Analysis of Reactor Simulations Using Surrogate Models

by

Artem Yankov

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Nuclear Engineering and Radiological Sciences)
in the University of Michigan
2015

Doctoral Committee:

Professor Thomas J. Downar, Chair
Reactor Physics Nuclear Engineer Benjamin S. Collins, ORNL
Professor Karthik Duraisamy
Professor Krzysztof J. Fidkowski
Professor John C. Lee

ACKNOWLEDGMENTS

Foremost, I am foreever indebted to Professor Thomas Downar for taking a risk on an unknown mathematics major coming from a small engineering school in Terre Haute, Indiana. Needless to say, without Prof. Downar's willingness to take me in it is questionable whether I would have ever been given the opportunity to study computation and nuclear engineering with some of the best around. Also, I must thank Dr. Benjamin Collins for providing guidance and mentorship, especially in the first few years of the doctoral program when it is needed most. I am thankful for his patience and for instilling in me confidence in being able to conduct original research.

The sacrifices and risks made by my family in leaving all they knew in Russia for the chance of a better life in the United States has provided me with inspiration every day. My family has taught me how to be gritty and showed me how far a little elbow grease can go. I owe everything to my family for giving me the personality traits necessary to complete a Ph.D.

My wife Kaela deserves an enormous amount of credit for the completion of this dissertation. Ever since our first date as undergraduates my study and organizational habits have miraculously improved. Thank you for putting me in place and supporting all my endeavors.

I am thankful to all my committee members, Dr. Giovanni Pastore and Dr. Laura Swiler. Discussions with these professionals helped to cement my understanding of concepts that would ultimately be the cornerstone of my thesis. I appreciate their patience in answering all my questions.

Last but not least, my gratitude is given to anyone who has had deskspace in the "Boiler Room" during my time there. It's highly likely that at one point we took a trip to the Cooley Building to get coffee, a trip that has been invaluable in clearing my mind and putting things in perspective. Thank you for maintaining a light-hearted atmosphere at all times.

TABLE OF CONTENTS

Chapter

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

LIST OF ABBREVIATIONS

PDE partial differential equation

SPDE stochastic partial differential equation

HDMR high-dimensional model representation

ANOVA analysis of variance

TMI Three Mile Island

UAM Uncertainty Analysis in Modeling

PARCS Purdue Advanced Reactor Core Simulator

SCALE Standardized Computer Analyses for Licensing Evaluation

ENDF Evaluated Nuclear Data File

ADFs assembly discontinuity factors

LHS Latin Hypercube Sampling

MPACT Michigan Parallel Characteristics Transport Code

SIFGRS Simple Integrated Fission Gas Release and Swelling

PCA Principal Component Analysis

RMSE Root Mean Square Error

EGO Efficient Global Optimization

COBYLA Constrained Optimization by Linear Approximation

ABSTRACT

The relatively recent abundance of computing resources has driven computational scientists to build more complex and approximation-free computer models of physical phenomenon. Often times, multiple high fidelity computer codes are coupled together in hope of improving the predictive powers of simulations with respect to experimental data. To improve the predictive capacity of computer codes experimental data should be folded back into the parameters processed by the codes through optimization and calibration algorithms. However, application of such algorithms may be prohibitive since they generally require thousands of evaluations of computationally expensive, coupled, multiphysics codes. Surrogates models for expensive computer codes have shown promise towards making optimization and calibration feasible.

In this thesis, non-intrusive surrogate building techniques are investigated for their applicability in nuclear engineering applications. Specifically, Kriging and the coupling of the anchored-ANOVA decomposition with collocation are utilized as surrogate building approaches. Initially, these approaches are applied and naively tested on simple reactor applications with analytic solutions. Ultimately, Kriging is applied to construct a surrogate to analyze fission gas release during the Risø AN3 power ramp experiment using the fuel performance modeling code Bison. To this end, Kriging is extended from building surrogates for scalar quantities to entire time series using principal component analysis. A surrogate model is built for fission gas kinetics time series and the true values of relevant parameters are inferred by folding experimental data with the surrogate. Sensitivity analysis is also performed on the fission gas release parameters to gain insight into the underlying physics.

CHAPTER 1

Introduction

1.1 Overview of Previous Work

Modeling nuclear reactor stability and performance computationally has evolved into a multi-physics and multi-scale regime. Various computer codes have been developed and optimized to model individual facets of reactor operation such as neutronics, thermal-hydraulics, and kinetics. These codes are most often coupled to produce more physical results. While it is crucial to be able to produce best-estimate calculations for the design and safety analysis of nuclear reactors, it is equally important to obtain design margins by propagating uncertainty information through the entire computational process. Replacing seemingly arbitrary design margins based on engineering judgment with design margins based on the uncertainty quantification of computer codes can help overcome significant human shortcomings [18].

The methodologies used for uncertainty quantification in nuclear engineering have generally mirrored the computational resources available during the time of development. When computational resources were relatively limited, methods based on perturbation theory were derived since the method computation times are independent of the number of input parameters [76]. Perturbation theory applies a number of linear approximations and adjoint operators to allow for the retrieval of sensitivity coefficients and basic statistical moments of responses of interest. However, in perturbation theory each desired response requires the solution of a deterministic equation and in some inhomogeneous problems in reactor physics the linear approximations may fail [21]. In addition, application of perturbation theory techniques to computer codes is highly intrusive. Coupled, multiphysics uncertainty analyses using perturbation theory are prohibitive especially when legacy codes are involved. Yet another drawback of perturbation theory in neutronics applications arises in the adjoint-based formulation for cross section uncertainty propagation. In this case, exceptions must be made to responses that cannot be expressed as ratios of reaction rates, as in the transport cross section and assembly discontinuity factors (ADFs) [84] [83].

As computer resources increased, the linear approximations of perturbation theory were gradually relaxed until routine Monte Carlo sampling became feasible. In the XSUSA method a 44-group covariance matrix is stochastically sampled to generate perturbed few-group cross sections that can be used as input to a Monte Carlo uncertainty analysis in a core simulator [43]. While uncertainty quantification using Monte Carlo sampling does not apply any approximations to the mathematical system under investigation, the computational expense is tremendous, and the extraction of information such as sensitivity coefficients can be difficult. Nevertheless, sampling provides a brute force means by which many uncertainty quantification algorithms can be validated.

Recently, an approach based on the stochastic finite element method, referred to as polynomial chaos expansion, has been developed for the purposes of uncertainty quantification [24]. The basic idea behind polynomial chaos expansions is to expand the random variables of some governing stochastic partial differential equations in terms of orthogonal polynomials and then to apply Galerkin projections [51]. This idea is similar to the angular expansion of the scatter cross section in terms of Legendre polynomials in neutron transport theory. Methods for uncertainty quantification that employ polynomial chaos expansions fall under the general mathematical framework of spectral methods. The motivation behind spectral methods, and specifically collocation, for uncertainty quantification is to combine the rapid convergence rates of deterministic methods with the generality of stochastic sampling methods.

While used extensively in structural and materials engineering, polynomial chaos expansions have been seldom applied in nuclear engineering. In [80] and [79] the author applies analytic polynomial chaos techniques to investigate the effect of random material properties on radiation transport through a slab in the P1 approximation. The authors in [5] apply non-linear polynomial chaos to examine static and dynamic eigenvalue uncertainties due to uncertainties in cross section values. Finally, the authors in [19] use polynomial chaos-collocation to study the effects of total cross section uncertainties on the probability density function of the scalar flux in absorbing and diffusive media.

The polynomial chaos expansions based on stochastic Galerkin methods applied in the research above exhibit several substantial advantages over Monte Carlo and perturbative methods. Mainly, spectral convergence rates can be achieved. In addition, while nonlinearities and large uncertainties in random variables pose problems for perturbation methods, the stochastic Galerkin methods easily deal with such factors. However, the primary drawback of polynomial chaos expansions with stochastic Galerkin methods lies in their intrusive nature. In other words, unless an engineering code is initially developed with stochastic Galerkin capabilities, which is unlikely, extensive modifications will have to be made to the code. Modifications include adding the ability to solve an enlarged system of coupled, partial differential equations.

To this end the stochastic collocation method, a variant of polynomial chaos expansion, is considered since it circumvents the need to modify engineering code in order to implement spectrally converging uncertainty quantification methods. Stochastic collocation methods are essentially interpolatory, projecting a set of deterministic simulations onto a polynomial basis. With stochastic collocation codes can be treated as ”black boxes”. Such an attribute is highly desirable for the uncertainty analysis of nuclear engineering computer codes since often these consist of coupled, multi-physics, and multi-scale legacy code. The success of perturbatory methods, which are linear, in the uncertainty analysis of nuclear engineering codes is indicative of the potential for stochastic collocation. Being a spectral method, stochastic collocation performs best on smooth functions [12].

However, even when utilizing a stochastic collocation method based on sparse grid constructs, the method suffers from the so-called ”curse of dimensionality” [51]. The ”curse of dimensionality” arises when the convergence of a method is proportional to the exponent of the the function's dimensionality, as in a full tensor product interpolation scheme. While sparse grids help to mitigate the ”curse of dimensionality” researchers have shown that when applied to realistic engineering computer simulations the practicality of stochastic collocation diminishes for dimensions greater than $\mathscr{O}(10)$ [50]. Consequently, further mitigation is needed in order to apply stochastic collocation to real-world engineering problems. To this end, reduced order models, also known as surrogate models, are utilized. Reduced order models work to find a relatively small subspace of some function that is still representative of the original function space.

Performing uncertainty quantification on high-dimensional, coupled, multiphysics engineering codes can be prohibitive if the codes take many hours to run. To make uncertainty quantification feasible for such problems reduced order models are constructed. While a reduced order model may be computationally expensive to build, the ideal end product is a function that can be rapidly evaluated while simultaneously preserving the predictive capabilities of the large-scale model [13]. Due to the rapid evaluation of the reduced order model, uncertainty quantification, optimization studies, probabilistic analysis, and inverse problems can be carried out.

The popularity of collocation-based methods for uncertainty quantification have stemmed from greater accessibility to computing clusters. As recently as 2006 the theme in uncertainty quantification had been how to get the most out of a limited number of available simulations [27]. In [27] researchers investigated whether it is more accurate to estimate statistics directly from *N* unstructured simulations of a large-scale engineering code or from a trend model based on the *N* simulations. Krigging interpolation and multivariate adaptive regression splines are two common approaches for building trend models [69]. Collocation methods differ from such response surface approximation methods mainly in their use of structured grids at which the large-scale codes are evaluated. Use of structured grids generally allow for faster convergence to various error thresholds. However, to achieve the expected convergence of collocation algorithms the number of times a large-scale code must be executed is dictated by the algorithm and not the user.

1.2 Stochastic Partial Differential Equations

A stochastic partial differential equation (SPDE) is similar in flavor to the better known partial differential equation (PDE), the main difference being the presence of input uncertainties in the former's parameter space. If the parameters in some PDE are probabilistic and the affects of those variable parameters on the outputs are of interest, then the PDE must be reformulated into a SPDE. The following example aims to elucidate the difference between PDEs and SPDEs and to motivate a discussion of the unique features characteristic to SPDEs. While these features form the mathematical back-bone for this thesis they are somewhat abstract and not crucial to developing an understanding of the subject matter.

Consider the PDE describing one-speed diffusion of neutron flux in a slab nuclear reactor [17]:

$$\begin{aligned} \frac{1}{v}\frac{\partial\phi}{\partial t} - D\frac{\partial^2\phi}{\partial x^2} + \Sigma_a\phi(x,t) &= \nu\Sigma_f\phi(x,t) \\ \phi(x,0) &= \phi_0(x) \\ \phi(a,t) = \phi(-a,t) &= 0 \end{aligned} \tag{1.1}$$

The PDE in 1.1 can be solved for the flux ϕ as a function of both space x and time t. However, such a solution assumes that the parameters in the PDE, namely v, D, Σ_a and $\nu\Sigma_f$, are fixed values. What would happen to the flux if these parameters were described by probability distributions? The flux would be influenced by perturbations to any of the parameters and so the initial two-dimensional problem is converted to a six-dimensional problem, uncertainty in initial and boundary conditions aside. How would the solution to this SPDE be found? A few preliminaries are in order.

1.2.1 Preliminaries

Most of the language used to describe SPDEs comes from the field of probability theory, which is based on the ideas of sets, fields, and events. The notation and definitions described here come mainly from [54]. The ultimate purpose of introducing the proceeding ideas from probability theory is to be able to understand random variables and the Doob-Dynkin Lemma. Some of the jargon used to describe sets will also be utilized when discussing dimension-wise function decompositions. A *set* is simply a collection of objects while a *subset* is a collection of objects contained within the larger set. A *sample space* is the set of all outcomes of an experiment and is usually denoted by Ω. For example, if the experiment is flipping a fair coin then the sample space is $\Omega = \{H, T\}$. Subsets of Ω are referred to as *events*. If $\Omega = \{\omega_1, \omega_2, ..., \omega_N\}$ then the total number of subsets of Ω is 2^N, where both the empty set $\emptyset$ and all of Ω are included in the count.

A few definitions from set algebra are required before moving on to the Doob-Dynkin Lemma. A *union*, or sum, of two sets A and B is the set of elements that are in at least A or B and is denoted by $A \cup B$. The *intersection* of sets A and B consists of all the elements belonging to both A and B and is denoted by $A \cap B$. Finally, the *complement* of a set A, denoted by A^C, consists of all the elements not in A. From these definition, it follows that $A \cup A^C = \Omega$ and $A \cap A^C = \emptyset$.

Now, let's use the ideas of sets, unions, and intersections to define what is meant by a field and sigma field. Let A and B be subsets of the set Ω. The subsets A and B form a *field* M if,

- $\emptyset \in M$, $\Omega \in M$
- If $A \in M$ and $B \in M$ then $A \cup B \in M$ and $A \cap B \in M$
- If $A \in M$ then $A^C \in M$

A *sigma field* $\mathscr{F}$ is a field that is closed under any countably infinite set of unions and intersections. In other words if subsets $A_1, ..., A_n, ...$ belong to $\mathscr{F}$ then so do $\bigcup_{i=1}^{\infty} A_i$ and $\bigcap_{i=1}^{\infty} A_i$. Consider the sample space $\Omega = \{1, 2, 3\}$. Following the definition of a sigma field, $\mathscr{F} = \{\emptyset, \Omega, \{1\}, \{2, 3\}\}$ is a sigma field while $\mathscr{G} = \{\emptyset, \Omega\{2\}\}$ is not. Rather, the correct sigma field of $\mathscr{G}$ is $\sigma(\mathscr{G}) = \{\emptyset, \Omega, \{2\}, \{1, 3\}\}$. The notation $\sigma(\mathscr{U})$ is used to denote the smallest sigma field containing $\mathscr{U}$, where $\mathscr{U}$ is a collection of subsets of Ω. In general, such a sigma field can be constructed by,

$$\sigma(\mathscr{U}) = \bigcap_{\mathscr{A}} \{\mathscr{U} \subset \mathscr{A} : \mathscr{A} \text{ is a sigma field}\} \tag{1.2}$$

A *Borel sigma algebra* is $\mathscr{B} = \sigma(\mathscr{U})$ where $\mathscr{U}$ consists of all the open sets in $\mathbb{R}^N$.

Combining the ideas described above, a *probability space* $(\Omega, \mathscr{F}, P)$ consists of a sample space Ω, a sigma field $\mathscr{F}$ of subsets of Ω, and a probability function P on $(\Omega, \mathscr{F})$. The probability function must satisfy the three axioms of probability, which are given as:

1. $P(A) \geq 0$
2. $P(\Omega) = 1$
3. $P(A \cup B) = P(A) + P(B)$ if $A \cap B = \emptyset$

An example will help make some of these abstract concepts more concrete. Consider the experiment of tossing a fair coin. The sample space is $\Omega = \{H, T\}$ and the sigma field of events is $\mathscr{F} = \{\{H\}, \{T\}, \Omega, \emptyset\}$. Probabilities of events in $\mathscr{F}$ are $P(H) = P(T) = 1/2$, $P(\Omega) = 1$ and $P(\emptyset) = 0$.

At this point, enough background has been given to describe a *random variable* in general terms. Uncertainty quantification pioneer Gianluca Iaccarino writes, ”Random variables are the building blocks for studying uncertainties in a probabilistic framework” [32]. In the simplest terms, a random variable takes events and assigns them a real number. Let X be a random variable. Then X takes an event ω from the event space Ω and maps it to a number on the real line $X(\omega) : \Omega \to \mathbb{R}$. Under such a mapping a whole region $A_B \in \Omega$ gets mapped to an interval B on the real line. A more formal definition of a random variable can be made using the language of set theory. Let $(\Omega, \mathscr{F}, P)$ be a probability space. A mapping $X : \Omega \to \mathbb{R}^N$ measurable with respect to $\mathscr{F}$ is a random variable. In other words, for any open set A in $\mathbb{R}^N$, $X^{-1}(A) \in \mathscr{F}$ is a random variable.

As an example, consider the event space $\Omega = \{1, 2, 3\}$ and the sigma field $\mathscr{F} = \{\emptyset, \Omega, \{1\}, \{2, 3\}\}$. Then if we define $Y(1) = 1$, $Y(2) = 0$, and $Y(3) = -1$ then Y is not a random variable because $\{3\} \notin \mathscr{F}$. Now, if we define $Z(1) = 1$, and $Z(2) = Z(3) = 0$ then Z is a random variable because $\{1\}$ and $\{2, 3\}$ are both in $\mathscr{F}$. This example serves to demonstrate the interconnectedness between sigma fields and event spaces.

The purpose of the preceding discussion was to build a sufficient framework to be able to state the Doob-Dynkin lemma. Let $X : \Omega \to \mathbb{R}^N$ be a random variable and let $\mathscr{B}$ be the Borel sigma algebra. The sigma algebra generated by X is $\sigma(X) = \left\{X^{-1}(F) : F \in \mathscr{B}\right\}$. The Doob-Dynkin lemma describes the relationship between a random variable and the sigma field it generates. Let $X, Y : \Omega \to \mathbb{R}^N$ be two functions. Then Y is $\sigma(X)$ measurable if and only if there exists a Borel measurable function $g : \mathbb{R}^N \to \mathbb{R}^N$ (for any $A \in \mathscr{B}, g^{-1}(A) \in \mathscr{B}$) such that $Y = g(X)$. To say that Y is ”$\sigma(X)$ measurable” means that if X is known then Y is known as well [54]. When a PDE is transformed into a SPDE by treating the parameters as random variables it is appropriate to question whether the solution of the SPDE can be described in terms of the same random variables [49]. The Doob-Dynkin lemma answers this question with a resounding yes.

1.2.2 Application to Engineering Systems

In this section, the theory of SPDEs is applied to the types of problems that typically arise in nuclear engineering. Much of the notation used in the proceeding discussion is borrowed from [82]. First, define the physical domain as $\mathscr{D} \subset \mathbb{R}^d$ where d can be 1, 2, or 3 depending on the number of spatial dimensions being considered. The boundary of the domain is designated as $\partial\mathscr{D}$. Any coordinate living in the spatial domain can be described by some vector $\mathbf{x} = (x_1, ..., x_d)$. The most general mathematical systems that are of interest can be written as,

$$\mathscr{L}(\mathbf{x}, \omega; u) = f(\mathbf{x}, \omega) \qquad \mathbf{x} \in \mathscr{D} \tag{1.3}$$

$$\mathscr{B}(\mathbf{x}; u) = g(\mathbf{x}) \qquad \mathbf{x} \in \partial\mathscr{D}. \tag{1.4}$$

Ultimately, $u : \Omega \times \mathscr{D} \to \mathbb{R}$ is sought after since u is the solution to 1.3 and 1.4, where $\mathscr{L}$ is a linear/non-linear differential operator, $\mathscr{B}$ is a boundary operator, and f and g are driving terms. The system in 1.3 and 1.4 is defined over a complete probability space $(\Omega, \mathscr{F}, P)$ with $\omega \in \Omega$, as defined previously. Of course, 1.3 and 1.4 must be well-posed in the sense of Hadamard [4]. In Hadamard's definition, well-posed mathematical models of physical phenomena satisfy three conditions:

1. Existence of a solution
2. Uniqueness of the solution
3. The solution is not sensitive to small perturbations in initial conditions

Generally, well-posed problems can be solved using stable computer algorithms. However, it is important not to confuse posedness with conditioning as the two describe very different things. Problems that are typically not well-posed, such as inverse problems, can be formulated as well-posed problems through regularization.

The problem in 1.3 and 1.4 is continuous and as such, will not have an analytic solution for practical problems in engineering applications. An infinite number of random variables are needed to fully describe the stochastic process. Since modeling such a process on a computer is impossible, the infinite-dimensional probability space must be reduced to a finite-dimensional space. The procedure to do this is referred to as the ”finite-dimensional noise assumption". The Karhunen-Loeve expansion of the stochastic space achieves this reduction in dimensionality with the benefit of being able to fully model the full stochastic space if desired [82].

Applying the Karhunen-Loeve expansion to 1.3, the random inputs can be characterized by a set of N random variables as,

$$\mathscr{L}(\mathbf{x}, Y_1(\omega), ..., Y_N(\omega); u) = f(\mathbf{x}, Y_1(\omega), ..., Y_N(\omega)) \tag{1.5}$$

where $\{Y_i(\omega)\}_{i=1}^N$ are uncorrelated random variables. By the Doob-Dynkin lemma, the solution of 1.3 and 1.4 can be expressed in terms of the same random variables $\{Y_i(\omega)\}_{i=1}^N$. Hence, the solution to the SPDE can be written as $u(\mathbf{x}, \omega) = u(\mathbf{x}, Y_1(\omega), ..., Y_N(\omega))$. Equations 1.3 and 1.4 with a discrete stochastic space can be restated as,

$$\mathscr{L}(\mathbf{x}, \mathbf{Y}; u) = f(\mathbf{x}, \mathbf{Y}) \qquad (\mathbf{x}, \mathbf{Y}) \in D \times \Gamma \tag{1.6}$$

$$\mathscr{B}(\mathbf{x}, \mathbf{Y}; u) = g(\mathbf{x}, \mathbf{Y}) \qquad (\mathbf{x}, \mathbf{Y}) \in \partial D \times \Gamma \tag{1.7}$$

where Γ_i is the image of the independent random variables $Y_i(\omega)$. Without loss of generality Γ_i can be restricted to $[0,1]$ for $i = 1, ..., N$. Consequently, the bounded stochastic space, or support, is an N-hypercube $\Gamma = [0,1]^N$. No limits have really been imposed on the stochastic space since any bounded region can be mapped to the unit hypercube. In 1.6 and 1.7 what remains is a set of independent, deterministic equations in space that can be solved using any deterministic discretization technique. The stochastic problem has essentially been decoupled from the spatial problem.

1.3 Uncertainties in Nuclear Data

1.3.1 Cross Section Uncertainty Overview

While the methods described in this thesis are applicable to a plethora of engineering applications, the target application here concerns the simulation of nuclear reactors. A main component of any reactor analysis is a description of the neutronics, the physical processes behind neutron transport inside a reactor. When studying uncertainty quantification and sensitivity analysis of the neutronics in some reactor system, the primary player has been shown to be uncertainty in cross section data [1] [35]. Cross section uncertainties for various reactions, energies, and nuclides are accumulated during their experimental determination. Many of the cross sections exhibit correlations among themselves that must be taken into account. A problem arises when one becomes interested in performing uncertainty and sensitivity analysis on a core simulator since the simulators use processed cross section data. The experimental cross section uncertainties and covariances must be carefully propagated down through the standard cross section processing regime.

Until recently, the primary approach for cross section uncertainty propagation involved the application of first-order perturbation theory. The perturbation theory approach is described in detail in [76]. Basically, for neutronics uncertainty analysis this approach entails the solution of an adjoint transport equation for each response of interest, allowing for the efficient retrieval of sensitivity coefficients for all input data. The sensitivity coefficients can then be used to obtain response uncertainties by being operated on by the inputs' covariance matrix. However, the linear approximations in perturbation theory and generalized perturbation theory may fail in certain scenarios. In particular, if the ratio of neutron loss due to absorption is high, such as when a control rod is rapidly inserted, second order effects may become substantial [76]. In addition, while perturbation theory may be computationally efficient for problems where only a handful of responses are of interest, problems with many responses may be burdensome due to the need to solve the generalized adjoint transport equation for each response. Not to mention, implementation of the perturbation theory approach is intrusive to engineering codes which makes uncertainty quantification of coupled, multiphysics code systems infeasible.

With the recent increase in accessibility to parallel computing environments stochastic sampling methods have become popular since they do not apply any kind of approximations to the physics at hand. Consequently, for cross section uncertainty propagation sampling methods have also been adopted to replace perturbation theory. A description of the process for propagating experimental uncertainties in cross section data to few group cross sections is described in this section. The statistical sampling framework for cross section data is extremely flexible since uncertainties can be calculated for any few group parameters. The same cannot be said of perturbation theory, which must formulate responses as ratios of reaction rates. Consequently, uncertainties for few group transport cross sections and assembly discontinuity factors are difficult to obtain using generalized perturbation theory.

1.3.2 Sampling Method

The method for producing stochastically sampled few-group cross sections will be described. Although the method described is quite general, to provide clarity it will be described in the context of the Standardized Computer Analyses for Licensing Evaluation (SCALE) code [9]. Specifically, SCALE's Evaluated Nuclear Data File (ENDF) libraries and cross section processing utilities will be detailed. In the ENDF libraries the multigroup cross sections are assumed to be Gaussian and consequently, they can be fully characterized by their means and covariances. The SCALE 44-group ENDF/B-VII covariance matrix contains generic multigroup covariance data. For a problem where m nuclide-reaction combinations are of interest the pertinent covariance matrix is expanded to a size of $[m \cdot 44] \times [m \cdot 44]$. The Gaussian assumption imposed on the cross section values implies the cross sections can take on negative values. Since cross sections physically cannot take on negative values their distributions are truncated [43].

The first step towards obtaining perturbed few-group cross sections is to sample the generic multigroup covariance library. The covariance data schema in SCALE is given as relative values of infinitely dilute cross sections [78]. Denote the 1D energy dependent, pointwise cross section for reaction x as $\sigma_x(E)$. The infinitely dilute cross section in energy group g can then be calculated as,

$$\sigma_{x,g}(\infty) = \frac{\langle \sigma_x(E) \rangle}{\Delta u_g} \tag{1.8}$$

where the bracket notation indicates a lethargy inner product over the energy interval covered by group g. Consequently, when the multigroup covariance matrix in SCALE is sampled the resulting perturbation $\sigma'_{x,g}$ to the infinitely dilute cross sections can be expressed in terms of a perturbation factor $Q_{x,g}$,

$$\begin{aligned} \sigma'_{x,g}(\infty) &= \left(1 + \frac{\Delta\sigma_{x,g}(\infty)}{\sigma_{x,g}(\infty)}\right)\sigma_{x,g}(\infty) \\ &= Q_{x,g}\sigma_{x,g}(\infty). \end{aligned} \tag{1.9}$$

The perturbations in Eq. 1.9 will provide generic multigroup cross section values to propagate through a transport solver, which can then yield perturbed few-group cross section values. However, to make the perturbed cross sections problem specific they must undergo adjustments due to resonance self shielding. For this purpose, perturbed pointwise cross sections and perturbed Bondarenko self-shielding factors are required. The pointwise, or continuous energy, cross sections are needed to perform 1D transport calculations in the resolved resonance range while the Bondarenko factors are applicable in the unresolved energy range. Since the pointwise cross sections, Bondarenko factors, and generic infinitely dilute multigroup cross section are related algebraically it is necessary to perturb each in terms of the factor $Q_{x,g}$.

In [78] the authors show that, given a perturbation factor $Q_{x,g}$, the appropriate pointwise cross sections perturbation can be obtained by,

$$\sigma'_x(E) = Q_{x,g}\sigma_x(E) \qquad E \in g. \tag{1.10}$$

From Eq. 1.10 it is clear that the pointwise cross section data gets perturbed uniformly in a given energy group g, which is a valid approximation if the energy width is small. The authors in [78] note that although this approximation leads to discontinuities at the energy group boundaries, it does not impact the resulting multigroup data significantly. No significant impact is expected because the multigroup data is averaged using fluxes and cross sections from the same energy bin.

At this point, infinitely dilute multigroup cross sections and pointwise cross sections can be consistently perturbed using the SCALE ENDF covariance library. To proceed with a transport calculation using perturbed parameters, perturbed Bondarenko self-shielding factors $f'_{x,g}(\sigma_0)$ are also needed. The Bondarenko factors are expressed in terms of the background cross section σ_0, pointwise cross sections, and infinitely dilute cross sections as,

$$f_{x,g}(\sigma_0) = \frac{1}{\sigma_{x,g}(\infty)}\Big\langle\frac{\sigma_x(E)}{\sigma_t(E)+\sigma_0}\Big\rangle\Big/\Big\langle\frac{1}{\sigma_t(E)+\sigma_0}\Big\rangle. \tag{1.11}$$

Substituting Eqs. 1.9 and 1.10 into Eq. 1.11, the perturbed Bondarenko factor is obtained,

$$f'_{x,g}(\sigma_0) = \frac{1}{Q_{x,g}\sigma_{x,g}(\infty)}\Big\langle\frac{Q_{x,g}\sigma_x(E)}{Q_{t,g}\sigma_t(E)+\sigma_0}\Big\rangle\Big/\Big\langle\frac{1}{Q_{t,g}\sigma_t(E)+\sigma_0}\Big\rangle. \tag{1.12}$$

The authors in [78] show that Eq. 1.12 can be evaluated by simply evaluating the unperturbed Bondarenko equation in Eq. 1.11 at a perturbed background cross section $\sigma'_0 = \sigma_0/Q_{t,g}$. Note that only those cross sections whose uncertainties are specified in the SCALE covariance library can be processed and propagated through transport calculations. Since covariance data for 2D scattering distributions is not available in the SCALE ENDF/B-VII covariance library, their uncertainty affects cannot be quantified in any analyses [78].

1.3.3 Cross Section Sampling in SCALE

The method described in section 1.3.2 for producing perturbed cross sections will be further described in the context of the modules in SCALE. The first step is to produce perturbation factors using the module XSUSA by sampling the SCALE ENDF/B-VII covariance library. If a total of N_1 output files of transport solver are desired then N_1 sets of perturbation factors are to be obtained. Using the perturbation factors, the SCALE module CLAROL+ yields perturbed, infinitely dilute, multigroup cross sections by essentially applying Eq.

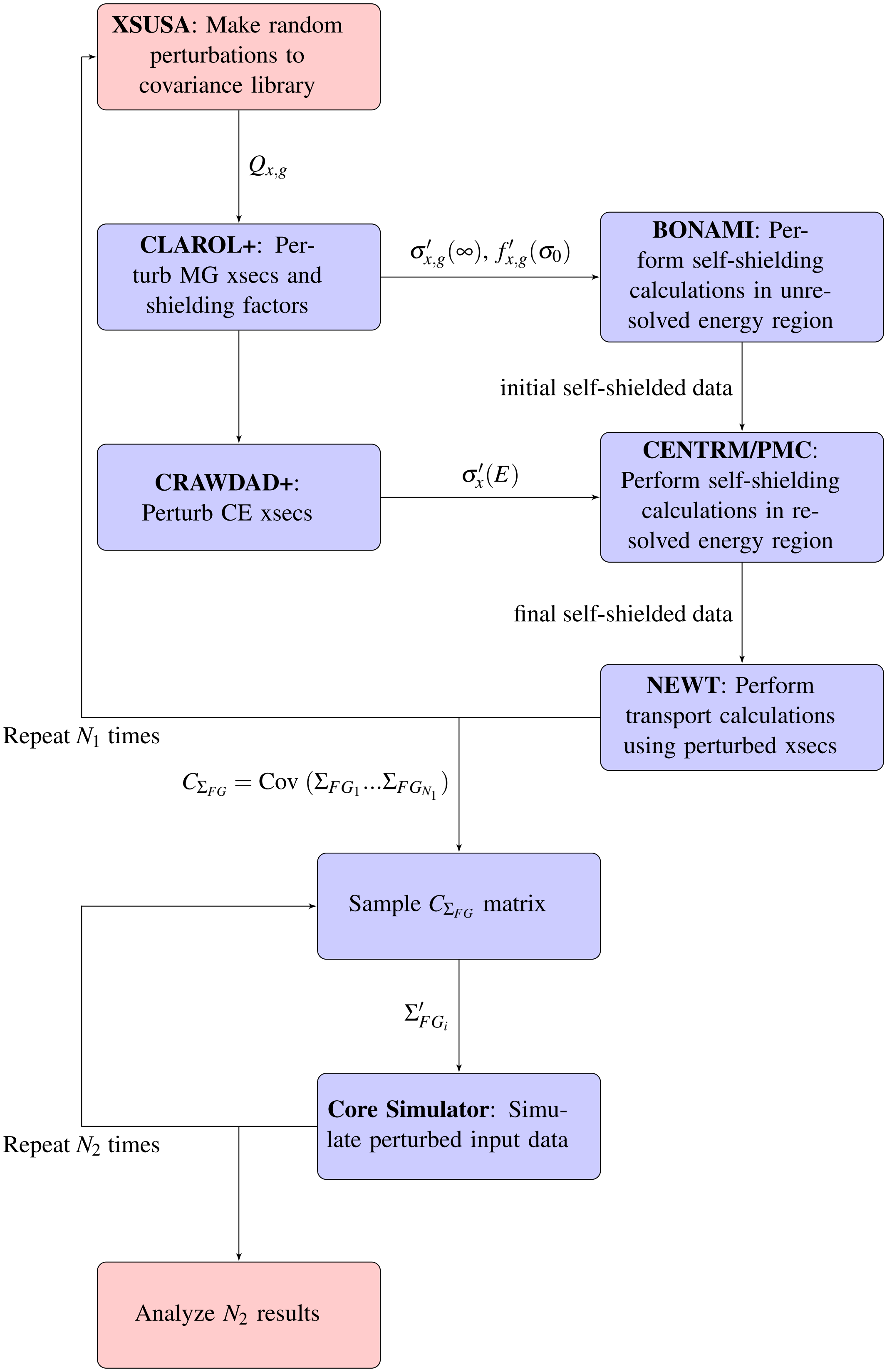
1.9. The resulting cross sections are then passed to the module CRAWDAD+, which outputs perturbed, continuous energy cross sections. To make the multigroup cross sections problem specific they are passed to the BONAMI module, which adjusts the cross sections to account for self-shielding effects in the unresolved energy region using the Bondarenko shielding factor method. Perturbed Bondarenko factors are used in BONAMI, as output from CLAROL+ upon application of Eq. 1.12.

At this point the cross sections still need to be processed for self-shielding effects in the resolved energy region. The SCALE modules CENTRM/PMC are responsible for folding in these effects into the multigroup cross sections. CENTRM solves the neutron slowing down equation on an ultra fine spectra on the order of $\mathscr{O}(10^4)$ energy points. For the flux a linear interpolation scheme is used between energy points to produce a continuous energy solution over the entire energy range [77]. The CENTRM module incorporates actual composition, temperature, and environment dependent effects into the resonance shielded cross sections. PMC accepts the high resolution flux spectra output by CENTRM and averages pointwise nuclear data into multigroup cross sections, which are now tailored to solve the specific problem of interest.

Finally, the problem specific multigroup cross sections output by CENTRM/PMC are sent to a lattice transport solver, which in this case is the SCALE module NEWT. In general, NEWT can be replaced by any transport solver. However, for the purposes of producing few group cross sections a lattice transport solver will generate the multigroup flux distribution for some heterogeneous configuration, which is then used to generate homogenized few group cross sections and intra-assembly flux shapes. A script to extract desirable output quantities from the transport output, such as few group cross sections and kinetics parameters, is needed. Such a script is executed after each of N_1 calls to NEWT. Each call extracts a set of perturbed, few group cross sections that are ultimately propagated through a core simulator. The SAMPLER sequence in SCALE automates the process of producing perturbed cross section sets [78].

With the availability of N_1 perturbed few group cross section sets, each set containing all the nuclear data required by a core simulator, the sets can be directly propagated through the core simulator. The desired results from each simulation can then be gathered and statistically analyzed in what is referred to as the 'one-step' method. However, in what is referred to as the 'two-step' method, the N_1 perturbed few group cross section sets can be used to create a few group covariance matrix [84]. The few group covariance matrix is then sampled and each sample is propagated through the core simulator. Both the 'one-step' and 'two-step' methods have been shown to produce consistent results [84]. However, the 'two-step' method has the advantage of breaking the limitation on the number of samples that can be propagated through the core simulator. In the 'one-step' method each core simulation requires a corresponding transport solution. Transport solutions are relatively expensive to compute when compared to core simulations. A flow diagram of the 'two-step' method is displayed in Figure 1.1. For the 'one-step' method $N_1 = N_2$ whereas for the 'two-step' method $N_2 >> N_1$, allowing for the acquisition of better statistics.

Figure 1.1: Flow diagram of ’two-step’ method for core simulators.


XSUSA: Make random perturbations to covariance library
$Q_{x,g}$
CLAROL+: Perturb MG xsecs and shielding factors
$\sigma'_{x,g}(\infty)$, $f'_{x,g}(\sigma_0)$
BONAMI: Perform self-shielding calculations in unresolved energy region
initial self-shielded data
CRAWDAD+: Perturb CE xsecs
$\sigma'_x(E)$
CENTRM/PMC: Perform self-shielding calculations in resolved energy region
final self-shielded data
NEWT: Perform transport calculations using perturbed xsecs
Repeat N_1 times
$C_{\Sigma_{FG}} = \text{Cov}\ (\Sigma_{FG_1}...\Sigma_{FG_{N_1}})$
Sample $C_{\Sigma_{FG}}$ matrix
Σ'_{FG_i}
Core Simulator: Simulate perturbed input data
Repeat N_2 times
Analyze N_2 results

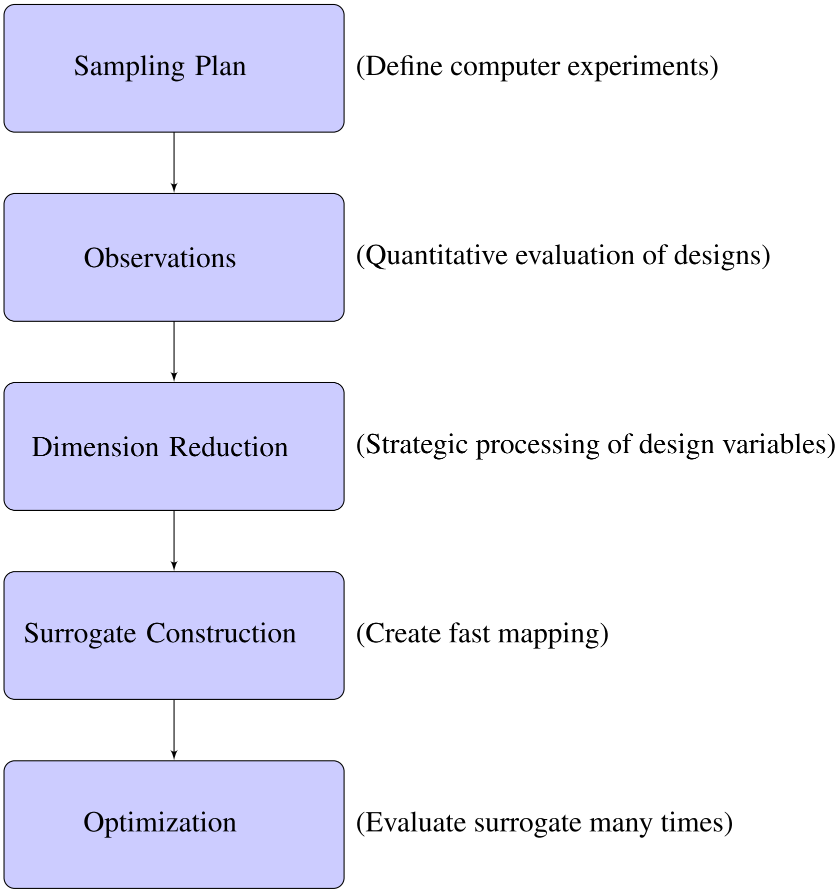
CHAPTER 2

Surrogate Models for Computer Codes

Often an engineering team seeks to perform optimization and calibration routines on a set of computer codes which may require several hours, if not days, to complete a single simulation. Given the limited computational budget available to such teams a surrogate model is typically sought for the computational codes. The use of a surrogate model for an expensive computer code effectively exchanges predictive accuracy for simulation execution time. Unfortunately, to construct a surrogate model the expensive computer code must still be executed a certain number of times, although still significantly less times than required to conduct an optimization study. At this point there are two directions that can be taken to construct the surrogate model. If a computational budget is limited to say, *N* evaluations of the objective function then it is necessary to somehow optimize the set of input points at which the objective computer code is evaluated. The general strategy for constructing a surrogate is outlined in Fig. 2.1, with each component described in detail in the proceeding sections. Sampling plan optimization strategies are described in section 2.1, while surrogates based on the resulting sampling plan, namely Kriging, is described in section 2.2. If there is more flexibility in the computational budget then collocation-based surrogates should be considered since they offer the benefits of built-in convergence metrics.

