ADJOINT-BASED UNCERTAINTY QUANTIFICATION FOR NEUTRON TRANSPORT CALCULATIONS

A Dissertation

by

DONALD EUGENE BRUSS

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Jim E. Morel
Co-Chair of Committee,	Jean C. Ragusa
Committee Members,	Marvin L. Adams
	Jean-Luc Guermond
Head of Department,	Yassin A. Hassan

August 2016

Major Subject: Nuclear Engineering

ABSTRACT

The accurate modeling of complex physical phenomena, such as radiation transport in a laboratory experiment or a nuclear reactor, challenges the limits of modern computing resources and helps drive the requirement for next-generation exascale computers. The efficient and accurate propagation of uncertainty through these models is an area of ongoing research. Uncertainties in material properties contribute to the uncertainty in quantities of interest (QoIs) for which a problem is solved, but quantifying the uncertainty in a QoI is often prohibitively expensive.

In this research an integrated approach to uncertainty quantification (UQ) for radiation transport problems with uncertain nuclear data is introduced. A novel dimension reduction method is applied to the nuclear data characterizing cross-section uncertainty. An adjoint-based sensitivity analysis is performed to yield sensitivity coefficients for the QoI with respect to the reduced-dimensional space. Finally, response surfaces are constructed for the QoI over the reduced-dimensional input space. These surfaces yield information about the distribution of the QoI over the original uncertain input space. This multi-step approach is applied to several radiation transport problems for which traditional UQ methods are prohibitively expensive.

DEDICATION

To my wife Lauren, who has encouraged and supported me throughout this project, and to my parents Larry and Gwen, who instilled in me and promoted my love of science and mathematics.

“I try all things; I achieve what I can.” Herman Melville, *Moby Dick*.

ACKNOWLEDGEMENTS

I would like to thank my committee chairs, Drs. Jim Morel and Jean Ragusa, for their mentoring and guidance throughout this project. I would also like to thank Drs. Marvin Adams, Ryan McClarren, and Derek Bingham (of Simon Frasier University in Vancouver, BC), for their input and direction. Finally, I am extremely grateful to my peers, Andrew Till, Jarrod Edwards, and Carolyn McGraw, for years of discussions, collaboration, and encouragement during our time at Texas A&M.

This research was funded by the Department of Energy Office of Science, through the Center for the Exascale Simulation of Advanced Reactors (CESAR) (contract number DE-AC02-06CH11357). This research was also funded by the Department of Energy National Nuclear Security Administration's Predictive Science Academic Alliance Program (PSAAP-II) (contract number DE-NA0002376).

NOMENCLATURE

DOE	Department of Energy
NNSA	National Nuclear Security Agency
PSAAP	Predictive Science Academic Alliance Program
CERT	Center for Exascale Radiation Transport
UQ	Uncertainty Quantification
IMSDP	Infinite-Medium Slowing Down Problem
PDT	The Parallel Deterministic Transport research code
†	The “dagger” superscript denotes “adjoint”
LPoU	Linear Propagation of Uncertainty
MC	Monte Carlo
MCMC	Markov Chain Monte Carlo
GEMARS	Gradient-Enhanced Multivariate Adaptive Regression Splines
IM-1	Impurity Model 1 (an experiment performed for CERT)
1E5	10^5 (this format maximizes the size of the exponent for ease of reading)

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Accurate modeling of complex physical phenomena, such radiation transport in a laboratory experiment or neutron distributions in a nuclear reactor, challenge the limits of modern computing resources and help drive the requirement for next-generation exascale computers. The efficient and accurate propagation of uncertainty through these models is an area of ongoing research. Uncertainties in material properties contribute to the uncertainty in a quantity of interest (QoI) calculated as problem output. Understanding and quantifying the uncertainty in QoIs is an important part of solving complex systems of equations. A response surface is one tool used to investigate the contribution of uncertainty in input parameters to uncertainty in a QoI, but the cost of building accurate response surfaces greatly increases as the dimensionality of the input space increases. Deterministic neutron transport calculations may require thousands or tens of thousands of uncertain material cross sections as input, making the generation of a response surface for these variables by traditional means untenable.

1.1 The UQ Process

The uncertainty quantification (UQ) process can be thought of as three consecutive tasks. First, the uncertain input parameters must be characterized and quantified so that they can be sampled. Second, a physics calculation must be performed to determine the quantity of interest, and, if possible, the partial derivative of the QoI to the input parameters at a given input state (referred to as sensitivity coefficients). Third, given QoIs and sensitivity coefficients at sample points, a response surface must be

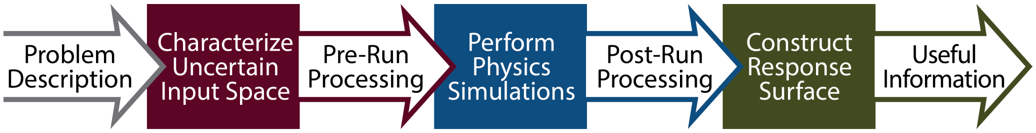
Problem Description
Characterize Uncertain Input Space
Pre-Run Processing
Perform Physics Simulations
Post-Run Processing
Construct Response Surface
Useful Information

Figure 1.1: A graphical representation of the UQ process.

constructed for the QoI. Finally, this surface can be sampled to generated statistics about the QoI. These statistics can be used to inform the design of experiments or quantify irreducible sources of uncertainty in an experiment.

These three main phases of the UQ process (characterizing the input space, the physics calculations, and the generation of a response surface) are independent steps, with the output of each step processed and passed to the next step. These tasks and the flow of information through the UQ process are visualized in Figure 1.1.

This research project was originally devised to inform and understand calculations of a laboratory experiment being performed at the Nuclear Science Center at Texas A&M University. Radiation transport calculations are performed to model the neutron propagation through the experiment and predict detector response. Uncertainty in the nuclear data used in these calculations gives rise to uncertainty in the detector response. Quantifying this uncertainty due to nuclear data is necessary to interpret the calculation results.

Given this understanding of the project's origin, the steps of the UQ analysis presented in Figure 1.1 can be addressed more specifically. The source of uncertainty considered in this work are material properties quantifying neutron-nucleus interactions. This nuclear data contains both aleatory and epistemic uncertainties. The quantum mechanical nature of many nuclear reactions may contribute irreducible or aleatory uncertainties, especially in the unresolved resonance range of nuclear interactions. Epistemic uncertainty is introduced by the complexity of neutron-nucleus interactions and the necessarily complex experiments necessary to measure these interactions. Libraries of nuclear data do not distinguish between these two sources of uncertainty but present an estimate of the total uncertainty in the nuclear data.

The capability to access the uncertainty data in the nuclear data libraries and put it to use in the analysis of radiation transport problems had not previously been developed at Texas A&M University. A Python tool was written to write input decks for the nuclear data processing code NJOY, run the code, and extract the processed nuclear data. This data is then used to draw samples from the nuclear data's distribution and format it for use in a neutron transport code.

The physics calculations in this analysis are radiation transport simulations. These calculations are performed with PDT, a parallel deterministic radiation transport code developed at Texas A&M. The dimensionality of the uncertain nuclear-data space and the cost of radiation transport simulations for realistic problems make sampling-based UQ methods prohibitively expensive. An adjoint sensitivity method recently implemented in PDT is applied to maximize the useful information extracted from limited computational resources.

A C++ tool was written to construct a response surface given the data calculated with the adjoint sensitivity PDT simulations. This tool was specifically written for application to problems with large numbers of independent parameters. The surfaces generated with this tool are accurate and predictive. Once constructed, these surfaces can be used to generate statistics about the quantity of interest or serve as an emulator for future calculations.

1.2 Novelty

The novelty of this work is three-fold. First, the uncertain input space considered in this work is of very high dimension. The dimensionality of the problem is reduced by considering the physics underlying the cross-section data, the uncertainty in the cross-section data, and the sensitivity of the QoI to the cross sections. This multifaceted approach to dimension reduction is the major innovation in this work.

Second, the reduced dimensionality of the input space remains large enough to defy traditional Monte Carlo-based sampling techniques for response surface generation. A tool was written to construct a response surface using gradients of the QoI at sparse sampling points. An early version of this tool was presented at the M&C 2015 conference by Bruss [7]. This tool has been refined and rewritten to reduce the time required to generate the response surfaces.

Finally, these novel components will be applied to assess uncertainty in predicted detector response in an experiment being performed at Texas A&M University. The simulations performed to model this experiment, referred to as the "Impurity Model 1" or "IM-1" experiment, require on the order of 10,000 (O(10,000)) CPU hours on the supercomputer Syrah at Lawrence Livermore National Laboratory. The complexity of this problem makes traditional Monte Carlo-based UQ analysis methods impractical. The combination of recently developed capabilities and the novel methods developed in this work make feasible an uncertainty quantification analysis for a problem of this magnitude.

1.3 Preview of Sections

The remainder of this work is divided into three theory sections and a section of results. The first theory section explains the generation of first and second moments of the multigroup cross sections from the evaluated nuclear data files using the Los Alamos National Laboratory-produced code NJOY 2012 [22]. The modeling of these cross sections as multivariate Gaussian distributions is outlined.

The second theory section covers the equations of radiation transport. The Boltzmann equation for neutron transport is introduced along with the concept of a Quantity of Interest. The adjoint equations used to calculate the sensitivity of a quantity of interest with respect to the multigroup cross sections are derived. Finally, several notes are included on how these quantities are calculated in PDT.

In the third theory section we address the curse of dimensionality. A novel dimension reduction scheme is introduced to map from the highly-correlated high-dimensional cross-section space to a smaller space containing uncorrelated parameters. The construction and use of a response surface using gradient-enhanced multivariate adaptive regression splines is explored.

Finally, results are presented for a variety of test problems of increasing complexity and computational cost. An analytic function is used to test the construction of response surfaces. A simple neutron transport problem with an analytic solution is used to test the dimension reduction machinery introduced in Section 4.4. A one-dimensional slab problem is used as an inexpensive surrogate for the complex physical experiment. The most computationally expensive test problem corresponds to the physical experiment being performed at Texas A&M University. Finally, two problems related to the analysis of nuclear reactors are examined.

A few short appendices include instructions to run the two Python tools written in the course of this project. The first Python tool is used to run NJOY to produce multigroup cross sections and write PDT-formatted cross-section files. The second Python tool is used to drive the UQ analysis.

2. NUCLEAR DATA AND CROSS SECTION PREPARATION

The physical problems considered in this work primarily concern the transport of neutrons through various laboratory experiments. The distribution of neutrons in these experiments is described by the radiation transport equation. The interaction of neutrons with the medium through which they pass is characterized by the medium's material properties. Nuclear interactions may remove a neutron from the problem or cause the neutron to undergo a change in energy or direction of travel. The interaction probabilities for these various reactions are referred to as nuclear cross sections. The generation of cross sections for use in deterministic radiation transport calculations is discussed in this section. The radiation transport equations themselves are discussed in Section 3.

2.1 Cross Section Fundamentals

Cross sections are used to describe the interaction probabilities of various neutron-nucleus reactions that may occur as a neutron approaches a nucleus. The cross sections depend both upon the nuclear properties unique to each isotope and the material density specific to a particular problem or experiment. To distinguish between these two quantities, macroscopic and microscopic cross sections are defined. The microscopic cross section σ has units of area and characterizes an isotope's interaction probability. The macroscopic cross section Σ has units of inverse length and characterizes a specific material's interaction properties. The macroscopic and microscopic cross sections are related by the isotope's atom density in a material N, which has units of atoms per volume,

$$\Sigma = N\sigma. \tag{2.1}$$

A useful unit of area for describing nuclear interactions is the barn. A barn is defined as 10^{-24} cm^2 and is roughly equivalent to the geometric cross section of an atomic nuclei [14]. Microscopic cross sections are typically reported in barns, and number densities are typically reported with units of atoms per barn-cm. The macroscopic cross section is reported in units of inverse cm, as these units are convenient for the calculation of the neutron's mean free path. The mean free path has units of length and is the average distance the neutron travels between interactions. The mean free path is calculated as one over the macroscopic cross section Σ.

The cross section for a material m comprised of J isotopes is simply the sum of the macroscopic cross sections corresponding to each isotope,

$$\Sigma_m = \sum_{j=1}^{J} N_j \sigma_j. \tag{2.2}$$

As the number density of a material is problem specific and easily derived from the microscopic cross sections, the microscopic cross sections are considered exclusively in this work.

The interaction of a neutron and a nucleus may involve elastic scattering, in which the neutron scatters off of the nucleus directly, or the neutron may be absorbed by the nucleus, resulting in the creation of an unstable compound nucleus. The compound nucleus subsequently decays with the emission of a particle of some sort-either a neutron, a gamma ray, a secondary particle, or by complete disintegration.

If the sum of the neutron's relative kinetic energy and the binding energy associated with that neutron in the compound nucleus correspond very precisely to an excited nuclear energy state of the compound nucleus, the interaction probability exhibits a sharp peak at that energy. These peaks are called resonances and are very strongly a function of the neutron energy E. The cross section corresponding to elastic scattering, in contrast, is a function of the nucleus' geometric cross section [14].

The resonance behavior of many nuclear reactions is extremely complex and can be difficult to measure. Figure 2.1, generated with the ENDF database maintained by the International Atomic Energy Agency (IAEA) [34], plots the total interaction probability for uranium 235 as a function of neutron energy. The cross section varies

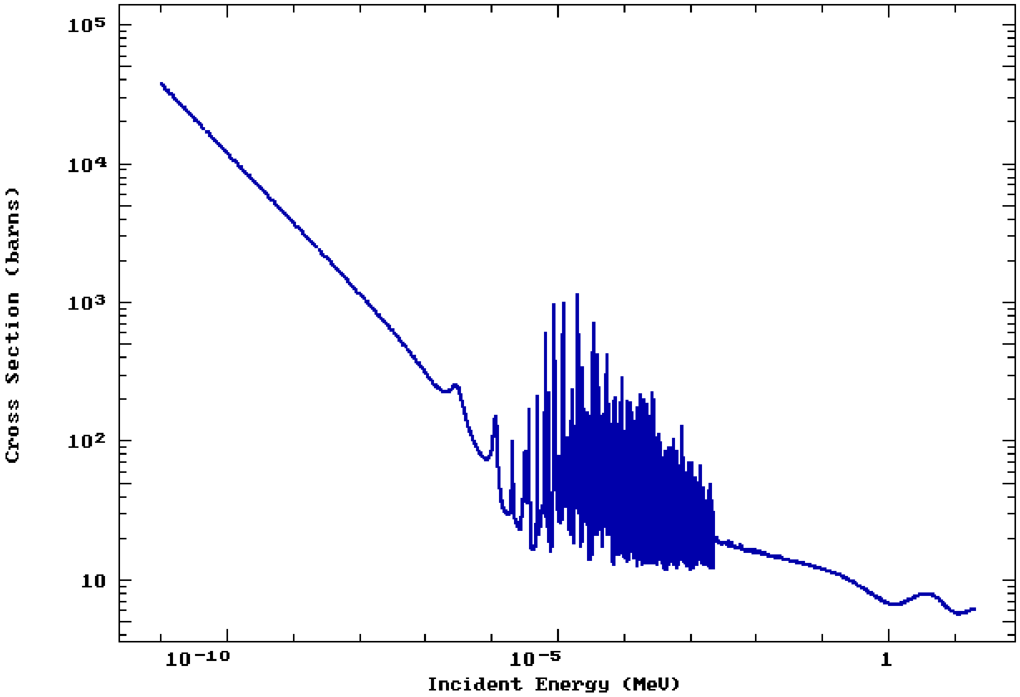

Figure 2.1: Total cross section for U-235 [34].

smoothly at low energy. Strong resonances are clearly visible starting at a neutron energy of about 10 eV and continuing through about 5E-3 MeV. This energy region is referred to as the “resolved resonance region.” Above this range the plot of the cross sections appears relatively smooth through about 5E-2 MeV. This region is referred to as the “unresolved resonance region.” Although nuclear resonances are present at these energies, their structure is so fine that they cannot be resolved. At high energy, the resonance structure vanishes entirely and the cross section is again smooth.

Reactions that result in the removal of a neutron from the neutron population are said to be absorptive. These reactions may result in the production of a secondary particle. Examples of this kind of interaction include the emission of a gamma ray (n,γ), the emission of a proton (n,p), the emission of an alpha particle (n,α), the emission of a deuteron or triton, (n,d) and (n,t), respectively, or other similar reactions. These reactions are described by the cross section as a function of neutron energy, $\sigma(E)$.

Various reactions result in the production of one or more neutrons. These include elastic scattering (n,el), inelastic scattering (n,in), the production of two neutrons (n,2n), and a variety of reactions that result in the production of a neutron and a secondary particle. These secondary particles may include a proton (n,n+p), deuteron (n,n+d), triton (n,n+t), or alpha particle (n,n+α). These reactions may change the incident neutron’s direction of flight and energy and are therefore described as a function of the initial and final energies E and E' and incident and final angle $\vec{\Omega}$ and $\vec{\Omega}'$, respectively, $\sigma(E \rightarrow E', \vec{\Omega} \rightarrow \vec{\Omega}')$. Except in rare cases (such as very low energy neutrons and materials exhibiting bulk crystal structure), cross sections do not depend upon the neutrons initial and final directions of travel, but rather the change in the direction of travel, $\mu = \vec{\Omega} \cdot \vec{\Omega}$ [22]. Cross sections that result in the production of a neutron are generally stored as $\sigma(E \rightarrow E', \mu)$.

Incident neutrons that cause the target isotope to fission produce several neutrons in addition to two fission products and gamma rays. The neutron production term is described by the number of neutrons produced per fission $\nu(E)$, a distribution function in energy $\chi(E \rightarrow E')$, and an assumed isotropic distribution in angle.

The measurement and use of nuclear cross sections in deterministic radiation transport calculations is described in the following sections.

2.2 The Evaluated Nuclear Data Files

Nuclear cross-section data is generated by theoretical calculation and by direct experimentation. Experimental sources of uncertainty give rise to uncertainty in the value of measured cross sections. Uncertainty in model parameters themselves measured by experiment produce uncertainty in calculated cross-section values. Significant disagreement may be present in the cross-section data measured between different experiments, and these results may not match theoretical calculations for the same cross section [4][14]. The process of collecting data from various experiments and calculations and producing a cohesive nuclear data set is called evaluation. Evaluated nuclear data is collected in data sets called the Evaluated Nuclear Data Files (ENDF). These files contain both the best estimate of the energy-dependent cross sections and an estimate of the uncertainty in those cross sections.

Several standardized formats for the ENDF files exist, primarily corresponding to the county in which the data is produced. The ENDF/B format is used in the United States; the most current data (and the data used in this work) is released as ENDF/B-VII.1. The ENDF/B data set does not directly contain cross sections as a function of energy across the entire energy spectrum. Cross-section data in the smooth region of the energy spectrum is stored as data points on an energy grid and a rule for interpolation between the data points. The resolved resonance region is described by resonance parameters. Cross sections in the unresolved resonance region are computed from a statistical distribution of the resonance parameters [22]. Various codes exist to process the ENDF/B formatted data files, including AMPX, a code written by Oak Ridge National Laboratory, and NJOY, a code written by Los Alamos National Laboratory.

The use of NJOY to process the ENDF/B data files and produce the multigroup cross-section data needed for deterministic radiation transport calculations is discussed in Section 2.5.

2.3 The Multigroup Method

The continuous energy cross-section data is discretized in energy by the multigroup method. The central approximation of the multigroup method is that the cross sections may be represented as a collection of piecewise constant step functions in energy. This requires the division of the continuous energy variable into bins called groups. Ideally these groups are chosen to reflect the physics of the problem of interest. Given boundaries E_g and E_{g+1} for energy group g, the multigroup cross sections σ_g are obtained by integrating the continuous energy cross section $\sigma(E)$ from E_g to E_{g+1} with some spectral weighting function $f(E)$,

$$\sigma_g = \frac{\int_{E_g}^{E_{g+1}} f(E)\sigma(E)dE}{\int_{E_g}^{E_{g+1}} f(E)dE}. \tag{2.3}$$

The choice of the weighting function $f(E)$ depends strongly upon the problem for which the cross sections are to be used. Ideally the exact solution of the radiation transport equations for the specific problem of interest would be used as the spectral weighting function [4]. This approach would produce cross sections that are spatially dependent as well as energy dependent. While it may be theoretically possible to iteratively solve for this solution, it is wholly impractical for problems of interest.

A second method to generate accurate multigroup cross sections is to select an extremely fine group structure such that resonances are well resolved and errors in the weighting spectrum are relatively unimportant. In this case a constant weighting spectrum would yield accurate multigroup cross sections [21][22]. This approach also eliminates the spatial dependence of the cross sections introduced by using the actual solution as a weighting spectrum. However, the number of groups required to resolve the cross sections in this fashion may be one or more orders of magnitude larger than desired for a calculation.

The weighting spectrum may be approximated accurately if the problem of interest is relatively simple and has a well-known solution. A source-detector laboratory experiment, for example, with neutrons produced at high energy and scattered through materials with small atomic weights, should exhibit a 1/E slowing down spectrum in the intermediate energies and a thermal Maxwellian distribution at low energies. A spectrum containing these components would be expected to produce accurate cross sections. A variety of precomputed weighting spectrums are included in NJOY for various problems with well-known approximate solutions.

The presence of strong resonances at epithermal and fast neutron energies complicate the calculation of cross sections. The presence of a strong resonance depresses the solution $\phi(E)$ in the vicinity of the resonance in an effect called “self shielding” [22]. This fine structure of the self-shielded weighting spectrum prevents the use of a simple, smooth weighting spectrum for problems containing isotopes with strong resonances. Cross sections calculated with the assumption that an isotope’s resonances have no effect on the weighting spectrum are referred to as “infinitely dilute” cross sections [22]. These infinitely dilute cross sections are not appropriate for applications which involve significant quantities of materials that exhibit strong resonances, such as nuclear reactor calculations. Various methods for calculating appropriately self-shielded weighting spectrums exist and are discussed at length in [22]. The infinitely dilute cross sections were appropriate for most of the problems considered in this work.

Weighting spectrums for problems containing strong resonances are commonly generated by solving an infinite-medium slowing down problem (IMSDP) containing the same materials as the problem of interest. By eliminating the spatial dependence of the solution the IMSDP can be solved using either the ENDF pointwise data and resonance parameters or a very fine group structure generated with a constant weighting spectrum. In the case of a very fine group structure the impact of the weighting spectrum used to generate the cross sections is minimized. The solution of the IMSDP is used as the weighting spectrum to generate the multigroup cross sections for the problem of interest [18][21].

The piecewise-step nature of the multigroup cross sections is illustrated in Figure 2.2, which contains the continuous energy and multigroup total cross section for natural carbon in blue and red, respectively. The group boundaries and weighting spectrum used to generate these multigroup cross sections were derived from the problem of

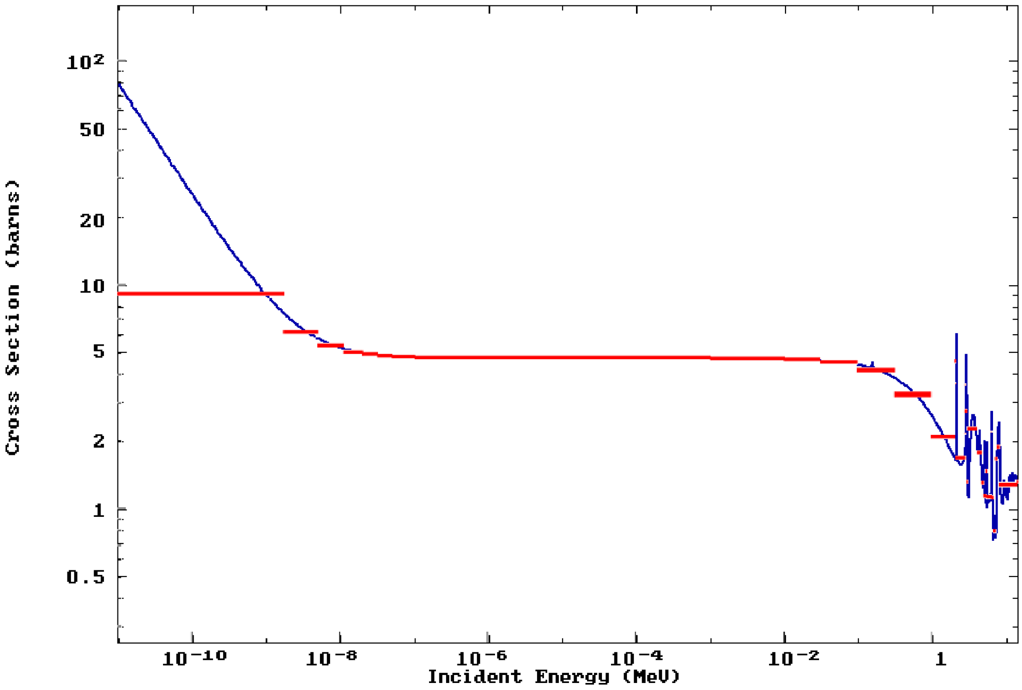

Figure 2.2: Continuous-energy (blue) and multigroup (red) total cross sections for natural carbon [34].

interest to the CERT project, as discussed in the results section.

2.4 Uncertainty Information in the ENDF Data and Multigroup Cross Sections

The ENDF/B data sets contain the evaluators' best estimate for the first and second moments of the cross sections. The first moments, or expectation values, are the best estimate for the cross sections. These will be referred to as the mean value of the cross sections in this work. The second moment of the cross-section distribution describes the uncertainty in the cross-section values as well as the correlations that exist across energy and across different reactions.

As defined in the NJOY manual [22], given two parameters x and y and the expectation operator $E\left[x\right]$, which returns the average value over the probability distribution function of x, the expected value of x is

$$x_0 = E\left[x\right]. \tag{2.4}$$

The covariance between x and y is defined

$$cov(x,y) = E\left[(x-x_0)(y-y_0)\right]. \tag{2.5}$$

The covariance between x with itself is the variance in x, σ_x^2,

$$cov(x,x) = \sigma_x^2 = E\left[(x-x_0)^2\right], \tag{2.6}$$

and the standard deviation in x, σ_x, is the square root of the variance in x.

The correlation between x and y is calculated from the covariance and standard deviations in x and y,

$$corr(x,y) = \frac{cov(x,y)}{\sigma_x \sigma_y}. \tag{2.7}$$

The covariance is therefore the product of the standard deviation of x, the standard deviation of y, and a correlation parameter. The correlation parameter is bound between -1 and 1.

The ENDF/B data sets contain uncertainty and correlation data for the pointwise nuclear data and for the resonance parameters introduced in Section 2.2. This data is processed by NJOY and reported in multigroup form as a vector of the mean value of the cross sections $\vec{x}_\mu$ and the covariance matrix for those cross sections $\overline{\overline{\Sigma_x}}$. The uncertainty in the multigroup cross sections is calculated by NJOY at infinite dilution [22]. The multigroup form of the covariance data produced by NJOY can then be used in a UQ analysis using multigroup neutron transport codes.

(As an aside, the use of the Greek letter sigma for summation, microscopic and macroscopic cross sections, variance and standard deviation, and the covariance matrix makes the clear communication of these concepts difficult. We have tried to keep the different uses of sigma unambiguous in the text.)

Given the first and second moments of the multigroup cross-section data $\vec{x}_\mu$ and $\overline{\overline{\Sigma_x}}$, the multigroup cross sections may be approximated as a multivariate Gaussian distribution,

$$\vec{x} \sim N(\vec{x}_\mu, \overline{\overline{\Sigma}}_x). \tag{2.8}$$

Drawing from this distribution generates realizations R_i of the cross sections x_{R_i} that reflect both the recorded uncertainty in the data and the correlations across energy groups and interactions. A computational method for drawing from this distribution is discussed in Section 4.4.1.

2.5 Cross Section Generation with NJOY

The cross-section-preparation code NJOY 2012 was used to generate multigroup cross sections and covariance matrices from the ENDF/B-VII.1 data in this work. NJOY is a product of Los Alamos National Laboratory. NJOY's capabilities extend far beyond those utilized in this work; an overview of the code's components used in this work is presented in this section.

A Python tool was written to generate the NJOY input files, run NJOY, and read the multigroup cross sections and covariance matrices from the NJOY output files.

The use of this tool is covered in detail in Appendix C.

2.5.1 NJOY Modules

NJOY 2012 processes the ENDF/B-VII.1 data files through a collection of modules to generate the multigroup cross sections. These calculations are performed for one isotope at a time. The code is run using a script comprised of consecutive input decks for each of these modules. The modules that are used to generate the multigroup cross sections and covariance data are briefly introduced in this section. Module names will be listed in all capital letters, following the NJOY convention. For a detailed description of the NJOY code, see [22].

For historical reasons, data files are referred to as "tapes" which are labeled only by number. Each module's deck specifies an input and output tape. NJOY requires each isotopes' ENDF/B data file, labeled as a tape, as input. These data files may be downloaded from the website of the Nuclear Information Service website managed by T-2 Division at Los Alamos National Laboratory [1].

The first NJOY module used in the generation of multigroup cross-section data is MODER. This module is used to convert text files to binary files and vice versa. This module is used to convert the human-readable ENDF/B file to binary for use by NJOY.

The RECONR module is used to read the ENDF/B data file and reconstruct an isotope's cross sections as a function of energy. This involves reading the pointwise data points and interpolation scheme information and reconstructing the resonances from the resonance parameters. This step produces a "pointwise-ENDF" or "PENDF" tape as output.

The BROADR module Doppler-broadens the cross sections for a given material temperature. BROADR requires as input the ENDF/B data file and the PENDF data file produced by the RECONR module and produces PENDF file as output.

The THERMR module is used to generate thermal cross sections for a given material temperature. This step can produce cross sections for thermal scattering in materials in which the material structure contributes significantly to the scattering, such as hydrogen in water, hydrogen in polyethylene, and carbon in graphite. THERMR requires as input the ENDF/B data file and the PENDF data file produced by the BROADR module. The PENDF file produced by BROADR is appended with the output of THERMR.

The GROUPR module produces multigroup cross sections from the PENDF tape produced by THERMR. In addition to a material temperature, an energy group structure and weighting spectrum must be provided in the GROUPR input. Custom group structures and weighting spectrums may be provided by the user, or the user may specify an option built into NJOY. A brief description of these default options are available in the NJOY manual. The GROUPR module produces a "groupwise-ENDF" or "GENDF" tape as output.

Finally, the ERRORR module calculates the covariance matrices associated with the multigroup cross sections generated with the GROUPR module. ERRORR requires the ENDF/B, PENDF, and GENDF data files as input. The material temperature, group structure, and weighting spectrum must be specified. ERRORR produces a covariance tape which may be further processed by modules which are not used in this work.

2.5.2 Data Produced by NJOY

The output tape produced by the ERRORR module is in a human-readable format. The multigroup cross sections and covariance matrices are read from this file. The form of this data is mentioned in this section for clarity.

Let us define a bit of notation before continuing. Cross sections are referred to as “removal” cross sections and as “transfer” cross sections in this work. Transfer cross sections appear as a matrix of cross sections from group g to group g', $\sigma_{g\to g'}$. Removal cross sections appear as a vector of cross sections for group g. For example, the elastic scattering cross sections are given as both a transfer matrix and as a vector of the removal cross section in each group, that is, the sum over g' of the transfer scattering cross sections $\sigma_{(n,el),g\to g'}$.

The multigroup cross sections are reported for all neutron interactions with at least one non-zero cross section. A transfer matrix is produced for all reactions that result in the production of one or more neutrons (except fission, in which case the fission spectrum χ is produced). In a few specific cases, removal cross sections produced by summing the transfer matrix across the groups to which the neutrons scatter g' was not equal to the removal cross section in that group for that reaction. The Python tool therefore sums the transfer matrices to produce the multigroup removal cross sections. The total cross section is similarly calculated by summing the removal cross sections for all reactions.

Covariance matrices are produced for each interaction with nonzero cross sections. Covariance data is available for the removal cross sections. Data is not available for the group-to-group components of the transfer matrix. This means that when samples are drawn from the cross-sections' multivariate Gaussian distribution, realized values of the transfer matrices must be calculated from the removal cross sections and the transfer matrices for the mean value of the cross sections. This was accomplished by multiplying the GROUPR-produced transfer scattering cross sections from group g by the ratio of the realized removal cross section, $\sigma_{R_i,g}$, to the mean value of the removal cross section, $\sigma_{\mu,g}$,

$$\sigma_{R_i,g\to g'} = \frac{\sigma_{R_i,g}}{\sigma_{\mu,g}}\sigma_{\mu,g\to g'}. \tag{2.9}$$

The multigroup cross sections are correlated across energy groups and across reactions. In some cases the cross sections are correlated across isotopes, but it proved extremely difficult to generate covariance data for all reactions across multiple isotopes. In a previous paper that considered only total and elastic scattering reactions, cross-isotope covariances were generated [7]. In this work, in order to include all neutron reactions, cross-isotope covariances were neglected.

The covariance matrices may be plotted to visualize the covariance data. Figure 2.3 shows the covariance matrix produced by NJOY 2012 for boron 10 using 99 energy groups and a weighting spectrum that follows a thermal Maxwellian in the low energy region, a 1/E spectrum in the slowing down region, and a fission spectrum at high energy. These cross sections were generated for one of the test problems in the results section.

The figure is organized by reaction and then ordered by energy group, from low energy to high energy, for each reaction. The white space in the figure represents regions for which the covariances are zero. The blocks of color near the diagonal correspond

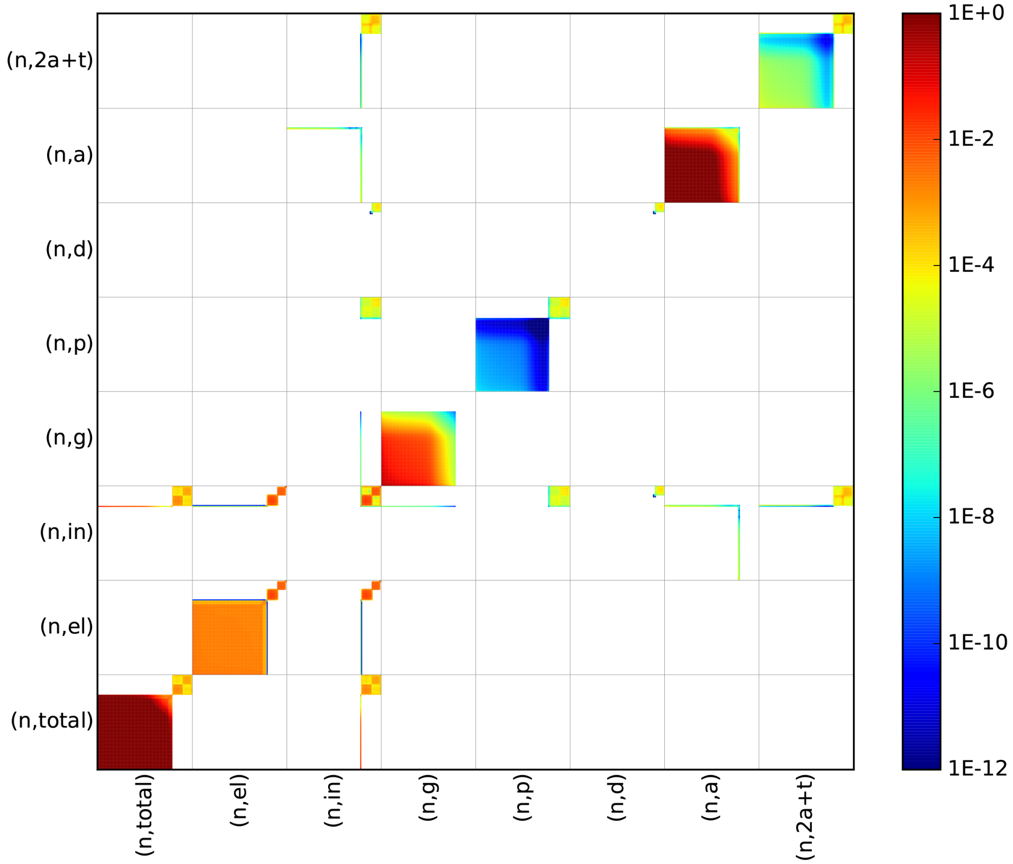

Figure 2.3: Covariance matrix for boron 10 produced with 99 energy groups and a thermal Maxwellian, 1/E, and fission spectrum weighting spectrum.

to cross-group covariance data. The (n,total),(n,total) block is expanded in Figure 2.4 to more closely examine this dense data. The off-diagonal data corresponds to covariances across reactions. In this case, all reactions are correlated to the inelastic scattering reaction.

Each diagonal block contains, in this example, 99 rows and 99 columns, corresponding to the 99 energy groups used in this problem. Data is displayed for eight different nuclear interactions; interactions for which all of the cross sections were zero were left out of the figure. Figure 2.5 shows the correlation matrix associated with this

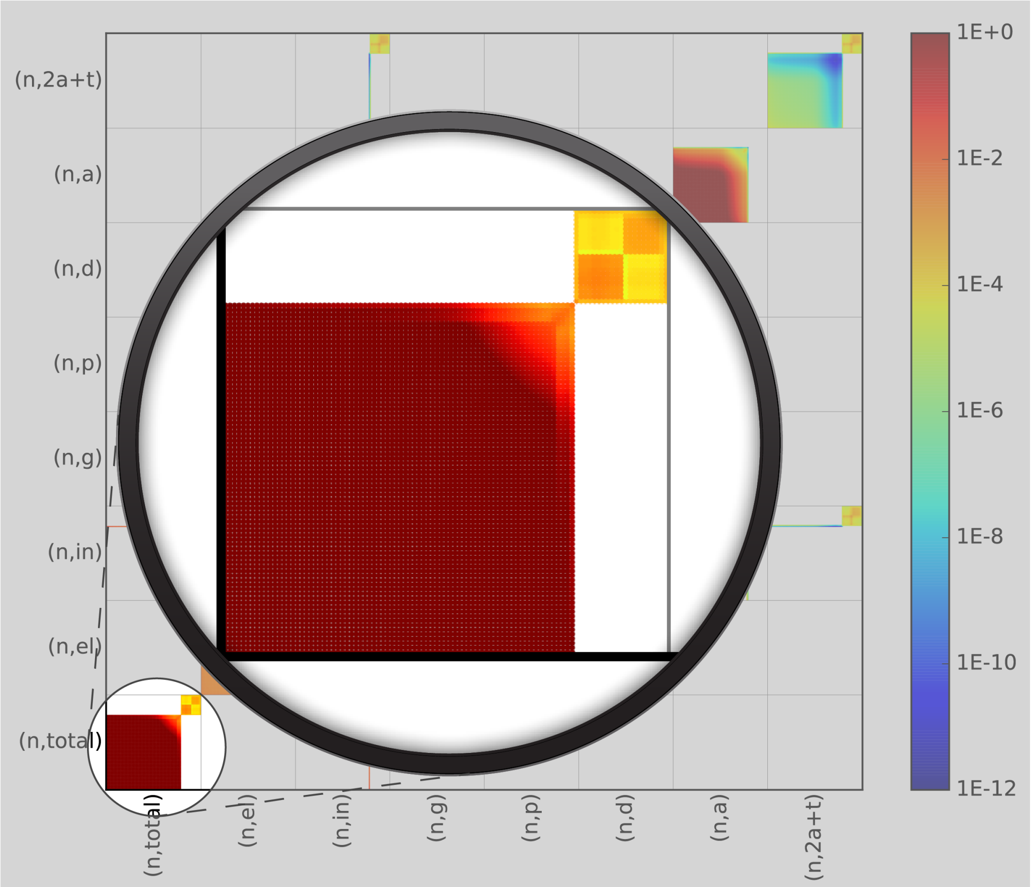

Figure 2.4: Enlarged view of a component of Figure 2.3, the covariance matrix for boron 10, to better visualize the data's fine structure.

covariance matrix.

The correlation matrix shows strong correlation between most energy groups for a given reaction and weak correlation across reactions.

2.6 Cross-Section Files for PDT

The deterministic transport code PDT was used to perform the radiation transport calculations analyzed in this work. For a thorough discussion of the radiation trans-

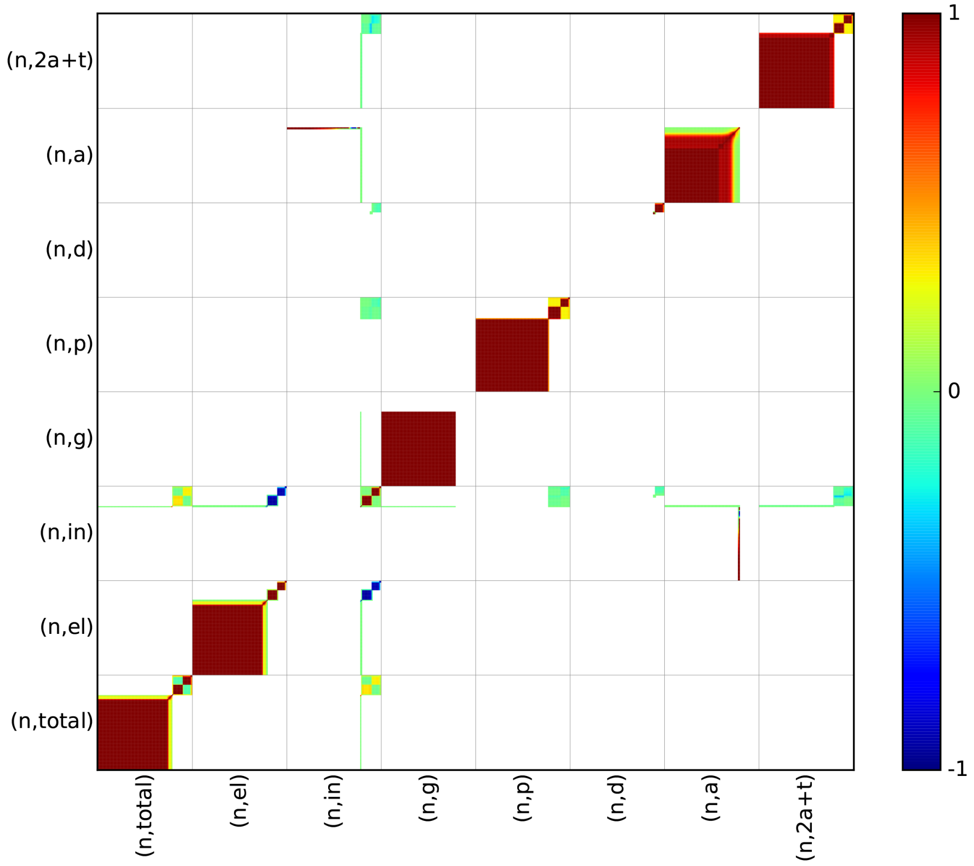

Figure 2.5: Correlation matrix for boron 10 produced with 99 energy groups and a thermal Maxwellian, 1/E, and fission spectrum weighting spectrum.

port equations, their discretization, and their solution, see Section 3. Here we briefly discuss the supply of nuclear data to PDT.

PDT requires a cross-section file for each isotope containing the temperature at which the cross sections were calculated, the energy group structure, the microscopic total cross section, and the microscopic total transfer matrix. Fissile isotopes must include data for the fission cross section, the fission spectrum, and the number of neutrons produced per fission. Alternatively, each individual reaction an isotope may undergo may be included in the cross-section file; in this case, PDT determines the total cross section and total transfer matrix internally.

The PDT input file specifies the number density of each isotope in each region, from which the macroscopic cross sections are calculated, and the location of each nuclear data file. A simple Python tool was written as a part of this work to write PDT-formatted nuclear data files given a set of multigroup cross sections.

3. THE EQUATIONS OF RADIATION TRANSPORT

The so-called neutron transport equation...strikes far more terror in the hearts of fledgling nuclear engineers who are intimidated by its frightening reputation within the nuclear reactor community. Neutron transport theory has come to be associated with a hideous plethora of impenetrable mathematics, unwieldy formulas, and (eventually) the expenditure of enormous amounts of money on computer number-crunching.

— Duderstadt and Hamilton, *Nuclear Reactor Analysis*, page 104 [14]

3.1 The Neutron Transport Equation

The behavior of neutrons in a background material is accurately described by the linear Boltzmann equation for sparse gases, which is also known as the neutron transport equation [14]. The principal unknown is the neutron angular flux ψ, which has units of neutrons per second per steradian per energy per centimeter squared. The angular flux depends upon independent parameters in time t, space r, angle Ω, and energy E.

$$\begin{aligned}\frac{1}{v}\frac{\partial\psi}{\partial t}+&\vec{\Omega}\cdot\triangledown\psi(\vec{r},E,\vec{\Omega},t)+\Sigma_t(\vec{r},E)\psi(\vec{r},E,\vec{\Omega},t)= \\ &\int_{4\pi}\int_0^\infty\Sigma_s(\vec{r},E'\to E,\vec{\Omega}'\to\vec{\Omega})\psi(\vec{r},E',\vec{\Omega}',t)dE'd\Omega'+s(\vec{r},E,\vec{\Omega},t).\end{aligned} \tag{3.1}$$

Here v is the neutron speed in units of cm/s, Σ_t and Σ_s are the macroscopic total and scattering cross sections with units of 1/cm, and s is a source term with units of neutrons per cm^2 per second per steradian.

