ABSTRACT

KENNEDY, CHRISTOPHER BRANDON. GPT-Free Sensitivity Analysis for Reactor Depletion and Analysis. (Under the direction of Dr. Hany S. Abdel-Khalik).

Introduced in this dissertation is a novel approach that forms a reduced-order model (ROM), based on subspace methods, that allows for the generation of response sensitivity profiles without the need to set up or solve the generalized inhomogeneous perturbation theory (GPT) equations. The new approach, denoted hereinafter as the generalized perturbation theory free (GPT-Free) approach, computes response sensitivity profiles in a manner that is independent of the number or type of responses, allowing for an efficient computation of sensitivities when many responses are required. Moreover, the reduction error associated with the ROM is quantified by means of a Wilks' order statistics error metric denoted by the κ-metric.

Traditional GPT has been recognized as the most computationally efficient approach for performing sensitivity analyses of models with many input parameters, e.g. when forward sensitivity analyses are computationally overwhelming. However, most neutronics codes that can solve the fundamental (homogenous) adjoint eigenvalue problem do not have GPT (inhomogenous) capabilities unless envisioned during code development. Additionally, codes that use a stochastic algorithm, i.e. Monte Carlo methods, may have difficult or undefined GPT equations. When GPT calculations are available through software, the aforementioned efficiency gained from the GPT approach diminishes when the model has both many output responses and many input parameters. The GPT-Free approach addresses these limitations, first by only requiring the ability to compute the fundamental adjoint from perturbation theory, and second by constructing a ROM from fundamental adjoint calculations,

constraining input parameters to a subspace. This approach bypasses the requirement to perform GPT calculations while simultaneously reducing the number of simulations required.

In addition to the reduction of simulations, a major benefit of the GPT-Free approach is explicit control of the reduced order model (ROM) error. When building a subspace using the GPT-Free approach, the reduction error can be selected based on an error tolerance for generic flux response-integrals. The GPT-Free approach then solves the fundamental adjoint equation with randomly generated sets of input parameters. Using properties from linear algebra, the fundamental *k*-eigenvalue sensitivities, spanned by the various randomly generated models, can be related to response sensitivity profiles by a change of basis. These sensitivity profiles are the first-order derivatives of responses to input parameters. The quality of the basis is evaluated using the κ-metric, developed from Wilks' order statistics, on the user-defined response functionals that involve the flux state-space. Because the κ-metric is formed from Wilks' order statistics, a probability-confidence interval can be established around the reduction error based on user-defined responses such as fuel-flux, max-flux error, or other generic inner products requiring the flux. In general, The GPT-Free approach will produce a ROM with a quantifiable, user-specified reduction error.

This dissertation demonstrates the GPT-Free approach for steady state and depletion reactor calculations modeled by SCALE6, an analysis tool developed by Oak Ridge National Laboratory. Future work includes the development of GPT-Free for new Monte Carlo methods where the fundamental adjoint is available. Additionally, the approach in this dissertation examines only the first derivatives of responses, the response sensitivity profile; extension and/or generalization of the GPT-Free approach to higher order response sensitivity profiles is natural area for future research.

GPT-Free Sensitivity Analysis for Reactor Depletion and Analysis

by
Christopher Brandon Kennedy

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Nuclear Engineering

Raleigh, North Carolina

2015

APPROVED BY:

Dr. Hany S. Abdel-Khalik Committee Chair	Dr. Cristian Rabiti
Dr. Yousry Y. Azmy	Dr. Ralph C. Smith

DEDICATION

This dissertation is dedicated to my mother.

BIOGRAPHY

Christopher Brandon Kennedy (Chris) was born March 18th, 1985. He has one older brother of seven years. Shortly after Chris was born in Whittier, CA, his family relocated first to Aiken, SC and again later to Raleigh, NC. He received his B.S. of Nuclear Engineering from North Carolina State University in May of 2009. He promptly joined Dr. Hany-Abdel-Khalik's research group as a research assistant in Nuclear Engineering.

ACKNOWLEDGMENTS

First and foremost, I'd like to acknowledge my family for supporting and believing in me to reach the academic milestone of a doctorate of philosophy. Similarly, without the rigorous motivation, support, and opportunity provided by my academic advisor, Dr. Hany Abdel-Khalik, a Ph.D. would not have been possible. He pushed me to succeed and do my best even at great personal sacrifice. To my committee members, Dr. Abdel-Khalik, Dr. Azmy, Dr. Rabiti, and Dr. Smith, thank you for taking the time to work with me on the path to graduation. I would like to also take the time to recognize my professors in my graduate career: Dr. Turinsky, Dr. Anistratov, Dr. Gardner, Dr. Eapen, Dr. Shannon, Dr. Hawari, Dr. Doster, Dr. Bourham, Dr. Wehring, and Dr. Mattingly. I'd also like to thank Hermine, Lisa, and Ganga for their assistance during my time at NCSU. Last but not least, I would like to acknowledge the support of my peer graduate students at NCSU.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

PWR	Pressurized Water Reactor
BWR	Boiling Water Reactor
HTGR	High Temperature Gas Reactor
PT	Perturbation Theory
GPT	Generalized Perturbation Theory
FF	Fission Fragment
DOE	Department of Energy
XS	Cross-Section
σ	Microscopic Cross-Section
Σ	Macroscopic Cross-Section
k	Multiplication Eigenvalue
λ	$1/k$ (System Eigenvalue)
FA	Fuel Assembly
UQ	Uncertainty Quantification
DA	Data Assimilation
DO	Design Optimization
V&V	Verification and Validation
ROM	Reduced Order Model
SA	Sensitivity Analysis
T-H	Thermal-Hydraulics
SVD	Singular Value Decomposition

QR	QR-Decomposition
RFA	Range-Finding Algorithm
E	Kinetic Energy (neutron)
N	Volumetric Atomic Number Density ($atoms \cdot cm^{-3}$)
r	Matrix Rank
m, n	Matrix and Vector Dimensions
ρ	Phase Space (Volume, Energy, etc.)
ρ_k	Reactivity $\left((k-1)/k\right)$
ρ_x	Density ($grams \cdot cm^{-3}$)
N_a	Avogadro's Number ($atoms \cdot mol^{-1}$)
A	Atomic Mass ($grams \cdot mol^{-1}$)
cc	cm^3
BOC	Beginning of Cycle
EOC	End of Cycle
U	Uranium
Pu	Plutonium

1. INTRODUCTION

1.A Background and Motivation

Nuclear reactor design calculations are a computationally intensive endeavor because of but not limited to the physical complexity of reactor models, the extensive set of coupled physics equations, and the variable time-scales of nuclear phenomena. Iteration of reactor design requires repeated execution of software that takes not only numerous computer hours but also many engineering-hours. These iterations may often be encapsulated by an optimization problem: maximizing the economic potential of a very large complicated physical system while simultaneously meeting safety constraints across several physics (e.g. materials, fluids, neutronics, heat transfer, systems, etc.). When considering safety constraints or other design criteria, an uncertainty analysis can be used to estimate response uncertainties due to the propagation of input parameter uncertainties through the model. Because of the multi-scale approach to modeling with the coupling of several physics, a deterministic solution of response uncertainties is problematic to determine without repeated simulation. Additionally, a nuclear scientist may wish to improve the accuracy of the fundamental nuclear cross-sections to improve the accuracy of existing software through a technique known as data assimilation; however, such a technique requires knowledge of the relationship between input parameters and output responses, which for the same reasons as the uncertainty analysis are difficult to obtain in closed-form. For these three approaches, optimization, uncertainty analysis, and data assimilation, a sensitivity analysis is completed to determine the relationship between input parameters and output responses. The sensitivity

analysis aids the nuclear engineer in understanding the relationships among the input parameters, e.g. cross-sections, geometry, etc., and output responses, e.g. fluxes, reaction rates, power-densities, etc. Depending on the nuclear reactor model, there exist millions of input parameters and output responses; design decisions quickly become problematic without knowledge of model sensitivities. While underlying physical phenomena may be very complex, a first-order sensitivity analysis may simplify the relationship by calculating the first derivative of output responses to input parameters. In general, higher-order derivatives can be determined; however, the curse of dimensionality makes higher-order sensitivity analysis significantly more expensive. Regardless of the sensitivity analysis order, the objective is to speed up engineering design calculations by taking advantage of the correlations among output responses, thereby enabling faster computation of sensitivities using existing nuclear modeling software.

Accurate modeling of nuclear reactor systems requires the processing of millions of input parameters and output responses due to the heterogeneity of a reactor core and the complexity of neutron interactions. A full-core 3D heterogeneous model coupling all interacting physics is intractable because of the multi-scale, from nanometers to meters, and multi-physics, e.g. neutronics and fluids, nature of the problem. For example, if a reactor model were spatially discretized to a fraction of a fuel rod radius, with tens of thousands of fuel rods in a reactor, there would exist hundreds of millions of spatial regions; when including a coarse energy grid – neutron energies span orders of magnitude in a nuclear reactor – the state-space easily exceeds billions of regions. Even with today's available computing power, billions of discretization regions are not practically solved; the problem

must be broken down strategically, e.g. fuel pins and assemblies, to preserve accuracy while reducing computational overhead. The process of breaking up design calculations into a sequence that addresses the various scales of the underlying physics, e.g. multi-scale modeling, in nuclear engineering requires homogenization theory. The homogenization process, while making the calculations feasible, may still be computationally expensive and is still iterative. Any process or approach that can reduce the number of steps to solve a reactor design calculation is computationally favorable.

Since 1945, nuclear scientists have successfully employed perturbation theory (PT) to efficiently calculate *k*-eigenvalue sensitivity profiles; generalized perturbation theory (GPT), developed shortly after PT, expands on the theory, describing the first order derivatives of generic output responses with respect to input parameters[†]. Both PT and GPT are general mathematical tools applied to nuclear engineering to compute sensitivity profiles. By forming the adjoint neutronics equations, the number of calculations required to compute sensitivities is significantly reduced. More specifically, GPT formulates the response and region specific adjoint model, the solution of which can be used in conjunction with the solution to the forward model to directly calculate the sensitivities of a given response with respect to all input parameters via inner product relations. Therefore, the computational cost of evaluating sensitivities for a model with m responses and n input parameters reduces to m adjoint evaluations and 1 forward evaluation as opposed to $n + 1$ forward evaluations with a forward sensitivity approach. When m is much less than n, the adjoint approach becomes superior; however, when both m and n are large, there may not be a clear advantage to using the GPT

[†] Generic output responses must still be *allowable* under GPT theory. See Williams 1986 for additional details.

approach over the forward approach. Because of these advantages when the responses of interest are far fewer than the number of input parameters, PT and GPT have been employed to solve a variety of analysis problems such as: control rod worth, fuel cycle optimization, reactivity coefficients, boundary condition analysis, and sensitivity analysis.

Despite the widespread use and significant power of GPT, it has several well-known limitations. First, the ability to compute general response sensitivities requires the capability to form and solve the GPT adjoint equations. The GPT adjoint equations are similar in form to the forward equations (in the form of a generalized eigenvalue problem) but they require an inhomogeneous source definition based on the response of interest. For deterministic neutronics models, the adjoint equations can be crudely described by the transpose matrix operator of the forward model. When constructing the adjoint for multi-physics models, not only must the boundary conditions and source terms be carefully considered, but also custom software for the multi-physics models will be required to solve the GPT equations. For existing codes that lack these features, the cost of implementation can be expensive and/or prohibitive. For probabilistic models, an adjoint model is set up by reversing the track of neutrons. While this approach has been demonstrated for source driven problems, there currently exists no general theory on how to implement an adjoint model for eigenvalue problems. Second, typical adjoint simulations in PT and GPT problems require significantly more computational effort compared to the forward simulation; one forward run is not equivalent to one adjoint run[‡]. As the number of output responses of interest increases, the GPT approach suffers from the same problem as forward simulations for large models with

[‡] The actual difference is problem dependent; the adjoint typically took 2-3x longer than a forward run for the models examined in this dissertation.

many output responses, *m,* and many input parameters, *n*. Because nuclear cross-sections span many orders of energy and are reaction, isotope, and temperature dependent, the input parameter space can quickly grow as computational power and data accuracy allows. Output responses depend on the state-space, which is lower dimensional than the input parameter space; however, as the state-space, i.e. flux, is refined in energy and in space, through either a decrease in homogenization or an increase in mesh refinement, the output parameter dimension can begin to make GPT approach computationally intractable.

The objective of this dissertation is to address the limitations of GPT by allowing for GPT calculations when either a) the GPT calculation is not possible or b) when the number of GPT responses is large and computationally intractable. The goal is to allow for wider-spread usage of sensitivity profiles in engineering design and optimization, uncertainty analysis, and data assimilation. More specifically, the objectives are as follows:

1. Develop an approach to efficiently compute response sensitivities without formation or solution of the GPT equations using only the fundamental adjoint from PT.
2. Quantify the reduction error from reduced order models used in this approach.
3. Quantify the error in response sensitivities determined using this approach.
4. Couple depletion and neutronics for depletion perturbation theory GPT problems.

This dissertation meets these objectives by means of a new hybrid approach, denoted generalized perturbation theory free (GPT-Free), to sensitivity analysis. The first problem addressed is the ever-increasing model dimensionality in both input parameters and output responses. The hybrid approach attempts to form a reduced-order model (ROM) using

information from both the input parameters and the output responses. The ROM forms a problem-specific basis of lower dimension that allows one to more quickly perform a forward sensitivity analysis. Additionally, construction of the ROM can be set such that the model reduction error is below a quantifiable user-specified threshold. In summary, the ROM is to allow for a more computationally efficient forward sensitivity analysis. The second problem addressed is when the construction of GPT equations is not feasible; this happens when the software source is unavailable or formulation of the GPT equations is not amenable e.g. Monte Carlo models or complex multi-physics problems. The hybrid approach allows for the calculation of the response sensitivities using only the fundamental (homogenous) adjoint as opposed to the GPT response-specific adjoint. As proof of principle, this approach will be applied to deterministic models to evaluate responses sensitivities with respect to the multi-group cross-sections.

A statistical metric introduced in this dissertation, denoted the κ-metric, meets the objectives for quantifying the reduction error from using the GPT-Free ROM as well as an error estimate for response sensitivities calculated using the GPT-Free approach. One difficulty with measuring the effectiveness of a ROM is quantifying the error introduced by using the ROM, because the error may vary depending on parameters of the model. The κ-metric was designed to address this problem by implementing Wilks' order statistics in order to quantify the error with a probability/confidence interval. The κ-metric gives a probability p that a given ROM reduction error is met with confidence c given N samples of the ROM error. The metric is flexible in that the error can be tuned for the ROM or response error of interest.

This dissertation also expands on the GPT-Free approach by including a method to address depletion, a multi-physics problem. Depletion problems offer additional complexity because the reactor is no longer at steady-state due to burn-up of the reactor fuel. Current adjoint methodologies require specialized adjoint calculations at each time step for each response. The GPT-Free approach has been extended to consider depletion without requiring specialized generalized adjoint calculations.

1.B Document Organization

The dissertation is organized into seven major sections, each containing information to help the engineer or scientist better understand and apply the GPT-Free approach. The first section includes the background and motivation, organization of the document, and foundations for the mathematical notation used throughout the dissertation. The primary focus of this section is to demonstrate the importance of gradients in optimization problems common to nuclear engineering reactor calculations. While perturbation theory and generalized perturbation theory have been successfully used to find gradient information, they suffer from limitations for high-dimensional models and software packages that lack GPT capabilities. The GPT-Free approach is then introduced as a method to use existing perturbation theory technology to bypass the limitations of GPT.

The second section provides a brief overview of nuclear reactor calculations for the non-nuclear engineer. A brief overview of nuclear cross-sections, a fundamental nuclear parameter, and reactor calculations is included. The importance of the reactor calculation steps assists with understanding the level of calculation involved in a reactor model.

A review of the literature for the underlying methodologies for generalized perturbation theory, perturbation theory, Monte Carlo transport, and reduced order models is included in section three. Literature for both GPT and PT are explored and then narrowed down to the nuclear engineering field specific uses. Background equations for PT and GPT are developed to illustrate the additional work required to develop GPT formalisms in software. The Monte Carlo adjoint section is included to demonstrate the difficulty of developing a Monte Carlo transport generalized adjoint, especially with sufficient statistics. Lastly, because the GPT-Free approach ultimately develops a reduced order model, a brief overview of reduced order models and their applications to nuclear problems is included for completeness.

Section four begins the introduction of the GPT-Free approach applied to reactor problems, including details on the theory, formulation, the κ-metric, and studies of the method applied to different particular problems. The κ-metric addresses the problems with quantifying reduction errors for reduced order models and is not restricted to the GPT-Free approach. Furthermore, the effects of numerical inaccuracies such as mesh sizes, and geometry considerations for assemblies are considered while using the GPT-Free approach.

The fifth section builds on the fourth by extending GPT-Free to depletion and Monte Carlo problems by introducing additional theory, examples, and equations. Demonstration of the theory is included with some numerical examples to set the foundation for the theory. The formalism is then expanded to include an algorithm to apply the GPT-Free approach to a depletion model. The Monte carlo models section is based on Kiedrowski's continuous *k*-eigenvalue sensitivity calculation (Kiedrowski 2010).

The numerical demonstrations and benchmarks of the GPT-Free approach using the UAM depletion model and the TMI mini-core reactor model are included in the sixth section. This section includes benchmarks for a fast-metal sphere, JEZEBEL, the HTGR reactor lattice (Dehart 2009), The UAM depletion model (Ivanov 2007), and the TMI 9 assembly mini-core reactor model derived from the TMI-1 UAM 15x15 assembly model (Ivanov 1999). Sample calculations involve determination of the ROM dimension, reduction error introduced using the ROM, sample sensitivity profiles, and sample sensitivity demonstrations using the GPT-Free approach.

The last sections are reserved for conclusions, future work, references and appendices. These sections wrap up details and comments on the GPT-Free approach, feasibility, areas for improvement and opportunities, and limitations. Included in the dissertation appendices are notes on the different software implemented (e.g. SCALE6, PYTHON, MATLAB, etc.) and the various algorithms generated to benefit from high-performance computing clusters.

1.C Mathematical Notation

This section specifies the notation used for mathematical quantities in this dissertation. Unless otherwise specified, all calculations are completed with real numbers and are finite-dimensional. Fundamental linear algebra concepts explained in this section of the dissertation can be studied further in ref. Meyer, 1999.

Matrices and operators are denoted by bold-faced capital letters and capital Greek letters, e.g. $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a matrix with m rows and n columns. Similarly, $\Gamma \in \mathbb{R}^{p \times q}$ is a matrix

with p rows and q columns. Operators are understood when the dimension is not explicitly defined. Elements of a matrix are denoted by two indices, the first indicating the row, and the second indicating the column, e.g. $\mathbf{A}_{ij}$ denotes row i and column j. A vector of a matrix is denoted with an asterisks replacing the general dimension, e.g. $\mathbf{A}_{i*}$ denotes row i of $\mathbf{A}$ and $\mathbf{A}_{*j}$ denotes column j of $\mathbf{A}$. A single subscript on a matrix denotes a named subset of the matrix which will be explicitly defined when used, e.g. $\mathbf{A}_p = \left[\mathbf{A}_{1*} \ldots \mathbf{A}_{p*}\right]$. When the adjoint of a matrix or operator is considered, the matrix will be denoted with a superscript asterisks, e.g. $\mathbf{A}^*$. This is not to be confused with the complex matrix transpose as the field of numbers is reals.

Vectors and scalars are denoted by lower-case letters. Scalars are considered to be vectors of dimension one, e.g. $\sigma \in \mathbb{R}^n$ is a vector of length n and $\alpha \in \mathbb{R}$ is a vector of length one. By default, all vectors are column vectors unless otherwise specified such that the inner product $x^T x$ results in a scalar result, e.g. $x^T x = \sum_{i=1}^{n} x_i^2$ where $x \in \mathbb{R}^n$. Elements of vectors are denoted by an index, e.g. x_i. When the adjoint of a vector is considered, the vector will be denoted with a superscript asterisks, e.g. ϕ^*.

Inner products between two vectors x and y are denoted with the transpose notation, e.g. $x^T y$. For more complicated or generic inner-products involving matrices and vectors, the bracket notation may be used to improve clarity, e.g. see Eq. 1.

$$\left\langle x, \left(\mathbf{A} - \lambda \mathbf{B}\right) y \right\rangle \equiv \left(x^T \mathbf{A} y\right) - \lambda \left(x^T \mathbf{B} y\right) \quad \lambda \in \mathbb{R}^1 \tag{1}$$

Norms are defined with the double bar notation, $\|\cdot\|$, where the definition of the norm is typically defined in the problem context. When no definition is provided, the default is the l^2 norm of vector defined as $\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2} = \sqrt{x^T x}$ where $x \in \mathbb{R}^n$. For matrices, the l^2 norm is an induced matrix norm.

Orthogonal and parallel components of vectors are based on projections and orthogonal decompositions. Given an orthonormal basis **Q**, a vector x can, by projection, be broken apart into orthogonal and parallel components. For example, see Eq. 2 where x is broken into a component orthogonal to **Q** and parallel to **Q**. A visual illustration of this concept is shown in Fig. 1.

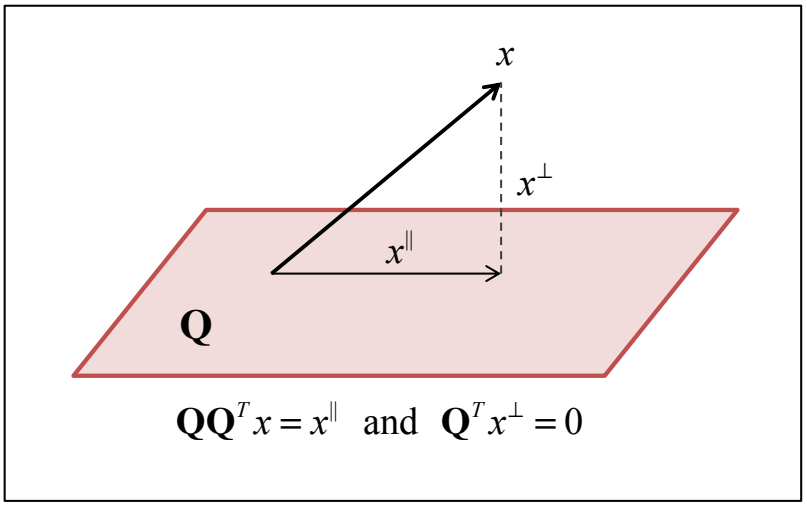

Figure 1: Projection of x onto the subspace represented by Q.

$$x = \underbrace{\left(\mathbf{Q}\mathbf{Q}^T\right)x}_{x^{\|}} + \underbrace{\left(\mathbf{I} - \mathbf{Q}\mathbf{Q}^T\right)x}_{x^{\perp}} \tag{2}$$

Matrices can be decomposed into their orthogonal components. The four spaces of a matrix are the column-space (range), row-space, null-space, and left null-space which are available through various orthogonal decompositions. For illustrative purposes, the singular value decomposition (SVD) of a matrix breaks a matrix down into these four spaces.

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T \quad \mathbf{A} \in \mathbb{R}^{m\times n} \quad \mathbf{U} \in \mathbb{R}^{m\times m} \quad \mathbf{V} \in \mathbb{R}^{n\times n} \quad \Sigma \in \mathbb{R}^{m\times n} \tag{3}$$

Where for a matrix $\mathbf{A}$ of rank r the spaces, from Eq. 3 are as follows:

$$\begin{aligned} \mathbf{U}_r &= \left[\mathbf{U}_{*1} \ldots \mathbf{U}_{*r}\right] && \text{Range of } \mathbf{A} \text{ or R } (\mathbf{A}) \\ \mathbf{U}_m &= \left[\mathbf{U}_{*(r+1)} \ldots \mathbf{U}_{*m}\right] && \text{Left null-space of } \mathbf{A} \text{ or N } \left(\mathbf{A}^T\right) \\ \mathbf{V}_r &= \left[\mathbf{V}_{*1} \ldots \mathbf{V}_{*r}\right] && \text{Row-space of } \mathbf{A} \text{ or R } \left(\mathbf{A}^T\right) \\ \mathbf{V}_n &= \left[\mathbf{V}_{*(r+1)} \ldots \mathbf{V}_{*n}\right] && \text{Null-space of } \mathbf{A} \text{ or N } \left(\mathbf{A}\right) \end{aligned} \tag{4}$$

These spaces of the matrix denote important properties on solutions of matrix-vector products. Shown in Eq. 5 are important properties related to the null-space and left null-space of a general matrix $\mathbf{A}$.

$$\begin{aligned} \mathbf{A}x = 0 \quad & \text{if } x \in \mathcal{N}\left(\mathbf{A}\right) \\ \mathbf{A}^T y = 0 \quad & \text{if } y \in \mathcal{N}\left(\mathbf{A}^T\right) \end{aligned} \tag{5}$$

Additionally, the spaces are complements of one another:

$$\mathcal{R}\left(\mathbf{A}\right) \perp \mathcal{N}\left(\mathbf{A}^T\right) \quad \text{and} \quad \mathcal{R}\left(\mathbf{A}^T\right) \perp \mathcal{N}\left(\mathbf{A}\right) \tag{6}$$

The inverse and pseudo-inverse (Moore-Penrose inverse) of a matrix are denoted distinctly in this dissertation. The generic inverse is the strict square-matrix definition such that right or left-multiplication yields the square identity matrix, e.g. $\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$, $\left\{\mathbf{A},\mathbf{I}\right\} \in \mathbb{R}^{n\times n}$. The pseudo-inverse operates on rectangular matrices, and is

often used in least-squares solutions, e.g. $x = \mathbf{A}^{\dagger} b,\ \mathbf{A} \in \mathbb{R}^{m \times n}, x \in \mathbb{R}^{n}, b \in \mathbb{R}^{m}$, and $m > n$. The pseudo-inverse varies for left and right multiplication depending on the structure of **A**.

The letters *m*, *n*, *r*, *i*, and *j* are reserved for indices and dimensions. For matrix quantities *m* typically denotes the numbers of rows and *n* indicates the number of columns. Likewise *m* and *n* may represent the number of entries in a vector. The indices *i*, and *j* are reserved for the specification of elements in both vectors and matrices. The letter *r* is reserved for the rank of a matrix, often representing the dimension of a related subspace.

Uppercase roman letters are reserved for special notations, typically special quantities such as atomic densities (*N*), energy groups (*G*), or upper limits on iteration indices. From the notation, these values are explicitly defined.

Macroscopic cross-sections are denoted with a capital sigma. This letter does not correspond with matrices and is a notational exception. Macroscopic cross-sections may be a scalar or a vector quantity, e.g. $\Sigma = N\sigma$ where $\sigma \in \mathbb{R}^{n}$ and $N \in \mathbb{R}^{1}$. As written, *N* is the number density of a nuclide and σ is the vector of microscopic cross-sections.

2. A BRIEF REVIEW OF REACTOR CALCULATIONS

2.A Nuclear Calculations

Nearly 15% of the world's electricity is generated from over 400 nuclear power plants internationally (USDOE 2012). This important resource is possible through untold numbers of person-hours devoted to research, experimentation, design, construction and simulation. Because of the complexity of nuclear reactor calculations, due to time-dependent multi-scale multi-physics dictating the behavior of a reactor system, modeling and simulation tools play an important role in the research and development of cutting-edge reactor design.

Simulation of these nuclear systems is difficult for many reasons, but for brevity, three important contributors are discussed herein. First, the complexity of the neutron transport equation, a seven dimensional integro-differential equation in time, space, energy, and angle shown in Eq. 7, does not have analytical solutions except for the most simplistic of systems (Duderstadt 1976, Bell 1970). Attempts to provide more sophisticated analytical benchmarks have since been undertaken by Ganapol in order to provide mathematical comparisons (2008). Without analytical benchmarks, the verification and validation of software to theory becomes a unique challenge of its own.

$$\frac{1}{v}\frac{\partial \psi(r,E,\Omega,t)}{\partial t} + \Omega \cdot \nabla \psi - \Sigma_t \psi = \int_{4\pi} d\Omega \int dE\, Q(r,E,\Omega,t) \tag{7}$$

Second, experimental mock-ups for reactor systems are not economically feasible. Even small-scale thermal-hydraulics simulations can cost millions of US dollars in excess of the majority private and public research engineering budgets. Evaluations must instead be made in comparison to the currently available body of experiments. The lack of experimental

benchmarks compounds the issue rising from a lack of analytical benchmarks, that is, the difficulty validating models representing Eq. 7 is increased.

The third difficulty discussed herein is the complications arising from the numerical modeling of Eq. 7. Currently available computing power makes the full reactor simulation impractical whether the simulation is deterministic solution of partial differential equations or a first-principles Monte Carlo particle flight simulation; a comparison of the two methodologies is discussed further in this section. In order to make the simulations computationally feasible, approximations and/or assumptions in Eq. 7 must be made, e.g. homogenization of spatial components, diffusion physics assumptions, reduction in spatial dimensions, selection of coarser energy groups, etc. (Stamm'ler 1983). As research and reactor design pushes forward for more accurate evaluations, more computational power and/or time is required.

Regardless of our ability to accurately model Eq. 7, the accuracy of results is also strongly dictated by our knowledge of the fundamental data, i.e. cross-sections. Nuclear cross-sections represent an area-based interaction probability between a nucleus, assumed at rest at a given temperature T, and an incident particle with energy E. These cross-sections, an example of which is shown in Fig. 2, are experimentally determined and are dependent on the isotope, the temperature of the target, the incident particle energy, and the nuclear reaction under consideration. Because the incident particle of interest, the neutron, lacks a chart, calibrating cross-section results to neutron energy is a challenging process that increases the uncertainty of experiments. Additionally, because all nuclear models require

fundamental cross-section data there will be measurable uncertainty in all derived results; even if Eq. 7 is modeled perfectly with infinite time and precision.

Neutron cross-section uncertainty exists because of the difficulty in measuring precise energy-dependent cross-sections. Neutrons are not charged particles, so determining the energy of neutrons involves costly time-of-flight measurements. These energy calibrations are typically completed by tuning equipment to nearby well-defined resonances. Additionally, due to the low interaction probability of some cross-sections, accurate measurements become cost-prohibitive to measure. For more details on the evaluation of nuclear cross-sections and their associated uncertainties, see chapter one in (Cacuci 2010).

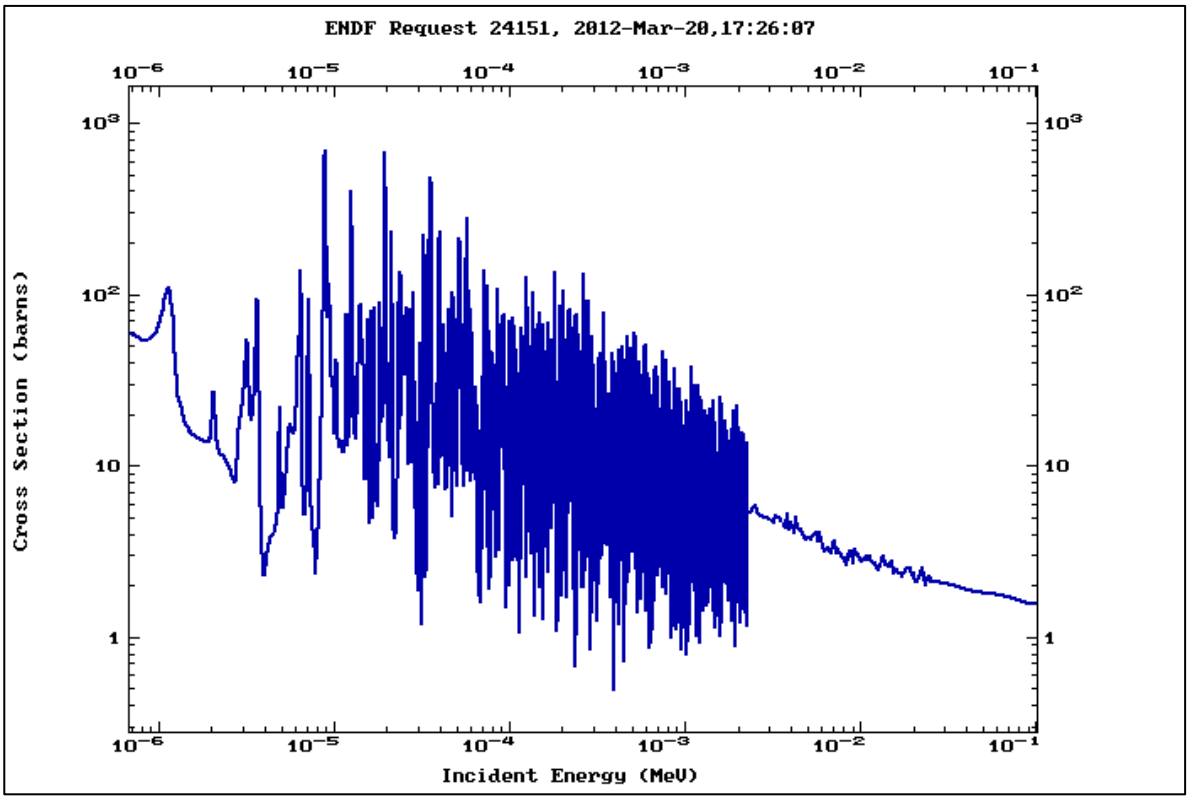

Figure 2: Uranium-235 fission cross-section at 293^{o} K.

To handle the computational limitations of reactor core transport simulations from Eq. 7, the calculation sequence to model an entire core must be carefully addressed. Traditionally a high-fidelity 1-D or 2-D simulation of fuel-pin cells are generated with many energy groups. Using this data, cross-sections are homogenized in energy and used for a 2-D assembly level calculation. After assembly-level calculations, additional spatial and energy homogenization occurs alongside physics reduction from transport to nodal diffusion to complete core-wide simulations. At this level reactor safety parameters can be checked, transients can be appropriately studied and so forth. See Fig. 3 from Stamm'ler (1983).

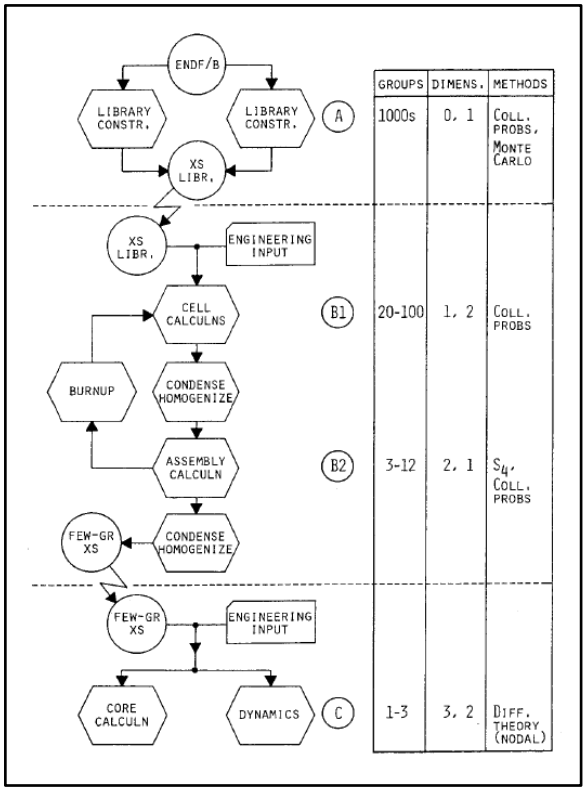

Figure 3: Reactor physics calculation flow-chart (Stamm'ler 1970).

The end goal for reactor calculations is the spatial distribution of neutrons and the power level in the reactor. From this information, appropriate safety and operational calculations can be made. Because of the computational intractability to model a fine-group time-dependent 3-D transport reactor core, calculation steps using homogenization according to the calculation flowchart are required.

Reactor calculations begin with the preparation of cross-section library data for a fine-energy group calculation of a localized pin-cell. The results of these fine spectrum calculations enable an accurate homogenization of the pin-cell cross-sections in preparation of assembly level calculations. A sample 2-D pin-cell transport model is shown in Fig. 4. Due to the near-standard thermal design of nuclear reactors, these few-group cross-section libraries are often widely available as packages alongside transport solver packages. Examples of few-to-many group pre-compiled thermal neutron libraries are the 44 and 238 group thermal cross section libraries included in the SCALE code package (2005).

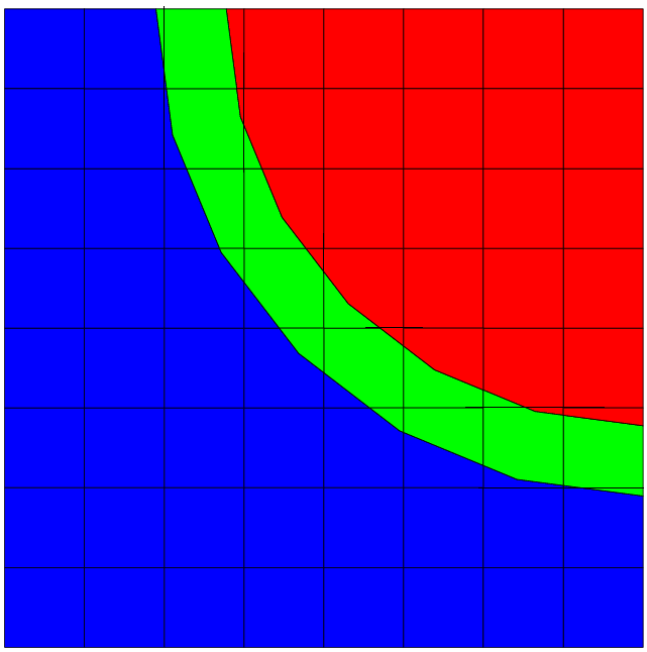

Figure 4: A simple 2D fuel pin-cell model

Upon generation of few-to-many group cross-section libraries assembly-level calculations become computationally feasible. A sample 2-D UAM assembly model is shown in Figure 5. For calculations on assemblies, typical calculations evaluate fuel-pin flux data using reflective boundary conditions. This flux data is used to collapse down to 2 – 4 energy groups and homogenize the spatial data to a single FA XS. These FA XS are used for core-wide nodal diffusion calculations for fast core simulators (Cacuci 2010, Stamm'ler 1983).

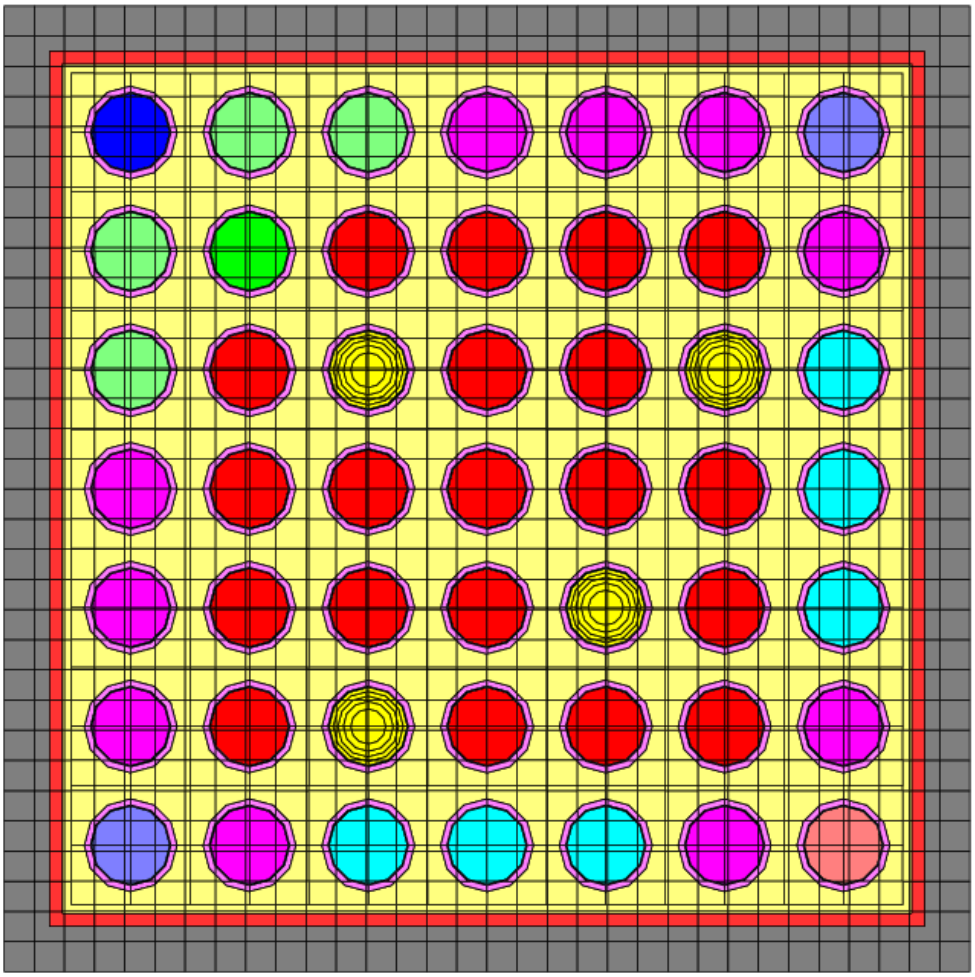

Figure 5: UAM assembly model from SCALE.

Calculations for the entire core using homogenized few group nodal-diffusion allow for reactor-core wide effects such as: FA relative powers, core-wide neutron distributions, etc. A few core simulators currently in use today include NESTLE, FORMOSA-B, HELIOS (Moore 1999, Turinsky 1995, Villarino 1992). For a sample numerical setup of a full-core calculation, see Figure 6. Because the technical details of this report focus on the pin and

assembly level numerical demonstrations, details for full-core calculations will be left to the interested reader.

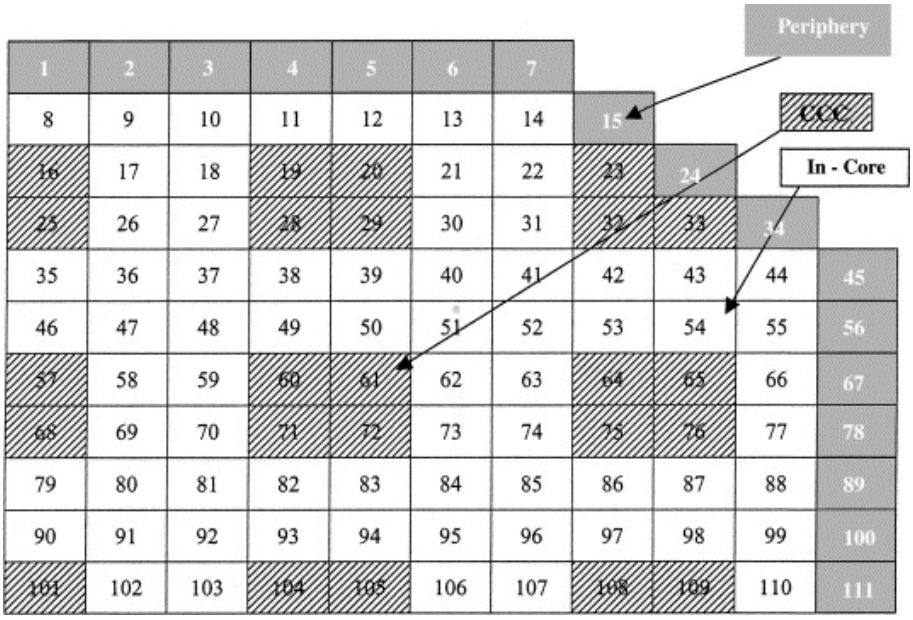

Figure 6: BWR quarter-core symmetry reactor model (Martin-del-Campo 2004).

While there are several steps to prepare XS to ultimately obtain full reactor-core calculations, there are two primary modeling philosophies that enable calculations of reactor components. The deterministic model involves solving variants of the neutron transport equation in Eq. 7. In solving the deterministic model, a generalized eigenvalue problem is solved over some region, usually at steady-state (time-independent) to obtain the average value of neutrons in mesh-cells. Any effects in time typically require a decoupling of the physics, e.g. depletion models, or a dramatic simplification of the physics, e.g. point reactor kinetics. In contrast, Monte Carlo models involve first-principles stochastic particle

simulation; individual neutrons are simulated within a region of space and reactions are simulated using continuous energy physics. Monte Carlo models, being stochastic averages of particle tracks, require many (millions or more) particles and a large amount of computation time (days to weeks); however, more complicated geometries can be modeled. Due to the excessive run-time and uncertainty introduced on outputs due to random sampling, Monte Carlo models are not suitable for applications where calculation time is a premium (e.g. online core simulators). Additionally, the uncertainty on outputs is problematic in regions with low particle counts such as in slab penetration problems. The disadvantage of deterministic codes includes homogenized/simple geometries and non-continuous energy representation for neutron transport. For additional details on Monte Carlo transport, see references (Carter 1975, Brown 2002). Similarly for deterministic models see (Duderstadt 1976, Cacuci 2010, Bell 1970, Stamm'ler 1983).

2.B Sensitivity Analysis Motivations

This dissertation introduces an approach to obtain response-gradient information, i.e. response sensitivity profiles, without the formulation or solution of the generalized adjoint equations. Any technique that yields sensitivity data falls into the category of a sensitivity analysis, relating output responses to input parameters. The motivation for sensitivity analyses is usually for one or more of the following: design optimization, uncertainty quantification, and data assimilation.

Design optimization is a mathematical technique that finds the local best-fit of a set of parameters to a model, often with a set of constraints. Common sets of engineering

constraints include minimizing costs, maximizing profits, meeting safety limits, and maintaining the system within physical constraints. These constraints or design objectives often are in competition of one another such that there are many local optimums; however, a better optimum may be nearby. For an optimization problem, the following conditions for an optimum must be met for a function f with $x \in \mathbb{R}^n$ input parameters:

$$\frac{\partial f\left(x^*\right)}{\partial x_i} = 0 \quad \forall \quad 1 \le i \le n \tag{8}$$

In Eq. 8, the optimum solution is denoted x^*, with the derivative with respect to all input parameters equal to zero at the optimum. Similarly, to denote the point as being a minimum (as opposed to a maximum or inflection point):

$$\frac{\partial^2 f\left(x^*\right)}{\partial x_i^2} > 0 \quad \forall \quad 1 \le i \le n \tag{9}$$

For high-dimensional systems, obtaining gradient information as in Equation 8 can be costly at best; however, when available, optimization schemes can benefit from gradient information to quickly find local optima (Kelley, 1999). Any scheme that allows for a faster determination of response sensitivities, i.e. response gradients, or allows for determination of these sensitivities that is otherwise unavailable, e.g. Monte Carlo methods, would be beneficial for numerous design optimization problems in the nuclear engineering community.

Uncertainty Quantification attempts to propagate uncertainty information from input parameters to output responses through a model. Consider any nuclear reactor model, reliant on cross-sections with inherent uncertainty. The output of safety parameters, pin-powers, and feedback parameters will have an uncertainty as a result of the uncertainty in the input

parameters. In a first order approach, the uncertainty can be propagated to the output parameter by means of the sandwich rule (Cacuci 2008, Cacuci 2010, Dean 2007). For a given covariance matrix $\mathbf{C}_x$ of n input parameters a corresponding covariance matrix $\mathbf{C}_y$ of m output responses can be constructed when the sensitivity matrix $\mathbf{S}$ is known (Eq. 10).

$$\mathbf{C}_y = \mathbf{S}^T \mathbf{C}_x \mathbf{S} \quad \mathbf{S} \in \mathbb{R}^{n \times m} \quad \mathbf{C}_x \in \mathbb{R}^{n \times n} \quad \mathbf{C}_y \in \mathbb{R}^{m \times m} \tag{10}$$

Propagation of uncertainties requires knowledge of the sensitivity profiles, stored in $\mathbf{S}$ and the covariance of the input, $\mathbf{C}_x$. Similar to design optimization, any scheme that provides these sensitivity profiles more quickly or when it is unavailable allows for the possibility of uncertainty propagation calculations. This relationship can be further optimized by constraining sensitivities and covariance data to input parameters that are relevant to the model (Bang 2011).

Data assimilation attempts to update the mean value of specific model parameters given covariance information and sensitivity information. This style of model parameter fitting has been widely used in the atmospheric sciences (Daley 1991, Ghil 1991) and in nuclear engineering (Gandini 1988); the topic of data assimilation has recently again become of interest to nuclear engineers in order to improve cross-section accuracy (Cacuci 2008, Palmiotti 2010).

Data assimilation usually requires two-steps, the first is the relation of all inherent models sensitivities to come up with a combined sensitivity (Eq. 11) relating a measured output parameter to an input parameter that will be updated.

$$\frac{\partial R_k}{\partial u_j} = \sum_{i=1}^{n} \frac{\partial R_k}{\partial x_i} \frac{\partial x_i}{\partial u_j} \tag{11}$$

Once this has been completed, a cost function, using the uncertainties of the input parameters and the resulting data are used to solve an optimization problem to update the input parameters, denoted u in Eq. 11. The details of this optimization procedure, development of the cost-function, and other details are not provided in this dissertation but a search of the literature or aforementioned references will lead to many available tools and methods.

3. LITERATURE REVIEW

3.A Generalized Perturbation Theory

Generalized perturbation theory (GPT) is a mathematical technique that develops both an adjoint relation and an inner product to estimate how model parameter perturbations affect user or problem-defined output responses. The original developments of perturbation theory (PT), and similarly GPT, were rooted in physics, being formally developed by Schrödinger (1926) and Lord Rayleigh (1894) and leading to the term Rayleigh-Schrödinger perturbation theory. Over the last century, PT and GPT have been expanded to the physical sciences from various fields such as nuclear engineering (Gandini 1967, Williams 1978), optics (Mitsunaga 1985), quantum mechanics (Pople 1955, Schrödinger 1926), chemistry (Sowers 1991, Andersson 1990), plasma physics (Al'tshul' 1966), and general mathematics (Primas 1963).

When the nuclear engineering field was first being developed, PT and GPT techniques were quickly adopted into research studies such as the calculation of pile period due to perturbations by Wigner (1945) and the evaluation of reaction-rate ratios by Usachev (1964). Since the introduction of GPT to the nuclear field, the technique has been applied to a wide body of problems of interest to the nuclear engineer. The study of reaction-rate ratios such as in Eq. 12 has been a common process studied by many scientists such as Usachev (1964), Williams (1978, 1986), Pomraning (1967, 1987), Lewins (1966), Gandini (1967), and Stacey (1972, 1974).

$$R = f\left(\langle \sigma_1, \phi \rangle, \langle \sigma_2, \phi \rangle, \ldots\right) \quad \text{e.g.} \quad R = \frac{\langle \sigma_s, \phi \rangle}{\langle \sigma_t, \phi \rangle} \tag{12}$$

GPT was formulated to calculate *k*-eigenvalue criticality sensitivity profiles (see Eq. 13) with respect to the nuclear cross-section data (Pomraning 1966, Pomraning 1967, Stacey 1972) which has since been used for reactor design calculations (Kando 1971).

$$s_{k,\sigma_{x,g,z}} = \frac{\partial k / k}{\partial \sigma_{x,g,z} / \sigma_{x,g,z}} \quad x \in \text{Reactions} \quad g \in \text{Energy Groups} \quad z \in \text{Regions} \tag{13}$$

Boundary condition perturbation theory was first used by Komata (1977) to estimate changes in the reactor neutron distribution and *k* due to perturbations of the reactor geometry, and has since been expanded further by Larsen (1981), McKinnely (2001), Rahnema (1983, 1999), Yamamoto (1997) and Petrov (1985). The study of reactor kinetics such as feedback and reactor period–building upon Wigner in 1945–has also benefitted from the GPT methodology by scientists such as Williams (1991), Harries (1978), and Kiedrowski (2010). Reactor operation burns the fuel in the reactor through depletion, which also benefits from the rigors of GPT (Williams 1978, Yang 1989). The formalism of GPT has been revised throughout the decades of research into the subjects. For a more complete list of references and mathematical formalisms, see the following references (Cacuci 1980, 2003, 2010; Oblow 1976, Williams 1986, and Stacey 1974, 2001).

Generalized Perturbation Theory is distinguished from perturbation theory in nuclear engineering by the definition of a source-term. For steady-state calculations of the neutron flux using the transport equation, the forward and adjoint models are closely related as in Eq. 14.

$$\mathbf{L}\phi - \frac{1}{k}\mathbf{F}\phi = 0 \quad \text{and} \quad \mathbf{L}^*\phi^* - \frac{1}{k^*}\mathbf{F}^*\phi^* = q^* \tag{14}$$

For perturbation theory, the adjoint source-term q^* is zero; however, for generalized perturbation theory the source depends on the response parameter of interest by the relation:

$$q^* = \frac{1}{R}\frac{dR}{d\phi} \quad R = f\left(\left\{\langle \phi, \sigma \rangle_i\right\}_{i=1}^{m}\right) \tag{15}$$

The response in Eq. 15 is a user or problem-defined functional composed of *m*, arbitrary, inner-products of the cross-sections and the flux. The term *generalized* comes from the fact that the homogenous adjoint used in perturbation theory is primarily used to calculate eigenvalue sensitivities whereas the generalized adjoint with arbitrary source term can handle any 'GPT-allowable' response. The GPT source term must be allowable due to mathematical constraints outlined in Williams (1986), which in summary expresses the following conditions on the adjoint formulation given some response *R* of the form in Eq. 16:

$$\begin{aligned} &1. \quad \left(\mathbf{L}^* - \frac{1}{k}\mathbf{F}^*\right)\phi^* = q^* \qquad q^* = \frac{1}{R}\frac{dR}{d\phi} \\ &2. \quad \left\langle q^*, \phi \right\rangle = 0 \\ &3. \quad \left\langle \phi^*, \mathbf{F}\phi \right\rangle = 0 \end{aligned} \tag{16}$$

If these three conditions can be met, the response is allowable by GPT and only one adjoint and one forward simulation is required in order to predict the change in some response *R* due to input parameter perturbations.

General GPT evaluations are designed to determine the sensitivity of a response of interest, of the form in Eq. 12 to some input parameter of interest, e.g. cross-sections, geometry, or composition. Whereas perturbation theory only provides the *k*-eigenvalue sensitivity to input parameters, GPT extends this to GPT-allowable responses in order to calculate the response sensitivity profiles of the form in Eq. 17.

$$S_R = \frac{\partial R / R}{\partial \alpha / \alpha} \tag{17}$$

Here, alpha is some general input parameter. The solution of the GPT formulation allows for the calculation of the indirect term in evaluating the change in a response due to a change in parameters. Consider a response perturbed by a change in the cross-sections which yields a change in the state-space (flux) in Eq. 18.

$$R + \Delta R = f\left(\langle \sigma + \Delta\sigma, \phi + \Delta\phi \rangle\right) = f\left(\langle \sigma, \phi \rangle + \langle \sigma, \Delta\phi \rangle + \langle \Delta\sigma, \phi \rangle + \langle \Delta\sigma, \Delta\phi \rangle\right) \tag{18}$$

The first term and the cross-section perturbation inner products can be evaluated using the forward solution; however, the indirect term including the change in the flux cannot be predicted without a corresponding forward simulation for each input parameter. The double delta term is dropped in a first-order sensitivity evaluation because the effect is second-order. To proceed, consider Eq. 14 with a perturbation in the input parameters introduced, yielding a change in the matrices, eigenvalue, and the state-space as in Eq. 19.

$$\left[(\mathbf{L} + \Delta\mathbf{L}) - (\lambda + \Delta\lambda)(\mathbf{F} + \Delta\mathbf{F})\right](\phi + \Delta\phi) = 0 \qquad \lambda = \frac{1}{k} \tag{19}$$

Recall that Eq. 14 holds, and by rearranging the remaining terms while neglecting second order effects (e.g. combinations that include 2 or more delta terms multiplied together), the following equation remains:

$$(\mathbf{L} - \lambda\mathbf{F})\Delta\phi = \Delta\lambda\mathbf{F}\phi - (\Delta\mathbf{L} - \lambda\Delta\mathbf{F})\phi \tag{20}$$

Recall the adjoint formulation from Eq. 14, and take the inner product of ϕ^* with Eq. 20 and $\Delta\phi$ with Eq. 14:

$$\langle \phi^*, (\mathbf{L} - \lambda\mathbf{F})\Delta\phi \rangle = \Delta\lambda \langle \phi^*, \mathbf{F}\phi \rangle - \langle \phi^*, (\Delta\mathbf{L} - \lambda\Delta\mathbf{F})\phi \rangle \tag{21}$$

$$\left\langle \Delta\phi, \left(\mathbf{L}^* - \lambda\mathbf{F}^*\right)\phi^* \right\rangle = \left\langle \Delta\phi, \frac{1}{R}\frac{dR}{d\phi} \right\rangle \tag{22}$$

Recall the property of the adjoint:

$$\left\langle \mathbf{A}x, y \right\rangle = \left\langle x, \mathbf{A}^* y \right\rangle \tag{23}$$

Combining Eq. 21 and 22 and using the property of the adjoint the following relation for the indirect flux term can be formed, allowing the calculation of Eq. 24.

$$\Delta\lambda\left\langle \phi^*, \mathbf{F}\phi \right\rangle - \left\langle \phi^*, \left(\Delta\mathbf{L} - \lambda\Delta\mathbf{F}\right)\phi \right\rangle = \left\langle \frac{1}{R}\frac{dR}{d\phi}, \Delta\phi \right\rangle \tag{24}$$

The evaluation of response sensitivities as in Eq. 17 is computed in several steps.

1. **The problem adjoint must be constructed**. *This step is straightforward for neutronics, but, Williams (1986) shows that coupling additional physics, such as depletion, greatly increases the adjoint formulation complexity.*
2. **The source-term for the response must be allowable**. *This involves an inner-product check of the source term with the forward solution for the neutron flux.*
3. **The adjoint problem must be solved**. *This requires a software package that supports generalized adjoint calculations*.
4. **The inner product must be evaluated**.
5. **Steps 2 through 4 must be repeated for each response of interest**.

Three fundamental challenges exist in nuclear engineering software that limits or prevents GPT calculations: the problem of dimensionality, the cost of implementation, and limitations of the modeling philosophies. 1) The dimensionality concern is that an adjoint simulation as in Eq. 24 must be evaluated for *every* response of interest. Consider an

arbitrary model with n input parameters and m output responses. There are two common modes of determining sensitivity profiles, the forward approach, which is dependent on the number of input parameters requiring approximately n calculations of the forward (original) model. The adjoint approach, using PT or GPT, is dependent on the number of responses, requiring approximately m adjoint equation calculations. The optimal method of evaluating sensitivities depends on the ratio of input parameters to output responses and the additional time required to solve the adjoint problem. Traditionally the adjoint method has been favored by reducing the dimensionality of the outputs via homogenization and model reduction so that the number of adjoint calculations is sufficiently fewer than the number of input parameters. As models grow in complexity and accuracy, e.g. the modeling of a Gadolinium fuel pin in assemblies with a high number of both spatial regions and energy groups, the number of output parameters can sufficiently increase such that both m and n is large. In these cases, the model is high dimensional and neither scheme, forward or adjoint, is immediately optimal. 2) Many software packages used by the nuclear industry perform homogenous adjoint calculations, allowing for PT evaluations; however, because of a lack of vision, resources, or need, these same software packages do not support the additional components to allow for GPT evaluations. In these cases, the software designers and programmers are often no longer available to operate or provide advice on the software. Furthermore, a lack of documentation and commenting in the source code makes it difficult and/or intractable to retrofit these software packages with GPT capabilities. Under these circumstances, GPT evaluations are not possible and are not probable in the near future despite their heavy use by the nuclear industry. 3) The Monte Carlo modeling philosophy is

challenged by the difficulty of accurately and with good statistics providing an adjoint methodology. For the Monte Carlo models, a continuous energy adjoint is often difficult to achieve with statistically meaningful results if the calculation is even available. Because an inner product for the GPT adjoint in a Monte Carlo model is not yet clear, GPT is still unavailable for this modeling philosophy. If any of these three challenges can be overcome, GPT calculations can be either more available, more accurate, or both.