Computer codes that model physical phenomena typically accept an input file whose purpose is to describe specific conditions in the universe being modeled by the code. The code is executed and the affects of the conditions on some dependent quantities are output. From this perspective computer codes can be viewed and treated in much the same way functions mathematicians deal with are treated. Like matrices, functions with certain properties can undergo various decompositions that offer insight into their structure. The purpose of section 2.3 is to describe a technique for decomposing functions into orthogonal components, with the ultimate intention of applying the technique to computer codes. It is hoped that the decomposition of the computer code in terms of its inputs reveals which inputs play the most active roles in the underlying physics. Keeping only the most active dimensions in the decomposition, a reduced order model is effectively built. However, the

Figure 2.1: General flow diagram for constructing a surrogate for an expensive computer code.


Sampling Plan
(Define computer experiments)
Observations
(Quantitative evaluation of designs)
Dimension Reduction
(Strategic processing of design variables)
Surrogate Construction
(Create fast mapping)
Optimization
(Evaluate surrogate many times)

function decomposition technique described in section 2.3 describes only half the story. To create a reduced order model of a computer code that can be efficiently evaluated at any state point in the original parameter space an efficient, multidimensional, interpolation scheme is needed. Such a scheme is discussed in section 2.4. The coupling between interpolation and function decomposition that creates a usable reduced order model is then described in section 2.5.

2.1 Optimized Sampling Plans

The following section is concerned with the problem of identifying an optimal set of points at which to build a surrogate model for an expensive computer code when only N evaluations can be afforded. All surrogate models are built around a set of points at which the objective computer code is actually evaluated. Intuitively, the surrogate accuracy is expected to decrease as one moves further away from such points. Consequently, it is important to spread N points as uniformly as possible across the design space.

2.1.1 Morris Algorithm

Expensive computer codes generally have many input parameters because they attempt to model some phenomena on a very fine scale as accurately as possible. Of course, when writing such computer codes engineers are not aware which input parameters have the greatest impact on the outputs of interest or else only these parameters would be modeled. Contrarily, engineers typically only know that certain parameters are involved in the pertinent physics in some way but not to what extent. Due to the curse of dimensionality, the less design variables considered in the construction of a surrogate the cheaper the computational cost will be [20]. Consequently, before attempting to construct a surrogate for an expensive computer code it is worth identifying which design variables are most active. After all, if a design variable has a trivial effect on some output of interest then its presence in the surrogate should be minimized, for this is the very purpose of a surrogate model. One method for weeding out unimportant design variables is described by Morris [52], which is summarized in algorithm 1.

The premise of Morris' algorithm is that if the derivative of some output parameter with respect to a design variable changes significantly throughout the design space then the variable is important. If the output parameter does not change with respect to the design variable then the variable can safely be ignored. To this end, the metric in Eq. 2.1 is introduced by Morris to estimate the so-called elementary effect $d_i(\mathbf{x})$ of design variable x_i.

$$d_i(\mathbf{x}) = \frac{f(x_1, x_2, ..., x_{i-1}, x_i + \Delta, x_{i+1}, ..., x_k) - f(\mathbf{x})}{\Delta} \tag{2.1}$$

In Eq. 2.1, Δ is a step-length size for the perturbation. For convenience all variables x_i are normalized to unit length and divided into p segments such that $x_i \in \{0, 1/(p-1), 2/(p-1), ..., 1\}$. Choosing a set of $\mathbf{x}$ carefully, it is possible to calculate an elementary effect for each of k design variables using only $k+1$ function evaluations. Indeed, as described in [52], a $[k+1] \times [k]$ random orientation matrix B^* can be constructed using the equation,

$$\mathbf{B}^* = \left(\mathbf{1}_{k+1,1}\mathbf{x}^* + \frac{\Delta}{2}\left[\left(2\mathbf{B} - \mathbf{1}_{k+1,k}\right)\mathbf{D}^* + \mathbf{1}_{k+1,k}\right]\right)\mathbf{P}^*. \tag{2.2}$$

In Eq. 2.2, $\mathbf{1}$ is a matrix of ones with size specified by its subscript and $\mathbf{B}$ is a $[k+1] \times [k]$ matrix of zeros and ones with the characteristic that for each column there exists a pair of rows differing in only their i^{th} entry for $i \in \{1, 2, ..., k\}$. Also, $\mathbf{D}^*$ is a $[k] \times [k]$ diagonal matrix with ones of differing parity uniformly spread, $\mathbf{P}^*$ is a $[k] \times [k]$ random permutation matrix, and $\mathbf{x}^*$ is a random point chosen in the p-level design space. When the rows of $\mathbf{B}^*$ are evaluated by the objective function and substituted into Eq. 2.1 an elementary effect is calculated for each design variable.

The more elementary effects that can be calculated for each design variable, the better the estimate as to the effect of each design variable on the objective function. Consequently, r random orientation matrices are typically created to obtain a total of r elementary effects for each design variable. Taking the mean and standard deviation of each variable's r effects can yield insight into the most important variables. Plotting the mean and standard deviation of each variable's effects on a scatter plot, variables that have a negligible effect on the objective function will cluster around the origin. Large fluctuations in standard deviation are indicative of nonlinear and interactive effects [52].

Algorithm 1 Uses Morris' Algorithm to determine which of a function's design variables induce the most significant effects and interactions.

```
Initialize zeros matrix $d_{stats}$ of size $[r] \times [k]$
for $i = 1 : r$ do
    $X \rightarrow$ Create random orientation matrix                ▷ Eq. 2.2
    $f_{base} = f(X[0,:])$
    for $j = 1 : k$ do
        $f_{new} = f(X[k,:])$
        $l \rightarrow$ Find index of effect being perturbed
        $d_{effect} \rightarrow$ Calculate elementary effect          ▷ Eq. 2.1
        $d_{stats}[i,l] = d_{effect}$
        $f_{base} := f_{new}$
    end for
end for
Return mean and standard deviation of $d_{stats}$
```

2.1.2 Latin Hypercube Sampling

One of the major problems with random sampling occurs when a relatively limited sample size is utilized. In this case, subsets of the sample space with high consequence but low probability are likely to be missed [28]. In addition, the proximity of sampled values caused by random sampling is inefficient, which often causes slow convergence. In order to resolve such issues LHS was conjured. The basis of LHS rests upon dividing the normalized space of each design variable into n equally sized bins if n samples are required. As a result, when the n samples are taken it is guaranteed that the entire spectrum of each design variable's space has been visited. Algorithmically an n sample Latin hypercube in k dimensions can be easily calculated, as described in algorithm 2.

Algorithm 2 Creates a random Latin hypercube consisting of n samples in k dimensions.

```
Initialize zeros matrix $X$ of size $[n] \times [k]$
for $i = 1 : k$ do
    $p \rightarrow$ Create random permutation of the set $\{1, 2, ..., n\}$
    $X[:, i] := p[:]$
end for
Map each entry of $X$ into hypercube
Return $X$
```

Each row of the output from algorithm 2 is a sample point normalized to a hypercube. As mentioned previously, when sampling a design space it is important to take sample points uniformly. While LHS increases the likelihood of obtaining such a uniform sampling space at random it is actually possible to obtain an optimized sampling space based on the maximin metric [20].

The maximin metric describe by Morris and Mitchell [53] makes use of two notions in an attempt to quantify the uniformity, or 'space-fillingness', of a set of sampling points. In order to describe the notions for each sampling plan it is useful to gather $\{d_1, d_2, ..., d_m\}$ and $\{J_1, J_2, ..., J_m\}$, the unique distances between all points in the plan sorted in ascending order and the number of occurrences of each distance, respectively. In words, the Morris and Mitchell criteria states that an optimized sampling plan will minimize all J_i while maximizing the corresponding d_i. More formally, Morris and Mitchell define the maximin sampling plan as one which maximizes d_1, and among plans for which this is true, minimizes J_1, among plans for which this is true, maximizes d_2, and so on [20]. The previous definition can be restated into pseudo-equivalent minimization problem by introducing the parameter $\Phi_q(\mathbf{X})$,

$$\Phi_q(\mathbf{X}) = \left(\sum_{j=1}^{m} J_j d_j^{-q} \right)^{1/q} \tag{2.3}$$

where $\mathbf{X}$ is a sampling plan and q is a control parameter inherent in the minimization problem.

The minimization of Eq. 2.3 and the Morris and Mitchell definition of the maximin sampling plan are generally used in unison to obtain a locally optimal sampling plan since finding the globally optimal plan is computationally infeasible. In this approach a random sampling plan $\mathbf{X}_0$ is initially generated using algorithm 2. Using a range of q values, usually from one to one-hundred, an optimal latin hypercube plan is found for each value based on the initial sampling plan $\mathbf{X}_0$. To obtain an optimized plan for each q algorithms such as simulated annealing [42] and evolutionary operation [10] can be used. These algorithms work to minimize Eq. 2.3 by comparing an initial sampling plan to perturbed versions, which are obtained by switching pairs of column entries in the output of algorithm 2. Once an optimized LHS plan is found for each q value, the resulting set of plans are contested directly against each other by explicit application of Morris and Mitchell's maximin definition. The sampling plan satisfying the maximin criteria is the locally optimal sampling plan to be used for proceeding surrogate model construction. Algorithm 3 summarizes the search for a locally optimal LHS plan.

Algorithm 3 Obtains a locally optimal LHS plan using the Morris-Mitchell minimax criteria.

```
Initialize sampling plan $\mathbf{X}_0$                                  ▷ Apply algorithm 2
$q = [1, 2, 5, 10, 20, 50, 100]$
$\mathbf{X}^{cands.} \rightarrow$ Initialize empty array for optimal $\mathbf{X}$ candidates
for $q_i$ in $q$ do
    $\mathbf{X}_{opt}(q_i) \rightarrow$ Find optimal $\mathbf{X}$ for $q_i$ using simulated annealing    ▷ Minimize Eq. 2.3
    Add $\mathbf{X}_{opt}(q_i)$ to $\mathbf{X}^{cands.}$
end for
Apply Morris-Mitchell criterion between all $(\mathbf{X}_i^{cands.}, \mathbf{X}_j^{cands.})$ to find optimal plan
```

2.2 Kriging

The formulation of kriging is such that a deterministic computer code's output is assumed to be a stochastic process, enabling a statistical approach to a surrogate construction [63]. To this end, consider the sampling data $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(n)}\}$, which may have come from the result of an optimized sampling plan as described in section 2.1.2. At each datum $\mathbf{x}^{(k)}$ a random process $Y(\mathbf{x}^{(k)})$ induces an observation $y^{(k)}$. In kriging the random field can be described with a mean value of $\mathbf{1}\mu$, where $\mathbf{1}$ is a vector of length n, and a correlation matrix of the random field instances,

$$\Psi = \begin{pmatrix} cor[Y(\mathbf{x}^{(1)}), Y(\mathbf{x}^{(1)})] & \cdots & cor[Y(\mathbf{x}^{(1)}), Y(\mathbf{x}^{(n)})] \\ \vdots & \ddots & \vdots \\ cor[Y(\mathbf{x}^{(n)}), Y(\mathbf{x}^{(1)})] & \cdots & cor[Y(\mathbf{x}^{(n)}), Y(\mathbf{x}^{(n)})] \end{pmatrix}. \tag{2.4}$$

The correlation between two stochastic processes $Y(\mathbf{x}^{(i)})$ and $Y(\mathbf{x}^{(l)})$ is expressed as,

$$cor[Y(\mathbf{x}^{(i)}), Y(\mathbf{x}^{(l)})] = \exp\left(- \sum_{j=1}^{k} \theta_j |x_j^{(i)} - x_j^{(l)}|^{p_j} \right) \tag{2.5}$$

where θ_j and p_j are optimization parameters controlling the magnitude of correlation at each basis. Specifically the parameter p_j controls a correlation's smoothness while θ_j defines the spread.

Given the formulation of the observations occurring at $\mathbf{x}^{(k)}$ as instances of a stochastic process it is appropriate to discuss the likelihood of seeing the observed data. Of course, the likelihood is conditional on the parameters described in Eq. 2.5 as well as the mean and standard deviation of the random field. The likelihood of seeing the observed data can be written as,

$$L\left(\mathbf{Y}^{(1)},...,\mathbf{Y}^{(n)}|\mu,\sigma,\{\theta_1,...,\theta_k\},\{p_1,...,p_k\}\right) = \frac{1}{(2\pi\sigma^2)^{n/2}|\Psi|^{1/2}} \times \cdots \\ \times \exp\left[\frac{(\mathbf{y}-\mathbf{1}\mu)^T\Psi^{-1}(\mathbf{y}-\mathbf{1}\mu)}{2\sigma^2}\right]. \tag{2.6}$$

In order to maximize the likelihood L, that is, picking values for the mean and standard deviation that maximize the probability of seeing the observed data, the standard procedures taught in calculus can be applied to obtain the maximum likelihood estimators,

$$\hat{\mu} = \frac{\mathbf{1}^T\Psi^{-1}\mathbf{y}}{\mathbf{1}^T\Psi^{-1}\mathbf{1}} \tag{2.7}$$

$$\hat{\sigma}^2 = \frac{(\mathbf{y}-\mathbf{1}\mu)^T\Psi^{-1}(\mathbf{y}-\mathbf{1}\mu)}{n}. \tag{2.8}$$

The estimators in Eq. 2.7- 2.8 are actually estimators of the natural logarithm of the likelihood since this form is easier to work with. Substitution of Eq. 2.7- 2.8 into the natural logarithm likelihood leads to the so called concentrated ln-likelihood function [20], given as,

$$\log(L) \approx -\frac{n}{2}\log\left(\hat{\sigma}^2\right) - \frac{1}{2}\log|\Psi|. \tag{2.9}$$

While it is relatively easy to optimize the natural logarithm likelihood with respect to μ and σ it is much more difficult to optimize Eq. 2.9 with respect to the θ and p parameters discussed earlier. Indeed, global search algorithms such as simulated annealing [42] are generally used to find optimizing θ and p parameters. Since such search algorithms require repetitive calls to the objective function, in this specific optimization problem it is important to be able to evaluate Eq. 2.9 efficiently. Quick inspection of Eqs. 2.7 - 2.9 leads to the observation that Ψ must be inverted, a common computational bottleneck for large matrices. Fortunately, being symmetric positive-definite, the matrix inversion action of Ψ can be completed efficiently by performing Cholesky decomposition followed by calls to forward and backwards substitution routines.

Once all optimizing parameters are available the goal is to utilize the parameters to build a model that makes function predictions on new points $\mathbf{x}$. Of course, the prediction should be consistent with all correlation parameters. To this end, for some new data point $\mathbf{x}$ a vector of correlations with existing points must be constructed using Eq. 2.5,

$$\psi = \begin{pmatrix} cor[Y(\mathbf{x}^{(1)}), Y(\mathbf{x})] \\ \vdots \\ cor[Y(\mathbf{x}^{(n)}), Y(\mathbf{x})] \end{pmatrix}. \tag{2.10}$$

Using Eq. 2.10 new predictions can be made at $\mathbf{x}$ using the maximum likelihood estimator,

$$\hat{y}(\mathbf{x}) = \hat{\mu} + \psi^T \Psi^{-1} (\mathbf{y} - \mathbf{1}\hat{\mu}). \tag{2.11}$$

To get some intuition for Eq. 2.11, it is worth while to consider the second term on the right-hand side as a perturbation to the mean of the stochastic process. The term $\Psi^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu})$ is a vector of coefficients of the unique linear expansion of $(\mathbf{y} - \mathbf{1}\hat{\mu})$ in the basis columns of Ψ. From this point of view, prediction using kriging works to estimate a function value at a certain point by computing a weighted average of known function values in the vicinity of the objective points [33].

2.3 Function Decompositions

In order to reduce the dimensionality of some function there must exist a metric to determine the importance of each dimension with respect to the others. It's important to have a framework that isolates the effects of each dimension on the output of the function. The framework chosen to perform dimension reduction is formally known as high-dimensional model representation (HDMR). In statistics, the ANOVA decomposition is a special case of HDMR where the Lebesgue measure is used to perform all integrations.

2.3.1 Dimension-wise Decompositions

The dimension-wise HDMR is algorithmically similar to Gram-Schmidt for matrix orthogonalization in that orthogonal components are systemically removed to create a linearly independent basis. As in Gram-Schmidt, the essential operator in dimension-wise HDMR is the projection operator. Before introducing the projection operator of interest in this thesis, define the d-dimensional product measure to be,

$$d\mu\left(\mathbf{x}\right) = \prod_{j=1}^{d} d\mu_j\left(x_j\right) \tag{2.12}$$

where μ_j are probability measures defined over some Ω. Two functions $f,g : \Omega^d \rightarrow \mathbb{R}$ are considered orthogonal with respect to the product measure defined in 2.12 if the inner product,

$$(f,g) = \int_{\Omega^d} f(\mathbf{x})g(\mathbf{x})d\mu(\mathbf{x}) \tag{2.13}$$

is equal to zero. To introduce the projection operator $P_{\mathbf{u}}$, the notation used in [30] is adopted. The operator $P_{\mathbf{u}}$ projects from a d-dimensional space to a $|\mathbf{u}|$-dimensional space for some set $\mathbf{u} \subseteq \mathscr{D}$, where $\mathscr{D} = \{1,...,d\}$ consists of the set of all coordinate indices in $\mathbf{x}$. The projection on to $\mathbf{u}$ is given as,

$$P_{\mathbf{u}}f\left(\mathbf{x_u}\right) = \int_{\Omega^{d-|\mathbf{u}|}} f\left(\mathbf{x}\right) d\mu_{\mathscr{D}\setminus\mathbf{u}}\left(\mathbf{x}\right) \tag{2.14}$$

where $\mathbf{x_u}$ has length $|\mathbf{u}|$ and consists of the $\mathbf{x}$ coordinates specified in $\mathbf{u}$. Also, the notation $\mathscr{D} \setminus \mathbf{u}$ signifies all the coordinates in $\mathscr{D}$ not contained in $\mathbf{u}$. From 2.14 it is clear that the projection operator works to integrate out all coordinate indices not contained in $\mathbf{u}$ from f. For some coordinate indice sets $\mathbf{u}$ and $\mathbf{v}$, where $\mathbf{u} \neq \mathbf{v}$, the following orthogonality relation holds,

$$(f_{\mathbf{u}}, f_{\mathbf{v}}) = 0. \tag{2.15}$$

The notation $f_{\mathbf{u}}$ is used to denote the function of only the coordinate indices contained in $\mathbf{u}$. From 2.15, it follows that a function can be written in terms of its 2^d orthogonal components,

$$f\left(\mathbf{x}\right) = \sum_{\mathbf{u}\subseteq\mathscr{D}} f_{\mathbf{u}}\left(\mathbf{x_u}\right) \tag{2.16}$$

where the component functions $f_{\mathbf{u}}$ are defined recursively as [30],

$$f_{\mathbf{u}}\left(\mathbf{x_u}\right) = P_{\mathbf{u}}f\left(\mathbf{x_u}\right) - \sum_{\mathbf{v}\subset\mathbf{u}} f_{\mathbf{v}}\left(\mathbf{x_v}\right). \tag{2.17}$$

The recursive definition in 2.17 can be written explicitly as,

$$f_{\mathbf{u}}\left(\mathbf{x_u}\right) = \sum_{\mathbf{v}\subseteq\mathbf{u}} (-1)^{|\mathbf{u}|-|\mathbf{v}|} P_{\mathbf{v}}f\left(\mathbf{x_v}\right) \tag{2.18}$$

For most functions arising in engineering applications, especially if the function is a computer code, the decomposition in 2.16 is not possible to obtain because each component function $f_{\mathbf{u}}$ will require a high-dimensional integral to be performed. Of course, this statement assumes a Lebesgue measure in the definition of $d\mu$ in 2.12. Alternatively, if a Dirac measure is used then the computationally burdensome integral in 2.14 is reduced to a single function evaluation. In this case, the decomposition in 2.16 is referred to as an anchored-ANOVA decomposition, or CUT-HDMR [50].

2.3.2 Anchored-ANOVA Decomposition

Using the Dirac measure $\delta(\mathbf{x}-\mathbf{a})d\mathbf{x}$ to evaluate the projection operator at a fixed point $\mathbf{a} \in [0,1]^d$ in the hypercube, equation 2.14 becomes,

$$P_{\mathbf{u}} f(\mathbf{x}_{\mathbf{u}}) = f(\mathbf{x})\,|_{\mathbf{x}=\mathbf{a}\setminus\mathbf{x}_{\mathbf{u}}}. \tag{2.19}$$

The notation $\mathbf{a} \setminus \mathbf{x}_{\mathbf{u}}$ is the anchor point $\mathbf{a}$ except at the coordinate indices specified in $\mathbf{u}$. At the coordinate indices $\mathbf{u}$, the anchor point takes upon the corresponding values in $\mathbf{x}$. Using the Dirac measure to evaluate the projections comprising 2.16, the objective function is expressed as a linear combination of its values along lines, faces, hyperplanes,..., etc [30]. As mentioned previously, using the Dirac measure to perform the projection operations in HDMR results in enormous computational savings since high-dimensional integrals are replaced with single function evaluations.

Given the structure of anchored-ANOVA, it is not surprising to learn that there exists a close connection to multivariate Taylor series [47]. This connection provides insight into some of the properties of the anchored-ANOVA decomposition. The multivariate Taylor series of $f(\mathbf{x})$ about a point $\bar{\mathbf{x}}$ can be written as,

$$f(\mathbf{x}) = f(\bar{\mathbf{x}}) + \sum_{i=1}^{d} \frac{\partial f(\mathbf{x})}{\partial x_i}(x_i - \bar{x}_i) + \frac{1}{2!}\sum_{i,j=1}^{d} \frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j}(x_i - \bar{x}_i)\left(x_j - \bar{x}_j\right) + \ldots \tag{2.20}$$

As an example, consider what happens if 2.20 is evaluated at $\mathbf{x} = \mathbf{a} \setminus x_i$,

$$f(\mathbf{x})\,|_{\mathbf{x}=\mathbf{a}\setminus x_i} = f(\bar{\mathbf{x}}) + \frac{\partial f(\mathbf{x})}{\partial x_i}(x_i - \bar{x}_i) + \frac{1}{2!}\frac{\partial^2 f(\mathbf{x})}{\partial x_i^2}(x_i - \bar{x}_i)^2 + \ldots \tag{2.21}$$

Since $f_i(x_i) = f(\mathbf{x})\,|_{\mathbf{x}=\mathbf{a}\setminus x_i} - f(\bar{\mathbf{x}})$,

$$f_i(x_i) = \frac{\partial f(\mathbf{x})}{\partial x_i}(x_i - \bar{x}_i) + \frac{1}{2!}\frac{\partial^2 f(\mathbf{x})}{\partial x_i^2}(x_i - \bar{x}_i)^2 + \ldots \tag{2.22}$$

Expression 2.22 shows that the first-order component functions in anchored-ANOVA consist of entire Taylor series expansions. Similarly, second-order component functions will consist of their respective entire Taylor series expansions and so on. Consequently, a truncated anchored-ANOVA expansion will always provide a better approximation to a function than a truncated Taylor expansion [47].

2.3.2.1 Effective Dimensions

The ultimate purpose of introducing an expansion such as anchored-ANOVA is to truncate it and use the truncated portion as an approximation to the objective function. Of course, evaluation of the truncated anchored-ANOVA expansion is expected to be much more computationally efficient than the objective function. When an anchored-ANOVA decomposition is truncated, the loss incurred becomes the components 2.17 that are not being represented. Of course, in practical construction the components not represented are calculated to contribute relatively trivially. Two notions exist for classifying the dimension of a truncated anchored-ANOVA decomposition. Both notions depend on $\hat{\sigma}(f)$, which is the sum of the absolute values of the integrals of all anchored-ANOVA terms [30]

$$\hat{\sigma}(f) = \sum_{\substack{u \subseteq \mathscr{D} \\ \mathbf{u} \neq \emptyset}} |If_{\mathbf{u}}| \approx \sum_{\substack{u \subseteq \mathscr{D} \\ \mathbf{u} \neq \emptyset}} |q_{\mathbf{u}}|. \tag{2.23}$$

The notation $I\cdot$ represents an exact integral but, in practice the integral will be evaluated using some multivariate quadrature scheme and so the exact integral's approximation is denoted by $q_{\mathbf{u}} \approx If_{\mathbf{u}}$. For some user-defined threshold $\alpha \in [0,1]$ the truncation and superposition dimensions of a truncated anchored-ANOVA expansion can be defined. The truncation dimension attempts to quantify the importance of a certain number of dimensions d_t. Mathematically, the truncation dimension is the smallest integer d_t such that,

$$\sum_{\substack{\mathbf{u} \subseteq \{1,\ldots,d_t\} \\ \mathbf{u} \neq \emptyset}} |q_{\mathbf{u}}| \geq \alpha \hat{\sigma}(f).$$

Contrastingly, the superposition dimension attempts to quantify the order of important dimensions d_s. Mathematically, the superposition dimension is the smallest dimension d_s such that,

$$\sum_{\substack{|\mathbf{u}| \leq d_s \\ \mathbf{u} \neq \emptyset}} |q_{\mathbf{u}}| \geq \alpha \hat{\sigma}(f).$$

Both definitions for the effective definition of a truncated anchored-ANOVA expansion can be directly related to the exact integral of the objective function If. Specifically, for the truncation dimension the following relation holds [30],

$$|If - \sum_{\mathbf{u}\subseteq\{1,\ldots,d_t\}} If_{\mathbf{u}}| \leq (1-\alpha)\,\hat{\sigma}\,(f)\,. \tag{2.24}$$

Similarly, for the superposition dimension the following inequality holds,

$$|If - \sum_{|\mathbf{u}|\leq d_s} If_{\mathbf{u}}| \leq (1-\alpha)\,\hat{\sigma}\,(f)\,. \tag{2.25}$$

Inequalities 2.24 and 2.25 suggest that if all the anchored-ANOVA terms are used then the exact integral of the objective function can be reproduced. However, in general the set of effective dimensions as determined by anchored-ANOVA will not be equal to the set determined by a classic ANOVA decomposition. The choice of the anchor point **a** has a direct influence on the accuracy and truncation dimension of the anchored-ANOVA expansion [22]. In [22] the authors argue that choosing the anchor point to be the centroid of the parameter space is an excellent choice for most applications. As such, in this thesis the anchor point is always chosen to be the centroid of the working parameter space Ω^d.

2.4 Smolyak Sparse Grids

In order to create a reduced-order model for some objective function the anchored-ANOVA decomposition plays a crucial role but more is needed [23]. Recall that the primary purpose for constructing a reduced-order model is to replace the presumably computationally intensive objective function with something that is trivial to evaluate. Consequently, in evaluating the anchored-ANOVA decomposition at some point **x** the projections in 2.19 must be trivial to evaluate as well. As it stands, evaluating the anchored-ANOVA decomposition for some objective function is significantly more expensive than simply evaluating the function itself. To resolve this issue, a Smolyak sparse grid interpolant is created for each projection. While creating each such interpolant incurs some initial overhead, the payoff is the desired reduced order model.

2.4.1 Motivation

To describe multivariate function interpolation based on Smolyak sparse grids it makes sense to speak in the context of quadrature since a quadrature rule consists of interpolating a function using polynomials and then integrating the polynomials exactly. For the moment consider some smooth 1D function $f(x)$. The function $f(x)$ can be approximated arbitrarily well through the summation,

$$f(x) \approx \sum_{i=1}^{P} f(x_i) C_i(x) \tag{2.26}$$

where $C_i(x)$ are cardinal functions of degree P with the property that $C_i(x_j) = \delta_{ij}$, δ_{ij} being the Kronecker δ-function [11]. By the Weierstrass approximation theorem, smooth functions can be uniformly approximated as closely as desired by polynomial functions [73]. At the collocation points, or abscissas, in 2.26 the function $f(x)$ is interpolated exactly at x_i. The function $f(x)$ is comprised of various constant, linear, quadratic, cubic,..., etc terms and so exact integration of $f(x)$ amounts to integrating its monomial constituents.

Suppose that instead of interpolating a 1D function, a multivariate function of d dimensions is to be interpolated. The naive approach to multivariate interpolation is to take a Cartesian product of 1D rules, such as in 2.26, d times. Consequently, the product grid will contain P^d points, each of which requires a unique function evaluation. Such exponential growth is coined the ”curse of dimensionality” [51]. As a rule of thumb, exact integration of a monomial constituent comes at the cost of a single function evaluation [14]. Considering the space of d-dimensional, P-degree polynomials has some,

$$\binom{P+d}{d} \approx \frac{d^P}{P!} \tag{2.27}$$

dimensions, for high dimensional problems the full Cartesian product approach integrates a superfluous number of monomials. Russian mathematician Sergei Smolyak was one of the first to realize the potential computational savings in his paper [67].

2.4.2 Algorithm Mechanics

A Smolyak sparse grid is the set of collocation points used to build an interpolant for some multivariate objective function while the Smolyak algorithm is the whole procedure of building the interpolant. To begin, the Smolyak algorithm will be stated and pertinent notation will be introduced. Since indice tracking comprises the brunt of understanding the Smolyak algorithm, it is crucial to choose a clear notation. Consequently, the notation used here closely follows that of [6].

Slightly generalizing 2.26, for the case of some smooth 1D function f, let U^i be the interpolant of f comprised of m_i collocation points.

$$U^i = \sum_{j=1}^{m_i} f\left(x_j^i\right) a_j^i \tag{2.28}$$

In 2.28, $i \in \mathbb{N}$, and $a_j^i \in C([-1,1])$ are basis functions imposing the demand that U^i exactly be able to reproduce f at the collocation points x_j^i. The notation $x_j^i \in [-1,1]$ refers to the j^{th} collocation point of m_i total points. Restricting the domain of the collocation points to $[-1,1]$ does not impose any limitations on being able to interpolate f arbitrarily well since $[-1,1]$ can always be mapped to the parameter space of f.

To generalize from 1D interpolation to multivariate interpolation 1D interpolation formulas, such as the one in 2.28, are combined using tensor products.

$$\left(U^{i_1} \otimes \cdots \otimes U^{i_d}\right)(f) = \sum_{j_1=1}^{m_{i_1}} \cdots \sum_{j_d=1}^{m_{i_d}} f\left(x_{j_1}^{i_1}, \cdots, x_{j_d}^{i_d}\right) \cdot \left(a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_d}^{i_d}\right) \tag{2.29}$$

Tensor products are a mathematical convenience used to represent all combinations of some entity, in this case U^i. The scheme in 2.29 suffers from the, ”curse of dimensionality” since a total of,

$$\prod_{k=1}^{d} m_{i_k} \tag{2.30}$$

function evaluations are needed to form the interpolant. The Smolyak algorithm is based on 2.29, the only difference being not all the tensor products are used. In explicit form, the Smolyak formula for approximating the left-hand side of 2.29 is given as [6],

$$A(q,d) = \sum_{q-d+1 \leq |\mathbf{i}| \leq q} (-1)^{q-|\mathbf{i}|} \binom{d-1}{q-|\mathbf{i}|} \left(U^{i_1} \otimes \cdots \otimes U^{i_d}\right). \tag{2.31}$$

Each entry i_k in the vector $\mathbf{i} \in \mathbb{N}^d$ contains the indice corresponding to the level of interpolation in dimension k. The more collocation points being utilized, the higher the level of interpolation since the interpolant becomes increasingly accurate. The magnitude of $\mathbf{i}$ is $|\mathbf{i}| = |i_1 + \cdots + i_d|$. Since each $i_d \geq 1$, the variable $q \geq d$. The variable q essentially keeps track of the level of interpolation of the Smolyak algorithm. As q is increased, more tensor product combinations are allowed. From 2.31 it is clear that the Smolyak algorithm is able to reduce the total number of tensor product components by limiting the entries of $\mathbf{i}$. Note, when performing analysis using the Smolyak algorithm the ”interpolation level” typically refers to $q-d$ in order to ground the analysis at an interpolation level of zero.

The Smolyak formula in 2.31 can be rewritten in several ways, all of which use the idea of the incremental interpolant Δ^i defined as,

$$\begin{aligned} U^0 &= 0 \\ \Delta^i &= U^i - U^{i-1} \end{aligned} \tag{2.32}$$

The incremental interpolant operator is simply the difference between interpolants at two successive levels. Using the notion of the incremental interpolant, the Smolyak formula can be rewritten as,

$$A\left(q,d\right) = \sum_{|\mathbf{i}| \leq q} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d}\right) \tag{2.33}$$

At first sight, 2.31 and 2.33 seem inefficient since neither exposes the recursiveness inherent in the Smolyak formula. In other words, when moving from index q to $q+1$ the work done to get to level q is not lost. Rewriting the Smolyak formula in a recursive fashion is advantageous for implementation on a computer.

$$\begin{aligned} A(q,d) &= A(q-1,d) + \Delta A(q,d) && (2.34) \\ \Delta A(q,d) &= \sum_{|\mathbf{i}|=q} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d}\right) && (2.35) \end{aligned}$$

While the Smolyak algorithm representation in 2.34 has the advantage of being represented recursively, it does not provide any type of indicator for when the Smolyak sparse grid should be refined. Collocation points should be added to a Smolyak sparse grid until the resulting interpolant is able to reproduce the objective function to some user-defined threshold. The authors in [49] are able to rewrite 2.35 in terms of what's referred to as a hierarchical surplus,

$$\Delta A(q,d) = \sum_{|\mathbf{i}|=q} \left(f(x_{j_1}^{i_1},\ldots,x_{j_d}^{i_d}) - A(q-1,d)(x_{j_1}^{i_1},\ldots,x_{j_d}^{i_d})\right) \cdot \left(a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_d}^{i_d}\right) \tag{2.36}$$

which appears as the first term in the summation as the difference between the function value at the point $(x_{j_1}^{i_1},\ldots,x_{j_d}^{i_d})$ and the Smolyak $q-1$ level interpolant value at the same point. Level q of the Smolyak algorithm generally contains all the points comprising level $q-1$ plus some new collocation points. Consequently, the level q Smolyak interpolant is expected to exactly evaluate any collocation points born in previous levels. The summation in 2.36 is taken over all the new points in level q that have not appeared in level $q-1$ since the hierarchical surplus for these will be identically equal to zero. The hierarchical surpluses provide an indicator for how well the Smolyak algorithm is interpolating some objective function. If the hierarchical surpluses are decreasing with each successive level then the Smolyak algorithm is converging.

Following the notation in [6], let $X^i = \{x_1^i,\ldots,x_{m_i}^i\}$ be the collocation points comprising U^i. From 2.31, the total number of collocation points in a Smolyak sparse grid can be written as,

$$H(q,d) = \bigcup_{q-d+1 \leq |\mathbf{i}| \leq q} \left(X^{i_1} \times \cdots \times X^{i_d} \right). \tag{2.37}$$

2.4.3 Basis and Collocation Points

The exactness of the Smolyak algorithm is decided mainly by the choice of collocation points $x^{i_k}_{j_k}$ used to build $H(q,d)$. The basis functions $a^{i_k}_{j_k}$ work to weave the collocation points together. Gaussian quadrature is a favorite of many since with only $n+1$ collocation points, all polynomials of degree $2n+1$ or less can be integrated exactly [29]. However, collocation points derived from Gaussian quadrature schemes are not nested in that $X^i \not\subset X^{i+1}$. Nestedness in the choice of collocation points is an essential feature for reducing the computational expense of applying the Smolyak algorithm. If nested collocation points are chosen for each X^{i_k} then the Smolyak sparse grid will also be nested such that $H(q-1,d) \subset H(q,d)$ [6]. Consequently, when improving the Smolyak interpolant from level $q-1$ to level q one will only have to evaluate the objective function at the points that are unique to X^i, which are given as $X^i_\Delta = X^i \setminus X^{i-1}$ [49]. The set of new points in level q of a Smolyak sparse grid are given as,

$$\Delta H(q,d) = \bigcup_{|\mathbf{i}|=q} X^{i_1}_\Delta \times \cdots \times X^{i_d}_\Delta. \tag{2.38}$$

A viable alternative to Gaussian quadrature collocation points for the Smolyak algorithm is to use Clenshaw-Curtis collocation points, which consist of the extrema of Chebyshev polynomials. While $n+1$ Clenshaw-Curtis abscissas can only exactly integrate polynomials of degree n, they have the advantage of being nested. Accuracy is sacrificed for nestedness in the Smolyak algorithm, at least in theory. In practice it has been shown that for most functions Clenshaw-Curtis quadrature performs almost on par to Gaussian quadrature [72]. In other words, the double accuracy of Gaussian quadrature is rarely realized. For some level i the Clenshaw-Curtis collocation points are given by,

$$x^i_j = \begin{cases} \cos\frac{\pi(j-1)}{m_i-1} & j = 1,\ldots,m_i \text{ if } i > 1 \\ 0 & j = 1 \text{ if } i = 1 \end{cases} \tag{2.39}$$

In order for the level i Clenshaw-Curtis abscissas to contain the level $i-1$ abscissas, a total of 2^{i-1} new points must be added. Consequently, the total number of abscissas appearing in the level i Clenshaw-Curtis scheme is given as,

$$m_i = \begin{cases} 2^{i-1}+1 & i > 1 \\ 1 & i = 1 \end{cases} \tag{2.40}$$

Another alternative to the Gaussian and Clenshaw-Curtis abscissas is Gauss-Patterson. The Gauss-Patterson set of collocation points are nested and provide a polynomial exactness of $(3n-1)/2$ with n points, which is right in between the exactness of Clenshaw-Curtis and Gaussian sets. Obtaining the Gauss-Patterson abscissas involves a rather convoluted, iterative process and so the reader is referred to [15] to review the methodology and obtain tables of the actual points. The growth rule for Gauss-Patterson goes as $2^i - 1$, which is some factor of two greater than the growth rule for Clenshaw-Curtis. In [48], the authors conclude the Gauss-Patterson collocation points are competitive with Clenshaw-Curtis when comparing the cost and accuracy of computing quadratures using the same number of function evaluations.

To weave together the collocation points forming a Smolyak sparse grid, some type of basis function a_j^i is needed, as defined in 2.28. Although the basis functions will be applied to multi-dimensional interpolation, the Smolyak algorithm conveniently scales 1D basis functions to multiple dimensions through the use of tensor products. One basis commonly used in adaptive sparse grids is the linear hat basis function [3]. For the scheme in 2.40 the linear hat functions are given as,

$$\begin{aligned} a_1^1 &= 1 \text{ for } i = 1, \\ a_j^i &= \begin{cases} 1-(m_i-1)|x-x_j^i| & \text{if } |x-x_j^i| < 1/(m_i-1) \\ 0 & \text{else} \end{cases} \end{aligned} \tag{2.41}$$

for $i > 1$ and $j = 1, ..., m_i$. While the linear hat functions have the advantage of local support they are limited to relatively slow convergence due to their lack of curvature. Offering faster error decay are the global Lagrange characteristic polynomials,

$$a_j^i = \begin{cases} 1 & \text{if } i = 1 \\ \displaystyle\prod_{\substack{k=1 \\ k\neq j}}^{m_i} \frac{x-x_k^i}{x_j^i - x_k^i} & j = 1, ..., m_i \text{ for } i > 1 \end{cases} \tag{2.42}$$

However, the Lagrange characteristic polynomials are plagued by the fact that each evaluation of 2.28 requires $\mathscr{O}(m_i^2)$ operations and often the computation is numerically unstable [7]. To remedy these concerns, the barycentric form of Lagrange characteristic polynomials is used to form a basis. The barycentric Lagrange basis is given as,

$$a_j^i = \begin{cases} 1 & \text{if } i = 1 \\ \dfrac{\frac{w_j^i}{x - x_j^i}}{\sum_{j=0}^{m_i} \frac{w_j^i}{x - x_j^i}} & j = 1, \ldots, m_i \text{ for } i > 1 \end{cases} \tag{2.43}$$

where w_j^i are barycentric weights defined by,

$$w_j^i = \frac{1}{\prod\limits_{k \neq j} \left(x_j^i - x_k^i\right)} \qquad j = 1, \ldots, m_i. \tag{2.44}$$

For special collocation sets, such as Clenshaw-Curtis in 2.39, explicit forms exist for the barycentric weights. Generally, forming the weights is an $\mathscr{O}(m_i^2)$ operation and then evaluation of an interpolant based on the barycentric Lagrange basis is only a $\mathscr{O}(m_i)$ operation [7]. With an explicit form in hand, evaluation of the barycentric Lagrange basis is significantly cheaper than the Lagrange basis. For the Clenshaw-Curtis collocation points in 2.39, the barycentric weights are given by [65],

$$w_j^i = (-1)^{j+1} \delta_j^i \qquad \delta_j^i = \begin{cases} .5 & j = 1 \text{ or } j = m_i \\ 1 & \text{else} \end{cases}. \tag{2.45}$$

From 2.43, an apparent problem exists if the barycentric basis is to be evaluated at a collocation point. As [7] explains, the problem can be circumvented by simply perturbing the value of x by an ε on the order of machine precision. In this case, the numerator and denominator in 2.43 will effectively cancel each other such that $a_j^i = 1$. The barycentric Lagrange basis is therefore numerically stable.