The real-world problems considered in this work do not vary in time. This simplifies the neutron transport equation by eliminating the time derivative and time dependence of ψ,

$$\begin{aligned}\vec{\Omega} \cdot \nabla\psi(\vec{r}, E, \vec{\Omega}) + &\Sigma_t(\vec{r}, E)\psi(\vec{r}, E, \vec{\Omega}) = \\ &\int_{4\pi}\int_0^\infty \Sigma_s(\vec{r}, E' \to E, \vec{\Omega}' \to \vec{\Omega})\psi(\vec{r}, E', \vec{\Omega}')dE'd\vec{\Omega}' + s(\vec{r}, E, \vec{\Omega}).\end{aligned} \tag{3.2}$$

The scattering source on the right hand side of the transport equation requires further attention. The integral over angle of the angular flux ψ is the scalar flux ϕ,

$$\phi(\vec{r}, E) = \int_{4\pi} \psi(\vec{r}, E, \vec{\Omega})d\Omega. \tag{3.3}$$

In the case of isotropic scattering the scattering source has no angular dependence and can be simplified, appearing

$$\int_{4\pi}\int_0^\infty \Sigma_s(\vec{r}, E' \to E, \vec{\Omega}' \to \vec{\Omega})\psi(\vec{r}, E', \vec{\Omega}')dE'd\Omega' = \frac{1}{4\pi}\int_0^\infty \Sigma_s(\vec{r}, E' \to E)\phi(\vec{r}, E')dE'. \tag{3.4}$$

To accurately model the anisotropic behavior of neutrons scattering in realistic problems, the scattering operator is typically expressed using a spherical harmonics expansion of the angular flux [6][14][24]. The scattering cross sections can be expanded using the Legendre polynomials P_ℓ^0,

$$\Sigma_s(\vec{r}, E' \to E, \vec{\Omega}' \to \vec{\Omega}) = \sum_{\ell=0}^{\infty} \frac{2\ell+1}{4\pi}\Sigma_{s,\ell}(\vec{r}, E' \to E)P_\ell^0(\vec{\Omega}' \cdot \vec{\Omega}). \tag{3.5}$$

We define the dot product of $\vec{\Omega}'$ and $\vec{\Omega}$ as ξ_s, the cosine of the scattering angle. The scattering cross sections depend only upon the change in direction of the neutron, so we replace $\vec{\Omega}' \to \vec{\Omega}$ with ξ_s. With this notation,

$$\Sigma_s(\vec{r}, E' \to E, \xi_s) = \sum_{\ell=0}^{\infty} \frac{2\ell+1}{4\pi} \Sigma_{s,\ell}(\vec{r}, E' \to E) P_\ell^0(\xi_s). \tag{3.6}$$

The Legendre moments of the cross sections $\sigma_{s,\ell}$ are produced from the ENDF files by NJOY and are easily converted to the macroscopic $\Sigma_{s,\ell}$.

The angular flux can similarly be expanded in the spherical harmonics. Defining Y_ℓ^m as the spherical harmonic function of degree ℓ and order m, the expanded angular flux is

$$\psi(\vec{r}, E, \vec{\Omega}) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{+\ell} \frac{2\ell+1}{4\pi} \phi_\ell^m(\vec{r}, E) Y_\ell^m(\vec{\Omega}). \tag{3.7}$$

The expansion coefficients for the angular flux moments ϕ_ℓ^m can be obtained from this expression by taking advantage of the orthogonality of the spherical harmonic functions, yielding

$$\phi_\ell^m(\vec{r}, E) = \int_{4\pi} \psi(\vec{r}, E, \vec{\Omega}) Y_\ell^m(\vec{\Omega}) d\Omega. \tag{3.8}$$

Finally, the scattering term is assembled from the moments of the scattering cross section in Eq. (3.6) and the moments of the scalar flux in Eq. (3.8) in a long process involving the addition theorem,

$$\begin{aligned} &\int_{4\pi} \int_0^{\infty} \Sigma_s(\vec{r}, E' \to E, \vec{\Omega}' \to \vec{\Omega}) \psi(\vec{r}, E', \vec{\Omega}') dE' d\Omega' = \\ &\qquad \int_0^{\infty} \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{+\ell} \frac{2\ell+1}{4\pi} \Sigma_{s,\ell}(\vec{r}, E' \to E) \phi_\ell^m(\vec{r}, E') Y_\ell^m(\vec{\Omega}) dE'. \end{aligned} \tag{3.9}$$

The anisotropic scattering order is truncated after some number of moments L. The scattering source is then

$$\int_{4\pi}\int_0^\infty \Sigma_s(\vec{r}, E' \to E, \vec{\Omega}' \to \vec{\Omega})\psi(\vec{r}, E', \vec{\Omega}')dE'd\Omega' = \tag{3.10}$$
$$\int_0^\infty \sum_{\ell=0}^{L}\sum_{m=-\ell}^{+\ell} \frac{2\ell+1}{4\pi}\Sigma_{s,\ell}(\vec{r}, E' \to E)\phi_\ell^m(\vec{r}, E')Y_\ell^m(\vec{\Omega})dE'.$$

The moments of the scattering cross section $\Sigma_{s,\ell}$ are calculated from the evaluated nuclear data files by the data processing code NJOY, discussed in Section 2.3.

3.2 Discretization of the Neutron Transport Equation in PDT

The discretization of the neutron transport equation is a topic of overwhelming breadth. The discretization schemes used in the research code "PDT" being developed at Texas A&M University and used in all of the calculations performed in this work are outlined in the following paragraphs, but this discussion is by no means comprehensive.

3.2.1 Angular Discretization: Discrete Ordinates

The angular flux variable is discretized in angle using the discrete ordinates (S_N) approximation, in which the solution is calculated in discrete directions D that correspond to quadrature points used to numerically integrate the angular flux. The angle-integrated scalar flux $\phi(\vec{r}, E)$ is computed from the angular flux by a quadrature rule, that is,

$$\phi(\vec{r}, E) = \int_{4\pi}\psi(\vec{r}, E, \vec{\Omega})d\Omega \approx \sum_{d=1}^{D} w_d\psi_d(\vec{r}, E). \tag{3.11}$$

There are many methods by which quadrature points may be chosen. The “product Gauss Legendre Chebyshev” quadrature set was applied for the problems considered in this work [3][27]. This discretization uses a Gauss-Legendre quadrature set in the polar ($z-$) direction and a Gauss-Chebyshev quadrature set in the azimuthal (x- and y-) directions. Three-dimensional slab-like problems were defined such that the direction of interest corresponds to the $z-$ direction; in this case, the problem is similar to a one-dimensional problem using the Gauss-Legendre quadrature points.

The product Gauss Legendre Chebyshev quadrature set requires the specification of the number of polar and azimuthal angles per octant. The quadrature points are then chosen according to the formulas associated with this scheme (as in [3]). The number of angles required to resolve the angular flux is highly problem dependent and requires a resolution study to examine the sensitivity of the QoI to both the polar and azimuthal angular resolution.

3.2.2 Energy Discretization: Multigroup

The angular flux variable is discretized in energy using the multigroup approximation. Energy dependent quantities such as the angular flux and microscopic cross sections are assumed to be piecewise constant over the width of each energy group. As with the angular discretization, the number of energy groups required to resolve the solution is highly problem dependent. The discretization of the continuous energy variable into groups and the calculation of multigroup cross sections was discussed in Section 2.3.

Integrating the scalar flux over energy is now accomplished by summing over the G

energy groups,

$$\int_0^\infty \phi(\vec{r}, E)dE \approx \sum_{g=1}^{G} \phi_g(\vec{r}). \tag{3.12}$$

The multigroup form of the continuous energy anisotropic scattering source, Eq. (3.10), is

$$\begin{aligned}\int_{4\pi}\int_0^\infty \Sigma_s(\vec{r}, E' \to E, \vec{\Omega}' \to \vec{\Omega})\psi(\vec{r}, E', \vec{\Omega}')dE'd\Omega' = \\ \sum_{g'=1}^{G}\sum_{\ell=0}^{L}\sum_{m=-\ell}^{+\ell} \frac{2\ell+1}{4\pi}\Sigma_{s,\ell}(\vec{r}, g' \to g)\phi_\ell^m(\vec{r}, g')Y_\ell^m(\vec{\Omega}).\end{aligned} \tag{3.13}$$

3.2.3 Spatial Discretization: Piece-Wise Linear Discontinuous Finite Elements

The spatial discretization of the neutron transport equation is accomplished using a discontinuous finite element method. All of the calculations performed in the scope of this work used the piece-wise linear discontinuous (PWLD) spatial discretization [5]. The PWLD discretization assigns a basis function to each vertex of a spatial cell (that is, four basis functions per cell in two dimensions using quadrilaterals and eight basis functions per cell in three dimensions using hexahedrons). The basis functions are tent functions which are one at their vertex and linearly decrease to zero at the other cell vertices. Boundary conditions for each cell are determined from the solution in the upwind cell.

The basis functions in each cell have an associated volume. The cell-averaged angular flux in cell c, ψ_{c_A}, is the sum of the angular flux coefficient associated with each basis function $\psi_{c,b}$ and a volume term associated with each basis function,

$$\psi_{c_A}(E,\vec{\Omega}) = \sum_b V_{c,b}\psi_{c,b}(E,\vec{\Omega}). \tag{3.14}$$

The PWLD discretization has several properties that are desirable for radiation transport calculations. The scheme can be applied to arbitrary cells, is second order accurate in space, and preserves the thick diffusion limit [5]. For these reasons the PWLD discretization is the most commonly used spatial discretization in PDT calculations.

3.2.4 Operator Form of the Discretized Transport Equation

The discretization of the transport equation in space, angle, and energy yields a collection of coupled equations. The total number of coupled equations is the product of the number of spatial cells, the number of basis functions per spatial cell, the number of energy groups, and the total number of angular directions (which is itself the number of polar angles per quadrant times the number of azimuthal angles per octant times eight octants on the unit sphere). For convenience these coupled equations are often written in operator or matrix form.

In this notation the fully discretized angular flux is represented as the vector ψ. The streaming and total interaction terms are collected into the matrix L,

$$L\psi = \vec{\Omega} \cdot \nabla\psi(\vec{r},E,\vec{\Omega}) + \Sigma_t(\vec{r},E)\psi(\vec{r},E,\vec{\Omega}). \tag{3.15}$$

The anisotropic scattering operator is represented S,

$$S\psi = \sum_{g'=1}^{G} \sum_{\ell=0}^{L} \sum_{m=-\ell}^{+\ell} \frac{2\ell+1}{4\pi} \Sigma_{s,\ell}(\vec{r}, g' \to g) \phi_\ell^m(\vec{r}, g') Y_\ell^m(\vec{\Omega}), \tag{3.16}$$

with Eq. (3.8) used to calculate moments of the scalar flux from the angular flux. Finally, the inhomogeneous neutron source is represented as q. Combining these, the radiation transport equation can be written

$$L\psi = S\psi + q. \tag{3.17}$$

This compact notation for the fully discretized transport equation will be used from this point forward.

3.3 Inner Product Definition

It is useful to define a compact notation for the inner product of vectors a and b. We define the inner product as the integration over all phase space (that is, angle $\vec{\Omega}$ and spatial domain $\mathcal{D}$) of the multigroup functions $a_g(\vec{r}, \vec{\Omega})$ and $b_g(\vec{r}, \vec{\Omega})$,

$$\langle a, b \rangle = \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \sum_g a_g(\vec{r}, \vec{\Omega}) b_g(\vec{r}, \vec{\Omega}). \tag{3.18}$$

For the discretized angular flux, this integration is accomplished by summing the cell-averaged values of the two functions a_{c_A} and b_{c_A} over the energy groups g, discrete angles d with associated quadrature weights w_d, and spatial cells c with volumes V_c. This summation is

$$\langle a, b \rangle = \sum_d w_d \sum_c V_c \sum_g a_{c_A,d,g} b_{c_A,d,g}. \tag{3.19}$$

Using this definition of the inner product, we define an adjoint matrix $M^\dagger$ such that for matrix M,

$$\langle Ma, b \rangle = \left\langle a, M^\dagger b \right\rangle . \tag{3.20}$$

This definition of an adjoint operator in the context of inner products will be used to derive the adjoint equation and the equations for the sensitivity coefficients.

We note one final property of the inner product that will prove useful in this work. By the inner product's definition, Eq. (3.18), it is clear that the inner product of a vector with the identity matrix $\overline{\overline{I}}$ is equal to the inner product of the vector alone,

$$\left\langle \overline{\overline{I}}, a \right\rangle = \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \sum_g \overline{\overline{I}} a = \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \sum_g a = \langle a \rangle . \tag{3.21}$$

3.4 The Quantity of Interest

The quantities of interest Q examined in this work are reaction rates, which are integral quantities of the scalar flux and the macroscopic cross section corresponding to the reaction of interest in a region of interest such as a detector volume. Let us define the multigroup microscopic cross section that corresponds to the reaction of interest R in the region of interest as $\sigma_{R,g}(\vec{r})$ and the corresponding macroscopic cross section as $\Sigma_{R,g}(\vec{r})$. The QoI is calculated in PDT from the discretized angular flux as

$$Q = \sum_d w_d \sum_c V_c \sum_g \Sigma_{R,g} \psi_{c_A,d,g} . \tag{3.22}$$

This is the inner product of the angular flux and the cross section corresponding to the reaction of interest,

$$Q = \langle \Sigma_r, \psi \rangle . \tag{3.23}$$

As a concrete example of the quantity of interest, consider a problem that contains a neutron source and a neutron detector. The QoI is specified as the count rate in the detector. The cross section Σ_r is any cross section in the materials present in the detector which produce detection events. These cross sections only appear within the detector volume. The inner product of Σ_r and ψ then yields the number of interactions in the detector per second that should yield detection events.

The term that is multiplied by the angular flux ψ and then integrated over the phase space to generate the QoI is called a response function and denoted $q^\dagger$. The QoI is now generally defined as

$$Q = \left\langle \psi, q^\dagger \right\rangle . \tag{3.24}$$

3.5 Derivation of the Adjoint Neutron Transport Equations

The adjoint problem is mathematically related to the forward problem and used to solve for the sensitivity of the quantity of interest Q to uncertain input parameters p. The uncertain parameters in this work are the microscopic cross sections σ, as introduced in Section 2.1. The derivation of the adjoint equation follows the variational approach, as presented in [6]. Stripling ([28] and [29], citing primarily [10]) followed this approach and implemented the resulting adjoint equations in PDT.

The derivation is performed entirely in terms of the operator forms of the fully discretized equations introduced in Section 3.2.4. The inner product notation from Section 3.3 and the $q^\dagger$ term defined in Section 3.4 are used extensively in this derivation.

3.5.1 Source-Detector Form of the Neutron Transport Equations

The derivation of the adjoint sensitivity equations begins by forming a Lagrangian system $\mathcal{L}$ by taking an inner product of the discretized radiation transport equation Eq (3.17) with the as-of-yet undefined adjoint function $\psi^\dagger$ and subtracting it from the QoI, Eq. (3.24),

$$\mathcal{L} = \left\langle \psi, q^\dagger \right\rangle - \left\langle \psi^\dagger, L\psi - S\psi - q \right\rangle, \tag{3.25}$$

$$\mathcal{L} = \left\langle \psi, q^\dagger \right\rangle - \left\langle \psi^\dagger, L\psi \right\rangle + \left\langle \psi^\dagger, S\psi \right\rangle + \left\langle \psi^\dagger, q \right\rangle. \tag{3.26}$$

The Lagrangian is a function of the angular flux ψ, the adjoint angular flux $\psi^\dagger$, and the parameters of interest p,

$$\mathcal{L} = \mathcal{L}(\psi, \psi^\dagger, p), \tag{3.27}$$

but this notation is suppressed to simplify the following algebra.

We will see that the as-yet-undefined $\psi^\dagger$ is a Lagrangian multiplier. As long as ψ satisfies the forward problem, the Lagrangian is identically equal to the QoI. The derivative of $\mathcal{L}$ with respect to the uncertain parameters p is obtained through the chain rule,

$$\frac{d\mathcal{L}}{dp} = \frac{\partial \mathcal{L}}{\partial p} + \frac{\partial \mathcal{L}}{\partial \psi}\frac{\partial \psi}{\partial p} + \frac{\partial \mathcal{L}}{\partial \psi^\dagger}\frac{\partial \psi^\dagger}{\partial p}. \tag{3.28}$$

Applying the chain rule to Eq. (3.26),

$$\begin{aligned} \frac{d\mathcal{L}}{dp} =& \frac{\partial}{\partial p}\left[\left\langle \psi, q^\dagger \right\rangle - \left\langle \psi^\dagger, L\psi \right\rangle + \left\langle \psi^\dagger, S\psi \right\rangle + \left\langle \psi^\dagger, q \right\rangle\right] \\ &+ \frac{\partial}{\partial \psi}\left[\left\langle \psi, q^\dagger \right\rangle - \left\langle \psi^\dagger, L\psi \right\rangle + \left\langle \psi^\dagger, S\psi \right\rangle + \left\langle \psi^\dagger, q \right\rangle\right]\frac{\partial \psi}{\partial p} \\ &+ \frac{\partial}{\partial \psi^\dagger}\left[\left\langle \psi, q^\dagger \right\rangle - \left\langle \psi^\dagger, L\psi \right\rangle + \left\langle \psi^\dagger, S\psi \right\rangle + \left\langle \psi^\dagger, q \right\rangle\right]\frac{\partial \psi^\dagger}{\partial p}. \end{aligned} \tag{3.29}$$

In general, the sensitivity of the solution of the radiation transport equation to the uncertain parameters $\frac{\partial\psi}{\partial p}$ is unknown and difficult to calculate. In fact, it is usually more difficult to determine the sensitivity of ψ in all phase space to p, $\frac{\partial\psi}{\partial p}$, than the sensitivity of the scalar QoI Q to the parameters, $\frac{\partial Q}{\partial p}$. Similarly, $\frac{\partial\psi^\dagger}{\partial p}$ is not known and not easily calculated. The strategy to derive the sensitivity coefficients is therefore to manipulate Eq. (3.29) to eliminate $\frac{\partial\psi}{\partial p}$ and $\frac{\partial\psi^\dagger}{\partial p}$ and leave only calculable terms in the expression for $\frac{d\mathcal{L}}{dp}$.

Applying the partial derivatives to Eq. (3.29) and applying the chain rule to the operator terms

$$\frac{\partial}{\partial p}\left(L\psi\right) = \left(\frac{\partial}{\partial p}L\right)\psi + L\left(\frac{\partial}{\partial p}\psi\right) \tag{3.30}$$

and all other terms yields

$$\begin{aligned}
\frac{d\mathcal{L}}{dp} = \Bigg[& \left\langle \frac{\partial}{\partial p}\psi, q^\dagger \right\rangle + \left\langle \psi, \frac{\partial}{\partial p} q^\dagger \right\rangle \\
& - \left\langle \frac{\partial}{\partial p}\psi^\dagger, L\psi \right\rangle - \left\langle \psi^\dagger, \left(\frac{\partial}{\partial p}L\right)\psi \right\rangle - \left\langle \psi^\dagger, L\left(\frac{\partial}{\partial p}\psi\right) \right\rangle \\
& + \left\langle \frac{\partial}{\partial p}\psi^\dagger, S\psi \right\rangle + \left\langle \psi^\dagger, \left(\frac{\partial}{\partial p}S\right)\psi \right\rangle + \left\langle \psi^\dagger, S\left(\frac{\partial}{\partial p}\psi\right) \right\rangle \\
& + \left\langle \frac{\partial}{\partial p}\psi^\dagger, q \right\rangle + \left\langle \psi^\dagger, \frac{\partial}{\partial p} q \right\rangle \Bigg] \\
+ \Bigg[& \left\langle \frac{\partial}{\partial \psi}\psi, q^\dagger \right\rangle + \left\langle \psi, \frac{\partial}{\partial \psi} q^\dagger \right\rangle \\
& - \left\langle \frac{\partial}{\partial \psi}\psi^\dagger, L\psi \right\rangle - \left\langle \psi^\dagger, \left(\frac{\partial}{\partial \psi}L\right)\psi \right\rangle - \left\langle \psi^\dagger, L\left(\frac{\partial}{\partial \psi}\psi\right) \right\rangle \\
& + \left\langle \frac{\partial}{\partial \psi}\psi^\dagger, S\psi \right\rangle + \left\langle \psi^\dagger, \left(\frac{\partial}{\partial \psi}S\right)\psi \right\rangle + \left\langle \psi^\dagger, S\left(\frac{\partial}{\partial \psi}\psi\right) \right\rangle \\
& + \left\langle \frac{\partial}{\partial \psi}\psi^\dagger, q \right\rangle + \left\langle \psi^\dagger, \frac{\partial}{\partial \psi} q \right\rangle \Bigg] \frac{\partial \psi}{\partial p} \\
+ \Bigg[& \left\langle \frac{\partial}{\partial \psi^\dagger}\psi, q^\dagger \right\rangle + \left\langle \psi, \frac{\partial}{\partial \psi^\dagger} q^\dagger \right\rangle \\
& - \left\langle \frac{\partial}{\partial \psi^\dagger}\psi^\dagger, L\psi \right\rangle - \left\langle \psi^\dagger, \left(\frac{\partial}{\partial \psi^\dagger}L\right)\psi \right\rangle - \left\langle \psi^\dagger, L\left(\frac{\partial}{\partial \psi^\dagger}\psi\right) \right\rangle \\
& + \left\langle \frac{\partial}{\partial \psi^\dagger}\psi^\dagger, S\psi \right\rangle + \left\langle \psi^\dagger, \left(\frac{\partial}{\partial \psi^\dagger}S\right)\psi \right\rangle + \left\langle \psi^\dagger, S\left(\frac{\partial}{\partial \psi^\dagger}\psi\right) \right\rangle \\
& + \left\langle \frac{\partial}{\partial \psi^\dagger}\psi^\dagger, q \right\rangle + \left\langle \psi^\dagger, \frac{\partial}{\partial \psi^\dagger} q \right\rangle \Bigg] \frac{\partial \psi^\dagger}{\partial p}.
\end{aligned} \tag{3.31}$$

Many of these terms are zero. The derivative of the adjoint angular flux $\psi^\dagger$ with respect to the angular flux ψ is zero, $\frac{\partial}{\partial\psi}\psi^\dagger = 0$; similarly, $\frac{\partial}{\partial\psi^\dagger}\psi = 0$. In this project, the driving source q does not depend upon the angular flux ψ or the cross sections, which are the parameter of interest in this work, so $\frac{\partial}{\partial\psi}q = \frac{\partial}{\partial\psi^\dagger}q = \frac{\partial}{\partial p}q = 0$.

The inner product of a vector with zero is zero, so these terms are removed from the equation for the derivative of the Lagrangian with respect to the parameters of interest. Eq. (3.31) is repeated here with terms that are zero denoted with a strikethrough,

$$
\begin{aligned}
\frac{d\mathcal{L}}{dp} = \Bigg[& \left\langle \frac{\partial}{\partial p}\psi, q^\dagger \right\rangle + \left\langle \psi, \frac{\partial}{\partial p}q^\dagger \right\rangle \\
& - \left\langle \frac{\partial}{\partial p}\psi^\dagger, L\psi \right\rangle - \left\langle \psi^\dagger, \left(\frac{\partial}{\partial p}L\right)\psi \right\rangle - \left\langle \psi^\dagger, L\left(\frac{\partial}{\partial p}\psi\right) \right\rangle \\
& + \left\langle \frac{\partial}{\partial p}\psi^\dagger, S\psi \right\rangle + \left\langle \psi^\dagger, \left(\frac{\partial}{\partial p}S\right)\psi \right\rangle + \left\langle \psi^\dagger, S\left(\frac{\partial}{\partial p}\psi\right) \right\rangle \\
& + \left\langle \frac{\partial}{\partial p}\psi^\dagger, q \right\rangle + \cancel{\left\langle \psi^\dagger, \frac{\partial}{\partial p}q \right\rangle} \Bigg] \\
+ \Bigg[& \left\langle \frac{\partial}{\partial \psi}\psi, q^\dagger \right\rangle + \cancel{\left\langle \psi, \frac{\partial}{\partial \psi}q^\dagger \right\rangle} \\
& - \cancel{\left\langle \frac{\partial}{\partial \psi}\psi^\dagger, L\psi \right\rangle} - \cancel{\left\langle \psi^\dagger, \left(\frac{\partial}{\partial \psi}L\right)\psi \right\rangle} - \left\langle \psi^\dagger, L\left(\frac{\partial}{\partial \psi}\psi\right) \right\rangle \\
& + \cancel{\left\langle \frac{\partial}{\partial \psi}\psi^\dagger, S\psi \right\rangle} + \cancel{\left\langle \psi^\dagger, \left(\frac{\partial}{\partial \psi}S\right)\psi \right\rangle} + \left\langle \psi^\dagger, S\left(\frac{\partial}{\partial \psi}\psi\right) \right\rangle \\
& + \cancel{\left\langle \frac{\partial}{\partial \psi}\psi^\dagger, q \right\rangle} + \cancel{\left\langle \psi^\dagger, \frac{\partial}{\partial \psi}q \right\rangle} \Bigg] \frac{\partial \psi}{\partial p} \\
+ \Bigg[& \cancel{\left\langle \frac{\partial}{\partial \psi^\dagger}\psi, q^\dagger \right\rangle} + \cancel{\left\langle \psi, \frac{\partial}{\partial \psi^\dagger}q^\dagger \right\rangle} \\
& - \left\langle \frac{\partial}{\partial \psi^\dagger}\psi^\dagger, L\psi \right\rangle - \cancel{\left\langle \psi^\dagger, \left(\frac{\partial}{\partial \psi^\dagger}L\right)\psi \right\rangle} - \cancel{\left\langle \psi^\dagger, L\left(\frac{\partial}{\partial \psi^\dagger}\psi\right) \right\rangle} \\
& + \left\langle \frac{\partial}{\partial \psi^\dagger}\psi^\dagger, S\psi \right\rangle + \cancel{\left\langle \psi^\dagger, \left(\frac{\partial}{\partial \psi^\dagger}S\right)\psi \right\rangle} + \cancel{\left\langle \psi^\dagger, S\left(\frac{\partial}{\partial \psi^\dagger}\psi\right) \right\rangle} \\
& + \left\langle \frac{\partial}{\partial \psi^\dagger}\psi^\dagger, q \right\rangle + \cancel{\left\langle \psi^\dagger, \frac{\partial}{\partial \psi^\dagger}q \right\rangle} \Bigg] \frac{\partial \psi^\dagger}{\partial p}.
\end{aligned}
\tag{3.32}
$$

Dropping the terms that are equal to zero, this reduces to

$$\begin{aligned}\frac{d\mathcal{L}}{dp} = & \left[\left\langle \frac{\partial}{\partial p}\psi, q^\dagger \right\rangle + \left\langle \psi, \frac{\partial}{\partial p} q^\dagger \right\rangle \right. \\ & \qquad - \left\langle \frac{\partial}{\partial p}\psi^\dagger, L\psi \right\rangle - \left\langle \psi^\dagger, \left(\frac{\partial}{\partial p}L\right)\psi \right\rangle - \left\langle \psi^\dagger, L\left(\frac{\partial}{\partial p}\psi\right) \right\rangle \\ & \qquad + \left\langle \frac{\partial}{\partial p}\psi^\dagger, S\psi \right\rangle + \left\langle \psi^\dagger, \left(\frac{\partial}{\partial p}S\right)\psi \right\rangle + \left\langle \psi^\dagger, S\left(\frac{\partial}{\partial p}\psi\right) \right\rangle \\ & \qquad \left. + \left\langle \frac{\partial}{\partial p}\psi^\dagger, q \right\rangle \right] \\ & + \left[\left\langle \frac{\partial}{\partial \psi}\psi, q^\dagger \right\rangle - \left\langle \psi^\dagger, L\left(\frac{\partial}{\partial \psi}\psi\right) \right\rangle + \left\langle \psi^\dagger, S\left(\frac{\partial}{\partial \psi}\psi\right) \right\rangle \right] \frac{\partial \psi}{\partial p} \\ & + \left[-\left\langle \frac{\partial}{\partial \psi^\dagger}\psi^\dagger, L\psi \right\rangle + \left\langle \frac{\partial}{\partial \psi^\dagger}\psi^\dagger, S\psi \right\rangle + \left\langle \frac{\partial}{\partial \psi^\dagger}\psi^\dagger, q \right\rangle \right] \frac{\partial \psi^\dagger}{\partial p}. \end{aligned} \tag{3.33}$$

The derivative of a vector with respect to a vector is a Jacobian matrix. The derivative of ψ with respect to ψ is the identity matrix I. This simplifies the last two lines of Eq. (3.33),

$$\begin{aligned}\frac{d\mathcal{L}}{dp} = & \left[\left\langle \frac{\partial}{\partial p}\psi, q^\dagger \right\rangle + \left\langle \psi, \frac{\partial}{\partial p} q^\dagger \right\rangle \right. \\ & \qquad - \left\langle \frac{\partial}{\partial p}\psi^\dagger, L\psi \right\rangle - \left\langle \psi^\dagger, \left(\frac{\partial}{\partial p}L\right)\psi \right\rangle - \left\langle \psi^\dagger, L\left(\frac{\partial}{\partial p}\psi\right) \right\rangle \\ & \qquad + \left\langle \frac{\partial}{\partial p}\psi^\dagger, S\psi \right\rangle + \left\langle \psi^\dagger, \left(\frac{\partial}{\partial p}S\right)\psi \right\rangle + \left\langle \psi^\dagger, S\left(\frac{\partial}{\partial p}\psi\right) \right\rangle \\ & \qquad \left. + \left\langle \frac{\partial}{\partial p}\psi^\dagger, q \right\rangle \right] \\ & + \left[\left\langle I, q^\dagger \right\rangle - \left\langle \psi^\dagger, LI \right\rangle + \left\langle \psi^\dagger, SI \right\rangle \right] \frac{\partial \psi}{\partial p} \\ & + \left[-\left\langle I, L\psi \right\rangle + \left\langle I, S\psi \right\rangle + \left\langle I, q \right\rangle \right] \frac{\partial \psi^\dagger}{\partial p}. \end{aligned} \tag{3.34}$$

The terms within the three sets of brackets are rearranged to further clarify this expression.

$$\begin{aligned}\frac{d\mathcal{L}}{dp} = &\Bigg[\left\langle \frac{\partial}{\partial p}\psi, q^\dagger \right\rangle - \left\langle \psi^\dagger, L\left(\frac{\partial}{\partial p}\psi\right)\right\rangle + \left\langle \psi^\dagger, S\left(\frac{\partial}{\partial p}\psi\right)\right\rangle \\ &\quad - \left\langle \frac{\partial}{\partial p}\psi^\dagger, L\psi \right\rangle + \left\langle \frac{\partial}{\partial p}\psi^\dagger, S\psi \right\rangle + \left\langle \frac{\partial}{\partial p}\psi^\dagger, q \right\rangle \\ &\quad + \left\langle \psi, \frac{\partial}{\partial p}q^\dagger \right\rangle - \left\langle \psi^\dagger, \left(\frac{\partial}{\partial p}L\right)\psi \right\rangle + \left\langle \psi^\dagger, \left(\frac{\partial}{\partial p}S\right)\psi \right\rangle\Bigg] \\ &+ \left[- \left\langle \psi^\dagger, LI \right\rangle + \left\langle \psi^\dagger, SI \right\rangle + \left\langle I, q^\dagger \right\rangle\right] \frac{\partial \psi}{\partial p} \\ &+ \left[- \left\langle I, L\psi \right\rangle + \left\langle I, S\psi \right\rangle + \left\langle I, q \right\rangle\right] \frac{\partial \psi^\dagger}{\partial p}.\end{aligned} \tag{3.35}$$

The matrix property of the inner product definition, Eq. (3.20), is repeatedly applied to terms in the first and second sets of brackets. The adjoint matrices are left undefined for now, but will be derived and compared to the forward operators in in Section 3.6. After the application of Eq. (3.20) the resulting terms are reordered for clarity.

$$\begin{aligned}\frac{d\mathcal{L}}{dp} = &\Bigg[- \left\langle \frac{\partial}{\partial p}\psi, L^\dagger\psi^\dagger \right\rangle + \left\langle \frac{\partial}{\partial p}\psi, S^\dagger\psi^\dagger \right\rangle + \left\langle \frac{\partial}{\partial p}\psi, q^\dagger \right\rangle \\ &\quad - \left\langle \frac{\partial}{\partial p}\psi^\dagger, L\psi \right\rangle + \left\langle \frac{\partial}{\partial p}\psi^\dagger, S\psi \right\rangle + \left\langle \frac{\partial}{\partial p}\psi^\dagger, q \right\rangle \\ &\quad + \left\langle \psi, \frac{\partial}{\partial p}q^\dagger \right\rangle - \left\langle \psi^\dagger, \left(\frac{\partial}{\partial p}L\right)\psi \right\rangle + \left\langle \psi^\dagger, \left(\frac{\partial}{\partial p}S\right)\psi \right\rangle\Bigg] \\ &+ \left[- \left\langle I, L^\dagger\psi^\dagger \right\rangle + \left\langle I, S^\dagger\psi^\dagger \right\rangle + \left\langle I, q^\dagger \right\rangle\right] \frac{\partial \psi}{\partial p} \\ &+ \left[- \left\langle I, L\psi \right\rangle + \left\langle I, S\psi \right\rangle + \left\langle I, q \right\rangle\right] \frac{\partial \psi^\dagger}{\partial p}.\end{aligned} \tag{3.36}$$

Combining like terms into inner products and taking advantage of the fact that the inner product of a vector with the identity matrix is just the inner product of the vector, Eq. (3.21),

$$
\begin{aligned}
\frac{d\mathcal{L}}{dp} = \Bigg[& \left\langle \frac{\partial}{\partial p}\psi, -L^\dagger\psi^\dagger + S^\dagger\psi^\dagger + q^\dagger \right\rangle \\
& \left\langle \frac{\partial}{\partial p}\psi^\dagger, -L\psi + S\psi + q \right\rangle \\
& + \left\langle \psi, \frac{\partial}{\partial p}q^\dagger \right\rangle - \left\langle \psi^\dagger, \left(\frac{\partial}{\partial p}L\right)\psi \right\rangle + \left\langle \psi^\dagger, \left(\frac{\partial}{\partial p}S\right)\psi \right\rangle \Bigg] \\
& + \left[\langle -L^\dagger\psi^\dagger + S^\dagger\psi^\dagger + q^\dagger \rangle \right] \frac{\partial\psi}{\partial p} \\
& + \left[\langle -L\psi + S\psi + q \rangle \right] \frac{\partial\psi^\dagger}{\partial p}
\end{aligned} \tag{3.37}
$$

Notice that at this point all of the terms that include the impossible-to-calculate $\frac{\partial\psi}{\partial p}$ and $\frac{\partial\psi^\dagger}{\partial p}$ quantities have been isolated. If the forward equation is satisfied,

$$L\psi - S\psi - q = 0, \tag{3.38}$$

all of the terms that multiply $\frac{\partial\psi^\dagger}{\partial p}$ cancel to 0. Similarly, if

$$L^\dagger\psi^\dagger - S^\dagger\psi^\dagger - q^\dagger = 0, \tag{3.39}$$

all of the terms that multiply $\frac{\partial\psi}{\partial p}$ are reduced to 0. This is the source of the adjoint equation. If the forward and adjoint equations are satisfied, than the expression for the sensitivity coefficients contains only terms that are calculable,

$$\frac{d\mathcal{L}}{dp} = \left\langle \psi, \frac{\partial}{\partial p}q^\dagger \right\rangle - \left\langle \psi^\dagger, \left(\frac{\partial}{\partial p}L\right)\psi \right\rangle + \left\langle \psi^\dagger, \left(\frac{\partial}{\partial p}S\right)\psi \right\rangle. \tag{3.40}$$

In the following section the as-of-yet undefined $L^\dagger$ and $S^\dagger$ operators are defined.

Finally, recalling the original definition of the Lagrangian system, Eq. (3.26), the

Lagrangian is equal to the QoI as long as the forward equation is satisfied. The derivative of the Lagrangian with respect to some parameter p is therefore also the derivative of the QoI to that parameter,

$$\frac{dQ}{dp} = \left\langle \psi, \frac{\partial}{\partial p} q^\dagger \right\rangle - \left\langle \psi^\dagger, \left(\frac{\partial}{\partial p} L \right) \psi \right\rangle + \left\langle \psi^\dagger, \left(\frac{\partial}{\partial p} S \right) \psi \right\rangle. \tag{3.41}$$

This expression for the sensitivity coefficients $\frac{\partial Q}{\partial p}$ requires the evaluation of these three inner products for each parameter of interest p. For multigroup cross sections, this means that this expression must be evaluated for each energy group, each reaction of interest, and each material. The forward and adjoint equations must be solved only once for ψ and $\psi^\dagger$, respectively, which are then used in the inner products for any number of parameters p.

3.5.2 k-Eigenvalue Form of the Neutron Transport Equations

The derivation of the adjoint sensitivity equations may be expanded to be applicable to k-eigenvalue neutron transport problems. In these problems the neutron transport equation includes an extra source term corresponding to the production of neutrons by fission, and the external source of neutrons is often neglected,

$$\begin{aligned} \vec{\Omega} \cdot \bigtriangledown \psi(\vec{r}, E, \vec{\Omega}) + & \Sigma_t(\vec{r}, E) \psi(\vec{r}, E, \vec{\Omega}) = \\ & \int_{4\pi} \int_0^\infty \Sigma_s(\vec{r}, E' \to E, \vec{\Omega}' \to \vec{\Omega}) \psi(\vec{r}, E', \vec{\Omega}') dE' d\vec{\Omega}' + \\ & \lambda \frac{\chi(\vec{r}, E)}{4\pi} \int_0^\infty \nu(E') \Sigma_f(\vec{r}, E') \int_{4\pi} \psi(\vec{r}, E', \vec{\Omega}') d\Omega' dE'. \end{aligned} \tag{3.42}$$

This equation introduces several terms that have not yet been seen. The fission cross section Σ_f and the number of neutrons produced per fission ν are functions of material and energy. The fission spectrum χ describes the distribution in energy of neutrons produced by fission. Finally, the eigenvalue λ is defined as one over the system's multiplication factor,

$$\lambda = \frac{1}{k_{eff}}. \tag{3.43}$$

The fission terms can be collected into the operator F such that the operator form of the neutron transport equation can be written

$$L\psi = S\psi + \lambda F\psi. \tag{3.44}$$

The Lagrangian system is defined in the same fashion as in the source-driven neutron transport case, but now must include a dependence upon the eigenvalue λ,

$$\mathcal{L} = \mathcal{L}(\psi, \psi^\dagger, \lambda, p), \tag{3.45}$$

and the derivative of the Lagrangian system $\mathcal{L}$ with respect to the parameters of interest p must be expanded to include the k-eigenvalue λ,

$$\frac{d\mathcal{L}}{dp} = \frac{\partial\mathcal{L}}{\partial p} + \frac{\partial\mathcal{L}}{\partial\psi}\frac{\partial\psi}{\partial p} + \frac{\partial\mathcal{L}}{\partial\psi^\dagger}\frac{\partial\psi^\dagger}{\partial p} + \frac{\partial\mathcal{L}}{\partial\lambda}\frac{\partial\lambda}{\partial p}. \tag{3.46}$$

For brevity the algebra that follows the application of Eq. (3.46) to the Lagrangian system is not repeated here. This derivation eventually yields equations for the forward and adjoint equations,

$$L\psi - S\psi - \lambda F\psi = 0, \tag{3.47}$$

$$L^\dagger\psi^\dagger - S^\dagger\psi^\dagger - \lambda F^\dagger\psi^\dagger = 0, \tag{3.48}$$

as well as one additional constraint,

$$\left\langle \psi^{\dagger}, F\psi \right\rangle = 1. \tag{3.49}$$

With these constraints the equation for the sensitivity coefficients matches the equation derived for the source-driven problem, Eq. 3.41. The adjoint sensitivity method is therefore applicable to both source-driven and k-eigenvalue problems and was implemented in PDT for both of these cases [28].

3.6 The Adjoint Operators

In the previous section an equation for the sensitivity of the QoI to parameters p was derived which required the adjoint transport equation, Eq. (3.39), be satisfied. The adjoint analog to the streaming and interaction matrix, $L^{\dagger}$, and the adjoint analog to the scattering operator $S^{\dagger}$ were left undefined. To derive these we examine the relationship between the adjoint and forward operators, Eq. (3.20), repeated here for convenience,

$$\langle Ma, b \rangle = \left\langle a, M^{\dagger} b \right\rangle .$$

We will deal with the operators in their continuous forms in space and angle and in the multigroup form in energy, as in the definition of the inner product in Eq. (3.18). The streaming and removal operator L is addressed as though it were two operators-one containing the streaming term and the other containing the interaction term. These derivations may be found in [6] and in [21].

3.6.1 The Streaming Component of L

First we consider the streaming component of the streaming and interaction operator L, denoted L_s to distinguish it from the full operator. By Eq. (3.20),

$$\left\langle L_s^\dagger \psi^\dagger, \psi \right\rangle = \left\langle \psi^\dagger, L_s \psi \right\rangle . \tag{3.50}$$

Expanding the inner product notation using Eq. (3.18) on the right hand side of this equation and replacing L_s with the continuous form of the streaming operator yields

$$\left\langle L_s^\dagger \psi^\dagger, \psi \right\rangle = \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \sum_g \psi_g^\dagger(\vec{r}, \vec{\Omega}) \vec{\Omega} \cdot \triangledown \psi_g(\vec{r}, \vec{\Omega}). \tag{3.51}$$

The volume integral can be expanded by integrating by parts,

$$\begin{aligned} \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \sum_g \psi_g^\dagger(\vec{r}, \vec{\Omega}) \vec{\Omega} \cdot \triangledown \psi_g(\vec{r}, \vec{\Omega}) = \\ \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \sum_g \triangledown \cdot \Big(\psi_g^\dagger(\vec{r}, \vec{\Omega}) \vec{\Omega} \psi_g(\vec{r}, \vec{\Omega}) \Big) \\ - \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \sum_g \psi_g(\vec{r}, \vec{\Omega}) \vec{\Omega} \cdot \triangledown \psi_g^\dagger(\vec{r}, \vec{\Omega}). \end{aligned} \tag{3.52}$$

The divergence theorem can be applied to the second term of Eq. (3.52),

$$\int_{\mathcal{D}} dV \, \triangledown \cdot \Big(\psi_g^\dagger(\vec{r}, \vec{\Omega}) \vec{\Omega} \psi_g(\vec{r}, \vec{\Omega}) \Big) = \oint_{\partial \mathcal{D}} dA \psi_g^\dagger(\vec{r}, \vec{\Omega}) \psi_g(\vec{r}, \vec{\Omega}) \left(\Omega \cdot \hat{n} \right), \tag{3.53}$$

where $\partial\mathcal{D}$ is the boundary of domain $\mathcal{D}$, $\hat{n}$ is the unit normal vector across the boundary $\partial\mathcal{D}$, and ∂A an infinitesimal area on $\partial\mathcal{D}$.