3.B Perturbation Theory

In nuclear engineering, perturbation theory is a special case of generalized perturbation theory where the source term, q^* in Eq. 14 is zero, allowing for the calculation of the *k*-eigenvalue sensitivities with both a forward and an adjoint calculation (Pomraning 1966, Pomraning 1967, Stacey 1972, Williams 1986). Because the homogenous adjoint is much simpler to calculate, perturbation theory is more widely available in nuclear software today. In order to evaluate the sensitivity of the fundamental eigenvalue, fewer steps and calculation complexities are required.

Consider again the forward and homogenous adjoint transport models in Eq. 25

$$\mathbf{L}\phi - \lambda\mathbf{F}\phi = 0 \quad \mathbf{L}^*\phi^* - \lambda^*\mathbf{F}^*\phi^* = 0 \quad \lambda = \lambda^* \tag{25}$$

First, the eigenvalues for the forward and adjoint calculation must be shown to be equivalent. Compute inner products with the forward and adjoint equations:

$$\left\langle \phi^*, \left(\mathbf{L} - \lambda\mathbf{F}\right)\phi \right\rangle = 0 \quad \text{and} \quad \left\langle \phi, \left(\mathbf{L}^* - \lambda^*\mathbf{F}\right)\phi^* \right\rangle = 0 \tag{26}$$

Subtract these two equations and use the property of the adjoint (Eq. 23) to come up with a relation between the forward and adjoint eigenvalue:

$$\begin{aligned}
&\left\langle \phi^*, \left(\mathbf{L}-\lambda\mathbf{F}\right)\phi \right\rangle - \left\langle \phi, \left(\mathbf{L}^*-\lambda^*\mathbf{F}\right)\phi^* \right\rangle = 0 \\
&\left\langle \phi^*, \mathbf{L}\phi \right\rangle - \lambda\left\langle \phi^*, \mathbf{F}\phi \right\rangle - \left\langle \phi, \mathbf{L}^*\phi^* \right\rangle + \lambda^*\left\langle \phi, \mathbf{F}^*\phi^* \right\rangle = 0 \\
&\left\langle \phi^*, \mathbf{L}\phi \right\rangle = \left\langle \mathbf{L}^*\phi^*, \phi \right\rangle = \left\langle \phi, \mathbf{L}^*\phi^* \right\rangle \\
&\left\langle \phi^*, \mathbf{F}\phi \right\rangle = \left\langle \mathbf{F}^*\phi^*, \phi \right\rangle = \left\langle \phi, \mathbf{F}^*\phi^* \right\rangle \\
&\left(\lambda^*-\lambda\right)\left\langle \phi^*, \mathbf{F}\phi \right\rangle = 0
\end{aligned} \tag{27}$$

Because the inner product $\left\langle \phi^*, \mathbf{F}\phi \right\rangle$ is non-negative and usually greater than zero except for extreme cases, the forward and adjoint eigenvalues must be equivalent (Williams 1986). Expanding the forward model with first order perturbation theory yields the following equation:

$$\left(\mathbf{L}\phi - \lambda\mathbf{F}\phi\right) + \left(\mathbf{L}\Delta\phi - \lambda\mathbf{F}\Delta\phi\right) + \left(\Delta\mathbf{L}\phi - \lambda\Delta\mathbf{F}\phi\right) - \Delta\lambda\mathbf{F}\phi = 0 \tag{28}$$

From the forward model, the first term is zero. Take an inner product of Eq. 28 with the adjoint flux from Eq. 25:

$$\Delta\lambda\left\langle \phi^*, \mathbf{F}\phi \right\rangle = \left\langle \phi^*, \left(\mathbf{L}\Delta\phi - \lambda\mathbf{F}\Delta\phi\right) \right\rangle + \left\langle \phi^*, \left(\Delta\mathbf{L}\phi - \lambda\Delta\mathbf{F}\phi\right) \right\rangle \tag{29}$$

Using the property of the adjoint, the middle term can be shown to be zero from the adjoint formulation because of Eq. 25.

$$\left\langle \phi^*, \left(\mathbf{L}-\lambda\mathbf{F}\right)\Delta\phi \right\rangle = \left\langle \left(\mathbf{L}^*-\lambda\mathbf{F}^*\right)\phi^*, \Delta\phi \right\rangle = 0 \tag{30}$$

Thus the change in the eigenvalue can be shown to be found using the forward and adjoint solution of the flux:

$$\Delta\lambda = \frac{\left\langle \phi^*, \left(\Delta\mathbf{L}-\lambda\Delta\mathbf{F}\right)\phi \right\rangle}{\left\langle \phi^*, \mathbf{F}\phi \right\rangle} \tag{31}$$

By calculating once the reference (forward) solution for ϕ and additionally one more calculation for the adjoint solution ϕ^*, the first order change in the eigenvalue can be estimated due to changes in input parameters. This capability can be extended to determine derivative sensitivities of the eigenvalue to input parameters by a change of variables:

$$\Delta\lambda = \frac{\partial\lambda}{\partial\alpha}\Delta\alpha = \frac{\left\langle \phi^*, \left(\frac{\partial \mathbf{L}}{\partial\alpha} - \lambda\frac{\partial \mathbf{F}}{\partial\alpha}\right)\phi\Delta\alpha \right\rangle}{\left\langle \phi^*, \mathbf{F}\phi \right\rangle}$$
$$\frac{\partial\lambda}{\partial\alpha} = \frac{\left\langle \phi^*, \left(\frac{\partial \mathbf{L}}{\partial\alpha} - \lambda\frac{\partial \mathbf{F}}{\partial\alpha}\right)\phi \right\rangle}{\left\langle \phi^*, \mathbf{F}\phi \right\rangle} \quad (32)$$

Converting to the system multiplication parameter *k*:

$$\frac{\partial k}{\partial\alpha} = -k^2\frac{\left\langle \phi^*, \left(\frac{\partial \mathbf{L}}{\partial\alpha} - \frac{1}{k}\frac{\partial \mathbf{F}}{\partial\alpha}\right)\phi \right\rangle}{\left\langle \phi^*, \mathbf{F}\phi \right\rangle} \quad (33)$$

The advantage of this perturbation theory formulation is that the first order calculation of eigenvalue changes for an arbitrary number of input parameters can be determined with only two simulations, a significant gain over the forward approach.

3.C Monte Carlo GPT and Adjoint Development History

Simulation of the adjoint transport equations using the Monte Carlo method has conventionally been difficult due to the nature of neutron transport and the problems being solved. A brief overview of the literature summarizing the approaches and challenges to calculate a Monte Carlo adjoint is provided to explain why GPT calculations using the Monte

Carlo philosophy are currently less available. The majority of the literature that pertains to a Monte Carlo adjoint, both continuous and multi-group, is limited to the fundamental adjoint sometimes denoted the Importance function. Furthermore, the adjoint methodologies to be discussed are primarily for fixed-source problems and are thusly not capable of handling eigenvalue calculations. Assuming such a continuous energy representation of the fundamental adjoint is possible in the eigenvalue operating mode; construction of the GPT-specific source terms is unclear at best and is a modern topic of research.

The research and development of the fundamental adjoint has had to overcome significant limitations and problems throughout recent years. In 1970 and later in 1975, Carter discusses the benefits and limitations of the adjoint Monte Carlo method from the integro-differential viewpoint of neutron transport. Two major limitations cited by Carter arise from statistical difficulties: First, the adjoint (pseudo) neutrons do not necessarily migrate towards the region of interest. Those adjoint neutrons that do make it to the region of interest typically have highly varying weights, damaging the statistics of the solution. Highly varying weights contaminates the variance due to the difference between the squared mean sample weight and the mean of the squared weights:

$$\bar{\sigma}^2 \cong \frac{1}{N}\sum_{i=1}^{N} w_i^2 - \left(\frac{1}{N}\sum_{i=1}^{N} w_i\right)^2 \tag{34}$$

Wagner (1994) cites a similar problem generating the adjoint importance function due to statistical limitations using MCNP. Furthermore, the MCNP adjoint multi-group option requires pre-model calculations to accurately calculate the source term and response tallies. Because of the limitations and considerations, multi-group adjoint calculations are much

more complicated than their forward counterpart in MCNP. In contrast to Wagner, Carter's solution to handling of the adjoint source terms requires a complete cross-section library revision. Even with this solution, the problem again is the statistical weights of pseudoneutrons.

Hoogenboom expanded on Carter's approach to reducing the variance of adjoint particle weights by showing that the optimal weight function was different from that of Carter and McCormick (Hoogenboom 1981); however, this improvement makes the sampling scheme problem-dependent and again the problem still requires preparation of adjoint cross-sections. Hoogenboom further went on to develop the FOCUS code which allowed for continuous energy eigenvalue-mode fundamental adjoint evaluations; however, the uncertainty in the response functions increases with each generation, an undesirable trait due to a need for a large number of generations to remove higher eigenfunction mode contamination.

An alternative approach, the iterated fission probability concept of the fundamental adjoint, was introduced by Hurwitz in 1948, which was later expanded upon with collaboration with Ehrlich in 1954 and by authors such as Lewins in 1960. The iterated fission probability uses the forward calculation to determine the steady-state neutron flux exhibited by original generation neutrons. The advantage of this approach is that no cross-section processing is required, and the forward model can be modified to provide these evaluations without changes to the operators or source terms. The disadvantage to the method is again the statistical problem arising from a steady-state flux evaluated for each progenitor neutron, requiring significant tracking and populations to provide meaningful results.

Furthermore, accurately tracking neutron generations through scattering physics is challenging.

A recent development by Kiedrowski in 2010 expands on the iterated fission probability concept to calculate adjoint-weighted tallies measuring point reactor kinetics parameters such as the neutron generation time, the delayed neutron fraction, and Rossi-alpha. This work is being expanded to support general adjoint-weighted tallies of the form to calculate the fundamental adjoint sensitivity profile in the latest version of MCNP. From the view of the literature, this method has the highest hope for statistically satisfactory adjoint-weighted tallies that may be developed into a GPT framework for Monte Carlo.

In review, the adjoint Monte Carlo solution methodology has been approached from many avenues via different forms of the transport equation (Carter, Hurwitz, and Ehrlich), different approaches to handling the importance function (Carter, Hoogenboom, and Eriksson), multi-group approaches (Ehrlich and Wagner), and the general adjoint weighted tallies (Kiedrowski). While the literature has many more articles from theses authors and colleagues that expand on the various methods, the fundamental objectives typically are modifications to the overall method for improved statistics, i.e. variance reduction biasing schemes. While significant advances have been made for fixed-source problems, the ability to calculate GPT inner-products using a continuous energy Monte Carlo adjoint is currently unavailable, with the best hope being expansions on Kiedrowski's methodology. As the ability for a *k*-eigenvalue sensitivity profile is within reach by Kiedrowski, there is motivation to employ the GPT-Free methodology to evaluate GPT style responses.

3.D Reduced Order Models

Modeling of physical phenomena oftentimes results in very complicated mathematical systems. While these models may faithfully represent the underlying physics or behavior of a physical system, their practical use may be limited by computational limitations or by needs for repeated execution. In these circumstances, reduced order models (ROM) are generated for the purpose of making a calculation feasible or expediting calculations for repeated model execution. In general, a ROM attempts to replicate the results of a higher fidelity model with less complicated systems that are often easier and/or quicker to solve. With modern engineering calculations becoming larger, involving more than millions of inputs and/or equations there is a significant need for model order reduction.

Historically, reduced order models have always been present in the sciences when attempting to model physical phenomena. First-order perturbation theory (PT) which has been around for centuries is a common example of forming a reduced order model of a system about some reference point. Truncating all terms after the first order derivative of a Taylor series expansion provides a linear representation of phenomena that is often easier to study. Similarly the classical equation of motion can be thought of us a ROM for the relativistic motion equations when the velocity is much less than the speed of light, $v << c$ (which is true for everyday physical phenomena at the macroscopic scale). Reduced order models received renewed attention with the development of computing power starting at the end of the 20^{th} century. As computing power increases, it is possible to solve larger and more complicated physically phenomena. While individual simulations of large-scale computing problems may be feasible, analysis of behavior requiring repeated model execution may

make these large-scale models intractable, hence the need for a ROM. Specific to nuclear reactor calculations, a full-core Monte Carlo particle simulation of a nuclear reactor can be completed with significant computational resources; however, repeat execution of the model is intractable for design purposes and thus the need for a ROM is present.

There are a plethora of ROM construction philosophies in the literature (Schilders 2008, Cacuci 2003); however, these can be condensed to approximately three methodologies in the physical sciences. The first methodology attempts to reduce model dimensionality by fitting input parameters and output responses to some user-defined function and is denoted response-surface modeling. While response surface modeling need not form a reduced order model, the methodology is often used in engineering for this purpose. An example of response surface modeling is fitting perturbations of a model function to a first-order derivative model where the original model and resulting reduced model can be viewed in Eqs. 35 and 36.

$$f\left(\alpha_0+\Delta\alpha\right)=f\left(\alpha_0\right)+\left(\left.\frac{\partial f}{\partial\alpha}\right|_{\alpha_0}\right)\Delta\alpha+\left(\left.\frac{\partial^2 f}{\partial\alpha^2}\right|_{\alpha_0}\right)\Delta\alpha^2+\ldots \tag{35}$$

$$f\left(\alpha_0+\Delta\alpha\right)\approx f\left(\alpha_0\right)+\left(\left.\frac{\partial f}{\partial\alpha}\right|_{\alpha_0}\right)\Delta\alpha \tag{36}$$

The concept can be generalized to many outputs and many responses; however, additional notation would be required to describe the resulting tensors. For an overview of response-surface modeling applied to reduced order models, see Schilders (2008) for a list of references and examples.

The second methodology involves a physics-based reduction of the model. Typically these reductions are for physical and semi-empirical partial differential equations. Some examples in the nuclear field include the diffusion approximation of transport as in Eq. 37, the assumption for isotropic scattering in transport (Eq. 38), Quasi-static depletion in Eq. 39, and point-reactor kinetics, Eq. 40 (Bell 1970, Williams 1986, Allen 1975, Lamarsh 2001).

$$\psi\left(r,\Omega,E,t\right) \rightarrow \phi\left(r,E,t\right) \text{ and } J\left(r,E,t\right) \tag{37}$$

$$\Sigma_s\left(E'\rightarrow E,\Omega\rightarrow\Omega'\right) \rightarrow \frac{1}{4\pi}\Sigma_s\left(E'\rightarrow E\right) \approx \int_{4\pi}\Sigma_s\left(E'\rightarrow E,\Omega\rightarrow\Omega'\right)d\Omega \tag{38}$$

$$\mathbf{L}\left(\Sigma(t)\right)\phi\left(t\right)-\frac{1}{k\left(t\right)}\mathbf{F}\left(\Sigma(t)\right)\phi(t)=0 \rightarrow \mathbf{L}_i\phi\left(t_i\right)-\frac{1}{k_i}\mathbf{F}_i\phi\left(t_i\right)=0 \tag{39}$$

$$\Sigma\left(r,t\right),\rho\left(t\right)\rightarrow\Sigma\left(t\right),\rho\left(t\right) \tag{40}$$

The third methodology attempts to reduce the model dimensionality by reducing the spaces of the problem be it the input parameter space, output response space, or the state-space of a system (e.g. temperature or flux). Consider a linear model with *n* input parameters, a state-space of dimension *w*, and *m* output responses. Reduction of the dimension of any of these three spaces can reduce problem complexity by restricting some space to the *r* most important directions. For example, consider a model in which the user desires to perturb input parameters one at a time according to a sensitivity analysis. Reduction to the *r* most important input parameters will save *n* – *r* simulations at some cost to the accuracy for reducing the calculation complexity, which can be denoted the *reduction error*.

For the purpose of comparing the methodologies, consider a state equation Π with input parameters σ of dimension *n*, state space ϕ and *m* general response functionals denoted by R_i for all *i*.

$$\begin{aligned} \Pi(\sigma,\phi) &= 0 \\ R_i &= \langle \phi,\sigma \rangle \end{aligned} \tag{41}$$

Consider first the response surface model. For example, a user wishes to understand the effect of input parameter perturbations on the model in Eq. 41 by constructing a first-order sensitivity model relating output responses to input parameters. The resulting relationship forms an *m* by *n* Jacobian matrix as in Eq. 42:

$$\Delta R = \mathbf{J}\Delta\sigma \qquad \mathbf{J}_{xy} = \frac{\Delta R_x}{\Delta\sigma_y} \tag{42}$$

The reduction occurs because higher-order terms are removed as in Eq. 35, simplifying to Eq. 36. Furthermore, the user need not consider all responses or input parameters such that the Jacobian, **J** may be significantly smaller than *m* by *n*. Another example of response-surface modeling is polynomial chaos (Gentle 2009, Le Maitre 2004, Schilders 2008) as shown in Eq. 43, where higher-order terms are considered in an efficient manner.

$$f(\sigma) = \sum_{i=0}^{r}\sum_{k}^{n_i}\beta_{ik}\phi_{ik}(\sigma) \tag{43}$$

Similarly, response surface modeling is used in perturbation theory and generalized perturbation theory to relate first and second order derivatives of system properties to input parameters. The resulting short-hand model provides the ability to quickly estimate responses of interest without re-execution of the simulation for each test case (Cacuci 2008).

The second methodology attempts to rewrite Eq. 41 by simplifying underlying physics. Consider the following representation for Π:

$$\Pi(\sigma,\phi)=0 \quad \equiv \quad \mathbf{L}(\sigma)\phi-\lambda\mathbf{F}(\sigma)\phi=0 \tag{44}$$

Approximations could be introduced to convert the system to diffusion such that directional terms are removed in **L**. Alternatively, thermal scattering could be removed to change the shape of **L** so that it is triangular by removal of upscattering as in Fig. 7.

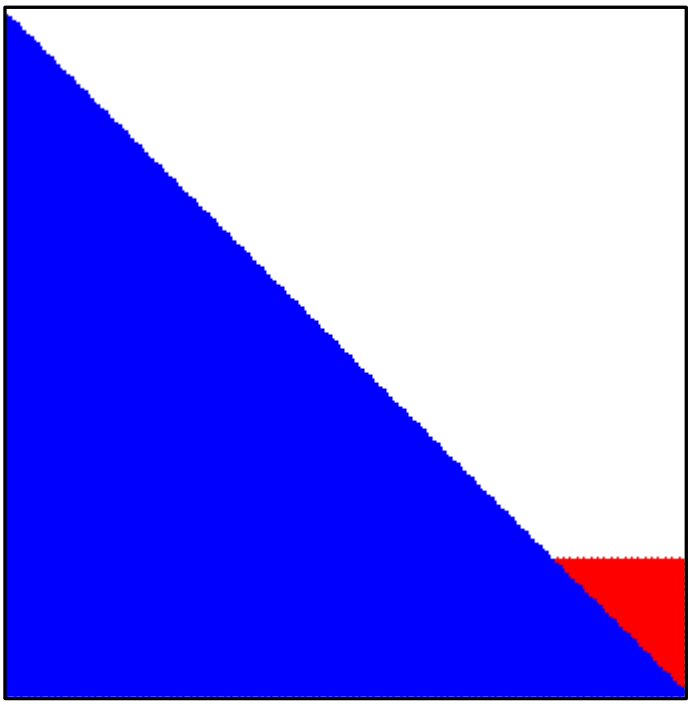

Figure 7: L matrix construction with upscattering terms marked by the red portion of the matrix.

Alternatively, the model could be homogenized so that the spatial and energy regions in **L** and **F** are reduced, thus directly reducing model dimensionality. The errors for the latter two methods can be viewed as in Eqs. 45 and 46. When removing upscattering in thermal

systems, the shape-function introduces a distortion to the state-space and the associated eigenvalue. Homogenization will produce similar effects, but the residual an integral comparison.

$$\left(\mathbf{L}-\mathbf{L}_{th}\right)\tilde{\phi}-\lambda\left(\mathbf{F}\right)\tilde{\phi}=0 \quad r=\phi-\tilde{\phi} \tag{45}$$

$$\mathbf{L}_H\phi_H-\lambda\mathbf{F}_H\phi_H=0 \quad r_{ij}=\int_{\Delta E_i}dE\int_{\Delta V_j}dr^3\phi\left(r,E\right)-\phi_{H(i,j)} \tag{46}$$

The removal of angular physics with the Diffusion approximation and the homogenization of the system directly reduces the dimensionality of the model at a sacrifice to accuracy. The removal of certain physics such as thermal upscattering does not change the dimensionality of the model but does reduce the complexity of obtaining a solution and by extension, the time to compute, thus reducing the time-cost at a sacrifice to accuracy. Other reductions include a reduction in spatial dimension from 3D to 2D and 1D depending on the problem symmetry (Duderstadt 1976, Bell 1970, Stamm'ler 1983). These schemes reduce the computationally complexity such that either a problem can be solved more quickly or a more complicated problem can be solved that may otherwise have been computationally intractable.

Consider again Eq. 41 with the intent to reduce model dimensionality by reducing the spaces of the problem. Similar to homogenization, a simple reduction in numerical quantities such as the number of mesh divisions in space, energy, or angle can reduce model dimensionality at a sacrifice of accuracy; however, there are techniques that can, without making the model numeric coarser, reduce model dimensionality. The input parameter space of dimension *n* could be restricted to some basis selected *a priori* or *a posteriori* with

dimension *r* where *r* is less than *n*. As long as the new basis does not significantly change the state-space or responses of interest, a ROM with a reduced dimensionality basis is feasible. An example of when this is possible is systems with high degrees of correlation among inputs, or systems where many inputs do not significantly contribute to the solution of the problem. Reduction of the input space is favorable when interested in completing a forward sensitivity analysis procedure as the calculation complexity reduces from *n* to *r*. Consider the simple example of a fast reactor system, the input space can effectively be reduced by removing the consideration of thermal cross-sections that do not contribute to the solution. The goal, however, is to automatically select components that are unimportant to be removed from the system.

State-space reduction attempts to restrict the model based on possible combinations of the state-space. For eigenvalue problems, any reduction in the state-space directly affects the model complexity and allows directly for reductions in the input and output spaces. For nuclear engineering problems, the flux (state-space vector) often has a much lower dimensionality than the number of mesh points in a problem, allowing for this sort of reduction. Additionally, any state-space reduction can be projected onto the eigenvalue matrices to constrain the model to a set of possible states.

Output space reduction methods constrain responses to a specific basis. Similar to the input space projections, the output space could be reduced from dimension *m* to dimension *r* based on the physics of the model. Typically the output response basis is selected *a posteriori* unless the responses are known to behave in a certain manner. Reduction of the output space is ideal for adjoint methodologies and for correlated outputs. Linear algebra principles can

estimate the dimensionality of a linear model using *subspace* methods or rank determination methods acting on the model.

Restating, the input and output space reduction methods attempt to restrict these vectors to a lower dimension by elimination or change of variables such as a basis transform. Consider a perturbed model $\Theta\left(\sigma_0 + \Delta\sigma\right)$ with *n* inputs, where only a subset of the inputs yield a change in the response:

$$\begin{gathered}\Theta\left(\sigma_0 + \Delta\sigma\right) \equiv \Theta\left(\sigma_0 + \Delta\sigma^{\|}\right) = \phi_0 + \Delta\phi \\ and \\ \Theta\left(\sigma_0 + \Delta\sigma^{\perp}\right) = \phi_0 \\ where \quad \Delta\sigma = \Delta\sigma^{\|} + \Delta\sigma^{\perp}\end{gathered} \tag{47}$$

The orthogonal perturbations yield no change in the model. If the parallel input terms are much lower dimension then the input space *n*, a ROM could be formed such that perturbations are constrained to the reduced input space of dimension *r*. Consider a projector **P** that represents input space of dimension *r*, the model can be rewritten to:

$$\begin{gathered}\tilde{\sigma} = \mathbf{P}^T \Delta\tilde{\sigma} \\ \Theta\left(\sigma_0 + \mathbf{P}\mathbf{P}^T \Delta\sigma\right) \equiv \Theta\left(\sigma_0 + \mathbf{P}\tilde{\sigma}\right) = \phi_0 + \Delta\phi\end{gathered} \tag{48}$$

According to Eq. 48 the model now is determined by the *r* inputs represented by $\tilde{\sigma}$. These schemes may be considered projection schemes (Schilders 2008, Gentle 2009, Abdel-Khalik 2004, Meyer 1999). Similar considerations can be made with an adjoint model to reduce the output space. ROM methods that operate on a reduced basis come up with the same effective

end result; however, the means to obtain the reduced basis differ depending on the physics and the approach chosen.

3.D.2 Subspace Methods

Subspace methods take advantage of correlations in high dimensional models via automatic *a posteriori* selection of basis functions, the foundations for a ROM. From linear algebra, the rank of a matrix (here representing the model or model ROM) can be determined by taking the inner product with linearly independent vectors and using a rank-revealing decomposition such as SVD or QR on the set of resulting vectors. Gaussian random perturbations are used to explore the spaces of a model due to satisfaction of randomness while maintaining superior orthogonality versus uniform random number distributions as shown in Fig. 8:

Consider a forward and adjoint model:

$$\begin{aligned} \Pi(\sigma,\phi)=0 \qquad & \Pi(\sigma+\Delta\sigma,\phi+\Delta\phi)=0 \\ \Pi^*(\sigma,\phi^*)=0 \qquad & \Pi^*(\sigma+\Delta\sigma,\phi^*+\Delta\phi^*)=0 \end{aligned} \tag{49}$$

Perturbations introduced into the model exact a change on the state-space in both the forward and adjoint model. Replacing this model with some arbitrary model that can be represented by a matrix, $\Theta(\Delta\sigma)=\Delta\phi$, the spaces of the model represented by Θ can be explored. Consider a matrix **G** of dimension m by n with rank $r << \min(m, n)$, and two Gaussian random matrices **U** of dimension $n \times p$ and V of dimension $m \times p$ where $p >$ r where:

$$\mathbf{GU} = \mathcal{R}(\mathbf{G}) = \mathbf{Y}$$
$$\mathbf{G}^T\mathbf{V} = \mathcal{R}(\mathbf{G}^T) = \mathbf{X} \tag{50}$$

In this representation, R denotes the range of the matrices **G** and $\mathbf{G}^T$. The rank of the right hand product can be at most *r*. A rank revealing decomposition such as QR or SVD will extract *r* basis vectors from **Y** and **X**. The basis vectors from **Y** form a basis for the output space in **G** or loosely Θ. Similarly the basis vectors from **X** form a basis for the input space in **G** or similarly Θ.

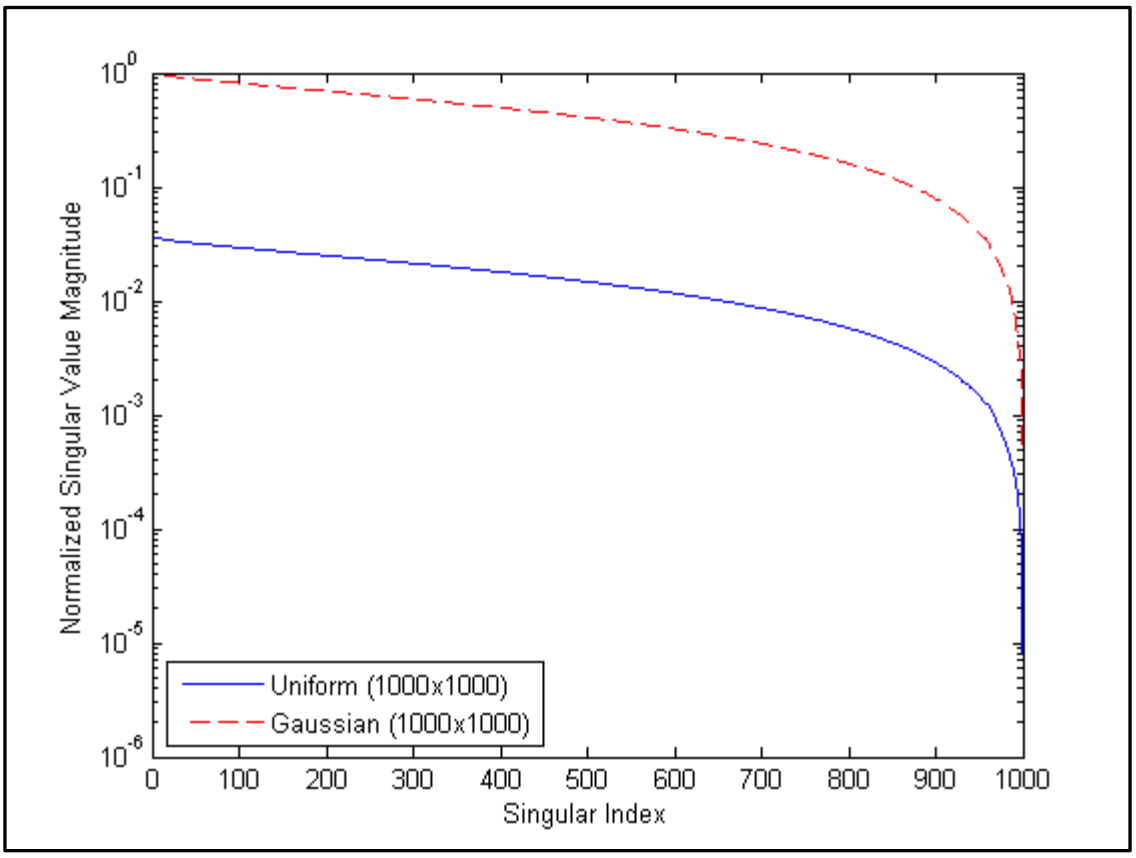

Figure 8: Singular value decomposition of two types of random matrices of dimension 1000x1000. A set of orthonormal vectors would be indicated by a flat horizontal line along *y =1*.

Using the properties of a linear algebra, any basis can be represented by another in a linear space.

$$\begin{aligned} \Delta\sigma_i &= \mathbf{X}_r \mathbf{a}_i \\ \Delta\phi_i &= \mathbf{Y}_r \mathbf{b}_i \end{aligned} \tag{51}$$

Such that once these basis are found of reduced dimension *r*, a change of basis or variables can be introduced to form the ROM.

One of the most prominent benefits of subspace methods is that the nature of Gaussian random perturbations or any orthogonal full-vector perturbation fully probes the system spaces. This means that only *r* runs are required to extract the basis as opposed to *n* or *m* runs using a standard approach of perturbing a single input/output at a time. Additional runs do not provide additional information about the basis and are only useful for determining stopping points if an appropriate error stopping metric is not used. This feature makes subspace methods extremely powerful compared to other methods where a basis is chosen before-hand or chosen according to some algorithm or Krylov subspace. In scenarios where there is not much reduction available or small reducible systems, subspace methods become less attractive.

4. GPT-FREE SENSITIVITY ANALYSIS

4.A GPT-Free Fundamentals

The dimension of nuclear engineering models is continuously increasing, making a sensitivity analysis (SA), in either the forward or adjoint mode, computationally expensive. Additionally, the generalized adjoint requires generalized perturbation theory (GPT), a construction that is often unavailable due to computational complexities and may be cost-prohibitive to implement into software. As a result, design calculations that require repeated execution may be less efficient, and some analysis cannot be completed in a reasonable amount of time with the desired level of accuracy. To address concerns where GPT is unavailable and/or for models with high-dimensionality, the GPT-Free approach has been developed in order to obtain sensitivity information in a computationally efficient manner.

The objective of the GPT-Free approach is to generate a reduced order model (ROM) using the fundamental (homogenous) adjoint that can calculate in a computationally efficient manner the variations or gradients (sensitivities) of all output responses generated by the model resulting from perturbations in all model input parameters. This is to be completed using classical perturbation theory (PT) and avoiding both the construction and the solution of the GPT equations. Classical PT requires only the execution of the forward model and the adjoint model for the homogenous eigenvalue problem. With the response sensitivities from GPT-Free, design optimization, uncertainty quantification, data assimilation, or other engineering tasks can be completed that may otherwise be computationally intractable or unavailable.

The GPT-free approach is based on the observation that the system multiplication, k, can be written as an unknown function of the flux, ϕ (the state-space of a neutronics calculation).

$$k = f(\phi) \tag{52}$$

Because most responses of interest to the nuclear engineer involve an inner product with the flux, the equation can be rewritten as a function of m arbitrary response functionals. Alternatively, the responses of interest can be taken as the flux, where m is the number of spatial and energy regions cells in the model.

$$k = f(R_1 \ldots R_m) \tag{53}$$

The responses are defined as an inner product between the flux and the cross-sections.

$$R = R(\langle \sigma, \phi \rangle) \tag{54}$$

This response functional expression can be extended to account for more general responses, e.g. bilinear ratios of responses:

$$R = R(\langle \sigma_1, \phi \rangle, \langle \sigma_2, \phi \rangle, \ldots, \langle \sigma_m, \phi \rangle) \tag{55}$$

Since the flux is a function of model parameters, both the system multiplication k and the responses are also expected to depend on the input parameters. For this work, the input parameters of interest are the nuclear cross-sections. The responses include multi-group fluxes and reaction rates at various spatial locations in the given model.

Differentiate Eq. 53 with respect to the microscopic cross-sections to obtain gradient information or the sensitivity profile for k:

$$\frac{dk}{d\sigma} = \sum_{i=1}^{m} \frac{\partial k}{\partial R_i} \frac{dR_i}{d\sigma} \tag{56}$$

In this representation, the left hand side of the equation is a gradient or sensitivity profile of the *k*-eigenvalue. The sensitivity profile for *k* is with respect to the isotope, energy group, and reaction specific microscopic cross-sections. The right hand side of the equation is a linear combination of weighted response sensitivity profiles, $dR_i/d\sigma$. The weighting terms, $\partial k/\partial R_i$, are scalar quantities representing the derivative of *k* with respect to the i^{th} response, which is expected to be a nonlinear function of cross-sections, composition, and geometry. Equation 56 implies that the *k*-eigenvalue sensitivity profile can be written as a linear combination of all response sensitivity profiles.

It can be shown with first order accuracy that an input parameter perturbation orthogonal to arbitrary response sensitivity yields no change in the response (Kennedy 2012).

$$\Delta R_i = \Delta\sigma^T \left(\frac{dR_i}{d\sigma} \right) = 0 \quad if \quad \Delta\sigma \perp \left(\frac{dR_i}{d\sigma} \right) \tag{57}$$

Each response sensitivity profile in the summation for the *k*-eigenvalue sensitivity profile in Eq. 56 represents an input parameter direction that changes a system response. If one can show that the sensitivity profiles for all responses span a subspace of size $r << n$, the number of inputs, a ROM could be constructed to restrict input perturbations along a basis of dimension r[§]. If *r* is relatively small, one could employ a forward rather than an adjoint sensitivity analysis to calculate response variations and/or sensitivities. The subspace relationship for Eq. 56 and Eq. 57 is shown graphically in Figure 9.

[§] For more on the math of subspaces, bases, and low-rank matrices as it pertains to ROM, consult Meyer (1999).

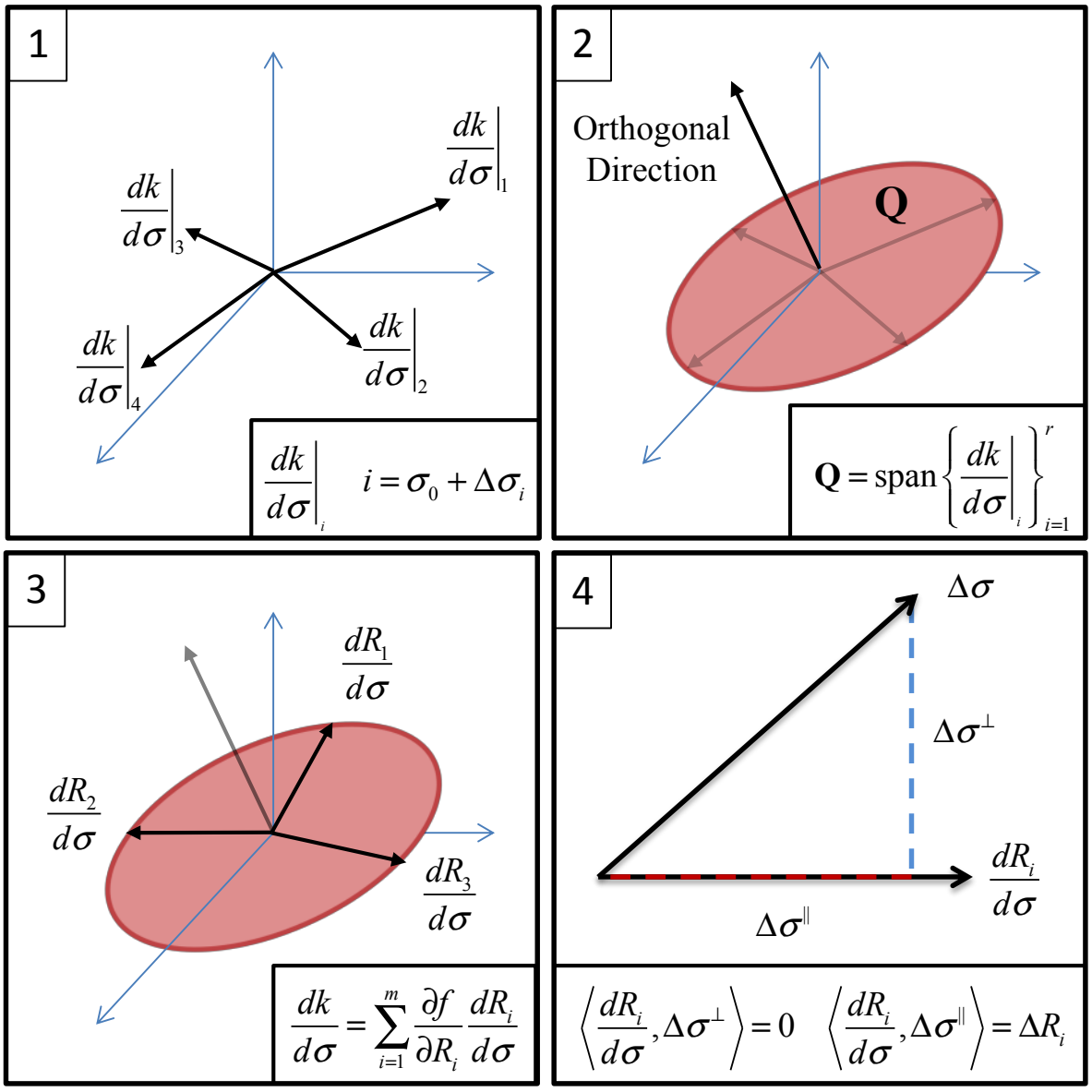

Figure 9: Visual demonstration of Subspace approach to GPT-Free ROM formation. 1) Cross-section and number density perturbations yielding various sensitivity profiles. 2) The span of the set of sensitivities form a subspace Q. 3) Relationship of Eq. 56 demonstrating that response sensitivities exist in the same subspace. 4) Restriction of input parameters to the subspace Q for responses as in Eq. 57.

At a first glance, it might appear counter-intuitive that the input parameters space represented by cross-sections can be reduced significantly. Success and experience with few-group cross-section models for modeling real reactors demonstrates that the cross-section data can be reduced at least in energy (Stamm'ler 1983). Furthermore, recent research (Kennedy 2011, Abdel-Khalik 2011, Kennedy 2012) has demonstrated that flux variations generated from all possible cross-section perturbations can be accurately represented by a subspace. To identify this reduced order model subspace, a result from linear algebra (Meyer 1999, Golub 1996) is employed – that is any subspace can be represented by an alternative basis.

Construction of the ROM basis requires the *k*-eigenvalue sensitivity profile to change in order to construct the spanning set. The most straightforward manner to introduce changes into the sensitivity profile is to introduce perturbations into the cross-sections and nuclide number densities to change the weighting factors, $\partial k/\partial R_i$, from Eq. 56 and thus stimulate the various response sensitivity profiles. As these perturbations are introduced, the eigenvalue sensitivity profiles can be collected and a new basis can be formed. A straightforward question is how many perturbations are required until a sufficient input basis is found?

A range-finding algorithm (RFA) is required to determine stopping criteria for the input basis. A tolerance error metric, denoted the κ-Metric (Kennedy 2011), is explained in later sections in detail, but in brief summary, the input perturbations continue until the input model basis sufficiently describes the system response to some user denoted tolerance.

Consider a steady-state neutronics model and a response functional:

$$\Pi\left(\phi,\sigma\right)=0$$
$$R=\left\langle \phi,\sigma \right\rangle \tag{58}$$

The flux, ϕ, which is the state-space in the state equation, is dependent on both input parameters and problem geometry. The response may be either global or local over some region of the reactor. Examining some response of interest, consider a first order perturbation:

$$\Delta R=R\left(\sigma_0+\Delta\sigma,\phi\right)-R\left(\sigma_0,\phi\right) \tag{59}$$

Restricting this response to some input subspace, **Q** with rank *r***:**

$$\Delta R=R\left(\sigma_0+\mathbf{Q}\mathbf{Q}^T\Delta\sigma\right)-R\left(\sigma_0\right) \tag{60}$$

If the subspace represented by **Q** is sufficiently small for all responses, then response sensitivities could be calculated for the *r* input directions according to **Q** by running a forward sensitivity analysis. GPT-Free attempts to obtain a ROM based on automatic basis reduction.

Consider a set of *p k*-eigenvalue sensitivities formed from running the forward/adjoint model using cross-sections of dimension *n*, corresponding to the isotope, reaction, energy specific microscopic cross-sections, perturbed independently from a Gaussian distribution. Collect the set of *p* sensitivity profiles into a matrix $\mathbf{Z}\in\mathbb{R}^{n\times p}$ to form Eq. 61.

$$\mathbf{Z}=\left[\left.\frac{dk}{d\sigma}\right|_1\left.\frac{dk}{d\sigma}\right|_2\cdots\left.\frac{dk}{d\sigma}\right|_p\right] \tag{61}$$

Consider a matrix **R** of response sensitivities with *m* columns and a vector of weighting factors, **w**. The matrix **Z** can be represented by **R** and the inner product of various **w**.

$$\mathbf{R} = \left[\frac{dR_1}{d\sigma} \cdots \frac{dR_m}{d\sigma} \right] \quad \mathbf{w} = \left[\frac{\partial k}{\partial R_1} \cdots \frac{\partial k}{\partial R_m} \right]^T \tag{62}$$
$$\mathbf{Z} = \left[\mathbf{R}\mathbf{w}_1 \; \mathbf{R}\mathbf{w}_2 \ldots \mathbf{R}\mathbf{w}_n \right]$$

If the rank of **Z** is sufficiently small, the rank of **R** is likewise small, and a subspace **Q** can be determined from **Z** to form a reduced order model for the inputs. Consider a matrix **Q** as determined from the SVD of Z, where the first *r* columns of **U**, denoted **Q,** represent the range of **Z**:

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Q} \\ \mathbf{U}_N \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_r & 0 \\ 0 & \mathbf{\Sigma}_N \end{bmatrix} \begin{bmatrix} \mathbf{V}_r^T & \mathbf{V}_N^T \end{bmatrix} \quad \mathbf{Q} = \mathcal{R}(\mathbf{Z}) \in \mathbb{R}^{n \times r} \tag{63}$$

This can be shown when considering the effect of a cross-section perturbation on the transpose of **Z**:

$$\left(\mathbf{Z}_{*i}\right)^T \Delta\sigma = \Delta k_i \quad = \left(\frac{dk}{d\sigma} \right)_i^T \Delta\sigma = \Delta k \tag{64}$$

If the range of **Z** is sufficiently small, it may be computationally efficient to complete a forward sensitivity analysis along the first *r* directions of **Z** which is represented by **Q** as determined from a rank-revealing decomposition such as the QR or SVD[**]. In order to meet this objective, a sufficient estimate for the rank of ***Z*** must be available. To do this, a subspace termination criterion is required.

[**] For more on rank revealing decompositions such as SVD or QR, see Meyer (1999).

4.B Subspace Termination Criterion: κ-Metric

Accurate determination of the effective rank of a ROM is a prerequisite for the practical application of the GPT-free approach. Past work has primarily employed an engineering judgment type approach based on trial and error for rank determination (Kennedy 2011, Abdel-Khalik 2011). In particular, a truncated SVD (Meyer 1999) approach was employed where the rank is determined such that reconstruction error for the response of interest is less than some user-defined or response target accuracy. The rank is increased until the target accuracy is reached. For the purpose of a rigorous method, a new more detailed approach has been constructed in this dissertation.

Consider an arbitrary response metric where some response vector, f, is compared to the reference value f_0:

$$\kappa = \left|\left| f - f_0 \right|\right| \tag{65}$$

In engineering applications, a tolerance or criteria can be set such that κ can be sought to be less than some defined value $\mathbf{U}$.

$$\kappa \overset{def}{=} \left|\left| f - f_0 \right|\right| \leq \epsilon \tag{66}$$

For any reduced order model Θ, one can assume that the output has some expected value $\mathbf{y}$ and some associated error distribution $\mathbf{e}$ given inputs $\mathbf{x}$:

$$\Theta(\mathbf{x}) = \mathbf{y} + \mathbf{e} \tag{67}$$

Here the vector $\mathbf{e}$ could be a normally distributed error vector with some possibly unknown standard deviation ς. It is possible that with a single realization of the model the metric can be met as an outlier of Θ. When the underlying distribution for $\mathbf{e}$ is unknown (as often is the case for a ROM), a single measure of Θ is even riskier to trust for some metric as defined in Eq. 66. To visualize this effect, consider Fig. 10. In this figure a ROM is considered based on true model evaluations with a specified error tolerance bound.

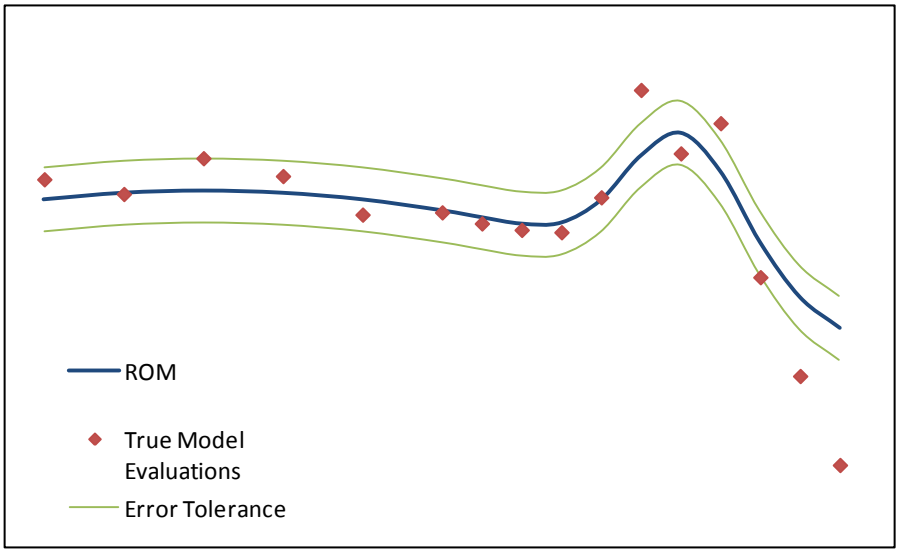

Figure 10: A comparison of a hypothetical model and an associated ROM with some error tolerance.

If a single evaluation had been taken from the ROM it is likely to have been considered to be satisfactory. In order to reduce the risk of a 'false positive' acceptance of a ROM, an approach based on Wilks' order statistics has been applied to metrics of the form in Eq. 66 to provide better evidence that a ROM meets the specified tolerance.

Wilk's order statistics allows one to assign a tolerance interval for a response of interest and a probability for meeting this tolerance given an unknown distribution (Wilks 1941). The only practical requirement for Wilks' order statistics is that the underlying distribution is continuous and samples from the distribution are random. Exceptions that violate this criterion could be an ROM where the software does not run, an illegal operation contaminates the data to result in NaN or zero, or a non-physical output occurs near local points in the space of evaluation. By arguments given prior for a reduced order model, this is typically satisfactory with nuclear engineering calculations. Given some ROM and some metric, the minimum number of samples for some confidence in the metric can be determined from Wilks' formula (Wilks 1941, Strydom 2011):

$$(1-p^N)-N(1-p)p^{N-1}\geq c \tag{68}$$

where p is the probability percentile, N is the number of samples, and c is the confidence. The κ-metric is defined as a measure of the accuracy of the ROM of dimension r. To evaluate the rank of a model using this metric, reform Eq. 66:

$$\kappa_c \overset{def}{=} \left\{\left\|f_i - f_0\right\| \leq \epsilon\right\}_{i=1}^{n} \quad \text{Accept if at least } \left(pn\right) \text{ pass} \tag{69}$$

Here Eq. 69 requires pn of n samples to meet the tolerance and provides a confidence c in the estimate of the metric. In summary, the κ-metric tests the error between the ROM and the true model by sampling the error **e** in Eq. 67 and using the property of Wilks' order statistics to make engineering claims based on the resulting distribution of **e** in the form of Eq. 69 (Halko 2009, Wilks 1941).

For the purpose of GPT-Free, three metrics are of given as examples for construction of a ROM from sensitivity data. Global responses, k and ϕ, are used for determining the ROM rank. The κ-metric for the k-eigenvalue is straightforward and best viewed in units of *pcm* where the tolerance is set at or near the convergence criteria for k:

$$\kappa = \left(\underbrace{10^5 \times \left| \frac{k - k_0}{k_0} \right|}_{|pcm|} \right) \leq \epsilon_{tol} \tag{70}$$

For vector quantities such as the flux, two different metrics have been utilized in this technical report. The first is a 2-norm metric normalized to the number of cases n. The second is an absolute average error metric visualized for each test case. They are defined as:

$$\kappa_{\phi 1} = \frac{1}{n} \left\| \left| \frac{\phi - \phi_0}{\phi_0} \right| \right\|_2 \tag{71}$$

$$\kappa_{\phi 2,i} = \frac{\sum_{j \in \rho_x} \left| \phi_{ij} - \phi_{0j} \right|}{\sum_{j \in \rho_x} \phi_{0j}} \equiv \frac{\sum_{j \in \rho_x} \left| \Delta \phi_{ij} \right|}{\sum_{j \in \rho_x} \phi_{0j}} \tag{72}$$

These metrics can also be normalized to a set of reference perturbation cases; however, the same result is obtained if the engineering tolerance is properly defined/scaled to the accuracy of the convergence criteria in the software or to some output response of interest.

While these metrics have been defined, a metric can be defined for any response of interest. Using this metric, any response error can be quantified in this manner. Consider a

response for a single-group fission cross-section for U-235, the error in the model could be estimated similarly for some selected ROM:

$$\kappa = \left|\widetilde{\sigma}_f - \sigma_{f0}\right| \equiv \left| \frac{\sum_{g=1}^{G} \tilde{\sigma}_{f,g} \sum_{j\in\rho_x} \phi_{j,g}}{\sum_{g=1}^{G} \sum_{j\in\rho_x} \phi_{j,g}} - \frac{\sum_{g=1}^{G} \sigma_{f,g} \sum_{j\in\rho_x} \phi_{0j,g}}{\sum_{g=1}^{G} \sum_{j\in\rho_x} \phi_{0j,g}} \right| \tag{73}$$

From this evaluation with some given ROM, an estimate in the error for this response for all simulations can be predicted. This capability is important because once an ROM is selected using the metrics for ϕ and k, all responses will have some inherent errors due to model reduction, regardless of the model/approximations used for sensitivities.

In review, The κ-metric represents the absolute error in the response of interest resulting from constraining input parameter perturbations to a subspace. An upper bound on this metric could easily be set since target accuracy requirements are often available for the responses of interest.

4.C GPT-Free Algorithm

4.C.1 ROM Construction

Consider a model that calculates the k-eigenvalue sensitivities with respect to l nuclides, h reaction types, and g groups. Define the k-eigenvalue sensitivity vector z as:

$$z \in \mathbb{R}^{n=(l\times h\times g)} \rightarrow z = \frac{\partial k}{\partial \sigma} \tag{74}$$

where n is the total number of cross-section types in all groups and for all isotopes and reactions. Now, introduce random Gaussian perturbations, uniquely for each nuclide, reaction, and energy group, into the cross-sections to construct $\Delta\sigma_i$, and re-evaluate the k-eigenvalue sensitivity vector:

$$z_i = \left.\frac{dk}{d\sigma}\right|_{\sigma_0+\Delta\sigma_i} \tag{75}$$

The subscript i refers to the i^{th} cross-sections perturbation. Notice that as implied by Eq. 56, the vector z_i is a random linear combination of the unknown responses sensitivity vectors. Next, repeat this process p times, and construct a matrix:

$$\mathbf{Z} \in \mathbb{R}^{n\times p} = \left[z_1 \dots z_i \dots z_p \right] \tag{76}$$

In this process, p is increased until an effective rank for the matrix $\mathbf{Z}$ is reached. An effective rank implies that the matrix $\mathbf{Z}$ could be approximated by matrices of lower dimensions such that the maximum discrepancy between the original matrix and the approximation does not exceed a user-specified tolerance. We employ here the rank-revealing singular value decomposition (SVD) of the form:

$$\mathbf{Z} \approx \mathbf{U}_r \mathbf{W}_r \mathbf{V}_r^T \rightarrow \quad \mathbf{W}_r \in \mathbb{R}^{r\times r} \text{ and } \mathbf{U}_r \in \mathbb{R}^{n\times r} \tag{77}$$

The rank r is considered an effective rank for the matrix $\mathbf{Z}$ when the κ-metric for k and ϕ is satisfied using the input space defined by the first r columns of $\mathbf{U}_r$. The implication of this

decomposition is that instead of having *n* input parameters that can change the response, only a small number *r*, represented by the columns of the matrix $\mathbf{U}_r$ is shown to capture all possible variations of the responses to a given tolerance.

Consider the governing equation for the flux:

$$\Pi\left(\sigma,\phi\right)=0 \tag{78}$$

where σ denotes all cross-sections. Write the response of interest as a linear function of the flux:

$$R=\left\langle \phi,\sigma \right\rangle \tag{79}$$

Without loss of generality, the discussion below will focus on a single response, as implied by Eq. 79, to simplify the resulting expressions. The procedure however is applicable to all model responses that are described as functions of the flux. The ROM is given by:

$$\Pi\left(\sigma_0+\mathbf{U}_r\mathbf{U}_r^T\Delta\sigma\ ,\phi\right)=0 \tag{80}$$

$$R=\left\langle \phi,\sigma_0+\mathbf{U}_r\mathbf{U}_r^T\Delta\sigma \right\rangle \tag{81}$$

This representation implies that the cross-sections are constrained to vary along a subspace that is spanned by the columns of the matrix $\mathbf{U}_r$. This reduces the effective number of parameters to *r*, where now each effective parameter represents a perturbation of all cross-sections along one of the columns of the matrix $\mathbf{U}_r$.

To test this ROM with rank *r*, the *n* – *r* columns of **U** are used to construct perturbations that, according to the ROM under scrutiny, will yield no change in the system:

$$\Pi\left(\sigma_0+\left(\mathbf{I}-\mathbf{U}_r\mathbf{U}_r^T\right)\Delta\sigma,\phi\right)\approx\Pi\left(\sigma_0,\phi\right) \tag{82}$$

$$R=\left\langle \phi,\sigma_0+\left(\mathbf{I}-\mathbf{U}_r\mathbf{U}_r^T\right)\Delta\sigma \right\rangle\approx R=\left\langle \phi,\sigma_0 \right\rangle \tag{83}$$

The metrics of interest are those as described by Eq. 70-72. The procedure is described as follows for some rank estimate r for $\mathbf{Z}$:

1. Using Eq. 68, determine N given (p, c)
2. Generate N random Gaussian input parameter perturbations $\left\{\Delta\sigma_i\right\}_{i=1}^{N}$
3. Calculate the orthogonal components: $\Delta\sigma_i^{\perp} = \left(\mathbf{I} - \mathbf{U}_r\mathbf{U}_r^T\right)\Delta\sigma_i \quad i = 1\ldots N$
4. Execute the model n times to obtain k and ϕ
5. Calculate the κ metric for k and ϕ according to Eq. 70-72.
6. Determine the number of times M in which the metric is outside the user-defined confidence interval.
7. If $M > N(1-p)$, then the estimated rank is insufficient to meet user-defined accuracy requirements. Increase the rank by increasing the dimension of p and $\mathbf{Z}$, and return to step 3 with an updated $\mathbf{U}_r$.

The remaining $n - r$ vectors which span the rest of the space are assumed to have a negligible impact on the model state-space and response-space. Once this ROM has been generated that meets the criteria of the κ metric for some rank r, the first r columns of $\mathbf{U}$ represent the input space, effectively a reduced basis ROM along which a forward sensitivity analysis can proceed.

4.C.2 Sensitivity Analysis Procedure

This subsection describes how the ROM is used to produce sensitivities of the responses using a forward sensitivity analysis approach. First re-write Eq. 81 in a compact form recognizing that the flux variations depend on the cross-section variations:

$$R = R\left(\sigma_0 + \mathbf{U}_r \mathbf{U}_r^T \Delta\sigma\right) \tag{84}$$

This representation ensures that cross-section variations that are orthogonal to the columns of the $\mathbf{U}_r$ matrix do not change the responses; hence they need not be perturbed when employing a forward sensitivity analysis. This could be achieved by defining a set of *r* pseudo (denoted earlier as effective) input parameters defined by:

$$\mathbf{x}_i = \mathbf{u}_i^T \Delta\sigma \tag{85}$$

This equation implies that by perturbing the i^{th} pseudo input parameter, all cross-sections are perturbed along the direction $\mathbf{u}_i$. Rewrite Eq. 84 as:

$$R = R\left(\sigma_0 + \mathbf{U}_r \mathbf{x}_i\right) \tag{86}$$

This equation states that the response of interest is a function of *r* pseudo input parameters. If *r* is small, one could employ a forward sensitivity analysis instead of a GPT-based adjoint sensitivity analysis to determine the first order derivatives of the response with respect to the pseudo input parameters. And since the relationship between the pseudo input parameters and original input parameters is linear, as given by Eq. 85, one could back-calculate the responses derivatives with respect to the original input parameters using the chain rule of differentiation.

Some measure of caution must be taken when using the chain rule to determine sensitivities of the original cross-sections. The numerical limitations of this chain rule must be considered: columns of U span orders of magnitude that easily exceed single-precision and sometimes can push the limits of double precision. Because cross-sections in the SCALE code are stored in single-precision, some sensitivities are filtered out in the forward calculation. This result is desirable, because they will not contribute to a change in the result; however, when using the chain-rule, these low importance sensitivities return as noise near the numerical limits of the sensitivity analysis. Consider by means of an example of Figure 11 for a 44 energy group model. In this figure, any sensitivity components in the intermediate to fast energy ranges will effectively be filtered out. A change in the model will likely be observed due to the magnitude of the thermal regions of this hypothetical column of **U**.