2.4.4 Exactness and Error Bounds

The exactness of the Smolyak algorithm is determined by the space of polynomials the algorithm is exact on. Since the 1D interpolation rules, on which the Smolyak algorithm is based on, can exactly interpolate certain polynomials it is not presumptuous to expect the Smolyak algorithm to exactly interpolate certain polynomial spaces. Using the collocation set in 2.40 and 2.39 the Smolyak interpolant $A(q,d)$ is exact on [6],

$$\sum_{|\mathbf{i}|=q} \mathbb{P}(m_{i_1} - 1, 1) \otimes \cdots \otimes \mathbb{P}(m_{i_d} - 1, 1) \tag{2.46}$$

where $\mathbb{P}(k,d)$ is the space of all polynomials in d dimensions of total degree no greater than k. From 2.46 it follows that the Smolyak interpolant for $q = d + P$ is exact for all polynomials of degree P. In other words, the effects of any monomials containing x^l for $l \leq P$ will be captured by the Smolyak algorithm. Recall from 2.27 that the degrees of freedom of $\mathbb{P}(P,d)$ goes as $d^P/P!$. Any method aiming to reproduce $\mathbb{P}$ requires at least this many function evaluations. Since the number of collocation points in a Smolyak grid for $A(d+P,d)$ goes as $2^P d^P/P!$ the dependence on dimension is said to be optimal [6]. However, the asymptotic growth of points also indicates that the Smolyak algorithm requires still excessive function evaluations to achieve polynomial exactness.

Since the Smolyak algorithm is constructed using one-dimensional interpolation formulas, which all have error bounds, it is also possible to derive error bounds for a Smolyak interpolant $A(q = d+P,d)$. While the reader is instructed to consult [6] for a detailed derivation of the error bounds, they will nevertheless be stated here. Consider some d-variable function f with continuous derivatives of order P in each variable. The error in using a Smolyak interpolant to approximate f can be given as,

$$\|f - A(d+P,d)(f)\|_\infty \leq c_{d,P} M^{-P} (\log M)^{(P+2)(d-1)+1} \tag{2.47}$$

where M is the total number of knots used by $A(d+P,d)$ and $c_{d,P}$ is a constant depending on both d and P. From 2.47, the error in the Smolyak interpolant heavily depends on the smoothness of the function being interpolated and on the total number of collocation points used to form the interpolant.

2.4.5 Computer Implementation

To implement Smolyak's algorithm on a computer equations 2.34 and 2.36 should be utilized since together they provide a recursive definition. Much of the implementation efforts are concerned with indice book keeping. Although the pseudocode for Smolyak's algorithm used in this thesis is provided here, the reader is directed to [44] for more elaborate details. Efficiency of the algorithm can be increased by pre-calculating the desired abscissas as in 2.39, the number of abscissas at a given level as in 2.40, and corresponding barycentric weights as in 2.45. A data structure for quick retrieval of the desired values is also necessary. Abscissa information is constantly being reused in Smolyak's algorithm and so it is inefficient to have to recalculate values each time.

The pseudocode for Smolyak's algorithm in algorithm 4 will now be discussed in some detail. To initialize the algorithm a data structure must be created to store all information for each index in the sparse grid. For level q of the Smolyak algorithm the summation in

Algorithm 4 Smolyak's algorithm for creating an interpolant for a function f of d dimensions. The algorithm will exit if the maximum Smolyak level is reached or if one of the convergence criteria is met.

```
Create data structure that stores indice coordinates and hierarchical surplus.
for $q = d,$ maximum level do
    for $(i_1,...,i_d)$ in enumerations of $i_1 + ... + i_d = q$ do        ▷ See Alg. 5
        for $(j_1,...,j_d)$ in enumerations of $(i_1,...,i_d)$ do          ▷ See Alg. 6
            Turn each $(j_1,...,j_d)$ into a knot.
            Categorize as either processed or unprocessed knot.
        end for
        Evaluate $f$(unprocessed knots).
        Calculate hierarchical surplus at unprocessed knots.               ▷ Eq. 2.36
        Archive newly processed knots.
    end for
    Check for convergence.
end for
```

2.34 is over all sets $(i_1, i_2, ..., i_d)$ such that $i_1 + i_2 + ... + i_d = q$. Each such set corresponds to a knot $(x^{i_1}_{j_{i_1}}, ..., x^{i_d}_{j_{i_d}})$ in the random space defined by the hypercube $[-1,1]^d$. The knot, function value at the knot, and the corresponding hierarchical surplus should all be stored in the data structure.

The first loop in the pseudocode tells the code to keep increasing the interpolation level in the Smolyak algorithm until some maximum level is reached, which is specified by the user. The purpose of this loop is to make sure the algorithm ends eventually. Of course, other convergence criteria are in place in hope that the algorithm terminates long before the maximum level is reached. The second loop goes through all the enumerations of $(i_1, i_2, ..., i_d)$ such that $i_1 + i_2 + ... + i_d = q$. The algorithm for producing such enumerations is provided in algorithm 5. The third loop takes each enumeration and again enumerates over each index to obtain each component in the tensor product appearing in the Smolyak formulation. An algorithm to execute this enumeration is provided in algorithm 6.

In the main body of the pseudocode each output from algorithm 6 is first converted to a knot $(x^{i_1}_{j_{i_1}}, ..., x^{i_d}_{j_{i_d}})$. Each potential knot must then be sorted into one of two categories. The motivation for the two categories arises from the fact that the same knot may be expressed in several ways. Since each knot corresponds to a function value and hierarchical surplus, significant computational savings can be incurred by not reevaluating the objective function at the same knots. Consequently, each potential knot is binned into either a category of knots that have already been evaluated at f or a category of unevaluated knots.

Once all components in the tensor product for a given $(i_1, i_2, ..., i_d)$ have been converted and sorted, the unevaluated knots are processed. In this step of the algorithm the previously unevaluated knots should be evaluated at f in parallel if possible since each evaluation is completely independent. The resulting functions values should then be used to compute the hierarchical surplus in 2.36 for each knot. Once the function value and hierarchical surplus is available for each new knot the results should be archived in the indice data structure.

Finally, once the second loop is complete, the level of the Smolyak interpolant has been effectively increased and it's time to check whether additional levels are required based on user-defined convergence criteria. Perhaps the best indicator of a Smolyak interpolant's convergence is the maximum hierarchical surplus calculated for all newly processed knots at the current interpolation level. The hierarchical surplus is a measure of how well the interpolant is able to match the objective function and therefore, if the hierarchical surpluses being calculated are decreasing with each level the interpolant is converging. An additional convergence criteria includes comparison of the relative change in computed mean and variance between two successive interpolation levels. For this thesis, the Smolyak algorithm is terminated only after the maximum hierarchical surplus is below a certain threshold, the relative change in interpolant mean is below a threshold, and the relative change in variance does not exceed a threshold.

2.5 Combining Decomposition and Smolyak's Algorithm

As hinted at in the beginning of chapter 2, the Smolyak algorithm combines with the anchored-ANOVA decomposition to create a reduced order model for any well behaving computer code. To see how the Smolyak algorithm fits into the functional decomposition described in this chapter, begin by substituting 2.18 into 2.16 to get,

$$f(\mathbf{x}) = \sum_{\mathbf{u} \subseteq \mathscr{D}} \sum_{\mathbf{v} \subseteq \mathbf{u}} (-1)^{|\mathbf{u}|-|\mathbf{v}|} P_{\mathbf{v}} f(\mathbf{x}_{\mathbf{v}}). \tag{2.48}$$

Now, insert the Dirac projection operator from 2.19 into 2.48 to arrive at,

$$f(\mathbf{x}) = \sum_{\mathbf{u} \subseteq \mathscr{D}} \sum_{\mathbf{v} \subseteq \mathbf{u}} (-1)^{|\mathbf{u}|-|\mathbf{v}|} f(\mathbf{x})|_{\mathbf{x}=\mathbf{a} \backslash \mathbf{x}_{\mathbf{v}}}. \tag{2.49}$$

To create a reduced order model the set $\mathscr{D}$ is ultimately shrunk to only contain a subset of all the variables of f but this is discussed later. The important aspect of 2.49 to realize is that in order to evaluate $f(\mathbf{x})|_{\mathbf{x}=\mathbf{a} \backslash \mathbf{x}_{\mathbf{v}}}$ evaluation of the expensive computer code $f(\mathbf{x})$ is required. Consequently, the desirable property of reduced order models, that of rapid evaluation, is not achieved in 2.49. The remedy is to approximate each $f(\mathbf{x})|_{\mathbf{x}=\mathbf{a} \backslash \mathbf{x}_{\mathbf{v}}}$ using Smolyak interpolants. Substituting 2.33 into 2.49, the formulation for creating a reduced order model of f is complete.

$$f(\mathbf{x}) = \sum_{\mathbf{u}\subseteq\mathscr{D}} \sum_{\mathbf{v}\subseteq\mathbf{u}} (-1)^{|\mathbf{u}|-|\mathbf{v}|} \sum_{|\mathbf{i}|\leq q} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_{|\mathbf{v}|}}\right) \tag{2.50}$$

While there is initial overhead to create an interpolant for each component in 2.49 the result is quick evaluation of the reduced order model. Details regarding the implementation and application of 2.50 will be discussed in the proceeding sections.

2.5.1 Combinatorics Routines

In order to implement 2.50 on a computer several enumeration routines need to be available. Unfortunately, these routines are not available in most numerical math libraries containing combinatorics routines. The first routine of interest solves the problem of how to enumerate all the ways d positive integers can be summed to equal another integer. In other words, what are all the sets $\{i_1,...,i_d\}$ such that $i_1+i_2+...+i_d = q$? This problem inserts itself in 2.50 in the summation index for Smolyak interpolation. As the Smolyak interpolant becomes refined from level to level—q is increased by one in each refinement—the Smolyak algorithm must newly account for all $\mathbf{i}$ such that $|\mathbf{i}| = q$. The following enumeration algorithm, a slight modification of the original algorithm found in [30], finds all the desired indice sets:

With all the index sets for some Smolyak level q available through the code segment in 5, the tensor product appearing in 2.33 can be evaluated with the aid of an additional enumeration routine. Each indice in an index set $\{i_1+i_2+...+i_d\}$ corresponds to certain number of knots, which for example, is given by 2.40 for Clenshaw-Curtis. All the components in the tensor product can be given by the following enumeration algorithm, which is based on the algorithm in [30]. Input to algorithm 6 should be based on output from algorithm 5. Specifically for some index set $\{i_1,i_2,...,i_d\}$ returned by algorithm 5, each indice should be converted to a corresponding number of knots and input to algorithm 6.

2.5.2 Sampling Sparse Grid Interpolant of Correlated Variables

The reduced order model in 2.50 consists of linear combinations of Smolyak interpolants. Evaluation of the reduced order model is equivalent to finding the value at several Smolyak interpolants and summing the results. However, recall that in the Smolyak algorithm for building the interpolants described in 2.4.2 each dimension in the sparse grid is built orthogonal to the others. Implicit in this construction is the assumption that the random variables comprising the objective function are independent. In fact, in many computer codes the

Algorithm 5 For positive integers d and q this code outputs all sets $\{i_1, i_2, ..., i_d\}$ such that $i_1 + i_2 + ... + i_d = q$.

```
$p = 0$
$m = q - d + 1$
$k = [0, 1, ..., 1]$                                        ▷ vector of length d
$\hat{k} = [m, m, ..., m]$                                  ▷ vector of length d
repeat
    $k(p) = k(p) + 1$
    if $k(p) > \hat{k}(p)$ then
        if p=d then
            All indices enumerated!
        else
            $k(p) = 1$
            $p = p + 1$
        end if
    else
        for $j = 0 : p$ do
            $\hat{k}(j) = \hat{k}(p) - k(p) + 1$
        end for
        $k(0) = \hat{k}(0)$
        $p = 1$
        Return valid index set k!
    end if
until $k = [0, ..., 0]$
```

Algorithm 6 Code for enumerating all components of a tensor product. The input is a d dimensional vector m where each entry m_j corresponds to the number of knots in a collocation scheme of level i_j.

```
$p = 0$
$s = [0, 1, 1, ..., 1]$                                     ▷ vector of length d
repeat
    $s(p) = s(p) + 1$
    if $s(p) > m(p)$ then
        if $p = d - 1$ then
            All indices enumerated!
        else
            $s(p) = 1$
            $p = p + 1$
        end if
    else
        $p = 0$
        Return valid enumeration set!
    end if
until $s = m$
```

random variables forming the parameter space are correlated. The degree of correlation among the random variables is generally described using a covariance matrix.

With the availability of a covariance matrix for the input random variables it is possible to sample the reduced order model hundreds or thousands of times to get accurate and precise statistical moments. Evaluating a Smolyak interpolant is relatively cheap and fast so this is not a computational problem. To produce correlated random variables from a Gaussian population the Kaiser-Dichman method can be applied [39]. Say some inputs to a computer code are normally distributed with mean μ and covariance Σ. To produce a random sample μ' from Σ one can apply,

$$\mu' = \mu + U^T \pi \tag{2.51}$$

where U^T is the lower triangular matrix arising from the Cholesky decomposition of Σ and π is a standard normal random vector.

The Cholesky decomposition of a covariance matrix $C = U^T U$ is the equivalent to taking the square root of a matrix. A Cholesky matrix transform, or left multiplication by U^T, maps uncorrelated variables into correlated variables with covariance matrix Σ. Consequently, in 2.51 μ' is normally distributed with mean μ and covariance Σ. A statistically significant number of instances of μ' should be calculated and evaluated at the reduced order model. Even though the model is built assuming independent variables this method takes into account any correlations when calculating statistical moments. When evaluating any Smolyak interpolant at a number of sampled points one must always make sure all the sampled points lay inside the bounds of the hypercube forming the sparse grid.

2.5.3 Dimension Truncation

By definition a reduced order model contains less dimensions than the original model whose reduction is intended. Consequently, a methodology for identifying important dimensions is necessary. For the purposes of this thesis, the importance of a random input variable on some objective function is determined by its contribution to the function's variance. While there are methods for exactly calculating the truncation and superposition dimensions of a function [30], these methods require the computation of all 2^d component functions in an ANOVA decomposition. For typical computer codes used in engineering this requirement is not feasible and so the effective dimensions of a function must be estimated adaptively.

To this end, construction of a reduced order model based on the anchored-ANOVA decomposition begins with calculation of the zeroth-order and all first-order components.

The zeroth-order component function is simply the function evaluated at the anchor point,

$$f_{\{\emptyset\}} = f(\bar{\mathbf{x}}). \tag{2.52}$$

From the recursive definition of the anchored-ANOVA decomposition in 2.17, the i^{th} component function is given as,

$$f_{\{i\}} = f(\mathbf{x})|_{\mathbf{x}\backslash x_i} - f_{\{\emptyset\}}. \tag{2.53}$$

The first order components solely measure the affect of the i^{th} random variable on the function output. Therefore, for each of d random variables contributing to a function's variability a sensitivity coefficient can be calculated as [50],

$$\eta_i = \frac{\int_{\Omega_i} \left[f(\mathbf{x})|_{\mathbf{x}=\bar{\mathbf{x}}\backslash x_i} - f(\mathbf{x}) \right] \rho(x_i) dx_i}{f_{\{\emptyset\}}} \tag{2.54}$$

where $\rho(x_i)$ is the probability density of x_i. If f is a function of spatial coordinates then the L_2 norm can be applied to 2.54. The greater the affect of the i^{th} random variable on the function output, the greater the sensitivity coefficient.

Once all the first order anchored-ANOVA components are calculated the sensitivity coefficients in 2.54 can be used to identify the important dimensions. To start, each η_i should be normalized by the sum of all η_i. The important dimensions can then be obtained by taking all i variables such that $\eta_i > \theta_1$. A value of $\theta_1 = .02$ has been shown to be effective for many problems [23]. At the very least, the reduced order model described here will consists of the zeroth order and all first order anchored-ANOVA components, giving it a superposition dimension of one. The variance and mean of the reduced order model with superposition dimension one should be obtained through sampling using 2.51.

To improve accuracy higher order components can be included. Higher order components should only be built using combinations of the dimensions deemed important. If all dimensions were to be included construction of a reduced order model based on the ANOVA decomposition would quickly become intractable for most engineering problems. It can be argued that if a set of random variables independently affect a function then their interactions will likely also affect the function. Say all first order components have been constructed and the important dimensions have been added to a set $\mathscr{T}$. The next step in creating a higher order model is to construct anchored-ANOVA components utilizing dimensions t such that $t \subset \mathscr{T}$ and $|t| = 2$. In total there should be $C(|\mathscr{T}|, 2)$ such components.

In general, the number of p order components can be given as,

$$C(|\mathscr{T}|,p) = \frac{|\mathscr{T}|!}{(p!)(|\mathscr{T}|-p)!}. \tag{2.55}$$

To determine whether additional orders need to be included in the reduced order model, the mean of the current reduced order model should be compared to the mean of the previous order model. If the relative change in mean doesn't exceed some threshold θ_2 then the reduced order model is considered converged. Otherwise, higher order components should be added. In this manner, a reduced order model for a computer code can be constructed adaptively. The procedure for adaptively creating a reduced order model is summarized in algorithm 7.

Algorithm 7 Adaptively creates a reduced order model for some function f of d dimensions.

```
Create zeroth-order component function.                              ▷ Eq. 2.52
for $i = 1, d$ do
    Construct 1D Smolyak interpolant for $P_{\{i\}} f$.               ▷ Eq. 2.19
    Create first-order anchored-ANOVA component $f_{\{i\}}$.          ▷ Eq. 2.53
end for
Identify important dimensions and put into set $\mathscr{T}$.          ▷ Eq. 2.54
for $p = 2, d$ do
    for $t \subset \mathscr{T}$ s.t. $|t| = p$ do
        Construct $p$-dimensional Smolyak interpolant for $P_t f$.   ▷ Eq. 2.19
        Create $p$-order anchored-ANOVA component $f_t$.             ▷ Eq. 2.17
    end for
    Check for convergence of reduced order model.
end for
```

CHAPTER 3

Application to Simplified Reactor Systems

The methodologies described in Chapter 2 will initially be applied to three relatively simple problems in reactor physics, each building on its predecessors in complexity. The purpose of the simple problems is to develop some intuition about the workings of the described reduced order model algorithms. Ultimately, the methods will be applied to solve a state of the art problem in reactor uncertainty quantification and sensitivity analysis.

3.1 Infinite Lattice Multiplication Factor

3.1.1 Problem Statement

In an infinite lattice a reactor of infinite size is considered and therefore neutrons are not capable of leaking out of the system. An infinite lattice effectively removes the effects of geometry in neutron transport and characterizes the system entirely in terms of its material properties. Since the physics of an infinite lattice are greatly simplified an analytic analysis of the system is possible. Consequently, the infinite lattice problem is ideal for an initial analysis of any new computational method.

To begin, the mathematical formulation of an infinite lattice will be described. In matrix form the two-group neutron balance equations for an infinite lattice can be written as,

$$\begin{pmatrix} \Sigma_{a_1}+\Sigma_{1\to 2} & 0 \\ -\Sigma_{12} & \Sigma_{a_2} \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix} = \frac{1}{k_\infty} \begin{pmatrix} \nu\Sigma_{f_1} & \nu\Sigma_{f_2} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix}. \tag{3.1}$$

Solving the system in 3.1 for the infinite multiplication factor, the following analytic expression is obtained,

$$k = \frac{\Sigma_{a_2}\nu\Sigma_{f_1}+\Sigma_{1\to 2}\nu\Sigma_{f_2}}{\Sigma_{a_2}\left(\Sigma_{a_1}+\Sigma_{1\to 2}\right)}. \tag{3.2}$$

The infinite multiplication factor is a function of five material parameters. Since this thesis is concerned with the affect of uncertainties in input parameters on computer code outputs, variations in k_∞ as a function its stochastic input variables are of interest. Assume all variation in k_∞ can be attributed to its input cross sections, whose distributions follow a multivariate Gaussian. To obtain physical homogenized, two-group cross section values a real system must first be modeled in a transport code. The UAM Benchmark is sought for this purpose [34]. Specifically, the TMI assembly is modeled using the two-step method described in [84]. A total of 300 perturbed cross section sets were produced to obtain the few-group mean and covariance data used in the proceeding analysis. The cross section data is summarized in Table 3.1.

Table 3.1: Two-group cross section data for an infinite TMI lattice.

			Correlation Coefficient Matrix				
	Mean	**Standard Dev.**	Σ_{a_1}	Σ_{a_2}	$\nu\Sigma_{f_1}$	$\nu\Sigma_{f_2}$	$\Sigma_{1\to2}$
Σ_{a_1}	1.04E-02	9.06E-05	1	0.07	-0.13	0.02	0.75
Σ_{a_2}	1.10E-01	2.31E-04	0.07	1	0.06	0.31	-0.07
$\nu\Sigma_{f_1}$	9.00E-03	4.85E-05	-0.13	0.06	1	0.33	-0.10
$\nu\Sigma_{f_2}$	1.91E-01	8.87E-04	0.02	0.31	0.33	1	0.01
$\Sigma_{1\to2}$	1.80E-02	2.18E-04	0.75	-0.07	-0.10	0.01	1

Multiple methods will be applied to obtain basic statistical and sensitivity data on the infinite multiplication factor. Of course, Monte Carlo sampling using the input cross sections' covariance matrix and applying 2.51 will provide the mean and variance of k_∞. Another approach to get at the variance of k_∞ is through the "Sandwich Equation" [36],

$$\sigma^2(k_\infty) = S^T C S \tag{3.3}$$

where C is the covariance matrix for the input data. In equation 3.3 the array S contains sensitivities of the output to the input parameters. For this problem the vector S contains,

$$S^T = \left(\begin{array}{ccccc} \frac{\partial k_\infty}{\partial \Sigma_{a_1}} & \frac{\partial k_\infty}{\partial \Sigma_{a_2}} & \frac{\partial k_\infty}{\partial \nu\Sigma_{f_1}} & \frac{\partial k_\infty}{\partial \nu\Sigma_{f_2}} & \frac{\partial k_\infty}{\partial \Sigma_{1\to2}} \end{array} \right). \tag{3.4}$$

Since there exists an analytic expression for k_∞ the sensitivity vector for this problem in 3.4 is exact. As a side-check the sensitivity vector S can also be constructed using central differencing. The sensitivity of k_∞ to the i^{th} cross section Σ_i using central differencing is expressed as,

$$\left.\frac{\partial k_\infty}{\partial \Sigma_i}\right|_{\Sigma_{j\neq i}=\bar{\Sigma}_j} \approx \frac{k_\infty(\Sigma_i+\Delta\Sigma_i) - k_\infty(\Sigma_i-\Delta\Sigma_i)}{2\Delta\Sigma_i} \tag{3.5}$$

where all cross sections Σ_j, $j \neq i$, are held at their mean values. Generally a one percent perturbation $\Delta\Sigma$ is sufficient to obtain accurate sensitivities although this rule of thumb is dependent on the smoothness of the objective function. With a variety of methods available to obtain sensitivities and statistical moments of k_∞ it is possible to thoroughly assess the potential of a reduced order model.

3.1.2 Analysis

Several elements of the methodologies described in Chapter 2 will be tested in this section and compared to results obtained using analytic and Monte Carlo approaches. For all Smolyak sparse grids constructed the hypercube domain extends to six standard deviations in each random variable. Since k_∞ is a function of only five random variables a sparse grid interpolant will be constructed without applying any function decomposition in order to demonstrate the accuracy and convergence of the method. The convergence criteria for the sparse grid interpolant is set such that the maximum hierarchical surplus at a given level is not to exceed 10^{-10}. Both Clenshaw-Curtis and Gauss-Patterson abscissas are tested.

From Fig. 3.1 the Clenshaw-Curtis and Gauss-Patterson schemes perform similarly in terms of level to level convergence. However, observe that at each interpolation level the Gauss-Patterson scheme requires significantly more nodes in exchange for a small increase in convergence speed. Both schemes converge to the threshold around level five although the Gauss-Patterson scheme requires more than twice as many function evaluations to get there than Clenshaw-Curtis. Based on the graphical determination of order of convergence [11], it is clear from 3.1 that the Smolyak interpolant for k_∞ converges geometrically.

With the Smolyak interpolation routines working as expected it's safe to apply them to an anchored-ANOVA decomposition of k_∞. To start, only the first order components will be built and analyzed. The first order components are relatively cheap to produce and often collectively produce very accurate reduced order models [50]. Afterwards, all higher order components will be added in order to show that the full anchored-ANOVA decomposition can fully reproduce the objective function. Since k_∞ is a function of only five random variables there is no point in adaptively constructing the reduced order model as described in Section 2.5.3.

As a first comparison between all the models developed for quantifying the uncertainty in k_∞ the mean and variance values of each model will be compared. With the exception of the variance obtained using the Sandwich Equation, each model's variance was obtained by propagating 1000 samples of Eq. 2.51 through the model. All samples produced for each model were seeded identically and so the same random numbers were drawn. The mean and variance results, along with 99% confidence intervals, are summarized in Table 3.2.

The five dimensional sparse grid interpolant results are entirely self consistent with the

Figure 3.1: Convergence study of a five dimensional sparse grid interpolant for the multiplication factor of an infinite TMI lattice. The boxed numbers represent the current number of knots in the sparse grid.

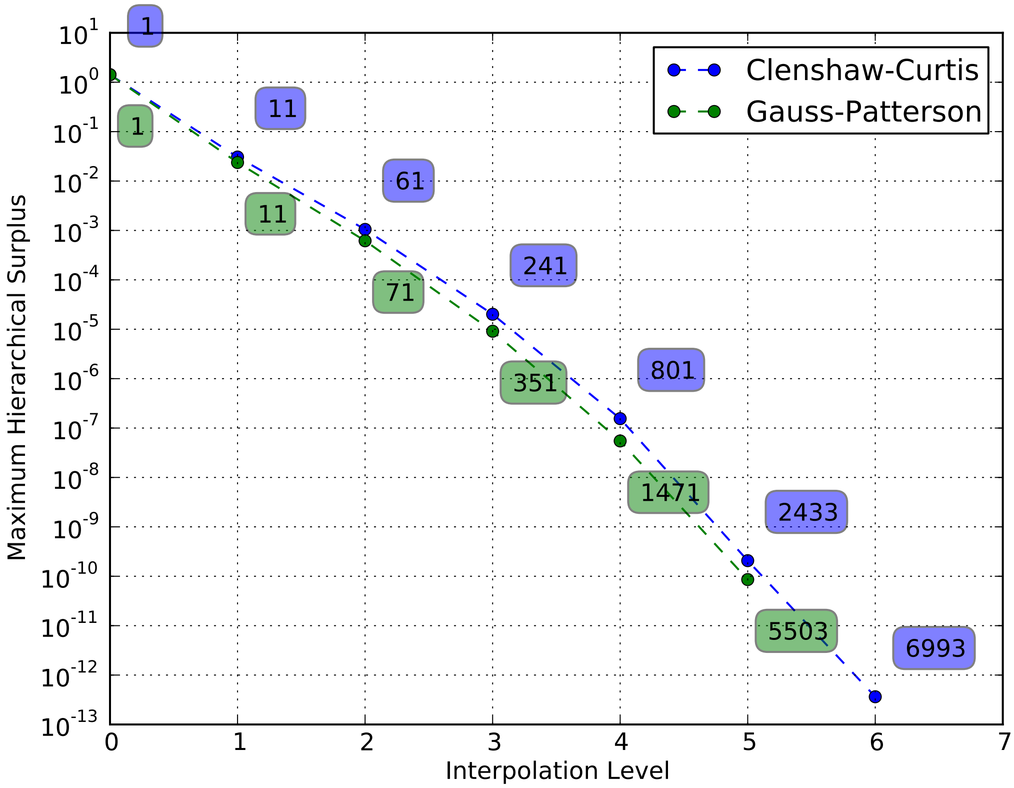
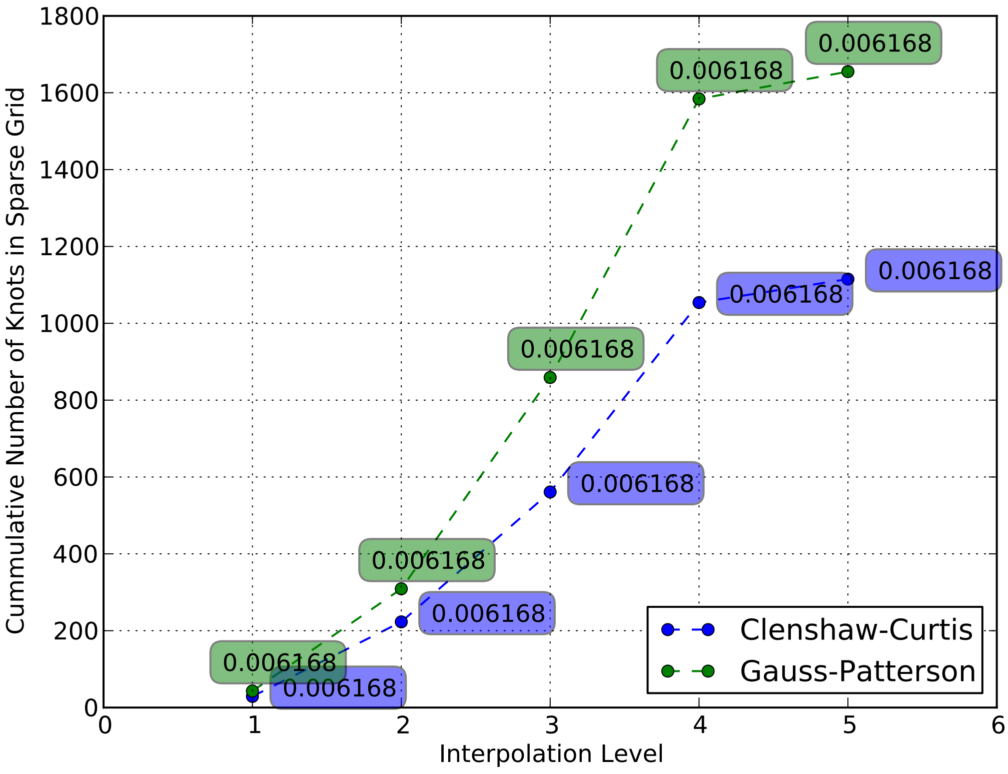


Table 3.2: Mean and variance data for the multiplication factor of an infinite TMI lattice obtained using Monte Carlo sampling. Wherever sampling was utilized the same random numbers were used.

Method	Mean	99% CI	Standard Dev.	99% CI
5D Sparse Grid CC	1.41562	(1.41512, 1.41612)	0.006168	(0.005909, 0.006544)
5D Sparse Grid GP	1.41562	(1.41512, 1.41612)	0.006168	(0.005831, 0.006544)
1D ANOVA CC	1.41560	(1.41510, 1.41610)	0.006168	(0.005831, 0.006544)
All ANOVA CC	1.41562	(1.41512, 1.41612)	0.006168	(0.005831, 0.006544)
1D ANOVA GP	1.41560	(1.41510, 1.41610)	0.006168	(0.005831, 0.006544)
All ANOVA GP	1.41562	(1.41512, 1.41612)	0.006168	(0.005831, 0.006544)
True Function	1.41562	(1.41512, 1.41612)	0.006168	(0.005831, 0.006544)
Sandwich			0.006540	

Figure 3.2: Cumulative number of knots required at each level of an anchored-ANOVA decomposition of the multiplication factor of an infinite TMI lattice. Boxes contain the calculated standard deviation at the current level.



anchored-ANOVA results. Further, both of these methods produce identical results to those obtained using Monte Carlo sampling. Although the analytic variance from the Sandwich Equation is within the 99% confidence bounds of each model's results, there is a notable difference due to the fact that only 1000 samples were used to obtain each model's statistics. Increasing the number of samples decreased the difference. Note that in Table 3.2 the anchored-ANOVA the reduced order models consisting of only one dimension anchored-ANOVA components perform just as well as the full decomposition and the sparse grid interpolants over all five random variables. However, the 1D component models require only 29 function evaluations to produce, which is some ten times fewer evaluations than the 5D interpolants, and some hundred times fewer evaluations than the full decomposition. The rapid convergence of the reduced order model containing only one dimensional anchored-ANOVA components is shown in 3.2. Construction of higher order components is very expensive. Fortunately for this problem, and perhaps others, construction of only

Table 3.3: Normalized sensitivity coefficients for the multiplication factor of an infinite TMI lattice.

	Normalized Sensitivity Coefficient of k_∞				
Method	Σ_{a_1}	Σ_{a_2}	$\nu\Sigma_{f_1}$	$\nu\Sigma_{f_2}$	$\Sigma_{1\to 2}$
5D Sparse Grid CC	-.367551	-.776087	.224060	.776010	.143491
5D Sparse Grid GP	-.367551	-.776087	.224060	.776010	.143491
1D ANOVA CC	-.367556	-.776098	.224063	.776020	.143493
All ANOVA CC	-.367551	-.776087	.224060	.776010	.143491
1D ANOVA GP	-.367556	-.776098	.224063	.776020	.143493
All ANOVA GP	-.367551	-.776087	.224060	.776010	.143491
Analytic	-.367520	-.775956	.224044	.775956	.143476
Central Difference	-.367551	-.776089	.224060	.776011	.143492

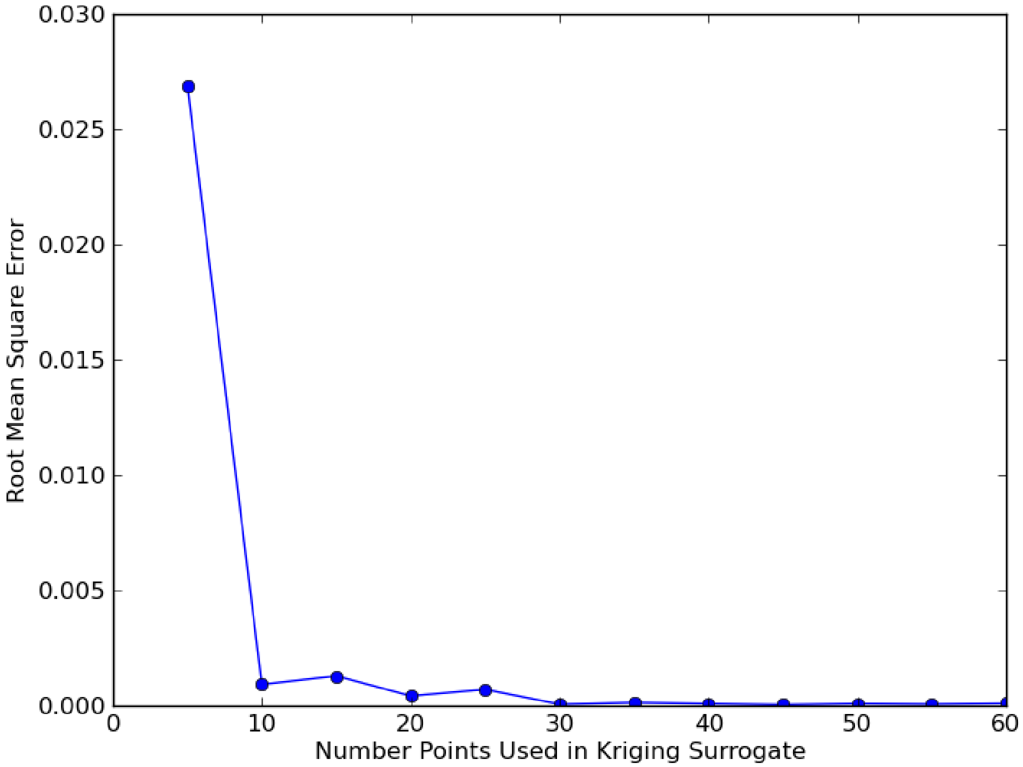
one dimensional components is completely sufficient to represent the objective function.

As another performance measure of the reduced order model methodologies, each model is used to obtain normalized sensitivity coefficients for k_∞. Central differencing is applied to each model, with perturbations made to each cross section at a time while holding the other cross sections at their mean values. Perturbations are taken to be 1% of each cross section's value. Using the analytic expression for k_∞ in 3.2, the central differencing results can be compared to the true sensitivity coefficients. The results are summarized in Table 3.3. Table 3.3, sensitivity coefficients are also obtained by applying central differencing to the true function as in Eq. 3.5. As expected, all models utilizing the central differencing formula produce self consistent sensitivity coefficients. The models differ from the analytic sensitivity coefficients only in the fourth decimal place, which is expected given the $\mathcal{O}(\Delta\Sigma^2)$ convergence of the central differencing formula.

Finally, the performance of a Kriging surrogate, as described in section 2.2, in modeling the infinite multiplication factor will be assessed. Before a Kriging surrogate is constructed the designer must agree on the number of points that will be used to construct the surrogate. As the number of points increases the accuracy of the Kriging surrogate is expected to increase. Recall that each point requires an evaluation of the true objective function, in this case Eq. 3.2. Since Eq. 3.2 is linear relatively few sampling points are expected to exactly reproduce the objective function. Indeed, these expectations are demonstrated in Fig. 3.3.

In Fig. 3.3 the number of points used to construct the Kriging surrogate is gradually increased. For each surrogate the root mean square error is calculated by having both Eq. 3.2 and the surrogate evaluate the same 100 randomly chosen points in the design space. Observe that for ten evaluation points, which is twice the number of design variables in the objective function, the error essentially goes to zero. This result can be explained by Eq.

Figure 3.3: Reduction in Kriging surrogate error for the infinite multiplication factor as the number of points used to build the surrogate increases.



3.2's linearity and the fact that Kriging is an interpolation method, thus requiring a pair of points in each dimension for exact interpolation.

3.2 Point Kinetics/Lumped Thermal Hydraulics

3.2.1 Problem Statement

A reduced order model based on the anchored-ANOVA decomposition will be constructed in this section for a simple system of ordinary differential equations modeling a transient in a BN800 sodium fast cooled reactor. The physical model of the reactor consists of point kinetics to model the neutronics and lumped thermal hydraulics equations to describe temperature feedback. The coupled system is nonlinear and only has a time dependence. Previous research groups have utilized point kinetics and lumped thermal hydraulics equations to model basic reactor systems in [26], [25], and [31]. In this section a reduced order model will be constructed for the maximum fuel temperature attained following a reactivity insertion as a function of the random variables exhibited in the description of the point kinetics/lumped thermal hydraulics system.

The six-group point kinetics equations modeling the neutronics of a reactor consist of a balance for reactor power $P(t)$ and a balance equation for each of the six precursor concentrations $C_k(t)$. Changes in reactor power are dependent on the precursor concentration, decays constants λ_k, delayed neutron fraction β and the mean neutron generation time Λ as detailed in,

$$\frac{dP}{dt} = \frac{\rho(T_f, T_c, t) - \beta}{\Lambda} P + \sum_{k=1}^{6} \lambda_k C_k. \tag{3.6}$$

The reactivity ρ depends on feedback from the fluids temperature models for the reactor fuel and coolant, which in turn depend on reactor power. The expression for each of the k precursor concentrations is written as,

$$\frac{dC_k}{dt} = -\lambda_k C_k + \frac{\beta_k}{\Lambda} P. \tag{3.7}$$

As for the ordinary differential equations describing the behavior of the reactor coolant system, two coupled equations suffice. For the fuel temperature T_f, the following lumped model is used,

$$M_f c_{pf} \frac{dT_f}{dt} = P + Ah(T_c - T_f) \tag{3.8}$$

where M_f is the lump fuel mass, c_{pf} is the specific heat capacity of the fuel, A is the heat transfer surface, and h is the heat transfer coefficient between the coolant and reactor fuel.

Finally, the coolant temperature is described as,

$$M_c c_{pc}\left(\frac{dT_c}{dt}+v\frac{T_c-T_{in}}{L}\right)=Ah(T_f-T_c) \tag{3.9}$$

where M_c is the lump coolant mass, c_{pc} is the specific heat capacity of the coolant, L is the coolant channel length, v is the coolant flow velocity, and T_{in} is the inlet coolant temperature. The initial conditions for P, C_k, T_f, and T_c depend on the initial power in the reactor P_0 before any kind of transient occurs and are listed in 3.10.

$$\begin{aligned} P(0) &= P_0 \\ C_k(0) &= \frac{\beta_k}{\lambda_k\Lambda}P_0 \\ T_f(0) &= T_c(0)+\frac{P_0}{Ah} \\ T_c(0) &= T_{in}+\frac{P_0 L}{M_c c_{pc} v} \end{aligned} \tag{3.10}$$

Serving as the coupling device between the lumped thermal hydraulics model and point kinetics model is the reactivity, which is proportional to the coolant temperature and the fuel temperature. Of course, any external reactivity ρ_{ex} added to the reactor is also a contributor. The time dependent reactivity is given explicitly as,

$$\rho(T_f,T_c,t)=\rho_{ex}+\alpha_d(T_f-T_f(0))+\alpha_c(T_c-T_c(0)) \tag{3.11}$$

where α_d and α_c are the doppler and coolant coefficients of reactivity, respectively.