Substituting the result from Eq. (3.53) into Eq. (3.52),

$$\begin{aligned}\int_{4\pi} d\Omega \int_{\mathcal{D}} dV \sum_g \psi_g^\dagger(\vec{r},\vec{\Omega})\vec{\Omega}\cdot\bigtriangledown\psi_g(\vec{r},\vec{\Omega}) = \\ \int_{4\pi} d\Omega \oint_{\partial\mathcal{D}} dA \sum_g \psi_g^\dagger(\vec{r},\vec{\Omega})\psi_g(\vec{r},\vec{\Omega})\,(\Omega\cdot\hat{n}) \\ -\int_{4\pi} d\Omega \int_{\mathcal{D}} dV \sum_g \psi_g(\vec{r},\vec{\Omega})\vec{\Omega}\cdot\bigtriangledown\psi_g^\dagger(\vec{r},\vec{\Omega}).\end{aligned} \tag{3.54}$$

The surface term in Eq. (3.54) can be expanded by breaking it into exiting directions $\vec{\Omega}\cdot\hat{n} > 0$ and entering directions $\vec{\Omega}\cdot\hat{n} < 0$

$$\begin{aligned}\int_{4\pi} d\Omega \oint_{\partial\mathcal{D}} dA \sum_g \psi_g^\dagger(\vec{r},\vec{\Omega})\psi_g(\vec{r},\vec{\Omega})\,(\Omega\cdot\hat{n}) = \\ \int_{\vec{\Omega}\cdot\hat{n}<0} d\Omega \oint_{\partial\mathcal{D}} dA \sum_g \psi_g^\dagger(\vec{r},\vec{\Omega})\psi_g(\vec{r},\vec{\Omega})\,(\Omega\cdot\hat{n}) \\ +\int_{\vec{\Omega}\cdot\hat{n}>0} d\Omega \oint_{\partial\mathcal{D}} dA \sum_g \psi_g^\dagger(\vec{r},\vec{\Omega})\psi_g(\vec{r},\vec{\Omega})\,(\Omega\cdot\hat{n})\,.\end{aligned} \tag{3.55}$$

For a source-detector problem of the types examined in this work, the angular flux in the incident directions on the problem boundary is zero,

$$\psi_g(\vec{r}\in\partial\mathcal{D},\vec{\Omega}\cdot\hat{n}<0) = 0. \tag{3.56}$$

This eliminates the second term in Eq. (3.55). If we enforce a zero-exiting boundary condition upon the adjoint angular flux

$$\psi_g^\dagger(\vec{r}\in\partial\mathcal{D},\vec{\Omega}\cdot\hat{n}>0) = 0, \tag{3.57}$$

the third term in Eq. (3.55) is similarly eliminated.

With these boundary conditions, Eq. (3.54) is simplified to

$$\int_{4\pi} d\Omega \int_{\mathcal{D}} dV \sum_g \psi_g^\dagger(\vec{r},\vec{\Omega})\vec{\Omega}\cdot\bigtriangledown\psi_g(\vec{r},\vec{\Omega}) = \\ -\int_{4\pi} d\Omega \int_{\mathcal{D}} dV \sum_g \psi_g(\vec{r},\vec{\Omega})\vec{\Omega}\cdot\bigtriangledown\psi_g^\dagger(\vec{r},\vec{\Omega}). \tag{3.58}$$

Substituting this result all the way back to the original expansion of the streaming term inner product, Eq. (3.51), moving the negative sign in the second term into the operator, and reordering the two angular flux terms,

$$\left\langle L_s^\dagger\psi^\dagger,\psi\right\rangle = \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \sum_g -\vec{\Omega}\cdot\bigtriangledown\psi_g^\dagger(\vec{r},\vec{\Omega})\psi_g(\vec{r},\vec{\Omega}). \tag{3.59}$$

It is clear that the adjoint streaming term is the forward streaming term multiplied by negative one. This adjoint streaming operator is very easily treated within PDT and does not require modification to the transport solver beyond the boundary condition requirements set forth in Eq . (3.56) and Eq. (3.57).

3.6.2 The Interaction Component of L

Next we will examine the interaction component of the L operator, denoted L_i. As before, by the operator property of the inner product,

$$\left\langle L_i^\dagger\psi^\dagger,\psi\right\rangle = \left\langle\psi^\dagger, L_i\psi\right\rangle. \tag{3.60}$$

Expanding the right hand side of this expression gives us

$$\left\langle L_i^\dagger\psi^\dagger,\psi\right\rangle = \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \sum_g \psi_g^\dagger(\vec{r},\vec{\Omega})\Sigma_{t,g}(\vec{r})\psi_g(\vec{r},\vec{\Omega}). \tag{3.61}$$

The three terms within the integrals on the right hand side of this equation can be reordered at will, so

$$\left\langle L_i^\dagger \psi^\dagger, \psi \right\rangle = \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \sum_g \psi_g(\vec{r}, \vec{\Omega}) \Sigma_{t,g}(\vec{r}) \psi_g^\dagger(\vec{r}, \vec{\Omega}). \tag{3.62}$$

It is clear that the forward and adjoint interaction operators are equivalent,

$$\left\langle L_i^\dagger \psi^\dagger, \psi \right\rangle = \left\langle L_i \psi^\dagger, L_i \psi \right\rangle = \left\langle \psi^\dagger, L_i \psi \right\rangle. \tag{3.63}$$

Note that the adjoint interaction operator, being equal to the forward interaction operator, uses the physical cross sections employed in the forward problem.

3.6.3 The Scattering Operator S

With the streaming and interaction adjoint operators defined, we turn to the adjoint form of the scattering operator S. We denote the transfer scattering cross section from group g' to group g and angle $\vec{\Omega}'$ to $\vec{\Omega}$ as $\Sigma_{s,g'\to g}(\vec{r}, \vec{\Omega}' \to \vec{\Omega})$. Writing the inner product relation

$$\left\langle S^\dagger \psi^\dagger, \psi \right\rangle = \left\langle \psi^\dagger, S\psi \right\rangle \tag{3.64}$$

and expanding the right hand side into the continuous form,

$$\left\langle S^\dagger \psi^\dagger, \psi \right\rangle = \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \sum_g \psi_g^\dagger(\vec{r}, \vec{\Omega}) \int_{4\pi} d\Omega' \sum_{g'} \Sigma_{s,g'\to g}(\vec{r}, \vec{\Omega}' \to \vec{\Omega}) \psi_{g'}(\vec{r}, \vec{\Omega}'). \tag{3.65}$$

Recalling from the discussion of the anisotropic scattering operator that the scattering cross section depends upon ξ_s, the dot product of $\vec{\Omega}$ and $\vec{\Omega}'$, this can be slightly simplified as

$$\left\langle S^\dagger \psi^\dagger, \psi \right\rangle = \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \sum_g \psi_g^\dagger(\vec{r}, \vec{\Omega}) \int_{4\pi} d\Omega' \sum_{g'} \Sigma_{s,g' \to g}(\vec{r}, \vec{\Omega}' \cdot \vec{\Omega}) \psi_{g'}(\vec{r}, \vec{\Omega}'). \quad (3.66)$$

Collecting all of the integrals, we have

$$\left\langle S^\dagger \psi^\dagger, \psi \right\rangle = \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \int_{4\pi} d\Omega' \sum_g \sum_{g'} \psi_g^\dagger(\vec{r}, \vec{\Omega}) \Sigma_{s,g' \to g}(\vec{r}, \vec{\Omega}' \cdot \vec{\Omega}) \psi_{g'}(\vec{r}, \vec{\Omega}'). \quad (3.67)$$

Rearranging the order of ψ and $\psi^\dagger$ within the integrals does not change the value of the expression, so

$$\left\langle S^\dagger \psi^\dagger, \psi \right\rangle = \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \int_{4\pi} d\Omega' \sum_g \sum_{g'} \psi_{g'}(\vec{r}, \vec{\Omega}') \Sigma_{s,g' \to g}(\vec{r}, \vec{\Omega}' \cdot \vec{\Omega}) \psi_g^\dagger(\vec{r}, \vec{\Omega}). \quad (3.68)$$

Suppose that, without changing the above expression in any meaningful manner, we introduce a new variable f and use it to replace g everywhere it appears. Similarly, let us introduce a new variable h and use it to replace g':

$$\left\langle S^\dagger \psi^\dagger, \psi \right\rangle = \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \int_{4\pi} d\Omega' \sum_f \sum_h \psi_h(\vec{r}, \vec{\Omega}') \Sigma_{s,h \to f}(\vec{r}, \vec{\Omega}' \cdot \vec{\Omega}) \psi_f^\dagger(\vec{r}, \vec{\Omega}). \quad (3.69)$$

This expression is exactly equal to Eq. (3.68). Now let us again rename variables, this time replacing f with g' and h with g:

$$\left\langle S^\dagger \psi^\dagger, \psi \right\rangle = \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \int_{4\pi} d\Omega' \sum_{g'} \sum_g \psi_g(\vec{r}, \vec{\Omega}') \Sigma_{s,g \to g'}(\vec{r}, \vec{\Omega}' \cdot \vec{\Omega}) \psi_{g'}^\dagger(\vec{r}, \vec{\Omega}). \quad (3.70)$$

Comparing Eq. (3.70) and Eq. (3.68), we have simply reversed the notation for g and g' without otherwise changing the expression.

We apply the same logic to $\vec{\Omega}$ and $\vec{\Omega}'$, noting that the dot product $\vec{\Omega}' \cdot \vec{\Omega}$ is invariant with respect to the order of Ω and Ω'.

$$\left\langle S^\dagger \psi^\dagger, \psi \right\rangle = \int_{4\pi} d\Omega' \int_{\mathcal{D}} dV \int_{4\pi} d\Omega \sum_{g'} \sum_{g} \psi_g(\vec{r}, \vec{\Omega}) \Sigma_{s,g\to g'}(\vec{r}, \vec{\Omega}' \cdot \vec{\Omega}) \psi^\dagger_{g'}(\vec{r}, \vec{\Omega}'). \quad (3.71)$$

Finally, the order of the integrations can be changed,

$$\left\langle S^\dagger \psi^\dagger, \psi \right\rangle = \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \sum_{g} \int_{4\pi} d\Omega' \sum_{g'} \psi_g(\vec{r}, \vec{\Omega}) \Sigma_{s,g\to g'}(\vec{r}, \vec{\Omega}' \cdot \vec{\Omega}) \psi^\dagger_{g'}(\vec{r}, \vec{\Omega}'), \quad (3.72)$$

and $\psi_g(\vec{r}, \vec{\Omega})$ removed from the integral over $\vec{\Omega}'$ and g',

$$\left\langle S^\dagger \psi^\dagger, \psi \right\rangle = \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \sum_{g} \psi_g(\vec{r}, \vec{\Omega}) \int_{4\pi} d\Omega' \sum_{g'} \Sigma_{s,g\to g'}(\vec{r}, \vec{\Omega}' \cdot \vec{\Omega}) \psi^\dagger_{g'}(\vec{r}, \vec{\Omega}'). \quad (3.73)$$

This reveals the definition of the adjoint scattering operator,

$$S^\dagger \psi^\dagger = \int_{4\pi} d\Omega' \sum_{g'} \Sigma_{s,g\to g'}(\vec{r}, \vec{\Omega}' \cdot \vec{\Omega}) \psi^\dagger_{g'}(\vec{r}, \vec{\Omega}'). \quad (3.74)$$

The adjoint scattering operator uses the same transfer scattering cross sections as the forward scattering operator, but note that $\psi^\dagger_{g'}$ multiplies the transfer scattering cross section to group g' instead of from group g'.

3.6.4 The Fission Operator F

Finally, we address the adjoint form of the fission operator. As before, we begin with the relationship between the forward and adjoint operators,

$$\left\langle F^\dagger \psi^\dagger, \psi \right\rangle = \left\langle \psi^\dagger, F\psi \right\rangle, \quad (3.75)$$

expand the right hand side into the continuous form,

$$\left\langle F^\dagger\psi^\dagger, \psi\right\rangle = \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \sum_{g} \psi_g^\dagger(\vec{r}, \vec{\Omega})\chi_g \int_{4\pi} d\Omega' \sum_{g'} \nu_{g'}(\vec{r})\Sigma_{f,g'}(\vec{r})\psi_{g'}(\vec{r}, \vec{\Omega}'). \quad (3.76)$$

As in the derivation of the scattering operator, we can reverse the notation for g and g' and for $\vec{\Omega}$ and $\vec{\Omega}'$,

$$\left\langle F^\dagger\psi^\dagger, \psi\right\rangle = \int_{4\pi} d\Omega' \int_{\mathcal{D}} dV \sum_{g'} \psi_{g'}^\dagger(\vec{r}, \vec{\Omega}')\chi_{g'} \int_{4\pi} d\Omega \sum_{g} \nu_{g}(\vec{r})\Sigma_{f,g}(\vec{r})\psi_{g}(\vec{r}, \vec{\Omega}), \quad (3.77)$$

gather all of the integrals,

$$\left\langle F^\dagger\psi^\dagger, \psi\right\rangle = \int_{4\pi} d\Omega' \int_{\mathcal{D}} dV \sum_{g'} \int_{4\pi} d\Omega \sum_{g} \psi_{g'}^\dagger(\vec{r}, \vec{\Omega}')\chi_{g'}\nu_{g}(\vec{r})\Sigma_{f,g}(\vec{r})\psi_{g}(\vec{r}, \vec{\Omega}), \quad (3.78)$$

reorder the integrals,

$$\left\langle F^\dagger\psi^\dagger, \psi\right\rangle = \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \sum_{g} \int_{4\pi} d\Omega' \sum_{g'} \psi_{g'}^\dagger(\vec{r}, \vec{\Omega}')\chi_{g'}\nu_{g}(\vec{r})\Sigma_{f,g}(\vec{r})\psi_{g}(\vec{r}, \vec{\Omega}), \quad (3.79)$$

and then organize terms within the expression,

$$\left\langle F^\dagger\psi^\dagger, \psi\right\rangle = \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \sum_{g} \nu_{g}(\vec{r})\Sigma_{f,g}(\vec{r})\psi_{g}(\vec{r}, \vec{\Omega}) \int_{4\pi} d\Omega' \sum_{g'} \chi_{g'}\psi_{g'}^\dagger(\vec{r}, \vec{\Omega}'). \quad (3.80)$$

This defines the adjoint fission operator, $F^\dagger$,

$$F^\dagger\psi^\dagger = \nu_{g}(\vec{r})\Sigma_{f,g}(\vec{r}) \int_{4\pi} d\Omega' \sum_{g'} \chi_{g'}\psi_{g'}^\dagger(\vec{r}, \vec{\Omega}'). \quad (3.81)$$

The adjoint fission operator uses the same cross sections as the forward fission operator, but as with the scattering operator, the transfer matrix is transposed.

3.6.5 Solving the Adjoint Equations with PDT

Given the adjoint source defined in Section 3.4, the adjoint equations defined in Section 3.5, the boundary conditions defined in Section 3.6.1, and the adjoint operators defined in Sections 3.6.1, 3.6.2, and 3.6.3, the adjoint equations can be solved with PDT. The adjoint problem is defined in the PDT input deck alongside the forward problem; for an example of these input decks, see Appendix B. A flag is set within PDT at runtime to signify that an adjoint problem is being solved; this ensures the construction the adjoint operators where appropriate.

The source term $q^\dagger$ is usually the total cross section in the materials and volume of interest to the forward problem. The zero-exiting flux condition is set as a Dirichlet boundary condition in the outgoing directions. The interaction component of the streaming and interaction matrix L was demonstrated to be identical to the forward problem in section 3.6.2.

The adjoint scattering operator was shown to use the same cross-section data as the forward operator, but instead of multiplying $\psi_g^\dagger$ by scattering cross sections from group g to groups g', the adjoint angular flux is multiplied by cross sections to group g from groups g'. This is easily accomplished during the construction of the adjoint scattering operator.

The adjoint streaming operator differs from the forward streaming operator only in sign. Because the angular quadrature sets chosen for radiation transport problems are symmetric, the adjoint streaming operator is actually very simple to implement. The forward streaming operator is used as-is when solving an adjoint problem, but when the angular flux in direction d is requested, PDT returns the angular flux in the direction opposite d. The scalar flux is not changed.

With this outline of the adjoint solver in PDT, it is clear that, in general, the adjoint problem is as complex as the forward problem and will require a similar amount of time to solve.

3.7 Alternative Methods to Calculate Sensitivity Coefficients

The adjoint sensitivity method outlined in the previous section is an extremely efficient approach to calculate sensitivity coefficients for a problem containing a large number of parameters of interest and a small number of quantities of interest. For each QoI two solutions of the radiation transport problem are required: one forward solve and one adjoint solve. However, for problems with a large number of QoIs and few parameters of interest, the finite difference method may be more suitable for sensitivity analysis.

The finite difference method for calculating sensitivity coefficients requires one forward solve at the unperturbed parameters and one forward solve for each parameter of interest. Let us write the QoI as a function of the solution ψ, which is a function of the input parameters $p_1, p_2, p_3, ..., p_N$, $Q(\psi(p_1, ..., p_N))$. The sensitivity of the QoI to a particular parameter is evaluated

$$\frac{\partial Q}{\partial p_1} \approx \frac{Q(\psi(p_1 + \epsilon, p_2, ..., p_N)) - Q(\psi(p_1, p_2, ..., p_N))}{\epsilon}. \tag{3.82}$$

This approach allows the calculation of the sensitivity of any number of QoIs with only one forward solve per parameter of interest at a perturbation ϵ.

The finite difference method is ideally suited for problems with a relatively large number of QoIs and a relatively small number of parameters of interest. The adjoint sensitivity method is ideally suited for problems with a relatively small number of QoIs and a large number of parameters of interest. The problems considered in this work fall into the latter category.

3.8 Resonance Self-Shielding and Sensitivity Calculations

The calculation of the multigroup cross sections from the nuclear data (pointwise cross sections and resonance parameters) contained in the ENDF/B data files can be very sensitive to the choice of the weighting spectrum, as discussed in Section 2.3. This effect is especially important in problems that exhibit significant resonance self-shielding, complicating the calculation of sensitivity coefficients for these problems. These complications are discussed in detail in this section.

The experiments being performed at Texas A&M University in support of the CERT project, discussed in Section 5.2.1, primarily contain materials with low atomic number. Because these materials do not contain strong resonances the complications discussed in this section are minimal for problems related to the CERT project. Problems involving nuclear fuel, as discussed in Section 5.3, must take these complications into consideration.

3.8.1 Implicit Sensitivity Coefficients

The adjoint sensitivity method outlined in this section calculates the direct effect of perturbations to the multigroup cross sections on the QoI. This direct effect is referred to in the literature as the “explicit” component of the sensitivity coefficient [26]. This explicit sensitivity coefficient neglects a second-order effect on the QoI due to perturbations to the multigroup cross sections present in problems containing strong resonances. These problems include an "implicit" component of the sensitivity coefficient that must be calculated in a different manner.

The explicit sensitivity coefficient does not account for the fact that changes to one multigroup cross section may cause changes in the other multigroup cross sections due to self-shielding effects [33]. The implicit component of the sensitivity coefficient for a particular multigroup cross section σ_g accounts for the sensitivity of the QoI to all other multigroup cross sections $\sigma_{g'}$ and the sensitivity of those cross sections to perturbations in σ_g due to self-shielding effects [33]. The total sensitivity coefficient is the sum of the explicit and implicit components of the sensitivity, respectively,

$$\frac{\partial Q}{\partial \sigma_g}^{(total)} = \frac{\partial Q}{\partial \sigma_g} + \sum_{g'} \frac{\partial Q}{\partial \sigma_{g'}} \frac{\partial \sigma_{g'}}{\partial \sigma_g}. \tag{3.83}$$

The explicit effect $\frac{\partial Q}{\partial \sigma_g}$ and the sensitivity of the QoI to the multigroup cross sections in the implicit component of the total sensitivity $\frac{\partial Q}{\partial \sigma_{g'}}$ may be calculated with an adjoint sensitivity method like the one implemented in PDT.

The terms $\frac{\partial \sigma_{g'}}{\partial \sigma_g}$ in Eq. (3.83) may be calculated by the application of the finite difference method to the infinite-medium slowing down problem (IMSDP) used to generate the weighting spectrum. Recall that the IMSDP is solved using either pointwise cross section data or very fine multigroup cross sections. The perturbation to the multigroup cross sections $\sigma_{g'}$ may be calculated by uniformly perturbing the IMSDP cross sections corresponding to the energy range of the multigroup σ_g. Perturbing the IMSDP cross sections will perturb the solution of the IMSDP, which is the weighting spectrum used to generate the multigroup cross sections. This leads to the perturbation of the multigroup cross sections $\sigma_{g'}$.

Although codes like Oak Ridge National Laboratory's TSUNAMI module for SCALE computes these implicit sensitivity coefficients [26][33], this capability has not been implemented at Texas A&M University. The lack of implicit sensitivities may cause significant error in the calculation of sensitivity coefficients for problems containing strong resonances such as reactor physics calculations containing nuclear fuel. The implicit sensitivity coefficients are negligible for problems that do not contain strong resonances.

Given a group structure with very fine resolution of the resonances, a constant weighting spectrum will yield accurate multigroup cross sections [21][22]. In this limit the implicit sensitivity coefficients are reduced to zero as the weighting spectrum does not change with perturbations to the cross sections.

3.8.2 Realizations of the Nuclear Data

A second factor related to self-shielding complicates the calculation of even the explicit sensitivity coefficients. The ENDF files contain both the mean value of the nuclear data and the variances and covariances of that data. The weighting spectrum used to calculate multigroup cross sections is typically generated by solving the IMSDP using the mean values of the nuclear data. Realizations of the multigroup cross sections should be generated by sampling the uncertain nuclear-data space, solving the IMSDP at these realizations, and then computing the multigroup cross sections using the perturbed nuclear data and perturbed weighting spectrum. This approach yields realizations of the multigroup cross sections using a weighting spectrum consistent with the perturbed ENDF data.

Instead, the IMSDP is solved only once at the mean value of the nuclear data and this weighting spectrum is used to calculate the mean value of multigroup cross sections and the multigroup covariance matrices. Realizations of the multigroup cross sections are drawn directly from the multivariate Gaussian distribution defined by this data. These realizations are no longer consistent with the weighting spectrum used to generate the multigroup cross sections.

This limitation does not effect the estimation of variance with the linear propagation of uncertainty equation as it uses the sensitivity coefficients calculated at the mean value of the cross sections and the covariance matrix calculated at the mean value of the cross sections. This complication does impact Monte Carlo and response surface methods for UQ which rely upon sampling the cross-section space and solving the radiation transport equation at each of these samples.

The variance and covariance data contained within the ENDF file is not easily used to directly sample the nuclear data space. The nuclear data space could be approximated with a set of ultra-fine multigroup cross sections and their associated covariance matrices. This ultra-fine multigroup data may be generated with NJOY and a constant weighting spectrum. Samples can be drawn from the ultra-fine multivariate Gaussian distribution and weighting spectrums calculated by solving the IMSDP at each realization of the ultra-fine cross-section space. Multigroup cross sections can then be generated for the desired group structure using the realizations of the ultra-fine cross-section space and their associated weighting spectrums. In this fashion realizations of the multigroup cross sections may be generated using a weighting spectrum consistent with the realized nuclear data. This approach remains a topic of ongoing research and was not implemented in this work.

Note that even if consistently-weighted multigroup cross sections are generated and used in radiation transport calculations, the sensitivity coefficients still contain both explicit and implicit components. The use of an ultra-fine group structure in the problem of interest would eliminate both the necessity of computing a consistent weighting spectrum and the need to compute the implicit sensitivity coefficients. However, the computational costs associated with solving radiation transport problems with an ultra-fine group structure makes this approach impractical for all but extremely simple problems.

3.9 Notes on the Calculation of Sensitivity Coefficients with PDT

The adjoint sensitivity method was implemented in PDT by Hayes Stripling [28] as part of his doctoral work. The problems analyzed in this work utilized these PDT capabilities, but the differing motivations and goals of Stripling's' original project and this project led to several implementation differences in the use of the adjoint sensitivity method in PDT. These differences are outlined in this section.

3.9.1 Sensitivity Coefficients for Removal Cross Sections

The adjoint sensitivity method implemented in PDT calculates the sensitivity coefficients for the total cross section and the transfer scattering cross sections according to Eq. (3.41). If the quantity of interest is assumed to be the total reaction rate in a volume, the total cross section appears in the streaming and collision operator L and the adjoint source term $q^{\dagger}$. The transfer scattering cross sections appear in the scattering operator S.

As discussed in the Section 2.3 covering multigroup cross section generation, we calculate cross sections and covariance data for individual reactions and not the total cross section. PDT, however, calculates sensitivities only for total, scattering, and fission cross sections. In this section the relationship between the sensitivities calculated by PDT and the sensitivities needed in this work are outlined.

The total cross section for a given material m and group g, $\sigma_{t,g,m}$, is the sum of the zeroth moment scattering cross sections and each other cross section,

$$\sigma_{t,g,m} = \sum_{g'} \sigma_{s,g\to g',\ell=0,m} + \sigma_{(n,\gamma),g,m} + \sigma_{(n,\alpha),g,m} + \sigma_{(n,p),g,m} + ... \tag{3.84}$$

Interactions that result in the loss of a neutron, like the (n,γ), (n,α), (n,p), etc, reactions, appear only in the removal cross-section term. It is straightforward to demonstrate that, for a given material and energy group, the sensitivity of the QoI to any removal cross section is equal to the sensitivity of the QoI to the total cross section.

The derivative of the L operator with respect to the total cross section $\sigma_{t,g,m}$ is performed one matrix entry at a time. Consider a particular entry in row i and column j of the L matrix, $L_{i,j}$. This entry is either empty, in which case its derivative is also zero, or it contains some coefficients $C_{i,j}$ and the total cross section, $\sigma_{t,g,m}$, for each material in the cell corresponding to that entry in the matrix,

$$L_{i,j} = C_{i,j} \sum_m N_m \sigma_{t,g,m}. \tag{3.85}$$

Recall that the microscopic cross sections are multiplied by their number densities N when entered in the transport operators.

The derivative of the terms in $L_{i,j}$ with respect to a particular total cross section are straightforward to evaluate. Let us consider the total cross section in material m^*

and group g^*, σ_{t,g^*,m^*}:

$$\frac{d}{d\sigma_{t,g^*,m^*}} L_{i,j} = \begin{cases} 0 & \text{if} \quad \sigma_{t,g^*,m^*} \text{ does not appear in } L_{i,j} \\ C_{i,j} N_{m^*} & \text{if} \quad \sigma_{t,g^*,m^*} \text{ does appear in } L_{i,j} \end{cases} . \tag{3.86}$$

We next consider the case when $\sigma_{t,g,m}$ is replaced by the sum of the possible neutron interactions, as in Eq. (3.84). The matrix element $L_{i,j}$ now appears

$$L_{i,j} = C_{i,j} \sum_m N_m \left(\sum_{g'} \sigma_{s,g\to g',0,m} + \sigma_{(n,\gamma),g,m} + \sigma_{(n,\alpha),g,m} + \sigma_{(n,p),g,m} + ... \right). \tag{3.87}$$

Let us consider the (n,γ) reaction. The derivative of $L_{i,j}$ with respect to $\sigma_{(n,\gamma),g,m}$ for a particular group and material is

$$\frac{d}{d\sigma_{(n,\gamma),g^*,m^*}} L_{i,j} = \begin{cases} 0 & \text{if} \quad \sigma_{(n,\gamma),g^*,m^*} \notin L_{i,j} \\ C_{i,j} N_{m^*} & \text{if} \quad \sigma_{(n,\gamma),g^*,m^*} \in L_{i,j} \end{cases} . \tag{3.88}$$

Comparing Eq. (3.86) and Eq. (3.88), we see that

$$\frac{d}{d\sigma_{(n,\gamma),g^*,m^*}} L_{i,j} = \frac{d}{d\sigma_{t,g^*,m^*}} L_{i,j}. \tag{3.89}$$

For QoIs that are a function of the total interaction rate in a volume, a similar argument is applied to the $q^\dagger$ term of the equation for the sensitivity coefficients, Eq. (3.41), with the same result for the i-th entry in $q^\dagger$, $q_i^\dagger$,

$$\frac{d}{d\sigma_{(n,\gamma),g^*,m^*}} q_i^\dagger = \frac{d}{d\sigma_{t,g^*,m^*}} q_i^\dagger. \tag{3.90}$$

This result holds for all reactions that appear only in the transfer and removal operator L or the adjoint source term $q^\dagger$ (such as the (n,γ), (n,α), (n,p), (n,t), and similar reactions). Therefore, the sensitivity coefficient for any reaction that results in the net loss of a neutron is equal to the sensitivity coefficient for the total cross section. Reactions that appear in the transfer operator S are discussed in the following section.

3.9.2 Sensitivity Coefficients for Transfer Cross Sections

The adjoint sensitivity method implemented in PDT returns sensitivity coefficients for the total cross sections $\sigma_{t,g,m}$ and the zeroth moment of the transfer scattering cross sections $\sigma_{s,g\to g',\ell=0,m}$. The sensitivity coefficients returned by PDT for the scattering cross sections are computed by taking derivatives of the scattering operator S. If the total scattering cross section is expanded into its constituent components, however, the transfer scattering cross sections appear in the scattering operator S, the streaming and interaction operator L, and in the adjoint source term $q^\dagger$. If the total cross section is replaced in each operator with its constituent parts, then the partial derivatives of the L and $q^\dagger$ operators with respect to the transfer scattering cross sections are not zero. In fact, following the derivation in the previous section,

$$\frac{d}{d\sigma_{s,g^*\to g',0,m^*}} L_{i,j} = \frac{d}{d\sigma_{t,g^*,m^*}} L_{i,j} \tag{3.91}$$

and

$$\frac{d}{d\sigma_{s,g^*\to g',0,m^*}} q_i^\dagger = \frac{d}{d\sigma_{t,g^*,m^*}} q_i^\dagger . \tag{3.92}$$

Because the PDT implementation of the adjoint sensitivity method does not expand the total cross section into its constituent components it assumes that the partial derivatives of the L and $q^\dagger$ operators with respect to the transfer scattering cross sections are zero. Therefore, the sensitivity coefficient for the transfer scattering cross sections $\sigma_{s,g\to g',m}$ consistent with the expansion of the total cross section into its constituent components are the sum of the total and transfer scattering cross sections for that material and group,

$$\frac{dQ}{d\sigma_{s,g^*\to g',0,m^*}} = \left(\frac{dQ}{d\sigma_{t,g^*,m^*}}\right)_{PDT} + \left(\frac{dQ}{d\sigma_{s,g^*\to g',0,m^*}}\right)_{PDT}. \tag{3.93}$$

Here we have denoted the PDT-calculated sensitivity coefficient for the transfer scattering cross section with a “PDT” subscript for clarify. At all points in this work besides this section (3.9.2), the sensitivity coefficient for the scattering cross section will be the consistent coefficient as defined in Eq. (3.93).

This result holds for all reactions that result in a neutron as a reaction product, including elastic and inelastic scattering, $(n, 2n)$ reactions, $(n, n + \alpha)$ reactions, and other reactions of a similar type.

The higher-order scattering cross sections do not appear in the total cross sections, so the sensitivity coefficients produced by PDT are correct and do not need to be modified.

3.9.3 Calculating Sensitivity Coefficients for the Removal Scattering Cross Sections from the Transfer Scattering Cross Sections

In this section we distinguish between transfer scattering cross sections, $\sigma_{s,g\to g',\ell,m}$ and removal scattering cross sections, $\sigma_{s,g,\ell,m}$. The “removal” designation for $\sigma_{s,g,\ell,m}$ denotes that this cross section accounts for the total removal of neutrons from group

g (although some of these neutrons scatter back into that group),

$$\sigma_{s,g} = \sum_{g'} \sigma_{s,g\to g'}. \tag{3.94}$$

To easily distinguish between these two cross sections, especially when in appearing in a numerator or denominator, let us denote the transfer scattering cross section $\sigma_{s,g\to g',\ell,m}$ as $\mathscr{S}(g \to g', \ell, m)$ and the removal scattering cross section $\sigma_{s,g,\ell,m}$ as $\mathcal{S}(g, \ell, m)$.

For brevity we drop the material subscript m from this section, as it will appear in every cross-section term in this section without change.

Sensitivity coefficients are calculated by PDT for the transfer scattering cross sections $\mathscr{S}(g \to g', \ell)$. The evaluated nuclear data files and the cross-section-preparation code NJOY, however, only provide covariance data for the zeroth-order removal scattering cross sections $\mathcal{S}(g, \ell = 0)$. We require (as will be seen in Section 4.3) the sensitivity coefficients with respect to the cross sections for which covariance data is available. It is therefore necessary to calculate sensitivity coefficients for the removal scattering cross sections from the sensitivity coefficients for the transfer scattering cross sections. We also outline how the higher-order moments of the scattering cross sections might be perturbed (without covariance data) to match a realization of the zeroth-order removal scattering cross sections.

Before the relationship between the transfer and removal scattering cross sections is presented, a final bit of notation must be introduced. This notation is used extensively in the portion of the multigroup cross-section generation section covering multivariate Gaussian distributions, Section 2.4. The nominal or average value of the multigroup cross sections is denoted with a subscript μ. A perturbed set of cross sections is denoted with a subscript R. These perturbed cross sections may be generated as part of a finite difference analysis or by the data produced by NJOY.

Let us assume that the ratio of the realized removal cross section to the mean value of the removal scattering cross section is constant across all moments,

$$\frac{\mathcal{S}_R(g,\ell)}{\mathcal{S}_\mu(g,\ell)} = \frac{\mathcal{S}_R(g,0)}{\mathcal{S}_\mu(g,0)}, \tag{3.95}$$

and that the ratios of group g's transfer cross sections to group g's removal cross section is constant across all moments and realizations of the cross sections,

$$\frac{\mathscr{S}_R(g \to g',\ell)}{\mathcal{S}_R(g,\ell)} = \frac{\mathscr{S}_\mu(g \to g',\ell)}{\mathcal{S}_\mu(g,\ell)}. \tag{3.96}$$

These two assumptions allow the calculation of $\mathscr{S}_R(g \to g',\ell)$ and $\mathcal{S}_R(g,\ell)$ from the known values of $\mathscr{S}_\mu(g \to g',\ell)$, $\mathcal{S}_\mu(g,\ell)$, and the covariance data for $\mathcal{S}(g,0)$.

The change in the QoI ΔQ (from the QoI at the mean value of the cross sections) due to a perturbation in the transfer cross sections from a particular group g to any or all groups g', $\Delta\mathscr{S}(g \to g',\ell)$, is

$$\Delta Q = \sum_{g'} \frac{\partial Q}{\partial \mathscr{S}(g \to g',\ell)} \Delta\mathscr{S}(g \to g',\ell). \tag{3.97}$$

The change in the transfer scattering cross section can be expanded

$$\Delta\mathscr{S}(g \to g',\ell) = \mathscr{S}_R(g \to g',\ell) - \mathscr{S}_\mu(g \to g',\ell). \tag{3.98}$$

The realized value of the transfer cross section is calculated from the realized value of the removal cross section using Eq. (3.96),

$$\mathscr{S}_R(g \to g', \ell) = \mathscr{S}_\mu(g \to g', \ell)\frac{\mathcal{S}_R(g,\ell)}{\mathcal{S}_\mu(g,\ell)}. \tag{3.99}$$

Substituting this into Eq. (3.98) and then simplifying where possible,

$$\Delta\mathscr{S}(g \to g', \ell) = \mathscr{S}_\mu(g \to g', \ell)\frac{\mathcal{S}_R(g,\ell)}{\mathcal{S}_\mu(g,\ell)} - \mathscr{S}_\mu(g \to g', \ell), \tag{3.100}$$

$$\Delta\mathscr{S}(g \to g', \ell) = \left(\frac{\mathcal{S}_R(g,\ell)}{\mathcal{S}_\mu(g,\ell)} - 1\right)\mathscr{S}_\mu(g \to g', \ell), \tag{3.101}$$

$$\Delta\mathscr{S}(g \to g', \ell) = \left(\frac{\mathcal{S}_R(g,\ell) - \mathcal{S}_\mu(g,\ell)}{\mathcal{S}_\mu(g,\ell)}\right)\mathscr{S}_\mu(g \to g', \ell). \tag{3.102}$$

This is substituted back into Eq. (3.97), the expression for the change in the QoI,

$$\Delta Q = \left(\frac{\mathcal{S}_R(g,\ell) - \mathcal{S}_\mu(g,\ell)}{\mathcal{S}_\mu(g,\ell)}\right)\sum_{g'}\frac{\partial Q}{\partial\mathscr{S}(g \to g', \ell)}\mathscr{S}_\mu(g \to g', \ell). \tag{3.103}$$

Dividing both sides of this expression by the difference between the realized and mean values of the removal cross section $\Delta\mathcal{S}(g,\ell)$ and then simplifying the resulting equation yields

$$\frac{\Delta Q}{\Delta\mathcal{S}(g,\ell)} = \sum_{g'}\frac{\partial Q}{\partial\mathscr{S}(g \to g', \ell)}\frac{\mathscr{S}_\mu(g \to g', \ell)}{\mathcal{S}_\mu(g,\ell)}. \tag{3.104}$$

Taking a limit as $\Delta\mathcal{S}(g,\ell)$ goes to zero transforms the fraction on the left side of the expression to the partial derivative of the QoI with respect to the removal cross sections,

$$\lim_{\Delta\mathcal{S}(g,\ell)\to 0}\frac{\Delta Q}{\Delta\mathcal{S}(g,\ell)} = \frac{\partial Q}{\partial\mathcal{S}(g,\ell)}, \tag{3.105}$$

$$\frac{\partial Q}{\partial \mathcal{S}(g,\ell)} = \sum_{g'} \frac{\partial Q}{\partial \mathscr{S}(g \to g',\ell)} \frac{\mathscr{S}_\mu(g \to g',\ell)}{\mathcal{S}_\mu(g,\ell)}. \tag{3.106}$$

The sensitivity of the QoI to the removal scattering cross sections is the weighted sum of the sensitivity of the QoI with respect the transfer cross sections, where the weighting coefficients for the transfer scattering cross section from group g to g' in moment ℓ are the ratio of that cross section to the removal cross section in group g and moment ℓ.

3.9.4 Sensitivity Coefficients for Higher-Order Scattering Moments

Although PDT calculates the sensitivity coefficients for all moments of the scattering source, the evaluated nuclear data files and the cross-section-preparation code NJOY only produce covariance data for the zeroth-moment scattering cross sections.

In addition to this gap in the data, the number of inner products that must be performed is proportional to the number of number of energy groups (for the total cross section) plus the number energy groups squared (the transfer cross sections); that is, for a problem with M materials, N energy groups, and L moments of the scattering source, the total number of inner products that must be performed is proportional to $M(N + LN^2)$.

In the problems analyzed in this project, more time was spent performing inner products than solving both the forward and adjoint transport problems. This was true for problems with only isotropic scattering, that is, problems containing one total moment of the scattering source, the zeroth moment. The computational cost to solve problems containing anisotropic scattering was dominated by the calculation of the sensitivity coefficients.

Rather than extrapolating data from the zeroth moment scattering cross sections to the higher order moments, sensitivity coefficients were only calculated for the zeroth moment scattering cross sections. This proved necessary for the test problem corresponding to the Impurity Model One experiments, as discussed in the results section.

4. UNCERTAINTY QUANTIFICATION FOR RADIATION TRANSPORT CALCULATIONS

In the previous sections uncertainty in the multigroup cross sections was quantified, the radiation transport equations were introduced, a quantity of interest was defined, and a method for calculating the partial derivative of a QoI with respect to each multigroup cross section was derived. In this section these separate components will be assembled to estimate the uncertainty in a QoI due to the uncertainty in the nuclear data.

This section opens with the dreaded curse of dimensionality in Section 4.1. In Section 4.2 the standard sampling-based or Monte Carlo method for uncertainty quantification is outlined. In Section 4.3 a standard propagation of uncertainty approach commonly used in conjunction with adjoint sensitivity methods is introduced. A novel scheme for a physics-preserving dimension reduction scheme is presented in Section 4.4. The construction and use of a response surface for this reduced-dimensional space is examined in detail in Section 4.5.

A Python tool was written to perform the actions described in this section. The use of this tool is described in detail in Appendix D.

4.1 The Curse of Dimensionality

Performing an UQ analysis for a problem with a high-dimensional uncertain input space is extremely challenging because of the curse of dimensionality. The curse of dimensionality states that the number of samples required to scrutinize a multidimensional space grows at a prohibitive rate as the number of dimensions in the input space increases. Consider a unit space of dimension D. If $D = 1$, this is a line of length 1. If $D = 2$, this is a unit square; for $D = 3$ a unit cube, and for $D > 3$ a unit hypercube. Suppose we wish to sample this space with a fixed maximum spacing between points L. In each dimension, at least $1/L$ points are required to satisfy this constraint. Let us denote the number of samples required per dimension as a constant C, with $C = 1/L$.

As the dimensionality of the space grows, the number of points required to sample the space N grows as

$$N = C^D. \tag{4.1}$$

In the limiting case of $C = 1$, a single sample is taken at the mean value of the uncertain input parameters. The exponential D term makes regularly-spaced sampling impossibly expensive for problems with more than a few dimensions. Stratified sampling methods and Latin Hypercube Sampling require that many samples are taken per dimension; these approaches are prohibitively expensive for problems with several dimensions [8]. Even the use of sparse evenly-spaced sampling points become impossible as the number of dimensions increases.

As discussed in Section 2.3, radiation transport calculations can contain hundreds or thousands of correlated multigroup cross sections. An input space of this size is impossible to sample with a regular mesh no matter how inexpensive it is to solve the problem of interest. Of course, realistic, interesting problems are expensive to solve and prohibitively expensive to run in large ensembles. At best an extremely sparse set of sample points can be generated as data points with which to perform a UQ analysis.

4.2 Statistical Methods for UQ Analysis

The distribution of a QoI corresponding to a known distribution of uncertain input parameters can be estimated by sampling the uncertain input space and evaluating the QoI at each of these sample points. This is known as a Monte Carlo method. The distribution of the sampled QoI from an unbiased sampling scheme will converge to the QoI's actual distribution as the number of samples increases [9]. An unbiased sampling method randomly samples points in the input space based upon the distributions assumed for those parameters.

More complex sampling schemes, like Stratified Sampling or Latin Hypercube Sampling, can be employed to reduce the number of samples required to calculate the QoI's distribution. These schemes essentially divide the range of each uncertain parameter into equally-likely bins and then sample randomly from within each of these bins. This approach preserves the original probability distribution of the uncertain parameter but ensures that samples are spread across the parameters' domain.

The error in a Monte Carlo analysis E_{MC} using unbiased sampling converges at a rate proportional to the square root of the number of samples N,

$$E_{MC} \propto \frac{1}{\sqrt{N}}. \tag{4.2}$$

This result means that to halve the error in the MC solution, the number of simulations must be quadrupled [21]. While this makes the variance reduction for a Monte Carlo method predictable, these methods are often prohibitively expensive for complex problems with a large number of uncertain parameters.

The Monte Carlo method with straightforward unbiased sampling is employed in the simple example problems examined in this work as the reference solution to which other solutions can be compared. The complex problems which motivated this work are expensive enough that a Monte Carlo UQ analysis is impossible.

4.3 Deterministic Methods for UQ Analysis

Deterministic methods for uncertainty quantification, like the adjoint sensitivity method (Section 3.5) or the finite difference sensitivity method (Section 3.7), are typically used in conjunction with the propagation of uncertainty relationship to approximate uncertainty in the QoI. This is in contrast to Monte Carlo methods for UQ which estimate the mean and variance of a QoI directly from the QoI's distribution at a large number of sample points.

The "propagation of error" or "propagation of moments" equation is derived from a Taylor series expansion of the QoI as a function of the input parameters. Terms beyond first order are dropped; although it is possible to derive adjoint equations to yield second- (and higher-) order derivatives of the QoI with respect to the parameters of interest, the adjoint equations for these derivatives become increasingly complex [15]. After neglecting higher order terms and taking moments of the QoI, the variance in the QoI σ_Q^2 is approximated as

$$\sigma_Q^2 = \vec{S_p}^T \overline{\overline{\Sigma_p}} \vec{S_p}. \tag{4.3}$$

Here $\overline{\overline{\Sigma_p}}$ is the covariance matrix introduced in Section 2.4 and $\vec{S_p}$ is the vector of partial derivatives of Q, $\frac{\partial Q}{\partial p}$ [8]. From this equation we see that the uncertainty in a quantity of interest depends both upon the sensitivity of that QoI to each uncertain parameter S_p and the variances in and covariances between the parameters p.

Let us briefly consider a problem which contains only uncorrelated parameters. Qualitatively, if either the sensitivity $\frac{\partial Q}{\partial p}$ to a particular parameter p or the uncertainty (quantified as the variance, σ_p^2) in a particular parameter p is zero, then that parameter gives rise to no uncertainty in the QoI. Extending that logic, if the QoI is only weakly sensitive to a parameter with small uncertainty, the uncertainty in the QoI due to that parameter can be expected to be quite small. All other combinations of σ_p^2 and S_p will contribute to uncertainty in the QoI. These qualitative limiting cases are organized in Table 4.1.

σ_p^2	S_p	σ_Q^2
zero	zero	zero
zero	large	zero
large	zero	zero
small	small	very small
non-zero	non-zero	significant

Table 4.1: Qualitative analysis of several limiting cases for a QoI Q with a single uncertain parameter p and sensitivity S_p.