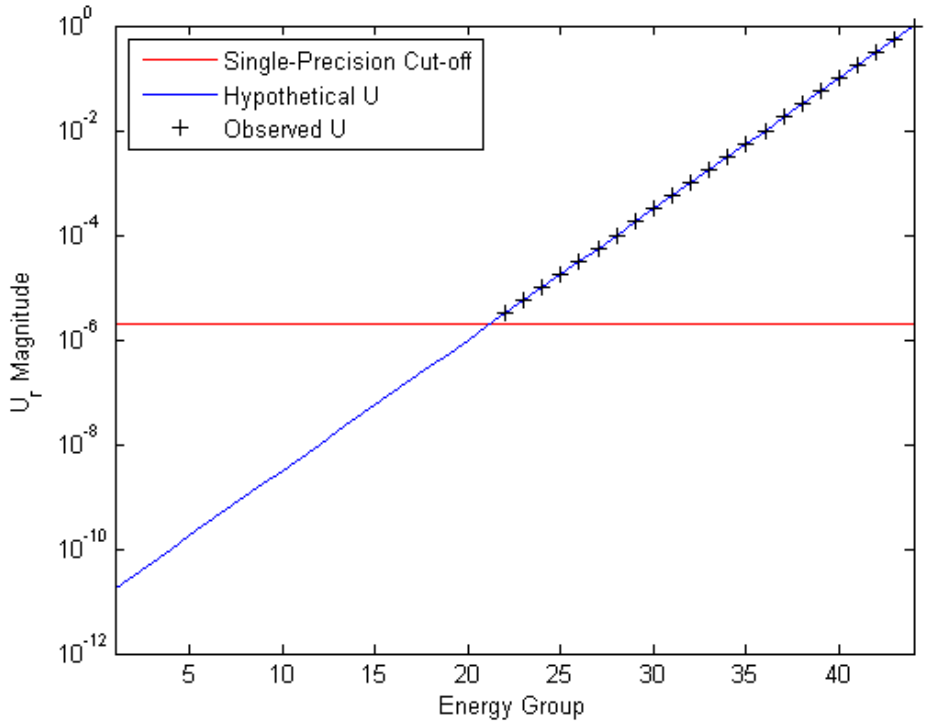

Figure 11: A hypothetical column of U_r for a single isotope-reaction. The numerical limits for a single-precision cross-section perturbation of magnitude 5% is included for comparison.

However, consider that the sensitivity information for intermediate and fast ranges (groups 1 to 20) will not change the model: the actual sampled sensitivity is the set of plus markers in the figure. The problem then arises in situations as in Figure 12:

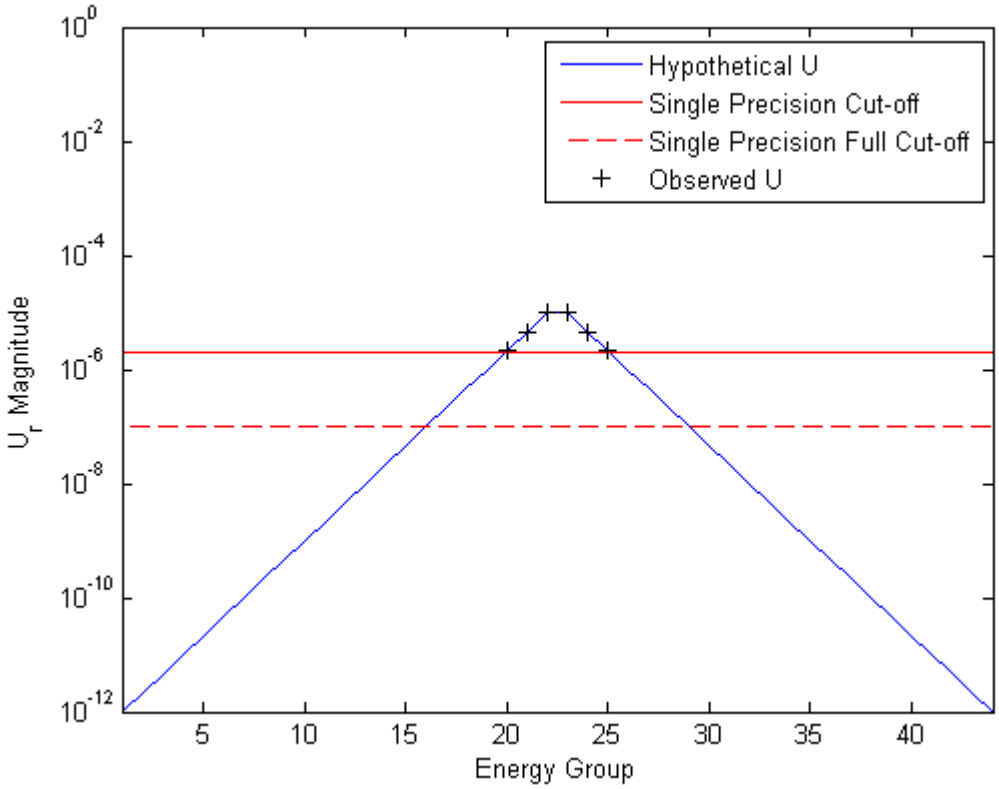

Figure 12: A hypothetical column of U_r for a single isotope-reaction. The numerical limits for a single-precision cross-section perturbation of magnitude 5% is included for comparison.

In Figure 12, the observed change in the model is likely to be small; however, due to Gaussian sampling, some samples may bleed along the lines of the numerical tolerance causing sensitivities for energy groups 15-20 and 26-30 to lack fidelity. When these appropriate sensitivities are back-calculated using the chain-rule, their noise will contaminate nearby data for energy groups 20 – 25. Effectively, only the largest sensitivities can be accepted; even though the ROM may suggest numerical tolerance is achievable to a certain level, the sensitivity analysis, in its current form, has numerical limitations that must be

considered. To avoid this problem, the best solution is instead to use sensitivities in the form of **U** instead of back-calculating to the original cross-sections.

The literature is replete with many techniques that can be employed to conduct a forward sensitivity analysis (Cacuci 1980). We employ a regression-based approach in this work. The idea behind regression techniques is fairly simple: first, one assumes a regression surface that approximates the relationship between the responses of interest and input parameters; the surface has a number of undetermined coefficients that can be related to the response derivatives; second, the forward model is executed a number of times that is greater than or equal to the number of unknown coefficients to render a well-posed problem; finally, the method of least-squares is employed to determine the coefficients (Seber 2003, Bates 1988).

The primary source of error in this approach lies in the choice of the regression surface. Given that we are primarily interested in first-order derivatives evaluated locally around some reference values for the cross-sections, a linear regression surface is employed with all cross-sections perturbations employed to generate the coefficients small enough to ensure the linearity approximation is valid. The surface selected is given by:

$$\Delta R = \sum_{i=1}^{r} \frac{\partial R}{\partial x_i} \Delta x_i \tag{87}$$

where $\partial R/\partial x_i$ is the unknown derivative with respect to the i^{th} pseudo input parameter. Now, consider a set of J random cross-sections perturbations $\{\Delta\sigma_i\}_{i=1}^{J}$, calculate the corresponding change in the pseudo input parameters as:

$$\mathbf{x}_i = \mathbf{U}_r^T \Delta\sigma_i \quad i = 1 \ldots J \quad \mathbf{X} = [x_1 \ldots x_J] \tag{88}$$

Next, execute the forward model j times and calculate the corresponding change in the response:

$$\Delta R_i = R(\sigma_0 + \Delta\sigma_i) - R(\sigma_0) \quad i = 1 \ldots J \tag{89}$$

Using the method of least-squares and the Moore-Penrose pseudo inverse (Meyer 1999), determine the unknown coefficients in the regression surface as follows, where Δr is the vector of J response deltas.

$$\frac{dR}{dx} = \left(\mathbf{X}^T\right)^{\dagger} \Delta r \tag{89}$$

Finally, using the chain rule of differentiation, back-calculate the response sensitivity profile as:

$$\frac{dR}{d\sigma} = \frac{dR}{dx}\frac{dx}{d\sigma} = \mathbf{U}_r \left(\mathbf{X}^T\right)^{\dagger} \Delta r \tag{90}$$

The above regression surface is constructed for a single response. For multiple responses, e.g. flux distribution, the analysis could be easily repeated for all responses of interest. Note that one needs J forward model executions only regardless of the number of responses. This follows as in each forward model execution all responses variations are available which represents one of the advantages of forward sensitivity analysis. The extra computational efforts would be the solution of the minimization problem in Eq. 91 for each given response which is computationally cheap compared to the execution of the forward model. Finally, the value of J is often selected to be slightly larger than the number of unknowns, r, in order to

render an over-determined least-squares problem that is less sensitive to the numerical noise of the calculation.

4.C.3 Concluding Notes on the GPT-Free Sensitivity Analysis Algorithm

In conclusion of this section, we remark that there are two sources of error in the GPT-free-generated sensitivities. The first source occurs during the construction of the subspace. As mentioned earlier, this error could be controlled by expanding the subspace until the maximum error meets a user-defined criterion. The second source of error results from the regression step. This error is directly related to the choice of the regression surface. For neutronics problems with small cross-sections perturbation, this error source is expected to be small given the adequacy of the linearity assumption. If one desires to employ the GPT-free approach for a general nonlinear model, more sophisticated strategies should be employed in the selection of the regression surface. We recognize here that a great deal of research in the applied mathematics and statistical communities has been conducted over the years to address this very issue in a general context (Seber 2003, Bates 1988, Ghanem 1991). By way of few examples, response surface methods such as polynomial chaos and stochastic collocation and their gradient-enhanced implementations have proven effective as regression surfaces. The focus in this report is on a simple linear regression model, leaving the implementation of the GPT-free approach to general nonlinear models to future work.

5. GPT-Free Extensions

5.A Multi-Region Analysis

When considering the effect of a local perturbation in a global system for the purpose of constructing a reduced-order model, tests were derived using the UAM assembly model. If the ROM formed from a single pin perturbation is sufficiently small and the effects on neighboring pins is similarly small, it is expected that the ROM for an entire assembly should by extension be low-dimensional. The effect of a local pin perturbation an assembly was undertaken using the modified, specifying each fuel-pin with a unique mixture, UAM assembly model (Ivanov 2007). The graphic of the assembly model as output from NEWT is shown in Fig. 13.

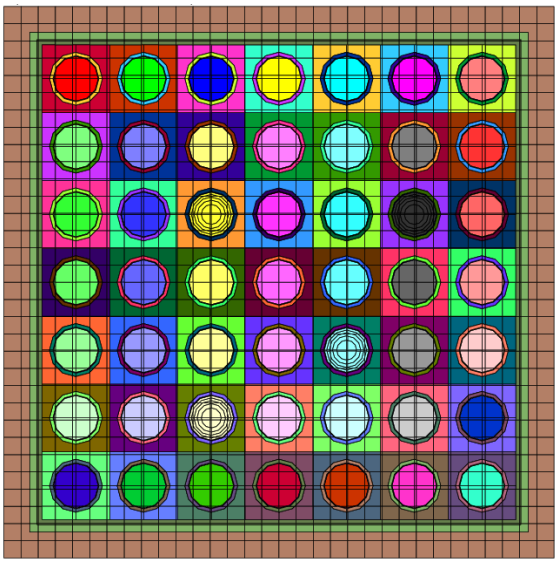

Figure 13: UAM assembly model: mixture-dependent perturbation model.

Introducing 10% RMS Perturbation of Uranium fission and capture cross-sections in a central pin introduces local and global changes to the sensitivity profile in the assembly system. Because of the nonlinearity in the model, the sensitivity profile changes for the system; however, changes external to the center pin are about 2 orders of magnitude smaller for neighboring pins and about 3 orders smaller for the remaining pins. A total of 64 different perturbation cases were studied to determine these effects. Indexing pins by row, the average relative changes in sensitivities for each pin cell is recorded in Fig. 14. The perturbed pin (25) is exhibiting changes of the same order of magnitude as cross-section perturbations or about 10% RMS; however, the relative change in sensitivities of the four adjacent pins are between 1.5 to 2.5 orders of magnitude smaller. Pins further away are 2 to 3.5 orders of magnitude less. A sample of four representative perturbations is plotted in Figure 15.

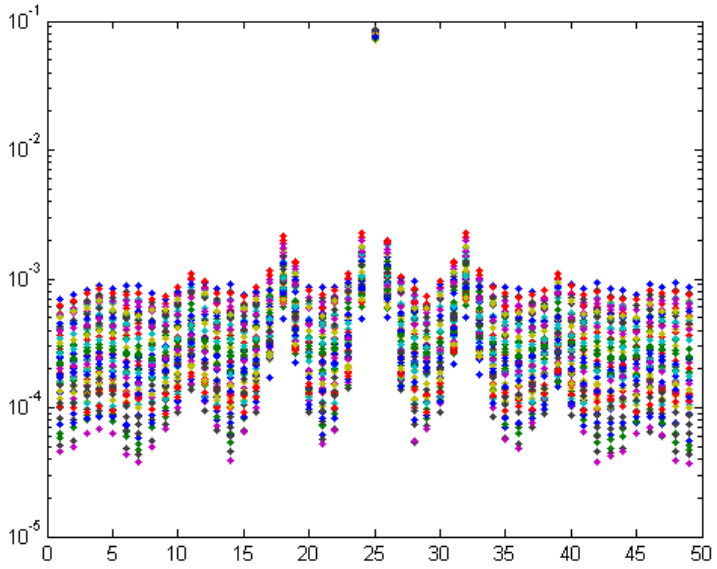

Figure 14: Perturbations in pin 25: Average relative pin sensitivity changes.

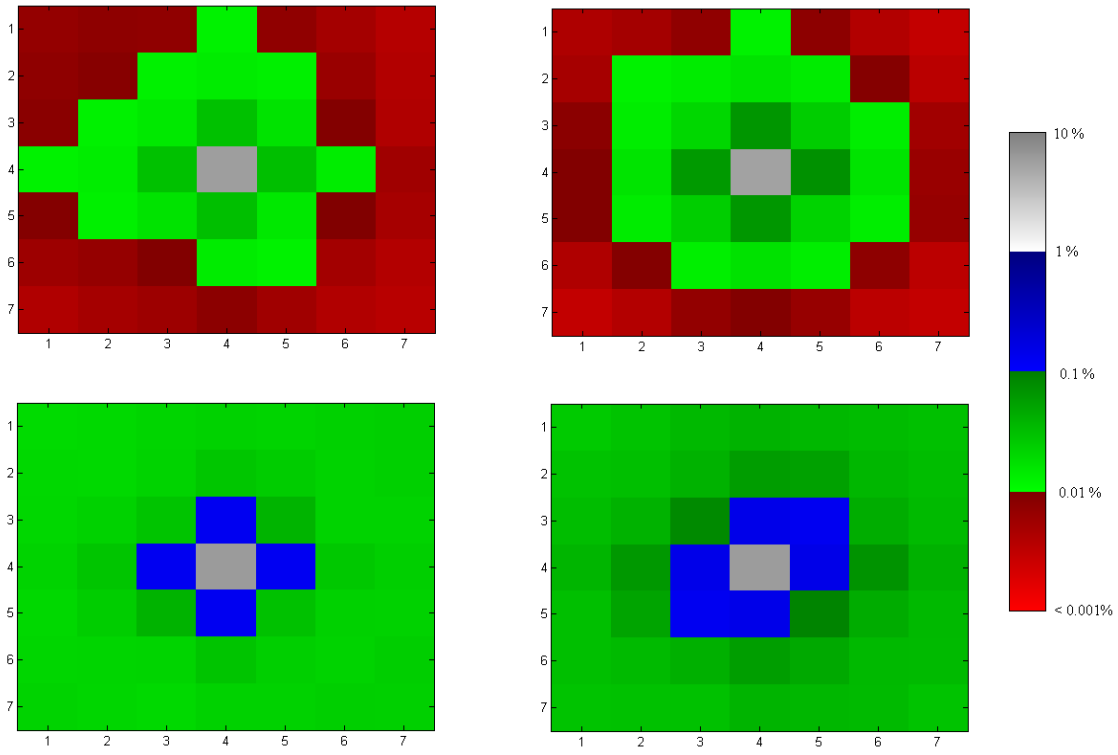

Figure 15: Average pin sensitivity magnitudes of four sample center-pin perturbations.

Given the nature of the sensitivity results, a ROM was formed by introducing perturbations globally in the assembly and examining a single pin-cell for study. The κ-metric for the *k*-eigenvalue and the flux is demonstrated with increasing ROM dimension in Figures 16 and 17. The ROM dimension necessary to reduce changes in *k* to the specified software tolerance of 10^{-6} is around the order of 20; however, the flux 2-norm result reduces more slowly, suggesting a larger dimension ROM depending on the tolerance requested. A more instructive plot is shown in Figure 18 demonstrating the flux data point-wise reduction in error via increased ROM throughout the assembly. Reduction is possible with lower order models depending on the desired tolerance.

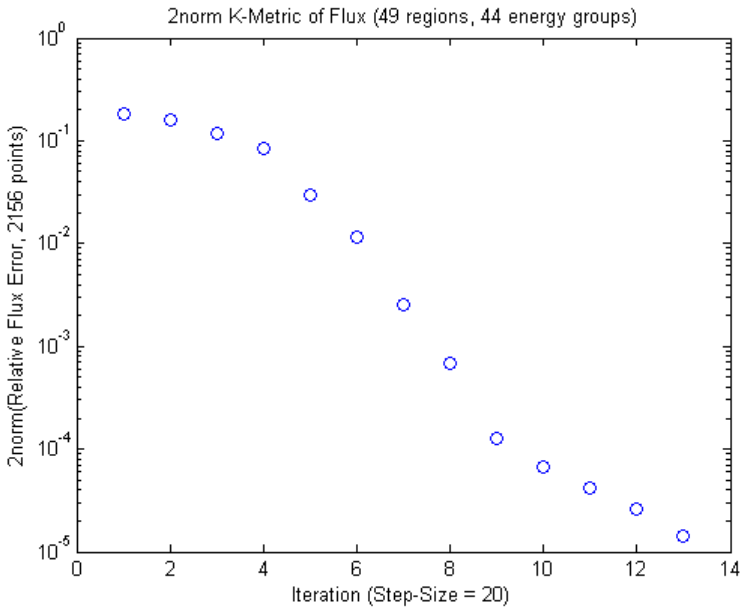

Figure 16: Single-pin 2-Norm flux κ-metric.

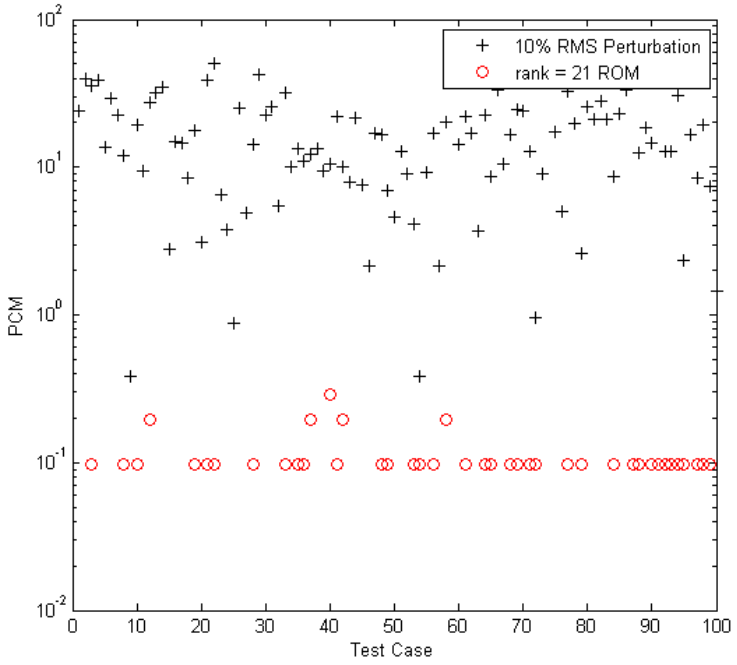

Figure 17: Single-pin *k*-eigenvalue κ-metric.

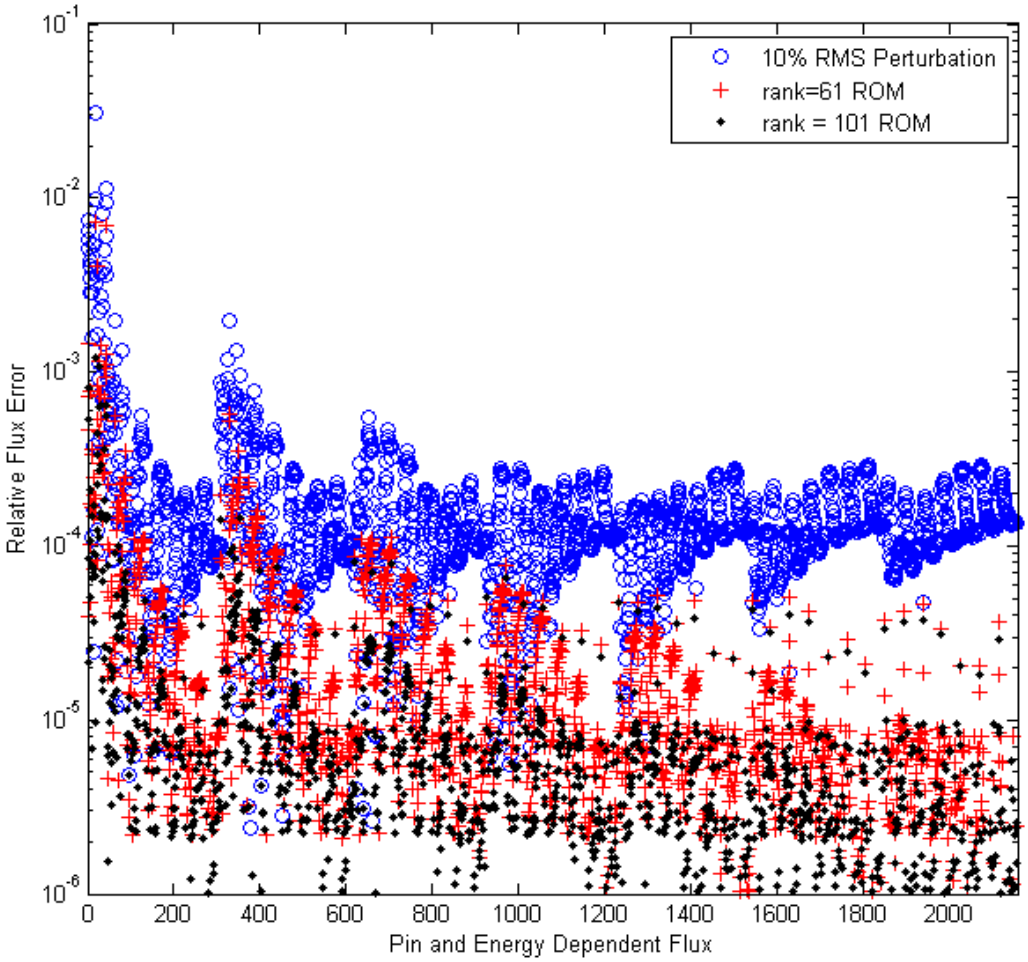

Figure 18: Pointwise flux relative plots from a single-pin perturbation for various ROM.

Applying the GPT-Free methodology to the full assembly system as opposed to a single fuel-pin does not significantly increase the ROM dimension. The modification was made by considering each fuel-pin a unique material, such that the ROM could examine each region independently. Figures 19, 20, and 21 are similar to Figures 16 and 17 demonstrating only a mild increase in ROM dimension due to a full-assembly examination.

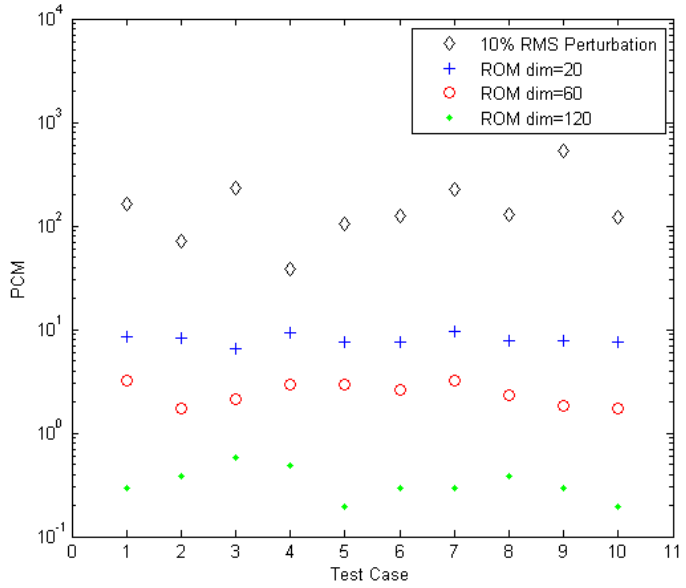

Figure 19: Full-core *k*-eigenvalue κ-metric.

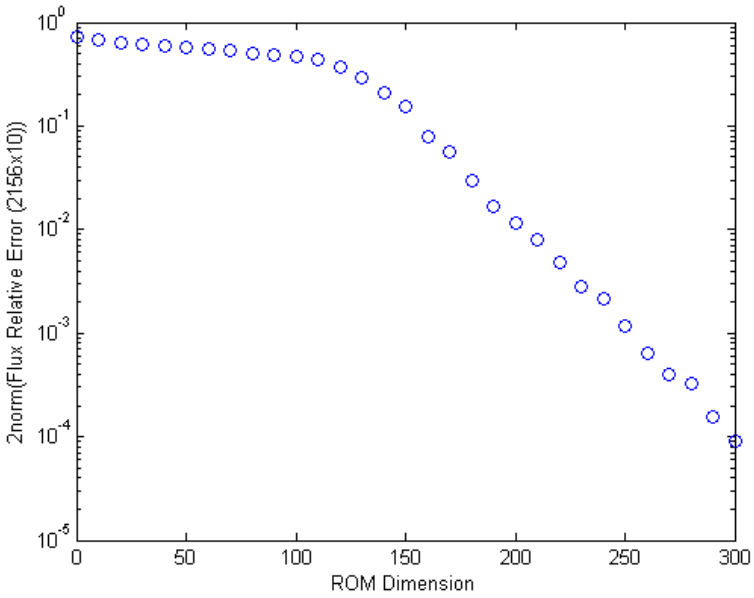

Figure 20: Full-core 2Norm flux κ-metric.

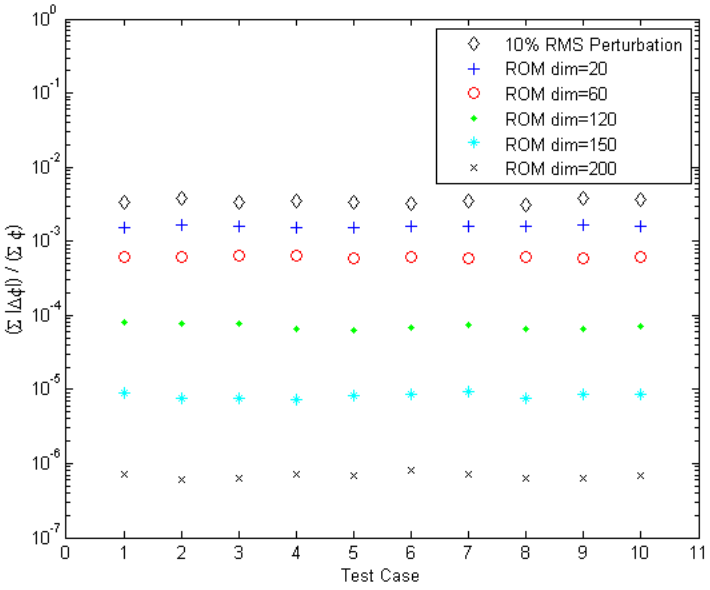

Figure 21: Full-core average flux error κ-metric.

Using the aforementioned GPT-Free approach with a ROM of dimension 200, response sensitivities can be obtained from a model with both an input and output space two orders of magnitude larger. This spatial independence may be a result of the independence of cross-section data assumed for each fuel-pin; however, in an actual assembly, perturbations are not independently placed within each pin and are instead globally introduced, significantly constraining the outputs. If the fuel-pins are nearly independent as in this case, large reductions are available.

5.B Mesh-Refinement Effects

The GPT-Free approach constructs a reduced-order model regardless of a user-specified mesh. The errors in the ROM formed using the GPT-Free approach introduced by means of mesh-refinement were studied. Based on the methodology implemented to form the ROM, errors introduced from a lower order mesh are expected to be minimal. A modified κ-metric of the form in Eq. 92 will be used to study the effects of mesh refinement on the quality of a ROM.

$$\kappa_{\phi M} = \frac{\sum_{i=1}^{m} \left| \Delta \phi_i^{\perp} \right|}{\sum_{i=1}^{m} \left| \Delta \phi_i \right|} \tag{91}$$

A single reflected 2.45 w/o ^{235}U pin-cell from the SCALE sample problems folder was modeled using NEWT and TRITON/TSUNAMI-2D with the standard v5-44 cross-sections library (Dehart 2009, SCALE 2005). Two tests are completed using this model, the first examining the incremental relative change in sensitivities and flux due to mesh-refinement and comparing it to 10% RMS cross-section perturbations of the model, and the second constructing a ROM at a given mesh-grid and evaluating its quality using the κ-metric with different mesh-grids.

For the first test, the isotope, energy, and reaction-dependent cross-sections are randomly perturbed with a 10% RMS via a Gaussian distribution, and the TSUNAMI-2D sequence is executed to evaluate both the exact changes in fuel-region flux and the fuel sensitivity profiles due to these perturbations. Next, the TSUNAMI-2D is executed with increasing mesh-refinement and no cross-section perturbations to evaluate the incremental change in the

flux. The results are compared in Fig. 22 showing the reduction in average and maximum relative error in the flux under mesh-refinement as circles and plusses respectively. The relative change in average and maximum flux due to cross-section perturbations is shown as the solid and dashed lines. The relative flux error is computed as follows:

$$\Delta\overline{\phi_m} = \frac{1}{44}\sum_{g=1}^{44}\frac{\left|\phi_g^{m+2}-\phi_g^m\right|}{\phi_g^m}$$
$$\max\left(\Delta\phi_m\right) = \max_{g=1..44}\left(\frac{\left|\phi_g^{m+2}-\phi_g^m\right|}{\phi_g^m}\right) \tag{91}$$

The m indicates the flux calculated with the m^{th} mesh division. The relative change in flux due to 10% RMS cross-section perturbations is computed similarly, comparing the perturbed flux to the reference flux in Eq. 94. Here $N = 20$, the number of perturbation cases.

$$\Delta\overline{\phi_\sigma} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{44}\sum_{g=1}^{44}\frac{\left|\phi_g^i-\phi_g^{ref}\right|}{\phi_g^{ref}}$$
$$\max\left(\Delta\phi_\sigma\right) = \max_{i=1..N}\left\{\max_{g=1..44}\left(\frac{\left|\phi_g^i-\phi_g^{ref}\right|}{\phi_g^{ref}}\right)\right\} \tag{91}$$

Note that with coarse mesh refinement, the relative change in flux due to 10% RMS cross-section perturbations is of similar order to the discretization errors. As the mesh is refined the discretization error diminishes, and becomes noticeably different from the flux variations due to cross-sections perturbation. The level-off beyond 12 mesh divisions/cell indicates that the best accuracy possible by the employed numerical scheme is reached.

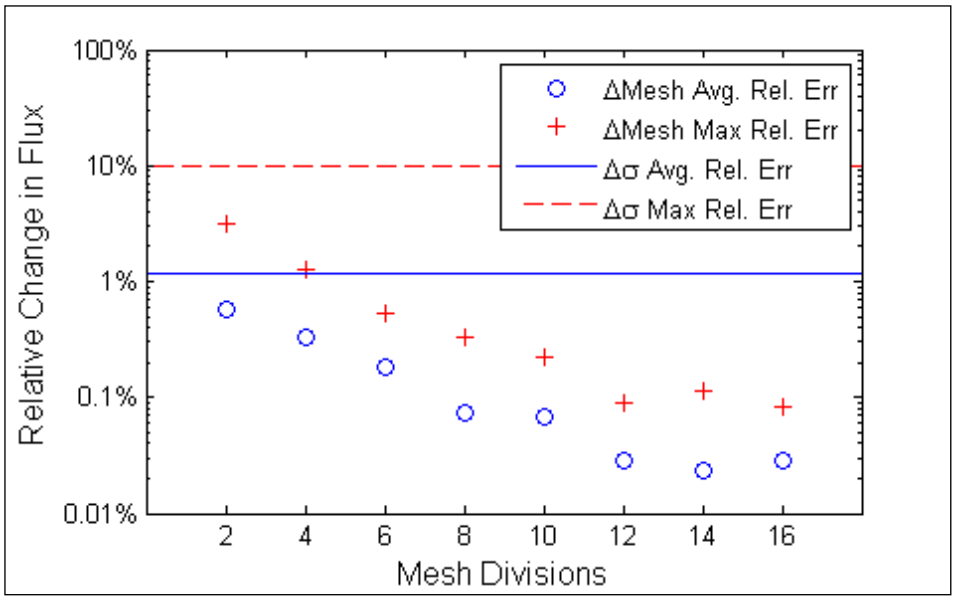

Figure 22: Variations in the flux due compared between cross-section perturbations and mesh-refinement.

The second test uses the pin-cell model with a 2x2 mesh-grid to generate the reference GPT-Free ROM with dimension r = 200. A set of 20 reference cross-section perturbations are generated at each mesh-refinement grid. The quality of the ROM is evaluated using the κ-metric with increasing mesh refinement. The orthogonal contribution from benchmark cross-section perturbations at each mesh-grid is computed as in Eq. 94. The results are plotted in Fig. 23 comparing the changes in the evaluations as a function of mesh-refinement. The red plus indicates the magnitude of the flux normed across energy. The blue circles show the 2-norm of the flux error for the 20 samples calculated as follows:

$$\text{ROM Flux Error} = \left|\left|\phi^{\perp} - \phi^{ref}\right|\right| \tag{91}$$

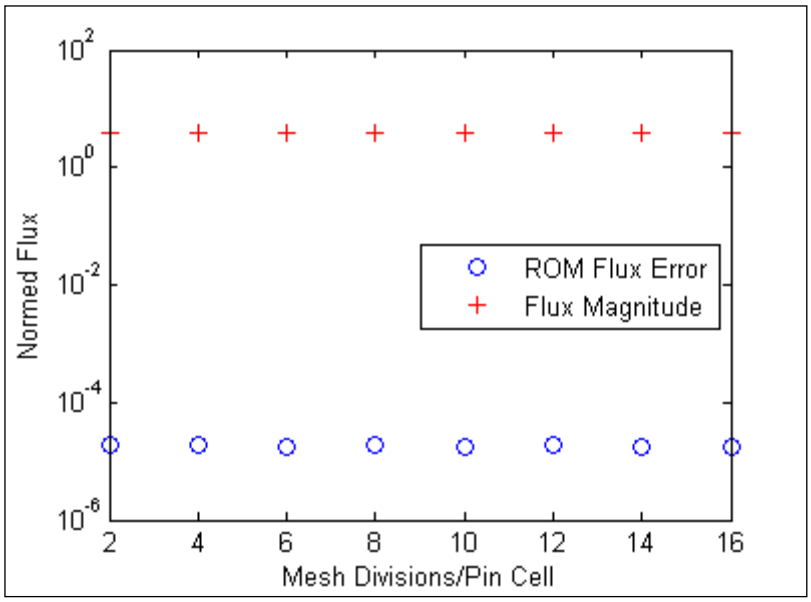

Figure 23: GPT-Free error vs. mesh-refinement

The orthogonal flux is generated via flux perturbations projected orthogonal to the ROM constructed with a 2x2 mesh. Note that the error resulting from the GPT-free subspace is independent of the mesh refinement. This implies one can use a coarse mesh to generate the subspace, and employ a finer mesh to complete the forward sensitivity analysis.

5.C Depletion GPT-Free Sensitivity Analysis

Construction of a ROM can be extended over the life of the reactor system due to depletion. Consider the macroscopic cross-section, a product of the microscopic cross-section, σ, and the number densities of materials, *N*.

$$\Sigma = N\sigma \tag{92}$$

The ROM for a steady-state problem perturbs only cross-section data to find a space from the fundamental sensitivities, $dk/d\sigma$. To address depletion, burn-up must be incorporated in order to sample sensitivities over the time horizon.

Burn-up changes sensitivity information in two manners. First, fissile material is being depleted. While other fissile materials are being produced, the net effect is a decrease in fissile material that in time hardens the spectrum with the introduction of Pu-239 (Cacuci 2010). Second, burn-up introduces additional actinides into the system. These two mechanisms strongly influence sensitivity data over the lifetime of a reactor. A sample of the most important actinides depletion evolution curves from a VHTR infinite lattice cell are shown in Figure 24. Also, the evolutions of sensitivity profile magnitudes are plotted in Figure 25. These two figures demonstrate the time-evolution of a reactor undergoing depletion.

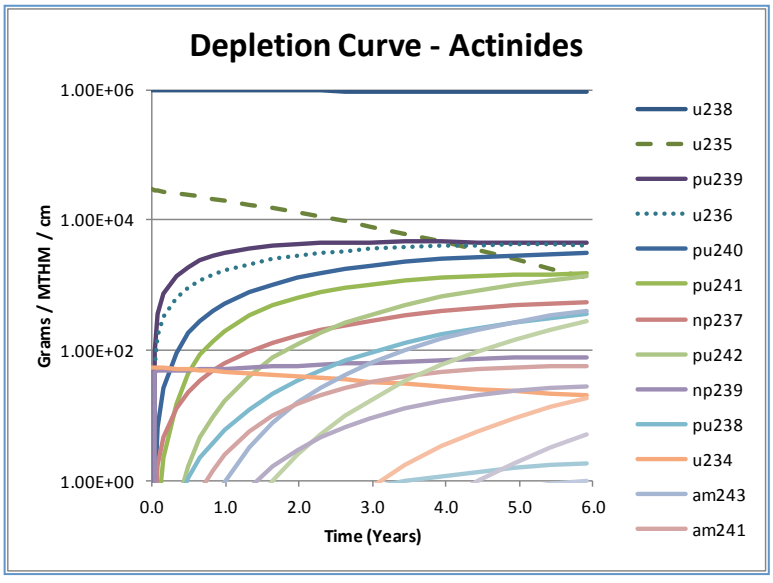

Figure 24: Depletion curves for actinides in a HTGR assembly model

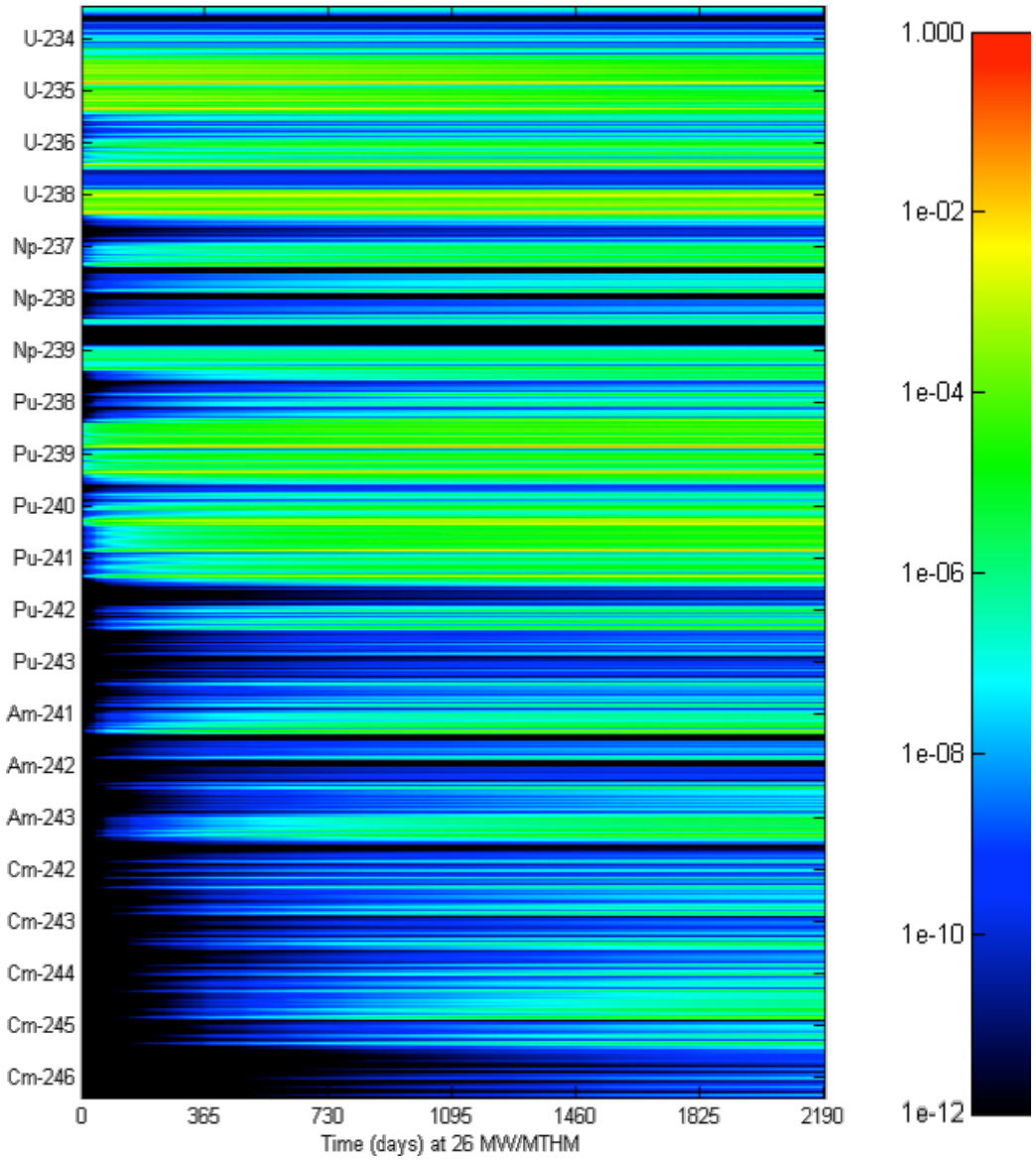

Figure 25: Time and energy dependent k-sensitivity profiles of 21 actinides for fission and radiative capture modeled in TRITON using the HTGR infinite prismatic lattice model.

.

Typical depletion evaluations use a quasi-static approximation in order to efficiently model a reactor system. The quasi-static approach decouples the time variable in neutron

transport and depletion by considering that the reactor is steady-state at any time point and that the flux is relatively insensitive to changes in the nuclide densities over short time-spans. In other words, the flux is separable in time (Williams 1978). Such an approximation separates the depletion calculation into an iterative process of two steps: A steady-state critical eigenvalue transport solution followed by an evaluation in time of ordinary differential depletion equations. Using the quasi-static approximation in Eq. 97, the time-dependent transport equation is approximated by steady-state evaluations at T points in time $\{t_i\}_{i=1}^{T}$.

$$\mathbf{M}\left(N,\sigma,t\right)\phi = 0 \quad \rightarrow \quad \mathbf{M}\left(N_{t_i},\sigma\right)\phi_{t_i} = 0 \tag{92}$$

By decoupling the transport calculation from depletion, which is computed between transport time-steps, the burn-up equations can be simplified as in Eq. 98.

$$\frac{dN\left(t\right)}{dt} = \mathbf{A}\left(\phi,\sigma,t\right)N\left(t\right) \quad \rightarrow \quad \frac{dN\left(t\right)}{dt} \approx \mathbf{A}\left(\phi_{t_i},\sigma\right)N\left(t\right) \tag{92}$$

The removal of a time component from the transmutation matrix **A** greatly simplifies the depletion evaluation by reducing it to a system of ODEs. For a more extensive examination of the burn-up equations and related topics, see ref. (Turinsky 2010).

Typical perturbation theory calculations are evaluated with steady-state neutronics models. Similar to depletion perturbation theory (Williams 1978), additional steps must be taken to apply the GPT-Free methodology to depletion calculations because of model variations in time. First, the number of computations must be reduced in order to effectively sample the sensitivity space. Running the entire depletion sequence requires repeated

evaluation of the model in order to accurately determine both the flux and the nuclide density vector. Second, because the flux is evaluated only at a finite number of points in time, the sensitivities are not necessarily guaranteed to be represented fully at these pre-selected time-points. The nonlinear change in sensitivity profiles in time is best seen in Figure 25, depicting the change in sensitivity profiles in an HTGR lattice (Dehart 2009) in time due to depletion. The depletion GPT-Free calculation is broken down into additional steps to address these concerns.

The first step is to determine the depletion-evolution curves of the nuclide density profiles subject to cross-section perturbations. To form the depletion-evolution curves, cross-sections perturbations are introduced into the model at time zero. Enough burn-up time-steps are completed such that linear interpolation of nuclide densities between time-points minimizes the error below a user-defined threshold, similar to the method in which point-wise cross-section evaluations are formed (Leppänen 2009). Using a series of perturbations, the mean and standard deviation of nuclides can be formed in time. The number of samples required must be statistically sufficient and thus is based on the Wilks' criteria, shown in Eq. 99, to select the number of samples, N, based on the desired probability/confidence interval, p and c (Wilks 1941, Strydom 2011, Kennedy 2011).

$$\left(1-p^{N}\right)-N\left(1-p\right)p^{N-1}\geq c \tag{92}$$

The second step is to include a time-sampling random-variable into the GPT-Free methodology. By sampling times randomly, a nuclide density vector is formed by sampling a nuclide density vector from the depletion-evolution curves. By linearly interpolating between

both the neighboring time-points mean and standard deviation, a model-plausible nuclide density vector can be prepared according to the calculations in the prior step. This scheme allows for the steady-state evaluation of any point in time between t_1 and t_T that follows from the depletion-evolution curves. If too few depletion burn-up points are selected in developing the depletion-evolution curve, the sampling method will not accurately capture possible nuclide configurations in time; however, if sufficient time points are used, the time-sampling scheme does not require repeated computation of prior time-steps.

From evaluations of the UAM assembly model (Ivanov 2007) using TRITON (Dehart 2009) as shown in Figure 26, modeled with 10% RMS Gaussian cross-section perturbations introduced at the beginning of the evaluation over a time horizon of 1080 days at a specific power of 21.220 MWth/MTHM, the largest absolute uncertainty in any isotope is less than 3mg per cubic centimeter for a corner-pin ^{238}U at the final time-point. Gadolinium pins, denoted yellow in Figure 26, were depleted by flux in TRITON as opposed to the deplete-by-power per SCALE manual recommendations (SCALE 2005). A comparison of all nuclides with uncertainty greater than 1μg was plotted vs. their sorted relative standard deviation in Figure 27. For all nuclides with relative uncertainty greater than 4%, their absolute uncertainty lies below 10^{-7} atoms per barn-cm.

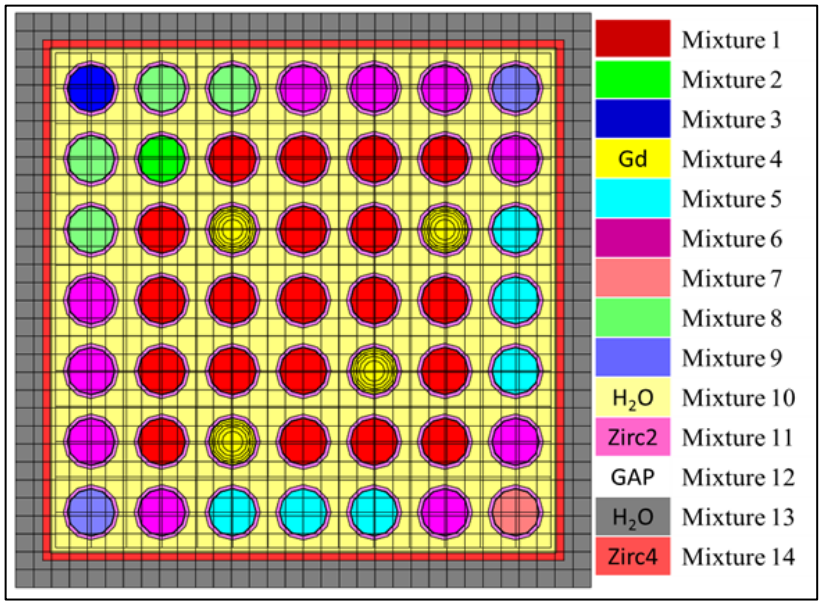

Figure 26: UAM 7x7 assembly model with 9 mixture-depletion zones.

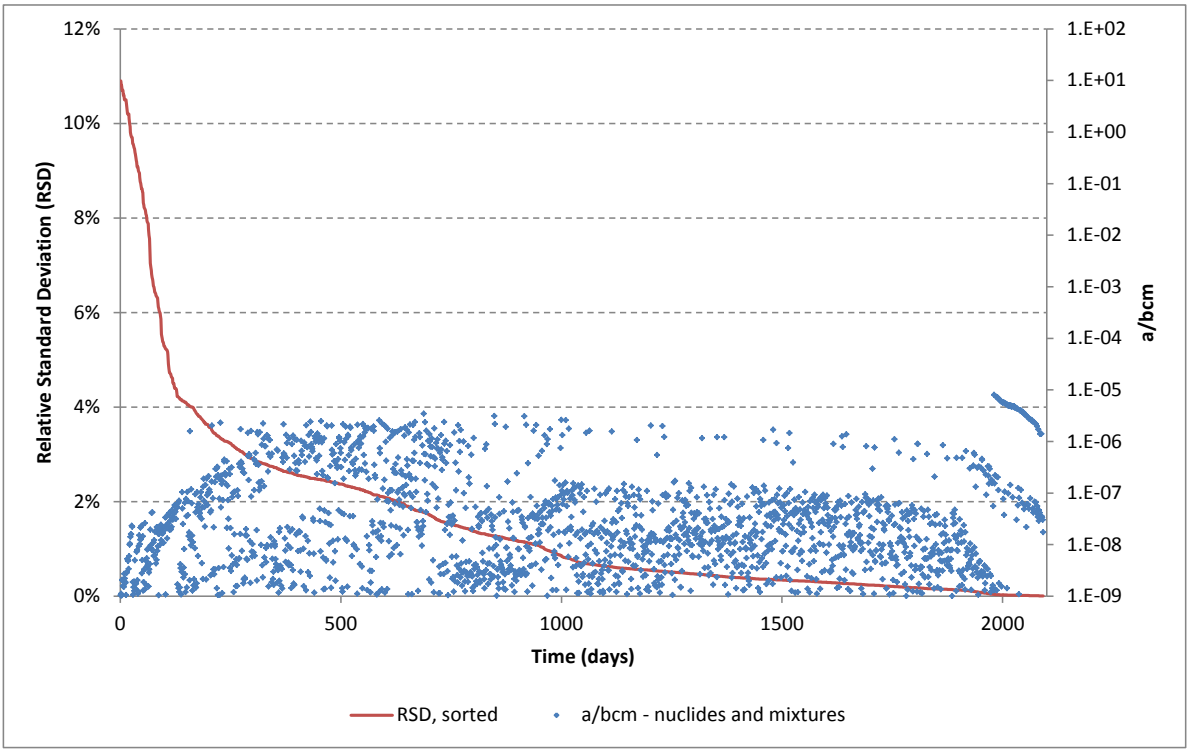

Figure 27: RSD *vs.* absolute uncertainty for all mixture isotopes at all depletion time-points.

The depletion curves of four isotopes of Uranium and Plutonium in the interior pin, denoted red in Figure 26, are plotted *vs.* their relative standard deviation in Figures 28 - 31. The lines designate the linearly interpolated nuclide density value. Additional depletion curves are available but are not reproduced here due to their similarity to the four provided plots. For nuclides that have a large value, e.g. existing prior to depletion, the relative uncertainty tends to increase as time progresses; however, nuclides that are building up due to depletion tend exhibit a single or double peak behavior: the initial peak being the uncertainty in its initial early production (high uncertainty on a very small value), and the second peak building up due to uncertainty increasing as time progresses. For ^{238}U, the dominant isotope in the reactor, the uncertainty is fairly low throughout the cycle, resulting in the erratically shaped uncertainty plot.

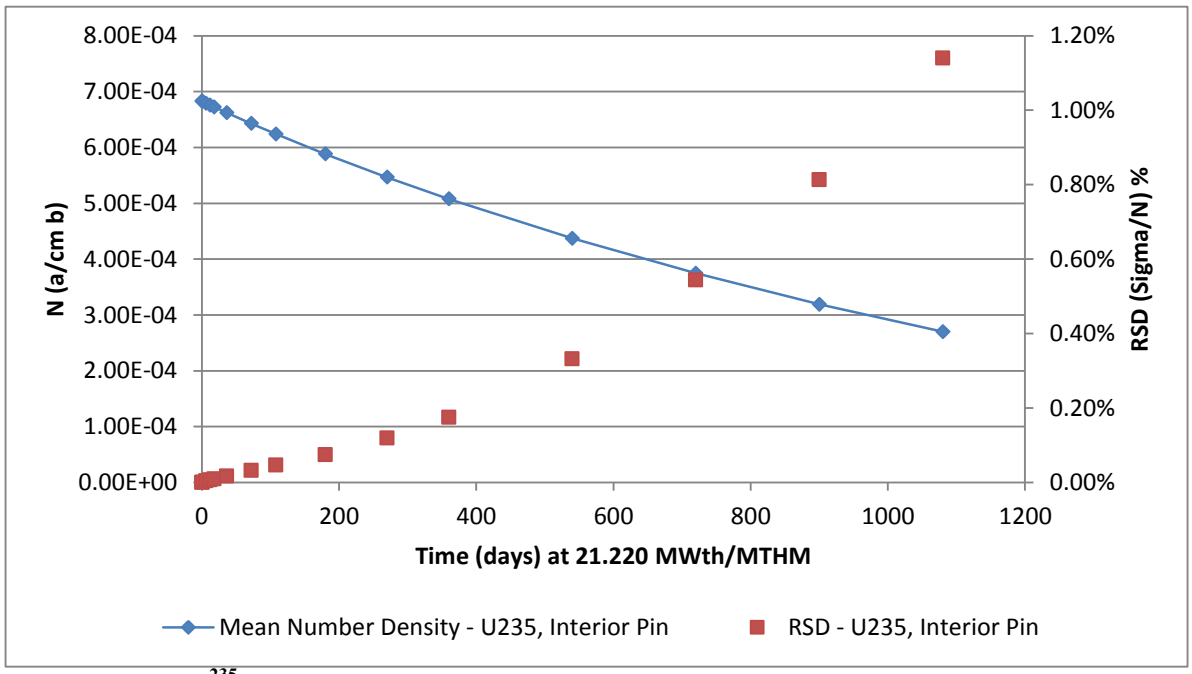

Figure 28: ^{235}U number density and relative uncertainty *vs.* operating time at 21.220 MWth/MTHM.

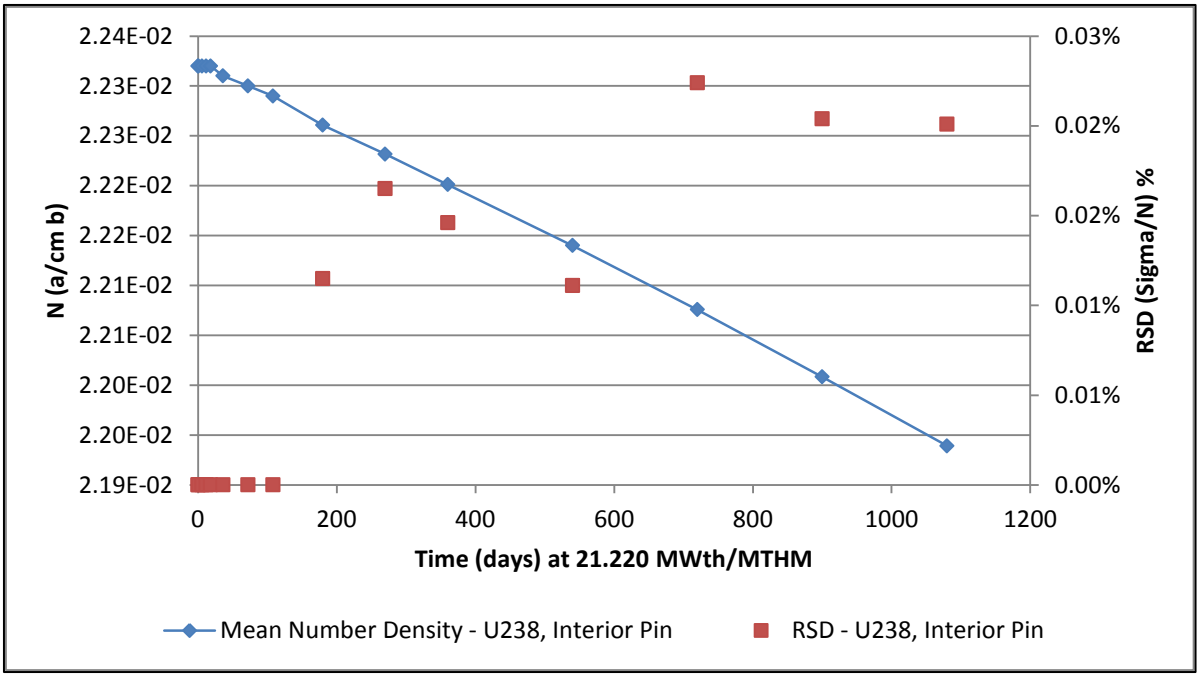

Figure 29: ^{238}U number density and relative uncertainty *vs.* operating time at 21.220 MWth/MTHM.

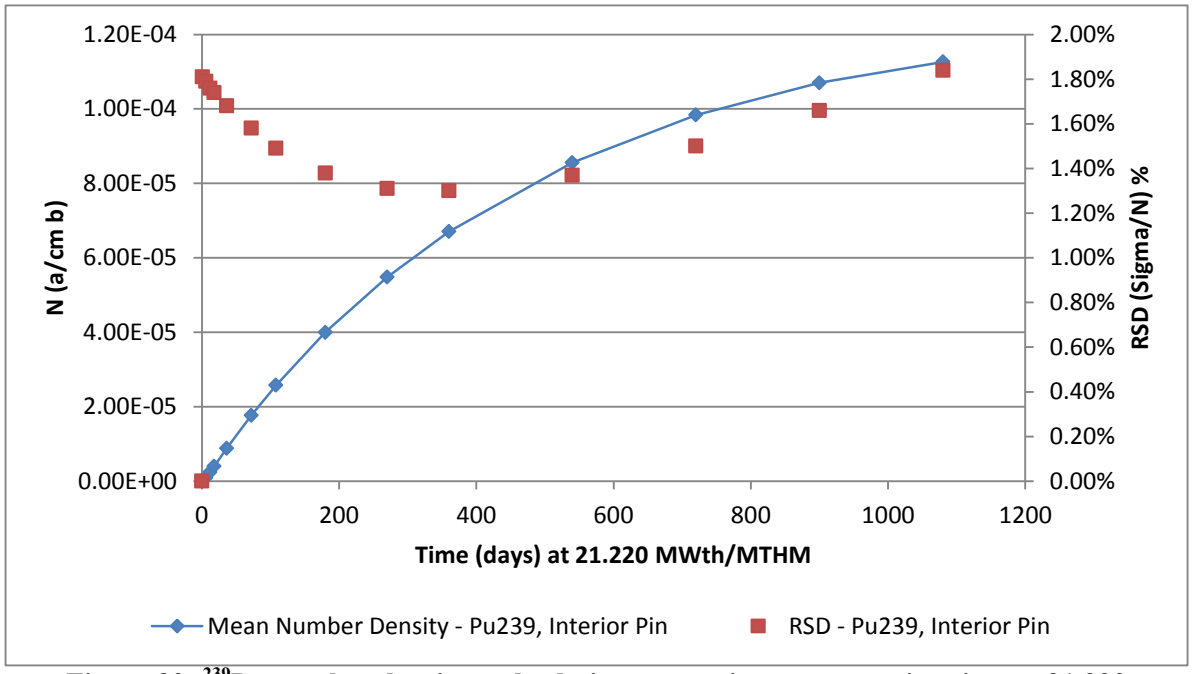

Figure 30: ^{239}Pu number density and relative uncertainty *vs.* operating time at 21.220 MWth/MTHM.