The equations in 3.6, 3.7, 3.9, and 3.8 are used to model the transient resulting from a half sawtooth external reactivity insertion, as shown in 3.12.

$$\rho_{ex}(t)=\begin{cases} t\rho_{max}/20 & t\leq 20 \\ 0 & t>20 \end{cases} \tag{3.12}$$

By treating the coefficients in the point kinetics/lumped thermal hydraulics model as random variables, the objective function investigates the response surface for the maximum fuel temperature attained during transient. The reduced order methodologies will be tested against the stated problem. A depiction of the transient at the random variables' mean values, along with the external reactivity is shown in Figure 3.4. A total of twenty two random variables will be investigated for their affect on the maximum fuel temperature attained during transient. The random variables' mean values, along with their standard deviations are listed in 3.4. Note that all standard deviations are taken to be 5% of the mean value. All

Figure 3.4: Transient resulting from a half sawtooth external reactivity insertion, as modeled using the mean parameter values of the point kinetics/lumped thermal hydraulics system.

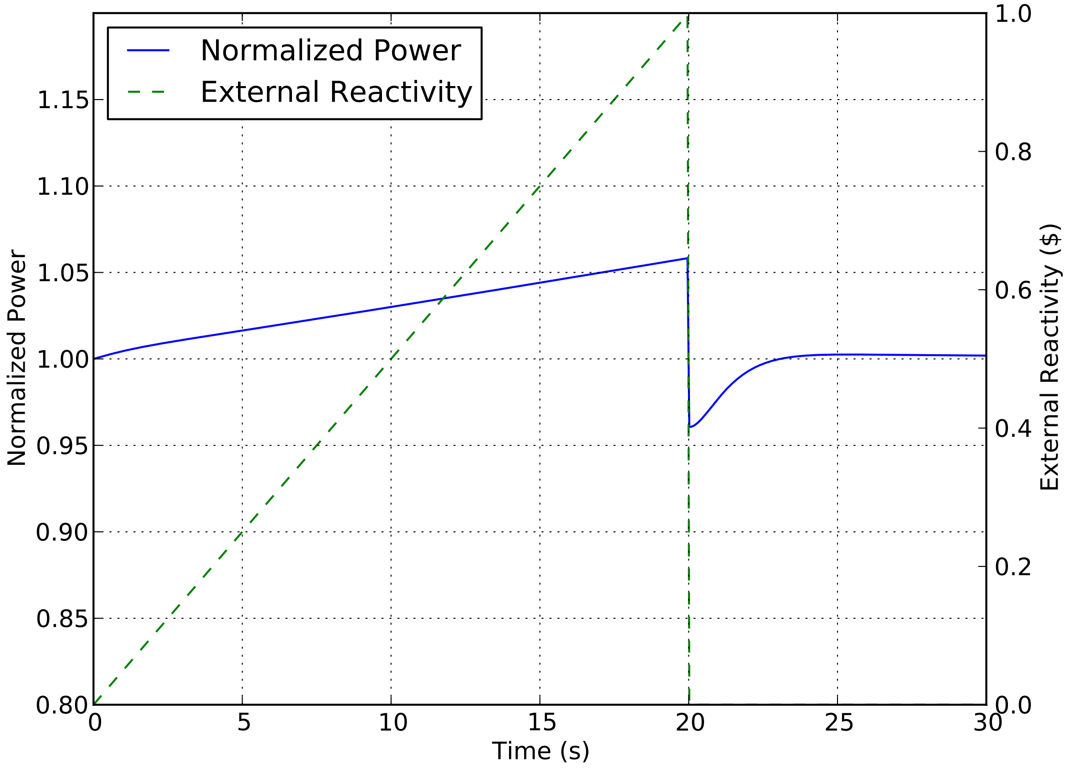


Table 3.4: Mean parameter values used in the point kinetics/lumped thermal hydraulics model for the analysis of a BN800 fast sodium cooled reactor.

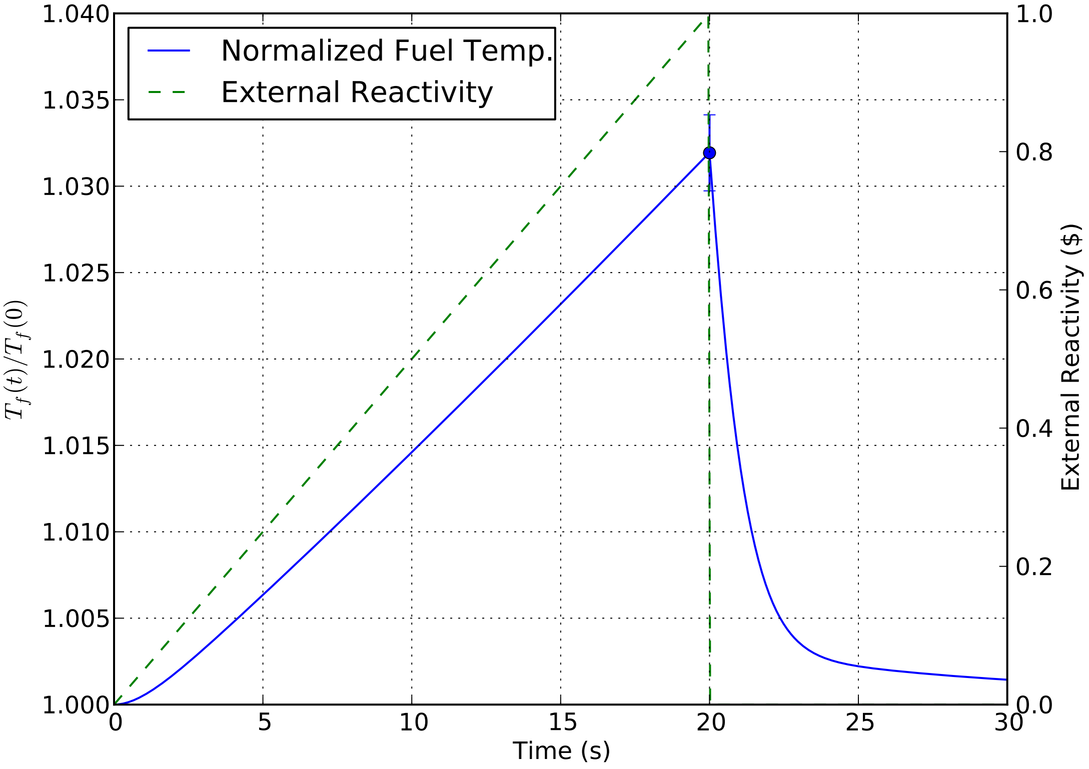
Random Variable	Units	Mean	Standard Dev.
λ_1	s^{-1}	1.24E-02	6.20e-04
λ_2	s^{-1}	3.05E-02	1.52e-03
λ_3	s^{-1}	1.11E-01	5.55e-03
λ_4	s^{-1}	3.01E-01	1.50e-02
λ_5	s^{-1}	1.14E+00	5.70e-02
λ_6	s^{-1}	3.01E+00	1.50e-01
β_1		9.00E-05	4.50e-06
β_2		8.53E-04	4.26e-05
β_3		7.00E-04	3.50e-05
β_4		1.40E-03	7.00e-05
β_5		6.00E-04	3.00e-05
β_6		5.50E-04	2.75e-05
Λ	s	4.00E-07	2.00e-08
Ah	kW/K	2.50E+06	1.25e+05
M_c	kg	1.16E+03	5.84e+01
M_f	kg	9.67E+03	4.83e+02
c_{pc}	$J/kg \cdot K$	1.20E+03	6.00e+01
c_{pf}	$J/kg \cdot K$	5.00E+02	2.50e+01
v	m/s	7.50E+00	3.75e-01
α_d	pcm/K	6.87E-06	3.43e-07
α_c	pcm/K	1.23E-06	6.15e-08
ρ_{max}		4.19E-04	2.09e-05

random variables are assumed to be independent of one another, as was assumed in [25]. A plot of the fuel temperature as a function of time due to the external reactivity profile shown in the same figure is depicted in Figure 3.5.

3.2.2 Analysis

An adaptive reduced order model, whose formulation is summarized in Algorithm 7, will be created for the problem described in section 3.2.1. The reduced order model will be investigated for its ability to reproduce statistics of interest by comparing its results with those obtained from sampling the true function. As described in Algorithm 7, the first step in creating a reduced order model for the maximum fuel temperature is to construct all first order components in the anchored-ANOVA decomposition and to identify the important ones. The sparse grids comprising the reduced order model are assumed to be converged

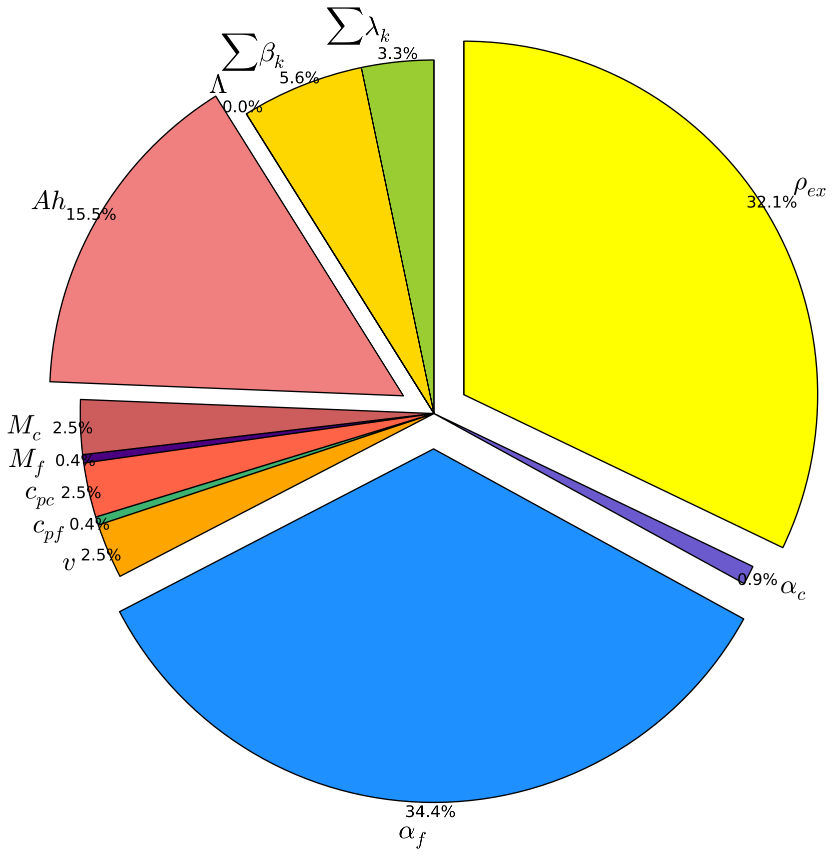
Figure 3.5: Fuel temperature transient resulting from a half sawtooth reactivity insertion. All parameters in the coupled point kinetics/lumped thermal hydraulics equations are held at their mean values.



when the maximum hierarchical surplus for a given level is less than 10^{-5}. Consequently, at least five digits of accuracy are expected. Important dimensions are those whose normalized sensitivity index in Eq. 2.54 exceeds 5%.

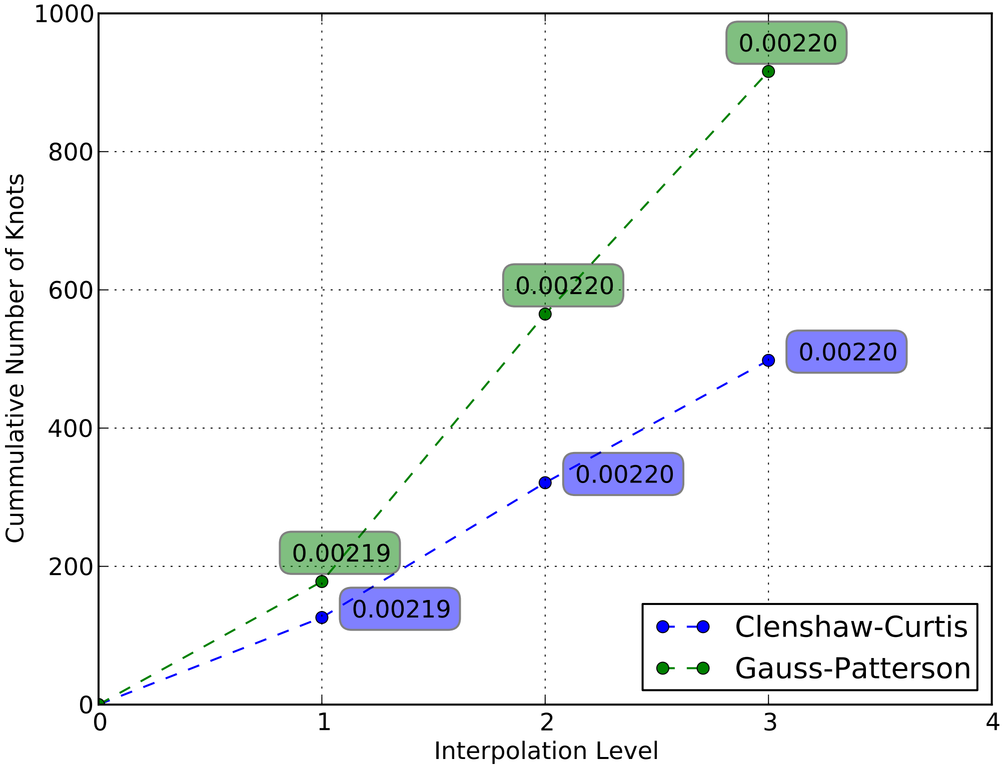
From Figure 3.6 the ”important” variables are identified to be Ah, α_d, and ρ_{max}. Collectively these three ”important” random variables comprise 82% of the total sensitivity. The indication that the maximum fuel temperature is sensitive to the heat transfer Ah from fuel to coolant is not surprising since in Eq. 3.8 the fuel temperature is directly proportional to Ah. Further, the random variables ρ_{max} and α_d determine the slope of the increase in fuel temperature, as seen in Figure 3.5, and so a strong sensitivity to these random variables is expected. The sensitivity of the maximum fuel temperature to α_c is not as great since the increase in coolant temperature during the transient is significantly smaller than the rise in fuel temperature. Relatively weak sensitivity to M_f and c_{pf} can perhaps be attributed to cancellation of error since these two variables are multiplied together.

Figure 3.6: Normalized sensitivity indices for all random variables comprising the coupled point kinetics/lumped thermal hydraulics equations. The effects of all β_k and λ_k have been lumped into single β and λ effects, respectively.



With only three random variables deemed as ”important” only three second order anchored-ANOVA components must be built for the reduced order model. Neglecting any convergence criteria, third order component describing the interaction effects among the three important random variables is also built. A summary of the total number of function evaluations needed to construct the adaptive reduced order model for the maximum fuel temperature is shown in Figure 3.7. The Gauss-Patterson scheme required almost twice as many knots as Clenshaw-Curtis to adaptively build the reduced order model. From Figure 3.7 it’s clear that the reduced order model consisting of only one dimensional components is effectively just as accurate as the full model, but requiring only 126 function evaluations to

Figure 3.7: Number of knots needed to adaptively construct a reduced order model for the maximum fuel temperature in both the Clenshaw-Curtis and Gauss-Patterson schemes. Boxed values state the standard deviation calculated at each level.



build using Clenshaw-Curtis. To see how well the reduced order models are able to reproduce the mean and variance of the true function Monte Carlo simulation is utilized. The models produced using anchored-ANOVA decomposition with superposition dimensions of one and three are sampled along with the true function. Mean, variance, and pertinent 99% confidence intervals for the sampling are summarized in Table 3.5. A total of 1000 samples were used for each method, each using the same random numbers. As evidenced in Table 3.5 the statistical results for each method are consistent. While the reduced order model with superposition dimension of three is able to replicate the Monte Carlo results to five significant figures, the expected accuracy, the order-one superposition model is slightly short. Of course, this is due to the absence of higher order components. However, the proximity of the order-one superposition model's results to the true results indicate that for this problem the higher order components do not have a significant impact. To further verify the ability of the reduced order models to accurately reproduce basic statistical moments, the

Table 3.5: Mean and variance data for the maximum fuel temperature achieved during transient obtained using Monte Carlo sampling. The same random numbers were used for all 1000 samples for each method.

Method	**Mean**	**99% CI**	**Standard Dev.**	**99% CI**
1D ANOVA CC	1.03193	(1.03175, 1.03211)	0.002187	(0.002068, 0.002320)
All ANOVA CC	1.03193	(1.03175, 1.03211)	0.002196	(0.002076, 0.002330)
1D ANOVA GP	1.03193	(1.03175, 1.03211)	0.002187	(0.002068, 0.002320)
All ANOVA GP	1.03193	(1.03175, 1.03211)	0.002196	(0.002076, 0.002330)
True Function	1.03193	(1.03175, 1.03211)	0.002196	(0.002076, 0.002330)

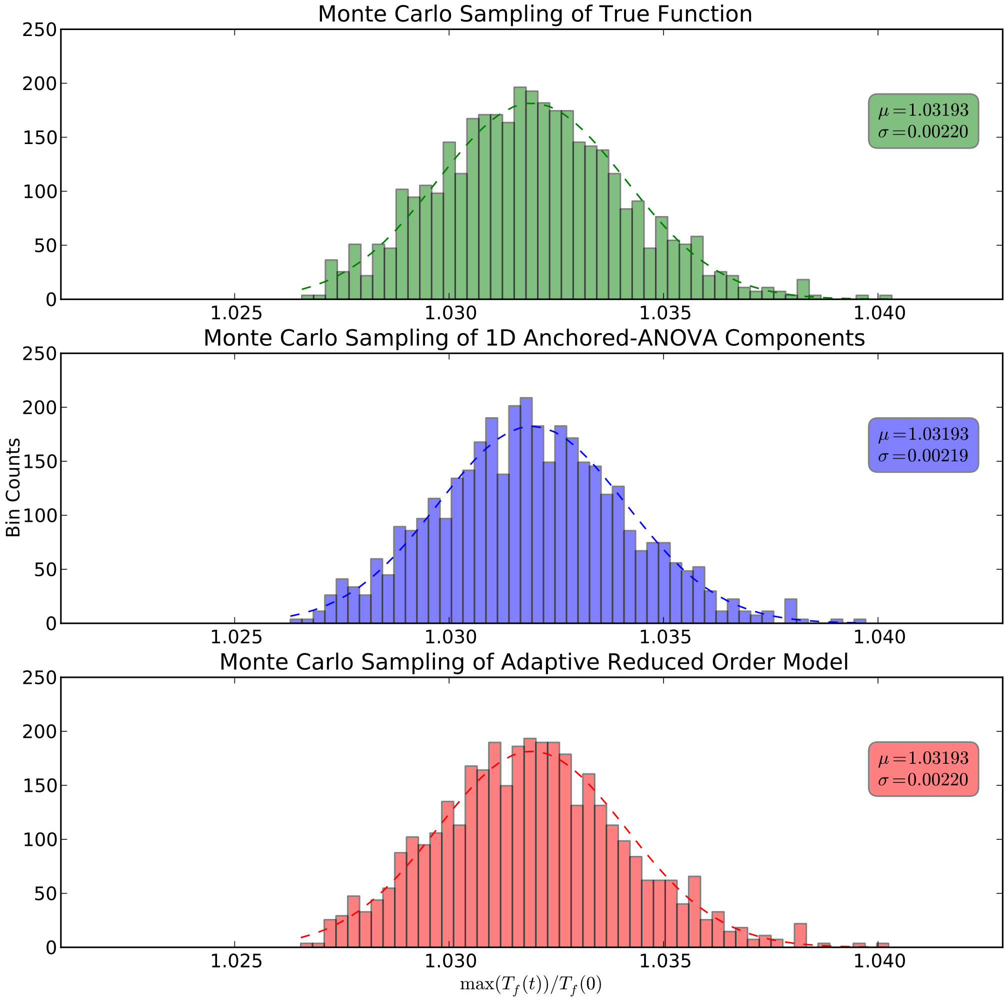
probability distributions for the normalized maximum fuel temperature produced by each model are compared in Figure 3.8. All of the tested reduced order models are able to reproduce the Gaussian probability distribution for the normalized maximum fuel temperature.

As done in section 3.1.2, a sensitivity analysis will be completed for the reduced order model and compared to the normalized sensitivity coefficients obtained using central differencing. From Table 3.6 notice that the largest sensitivity coefficients are those of the random variables deemed ”important” using the adaptive reduced order model algorithm. Only the sensitivity coefficients for the Clenshaw-Curtis sparse grid are shown in Table 3.6 since the Gauss-Patterson sparse grid returns identical results. The normalized sensitivity coefficients calculated using the reduced order models are the same as those calculated using central differencing to the expected number of significant digits.

To further analyze the coupled point kinetics/lumped thermal hydraulics problem in hand, Morris' algorithm is applied, as described in section 2.1.1. As previously described, algorithm 1 aims to identify the design variables that have the greatest influence on an objective function's behavior, which in this case is the fuel temperature. To effectively determine such design variables it is convenient to plot the mean against the standard deviation of each design variable's elementary effects. From the problem at hand, the elementary effect statistics are plotted in Fig. 3.9. Figure 3.9 was obtained using five divisions and ten elementary effects per design variable. Observe how eighteen of the twenty two points are clustered around the origin, implying that each of these clustered variables have minimal influence on the fuel temperature. In contrast, the variables ρ_{ex}, α_f, and Ah are likely to be primary contributors. The results of Fig. 3.9 are entirely consistent with Fig. 3.6, which also identifies the variables ρ_{ex}, α_f, and Ah as being principally important.

Now, with the three key variables identified a kriging surrogate is constructed for the normalized fuel temperature. Sampling the three-dimensional surrogate 1000 times, as done to obtain the results in Table 3.5, a mean normalized temperature of 1.03198 ± 0.002299 was obtained which is well within the 99% confidence interval of the sampled

Figure 3.8: Histograms produced by sampling the true function, order-one superposition reduced order model, and the full adaptive reduced order model.



true result. Note that some of the statistical discrepancies being observed can be attributed to the fact that the same random number were not used in the evaluation of the surrogate and true function.

Table 3.6: Normalized sensitivity coefficients of the maximum fuel temperature to random variables.

Random Variable	1D ANOVA CC	All ANOVA CC	Central Diff.
λ_1	3.7894E-05	3.7894E-05	3.7895E-05
λ_2	7.0387E-04	7.0387E-04	7.0387E-04
λ_3	8.4244E-04	8.4244E-04	8.4215E-04
λ_4	9.7308E-04	9.7309E-04	9.7379E-04
λ_5	1.1572E-04	1.1572E-04	1.1607E-04
λ_6	4.4638E-05	4.4639E-05	4.0498E-05
β_1	-3.2992E-04	-3.2992E-04	-3.2976E-04
β_2	-2.6582E-03	-2.6582E-03	-2.6616E-03
β_3	-1.1953E-03	-1.1953E-03	-1.2040E-03
β_4	-1.0129E-03	-1.0129E-03	-1.0232E-03
β_5	-1.1689E-04	-1.1689E-04	-1.1810E-04
β_6	-4.0718E-05	-4.0718E-05	-4.1134E-05
Λ	-8.9294E-08	-8.9295E-08	-8.9364E-08
Ah	1.2553E-02	1.2553E-02	1.2584E-02
M_c	1.8753E-03	1.8753E-03	1.8716E-03
M_f	-3.6695E-04	-3.6695E-04	-3.6360E-04
c_{pc}	1.8753E-03	1.8753E-03	1.8903E-03
c_{pf}	-3.6695E-04	-3.6695E-04	-3.5976E-04
v	1.8838E-03	1.8839E-03	1.9177E-03
α_d	-2.6655E-02	-2.6656E-02	-2.6625E-02
α_c	8.4387E-04	8.4387E-04	8.7194E-04
ρ_{max}	3.1164E-02	3.1164E-02	3.1272E-02

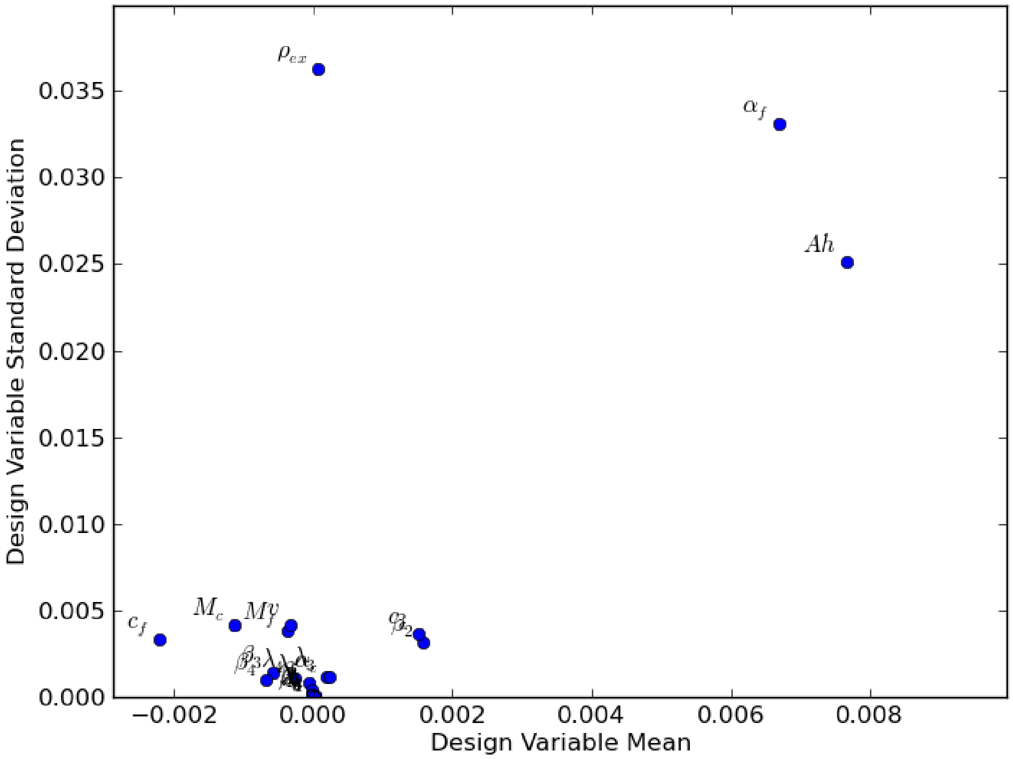
3.3 TMI Minicore

3.3.1 Problem Statement

The previous two problems dealt with relatively simple functions that don't require industrial engineering codes to solve. However, the main intention of this thesis is to construct reduced order models for computer codes that aim to model large and complex engineering systems. Interaction with such computer codes consist of input and output files; the governing equations and their solvers are rarely seen. The primary purpose of this demonstration problem is to show that the same algorithms applied to analyze the previous problems are also functional when applied to engineering computer codes.

In this demonstration problem the reactor core simulator code PARCS [16] is applied to the TMI minicore described in the first phase of the UAM Benchmark [34]. The minicore problem consists of a three-by-three fuel assembly configuration with reflector blocks

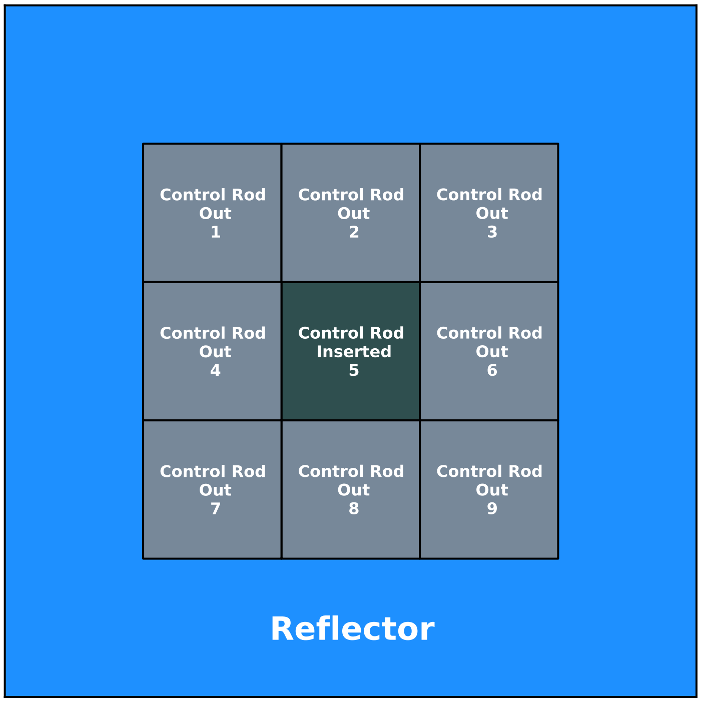
Figure 3.9: Statistics of each design variable's elementary effects, as applied to the coupled point kinetics/lumped thermal hydraulics problem.



placed around the assemblies, as seen in Figure 3.10. In the minicore the central assembly is rodded while the periphery fuel assemblies are unrodded. Vacuum boundary conditions are applied. The few-group, homogenized cross section description for each fuel assembly consists of transport, absorption, nu-fission, and scatter cross sections along with values for ADFs. For a two-group problem the total number of cross sections to describe an assembly is nine. Since the homogenized reflector region does not support fission only seven homogenized cross sections are required to describe it. Consequently, to model the minicore configuration in Fig. 3.10 in PARCS a total of twenty five homogenized, two-group cross sections are needed.

In order to study the effects of the uncertainties inherent in the few-group cross sections on output parameters of interest in PARCS, a few-group covariance matrix is necessary. The few-group covariance matrix is obtained using the 'two-step' method depicted in Fig. 1.1. A total of 300 transport calculations with perturbed multigroup cross sections were

Figure 3.10: TMI minicore configuration used for analysis, as defined in the UAM Benchmark specifications.



executed to generate the few-group covariance matrix. In the previous example problems only one output parameter was investigated at a time. However, recall in the discussion of the Smolyak algorithm that interpolation is only performed on the random space. The objective function is evaluated at each abscissa in the random space, returning an output that can still be a function of physical space. In this case the hierarchical surplus is no longer one dimensional. Rather, the hierarchical surplus in Eq. 2.36 is rewritten as,

$$f(\mathbf{x},x_{j_1}^{i_1},\ldots,x_{j_d}^{i_d})-A(q-1,d)(\mathbf{x},x_{j_1}^{i_1},\ldots,x_{j_d}^{i_d}) \tag{3.13}$$

where $\mathbf{x}\in\mathscr{D}$ is a coordinate in the spatial domain. When the objective function is a function of spatial coordinates the Smolyak algorithm approximates the function as a linear combination of vectors, the linear weights still being tensor products of basis functions for the random space. To determine the mean and variance of a reduced order model approximation of some space-dependent objective function the L_2 norm is taken over the spatial domain $\mathscr{D}$. Similarly, to identify important dimensions using the sensitivity coefficient defined in Eq. 2.54 the L_2 norm is taken of $\eta_i(\mathbf{x})$.

The problem in this section considers the core box power distribution of the minicore in Fig. 3.10 and consequently, the objective function is space dependent. For each simulation in PARCS a vector of length nine is returned with each entry containing the relative box power in the fuel assemblies. The box powers are calculated such that the average of all nine entries is identically equal to unity. In the PARCS output file the box powers are only given to four digits of accuracy and therefore roundoff error in this problem warrants some attention.

3.3.2 Analysis

In this analysis a pure 'two-step' approach is used to compare the results obtained using a reduced order model for the box powers output in PARCS. For the 'two-step' approach a few group covariance matrix is constructed and sampled 500 times, with each sample containing perturbed few group cross sections. Each cross section set is propagated through the PARCS code and the box powers for the fuel assemblies are extracted from the output file. Ultimately, 500 box power outputs are analyzed statistically to obtain correlations, means, and variances. The same few group covariance matrix is used to sample the reduced order model for the box powers.

To build a reduced order model, Algorithm 7 is applied with a hierarchical surplus threshold of 10^{-4} since the PARCS box powers are only output to four decimal places. Consequently, roundoff error may accumulate in the fourth decimal point of the reduced order model. A total of 123 executions of PARCS were required to construct all single order components of the reduced order model using Clenshaw-Curtis abscissas. Due to the geometry of the TMI minicore, 1/8 symmetry is expected in the box power results. Consequently, only fuel assemblies one, four and five are investigated, as defined in Fig. 3.10. Sampling results for the mean and standard deviation for the true box power PARCS model and the reduced order model are summarized in Tables 3.7 and 3.8, respectively. As always, the same random numbers are used to produce each sample when comparing two different methods. The results are identical in all cases to four significant digits and in most cases, to five significant digits. In the reduced order model results notice that the standard deviation for assembly six is slightly off in the fourth decimal place when compared to assemblies two, four, and eight. The standard deviation should be equal in these four assemblies due to symmetry, indicating the presence of roundoff error.

Adding higher order components in accordance with Algorithm 7 did not improve the

Table 3.7: Mean and standard deviation data for TMI minicore box powers where PARCS code is used as objective function. A total of 500 samples were used. Assembly numbers correspond to Fig. 3.10.

Assembly	Mean	99% CI	Standard Dev.	99% CI
1	0.8387	(0.8386, 0.8388)	0.0007	(0.0006, 0.0008)
2	1.1499	(1.1498, 1.1500)	0.0011	(0.0010, 0.0012)
3	0.8387	(0.8386, 0.8388)	0.0007	(0.0006, 0.0008)
4	1.1499	(1.1498, 1.1500)	0.0011	(0.0010, 0.0012)
5	1.0453	(1.0450, 1.0456)	0.0027	(0.0025, 0.0029)
6	1.1499	(1.1498, 1.1500)	0.0011	(0.0010, 0.0012)
7	0.8387	(0.8386, 0.8388)	0.0007	(0.0006, 0.0008)
8	1.1499	(1.1498, 1.1500)	0.0011	(0.0010, 0.0012)
9	0.8387	(0.8386, 0.8388)	0.0007	(0.0006, 0.0008)

Table 3.8: Mean and standard deviation data for TMI minicore box powers where the objective function is a reduced order model for PARCS containing only 1D components. A total of 500 samples were used. Assembly numbers correspond to Fig. 3.10.

Assembly	Mean	99% CI	Standard Dev.	99% CI
1	0.8387	(0.8386, 0.8388)	0.0007	(0.0006, 0.0008)
2	1.1500	(1.1499, 1.1501)	0.0012	(0.0011, 0.0013)
3	0.8387	(0.8386, 0.8388)	0.0007	(0.0006, 0.0008)
4	1.1500	(1.1499, 1.1501)	0.0012	(0.0011, 0.0013)
5	1.0455	(1.0452, 1.0458)	0.0027	(0.0025, 0.0029)
6	1.1500	(1.1499, 1.1501)	0.0011	(0.0010, 0.0012)
7	0.8387	(0.8386, 0.8388)	0.0007	(0.0006, 0.0008)
8	1.1500	(1.1499, 1.1501)	0.0012	(0.0011, 0.0013)
9	0.8387	(0.8386, 0.8388)	0.0007	(0.0006, 0.0008)

statistics of the reduced order model. For the purposes of uncertainty quantification the reduced order model consisting entirely of 1D components is sufficient for this problem. The bivariate distributions for the box powers for all combinations of assemblies one, four, and five are given in Fig. 3.11. For each case in Fig. 3.11 the bivariate distributions are obtained by sampling the reduced order model and PARCS. Variations between the overlaid distributions can be attributed to roundoff error. A decrease in the box power of assembly five is equivalent to an increase in the absorption of the assembly's control rods, which effectively shifts the neutron flux away from the central assembly. Consequently, a negative correlation is expected between assemblies one and five and between assemblies one and four. Indeed, statistical sampling has the Pearson correlation coefficient for the box powers of assemblies one and four to be -0.83 and -0.84 between assemblies one and five. However, negative correlation coefficients between assembly one and assemblies four and five implies a positive correlation coefficient between assemblies four and five. The positive correlation coefficient of $+0.40$ between assemblies four and five is evident in Fig. 3.11.

Further investigation into the decisive parameters effecting the core power distribution can be obtained with a design variable screening study. Applying algorithm 1 to the normalized power distribution of assemblies one, two, four, and five, the plot in Fig. 3.12 is obtained. As seen in Fig. 3.12, a cluster of design variables are centered around the origin indicating a lack of activity relating to the core power. For example, the ADFs in each of the four assemblies have a negligible effect on the core power. On the contrary, the cross sections Σ^i_{12}, Σ^o_{12} and Σ^o_{tr} stand out from the cluster, which is no surprise considering the TMI minicore is a scattering dominant problem. The key design variables effecting assembly five's core power, which essentially controls the remaining power in the core, are Σ^i_{a1}, $\nu\Sigma^i_{f2}$, $\nu\Sigma^o_{f2}$, and the scattering cross sections. Note that the elementary effects of the important design variables in assembly five are an order of magnitude greater than in the other assemblies.

From the observation of the important design variables in Fig. 3.12, a Kriging surrogate is constructed for the core power as a function of Σ^i_{a1}, $\nu\Sigma^i_{f2}$, Σ^i_{12}, Σ^o_{tr1}, $\nu\Sigma^o_{f2}$ and Σ^o_{12}. Using these variables a sampling plan consisting of twenty true PARCS evaluations was obtained using algorithm 3. Based on the optimized sampling plan a Kriging surrogate was constructed and sampled 500 times in accordance with the six design variables' covariance matrix. The results are summarized in Table 3.9 and should be compared to Tables 3.7 and 3.8. The random numbers used for sampling the the Kriging surrogate were not the

Table 3.9: Mean and standard deviation for TMI minicore box powers using a six dimensional Kriging surrogate. A total of 500 samples were used. Assembly numbers correspond to Fig. 3.10.

Assembly	Mean	99% CI	Standard Dev.	99% CI
1	0.8385	(0.8384, 0.8386)	0.0010	(0.0009, 0.0011)
2	1.1499	(1.1497, 1.1500)	0.0013	(0.0012, 0.0014)
3	0.8384	(0.8383, 0.8386)	0.0010	(0.0009, 0.0011)
4	1.1499	(1.1497, 1.1500)	0.0013	(0.0012, 0.0014)
5	1.0465	(1.0462, 1.0467)	0.0022	(0.0020, 0.0024)
6	1.1499	(1.1497, 1.1500)	0.0013	(0.0012, 0.0014)
7	0.8384	(0.8383, 0.8386)	0.0010	(0.0009, 0.0011)
8	1.1499	(1.1497, 1.1500)	0.0013	(0.0012, 0.0014)
9	0.8384	(0.8383, 0.8386)	0.0010	(0.0009, 0.0011)

same as those used to sample the true function and the collocation-based surrogate. Consequently, some of the discrepancies between Table 3.9 and Tables 3.7 and 3.8 are purely statistical. Despite this fact each assembly's core power mean and standard deviation confidence intervals overlap between the different modeling approaches. The Kriging surrogate consisting of six of the original twenty five design variables can adequately represent the physical phenomenon occurring in the TMI minicore. Of course, this conclusion can be made for this problem because the uncertainties in the cross sections are relatively small. Larger uncertainties in the design variables are likely to effect the ability of a surrogate's ability to represent its objective function.

3.4 General Observations

Before moving on to a large-scale engineering system a few general observations regarding the application of collocation-based and Kriging reduced order models to the uncertainty quantification of the reactor systems described are in order. Primarily, the demonstration problems indicate that use of only 1D components in the anchored-ANOVA expansion provides a very good approximation to the true system under investigation. From a computational point of view, 1D components are cheap to build and their quantity is equal to the number of random variables modeled. Higher order components generally require higher levels of interpolation and so their construction should be minimized if possible. As mentioned in [50], the interaction effects between random variables for most realistic physical systems have a negligible effect on outputs of interest.

Nevertheless, if multivariate components are needed algorithm 7 is able to identify the combination of components most likely to affect the output, which offers significant computational savings. For example, if all components of two random variables were to be included in the anchored-ANOVA expansion then 4950 two-dimensional sparse grid interpolants would need to be built. However, if only 5 of the 100 variables are deemed ”important” by algorithm 7 then only 10 two-dimensional sparse grid interpolants will be built. Not to mention, identification of variables that most affect the variability in some computer code output of interest offers insight into the physical system under consideration. It should be reemphasized that components of the anchored-ANOVA used to build reduced order models do not represent the order of effects on the output. Even the 1D component functions can model nonlinear behavior as discussed in section 2.3.2. Rather, the multivariate components describe the effect of input variables upon some output when acting together.

For the demonstration problems investigated in this chapter, the Clenshaw-Curtis collocation abscissas performed significantly better than Gauss-Patterson. The Clenshaw-Curtis knots offered effectively the same convergence rates as Gauss-Patterson with far fewer function evaluations. Not to mention, Clenshaw-Curtis knots are much easier to generate.

The application of Morris' Algorithm for design variable screening was found to be a visually useful tool for dimension reduction. Whether a computational problem has a large number of dimensions or not the results of Morris' Algorithm can be viewed on a two-dimensional plot. Design variables having the greatest influence on an objective output's behavior become instantly recognizable. After identifying each example problem's key design variables, a reduced order model based on surrogate Kriging was constructed. The Kriging models' ability to reproduce statistics calculated used exact and collocation-based models using only twenty or so basis points demonstrated great promise for further applications.