In the case of uncorrelated parameters, the contribution of each parameter to the variance in the QoI is straightforward to calculate. The covariance matrix $\overline{\overline{\Sigma_p}}$ for an uncorrelated set of N parameters p is simply a diagonal matrix containing the variance in the N parameters. This simplifies Eq. (4.3) to

$$\sigma_Q^2 = \sum_{i=1}^{N} S_{p_i}^2 \sigma_{p_i}^2. \tag{4.4}$$

This qualitative analysis is more complex for a problem with multiple correlated parameters. Because the off-diagonal terms in the covariance matrix are non-zero, the simplification which allowed us to form Eq. (4.4) from Eq. (4.3) can not be made. The covariance matrices generated in Section 2.3 were produced per isotope; at best, then, Eq. (4.3) can be applied on a per-isotope basis to estimate the contribution of each isotopes' uncertainty to the uncertainty in the QoI.

4.4 Dimension Reduction and the Identification of Important Independent Parameters

A novel dimension-reduction scheme is introduced in this section to reduce the dimensionality of the highly-correlated multigroup cross-section space. This scheme is notable for preserving the correlations between the multigroup data and for selecting parameters of interest based upon their contribution to the uncertainty in the QoI. This scheme is fundamentally unlike the dimension reduction schemes put forth by Hany et al. in [2] and by Constantine et al. in [11]. A brief summary of the difference between the method introduced here and the methods in [2] and [11] is included in Section 4.4.5.

4.4.1 Computational Method to Draw Samples from a Multivariate Gaussian Distribution

The evaluated nuclear data files approximate the multigroup cross sections as a multivariate Gaussian distribution with mean $\vec{\mu}_x$ and covariance matrix $\overline{\overline{\Sigma_x}}$. The vector of multigroup cross sections will be represented by $\vec{x}$ and the subscript x will be used to denote a value for the cross-section data, as in Section 2.3.

$$\vec{x} \sim N(\vec{\mu}_x, \overline{\overline{\Sigma_x}}). \tag{4.5}$$

A computationally efficient method to sample a multivariate Gaussian distribution is presented in [16] and outlined here. To sample the cross-section distribution we must compute a matrix $\overline{\overline{A_x}}$ such that

$$\overline{\overline{A_x}}\,\overline{\overline{A_x}}^T = \overline{\overline{\Sigma_x}}. \tag{4.6}$$

Recalling that the covariance matrix $\overline{\overline{\Sigma_x}}$ is symmetric, a Cholesky factorization satisfies this relationship. Let us denote the matrices generated with the Cholesky factorization $\overline{\overline{C_x}}$,

$$\overline{\overline{C_x}}\,\overline{\overline{C_x}}^T = \overline{\overline{\Sigma_x}}. \tag{4.7}$$

The covariance matrices generated from the evaluated nuclear data files with NJOY are not always symmetric positive definite, which can cause a Cholesky decomposition to fail. As a second approach to computing a matrix $\overline{\overline{A_x}}$ that satisfies Eq. (4.6), let us take an eigenvalue decomposition of the covariance matrix

$$\overline{\overline{\Sigma_x}} = \overline{\overline{E_x}}\,\overline{\overline{\Lambda_x}}\,\overline{\overline{E^T_x}}. \tag{4.8}$$

The matrix $\overline{\overline{\Lambda_x}}$ is diagonal and contains the eigenvalues of $\overline{\overline{\Sigma_x}}$. The matrix $\overline{\overline{E_x}}$ contains the right eigenvectors corresponding to the eigenvalues in $\overline{\overline{\Lambda_x}}$. If we split the diagonal $\overline{\overline{\Lambda_x}}$ into $\overline{\overline{\Lambda_x}}^{1/2}\overline{\overline{\Lambda_x}}^{1/2}$, Eq. (4.8) can be written as

$$\overline{\overline{\Sigma_x}} = \overline{\overline{E_x}}\,\overline{\overline{\Lambda_x}}^{1/2}\overline{\overline{\Lambda_x}}^{1/2}\overline{\overline{E^T_x}}, \tag{4.9}$$

and we see that we can compute a transform matrix $\overline{\overline{A_x}}$ that satisfies Eq. (4.6)

$$\overline{\overline{A_x}} = \overline{\overline{E_x}}\,\overline{\overline{\Lambda_x}}^{1/2}. \tag{4.10}$$

The eigenvalue decomposition (with minor modification, outlined in the following section) is suitable for application to matrices for which the more common Cholesky factorization fails. Note that the matrix produced from this approach $\overline{\overline{A_x}}$ is not the same matrix as produced by the Cholesky decomposition $\overline{\overline{C_x}}$,

$$\overline{\overline{A_x}} \neq \overline{\overline{C_x}}, \tag{4.11}$$

although both satisfy Eq. (4.6).

Given the transform matrix $\overline{\overline{A_x}}$ calculated from the eigenvalue decomposition, we next define a vector $\vec{z}$ containing independent normal members z_i with mean zero and variance one,

$$z_i \sim N(0,1). \tag{4.12}$$

We refer to the $N(0,1)$ distribution as the "unit normal" distribution.

We will denote samples of the cross sections $\vec{x}$ and the independent-parameter space $\vec{z}$ with a subscript R_i for "the i-th realization." Samples of the cross-section space are produced as the product of $\overline{\overline{A_x}}$ and samples of $\vec{z}$,

$$\vec{x}_{R_i} = \overline{\overline{E_x}}\,\overline{\overline{\Lambda}}_x^{1/2} \vec{z}_{R_i} + \vec{\mu}. \tag{4.13}$$

These samples of the cross-section space were written as PDT-formatted nuclear data files using the Python tool mentioned in Section 2.6.

4.4.2 Dimension Reduction by the Eigenvalue Decomposition of the Covariance Matrix

In general, a covariance matrix should be positive semi-definite [16]. However, the matrices calculated with NJOY contain negative eigenvalues. We have observed that the negative eigenvalues encountered in the covariance matrices generated in this work were generally several orders of magnitude smaller than the largest positive eigenvalues. We therefore set the negative eigenvalues in $\overline{\overline{\Lambda}}_x$ to zero.

The positive eigenvalues were truncated based upon the contribution of each eigenvalue to the sum of the positive eigenvalues. Ordering the eigenvalues λ_i from largest to smallest, the first N eigenvalues were kept and the rest set to zero such that

$$\frac{\sum_{i=1}^{N} \lambda_i}{\sum \lambda_i} \geq 1 - \epsilon. \tag{4.14}$$

The value of ϵ in this work was conservatively chosen to be 1E-8.

The covariance matrix and its associated eigenvalues for a set of uranium-235 cross sections is presented as a representative example of the magnitude and sign of an NJOY-produced covariance matrix's eigenvalues. The generation of this eighty group multigroup cross-section data set is discussed in Section 5.3.1. The covariance matrix is visualized in Figure 4.1.

The full covariance matrix contains seventeen different nuclear reactions. Of these, only the reactions with non-zero cross sections are displayed in Figure 4.1. With eighty energy groups, the covariance matrix has 1360 rows and columns (and eigenvalues). These eigenvalues are plotted from largest to smallest in Figure 4.2.

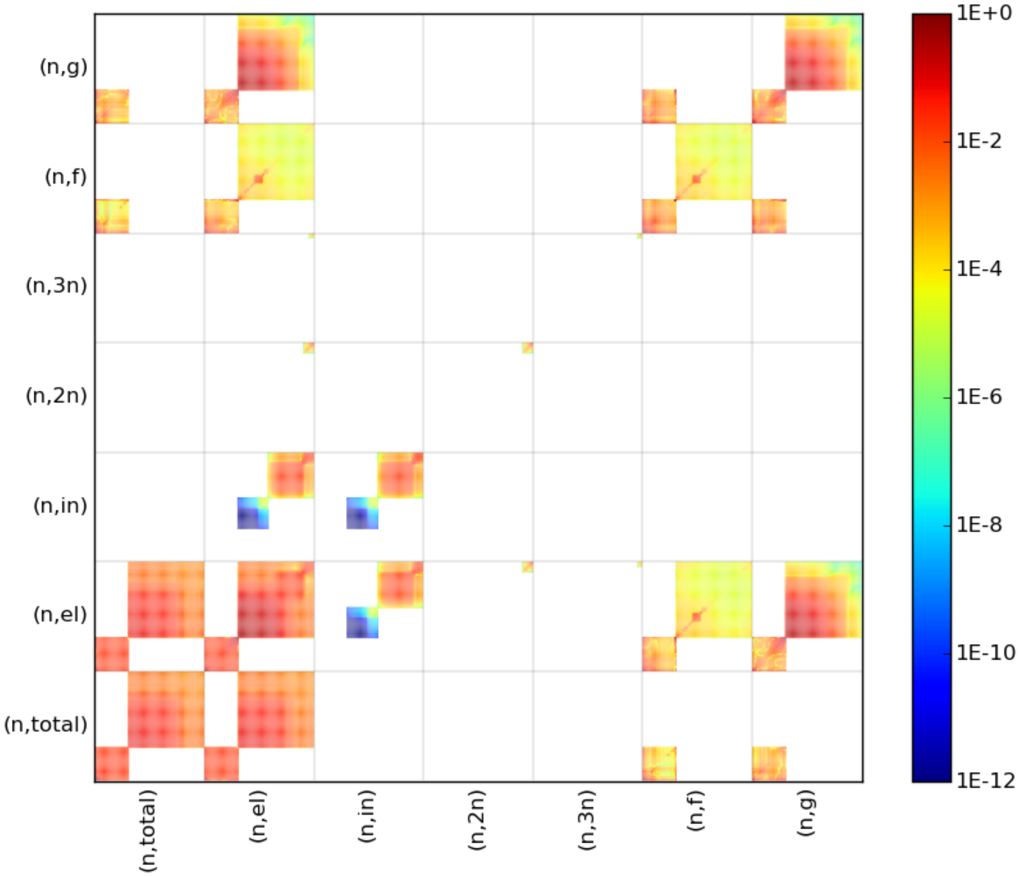

Figure 4.1: Covariance matrix for uranium 235 produced with 80 energy groups and a thermal Maxwellian, 1/E, and fission spectrum weighting spectrum.

The largest positive eigenvalue is 8.91. The largest negative eigenvalue is -3.2E-4. This disparity in magnitudes suggests that setting the negative eigenvalues to zero will not cause a significant loss of information.

The $\overline{\overline{\Lambda_x}}$ and $\overline{\overline{A_x}}$ matrices may be reshaped to reflect the truncated collection of eigenvalues present in $\overline{\overline{\Lambda_x}}$. This reduces the size of the vector of independent unit normal parameters $\vec{z}$ which must be drawn to sample the multivariate Gaussian distribution. This truncation is therefore a matrix-based dimension reduction of the cross-section space. In the U-235 example, the cross-section space contained 387

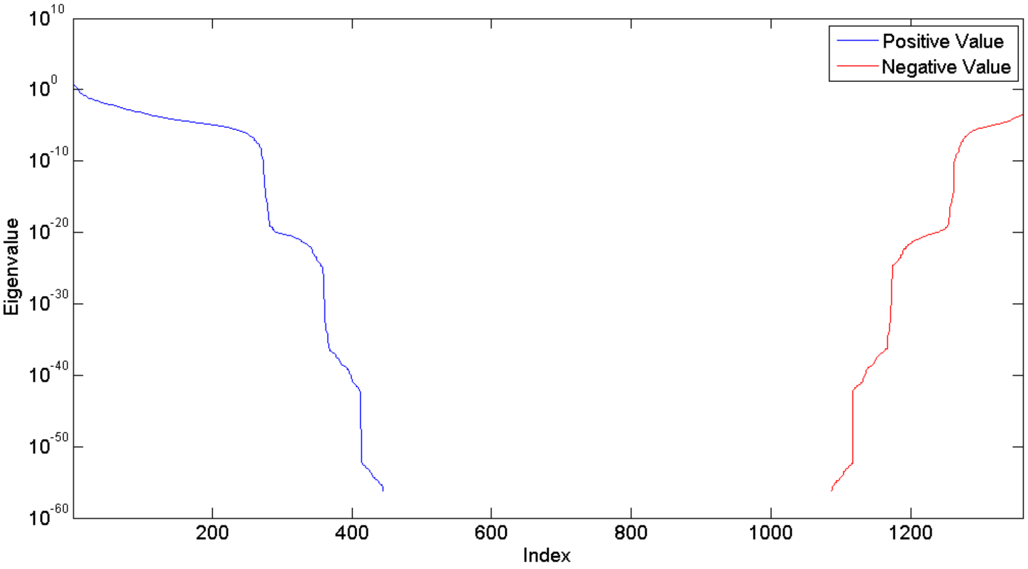

Figure 4.2: All eigenvalues of the 80-group uranium 235 covariance matrix visualized in Figure 4.1.

nonzero multigroup cross sections, 445 positive eigenvalues, and 261 eigenvalues after truncation.

4.4.3 Mapping Between Cross Sections and Uncorrelated Parameters

A mapping can be defined for the transformation between the highly correlated multigroup cross sections and the uncorrelated, unit normal parameters used to draw samples from the cross-section distribution. A Jacobian matrix can be calculated from Eq. (4.13) by considering the general case (not specific to a particular realization) and then taking a derivative of $\vec{x}$ with respect to $\vec{z}$,

$$\frac{d\vec{x}}{d\vec{z}} = \frac{d}{d\vec{z}}\left(\overline{\overline{E_x}}\,\overline{\overline{\Lambda}}_x^{1/2}\vec{z} + \vec{\mu}\right), \tag{4.15}$$

Applying the chain rule to this yields

$$\overline{\overline{\frac{dx}{dz}}} = \overline{\overline{E_x}}\,\overline{\overline{\Lambda}}_x^{1/2}. \tag{4.16}$$

This expression defines the mapping between the uncorrelated-parameter space containing $\vec{z}$ and the correlated multigroup cross-section space containing $\vec{x}$. In general there may be as many uncorrelated parameters z_i as cross sections x_i, but in practice the number of independent parameters is less than the number of cross sections due to the truncation of the eigenvalues in $\overline{\overline{\Lambda}}$. This means that the matrix $\overline{\overline{\frac{dx}{dz}}}$ is not necessarily square, but rather $n \times m$, where n is the number of multigroup cross sections x_i and m the number of uncorrelated parameters z_i.

A similar mapping can be defined for the derivatives of the QoI and the two spaces. The derivative of the QoI with respect to the parameters of interest is related to the derivative of the QoI with respect to the cross sections by the chain rule,

$$\frac{dq}{dz} = \frac{dq}{dx}\frac{dx}{dz}. \tag{4.17}$$

The $\frac{dq}{dz}$ and $\frac{dq}{dz}$ terms are vectors and the $\frac{dx}{dz}$ term a matrix, as defined in Eq. (4.16). Given the shape of the transform matrix $\frac{dx}{dz}$, the vectors $\frac{d\vec{q}}{dz}$ and $\frac{d\vec{q}}{dx}$ are $1 \times m$ and $1 \times n$, respectively.

$$\frac{d\vec{q}}{dz}^T = \frac{d\vec{q}}{dx}^T \overline{\overline{\frac{dx}{dz}}}. \tag{4.18}$$

This expression is simplified by applying the transpose operator to (4.18),

$$\frac{d\vec{q}}{dz} = \overline{\overline{\frac{dx}{dz}}}^T \frac{d\vec{q}}{dx}. \tag{4.19}$$

Substituting Eq (4.16) for the transform matrix $\overline{\overline{\frac{dx}{dz}}}$,

$$\frac{d\vec{q}}{dz} = \left[\overline{\overline{E_x}}\,\overline{\overline{\Lambda}}_x^{1/2}\right]^T \frac{d\vec{q}}{dx}. \tag{4.20}$$

The matrix of eigenvalues is diagonal, so transposing it does not change it.

$$\frac{d\vec{q}}{dz} = \overline{\overline{\Lambda}}^{1/2}\,\overline{\overline{E_x}}^T \frac{d\vec{q}}{dx}. \tag{4.21}$$

Finally, for clarity let us replace the vectors of sensitivity coefficients $\frac{d\vec{q}}{dx}$ and $\frac{d\vec{q}}{dz}$ with $\vec{S}_x$ and $\vec{S}_z$, respectively,

$$\vec{S}_z = \overline{\overline{\Lambda}}_x^{1/2}\,\overline{\overline{E_x}}^T \vec{S}_x. \tag{4.22}$$

We have established in this section a mapping between the uncorrelated-parameter space and the multigroup cross-section space, Eq. (4.16), and a mapping from the partial derivatives with respect to the multigroup cross sections to the partial derivatives of the QoI with respect to the uncorrelated parameters z, Eq. (4.22). This mapping allows us to work in the uncorrelated-parameter space instead of the cross-section space.

There are several advantages to working in this space. First, as noted earlier, the dimensionality of the independent-parameter space is usually significantly smaller than the dimensionality of the multigroup cross-section space. Second, because the z-parameters are unit normal, they are all centered at 0 and have variance 1. This greatly simplifies the construction of a response surface because each parameter shares the same range. Finally, the mapping between spaces preserves the physical correlations between the multigroup data while allowing the response surface to be constructed in terms of uncorrelated parameters.

4.4.4 Dimension Reduction in the Uncorrelated-Parameter Space

A major reduction in the number of important dimensions may be accomplished with the information now available to us. First, from Eq. (3.24), the QoI can be expressed as a function of the solution to the forward transport equation,

$$Q = \left\langle \psi, q^{\dagger} \right\rangle .$$

Let us denote ψ and $q^{\dagger}$ functions of the cross sections at each realization, $\vec{x}_{R_i}$, and denote by Q_{R_i} the QoI at specific realization R_i of the cross sections $\vec{x}_{R_i}$,

$$Q_{R_i} = \left\langle \psi(\vec{x}_{R_i}), q^{\dagger}_{R_i} \right\rangle . \tag{4.23}$$

The forward solution can also be considered a function of the uncorrelated parameters,

$$Q_{R_i} = \left\langle \psi(\vec{z}_{R_i}), q^{\dagger}_{R_i} \right\rangle . \tag{4.24}$$

The partial derivatives of the QoI with respect to the multigroup cross sections are calculated at each realization of the cross sections, and by Eq. (4.22) these partial derivatives are mapped to the uncorrelated-parameter space,

$$\vec{S}_z\Big|_{R_i} = \overline{\overline{\Lambda}}_x^{1/2} \overline{\overline{E_x}}^T \, \vec{S}_x\Big|_{R_i} .$$

The variance in the QoI can be estimated at each realization of the cross sections by Eq. (4.3) using either the sensitivities and covariance matrix corresponding to the multigroup cross sections,

$$\sigma_Q^2\Big|_{R_i} = \vec{S}_x\Big|_{R_i} \overline{\overline{\Sigma_x}}\ \vec{S}_x\Big|_{R_i}, \tag{4.25}$$

or the uncorrelated parameters,

$$\sigma_Q^2\Big|_{R_i} = \vec{S}_z\Big|_{R_i} \overline{\overline{\Sigma_z}}\ \vec{S}_z\Big|_{R_i}. \tag{4.26}$$

Note that this is true even though there are a different number of multigroup cross sections x_i and uncorrelated parameters z_i.

The covariance matrix for the uncorrelated parameters is diagonal, and because the z parameters are unit normal, the diagonal entry of Σ_z is one. Let us denote the number of z parameters as N_z. As we saw in Eq. (4.4), this means the variance the QoI is the sum of the N_z squared sensitivity coefficients $\vec{S}_z$,

$$\sigma_Q^2\Big|_{R_i} = \sum_{j=1}^{N_Z} S_{z_j}^2\Big|_{R_i}. \tag{4.27}$$

An average variance in the QoI $\widehat{\sigma_Q^2}$ can be computed by averaging the variance estimates at each of the M realizations of the cross section,

$$\widehat{\sigma_Q^2} = \frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{N_Z} S_{z_j}^2\Big|_{R_i}. \tag{4.28}$$

This averaging attempts to compute a global variance in the QoI from the local sensitivity coefficients. If the partial derivatives of the QoI with respect to the uncorrelated parameters are slowly varying, than this averaging is unnecessary. However, if the sensitivity coefficients vary significantly across the input space, this step captures that change.

The average contribution of each z-parameter to the variance in the QoI $\widehat{\sigma^2_{Q,z_j}}$ is computed

$$\widehat{\sigma^2_{Q,z_j}} = \frac{1}{M} \sum_{i=1}^{M} S^2_{z_j}\Big|_{R_i}. \tag{4.29}$$

The fraction of the total variance contributed by each independent parameter z_j is easily calculated from Eq. (4.28) and Eq. (4.29). The independent parameters can be ordered from most important to least important by their contribution to the variance in the QoI. We denote the ordered average contributions to the total variance by each z-parameter $\left[\widehat{\sigma^2_{Q,z_j}}\right]^{\text{ranked}}$ and indicate the index in this list with a subscript k. The K important independent parameters are chosen such that

$$\frac{\sum_{k=1}^{K} \left[\widehat{\sigma^2_{Q,z_j}}\right]^{\text{ranked}}_k}{\widehat{\sigma^2_Q}} \geq 1 - \epsilon. \tag{4.30}$$

These K parameters are the important independent z-parameters which are considered in the construction of a response surface. In this work the fraction of the variance which was retained in the selection of important parameters was 0.99 ($\epsilon = 0.01$) in all but one problem of interest. The problem which departed from this setting is clearly noted in the results. This step tremendously reduces the number of dimensions which must be considered while retaining the dimensions responsible for almost all of the variance in the QoI.

In summary, the reduction of the multigroup cross-section space is accomplished in two distinct steps in the process of mapping to the independent-parameter space. First, a matrix-based dimension reduction is accomplished based upon the magnitude of the eigenvalues of the covariance matrix. This step is responsible for a small decrease in the problem dimensionality. Second, an importance-based reduction is accomplished based upon the linear propagation of uncertainty and evaluations of the sensitivity at various points in the input space. Most of the total reduction in the problem's dimensionality is accomplished in this importance-based step.

4.4.5 A Note on Active Subspace Methods

The dimension reduction scheme presented in this section preserves the physical relationships between the correlated multigroup cross sections by the nature of the matrix decomposition and reduces the dimensionality of the problem by isolating parameters that are responsible for the variance in the QoI. The active subspace methods in [2] and [11] are fundamentally different in that they construct a Jacobian matrix in the QoI space. A matrix decomposition of this matrix is used to identify a low-dimensional active subspace of the full-dimensional space. The method outlined in this work has two advantages when compared to these active subspace methods.

First, the scheme outlined in this work is based not just upon the sensitivity of the QoI to each dimension, but also upon the uncertainty in that dimension. As discussed in Section 4.3, both of these components must be considered to determine the contribution of any parameter to the variance in the QoI. The active subspace methods might miss a parameter that has a small sensitivity but a large uncertainty, therefore contributing significant uncertainty to the QoI.

Second, the dimensions that comprise the uncorrelated space used in this work preserve the relationships between multigroup cross sections. When using the active subspace methods, it might be possible to select some cross sections as important while discarding tightly correlated but relatively unimportant cross sections. This does not happen using the uncorrelated parameters in this work. If any of the multigroup cross sections associated with an uncorrelated parameter are important, then all of the cross sections tightly correlated to that parameter are automatically preserved by the selection of the uncorrelated parameter. While this may be a negligible difference between the two approaches, the scheme outlined in this work preserves these physical relationships between the multigroup cross sections.

4.5 Construction and Use of a Response Surface for UQ

The central novel idea in this work is the use of the adjoint sensitivity method introduced in Section 3.5 and the dimension reduction scheme introduced in Section 4.4 to skirt the curse of dimensionality and construct an accurate and predictive response surface from extremely sparse samples of the cross-section space. In this section the concept of a response surface is introduced and an algorithm for efficiently constructing such a surface is presented.

A response surface is an approximate model of the QoI as a function of the input parameters [23]. Let us rephrase Eq. (3.24) such that the QoI is a function of the cross sections, $\vec{x}$,

$$Q = f(\vec{x}). \tag{4.31}$$

A response surface seeks to construct a model $\tilde{f}$ for the QoI such that

$$Q = \tilde{f}(\vec{x}). \tag{4.32}$$

The surrogate model $\tilde{f}(x)$ must be inexpensive to evaluate compared to the full model. This surrogate model is constructed from evaluations of the full problem at various points in the input space. Ideally these points are chosen by stratified sampling or latin hypercube sampling to ensure points are spread across the input space. Once constructed, a response surface can be used to rapidly sample the QoI over the input space, from which moments of the distribution or distribution extrema can be determined [23][30].

In order to use a response surface for uncertainty quantification the model must be predictive. This means that it must accurately reflect the behavior of the QoI at both the points in the input space where the code has been run and at points in the input space away from the existing evaluations. In this section an algorithm to construct a predictive response surface for radiation transport calculations with uncertain cross-section data is explained.

4.5.1 Construction of a Response Surface Using Sensitivity Data

Although the dimension reduction scheme introduced in 4.4.4 greatly reduces the number of dimensions over which a response surface must be constructed, the curse of dimensionality still prevents thorough sampling of the input space for most problems. The partial derivatives calculated with the adjoint sensitivity method are used to greatly increase the amount of information extracted from each run of the full problem at points in the input space.

For an input space with D uncertain parameters or dimensions, each forward simulation produces the QoI at that point in the input space. N simulations, therefore, produce N data points with which the surface can be constructed. The adjoint sensitivity method requires two solutions of the problem per point in the input space (the forward and adjoint solves) and the calculation of numerous inner products, but produces produces the QoI and the partial derivatives $\frac{\partial Q}{\partial d}$ in each of the D dimensions at that point in the input space.

The calculation of the inner products for each parameter of interest is a significant cost associated with the adjoint sensitivity method. Although each inner product is inexpensive, the total number of required inner products is proportional to the number of isotopes times the number of groups plus the number of groups squared for the total and transfer scattering cross sections, respectively. A more detailed analysis of this cost is included in the results for the IM-1 test problem in Section 5.2.2, but in general the cost of computing forward and adjoint solutions and then performing the inner products was similar in total cost to between five and ten solutions of the forward problem. If the parameters of interest do not include the transfer cross sections, however, the inner products are much less expensive than the solution of the forward and adjoint transport problems.

If the partial derivatives do not vary rapidly across the QoI space, than at each point in the input space the adjoint sensitivity method provides enough data to accurately model the surface in the vicinity of that point. If the partial derivatives are slowly varying, this approach should make it possible to construct a response surface with very sparse sampling points. In the simple limit, if the QoI varies linearly in each dimension, than a single pair of forward and adjoint solutions and one set of inner product calculations for the parameters of interest will allow the QoI to be exactly modeled. Using forward simulations alone would require a $D+1$ simulations to reach the same conclusion.

In the complex limit, if the QoI varies on a fine scale, than it may not be possible to construct an accurate model with a sparse sampling points. This scenario, however, is intractable whether statistical methods or deterministic methods are used for a UQ analysis.

4.5.2 The Gradient-Enhanced Multivariate Adaptive Regression Spline (GEMARS) Model

A multivariate adaptive regression spline (MARS) model was chosen as the method by which a response surface is constructed for this project. This choice reflects the need for a model that is relatively rapid to construct, possible to build in a high-dimensional space in which all of the dimensions are important, and which can incorporate gradient information. The gradient-enhanced MARS (GEMARS) model used in this project is based on work by Stripling (of the Texas A&M Nuclear Engineering Department) [30], which built upon work by Holmes [12] and Denison et al. [13]. An early version of the algorithm developed in this work was presented in [7].

A MARS algorithm assembles a collection of splines to model available data. Splines are randomly proposed and then fit to the data by solving a least-squares problem. The splines used in this work are defined by their dimension(s), knot, orientation and order. As example spline is presented in Fig. 4.3 to assist in the explanation of these components.

The spline presented in Fig. 4.3 has one dimension, which will be referred to as x. The knot point at $x = 0.2$ is clearly visible. The orientation of the spline refers to the side of the knot which is non-zero; in this case the spline's orientation is positive. The spline is linear, so it is of order one. This equation for this spline can be represented

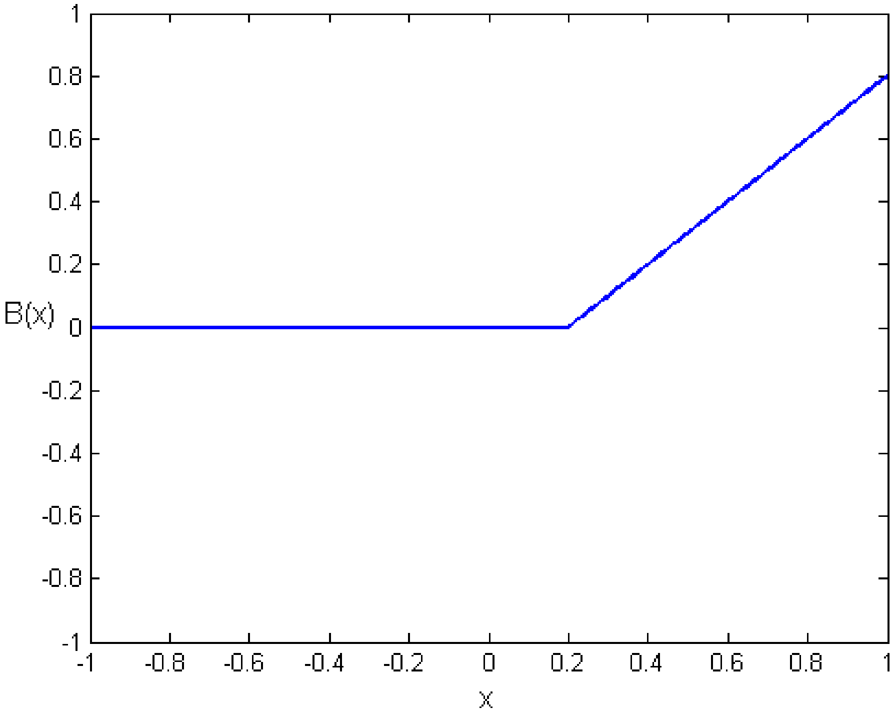

Figure 4.3: An example spline.

mathematically as

$$B(x) = (x - 0.2)_+ \, , \tag{4.33}$$

where the power of 1 is implicit and the subscript + notation is used to signify that the spline's orientation is positive, that is,

$$B(x) = \begin{cases} 0 & \text{if} \quad x - 0.2 < 0 \\ B(x) & \text{if} \quad x - 0.2 \geq 0 \end{cases} \, . \tag{4.34}$$

A general spline B_k can be defined using this notation as

$$B_k(\vec{x}) = \prod_{d=0}^{D} (x_d - t_{k,d})_{\pm}^{o_{k,d}}. \tag{4.35}$$

Note that a spline can exist in multiple dimensions $d = 1, 2, ..., D$ as the product of terms in each dimension. The position in the multidimensional input space now appears $\vec{x}$, with the x-position in dimension d x_d, and the knot point for the k-th spline in dimension d is labeled $t_{k,d}$. The $o_{k,d}$ term is the order of the k-th spline in dimension d. As above, the $()_{\pm}$ notation specifies that

$$(x_d - t_{k,d})_{+}^{o_{k,d}} = \begin{cases} 0 & \text{if} \quad (x_d - t_{k,d}) < 0 \\ (x_d - t_{k,d})^{o_{k,d}} & \text{if} \quad (x_d - t_{k,d}) \geq 0 \end{cases} \tag{4.36}$$

and

$$(x_d - t_{k,d})_{-}^{o_{k,d}} = \begin{cases} (x_d - t_{k,d})^{o_{k,d}} & \text{if} \quad (x_d - t_{k,d}) < 0 \\ 0 & \text{if} \quad (x_d - t_{k,d}) \geq 0 \end{cases}. \tag{4.37}$$

The MARS model $B(\vec{x})$ is a collection of splines each multiplied by a coefficient β_k,

$$B(\vec{x}) = \beta_0 + \sum_{k=1}^{K} \beta_k \prod_{d=0}^{D} (x_d - t_{k,d})_{+}^{o_{k,d}}, \tag{4.38}$$

and a lift function β_0. Given a collection of splines, the coefficients β_k are determined by solving a least-squares problem for the given data. The least squares problem takes the form

$$\overline{\overline{B}}\vec{\beta} = \vec{q}, \tag{4.39}$$

where the spline coefficients β_k are located in the k-th row of the vector $\vec{\beta}$, the QoI at realization R_i of the input space Q_{R_i} is in the i-th row of the vector q, and the matrix

$\overline{\overline{B}}$ contains in the i-th row and k-th column the evaluated terms $B_k(\vec{x}_{R_i})$.

The matrix B has dimension $M \times K$, where M is the number of realizations of the input space and K is the number of splines in the model. The matrix inversion necessary to solve the least-squares problem is performed directly.

Gradient information can be incorporated in the MARS model by including the sensitivity coefficients $\frac{\partial Q}{\partial x_j}$ in the least squares problem. The derivative of the MARS model with respect to the j-th parameter is

$$\frac{\partial B}{\partial x_j} = \sum_{k=1}^{K} o_{k,n} \beta_k \prod_{d=0}^{D} (x_d - t_{k,d})_+^{o^*_{k,d}}, \tag{4.40}$$

where $o^*_{k,d}$ is defined

$$o^*_{k,d} = \begin{cases} o_{k,d} - 1 & \text{if} \quad d = j \\ o_{k,d} & \text{if} \quad d \neq j \end{cases} . \tag{4.41}$$

These derivative terms must be included in the least squares problem. The vector $\vec{\beta}$ is unchanged, but the matrix $\overline{\overline{B}}$ must now include a row for each of the M realizations of the input space and each of the D dimensions containing $\left.\frac{\partial B}{\partial x_j}\right|_{x_{R_i}}$. Similarly, the vector $\vec{q}$ contains Q_{R_i} in the first M rows and then partial derivatives $\left.\frac{\partial Q}{\partial x_j}\right|_{x_{R_i}}$ for each of the M realizations and each of the D dimensions on the input space.

The dimension of the matrix $\overline{\overline{B}}$ in now $(M + MD) \times K$, where M is the number of realizations of the input space, D is the dimensionality of the input space, and K is the number of splines in the MARS model. The cost of solving the least squares problem grows with the number of splines, the number of realizations of the input space, and the dimensionality of the input space. It is therefore desirable to minimize each of these factors.

The least-squares solve selects spline coefficients β_k to best satisfy the available data given the existing collection of splines. The generation of the collection of splines and the selection of the spline coefficients is discussed in the following section.

4.5.3 Implementation of the GEMARS Algorithm

The MARS splines introduced in the last section contain a variety of parameters that must be specified before the least-squares problem can be solved for the spline coefficients β_k. In this section the algorithm for populating the GEMARS model with splines, choosing the spline coefficients, and building a predictive model is described.

The requirement that the response surface be predictive necessitates dividing the simulation data into "training points" and "testing points." The training data is used in the least-squares fit of the spline model to the data. The testing points are used to evaluate the error in the surface at points not included in the surface fitting step. For a given set of data, 75% of the realizations were used as training data and 25% of the realizations were reserved as testing data.

A Metropolis-Hastings Markov Chain Monte Carlo (MCMC) algorithm is used to construct the GEMARS model [17]. The general idea of a MCMC MARS algorithm is to iteratively perturb the collection of splines, solve the least-squares problem for the best possible model fit to the training data with the current collection of splines, and then test the model's fit to the testing data. If the model is improved by the random perturbation, the perturbation is accepted; if the unperturbed model is a better fit to the data than the perturbed model, the change is rejected.

The random perturbations may take three forms. First, a new spline may be birthed.

In this case the number of dimensions, the dimension(s), knot point(s), spline order(s), and orientation(s) of the spline are randomly chosen. The spline is evaluated at each training point and a column is added to the matrix $\overline{\overline{B}}$. Because the parameters upon which the response surface is constructed are unit normal, the knot points are limited to the range $x \in [-4, 4]$. More than 99.9% of points sampled from a normal distribution should fall within this range.

The second possible perturbation is to change an existing spline. In this work, this change was taken to be the random relocation of the spline's knot point. This requires re-evaluating the spline at each of the training points and updating the appropriate column of the matrix $\overline{\overline{B}}$.

The final possible perturbation to the model is to kill an existing spline. In this case a column of the matrix $\overline{\overline{B}}$ is removed.

Two of these three possible perturbations requires the evaluation of one spline at each of the training points. The third perturbation, killing a spline, requires no additional spline evaluations. The matrix $\overline{\overline{B}}$ does not need to be built from scratch at each iteration; the existing matrix changes in only one column. At each iteration solving the least-squares problem dominates the computational cost of the iteration.

After solving the least-squares problem for the spline coefficients, the model must be evaluated at the testing points. As with the training points, the unperturbed splines need not be re-evaluated at each testing point. Instead, the unperturbed, previously-evaluated splines need only be multiplied by the newly-computed spline coefficients. The perturbed spline must be evaluated at each testing point and multiplied by its spline coefficient. The model is then evaluated via Eq. (4.38). The error in the model $\tilde{f}$ at iteration i, E_i, is calculated as a normalized L_2 norm of the relative error in both the QoI and the sensitivities at the T training points, x_{R_t},

$$E_i = \frac{1}{T}\sqrt{\sum_{t=1}^{T}\left(\frac{Q(\vec{x}_{R_t}) - \tilde{f}(\vec{x}_{R_t})}{Q(\vec{x}_{R_t})}\right)^2 + \sum_{t=1}^{T}\sum_{d=1}^{D}\left(\frac{\left.\frac{\partial Q}{\partial x_d}\right|_{R_t} - \left.\frac{\partial \tilde{f}}{\partial x_d}\right|_{R_t}}{\left.\frac{\partial Q}{\partial x_d}\right|_{R_t}}\right)^2}\,. \tag{4.42}$$

If the error in the model at the testing points is decreased by the random perturbation to the model, the change is accepted. If the error is increased, the change is rejected and the model reverts to its unperturbed state. Note that in order to efficiently revert to the unperturbed state, a copy of the unperturbed collection of splines and the unperturbed matrix $\overline{\overline{B}}$ must be stored at each iteration. This copy command can be computationally expensive.

If the error in the model at the testing points satisfies some preselected error tolerance, the model is accepted and the iterations are terminated.

4.5.4 Use of a Response Surface for UQ

Once the MCMC GEMARS model converges to the specified tolerance, the response surface can be used for uncertainty quantification. Because the model is extremely quick to sample it can be thoroughly sampled. The C++ algorithm written for this work, for example, was able to perform 1E5 samples in O(10) seconds on a typical response surface. From this large number of samples the mean, variance, and any desired higher-order moments of the QoI distribution can be directly calculated.

The construction of a response surface from available data is much less expensive than generating the data for complex problems. Given this relatively inexpensive cost, the MCMC GEMARS algorithm is applied to each collection of data points a large number of times. This is done for two reasons. First, because the MCMC algorithm is a random walk process, two MCMC calculations starting from identical initial conditions will not produce identical final models. Second, the division of data into training and testing points is random, and different divisions of the data will produce different final models. Therefore a large number of GEMARS surfaces are constructed for each available set of data. In this work between 20 and 50 models were constructed for each available collection of data points.

It is possible to overfit the training data. A set of data points that may be accurately modeled by a low-order polynomial can also be accurately modeled as a higher-order polynomial. As the polynomial order increases, even if the testing points are exactly met, the model may become highly oscillatory; this is obviously undesirable if the actual QoI corresponds to a low-order polynomial. To guard against this phenomena the variance of each surface is compared against the median variance of the surfaces. Any surface whose variance differs from the mean variance by a factor of more than 10 is rejected. Although this method of screening the response surfaces is not foolproof, it is a useful indicator of the quality of the surfaces. In general, for the problems considered in this work, fewer than 10% of the converged surfaces were rejected by this screening step. This screening step implicitly implies that the QoIs we seek are relatively smoothly varying across the parameter space.

Given a collection of screened response surface models and some scalar values determined on each model, the means and variances of these values are easily calculated. The values of interest calculated on each surface are the mean and variance of the QoI represented by that surface. This gives rise to the confusing nomenclature of the mean mean, the variance in the mean, the mean variance, and the variance in the variance. The trick here is that the first descriptor refers to the collection of surfaces and the second descriptor refers to the QoI modeled by each surface, that is, we calculate the mean across the surfaces of the mean QoI on each surface, the mean across the surfaces of the variance in the QoI on each surface, the variance across the surfaces of the mean QoI on each surface, and the variance across the surfaces of the variance in the QoI on each surface.

Given a collection of M points in the input space at which the QoI has been calculated, response surfaces can be generated with varying numbers of available data points from 1 to M. The magnitude of M will usually be limited by the computational resources available for radiation transport calculations. By plotting the QoI's mean and variance (and their associated confidence intervals) as calculated by the MCMC GEMARS algorithm against the number of data points used in the algorithm, it is possible to observe convergence of the response surface models. By plotting the data in this fashion it is possible to argue that "enough" radiation transport calculations have been performed to construct accurate, predictive response surfaces.

4.5.5 An Initial Guess for the MCMC GEMARS Algorithm

The MCMC GEMARS algorithm is seeded with an initial guess for the response surface. A one-dimensional, linear spline with knot point $x_d = -4.0$ and positive orientation is created in each dimension. The least-squares problem is solved for this configuration to yield a linear hyperplane model. This initial guess for the surface will be exactly correct if the QoI is a linear function of the parameters of interest, and will always be accepted as a converged solution for a model with only a single data point.

Note that the assumption used to derive the linear propagation of uncertainty equation introduced in Section 4.3, that is, that the sensitivity coefficients are constant over the uncertain interval of the parameters of interest, is true for this linear response surface. Therefore the variance calculated from a set of sensitivity coefficients is equal whether it is calculated with the linear propagation of uncertainty equation or this linear response surface. Note also that a linear surface yields as its mean the QoI which was supplied to construct the surface.

4.5.6 Search Strategy to Avoid Local Minima

The Markov Chain Monte Carlo algorithm outlined above for optimizing a GEMARS model may be prone to failure due to the presence of local minima. In the case of a multidimensional local minima, the error in the model cannot be reduced with only a small perturbation to the model, as any small perturbation will increase the error in the model. This topic is of considerable interest in the machine learning community and is related to the "Multi-Armed Bandit Problem." In general, the multi-armed bandit problem deals with the tradeoff between exploration of an uncertain design space and the exploitation of knowledge already gathered about the space.

Although many complicated algorithms have been proposed and studied for this problem, it is difficult to outperform the extremely straightforward "epsilon-greedy" or ϵ-greedy algorithm [19][32]. The ϵ-greedy algorithm stipulates that at some small rate ϵ sub-optimal choices should be made in order to more fully explore the design space. In the context of this work, then, perturbations to the spline model which increase the model error are accepted at the rate ϵ.

Of course, a local minima is acceptable if the error in or around that minima meets the convergence tolerance. The MCMC GEMARS algorithm written for this work used an ϵ of 0.002, such that poorly performing changes were accepted at a rate of about 1 in 500. Note that $\epsilon = 0$ is a purely greedy algorithm and that as ϵ increases, the model becomes increasingly difficult to converge, as it may take many iterations to move or remove a model-degrading spline.

4.5.7 Error-Informed Selection of Spline Dimension

Splines are birthed in the MCMC GEMARS algorithm with some specified frequency. Instead of randomly choosing the dimension(s) of the parameter space into which the new spline is placed, it is possible to select dimensions to try to decrease the model error as efficiently as possible. The error in the spline model is assessed using Eq. (4.42) evaluated at the training points. Two sources of error are present in this equation: the QoI and the partial derivative of the QoI in each dimension.

The magnitude of the partial derivative term in the error equation and the contribution of each dimension to this term are easily calculated. The relative contribution of each dimension is then readily available and can be expressed as a probability density function (PDF). A cumulative distribution function (CDF) can be constructed for the different dimensions from this PDF. A random number drawn from a uniform distribution between 0 and 1 is used to select the dimension of the new spline using the CDF. This approach preferentially places new splines into the dimensions that most need additional samples without excluding dimensions that exhibit relatively good agreement with the test points.

4.5.8 Distribution of Training and Testing Points

The multigroup cross sections are described by a multivariate Gaussian distribution. The uncorrelated parameters are each unit normal distributions. Statistical

UQ methods require drawing from these normal distributions. Once the surfaces are constructed, the models are sampled at O(1E5) normally-drawn samples. However, during the construction of the response surface, the training and testing points need not be normally distributed across the input space. Ideally a Latin Hypercube Sampling or Stratified Sampling technique would be employed to ensure even sampling of the space, but the required sparseness of the samples prevents this method for most test problems.

Response surfaces were constructed for one of the test problems using both normal and uniform samples of the uncorrelated-parameter space. The mean value of the uncorrelated parameters ($p_i = 0$) was always included as one sample point, as this data point would always be available when performing an analysis of this type; no matter how sparse the sampling of the input space, the problem must be run at the nominal value of the cross sections to calculate the nominal QoI. Uniform samples were drawn from the interval $[-3, 3]$ in the uncorrelated-parameter space. The use of a uniform distribution spreads the samples more evenly across the uncertain parameter space, which better informs model construction.