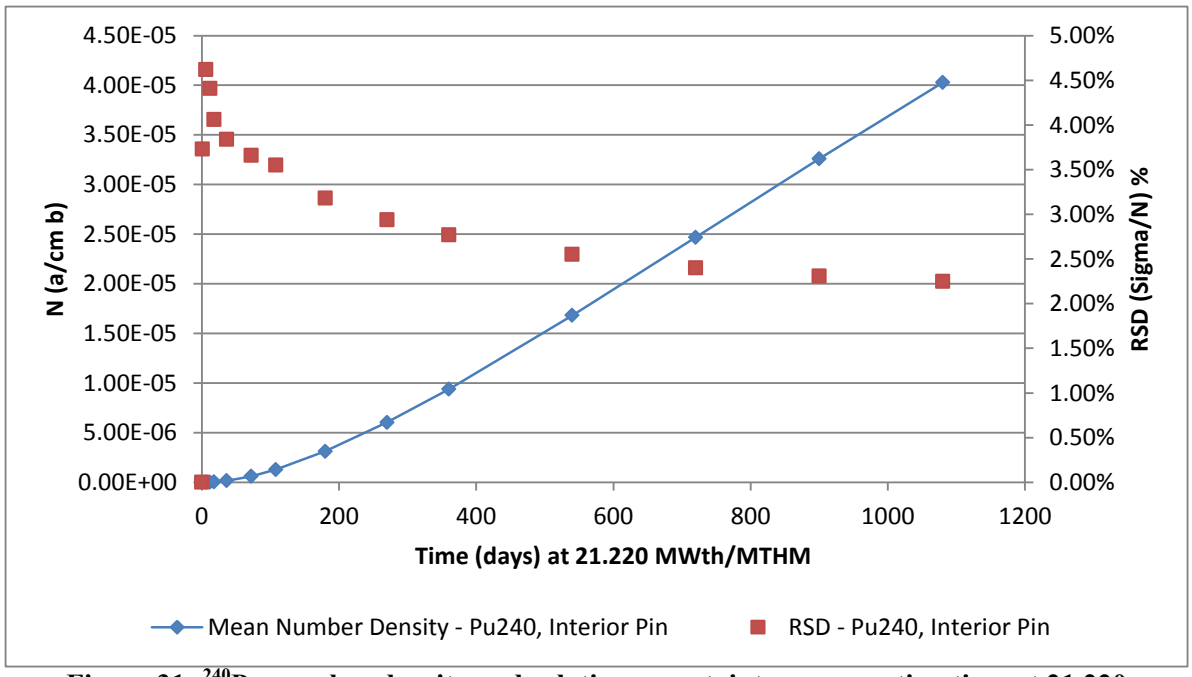

Figure 31: ^{240}Pu number density and relative uncertainty *vs.* operating time at 21.220 MWth/MTHM.

Prior analysis of the HTGR lattice compared the singular value spectrum for the flux with and without depletion in conjunction with cross-section perturbations. The model was varied within the time-scale of the depletion-evolution curve with 2% RMS nuclide density variations and with cross-sections perturbed by sampling from a Gaussian distribution with 10% RMS variation. The resulting flux vectors collected from the samples was arranged into a matrix and analyzed using the SVD (Meyer 1999). The resulting singular value spectrum of the flux matrix is normalized with respect to the largest singular value, σ_1 and is plotted in Fig. 32. A sample of 200 perturbations at the reference conditions, without depletion, are compared to sample sizes of 200 and 1000 perturbations including depletion with nuclide density perturbations. Variations primarily exist within a few percent of the reference case, with the largest discrepancies in the first 20 singular values. Larger discrepancies exhibited at

the higher end of the singular indices are within the realm of errors in the calculation due to a set model convergence tolerance. Based upon both the similar shape and magnitude of the singular value plots, it is expected that depletion does not significantly add information to the output space, i.e. model order reduction via subspace methods should extend naturally to depletion problems.

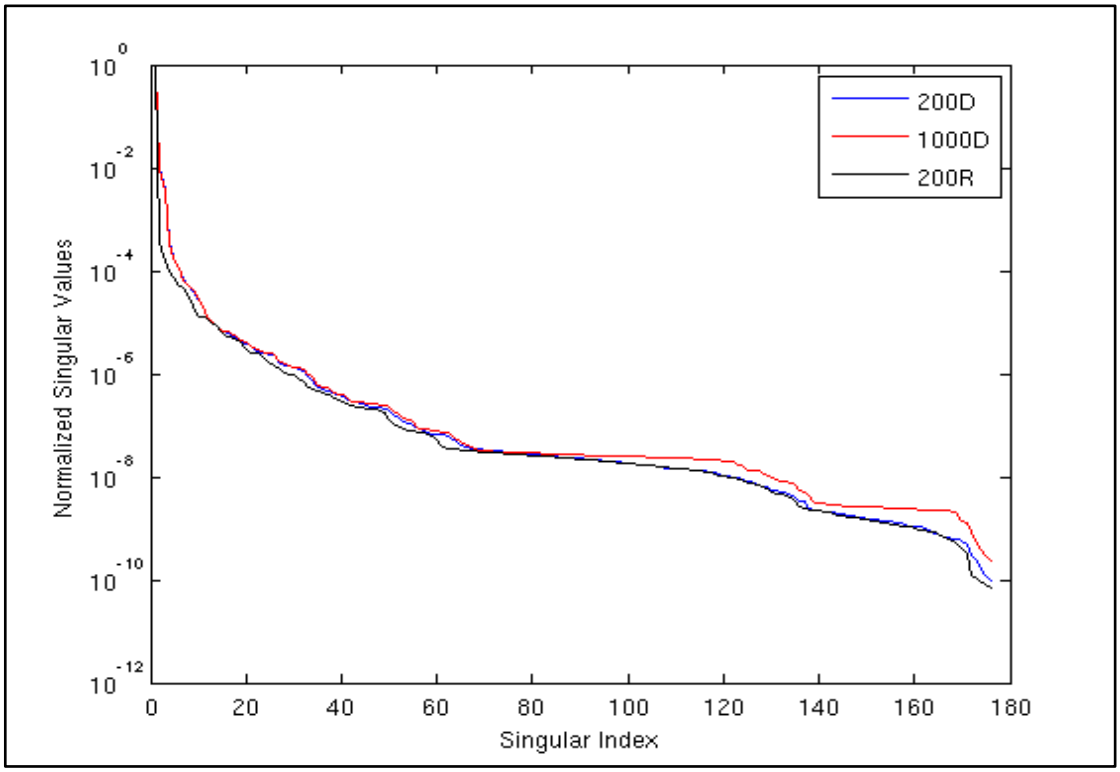

Figure 32: Flux singular values for steady-state and depletion due to cross-section perturbations.

Analysis of a ^{235}U enriched pin-cell under depletion provides a closer look at the effect of depletion on the flux and the ROM formed using the GPT-Free approach. In Fig. 33, the spread of the relative change in the flux using the SCALE 44 energy group structure is shown for twenty samples with 10% RMS cross-section perturbations drawn from a Gaussian

distribution in addition to random time-point sampling on the depletion-evolution curve. Variations increase in the thermal region due to the pin-cell being water moderated. Additionally energy group 20 shows a peak variance consistent with resonances due to the presence of ^{238}U.

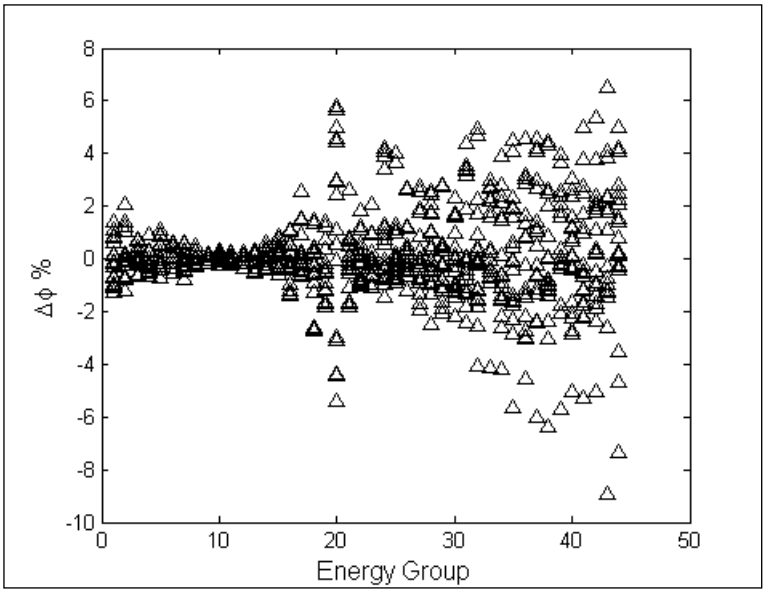

Figure 33: Relative change in fuel-pin flux from 20 random times and cross-section perturbations

The GPT-Free approach has been applied to the pin-cell model both with and without depletion effects in order to compare the dimensionality of the reduced order model. Fig. 34 compares the results of various dimensional ROMs both with and without depletion contributions by means of the average flux error κ-metric in Eq. 100.

$$\kappa_{\phi} = \frac{\sum_{i=1}^{m}\sum_{g=1}^{44}\left|\Delta\phi_{i,g}^{\perp}\right|}{\sum_{i=1}^{m}\sum_{g=1}^{44}\left|\Delta\phi_{i,g}\right|} \tag{92}$$

The m in Eq. 100 designates the number of samples tested under the Wilks' criteria. The orthogonal term designates the ROM projected change in flux. While the effects of depletion do increase the dimension of the ROM, the change is less than a factor of 3. Compared with a traditional analysis of a depletion model where an evaluation for each response (adjoint-mode) or input (forward-mode) requires evaluation at T time-points, the GPT-Free approach is expected to yield favorable results.

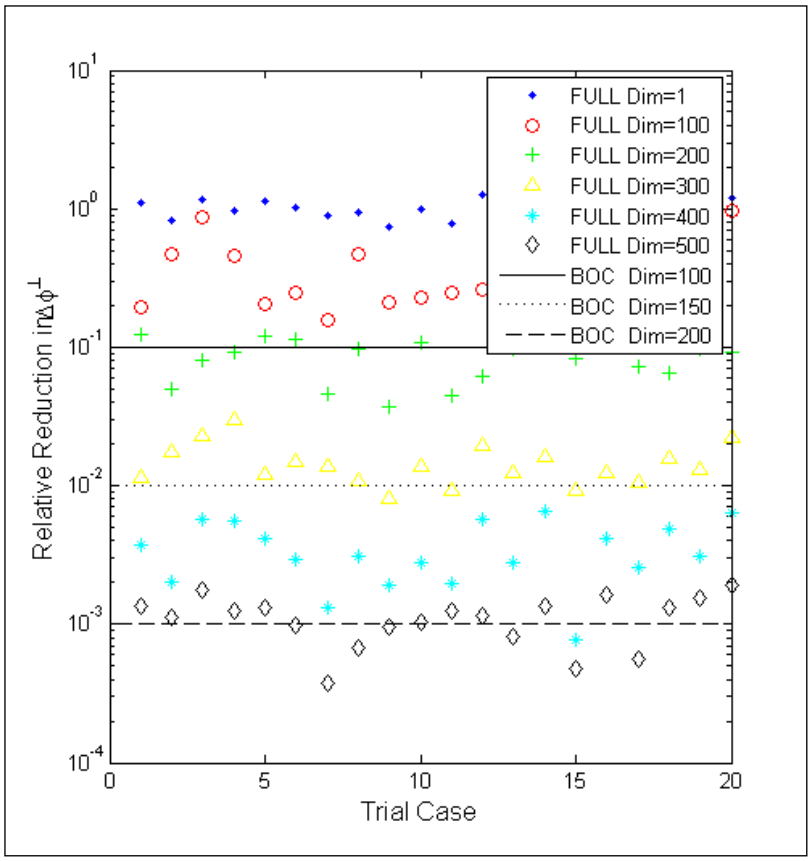

Figure 34: Average flux error κ-metric comparing the BOC ROM to the depletion (FULL) ROM.

The following algorithm has been designed to for a depletion GPT-Free approach:

1. Compute depletion-evolution curves between user-defined times $\left[t_0, t_T\right]$.
2. Pre-compute a sample of *N* benchmark perturbations:
    a. Sample a random time within $\left[t_0, t_T\right]$ and similarly sample the nuclide density vector.
    b. Perturb cross-sections using a Gaussian distribution
    c. Set aside the output and input parameter perturbations for later analysis

3. Determine the GPT-Free ROM with initial guess of dimension Δp with p = 0 for the first iteration
    a. Sample Δp random times within $[t_0, t_T]$ and similarly sample the nuclide density vector.
    b. Perturb cross-sections using a Gaussian distribution
    c. Run the forward and fundamental adjoint model to obtain *k*-sensitivity profiles
    d. Collect $p + \Delta p$ sensitivity profiles into a matrix and form an orthogonal decomposition
    e. Using the input parameter space as determined by step 3.d, project this space onto the *N* cross-section perturbations from step 2.
    f. Run the model *N* times, evaluating the κ-metric to determine the ROM quality.
        i. If the ROM passes, $r = p + \Delta p$ is the ROM dimension, move to step 4.
        ii. Otherwise, $p = p + \Delta p$, return to step 3 and compute more perturbations.
4. Using the ROM with dimension *r*, evaluate $r + \delta$ forward depletion evaluations
    a. Perturb cross-sections at time zero using a Gaussian distribution
    b. Run the forward-depletion model
    c. Compute response sensitivities using a least-squares approach
        i. The δ additional perturbations is a small number that helps the least-squares approach evaluate sensitivities appropriately

4*. Alternative ROM formation strategy:

a. Perturb cross-sections at time zero along the directions of the ROM matrix

b. Run the forward-depletion model

c. Compute response sensitivities using a finite-difference approach

5.D Monte Carlo Extensions

Recent work by Kiedrowski has enabled the possibility of adjoint-weighted tallies with the interest of ultimately providing sensitivity calculations for the fundamental eigenvalue (Kiedrowski 2010). The evaluation from Kiedrowski enables adjoint-weighted tallies of the form:

$$k\Delta\rho_k = -\frac{\left\langle \psi^*, \mathbf{P}\psi \right\rangle}{\left\langle \psi^*, \mathbf{F}\psi \right\rangle} \tag{93}$$

Here, ρ_k is the reactivity, and both **P** and **F** are the loss and production operators respectively; however, the operator terms must be specially described in Monte Carlo:

$$\mathbf{P} = \Delta\Sigma_t - \Delta\mathbf{S} - \frac{1}{k}\Delta\mathbf{F} \tag{94}$$

In Eq. 102, **S** is the perturbed scattering term which for simplifications is assumed to be zero, an approximation.

Consider a relative sensitivity coefficient of the *k*-eigenvalue to cross-sections:

$$\mathbf{s}_k = \frac{\sigma}{k}\frac{dk}{d\sigma} \approx \frac{\sigma}{\Delta\sigma}\frac{\Delta k}{k} \tag{95}$$

Using reactivity relationships from Kiedrowski, the following relationships are possible for sensitivity coefficients formulated from Eq. 101 and 103:

$$\mathbf{s}_k \approx \frac{\sigma}{\Delta\sigma}\left(\frac{k\Delta\rho}{1-k\Delta\rho}\right) \tag{96}$$

This formulation in Eq. 104 at first glance suggests that the sensitivity space would be full using continuous perturbations of cross-section data as the term in parenthesis is a scalar weighting factor; however, as certain vectors will significantly change the weighting factor term it is expected that some terms will be nearly zero, enabling reduction.

Consider a Gaussian normal perturbation ξ_i and a weight factor for the parenthesis term, α_i. For a single cross-section perturbation and a vector of perturbations, the following relation is revealed:

$$\begin{aligned} \mathbf{s}_{k,ix} &\approx \alpha_i \frac{\sigma_x}{\Delta\sigma_{i,x}} \equiv \alpha_i \frac{\sigma_x}{\xi_i \sigma_x} = \frac{\alpha_i}{\xi_i} \\ \mathbf{s}_{k,i} &\approx \alpha_i \frac{1}{\mathbf{v}_i} \qquad \mathbf{v} = \left[\xi_1 \ldots \xi_n\right]^T \end{aligned} \tag{97}$$

Because of the nature of Gaussian perturbations, the model is expected to be full-rank; however, the weighting factor depends on the physics of the system.

Gaussian perturbations lead to poorly conditioned matrix without consideration of the impact of the weighting factors. A comparison of Gaussian perturbations and their inverse is available in Figure 35. This difficulty can be overcome by an alternative sampling scheme. Consider both the transport equation and the *k*-eigenvalue sensitivity equation for a deterministic model.

$$\mathbf{L}\phi = \lambda \mathbf{F}\phi \tag{98}$$

$$S_{k,\Sigma_i} \frac{\partial k/k}{\partial \Sigma_i/\Sigma_i} = -\frac{\Sigma_i}{k} \frac{\left\langle \phi^*, \left(\frac{\partial \mathbf{L}}{\partial \Sigma_i} - \frac{1}{k} \frac{\partial \mathbf{F}}{\partial \Sigma_i} \right) \phi \right\rangle}{\left\langle \phi^*, \frac{1}{k^2} \mathbf{F} \phi \right\rangle} \tag{99}$$

In order to sample various model-evaluations effectively, perturbations in the input space must be well-conditioned. However, sensitivity coefficients, which develop the GPT-Free ROM, must also be sufficiently varied efficiently for the methodology to be efficient. For any variation, the vector of sensitivities will be highly dependent on ϕ, ϕ^*, and Σ. Because the microscopic cross-sections vary only mildly and both the forward and adjoint flux vary only in shape, the variations will be minor. The step to address this is to introduce number density perturbations.

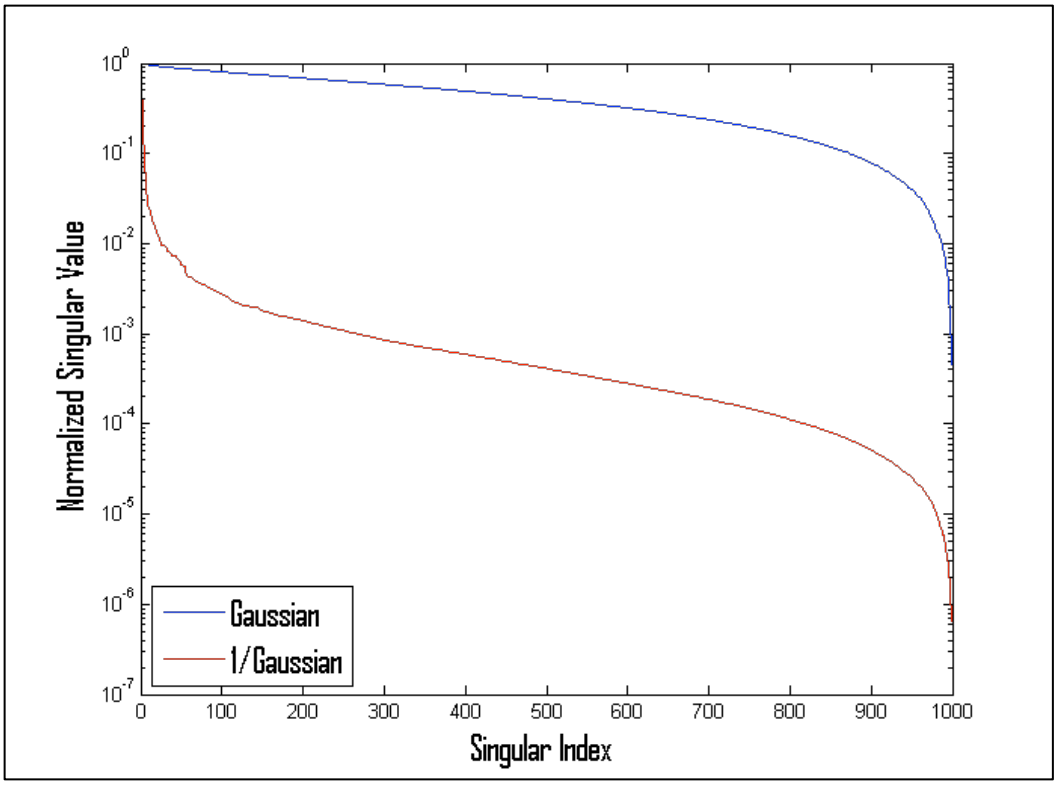

Figure 35: Comparison of the singular value spectrum of Gaussian and 1/Gaussian perturbations.

A study was completed on a ^{235}U enriched pin-cell containing traces of ^{234}U and ^{236}U to determine the effect of cross-section and nuclide perturbations on a model to investigate the hypothesis of nuclide perturbations. A series of tests perturbing both number density and cross-sections by sampling from a Gaussian distribution, outlined in Table 1, were evaluated with TSUNAMI-2D to generate sensitivity profiles. The singular value plots from the sensitivity curves using 100 samples are plotted in Figure 36. As can be seen from the graph, cross-section perturbations remove the steep fall-off prevalent from nuclide density perturbations. This is a result of the degrees of freedom from independently perturbing energy and reaction specific cross-sections. Increasing the number density perturbation adds

more information to the first singular values as seen by the shift in the knee of the curve. The first singular values represent the most important directions.

Table 1: Perturbation test cases for the ^{235}U Pin-Cell

Case	$\Delta\sigma$ RMS%	ΔN RMS%
1	0	10%
2	0	50%
3	10%	0%
4	10%	10%
5	10%	50%

Due to ^{234}U and ^{236}U existing in small quantities in a ^{235}U fuel-pin, large number density variations are required in order to add importance to their respective components of the sensitivity profile. A detailed study of the ^{236}U fission sensitivities shows that number density perturbations drive up the magnitude of sensitivities with little variation in shape; however, cross-section perturbations can modify the shape of the sensitivity profile but have a lesser impact on the overall magnitude. This can be demonstrated in Figs. 37 - 40 for cases 1, 3, 4, and 5 from Table 1. These figures plot the energy-dependent fission *k*-sensitivity profile for ^{236}U for 100 samples of the respective test cases. The spread in the data measures the relative effect the perturbation case had on the spread of the *k*-sensitivity profile. For equivalent perturbation magnitudes, number densities variations have a stronger effect than cross-sections on the magnitude of *k*-sensitivity profiles.

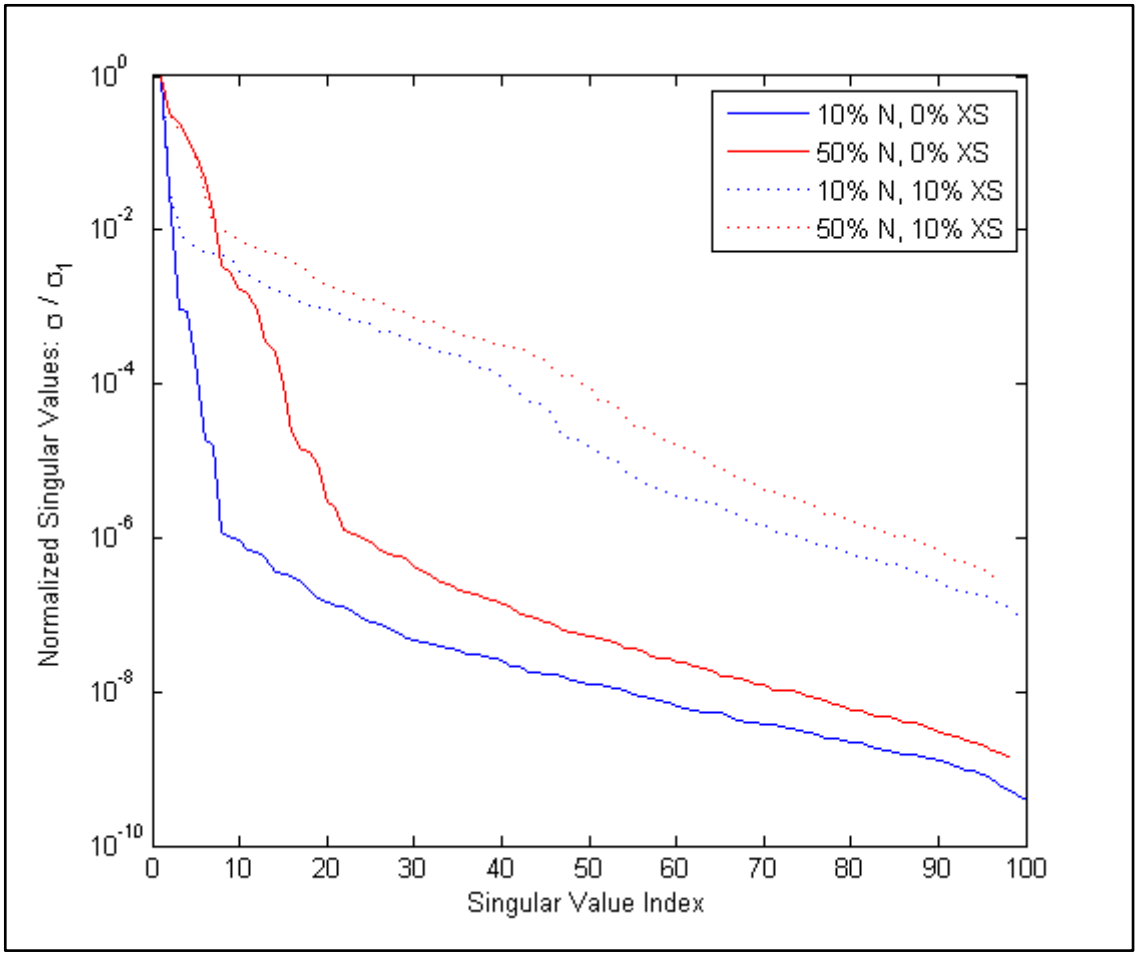

Figure 36: Singular values of sensitivity profiles for nuclide density (N) and cross-section (XS) perturbations of a ^{235}U enriched fuel pin modeled in TSUNAMI-2D.

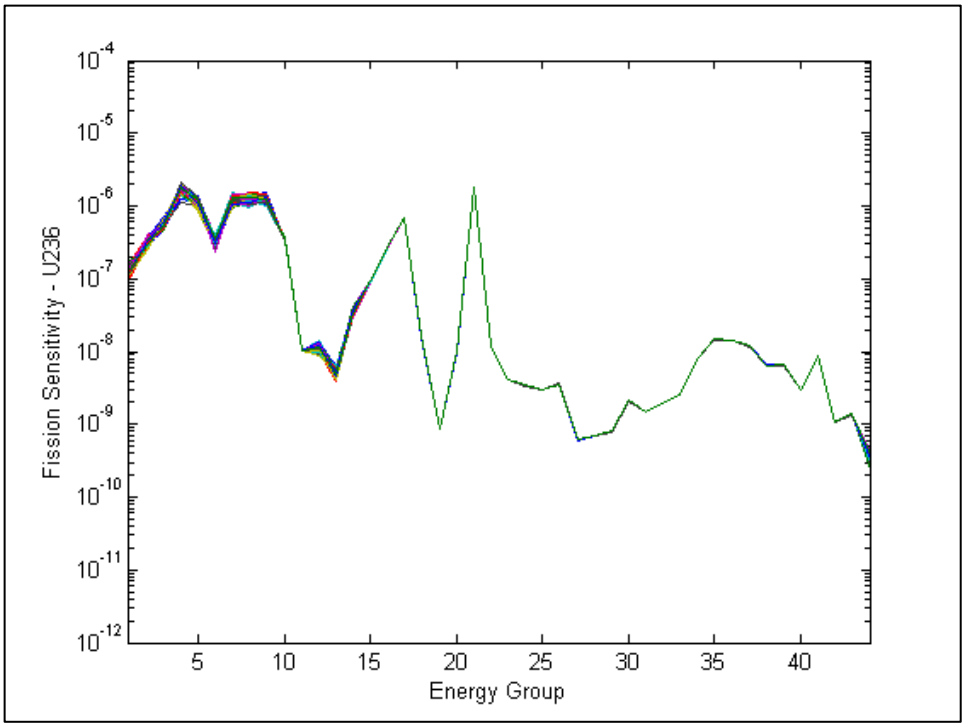

Figure 37: Case 3, ^{236}U *k*-sensitivity profile, 10% RMS cross-section perturbations

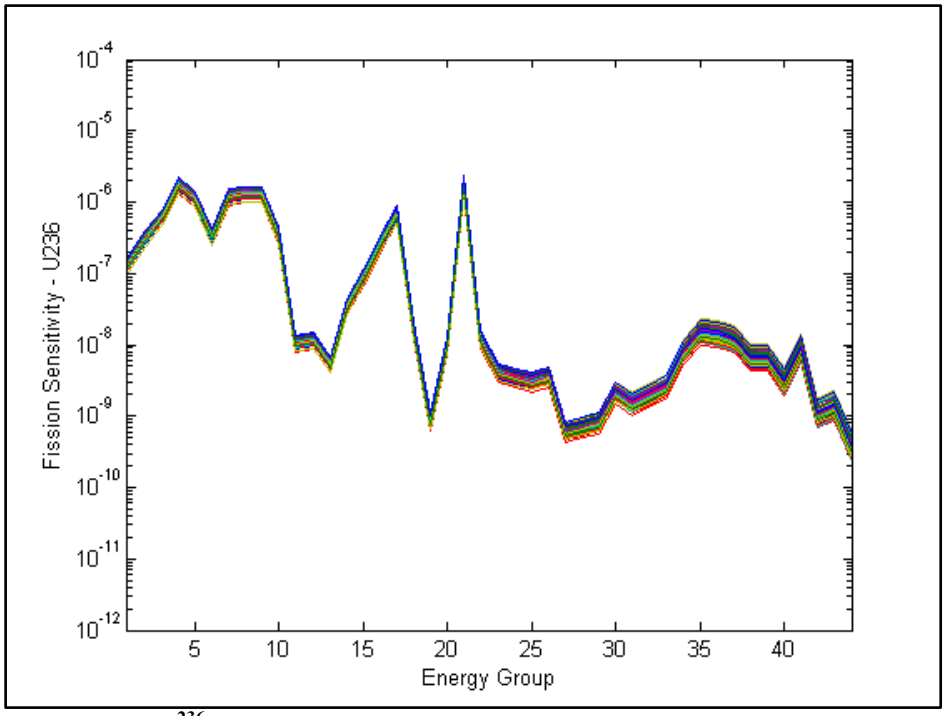

Figure 38: Case 1, ^{236}U *k*-sensitivity profile, 10% RMS number density perturbations

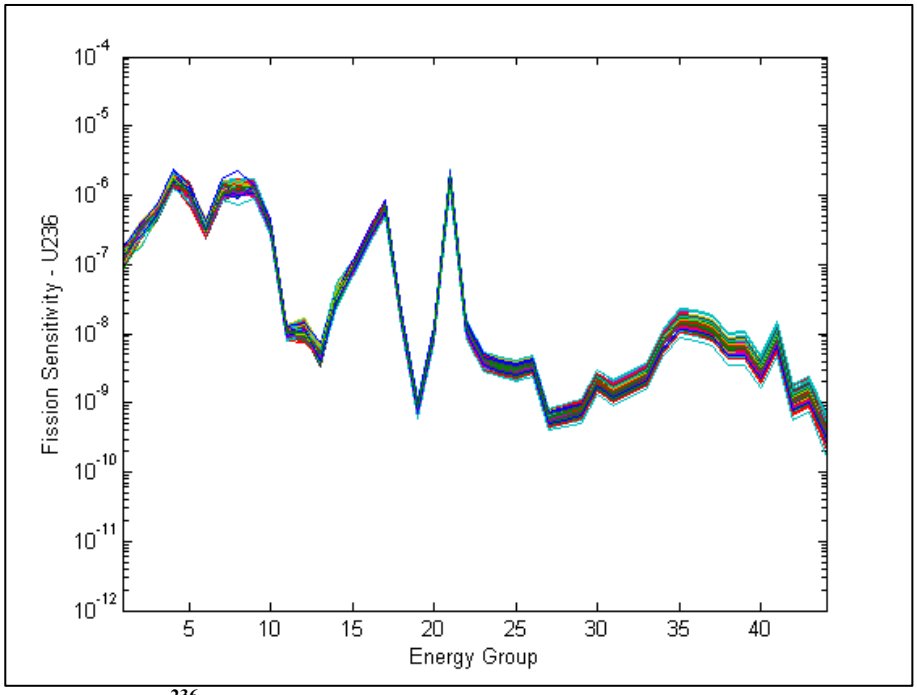

Figure 39: Case 4, ^{236}U *k*-sensitivity profile, 10% RMS number density and 10% XS perturbations

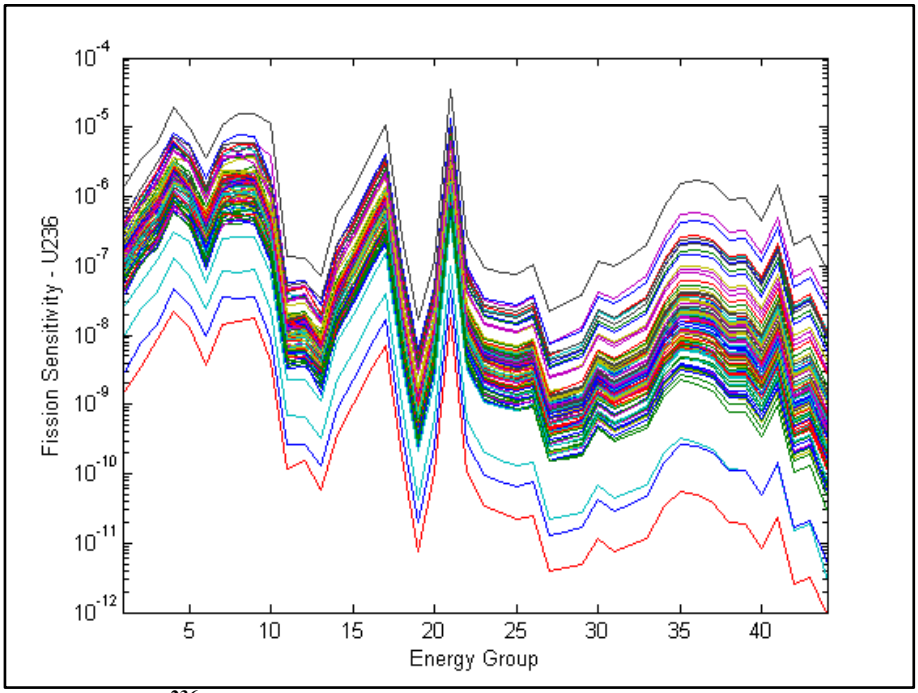

Figure 40: Case 4, ^{236}U *k*-sensitivity profile, 50% RMS number density and 10% XS perturbations

When working with Monte Carlo models, it is expected that both number density and cross-section variations will be required for effectively sampling a model. While there are many ways to sample input parameters, one recent method used effectively for continuous cross-section perturbations is the rQuad polynomial perturbation method for continuous cross-section data (Kennedy 2011). The rQuad polynomial method maps random quadratic functions to the cross-section data to preserve the physical shape of the cross-sections. For a sample of the perturbation, see Figure 41 where the relative perturbation is plotted as blue squares. The rQuad method was compared to Gaussian perturbations for the purpose of Monte Carlo sensitivity sampling. The singular values for 1000 samples using each methodology are plotted in Figure 42, with the rQuad method having a slightly improved singular value spectrum (Shown in blue).

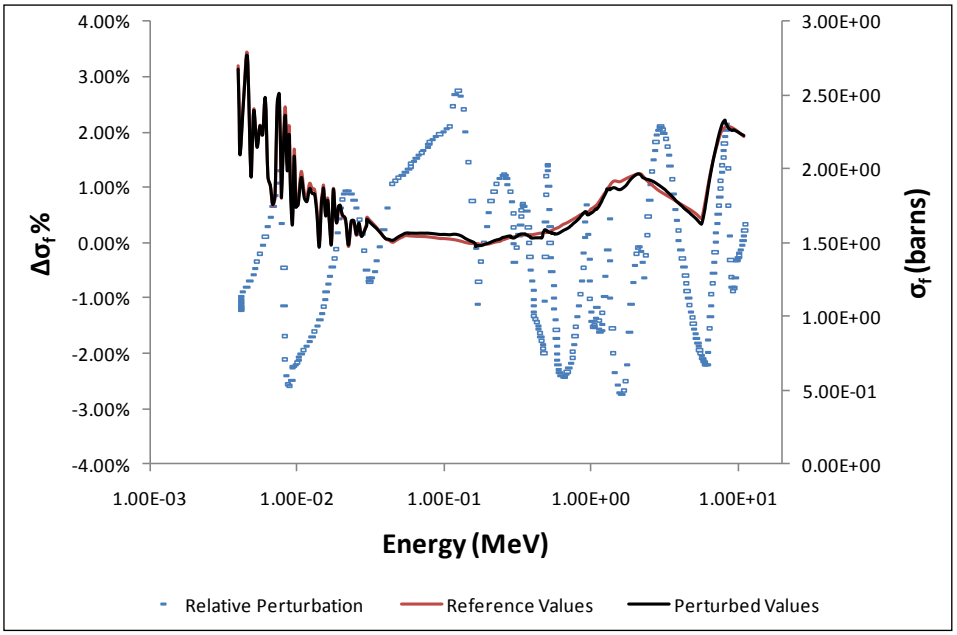

Figure 41: Sample rQuad perturbation of the ^{239}Pu ENDF VI cross-section from MCNP.

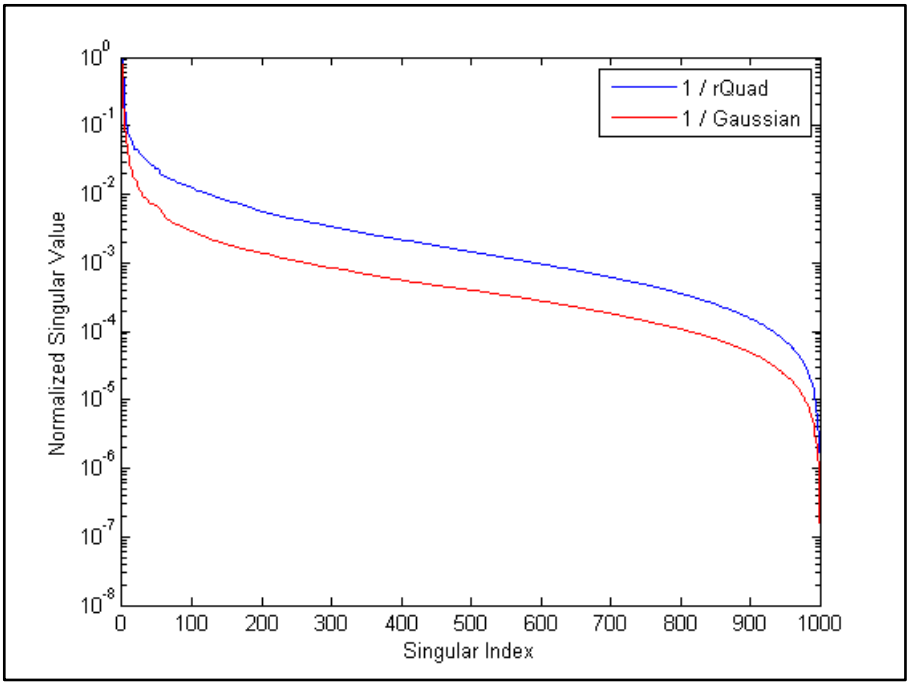

Figure 42: The singular value spread of rQuad and Gaussian inverse perturbations

While the rQuad method was shown to be a practical tool for sensitivity analysis with Monte Carlo models (Kennedy 2011), the method only slightly improves on the Gaussian perturbations in the conditioning of the Monte Carlo sensitivities. The large drop-off in the sensitivity spectrum is due to the domination of near-zero perturbations that do not strongly affect the model. As a result, the Gaussian distribution was modified to the following form:

$$p(x) = \begin{cases} \dfrac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} & |x| > \sigma_0 \\ 0 & |x| \le \sigma_0 \end{cases} \tag{100}$$

A test of 10,000 samples of the distribution using a standard deviation of 10% $(\sigma = 0.10)$ and a cutoff of 1% $(\sigma_0 = 0.01)$ is demonstrated in Fig. 43.

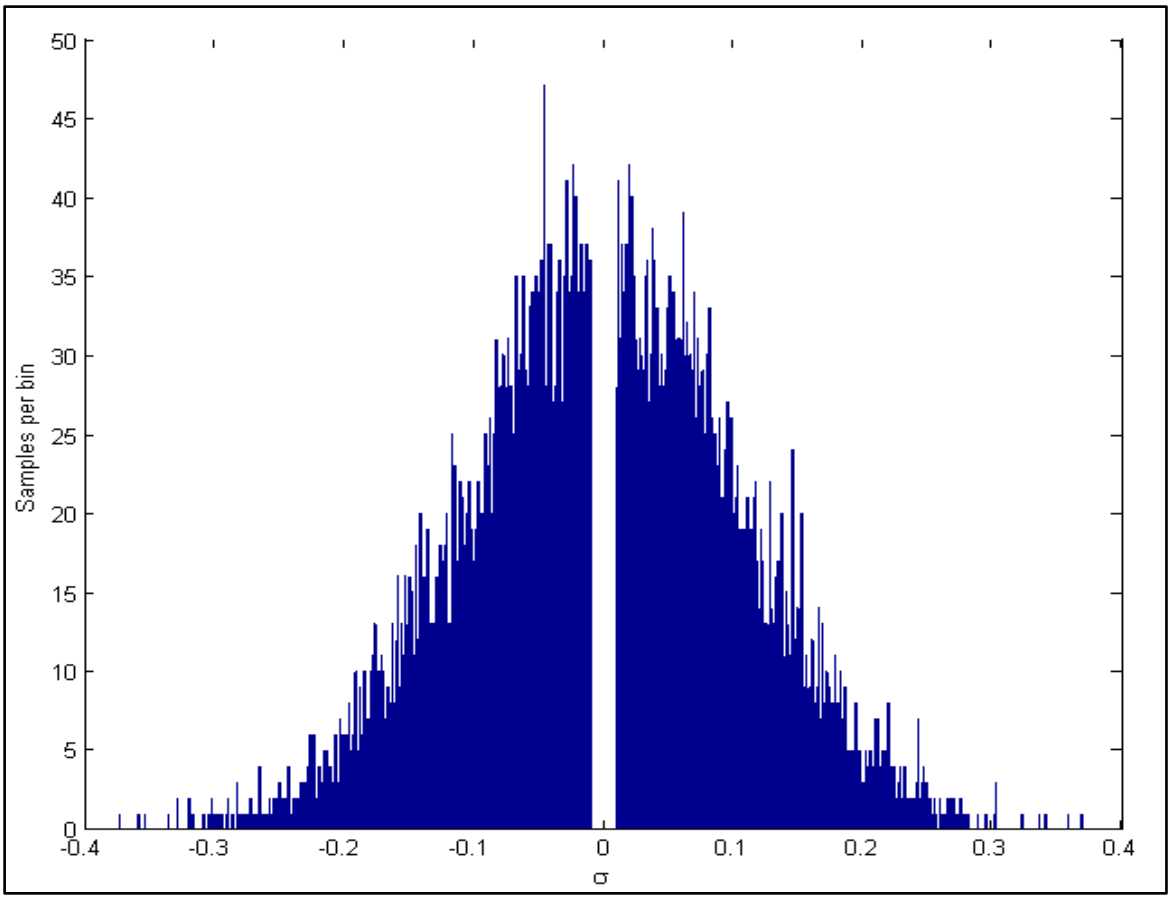

Figure 43: Numerical test of modified Gaussian distribution using 10k samples with 100 bins.

The inverse of the modified Gaussian distribution is compared to the regular Gaussian in Fig. 42. The result demonstrates the effectiveness of having a minimum threshold of perturbations in the conditioning of the sensitivity vector coefficients.

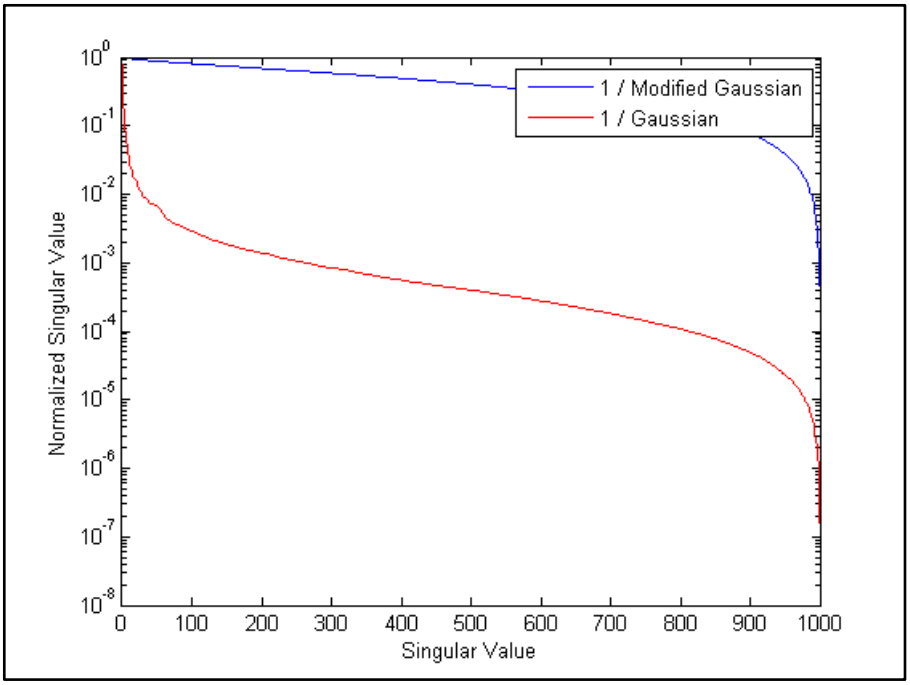

Figure 44: Singular values of the modified Gaussian inverse compared with the standard Gaussian

The next phase of research is to combine the modified Gaussian distribution with the rQuad methodology to gain the physical benefits of the rQuad methodology with the conditioning of the modified Gaussian sampling scheme. Combining this approach to cross-section perturbations with the GPT-Free methodology and Kiedrowski's methodology for solving for the fundamental *k*-sensitivity eigenvalue should allow for implementation of the GPT-Free approach to Monte Carlo models.

6. NUMERICAL RESULTS

6.A Thermal HTGR Infinite Prismatic Lattice Model

An HTGR infinite prismatic lattice was modeled using the deterministic package NEWT (Dehart 2009, Dehart 2011). A 2D NEWT model of this lattice is shown in Fig. 45; the shaded circles represent coolant channels and the smaller circles consist of the fuel compact. The standard 44-group cross-section library from SCALE is used with flux spatially homogenized over the fuel, coolant, and moderator regions. The input parameter space has dimension n = 1584 representing fission and capture cross-sections for 18 actinides. The original model from Dehart uses the material data in Table 2. To ensure all actinides are present, the model is depleted using TRITON to 40GWd/MTHM; Table 3 shows a list of these actinides. A numerical convergence criterion of 10^{-6} is used for both the flux and the k-eigenvalue in NEWT.

Table 2: Parameters for the HTGR infinite prismatic lattice

Region	Material	a cm^{-1} b^{-1}
Fuel	U-238	2.12877E-02
Fuel	U-235	1.92585E-03
Fuel	O	4.64272E-02
Fuel	B-10	1.14694E-07
Fuel	B-11	4.64570E-07
Matrix	C (Natural)	8.77414E-02
Matrix	B-10	9.64977E-09
Matrix	B-11	3.90864E-08
Coolant	He-3	3.71220E-11
Coolant	He-4	2.65156E-05

Table 3: List of actinides modeled in fuel region (40 GWd/MTHM)

Am-241	Cm-243	Cm-246
Am-243	Cm-244	U-234
Cm-242	Cm-245	U-235
U-236	Pu-238	Pu-241
U-238	Pu-239	Pu-242
Np-237	Pu-240	Pu-243

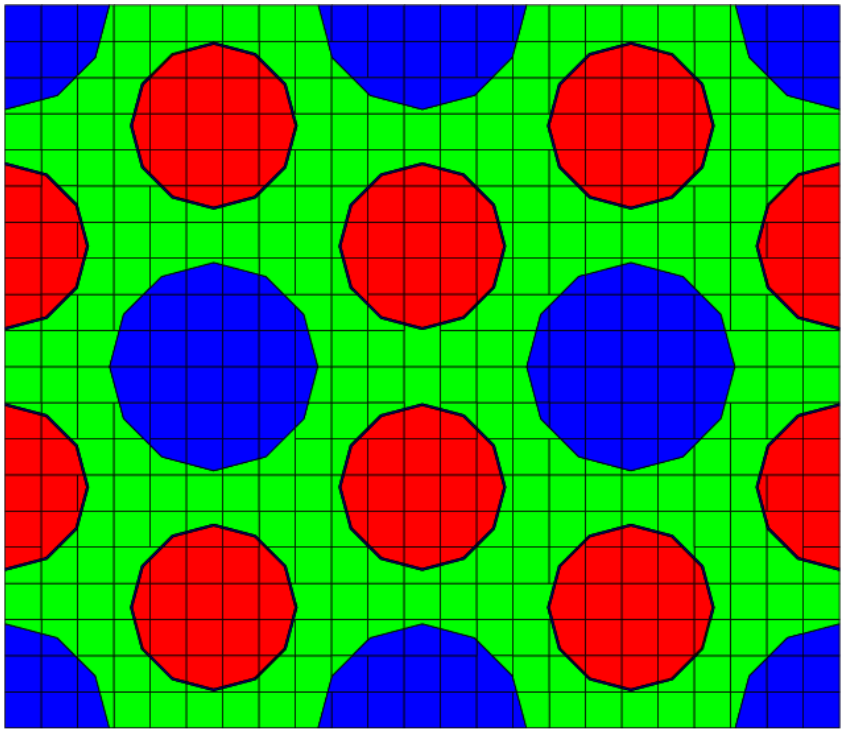

Figure 45: HTGR infinite prismatic lattice: Large circles – coolant channels, smaller circles – fuel.

Fig 46 demonstrates the effect of increasing the rank of the ROM to capture the null space of the model[††]. The figure is constructed as follows: the forward model is executed with N = 100 independent cross-section perturbations and the corresponding k-eigenvalue variations are recorded. After the ROM is constructed, the k-eigenvalue variations resulting from parameters perturbations along the null space are also calculated using direct forward calculations. The value N is determined based on values for the probability and confidence of (p = 95%/c = 95%) from Eq. 99. The red plusses describe the exact variation in the k-eigenvalue from reference value as predicted by the original forward model. The blue circles describe the error in k-eigenvalue resulting from using the ROM with an effective rank r = 40. The black diamonds show the same results but with an r = 90 ROM. The user may choose to increase the rank until a specified tolerance limit is reached. Note that the minimum ROM error cannot ever be lower than the tolerance of the forward calculations for the k-eigenvalue.

Fig. 47 shows the microscopic fission density response results. The blue bars indicate the exact response variations for 100 different random simulations and the red bars represent the ability of sensitivities generated from the ROM in predicting the variation. Results indicate that response variations that are too small, i.e., within the numerical precision of the calculations, may not be well captured by the ROM model.

[††] In our context, the null space refers to the subspace that is orthogonal to the ROM subspace. As described earlier, perturbations that are orthogonal to the ROM subspace (i.e., the null space) are expected to produce negligible variations in the responses.

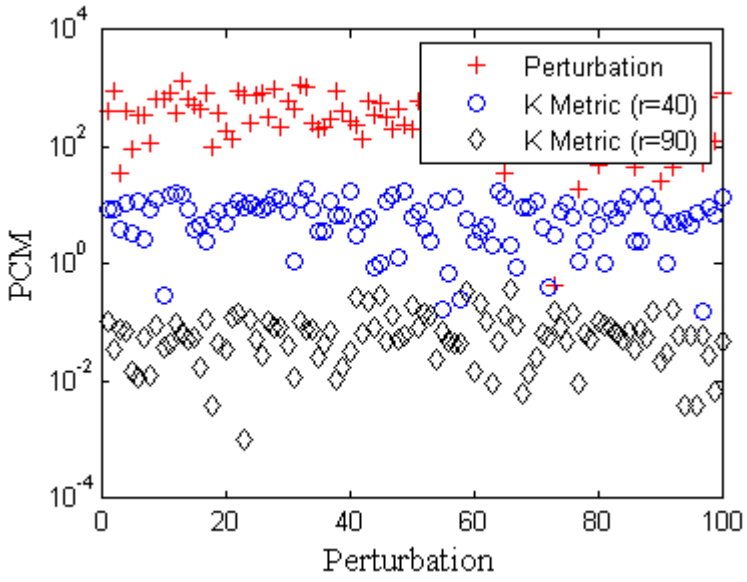

Figure 46: ROM Accuracy for *k*-eigenvalue (HTGR).

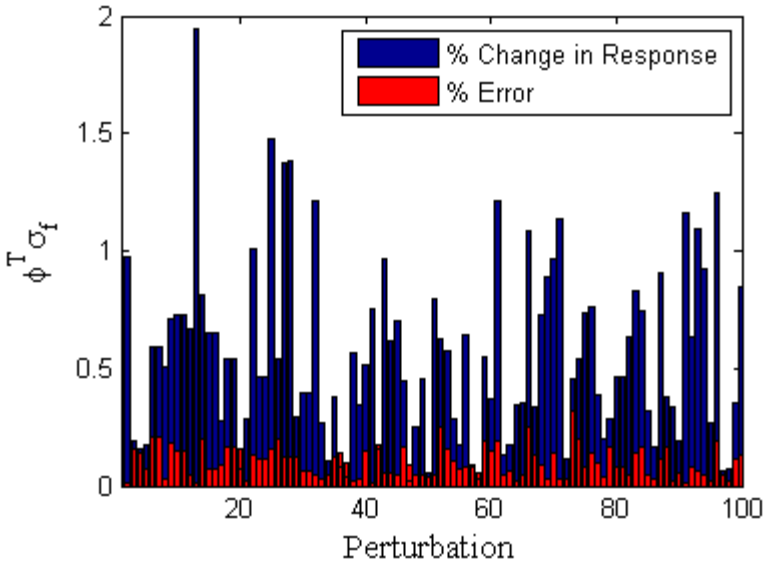

Figure 47: ROM Accuracy for Fission Density (HTGR)

To develop insight into the behavior of the κ-metric, Fig. 48 plots the metric against different rank estimates using both the flux and the *k*-eigenvalue as responses. For reference, the singular values of the matrix **Z** are compared in the same figure. Note that the singular values drop precipitously after approximately $r = 90$, after which the κ-metric levels off showing a plateau type behavior indicating that increasing the rank does not improve the accuracy of the ROM. This behavior is a result of the finite precision of the calculations. In general, if the model outputs have infinite precision, one would expect the accuracy of the ROM to improve until it matches the original model, that is when $r = n$. Given that the *k*-eigenvalue derivatives are employed to construct the ROM, the accuracy of the ROM is not expected to be better than the accuracy of these derivatives. In our case, the derivatives from SCALE were available to five significant digits only which were read from a text file output.

To assess the performance of the ROM, it is employed to predict the variation in the fuel group fluxes for N different perturbations. The comparison of the exact group flux variation calculated by direct forward perturbation is compared to ROM predictions in Fig. 49 in a similar manner to Fig. 47 using 100 benchmark perturbation cases.

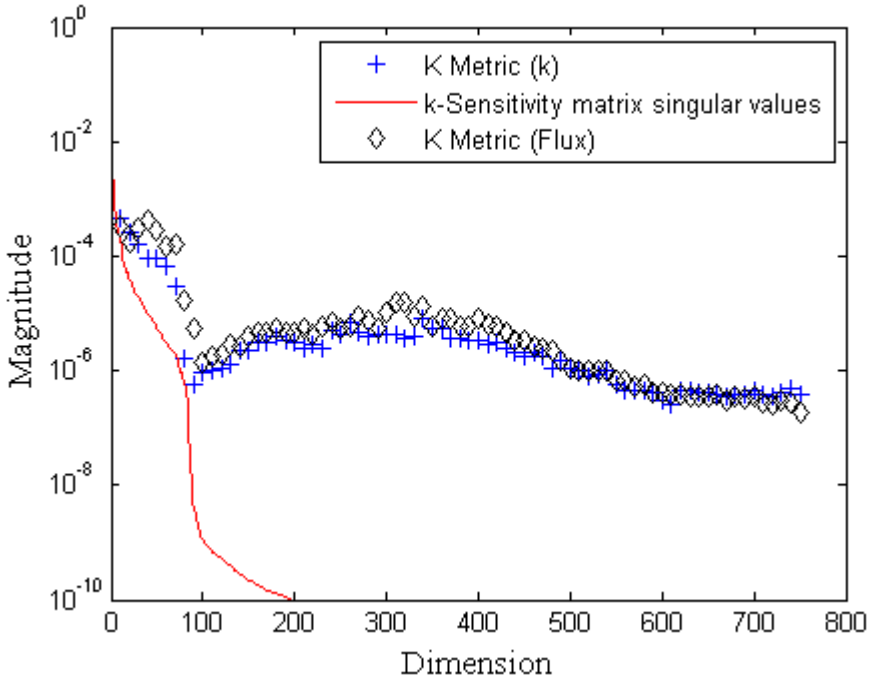

Figure 48: The κ-metric compared to *k*-eigenvalue sensitivity singular values

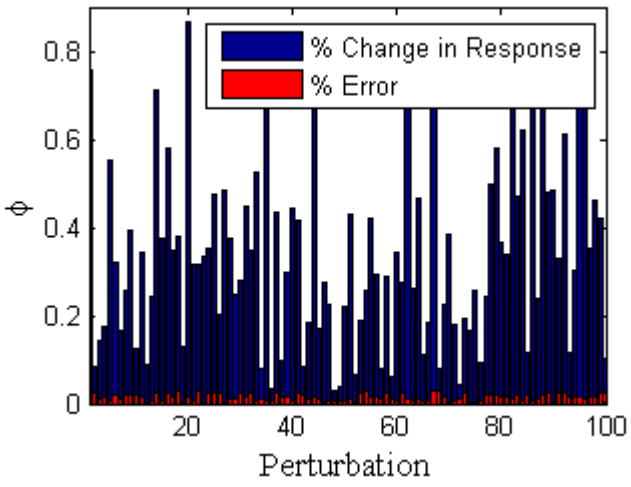

Figure 49: ROM Accuracy for the thermal group flux (HTGR)

Several experiments and responses were used to demonstrate the GPT-free approach to determine response variations with respect to parameters perturbations. Overall, we notice that responses that are strongly correlated with the *k*-eigenvalue, their associated prediction errors are found to be very small. For responses that are weakly correlated with the *k*-eigenvalue, their exact variations are very small and the associated ROM errors are larger. This error however can be user-controlled employing the κ-metric.

6.B Depletion UAM Assembly Model

The UAM assembly model is a component of a benchmark for uncertainty analysis in modeling. The 7x7 LWR assembly contains both UO_2 pins and Gd-loaded pins. The model was readily available in SCALE6.1 (Dehart 2009). A short summary of the assembly parameters is available in Table 4 and a figure of the assembly layout as output from SCALE is in Figure 50. The SCALE 44-Group cross-section library was used for all subsequent SCALE calculations. The model was depleted using TRITON with 14 burn-up time steps as shown in Table 5 with 10% RMS Gaussian random cross-section perturbations introduced at the beginning of cycle (BOC). Cross-sections are introduced independently by isotope, reaction, and energy group. The resulting number densities from the depletion calculation were used to determine a depletion evolution curve, storing both the mean value and the standard deviation of number densities as a function of time for each isotope. Samples of the depletion evolution curve are available in Figures 28-31 in section 5.C.