However, in all of the example problems analyzed in the current chapter the input uncertainties were calculated systematically beforehand and were therefore relatively small. In other words, engineering judgment never had to be applied in order to estimate the uncertainty of some model input parameter. Uncertainty estimates based on engineering judgment are reasonably expected to be significantly greater than those that can be calculated because more information is known about the latter. The introduction of large uncertainty estimates will be expected in the proceeding application, largely due to the fact that the sources of uncertainty are unknown. Such estimates are expected to negatively impact the smooth construction of surrogate models to represent the applications of interest.

Figure 3.11: Multivariate distributions for the box powers of two assemblies. Dashed distributions were obtained by sampling the reduced order model consisting of only 1D components.

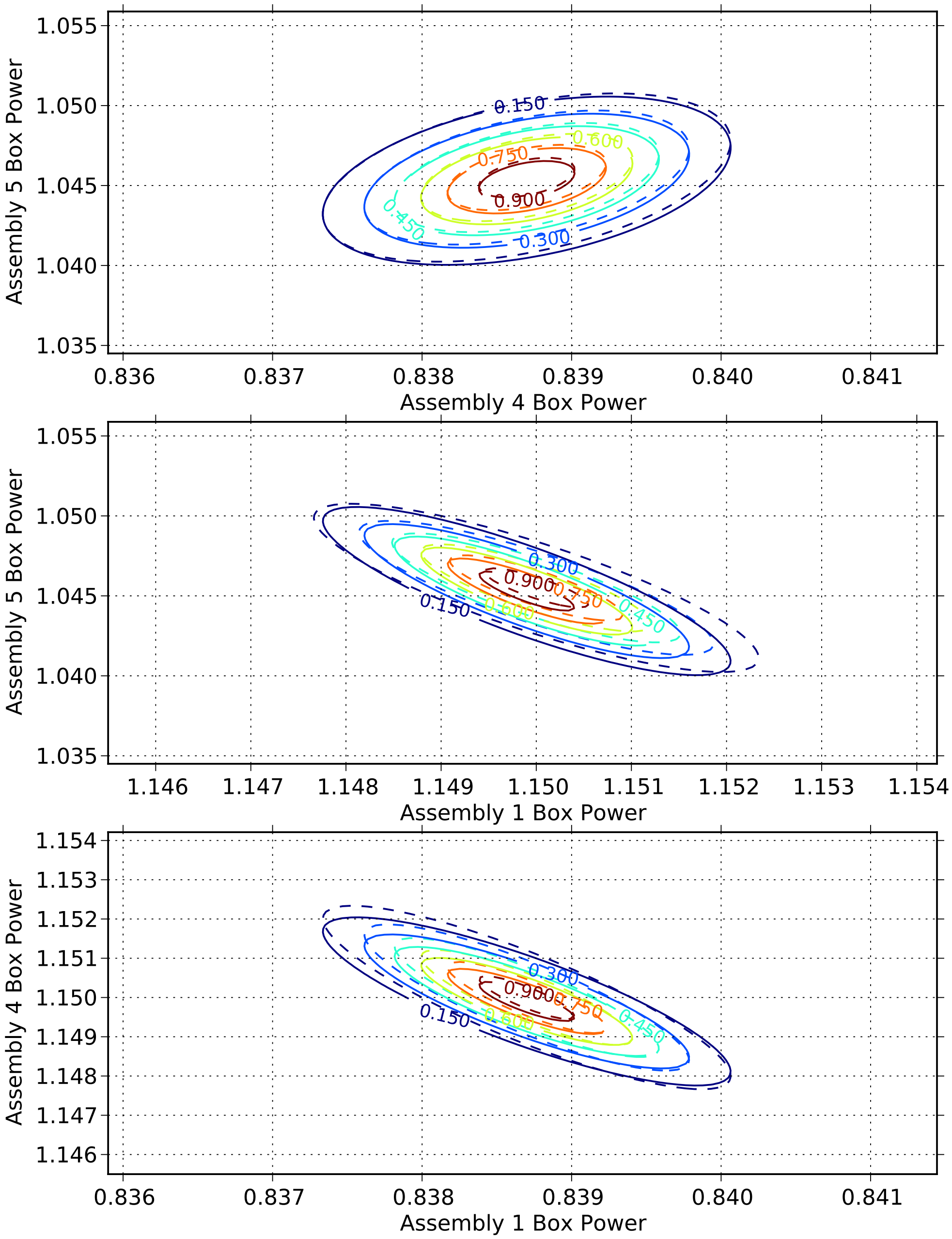
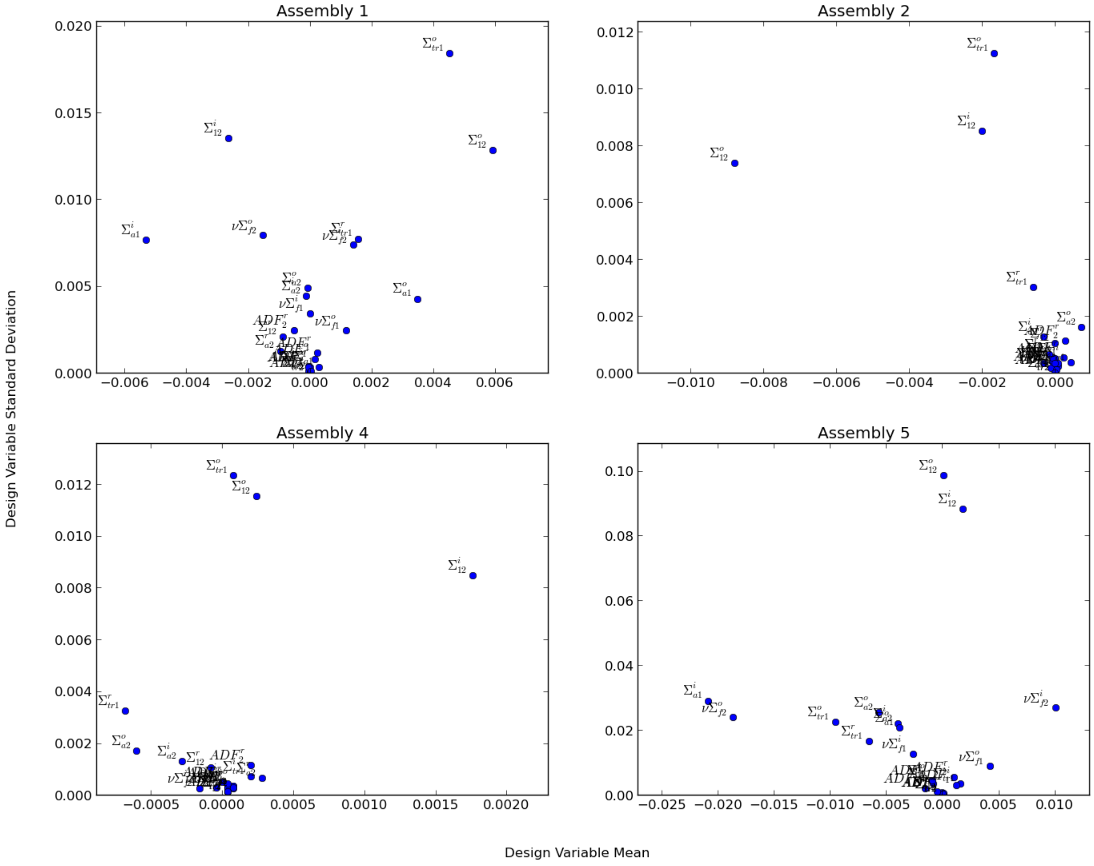


Figure 3.12: Design variable screening study for the TMI Minicore.



CHAPTER 4

Application to Fission Gas Release

4.1 Problem Overview

In the previous chapters collocation-based and Kriging surrogates were applied to several play problems in hopes of demonstrating the surrogates' efficacy. Ultimately, the same surrogate methods will be applied to a difficult problem in nuclear fuel performance modeling. Specifically, modeling the depletion behavior of high burnup fuel is of interest. The field of fuel performance modeling is an ideal application for surrogates because most modern fuel performance codes such as Bison [81] are computationally expensive and there is a relative abundance of experimental data to complement the computer codes. Recall that the true promise of surrogates arises when thousands of simulations of an expensive computer code are required, as is the case for optimization and calibration studies. The idea is to fold together computer simulations and experimental data to improve the computer code's predictive accuracy.

Calibration studies involving the fuel performance code Bison have already been conducted by Swiler et. al. in [71]. In [71], an optimal fuel relocation activation parameter is identified by an aggregated calibration to experimental observations of several Halden fuel rods. Surrogates were not required for the calibration study due to the relative simplicity of the fuel rod model used. The Dakota [2] framework was used to complete the calibration study.

For the proposed thesis research the application of interest arises from the extensive validation base for fuel performance modeling found in the Fumex-II database [40]. Specifically, the application involves calibrating fission gas parameters involved in modeling the Risø AN3 experiment since these parameters are notoriously uncertain. The Risø AN3 experiment consists of a base irradiation of four reactor cycles up to some 41 GWd/t, as shown in Fig. 4.1, followed by a power ramp. The base irradiation takes place in the Biblis A pressurized water reactor [40]. After the base irradiation period, a fuel rod is extracted

Figure 4.1: Base irradiation history for the Risø AN3 experiment.

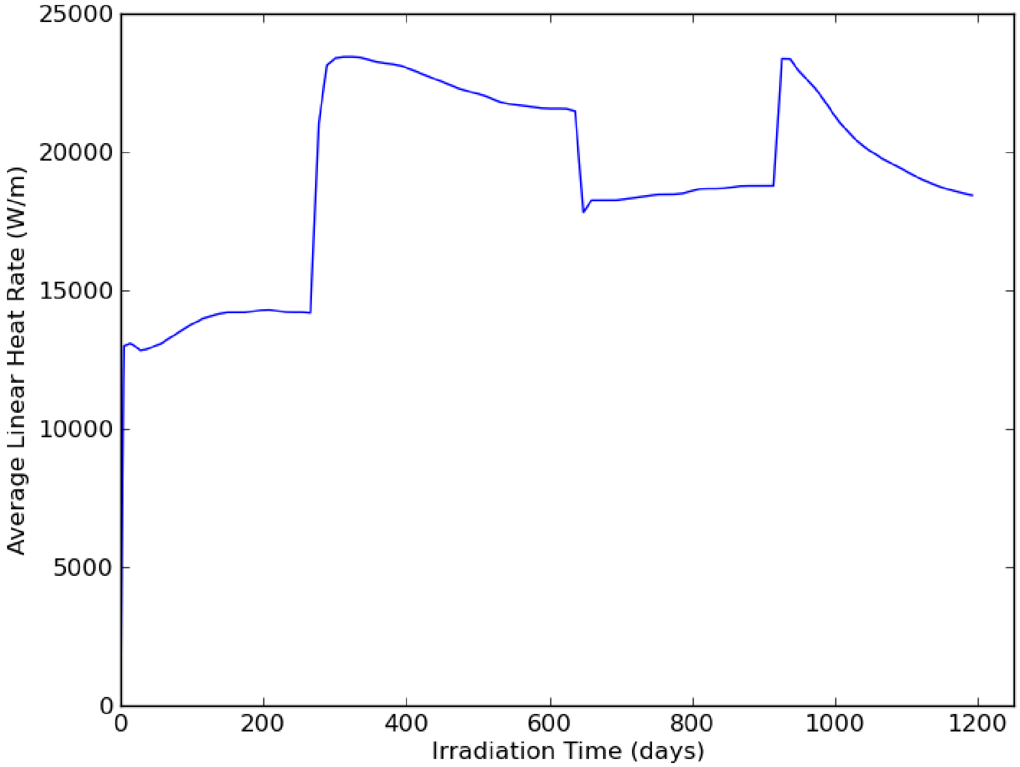
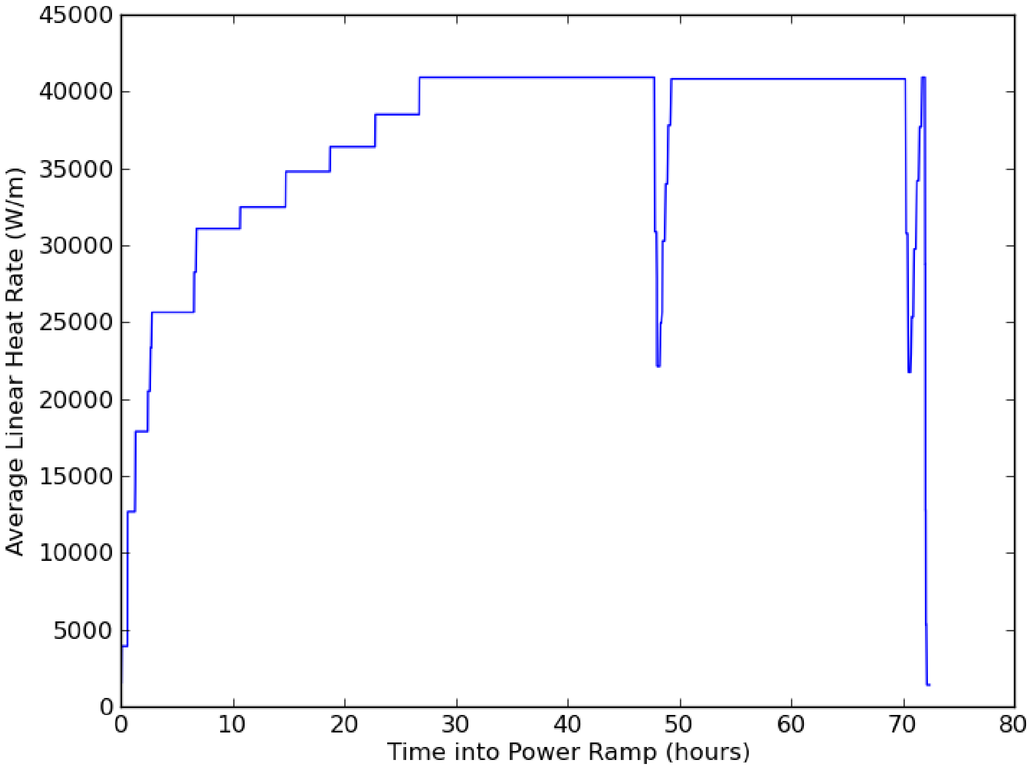


Figure 4.2: Power ramp experiment for the the Risø AN3 experiment.



and refabricated to a shorter length before undergoing the power ramp in Fig. 4.2. The refabricated fuel rod was outfitted with various instrumentation such that fuel centerline temperature, fission gas release and rod internal pressure measurements could be obtained. The Risø AN3 test rod specifications are summarized in Table 4.1. The reader is encouraged to consult [59] for more in-depth specifications and modeling details.

The Risø AN3 experiment was modeled using solely Bison in [60] and is included as a validation case for the Bison code. The relevant Bison input file and pertinent operating conditions, as utilized in this thesis, are included in Appendices A, B. Note, the Bison Risø AN3 model is based upon the conditions and specifications in Table 4.1 and [59]. To model fission gas release the SIFGRS model is utilized in Bison. Bison's prediction of fission gas release during the power ramp is displayed alongside the corresponding experimental results in Fig. 4.3. As seen in Fig. 4.3, Bison over predicts fission gas release by a factor of two some forty hours into the power ramp. Similarly, Bison's prediction of fuel centerline temperature is plotted against experimental data in Fig. 4.4. Since fission gas re-

Table 4.1: Risø AN3 test configurations and operating conditions.

Description	**Units**	**Value**
Overall fuel rod length	m	0.39058
Fuel stack length	m	0.286
Nominal plenum height	mm	61.0
Pre-ramp rod fill gas composition		He
Pre-ramp rod fill gas pressure	MPa	2.31
Refabfricated rod fill gas composition		He
Refabfricated rod fill gas pressure	MPa	1.57
Fuel material		UO_2
Fuel enrichment	%	2.95
Fuel density	%	93.74
Fuel inner diameter	mm	2.5
Fuel outer diameter	mm	9.053
Pellet dish diameter	cm	0.665
Pellet dish depth	cm	0.013
Pellet chamfer width	cm	0.046
Pellet chamfer depth	cm	0.016
Cladding material		Zr-4
Cladding outer diameter	mm	10.81
Cladding inner diameter	mm	9.258
Cladding wall thickness	mm	0.776
Base irradiation coolant inlet temperature	C	287.7
Base irradiation coolant pressure	MPa	15.52
Power ramp coolant pressure	MPa	15.3

Figure 4.3: Comparison of Bison fission gas release prediction to experimental results during the Risø AN3 power ramp.

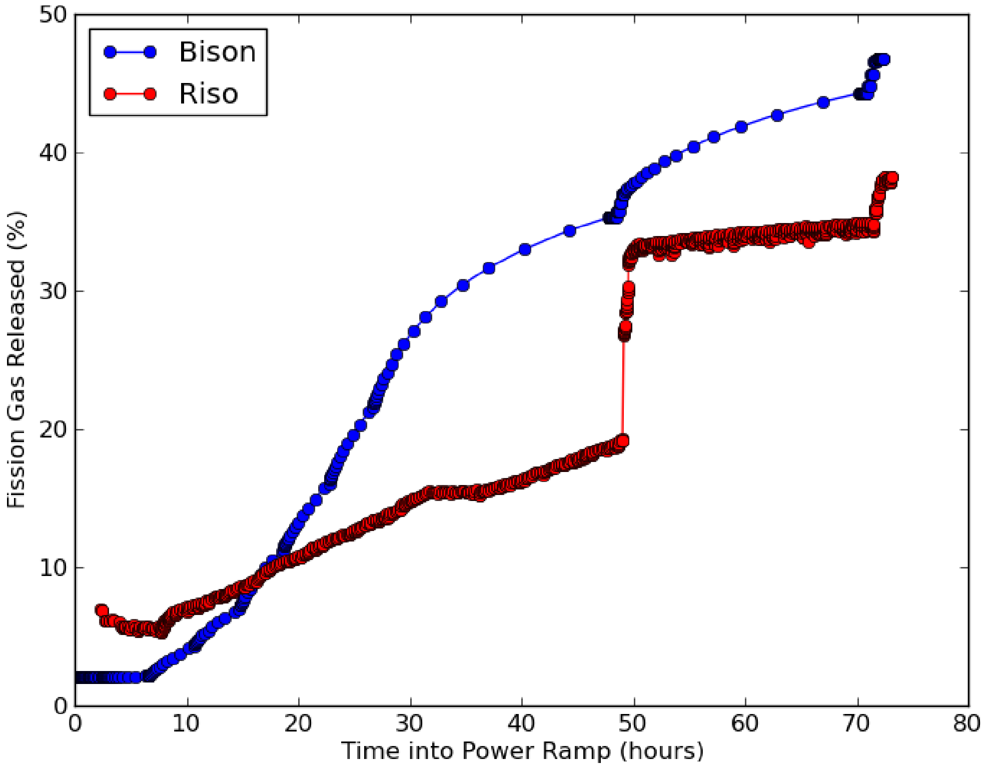
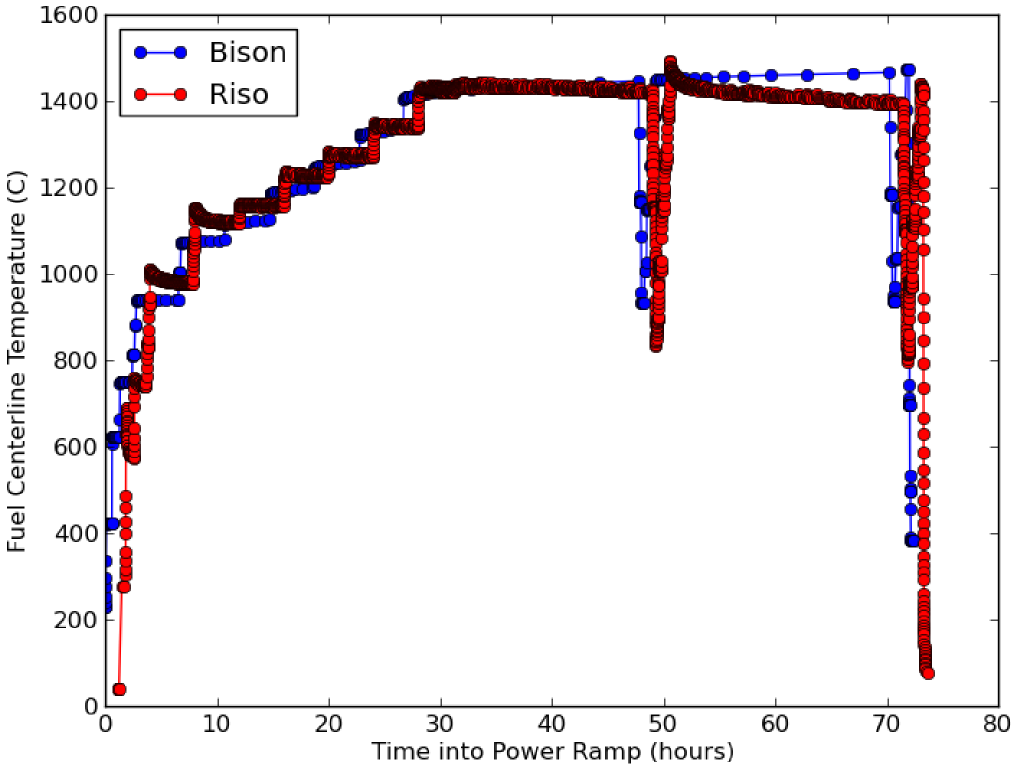


Figure 4.4: Comparison of Bison fuel centerline temperature prediction to experimental results during the Risø AN3 power ramp.

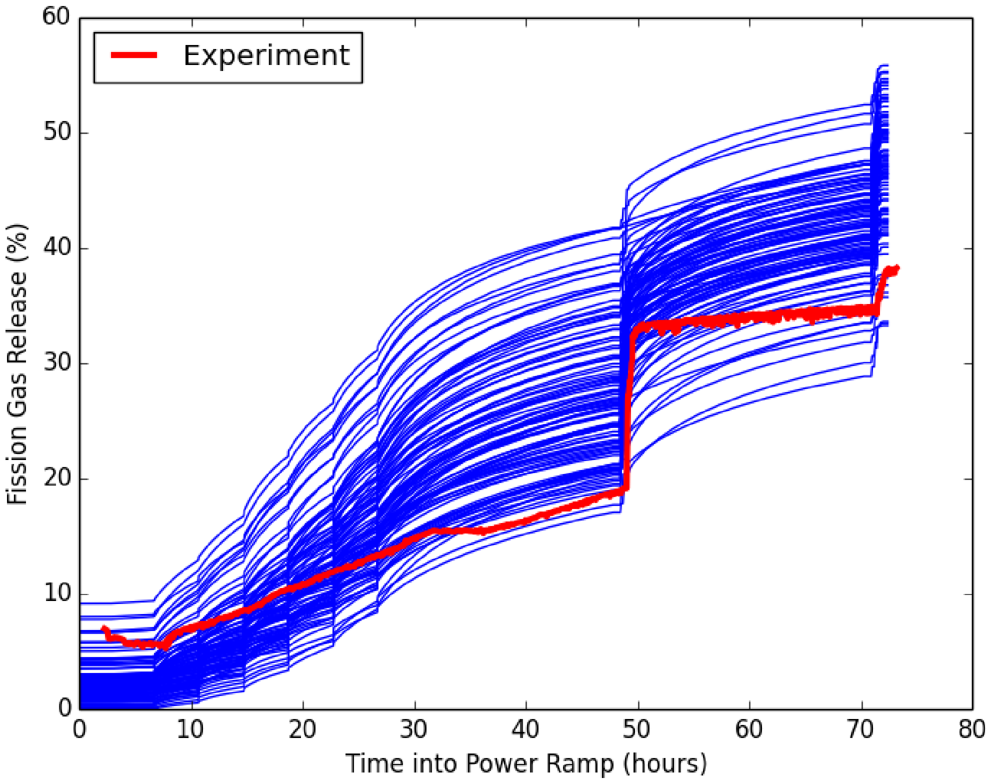


lease and fuel temperature are strongly coupled [60], it is anticipated that better fission gas release predictions will result in a more accurate fuel centerline temperature comparison.

Calibrated parameters in SIFGRS are expected to decrease the error between Bison's predicted output and the experimental data. Input parameters to SIFGRS such as the fuel grain radius, hydrostatic stress, fuel porosity, bubble surface tension, and the gas diffusion coefficient are quite generic and uncertain. Consequently, adjusting such parameters to better match experimental data is justified. However, Bison's fission gas release predictions are computationally expensive. Considering the calibration of parameters in SIFGRS will requires thousands of Bison instances, a surrogate model for fission gas release behavior becomes necessary. As in [69], the Dakota framework will be utilized to construct surrogate models for fission gas release behavior and to perform calibration routines.

The primary culprits in the SIFGRS model have been identified in [70] and [58] to be the initial fuel grain radius, fuel porosity, surface tension, temperature, fuel grain radius,

Figure 4.5: Fission gas release time series for 100 LHS Bison simulated Risø AN3 power ramp experiments.



diffusion coefficient, resolution parameter and grain boundary coefficient. The importance and role of of these variables will be described in section 4.2. For now, suffice it to say that the true values of these variables are unknown and only uniform probability distributions can be attached to each variable. As a starting point for analysis, the LHS module in Dakota was utilized to sample the variable space and simulate the resulting fission gas release time series resulting from the Risø AN3 power ramp. The 100 resulting time series are shown in Fig. 4.5. From Fig. 4.5, it is clear that Bison tends to over predict fission gas release. However, it is also apparent that the range of variables is sufficient to contend with the experimental data.

The problem of finding ”true” fission gas release parameters that will yield a release profile similar to that provided by experimental data is complicated by the quality of the experimental data. As seen in Fig. 4.3, the fission gas release is not monotonically increasing at all time steps in the power ramp. Of course, such a result is not physical. One possible explanation for the fission gas release decrease attributes the pressure transducers used to measure plenum pressure. Under some conditions released fission gases get trapped in fuel gaps and cracks, thereby not contributing to plenum pressure. The same phenomenon is believed to cause the large release of fission gas occurring at approximately hour fifty into the power ramp [55]. Observe the large drop in power occurring at the same time as the spike in rod internal pressure measured by the transducer. As the power drops the cracks where fission gases are stored open. Consequently, although such fission gases were actually released earlier in the power ramp, the pressure transducer registers the gases in one burst. A burst of fission gases is expected to occur in the event of pellet-clad mechanical interaction. However, no such interaction took place during the Risø AN3 power ramp. The fission gas burst is controversial and currently not well understood [55]. An alternative explanation suggests the burst is caused by gases accumulated at grain boundaries. Given the active research and discussion in this area, it is safe to say the fission gas release model implemented in Bison is not capable of exactly reproducing experimental observations. Indeed, no fission gas release burst mechanism exists in the Bison code. Such shortcomings will have to be considered when performing calibration on fission gas parameters.

4.2 Theory of Fission Gas Release

Initial attempts to create models describing fission gas release were based on empirical correlations. Analyses performed with such models are limited in scope since they depend on a limited set of experimental observations. Simulations with conditions set outside the realm of the experimental data are extrapolating and therefore unreliable, especially when it is acknowledged that the input parameters to the fission gas release models have high uncertainty. To this end, a physics-based fission gas release model was developed by Pastore et. al [56]. Physics based models are based on the solution of partial differential equations describing the various processes in fission gas release such as fuel swelling and diffusion of gas bubbles to the rod free volume.

There are two primary physical processes that lead to fission gas release into the rod free volume. Both processes involve the formation of Xenon and Krypton gas bubbles along UO_2 grain boundaries. In athermal gas release, which occurs at relatively low operating reactor temperatures, the diffusion of gas bubbles from within the fuel grains is negligible. However, in athermal gas release bubbles along fuel grain surfaces are able to escape into the rod free volume through the release mechanisms of recoil and knockout [55]. At high temperatures fission gas release is dominated by thermal release, which involves the coupling of several physical processes. In thermal release Xenon and Krypton gas atoms formed within fuel grains diffuse towards the grain boundaries through trapping-in and irradiation-induced resolution [56]. Additional gas bubbles appear onto grain boundaries through sweeping due to grain growth. As more and more gas bubbles diffusion towards grain boundaries the bubbles experience growth and ultimately coalesce. Eventually, tunnel networks form that provide a means for fission gases to escape into the rod free volume. The internal pressure causing bubble growth can be subdued by hydrostatic stress from the rod internal pressure and pellet/cladding contact pressure. A primary motivation for developing a physics-based fission gas release model is to accurately describe the interplay between these opposing forces [56].

The physics-based models for athermal and thermal fission gas release are given mathematical treatments in the proceeding sections. The implementation of the physics-based fission gas release model by Pastore et. al., namely SIFGRS, is implemented in the Bison code and utilized in this research. The implementation is validated in [55], [57], [60], and [58]. Note that the models on which the SIFGRS implementation is based on is kept relatively simple in order to provide timely estimates of fission gas release in the computationally expensive, finite-element based Bison code. The relative simplicity of the models is justified considering the large uncertainties in both fuel performance modeling codes and model parameters in the fission gas release model [57] [46].

The primary challenges of modeling thermal fission gas release are in describing the diffusional process of fission gases to grain boundaries and the resulting bubble growth and coalescence. As mentioned previously, the thermal release model is intended to be relatively simple, and as such, makes several assumptions that are in contrast to the current understanding of materials science. Upon reaching a fuel grain boundary fission gas atoms can be thrust back into the grain by irradiation. Such behavior is neglected in the proceeding model. Further, the complexity of grain edges, most notably triple grain junctions, are completely neglected. In addition, the initial concentration of grain face bubbles is static, which is not entirely correct because new bubbles will form during any irradiation process. Bubbles are also assumed to absorb any gas atoms arriving at grain faces. Finally, all grain face bubbles are assumed to have the same size and shape [55].

Fission gas bubble growth is known to result from two channels. First, growth occurs due to absorption of vacancies caused by a pressure differential between the bubble's expansion and external forces acting on the bubble. Second, bubble growth is accelerated due to the diffusional process bringing gas atoms from the inner fuel grain to the grain faces. To this end, bubble growth rate can be described as,

$$\frac{dV_{gf}}{dt} = \omega\frac{dn_g}{dt} + \Omega_{gf}\frac{dn_v}{dt} \tag{4.1}$$

where V_{gf} is the bubble volume, n_v is the number of vacancies per bubble, and n_g is the number of fission gas atoms per bubble. The first term in Eq. 4.1 describes the contribution from diffusional gas atoms while the second term describes absorption of vacancies. The factors ω and Ω_{gf} are respective proportionality constants commonly described in the literature as Van Der Waal's' volume of a fission gas atom and the vacancy volume in grain boundary bubbles. The term due to diffusional gas atoms is obtained by solving the diffusion equation in spherical geometry for the concentration of intra-granular gas atoms C_{ig},

$$\frac{dC_{ig}}{dt} = \frac{b}{b+g} D_s \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial C_{ig}}{\partial r} \right) + \beta. \tag{4.2}$$

The effective diffusion coefficient in Eq. 4.2, which regulates the rate at which gas atoms in a UO_2 lattice diffuse from fuel grain to face, consists of several components. The factor g represents the trapping parameter while b is the resolution parameter. Together, the ratio $b/(b+g)$ is the fraction of all intra-granular gas atoms that can potentially diffuse towards the grain boundary [56]. The final component in the effective diffusion coefficient is the intra-granular gas atom diffusion coefficient D_s, which carries the classic interpretation of a diffusion coefficient. The source term in Eq. 4.2 is the rate at which gas is generated.

Changes in vacancy absorption and emission, the second term in Eq. 4.1, is proportional to the pressure differential of the gas bubble. If n_v is the vacancy number in a bubble then the time rate of change of n_v is given by,

$$\frac{dn_v}{dt} = \frac{2\pi D_v \delta_g}{kTs} \left(p - p_{eq} \right). \tag{4.3}$$

The internal pressure of the gas bubble can be expressed using Van Der Waal's equation,

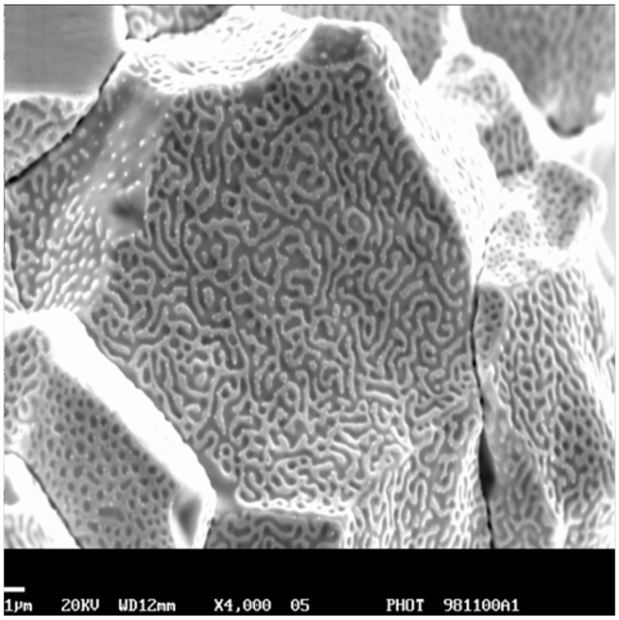
$$p = \frac{kT}{\Omega_{gf}} \frac{n_g}{n_v} \tag{4.4}$$

where k is Boltzman's constant and T is temperature. The pressure p_{eq} acting on the bubble is given as the difference between the bubble surface tension and the hydrostatic stress σ_h of the surrounding medium. More formally,

$$p_{eq} = \frac{2\gamma}{R_{gf}} - \sigma_h \tag{4.5}$$

where R_{gf} is the bubble radius of curvature. The factor γ is the surface tension, which is also commonly referred to as the UO_2/gas specific surface energy. The surface tension parameter strongly affects the kinetics of bubble growth [70]. To solve for the vacancy

Figure 4.6: Bubble coalesce on a fuel grain face from the AGR/Halden Ramp Test Programme [75].



number in Eq. 4.3 the grain boundary diffusion layer thickness δ_g is needed in addition to the vacancy diffusion coefficient D_v. The factor s in Eq. 4.3 is closely related to the grain face fractional coverage [55]. At this point all the elements are in place to model bubble growth using Eq. 4.1.

As bubbles increase in size along grain boundaries they will eventually run into each other and coalesce, as exemplified in Fig. 4.6. Under the premise of conservation of total bubble volume, the relationship between bubble number density N_{gf} and the projected bubble area along the grain face A_{gf} can be expressed as [56],

$$\frac{dN_{gf}}{dt} = -\frac{6N_{gf}^2}{3+4n_{gf}A_{gf}}\frac{dA_{gf}}{dt}. \tag{4.6}$$

The bubble number density is obtained by solving 4.1. The processes of bubble growth and coalescences along fuel grain boundaries lead to fission gas release once certain saturation criteria have been met. Let $F_c = N_{gf}A_{gf}$ denote the fraction of a grain face covered by bubbles. The saturation condition holds when the time rate of change of F_c is static. When the saturation condition holds, the relation between bubble number density and projected grain face area can be reformulated as,

$$\frac{dN_{gf}}{dt} = -\frac{N_{gf}}{A_{gf}}\frac{dA_{gf}}{dt}. \tag{4.7}$$

Only when the saturation condition has been reached can fission gas escape into the rod free volume. Once the saturation condition is met, Eq. 4.8 can be used to model the concentration of gas atoms released per unit grain face.

$$\frac{d\psi_{thr}}{dt} = n_g \frac{1}{2}\frac{2N_{gf}A_{gf} - 3}{4N_{gf}A_{gf}}\frac{dN_{gf}}{dt} \tag{4.8}$$

where n_g is the number of fission gas atoms per grain face bubble. The authors in [56] argue that the process of fission gas release works strictly to reduce the bubble gas content and size but not concentration.

While 4.8 is useful for quantifying fission gas release, a more useful metric is the concentration of gas atoms release per unit fuel volume C_{thr}. The metric C_{thr} is related to ψ_{thr} through the proportionality constant equal to the grain surface to volume ratio [57], as expressed in 4.9.

$$\frac{dC_{thr}}{dt} = \frac{3}{r_{gr}}\left(1 - P_f\right)\frac{d\psi_{thr}}{dt} \tag{4.9}$$

In Eq. 4.9 the factors r_{gr} and P_f represent the fuel grain radius and fuel porosity, respectively. Both of these parameters are omnipresent in fuel performance modeling and have a significant effect on fission gas release values. As the fuel grain radius increases during irradiation the grain surface to volume ratio decreases. Consequently, fuel grain faces generally are less able to hold fission gases [55]. In addition, an increase in fuel grain radius implies a larger travel distance for the fission gases to reach the grain face, per Eq. 4.2. Furthermore, in the phenomenon of grain boundary sweeping additional fission gases are swept to grain faces resulting from fuel grain growth, and specifically, moving grain boundaries. The fuel grain radius is the determinant in governing the fraction of intra-granular gas atoms get swept to the boundaries, as evidenced in Eq. 4.10.

$$f = \frac{r^3_{gr,i} - r^3_{gr,i-1}}{r^3_{gr,i}} \tag{4.10}$$

The i in Eq. 4.10 refers to the i^{th} time-step since grain boundary sweeping makes time-dependent contributions.

In the Bison implementation of the fission gas release model described above, integral fission gas release values are reported. Being a finite element code, Bison computes the gas release at each integration point. The ratio of total fission gas released into the rod free volume to the total fission gas generated at each integration point is the quantity of interest in this thesis. Also, note that since Bison makes time-dependent calculations and the SIFGRS model accepts parameter values such as temperature from the greater code, how to go about a parametric or sensitivity analysis is not straight forward. To this end, scaling factors have been introduced into SIFGRS. At each time-step in a Bison simulation the a set of predetermined scaling factors are applied to their corresponding parameters [57]. As discussed in [57], the existing framework in Bison for propagating uncertainty values is not ideal. For example, to accurately asses the affect of temperature on fission gas release the material properties directly determining temperature fields should be parameterized.

As a final note, due to the relatively high temperatures involved in the Risø AN3 power ramp experiment athermal fission gas release is mostly trumped by its thermal counterpart. Nevertheless, for completeness the equations implemented in Bison for modeling the recoil-and-knockout phenomena leading to athermal gas release is included in Eq. 4.11.

$$\frac{dC_{atr}}{dt} = \frac{yF}{4}\left(\frac{S_g}{V}\mu_f + 2\frac{S_t}{V}\mu_{U,ko}\right) \tag{4.11}$$

The parameters y and F are the yield of fission gases and fission rate density, respectively. Also, S_g, S_t and $\mu_{U,ko}$ are the geometric surface area of the fuel, total fuel surface area, and range of higher-order knock-on in uranium oxide [57].

4.3 Kriging-based Surrogate for Parameter Calibration

In this section Bison is used to model the Risø AN3 power ramp fission gas release time series. More specifically, given the wide uncertainty in the input fission gas release parameters the problem of identifying parameter values resulting in Bison predictions most similar to experimental data is tackled. Since a single Bison simulation of the Risø AN3 power ramp is a computationally expensive, the surrogate construction techniques described in chapter 2 will be employed. Recall that calibration exercises require thousands of instances of computer code simulations. Consequently, without the use of surrogates parameter calibration would not be feasible in this problem. Of the two surrogate construction techniques described, namely Kriging and the collocation approach, Kriging is more applicable for the problem in hand. Both surrogate techniques are designed for modeling surrogates of scalar quantities. However, in this problem a surrogate for an entire time series is desired. Of course, a surrogate can be constructed at each time-step but this would be extremely expensive considering there are $\mathscr{O}(100)$ time-steps. Kriging is used here mainly because it

Table 4.2: Fission gas release parameters used for calibration along with their uniform probability distributions.

Description	Symbol	Lower Bound	Upper Bound	Scaled
Initial Fuel Grain Radius	$r_{g,0}$	2.0E-6	15.0E-6	no
Fuel Porosity	P_f	0.0	0.1	no
Surface Tension	γ	0.5	1.0	no
Temperature	T	0.95	1.05	yes
Fuel Grain Radius	r_g	0.4	1.6	yes
Vacancy Diffusion Coef.	D_v	0.1	10.0	no
Resolution Parameter	b	0.1	10.0	no
Intra-granular Diffusion Coef.	D_s	0.316	3.162	no

can most easily be extended when only a set and limited number of Bison simulations can be afforded.

4.3.1 Uncertain Parameters

In order to calibrate certain parameters it's necessary to first assign a valid range of values each parameter can potentially take on. In light of previous fission gas release parameter sensitivity research in [57], [70], and [37] a total of eight parameters have been chosen for calibration in this research due to their propensity for influencing fission gas release behavior. The eight parameters are the initial fuel grain radius, fuel porosity, bubble surface tension, temperature, fuel grain radius, intra-granular gas atom diffusion coefficient, vacancy diffusion coefficient and resolution parameter. Each of the parameters is assumed to carry a uniform uncertainty distribution with lower and upper bounds estimated by Pastore et. al. [70] [57] using experimental values cited in published literature. The distributions are summarized in Table 4.2.