The ability to sample points at any point in the input space for the construction of the response surface presents a straightforward opportunity for the extension of this algorithm. Given a collection of response surfaces generated with some set of data points, regions of disagreement between the surfaces should be selected for the generation of new cross sections and additional radiation transport simulations. This approach should yield well-converged response surfaces with fewer data points than a model constructed entirely from randomly selected data points.

4.5.9 Model Limitations

There are two limitations inherent in the MCMC GEMARS algorithm due to design decisions. First, the MCMC GEMARS model assumes that the QoI is relatively slowly varying across the input space. This assumption appears to be well grounded in the nuclear data and by experiments performed in the nuclear community. If a QoI such as detector response or the multiplication factor of a nuclear reactor varies rapidly with minor changes in the cross-section data, calculations performed using different sources of cross-section data should show significant disagreement. Similarly, these calculations would not be expected to match experimental data. Because these effects are not widely observed, it seems a justifiable assumption that the QoIs change slowly with varying cross sections.

The second major assumption in this work proved reasonable for the problems considered thus far, but may not be valid for all problems. In this work the number of dimensions spanned by each spline was limited to a small number. The uncorrelated parameters isolated in Section 4.4.4 are uncorrelated in the sense that they represent independent sets of multigroup cross sections. A QoI which is calculated from the product of several cross sections may include terms that are the product of multiple independent parameters. The inclusion of cross terms rapidly increases the dimensionality of the parameter space in which the response surface is calculated.

Consider an input space containing P independent parameters. A model constructed of one-dimensional splines contains P dimensions which must be sampled. A two-dimensional spline is drawn from a space of $P(P-1)$ dimensions; a three-dimensional spline is drawn from $P(P-1)(P-2)$ dimensions. The presence of cross-dimensional terms massively increases the possible combinations of dimensions in which the splines may be sampled.

A small number of cross terms were specified in the algorithm for the problems in this work. This setting proved sufficient for the test cases considered. In general, however, there is no guarantee that the number of required cross terms will be small.

4.5.10 MCMC GEMARS Settings in this Work

Although various combinations of settings were tested on the various test problems examined in the course of this project, the MCMC GEMARS algorithm was applied using linear splines with a maximum of three dimensions per spline. These settings were chosen primarily to minimize the spline-parameter space that is explored. These settings proved sufficient for the slowly-varying QoIs examined in this work.

The ratios of spline birthing, spline movement, and spline killing in the MCMC algorithm were adjusted to discourage model growth except when necessary. To that end, a birth rate of 0.10, a movement rate of 0.40, and a killing rate of 0.50 was chosen. This collection of parameters was observed to allow the number of splines in the model to increase slowly until the model converged.

An error tolerance of 0.05 was specified for the surface convergence criteria for the problems in this work. Again, larger and smaller error tolerances were tested in the model design stage. This setting was found to converge after several tens of thousands of iterations for most problems. At tolerances below 0.01 models often failed to converge after running overnight on test problems.

Together, these settings were used to successfully apply the MCMC GEMARS surface tool to a variety of radiation transport problems.

5. RESULTS

In this section a series of problems are analyzed to test the methods outlined in the three theory sections. These test problems fall into three categories. The first pair of problems is designed to verify that the UQ methodology, algorithms, and software developed in this work are properly implemented. The second set of problems is performed to examine a physical experiment being performed at Texas A&M University. Finally, a third set of problems is related to a nuclear reactor application.

The goal of the UQ analysis performed in this work is to calculate the quantity of interest and variance in the QoI for realistic neutron transport calculations using a gradient-informed response surface. The number of data points required to construct an accurate, predictive response surface is not known a priori. Therefore, for each test problem, the QoI's mean and variance are plotted versus the number of data points available to construct the surface. From these plots the number of data points required for the surfaces to converge to a solution for the QoI and the variance can be determined.

The response surfaces are constructed using an increasing pool of evaluations of the full model in the independent-parameter space. As the response surfaces are quick to generate compared to the cost of the forward and adjoint solves of the transport problem and the inner product evaluations, the surfaces may be constructed as QoI and sensitivity data is slowly generated. As each adjoint sensitivity calculation in the input space is completed the QoI and sensitivity coefficients are appended to the existing data and passed to the surface construction tool. This pool of data is then used to construct a large number of surfaces as outlined in Section 4.5. The first physics calculation is always performed at the mean value of the cross sections, which corresponds to each independent parameter being equal to zero.

The actual value of the QoI's mean and variance is impossible to determine for all but a contrived simple problem. The "correct" values for the statistics of interest can be calculated for inexpensive problems by applying a Monte Carlo sampling method. We begin this section by examining computationally simple problems for which a Monte Carlo analysis can be performed in a reasonable amount of time. The UQ methodology was tested by comparing results generated with the response surfaces to the brute force Monte Carlo results. The Monte Carlo and response surface results are further compared to the variance estimate produced by the linear propagation of uncertainty analysis, Eq. (4.3).

The goal of this research is to produce accurate estimates of the mean and variance of a QoI for a complex problem with relatively few evaluations of the physics code at points in the input space. It is therefore desirable to demonstrate quick convergence to the Monte Carlo results for a verification problem, and quick convergence of the response surface results on the realistic problems. It will also prove interesting to compare the correct value of the variance in the QoI to the value estimated by the linear propagation of uncertainty equation. These results are presented in the following sections.

The linear propagation of uncertainty equation will be referred to as the LPoU equation in this section, as it is invoked often and repeatedly in the following pages.

5.1 Algorithm Verification Problems

Two simple test problems were designed to verify the correctness of various pieces of code written for this work. The first problem was used to test the GEMARS surface algorithm. The second problem was used to test the multivariate Gaussian sampling, change of basis, and dimension reduction algorithms.

5.1.1 GEMARS Surface Test Problem

The first test problem verifies that the C++ GEMARS code produces response surfaces that accurately model the input data. The QoI and sensitivity coefficients are generated by sampling an analytic function and provided to the GEMARS algorithm as input data. The mean and variance of the GEMARS-produced response surface are compared to the mean and variance of the analytic function. A two-dimensional problem was chosen so that the analytic function may be visualized. This function was chosen from [13], where it was referred to as an "additive function." The original problem was given for the domain $x \in (0, 1)$. For this work the analytic function was scaled such that the two dimensions x_1 and x_2 have range $(-5, 5)$ so that samples drawn from the Gaussian distributions of x_1 and x_2 are within this region. The function for the surface $f(x)$ is

$$x_{1,2} \in (-5, 5), \tag{5.1a}$$

$$\hat{x}_{1,2} = \frac{x_{1,2}}{10} + 0.5, \tag{5.1b}$$

$$
\begin{aligned}
f(x) = 1.3356\,[\quad & 1.5(1-\hat{x}_1)+ \\
& \exp(2\hat{x}_1-1)\sin\left((3\pi(\hat{x}_1-0.6)^2\right)+ \\
& \exp(3\hat{x}_2-1.5)\sin\left(4\pi(\hat{x}_2-0.9)^2\right)]\,.
\end{aligned}
\tag{5.1c}
$$

The function is visualized in Figure 5.1.

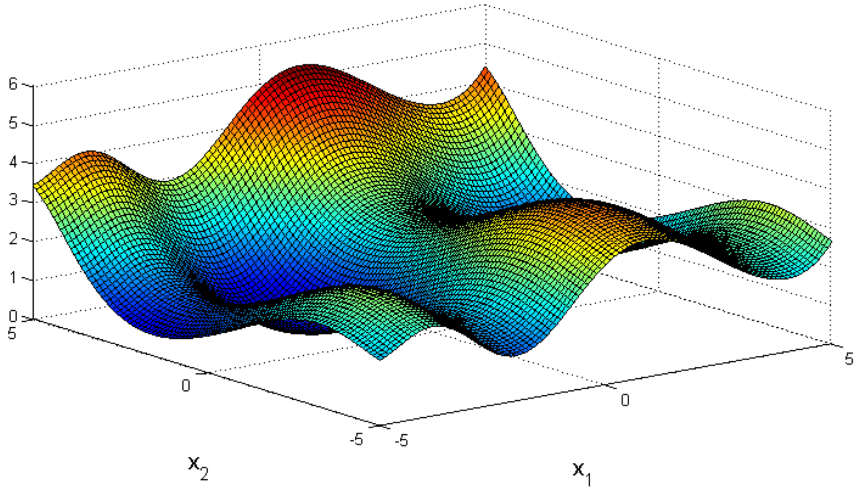

Figure 5.1: Analytic function $f(x)$ for which a response surface is constructed.

There are two main challenges in the surface construction step. First, the model must capture the surface structure as it changes over the problem domain. Second, the complexity of the surface grows rapidly as the number of dimensions increases, especially if the structure in each dimension is complex. This test problem is designed to address the first complexity. The surface examined shows significantly more structure than we expect to see in the QoIs for radiation transport calculations. The remaining test problems demonstrate the effectiveness of the GEMARS tool when applied to multidimensional test functions with simple distributions in each dimension.

The reference value of the function's mean and variance was calculated by a Monte Carlo analysis by drawing from mean zero variance one Gaussian distributions $N(0,1)$ for x_1 and x_2. The QoI and variance in the QoI as a function of the number of Monte Carlo samples are plotted in Figure 5.2. The function's mean and variance calculated with 100,000 (1E5) samples is reported in Table 5.1.

The Monte Carlo analysis indicates that approximately 500 samples of the input space are necessary to converge the values of the QoI and the variance in the QoI. This cost will be compared to the number of samples required to converge these values as calculated with the response surfaces.

The response surface tool constructs a surface over the range $x_i \in (-4,4)$. Approximately 99.994% of samples drawn from unit normal distributions fall within this range. Points in the input space $x_{1,2}$ were chosen for the construction of the response surface for the test function by latin hypercube sampling in the range $x_{1,2} \in (-3.75, 3.75)$, which covers the range over which samples are likely to be drawn but avoids drawing samples very near the boundaries of the response surface tool. The point (0,0) was included as the first sample point. The one hundred sampling points are visualized in Figure 5.3.

Response surfaces were constructed with linear splines, a maximum connectivity of two (the number of dimensions), and a convergence tolerance of 0.05. A total of fifty surfaces were constructed for each collection of samples of the input space. Once converged, the mean and variance of each surface was calculated by sampling the surface at 100,000 points drawn from normal distributions in each variable. The surfaces were screened by calculating the mean value of the fifty variances and rejecting any surface with a variance that differed from the mean variance by a factor of more

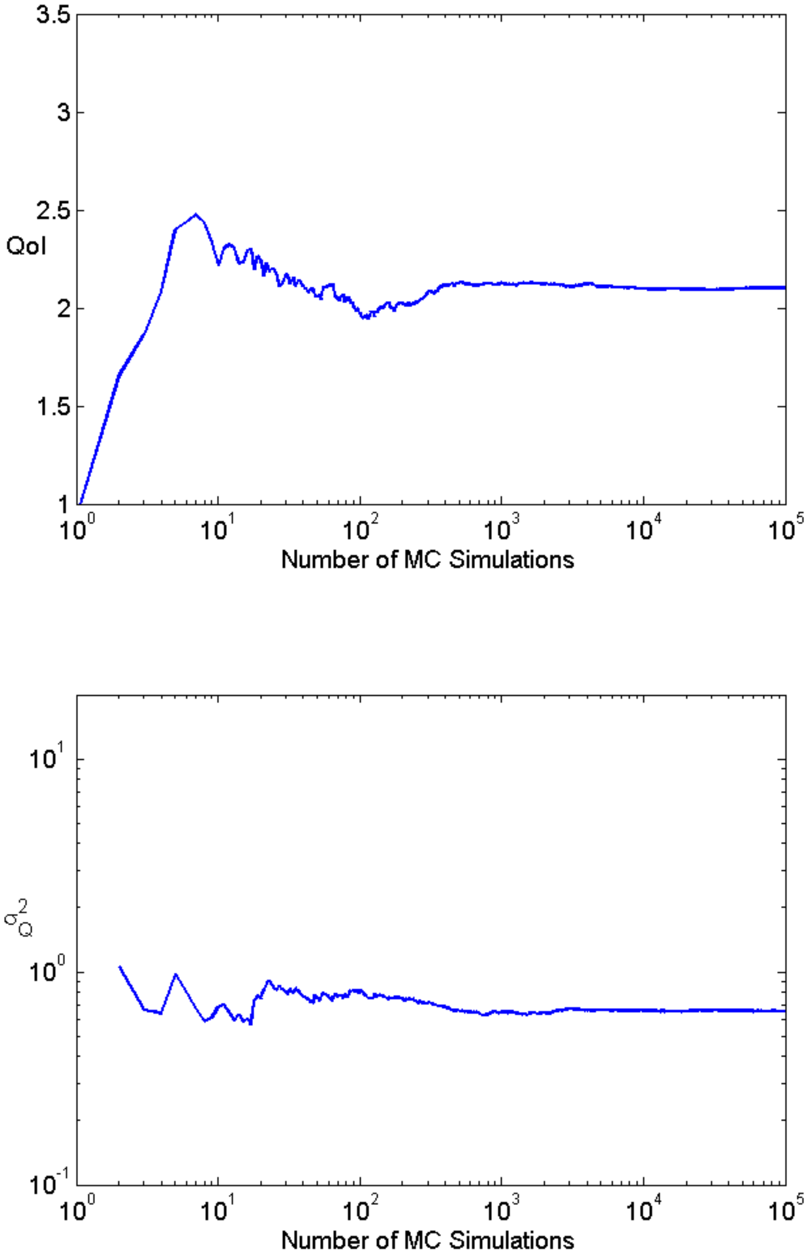

Figure 5.2: Mean and variance for the surface test problem calculated with a Monte Carlo analysis as a function of the number of samples.

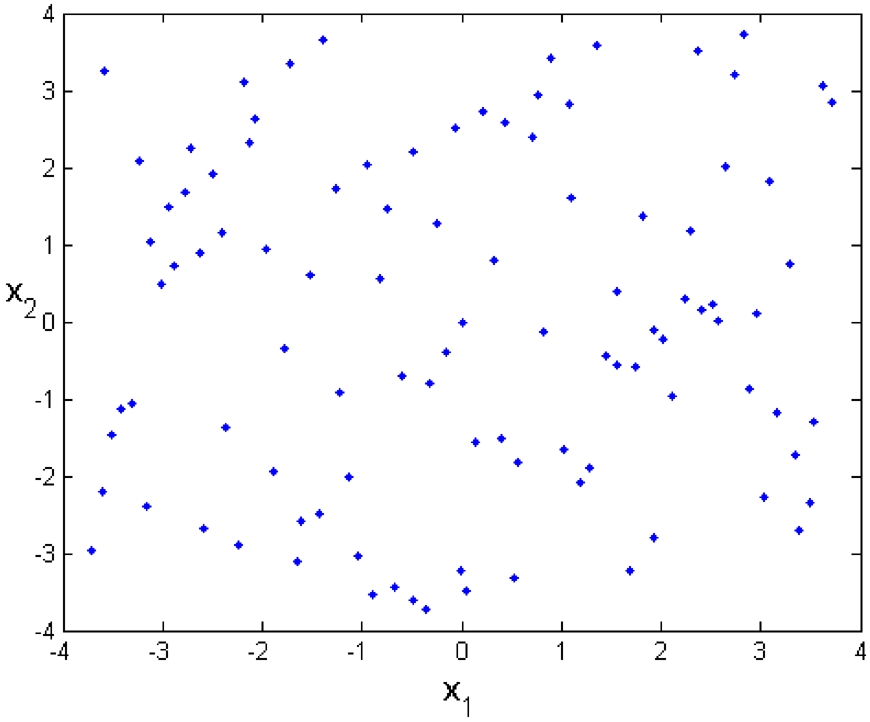

Figure 5.3: One hundred sample points in the two-dimensional problem space, including the mean value of both parameters and ninety nine points chosen with latin hypercube sampling.

than ten. The raw and screened collections of surfaces calculated with each pool of input data were averaged to produce the estimate for the QoI and variance given the available samples.

The number of surfaces that passed the variance-based screening process and the total number of samples of the input space from which the response surface may be constructed is reported in Figure 5.4. In all cases the number of rejected surfaces is very small, indicating that the surfaces very rarely suffer from dramatic overfitting of the data.

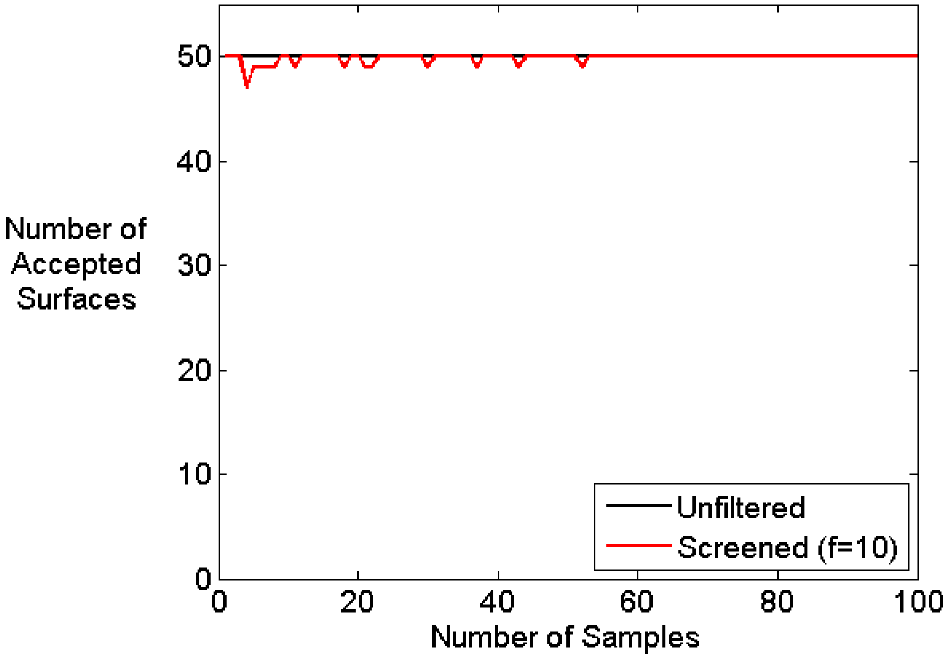

Figure 5.4: Number of response surfaces for the surface test problem which pass the variance screening step with a screening factor of 10.

The mean and variance calculated from the unscreened surfaces is presented as two subplots in Figure 5.5. The QoI and variance are on the y-axis plotted against the number of samples of the input space with which the surface was constructed on the x-axis. The mean and variance calculated directly from the increasing pool of forward evaluations of the input space, as plotted in Figure 5.2, is plotted in black and labeled as the Monte Carlo (MC) Value. The dashed black line shows the mean and variance calculated from 100,000 Monte Carlo samples of the test function. The QoI at the mean value of the input space is plotted in red in the first figure. Finally, the blue dots represent the values generated with the response surfaces.

The mean and variance of the screened surfaces for the surface test problem are presented in Figure 5.6. Based on the low rate of surface rejection visible in Figure 5.4, the blue dots in Figures 5.5 and 5.6 will be identical at most points and differ modestly in only the few locations in which surfaces were rejected.

Let us analyze the results displayed in Figures 5.4, 5.5, and 5.6 in two parts. First we address the screening process used to filter the response surfaces. Comparing the screened and unscreened surface results for the variance, at small numbers of data points the unscreened results contain points that differ from their neighbors by an order of magnitude. Although the surfaces can not be expected to monotonically improve as data points are added to the model, we do not expect the surfaces' quality to drastically degrade with the addition of points. These variations are damped in the screened results.

Note that by Figure 5.4 very few surfaces are rejected when they are screened by their variances and the mean variance across the surfaces. In general, as the number of data points with which the surface is constructed increases the acceptance ratio also increases, although this effect is not pronounced in this example problem. We

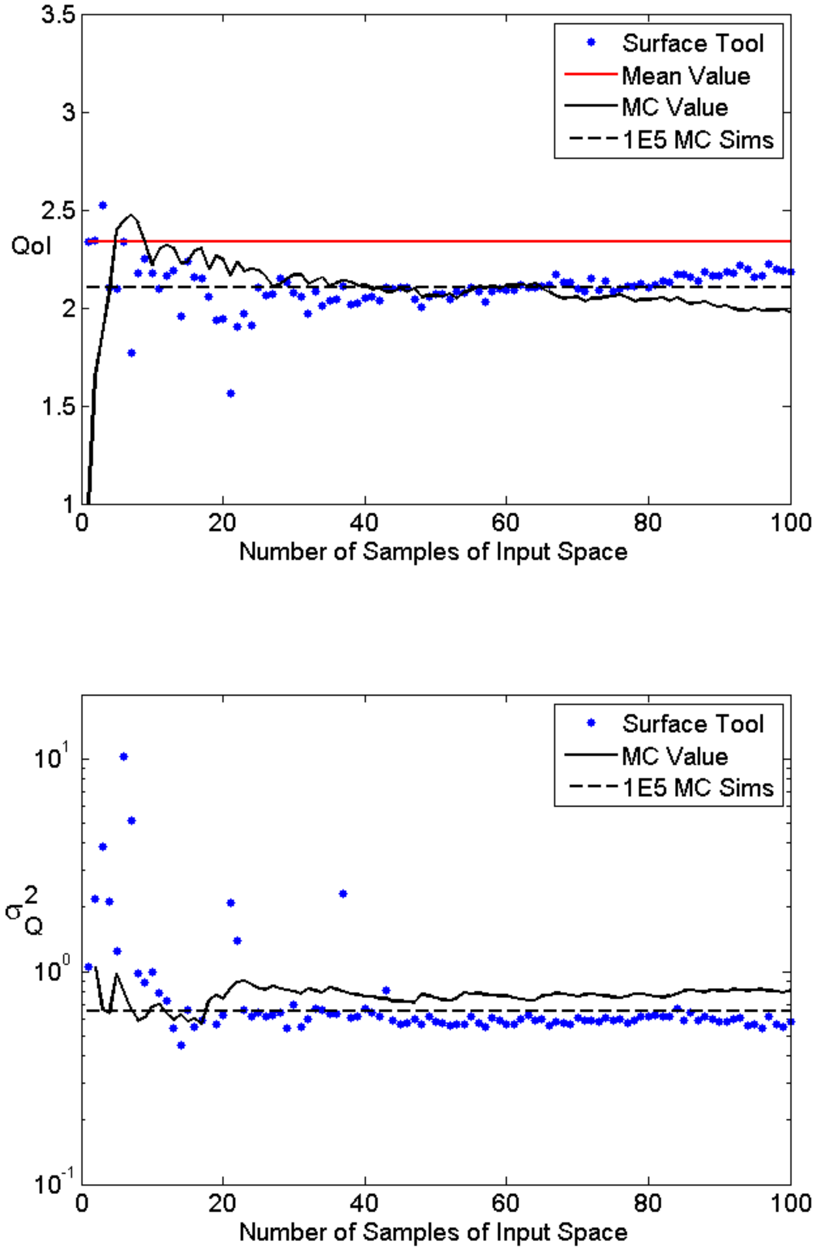

Figure 5.5: Mean and variance for the surface test problem calculated from the unscreened response surfaces.

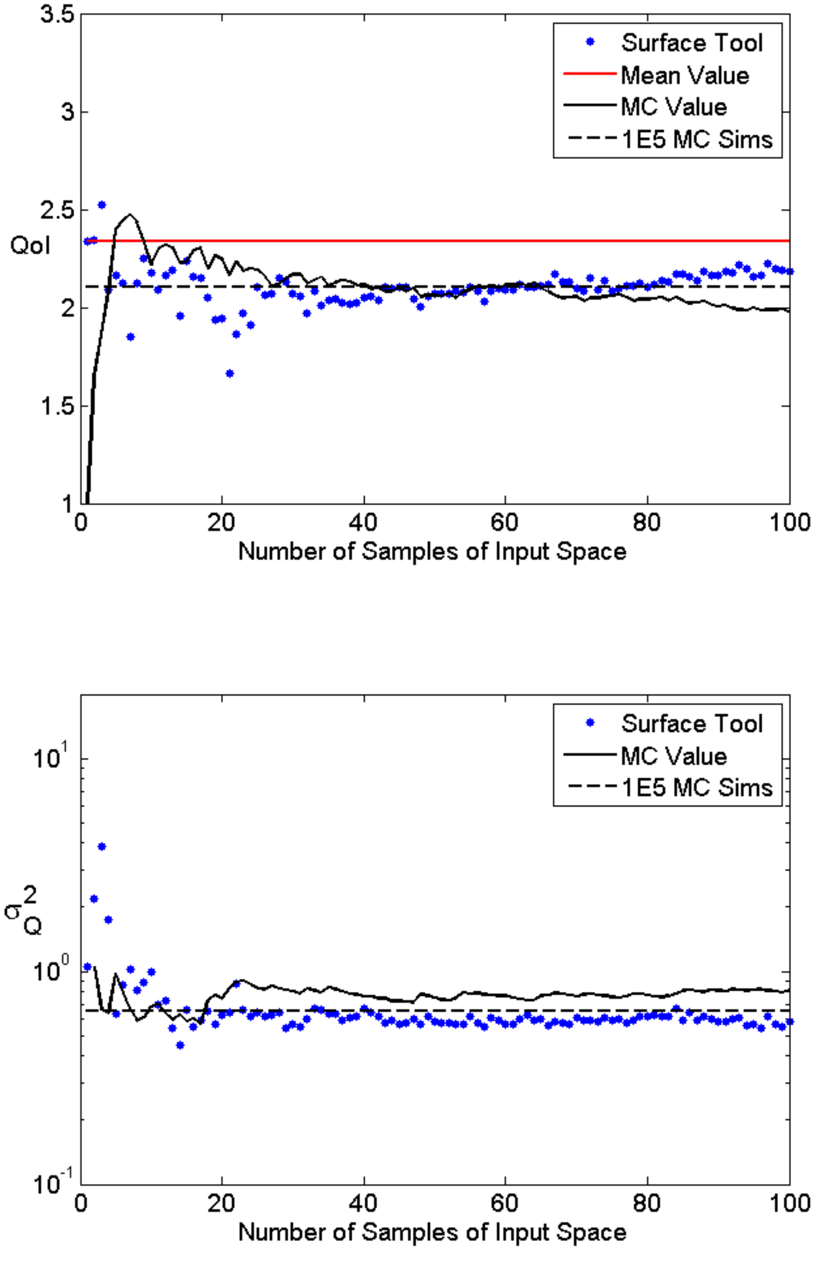

Figure 5.6: Mean and variance for the surface test problem calculated from the screened response surfaces.

will present only the acceptance ratio (Figure 5.4) and the screened results (Figure 5.6) for the remaining test problems.

We next analyze the performance of the screened response surface tool for this test problem. Observing Figure 5.6, the surfaces began to converge to the correct value of the QoI with about thirty samples of the test function. As the number of samples increased, the estimated QoI drifted from slightly below the value predicted by 1E5 Monte Carlo simulations to slightly above the correct value, but did not change dramatically. The variance in the surface was consistently predicted after about about twenty samples of the input space.

Choosing the point at which to cease surface construction can be difficult. In this problem, the variance appears to have converged with about twenty samples of the input space, and the QoI is slowly varying after about thirty samples of the input space. Given the correct value produced by a Monte Carlo analysis, we can see that the surfaces produced with about sixty sample points match the exact solution extremely well, but this data is not available a priori. Values for the QoI and the standard deviation in the QoI are presented in Table 5.1 using various numbers of sample points. Surface results are averaged over small ranges of input points to smooth some of the noise visible in Figure 5.6.

With between thirty and forty sample points, the response surfaces underestimate the QoI by about 3.3%. With between fifty and sixty data points, this underestimation is reduced to 1.3%. Finally, with between ninety and one hundred sample points, the surfaces overestimate the QoI by about 3.9%. Similarly, the estimate for the standard deviation varies from almost exactly correct to about a 10% underestimation. The QoI at the mean value of the independent parameters, in contrast, is about 11%

Method	**QoI**	**Std Dev** (% QoI)
Mean Value of x_1, x_2	2.3406	n/a
1E5 Monte Carlo Calculations	2.1053	38.4
Response Surfaces (30-40)	2.0347	38.6
Response Surfaces (50-60)	2.0777	36.6
Response Surfaces (90-100)	2.1865	34.7
50 Monte Carlo Calculations	2.0602	42.0
60 Monte Carlo Calculations	2.1155	41.4
70 Monte Carlo Calculations	2.0536	42.6

Table 5.1: QoI and variance in the QoI for the surface test problem.

larger than the mean value of the surface. The surfaces are demonstrated to yield significantly better estimates for the QoI that the mean value alone.

The response surfaces require at least thirty samples of the input space to begin to estimate the proper value of the QoI and variance. In addition to these thirty samples, further samples are required to demonstrate that the models have begun to converge. Because each adjoint sensitivity calculation requires both a forward and adjoint solve, this is equivalent in computational cost to requiring at least sixty forward simulations. The estimate for the QoI and variance using fifty, sixty, and seventy Monte Carlo samples of the input space are reported in Table 5.1. The QoI and variance as calculated by the Monte Carlo and response surface methods with a similar computational cost are similar. However, the response surfaces begin to converge with thirty samples of the input space, while the Monte Carlo analysis requires between four and five hundred samples to converge.

The response surface tool began to accurately model this complex test function with a relatively small number of sample points (thirty points). This analysis demonstrates that the response surface tool was correctly implemented and can be applied to the remaining test problems with confidence. We note that for a problem without a known solution, it may be difficult to judge when the surfaces have truly converged to the correct solution. It is therefore highly desirable to perform as many simulations of the physical problem as possible to increase both the accuracy of the response surfaces and build confidence that the surfaces have converged to the true distribution.

5.1.2 Analytic Problem

The second test problem we consider was designed to demonstrate that the machinery developed to sample the multivariate Gaussian distribution and the machinery used to transform between the cross-section space and the independent-parameter space were implemented without mistake. A neutron transport problem was constructed such that it has an analytic solution. The availability of an analytic solution allows several diagnostic tests to be performed on this problem.

This test problem consists of a one-dimensional purely absorbing homogeneous slab with an isotropic unit scalar neutron flux on one face. The QoI is defined as the absorption rate in a region within the slab. The angular flux solution at position x in the slab, angle θ, and initial value $\psi_{inc}(E)$ is given by

$$\psi(x, \theta, E) = \psi_{inc}(E) \exp\left(-\Sigma(E)\frac{x}{\cos(\theta)}\right). \tag{5.2}$$

The QoI is the total absorption rate $\Sigma(E)\phi(x, E)$ in a region of interest defined by the left and right boundaries of the region of interest, x_L and x_R, respectively. This is calculated by integrating the reaction rate over this range,

$$Q = \int_{x_L}^{x_R} \Sigma(E)\phi(x, E)dx. \tag{5.3}$$

We solve this problem in the discretized form in angle and energy. The boundary condition is isotropic,

$$\psi_{inc}(\theta, E) = \psi_{inc}(E). \tag{5.4}$$

The angular flux solution in each quadrature direction d is a function of the cosine of the angle associated with each direction. Defining $\mu_d = \cos(\theta_d)$,

$$\psi_d(x, E) = \psi_{inc}(E) \exp\left(-\Sigma(E)\frac{x}{\mu_d}\right). \tag{5.5}$$

The energy variable is discretized with the multigroup discretization and the incident neutron source is defined to be uniform in energy,

$$\psi_{inc,g} = \psi_{inc}. \tag{5.6}$$

Given this discretization in angle and energy, the angular flux is

$$\psi_{g,d}(x) = \psi_{inc} \exp\left(-\Sigma_g \frac{x}{\mu_d}\right). \tag{5.7}$$

With this expression for the angular flux, the QoI is

$$Q = \sum_d w_d \sum_g \int_{x_L}^{x_R} \Sigma_g \psi_{inc} \exp\left(-\Sigma_g \frac{x}{\mu_d}\right). \tag{5.8}$$

This spatial integral can be evaluated analytically, yielding a final expression for the QoI using the multigroup and discrete-ordinates discretizations,

$$Q = \sum_d w_d \sum_g -\mu_d \psi_{inc} \left(\exp\left(-\Sigma_g \frac{x_R}{\mu_d}\right) - \exp\left(-\Sigma_g \frac{x_L}{\mu_d}\right)\right). \tag{5.9}$$

By deriving an expression for the QoI using the multigroup and discrete ordinates approximations, the only difference between the QoI calculated with this expression and a QoI calculated with PDT for the same problem is the spatial discretization used by PDT. A spatial refinement study was performed to eliminate that source of error. This expression will allow PDT's QoI calculation capability to be tested.

Multigroup cross-section data is generated with NJOY and a ninety-nine group energy structure. The slab is assumed to be composed entirely of boron 10. Only the cross section for the absorption of a neutron and the emission of an alpha particle (n,α) was tracked-all other cross sections are set to zero to make the slab purely absorbing. The covariance matrix contains only data for the (n,α) cross section. Finally, the incoming source is isotropic and uniform in energy and defined such that the scalar flux on the boundary is one. This cross-section data is utilized in two separate calculations.

A Python tool was written to solve solve the analytic expression for the space- and energy-discretized transport equation for the QoI, Eq. (5.9). This code accepts as input the mean value of the radiative capture cross section and the 99-by-99 covariance matrix for the (n,α) cross section. Python's built in multivariate Gaussian sampling tool is used to draw 10,000 samples of the cross-section distribution. The mean value of the QoI and the variance in the QoI are determined by evaluating the analytic expression for the QoI at each of the 10,000 Monte Carlo sample points. Because this calculation of the QoI and variance in the QoI uses the covariance matrix as calculated by NJOY, these statistics allow us to test the multivariate Gaussian sampling code implemented in this work.

Realizations of the cross-section space are generated for PDT using the multivariate Gaussian sampling tool developed in this work. The PDT cross-section files are written using only the (n,α) cross section and excluding all transfer scattering reactions. PDT lacks a one-dimensional geometry capability, so a three-dimensional slab is modeled. The slab is arbitrarily large in the x- and y-directions, effectively making it a one-dimensional slab in the z-direction. Eight Gauss quadrature points are chosen in the polar directions; these directions are used in the Python analytic simulation as well. The atomic density of boron is set to 0.05 atoms per barn-cm. The slab is 11cm in length, with 50 cells in the first 10cm and 10 cells in the final 1cm, which is the region of interest.

One thousand forward PDT simulations are performed. The mean and variance of these 1,000 samples is compared to the mean and variance calculated with the analytic expression implemented with Python's multivariate Gaussian sampling routine.

One hundred forward and adjoint sensitivity calculations are performed in PDT and passed to the response surface tool. The number of surfaces which made it through the variance-based screening process is plotted in Figure 5.7.

The variance-based screening step is necessary to reject surfaces which are badly overfit to the training and testing data. This problem is clearly visible at small numbers of training and testing points. As the amount of data with which the surface is constructed increases, the fraction of surfaces which are rejected decreases rapidly. Due to the stochastic nature of the GEMARS algorithm some fraction of the surfaces may always be expected to suffer from overfitting. Figure 5.7 demonstrates that for large numbers of training and testing points increases the ratio of rejected surfaces decreases until between 95% and 100% of the surfaces are regularly accepted.

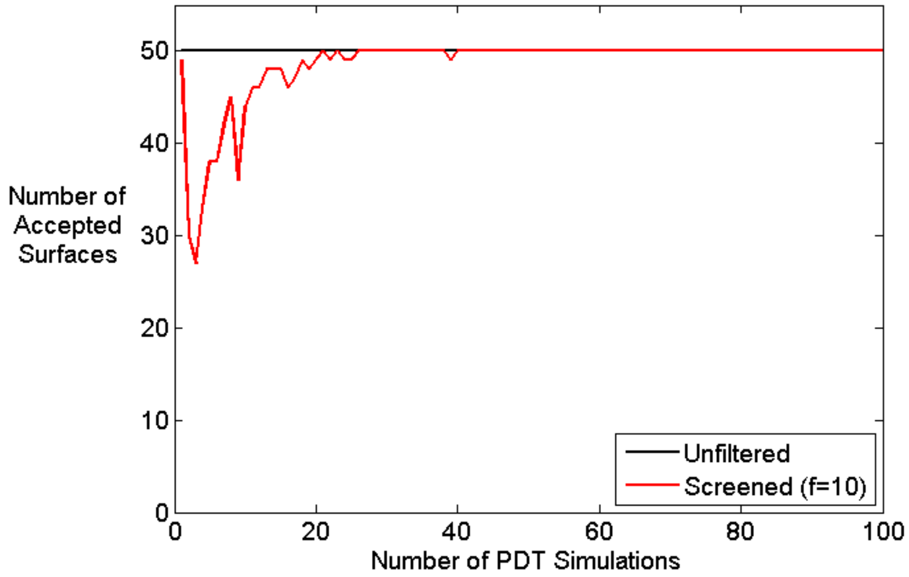

Figure 5.7: Number of response surfaces for the analytic test problem which pass the variance screening step with a screening factor of 10.

The acceptance ratio should not be used to test whether the response surface tool accurately predicts the QoI and variance in the QoI. Rather, the acceptance ratio is a necessary (but not sufficient) condition to indicate that enough training and testing points are available to construct predictive surfaces. The results displayed in Figure 5.7 suggest that at least eleven training and testing points (90%+ acceptance rate) are necessary to construct a family of response surfaces for this problem.

The QoI and variance in the QoI are calculated for this problem using a total of four different methods. The QoI is first calculated at the mean value of the cross sections and the variance in the QoI estimated using the LPoU equation evaluated at these cross sections. The QoI and variance are then calculated with the response surface tool, 100,000 analytic Monte Carlo samples, and 1,000 Monte Carlo forward PDT simulations. These results are plotted in Figure 5.8.

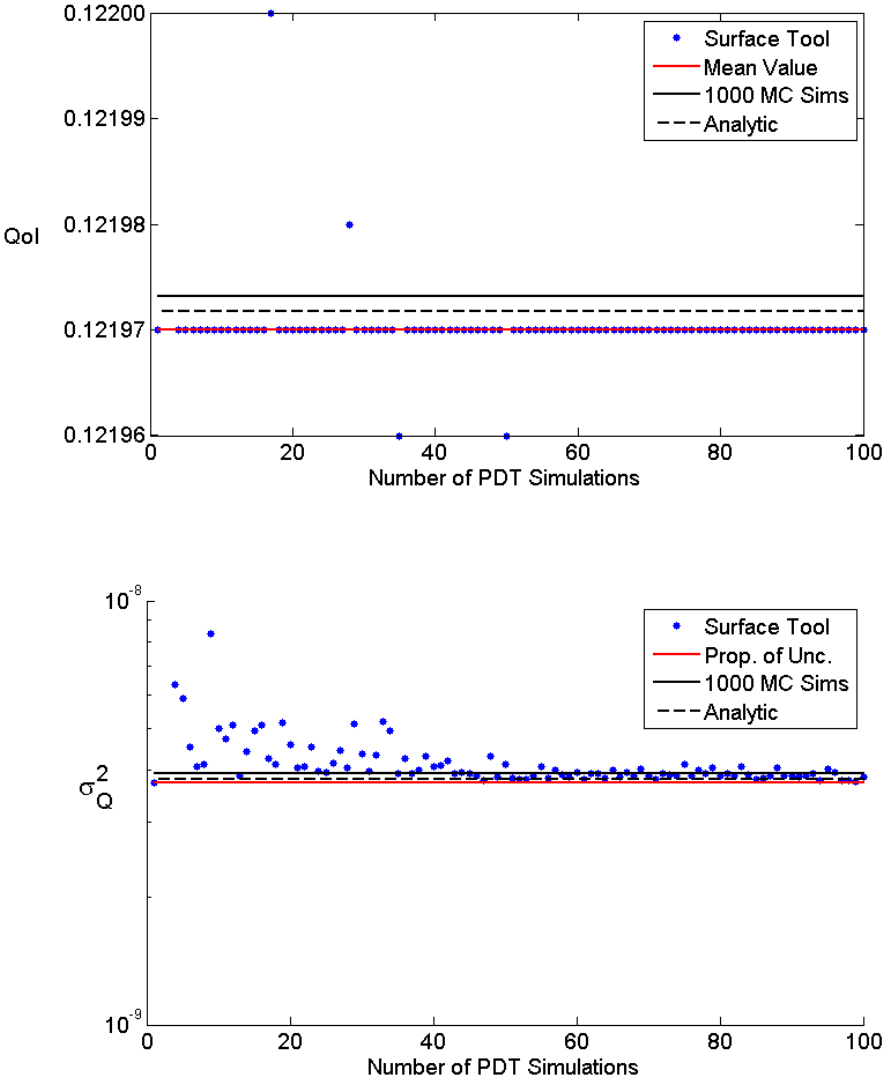

Figure 5.8: Mean and variance for the analytic test problem calculated from the screened response surfaces. Note the extremely fine axis for the QoI plot.

A variety of results are visible in Figure 5.8. First, the 100,000 samples of the analytic equation using Python's built in multivariate Gaussian sampling routine and the 1,000 forward PDT simulations using the sampling tool written in this work generated mean values of the QoI that were nearly identical. Similarly, the variance in the QoI as calculated with the two Monte Carlo methods are in extremely good agreement. The LPoU estimate for the variance was similarly matched the variance very closely. This agreement demonstrates that the multivariate Gaussian sampling tool was implemented correctly.

The agreement between the QoI as calculated with the analytic method and with PDT also confirms that PDT's routine to calculate the QoI was correctly implemented. A bug was actually found and corrected in this PDT routine in the process of running this problem.

The QoI calculated at the mean value of the cross sections matched the two Monte Carlo estimates for the QoI extremely well. The variance estimated with the LPoU equation similarly matched the actual variance very closely.

The response surfaces converge to the QoI calculated with the Monte Carlo methods after only four samples of the input space. The variance predicted by the response surfaces matches the Monte Carlo solutions extremely closely with about fifty samples of the input space. With only ten samples of the input space, the predicted variance was about 5E-9 instead of the correct value of 4E-9, a relatively small difference. The QoI and standard deviation in the QoI as calculated by each method are surmised in Table 5.2. The values generated with the response surfaces are shown with between ten and twenty samples of the input space and again with more than fifty samples of the input space.

Method	**QoI** (interactions/s)	**Std Dev** (% QoI)
Mean Value of XS	0.121970	0.050
1E5 Analytic Solutions	0.121972	0.050
1E3 Forward PDT Solutions	0.121973	0.051
Response Surfaces (10-20 samples)	0.121973	0.056
Response Surfaces (50-60 samples)	0.121969	0.051

Table 5.2: QoI and variance in the QoI for the analytic problem.

The small standard deviation in the QoI is explained by the uncertainty in boron-10's (n,α) cross section. The (n,α) cross sections have a fairly constant standard deviation of just less than 0.084% across most energy groups. A similarly small standard deviation is observed in the QoI.

The response surfaces accurately predicted both the QoI and the standard deviation of the QoI for this analytic problem. This result demonstrates that the code to map both the cross sections and the partial derivatives of the QoI between the cross-section space and the uncorrelated-parameter space is correctly implemented. This result is an early suggestion that the response surfaces may prove useful for the radiation transport problems we will consider next.

The two test problems considered in this section verified that the tools written to construct the response surfaces, sample from the multivariate Gaussian distribution, and map between the cross-section and independent-parameter spaces were properly implemented. The adjoint sensitivity response surface tool also showed initial promise for a simple radiation transport problem. We next examine realistic problems for which the UQ methodology was designed to address.

5.2 CERT Problems

This research project was originally conceived to analyze the uncertainty in detector response for an experiment being performed in the Nuclear Engineering Department at Texas A&M University. The Center for Exascale Radiation Transport (CERT) project, funded by the Predictive Science Academic Alliance Program II (PSAAP-II), seeks to predict the response of a detector placed in and around an complex graphite structure driven by a pulsed neutron source placed at one end of the structure. The experiments presently being performed involve the detailed analysis of individual bricks of graphite to determine the atomic density of any impurities in the brick. These steady-state experiments are titled the "Impurity Model 1" or IM-1 experiments. The experimental setup is pictured in Figure 5.9.

An americium-beryllium (AmBe) neutron source is located in the center and near the bottom of the large cylinder on top of the wooden table. The cylinder is made of high density polyethylene (HDPE) and encased in borated-HDPE. The HDPE thermalizes the neutrons before they enter the graphite brick. The neutron detector rests above the graphite brick encased in a borated aluminum shield. The detector is a BF_3 detector containing boron and fluorine. Aluminum and steel support structure are also present in the problem. A total of 17 isotopes must be included in the IM-1 calculations. These are summarized in Table 5.3.

The CERT problems use a 99-group energy structure with group boundaries chosen to resolve resonances in isotopes of interest to the CERT project. This structure is included in Appendix F.1. A spectrum for the AmBe source neutrons was assumed. P_3 anisotropic scattering was used for these problems.