Table 4: Assembly Parameter Data

UO_2 Fuel Density	10.42 g/cc	**Rod Pitch**	1.785 cm
Fuel Temperature	900 K	**Fuel Radius**	0.60579 cm
Cladding	Zircaloy-2	**Gap Radius**	0.62103 cm
Bin Material	Zircaloy-4	**Clad Radius**	0.71501 cm
Water Density	0.4577 g/cc	**NEWT Cells/Pin**	16 (4x4)
Water Temperature	560 K	**Ext Water Density**	0.738079 g/cc
Gap Material	Helium	**Gap Density (Helium)**	4.9560E-4 g/cc
Clad Temperature	630 K	**Energy Groups**	44
Zirc-2 Density	5.678 g/cc	**Convergence Tolerance**	10^{-6}
Zirc-4 Density	6.525 g/cc	**GadPin-Fuel Fraction**	0.97
GadPin Density	10.29 g/cc	**GadPin-Gd Fraction**	0.03

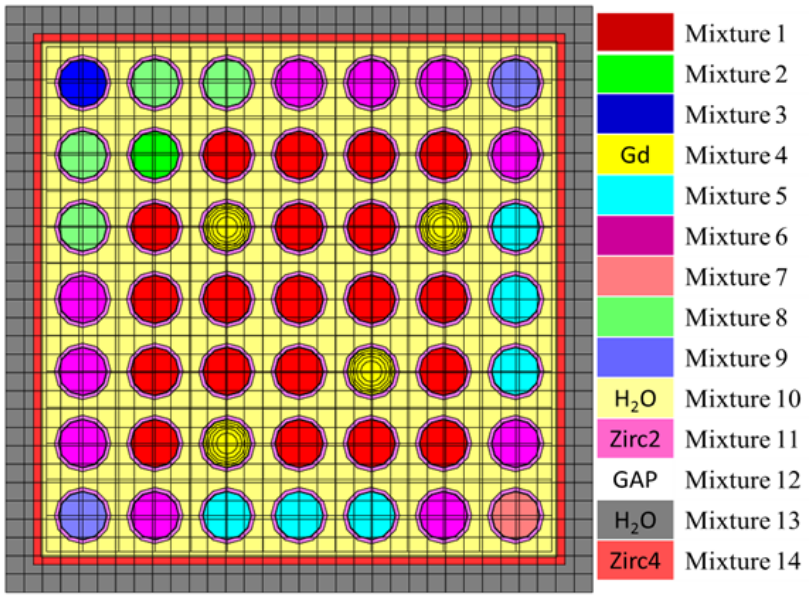

Figure 50: Depletion UAM assembly model

Table 5: Burn-up time points (days) at 22.220 MW/MTHM Specific Power

0	1	6	12	18
36	72	108	270	360
540	720	900	1080	

The GPT-Free approach was then run in order to construct the ROM by randomly sampling points in time. From these points in time, the nuclide densities were obtained by sampling the depletion-evolution curve data. Random Gaussian cross-section perturbations with magnitude of 10% RMS were then introduced independently by isotope, reaction, and energy group. The resulting model was generated by preprocessing PYTHON scripts and submitted to SCALE61 TSUNAMI-2D. The sensitivities of these many simulations were then collected into a matrix and evaluated by the κ-metric. This process was repeated to determine the rank of the ROM.

Table 6: Perturbed actinide list for the UAM assembly model

Am-241	Cm-243	Cm-246
Am-243	Cm-244	U-234
Cm-242	Cm-245	U-235
U-236	Pu-238	Pu-241
U-238	Pu-239	Pu-242
Np-237	Pu-240	Pu-243

The global κ-metric response figures in Figs. 51 and 52 are based on the following pair of equations:

$$\kappa = \left\{ 10^5 \left(\frac{\left| k^{\perp} - k_0 \right|}{k_0} \right) \right\} \tag{101}$$

$$\kappa = \left\{ \frac{\sum_{i=1,j=1}^{m,g} \left| \Delta\phi_{mg}^{\perp} \right|}{\sum_{i=1,j=1}^{m,g} \left| \Delta\phi_{mg} \right|} \right\} \quad m = \text{mixture} \quad g = \text{group} \tag{102}$$

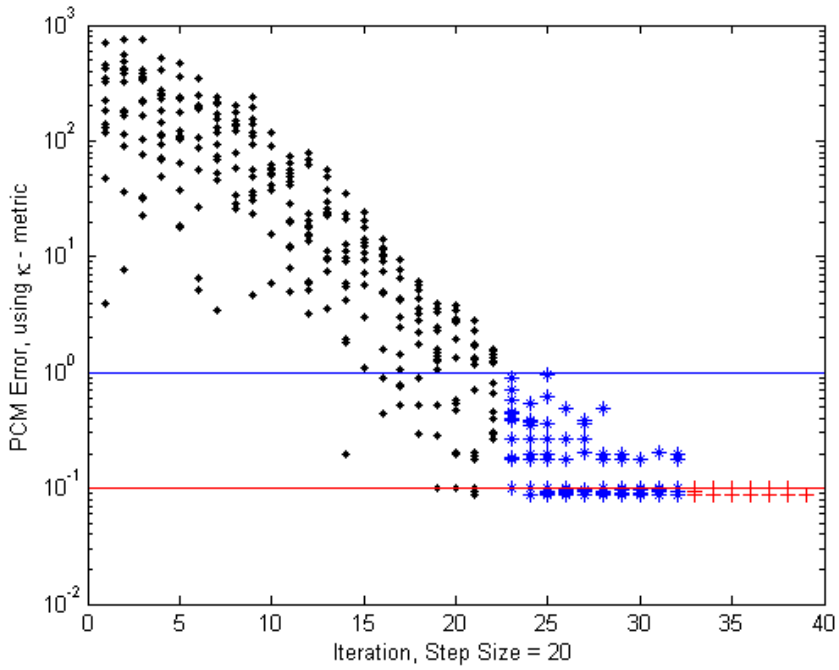

Figure 51: κ-metric plot using Eq. 109 for increasing ROM dimension.

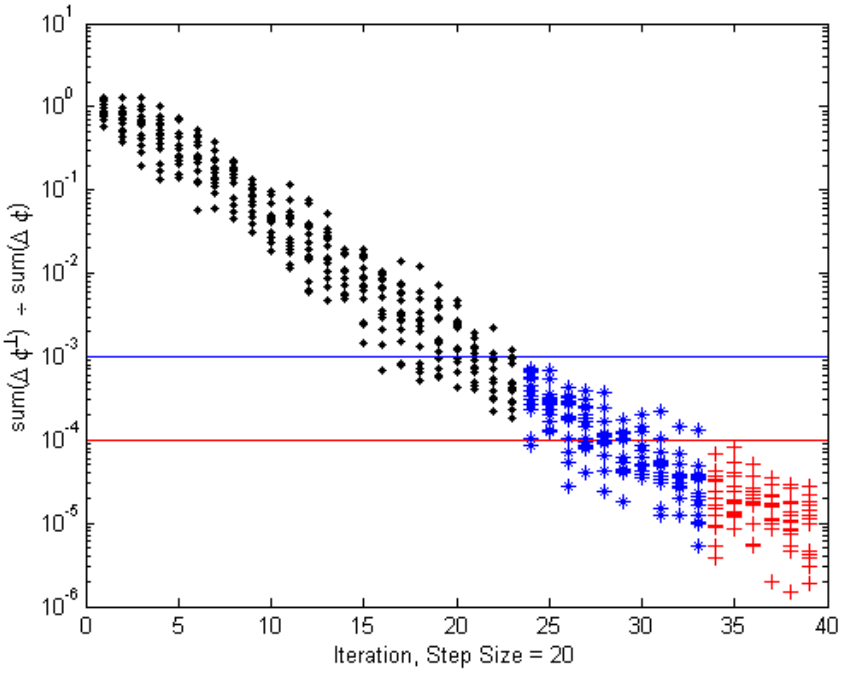

Figure 52: κ-metric plot using Eq. 110 for increasing ROM dimension.

In these figures, the black dots at points along the x-axis represent a sample of the κ-metric at a given ROM iteration. With a step-size of twenty, the ROM dimension at any point is 20*x*. The blue line and dots indicate the cutoff according to a tolerance of 1 pcm error. Furthermore, the red line indicates a cutoff at 10^{-6} corresponding to 0.1 pcm.

The process to test the ROM is to use a reference set of *N* benchmark calculations according to Eq. 99. The specific sets of cross-section perturbations introduced are stored so that the ROM can be tested by estimating the null-space. Given some input space **Q** as determined by the ROM of increasing dimension, the orthogonal component can be tested as follows:

$$\Pi\left(\sigma_0 + \left(\mathbf{I} - \mathbf{Q}\mathbf{Q}^T\right)\Delta\sigma, \phi^{\perp}, k^{\perp}\right) = 0 \tag{103}$$

If the ROM is accurate, the following should hold to a user-defined tolerance:

$$\begin{aligned} \phi^{\perp} - \phi &= 0 \\ k^{\perp} - k &= 0 \end{aligned} \tag{104}$$

As shown in Figs. 51 and 52, the error from the orthogonal components represented by Eq. 112 as measured by Eqs. 109 and 110 are decreasing with increasing ROM dimension up to the numerical tolerance of the software.

While the global ROM is of desire for general determination of the ROM dimension, the ROM can be tested further with the κ-metric by examining the error in the flux for each mixture in each energy group. Samples of these mixture and energy-group specific flux plots are available in Figs. 53-58. The error is calculated as follows:

$$\kappa_{mg} = \left\{\left|\phi_{m,g} - \phi_{0m,g}\right|\right\} \quad m = \text{mixture} \quad g = \text{group} \tag{105}$$

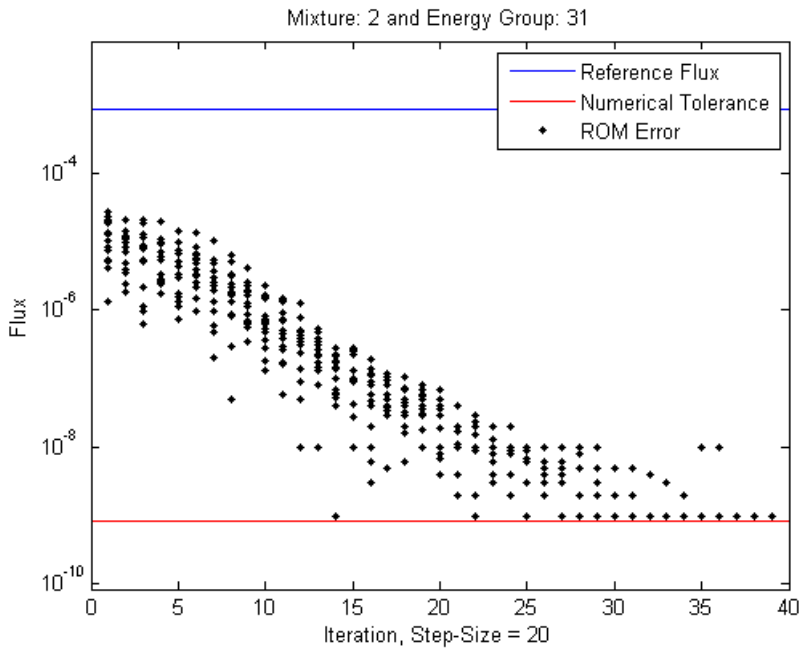

Figure 53: κ-metric plot for mixture 2, energy group 31.

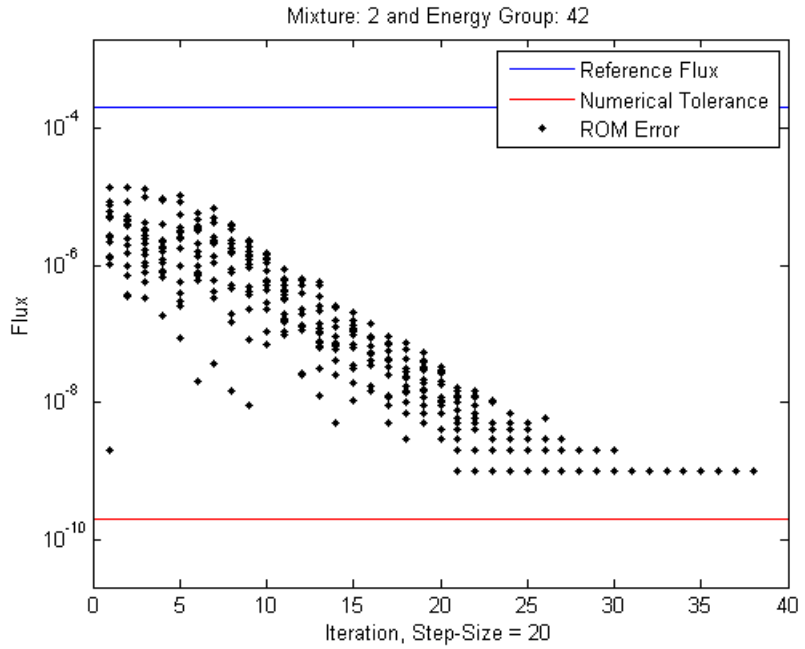

Figure 54: κ-metric plot for mixture 2, energy group 42.

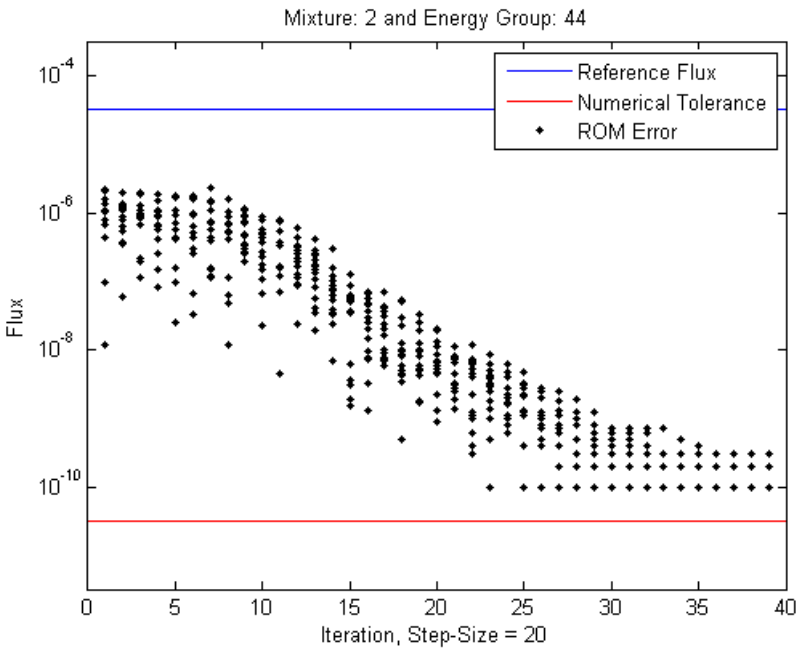

Figure 55: κ-metric plot for mixture 2, energy group 44.

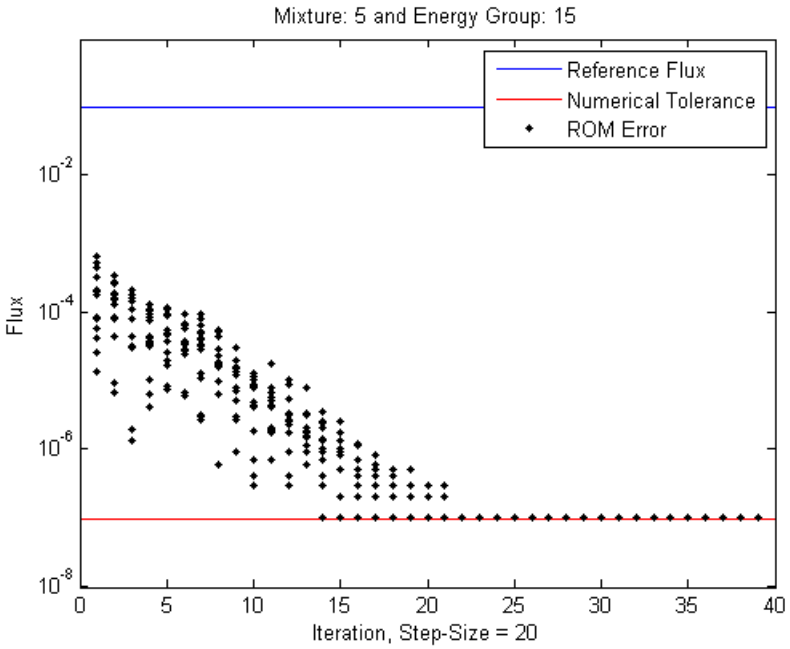

Figure 56: κ-metric plot for mixture 5, energy group 15.

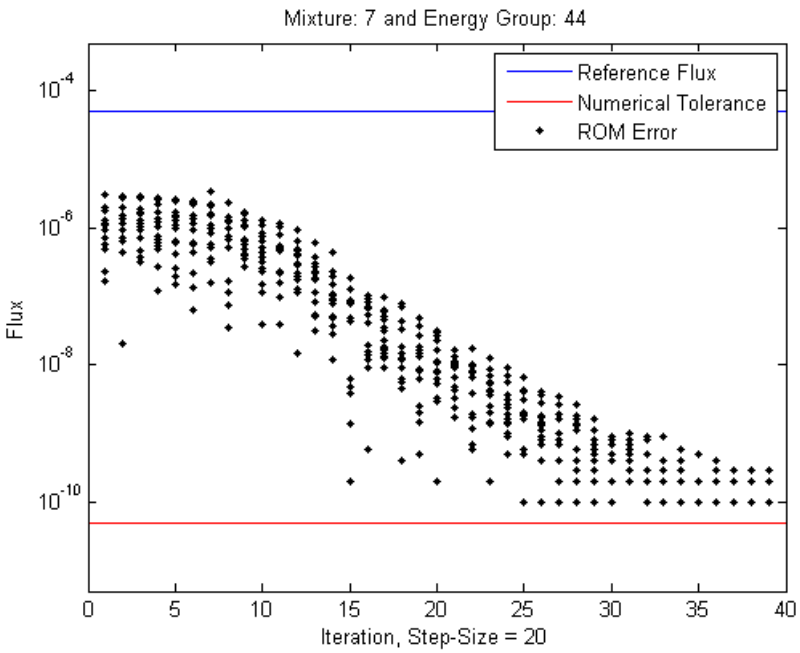

Figure 57: κ-metric plot for mixture 7, energy group 44.

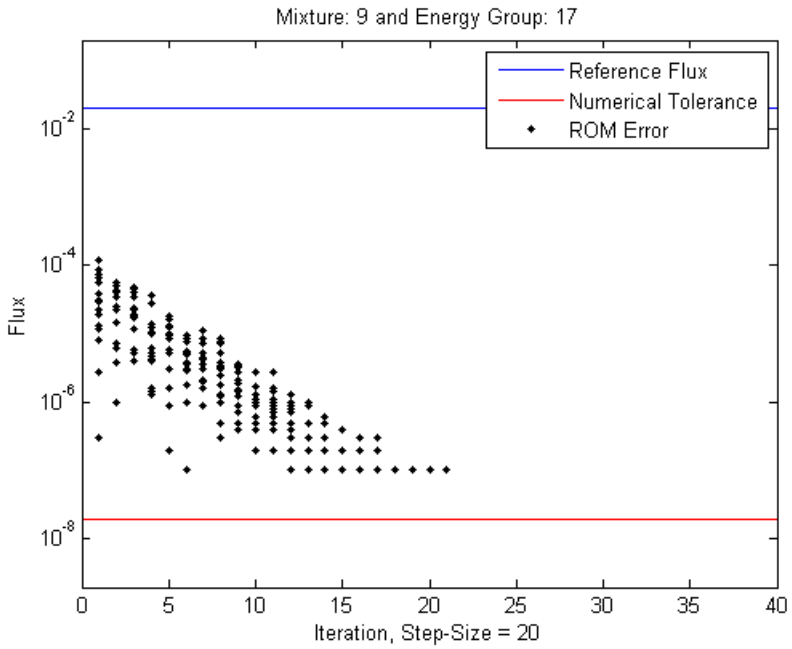

Figure 58: κ-metric plot for mixture 9, energy group 17.

As can be seen in Figs 53-58, some mixtures and energy groups require a larger dimensional ROM to capture the effects in a single pin and energy-group. Similarly, some pins and energy groups, e.g. Fig. 58, do not require as large of an ROM to capture the effect of the model. Overall the global responses in Figs 51 and 52 tend to be limited by the larger individual components such as in Fig. 57 and are thus fairly reliable.

One can use the κ-metric to estimate the effectiveness of the ROM in calculating reaction rates and reaction rate ratios. The following set of 14 reaction rate densities, ratios, and collapsed one-group cross-sections in Table 7 were selected as a sample to test the ROM effectiveness. The inner products are in energy, and are mixture specific.

Table 7: List of examined reaction inner products examined by the κ-metric.

Reaction	Description	Reaction	Description
1: $^{235}\mathrm{U}$: $\langle \sigma_f, \phi \rangle_E$	U-235 Fission	2: $^{235}\mathrm{U}$: $\langle \sigma_\gamma, \phi \rangle_E$	U-235 Radiative Capture
3: $^{238}\mathrm{U}$: $\langle \sigma_f, \phi \rangle_E$	U-238 Fission	4: $^{238}\mathrm{U}$: $\langle \sigma_\gamma, \phi \rangle_E$	U-238 Radiative Capture
5: $^{239}\mathrm{Pu}$: $\langle \sigma_f, \phi \rangle_E$	Pu-239 Fission	6: $^{239}\mathrm{Pu}$: $\langle \sigma_\gamma, \phi \rangle_E$	Pu-239 Radiative Capture
7: $^{235}\mathrm{U}$: $\dfrac{\langle \sigma_f, \phi \rangle_E}{\langle \phi, 1 \rangle_E}$	One-group U-235 Fission XS	8: $^{238}\mathrm{U}$: $\dfrac{\langle \sigma_\gamma, \phi \rangle_E}{\langle \phi, 1 \rangle_E}$	One-group U-238 Capture XS
9: $^{239}\mathrm{Pu}$: $\dfrac{\langle \sigma_f, \phi \rangle_E}{\langle \phi, 1 \rangle_E}$	One-group Pu-239 Fission XS	10: $\left(\dfrac{\langle \sigma_\gamma, \phi \rangle_E}{\langle \sigma_f, \phi \rangle_E} \right)_{\mathrm{U-235}}$	Capture-Fission ratio, U-235
11: $\left(\dfrac{\langle \sigma_\gamma, \phi \rangle_E}{\langle \sigma_f, \phi \rangle_E} \right)_{\mathrm{U-238}}$	Capture-Fission ratio, U-238	12: $\left(\dfrac{\langle \sigma_\gamma, \phi \rangle_E}{\langle \sigma_f, \phi \rangle_E} \right)_{\mathrm{Pu-239}}$	Capture-Fission ratio, Pu-239
13: $\left(\dfrac{^{238}\mathrm{U}: \langle \sigma_f, \phi \rangle_E}{^{235}\mathrm{U}: \langle \sigma_f, \phi \rangle_E} \right)$	Fission Fraction U-238 : U-235	14: $\left(\dfrac{^{239}\mathrm{Pu}: \langle \sigma_f, \phi \rangle_E}{^{235}\mathrm{U}: \langle \sigma_f, \phi \rangle_E} \right)$	Fission Fraction Pu-239 : U-235

In examining the ROM for the responses in Table 7, care must be taken to adequately describe the ROM error. Because the introduced changes in cross-sections are known, the error contribution should fully be the effect of flux variations, in effect this is a measurement of the error in calculating the indirect term. Thus the error, as compared to reference calculations, will be taken at the point of the perturbed cross-sections. Because of cross-section weighting, and the fact that the error is highest in the thermal groups where cross-sections are the largest, some reaction rate responses will be affected. The fourteen different responses are considered for the assembly-center pin, mixture 1 and plotted in Figs. 59-72.

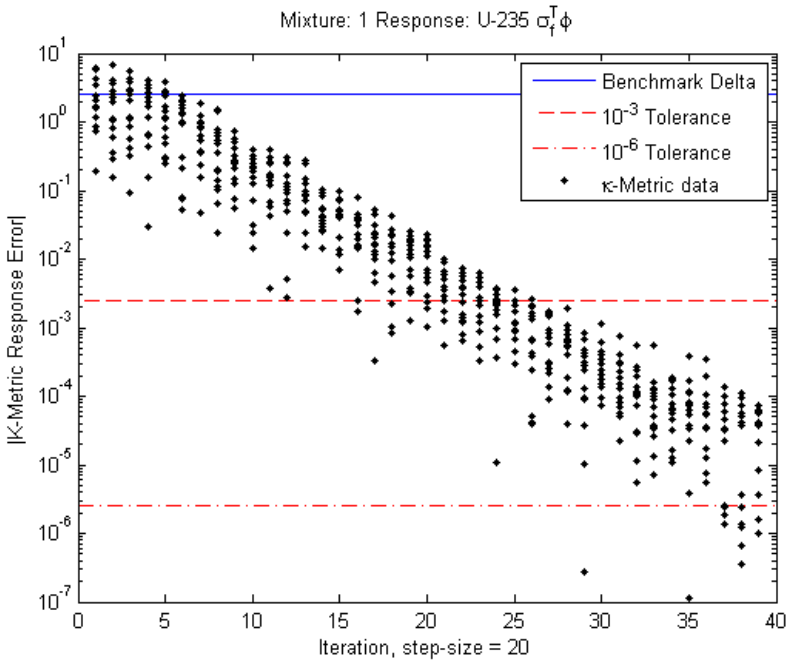

Figure 59: κ-metric plot for ^{235}U fission reaction density in mixture 1.

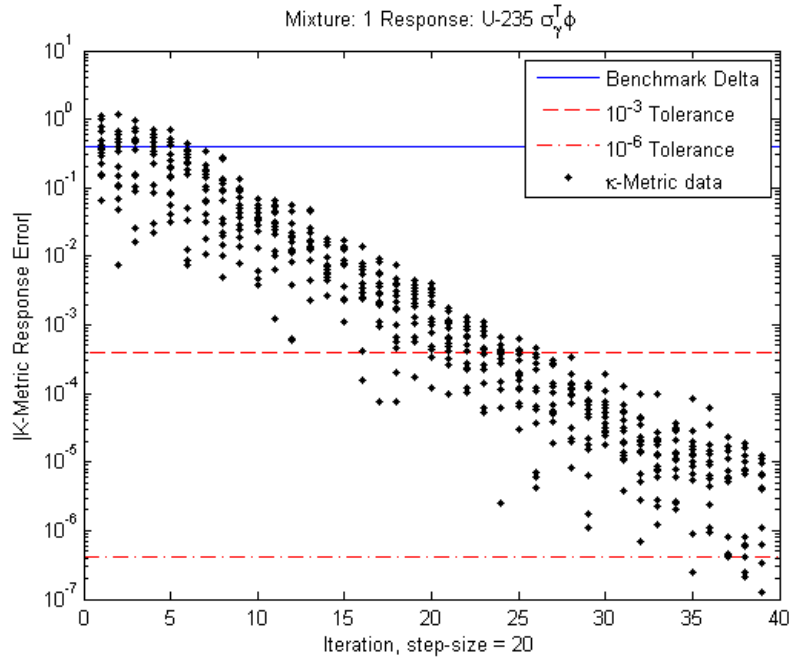

Figure 60: κ-metric plot for ^{235}U capture reaction density in mixture 1.

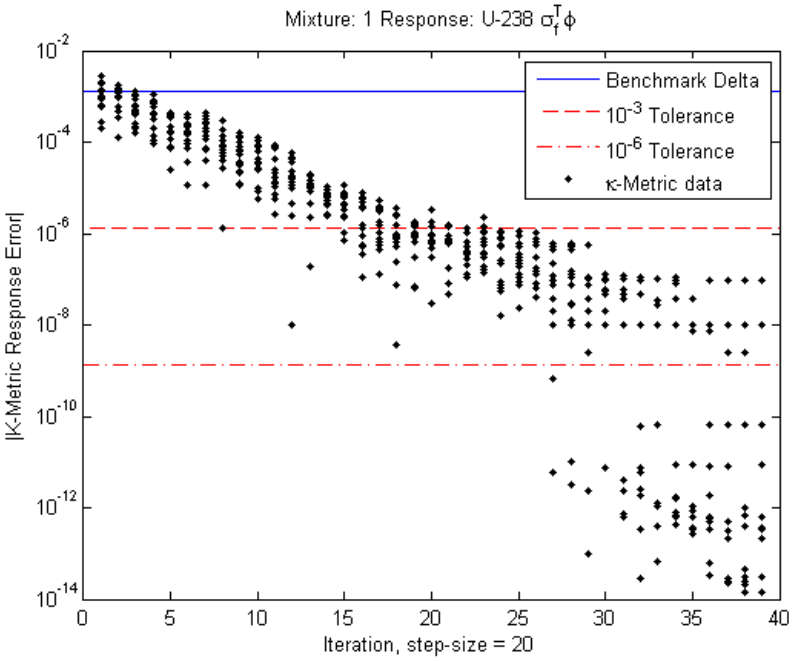

Figure 61: κ-metric plot for ^{238}U fission reaction density in mixture 1.

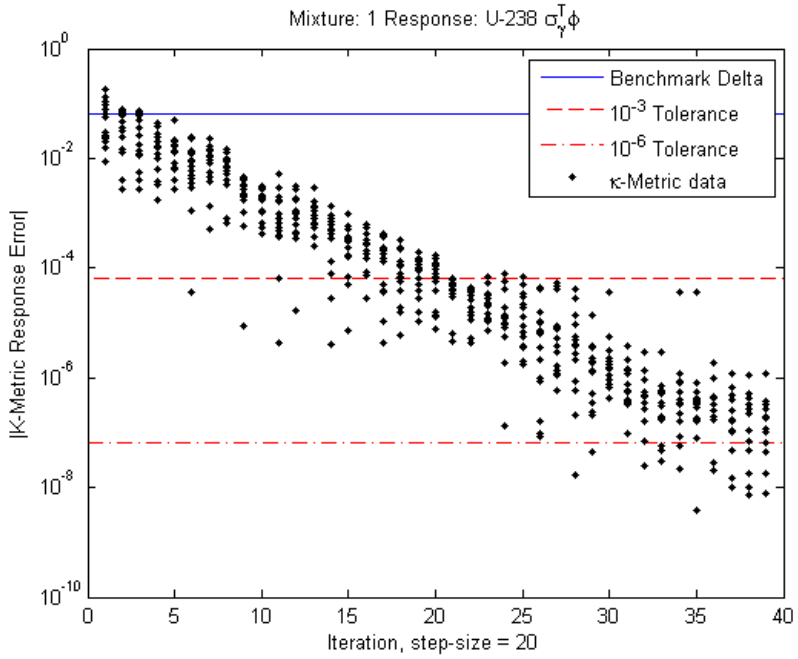

Figure 62: κ-metric plot for ^{238}U capture reaction density in mixture 1.

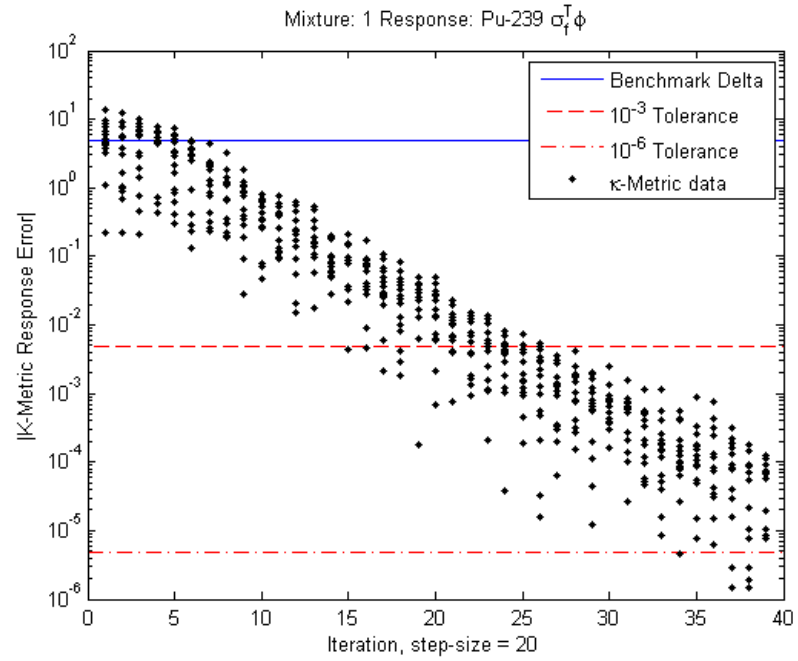

Figure 63: κ-metric plot for ^{239}Pu fission reaction density in mixture 1.

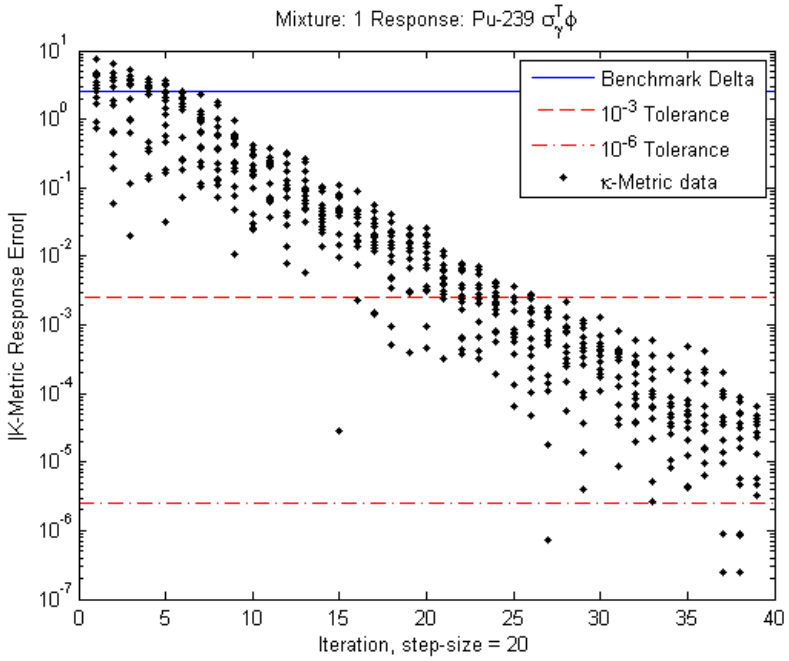

Figure 64: κ-metric plot for ^{239}Pu capture reaction density in mixture 1.

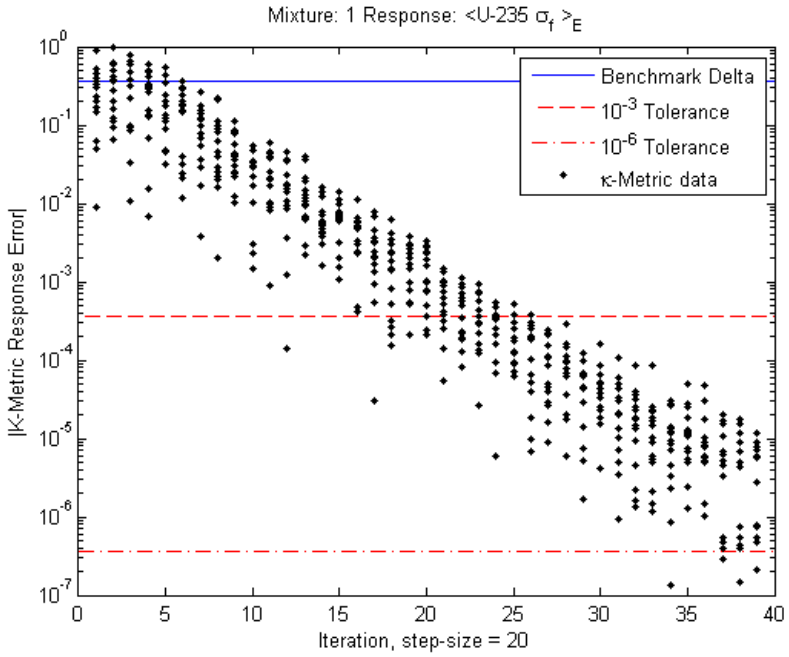

Figure 65: κ-metric plot for ^{235}U one-group fission cross-section in mixture 1.

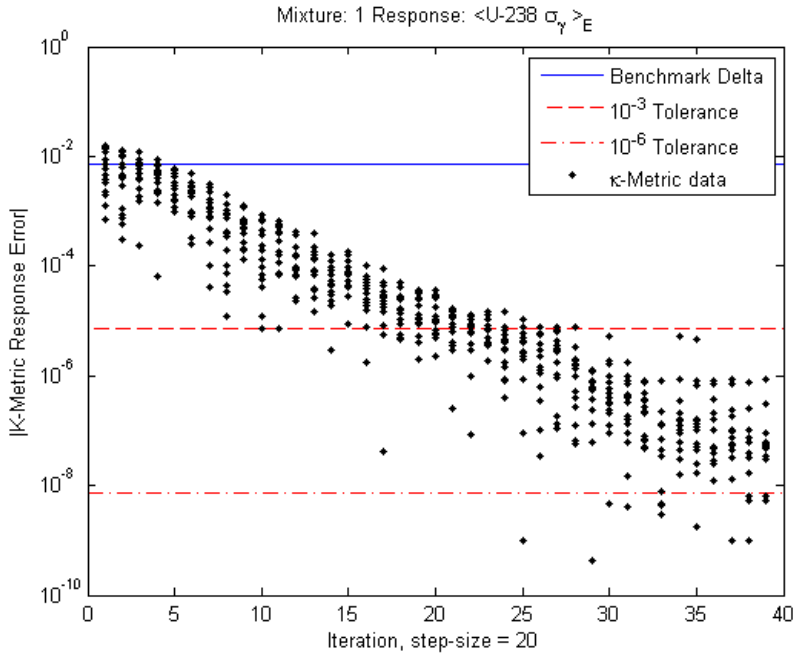

Figure 66: κ-metric plot for ^{238}U one-group capture cross-section in mixture 1.

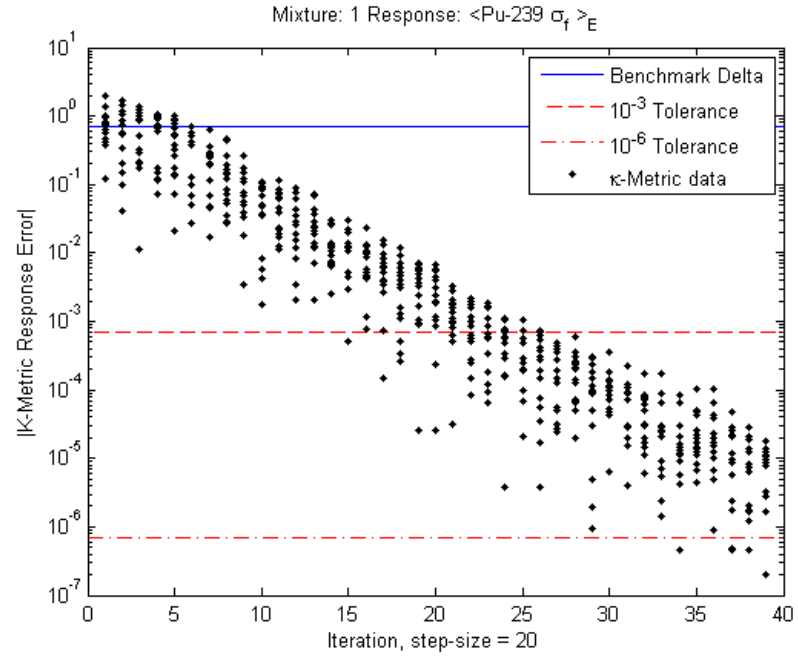

Figure 67: κ-metric plot for ^{239}Pu one-group fission cross-section in mixture 1.

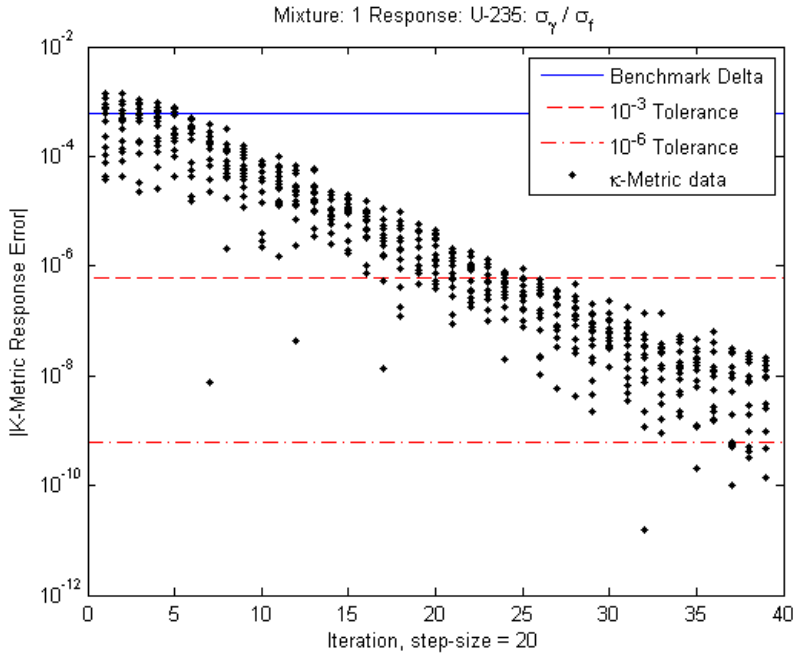

Figure 68: κ-metric plot for ^{235}U capture-to-fission ratio in mixture 1.

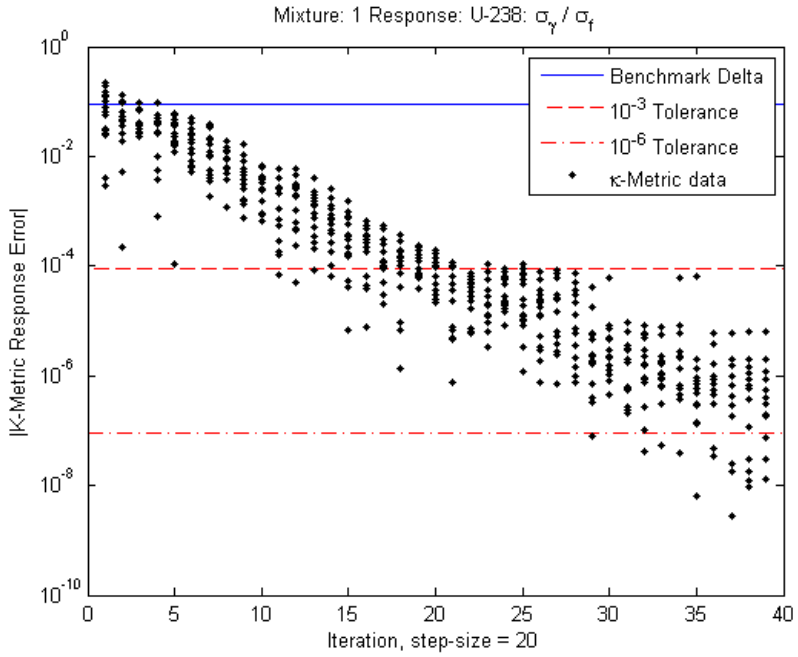

Figure 69: κ-metric plot for ^{238}U capture-to-fission ratio in mixture 1.

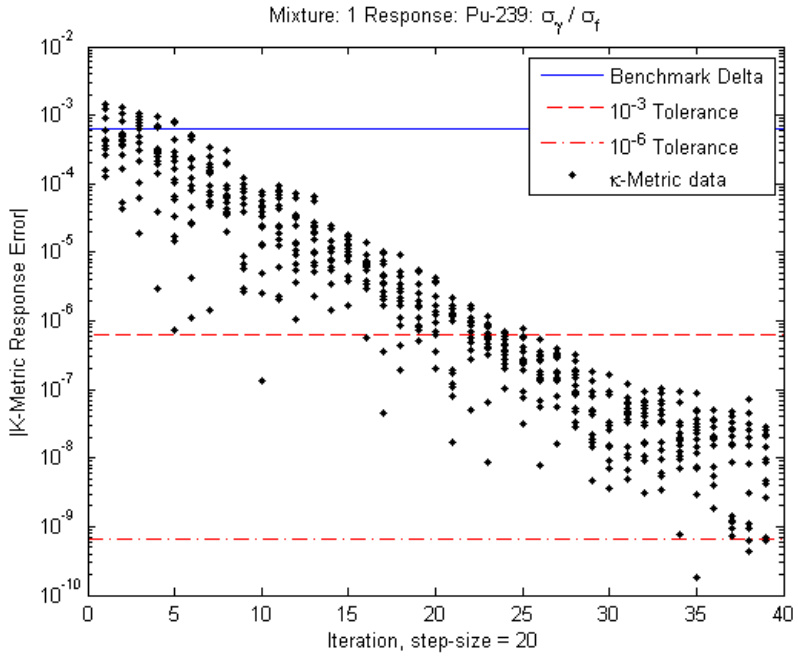

Figure 70: κ-metric plot for ^{239}Pu capture-to-fission ratio in mixture 1.

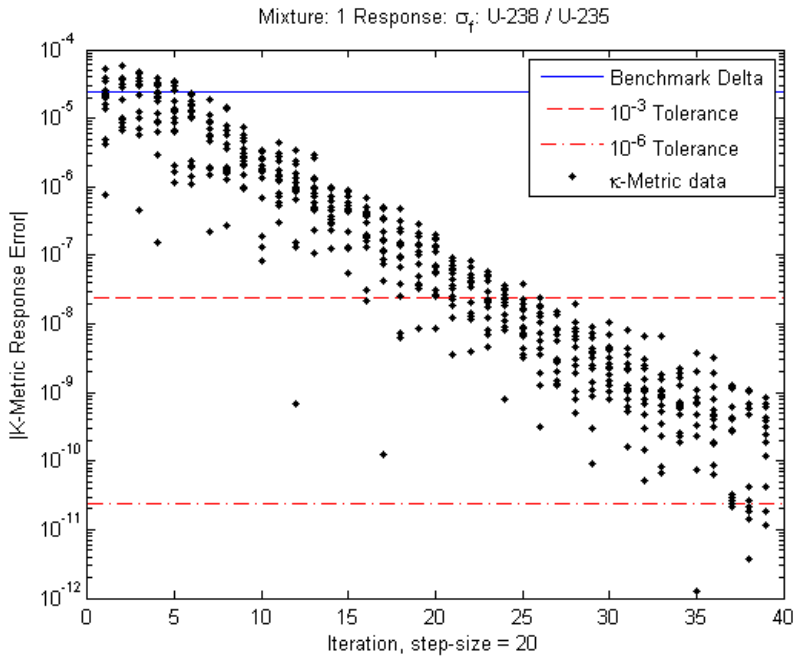

Figure 71: κ-metric plot for the fission ratio of ^{238}U to ^{235}U in mixture 1.

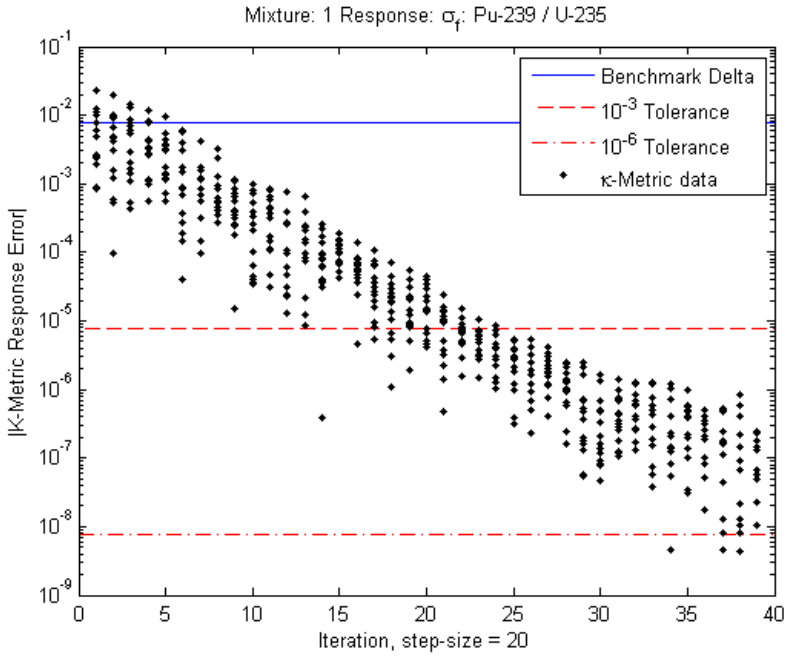

Figure 72: κ-metric plot for the fission ratio of ^{239}Pu to ^{235}U in mixture 1.

While results are provided for a single mixture, the results do not change appreciably for other fuel-pin mixtures. For comparison, consider Figs. 73 and 74, which show the same reaction rate ratio as Figs. 71 and 72 but for mixture 3, a corner fuel-pin. Comparable trends in reduction are visible. In each plot presented, the solid blue line indicates the average change from the reference case using 10% RMS Gaussian cross-section perturbations. The red dash and dash-dot lines indicate thresholds for 10^{-3} and 10^{-6} respectively relative to the average change marked by the blue line. The black dots indicate measurements of the ROM error using the orthogonal flux at the indicated ROM dimension 20*x*.

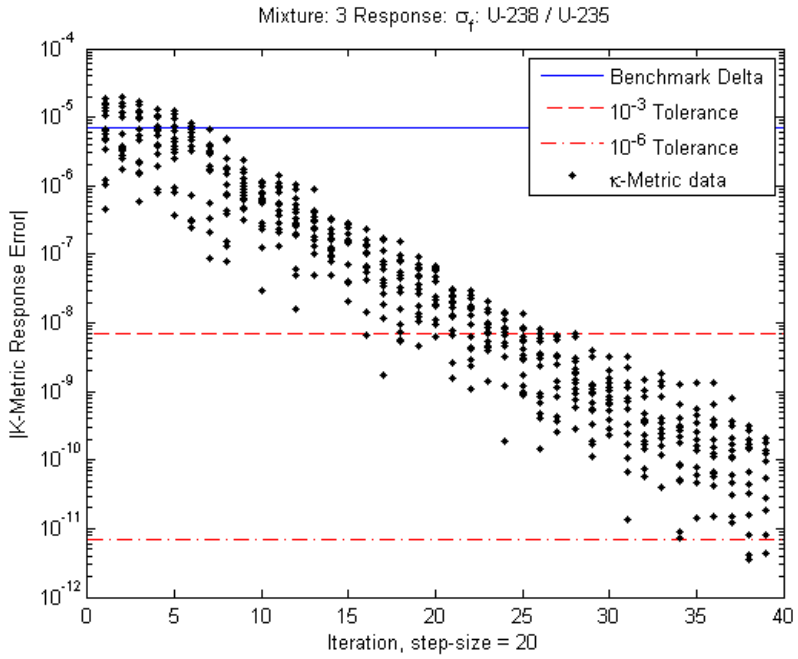

Figure 73: κ-metric plot for the fission ratio of ^{238}U to ^{235}U in mixture 3.

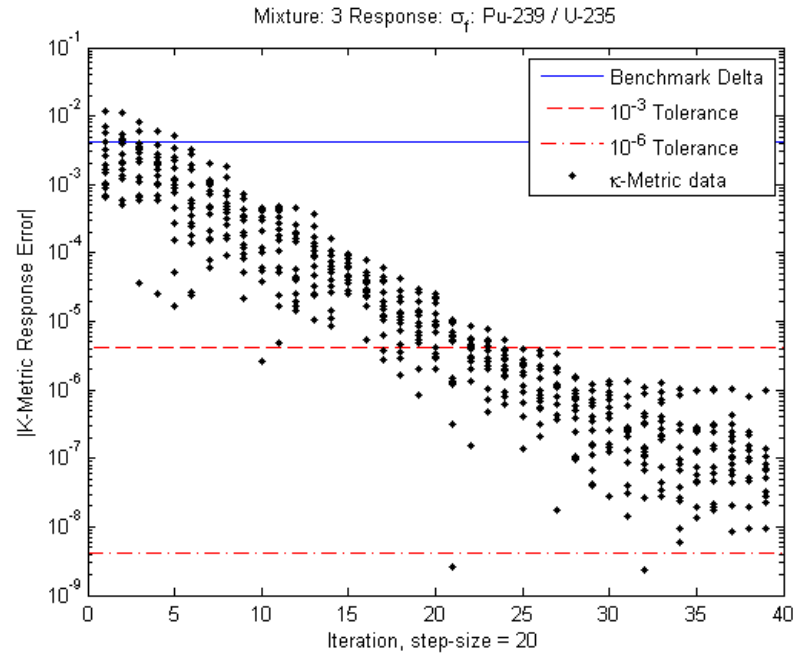

Figure 74: κ-metric plot for the fission ratio of ^{239}Pu to ^{235}U in mixture 3.

The sensitivities of the flux values for each mixture and energy group were calculated using the least squares approach mentioned in section V of this dissertation. Sensitivities were calculated using a ROM of dimension 660 given 700 samples. A set of 100 random 10% RMS cross-section perturbations were introduced into the model to give a set of actual model changes due to cross-section perturbations. The sensitivities were then used to predict the changes in the flux for all 100 test cases. The following figures demonstrate the result of this numerical experiment comparing the change in the flux value to the error in prediction due a combination of ROM error and nonlinearity. Cross-section perturbations were introduced at time zero, and depleted up to 990 days. The samples were carried through the 15 depletion points. Figures 75-83 demonstrate the test for the center-pin (mixture 1) at beginning of cycle for the fast, intermediate, and thermal energy ranges. Similarly, Figures 84-92 show the same energy groups, again for mixture 1, at the end-of-cycle (990 days). Overall results throughout the spectrum show errors similar to the fast and intermediate energy ranges; however, the error in the thermal regions is larger, due to a few appreciable factors. First, the flux in the lowest energy groups for the UAM assembly model are the smallest in the model, appreciating the highest ROM error as seen before using the κ-metric (see Figs. 55 and 57). Second, the thermal flux varies strongly (relatively) despite the small value (Fig. 33). Third, the variation of the lowest energy group is right near the numerical tolerance for the simulation (Fig. 55). These three factors, combined with the existence of nonlinearity due to 10% RMS cross-section perturbations, leads to the expectation that these results for the lowest energy groups are less favorable as compared to the intermediate and higher energy group using the GPT-Free ROM.