The scaled column in Table 4.2 denotes whether or not the parameter is scaled at each time-step in a Bison simulation, as described in Section 4.2. Temperature T is a ubiquitous field parameter in Bison that gets passed into the SIFGRS model, appearing most notably in Eq. 4.4 and 4.3 along with the vacancy diffusion coefficient D_v. The bubble surface tension parameterizes how internal bubble gas pressure behaves and appears in Eq. 4.4. The fuel grain radius appears in the grain boundary sweeping model 4.10 along with the equation describing the rate at which fission gases are released into the rod free volume in Eq. 4.9. In addition, the fuel grain radius r_g is a boundary condition in the gas diffusion process in Eq. 4.2. The parameters b and D_s also appear in Eq. 4.2.

4.3.2 Principal Component Analysis

The problem at hand involves a time-dependent objective function, namely the fission gas release fraction throughout the course of a power ramp. Considering most surrogate methodologies, including the ones of interest in this thesis, are designed to handle single objective functions at a time, constructing a surrogate for an entire time series is problematic since a surrogate must be constructed at every time step. Such is especially the case when one does not know a priori whether or not a given time series will contain interesting features mid-cycle, such as jumps and peaks in the objective function. In addition, if pertinent experimental time series data is available and a calibration study is desired, a surrogate model for the entire time series will be necessary. Time dependency is not an issue if one is only interested in investigating, for example, beginning of life or end of life behavior. PCA will be used in this thesis as a framework to create an efficient mapping between fission gas release parameters in SIFGRS and the Risø AN3 power ramp fission gas release time series.

4.3.2.1 Theory

As with many of the great ideas in linear algebra, the premise of PCA rests on a change of basis. PCA attempts to represent some original data samples in terms of a set of basis vectors that reduce redundancy and noise in the data. To this end, consider a matrix $\mathbf{X}$ of n observations and k variables where the k variables have been rescaled by their respective mean, as shown in Eq. 4.12. Rescaling by the mean will ensure that the projected data will live around the centroid of the new basis vectors.

$$\mathbf{X} = \begin{pmatrix} | & | & \cdots & | \\ X_1 & X_2 & \cdots & X_k \\ | & | & \cdots & | \end{pmatrix} \tag{4.12}$$

The problem PCA solves is that of choosing a set of expansion coefficients $\{p_{1j}\}_{j=1}^{k}$ such that,

$$\mathbf{Y}_1 = \mathbf{p}_1^T \mathbf{X}^T = p_{11}X_1 + p_{12}X_2 + \cdots + p_{1k}X_k \tag{4.13}$$

captures the largest variance in the data set. In other words, $\mathbf{Y}_1$ will point in the direction of largest variance. To bound the potential values of $\mathbf{p}_1$ the condition $\|\mathbf{p}_1\|_2 = 1$ is enforced. Since it is unlikely that $\mathbf{Y}_1$ will capture all the variance in the data, PCA goes on to find $\mathbf{Y}_2, ..., \mathbf{Y}_k$ such that all the variance in the data is accounted for. Each $\mathbf{Y}_j$ is independent from the other $\mathbf{Y}_{i \neq j}$ to make sure there is no redundancy in capturing variance. Each $\mathbf{Y}_j$ is referred to as the j-principal component. In matrix form, the workings of PCA result in,

$$\mathbf{Y} = \mathbf{P}^T\mathbf{X}^T. \tag{4.14}$$

From Eq. 4.14 it becomes clear that the operator $\mathbf{P}$ consisting of $\{p_{ij}\}_{i,j=1}^{k}$ has the effect of rotating the data in $\mathbf{X}$ onto an uncorrelated set of axis.

While the desired effect of operator $\mathbf{P}$ to produce output $\mathbf{Y}$ has been described, the question of how to find the expansion coefficients comprising $\mathbf{P}$ remains. As described in [66], the coefficients can be shown to be the loadings of the eigenvectors of the covariance matrix for $\mathbf{X}$. The columns of $\mathbf{P}$ are the eigenvectors of the symmetric matrix,

$$\begin{pmatrix} \sigma_{1,1} & \sigma_{1,2} & \cdots & \sigma_{1,k} \\ & \sigma_{2,2} & \cdots & \sigma_{2,k} \\ & & \ddots & \vdots \\ & & & \sigma_{k,k} \end{pmatrix} \tag{4.15}$$

where $\sigma_{i,j}$ represents the covariance between random variables X_i and X_j. The eigenvalues of the matrix in Eq. 4.15 represent the amount of variance covered by the respective eigenvector. Before being placed into $\mathbf{P}$, the eigenvectors should be sorted in descending order with respect to eigenvalue.

The utility of PCA lays in the fact that for most data sets the variance can be projected onto $\mathscr{O}(1)$ eigenvectors. To reveal this property the sorted eigenvalues can be plotted consecutively. PCA can be viewed as a tool for reducing the dimensionality of a data set by opting to keep only the first r-principal components since a majority of the variance can be projected onto these components. Indeed, if only r eigenvectors are kept then the projected data can written as,

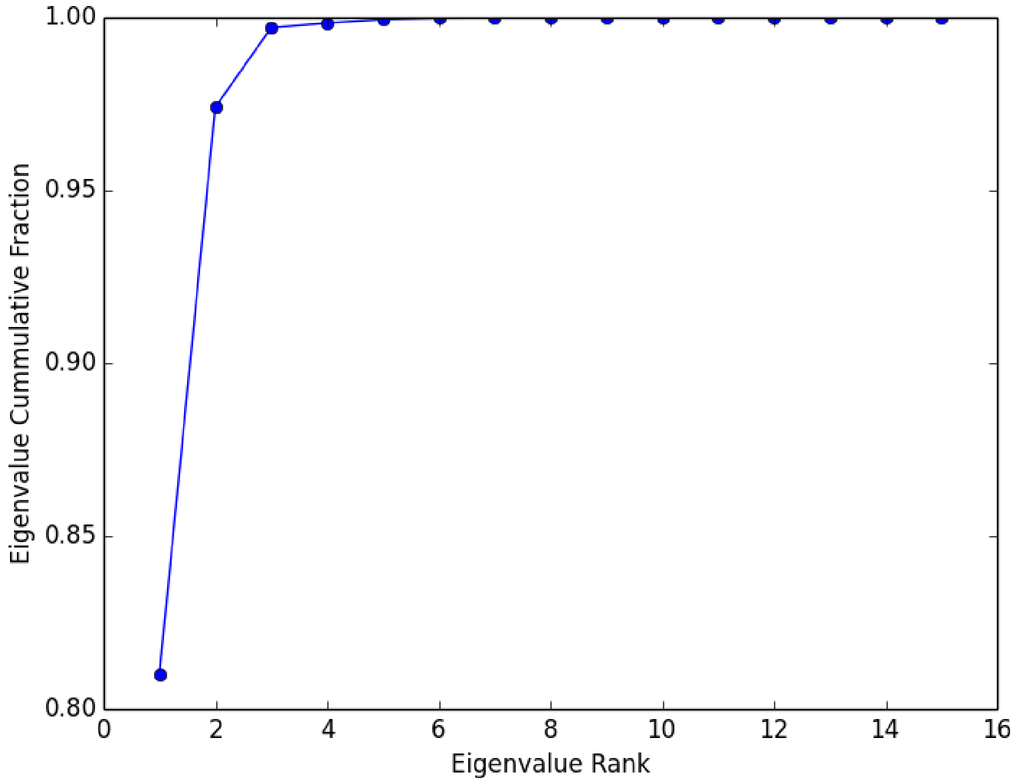
$$\mathbf{Y}_r = \mathbf{P}_r^T\mathbf{X}^T. \tag{4.16}$$

Using Eq. 4.16 the reduced-variance version of the data in $\mathbf{X}$ can be reconstructed as,

$$\mathbf{X}_*^T = \mathbf{P}_r\mathbf{Y}_r \tag{4.17}$$

where it's noted that the inverse of a matrix with orthonormal columns is its transpose. The reconstructed data in Eq. 4.17 represents perturbations around a centroid; the data mean must be added back in to obtain physical values. Using PCA to express a data set in terms of a more meaningful and truncated basis allows one to filter out noise and identify structure in the data. Such possibilities allow one to glean insights into the main contributors of a data set's variance [8].

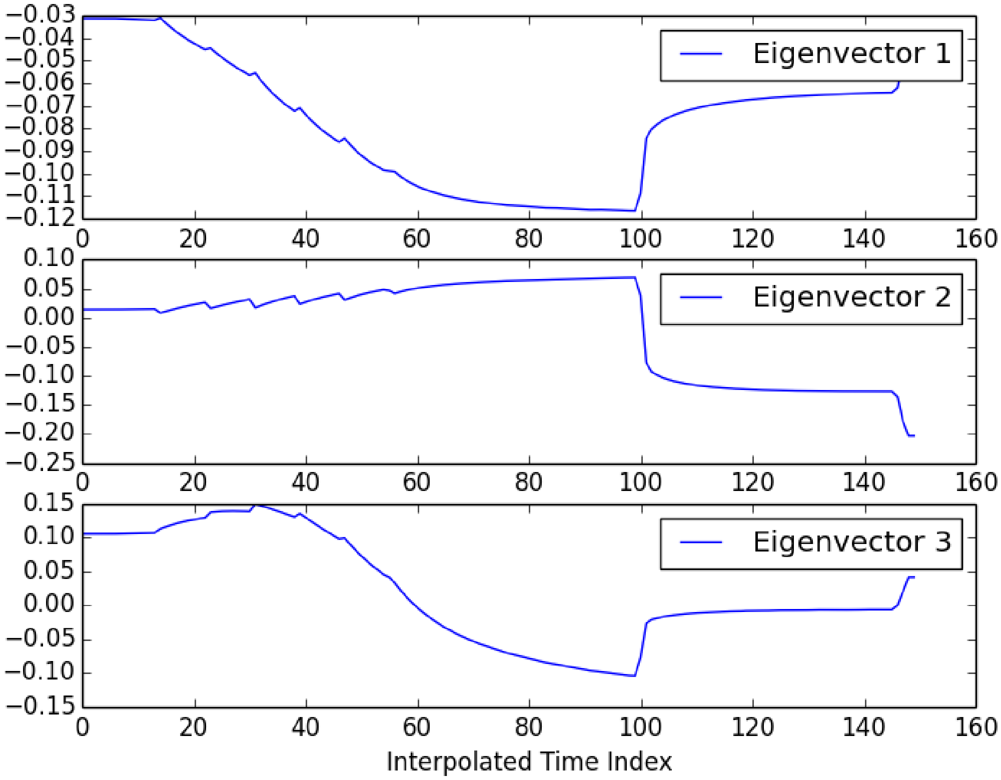
Figure 4.7: Cummulative variance carried by successive eigenvalues in the PCA covariance matrix.



4.3.2.2 Insights

To obtain insight into the contributors of variance in Bison fission gas release time series, PCA is applied to the data shown in Fig. 4.5. Each of the 100 time series shown in Fig. 4.5 is induced by a different set of SIFGRS parameters. Consequently, the time-stepping required for convergence in Bison varies from one time series to another. To perform PCA on the data, fission gas release fractions must be compared at identical times throughout the 100 different samples. Consequently every power ramp time series output by Bison is interpolated using cubic splines and then sampled at 150 evenly spaced points. The covariance matrix central to PCA in this case contains covariances between all 150 time steps. The eigenvalues of the covariance matrix are depicted in Fig. 4.7. As seen in Fig. 4.7, the three largest eigenvalues account for over 99% of the variance. If the original time series data is rotated onto the corresponding three principal components using Eq. 4.16, each time series can be represented using only three expansion coefficients. Consequently,

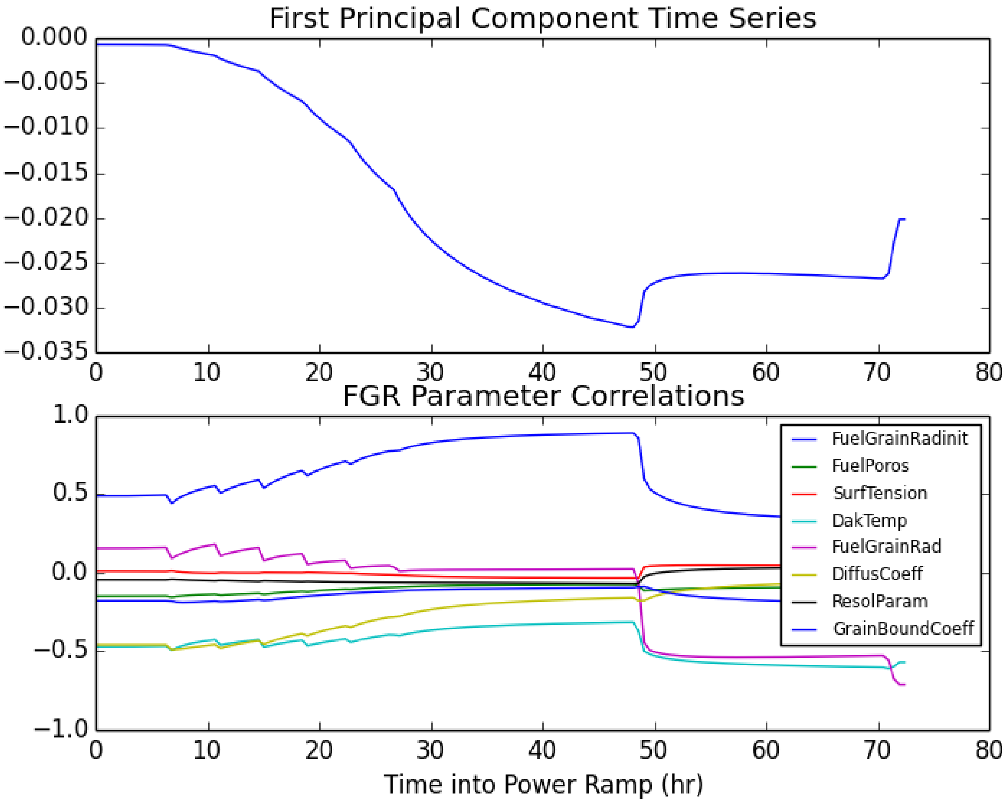
Figure 4.8: First three principal components of the time series covariance matrix.



surrogates can be constructed for only the three expansion coefficients as a function of the SIFGRS parameters. Instead of having to construct a surrogate at every time step, the principal component expansion coefficient surrogates act as a mapping to the entire time series.

The three principal components corresponding to the three largest eigenvalues of the time series covariance matrix are plotted in Fig. 4.8. The principal components enable insight into an underlying stochastic process by observing the magnitude of their coefficients [8]. Each time index's magnitude in a principal component represents its influence on the component. From Fig. 4.8 it appears as though the first principal component is strongly influenced by the times in the middle of the power ramp, the second principal component is influenced by the times at the end of the power ramp and the third principal component is influenced most by the early stages of the power ramp. Further insights can be gleaned by plotting the principal components as time series and correlating the loadings with the LHS values for the SIFGRS parameters as in Fig. 4.9. The initial fuel grain radius

Figure 4.9: Time series of the first principal component and correlations with SIFGRS parameters.



appears to be the leading driver of the variance in the first principal component judging by its high correlation in the middle of the power ramp. Observe how in the late stages of the power ramp, the temperature and grain radius become primary contributors.

4.3.3 Time-Series Surrogate

In Section 4.3.2 PCA was used used to show that three eigenvectors are sufficient to capture over 99% of the variance in Bison's simulated Risø AN3 power ramp fission gas release time series. The variation is caused by perturbations in the fission gas release parameters although they are not explicit in the PCA framework. The Dakota code is used to create the 100 time series in the previous section. In accordance with each parameter's uniform distribution described in Table 4.2, Dakota made 100 sets of perturbations to the parameters using the LHS method. Each of the 100 samples was then propagated through Bison and a time series of fission gas release was output. PCA was then performed on the covariance matrix of the 100 samples. Consequently, there is a clear mapping from a set of eight fission gas release parameters to a time series. More specifically, there is a mapping from the ith set of eight parameters R^i to the three expansion coefficients $\{p_{i1}, p_{i2}, p_{i3}\}$ that are capable of reproducing the time series, as described in Eq. 4.13. In order to predict new time series for a set of fission gas release parameters not in the set used to derive the three principal components this mapping must be generated. Kriging is used to achieve the mapping using the procedure summarized in Fig. 4.10. The procedure in Fig. 4.10 is an adaptation of the traditional surrogate construction process for scalar outputs of expensive computer codes summarized in Fig. 2.1 and is one of the original contributions of this thesis.

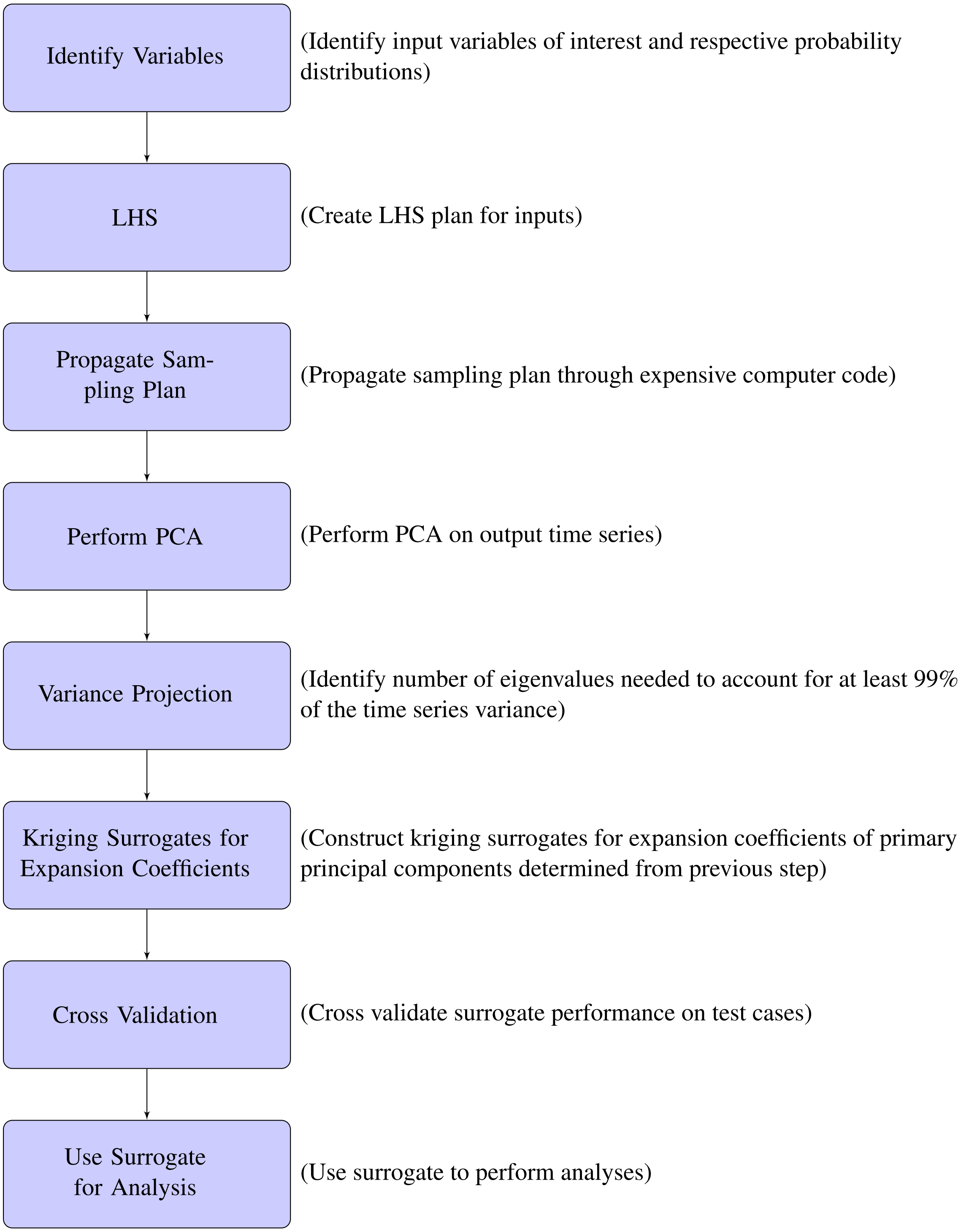
To start, using the Kriging description in Section 2.2 a surrogate is constructed for each of the expansion coefficients related to the three principal components responsible for over 99% of the variance. Each of these surrogates $\hat{p}_{ij}$ for $j \in (1,2,3)$ accepts a set of fission gas release parameters and outputs a scalar expansion coefficient for the principal component X_j. In accordance with Eq. 4.13 the predicted fission gas release time series is,

$$\hat{\mathscr{F}}^i(R^i) = \hat{p}_{i1}\left(R^i\right)X_1 + \hat{p}_{i2}\left(R^i\right)X_2 + \hat{p}_{i3}\left(R^i\right)X_3 + \mu \tag{4.18}$$

where μ is the mean release time series of the 100 simulations. Since Kriging surrogates return an uncertainty $\sigma_{\hat{p}_{ij}}$ along with a predicted scalar value, an uncertainty band can be derived for the time series in Eq. 4.18.

$$\sigma^2_{\hat{\mathscr{F}}^i} = \sigma^2_{\hat{p}_{i1}}X_1^2 + \sigma^2_{\hat{p}_{i2}}X_2^2 + \sigma^2_{\hat{p}_{i3}}X_3^2 \tag{4.19}$$

Figure 4.10: Flow diagram for constructing a time series surrogate using PCA and kriging for an expensive computer code.


Identify Variables
(Identify input variables of interest and respective probability distributions)
LHS
(Create LHS plan for inputs)
Propagate Sampling Plan
(Propagate sampling plan through expensive computer code)
Perform PCA
(Perform PCA on output time series)
Variance Projection
(Identify number of eigenvalues needed to account for at least 99% of the time series variance)
Kriging Surrogates for Expansion Coefficients
(Construct kriging surrogates for expansion coefficients of primary principal components determined from previous step)
Cross Validation
(Cross validate surrogate performance on test cases)
Use Surrogate for Analysis
(Use surrogate to perform analyses)

The shape of the uncertainty vector output by Eq. 4.19 is equal to the number of time-steps utilized in the principal components.

4.3.4 Analysis

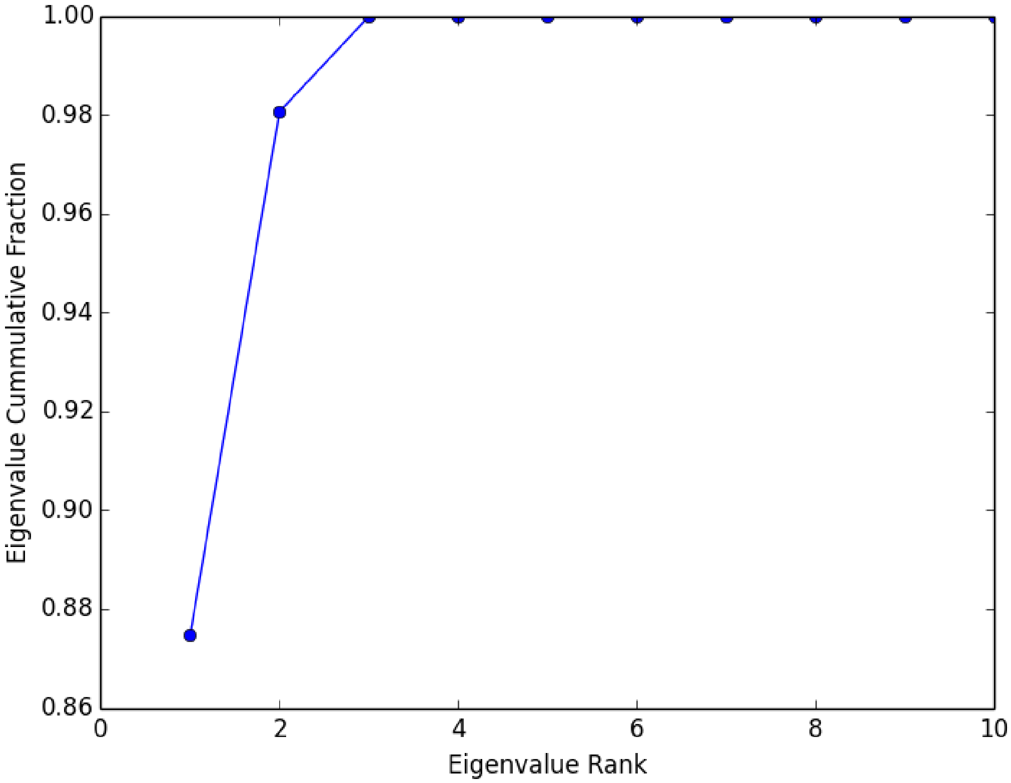
The time series surrogate formulation described in Section 4.3.3 is tested on the point kinetics problem in Section 3.2. Although maximum fuel temperature was studied in Section 4.3.3 the raw output of the solution to the system of differential equations in Eq. 3.6 - 3.12 yields a time series of reactor power, as depicted in Fig. 3.4. Since a time series solution can be obtained relatively quickly for any given set of input parameters the point kinetics problem in hand is ideal for investigating how sensitive the time series surrogate formulation is to training size. Indeed, the training size determines the quality of principal components and the accuracy of the kriging surrogates. The purpose of this section is analyze the trade-off between surrogate accuracy and the expense induced by additional expensive computer code simulations.

Due to the sensitivity analysis conducted in Section 3.2 the time series surrogate formulation will only be tested on the non-kinetics parameters in Table 3.4 since the kinetics parameters were deemed trivial. Specifically, the surrogates were built for the ten variables Λ, Ah, M_c, M_f, c_{pc}, c_{pf}, v, α_d, α_c and ρ_{max}. LHS was utilized to sample the parameter space to generate the design grid on which the principal components and kriging surrogates are calculated. Design grids are created for training set sizes of 15, 20, 30, 50, 75, 100, 200, 300 and 500 samples.

First, the minimum number of eigenvalues needed to accurately represent the variance in the point kinetics time series is investigated. After all, if the number of eigenvalues is on par with the number of time-steps there is no point in applying the time series surrogate formulation described in Section 4.3.3. Using 100 samples, the eigenvalues of the time series covariance matrix are found and sorted. The cumulative sum of the eigenvalues, which corresponds to the fraction of total variance resulting from the 100 samples explained by the corresponding eigenvectors, is plotted in Fig. 4.11. Note, each eigenvector consists of 500 time steps. From Fig. 4.11 it's clear that the three largest eigenvalues can explain over 99% of the variance in the 100 samples. Consequently, the 500 dimensional space initially faced with in the point kinetics time series can be reduced to only three dimensions with minimal loss of explainability.

The three principal components corresponding to the top three eigenvalues are plotted for training sizes of 20, 75, and 200 in Fig. 4.12 to ensure the shapes of the principal components do not change. It is important that the fundamental shapes of the principal compo-

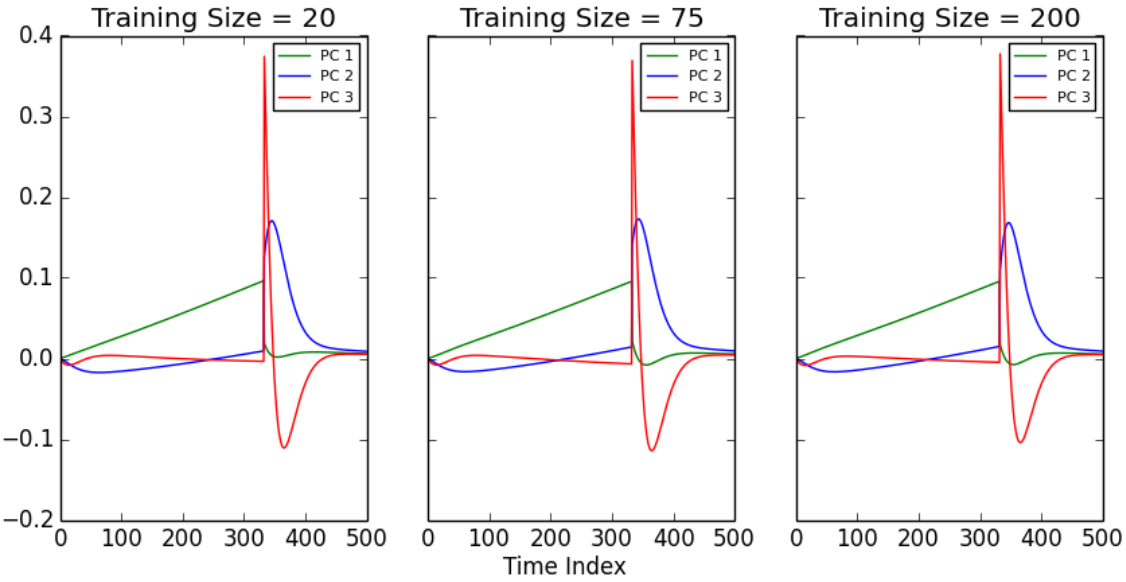
Figure 4.11: Cumulative eigenvalue contribution for point kinetics power ramp using 100 samples.



nents do not change significantly with training size because then surrogates for expansion coefficients values cannot be constructed since they are predicting different objects. For deterministic computer codes, as those investigated in this thesis, the shapes of the principal components are expected to remain relatively static with training set size. Contrarily, the magnitudes of the principal components are expected to change as each principal component is expected to account for more variance introduced with increased training size. Indeed, such a phenomenon is observed in Fig. 4.12. The first principal component in this problem closely follows the normalized reactor power while the second and third principal components correspond to the variance presented following the decrease in power once the sawtooth reactivity insertion ends.

To analyze how well the kriging surrogates are capable of predicting the principal component coefficients, the actual expansion coefficients are plotted against the predicted coefficients in Fig. 4.13 for various training set sizes. The test set in this case consists of 50 LHS sampled independently of those using the training sets. Note, the same test set is used for all proceeding analyses. As the training set size increases in Fig. 4.13 the rela-

Figure 4.12: Top three principal components for the point kinetics time series as calculated using various training set sizes.



tionship between predicted and true expansion coefficients is gradually rotated to 45°. For relatively low training sizes the predicted expansion coefficients are nearly zero, the mean value of the kriging surrogate. Part of the reason for this is the algorithm utilized for optimizing the θ values in the kriging construction. Although a global optimizer should be used a local optimization algorithm is generally used in R [62] and Python statistical packages for efficiency. Regardless of the sample size in Fig. 4.13 the surrogates struggle with predicting extreme expansion coefficients, which for this problem have magnitudes in the regions of -0.08 and 0.08. Since the expansion coefficients are normally distributed, as in Fig. 4.14, there simply are not as many training samples in these regions to provide accurate predictions. However, the kriging formulation is aware of this situation and accounts for the problem by outputting a relatively large uncertainty for these extreme expansion coefficients.

The performance of the time series kriging formulation, with uncertainties for the expansion coefficients accounted for, is gaged in Fig. 4.15 by utilizing the standardized cross-validated residual in Eq. 4.20.

$$\frac{p_{ij} - \hat{p}_{ij}}{\sigma_{\hat{p}_{ij}}}. \tag{4.20}$$

Since the expansion coefficients are normally distributed, some 99% of the values' standardized values should lay between -3 and +3. From Fig. 4.15 points lay outside the bounds

Figure 4.13: Comparison of predicted and true principal component expansion coefficients for the point kinetics problem using 45° plots.

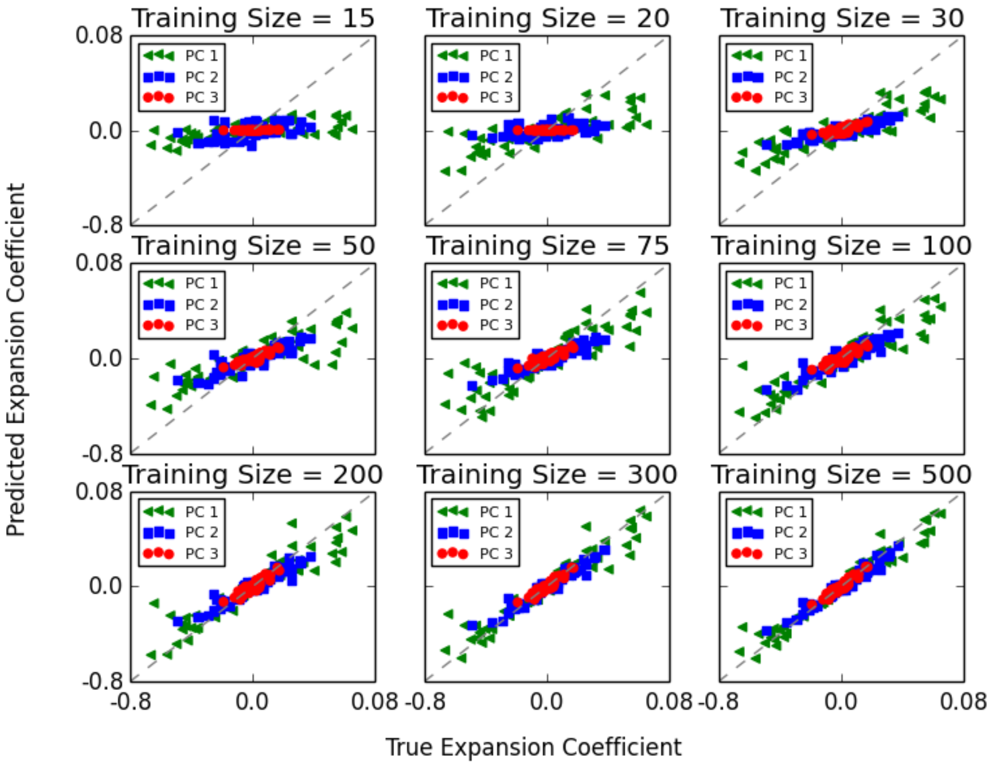
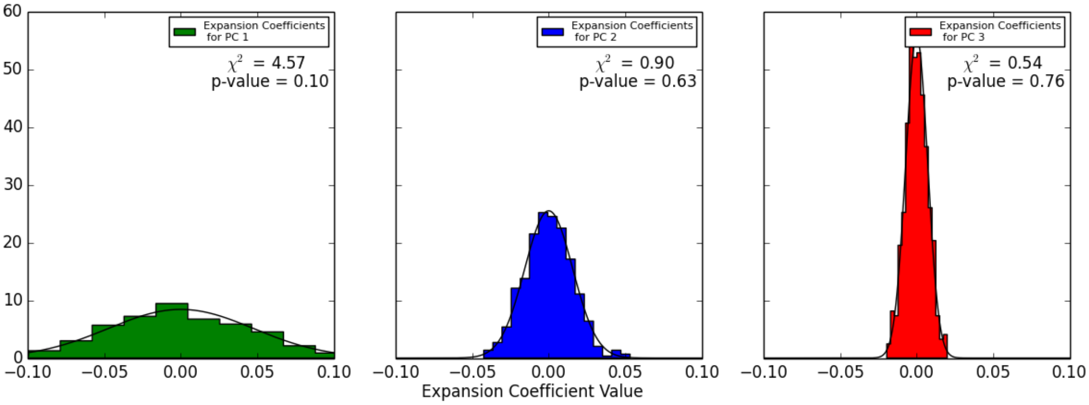
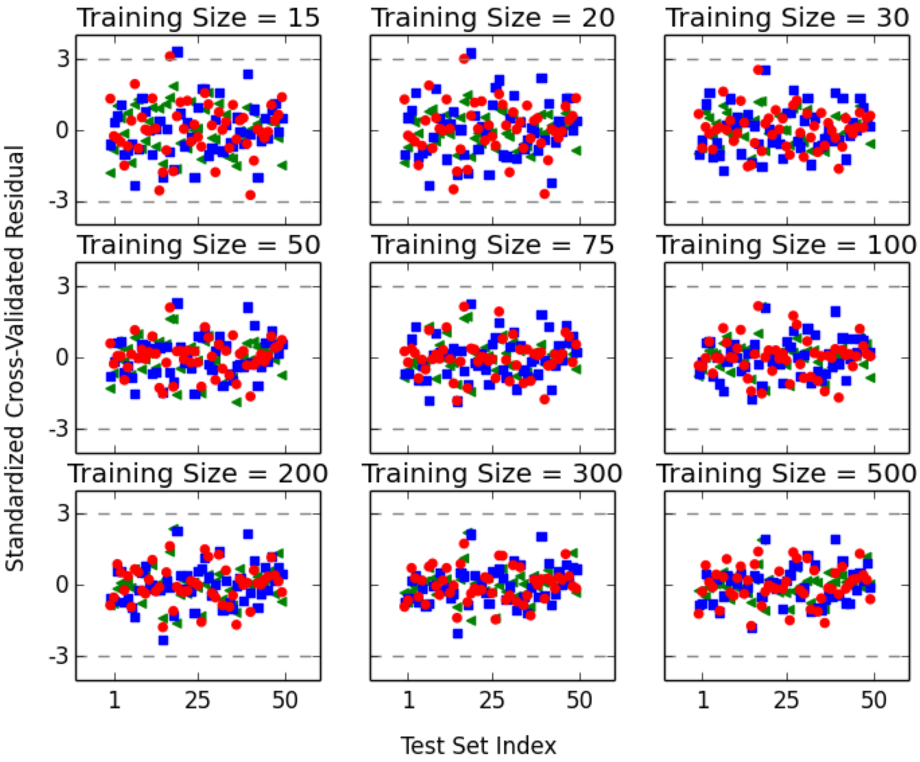


Figure 4.14: Distributions of point kinetics principal component expansion coefficients for a training size of 500 samples.



only for the small train sizes consisting of 15 and 20 samples. Indeed, for small train set sizes the uncertainty predictions are likely to be misrepresented. Recall, kriging predictions are initialized by the mean of the objective function in the training set and then error terms are added based on the proximity of the desired inputs to those in the training set. As expected, the standardized residuals decrease with increasing train set size. To visualize this trend better the absolute value of the standardized residuals are plotted against train set size in Fig. 4.16. After some 100 samples in the training set there appears to be only marginal gains in test set performance. Fig. 4.16 indicates that the first principal component for the point kinetics time series surrogate is predicted best followed by the third and second principal components. Since each successive principal component captures less variance than its predecessors it is most important to accurately predict the principal components corresponding to the largest eigenvalues.

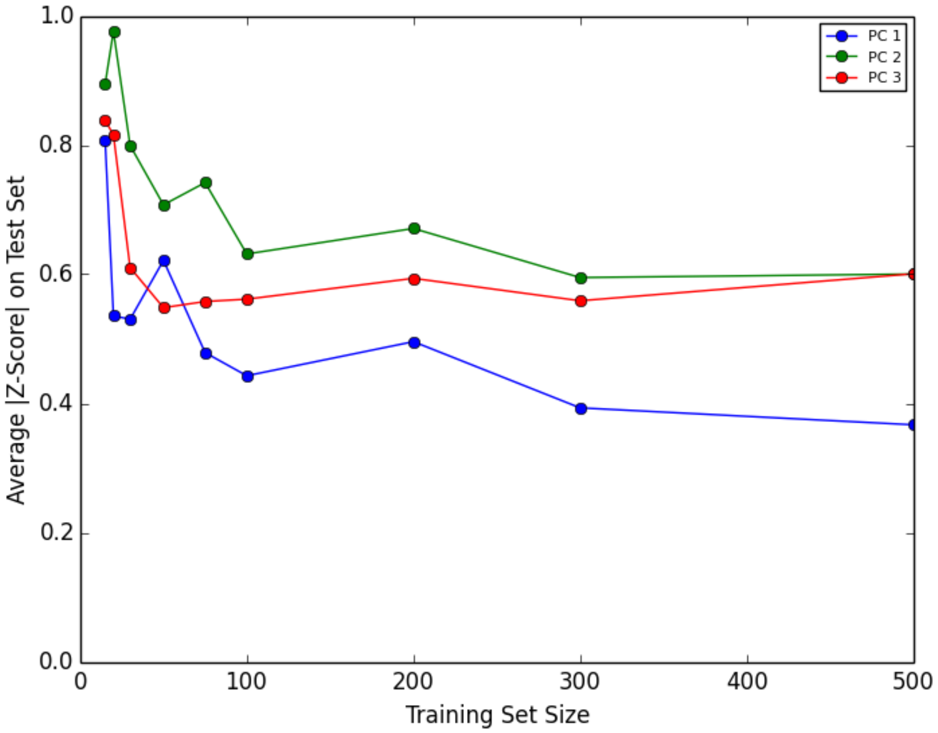
Figure 4.15: Cross validation of the point kinetics time series surrogate by examining standardized cross-validated residuals.



4.3.4.1 Cross Validation

With a Kriging formulation for predicting Risø AN3 power ramp fission gas release time series for any set of Bison fission gas release parameters in place, it is necessary to test the formulation. In other words, it's necessary to investigate the error in the formulation's

Figure 4.16: Average absolute value of standardized cross-validated residuals for the point kinetics time series surrogate for various train set sizes.



predictions. If the Kriging surrogates can accurately predict their respective PCA expansion coefficients then the formulation will be able to accurately predict gas release time series since Eq. 4.18 is just a linear combination of the predictions. To this end, a test set of 100 independent Bison simulations of the Risø AN3 power ramp were obtained. The simulations in the test set are completely isolated from those in the training set and not used in calculating the Kriging surrogates nor the principal components. There are several common approaches to judging the validity of a predictive model as described in [38]. With training and test sets available at disposal are true expansion coefficient values, predicted values, and uncertainties in the predicted values.