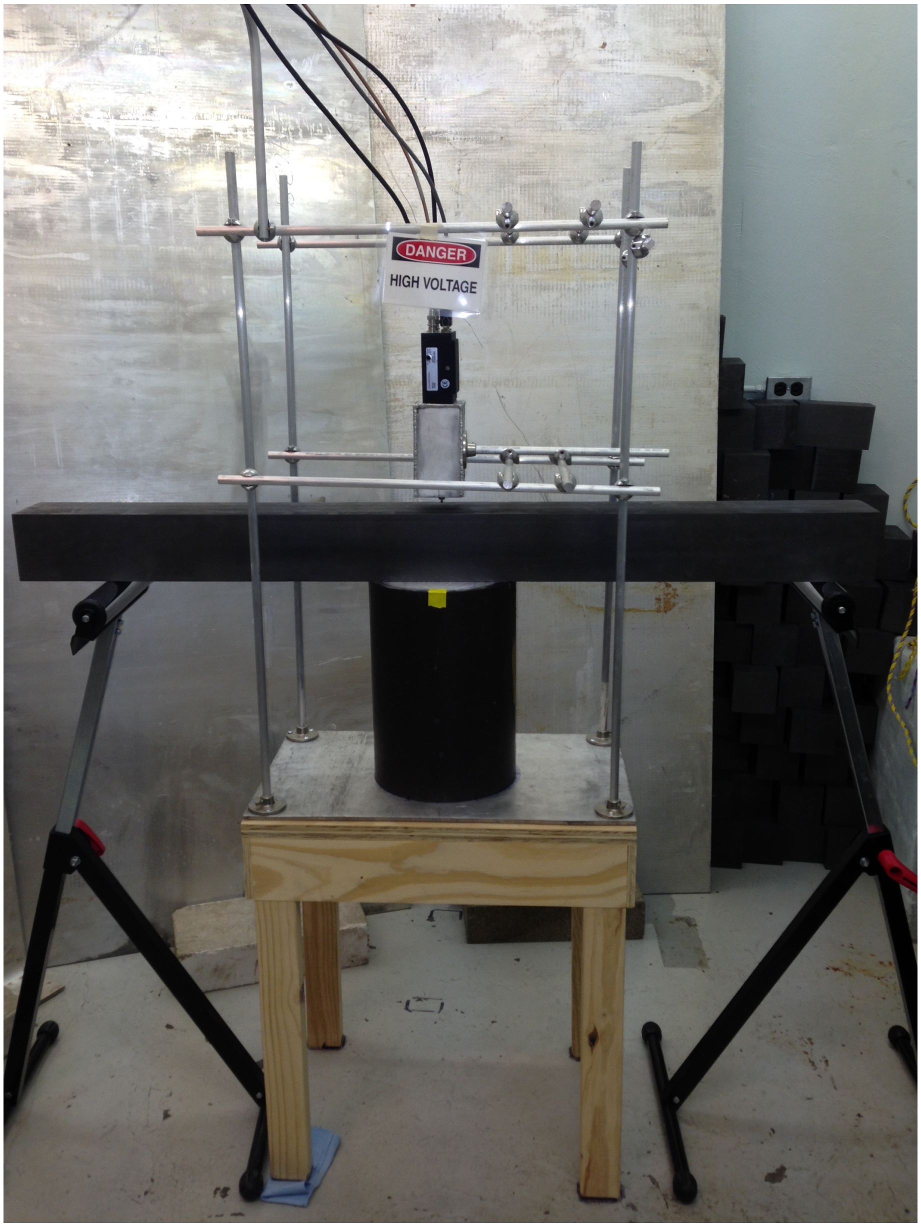

Figure 5.9: IM-1 experimental setup at the Nuclear Science Center.

Isotope	**Location**
Hydrogen-1	Wood
Hydrogen-1 (poly)	HDPE
Beryllium-9	AmBe source
Boron-10	B-HDPE, B-Al, detector
Boron-11	B-HDPE, B-Al, detector
Carbon (natural)	Graphite, HDPE, wood
Carbon (nat) (graphite)	Graphite
Nitrogen-14	Air
Oxygen-16	Air, wood
Fluorine-19	Detector
Aluminum-27	Aluminum structure
Chromium-52	Steel structure
Manganese-55	Steel structure
Iron-56	Steel structure
Nickel-58	Steel structure
Nickel-60	Steel structure
Americium-241	AmBe source

Table 5.3: Isotopes present in the IM-1 experiment.

Two test problems are considered in this section. First, a one-dimensional slab problem was considered to test the UQ machinery on a problem with realistic cross sections, energy structure, and scattering, but with simple geometry. When the UQ machinery was demonstrated to work on this simple problem the full IM-1 geometry was considered.

5.2.1 HDPE and Graphite Slab Problem

A relatively simple problem related the CERT IM-1 problem was designed to test the UQ machinery. The problem geometry consists of a one-dimensional slab containing 10 cm of HDPE, 10 cm of graphite, and 1 cm of boron. The QoI is absorption rate in the boron region. The HDPE and graphite regions contain 10 spatial cells each; the boron region contains five spatial cells. An isotropic incident neutron flux is applied to the boundary of the slab containing HDPE. The boundary source uses the AmBe source spectrum. The problem is solved with 99 energy groups, P_3 anisotropic scattering, 8 polar angles per octant, and PWLD spatial discretization.

Adjoint sensitivity calculations were performed at 100 cross-section realizations generated by drawing samples from uniform distributions of the independent parameters. Forward simulations were performed at 1,000 cross-section realizations generated by drawing samples normally from the input parameters.

This problem contains five isotopes: hydrogen in the polyethylene, boron-10 and boron-11 in the natural boron, and natural carbon as both the polyethylene and the graphite. The total number of non-zero cross sections across these five isotopes and the fourteen neutron interactions tracked in this work is 1,841. The eigenvalue decomposition of the covariance matrices (as discussed in Section 4.4.2) identified 568 independent parameters. The LPoU estimate for the variance (as discussed in Section 4.4.4), evaluated at the 100 realizations of the cross-section space drawn from uniform distributions of the independent parameters, revealed that 99% of the total variance is contributed by 31 of the independent parameters. These results are summarized in Table 5.4.

Isotope	**Number of Dimensions**
Non-Zero Cross Sections	1,841
Independent Parameters	568
Important Independent Parameters	31

Table 5.4: Dimension reduction in the CERT slab problem.

The relative contribution of each isotope to the total variance in the QoI was estimated by applying the LPoU equation to each isotope individually and comparing the variance contributed by each isotope to the total variance contributed by all of the isotopes. These contributions are presented in Table 5.5. Cross sections of hydrogen in the polyethylene are the most significant source of uncertainty in the detector response, followed by carbon in the graphite and boron-10 in the natural boron. Carbon in the HDPE and boron-11 in the natural boron contribute much less uncertainty than the other isotopes.

Isotope	**Contribution to Variance**
Hydrogen-1 (poly)	58.4%
Carbon (nat) (graphite)	19.3%
Boron-10	15.2%
Carbon (nat)	5.0%
Boron-11	2.3%

Table 5.5: Relative contribution of each isotope to uncertainty in the CERT slab problem (values are rounded). The (poly) and (graphite) comments denote thermally-treated bulk materials.

The number of surfaces that passed the variance-based screening step is presented in Figure 5.10.

Figure 5.10 demonstrates that at least eight samples of the input space are required to construct a family of response surfaces where more than 90% of the surfaces pass the variance-based screening step; at least twenty one samples of the input space are required to increase the acceptance ratio to at least 96%. This result states that at least twenty one training and testing points are be required to construct predictive surfaces for this problem.

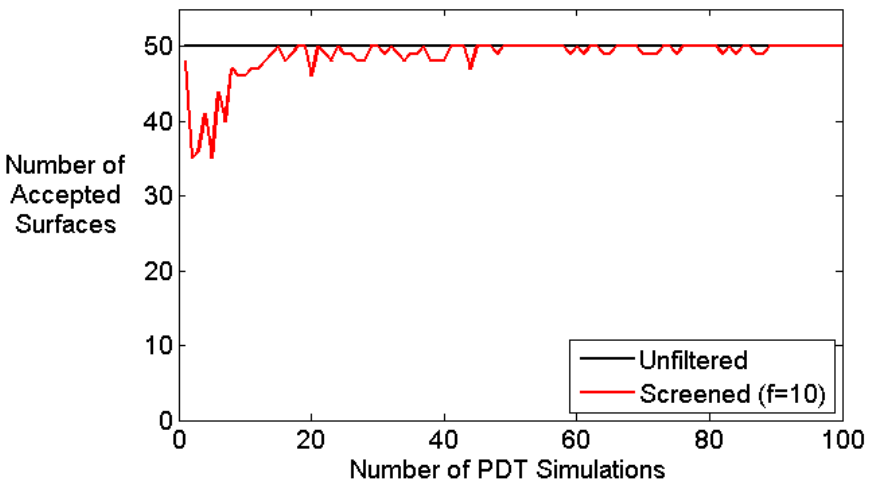

Figure 5.10: Number of response surfaces for the CERT slab problem which pass the variance screening step with a screening factor of 10.

The QoI and variance in the QoI calculated with the screened response surfaces are plotted in Figure 5.11.

The QoI and variance in the QoI as estimated by the response surfaces are reported in Table 5.6.

Method	**QoI** (interactions/s)	**Std Dev** (% QoI)
Mean Value of XS	3.213E-2	0.53
Response Surfaces (40-50 samples)	3.226E-2	0.52
Response Surfaces (90-100 samples)	3.226E-2	0.45
1E3 Forward PDT Simulations	3.214E-2	1.01

Table 5.6: QoI and variance in the QoI for the CERT slab problem.

The values of the QoI calculated at the mean value of the cross sections, from the response surfaces, and from 1,000 forward PDT simulations are all within about 0.4%

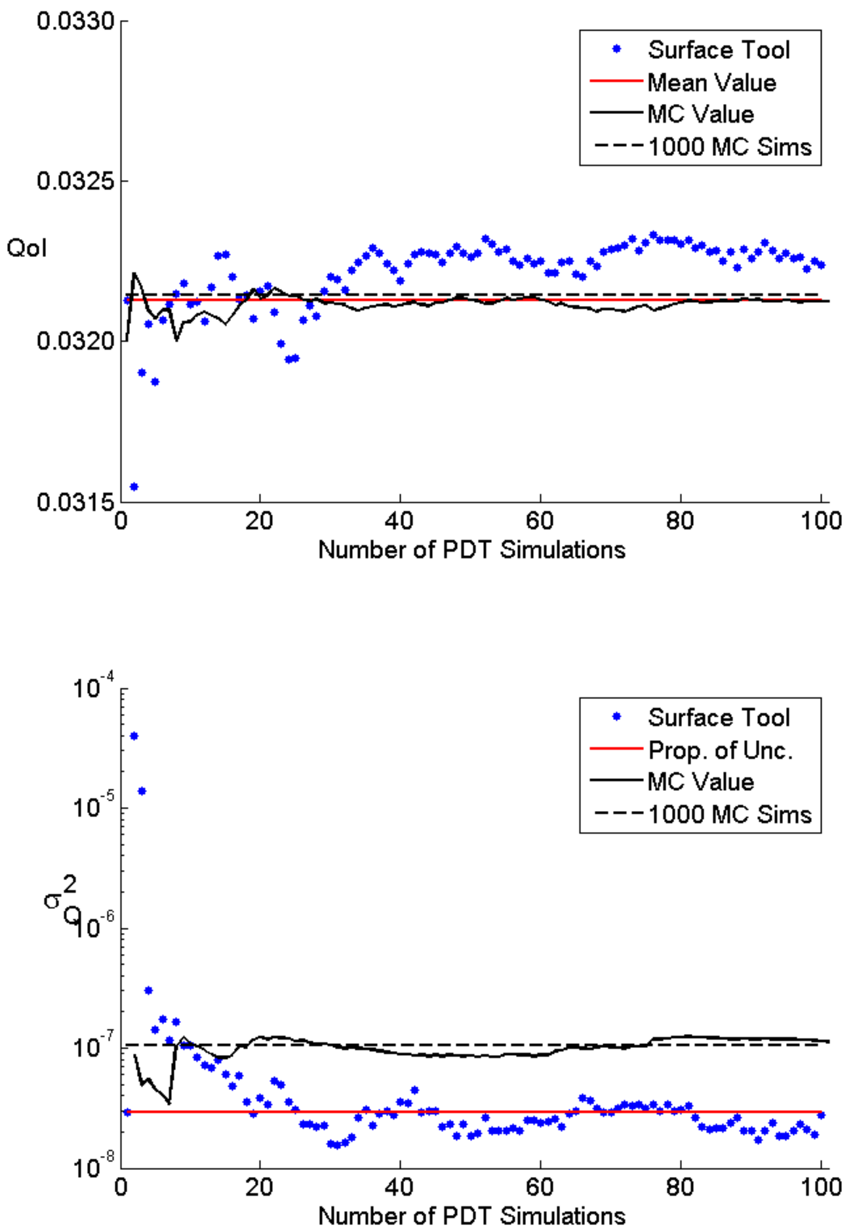

Figure 5.11: Mean and variance for the CERT slab problem calculated from the screened response surfaces.

of one another. The LPoU and response surfaces' estimate for the variance in the QoI are likewise similar, but the estimate of the variance calculated from the 1,000 forward PDT simulations is significantly larger than this value.

Recall from Section 4.3 that the LPoU estimate is accurate if the QoI is linear with respect to each independent parameter. The contribution of each independent parameter to the total variance can be assessed using the LPoU equation. For this problem there are thirty one independent parameters that contribute significantly to the variance in the QoI. Of these parameters the most important parameter is responsible for 53% of the total variance. The eighteen most important parameters together contribute 95% of the total variance. The partial derivatives of these eighteen parameters are plotted versus the parameters' value in Figure 5.12. (The remaining thirteen parameters were included in the response surface construction but not plotted for reasons of space.)

The sensitivity of the QoI to the most important parameter (responsible for more than half of the total variance) and to most of the rest of the important parameters is approximately linear. The third parameter shows significant noise in the sensitivity coefficient, but the trend across the noise appears to be close to linear. The near-linearity of the sensitivity coefficients for most of the important parameters suggests that the LPoU equation will provide an accurate estimation of the variance. The disagreement between the LPoU, response surface, and Monte Carlo estimates for the variance may be due to gross under-sampling of the input space in the Monte Carlo calculation. Given thirty independent parameters, the curse of dimensionality suggests that at least O(2E9) PDT forward calculations should be performed to sample the space.

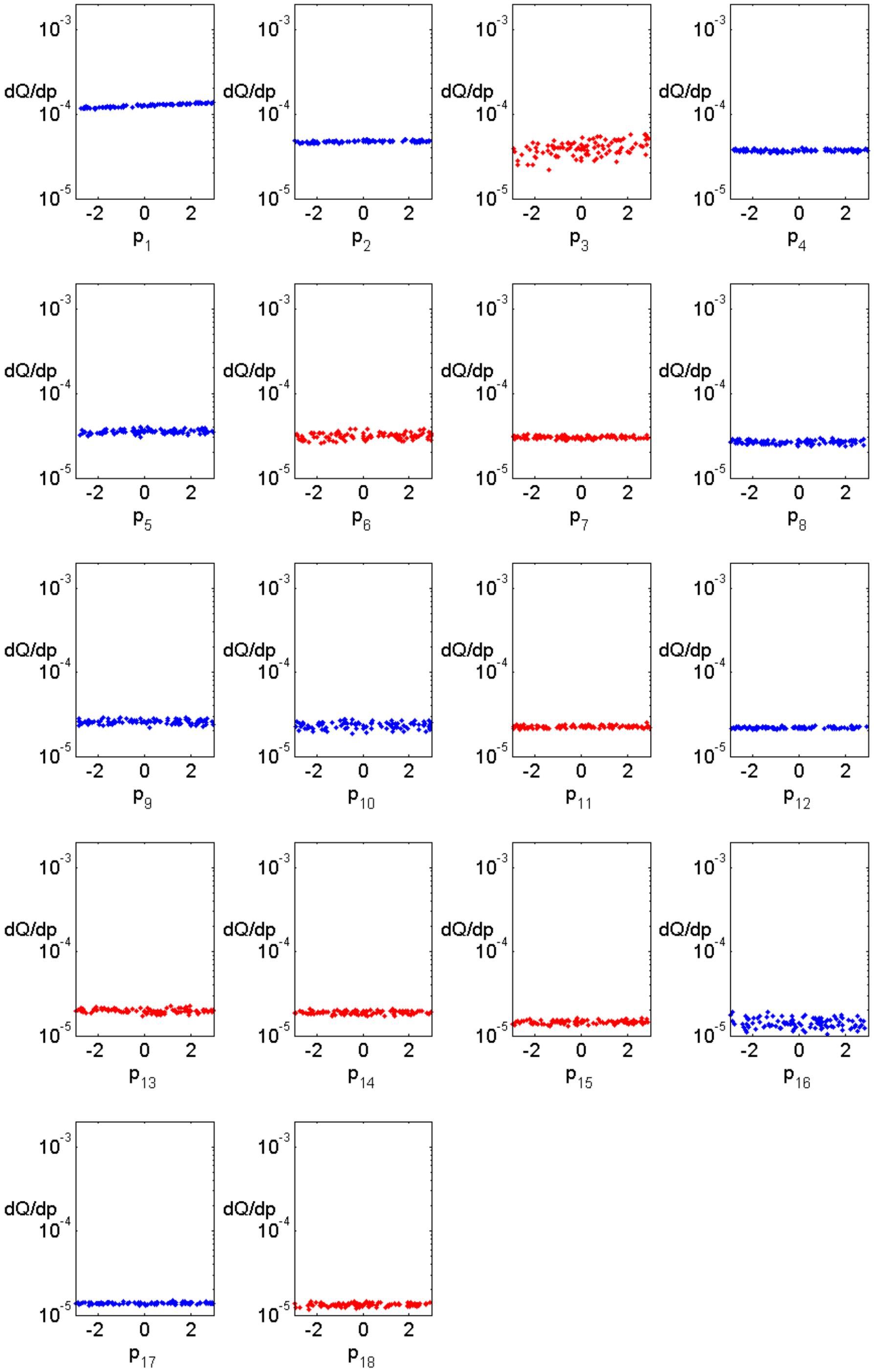

Figure 5.12: Partial derivatives $\frac{dQ}{dp}$ for the eighteen most important parameters for the CERT slab problem. Positive values are in blue, negative values are in red.

Having applied the multivariate Gaussian sampling tool, cross-section generation tool, dimension reduction technology, and response surface generation tool to this CERT slab problem, the full IM-1 problem is examined next.

5.2.2 IM-1 Like Problem

The full IM-1 problem geometry is much more complex than the simple slab geometry used in the previous problem. The neutron source is modeled as a volume source in the cells corresponding to the AmBe source. The QoI is the interaction rate in the cells corresponding to the detector volume. The full IM-1 problem geometry uses on the order of 250,000 brick cells containing a total of 14 different isotopes. The forward problem solved on this mesh takes just over thirty minutes to run on 3,456 cores on the supercomputer Vulcan at Lawrence Livermore National Laboratory. The forward solution, adjoint solution, and sensitivity coefficients for the 14 isotopes' total scattering cross sections required 1.5 hours. The calculation of the inner products for the transfer scattering cross sections, however, could not be completed within Vulcan's 12-hour wall clock limit.

The number of inner products that must be computed is proportional to the number of isotopes times the number of groups (the total cross sections) plus the number of isotopes times the number of groups squared (the transfer matrix). For this problem, calculating the total cross-sections' sensitivities required taking about 1,400 inner products. More than 137,000 inner products were required to calculate the sensitivity coefficients for the cross sections contained in the transfer matrices. These inner products require looping over all spatial cells, angles, and groups; the time required to calculate the sensitivity coefficients is therefore proportional to the number of groups cubed.

In order to reduce the required calculation time for these problems a 49-group energy structure was created from the CERT 99-group structure by collapsing some of the groups in the fast and epithermal neutron energies. This group structure is presented in Appendix F.2. The QoI was calculated with a forward calculation at the mean value of the cross sections generated using the 49- and 99-group energy structures. The QoI calculated with each of these group structures is presented in Table 5.7. The relative difference in the QoI between these two group structures was -0.21%.

Group **Structure**	**QoI** (interactions/s)
CERT 49	4.371278
CERT 99	4.362050

Table 5.7: IM-1 QoI with 49- and 99-group structure.

In order to further reduce the wall clock time required to compute the sensitivity coefficients for the IM-1 problem, a sensitivity study was performed for the 49-group problem to determine the required resolution in space and angle. The brick spatial cells are constructed by specifying each material boundary. Each volume is then divided into equally-sized cells based on the number of cells per division set in the x-, y-, and z-dimensions. This number of cells per division was constant over all volumes, but varied as a parameter in the sensitivity study. The angular quadrature is specified by the number of polar and azimuthal angles per octant.

The settings tested in thirteen IM-1 problems and the time required to complete the adjoint sensitivity calculation are reported in Table 5.8. These calculations were performed on 216 nodes of 16 processors each, for a total of 3,456 processors. For reference, the nominal IM-1 problem used 99 energy groups, 8 polar and 4 azimuthal angles per octant, 3 cells per volume in the x- and y- dimensions, and 4 cells per volume in the z-dimension. The final column of Table 5.8 indicates whether the inner product calculations were completed within the twelve hour wall clock time. PDT outputs the QoI when it finishes the forward simulation and outputs the sensitivity coefficients when it completes the adjoint calculation and the inner products. Therefore the QoI is written to the output file and may be included in the table even if the inner product calculations are not completed.

Test	**Cells in x,y**	**Cells in z**	**Polar Angles**	**Az. Angles**	**Run Time**	**QoI** (interact./s)	**I.P. Calc. Finished?**
A	3	3	1	1	0:05:29	0.1776986	Y
B	3	3	4	2	0:23:26	3.8482598	Y
C	3	3	6	4	1:04:19	4.2284746	Y
D	3	4	6	4	1:24:12	4.2623237	Y
E	3	4	8	4	1:53:15	4.3620495	Y
F	3	6	8	4	2:44:12	4.3586413	Y
G	3	6	8	6	4:00:07	4.3624837	Y
H	3	6	10	6	4:53:19	4.3812571	Y
I	3	6	12	6	6:01:49	4.3710807	Y
J	6	6	8	4	10:55:34	4.3676177	Y
K	6	6	8	6	12:00:00	4.3710992	N
L	6	6	12	4	12:00:00	4.3750985	N
M	6	6	12	6	12:00:00	Error	N

Table 5.8: Discretization study for the 49-group IM-1 problem.

The test using the finest mesh in space and angle, Test M, failed to a memory allocation error. The relative error in Test I compared to Test L, the finest test which ran in forward mode, was approximately 0.1%. The space and angle mesh specified in Test I was chosen for the IM-1 problem. The forward, adjoint, and inner product calculations required about six hours on 216 nodes on Vulcan to complete using this mesh.

The relative contribution of each isotope to the total variance, as calculated from the LPoU at the mean value of the cross sections, is presented in Table 5.9. Hydrogen in the polyethylene contributed the vast majority of problem's variance, and carbon in the graphite contributed most of the rest of the uncertainty. In fact, those two isotopes together were responsible for more than 99.5% of the total variance.

Isotope	**Contribution to Variance**
Hydrogen-1 (poly)	93.3%
Carbon (nat) (graphite)	6.3%
Fe-56	0.002%
Am-241	0.001%
B-10	0.001%

Table 5.9: Relative contribution of each isotope to uncertainty in the IM-1 problem (values are rounded).

One hundred IM-1 problems were performed on Vulcan at uniformly-sampled points in the parameter space. These calculations required about two million CPU hours to complete. From these simulations the important independent parameters were identified. The IM-1 problem had 3,287 non-zero multigroup cross sections, 1,344 independent parameters, and four important parameters, three of which corresponded to hydrogen in the polyethylene and one of which corresponded natural carbon in the graphite. The QoI and sensitivities of the QoI to these four important parameters were used to construct response surfaces. The number of surfaces that passed the variance-based screening step is presented in Figure 5.13. The mean and variance

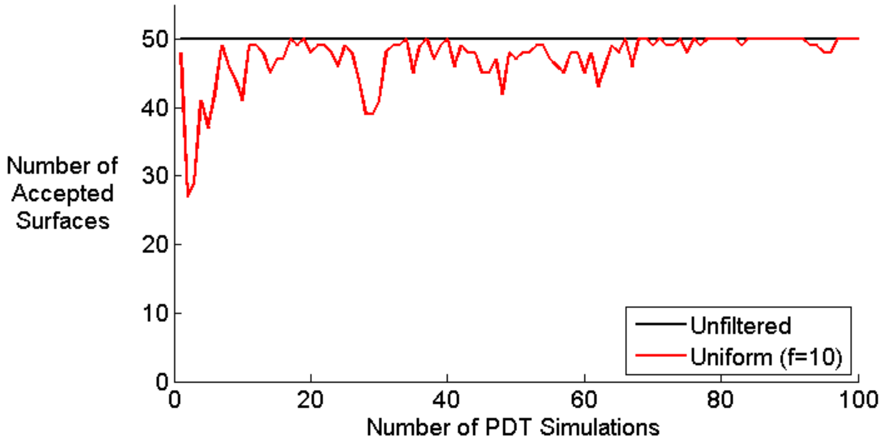

Figure 5.13: Number of response surfaces for the IM-1 problem which pass the variance screening step with a screening factor of 10.

calculated with these surfaces is presented in Figure 5.14.

The QoI at the mean value of the cross sections very closely matched the QoI calculated from the response surfaces with a large number of samples. Similarly, the LPoU equation evaluated at the mean value of the cross sections was in this case an excellent estimate for the variance in the QoI. The QoI and variance in the QoI as estimated by the response surfaces calculated with between 35 and 60 samples (the zone of initial convergence) and with more than 70 samples (the converged zone) are reported in Table 5.10.

Method	**QoI** (interactions/s)	**Std Dev** (% QoI)
Mean Value of XS	4.3559	1.95
Response Surfaces (35-60 samples)	4.3769	1.71
Response Surfaces (70-100 samples)	4.3513	1.98

Table 5.10: QoI and variance in the QoI for the IM-1 problem.

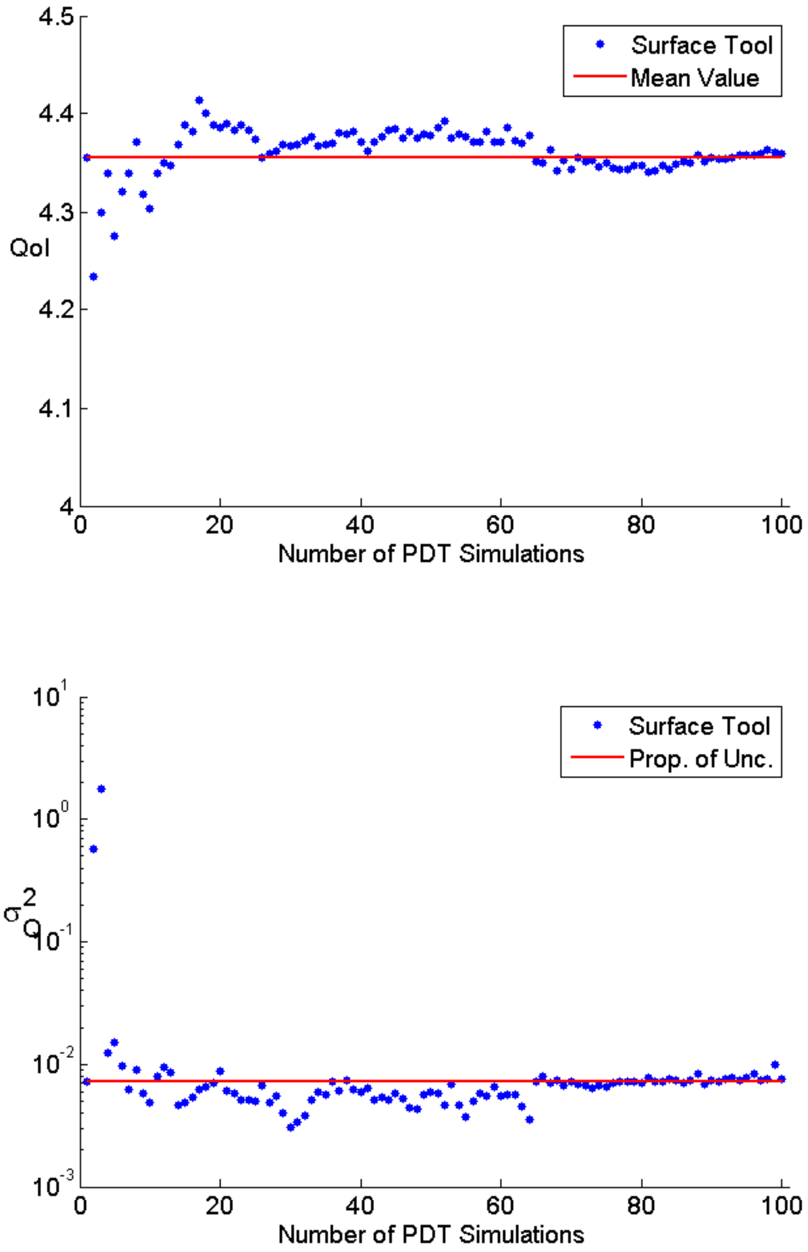

Figure 5.14: Mean and variance for the IM-1 problem calculated from the screened response surfaces.

In addition to considering the contribution to the variance of each isotope we can examine the contribution of each independent parameter to the total variance in the problem. The relative contribution of each of the four most important parameters is tabulated in Table 5.11.

z-Parameter	**Contribution to Variance**
P_1	91.7%
P_2	6.1%
P_3	1.0%
P_4	0.6%

Table 5.11: Relative contribution of the important parameters to uncertainty in the IM-1 problem (values are rounded).

Because this problem contains only four important parameters, the sensitivity of the QoI to each independent parameter over the parameter's range is easily visualized. Unlike the analytic test problem, which had only one important parameter, the QoI will not smoothly vary when plotted against each individual parameter. We do expect the partial derivative of the QoI with respect to each parameter to be slowly varying or constant over the parameter's range. These derivatives are plotted for the four important parameters, in order from most important to least important, in Figure 5.15.

Recall that the variance can be estimated from the partial derivatives of the QoI with respect to the independent parameters by squaring each derivative. It is clear, then, that the first parameter is significantly more important than the other three parameters, as quantified in Table 5.11. The first partial derivative slowly decreases across the parameter space. The second and third parameters appear relatively

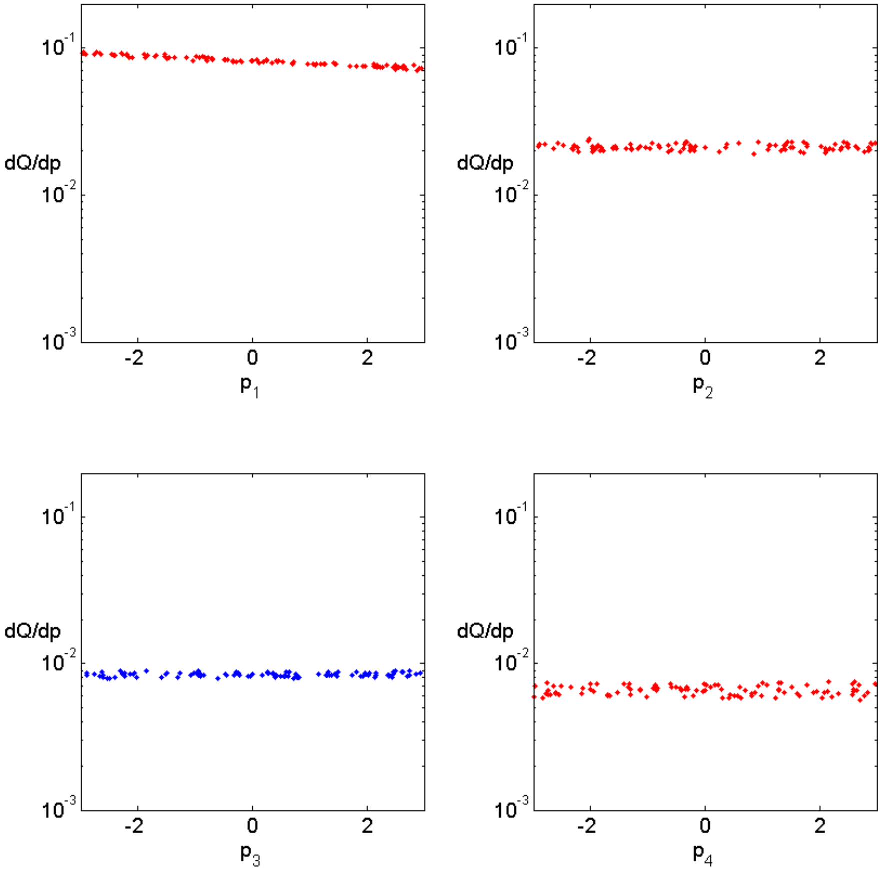

Figure 5.15: Partial derivatives $\frac{dQ}{dp}$ for the four important parameters for the IM-1 problem. Positive values are in blue, negative values are in red.

constant over the space. It is not surprising, therefore, that the LPoU equation yields a standard deviation very similar to the value calculated with the response surfaces.

Analysis of the transformation matrices used to map the uncorrelated z-parameters to the multigroup cross sections reveals the cross sections which correspond to each important parameter of interest. The Jacobian matrix used to map between the uncorrelated-parameter space and the cross-section space was defined earlier in Eq. (4.16), repeated here for convenience,

$$\overline{\overline{\frac{dx}{dz}}} = \overline{\overline{E_x}}\,\overline{\overline{\Lambda}}_x^{1/2}.$$

The rows of the Jacobian matrix correspond to multigroup cross sections and the columns of the matrix correspond to independent parameters. By plotting the columns of the Jacobian matrix corresponding to the four parameters of interest to the IM-1 problem, the cross sections of interest to the problem may be visualized. The cross sections corresponding to the four parameters of interest are visualized Figure 5.16.

The Jacobian matrix is used to map both the cross sections and the sensitivity coefficients between the independent z-parameters and the multigroup cross sections. The cross-section mapping involves the addition of the mean values of the cross sections to the Jacobian matrix times the z-parameters. In the case of the sensitivity coefficients, the mapping depends upon both the Jacobian matrix and the magnitude of the cross-sections' sensitivity coefficients. This means that the values plotted in Figure 5.16 are interesting in terms of position rather than in terms of magnitude.

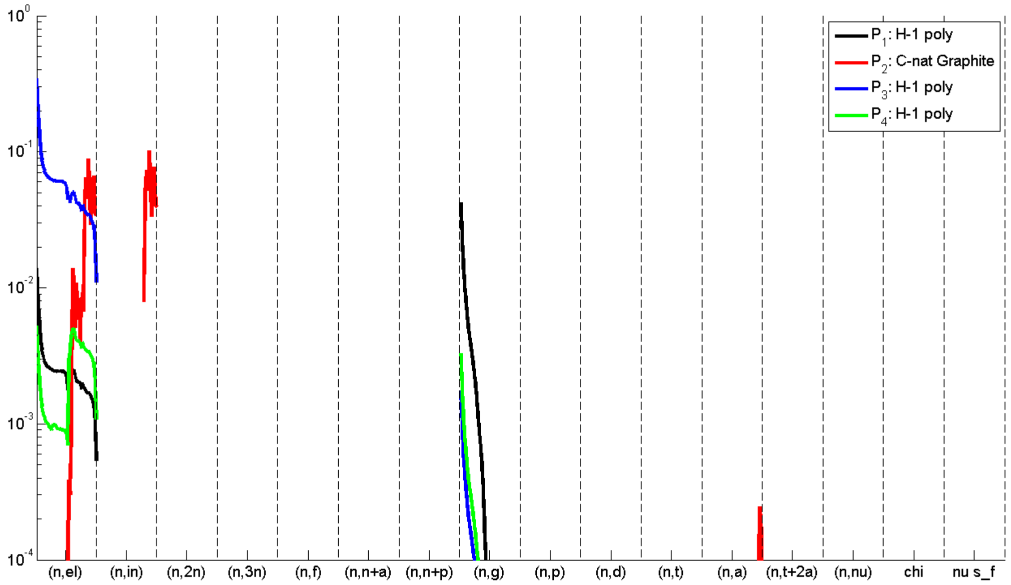

Figure 5.16: Visualization of the columns of the Jacobian matrix $\frac{dx}{dz}$ corresponding to the four parameters of interest to the IM-1 problem. Absolute values of the matrix entries are plotted.

The four important independent parameters for the IM-1 problem correspond to the elastic, inelastic, and radiative capture cross sections of hydrogen in the polyethylene and carbon in the graphite, and perhaps upon the (n,α) reaction in carbon in graphite. The most important parameter, responsible for more than 91% of the total variance, corresponds to elastic scattering and radiative capture in hydrogen in the HDPE. Polyethylene may be removed from the IM-1 problem (or a related future problem may be designed without HDPE) to reduce the uncertainty in the detector response due to the cross-section data.

5.3 Nuclear Reactor Problems

The final test problems examined in this work are related to the analysis of nuclear reactors. As with the IM-1 results, a relatively easy problem and a difficult problem are presented. The first is an infinite medium of a single isotope. The second problem models an infinite lattice of pins of nuclear fuel.

Reactivity was chosen as the QoI for these problems. A system's reactivity ρ is calculated from the system's multiplication factor k as

$$\rho = \frac{k-1}{k}. \tag{5.10}$$

A positive reactivity indicates a system with a multiplication factor greater than one. A negative reactivity indicates a system that is subcritical. An infinite lattice of pin cells containing fresh fuel should have a positive reactivity.

Adjoint sensitivity calculations are complicated by self-shielding effects due to the strong resonances present in uranium. As discussed in detail in Section 3.8, two factors contribute to this complication. First, the sensitivity coefficients calculated with the adjoint sensitivity method are the so-called explicit sensitivity coefficients. PDT cannot calculate the implicit sensitivity coefficients. Second, realizations of the multigroup cross section drawn from the multivariate Gaussian distribution do not account for the perturbations to the weighting spectrum that should be associated with each realization.

The first reactor problem analyzed in this section is designed to circumvent PDT's inability to calculate the implicit sensitivity coefficients. The second reactor problem neglects this shortcoming and applies the response surface tool using the explicit portion of the sensitivity coefficients. The issue of generating weighting spectra consistent with each realization of the cross-section space is an area of ongoing research and was not addressed in these problems.

5.3.1 Infinite Medium U-235 Problem

The first nuclear reactor-related problem considered in this work was designed to avoid the difficulties associated with calculating sensitivity coefficients for problems with strong resonance self-shielding. The results calculated for this problem may be compared against a similar problem analyzed in [20] and compared against results calculated with TSUNAMI-3D by [25].

5.3.1.1 Infinite Medium U-235 Problem Results

An infinite medium of uranium 235 was modeled in PDT using an eighty group energy structure optimized for fast systems. This structure is labeled the “lanl 80-group structure” and corresponds to the NJOY default group option 13. A weighting spectrum containing a fission spectrum, 1/E slowing down spectrum, and thermal Maxwellian (default option 4 in NJOY) was applied to generate the multigroup cross sections. The uranium was assumed to be 500K and of nominal density.

The infinite medium homogeneous problem is isotropic in angle and uniform in space and therefore requires very limited discretization in these dimensions. This problem is a slowing down problem requiring fine discretization in only the energy variable. A two-dimensional problem containing a single spatial cell, one polar and one azimuthal angle per octant, and P_0 (isotropic) scattering was specified.

This spectrum of the problem’s solution is very hard. Fission neutrons are produced in the fast portion of the spectrum. With no leakage and very little energy lost per scattering event due to the mass of the uranium nucleus, most neutrons are absorbed before they reach the resolved and unresolved resonance region. The scalar

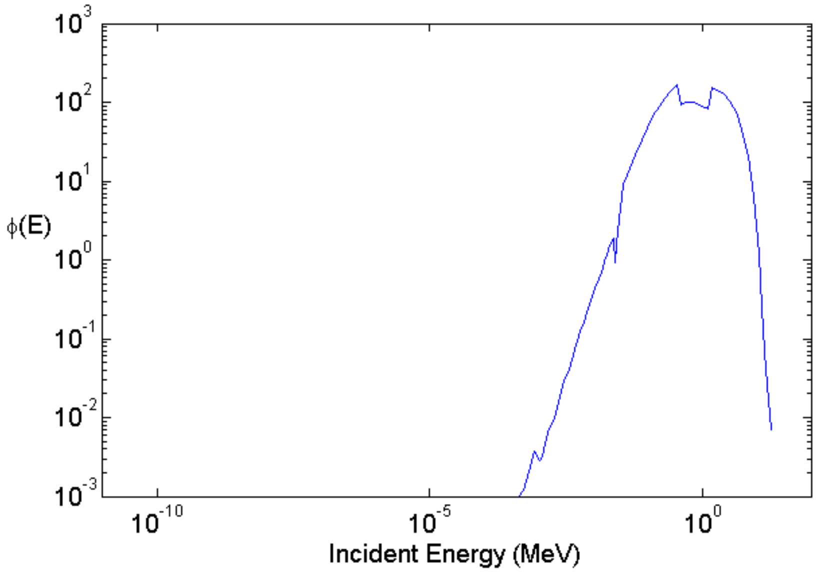

Figure 5.17: Scalar flux solution of the infinite medium U-235 problem plotted against energy.

flux solution of the infinite medium U-235 problem is plotted against energy in Figure 5.17. Uranium 235's total cross section, originally plotted in Section 2 as Figure 2.1, is repeated here as Figure 5.18 for proximity to Figure 5.17. Note that the scalar flux solution at the highest-energy portion of the unresolved resonance region is about two orders of magnitude smaller than the maximum scalar flux in the fast region.

Decomposition of the covariance matrix revealed 261 independent parameters. Six of the 261 parameters were determined to be important in the calculation of the variance. Of the six important parameters, the first was responsible for 75.9% of the total variance, the second parameter 16.0%, and the third parameter 5.4%. The fourth, fifth, and sixth parameters together contributed 1.8% of the total variance. The partial derivatives of the QoI with respect to these six parameters are plotted in Figure 5.19.

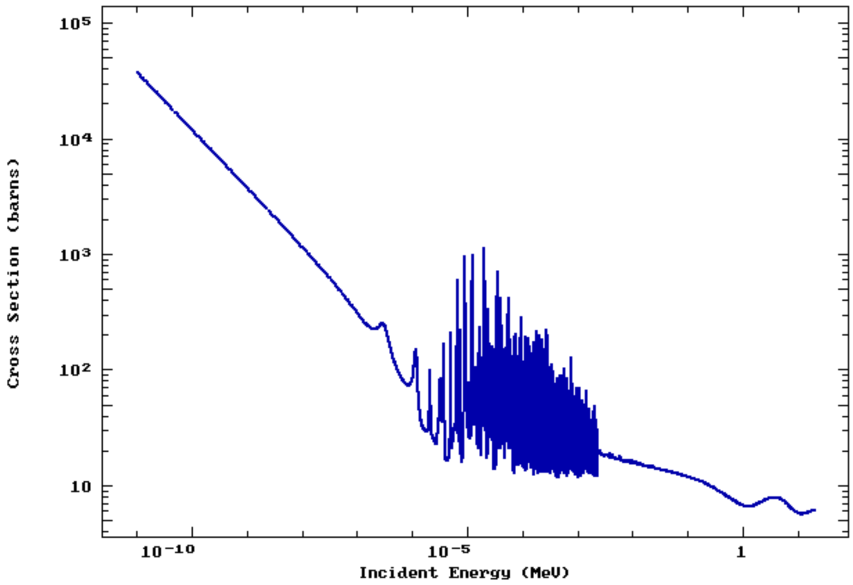

Figure 5.18: Total cross section for U-235 [34].

The partial derivatives of the six important parameters are fairly flat, suggesting that the LPoU estimate for the variance will be accurate. The two most important parameters have noisy partial derivatives, which is challenging for the response surface model, as it may encourage over fitting of the data. The number of surfaces that passed the variance-based screening step is presented in Figure 5.20. The mean and variance calculated with these surfaces is presented in Figure 5.21.

