[0, 9.453\cdot10^{-4}]$ |

# Appendix E – Experimental Design Data from FCRR Case Study

Table E.1. Original 27-point experimental design from Fong et al [31]

| Run | Inputs (Physical Values) | | | | | Inputs (Coded Values) | | | | | Relap Output |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | X1 | X2 | X3 | X4 | X5 | X1 | X2 | X3 | X4 | X5 | PCT (°C) |
| 1 | 0.000 | 6.85 | 0.85 | 0.000 | 26.85 | -2 | -2 | +2 | -2 | 0 | 705.400 |
| 2 | 0.150 | 46.85 | 0.75 | 0.075 | 46.85 | +2 | +2 | 0 | 0 | +2 | 739.900 |
| 3 | 0.075 | 46.85 | 0.65 | 0.150 | 46.85 | 0 | +2 | -2 | +2 | +2 | 753.700 |
| 4 | 0.150 | 26.85 | 0.85 | 0.075 | 26.85 | +2 | 0 | +2 | 0 | 0 | 707.100 |
| 5 | 0.000 | 46.85 | 0.75 | 0.150 | 26.85 | -2 | +2 | 0 | +2 | 0 | 723.300 |
| 6 | 0.150 | 6.85 | 0.65 | 0.000 | 46.85 | +2 | -2 | -2 | -2 | +2 | 746.200 |
| 7 | 0.000 | 26.85 | 0.85 | 0.000 | 6.85 | -2 | 0 | +2 | -2 | -2 | 688.900 |
| 8 | 0.075 | 26.85 | 0.75 | 0.150 | 6.85 | 0 | 0 | 0 | +2 | -2 | 699.400 |
| 9 | 0.075 | 6.85 | 0.65 | 0.075 | 6.85 | 0 | -2 | -2 | 0 | -2 | 706.700 |
| 10 | 0.075 | 46.85 | 0.85 | 0.075 | 26.85 | 0 | +2 | +2 | 0 | 0 | 704.600 |
| 11 | 0.075 | 6.85 | 0.75 | 0.000 | 46.85 | 0 | -2 | 0 | -2 | +2 | 729.400 |
| 12 | 0.000 | 46.85 | 0.65 | 0.075 | 6.85 | -2 | +2 | -2 | 0 | -2 | 709.100 |
| 13 | 0.150 | 46.85 | 0.85 | 0.150 | 6.85 | +2 | +2 | +2 | +2 | -2 | 684.600 |
| 14 | 0.075 | 26.85 | 0.75 | 0.000 | 26.85 | 0 | 0 | 0 | -2 | 0 | 711.300 |
| 15 | 0.150 | 26.85 | 0.65 | 0.000 | 6.85 | +2 | 0 | -2 | -2 | -2 | 700.900 |
| 16 | 0.000 | 6.85 | 0.85 | 0.075 | 46.85 | -2 | -2 | +2 | 0 | +2 | 718.100 |
| 17 | 0.150 | 6.85 | 0.75 | 0.150 | 26.85 | +2 | -2 | 0 | +2 | 0 | 708.400 |
| 18 | 0.000 | 26.85 | 0.65 | 0.150 | 46.85 | -2 | 0 | -2 | +2 | +2 | 750.500 |
| 19 | 0.075 | 46.85 | 0.85 | 0.000 | 46.85 | 0 | +2 | +2 | -2 | +2 | 719.700 |
| 20 | 0.150 | 6.85 | 0.75 | 0.075 | 6.85 | +2 | -2 | 0 | 0 | -2 | 704.400 |
| 21 | 0.000 | 6.85 | 0.65 | 0.150 | 26.85 | -2 | -2 | -2 | +2 | 0 | 714.600 |
| 22 | 0.150 | 26.85 | 0.85 | 0.150 | 46.85 | +2 | 0 | +2 | +2 | +2 | 711.200 |
| 23 | 0.000 | 26.85 | 0.75 | 0.075 | 46.85 | -2 | 0 | 0 | 0 | +2 | 728.700 |
| 24 | 0.150 | 46.85 | 0.65 | 0.000 | 26.85 | +2 | +2 | -2 | -2 | 0 | 717.500 |
| 25 | 0.075 | 6.85 | 0.85 | 0.150 | 6.85 | 0 | -2 | +2 | +2 | -2 | 671.400 |
| 26 | 0.000 | 46.85 | 0.75 | 0.000 | 6.85 | -2 | +2 | 0 | -2 | -2 | 687.000 |
| 27 | 0.075 | 26.85 | 0.65 | 0.075 | 26.85 | 0 | 0 | -2 | 0 | 0 | 715.600 |