The standardized cross-validated residuals for all three Kriging surrogates' predictions on the test set are shown in Fig. 4.17. Note the Kriging models used to predict the expansion coefficients are built on 100 samples analyzed previously in Sec. 4.3.2. Each point represents the number of standard deviations the true expansion coefficient is to the predicted coefficient. Some 99.7% of points are expected to lie within the band $[-3, +3]$. Indeed, of the 300 points appearing in Fig. 4.17 some six lie outside the bands, which is a first indication that the predictive model has merit.

Figure 4.17: Cross validation of Kriging predictions for PCA expansion coefficients using 3σ band approach.

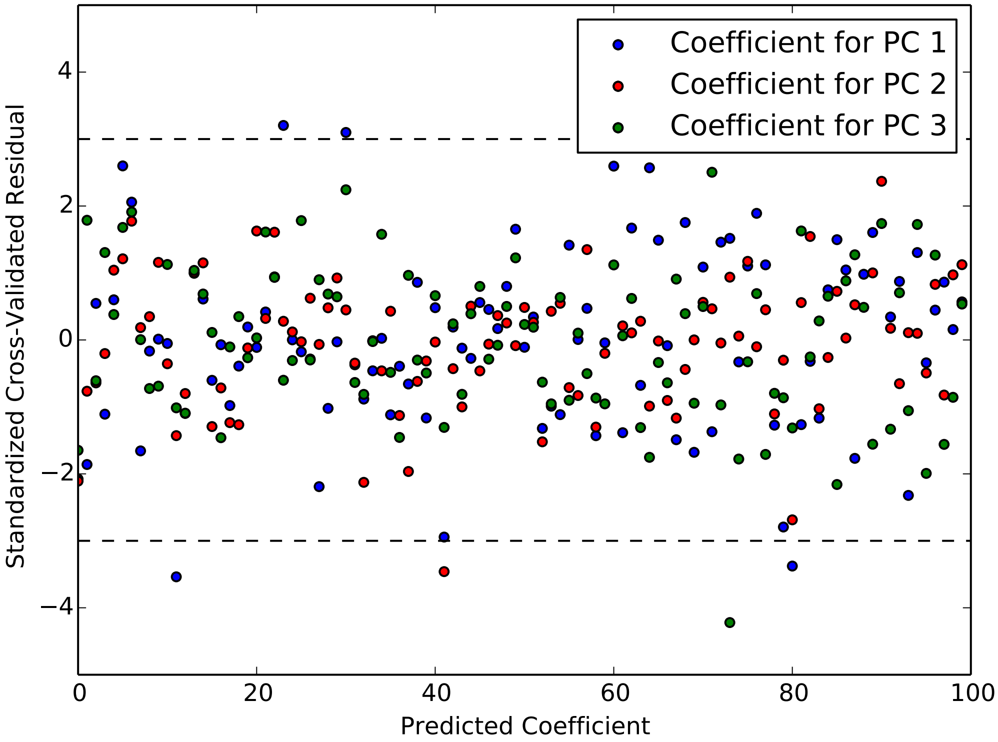
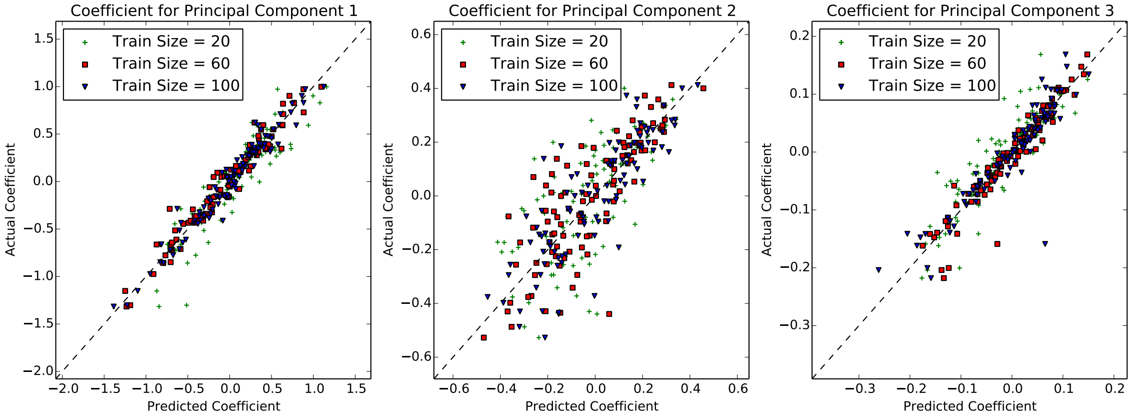


Figure 4.18: Cross validation of Kriging predictions for PCA expansion coefficients using 45° approach.



In the second cross validation approach, the predicted values are plotted directly against the true values as shown in Fig. 4.18. Ideally, all points would lie on the 45° line, which would indicate that predictions perfectly match the true expansion coefficient values. The plots in Fig. 4.18 show that the predicted values do generally approach this trend despite the presence of some noise. For predictions of each expansion coefficient in Fig. 4.18 the results are plotted when 20, 60, and 100 random samples from the test set are utilized. Of course, less noise is present as the number of samples used to build the surrogates increase. Surprisingly, regardless of the training set size, predictions for the first principal component expansion coefficient are of the highest quality followed by the predictions for the expansion coefficients of the third principal component. It is most important to accurately predict expansion coefficients for the top principal components since they explain the most variance. Accurate prediction of lower principal components offers diminishing returns. Both inverse and logarithmic transforms were applied to the data as suggested in [38] although no noticeable improvement was observed in prediction accuracy. Due to the relatively noisy values for the second expansion coefficient in Fig. 4.18 it is anticipated the largest prediction errors will occur from the release jump and onwards since the second principal components attempts to account for a majority of the variance in this area per Fig. 4.8.

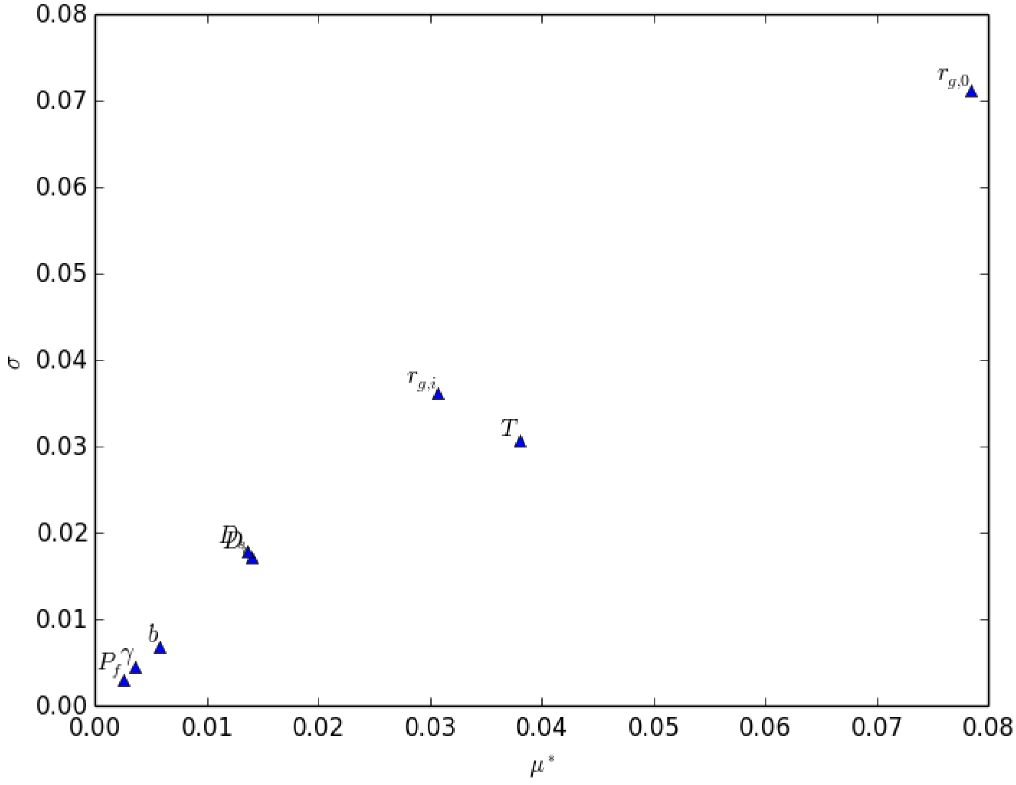
4.3.5 Calibration

Now that the predictive accuracy of the individual Kriging surrogates for each expansion coefficient have been cross validated it is time to investigate how well the surrogates can perform collectively in predicting Risø AN3 power ramp fission gas release time series. To determine the error between predicted time series the RMSE is used as a cost function. Recall that before any two time series are compared each is segmented into 150 identical locations. The objective of calibration is to find a set of fission gas release parameters R^i such that when they are input into Eq. 4.18 the RMSE is minimized.

Ideally, the landscape of possible RMSE values is such that a clear global minimum exists. In this case, an algorithm such as Efficient Global Optimization (EGO) can be used to find the minimal value and its corresponding parameter set [38]. EGO couples a computer model's surrogate prediction and uncertainty with the computer model itself in an iterative optimization process. At each iteration EGO estimates the parameter space location that is most likely to contain a RMSE value smaller than the current minimum. The true computer model is then evaluated at this point and therefore, the minimization procedure can be potentially expensive. However, if it is known a priori that the RMSE landscape is for example, convex then EGO should be strongly considered. Unfortunately, there is usually no way to tell a priori the shape of a landscape for most expensive engineering codes such as Bison. Due to the non-linearities inherent in such codes the landscape is often fairly ”flat”. A flat landscape can result when some parameters are insignificant and thus, many values of these parameters will result in similar cost function values. Also, a flat landscape can result if the minimization problem is non-unique, meaning many combinations of parameters can yield the same, or nearly the same, optimal values for the RMSE. Often times, these two issues may be interrelated and both will tremendously complicate the global optimizer search. Indeed, if EGO is used many expensive computer code evaluations will be wasted. A local optimization algorithm should be utilized if a flat landscape is suspected.

To decide which class of minimization algorithm to apply to solve the problem in-hand Sobol indices are calculated using the surrogate model. However, Morris' algorithm is first applied to get a sense of which of the eight fission gas release parameters in Table 4.2 most influence the RMSE. In reference to Section 2.1.1, 500 elementary effects are calculated for each fission gas release parameter. A total of 4500 evaluations of the surrogate in Eq. 4.18 were required to produce the plot in Fig. 4.19. The value μ^* in Fig. 4.19 is simply the average of the absolute values of each parameter's elementary effects, as defined in Eq. 2.1. Taking the absolute value helps to ensure cancellation effects in non-monotonic models do not obscure a parameter's true influence [52]. The initial fuel grain radius is clearly the most influential parameter followed by the fuel grain and temperature scaling parameters. Such results are consistent with the variable importances discovered through the PCA study that resulted in Fig. 4.9. The importance of fuel grain radius parameters is of no surprise given how omnipresent they are in the SIFGRS model, and specifically in

Figure 4.19: Morris' algorithm applied to the RMSE of the fission gas release time series.



determining diffusion distances and grain boundary sweeping. The positive distances away from the origin along the σ-axis are indicative of non-linear interaction effects between the fission gas release parameters and a first sign that a global minimum solution may be too computationally expensive for the calibration problem.

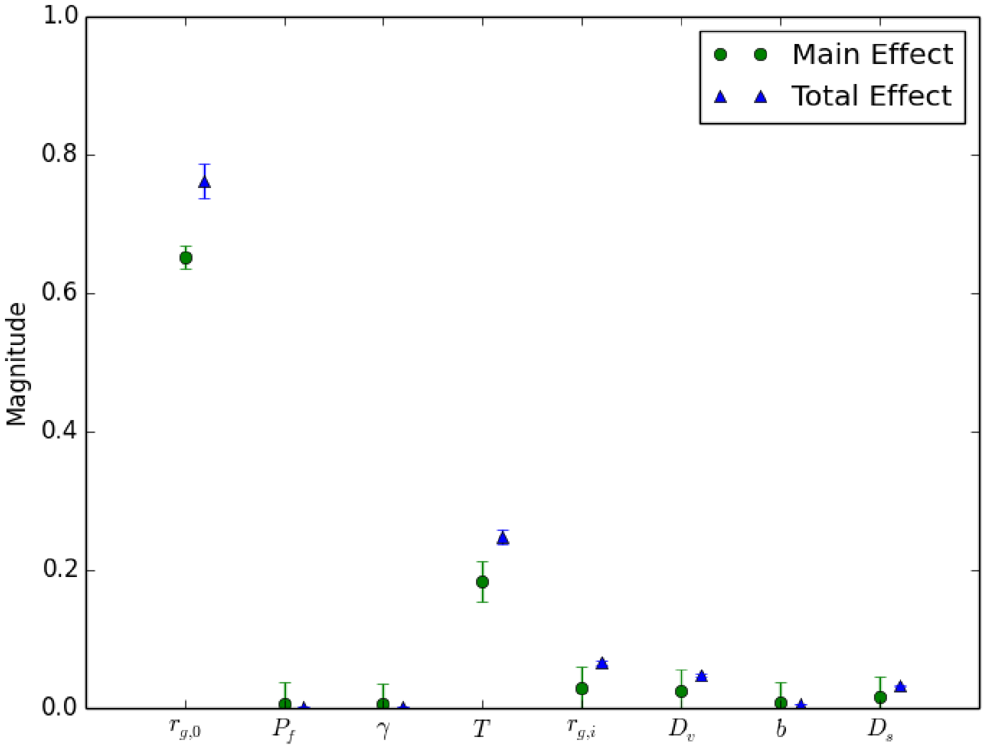
Further investigation of the sensitivity of the RMSE with respect to its input parameters is achieved by calculating Sobol indices. As described in [68], Sobol indices attribute a system's total variance to individual variables and their interactions with other variables by applying the anchored-ANOVA decomposition described in Section 2.3.2. In reference to Eq. 2.16, the "main effect index" is defined to be the variance of $f_{\mathbf{u}}(\mathbf{x_u})$ normalized by the system's total variance. The main effect determines the fraction of variance that is due to either one variable or any number of interacting variables. The anchored-ANOVA frameworks allows for the subtraction of variance contributions from other variables. If the main effect indices are added for all combinations of a systems d variables then the $2^d - 1$ components are expected to sum to unity.

However, the integrals involved in calculating the variance of $f_{\mathbf{u}}(\mathbf{x_u})$ are expensive and therefore, typically main effect indices are only calculated for the individual variables and not their interactions [64]. The ”total effect index” uses the anchored-ANOVA decomposition to estimate the variance due to a variable and all of its possible interactions with other variables in the system. The calculation of total effect indices is made possible thanks to the total variance theorem [64]. Unlike the sum of all main effect indices, the sum of all total effect indices is expected to sum to unity only when the model under consideration is purely additive.

Various methods exist for computing Sobol indices, all of which are based on the Monte Carlo method for estimating the integrals in the variance calculations when the model under consideration is not analytic. The R code package [62] is utilized for calculating Sobol indices for the problem in hand. Specifically, the method of Sobol and Jansen [64] is used initially to estimate main and total effect indices. The Sobol and Jansen algorithm requires $n(d+2)$ evaluations of the objective function to estimate both sets of indices, where n is the number of Monte Carlo samples to use. Both main and total effect indices for the RMSE are displayed in Fig. 4.20 using 10^5 Monte Carlo samples and 100 bootstrap samples to obtain 95% confidence intervals. Note, a total of 10^5 calculations for the RMSE were required to produce the data in Fig. 4.20. Without a surrogate model, the Bison code would have to be executed this same number of times, which would be prohibitive. The main effect magnitudes for each of the eight fission gas release parameters is entirely consistent with Fig. 4.19. The far greatest variance is due to the initial fuel grain radius followed by the temperature and fuel grain radius scaling factors. The intra-granular and vacancy diffusion coefficients have non-zero but relatively small Sobol sensitivity indices. However, P_f, γ, and b contribute trivially to any variability in the RMSE. From Fig. 4.20 it is evident that the parameters with non-trivial main effect indices also have significant total effect indices. Consequently, higher-order parameter interactions play a significant role in determining the RMSE. Such a discovery is no surprise given how, for example, tightly coupled the temperature and fuel grain radius are in the SIFGRS formulation described in Section 4.2.

In order to get a sense for which fission gas release parameters are strongly interacting, second-order Sobol indices are calculated in R. As seen in Fig. 4.21 the highest magnitude interaction effects are between parameters with the largest main effects from Fig. 4.20. The largest interaction variance is due to the initial fuel grain radius and temperature scaling factor. Relatively large interactions that contribute to RMSE variance are between the initial fuel grain radius and all the other fission gas release parameters. Of particular note is the magnitude of interaction between the vacancy diffusion coefficient and the temperature and initial fuel grain radius. A total of 3.7×10^5 RMSE calculations were required to produce Fig. 4.21. As mentioned previously, such an analysis is only feasible using an

Figure 4.20: Main and total effect indices for the RMSE using the Sobol-Jansen algorithm. A total of $n = 10^4$ Monte Carlo samples are used to estimate the indices.

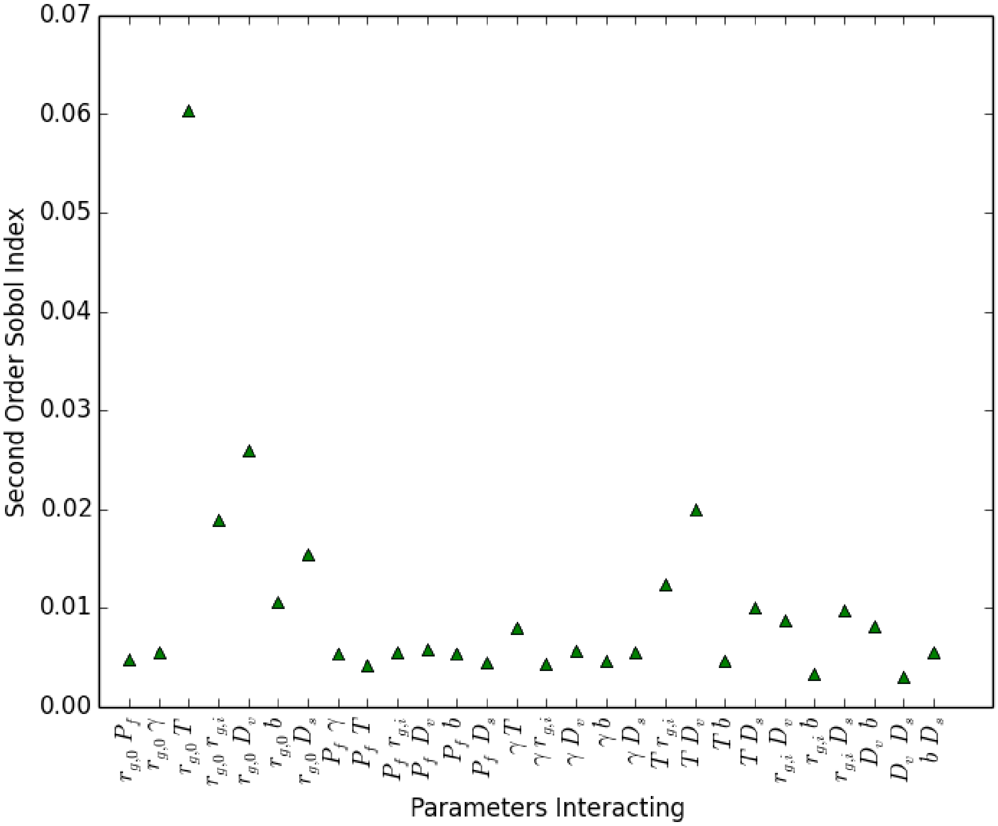


efficient surrogate.

Given the presence of strong interaction among the fission gas release parameters it is unlikely a global RMSE minimum can be found in reasonable time. However, a locally optimal solution can be found using the COBYLA algorithm. The COBYLA algorithm is of the simplex variety that does not require any gradient information while allowing for constraints to be placed on both the search parameters and objective function. For this problem it is necessary to not allow fission gas parameters that result in negative fission gas release values in predicted time series. Simplex algorithms are based on the fact that parameter constraints act as hyperplanes in d-dimensional space that collectively enclose a convex volume. The optimal objective function value must lay on one of the vertices of the intersection of the hyperplanes [61].

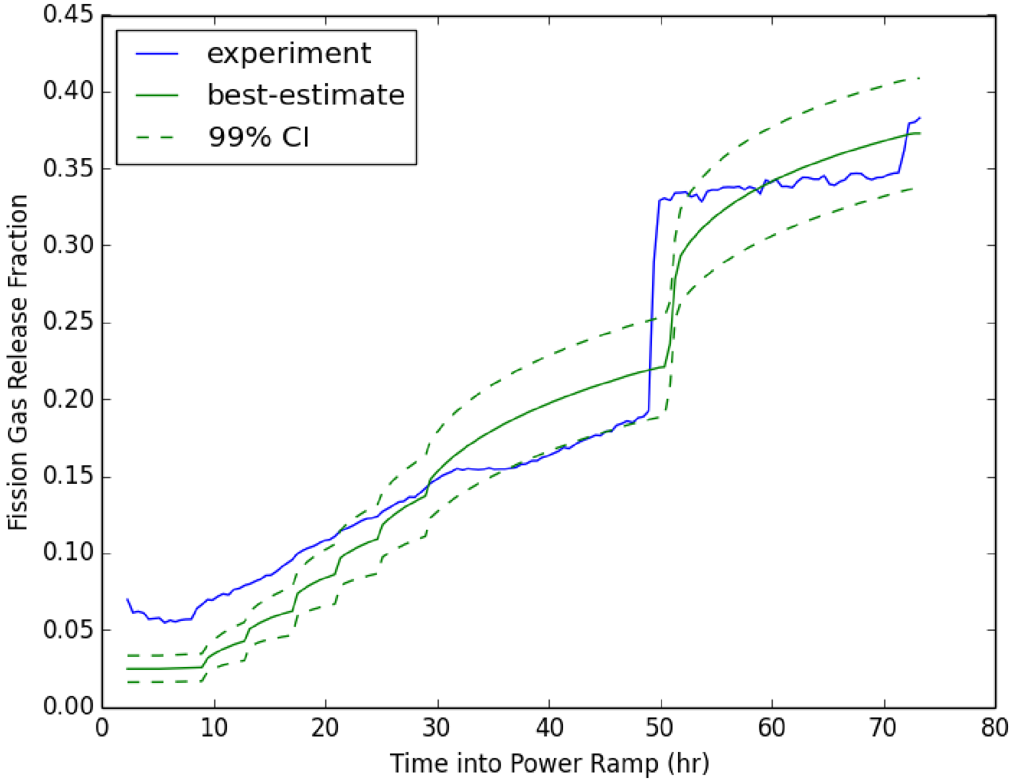
Since local optimization algorithms such as COBYLA are notorious for being sensitive to initial search conditions the algorithm is executed 100 different times, with each execution being seeded by one of the 100 LHS used to construct the expansion coefficient surrogates. Such a procedure increases the probability of finding a true minimum RMSE

Figure 4.21: Second order Sobol indices for the RMSE. A total of $n = 10^4$ Monte Carlo samples are used to estimate the indices.



and not one existing in a flat space. The minimum RMSE found was 0.02941, which corresponded to the predicted time series in Fig. 4.22. To identify the locally minimum RMSE some 10^5 instances of the surrogate were required. For each of the 100 seedings the COBYLA algorithm was terminated either after 2×10^3 iterations or a relative difference of 10^{-6} between two consecutive iterations. While the time series in Fig. 4.22 is certainly an improvement over the time series produced when the mean fission gas parameter values are used in Fig. 4.3, the predicted fission gas release time series leaves much to be desired. Discrepancies between predicted and experimental values may safely be attributed to the way in which gas release is modeled in SIFGRS. The optimal parameter values, with each parameter scaled to the unit hypercube, found for each of the 100 COBYLA seedings is summarized in a boxplot in Fig. 4.23. The length of the whiskers in Fig. 4.23 implies strong non-linear interaction effects, as substantiated by the investigation involving Sobol indices. The relative difference between the maximum RMSE and minimum RMSE among all 100 optimized seedings was only some 35%. The relatively low differences between the

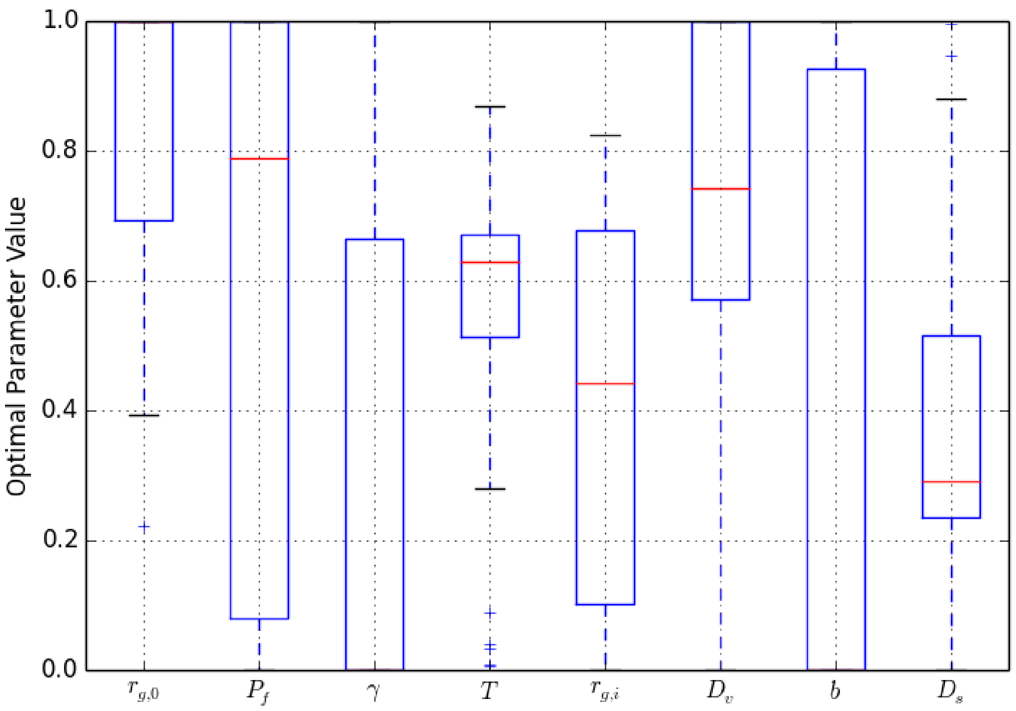
Figure 4.22: Best-estimate fission gas release time series compared to experimental data.



two extremes was achieved using a wide combination of parameter values.

One path taken to lessen the difference between experimental data and prediction was to smooth the experimental data. As mentioned previously, the measurements taken of fission gas release during the Risø AN3 power ramp may contain significant error. At some points in the experimental data the total fission gas release actually decreases, which is completely not physical. However, with only a single time series measurement there is no way of assigning uncertainties to fission gas release values measured at different times throughout the power ramp. Also, it would be unfounded to assign an uncertainty of say, 10% to all measurements taken during the power ramp. Consequently, in an attempt to smooth the experimental data local polynomial regression was applied with the minimal amount of smoothing necessary to make the fission gas release time series strictly monotonically increasing. Using the R ”loess” function with linear interpolants and a spanning parameter of 0.17 the desired smoothing was achieved. The smoothed experimental data along with the best-fit surrogate prediction is shown in Fig. 4.24. The RMSE for the best-estimate time series when compared to the smoothed experimental data was calculated to be 0.02486,

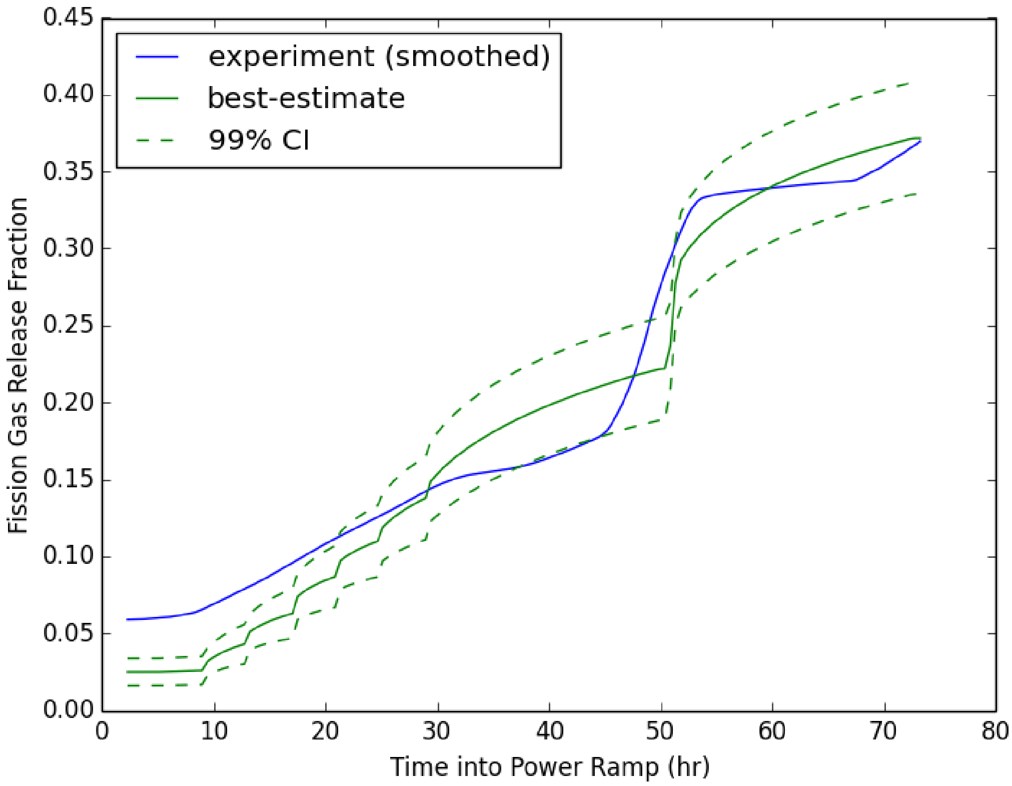
Figure 4.23: Boxplot of optimal parameter combinations for 100 seedings of the COBYLA algorithm.



which was a 15.5% reduction in RMSE from when raw experimental data was utilized.

Comparing Fig. 4.24 with Fig. 4.22, it is not clear whether the smoothed experimental data offers any advantages towards finding a set of Bison fission gas release parameters that best predict the available data. Although there are significant discrepancies between predicted and experimental time series, especially in the power burst occurring at hour fifty of the power ramp, the Bison predictions do a good job in predicting end-of-experiment fission gas release values. For the case when raw experimental data is used, as in Fig. 4.22, there is only a 2.6% relative error in the end-of-experiment fission gas release prediction. The beginning-of-experiment prediction error in this case is 64.8%. For the case of smoothed experimental date the prediction results are marginally improved with a beginning-of-experiment prediction error of 57.8% and an end-of-experiment error of 0.5%. Note, it's possible to enforce the conditions of matching the predicted beginning-of-experiment and end-of-experiment predictions to their respective experimental values in the COBYLA framework. However, the COBYLA algorithm is unable to converge to a solution that matches these conditions. Enforcing only one of the conditions to a tolerance of 10^{-3} was achievable although the resulting solution grossly over predicted the fission

Figure 4.24: Best-estimate fission gas release time series compared to experimental data that was smoothed using the local polynomial regression smoothing.



gas release elsewhere in the time series. A table of calibrated fission gas release parameters for both raw and experimental data is summarized in Table 4.3.

Table 4.3: Calibrated fission gas release parameters with respect to raw and smoothed experimental data.

Description	**Symbol**	**Smoothed**	**Raw**	**Relative Difference**
Initial Fuel Grain Radius	$r_{g,0}$	1.50E-05	1.50E-05	0.0E+00
Fuel Porosity	P_f	0.10	0.09	8.2E-02
Surface Tension	γ	0.50	0.50	1.4E-04
Temperature	T	1.02	1.02	1.8E-04
Fuel Grain Radius	r_g	1.26	1.26	5.1E-03
Vacancy Diffusion Coef.	D_v	7.49	7.50	3.8E-04
Resolution Parameter	b	0.10	0.19	4.8E-01
Intra-granular Diffusion Coef.	D_s	1.02	1.05	2.7E-02

CHAPTER 5

Summary and Conclusion

5.1 Summary

Advances in computing the past decade have motivated scientists and engineers to write computer code that models physical phenomenon using as few approximations as possible. While the predictive capabilities of such codes' simulations improve in representing experimental data, they typically require hours on parallel compute clusters to complete. Indeed, from a statistical vantage point the analysis of computer experiments landscape has not changed. Engineering computer codes are still expensive to execute and therefore, techniques must be used to somehow simplify the codes before analyzing them. Newly available computational power has been spent towards making individual computer codes more accurate and encompassing rather than on easing the ability to perform optimization, calibration and sensitivity analysis in hopes of gleaning insight into the physics at hand. The purpose of this thesis has been to apply and develop so-called surrogate methods towards the statistical analysis of expensive engineering computer codes when the inputs to the codes are uncertain.

Two surrogate approaches are investigated in this thesis. The first, anchored-ANOVA decomposition on Smolyak sparse grids is a relatively new approach researched in the 2000s. Kriging is the second approach taken to analyze the output of computer experiments and has been considered the bread-and-butter of the field since the 1950s. In Kriging, a computer code is treated as a stochastic process. For various inputs, the computer code's outputs are observed and a statistical model is built based on the distance between input sets and the marginal differences in output. Usually LHS is used to sample computer codes in an optimal fashion when only a limited number of computer simulations can be afforded. The anchored-ANOVA decompositions takes a more deterministic approach towards surrogate building. First, a computer code is sliced into components of first order, second order, and higher order combinations of input variables. For each component a polynomial interpolant is built on a Smolyak sparse grid. In the anchored-ANOVA approach one does not have the flexibility to evaluate the computer code at any arbitrary points. Rather, the computer code must be evaluated at points dictated by the collocation set and convergence criteria chosen. Based on the insight that higher order interaction effects are small compared to their first order counterparts, very accurate surrogates can be constructed with relatively few points.

Initially, both surrogate approaches were applied to text-book type problems in order to gain familiarity and to identify which approach would be more suitable when applied to modeling fission gas release using a computationally expensive computer code. In the anchored-ANOVA approach each variable considered requires special attention and consequently, for simple problems with relatively small uncertainties Kriging was observed to build as accurate of surrogate models as anchored-ANOVA using less objective function simulations. Spectral convergence was observed for each text-book problem considered when the anchored-ANOVA approach was applied. However, the number of new simulations of the objective function for each increasing level of Smolyak interpolants grows exponentially. In the classic formulations of both surrogate methods the input parameters are assumed to be independent of one another. For statistical analysis this assumption can be circumvented by building the surrogates with the independent assumption and then sampling the surrogates using a covariance matrix that describes the variables' interconnectedness. The performance of Clenshaw-Curtis and Gauss-Patterson collocation sets were both investigated in application to Smolyak interpolants. For the problems in this thesis, the higher theoretical accuracy of Gauss-Patterson did not show and consequently, Clenshaw-Curtis collocation is recommended due to its transparency and near-optimal performance.

Ultimately, Kriging was chosen as the surrogate approach to apply towards modeling fission gas release in Bison. The decision resulted from consideration on several fronts. The non-linearity of fission gas release models coupled with large uncertainties implied the need for modeling higher-order interaction effects with the surrogate. As indicated by the text-book problems, modeling such higher-order effects with anchored-ANOVA and Smolyak sparse grids can get very expensive, with no clear limit of how many objective function simulations will be needed to complete the surrogate. A Kriging surrogate makes more sense when faced with a limited computational budget. In addition, the transparency of Kriging was appealing when considering each Bison fission gas release simulation would have to be performed in parallel. For Kriging a total of n randomly sampled simulations are needed in order to construct the surrogate and thus, if one simulations fails to converge or experiences an error, the correction process is straight forward. Contrarily, there are a lot of moving pieces in the anchored-ANOVA surrogate approach. Repairing the damages to a surrogate's construction due to a failed simulation requires much book keeping. Finally, Kriging was chosen over the anchored-ANOVA approach for fission gas release due to Kriging's clear extension to time series. Both surrogate methods investigated in this thesis were primarily developed for scalar quantities. It was not clear how the anchored-ANOVA approach could be extended to construct a surrogate for a time series without taking on an immense computational expense.

Before investigating fission gas release kinetics for the Risø AN3 power ramp using Kriging, eight SIFGRS parameters were chosen for analysis. The eight parameters were the initial fuel grain radius, fuel porosity, bubble surface tension, temperature, fuel grain radius, intra-granular gas atom diffusion coefficient, vacancy diffusion coefficient and resolution parameter. Uniform distributions spanning orders of magnitudes were assigned to the parameters in order to reflect their large uncertainties. A total of 100 LHS of the parameters were then taken and propagated through Bison and the Risø AN3 fission gas release model. Principal component analysis was applied to model the variability in the fission gas kinetics output. Only three principal components were necessary to capture over 99% of the variance in the 100 fission gas release time series.

Kriging was extended to produce a surrogate for entire time series by constructing individual scalar Kriging surrogate models for the expansion coefficients of each of the three principal components. The time series Kriging model was then cross validated before applying it to calibrate Bison's SIFGRS parameters to experimental data. Although a locally optimal solutions was found by minimizing RMSE, there were apparent and irreconcilable differences between Bison's fission gas release predictions and the experimental data. While experimental data lays mostly outside of the calibrated fission gas kinetics output's 99% confidence intervals, the end-of-experiment error was under 3%. Some of the differences between prediction and experiment can be attributed to several fission gas release aspects not explicitly modeled in Bison. Namely, burst fission gas release due to microcracking and the effect of measuring fission gas release using pressure transducers. Experiments have identified scenarios where released fission gases get trapped in closed gaps and cracks in the fuel, thereby not contributing to changes in plenum pressure that can be measured by pressure transducers. Another factor contributing to discrepancies between predicted and experimental fission gas release time series is the uncertainty in SIFGRS parameters not modeled in this thesis.

Sobol indices were calculated for the fission gas release parameters using the Kriging surrogate. The starting fuel grain radius and fuel temperature had the highest sensitivity indices and produced the largest non-linear interaction effects with the other parameters. While sensitivity coefficients were calculated for the Risø AN3 problem, it is not certain that the same parameter conclusions would generalize to other fission gas kinetics problems. The same type of analysis as conducted in this thesis would likely have to be replicated for each unique problem. The lack of generalization is likely due to the unique profundity of physics in play for each type of problem.

The original contributions of this thesis are three fold. First, the construction of a surrogate model for the fuel performance code Bison and subsequent calibration of fission gas release parameters to experimental data from the FUMEX database. Second, the extension of Kriging to construct surrogates for entire time series through PCA. Finally, the application of anchored-ANOVA decomposition and Smolyak sparse grids to construct surrogates for classic nuclear engineering problems.

5.2 Suggestions for Future Research

The research conducted in this thesis can be extended in many ways, in both the application and theory of surrogate models. In addition, the results of the research are suggestive of areas in which fission gas release modeling can be improved. The most direct extension of this research is to apply it towards any of the fuel performance models in the FUMEX databases [40] [41] that contain experimental data such as the Halden IFA rods, Risø GE7, OSIRIS and REGATE test cases. Surrogates can be constructed for any of these models using the framework discussed in this thesis. The surrogates can then be folded together with available experimental data to analyze thermal responses during power ramps, fission gas release, fuel pellet swelling and pellet clad mechanical interaction. Surrogates for these models would allow for sensitivity analyses, which would allow modelers to gain valuable insights into underlying physics.

As scientists and engineers begin to write computer codes that are able to better emulate the fundamental physics in the material, thermalhydraulic, and neutronic components of nuclear processes surrogates should eventually be constructed for resulting multiphysics codes. Before such a task is undertaken it is essential to validate the individual codes to see if they are indeed capable of reproducing desired physics. Otherwise, coupling various codes will have no effect on the efficacy of the multiphysics systems and any surrogates built upon it since its success will only be as strong as its weakest constituent. Such a coupled, multiphysics code system is currently being developed and validated in Michigan Parallel Characteristics Transport Code (MPACT) [45]. If a coupled code system is more accurate than any of its constituent pieces individually then a surrogate model for the multiphysics system should be constructed. Performing optimization and calibration with such a surrogate and experimental data should yield more accurate parameter analyses. However, it should be noted that a more accurate objective code will not alleviate a resulting surrogate from the problems that have classically plagued surrogates. Namely, the existence of many local optima. The extension of surrogates to multiphysics computer codes is a worthwhile investigation.

In many of the problems where engineering computer codes are trying to model some physical phenomena there exists sparse experimental data to validate the codes. For example, in this thesis only a single time series measurement existed to measure fission gas release during the Risø AN3 power ramp. With the availability of more experimental data, more interesting and rigorous analyses can be conducted with surrogate models. Error bars on experimental data enable probabilistic inferences of parameter values in calibration. As in [74], Bayesian hierarchical models can be used to probabilistically infer parameter values and even reconstruct temperature and flux fields. Such Bayesian approaches have the advantage of indicating the likelihood of parameters taking on certain values using posterior probability distributions, which can then be used for probabilistic risk assessment. Given the relatively limited amount of experimental data available in the nuclear engineering field, application of hierarchical Bayes for solving stochastic inverse problems should strongly be considered because it allows for the natural incorporation of multiple data sources. In other words, multiple experimental results, such as both pellet elongation and fission gas release data for the same problem, can be folded together to provide a holistic calibration approach.