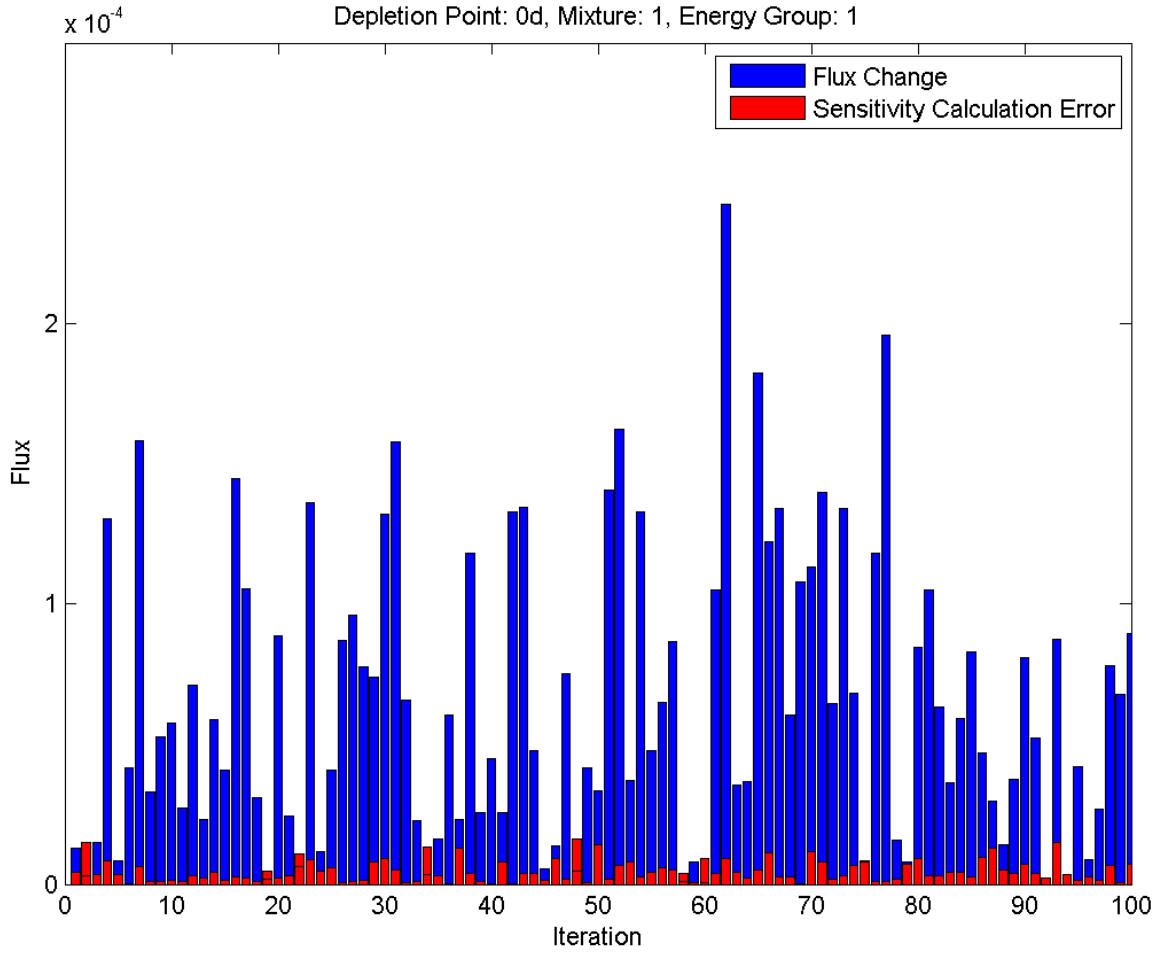

Figure 75: GPT-Free Numerical Demo, BOC, Mixture 1, Group 1, 100 Samples with 10% RMS XS Perturbations

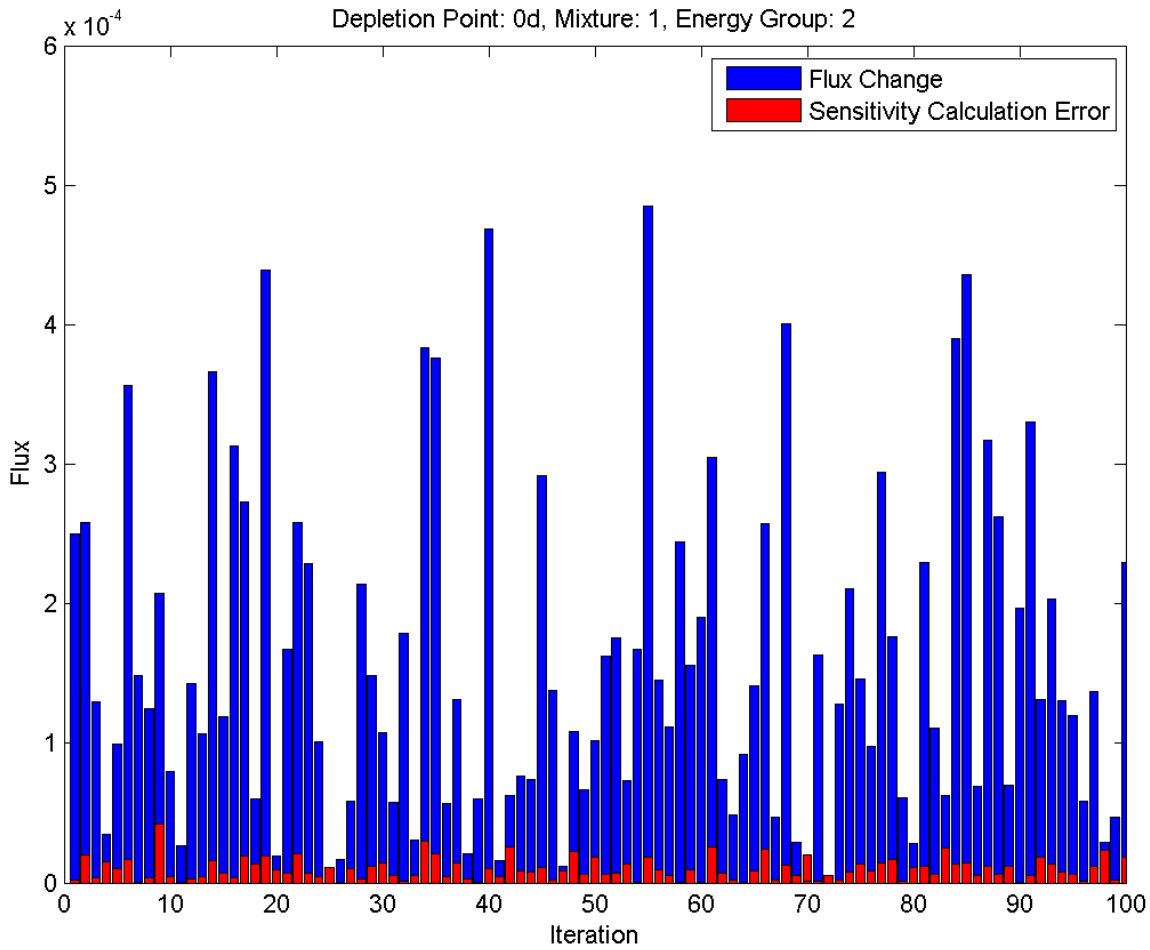

Figure 76: GPT-Free Numerical Demo, BOC, Mixture 1, Group 2, 100 Samples with 10% RMS XS Perturbations

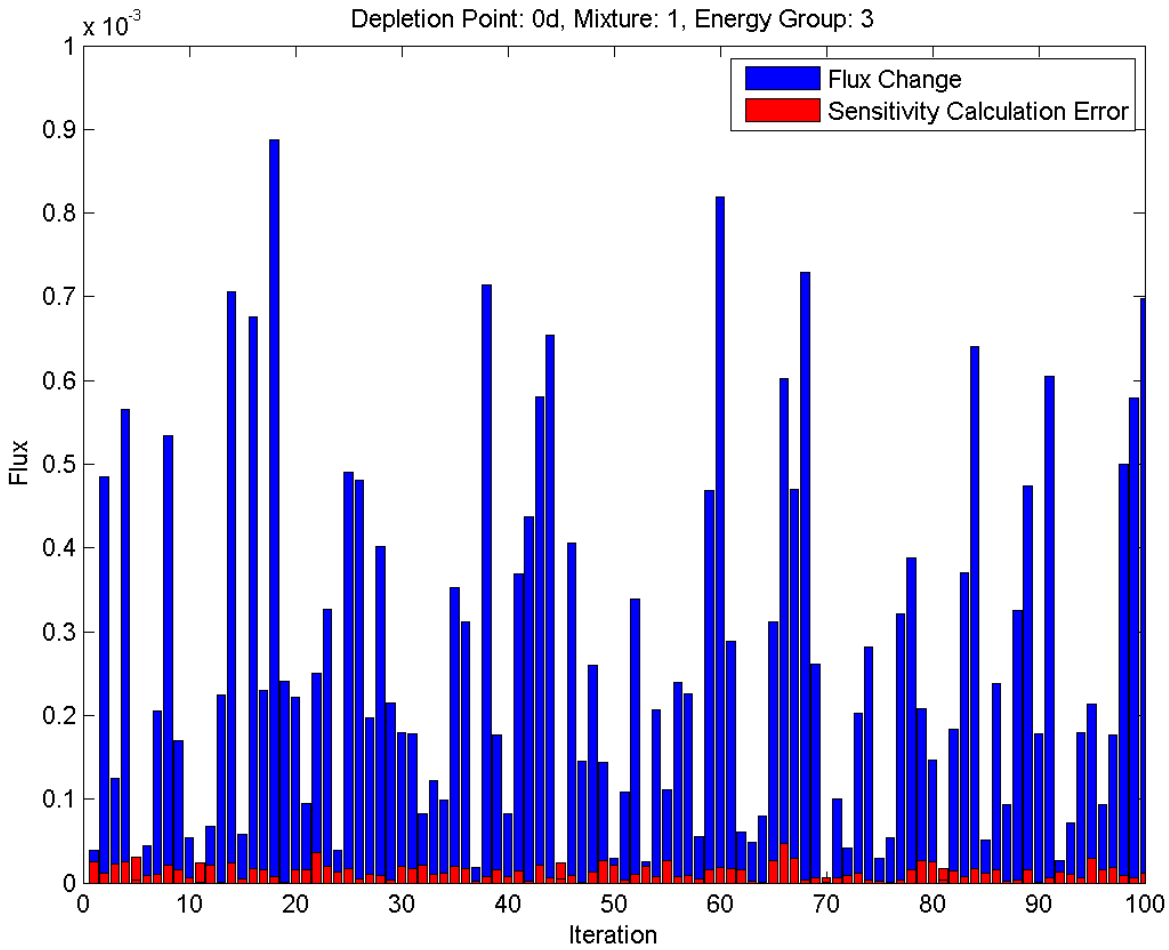

Figure 77: GPT-Free Numerical Demo, BOC, Mixture 1, Group 3, 100 Samples with 10% RMS XS Perturbations

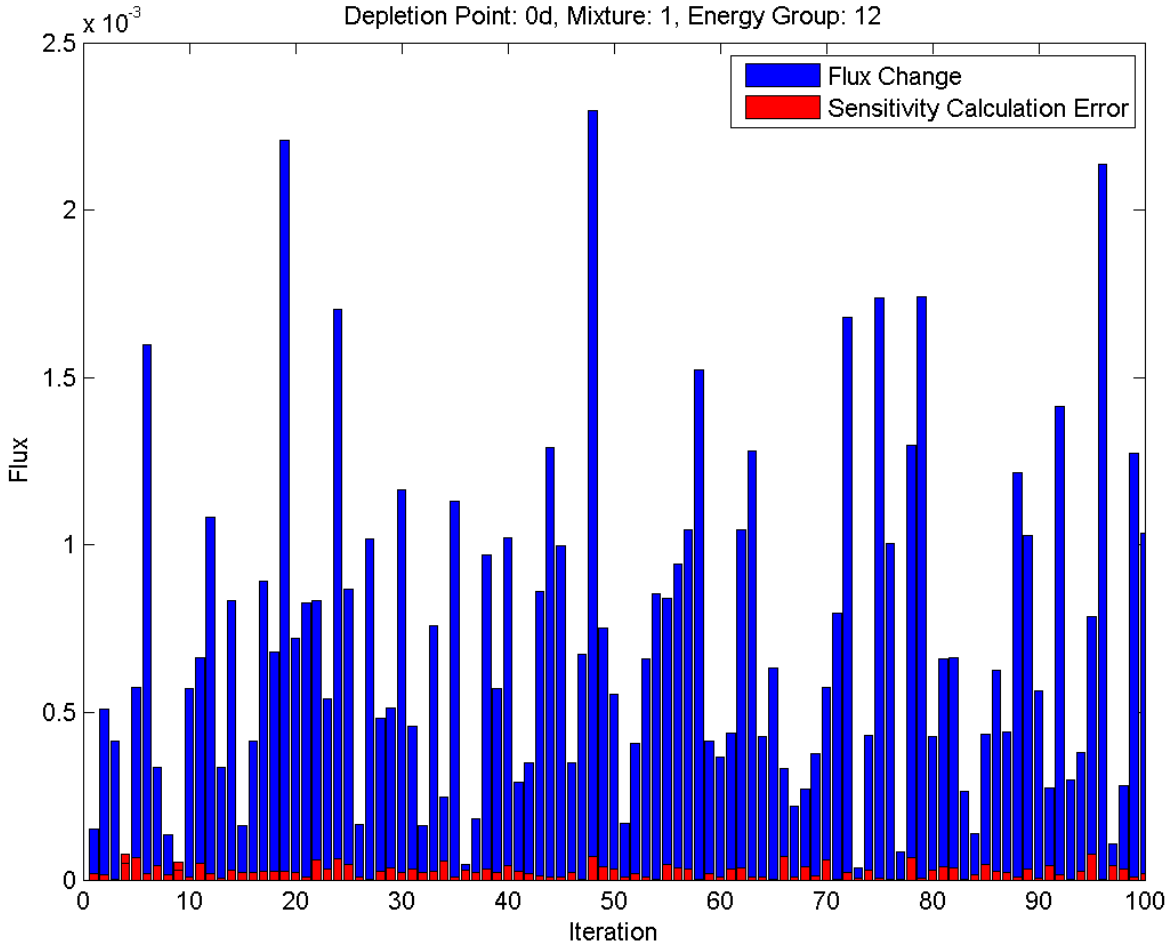

Figure 78: GPT-Free Numerical Demo, BOC, Mixture 1, Group 12, 100 Samples with 10% RMS XS Perturbations

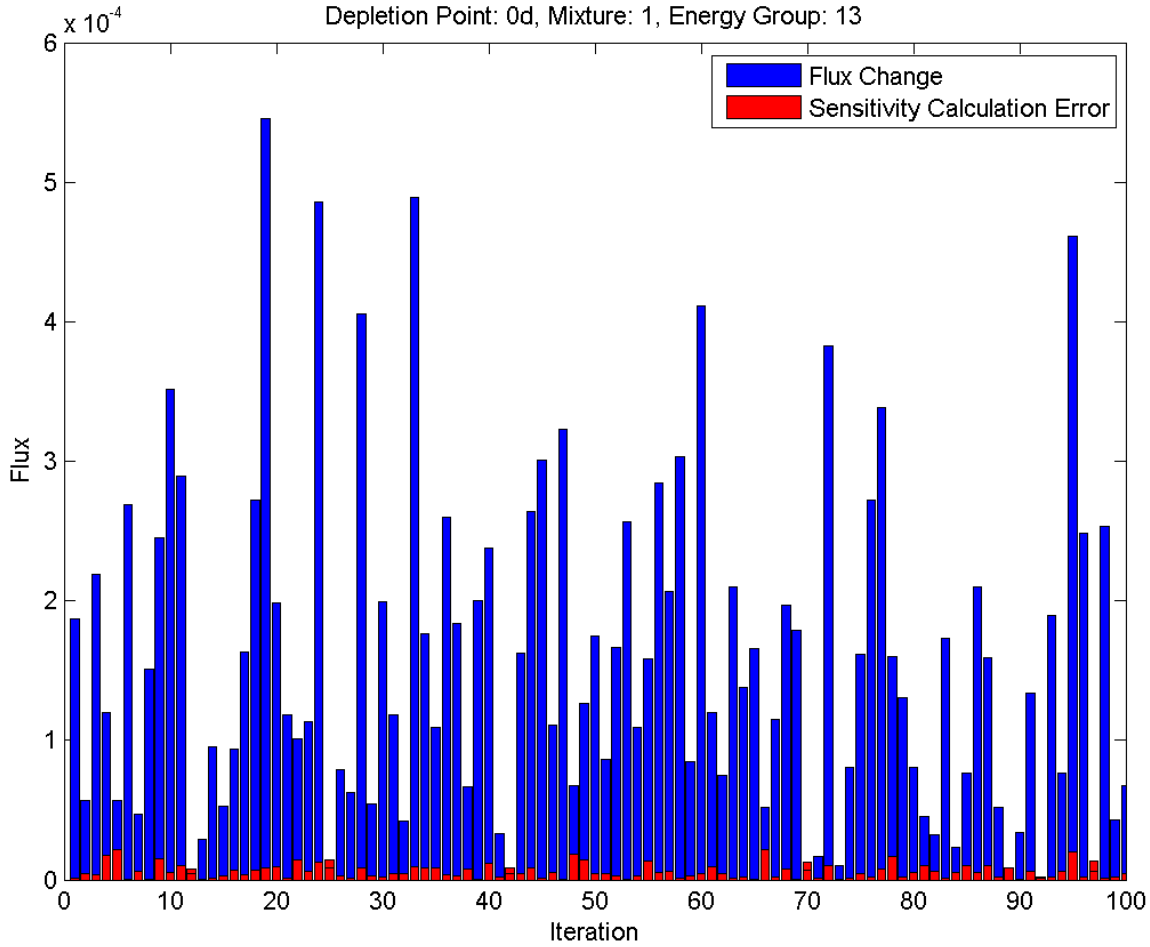

Figure 79: GPT-Free Numerical Demo, BOC, Mixture 1, Group 13, 100 Samples with 10% RMS XS Perturbations

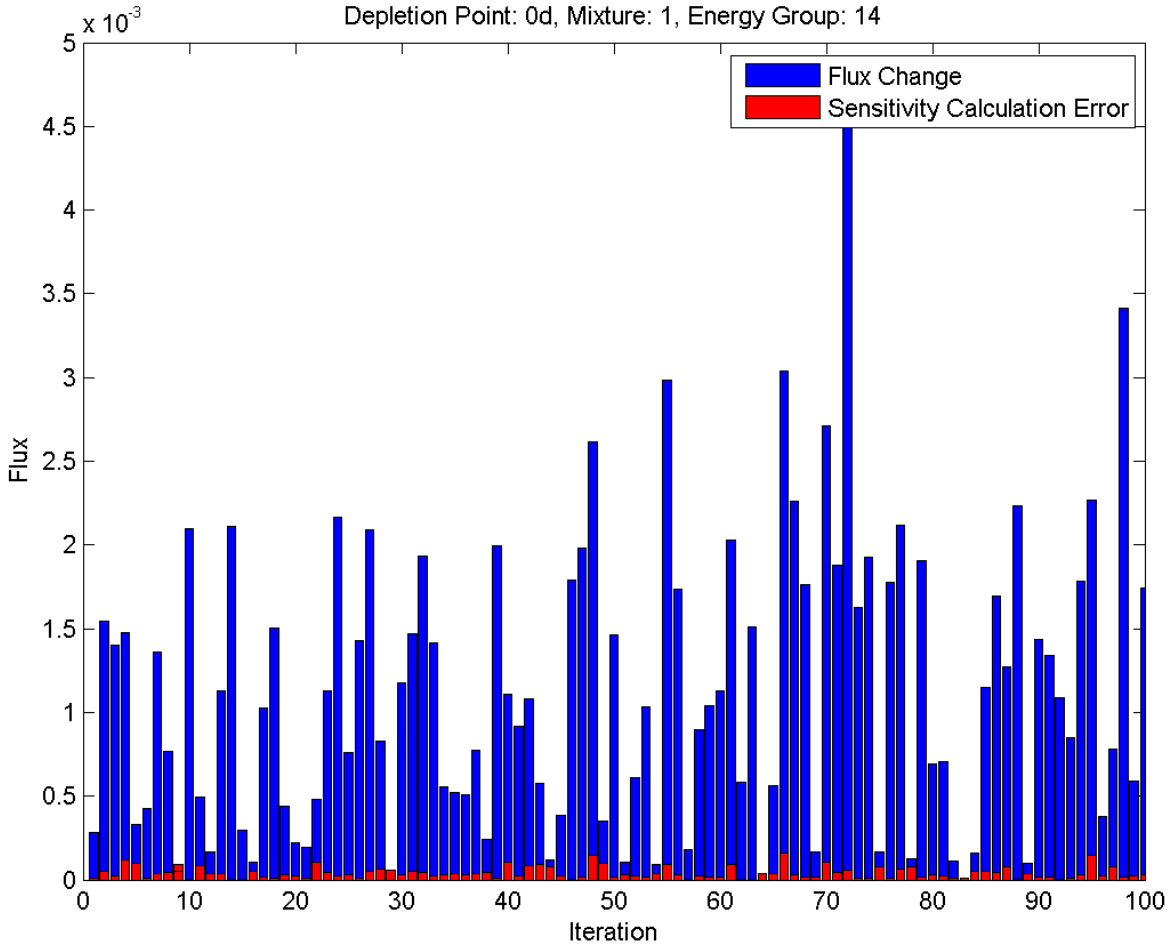

Figure 80: GPT-Free Numerical Demo, BOC, Mixture 1, Group 14, 100 Samples with 10% RMS XS Perturbations

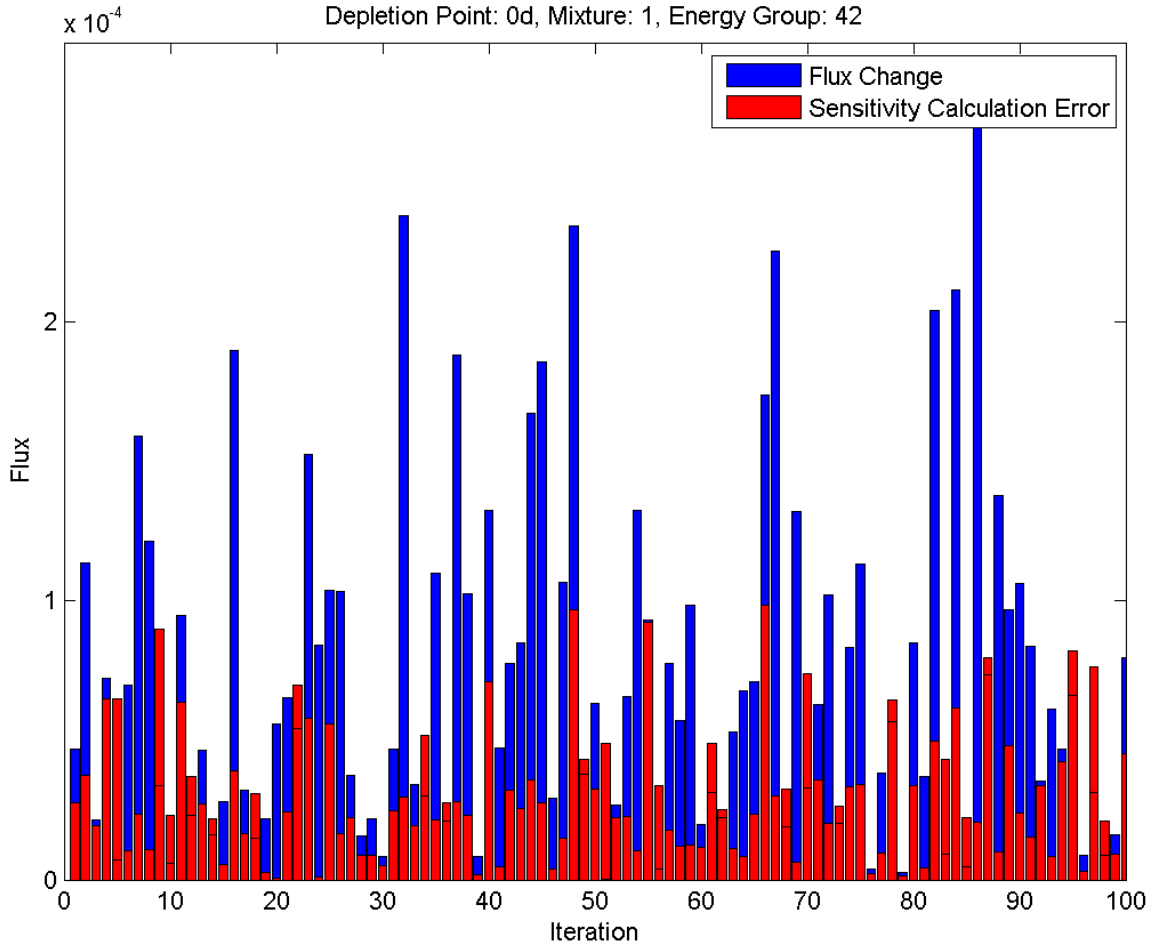

Figure 81: GPT-Free Numerical Demo, BOC, Mixture 1, Group 42, 100 Samples with 10% RMS XS Perturbations

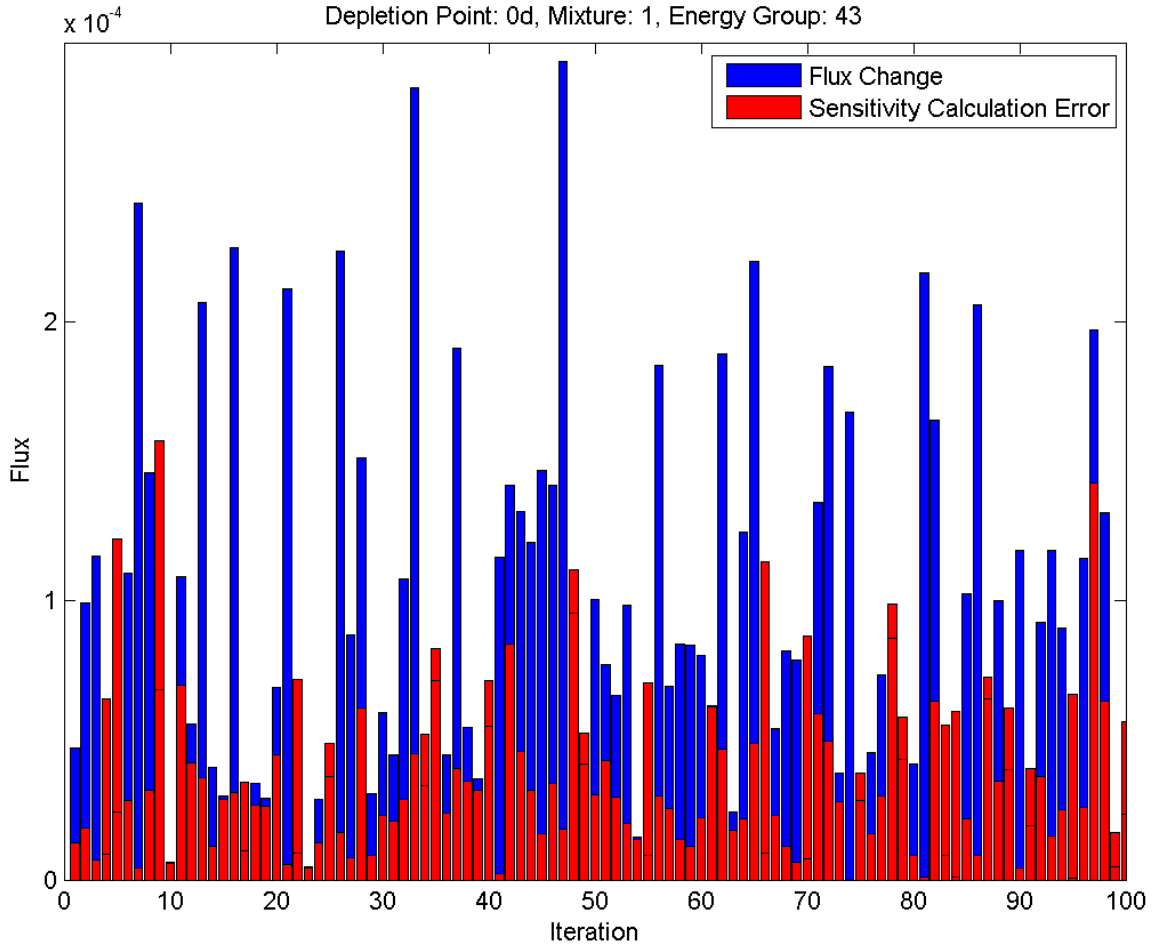

Figure 82: GPT-Free Numerical Demo, BOC, Mixture 1, Group 43, 100 Samples with 10% RMS XS Perturbations

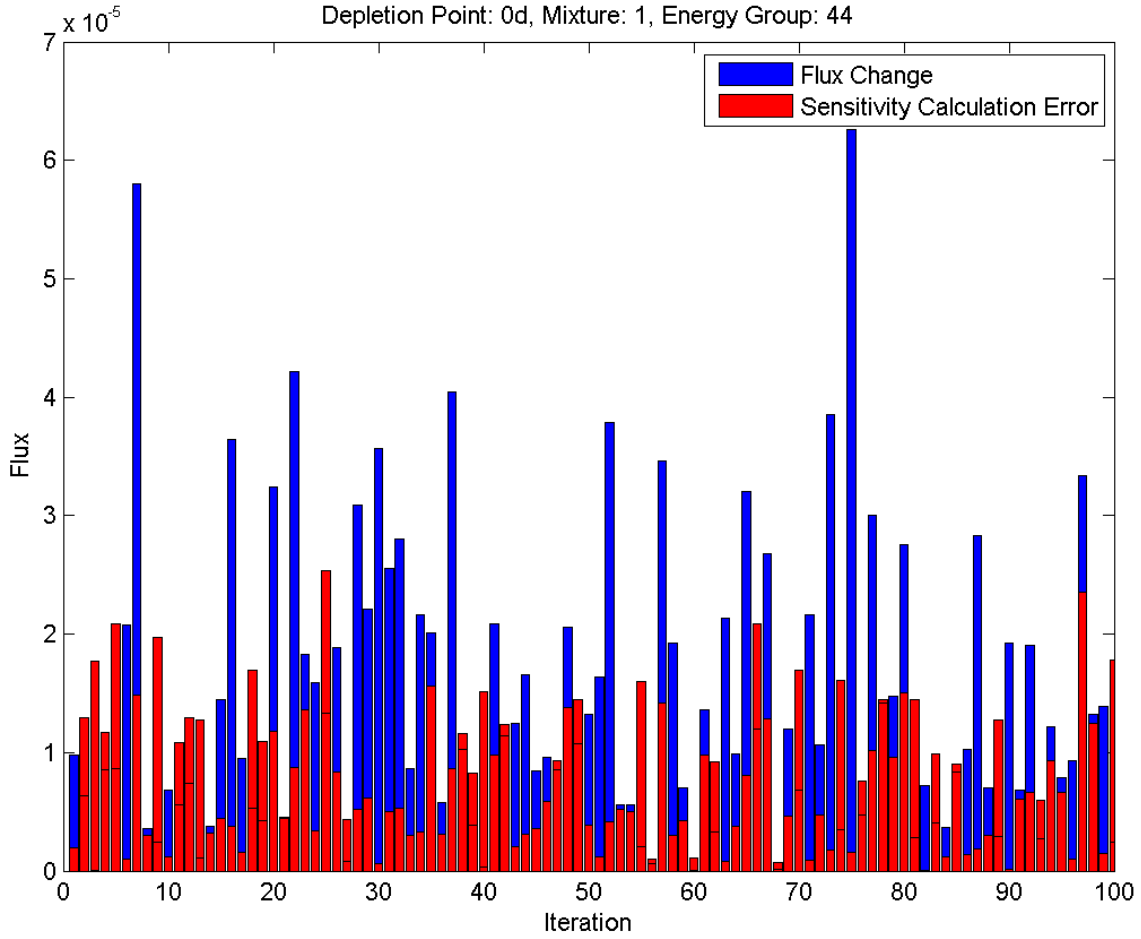

Figure 83: GPT-Free Numerical Demo, BOC, Mixture 1, Group 1, 100 Samples with 10% RMS XS Perturbations

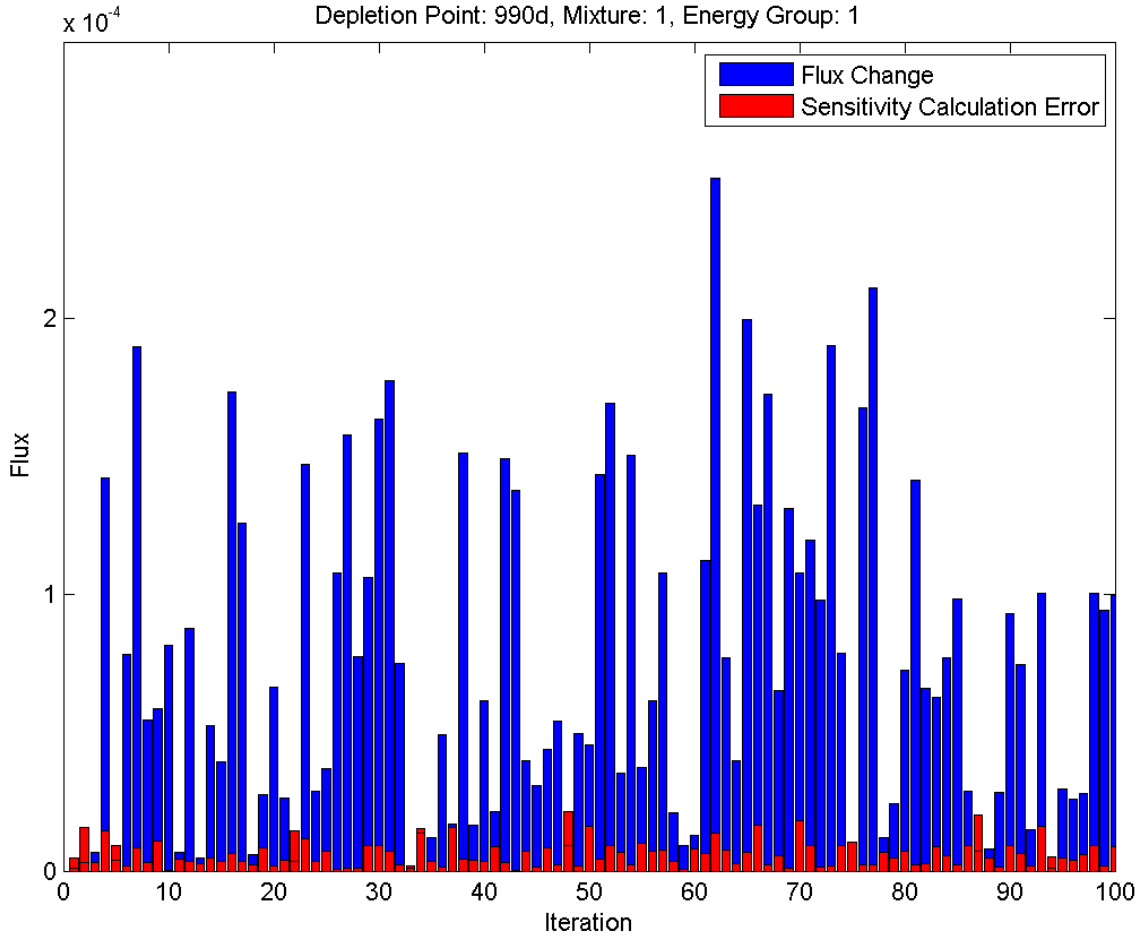

Figure 84: GPT-Free Numerical Demo, EOC, Mixture 1, Group 1, 100 Samples with 10% RMS XS Perturbations

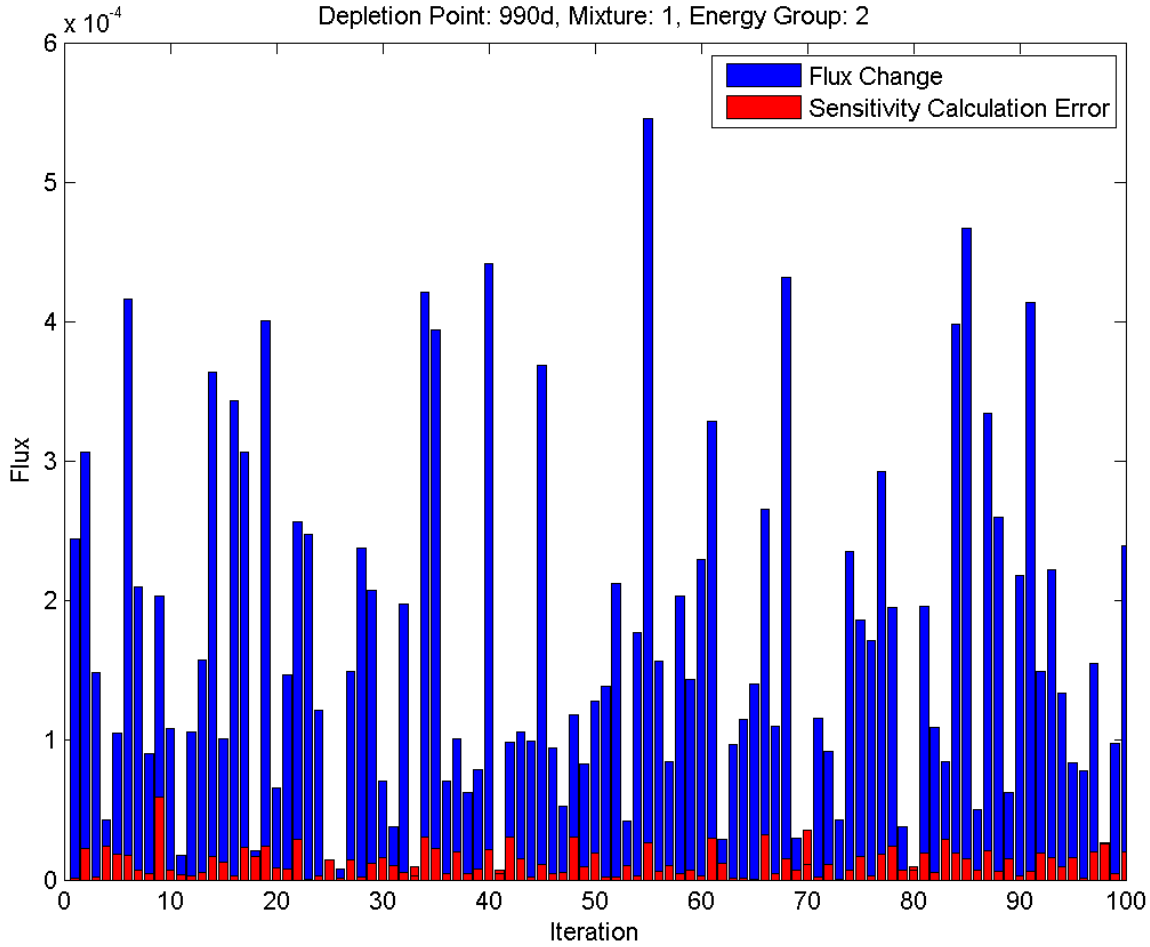

Figure 85: GPT-Free Numerical Demo, EOC, Mixture 1, Group 2, 100 Samples with 10% RMS XS Perturbations

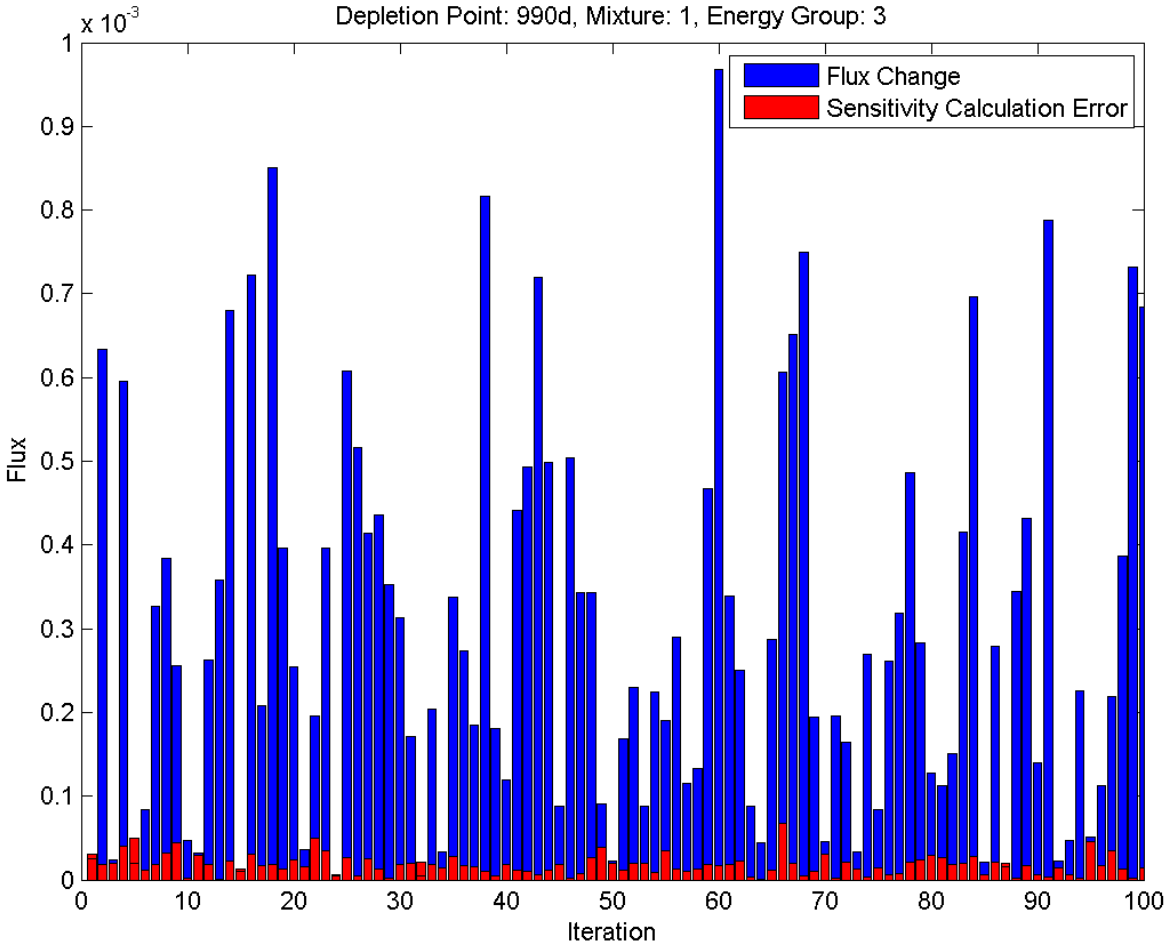

Figure 86: GPT-Free Numerical Demo, EOC, Mixture 1, Group 3, 100 Samples with 10% RMS XS Perturbations

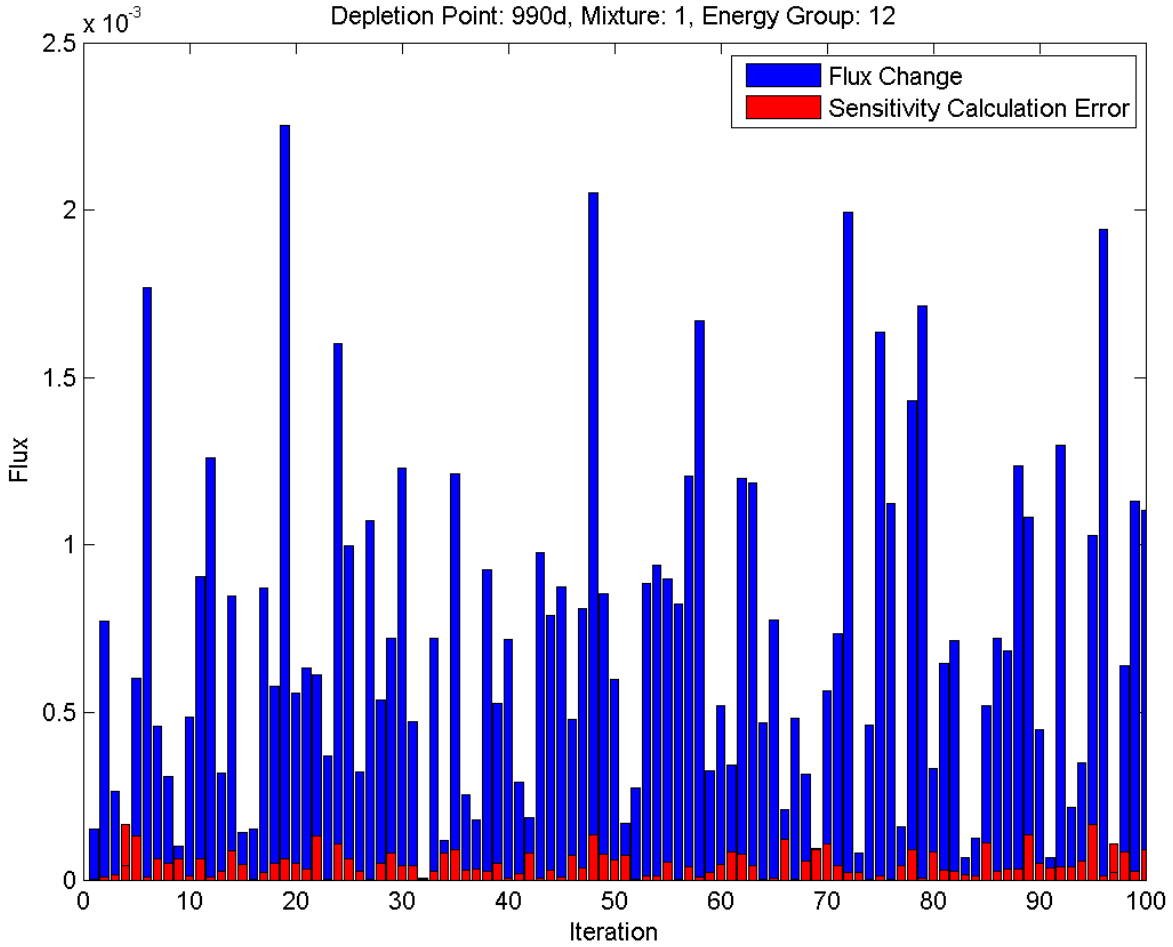

Figure 87: GPT-Free Numerical Demo, EOC, Mixture 1, Group 12, 100 Samples with 10% RMS XS Perturbations

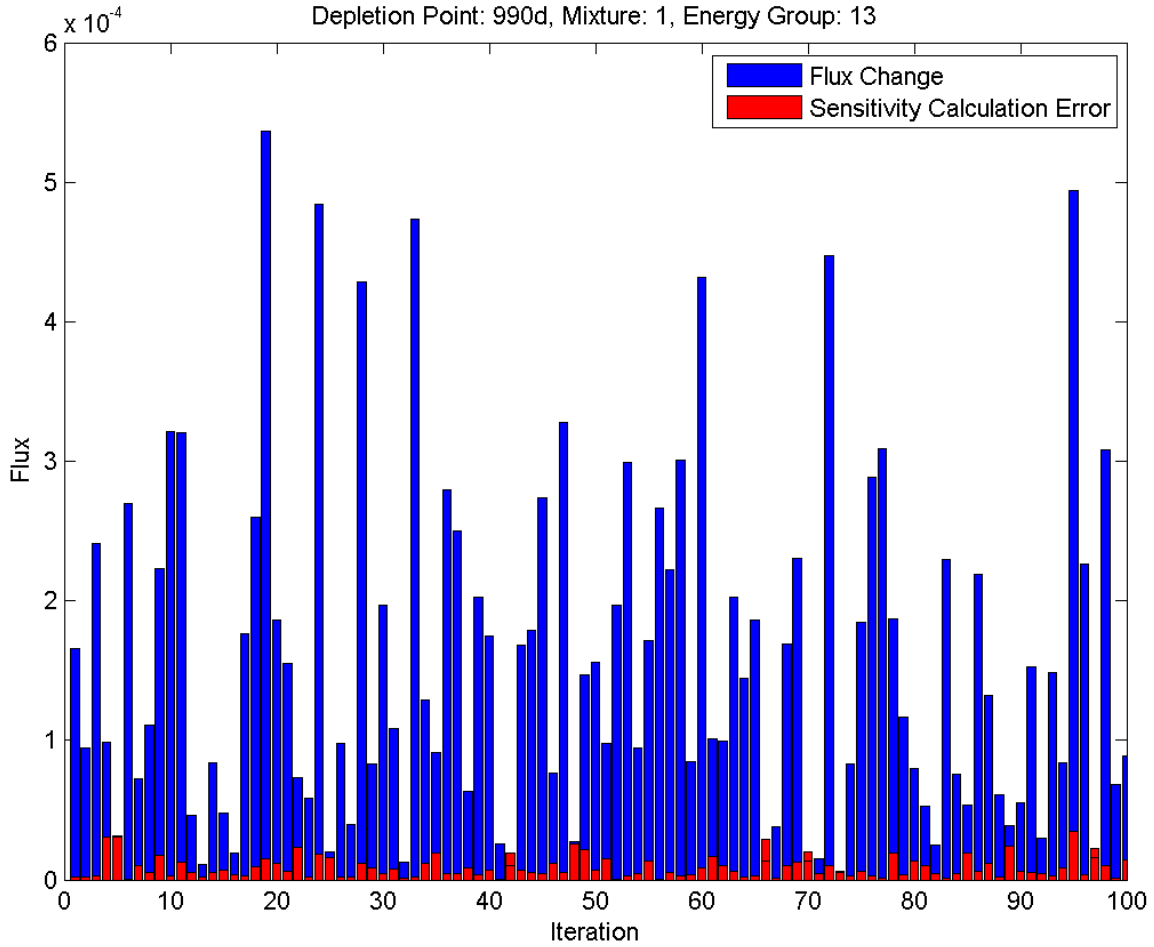

Figure 88: GPT-Free Numerical Demo, EOC, Mixture 1, Group 13, 100 Samples with 10% RMS XS Perturbations

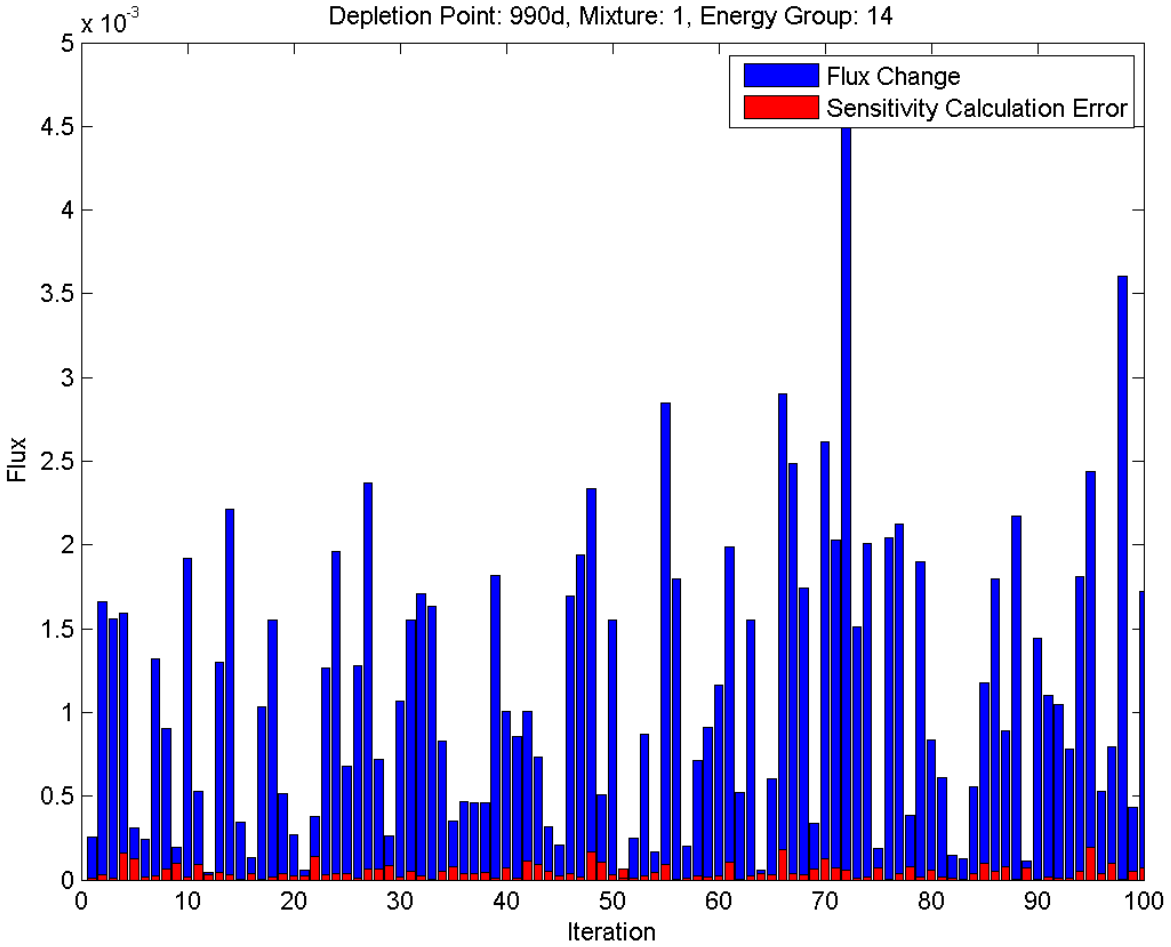

Figure 89: GPT-Free Numerical Demo, EOC, Mixture 1, Group 14, 100 Samples with 10% RMS XS Perturbations

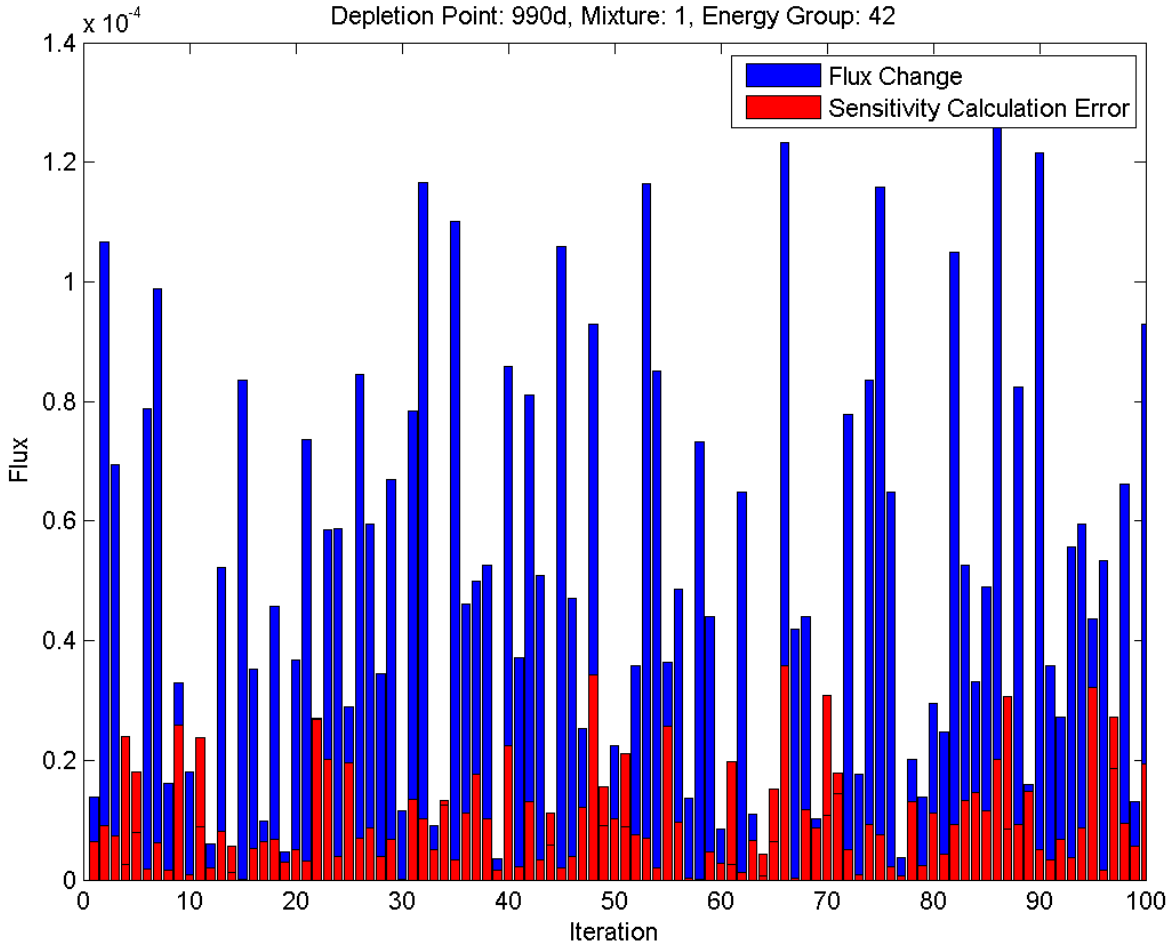

Figure 90: GPT-Free Numerical Demo, EOC, Mixture 1, Group 42, 100 Samples with 10% RMS XS Perturbations

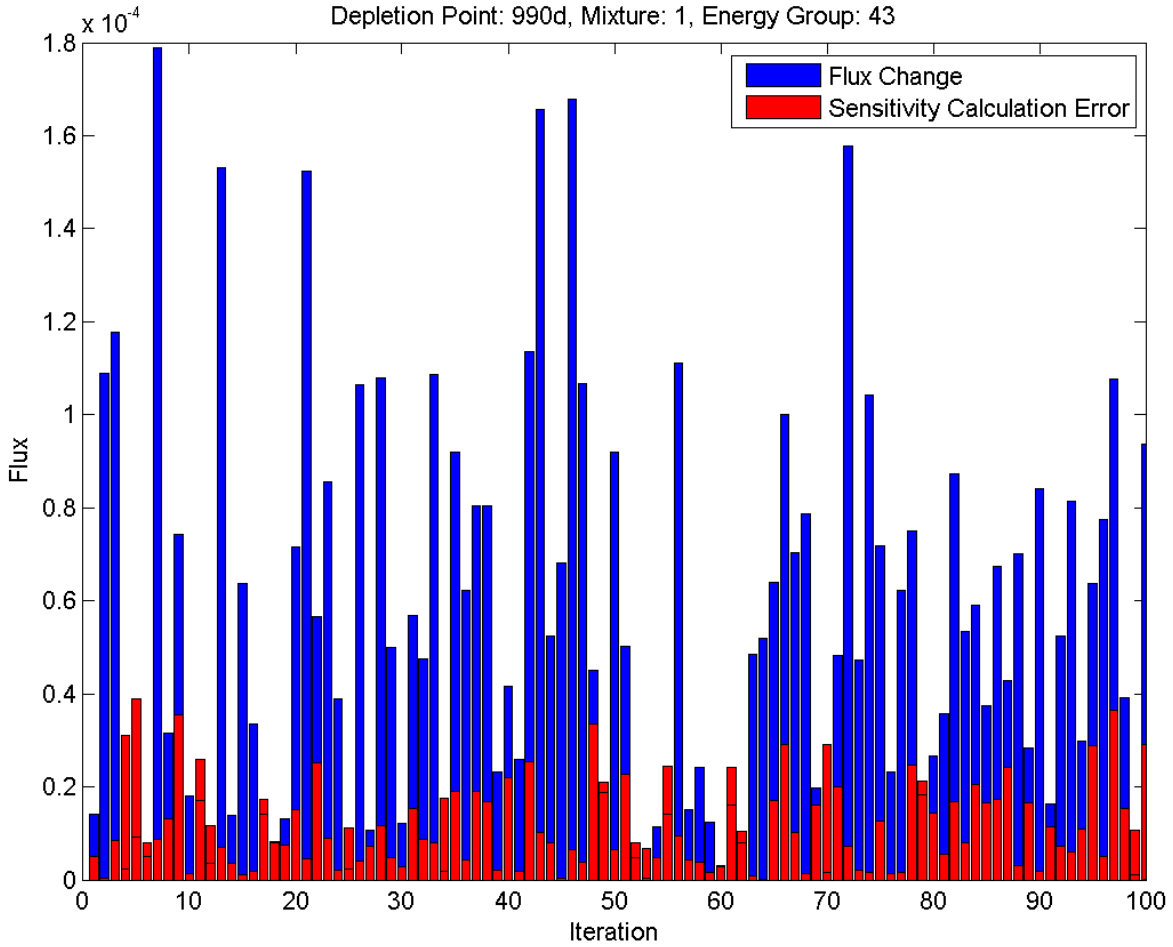

Figure 91: GPT-Free Numerical Demo, EOC, Mixture 1, Group 43, 100 Samples with 10% RMS XS Perturbations

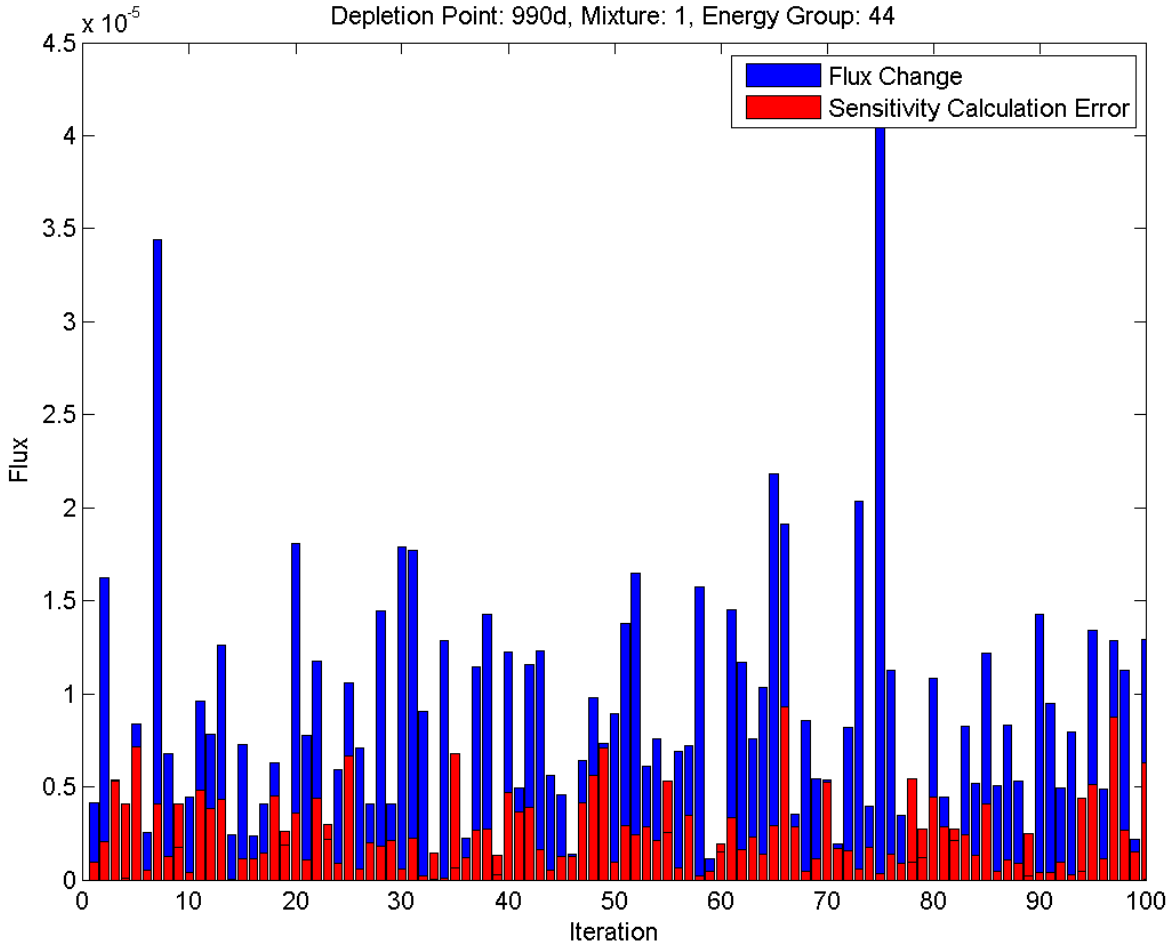

Figure 92: GPT-Free Numerical Demo, EOC, Mixture 1, Group 44, 100 Samples with 10% RMS XS Perturbations

Part of the nature of the error comes from the numerical contamination due to the limited accuracy of the numerical calculation. In effect, the least-squares operation is attempting to solve for the sensitivities for the thermal energy groups, which have less than two digits of precision using a numerical tolerance of 10^{-6}. This can be addressed partially by increasing the numerical tolerance which will reduce the numerical error in computing the least squares solve. Furthermore, if the least square solve is completed in the non-linear range, the sensitivities will also be prone to errors. While the result may be linear about some level of perturbations, this cannot be guaranteed for all possible linear combinations. One method to address this is to use finite-difference to calculate derivatives for each direction in the ROM, **Q**, represented by a column vector q_i.