Table E.2. Experimental design for 8-point validation set from Fong et al [31]

| Run | Inputs (Physical Variables) | | | | | Inputs (Coded Variables) | | | | | Relap Output |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | X1 | X2 | X3 | X4 | X5 | X1 | X2 | X3 | X4 | X5 | PCT (°C) |
| 28 | 0.150 | 46.85 | 0.65 | 0.000 | 36.85 | +2 | +2 | -2 | -2 | +1 | 741.66 |
| 29 | 0.075 | 26.85 | 0.65 | 0.150 | 36.85 | 0 | 0 | -2 | +2 | +1 | 741.20 |
| 30 | 0.075 | 6.85 | 0.65 | 0.075 | 36.85 | 0 | 0 | -2 | 0 | +1 | 740.37 |
| 31 | 0.075 | 6.85 | 0.75 | 0.150 | 36.85 | 0 | -2 | 0 | 2 | +1 | 726.48 |
| 32 | 0.000 | 6.85 | 0.75 | 0.075 | 36.85 | -2 | -2 | 0 | 0 | +1 | 726.30 |
| 33 | 0.150 | 46.85 | 0.85 | 0.000 | 36.85 | +2 | +2 | +2 | -2 | +1 | 718.86 |
| 34 | 0.075 | 6.85 | 0.75 | 0.075 | 36.85 | 0 | 0 | 0 | 0 | +1 | 717.58 |
| 35 | 0.000 | 6.85 | 0.85 | 0.000 | 36.85 | -2 | -2 | +2 | -2 | +1 | 714.74 |

**Table E.3. Experimental design for remaining 27-points used in FCRR study**

| Run | Inputs (Physical Variables) | | | | | Inputs (Coded Variables) | | | | | Relap Output |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | X1 | X2 | X3 | X4 | X5 | X1 | X2 | X3 | X4 | X5 | PCT (°C) |
| 36 | 0.000 | 6.85 | 0.85 | 0.000 | 16.85 | -2 | -2 | +2 | -2 | -1 | 695.61 |
| 37 | 0.000 | 6.85 | 0.75 | 0.075 | 16.85 | -2 | -2 | 0 | 0 | -1 | 706.64 |
| 38 | 0.113 | 46.85 | 0.65 | 0.113 | 46.85 | +1 | +2 | -2 | +1 | +2 | 753.05 |
| 39 | 0.150 | 36.85 | 0.65 | 0.150 | 46.85 | +2 | +1 | -2 | +2 | +2 | 752.33 |
| 40 | 0.150 | 46.85 | 0.70 | 0.150 | 46.85 | +2 | +2 | -1 | +2 | +2 | 746.78 |
| 41 | 0.113 | 36.85 | 0.70 | 0.113 | 46.85 | +1 | +1 | -1 | +1 | +2 | 745.72 |
| 42 | 0.113 | 46.85 | 0.65 | 0.150 | 36.85 | +1 | +2 | -2 | +2 | +1 | 742.70 |
| 43 | 0.150 | 36.85 | 0.65 | 0.113 | 36.85 | +2 | +1 | -2 | +1 | +1 | 742.33 |
| 44 | 0.150 | 46.85 | 0.70 | 0.113 | 36.85 | +2 | +2 | -1 | +1 | +1 | 737.44 |
| 45 | 0.113 | 36.85 | 0.70 | 0.150 | 36.85 | +1 | +1 | -1 | +2 | +1 | 735.45 |
| 46 | 0.075 | 6.850 | 0.800 | 0.150 | 46.850 | 0 | -2 | +1 | +2 | +2 | 730.42 |
| 47 | 0.000 | 26.850 | 0.800 | 0.000 | 46.850 | -2 | 0 | +1 | -2 | +2 | 733.03 |
| 48 | 0.150 | 46.850 | 0.800 | 0.075 | 46.850 | +2 | +2 | +1 | 0 | +2 | 734.03 |
| 49 | 0.150 | 6.850 | 0.800 | 0.075 | 36.850 | +2 | -2 | +1 | 0 | +1 | 719.76 |
| 50 | 0.000 | 26.850 | 0.800 | 0.150 | 36.850 | -2 | 0 | +1 | +2 | +1 | 724.78 |
| 51 | 0.075 | 46.850 | 0.800 | 0.000 | 36.850 | 0 | +2 | +1 | -2 | +1 | 725.03 |
| 52 | 0.075 | 6.850 | 0.800 | 0.075 | 26.850 | 0 | -2 | +1 | 0 | 0 | 710.62 |
| 53 | 0.150 | 26.850 | 0.800 | 0.000 | 26.850 | +2 | 0 | +1 | -2 | 0 | 710.79 |
| 54 | 0.000 | 46.850 | 0.800 | 0.750 | 26.850 | -2 | +2 | +1 | +2 | 0 | 717.02 |
| 55 | 0.150 | 46.850 | 0.850 | 0.150 | 46.850 | +2 | +2 | +2 | +2 | +2 | 729.740 |
| 56 | 0.150 | 6.850 | 0.650 | 0.150 | 6.850 | +2 | -2 | -2 | +2 | -2 | 707.050 |
| 57 | 0.000 | 46.850 | 0.650 | 0.150 | 46.850 | -2 | +2 | -2 | +2 | +2 | 754.730 |
| 58 | 0.000 | 6.850 | 0.850 | 0.150 | 6.850 | -2 | -2 | +2 | +2 | -2 | 687.240 |
| 59 | 0.150 | 46.850 | 0.850 | 0.000 | 6.850 | +2 | +2 | +2 | -2 | -2 | 689.660 |
| 60 | 0.000 | 46.850 | 0.650 | 0.000 | 6.850 | -2 | +2 | -2 | -2 | -2 | 712.460 |
| 61 | 0.000 | 6.850 | 0.850 | 0.000 | 46.850 | -2 | -2 | +2 | -2 | +2 | 724.200 |
| 62 | 0.075 | 26.850 | 0.750 | 0.075 | 26.850 | 0 | 0 | 0 | 0 | 0 | 718.810 |