Another suggested area of research that would have wide ranging consequences for all engineering fields utilizing surrogate models would be in the development of a systematic method for initially reducing the size of the input parameter space. Despite claims to the contrary, existing surrogate construction methods can only handle $\mathscr{O}(10)$ variables. Unfortunately, most computer codes, and especially multiphysics systems, have hundreds if not thousands of inputs. A method is needed to reduce the initial parameter space to one more suitable for surrogate construction. For the problem in this thesis, the SIFGRS model itself consisted of some dozen input parameters. Modeling the effects of all input parameters into Bison would have been a challenging task. Fortunately, much research had been conducted previously as to which fission gas release parameters generally have a large impact. Consequently, the initially large parameter space faced in this research was narrowed down by a thorough literature review. Ideally a mathematically rigorous method should exist to replace the literature review that is not immensely computationally burdensome.

Finally, as evidenced by the calibrated fission gas kinetics in Fig. 4.22, the Bison code fundamentally predicts fission gas release in a way different than experiment shows. As discussed in [58], the discrepancies are likely due to the trapping and sudden release of fission gases in fuel grain cracks [58]. Efforts should be made to model this phenomena in SIFGRS and the Bison code. Another area of development that may lessen any differences between predicted fuel performance metrics and observed values is the modeling of measurement devices. In this research, both thermocouples and pressure transducers are used to gather the experimental data. These instruments affected the quality of the experimental data and even the shape of the fission gas kinetics. Efforts should be made in Bison to incorporate the effects of instrumentation on predicted values in order for valid comparisons to be made between predicted and experimental data.

APPENDIX A

Bison Input File

The following input file was developed at Idaho National Laboratory. The input models Risø AN3 and attempts to mimic the actual conditions of the test fuel rod during the experimental power ramp and its preceding irradiation history. The same input file is packaged with the Bison code as a validation case.

```
# Set initial fuel density
[GlobalParams]
  density = 10303.0
  disp_x = disp_x
  disp_y = disp_y
  order = SECOND
  family = LAGRANGE
  energy_per_fission = 3.2e-11 # J/fission
[]

# Specify coordinate system type
[Problem]
  coord_type = RZ
[]

# Set problem dimension (2d-rz here) and import mesh file
[Mesh]
  file = riso_an3.e
  displacements = 'disp_x disp_y'
  patch_size = 1000
[]

# Define dependent variables, element order and shape function family, and initial conditions
[Variables]
 [./disp_x]
  [../]

  [./disp_y]
  [../]

  [./temp]
    initial_condition = 513
  [../]
[]

# Define auxillary variables, element order and shape function family
[AuxVariables]
  [./fast_neutron_flux]
    block = '1'
  [../]

  [./fast_neutron_fluence]
    block = '1'
  [../]

  [./grain_radius]
```

```
     block = '3 4'
     initial_condition = 4.000000000000000e-01
  [../]

  [./hydrostatic_stress]
    order = CONSTANT
    family = MONOMIAL
  [../]

  [./stress_xx]      # stress aux variables are defined for output;
    order = CONSTANT
    family = MONOMIAL
  [../]

  [./stress_yy]
    order = CONSTANT
    family = MONOMIAL
  [../]

  [./stress_zz]
    order = CONSTANT
    family = MONOMIAL
  [../]

  [./creep_strain_xx]
    order = CONSTANT
    family = MONOMIAL
  [../]

  [./creep_strain_yy]
    order = CONSTANT
    family = MONOMIAL
  [../]

  [./creep_strain_xy]
    order = CONSTANT
    family = MONOMIAL
  [../]

  [./creep_strain_hoop]
    order = CONSTANT
    family = MONOMIAL
  [../]

 [./vonmises]
    order = CONSTANT
    family = MONOMIAL
  [../]

  [./creep_strain_mag]
    order = CONSTANT
    family = MONOMIAL
  [../]

  [./gap_cond]
    order = CONSTANT
    family = MONOMIAL
  [../]

 []

# Define functions to control power and boundary conditions
[Functions]

  [./power_profile]
    type = PiecewiseLinear   # reads and interpolates an input file containing rod average linear power vs time
    data_file = riso_an3_power_history.csv
    format = columns
    scale_factor = 1
  [../]

  [./axial_peaking_factors]      # reads and interpolates an input file containing the axial power profile vs time
    type = PiecewiseBilinear
    data_file = an3_axial_peaking.csv
    scale_factor = 1
    axis = 1
  [../]

  [./pressure_ramp]             # reads and interpolates input data defining amplitude curve for coolant and fill gas pressure
    type = PiecewiseLinear
    x = '-100 0 103528800 103529000 103794477'
    y = '0 1   1     0.986  0.986'
  [../]

  [./flux]              # reads and interpolates input data defining fast neutron flux
    type = PiecewiseLinear
    data_file = riso_an3_fast_flux.csv
    format = columns
  [../]

 [./clad_temp_bc]
    type = PiecewiseLinear
    data_file = riso_an3_clad_bc.csv
    format = columns
 [../]

  [./q]
    type = PiecewiseLinear   # reads and interpolates an input file containing rod average linear power vs time
    data_file = riso_an3_power_history.csv
    format = columns
  [../]

[]

# Specify that we need solid mechanics (divergence of stress)
[SolidMechanics]
  [./solid]
    disp_r = disp_x
    disp_z = disp_y
    temp = temp
  [../]
[]

# Define kernels for the various terms in the PDE system (in all cases here, the axisymmetric (RZ) version is specified)
[Kernels]

  [./gravity]       # body force term in stress equilibrium equation
    type = Gravity
    variable = disp_y
    value = -9.81
  [../]

  [./heat]         # gradient term in heat conduction equation
    type = HeatConduction
    variable = temp
  [../]

 [./heat_ie]       # time term in heat conduction equation
    type = HeatConductionTimeDerivative
    variable = temp
  [../]

  [./heat_source_]  # source term in heat conduction equation
     type = NeutronHeatSource
     variable = temp
     block = '3 4'                    # fission rate applied to the fuel (block 2) only
     fission_rate = fission_rate  # coupling to the fission_rate aux variable
  [../]
[]

[Burnup]
  [./burnup]
    block = '3 4'
    rod_ave_lin_pow = power_profile
    axial_power_profile = axial_peaking_factors
    num_radial = 80
    num_axial = 11
    a_upper = 0.29334
    a_lower = 0.00734
    fuel_inner_radius = 0
    fuel_outer_radius = 0.0045265
    fuel_volume_ratio = 0.9889
    RPF = RPF
  [../]
[]

# Define auxilliary kernels for each of the aux variables
[AuxKernels]
  [./fast_neutron_flux]
    type = FastNeutronFluxAux
    variable = fast_neutron_flux
    block = '1'
    axial_power_profile = axial_peaking_factors
    function = flux
    factor = 4.9e17
    execute_on = timestep_begin
  [../]

  [./fast_neutron_fluence]
    type = FastNeutronFluenceAux
    variable = fast_neutron_fluence
    fast_neutron_flux = fast_neutron_flux
    execute_on = timestep_begin
  [../]

  [./grain_radius]
     type = GrainRadiusAux
     block = '3 4'
     variable = grain_radius
     temp = temp
     execute_on = residual
  [../]

  [./hydrostatic_stress]                  # include hydrostatic stress for possible use in ForMas
    block = '3 4'
    type = MaterialTensorAux
    tensor = stress
    variable = hydrostatic_stress
    quantity = hydrostatic
    execute_on = timestep
  [../]

  [./stress_xx]               # computes stress components for output
    type = MaterialTensorAux
    tensor = stress
    variable = stress_xx
    index = 0
    execute_on = timestep     # for efficiency, only compute at the end of a timestep
  [../]

  [./stress_yy]
    type = MaterialTensorAux
    tensor = stress
    variable = stress_yy
    index = 1
    execute_on = timestep
  [../]

  [./stress_zz]
    type = MaterialTensorAux
    tensor = stress
    variable = stress_zz
    index = 2
    execute_on = timestep
  [../]

  [./vonmises]
    type = MaterialTensorAux
    tensor = stress
    variable = vonmises
    quantity = vonmises
    execute_on = timestep
  [../]

  [./creep_strain_xx]               # computes stress components for output
    type = MaterialTensorAux
    tensor = creep_strain
    variable = creep_strain_xx
    block = 1
    index = 0
    execute_on = timestep     # for efficiency, only compute at the end of a timestep
  [../]

  [./creep_strain_yy]
    type = MaterialTensorAux
    tensor = creep_strain
    variable = creep_strain_yy
    block = 1
    index = 1
    execute_on = timestep
  [../]

  [./creep_strain_xy]
    type = MaterialTensorAux
    tensor = creep_strain
    variable = creep_strain_xy
    block = 1
    index = 3
    execute_on = timestep
  [../]

  [./creep_strain_hoop]
    type = MaterialTensorAux
    tensor = creep_strain
    variable = creep_strain_hoop
    block = 1
    index = 2
    execute_on = timestep
  [../]

  [./creep_strain_mag]
    type = MaterialTensorAux
    tensor = creep_strain
    variable = creep_strain_mag
    block = 1
    quantity = plasticstrainmag
    execute_on = timestep
  [../]

[]

[AuxBCs]
  [./conductance]
    type = MaterialRealAux
    property = gap_conductance
    variable = gap_cond
    boundary = 10
  [../]
[]

# Define mechanical contact between the fuel (sideset=10) and the clad (sideset=5)
[Contact]
  [./pellet_clad_mechanical]
    master = 5
    slave = 10
    disp_x = disp_x
    disp_y = disp_y
    formulation = penalty
    penalty = 1e10
    model = frictionless
    normal_smoothing_distance = 0.1
  [../]
[]

# Define thermal contact between the fuel (sideset=10) and the clad (sideset=5)
[ThermalContact]
```

```
  [./thermal_contact]
    type = GapHeatTransferLWR
    variable = temp
    master = 5
    slave = 10
    initial_moles = initial_moles       # coupling to a postprocessor which supplies the initial plenum/gap gas mass
    gas_released = fis_gas_released     # coupling to a postprocessor which supplies the fission gas addition
    roughness_clad = 1.0e-6
    roughness_fuel = 2.0e-6
    roughness_coef = 3.2
    plenum_pressure = plenum_pressure
    jump_distance_model = KENNARD
    initial_gas_fractions = '1 0 0 0 0 0 0 0 0 0'
    refab_time = 103529000
    refab_gas_fractions = '1 0 0 0 0 0 0 0 0 0'
    contact_pressure = contact_pressure
    quadrature = true
    normal_smoothing_distance = 0.1
  [../]
[]

# Define boundary conditions
[BCs]

# pin pellets and clad along axis of symmetry (y)
  [./no_x_all]
    type = DirichletBC
    variable = disp_x
    boundary = 12
    value = 0.0
  [../]

# pin clad bottom in the axial direction (y)
  [./no_y_clad_bottom]
    type = DirichletBC
    variable = disp_y
    boundary = '1'
    value = 0.0
  [../]

# pin fuel bottom in the axial direction (y)
  [./no_y_fuel_bottom]
    type = DirichletBC
    variable = disp_y
    boundary = 20
    value = 0.0
  [../]

  [./temp]
    type = FunctionDirichletBC
    boundary = '1 2 3'
    variable = temp
    function = clad_temp_bc
  [../]

  [./Pressure]
#  apply coolant pressure on clad outer walls
    [./coolantPressure]
      boundary = '1 2 3'
      factor = 1.552e7 #changes to 1.53e7 for bump tests
      function = pressure_ramp   # use the pressure_ramp function defined above
    [../]
  [../]

  [./PlenumPressure]
#  apply plenum pressure on clad inner walls and pellet surfaces
    [./plenumPressure]
      boundary = 9
      initial_pressure = 2.31e6 #changes to 1e5 for bump tests
      startup_time = 0
      R = 8.3143
      output_initial_moles = initial_moles       # coupling to post processor to get inital fill gas mass
      temperature = ave_temp_interior            # coupling to post processor to get gas temperature approximation
      volume = gas_volume                        # coupling to post processor to get gas volume
      material_input = fis_gas_released          # coupling to post processor to get fission gas added
      output = plenum_pressure                   # coupling to post processor to output plenum/gap pressure
      refab_time = 103529000
      refab_pressure = 1.57e6
      refab_temperature = 500
      refab_volume = 7.0e-6
    [../]
  [../]

 ]

# Define material behavior models and input material property data
[Materials]
  [./density1]
    type = Density
    block = '1'
    density = 6551.0
  [../]
  [./density2]
    type = Density
    block = '3 4'
  [../]

  [./fuel_thermal]                       # temperature and burnup dependent thermal properties of UO2 (bison kernel)
    type = ThermalFuel
    block = '3 4'
    temp = temp
    burnup = burnup
    model = 4
    initial_porosity = 0.06
  [../]

  [./fuel_solid_mechanics_swelling]      # free expansion strains (swelling and densification) for UO2 (bison kernel)
    type = VSwellingUO2
    block = '3 4'
    temp = temp
    burnup = burnup
  [../]

  [./fuel_creep]                        # thermal and irradiation creep for UO2 (bison kernel)
    type = Elastic #CreepUO2
    block = '3 4'
    disp_r = disp_x
    disp_z = disp_y
    temp = temp
    youngs_modulus = 2.e11
    poissons_ratio = .345
    thermal_expansion = 10e-6
    stress_free_temperature = 297
  [../]

  [./fuel_relocation]
    type = RelocationUO2
    block = '3 4'
    burnup = burnup
    diameter = 0.009053 #Fuel pellet diameter in m
    q = q
    gap = 2.05e-4 #diametral gap in m
    burnup_relocation_stop = 0.029
    relocation_activation1 = 5000 #initial relocation activation power in W/m
  [../]

  [./clad_thermal]                       # general thermal property input (elk kernel)
    type = HeatConductionMaterial
    block = '1'
    thermal_conductivity = 16.0
    specific_heat = 330.0
  [../]

  [./clad_solid_mechanics]               # thermoelasticity and thermal and irradiation creep for Zr4 (bison kernel)
    type = MechZry
    block = '1'
    disp_r = disp_x
    disp_z = disp_y
    temp = temp
    fast_neutron_flux = fast_neutron_flux
    fast_neutron_fluence = fast_neutron_fluence
    youngs_modulus = 7.5e10
    poissons_ratio = 0.3
    thermal_expansion = 5.0e-6
    absolute_tolerance = 1e-12
    output_iteration_info = false
    stress_free_temperature = 297
    model_irradiation_growth = false
    model_thermal_expansion = false
  [../]

  [./clad_irrgrowth]
    type = IrradiationGrowthZr4
    block = '1'
    fast_neutron_fluence = fast_neutron_fluence
  [../]

  [./fission_gas_release]
    type = Sifgrs
    block = '3 4'
    temp = temp
    fission_rate = fission_rate        # coupling to fission_rate aux variable
#   initial_grain_radius = 3.0e-6
    grain_radius = grain_radius
    gbs_model = true
    burnup = burnup
    ramp_model = true
    file_name = riso_an3_power_history.csv
    format = columns
    rod_ave_lin_pow = power_profile
    axial_power_profile = axial_peaking_factors

    temperature_scalef = 9.500000000000000e-01       # scaling factor for temperature
    grainradius_scalef = 4.000000000000000e-01       # scaling factor for grain radius
    igdiffcoeff_scalef = 1.000000000000000e+01       # scaling factor for intragranular diffusion coefficient
    resolutionp_scalef = 1.000000000000000e+01       # scaling factor for resolution parameter
    gbdiffcoeff_scalef = 3.160000000000000e-01       # scaling factor for grain boundary diffusion coefficient
    initial_grain_radius = 4.000000000000000e-01
    initial_porosity = 8.615294515923598e-02
    surface_tension = 9.512108845950570e-01
  [../]
[]

[Dampers]
  [./limitT]
    type = MaxIncrement
    max_increment = 50.0
    variable = temp
  [../]
[]

#[Debug]
# show_var_residual = 'disp_x disp_y temp'
#[]

[Executioner]
  type = Transient

  # PETSC options

  #Preconditioned JFNK (default)
  solve_type = 'PJFNK'
  petsc_options = '-ksp_gmres_modifiedgramschmidt'
  petsc_options_iname = '-ksp_gmres_restart -pc_type  -pc_composite_pcs
                         -sub_0_pc_hypre_type -sub_0_pc_hypre_boomeramg_max_iter
                         -sub_0_pc_hypre_boomeramg_grid_sweeps_all -sub_1_sub_pc_type
                         -pc_composite_type -ksp_type -mat_mffd_type'
  petsc_options_value = '201                 composite hypre,asm         boomeramg            2                                  2

  line_search = 'none'

  # controls for linear iterations
  l_max_its = 100
  l_tol = 8e-3

  # controls for nonlinear iterations
  nl_max_its = 10
  nl_rel_tol = 1e-4
  nl_abs_tol = 1e-10

  # time control
  start_time = -100
  end_time = 103789797
  num_steps = 5000.0
  dtmax = 1e6
  dtmin = 1

  # direct control of time steps vs time (optional)
  [./TimeStepper]
    type = IterationAdaptiveDT
    dt = 1.0e2
    optimal_iterations = 6
    growth_factor = 1.3
    linear_iteration_ratio = 100
    time_t  = '0         103528800 103529000 103529119 103531160
               103533619 103537700 103552640 103567520 103582160
               103596439 103611019 103625239 103701019 103701870
               103781900 103782929'
    time_dt = '2.0e2  1e2 1.0e1 1.0e2 1e2 1e2 1e2 1e2 1e2 1e2 1e2
               1e2 1e2 1e2 1e2 1e2 1e2 '
  [../]

 [./Quadrature]
    order = THIRD
 [../]

[]

# Define postprocessors (some are required as specified above; others are optional; many others are available)
[Postprocessors]
  [./ave_temp_interior]            # average temperature of the cladding interior and all pellet exteriors
     type = SideAverageValue
     boundary = 7
     variable = temp
   [../]

  [./clad_inner_vol]              # volume inside of cladding
   type = InternalVolume
    boundary = 7
    output = file
  [../]

  [./pellet_volume]               # fuel pellet total volume
    type = InternalVolume
    boundary = 8
    output = file
  [../]

  [./avg_clad_temp]               # average temperature of cladding interior
    type = SideAverageValue
    boundary = 7
    variable = temp
  [../]

  [./fis_gas_generated]
    type = ElementIntegralFisGasGeneratedSifgrs
    variable = temp
    block = '3 4'
   [../]

   [./fis_gas_released]
     type = ElementIntegralFisGasReleasedSifgrs
     variable = temp
     block = '3 4'
   [../]
   [./fis_gas_grain]
     type = ElementIntegralFisGasGrainSifgrs
     variable = temp
     block ='3 4'
   [../]

   [./fis_gas_boundary]
     type = ElementIntegralFisGasBoundarySifgrs
     variable = temp
     block = '3 4'
   [../]

  [./gas_volume]                # gas volume
    type = InternalVolume
    boundary = 9
  [../]

  [./flux_from_clad]           # area integrated heat flux from the cladding
    type = SideFluxIntegral
    variable = temp
    boundary = 5
    diffusivity = thermal_conductivity
  [../]

  [./flux_from_fuel]          # area integrated heat flux from the fuel
    type = SideFluxIntegral
    variable = temp
    boundary = 10
    diffusivity = thermal_conductivity
  [../]

  [./_dt]                     # time step
    type = TimestepSize
  [../]

  [./nonlinear_its]           # number of nonlinear iterations at each timestep
    type = NumNonlinearIterations
  [../]

  [./rod_total_power]
    type = ElementIntegralPower
    variable = temp
    fission_rate = fission_rate
    block = '3 4'
  [../]

  [./ave_fission_rate]
    type = ElementAverageValue
    variable = fission_rate
    block = '3 4'
  [../]

  [./TC_temp]
     type = NodalVariableValue
     variable = temp
     nodeid = 9738 #Global node ID 9739
  [../]

[]

# Define output file(s)
[Outputs]
  interval = 1
  output_initial = true
  csv = true
  exodus = true
  [./console]
    type = Console
    perf_log = true
    linear_residuals = true
    max_rows = 25
  [../]
#  iteration_plot_start_time = 1.035972e8
[]
```

APPENDIX B

Bison Operational Input Values

The parameter values below are inputs to the Risø AN3 input file described in Appendix A. Specifically, the values in the ”Clad Temperature” column are input, along with the time, into the csv file named ”riso_an3_clad_bc.csv”. The values in the ”Power History” column are input, along with the time, into the csv file named ”riso_an3_power_history.csv”. Finally, the values in the ”Fast Flux Multiplier” column are input, along with the time, into the csv file named ”riso_an3_fast_flux.csv”. Bison uses the fast flux multiplier, power history, and an additional scaling factor to determine the fast neutron flux.

Time (s)	Clad Temperature (C)	Fast Flux Multiplier	Power History (W/m)
0.0	300.0	0.000000	0
378000.0	571.5	0.901837	13000
1260000.0	571.6	0.901224	13100
2520000.0	571.3	0.886939	12800
5040000.0	571.6	0.909184	13100
7560000.0	572.0	0.949184	13600
10080000.0	572.4	0.976122	14000
12600000.0	572.5	0.991633	14200
15120000.0	572.5	1.001020	14200
17640000.0	572.6	1.006327	14300
20160000.0	572.5	1.010408	14200
22680000.0	572.5	1.009184	14200
23630400.0	572.4	1.007755	14100
24008400.0	579.2	1.466735	21900
24890400.0	580.2	1.541020	23100
26150400.0	580.5	1.557347	23400
28670400.0	580.5	1.569796	23400
31190400.0	580.3	1.586122	23200

33710400.0	580.2	1.593673	23100
36230400.0	579.9	1.593265	22800
38750400.0	579.7	1.593265	22500
41270400.0	579.4	1.583673	22200
43790400.0	579.2	1.590612	22000
46310400.0	579.0	1.584490	21700
48830400.0	578.9	1.587755	21600
51350400.0	578.8	1.593061	21500
53870400.0	578.8	1.608980	21500
55000800.0	578.7	1.618571	21400
55378800.0	574.9	1.345714	16900
56260800.0	576.0	1.441633	18200
57520800.0	576.0	1.442653	18200
60040800.0	576.0	1.449592	18200
62560800.0	576.1	1.467347	18300
65080800.0	576.1	1.487959	18400
67597200.0	576.1	1.500612	18400
70117200.0	576.3	1.530204	18600
72637200.0	576.3	1.541020	18600
75157200.0	576.4	1.559796	18700
77677200.0	576.4	1.567959	18700
79261200.0	576.4	1.584694	18700
79639200.0	580.3	1.910000	23200
80629200.0	580.5	1.925918	23400
81784800.0	580.0	1.884898	22900
84304800.0	579.3	1.822041	22100
86828400.0	578.4	1.739796	21000
89348400.0	577.7	1.686531	20200
91872000.0	577.3	1.648980	19700
94395600.0	576.9	1.620408	19300
96915600.0	576.6	1.597755	18900
99439200.0	576.3	1.580816	18600
101959200.0	576.1	1.573469	18400
103528800.0	576.1	1.578571	18300
103529000.0	500.0	0.000000	0
103529019.9	500.0	0.000000	0

103529119.9		0.012245	1500
103529299.9	611.0	0.012245	1500
103529309.9		0.031837	3900
103531160.0	611.3	0.031837	3900
103531170.0		0.102857	12600
103533619.9	611.8	0.102857	12600
103533629.9		0.116735	14300
103533740.1	611.8	0.116735	14300
103533750.1		0.145306	17800
103537700.1	611.9	0.145306	17800
103537710.1		0.166531	20400
103538479.9	612.0	0.166531	20400
103538489.9		0.176327	21600
103538600.1	612.0	0.176327	21600
103538610.1		0.189388	23200
103538839.9	612.0	0.189388	23200
103538849.9		0.208163	25500
103552640.1	612.1	0.208163	25500
103552650.1		0.229388	28100
103553239.9	612.1	0.229388	28100
103553249.9		0.252245	30900
103567520.0	612.2	0.252245	30900
103567530.0		0.263673	32300
103582160.1	612.2	0.263673	32300
103582170.1		0.282449	34600
103596439.9	612.3	0.282449	34600
103596449.9		0.295510	36200
103611019.9	612.3	0.295510	36200
103611029.9		0.312653	38300
103625239.9	612.3	0.312653	38300
103625249.9		0.332245	40700
103701019.9	612.3	0.332245	40700
103701029.9		0.295510	36200
103701140.1	612.3	0.295510	36200
103701150.1		0.250612	30700
103701739.9	612.2	0.250612	30700

103701749.9		0.226122	27700
103701860.1	612.1	0.226122	27700
103701870.1		0.179592	22000
103702700.0	612.0	0.179592	22000
103702710.0		0.202449	24800
103703480.1	612.1	0.202449	24800
103703490.1		0.208163	25500
103703600.0	612.1	0.208163	25500
103703610.0		0.245714	30100
103704500.0	612.2	0.245714	30100
103704510.0		0.275918	33800
103705339.9	612.2	0.275918	33800
103705349.9		0.287347	35200
103705460.1	612.3	0.287347	35200
103705470.1		0.306939	37600
103706239.9	612.3	0.306939	37600
103706249.9		0.331429	40600
103781900.0	612.3	0.331429	40600
103781910.0		0.293878	36000
103782019.9	612.3	0.293878	36000
103782029.9		0.249796	30600
103782680.1	612.2	0.249796	30600
103782690.1		0.235918	28900
103782800.0	612.2	0.235918	28900
103782810.0		0.202449	24800
103782919.9	612.1	0.202449	24800
103782929.9		0.176327	21600
103783639.9	612.0	0.176327	21600
103783649.9		0.186122	22800
103783760.1	612.0	0.186122	22800
103783770.1		0.205714	25200
103784539.9	612.1	0.205714	25200
103784549.9		0.241633	29600
103785439.9	612.2	0.241633	29600
103785449.9		0.277551	34000
103786339.9	612.2	0.277551	34000

103786349.9		0.306122	37500
103787239.9	612.3	0.306122	37500
103787249.9		0.332245	40700
103788200.0	612.3	0.332245	40700
103788210.0		0.233469	28600
103788319.9	612.2	0.233469	28600
103788329.9		0.103673	12700
103788440.1	611.8	0.103673	12700
103788450.1		0.043265	5300
103788679.9	611.4	0.043265	5300
103788689.9		0.011429	1400
103789797.0	611.0	0.011429	1400

BIBLIOGRAPHY

[1] H.S. Abdel-Khalik and P.J. Turinsky. Evaluation of core attributes uncertainties due to input data uncertainties. volume 92 of *Transactions of the American Nuclear Society*, San Diego, CA, 2005.

[2] B.M. Adams, W.J. Bohnhoff, K.R. Dalbey, J.P. Eddy, M.S. Eldred, D.M. Gay, K. Haskell, P.D. Hough, and L.P. Swiler. Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 5.0 user's manual. Technical Report SAND2010-2183, Sandia National Laboratory, December 2009.

[3] Nitin Agarwal and N.R. Aluru. A domain adaptive stochastic collocation approach for analysis of mems under uncertainties. *Journal of Computational Physics*, 228:7662–7688, 2009.

[4] John K. Au. *An Ab Initio Approach to the Inverse Problem-based Design of Photonic Bandgap Devices*. PhD thesis, California Institute of Technology, 2007.

[5] D. Ayres, M.M.R. Williams, and M.D. Eaton. Time and static eigenvalues of the stochastic transport equation by the methods of polynomial chaos. *Progress in Nuclear Energy*, 67:33–55, 2013.

[6] Volker Barthelmann, Erich Novak, and Klaus Ritter. High dimensional polynomial interpolation on sparse grids. *Advances in Computational Mathematics*, 12:273–288, 2000.

[7] Jean-Paul Berrut and Lloyd N. Trefethen. Barycentric lagrange interpolation. *SIAM Review*, 46(3):501–517, 2004.

[8] Soren Bisgaard and Murat Kulahci. *Time Series Analysis and Forecasting by Example*. John Wiley and Sons, 2011.

[9] S.M. Bowman. Scale 6: Comprehensive nuclear safety analysis code system. *Nuclear Technology*, 174(2):126–148, 2011.

[10] George Box. Evolutionary operation: A method for increasing industrial productivity. *Journal of the Royal Statistical Society*, 6(2):81–101, 1957.

[11] John Boyd. *Chebyshev and Fourier Spectral Methods*. Dover Publications, 2 edition, 2001.

[12] John P. Boyd. *Chebyshev and Fourier Spectral Methods*. Dover Publications, 2 edition, 2001.

[13] T. Bui-Thanh, K. Willcox, and O. Ghattas. Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM Journal on Scientific Computing*, 30(6):3270–3288, 2008.

[14] John Burkardt. Sparse grid collocation for uncertainty quantification. SCALA, Louisiana State University, 2012.

[15] Philip Davis and Philip Rabinowitz. *Methods of Numerical Integration*. Dover, 2007.

[16] Thomas Downar. Parcs: Purdue advanced reactor core simulator. Proceedings of the International Conference on the New Frontiers of Nuclear Technology, Seoul, South Korea, 2002.

[17] James Duderstadt and Louis Hamilton. *Nuclear Reactor Analysis*. John Wiley and Sons, 1976.

[18] D. Eardley. Quantification of margins and uncertainties. Technical report, MITRE Corporation, March 2005.

[19] Erin D. Fichtl and Anil K. Prinja. The stochastic collocation method for radiation transport in random media. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 112(4):646–659, 2011.

[20] Alexander Forrester, Andras Sobester, and Andy Keane. *Engineering Design via Surrogate Modelling*. John Wiley and Sons, 2008.

[21] A. Gandini. Implicit and explicit higher order perturbation methods for nuclear reactor analysis. *Nuclear Science and Engineering*, 67, 1978.

[22] Zhen Gao and Jan S. Hesthaven. On anova expansions and strategies for choosing the anchor point. *Applied Mathematics and Computation*, 217:3274–3285, 2010.

[23] Zhen Gao and Jan S. Hesthaven. Efficient solution of orindary differential equations with high-dimensional parameterized uncertainty. *Communications in Computational Physics*, 10(2):253–278, 2011.

[24] Roger Ghanem and P. D. Spanos. Polynomial chaos in stochastic finite elements. *Journal of Applied Mechanics*, 57(1):197–202, 1990.

[25] L. Gilli, D. Lathouwers, J.L. Kloosterman, and T.H. van der Hagen. Application of sensitivity analysis to a simplified coupled neutronic thermal-hydraulics transient in a fast reactor using adjoint techniques. In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, Rio de Janiero, Brazil, May 2011. American Nuclear Society.

[26] L. Gilli, D. Lathouwers, J.L. Kloosterman, and T.H. van der Hagen. Performing uncertainty analysis of a nonlinear point-kinetics/lumped paremeters problem using polynomial chaos techniques. *Annals of Nuclear Energy*, 40:35–44, 2012.

[27] A.A. Giunta, J.M. McFarland, L.P. Swiler, and M.S. Eldred. The promise and peril of uncertainty quantification using response surface approximations. *Structure and Infrastructure Engineering*, 2(3-4):175–189, 2006.

[28] J.C. Helton and F.J. Davis. Latin hypercube sampling and the propagation of uncertainty in analysis of complex systems. *Reliability Engineering and System Safety*, 81:23–69, 2003.

[29] Peter Henrici. *Essentials of Numerical Analysis with Pocket Calculator Demonstrations*. John Wiley and Sons, 1982.

[30] Markus Holtz. *Sparse Grid Quadrature in High Dimensions with Applications in Finance and Insurnance*. Springer, 2011.

[31] C. Housiadas. Lumped paramaters analysis of coupled kinetics and thermal-hydraulics for small reactors. *Annals of Nuclear Energy*, 29:1315–1325, 2002.

[32] Gianluca Iaccarino. Uncertainty quantification in computational science. Lehrstuhl fr Aerodynamik und Strmungsmechanik, Technische Universitt Mnchen, 2011.

[33] Edward H. Isaaks and Mohan Srivastava. *An Introduction to Applied Geostatistics*. Oxford University Press, 1990.

[34] K. Ivanov, M. Avramova, I. Kodeli, and E. Sartori. Benchmark for uncertainty analysis in modeling for design, operation and safety analysis of lwrs. Technical report, Nuclear Energy Agency, 2007.

[35] M.A. Jessee, H.S. Khalik, and P.J. Turinsky. Evaluation of bwr core attributes uncertainties due to multi- group cross section uncertainties. In *Joint International Topical Meeting on Mathematics and Computation and Supercomputing in Nuclear Applications*, Monterey, CA, April 2007. American Nuclear Society.

[36] M.A. Jessee, P.J. Turinsky, and H.S. Abdel-Khalik. Many-group cross-section adjustment techniques for boiling water reactor adaptive simulation. *Nuclear Science and Engineering*, 169:40–55, 2011.

[37] Russell C. Johns. Early user test: Bison for fission gas release modeling. Technical Report LA-UR-14-20508, Los Alamos National Laboratory, January 2014.

[38] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.

[39] H.F. Kaiser and K. Dichman. Sample and population score matrices and sample correlation matrices from an arbitrary population correlation matrix. *Psychometrica*, 27(2):179–182, 1962.

[40] C. Killeen, J.A. Turnbull, and E. Sartori. Fuel modelling at extended burnup: Iaea coordinated research project fumex-ii. In *Proceedings of the 2007 International LWR Fuel Performance Meeting*, San Francisco, CA, October 2007.

[41] J. Killeen, E. Sartori, and T. Tverbergi. Fumex-iii: A new iaea coordinated research project on fuel modeling at extended burnup. In *Proceedings of Top Fuel*, Paris, France, September 2009.

[42] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[43] M. Klein, L. Gallner, B. Krzykacz-Hausmann, A. Pautz, and W. Zwermann. Influence of nuclear data covariance on reactor core calculations. *Kerntechnik*, 3:174–78, 2011.

[44] Andreas Klimke and Barbara Wohlmuth. Peicewise multilinear hierarchical sparse grid interpolation in matlab. *ACM Transactions on Mathematical Software*, 31(4):561–579, 2005.

[45] Brendan Kochunas, Benjamin Collins, Thomas J. Downar, and William R. Martin. Overview of development and design of mpact: Michigan parallel characteristics transport code. In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, Sun Valley, ID, May 2013.

[46] K. Lassmann. The structure of fuel element codes. *Nuclear Engineering and Design*, 57(1):17–39, 1980.

[47] Genyuan Li, Sheng-Wei Wang, and Herschel Rabitz. High dimensional model representations: Concepts and applications. *Journal of Physical Chemistry A*, 105:7765–7777, 2001.

[48] Meilin Liu, Zhen Gao, and Jan Hesthaven. Adaptive sparse grid algorithms with applications to electromagnetic scattering under uncertainty. *Applied Numerical Mathematics*, 61:24–37, 2011.

[49] Xiang Ma and Nicholas Zabaras. An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. *Journal of Computational Physics*, 228:3084–3113, 2009.

[50] Xiang Ma and Nicholas Zabaras. An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations. *Journal of Computational Physics*, 229:3884–3915, 2009.

[51] O.P Le Maitre and Omar Knio. *Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics*. Springer, 2010.

[52] Max D. Morris. Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33(2):161–174, 1991.

[53] Max D. Morris and Toby J. Mitchell. Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference*, 43:381–402, 1995.

[54] Brent Oksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer-Verlag, 6 edition, 2005.

[55] G. Pastore, J.D. Hales, S.R. Navascone, D.M. Perez, B.W. Spencer, and R.L. Williamson. Analysis of fission gas release in lwr fuel using the bison code. Technical Report INL/CON-13-28389, Idaho National Laboratory, September 2013.

[56] G. Pastore, L. Luzzi, V. Di Marcello, and P. Van Uffelen. Physics-based modelling of fission gas swelling and release in uo_2 applied to integral fuel rod analysis. *Nuclear Engineering and Design*, (256):75–86, 2013.

[57] G. Pastore, L.P. Swiler, J.D. Hales, S.R. Navascone, D.M. Perez, B.W. Spencer, L. Luzzi, P. Van Uffelen, and R.L. Williamson. Uncertainty and sensitivity analysis of fission gas behavior in engineering-scale fuel modeling. *Elsevier*, (preprint), 2014.

[58] Giovanni Pastore, Jason Hales, Steve Navascone, Danielle Perez, Benjamin Spencer, and Richard Williamson. Analysis of fission gas release in lwr fuel using the bison code. In *LWR Fuel Performance MeetingTopFuel*, Charlotte, NC, 2013.

[59] D. M. Perez, R. L. Williamson, S. R. Novascone, G. Pastore, J. D. Hales, and B. W. Spencer. Assessment of bison: A nuclear fuel performance analysis code. Technical report, Idaho National Laboratory, November 2013.

[60] D.M. Perez, R.L. Williamson, S.R. Novascone, T.K. Larson, J.D. Hales, B.W. Spencer, and G. Pastore. An evaluation of the nuclear fuel performance code bison. In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, Sun Valley, ID, May 2013.

[61] M.J.D. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. *Advances in Optimization and Numerical Analysis*, 271:51–67, 1994.

[62] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. ISBN 3-900051-07-0.

[63] Jerome Sacks, William Welch, Toby Mitchell, and Henry Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435, 1989.

[64] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global Sensitivity Analysis*. John Wiley and Sons, 2008.

[65] H.E. Salzer. Lagrangian interpolation at the chebyshev points $x_{n,\nu} = \cos(\nu\pi/n)$. *The Computer Journal*, 15(2):156–159, 1972.

[66] Jonathon Shlens. A tutorial on principal component analysis. *Systems Neurobiology Laboratory, University of California at San Diego*, (82), 2005.

[67] Sergei A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Soviet Math. Dokl.*, 4:240–243, 1963.

[68] I. Sobol. Sensitivity estimates for nonlinear mathematical models. *Mathematical Modeling and Computational Experiment*, 1:407–414, 1993.

[69] L. Swiler, R. Slepoy, and A.A. Giunta. Evaluation of sampling methods in constructing response surface approximations. In *SDM non-Deterministic Approaches Conference*. American Institute of Aeronautics and Astronautics, May 2006.

[70] Laura Swiler, Giovanni Pastore, Richard Williamson, and Danielle Perez. Sensitivity analysis of the fission gas behavior in bison. Technical report, Sandia National Laboratories, May 2013.

[71] Laura Swiler, Richard L. Williamson, and Danielle M. Perez. Calibration of a fuel relocation model in bison. In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, Sun Valley, ID, May 2013.

[72] Lloyd Trefethen. Is gaussian quadrature better than clenshaw-curtis? *SIAM Review*, 50(1):67–87, 2008.

[73] Lloyd Trefethen. *Approximation Theory and Approximation Practice*. SIAM, 2012.

[74] Jingbo Wang and Nicholas Zabaras. Hierarchical bayesian models for inverse problems in heat conduction. *Inverse Problems*, 21(1):183, 2005.

[75] R.J. White, R.C. Corcoran, and J.P. Barnes. A summary of swelling data obtained from the agr/halden ramp test programme. Technical Report RT/NG/EXT/REP/0206/02, 2006.

[76] M.L. Williams. Perturbation theory for reactor analysis. In *CRC Handbook of Nuclear Reactor Calculations*, pages 63–188. CRC Press, 1986.

[77] M.L. Williams, J.C. Gehin, and W.S. Yang. Combining mc^2-2 and scale cross section methods to process nuclear data for gnep. In *International Conference on Reactor Physics, Nuclear Power: A Sustainable Resource*, Interlaken, Switzerland, September 2008.

[78] M.L. Williams, G. Ilas, M.A. Jessee, B.T. Rearden, D. Wiarda, W. Zwermann, L. Gallner, M. Klein, B. Krzykacz-Hausmann, and A. Pautz. A statistical sampling method for uncertainty analysis with scale and xsusa. *Nuclear Technology*, 183(3):515–526, 2012.

[79] M.M.R. Williams. Polynomial chaos functions and stochastic differential equations. *Annals of Nuclear Energy*, 33(9):774–785, 2006.

[80] M.M.R. Williams. Polynomial chaos functions and neutron diffusion. *Nuclear Science and Engineering*, 155(1):109–118, 2007.

[81] R.L. Williamson, J.D. Hales, S.R. Novascone, M.R. Tonks, D.R. Gaston, C.J. Permann, D. Andrs, and R.C. Martineau. Multidimensional multiphysics simulation of nuclear fuel behavior. *Journal of Nuclear Materials*, 423:149–163, 20012.

[82] Dongbin Xiu and Jan S. Hesthaven. High-order collocation methods for differential equations with random inputs. *SIAM Journal of Scientific Computing*, 27:1118–1139, 2005.

[83] A. Yankov, B.S. Collins, M.A. Jessee, and T.J. Downar. A generalized adjoint approach for quantifying reflector assembly discontinuity factor uncertainties. In *Advances in Reactor Physics*, Knoxville, TN, April 2012. American Nuclear Society.

[84] Artem Yankov, Benjamin Collins, and Markus Klein. A two-step approach to uncertainty quantification of core simulators. *Science and Technology of Nuclear Installations*, 2012.