Consider the Gateaux derivative for direction, q_i , scaled by α_i:

$$Df\left(\sigma_0\right) \approx \frac{f\left(\sigma_0\left(1+\alpha_i q_i\right)\right) - f\left(\sigma_0\right)}{\alpha_i} \tag{106}$$

The error between the finite difference derivative and the actual derivative is (Kelley 1999):

$$Df\left(\sigma_0\right) - f'\left(\sigma_0\right) = O\left(\alpha_i + \frac{\epsilon}{\alpha_i}\right) \tag{107}$$

Where ϵ is the machine precision, the error in equation 115 is minimized by selecting $\alpha_i = \sqrt{\epsilon}$.

$$Df\left(\sigma_0\right) - f'\left(\sigma_0\right) = O\left(2\sqrt{\epsilon}\right) \quad \text{and} \quad \alpha_i = \sqrt{\epsilon} \tag{108}$$

Equation 116 is valid for general calculations; however, a ROM input basis determined by an adaptive method requires additional consideration. Using the κ-metric, consider the relative importance for each ROM input direction:

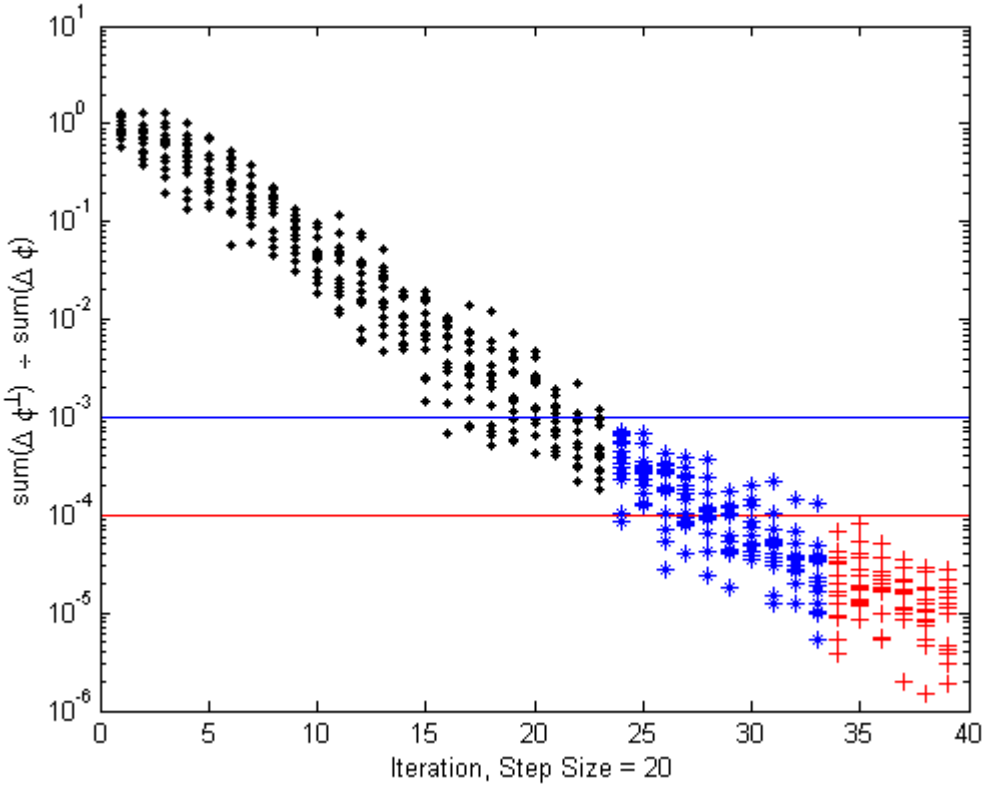

Figure 93: Flux relative reduction error κ-metric for the UAM assembly.

Each input direction is less important than the first, decreasing in a logarithmic fashion until numerical tolerance, 10^{-6} is reached. Restated, a perturbation using q_1 will yield a change in approximately 8000pcm; however, a perturbation using q_{660} will only change the model by approximately 0.25pcm at the final time-point. The benefit of an adaptive ROM is therefore

its own enemy when it comes to finite-differencing for a sensitivity analysis. Consider the following weighting scheme for the ROM considered in Figure 93:

$$Df\left(\sigma_0\right) \approx \frac{f\left(\sigma_0\left(1+\beta_i\alpha_i q_i\right)\right) - f\left(\sigma_0\right)}{\beta_i\alpha_i} \tag{109}$$

Where:

$$\beta_i = \frac{1}{\max\left(\kappa_i\left\{\phi\right\}_{j=1}^{N}\right)} \tag{110}$$

Equation 118 represents the relative scaling to bring direction *i* to the same magnitude as the first direction. Consider the last direction of importance which is of the magnitude of numerical tolerance. A perturbation along this direction would be smaller than the first direction by a factor of $1/\beta$. Without scaling up the perturbation, the model will not change within machine precision. The result is to select α_i such that the following holds:

$$\alpha_i = \min\left(\beta\sqrt{\epsilon}, 1\right) \tag{111}$$

The constraint of 1 is required such that cross-section perturbations do not become negative when introduced into the simulation. The result applied to Figure 91 yields a set of weighting parameters given a SCALE convergence criterion of 10^{-6} as shown in Figure 94. Unfortunately, this approach is further complicated by two factors. First, SCALE outputs only the first 6 digits for any flux response, with effective changes occurring in the last 3-4 output digits. With a tolerance at 10^{-6}, the calculation precision does not offer sufficient information for differencing. Second, even if the numerical tolerance is increased further, the

edited flux output will not be able to represent the code calculated changes. These two factors cause problems because the calculated sensitivities from Equation 117 are of the same order or smaller than the error predicted by Equation 116. Simply put, finite-differencing along the ROM directions using SCALE text-edited output does not compute reliable numerical derivatives given single-precision output. A sample finite-difference calculation using a ROM with dimension 660 demonstrates this problem for a particular flux-response in Figure 95. Most neutronics codes (NEWT included) do calculate the angular flux using double precision accuracy. By extracting this binary information, it is possible to mitigate this problem by increasing the numerical tolerance of the software such that sensitivities can be reliably calculated.

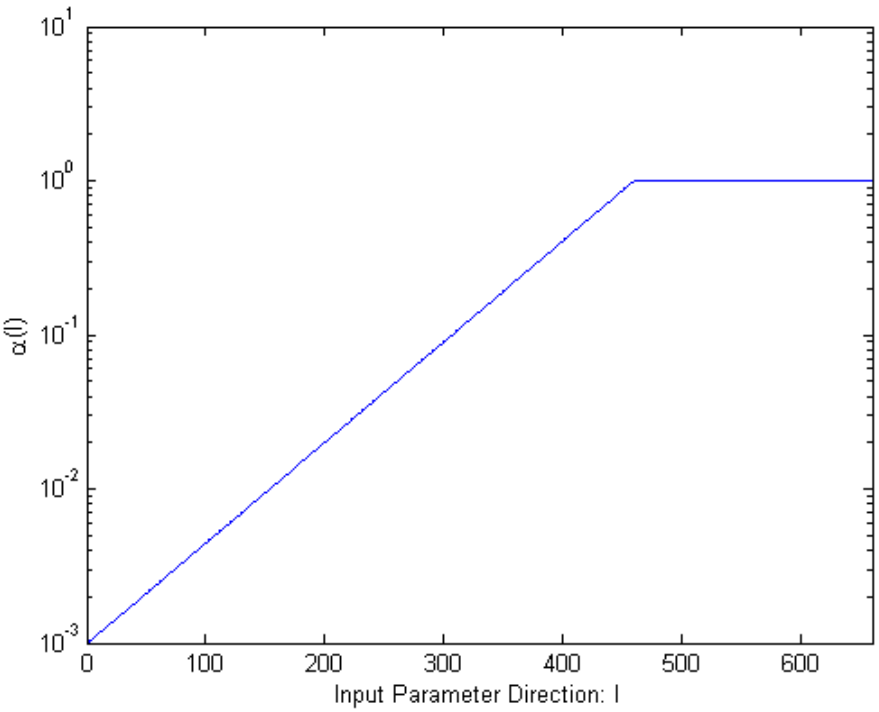

Figure 94: Recommended weighting parameters for finite-differencing

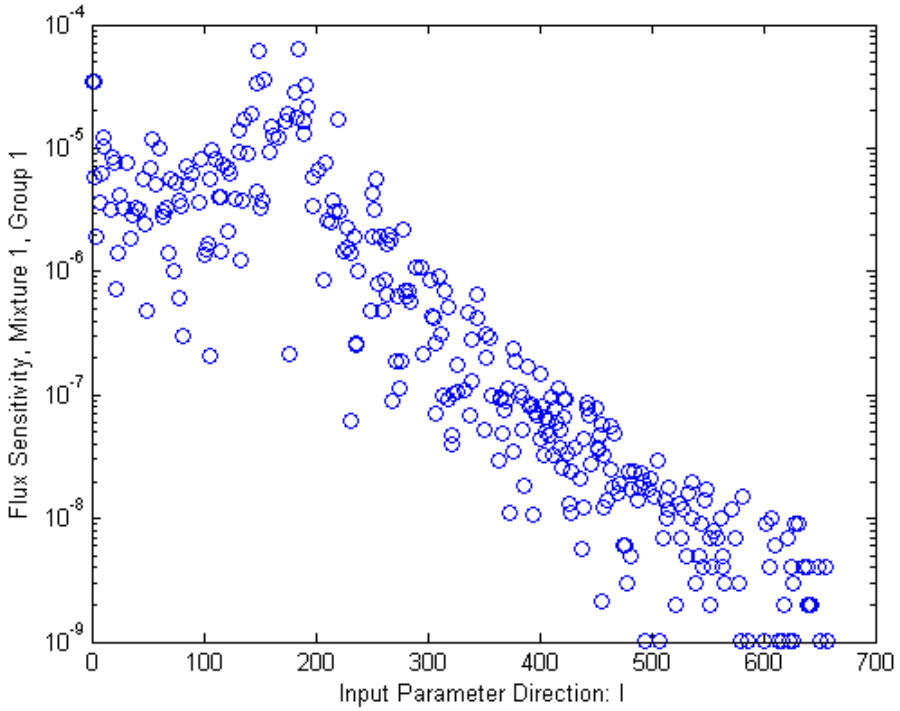

Figure 95: Sample sensitivity profile for flux, group 1, BOC, mixture 1.

To improve the quality of results affecting numerical limitations, two changes were made. First, a software tool was designed (See Appendix) to extract the binary flux data from the SCALE output files. Second, the numerical tolerance was increased from 10^{-6} to 10^{-9}, the highest precision that would still reliably converge to a solution. Using this improved accuracy significantly reduced the level of error in calculations. A sample of output using an ROM with dimension 270 after 3 years of burn-up is shown in Figure 96. Even though the error can be reduced beyond a rank of 270, the predictive error does not increase due to limitations of the sensitivity analysis as shown in Figure 97.

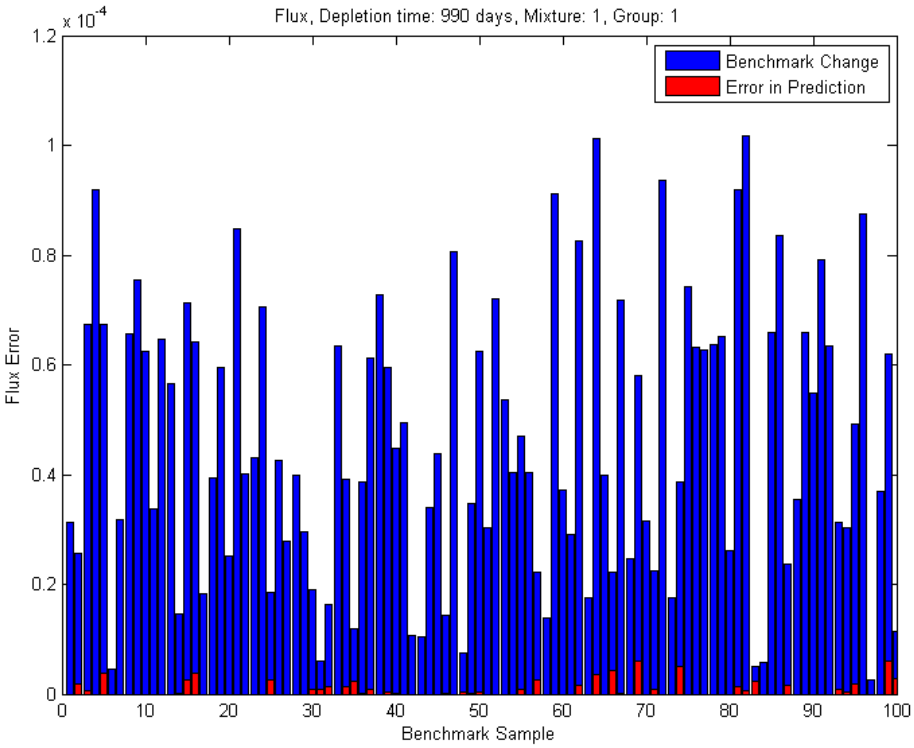

Figure 96: Error in first-order GPT-Free sensitivity fast flux prediction generated with r = 270.

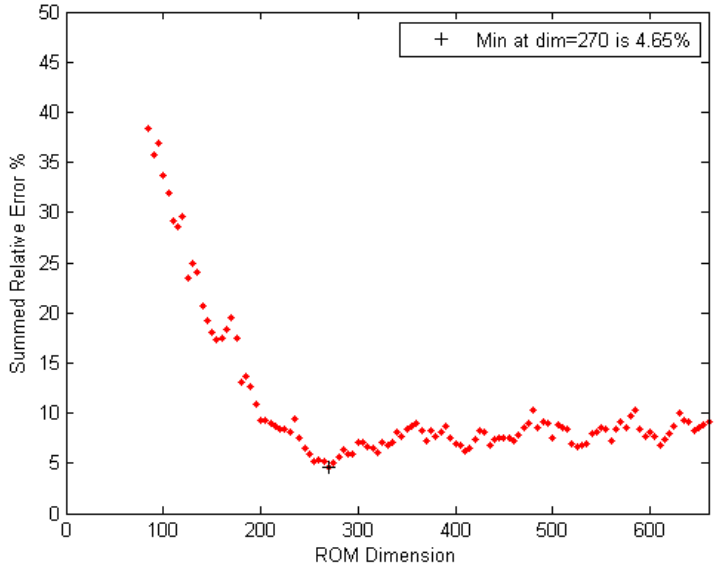

Figure 97: GPT-Free summed relative flux prediction error using 100 sample benchmarks.

Using the higher precision simulation, GPT-Free sensitivities were compared to a test of 132 forward finite differences for fission and capture cross-sections of both ^{235}U and ^{239}Pu. As mentioned in section 4.C.2, the numerical errors introduced by sensitivities near numerical tolerance place a limit on the accuracy of sensitivities determined via a forward sensitivity analysis; however, the largest sensitivities are determined as expected. A comparison between the forward finite-difference and the GPT-Free sensitivities is shown in Figure 98 at the beginning of cycle.

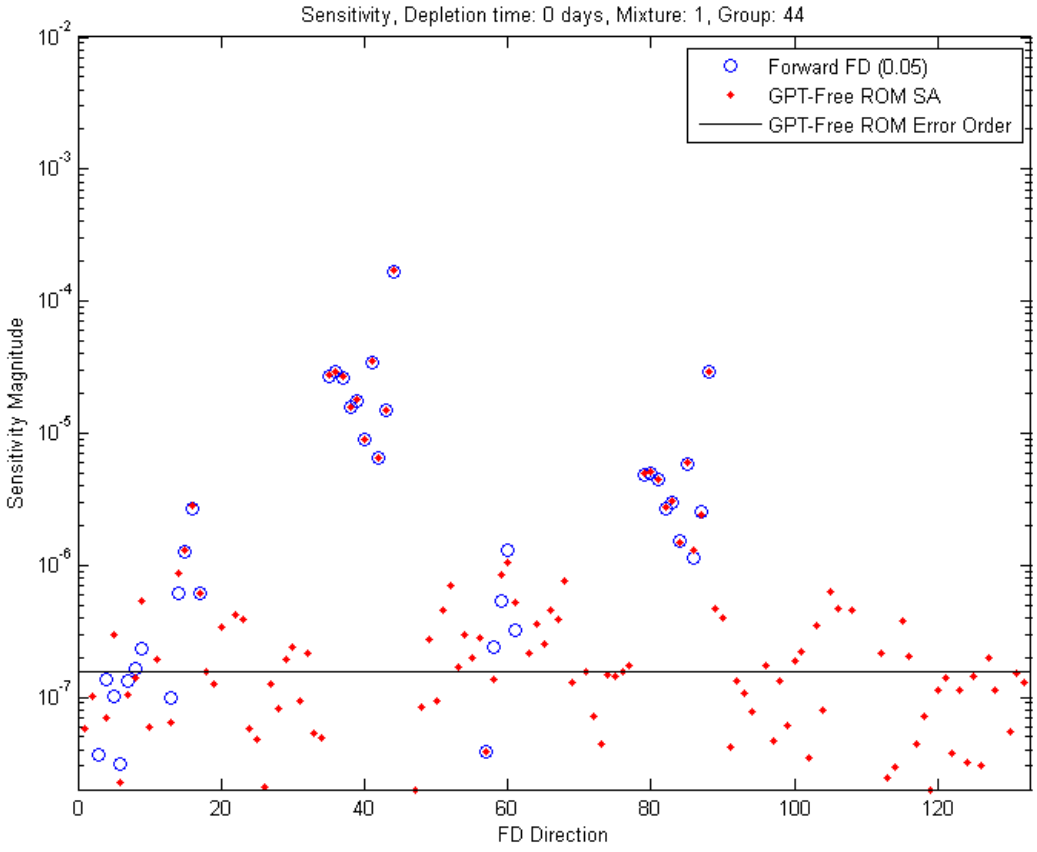

Figure 98: GPT-Free SA compared to finite difference for ^{235}U and ^{239}Pu fission and capture cross-sections, BOC.

A comparison of flux error using sensitivities generated from the GPT-Free sensitivity analysis to predict the change from an increment of the thermal ^{235}U fission cross-section is shown in Figure 98. On average the typical error from sensitivities is approximately six orders of magnitude smaller than the mixture and energy group specific flux. The figure shows the result from a test after a half year of burn-up.

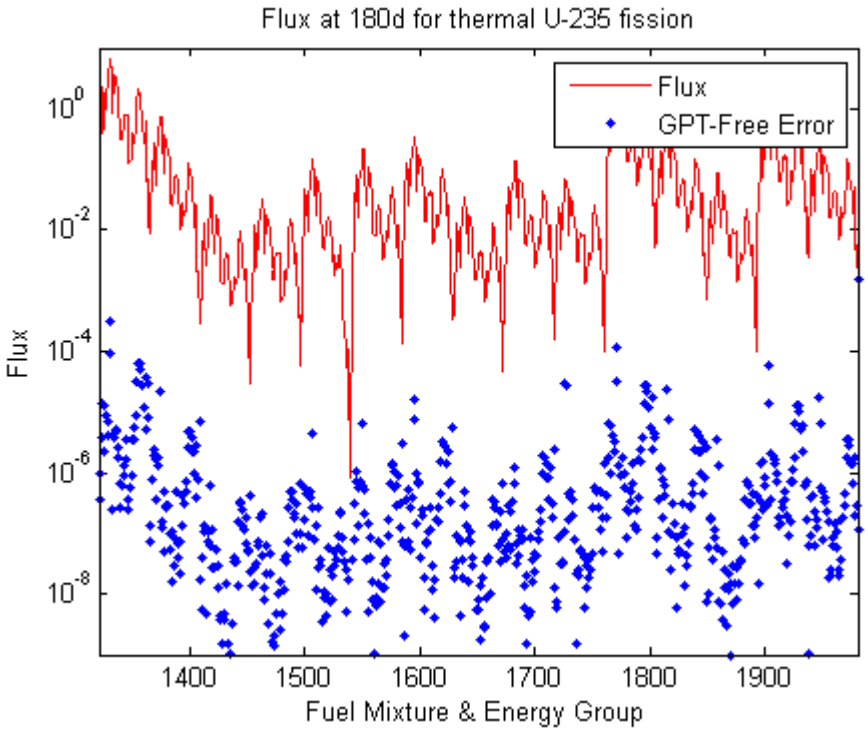

Figure 99: GPT-Free SA prediction of flux from a 5% increment of ^{235}U thermal fission cross-section after 180days of BU at 21.2 MWth/MTHM.

One final note of the depletion GPT-Free sensitivity model is the determination of resonance cross-sections. In particular, SCALE6 overwrites infinite dilution cross-sections. Even if cross-sections are in injected into the master cross-section library, the software tools will overwrite these values using continuous energy cross-sections. This causes the ^{238}U resonances to occasionally have errors due to the inability of the sensitivities to correct for these cross-sections. For future work, the depletion process needs to be separated from TRITON so that the working cross-section library, which is overwritten at each depletion point, can be used to update cross-sections.

6.C Depleted TMI Mini-Core Multi-Assembly Model

The TMI Mini-core model is a thermal PWR 45x45 reflected pin model simulating nine 15x15 assemblies with burnable poison, control rod channels, and fuel loading with pin-level detail (e.g. no homogenization). All pin-details such as gap, and clad are included with 100 mesh cells per pin (as compared to 16 for the UAM LWR model). The model, shown in Figs. 100 and 101, was derived from the TMI-1 UAM model specifications (Ivanov 1999). To build a complete inventory of isotopes, the model was depleted to 380 days at 21.220 MW/MTHM specific power using TRITON. All calculations are completed using the SCALE 44 energy group cross-section library with a numerical tolerance set to 10^{-9} for all transport calculations in TSUNAMI-2D and NEWT. Gadolinium pins were depleted by flux as opposed to power depletion per SCALE manual recommendations. The complete list of actinides is in Table 8, and all other isotopes modeled are included in Table 9.

Table 8: UO_2 Actinide Composition at 380 days.

Actinide	UO_2 SCALE Input Composition
u-235	9.14178E-04
u-238	2.17182E-02
u-234	1.26408E-08
u-236	3.96340E-05
np-237	1.17957E-06
pu-238	6.97396E-08
pu-239	6.36204E-05
pu-240	6.38089E-06
pu-241	1.81644E-06
pu-242	8.47834E-08
am-241	2.40074E-08
am-243	2.98918E-09
cm-242	1.39389E-09
cm-243	6.25819E-12
cm-244	1.26297E-10

Table 9: All remaining isotopes in UO_2 fuel pins at 380 days.

o-16	eu-151	sn-126	mo-97
h-1	eu-153	xe-131	mo-98
gd-152	eu-154	cs-134	ba-140
gd-154	eu-155	cs-135	i-129
gd-155	b-10	cs-137	pd-105
gd-156	b-11	ce-144	zr-95
gd-157	n-14	pr-143	pd-107
gd-158	kr-83	nd-143	pd-108
gd-160	i-135	nd-145	pr-141
in-115	pm-148	nd-146	eu-156
ag-109	xe-133	nd-147	ru-102
xe-135	cd-113	nd-148	nd-144
cs-133	ce-141	pm-147	zr-96
zr-94	ru-101	pm-149	i-127
nb-93	sm-153	sm-147	zr-91
mo-95	ru-103	sm-149	mo-100
tc-99	la-139	sm-150	ru-104
ru-106	ce-143	sm-151	ce-142
rh-103	zr-93	sm-152	nb-95
rh-105	mo-99		

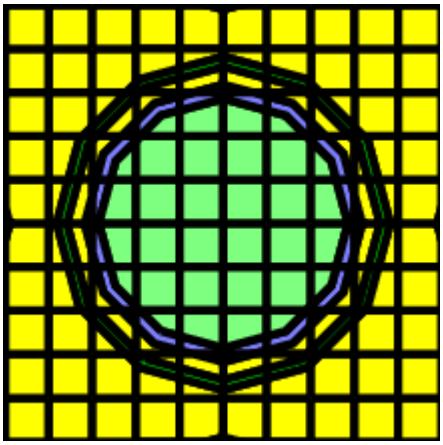

Figure 100: A single pin in the TMI mini-core model.

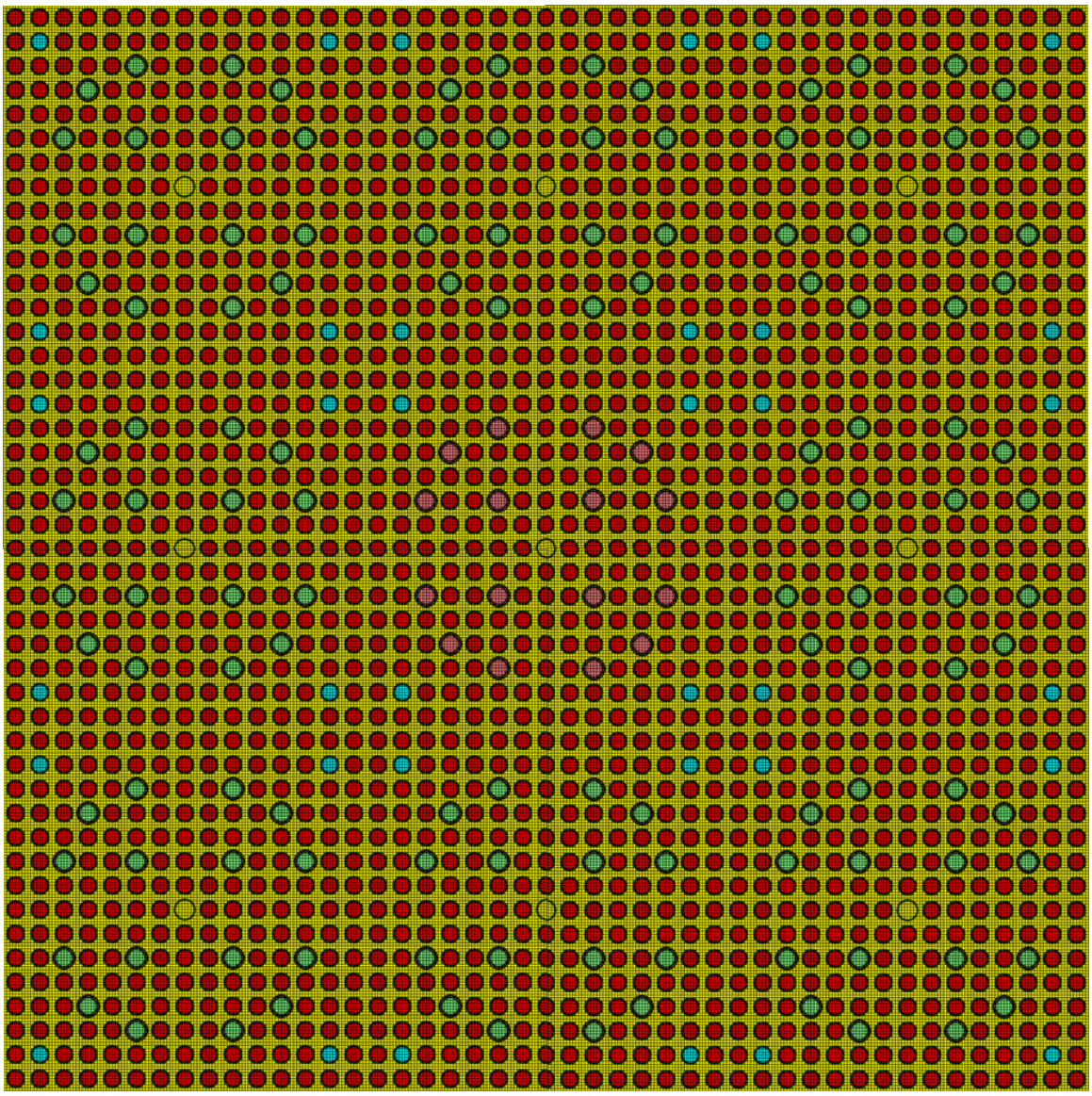

Figure 101: TMI Mini-core model, 45x45 with 10x10 mesh cells per pin.

The reference flux in the fuel, both UO_2 and Gd-UO_2, are plotted with the reference convergence criteria in Figure 102. Because of the large number of fuel regions and mixtures, the κ-metric is examined integrally for all fuel mixtures. In some cases, the burnable poison regions are extracted to illustrate the difference in the ROM reduction error for the two fuels.

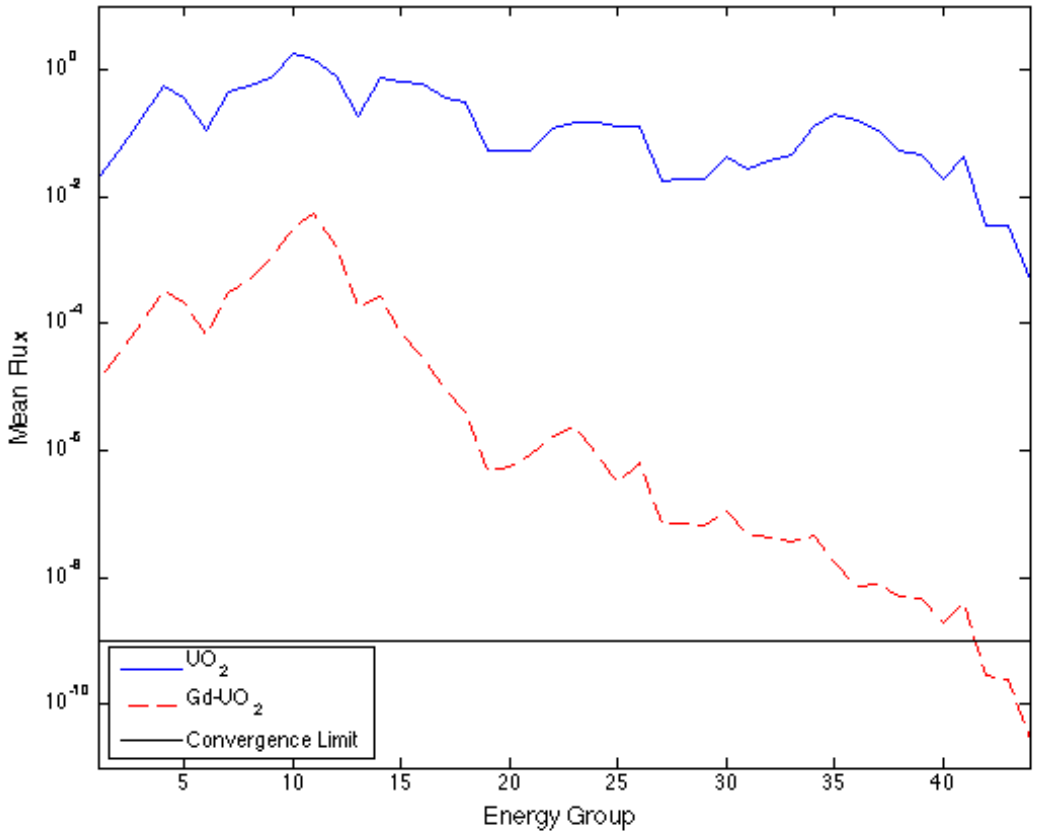

Figure 102: Mean Fuel Flux, TMI Mini-core model at 380 days at 21.220 MW/MTHM.

Even for a model with numerous cross-sections and flux-regions, the GPT-Free algorithm automatically selects input parameters that are important to the model using k-eigenvalue sensitivity profiles. Figure 103 shows the effect of filtering on the input parameter space. There are 2640 cross-section input parameters which have been perturbed at 5% RMS independently by isotope, energy group, reaction, and mixture (shown in blue). The red overlay shows the effect of the ROM determining the important cross-sections. Because the GPT-Free method filters out the unimportant directions, the remaining inputs, visualized as

blue in Figure 103, are the dominant or model dependent input parameters. By using the GPT-Free algorithm, a sensitivity analysis can be performed on the reduced input parameter space described by the blue region in Figure 103.

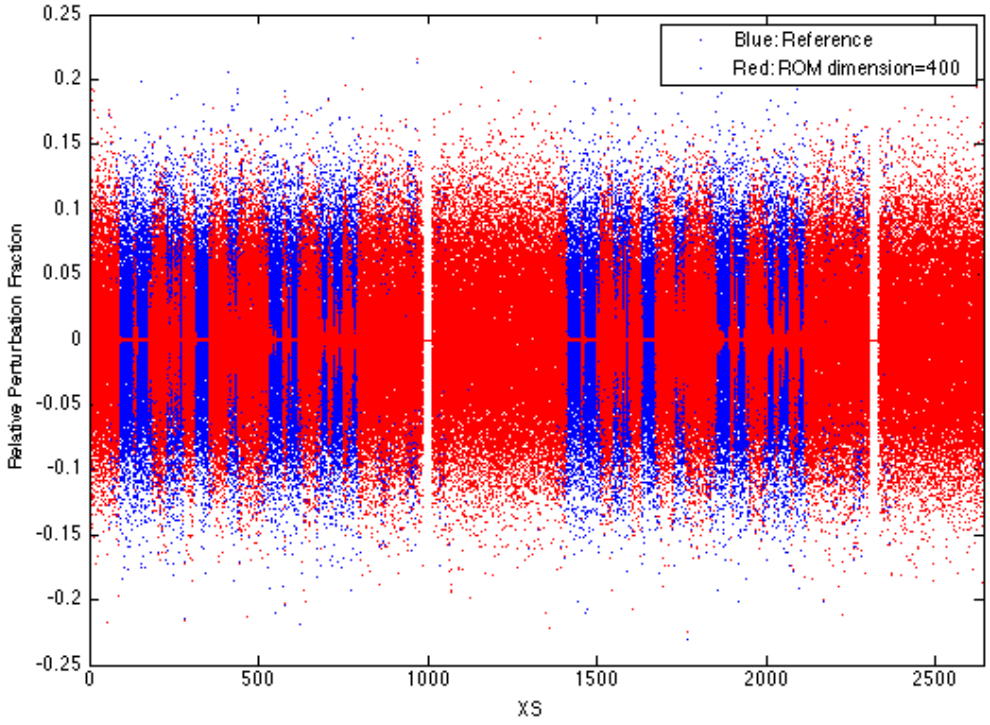

Figure 103: GPT-Free algorithm automatically selecting the important cross-section regions. Regions where the red is still spanning the full range of cross-section perturbations are cross-sections that have less impact on the model physics. Regions where the red is subtracted to 0 are regions that are highly important for the model. Cross-section perturbations mapped by this figure were generated by randomly sampling a 5% RMS Gaussian distribution.

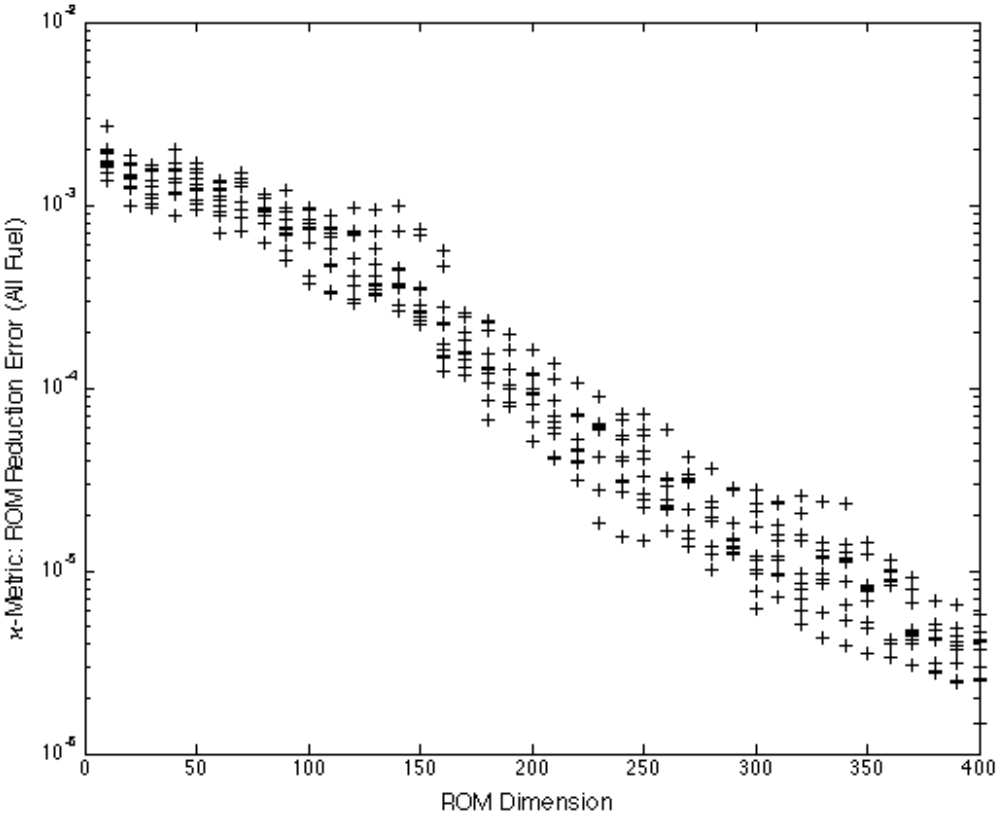

Figure 104: κ-Metric for all fuel fluxes *vs.* increasing ROM dimension.

Figure 104 is the κ-Metric for the UO_2 fuel regions showing a logarithmic decrease in ROM reduction error with increasing ROM dimension. Figures 105 and 106 show the similar result for the Gd-UO_2 fuel region and the total system (e.g. mixture independent) flux. Figure 107 compares data from Figs. 104 and 105 on the same chart. Equation 120 is the specific κ-Metric used for the set of figures and for ROM determination.

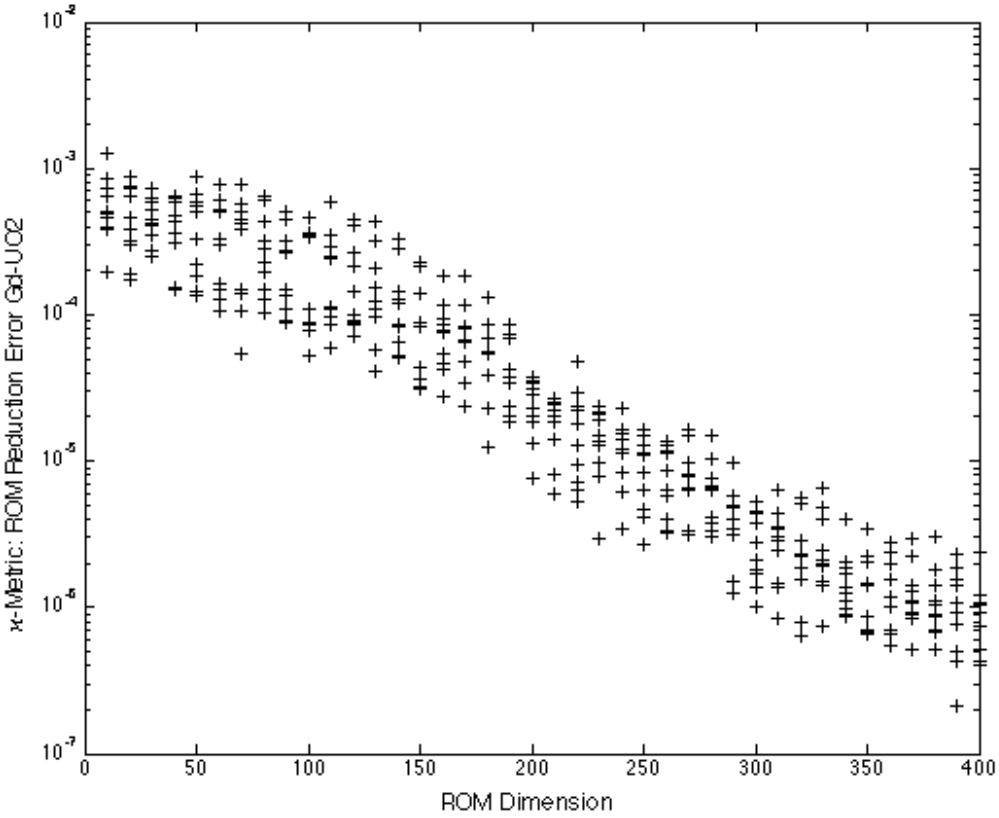

Figure 105: κ-Metric for all Gd-UO_2 fuel fluxes *vs.* increasing ROM dimension.

Equation 120 measures the reduction error from truncating the input parameter space to the specified dimension of ROM. As the ROM increases in dimension, more input parameter directions are included, reducing the error in the numerator term.

$$\kappa = \frac{\sum_{g}^{G}\sum_{x}\left|\phi^{\perp} - \phi_0\right|}{\sum_{g}^{G}\sum_{x}\left|\phi_0\right|} \tag{111}$$

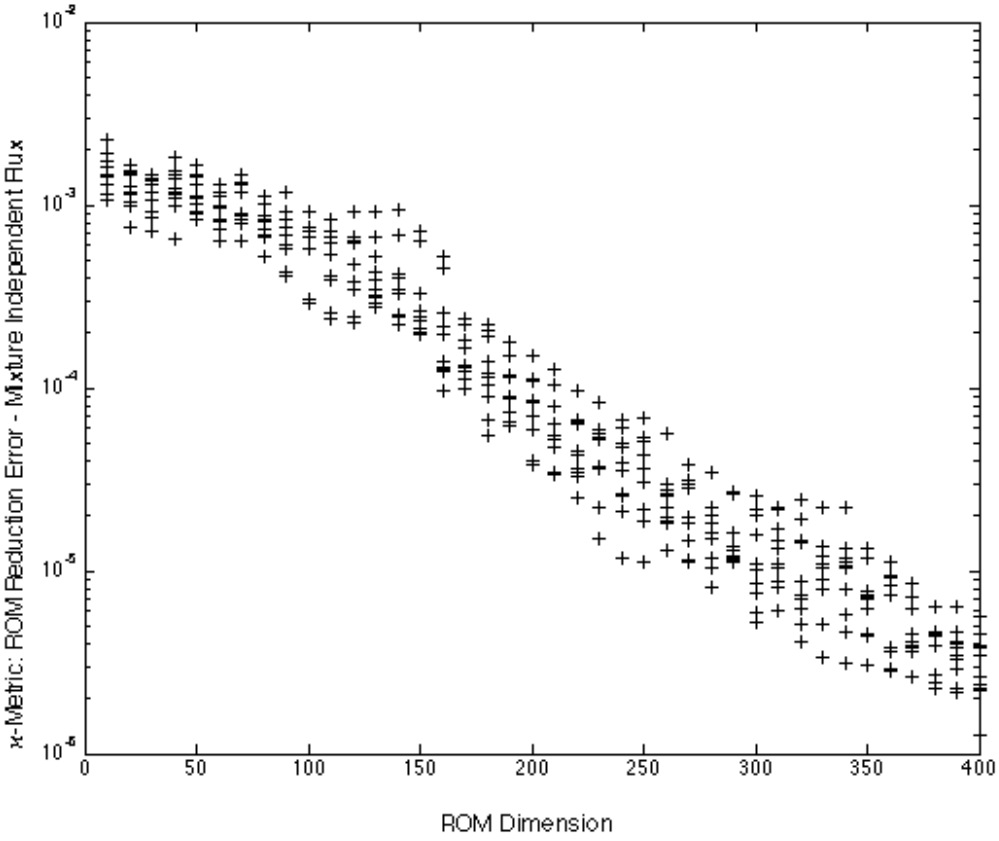

Figure 106: κ-Metric for the mixture independent flux *vs.* increasing ROM dimension.

Specific to Figure 106, the overall mixture flux is limited by the fuel flux because numerical convergence is limited to the UO_2 fuel pin convergence. Typically, the coolant and the Gd-UO_2 pins consistently converge more quickly. This makes sense because the non-burnable poison laced UO_2 fuel contributes the most to the overall power of the reactor system.

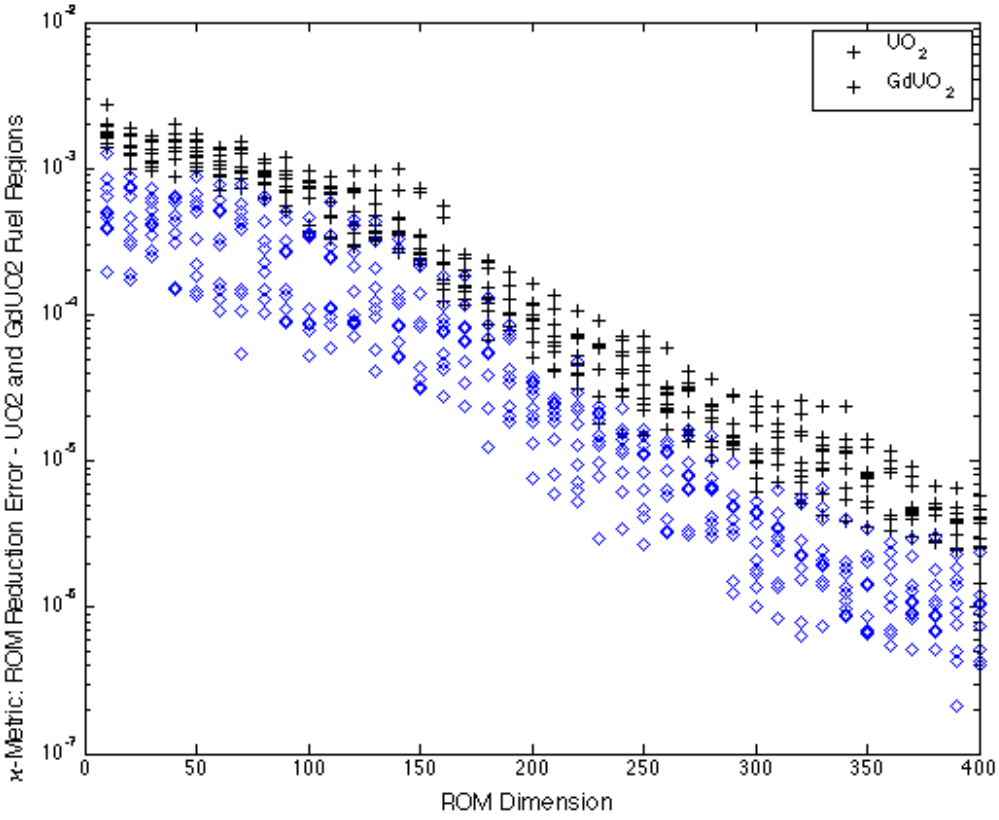

Figure 107: κ-Metric of both fuel types (UO_2 and $GdUO_2$) *vs.* increasing ROM dimension.

The ROM reduction error for fuel pins (UO_2 and Gd-UO_2) is demonstrated in Figure 108 by showing the energy dependent fuel flux and corresponding ROM error for variable ROM dimensions. The green bar indicates the numerical stopping criteria set in NEWT. As previously mentioned, the stopping criterion is limited by convergence in the UO_2 fuel pins; however, the flux elsewhere is fairly well resolved. Results below the convergence criteria, even if they appear feasible, must be questioned and used with caution.

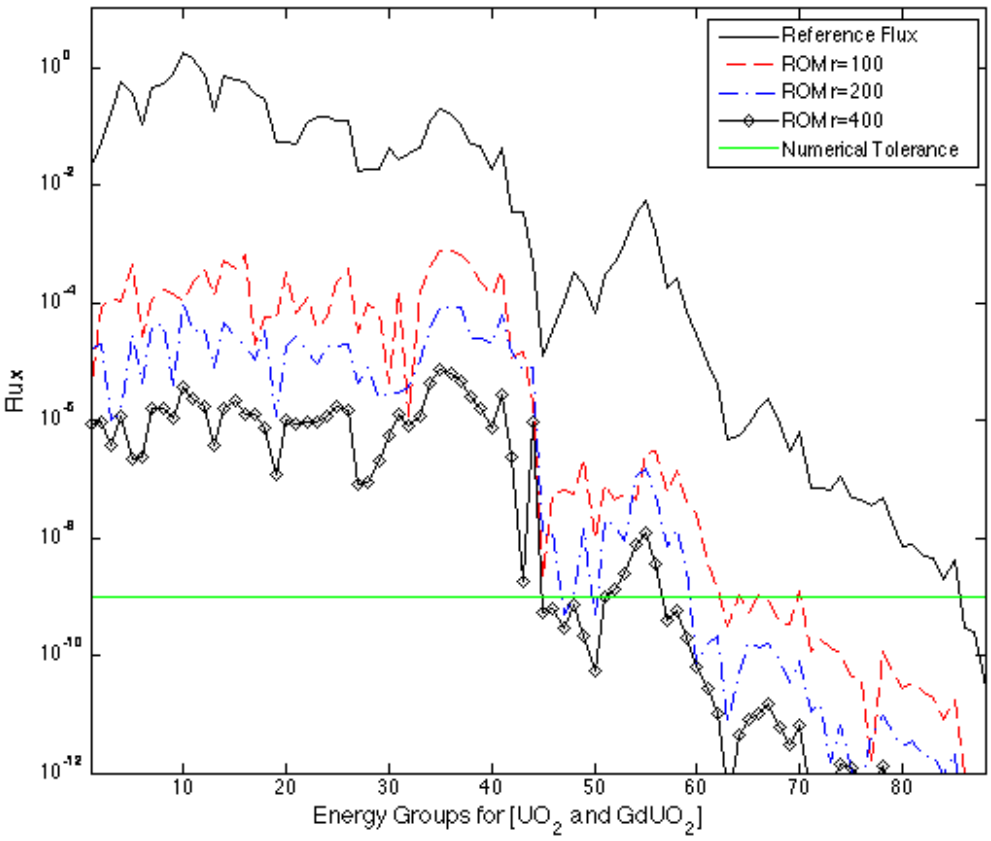

Figure 108: ROM Reduction Error for the two primary fuel regions UO_2 and $GdUO_2$. Note that the κ-Metric errors are dominated by the largest error terms corresponding to the largest flux.

Using an ROM of dimension 215 to reduce reduction error to approximately less than 1% of the flux, the GPT-Free algorithm generated a set of sensitivities that were used to examine the quality of the sensitivity analysis results as compared to the aforementioned reduction errors. The ROM was generated using 225 forward simulations with a least squares regression onto the first 215 columns of the subspace **Q** for the set of sensitivities generated by 5% RMS cross-section perturbations.

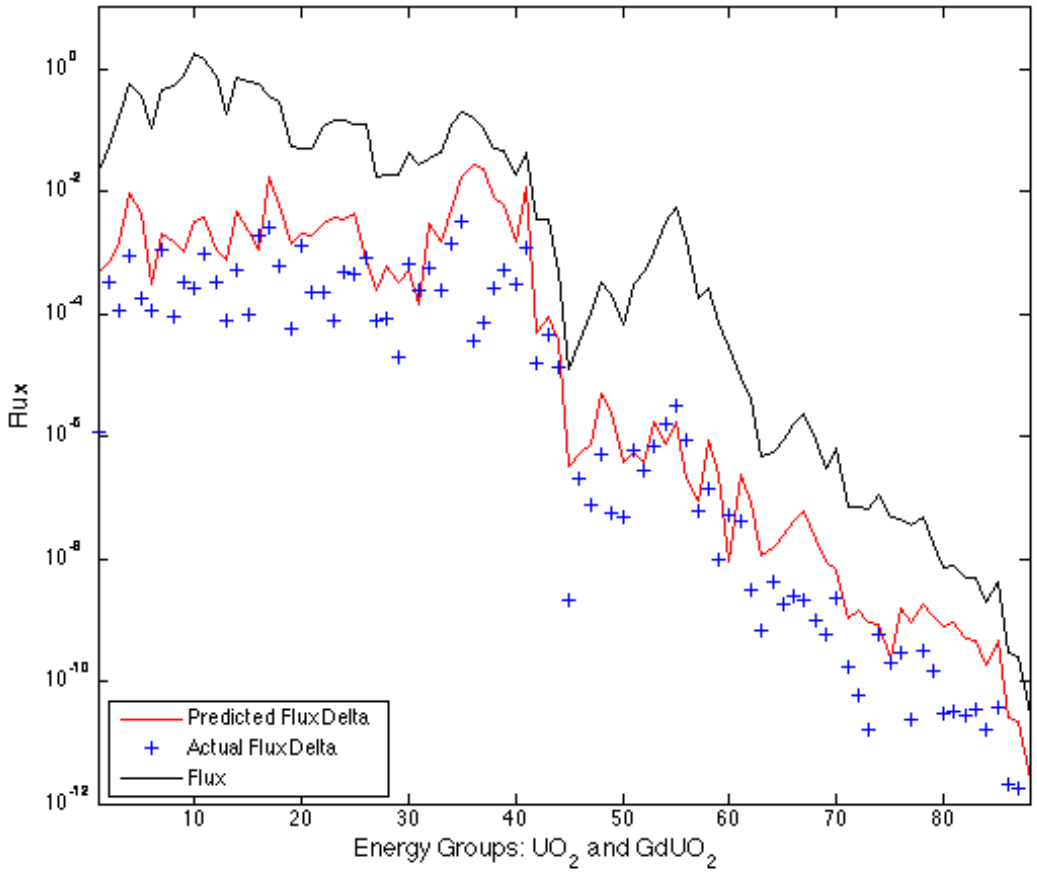

Figure 109: GPT-Free test case comparing an actual and predicated change in flux due to 5% RMS cross-section perturbations with an ROM dimension of 215

Figure 109 shows a sample test result using the aforementioned ROM for the flux. The predicted flux is comparable, even below the numerical tolerance at 10^{-9} with an exception near the epithermal region in the UO_2 fuel pins. This anomaly is likely due to the combination of mixtures to compute the final results. By increasing ROM dimension, it is expected that this error will decrease.

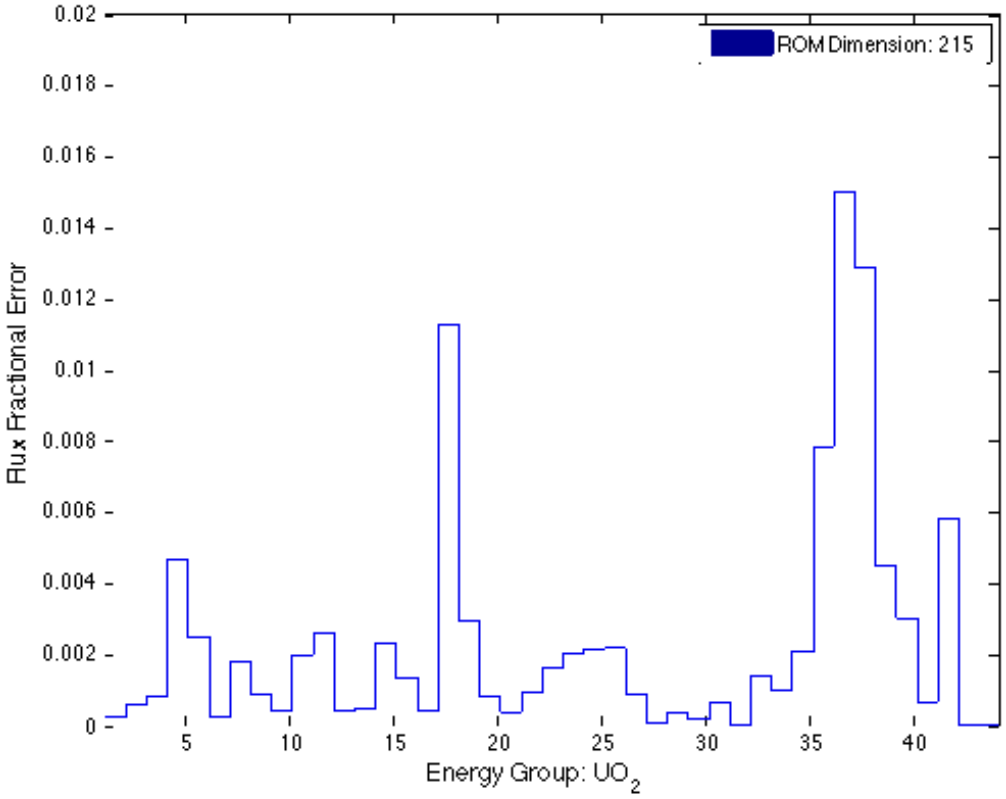

Figure 110: GPT-Free SA fractional flux error by energy group for a UO_2 fuel pin.

Figure 110 checks the reduction error estimate based on the ROM. For the UO_2 fuel pin, the fuel errors, with exceptions near epithermal and energy group 18, all fall below the expected 1% tolerance picked by the ROM. The average error is significantly lower for the UO_2 fuel pin, and is consistent with the κ-Metric as predicted in Figure 106.

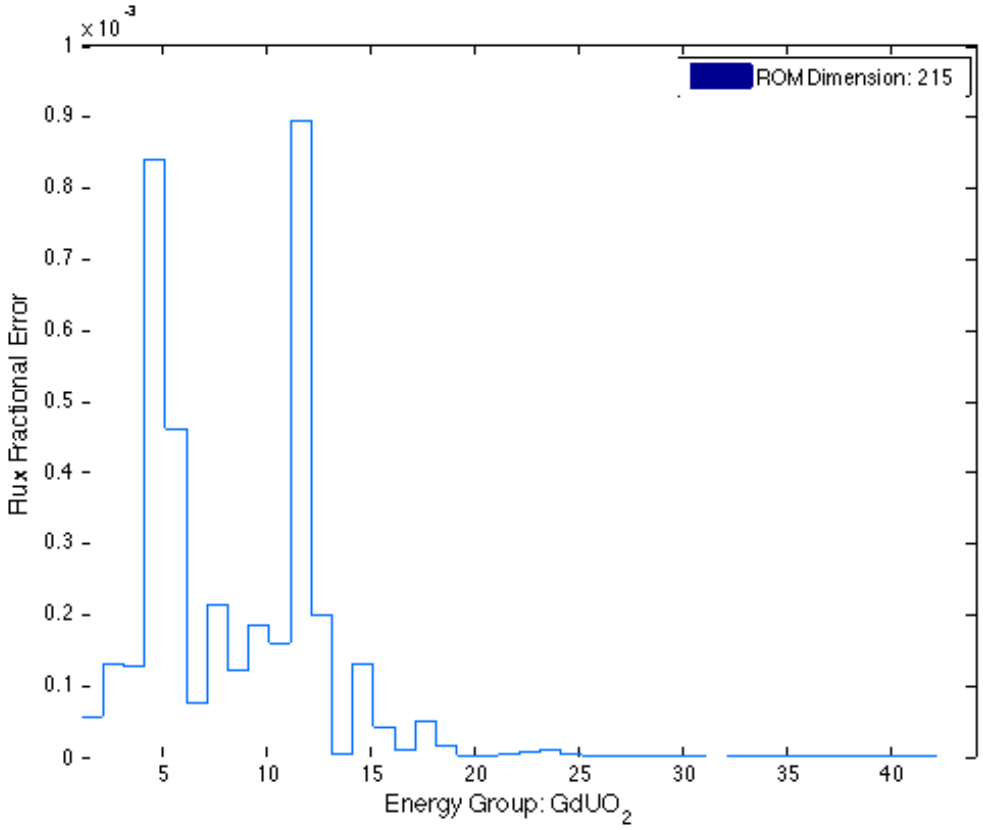

Figure 111: GPT-Free SA fractional flux error by energy group for a Gd-UO_2 pin.

Figure 111 checks the reduction error estimate based on the ROM again, but for the Gd-UO_2 fuel pin. All energy group errors fall below the expected 1% tolerance picked by the ROM. Actually, the error is also below 0.1%. Comparing this result with Figure 107, the observed error of a lower dimension ROM being sufficient to predict the Gd-UO_2 fuel pin is consistent with the reduction error predicted by the κ-Metric. Furthermore, the average error is significantly lower, approximately 0.01%.

7. CONCLUSIONS AND LIMITATIONS

This dissertation describes a GPT-free approach to compute responses sensitivities that does not require the explicit formulation of GPT equations. This approach requires the execution of classical PT to solve for the fundamental adjoint flux and construct the *k*-eigenvalue sensitivity profile vector. Unlike the GPT approach, it is not dependent on the number or type of responses, but instead the rank of the model. The approach employs the results generated from classical PT to find a subspace in the parameters space that can be used to construct a ROM that is more amenable for forward sensitivity analysis. Further, the ROM reduction error is user controlled. As observed in previous work, and further demonstrated in current work, the size of the subspace is significantly smaller than the original number of cross-sections for representative neutronics models. This implies that one can estimate all relevant responses variations using a forward sensitivity analysis rather than a traditional GPT approach.

The GPT-free approach has two sources of errors, one that results from constraining input parameters perturbations to a subspace, i.e. the reduction error in forming a ROM, and the other resulting from the first-order sensitivities surrogate model formed from regression/finite differencing. While increasing the size of the subspace can control for the former error-term; the latter however is highly sensitive to the choice of the regression surface. For linear and quasi-linear models, a regression approach is straightforward and represents the most accurate approach to calculating first-order sensitivities. When the GPT-free approach is extended to nonlinear models, more developments will be needed to optimally choose the regression surface.

Specific to depletion GPT-Free, results from the first order sensitivity analysis are more dominated by the second source of error due to the non-linearity of the number density at some future burn up due to cross-section perturbations at the BOC. While the ROM appropriately determines the space of all cross-sections that are important in the problem, future burn-up flux results are dependent on the change in cross-section introduced at BOC *and* the change in number densities that have evolved over the depletion cycle. This is effect is captured as the correction term at each time-step in William's paper in 1978.