The reactivity QoI is accurately estimated by the response surfaces with only three points in the input space. The estimate for the variance in the QoI is within a factor of two of the converged solution with only four samples of the input space, and converges to an answer very close to the LPoU estimate with about twelve samples of the input space. The QoI and variance in the QoI as calculated with the LPoU equation evaluated at the mean value of the cross sections and as calculated with the response surfaces are presented in Table 5.12. The multiplication factor associated

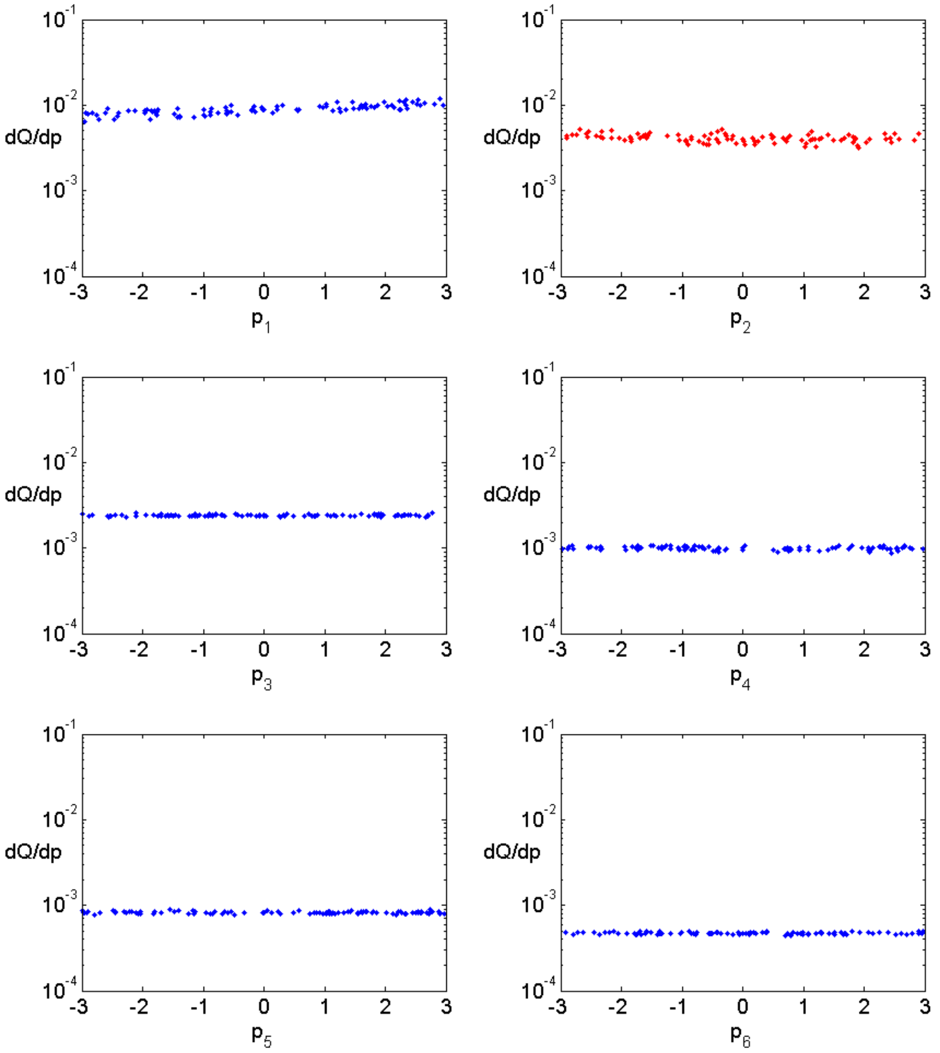

Figure 5.19: Partial derivatives $\frac{dQ}{dp}$ for the six important parameters for the infinite U-235 problem. Positive values are in blue, negative values are in red.

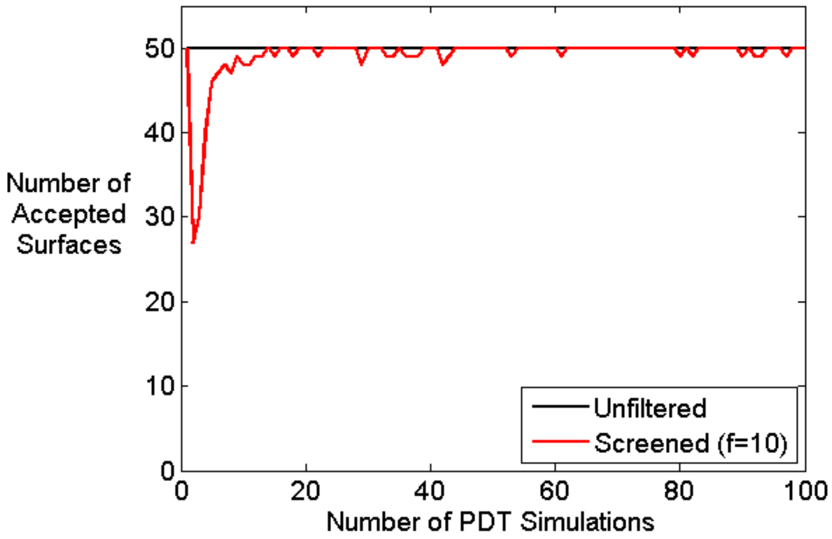

Figure 5.20: Number of response surfaces for the infinite U-235 problem which pass the variance screening step with a screening factor of 10.

with the QoI and the standard deviation in the multiplication factor are included in this table. The standard deviation in the multiplication factor is the average standard deviation calculated with $\rho + \sigma_\rho$ and $\rho - \sigma_\rho$.

Method	**QoI** (ρ)	**Std Dev** (% ρ)	**k**	**Std Dev** (% k)
Mean Value of XS	0.5607	1.79	2.2763	2.29
Response Surfaces (12-20)	0.5622	1.88	2.2841	2.42
Response Surfaces (20-100)	0.5607	1.91	2.2763	2.44

Table 5.12: QoI and variance in the QoI for the infinite medium U-235 problem.

The response surfaces converged to a solution for the QoI and variance in the QoI for this problem with a small number of samples of the input space (about 12 samples). Based upon the sensitivity coefficients calculated for the six important parameters

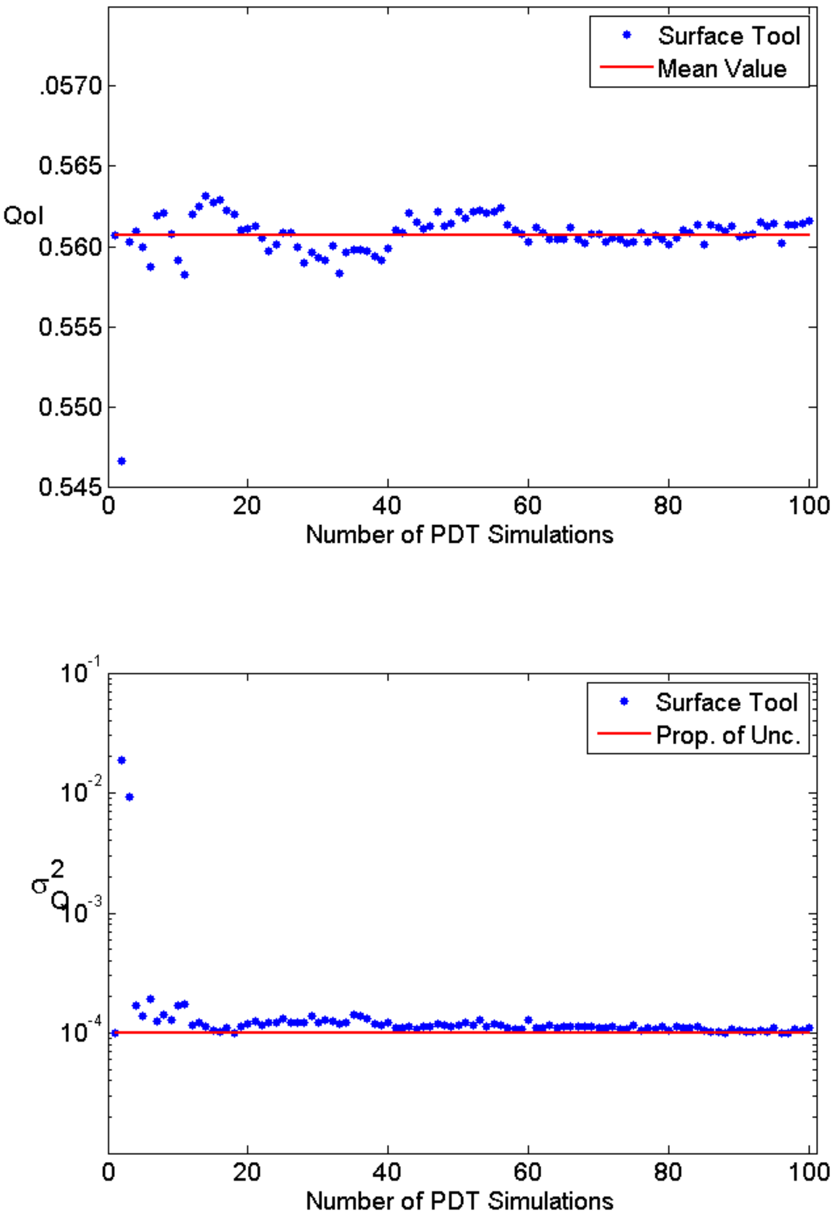

Figure 5.21: Mean and variance for the infinite U-235 problem calculated from the screened response surfaces.

of interest, the LPoU estimate for the variance was expected to be very accurate, and the response surfaces converged to that value almost exactly.

5.3.2 Comparison of Infinite Medium U-235 Problem Results with a Published Result for Godiva

Leal et al. performed a first-order sensitivity analysis for Godiva, a sphere of nearly pure, highly-enriched U-235, in [20]. The physics of this problem are very similar to the infinite medium pure uranium-235 problem except that in Godiva neutrons can escape by leakage from the system. Leal et al. used the SCALE code's KENO tool, a Monte Carlo neutron transport code, to calculate the multiplication factor in the system and the SCALE code's TSUNAMI-3D tool to calculate the total sensitivity coefficients for the uranium-235 multigroup cross sections.

Leal et al. report a multiplication factor $k = 0.9956 \pm 0.0005$ and a standard deviation in the multiplication factor due to uncertainty in the U-235 data of 1.1827 ± 0.0026 percent. This standard deviation is smaller than the uncertainty calculated for the infinite medium U-235 problem, but this may be explained by neutrons leaking from the surface of Godiva. Neutrons undergo many more collisions between birth and death in the infinite medium problem than in the finite volume problem, increasing the sensitivity of the multiplication factor to the cross sections.

5.3.2.1 Comparison of PDT and TSUNAMI-3D Results for the Infinite Medium U-235 Problem

Christopher Perfetti at Oak Ridge National Laboratory ran the infinite medium U-235 problem with TSUNAMI-3D to compare the results generated with TSUNAMI and SCALE to the results generated with PDT [25]. Perfetti ran the problem with both continuous energy TSUNAMI-3D and with multigroup TSUNAMI-3D.

The continuous energy TSUNAMI-3D calculation yielded a multiplication factor of 2.27744 ± 0.00076 and a standard deviation of 2.362% ± 0.0022. The multigroup TSUNAMI-3D calculation also produced an uncertainty in the multiplication factor of 2.362%. These values show excellent agreement with the multiplication factor and its standard deviation as calculated at the mean value of the cross sections by the LPoU equation ($k = 2.2763$, $\sigma_k = 2.29\%$) and with the response surface method (k between 2.2763 and 2.2841, σ_k between 2.42% and 2.44%, respectively) presented in Table 5.12.

The multigroup TSUNAMI-3D calculation reported that the implicit sensitivity coefficients were zero in all groups. The infinite medium problem was expected to have very small implicit sensitivity coefficients as neutrons are born at high energy and are mostly absorbed before reaching the unresolved and resolved resonance regions. These results confirm that the explicit sensitivity coefficients calculated by PDT for this problem are exactly equal to the total sensitivity coefficients.

The excellent agreement between the PDT and TSUNAMI-3D results for this problem strongly indicate that the UQ methodology utilized in this work was implemented properly and yields accurate estimates of the QoI and variance in the QoI for problems without strong self-shielding effects.

5.3.3 69-Group Pin Cell Calculation

The final test problem presented in this work models an infinite grid of nuclear fuel pins in water. A 69-group energy structure designed for light water reactors was used for this problem. The group structure corresponds NJOY's "epri-cpm" group structure, which is option 9. A weighting function was calculated for this fuel pin by Andrew Till, [31], another student in the computational methods research group at Texas A&M researching advanced methods of multigroup cross-section generation for reactor calculations.

Cross sections were generated at 500K for a uniformly hot, zero-power fuel pin. Five isotopes are present in this problem: uranium-235 and -238 in the fuel, oxygen-16 treated for thermal scattering in UO_2, hydrogen-1 thermally treated for scattering in water, and oxygen-16 in the water. The fuel pin had an outer radius of 0.97cm and a pitch of 1.29cm with reflecting boundary conditions. The fuel was 3.5 weight percent U-235 at the nominal density of UO_2. Cladding was neglected. The problem was solved with isotropic scattering, four polar and sixteen azimuthal angles per octant, and PDT's two-dimensional reactor-grid functionality with four rings of cells in the fuel, and one ring of cells in the water.

This problem suffers from PDT's inability to calculate the implicit sensitivity coefficients, but the UQ machinery can be exercised without these quantities as a first order estimate of the uncertainty in this problem's reactivity.

The contribution of each isotope to the total variance is displayed in Table 5.13.

The 69-group pin cell problem contains 1,035 non-zero multigroup cross sections, 651 independent parameters, and 45 important independent parameters. Of these 45 important parameters, 95% of the variance is accounted for in the largest 22 important parameters; the 23rd parameter is responsible for less that 0.5% of the total variance. To reduce the dimensionality of the input space, for this problem the

Isotope	**Contribution to Variance**
Uranium-238	61.6%
Uranium-235	34.4%
Hydrogen-1 (water)	3.9%
Oxygen-16 (UO_2)	0.03%
Oxygen-16	0.02%

Table 5.13: Relative contribution of each isotopes to uncertainty in the 69-group pin cell problem (values are rounded). The (water) and (UO_2) comments denote thermally-treated bulk materials.

threshold for importance was decreased to the parameters responsible for 95% of the total variance instead of 99% of the total variance.

The number of surfaces that passed the variance-based screening step is presented in Figure 5.22. The mean and variance calculated with these surfaces is presented in Figure 5.23.

The response surfaces converge to a value for the QoI with about sixty samples of the input space and converge to a value for the variance with about fifty samples of the input space. The QoI calculated at the mean value of the cross sections is very close to the QoI produced by the response surfaces. The variance predicted by the LPoU approximation using sensitivities calculated at the mean value of the cross sections, however, is significantly larger than the variance predicted by the response surfaces. These values are summarized in Table 5.14.

Although it appears in Figure 5.23 that the LPoU estimate of the variance vastly overestimates the variance in the QoI when compared to the response surfaces, we see that the two predicted standard deviations are 0.85% and between 0.46% and 0.51%, respectively. Differences in the variance are suppressed when the standard deviation is calculated by taking the square root of the variance. It should be noted

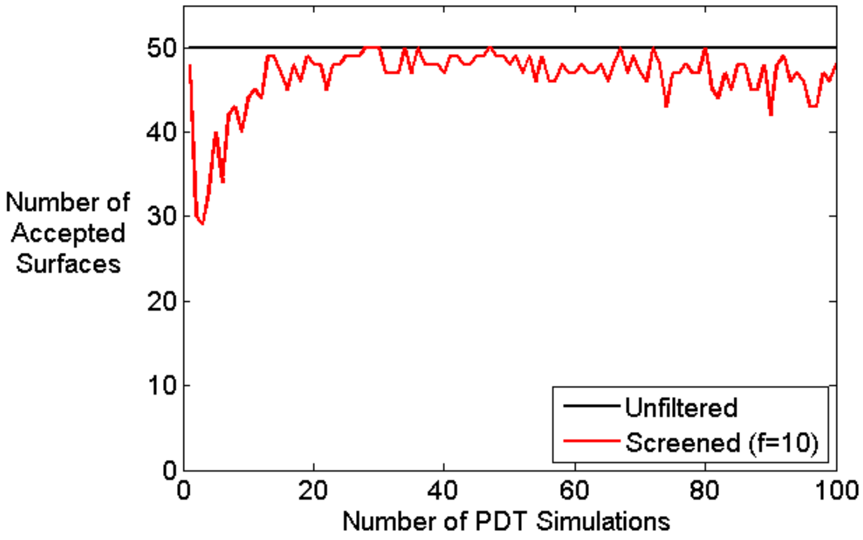

Figure 5.22: Number of response surfaces for the 69-group pin cell problem which pass the variance screening step with a screening factor of 10.

Method	**QoI** (ρ)	**Std Dev** (% QoI)
Mean Value of XS	0.3208	0.85
Response Surfaces (50-60)	0.3208	0.51
Response Surfaces (75-75)	0.3204	0.51
Response Surfaces (90-100)	0.3201	0.46

Table 5.14: QoI and variance in the QoI for the 69-group pin cell problem.

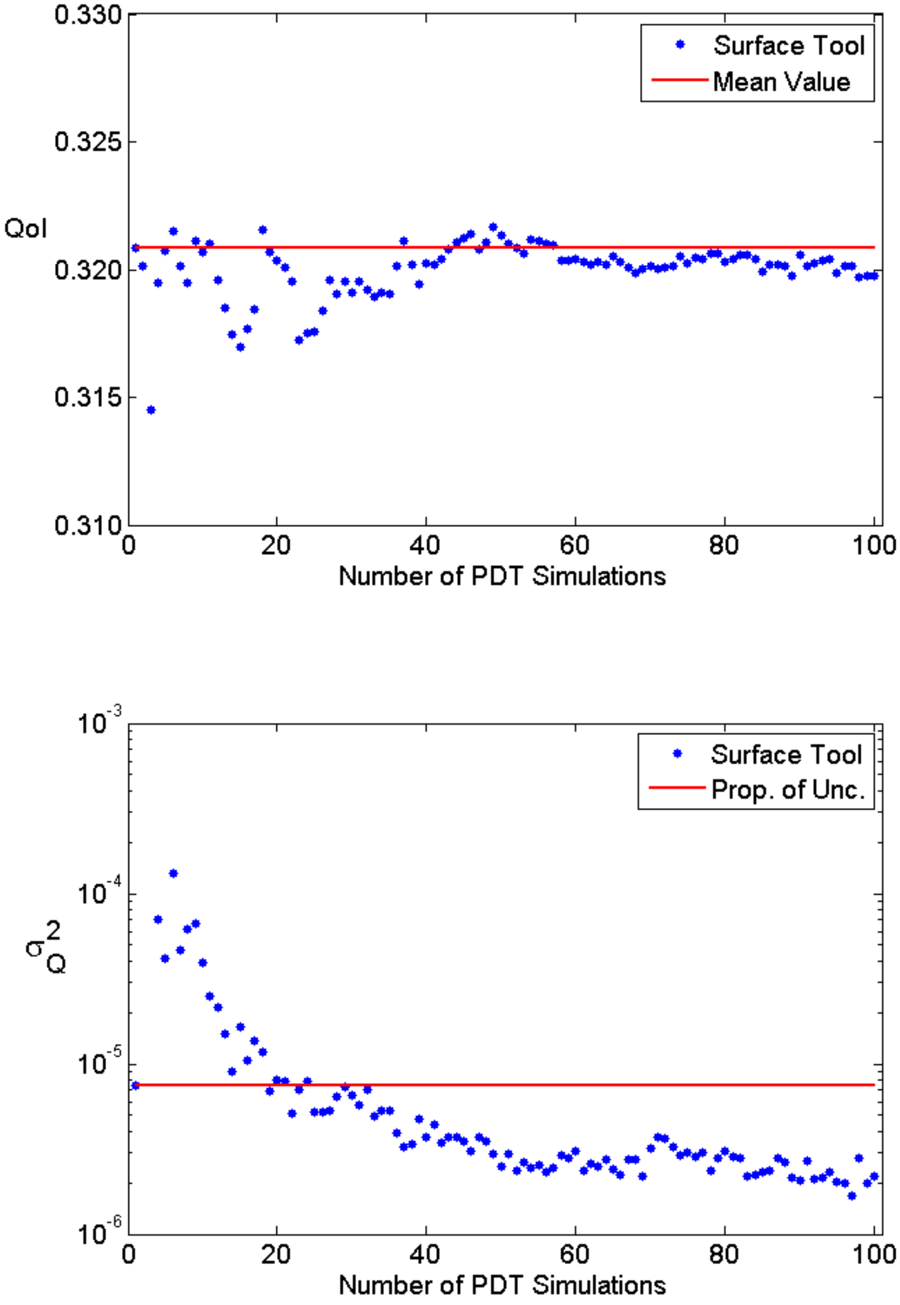

Figure 5.23: Mean and variance for the 69-group pin cell problem calculated from the screened response surfaces.

that the LPoU estimate of the variance, while 1.66 times as large as the response surface estimate, was at least fifty times less expensive to calculate than the response surface's estimate.

The partial derivatives of the QoI with respect to each of the uncertain parameters is plotted in Figure 5.24. Because there are so many important parameters for this problem, the axis labels are left off of the twenty two plots, but as in the previous plots of the partial derivatives, the x-axis corresponds to the parameter's value and the y-axis corresponds to the partial derivative.

Notice that the twenty-first parameter's partial derivative does not vary smoothly across the parameter's range. This parameter's partial derivative is plotted in Figure 5.25 so that it may be examined more closely.

This structure may imply that this parameter contributes to the QoI as the product of several parameters (which may or may not be included in the twenty-two important parameters), or it may reflect noise in the calculation of the terms in the inner products that yield this sensitivity. The derivative varies between 1E-3 and about 2.5E-3, meaning that the relative error of the slope in this parameter may on the order of 100%. The behavior of this partial derivative greatly complicates the construction of a response surface, as an extremely large number of splines would be required to accurately resolve the fine-scale behavior visible in this parameter.

Note that the range of partial derivatives (the spread in the y-axis of the plot) is approximately constant over the parameter's range (the x-axis of the plot), suggesting that a linear model at the average slope may be more appropriate than trying to fit the rapidly-varying data points.

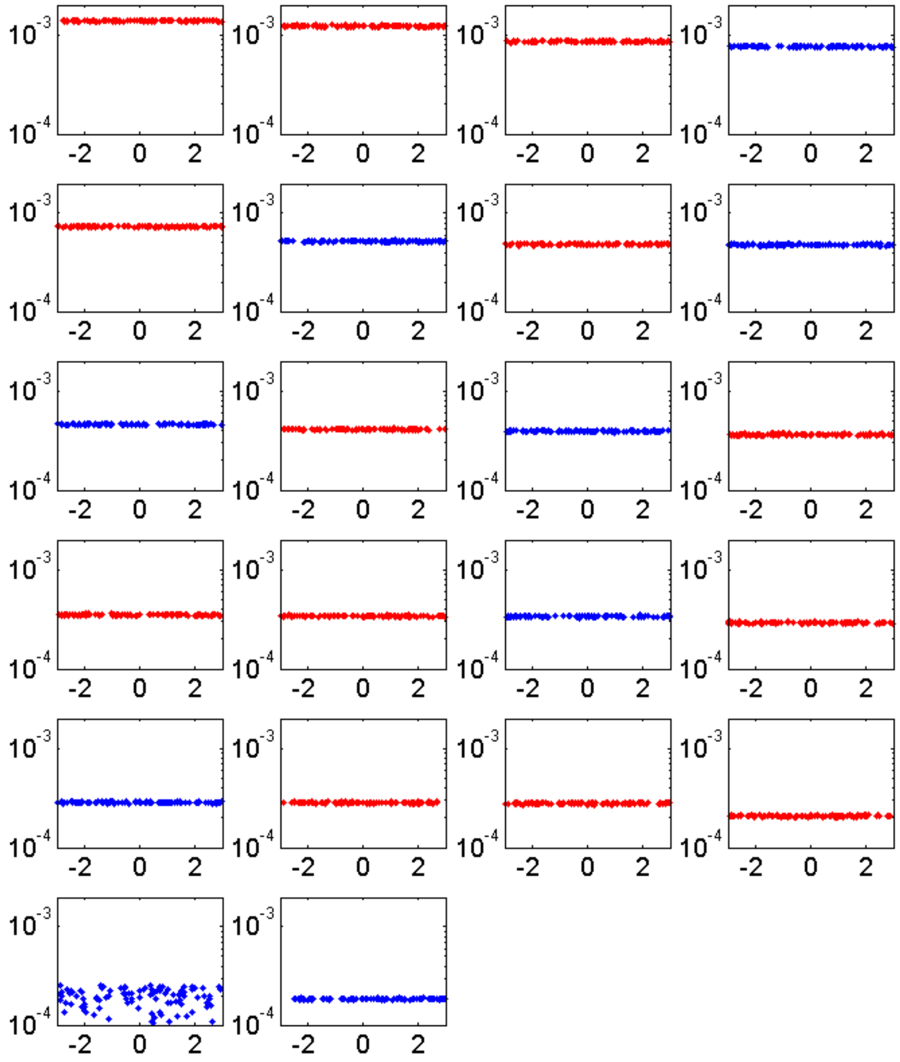

Figure 5.24: Partial derivatives $\frac{dQ}{dp}$ for the twenty two important parameters for the 69-group pin cell problem. Positive values are in blue, negative values are in red.

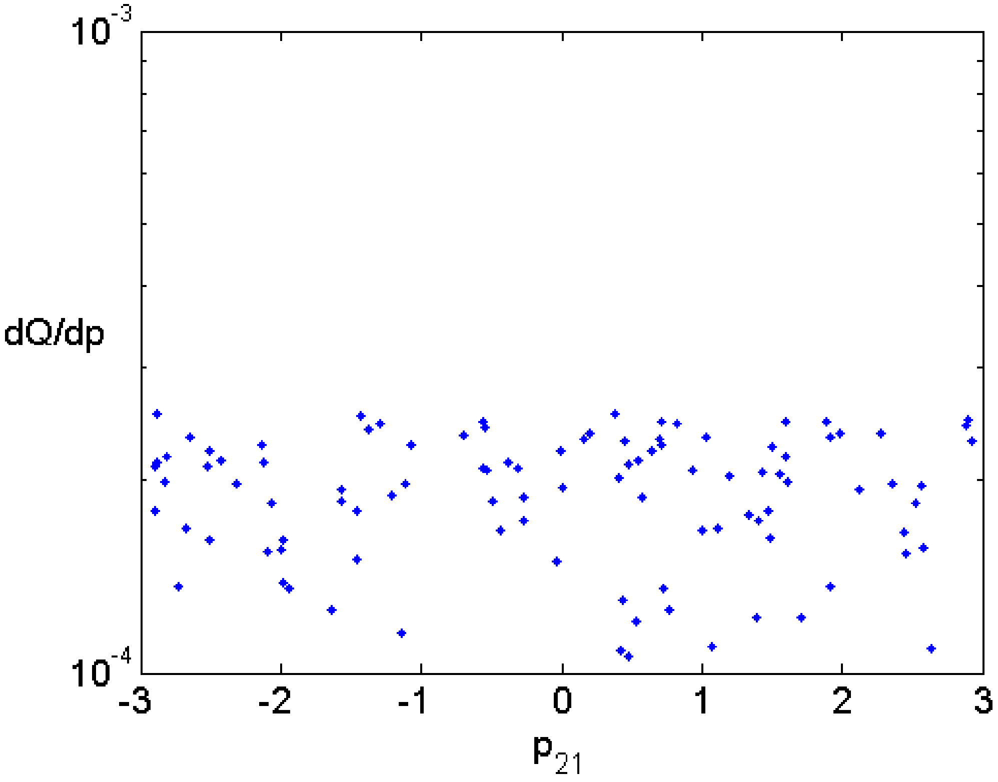

Figure 5.25: Partial derivatives $\frac{dQ}{dp}$ for the twenty first important parameter for the 69-group pin cell problem. Positive values are in blue, negative values are in red.

Although the adjoint sensitivity calculations performed with PDT do not include the implicit component of the sensitivity coefficients, this example problem demonstrated how the response surface machinery could be applied to a reactor problem with a large number of uncertain input parameters.

6. CONCLUSION

The novel elements of the UQ methodology introduced in this work are summarized in this section. The most interesting result generated in the course of this project and the real-world application of this result is highlighted. Finally, several avenues for continued research are proposed.

6.1 Summary of Method and Results

The UQ methodology introduced in this work was successfully applied to a variety of test problems related to various nuclear engineering applications. The cost of radiation transport calculations for realistic problems and the dimensionality of the uncertain cross-section data make traditional UQ analysis of these problems with Monte Carlo methods prohibitively expensive. The linear sensitivity analysis approach, which utilizes the linear propagation of uncertainty equation, was found to be accurate for some problems of interest, but was demonstrated to be insufficient for several of the problems examined in this work.

Significant capability has been added to the radiation transport calculations being performed at Texas A&M University. The Python tool written to create NJOY input decks, run NJOY, extract the multigroup cross-section data, and write PDT-formatted data files automated what had been a time-consuming and error-prone manual process. The ability to generate the covariance matrices associated with the multigroup cross sections did not exist at Texas A&M before this project. The covariance matrices may be used in conjunction with a single adjoint sensitivity calculation to estimate the variance in some quantity of interest using the linear propagation of uncertainty equation, or it may be used to perform the more thorough UQ analysis presented in this work.

The independent parameters revealed by the eigenvalue decomposition of the covariance matrix is much more easily sampled than the correlated multigroup cross sections. The way in which these parameters are identified and the mapping between the two spaces preserves the physical correlations between the multigroup cross sections. The idea of working in this independent-parameter space is applicable to more than just problems involving uncertain multigroup cross sections. The adjoint sensitivity method may be applied to calculate sensitivities with respect to a number of variables, including atomic densities, cell dimensions, and source spectra. Cases in which these variables are correlated may be described by a covariance matrix and an independent-parameter space can be identified with the eigenvalue decomposition of this matrix.

The variance-based dimension reduction scheme introduced in this work is novel and was demonstrated to very effectively reduce the dimensionality of the multigroup cross-section space. The reduced-dimensional space may be small enough for some problems that an outside code package like DAKOTA can be applied for UQ analysis. Other problems were characterized by a relatively large number of important, independent parameters that precludes the use of most UQ software. In these cases the response surface tool written in this work may be applied.

The UQ analysis performed in this project has already proved useful in the design of experiments for the CERT project, identifying hydrogen in the polyethylene as a major (and unexpected) source of uncertainty in the predicted detector response. As the complexity of the CERT experiment increases over the course of the project, these analysis will remain an important part of the design of the experiments and the interpretation of the computational results.

6.2 Future Work

Work could continue on this project in a number of directions. A few of the more promising avenues of investigation are outlined in this final section.

6.2.1 Gradient-Informed Surface Fitting

The GEMARS response surface tool seeks to fit splines to every partial derivative at every data point. Several of the test problems considered in this work contained important parameters with rapidly varying partial derivatives. The sixty-nine group pin cell problem's twenty-first important parameter demonstrated this behavior in Figures 5.24 and 5.25.

Convergence of the response surface tool relied upon the L_2 norm of the relative error in the QoI and sensitivity coefficients calculated at each point in the input space. The erratic behavior of the partial derivative in this dimension complicates the construction of a response surface for this problem. While this behavior may be explained by cross terms among the important parameters, it may also be reasonable to impose a single linear spline at the least-squares best estimate of the slope in that dimension. Further investigation of these special cases and how to treat them is warranted.

6.2.2 Dimension Reduction-Informed Sampling Methods

The UQ process presented in this work always drew random samples from uniform distributions of the uncorrelated parameters on the assumption that the dimensionality of the input space was too large to perform stratified sampling or latin hypercube sampling of these parameters. However, in several of the test problems, the number of important parameters was determined to be very small. This may present an opportunity to improve the method used to sample the input space.

Convergence of the response surface tool might be accelerated by performing a small number of physics calculations spread randomly across the input space and using these samples to estimate the number of important parameters. A sampling method can be chosen for the important parameters based upon this number. In the test problems with four and six important parameters, for example, advanced sampling techniques could be applied to these important parameters while the other parameters are sampled randomly. Problems containing a large number of important parameters would not benefit from this approach, but in these cases the original sampling scheme should be continued.

6.2.3 Adaptive Sampling Methods

The UQ machinery constructed for this work was structurally rigid in that each step in the UQ process was strictly sequential. The roadmap for UQ analysis introduced in the introduction and repeated here (Figure 6.1) presents the flow of information as strictly left-to-right.

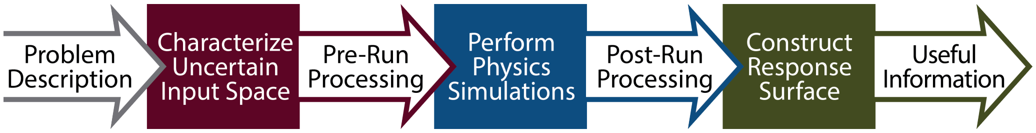
Problem Description
Characterize Uncertain Input Space
Pre-Run Processing
Perform Physics Simulations
Post-Run Processing
Construct Response Surface
Useful Information

Figure 6.1: A graphical representation of the UQ process.

In this conceptualization of the UQ process, the cross-section space is sampled, PDT cross-section files are written, and then PDT is run for all of the realizations of the cross-section space. This information is then passed to the response surface construction tool. While this approach was demonstrated to work in this paper, a more efficient implementation may be possible (and perhaps required) for the most expensive problems. Instead of choosing all of the sampling points a priori, convergence of the response surfaces might be accelerated by adaptively choosing sampling points during the UQ process.

In this adaptive approach to sampling the input space, an initial batch of samples are drawn randomly from uniform distributions over the input space. The adjoint sensitivity problem is run at each of these points and a family of response surfaces are generated with this data. In addition to sampling the surfaces normally to generate statistics for the QoI, the surfaces can be sampled at grid points and compared. Regions in which the surfaces show good agreement will probably not benefit from additional physics simulations at a nearby point. Regions in which the surfaces disagree, however, would likely benefit from addition QoI and sensitivity data in the vicinity of the disagreement. In this way additional sampling points can be chosen to generate data in the regions of the input space which most need additional information. This approach to UQ is visualized in Figure 6.2.

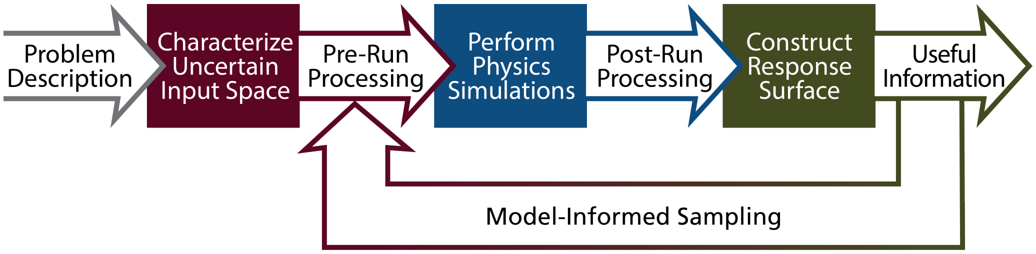

Figure 6.2: A graphical representation of the iterative UQ process using adaptive sampling of the input parameters.

This iterative process of choosing points in the input space, running the physics calculation, and construction response surfaces is especially well suited to large, expensive programs in which the physics simulations dominate the cost of the UQ analysis. In these cases, the delay due to the in-line construction of response surfaces is likely to be negligible compared to savings in the total number of simulations necessary to sample the input space.

6.2.4 Cross Section Samples with Consistent Weighting Spectra

Realizations of the multigroup cross-section space used in this work were generated by drawing samples of the multivariate Gaussian distribution associated with the multigroup cross sections. As discussed in Section 3.8.2, these realizations are inconsistent with the weighting spectrum used to generate the multigroup cross sections from the ENDF data file.

The process to generate consistently-weighted realizations of the multigroup cross sections proposed in 3.8.2 requires an intermediate step in which a set of ultra-fine multigroup cross sections are generated using a simple weighting spectrum. Realizations of the ultra-fine cross sections are drawn from the multivariate Gaussian distribution associated with these cross sections. The infinite medium slowing down problem is solved at each realization of the ultra-fine cross sections to generate weighting spectra consistent with the realized data. Finally, the realizations of the ultra-fine multigroup cross sections are collapsed to the desired group structure using the consistent weighting spectra.

It would be interesting to compare the realizations of the multigroup cross sections produced in this fashion to realizations of the multigroup cross sections drawn from the standard multivariate Gaussian distribution defined by the mean value of the multigroup cross sections and their associated covariance matrices. Implementing this approach and answering this question is an ongoing research project.

6.2.5 Implicit Sensitivities for Reactor Calculations

PDT's inability to calculate implicit sensitivity coefficients does not strongly impact the source-detector problems of interest to the CERT project but limits the applicability of the adjoint sensitivity method to reactor applications. The work of Andrew Till [31] may allow the construction of a multigroup-like energy discretization compatible with PDT and the adjoint sensitivity calculations that resolves the resonances in the resolved resonance region. This may eliminate the distinction between the explicit and implicit sensitivity coefficients.

Till's "FEDS" energy discretization scheme should be tested with PDT's adjoint sensitivity capability and the UQ machinery implemented in this work and compared to results generated with TSUNAMI. If successful, this would allow PDT to be applied as a tool for UQ for the analysis of complex nuclear reactor problems.

6.2.6 Extension to Other Parameters of Interest

The response surface tool written in the course of this work is not specific to neutron transport calculations with uncertain multigroup cross-section data. With minor extension of the adjoint sensitivity modules added to PDT by Stripling [28], sensitivities could be calculated for a QoI with respect to nuclide densities or the source spectrum. The response surface tool could then be applied to these problems.

REFERENCES

[1] US ENDF/B Libraries. Los Alamos National Laboratory, T-2 Nuclear Information Service, accessed May 2016, https://t2.lanl.gov/nis/data.shtml.

[2] H. S. Abdel-Khalik, Y. Bang, C. Kennedy, and J. Hite. Reduced order modeling for nonlinear multi-component models. *International Journal for Uncertainty Quantification*, Volume 2(4):341–361, 2012.

[3] I. K. Abu-Shumays. Compatible product angular quadrature for neutron transport in $x-y$ geometry. *Nuclear Science and Engineering*, Volume 64(2):299–316, 1977.

[4] M. L. Adams. Numerical methods for reactor analysis. NUEN 629 Class Notes, Spring 2013, Nuclear Engineering Department, Texas A&M University, College Station, Texas.

[5] T. S. Bailey. The piecewise linear discontinuous finite element method applied to the RZ and XYZ transport equations. Doctoral Dissertation, Nuclear Engineering Department, Texas A&M University, College Station, Texas, Spring 2008.

[6] G. I. Bell and S. Glasstone. *Nuclear Reactor Theory.* Van Nostrand Reinhold Company, New York, USA, 1970.

[7] D. E. Bruss, R. G. McClarren, M. L. Adams, and J. E. Morel. Construction of response surfaces for a reactor-like problem with realistic cross section uncer-

tainties. In *Joint International Conference on Mathematics and Computation (M&C)*, Nashville, Tennessee, April 19-23 2015.

[8] D. G. Cacuci. *Sensitivity & Uncertainty Analysis, Volume 1.* Chapman and Hall, New York, 2003.

[9] D. G. Cacuci. *Sensitivity & Uncertainty Analysis, Volume 2.* Chapman and Hall, New York, 2003.

[10] Y. Cao, S. Li, L. Petzold, and R. Serban. Adjoint sensitivity analysis for differential-algebraic equations: Algorithms and software. *Journal of Computational and Applied Mathematics*, Volume 149(1):171–191, 2002.

[11] P. G. Constantine, E. Dow, and Q. Wang. Active subspace methods in theory and practice: Applications to Kriging surfaces. *SIAM Journal of Scientific Computing*, Volume 36(4):A1500–A1524, 2014.

[12] D. G. T. Denison, C. C. Holmes, B. K. Mallick, and A. F. M. Smith. *Bayesian Methods for Nonlinear Classification and Regression.* Wiley, Hoboken, USA, 2002.

[13] D. G. T. Denison, B. K. Mallick, and A. F. M. Smith. Bayesian MARS. *Statistics and Computing*, Volume 8:337–346, 1998.

[14] J. J. Duderstadt and L. J. Hamilton. *Nuclear Reactor Analysis.* Wiley, Hoboken, New Jersey, USA, 1976.

[15] A. Gandini. Comments on higher order generalized perturbation theory. *Nuclear Science and Engineering*, Volume 73(3):289–294, 1980.

[16] J. E. Gentle. *Computational Statistics*. Springer, Dordrecht, New York, 2009.

[17] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, Volume 57:97–109, 1970.

[18] D. F. Hollenback and P. B. Fox. CENTRM validation. Technical report, Oak Ridge National Laboratory, 2005. ORNL/TM-2004/66.

[19] V. Kuleshov and D. Precup. Algorithms for the multi-armed bandit problem. *Journal of Machine Learning Research*, Volume 1:1–48, 2000.

[20] L. Leal, D. Mueller, G. Arbanas, D. Wiarda, and H. Derrien. Impact of the 235U covariance data in benchmark calculations. In *International Conference on the Physics of Reactors, Nuclear Power: A Sustainable Resource*, Interlaken, Switzerland, September 14-19 2008.

[21] E. E. Lewis and W. F. Miller, Jr. *Computational Methods of Neutron Transport*. American Nuclear Society, La Grange Park, Illinois, USA, 1993.

[22] R. E. MacFarlane and D. W. Muir. The NJOY nuclear data processing system, version 91. Technical report, Los Alamos National Laboratory, 1994. LA-12740-M.

[23] R. G. McClarren, D. Ryo, R. P. Drake, et al. A physics informed emulator for laser-driven radiating shock simulations. *Reliability Engineering and System Safety*, Volume 96:1194–1207, 2011.

[24] J. E. Morel. Neutron transport theory. NUEN 625 Class Notes, Fall 2012, Nuclear Engineering Department, Texas A&M University, College Station, Texas.

[25] C. M. Perfetti. Oak Ridge National Laboratory, personal communication, April 19, 2016.

[26] B. T. Rearden, D. E. Mueller, S.M. Bowman, R. D. Busch, and S. J. Emerson. TSUNAMI primer: A primer for sensitivity/uncertainty calculations with SCALE. Technical report, Oak Ridge National Laboratory, 2009. TM-2009/027.

[27] R. Sanchez and J. C. Ragusa. On the construction of Galerkin angular quadratures. *Nuclear Science and Engineering*, Volume 169(2):133–154, 2011.

[28] H. F. Stripling. Adjoint-based uncertainty quantification and sensitivity analysis for reactor depletion calculations. Doctoral Dissertation, Nuclear Engineering Department, Texas A&M University, College Station, Texas, Fall 2013.

[29] H. F. Stripling, M. Anitescu, and M. L. Adams. A generalized adjoint framework for sensitivity and global error estimation in time-dependent nuclear reactor simulations. *Annals of Nuclear Energy*, Volume 52:47–58, 2013.

[30] H. F. Stripling and R. G. McClarren. Gradient enhanced bayesian MARS for regression and uncertainty quantification. *Transactions of the American Nuclear Society*, Volume 105:454–457, 2011.

[31] A. T. Till. Finite elements with discontiguous support for energy discretization in particle transport. Doctoral Dissertation, Nuclear Engineering Department, Texas A&M University, College Station, Texas, Fall 2015.

[32] J. Vermorel and M. Mohri. Multi-armed bandit algorithms and empirical evaluation. In *In European Conference on Machine Learning*, pages 437–448. Springer, 2005.

[33] M. L. Williams and B. T. Rearden. SCALE-6 sensitivity/uncertainty methods and covariance data. *Nuclear Data Sheets*, 109(12):2796–2800, 2008.

[34] V. Zerpin. Evaluated nuclear data file database. Nuclear Data Services, International Atomic Energy Agency, accessed May 2016, https://www-nds.iaea.org/exfor/endf.htm.

APPENDIX A

ANALYTIC PROBLEM WITH NON-PHYSICAL DATA

An interesting feature was noted when the analytic test problem presented in Section 5.1.2 was first constructed. The purely absorbing slab was initially comprised of boron-10 and considered only the radiative capture (n,γ) cross section. Boron 10's radiative capture cross section is orders of magnitude smaller than it's (n,α) cross section and is relatively uncertain.

In the slowest group, the (n,γ) cross section is 2.50 barns and the standard deviation in that group's cross section is 1.00 barns. The (n,α) cross section in the slowest group, in contrast, is more than 19,000 barns and has a standard deviation of only 16.0 barns. The multivariate Gaussian distribution corresponding to the (n,γ) cross sections can produce realizations containing negative cross sections, which are physically impossible. These non-physical cross sections are set to zero when written as PDT-formatted cross-section files, effectively truncating the multivariate Gaussian distribution.

The analytic problem was performed using the (n,γ) cross sections for one important reason. Seventy-eight of the original 99 energy groups had non-zero cross sections. The dimension reduction revealed that only a single parameter was important for this problem. Because the QoI is a function of only a single parameter, the QoI and the sensitivity of the QoI to the single important parameter can be plotted for this problem. These plots are presented in Figure A.1.

The QoI and sensitivity vary smoothly with respect to the single parameter of interest p_0. Several features of this plot are worth further inspection. First, note that the QoI is not constant over the range of p_0. This indicates that the linear propagation of uncertainty equation would not yield an accurate estimate of the variance for this problem. Second, for -2.5 $< p_0$, the plot of the slope of the QoI in the second plot appears to match the slope of the QoI in the first plot. This confirms that the mapping of the partial derivative between the cross-section space and the independent-parameter space is accurate.

Finally, note that for $p_0 <$ -2.5 the QoI is exactly zero and the slope of the QoI in this region is clearly zero. The sensitivity coefficient calculated with the adjoint sensitivity method is plotted against p_0 in the second plot and clearly shows a large, non-zero sensitivity in this region. This disagreement would hinder the construction of a response surface for this problem as the data provided to the surface tool is contradictory in this region. The source of this contradiction is outlined below.

Realizations of the cross sections generated with values of the independent parameter -3 $< p_0 <$ -2.5 had the (n,γ) cross sections equal to zero in all groups. Recall the QoI is calculated as defined in Eq. (3.24),

$$Q = \left\langle \psi, q^{\dagger} \right\rangle . \tag{A.1}$$

The QoI is clearly zero if the cross section in the region of interest is zero,

$$Q = \langle \psi, 0 \rangle = 0. \tag{A.2}$$

The sensitivity coefficients, originally defined in Eq. (3.41), are repeated here for a problem with no scattering,

$$\frac{dQ}{dp} = \left\langle \psi, \frac{\partial}{\partial p} q^\dagger \right\rangle - \left\langle \psi^\dagger, \left(\frac{\partial}{\partial p} L \right) \psi \right\rangle . \tag{A.3}$$

These two terms are easily addressed. The adjoint solution $\psi^\dagger$ in the second term is calculated from the adjoint transport equation, Eq. (3.39), presented here for transport in a purely absorbing medium,

$$L^\dagger \psi^\dagger = q^\dagger . \tag{A.4}$$

If the cross section in the region of interest is zero than the adjoint source $q^\dagger$ is zero and the adjoint solution $\psi^\dagger$ must also be zero. The second term in the equation for the sensitivity coefficients is then zero,

$$\left\langle \psi^\dagger, \left(\frac{\partial}{\partial p} L \right) \psi \right\rangle = \left\langle 0, \left(\frac{\partial}{\partial p} L \right) \psi \right\rangle = 0. \tag{A.5}$$

The first term in A.3 contains the forward angular flux solution ψ and partial derivatives of the adjoint source with respect to the multigroup cross sections. The angular flux ψ streams through the problem from the boundary without interaction and is therefore equal to the imposed boundary condition.

The partial derivative of the adjoint source is applied with respect to the cross section in each energy group, $\sigma_{(n,\gamma),g}$. The $q^\dagger$ term contains these microscopic cross sections times the atomic density of boron-10 in the problem, N_{B-10}. The partial derivative of $q^\dagger$ with respect to the multigroup cross sections is therefore either 0 or N_{B-10}. The first term in Eq. (A.3) is seen to contain inner products of ψ and 0 and inner products of ψ and the fixed number density N_{B-10}. The adjoint sensitivity method therefore yields a nonzero sensitivity coefficient for cases in which the cross sections in the region of interest are zero.

This apparent shortcoming in the adjoint sensitivity method would not have been observed if the (n,γ) cross sections had been allowed to be negative. In this case the curve for the QoI in Figure A.1 would have continued smoothly in the region -3 $< p_0 <$ -2.5 and the derivatives calculated with the adjoint sensitivity method would have reflected this behavior. However, non-physical negative total cross sections are rejected by PDT before calculations begin. This causes the disagreement observed in this problem. Problems with realistic cross-section uncertainty should not exhibit this behavior.

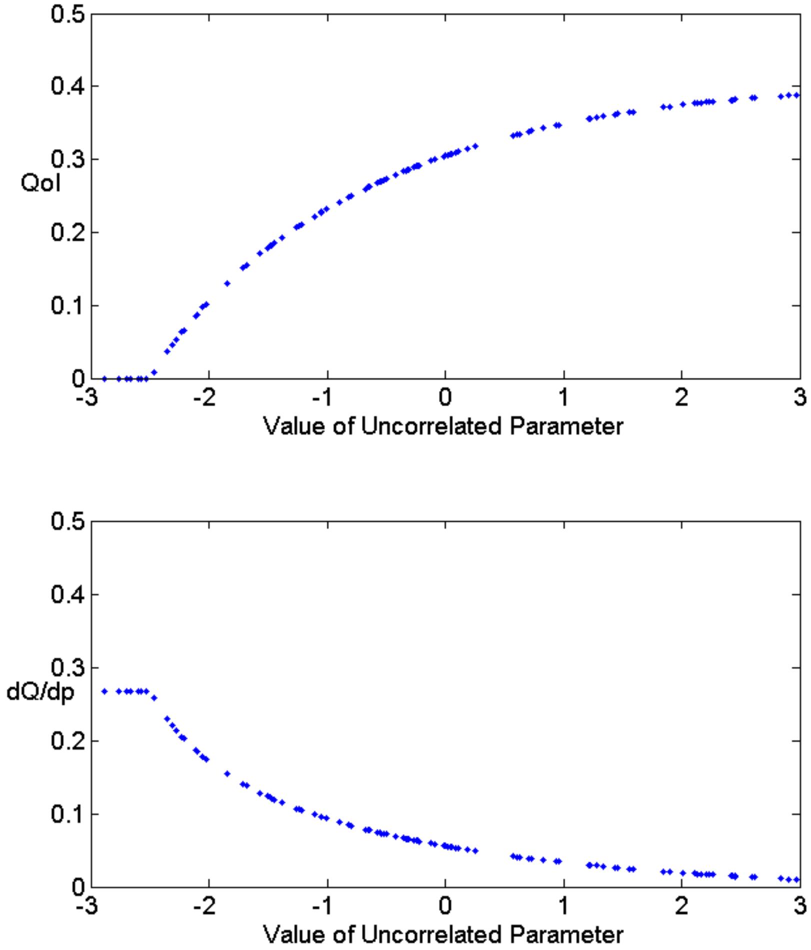

Figure A.1: QoI and sensitivity plotted against the single important parameter for the analytic test problem using only the (n,γ) cross sections.

APPENDIX B

PDT INPUT AND OUTPUT FOR ADJOINT PROBLEMS

Only minor modifications are required to the PDT input deck to solve perform adjoint sensitivity calculations. For detailed look at these options, refer to [28].

B.1 Adjoint Blocks in the Input Deck

Two blocks of information must be added to the PDT input deck to run adjoint sensitivity calculations for a radiation transport problem. These problems are always defined in addition to a full forward problem definition and are always performed immediately following the forward calculation.

The QoI and sensitivity coefficients are printed to the ".output_N" file produced by each of the N processors used to run the problem. For large problems, like the IM-1 problems run on 3500 cores on the supercomputer "Vulcan" at Lawrence Livermore National Laboratory, it is more practical to have the QoI and sensitivity coefficients printed to the standard out "slurm" file. This is accomplished by setting the "stdout_qoi" and "stdout_sens" flags in the "edits" category to "on",

```
<edits>
  ...
  <stdout_qoi>on</stdout_qoi>
  <stdout_sens>on</stdout_sens>
</edits>
```

Here "..." denotes other edits desired by the user.

The first required input block is placed within the “common” block at the start of the input deck. A problem name and QoI name are each specified. The “forward_solution” option is used for time-dependent problems for which the forward solution cannot be stored in memory at all time steps; for the problems considered in this work, this option should be set to “store”.

The “adjoint_QOI_do” block may be set to “compute_sensitivity” or “nothing”; this second option is useful when performing brute-force forward calculations for the QoI without the sensitivity coefficients. Finally, the parameters of interest are specified. For the problems considered in this work, these are the total, scattering, and fission cross sections, “sigma_t”, “sigma_s”, and “sigma_f”, respectively.

```
<problemID>problem_name</problemID>
<!-- Adjoint Problem Definition -->
<adjoint_def>
  <adjoint_QOI_id>qoi_name</adjoint_QOI_id>
  <forward_solution>store</forward_solution>
  <adjoint_QOI_do>compute_sensitivity</adjoint_QOI_do>
  <param_vec_def>sigma_t</param_vec_def>
  <param_vec_def>sigma_s</param_vec_def>
</adjoint_def>
```

The QoI is defined in a second block in the PDT input file. This block is located outside of the “common” block. The “QOI_def” block contains the QoI name specified in the first block, the QoI type, the material of interest “keyComp”, and the volume of the detector, specified in the format for PDT edits. Two QoI types were considered in this work: “detector_response” or “reaction_rate” (equivalent) and “reactivity”.

The detector volume is specified by dimension (0, 1, and 2 in a 3-d problem) and then by regions within those dimensions. In this slab-like example problem, the active component in the detector is B-10, and the detector volume is located in the 0th region in the x- and y-dimensions and the 2nd region in the z-dimension.

```
<!-- Specify a QOI -->
<QOI_def>
  <QOI_type>detector_response</QOI_type>
  <QOI_id>detecrr</QOI_id>
  <keyComp>B-10</keyComp>
  <QOI_reg.dim_bounds>
    <QOI_reg.dim_bounds.dim>0</QOI_reg.dim_bounds.dim>
    <QOI_reg.dim_bounds.div_start>0</QOI_reg.dim_bounds.div_start>
    <QOI_reg.dim_bounds.div_end>0</QOI_reg.dim_bounds.div_end>
  </QOI_reg.dim_bounds>
  <QOI_reg.dim_bounds>
    <QOI_reg.dim_bounds.dim>1</QOI_reg.dim_bounds.dim>
    <QOI_reg.dim_bounds.div_start>0</QOI_reg.dim_bounds.div_start>
    <QOI_reg.dim_bounds.div_end>0</QOI_reg.dim_bounds.div_end>
  </QOI_reg.dim_bounds>
  <QOI_reg.dim_bounds>
    <QOI_reg.dim_bounds.dim>2</QOI_reg.dim_bounds.dim>
    <QOI_reg.dim_bounds.div_start>2</QOI_reg.dim_bounds.div_start>
    <QOI_reg.dim_bounds.div_end>2</QOI_reg.dim_bounds.div_end>
  </QOI_reg.dim_bounds>
</QOI_def>
```

B.2 Example PDT Adjoint Sensitivity Output

The QoI is reported in the PDT output files as the “QOI_VALUE”,

```
BEGIN_RECORD TYPE = QOI_VALUE
32128138616.22537
END_RECORD
```

The sensitivity coefficients are reported as a “QOI_SENSITIVITY” record by parameter (total, scattering, and fission cross section) and then by material. In the case of total cross section, the coefficients are reported by group.

```
BEGIN_RECORD TYPE = QOI_SENSITIVITY
BEGIN_RECORD PARAMETER = sigma_t
BEGIN_RECORD COMPONENT_ID = H-1_poly
# group, dQOI/dp
0 0
1 -5624520623.069702
2 -662316425.5824511
3 -676004084.2877663
...
```

The scattering cross-section sensitivity coefficients are reported by material, and then by group to (“sink”), moment, and group from (“source”).

```
BEGIN_RECORD PARAMETER = sigma_s
BEGIN_RECORD COMPONENT_ID = H-1_poly
# sink (g), moment, source (g’), dQOI/dSigmaS(g’->g,m)
0 0 0 0
0 0 1 3691933641.574047
0 0 2 403851317.7467869
0 0 3 395782507.2423372
...
```

Note that the “BEGIN_RECORD PARAMETER” flag appears only once per parameter, not for each material.

APPENDIX C

PYTHON TOOL TO RUN NJOY

In this section the Python tool written to generate PDT-formatted cross-section files containing the multigroup cross sections is documented.

C.1 Overview

This python tool was created to automate the process of generating data files containing the multigroup cross sections used by PDT. This task was previously accomplished by hand each time cross sections were needed for a calculation. Given simple user input, this tool writes an input deck for NJOY 2012, calls NJOY, reads the NJOY output files, and writes PDT-formatted cross-section files.

Multigroup cross sections are generated for user-specified isotopes. The desired isotopes, group structure, weighting spectrum, and material temperature are specified in the Python tool's driver. The multigroup cross sections generated with this tool were verified by comparing PDT cross-section data files against files generated by Andrew Till and Weixiong Zheng, two students in the computational physics group that also run NJOY to produce cross sections for the CERT project, and by running problems using the data files generated with this tool and with data files provided by Till and Zheng and comparing quantities of interest between these problems.

The names of directories and files, as well as options in the Python driver file, will be denoted with in a `distinct font` in this documentation.

C.2 File Organization

After untarring this tool, the directory should contain a README file, a driver titled `njoy_driver.py`, and directories `njoy_step`, `pdt_xs_files`, `src`, and `xs_data`. These files are discussed individually in the following subsections.

C.2.1 Documentation File

This Appendix is included in the code tarball as the README file.

C.2.2 The Python Driver, `njoy_driver.py`

The Python driver `njoy_driver.py` contains the required user input and drives the process of writing the NJOY input file, running NJOY, reading the NJOY output file for the cross-section means and covariance matrices, writing the PDT-formatted cross-section file for the mean value of the cross sections, sampling the multivariate Gaussian distribution, and writing PDT-formatted cross-section files for each realization of the cross sections.

C.2.3 The Directory `src`

The `src` directory contains the Python code that performs each step of the cross-section generation process. This files contained in this directory and a brief explanation of each files' function are listed below.

- `write_run_njoy.py`: Write NJOY input files and then run NJOY 12.

- `read_njoy_out.py`: Read the NJOY 12 output for the multigroup cross sections, transfer matrices, and covariance matrices.

- `eig_investigation.py`: Take an eigenvalue decomposition of the covariance matrices for each isotope and report the total number of eigenvalues.

- `generate_realizations.py`: Draw samples of the multigroup cross-sections' multivariate Gaussian distribution.

- `write_pdtxs_file.py`: Write a PDT-formatted cross-section file for a given set of multigroup cross sections.

- `xsconverter.ext`: A `sanity check` tool written by Yunhuang Zhang to test PDT-formatted data files. Not currently used, but code to call this tool is included (and commented out) in the driver file.

C.2.4 The Directory `njoy_step`

The `njoy_step` directory contains five sub-directories. The names and contents of these directories are listed below.

- `endf_data`: Contains the evaluated nuclear data files (ENDF) needed to run NJOY. The user must provide these files. They are described in more detail in Section C.3.

- `group_bounds`: Custom group structures, described in Section C.3.

- `weight_functions`: Custom weight functions, described in Section C.3.

- `njoy_inputs`: The input files that are generated by the Python tool for NJOY 2012 are placed in this sub-directory. These files are useful to save as a record of the exact process used to generate the cross sections.

- `njoy_outputs`: The output files generated by NJOY at run time are placed in this sub-directory. These files are human-readable. These files are read by the Python tool to collect the multigroup cross sections and the covariance matrices.

C.2.5 The Directory `pdt_xs_files`

The directory `pdt_xs_files` contains a sub-directory titled `means` and a sub-directory titled `realizations.` PDT-formatted data files containing the mean value and realizations of the multigroup cross-section files are deposited into these two directories, respectively, at run time.

A directory titled `other_data` contains two sub-directories `transform_matrices` and `z_values.` At run time, the first sub-directory will be filled by the Python tool with the transform matrices used to map between the cross-section space and the independent-parameter space. The second sub-directory will contain the values of the independent (or `z`) parameters which were used to generate realizations of the cross sections. Finally, a file titled `scat_transfer_removal` is placed in this directory at run time. This file contains the removal and transfer scattering cross sections for each isotope.

C.2.6 The Directory xs_data

The directory xs_data must contain a directory titled figures and a directory titled matrices. figures will contain plots of the covariance and correlation matrices after NJOY is run. matrices will contain the covariance matrices after NJOY is run; matrices with components in the title break down the covariance matrix in an almost-human-readable format.

C.3 Prerequisites to Run the NJOY Python Tool

Modules required by the Python tool:

- NJOY12
- Python-2.7.8

The Texas A&M Nuclear Engineering Department has a license agreement for NJOY 12, but each individual user must sign the agreement. Dr. Marvin Adams is the sponsor at TAMU, and Daryl Hawkins and Cory Hearnsberger actually manage the software. Contact the Technology Transfer Division at Los Alamos National Laboratory to inquire about signing the agreement. As of the end of 2014, our contact was Mariann Johnston, mjonston@lanl.gov.

User supplied data:

- ENDF data files
- Custom group structure (if desired)
- Custom weighting spectrum (if desired)

The ENDF data files may be downloaded from the T-2 Nuclear Information Service at Los Alamos National Laboratory, https://t2.lanl.gov/nis/data.shtml. They are stored in the folder `./njoy_step/endf_data/` in the format `endf_mat`, where `mat` is the MAT number of the isotope of interest. Materials for which thermal scattering off of the bulk material is considered are stored in this format with an additional descriptor for the material of interest. The isotopes for which this treatment is available and their respective ENDF file names are, respectively,

- Hydrogen-1 in water, `endf_125_water`
- Hydrogen-1 in polyethylene, `endf_125_poly`
- Natural carbon in graphite, `endf_600_graphite`
- Oxygen-16 in UO_2 fuel, `endf_825_uo2`

In order to process these isotopes the free gas treatment of each isotope must be processed before the bulk material treatment (that is, in order to include hydrogen in water, free gas hydrogen must be specified in the NJOY input file even if no free gas hydrogen is needed for the problem of interest).

If a custom group structure is specified, it must stored in the folder `./njoy_data/group_bounds/` as a file titled `custom_group_struct.` Group boundaries are specified from smallest to largest. Several custom group structures are stored in this folder.

If a custom weighting function is specified, it must be stored in the folder `./njoy_data/weight_functions/` as a file titled `custom_weight_funct.` This file contains one line of title and then pointwise data in the format of energy in eV and weight w(E), ordered from lowest energy to highest energy.

Once the required modules are loaded and the required data is available, the command to run the Python program is `Python njoy_driver.py` from the directory containing the driver.

C.4 User Input

The Python NJOY tool is distributed as a tarball. Upon untarring the file the file structure described in Section C.2 should be present. After downloading the ENDF files from the LANL T2 site as described in Section C.3, the user may set problem specifics in the driver file. The required user input is a short section in `njoy_driver.py` between lines clearly labeled `Required User Input` and `End Required User Input.` These required inputs are listed below and then discussed in the following sub-sections.

- `scratch`
- `isotope_name`
- `mat`
- `group_option`
- `thermal_cutoff`
- `weight_function`
- `temp`
- `l_order`

- figure_format
- num_realizations

C.4.1 Location of Scratch, `scratch`

The address of a directory in which NJOY run-time files may be stored. This directory is deleted when NJOY execution is complete. The NJOY input and output files are saved after execution before the temporary directory containing the tapes generated with each NJOY module is deleted.

C.4.2 Isotope Names, `isotope_name`

A list of each isotope name. Natural carbon should be denoted "C-nat" and has MAT number 600.

```
isotope_name = [ 'H-1', 'H-1_poly', 'C-nat', 'C-nat_graphite' ]
```

Special cases involving thermal scattering off of bulk materials must be specified in this section. These isotopes are limited to

- Hydrogen-1 in water, `H-1_water`
- Hydrogen-1 in polyethylene, `H-1_poly`
- Natural carbon in graphite, `C-nat_graphite`
- Oxygen-16 in UO_2 fuel, `O-16_uo2`

C.4.3 MAT Numbers, `mat`

A list of the mat numbers. Example (to go with the isotope name example, above):

```
mat = [ 125, 125, 600, 600 ]
```

C.4.4 Group Structure, `group_option`

The group structure is specified with this option. Option 1 is a user-defined group structure. If selected, this requires the presence of a file `./njoy_step/custom_group_struct.` This file contains the group boundaries, one per line, from low energy to high energy.

NJOY contains several default group options which these may be selected by their number. A limited amount of detail about these built-in options is available in the NJOY manual in the GROUPR module [22]. The group structures used in this dissertation are listed below.

- `group_option = 1`: A user-supplied custom group structure is supplied.
- `group_option = 9`: EPRI 69-group light water reactor group structure.
- `group_option = 13`: LANL 80-group fast spectrum reactor group structure.

C.4.5 Thermal Cutoff, `thermal_cutoff`

Maximum energy for which the thermal treatment is applied. This number must match a group boundary. Suggested value is around 3 to 5 eV.

C.4.6 Weight Function, `weight_function`

Weighting function used by NJOY. The weighting functions used in this dissertation are listed below.

- `weight_function = 1`: A user-supplied custom weight spectrum.
- `weight_function = 3`: A 1/E spectrum.
- `weight_function = 4`: A spectrum containing 1/E, fission, and thermal Maxwellian components.

C.4.7 Material Temperature, `temp`

The temperature in Kelvin for which the cross sections are generated. This tool only supports generating cross sections for one temperature at a time.

C.4.8 Scattering Order, `l_order`

The Legendre order for scattering.

C.4.9 Covariance Matrix Figure Format, `figure_format`

Format of the figures produced to visualize the covariance matrices. Supported options are `png`, `pdf`, and `none`. Selecting `none` skips producing the figures. The `pdf` figures are better-looking than the `png` figures, but these figures can be quite large and time-consuming to generate. The `png` option produce figures that are usually of high-enough resolution while being much smaller than the PDF figures.

C.4.10 Number of Cross-Section Realizations, `num_realizations`

If the user wishes to generate samples of the cross-section space by drawing from the multivariate Gaussian distribution described by the cross-section means and the covariance matrices, this setting sets the number of realizations that will be generated. If set to zero, no samples will be drawn.

C.5 Code Execution

The Python NJOY tool may be run in two steps. The first step, `run_njoy`, write the NJOY input files and calls NJOY. The second step, `read_njoy_output`, reads the mean value of the scalar and transfer cross sections and the covariance matrices associated with the scalar cross sections. A PDT-formatted cross-section file is written for the mean value of the cross sections. If the number of realizations is set to any number besides zero, that many realizations of the cross-section space are generated and written as PDT-formatted data files.

The command to execute the Python tool is `python njoy_driver.py`. If NJOY is executed, NJOY input files are written. A temporary file is created in the location specified with `scratch` with the process ID number. The ENDF data files and NJOY input files are copied to this directory. After NJOY is executed, the NJOY output files are copied from the temporary directory to the `./njoy_step/njoy_outputs/` directory and the temporary directory is deleted.

APPENDIX D

PYTHON TOOL TO RUN UQ ANALYSIS

In this section the Python tool written to generate PDT input decks, run PDT at various points in the cross-section space in both forward and adjoint sensitivity mode, and read the PDT output files to extract the QoI and sensitivity coefficients is described. The Python tool also generates input files for the GEMARS surface C++ code, launches the construction of these surfaces, and processes the surface output files. This tool is relatively specific to the Texas A&M Nuclear Engineering cluster "Cluster," but may prove useful as an idea for a similar tool written for a different platform.

The names of directories and files, as well as options in the Python driver file, will be denoted with in a `distinct font` in this documentation.

D.1 File Organization

Upon unpacking the tarball in which this code is packaged, the directory should contain a README file, a driver titled `uq_driver.py`, and directories `src`, `pdt_step`, and `surface_step`. These directories and files are discussed individually in the following sections.

D.1.1 Documentation File

This Appendix is included in the code tarball as the README file.

D.1.2 The Python Driver, `uq_driver.py`

The UQ driver file contains the user settings for the UQ analysis and scripts to write PDT input decks, submit PDT jobs to a queue, read PDT output files to extract the QoI and the partial derivatives of the QoI with respect to the multigroup cross sections, convert the partial derivatives from the derivative of the QoI with respect to the multigroup cross sections $\frac{dQ}{d\sigma}$ to derivatives with respect to the independent parameters $\frac{dQ}{dp}$, write the input file for the GEMARS surface construction tool, launch the GEMARS calculations, and read the response surface output files.

This tool is very specific to the Texas A&M University cluster "Cluster" and the UQ analysis performed in the course of this work. It may be adapted with minor modification to run PDT calculations on Cluster for different problems of interest, but may not be suitable for a different project. Instead, by examining this code, a user might form an outline for their own script to complete a similar UQ project.

D.1.3 The Directory `src`

The `src` directory contains the Python code that performs each step of the UQ process. This files contained in this directory and a brief explanation of each files' function are listed below.

- `pdt_step`: Contains functions to write PDT input files using realizations of the cross-section space, launch PDT simulations, read the PDT output files to extract the QoI and sensitivity coefficients, read the transform matrices from `transform_mats`, and convert the partial derivatives $\frac{dQ}{d\sigma}$ to $\frac{dQ}{dp}$.

- surface_step: Contains functions to write input files for the GEMARS surface tool, call the surface_launch_script to launch the surface jobs, read the surface output files, and process the surface results.

- surface_launch_script: A script to launch parallel instances of the response surface construction tool. This file is separate from surface_step so that jobs may be submitted to the queue to run this script to launch the surface jobs.

D.1.4 The Directory pdt_step

The directory pdt_step contains a PDT executable and three subdirectories. The names and contents of these directories are listed below.

- pdt.exe: PDT executable. This should be supplied by the user to ensure a modern version of PDT is used.

- data: A directory containing data needed in the processing of the PDT output step. This directory contains two sub-directories: cov_mats and transform_mats. The data for these directories is generated with the Python NJOY tool discussed in Appendix C. This data must be copied to the scratch directory as outlined in Section D.2. The required data is then copied to this directory automatically at run time.

- pdt_inputs: This directory contains a sub-directory named for each PDT problem of interest. This directory name must match the problem name problem_name specified in the user input, as mentioned in Section D.3. Each of these directories must contain one or more PDT input files using the mean value of the cross sections. The mean value of the cross sections are stored in files titled `pdt_xs_MAT.cx`, where the "MAT" is the MAT number of the file. For example, the mean value of the U-235 cross sections are stored in a file titled `pdt_xs_9228.cx`.

Three types of PDT simulations may be run by the Python tool. Each problem requires its own input file. The "uniform" and "normal" designations run an adjoint sensitivity calculation using cross sections generated using uniform and normal distributions of the independent parameters, respectively. The "normal_qoi" designation runs only a forward calculation using cross sections generated using normal distributions of the independent parameters. These input files are titled, respectively, `test_mean_uniform`, `test_mean_normal`, and `test_mean_normal_qoi`.

- `processed_output`: This directory contains the QoI values and sensitivity coefficients read from the PDT output files. The QoI files are stored according to the distribution of the cross sections (`qoi_values_uniform`, etc). The value of the QoI may be useful in the production of histograms and are therefore named carefully. The sensitivity coefficients are stored in a file titled `dqds_values`. The sensitivity coefficients are immediately read and combined into an input file for the surface step and are therefore only stored during an intermediate step.

D.1.5 The Directory `surface_step`

The directory `surface_step` contains the GEMARS response surface tool executable, the surface input files produced by this Python tool, and the surface output files generated by the response surface calculations. A MATLAB tool to visualize the data output by the response surface tool is also included in this step.

- `surface_input_uniform` and/or `surface_input_normal`: A data file generated by the Python tool containing the QoI, sensitivity coefficients $\frac{dQ}{dp}$, and values of the z-parameters, respectively, at each realization of the input space, separated by a blank line. These files are named for the distribution used to generate the cross sections.

- `surface_results_uniform` and/or `surface_results_normal`: A data files generated by the Python tool containing the results from the response surface step. For each number of data points, the mean and variance of each surface is reported, as well as the average number of splines used in the model, average iterations required to converge the model, and the mean and variance in the mean and variance calculated with each surface.

  The end of this file contains a summary of the data in a format for the MATLAB plotting tool. The two sets of data correspond to the raw averages of the response surfaces at each number of sample points and values calculated from the filtered response surfaces.

- `plot_results_with_mc.m`: This MATLAB function plots the screening acceptance ratio and the QoI and variance as a function of the number of samples of the input space. It must be provided with the summary of the response surface data found at the bottom of the `surface_results_uniform` or `surface_results_normal` file. The MATLAB function may also be pointed to values of the QoI generated with a Monte Carlo analysis, stored in the `qoi_values_normal_qoi` file in the `pdt_step/processed_output/` directory. If a Monte Carlo analysis was not performed, this portion of the file should be commented out.

D.2 Prerequisites to Run the UQ Python Tool

Modules and executables required by the Python tool:

- Python-2.7.8.
- MATLAB R2011b (or newer) if plots of the response surface results are desired.

User supplied data and executables:

- PDT input file for each cross-section distribution (normal, uniform, normal_qoi) for which PDT simulations are to be run. For a problem named "example_prob," these are located in `/pdt_step/pdt_inputs/example_prob/example_prob_input_uniform`.
- Appropriate file structure on `/scratch/` to contain PDT run information.
- PDT executable.
- Response surface construction tool executable.

The file structure on `/scratch/` is organized by problem name. For a given problem, the directory with that name must contain a directory for PDT input files, PDT output files, PDT cross-section files, and surface output files. These must exist for each distribution of the independent parameters that are to be run (normal, uniform, and normal_qoi). These files are listed with the required names for the uniform distribution case,

- pdt_input_files_uniform,
- pdt_xs_files_uniform,
- pdt_output_files_uniform,
- surface_output_files_uniform.

The PDT simulations using normal and normal_qoi distributions of the independent parameters both use cross sections stored in the pdt_xs_files_normal directory. Surfaces are not generated for the normal_qoi distribution.

The pdt_input_files, pdt_output_files, and surface_output_files directories are filled when the UQ driver is executed. The pdt_xs_files directories must be filled using data generated by the Python NJOY tool. The cross-section directories must contain:

- The mean value of the cross sections,
- Realizations of the cross sections,
- A directory titled other_data, which contains
  - full_z: the values of the z-parameters at each realization,
  - scat_transfer_removal: the transfer and removal scattering cross sections for each isotope,
  - The directory cov_mats: the covariance matrices for each isotope used in the problem, and
  - The directory transform_mats: the transformation matrices for each

isotope used in the problem.

This data is all generated by the Python NJOY tool but must be moved to scratch and organized in this fashion for each distribution of the z-parameters and each problem of interest.

D.3 User Input

The Python UQ tool is distributed as a tarball. Upon untarring the file the file structure described in Section D.1 should be present. The required user input is a short section in `uq_driver.py` between lines clearly labeled `Begin User Input` and `End User Input.` The input is divided into three categories: specification of which problem is to be run, the UQ steps to be run for this problem, and problem-specific details. These three sets of required inputs are discussed in the following sub-sections.

D.3.1 UQ Problem Specification

The first section of user input contains information about the UQ problem to be run. These inputs are listed here.

- `problem_name`: the problem's name and the name of the directories corresponding to this problem,
- `xs_dist`: the distribution of the z-parameters to be used, either `uniform`, `normal`, or `normal_qoi`,
- `num_simulations`: the number of PDT simulations to be performed,

- `queue_available`: the queue on Cluster that may be used, either `normal`, `large`, or `both`. Access to the large-priority queue must be granted by the Nuclear Engineering IT staff with permission of a professor.

- `num_surfaces`: the number of surface models to be constructed with each number of PDT simulations. Suggested default: 50.

- `filter_strength`: the value of the filter that is applied to the ratio of the variance in a surface to the mean variance across the surfaces in the surface screening step. Suggested default: 10.

D.3.2 UQ Steps To Run

The steps to be run, signified with a `0` or `1` for no and yes, respectively:

- `run_pdt`,

- `process_pdt_output`,

- `run_surface_tool`,

- `process_surface_output`.

The first step, `run_pdt`, must be run by itself and the PDT jobs allowed to finish before the second step, `process_pdt_output`, can be run. If these these two steps are run consecutively with one call to `uq_driver.py`, the PDT jobs will be submitted to Cluster in the first step, and the second step will resubmit any job which has not yet been completed. This will result in each problem being submitted twice to the queue.

A five-second delay is included in the submission of the PDT jobs. If this delay is not included in the submission script the queuing system on Cluster fails. Jobs will move from “pending” to “running” and fill the available resources, but these jobs will not actually start and will never finish. The small delay between job submissions avoids this problem.

D.3.3 PDT Problem Details

Details about the radiation transport problem solved by PDT are supplied in the third section of the input. This section of the input contains information specific to each of the named problems set in the `problem_name` setting. These user settings are:

- `materials`: the name of each isotope in the problem in the naming convention established in the Python NJOY driver, Section C,
- `mat_nums`: the MAT numbers of each isotope in the problem,
- `mat_file_names`: the MAT numbers or the MAT number followed by a “_” and special material, for example, `H-1_poly` or `600_graphite`,
- `num_groups`: the number of energy groups used in the multigroup cross sections,
- `num_moments`: the Legendre order of the multigroup cross sections,
- `scratch_location`: the location of the directory containing the required file structure and PDT cross-section files described in Section D.2.

With these settings specified the UQ driver may be run.

D.4 Code Execution

The command to launch the Python UQ driver is `python uq_driver.py`.

The UQ driver is executed in four parts. These steps may be executed independently as long as the proceeding steps have been completed. These four steps are:

- Writing PDT input decks and launching the PDT simulations, `run_pdt`,
- Reading the PDT output files to extract the QoI and sensitivity coefficients, converting the sensitivity coefficients from the cross sections to the independent parameters, and writing an input file for the response surface calculations, `process_pdt_output`. If any PDT simulations are missing, those simulations are relaunched and the rest of this step is aborted.
- Launching response surface calculations, `run_surface_tool`, and
- Reading the response surface results, compiling the data, and filtering the results to screen for overfitting of the data, `process_surface_output`.

A report on the mean, variance, number of important parameters, contribution of each isotope to the total variance, and contribution of each important parameter to the total variance is printed to the screen at the end of the `process_pdt_output` step. The response surface input file is also generated in this step.

Results from the response surfaces are printed to an output file `/surface_step/surface_results_uniform` or `/surface_step/surface_results_normal`, depending upon the distribution of the z-parameters, at the completion of the `process_surface_output` step.

D.5 PDT Simulations on Vulcan

This tool was used to prepare to run and process PDT simulations on the supercomputer "Vulcan" at Lawrence Livermore National Laboratory. The UQ tool was used in the normal fashion to generate PDT input files, but the files were then moved to Vulcan and a separate script used to submit the jobs to the queuing system on Vulcan. The PDT output files were moved back to Cluster from Vulcan and processed using this tool. Response surfaces were then constructed with Cluster in the normal fashion.

PDT simulations performed using a large number of cores should have the QoI and sensitivity coefficients output to the `slurm` files produced for each run. Although this data makes the `slurm` files relatively large, it is much more efficient to copy a `slurm` to Cluster from Vulcan than it is to copy an output file containing a portion of the angular flux solution to Cluster from Vulcan. The `slurm` files require very minor processing before they can be handled with this tool; each line begins with a >> symbol that must be stripped before this tool can process the file to extract the QoI and sensitivity coefficients.

APPENDIX E

C++ GEMARS TOOL

The C++ tool used to construct Gradient-Enhanced Multivariate Adaptive Regression Spline (GEMARS) models of the QoI is explained in detail in Section 4.5.2 of this dissertation. This section contains an overview of the file structure, compilation instructions, and command-line options used to run this code.

E.1 File Organization

The C++ GEMARS surface construction tool is distributed as a tarball. The file structure within this tarball is outlined in this section.

E.1.1 Documentation File

This Appendix is included as a README file.

E.1.2 The Directory `src`

The directory `src` contains one C++ library in its own directory and the source code for the tool. This directory contains an external library `Eigen` which is used to solve the least-squares problem, a CMAKE file `CMakeLists` used in the compilation of the code, and the files containing the C++ code. These files and their content are listed below.

- `create_surface`: Driver file for GEMARS surface algorithm.

- `read_data_files`: Header (.h) and code (.cc) files to read the surface input file and organize the data.

- `model_class`: Header file and code files operations involving the model.

- `spline_class`: Header and code files for the operations involving the splines.

E.1.3 The Directory `test_response_surface`

This directory contains a MATLAB script `make_analytic_input.m` to generate an input file corresponding to the "Additive Surface" test problem described in [13] and solved in the test problem examined in Section 5.1.1. The results for this test problem are plotted with MATLAB with the `plot_results_with_mc.m` code. Various figures visualizing the test surface, the latin hypercube distribution used to sample the surface, and the surface performance as a function of the number of data points are included in this directory.

E.2 Compilation Instructions

A bash script is included in the distribution to compile the GEMARS code. The command to execute this command is `./create_exe`. This command generates the executable `ResponseSurface.exe`.

E.3 Runtime Instructions

Given a surface input file generated with the Python UQ function `uq_driver.py`, a response surface may be generated with the command `./ResponseSurface.exe` and the mandatory command line arguments. This process is automated in the

Python UQ driver function. The available command line arguments and their defaults are listed here.

- `-i`: path to and name of the input file generated with the Python UQ driver. This option must be specified.
- `-o`: path to and name of the output file to be generated with the response surface tool. This option must be specified.
- `-l`: maximum spline order. Default value: 1.
- `-c`: maximum spline connectivity. Default value: 1.
- `-n`: number of points in the input space used to construct the surface. Must be less than or equal to the number of points in the input file. This option must be specified.
- `-e`: convergence tolerance. Default value: 0.05.

An example call of the GEMARS surface tool from the directory containing the executable, with linear splines, a maximum connectivity between dimensions of three, an error tolerance of 0.05, and using ten data points would appear

```
./ResponseSurface.exe -i surface_input_uniform
-o surface_output_uniform_iter_01 -o 1 -c 3 -e 0.05 -n 10
```

APPENDIX F

SELECTED ENERGY GROUP STRUCTURES

In this appendix several energy group structures used in the problems in this work and not available through NJOY are recorded.

F.1 CERT 99-Group Energy Structure

Group	**Lower Bound**	**Upper Bound**
1	1.00000E-05	1.82202E-03
2	1.82202E-03	4.97106E-03
3	4.97106E-03	6.18000E-03
4	6.18000E-03	7.28878E-03
5	7.28878E-03	8.61545E-03
6	8.61545E-03	9.06000E-03
7	9.06000E-03	1.14000E-02
8	1.14000E-02	1.35465E-02
9	1.35465E-02	1.36000E-02
10	1.36000E-02	1.53687E-02
11	1.53687E-02	1.55000E-02
12	1.55000E-02	1.74000E-02
13	1.74000E-02	1.92000E-02
14	1.92000E-02	2.08353E-02
15	2.08353E-02	2.10000E-02
16	2.10000E-02	2.28000E-02
17	2.28000E-02	2.45000E-02
18	2.45000E-02	2.63000E-02

19	2.63000E-02	2.80000E-02
20	2.80000E-02	2.97000E-02
21	2.97000E-02	2.99463E-02
22	2.99463E-02	3.15000E-02
23	3.15000E-02	3.33000E-02
24	3.33000E-02	3.51000E-02
25	3.51000E-02	3.70000E-02
26	3.70000E-02	3.89000E-02
27	3.89000E-02	4.08000E-02
28	4.08000E-02	4.28000E-02
29	4.28000E-02	4.29973E-02
30	4.29973E-02	4.49000E-02
31	4.49000E-02	4.70000E-02
32	4.70000E-02	4.93000E-02
33	4.93000E-02	5.16000E-02
34	5.16000E-02	5.40000E-02
35	5.40000E-02	5.66000E-02
36	5.66000E-02	5.93000E-02
37	5.93000E-02	6.22000E-02
38	6.22000E-02	6.53000E-02
39	6.53000E-02	6.87000E-02
40	6.87000E-02	7.23000E-02
41	7.23000E-02	7.64000E-02
42	7.64000E-02	8.09000E-02
43	8.09000E-02	8.60000E-02
44	8.60000E-02	9.20000E-02
45	9.20000E-02	9.92000E-02
46	9.92000E-02	1.08000E-01
47	1.08000E-01	1.21000E-01

48	1.21000E-01	1.42000E-01
49	1.42000E-01	1.84000E-01
50	1.84000E-01	2.38000E-01
51	2.38000E-01	3.09000E-01
52	3.09000E-01	4.00000E-01
53	4.00000E-01	5.62341E-01
54	5.62341E-01	1.00000E+00
55	1.00000E+00	1.77828E+00
56	1.77828E+00	3.16228E+00
57	3.16228E+00	5.62341E+00
58	5.62341E+00	1.00000E+01
59	1.00000E+01	1.77828E+01
60	1.77828E+01	3.16228E+01
61	3.16228E+01	5.62341E+01
62	5.62341E+01	1.00000E+02
63	1.00000E+02	1.77828E+02
64	1.77828E+02	3.16228E+02
65	3.16228E+02	5.62341E+02
66	5.62341E+02	1.00000E+03
67	1.00000E+03	1.77828E+03
68	1.77828E+03	3.16228E+03
69	3.16228E+03	5.62341E+03
70	5.62341E+03	1.00000E+04
71	1.00000E+04	1.77828E+04
72	1.77828E+04	3.16228E+04
73	3.16228E+04	5.62341E+04
74	5.62341E+04	1.00000E+05
75	1.00000E+05	1.77828E+05
76	1.77828E+05	3.16228E+05

77	3.16228E+05	5.62341E+05
78	5.62341E+05	1.00000E+06
79	1.00000E+06	2.07016E+06
80	2.07016E+06	2.07831E+06
81	2.07831E+06	2.08769E+06
82	2.08769E+06	2.81313E+06
83	2.81313E+06	2.81512E+06
84	2.81512E+06	2.86312E+06
85	2.86312E+06	2.97494E+06
86	2.97494E+06	3.08125E+06
87	3.08125E+06	4.00230E+06
88	4.00230E+06	4.80107E+06
89	4.80107E+06	4.94256E+06
90	4.94256E+06	5.35484E+06
91	5.35484E+06	5.43000E+06
92	5.43000E+06	6.27650E+06
93	6.27650E+06	6.33875E+06
94	6.33875E+06	6.57250E+06
95	6.57250E+06	7.24687E+06
96	7.24687E+06	7.71093E+06
97	7.71093E+06	8.25392E+06
98	8.25392E+06	1.40000E+07
99	1.40000E+07	1.42000E+07

Table F.1: Energy group structure for the 49-group CERT group structure. Group bounds presented in eV.

F.2 CERT 49-Group Energy Structure

Group	**Lower Bound**	**Upper Bound**
1	1.00E-05	1.82E-03
2	1.82E-03	4.97E-03
3	4.97E-03	1.14E-02
4	1.14E-02	2.10E-02
5	2.10E-02	3.33E-02
6	3.33E-02	5.50E-02
7	5.50E-02	8.00E-02
8	8.00E-02	1.08E-01
9	1.08E-01	1.50E-01
10	1.50E-01	2.00E-01
11	2.00E-01	3.09E-01
12	3.09E-01	4.00E-01
13	4.00E-01	5.62E-01
14	5.62E-01	1.00E+00
15	1.00E+00	1.78E+00
16	1.78E+00	3.16E+00
17	3.16E+00	5.62E+00
18	5.62E+00	1.00E+01
19	1.00E+01	3.16E+01
20	3.16E+01	1.00E+02
21	1.00E+02	3.16E+02
22	3.16E+02	1.00E+03
23	1.00E+03	3.16E+03
24	3.16E+03	1.00E+04
25	1.00E+04	3.16E+04
26	3.16E+04	1.00E+05
27	1.00E+05	3.16E+05

28	3.16E+05	1.00E+06
29	1.00E+06	2.07E+06
30	2.07E+06	2.08E+06
31	2.08E+06	2.09E+06
32	2.09E+06	2.81E+06
33	2.81E+06	2.82E+06
34	2.82E+06	2.86E+06
35	2.86E+06	2.97E+06
36	2.97E+06	3.08E+06
37	3.08E+06	4.00E+06
38	4.00E+06	4.80E+06
39	4.80E+06	4.94E+06
40	4.94E+06	5.35E+06
41	5.35E+06	5.43E+06
42	5.43E+06	6.28E+06
43	6.28E+06	6.34E+06
44	6.34E+06	6.57E+06
45	6.57E+06	7.25E+06
46	7.25E+06	7.71E+06
47	7.71E+06	8.25E+06
48	8.25E+06	1.40E+07
49	1.40E+07	1.42E+07

Table F.2: Energy group structure for the 99-group CERT group structure. Group bounds presented in eV.

F.3 Reactor (C5G7) 7-Group Energy Structure

Group	**Lower Bound**	**Upper Bound**
1	1.00E-05	1.30E-01
2	1.30E-01	6.30E-01
3	6.30E-01	4.10E+00
4	4.10E+00	5.56E+01
5	5.56E+01	9.20E+03
6	9.20E+03	1.36E+06
7	1.36E+06	2.00E+07

Table F.3: Energy group structure for the 7-group reactor group structure. Group bounds presented in eV.