There are several important limitations in the work demonstrated by this dissertation. First, the κ-Metric is not monotonically decreasing with increasing rank. While there is a general downward trend in most graphs, a single step is insufficient to judge cutoff criteria. The κ-Metric attempts to overcome this limitation by means of sampling and Wilks' statistics by providing a likelihood of meeting the criteria set forth by the user. A second limitation is that this approach was applied specifically to absorption cross-sections; testing was not completed on scattering cross-sections. The expectation is that scattering will be more challenging for the GPT-Free approach to capture without larger perturbations in number-density and cross-section. Finally, the approach depends on computational power to compute generalized inverses to large matrices. Because the emphasis of this work is the approach to compute sensitivities by avoiding GPT formulation, regression and mathematical alternatives were not a focus of this dissertation. Additional considerations may be required when matrices exceed desktop computational limits. In summary, these limitations are not prohibitive of the approach, as demonstrated by this dissertation; however, they must be considered in context with future applications.

8. FUTURE WORK

An important future application of this approach is the determination of response sensitivities for continuous energy Monte Carlo models. Implementing a GPT approach in a Monte Carlo model is not straightforward given the different modeling philosophies (e.g. computing inner products equivalents without rerunning the model). This approach provides an immediate resolution to this problem. Its implementation is non-intrusive as it only requires the ability to perturb cross-sections and the ability to calculate the sensitivity of *k*-eigenvalue with respect to cross-sections, with work towards the latter being recently demonstrated in Kiedrowski as described in section 5.D.

Additional future work is to extend the current developments to nonlinear models. This is important for multi-physics models where the construction of a global GPT model is often computationally impractical. Several regression schemes will be investigated; we will take advantage of recent developments in the applied mathematical and statistical communities on the construction of higher order regression techniques for nonlinear models.

Additionally, two trends from the GPT-Free ROM occur consistently: First, the ROM selects automatically a set of input parameter basis; however, capturing the tail-effects becomes more difficult for the algorithm. This is an opportunity for optimization where the algorithm can be improved to better sample/select the basis as the ROM approaches the reference model. Second, the sensitivity analysis tends to run into trouble due to the order of magnitude difference between variable sensitivities. The sampling and differencing scheme likely can be improved to better account for this variability *vs*. a simple finite difference or linear least squares regression as employed in this dissertation.

The calculation of Depletion GPT-Free using the SCALE code runs into trouble for two reasons: First, any master-library perturbations that affect infinite-dilution cross-sections and resonance parameters is overwritten at every depletion cycle. Second, any attempts to use the working-library are impractical because the TRITON software is not easily disassembled into isolated SCALE input files. Steady-state calculations do not have this difficulty because the TSUNAMI routines are straightforward to reconstruct. To correct for this, either TRITON's sequence must be modified, or the depletion sequence for TRITON must be emulated such that cross-sections can be injected into the working library – Infinite Dilution XS are MT + 3000 – appropriately for each time point under consideration such that the resonance regions and respective sensitivities can be appropriately determined. While NITAWLST performs better than BONAMIST/CENTRM in adjusting sensitivities in the resonance ranges due to master library cross-section perturbations, implementation and accuracy could be significantly improved by addressing this limitation.

The Gaussian sampling scheme to construct a ROM is favorable compared to ordered basis such as the Fourier series or the canonical basis; however, the Gaussian basis, as described in the Monte Carlo discussion, is very inefficient because the majority of samples are very nearly zero, giving large-scale models difficulty. While the recommended alternative sampling scheme is a recommended starting point, this concept of sampling can be pushed further to optimize the ROM basis formation. Not only would this research reduce the ROM dimensionality, but also it would improve the numerical quality of the results.

REFERENCES

Abdel-Khalik, H. 2011, "On Nonlinear Reduced Order Modeling," *M&C 2011*, Rio de Janeiro, RJ, Brazil, May 8-12.

Abdel-Khalik, H. S. 2004, *Adaptive Core Simulation*, Ph.D. Dissertation.

Abdel-Khalik, H., et al. 2007, "Efficient Subspace Methods-Based Algorithms for Performing Sensitivity, Uncertainty, and Adaptive Simulation of Large-Scale Computational Models," *Nuclear Science and Engineering*, Vol. 159, pp. 256-272.

Allen, H. 1975, "Nuclear-Reactor Analysis," The MIT Press, MA.

Al'tshul', L. M. and Karpman, V. I. 1966, "Theory of Nonlinear Oscillations in a Collisionless Plasma," NASA TTF-10, 15.

Andersson, K., et. al. 1990, "Second-Order Perturbation Theory with a CASSCF Reference Function," *Journal of Physical Chemistry*, Vol. 94, pp. 5483-5488.

Bang, Y. 2012, "Hybrid Reduced Order Modeling Applied to Nonlinear Models," *International Journal for Numerical Methods in Engineering*, Vol. 91(9), pp. 929-949.

Bang, Y., Kennedy, C., and Abdel-Khalik, H. 2011, "On the Propagation of Uncertainties in High Dimensional Models," *Transactions of the American Nuclear Society*, Vol. 105, pp. 468-472.

Bates, D., et. al. 1988, *Nonlinear Regression Analysis and Its Applications,* John Wiley and Sons.

Bell, G. I. and Glasstone, S. 1970, *Nuclear Reactor Theory*, VNR Company.

Brown, F., et. al. 2002. "MCNP Version 5," LA-UR-02-3935.

Cacuci, D. 2003, *Sensitivity and Uncertainty Analysis: Theory*, CRC Press.

Cacuci, D. et. al. 1980, "Sensitivity Theory for General Systems of Nonlinear Equations," *Nuclear Science and Engineering*, Vol. 75, pp. 88-110.

Cacuci, D. G., ed. 2010. *Handbook of Nuclear Engineering*.

Carter, L. and Cashwell, E. 1975, *Particle-Transport Simulation with the Monte Carlo Method*, ERDA Critical Review Series.

Chadwick, M. B. et al. 2006, "ENDF/B-VII.0: Next Generation Evaluated Nuclear Data Library for Nuclear Science and Technology," *Nuclear Data Sheets*.

Daley, R. 1991, *Atmospheric Data Analysis*, Cambridge University Press.

Dean, V. F. 2007, "ICSBEP Guide to the Expression of Uncertainties," *Idaho National Laboratory.*

Dehart, M., et al. 2009, "Benchmark Specification for HTGR Fuel Element Depletion," NEA/NSC/DOC(2009)13.

Dehart, M., et al. 2009, "NEWT: A New Transport Algorithm for Two-Dimensional Discrete Ordinates Analysis in Non-Orthogonal Geometries," ORNL/TM-2005/39.

Dehart, M., et al. 2009, "TRITON: A Two-Dimensional Transport and Depletion Model for characterization of Spent Nuclear Fuel," ORNL/TM-2005/39.

Dehart, M., et al. 2011, "TRITON: A Two-Dimensional Transport and Depletion Model for characterization of Spent Nuclear Fuel," ORNL/TM-2005/39, section T1.2.4.

Duderstadt, J. and Hamilton L. 1976, *Nuclear Reactor Analysis*, Wiley.

Ehrlich R. and Hurwitz H. Jr. 1954, "The Multigroup Diffusion Equations," *Nukleonik*, Vol. 1, pp. 128.

Eriksson, B. et. al.1969, "Monte Carlo Integration of the Adjoint Neutron Transport Equation," *Nuclear Science and Engineering*, Vol 37, pp. 410.

Ganapol, B. D. 2008, "Analytical Benchmarks for Nuclear Engineering Applications," NEA-1827/02.

Gandini, A. 1967, "A Generalized Perturbation Method for Bilinear Functionals of the Real and Adjoint Neutron Fluxes," *Journal of Nuclear Energy*, Vol. 21, pp. 755.

Gandini, A. 1987, "Generalized Perturbation Theory Methods. A Heuristic Approach," *Advances in Nuclear Science and Technology*, Vol. 19, pp. 205.

Gandini, A. 1988, *Uncertainty Analysis and Experimental Data Transposition Methods in Uncertainty Analysis*, CRC Press.

Gandini, A., 2001. "HGPT Based Sensitivity Methods for the Analysis of Subcritical Systems," Annals of Nuclear Energy, Vol 28, pp. 1193.

Gentle, J. 2009, *Computational Statistics*, Springer.

Ghanem, R., et. al. 1991, *Stochastic Finite Elements – A Spectral Approach*, Springer-Verlag.

Ghil, M. 1991, "Data Assimilation in Meteorology and Oceanography," *Advances in Geophysics*, Vol. 33, pp. 141-266.

Golub, G. and Van Loan, C. 1996, *Matrix Computations*, The Johns Hopkins University Press.

Gustavsen, B., et. al. 1999, "Rational approximation of frequency domain responses by Vector Fitting," *IEEE Transactions on Power Delivery*, Vol. 14(3), pp. 1052-1061.

Halko, N., et. al. 2009, "Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions," *SIAM Review*, Vol. 53, pp. 217-288.

Harries, J. R. 1978, "Inverse Kinetics Reactivity Measurements on the Materials Testing Reactor HIFAR," AAEC/E456.

Hendrickx, W., et. al. 2004, "Some Remarks on the Vector Fitting Iteration," *Progress in Industrial Mathematics at ECMI*, pp. 134-138.

Hoogenboom, J. E. 1981, "A Practical Adjoint Monte Carlo Technique for Fixed-Source and Eigenfunction Neutron Transport Problems," *Nuclear Sci. and Eng*., Vol. 79, pp. 357-373.

Hurwitz H. Jr. 1948, "A Note on the Theory of Danger Coefficients," KAPL-98.

Ivanov, K. et al. 1999, "PWR MSLB Benchmark. Volume 1: Final Specifications," NEA/NSC/DOC(99).

Ivanov, K. et al. 2007, "Benchmark for Uncertainty Analysis in Modeling (UAM) for Design, Operation and Safety Analysis of LWRs," NEA/NSC/DOC(2007)23.

Jessee. M., et. al, "Development of Generalized Perturbation Theory Capability Within the SCALE Code Package"

Kando, H. et. al. 1971, "Consideration of Eigenvalue Sensitivity for a Reactor System," *Journal of Nuclear Science and Technology*, Vol 8(7), pp. 363-370.

Kelley, C. T. 1995, *Iterative Methods for Linear and Nonlinear Equations*, SIAM.

Kelley, C. T. 1999, *Iterative Methods for Optimization*, SIAM.

Kennedy, C. 2012, "GPT-Free Sensitivity Analysis for Eigenvalue Problems," *Nuclear Technology*, Vol. 179(2), pp. 169-179.

Kennedy, C., et. al. 2011, "On Rank Determination for Subspace Methods," *Transactions of the American Nuclear Society*, Vol. 105, pp. 461-463.

Kennedy, C. 2011, "Perturbation Analysis for Monte Carlo Continuous Cross Section Models," *International Conf. on Math. And Computational Methods Appl. to Nuclear Science and Engineering (M&C 2011)*, Rio de Janeiro, RJ, Brazil, May 8-12.

Kennedy, C. 2012, "Depletion GPT-Free Sensitivity Analysis for Eigenvalue Problems," submitted to the *Transactions of the American Nuclear Society*, Winter Meeting.

Kiedrowski, B. et al. 2010, "Calculating Kinetics Parameters and Reactivity Changes with Continuous-Energy Monte Carlo," *Proc. PHYSOR.*

Kiedrowski, B. et al. 2010, "Verification of k-Eigenvalue Sensitivity Coefficient Calculations Using Adjoint-Weighted Perturbation Theory in MCNP," *LA-UR-10-07261*.

Komata, M. 1977, "A Generalized Perturbation Theory Applicable to Reactor Boundary Changes," *Nuclear Science and Engineering*, Vol. 64, pp. 811.

Lamarsh, J. 2001, *Introduction to Nuclear Engineering*, Prentice Hall.

Lamarsh, J. 2002, *Introduction to Nuclear Reactor Theory*, ANS.

Larsen, E. and Pomraning G. 1981, "Boundary Perturbation Theory," *Nuclear Science and Engineering*, Vol 77, pp. 415.

Le Maitre, O., et. al. 2010, *Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics*, Springer.

Leppänen, J. 2009. "Two Practical Methods for Unionized Energy Grid Construction in Continuous-Energy Monte Carlo Neutron Transport Calculation," *Ann. Nucl. Energy*, Vol. 36, pp. 878-885.

Lewins, J. 1960, "A Derivation of the Time-Dependent Adjoint Equations for Neutron Importance in the Transport, Continuous Slowing-Down and Diffusion Models," *Journal of Nuclear Energy, Part A: Reactor Science*, Vol 13, pp. 1-5.

Lewins, J. 1966, "A Variational Principle for Ratios in Critical Systems," *Journal of Nuclear Energy Part A/B*, Vol 20, pp. 141.

Martin-del-Campo, C. 2004, "Development of a BWR Loading Pattern Design System Based on Modified Genetic Algorithms and Knowledge," *Annals of Nuclear Energy,* Vol 31(16), pp. 1901-1911.

McKinley, M. S. and Rahnema, F. 2001, "High-Order Boundary Condition Perturbation Theory for the Neutron Transport Equation," UCRL-ID-155132.

Meyer, C. 1999, *Matrix Analysis and Applied Linear Algebra*, SIAM.

Mitsunaga, M. and Brewer R. G. 1985, "Generalized Perturbation Theory of Coherent Optical Emission," *Physical Review A*, Vol 32(3), pp. 1605-1613.

Moore, B., et. al. 1999, "FORMOSA-B: A boiling water reactor in-core fuel management optimization package," *Nuclear Technology*, Vol. 126(2), pp. 153-169.

NCSU. 2012, "High Performance Computing Load Sharing Facility," http://www.ncsu.edu/itd/hpc/Documents/usefulLSF.php.

Oblow, E. 1976, "Sensitivity Theory from a Differential Viewpoint," *Nuclear Science and Engineering*, Vol. 59(2), pp. 187.

Palmiotti, G., et. al. 2010, "Use of Covariance Matrices in a Consistent (Multiscale) Data Assimilation for Improvement of Basic Nuclear Parameters in Nuclear Reactor Applications: From Meters to Femtometers," *International Conference on Nuclear Data for Science and Technology*, INL/CON-10-18506.

Petrov, Y. V. and Sakhnovsky, E. G. 1985, "On the Boundary Perturbation Theory as Applied to Nuclear Reactors," *Nuclear Science and Engineering*, Vol. 90, pp. 1-12.

Pomraning, G. 1966, "Generalized Variational Principles for Reactor Analysis," *Proceedings of the International Conference on the Utilization of Research Reactors and Mathematics and Computation*, Mexico, D.F., pp. 250.

Pomraning, G. 1967, "Variation Principle for eigenvalue equations," *Journal of Math and Physics*, Vol. 8, pp. 149.

Pomraning, G. 1967, "A Derivation of Variational Principles for Inhomogenous Equations," *Nuclear Science and Engineering*, Vol 29, pp. 220.

Pomraning, G. 1967. "The Calculation of Ratios in Critical Systems," *Journal of Nuclear Energy Part A/B*, Vol 21, pp. 755.

Pople, J. A. 1955, "Molecular Orbital Perturbation Theory. I. A Perturbation Method Based on Self-Consistent Orbitals," *Proceedings of the Royal Society of London. Series A*, pp. 233-241.

Primas, H. 1963, "Generalized Perturbation Theory in Operator Form," *Review of Modern Physics*, Vol 35(3), pp. 710-712.

PYTHON. 2012, "PYTHON programming language," http://www.python.org.

Rahnema, F. and Pomraning, G. 1983, "Boundary Perturbation Theory for Inhomogenous Transport Equations," *Nuclear Science and Engineering*, Vol 84, pp. 313.

Rahnema, F. and Ravetto, P. 1998, "On the Equivalence of Boundary and Boundary Condition Perturbations in Transport Theory and Its Diffusion Approximation," Nuclear Science and Engineering, 128, pp. 209-223.

Rayleigh, J. W. S. 1894, *Theory of Sound (Volume II)*. London: Macmillan. pp. 115–118.

Roache, P. 2002, "Code Verification by the Method of Manufactured Solutions," *Journal of Fluids Engineering*, 114 No. 1, pp. 4-10.

SCALE: *A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluations*, ORNL/TM-2005/39, Version 5, Vols. I–III, April 2005. Available from Radiation Safety Information Computational Center at Oak Ridge National Laboratory as CCC-725.

Schilders, W. H. A., et. al. 2008, *Model Order Reduction*, Springer.

Schrödinger, E. 1926, "Quantisierung als Eigenwertproblem," *Annalen der Physik*, Vol. 385(13), pp. 437.

Seber, G., et. al. 2003, *Nonlinear Regression Analysis*, John Wiley and Sons, Hoboken, N.J.

Sheldon, X., et. al. 2007, *Advanced Model Order Reduction Techniques in VLSI Design*, Cambridge University Press.

Sowers, G. and Sandler, S. I. 1991, "Equations of State from Generalized Perturbation Theory. Part 1. The Hard-Core Lennard-Jones Fluid," *Fluid Phase Equilibria*, Vol. 63, pp. 1-25.

Stacey, W. 2001, *Nuclear Reactor Physics*, Wiley-Interscience.

Stacey, W. 1972, "Variational Estimates of Reactivity Worths and Reaction Rate Ratios in Critical Systems," *Nuclear Science and Engineering*, Vol 48, pp. 444.

Stacey, W. 1972, "Variational Estimates and Generalized Perturbation Theory for Ratios of Linear and Bilinear Functionals," *Journal of Math and Physics*, Vol 13, pp. 1119.

Stacey, W. 1974, *Variational Methods in Nuclear Reactor Physics*, Academic Press, New York.

Stamm'ler, R. J. J. and Abbate, M. J. 1983, *Methods of Steady-State Reactor Physics in Nuclear Design*, Academic Press.

Strydom, G. 2011, "SUSA Uncertainty Analysis of the PEBBED-THERMIX CRP-5 PBMR DLOFC Benchmark."

Turinsky, P. J. 2010, Core Isotopic Depletion and Fuel management. In D. G. Cacuci, *Handbook of Nuclear Engineering*. Springer, New York, pp. 1243-1249.

Turinsky, P., et. al. 1995, "Computer code abstract: NESTLE," *Nuclear Science and Engineering*, 120(1), pp. 72-73.

US Department of Energy (DOE). 2012, "General Information," http://www.ne.doe.gov/.

Usachev, L. N. 1964, "Perturbation theory for the breeding ratio and for other ratios pertaining to various reactor processes," *Journal of Nuclear Energy A/B*, Vol. 18, pp. 571.

Villarino, E. 1992, "HELIOS; Angularly dependent collision probabilities," *Nuclear Science and Engineering*, Vol. 112(1), pp. 16-31.

Wagner, J. C.. et. al. 1994, "MCNP: Multigroup/Adjoint Capabilities," *LA-12704*.

Wang, C. 2012, "Exact-to-Precision Generalized Perturbation Theory for Eigenvalue Problem," *Nuclear Engineering Design* (submitted).

Wigner, E. 1945, "Effect of Small Perturbations on Pile Period," Chicago Report CP-G-3048.

Wilks, S. 1941, "Determination of Sample Sizes for Setting Tolerance Limits," *Annals of Math. Statistics*, Vol. 12, pp. 91.

Williams, M. L. 1979, "Development of Depletion Perturbation Theory for Coupled Neutron/Nuclide Fields," *Nuclear Science and Engineering*, Vol. 70, pp. 20.

Williams, M. L. 1978, "Development of depletion perturbation theory for coupled neutron/nuclide fields," ORNL/TM-5958.

Williams, M. L. 1986, *CRC Handbook of Nuclear Reactors Calculations: Perturbation Theory for Nuclear Reactor Analysis*, CRC Press, West Palm Beach, Florida, III, pp. 63-181.

Williams, M. L. 1991, "Reactivity due to deformation of a thin plate in a critical reactor," *Nuclear Science and Engineering*, Vol. 108(2), pp. 150-171.

Yamamoto. 1997, "Perturbation of general boundary condition for an eigenvalue change in the neutron Boltzmann transport equation," Nuclear Science and Engineering, Vol. 125(1), pp. 19.

Yang, W. S. and Downar, T. J. 1989, "Depletion Perturbation Theory for the Constrained Equilibrium Cycle", Nuclear Science and Engineering, Vol. 102(4), pp. 365-380.

APPENDIX

A.I GPT-Free Depletion Instructions with PYTHON + SCALE 6.x:

All instructions here are for the NCSU HPC using the load-sharing facility (NCSU 2012).

1. SETUP-PHASE
    a. Construct the following directories in a folder, henceforth denoted WORK:
        i. WORK/data/
        ii. WORK /pdata/
        iii. WORK /cx_data/
        iv. WORK /tmpdir/
        v. WORK /output/
        vi. WORK /utils/
    b. Run the model depletion case, abort the software run prior to Bonami and after TRITON.
    c. Copy the file *ft11f001*, the shortened master library file to the data/ directory
    d. Prepare the HPC files for this directory and place them in the data/ directory
        i. bsub_scale
        ii. scale_run
        iii. NOTE: These files must be saved with the UNIX EOL format.
    e. Copy all required utilities and scripts to the utils/ directory
    f. Copy the job instruction file (JIF) to the WORK/ directory
    g. Check that CLAROL+(Master) is installed and the path is included in the scale_run file
2. DEPLETION-EVOLUTION CURVE (DEC)
    a. Prepare the DEC PYTHON script with the HPC instructions for your specific system.
    b. Prepare the DEC Reference input file (modified to include the script-generated perturbation commands) in the data/ directory
    c. Call the DEC PYTHON script as follows:
    dec.py $J $NP $JIF $INP $Q $QT
        i. $J = Job #
        ii. $NP = # of perturbations to complete
        iii. $JIF = Job Instruction File
        iv. $INP = SCALE Input File
        v. $Q = HPC Queue
        vi. $QT = HPC Queue Time (In Minutes). This must be sufficiently large to allow completion.
    d. Collect all output files and parse for nuclide density information
    e. Build a nuclide-density depletion-evolution curve table

3. DEPLETION-ROM FORMATION (DRF)
    a. Prepare the DRF PYTHON script with the HPC instructions for your specific system.
    b. Prepare the DRF reference input file (modified to inject mixture-specific number densities).
        i. This file is model specific based on both the number and temperature of mixtures.
    c. Call the DRF PYTHON script as follows:
    drf.py $J $NP $JIF $INP $Q $QT
        i. Compute a set of *N* perturbations to set aside based on the Wilks' criteria.
        ii. Compute an initial guess of P perturbations to estimate the ROM.
    d. Begin the ROM determination loop
        i. Collect sensitivity data into the output directory
        ii. Build a ROM from the sensitivity data, and project onto the *N* perturbations
        iii. Run the model with the projected perturbations
        iv. Compare to the reference case using the κ-metric
        v. If insufficient accuracy, run additional perturbations and repeat
4. GPT-Free Evaluation
    a. Use a forward sensitivity analysis along the ROM input space from the DRF step to compute sensitivities of responses due to input parameters

A.II Depletion Perturbation Sequences

Depletion Perturbation Sequence (CLAROL+):

1. Run the depletion problem (T-DEPL) and exit after construction of the master library (ft11f001)
2. Run ENSTRUC to generate an energy group structure file (ft47f001)
3. Use VARIGEN or PPERT to generate cross-section perturbation parameter files for CLAROL+
4. Run CLAROL+ to generate a new perturbed master library (ft99f001)
5. Run the depletion problem with the new perturbed master library

Depletion Perturbation Sequence (AIM+)

1. Run AIM to convert the binary reference library to text.
2. Run PAIM to perturb the working text library
3. Run AIM again to package the perturbed library as a new master perturbed library
4. Run the depletion problem (T-DEPL)

Differences between CLAROL+ and AIM+

- CLAROL+ corrects for Bondarenko factors in the XS data. AIM+ does not currently support this feature. CLAROL+ incorrectly adjusts Bondarenko factors and this feature has thus been disabled.
- CLAROL+ supports 2D nuclide perturbations and more general MT perturbations. AIM+ supports MT 2 (Isotropic elastic scattering), MT 18 (Fission), and MT 102 (Radiative Capture)

Comparison of AIM+ and CLAROL+ due to the effect of Bondarenko factor correction for 5 test cases:

TEST CASE	**1**	**2**	**3**	**4**	**5**
***k*-eff (AIM+)**	0.979818	1.123007	1.048604	1.118065	1.096524
***k*-eff (CLAROL+)**	0.980174	1.120274	1.047523	1.118978	1.091639
PCM ERROR	36.25684	-243.954	-103.275	81.66467	-447.474

In general, CLAROL+ is superior to AIM+. AIM+ with modifications could be used to avoid the preliminary step required for CLAROL+. Alternatively, CLAROL+ could be upgraded to work directly with master binary cross-section libraries (e.g. xn44)

A.III Reference Input Files for UAM & TMI Depletion Test Cases

Job Specification File:

```
44 9 16 2 1 0.1                ;* NG NMIX NISO NRXN PMODE PMAG
1
2
4
201
202
212
203
213
500
92234
92235
92236
92238
94238
94239
94240
94241
94242
94243
95241
95243
96242
96243
96244
96245
18
102
```

NG = # of energy groups
NISO = # of Isotopes
NRXN = # of Reactions
PMODE = Perturbation Mode (1 = Gaussian)
PMAG = Perturbation Magnitude (RMS), e.g. 0.1 is equivalent to 10% RMS
*Omit the semicolon and remainder of text on the line.

Reference Input File (ROM Determination) UAM Model:

```
%%INJECT%%

=tsunami-2d   parm=(nitawlst)
Depletion Time-Point Run
pxn44
' Data taken from:
'        Benchmark for Uncertainty Analysis in Modeling (UAM)
'        for Design, Operation and Safety Analyses of LWRs,
'        Nuclear Energy Agency, 2007.
' JOB: %%JID%%
' PERT: %%PID%%
' TIME: %%TID%% days
read alias
 $gadpin 500 end
 $clad   101 102 103 104 105 301 302 303 304 305 end
 $mod    111 112 113 114 115 311 312 313 314 315 end
 $gap    121 122 123 124 125 321 322 323 324 325 end
end alias
read comp
' 2.93% enriched fuel pin
%%COMP_1%%
%%COMP_201%%
' 1.94% enriched fuel pin
%%COMP_2%%
%%COMP_202%%
%%COMP_212%%
' 1.69% enriched fuel pin
%%COMP_203%%
%%COMP_213%%
' 1.33% enriched fuel pin
%%COMP_4%%
' 3% Gd2O3 by weigh, 2.93% enriched fuel pin
%%COMP_500%%
' gap/clad/moderator
  he    $gap  den=4.9559E-4 1 711.15  end
  zirc2 $clad den=5.678 1 630 end
  h2o   $mod    den=0.4577 1  560  end
' channel
  zirc4 630 den=6.525 1 630 end
' water in bypass
  h2o   620 den=0.738079 1 560  end
end comp
read celldata
  latticecell squarep pitch=1.6359 111 fuelr=0.60579   1 gapr=0.62103 121
cladr=0.71501 101 end
  latticecell squarep pitch=1.6492 112 fuelr=0.60579   2 gapr=0.62103 122
cladr=0.71501 102 end
  latticecell squarep pitch=1.8588 114 fuelr=0.60579   4 gapr=0.62103 124
cladr=0.71501 104 end
  latticecell squarep pitch=1.7524 311 fuelr=0.60579 201 gapr=0.62103 321
cladr=0.71501 301 end
```

```
  latticecell squarep pitch=1.7524 312 fuelr=0.60579 202 gapr=0.62103 322
cladr=0.71501 302 end
  latticecell squarep pitch=1.7621 313 fuelr=0.60579 203 gapr=0.62103 323
cladr=0.71501 303 end
  latticecell squarep pitch=1.8646 314 fuelr=0.60579 212 gapr=0.62103 324
cladr=0.71501 304 end
  latticecell squarep pitch=1.8684 315 fuelr=0.60579 213 gapr=0.62103 325
cladr=0.71501 305 end
  latticecell squarep pitch=1.6333 115 fuelr=0.60579 500 gapr=0.62103 125
cladr=0.71501 105 end
end celldata

read model
Depletion Setup for UAM Model (GPT-Free)
read parm
 echo=yes timed=yes drawit=yes cmfd=1 epsilon=1e-6 inners=5 therm=yes
therms=1 outers=9999 xycmfd=4 saveangflx=yes prtbalnc=yes
end parm
read materials
  mix=  1         pn=1  com='2.93% UO2'         end
  mix=  2         pn=1  com='1.94% UO2'         end
  mix=  4         pn=1  com='1.33% UO2'         end
  mix=$gadpin     pn=1  com='2.93% UO2 (3% Gd)' end
  mix=201         pn=1  com='2.93% UO2, edge'   end
  mix=202         pn=1  com='1.94% UO2, edge'   end
  mix=212         pn=1  com='1.94% UO2, corner' end
  mix=203         pn=1  com='1.69% UO2, edge'   end
  mix=213         pn=1  com='1.69% UO2, corner' end
  mix=111         pn=2  com='H2O(void)'         end
  mix=101         pn=1  com='Zirc2'             end
  mix=121         pn=1  com='Helium'            end
  mix=620         pn=2  com='H2O(solid)'        end
  mix=630         pn=1  com='Zirc4'             end
end materials
read geom
  unit   11
    com="corner rod 1.33% enr"
    cylinder 1   0.60579
    cylinder 2   0.62103
    cylinder 3   0.71501
    cuboid   4 4p0.9375
    media   4 1 1
    media 121 1 2 -1
    media 101 1 3 -2
    media 111 1 4 -3
    boundary    4  2 2
  unit   12
    com="edge rod 1.69% enr"
    cylinder 1   0.60579
    cylinder 2   0.62103
    cylinder 3   0.71501
    cuboid   4 4p0.9375
    media 203 1 1
```

```
  media 121 1 2 -1
  media 101 1 3 -2
  media 111 1 4 -3
  boundary    4  2 2
unit   14
  com="edge rod 1.94% enr"
  cylinder 1   0.60579
  cylinder 2   0.62103
  cylinder 3   0.71501
  cuboid   4 4p0.9375
  media 202 1 1
  media 121 1 2 -1
  media 101 1 3 -2
  media 111 1 4 -3
  boundary    4  2 2
unit   17
  com="corner rod 1.69% enr"
  cylinder 1   0.60579
  cylinder 2   0.62103
  cylinder 3   0.71501
  cuboid   4 4p0.9375
  media 213 1 1
  media 121 1 2 -1
  media 101 1 3 -2
  media 111 1 4 -3
  boundary    4  2 2
unit   22
  com="interior rod 1.94% enr"
  cylinder 1   0.60579
  cylinder 2   0.62103
  cylinder 3   0.71501
  cuboid   4 4p0.9375
  media   2 1 1
  media 121 1 2 -1
  media 101 1 3 -2
  media 111 1 4 -3
  boundary    4  2 2
unit   23
  com="interior rod 2.93% enr"
  cylinder 1   0.60579
  cylinder 2   0.62103
  cylinder 3   0.71501
  cuboid   4 4p0.9375
  media   1 1 1
  media 121 1 2 -1
  media 101 1 3 -2
  media 111 1 4 -3
  boundary    4  2 2
unit   33
  com="interior gad pin"
  cylinder 1   0.270917524
  cylinder 2   0.383135237
  cylinder 3   0.469242916
```

```
    cylinder 4   0.541835048
    cylinder 5   0.60579
    cylinder 6   0.62103
    cylinder 7   0.71501
    cuboid   8 4p0.9375
    media 500 1 1
    media 500 1 2 -1
    media 500 1 3 -2
    media 500 1 4 -3
    media 500 1 5 -4
    media 121 1 6 -5
    media 101 1 7 -6
    media 111 1 8 -7
    boundary    8  2 2
  unit   37
    com="edge rod 2.93% enr"
    cylinder 1   0.60579
    cylinder 2   0.62103
    cylinder 3   0.71501
    cuboid   4 4p0.9375
    media 201 1 1
    media 121 1 2 -1
    media 101 1 3 -2
    media 111 1 4 -3
    boundary    4  2 2
  unit   77
   com="corner rod 1.94% enr"
    cylinder 1   0.60579
    cylinder 2   0.62103
    cylinder 3   0.71501
    cuboid   4 4p0.9375
    media 212 1 1
    media 121 1 2 -1
    media 101 1 3 -2
    media 111 1 4 -3
    boundary    4  2 2
global unit 100
   cuboid  1   4p6.70306
   array 1 1  place 4 4  0.0 0.0
   cuboid  2   4p6.90626
   cuboid  10  4p7.62
   media   111 1 1
   media   630 1 2 -1
   media   620 1 10 -2
   boundary 10 32 32
end geom
read array
  ara=1 nux=7 nuy=7 typ=cuboidal
  fill
    17  14  37  37  37  14  77
    14  23  33  23  23  23  14
    14  23  23  23  33  23  37
    14  23  23  23  23  23  37
```

```
    12  23  33  23  23  33  37
    12  22  23  23  23  23  14
    11  12  12  14  14  14  17 end fill
end array
read bounds
  all=refl
end bounds
end model
end
```

Reference Input File (ROM Determination) TMI Model (Depletion Routine):

```
=t-depl
TMI Minicore UAM (44 group)
v5-44

read comp
'UO2 Fuel, enriched to 4.85% by weight
uo2     1  den=10.283  1.0  551.0  92235  4.85  92238  95.15  end
'Cladding, zircaloy4
zirc4   2  den=6.55    1.0  551.0  end
'Helium gas gap
he      3  1.0  551.0  end
'Water coolant
h2o     4  den=0.766   1.0  551.0  end
'Gad, Gd2O3+UO2, Gd2O3 is 2 w/o, UO2 is enriched 4.12 w/o
wtptgad 5  10.144 4 92235 3.51 92238 82.87 64000 1.74 8016 11.88 1.0 551.0 end
'Cladding Material
inconels 6  1.0  551.0  end
'Absorber
cd      7  den=9.927  0.05  end
in      7  den=9.927  0.15  end
ag      7  den=9.927  0.80  end
'CR Coolant 1
h2o     8  den=0.766  1.0  551.0  end
'CR Coolant 2
h2o     9  den=0.766  1.0  551.0  end
'Coolant for the Gad
h2o    10  den=0.766  1.0  551.0  end
'Helium for the Gad
he     11  1.0  551.0  end
'Cladding for the Gad
zirc4  12  den=6.55    1.0  551.0  end
end comp
read celldata
 latticecell squarep hpitch=0.72135  4 fuelr=0.46955 1 gapr=0.47910  3 cladr=0.54640  2 end
 latticecell squarep hpitch=0.72135 10 fuelr=0.46955 5 gapr=0.47910 11 cladr=0.54640 12 end
end celldata

read depletion
 1 flux 5
end depletion

read burndata
  power=21.220  burn=40   end
  power=21.220  burn=40   end
  power=21.220  burn=40   end
  power=21.220  burn=40   end
  power=21.220  burn=40   end
  power=21.220  burn=40   end
  power=21.220  burn=40   end
  power=21.220  burn=40   end
  power=21.220  burn=40   end
  power=21.220  burn=40   end
end burndata

read model
read parm
 run=yes
 echo=yes
 drawit=yes
 prtflux=yes
 collapse=yes combine=no
 cmfd=yes
 xycmfd=4  timed=yes
 solntype=keff
 epsilon=1.0e-6  outers=5000
 inners=5 prthmmix=yes
end parm
read materials
1  2  "UO2 4.85"          end
2  2  "Clad"              end
3  2  "Gas Gap"           end
4  2  "Coolant"           end
5  2  "Gad"               end
6  2  "Inconel"           end
```

```
7  2  "Absorber"          end
8  2  "CR Coolant 1"      end
9  2  "CR Coolant 2"      end
10 2  "Gad Coolant"       end
11 2  "Gad Helium"        end
12 2  "Gad Clad"          end
end materials
read geometry
' Full fuel pin cell
unit 1
cylinder 1  0.46955  origin x=0.72135 y=0.72135
cylinder 2  0.47910  origin x=0.72135 y=0.72135
cylinder 3  0.54640  origin x=0.72135 y=0.72135
cuboid   4  1.44270  0  1.44270  0
media 1  1  1
media 3  1  2  -1
media 2  1  3  -2
media 4  1  4  -3
boundary 4
' Guide tube with CR out (full)
unit 2
cylinder 1  0.63245 origin x=0.72135 y=0.72135
cylinder 2  0.67310 origin x=0.72135 y=0.72135
cylinder 3  0.50292 origin x=0.72135 y=0.72135
cylinder 4  0.56007 origin x=0.72135 y=0.72135
cuboid   5  1.44270  0  1.44270  0
media 8  1  3
media 9  1  4  -3
media 4  1  1  -4
media 2  1  2  -1
media 4  1  5  -2
boundary 5
' Guide tube with CR in (full)
unit 3
cylinder 1  0.63245  origin x=0.72135 y=0.72135
cylinder 2  0.67310  origin x=0.72135 y=0.72135
cylinder 3  0.50292  origin x=0.72135 y=0.72135
cylinder 4  0.56007  origin x=0.72135 y=0.72135
cuboid   5  1.44270 0  1.44270  0
media 7  1  3
media 6  1  4  -3
media 4  1  1  -4
media 2  1  2  -1
media 4  1  5  -2
boundary 5
' Fuel with gad pin cell (full)
unit 4
cylinder 1  0.46955  origin x=0.72135 y=0.72135
cylinder 2  0.47910  origin x=0.72135 y=0.72135
cylinder 3  0.54640  origin x=0.72135 y=0.72135
cuboid   4  1.44270 0  1.44270  0
media 5  1  1
media 3  1  2  -1
media 2  1  3  -2
media 4  1  4  -3
boundary 4
' Full instrumentation tube
unit 5
cylinder 1  0.56005  origin x=0.72135 y=0.72135
cylinder 2  0.62610  origin x=0.72135 y=0.72135
cuboid   3  1.44270 0  1.44270  0
media 4  1  1
media 2  1  2  -1
media 4  1  3  -2
boundary 3
' Quarter instrumentation tube
unit 6
cylinder 1  0.56005 origin x=0.72135 y=0.72135 chord +x=0.72135 chord +y=0.72135
cylinder 2  0.62610 origin x=0.72135 y=0.72135 chord +x=0.72135 chord +y=0.72135
cuboid   3  1.4427  0.72135  1.4427  0.72135
media 4  1  1
media 2  1  2  -1
media 4  1  3  -2
boundary 3
' Half fuel pin cell bottom
unit 7
```

```
cylinder 1  0.46955 origin x=0.72135 y=0.72135  chord +y=0.72135
cylinder 2  0.47910 origin x=0.72135 y=0.72135  chord +y=0.72135
cylinder 3  0.54640 origin x=0.72135 y=0.72135  chord +y=0.72135
cuboid   4  1.44270 0  1.44270  0.72135
media 1  1  1
media 3  1  2  -1
media 2  1  3  -2
media 4  1  4  -3
boundary 4
' Half fuel pin cell left
unit 8
cylinder 1  0.46955  origin x=0.72135 y=0.72135  chord +x=0.72135
cylinder 2  0.47910  origin x=0.72135 y=0.72135  chord +x=0.72135
cylinder 3  0.54640  origin x=0.72135 y=0.72135  chord +x=0.72135
cuboid   4  1.4427  0.72135  1.4427  0
media 1  1  1
media 3  1  2  -1
media 2  1  3  -2
media 4  1  4  -3
boundary 4
' Half instrumentation tube left
unit 9
cylinder 1  0.56005  origin x=0.72135 y=0.72135   chord +x=0.72135
cylinder 2  0.62610  origin x=0.72135 y=0.72135   chord +x=0.72135
cuboid   3  1.4427  0.72135  1.4427  0
media 4  1  1
media 2  1  2  -1
media 4  1  3  -2
boundary 3
' Half instrumentation tube bottom
unit 10
cylinder 1  0.56005  origin x=0.72135 y=0.72135  chord +y=0.72135
cylinder 2  0.62610  origin x=0.72135 y=0.72135  chord +y=0.72135
cuboid   3  1.44270  0  1.44270  0.72135
media 4  1  1
media 2  1  2  -1
media 4  1  3  -2
boundary 3
' Central 1/4 rodded assembly
unit 11
cuboid   1  10.82025   0   10.82025   0
array    1  1   place 1 1 -0.72135 -0.72135
media 4  1  1
boundary 1
' Left-side 1/2 assembly unrodded
unit 12
cuboid   1  10.82025   0   21.6405   0
array    2  1   place 1 1 -0.72135 0
media 4  1  1
boundary 1
' Bottom-side 1/2 assembly unrodded
unit 13
cuboid   1  21.6405   0   10.82025   0
array    3  1   place 1 1  0 -0.72135
media 4  1  1
boundary 1
' Top corner full assembly
unit 14
cuboid   1  21.6405  0  21.6405  0
array    4  1   place  1  1  0  0
media 4  1  1
boundary 1
global unit 100
cuboid   1  32.46075  0  32.46075  0
array    5  1   place  1  1  0.0  0.0
media 4  1  1
boundary 1  180 180
end geom
read array
ara=1  nux=8  nuy=8  typ=square  pinpow=yes
fill
' 8 1 1 1 1 1 1 1
' 8 1 1 1 1 1 4 1
' 8 1 3 1 1 1 1 1
' 8 1 1 1 3 1 1 1
' 8 1 1 1 1 1 1 1
```

```
' 8 1 3 1 1 3 1 1
' 8 1 1 1 1 1 1 1
' 6 7 7 7 7 7 7 7
6 7 7 7 7 7 7 7
8 1 1 1 1 1 1 1
8 1 3 1 1 3 1 1
8 1 1 1 1 1 1 1
8 1 1 1 3 1 1 1
8 1 3 1 1 1 1 1
8 1 1 1 1 1 4 1
8 1 1 1 1 1 1 1
end fill
ara=2  nux=8  nuy=15 typ=square  pinpow=yes
fill
' 8 1 1 1 1 1 1 1
' 8 1 1 1 1 1 4 1
' 8 1 2 1 1 1 1 1
' 8 1 1 1 2 1 1 1
' 8 1 1 1 1 1 1 1
' 8 1 2 1 1 2 1 1
' 8 1 1 1 1 1 1 1
' 9 1 1 1 1 1 1 1
' 8 1 1 1 1 1 1 1
' 8 1 2 1 1 2 1 1
' 8 1 1 1 1 1 1 1
' 8 1 1 1 2 1 1 1
' 8 1 2 1 1 1 1 1
' 8 1 1 1 1 1 4 1
' 8 1 1 1 1 1 1 1
8 1 1 1 1 1 1 1
8 1 1 1 1 1 4 1
8 1 2 1 1 1 1 1
8 1 1 1 2 1 1 1
8 1 1 1 1 1 1 1
8 1 2 1 1 2 1 1
8 1 1 1 1 1 1 1
9 1 1 1 1 1 1 1
8 1 1 1 1 1 1 1
8 1 2 1 1 2 1 1
8 1 1 1 1 1 1 1
8 1 1 1 2 1 1 1
8 1 2 1 1 1 1 1
8 1 1 1 1 1 4 1
8 1 1 1 1 1 1 1
end fill
ara=3  nux=15  nuy=8  typ=square   pinpow=yes
fill
' 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
' 1  4  1  1  1  1  1  1  1  1  1  1  1  4  1
' 1  1  1  1  1  2  1  1  1  2  1  1  1  1  1
' 1  1  1  2  1  1  1  1  1  1  1  2  1  1  1
' 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
' 1  1  2  1  1  2  1  1  1  2  1  1  2  1  1
' 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
' 7  7  7  7  7  7  7  10 7  7  7  7  7  7  7
7  7  7  7  7  7  7  10 7  7  7  7  7  7  7
1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
1  1  2  1  1  2  1  1  1  2  1  1  2  1  1
1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
1  1  1  2  1  1  1  1  1  1  1  2  1  1  1
1  1  1  1  1  2  1  1  1  2  1  1  1  1  1
1  4  1  1  1  1  1  1  1  1  1  1  1  4  1
1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
end fill
ara=4  nux=15  nuy=15  typ=square  pinpow=yes
fill
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 4 1 1 1 1 1 1 1 1 1 1 1 4 1
1 1 1 1 1 2 1 1 1 2 1 1 1 1 1
1 1 1 2 1 1 1 1 1 1 1 2 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 2 1 1 2 1 1 1 2 1 1 2 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 5 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 2 1 1 2 1 1 1 2 1 1 2 1 1
```

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 2 1 1 1 1 1 1 2 1 1 1
1 1 1 1 1 2 1 1 1 2 1 1 1 1 1
1 4 1 1 1 1 1 1 1 1 1 1 1 4 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
end fill
ara=5  nux=2  nuy=2 typ=square   pinpow=yes
fill
' 12  14
' 11  13
11  13
12  14
end fill
end array
read bnds
    all=mirror
end bnds
end model
end
```

Python Run Script (GPT-Free):

```
#! Sequence Run for CLAROL+ and SCALE6
#!
#! ARG1 = Job ID
#! ARG2 = Number of Perturbations
#! ARG3 = Job Settings Input File
#! ARG4 = SCALE Input File
#! ARG5 = QUEUE Run
#! ARG6 = QUEUE Time

import os, sys, random, time

random.seed()

NJ_S = sys.argv[1]
NP_S = sys.argv[2]
JIFILE = sys.argv[3]
RIFILE = sys.argv[4]
QUEUEGO = sys.argv[5]
QUEUETIME = sys.argv[6]

NJ = int(NJ_S)
NP = int(NP_S)

print 'Beginning Job: ' + NJ_S + ' and Pert: ' + NP_S
print 'Job Input File: ' + JIFILE
print 'SCALE Input File: ' + RIFILE
print 'Queue: ' + QUEUEGO + '    with ' + QUEUETIME + ' minutes reserved per processor'

#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
#! Read the Input File
print 'Reading Input File: ' + JIFILE
f = open(JIFILE, 'r')
RFILE = f.readlines()
f.close()

#! Groups, Num Nuclides, Num Reactions, Perturbation Mode, Perturbation Magnitude, Library File Folder
NG = RFILE[0].split()[0]
NMIX = RFILE[0].split()[1]
NNUC = int(RFILE[0].split()[2])
NRXN = int(RFILE[0].split()[3])
PMODE = int(RFILE[0].split()[4])
PMAG = float(RFILE[0].split()[5])

Nlist = list()
NPlist = list()
Rlist = list()
Mlist = list()

for x in range(1,int(NMIX)+1):
        Mlist.append(int(RFILE[x]))

for x in range(1,int(NNUC)+1):
        NPlist.append(int(RFILE[x+int(NMIX)]))

for x in range(0, int(NNUC)):
        for y in range(0, int(NMIX)):
                z = 1000000 * Mlist[y] + NPlist[x]
                Nlist.append(z)

for x in range(1, int(NRXN)+1):
        Rlist.append(int(RFILE[int(NNUC)+x+int(NMIX)]))

#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
#! Prepare Directories
print 'Preparing Directories'
for x in range(1,NP+1):
        os.mkdir('tmpdir/J' + NJ_S + 'x' + str(x))

#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
#! Prepare XS Perturbations
print 'Preparing XS Perturbation Files'
for x in range(1,NP+1):
        for y in range(0,int(NNUC)):
                sISO = str(NPlist[y]).rjust(5,'0')
```

```
                for z in range(0,int(NRXN)):
                        sMT = str(Rlist[z]).rjust(4,'0')
                        sFull = 'pdata/param_' + sISO + '_' + sMT + '_' + NJ_S + '_' + str(x)
                        f = open(sFull, 'w')
                        #! print 'Writing file: ' + sFull
                        f.write('1'.rjust(12) + str(NG).rjust(12))
                        for ig in range(0,int(NG)):
                                if (PMODE == 1):
                                        #! Gaussian Random Number
                                        eta = random.gauss(0,PMAG)
                                elif (PMODE == 0):
                                        #! Uniform Distribution
                                        eta = (2.0 * random.random() - 1.0) * PMAG
                                else:
                                        sys.exit("Perturbation mode not recognized!!!")
                                        eta = 0.0
                                #! Write Value out to the file
                                sData = '{0: 14.6E}'.format(eta)
                                f.write(sData)
                        f.write('\n')
                        f.close()

#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
#! Preparing Scale Run Files
print 'Copying SCALE Perturbation Files'
for x in range(1,NP+1):
        for y in range(0,int(NNUC)):
                sISO = str(NPlist[y]).rjust(5,'0')
                for z in range(0,int(NRXN)):
                        sMT = str(Rlist[z]).rjust(4,'0')
                        sFull = 'pdata/param_' + sISO + '_' + sMT + '_' + NJ_S + '_' + str(x)
                        sTarget = 'tmpdir/J' + NJ_S + 'x' + str(x) + '/param_' + sISO + '_' + sMT + '_1'
                        os.system('cp ' + sFull + ' ' + sTarget)

#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
#! Preparing Scale INPUT File
print 'Preparing SCALE input files...'
for x in range(1,NP+1):
        INJECT = '=shell\n'
        for y in range(0,int(NNUC)):
                sISO = str(NPlist[y]).rjust(5,'0')
                for z in range(0,int(NRXN)):
                        sMT = str(Rlist[z]).rjust(4,'0')
                        sFULL = 'param_' + sISO + '_' + sMT + '_1'
                        INJECT = INJECT + 'cp ${RTNDIR}/' + sFULL + ' ' + sFULL + '\n'
        INJECT = INJECT + 'cp /abdelkhalik_data/cbkenned/PROJECTS/DEPL/FSA/data/ft11f001 ft11f001\n'
        INJECT = INJECT + 'cp /abdelkhalik_data/cbkenned/PROJECTS/DEPL/FSA/data/ft47f001 ft47f001\n'
        INJECT = INJECT + 'end\n'
        INJECT = INJECT + '=readparam_master\n'
        INJECT = INJECT + str(NNUC*NRXN).rjust(12) + str(NG).rjust(12) + '\n'
        for y in range(0,int(NNUC)):
                sISO = str(NPlist[y]).rjust(5,'0')
                for z in range(0,int(NRXN)):
                        sMT = str(Rlist[z]).rjust(4,'0')
                        sFULL = ' param_' + sISO + '_' + sMT + '_1'
                        INJECT = INJECT + sFULL + '\n'
        INJECT = INJECT + 'end\n'
        INJECT = INJECT + '=clarolplus_master\n'
        INJECT = INJECT + 'in=11 var=10 sam=1 out=99\n'
        INJECT = INJECT + 'end\n'
        INJECT = INJECT + 'end\n'
        INJECT = INJECT + '=shell\n'
        INJECT = INJECT + 'mv ft99f001 pxn44\n'
        INJECT = INJECT + 'mv ft11f001 oxn44\n'
        INJECT = INJECT + 'end\n'

        f = open('data/ref.input','r')
        FFILE = f.read()
        f.close()

        FFILE = FFILE.replace('%%INJECT%%',INJECT)

        g = open('tmpdir/J' + NJ_S + 'x' + str(x) + '/' + RIFILE, 'w')
        g.write(FFILE)
        g.close()
```

```
#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
#! Preparing Scale Run Files
print 'Preparing Scale HPC Files (BSUB and RUN Command)'
f = open('data/scale_run','r')
SC_IFILE = f.read()
f.close()

for x in range(1,NP+1):
     g = open('tmpdir/J' + NJ_S + 'x' + str(x) + '/scale_run', 'w')
     SC_OFILE = SC_IFILE.replace('%%FF%%', RIFILE)
     SC_OFILE = SC_OFILE.replace('%%P' , str(x))
     SC_OFILE = SC_OFILE.replace('%%J' , NJ_S)
     g.write(SC_OFILE)
     g.close()

f = open('data/bsub_scale', 'r')
BS_IFILE = f.read()
f.close()

for x in range(1, NP+1):
     g = open('tmpdir/J' + NJ_S + 'x' + str(x) + '/bsub_scale', 'w')
     BS_OFILE = BS_IFILE.replace('%%QQ%%', QUEUEGO)
     BS_OFILE = BS_OFILE.replace('%%QT%%', QUEUETIME)
     BS_OFILE = BS_OFILE.replace('%%P' , str(x))
     BS_OFILE = BS_OFILE.replace('%%J' , NJ_S)
     g.write(BS_OFILE)
     g.close()

#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
#! Wait 5 Seconds for HDD to catch up
print 'Sleeping for 3 seconds to let disks catch up'
time.sleep(3)

#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
#! Run SCALE
print 'Running Scale'
for x in range(1, NP+1):
     os.chdir('tmpdir/J' + NJ_S + 'x' + str(x) + '/')
     os.system('chmod +x scale_run')
     os.system('bsub < bsub_scale')
     os.chdir('../..')

print 'Script Complete'
```

Python NEWT-Binary Depletion Flux HPC Monitor Scripts:

Depletion calculations using T-DEPL overwrite the binary flux file at each depletion time-step. A monitoring script is required to maintain the binary file for each depletion time point. These two python scripts are included below (and must be added to the =shell section prior to the T-DEPL call in SCALE6).

Launcher.py

```
#! Monitor Launcher
import os, subprocess

FNULL = open(os.devnull,"w")
subprocess.Popen(["python","monitor.py","15"], stderr=FNULL)

print '\n  Launching Flux Monitor'
```

Monitor.py

```
#! Monitors SCALE depletion and maintains a record of NEWT Binary FLUX
Output
import time, sys, os

NT = int(sys.argv[1])

zi = 0
#! Look for first instance

while (zi == 0):
      time.sleep(10)
      zres = os.popen("ls --time-style='+%H:%M:%S' -l ft31f001").read()
      if ('\n' in zres):
            zi = zi + 1
            zts = zres.split()[5]
            #! Time in Seconds
            zt = int(zts.split(':')[0])*3600 + int(zts.split(':')[1])*60 +
int(zts.split(':')[2])

print '\n  Grabbing Newt flux output'
os.system('cp ft31f001 ft31f001_1')

while (zi < NT):
      time.sleep(10)
      zres = os.popen("ls --time-style='+%H:%M:%S' -l ft31f001").read()

      zts = zres.split()[5]
      zt_test = int(zts.split(':')[0])*3600 + int(zts.split(':')[1])*60 +
int(zts.split(':')[2])

      if (zt_test > zt):
            time.sleep(2)
            zi = zi + 1
            zt = zt_test
```

```
            os.system('cp ft31f001 ft31f001_' + str(zi))
            print '\n  Grabbing Newt flux output'

#! Monitoring Complete
for x in range(0,NT):
      os.system('cp ft31f001_' + str(x+1) + ' ${RTNDIR}/')
```

FExtract.F90

```
PROGRAM FExtract
! Program designed to read the NEWT 2D Scalar Flux Output from the Binary
File: ft31f001
! Chris Kennedy - NCSU
!
! Relevant Information extracted from VIPNEWT_I.f90, VIP_DATA.f90,
ebindump2d.f90

IMPLICIT NONE

! Precision Flags for SAMS/NEWT
INTEGER,PARAMETER :: isp = selected_int_kind(9)
INTEGER,PARAMETER :: rsp = selected_real_kind(6)
INTEGER,PARAMETER :: rdp = selected_real_kind(14)

! Record 1
INTEGER :: izm, im, mxx, ms, isct, mm, nom, igm
!izm # of zones (NEWT variable numzone)
!im # of mesh intervals (NEWT variable numcells)
!mxx # of mixtures (NEWT variable nummat)
!ms # of isotopes (NEWT variable nmix)
!isct scattering order (NEWT variable maxpn)
!mm # angles in quadrature (NEWT variable ndir)
!nom # of moments (NEWT variable nmom)
!igm # of groups (NEWT variable numen)

! Record 2
INTEGER :: ige, ibl, ibr, ibb, ibt, isn, iftg, mmt, nt1
! ige = 0 (Geometry Term)
! ibl - Left Boundary
! ibr - Right Boundary
! ibb - Bottom Boundardy
! ibt - Top Boundary
! isn - S(N) Quadrature Order
! iftg - First THermal Group
! mmt - # of Neutron Groups
! nt1 - Working Library FT #
CHARACTER(len=80) problem_title

! Record 3
REAL(rdp), ALLOCATABLE, DIMENSION(:) :: vol
! VOL Volume of each Mesh Intervale

! Record 4
REAL(8), ALLOCATABLE ::  weight(:)
REAL(8), ALLOCATABLE :: Ye(:,:)
! weight - Angular Quadrature Weights
! Ye - Scattering Constants for Flux Moments
! Ye(i,j),i=0:nmom,j=1:ndir, Y(0,j) = 1 (always) and is not in the file.

! Record 5
INTEGER(isp), ALLOCATABLE :: ma(:)
```

```
INTEGER, ALLOCATABLE :: mat(:)
INTEGER, ALLOCATABLE :: matndx(:)
! ma (Zone by Mesh) (i=1..NCELLs)
! mat(:) Mixture by Zone (i=1..NZONE)
! matndx(:) Mixture ID by Zone (i=1..NZONE)

! Record 6
INTEGER, ALLOCATABLE :: mix(:)
INTEGER, ALLOCATABLE :: nuc(:)
INTEGER, ALLOCATABLE :: matpn(:)
REAL, ALLOCATABLE :: den(:)
! mix(i),i=1,nmix - mixture number in the mixing table (integer)
! nuc(i),i=1,nmix - nuclide id in the mixing table (integer)
! matpn(i),i=1,nummat - the order of the Legendre polynomial being used
(integer)
! den(i),i=1,nmix - atom density in mixing table (single precision)

! Record 7
INTEGER, ALLOCATABLE, DIMENSION(:) :: matid
REAL(rsp), ALLOCATABLE, DIMENSION(:,:) :: chi, fisnu
! matid(i),i=1,nummat - material numbers for each mixture (integer)
! chi(i,j),j=1,numen, i=1,nummat - chi for each mixture (single precision)
! fisnu(i,j),j=1,numen, i=1,nummat - nubar times fission x-sect for each
mixture (single precision)

! Record 8
REAL(rdp) :: eigenf    ! Eigenvalue

! Record 9
REAL(rdp), ALLOCATABLE, DIMENSION(:,:,:) :: newt_flux
! newt_flux(i,j,k) = Cell(i), Moment(j), Group(k).  One record for each
energy group.
! Note that moment 0 = Scalar flux (but needs to be transformed with
Volume.

! Remaining Calculations
REAL(rdp), ALLOCATABLE, DIMENSION(:,:) :: volume_flux ! Volumetric Flux by
Zone and Energy Group
REAL(rdp), ALLOCATABLE, DIMENSION(:,:) :: scalar_flux ! Scalar Flux by
Mixture and Energy Group
INTEGER :: ix_g, ix_i, ix_j, ix_k ! Loop Variables

CHARACTER(4) :: FC

! ************************************************************
WRITE(*,*) "Extracting NEWT Flux Data..."

! Open File
OPEN(UNIT=31, FILE='ft31f001', STATUS='old', FORM='unformatted')

! Read Record 1 ==============================
READ(31) izm, im, mxx, ms, isct, mm, nom, igm
!izm # of zones (NEWT variable numzone)
```

```
!im # of mesh intervals (NEWT variable numcells)
!mxx # of mixtures (NEWT variable nummat)
!ms # of isotopes (NEWT variable nmix)
!isct scattering order (NEWT variable maxpn)
!mm # angles in quadrature (NEWT variable ndir)
!nom # of moments (NEWT variable nmom)
!igm # of groups (NEWT variable numen)

! Allocate Variables ************************
ALLOCATE ( weight(mm) )
weight = 0

ALLOCATE (matid(mxx) )
matid = 0

ALLOCATE ( matpn(mxx) )
matpn = 0

ALLOCATE ( ma(im) )
ma = 0

ALLOCATE ( mat(izm) )
mat = 0

ALLOCATE ( matndx(izm) )
matndx = 0

! ALLOCATE ( mb(ms) )
! mb = 0

ALLOCATE ( mix(ms) )
mix = 0

ALLOCATE ( nuc(ms) )
nuc = 0

ALLOCATE ( den(ms) )
den = 0

ALLOCATE ( chi(igm,mxx) )
chi = 0

ALLOCATE ( fisnu(igm,mxx) )
fisnu = 0

ALLOCATE ( vol(im) )
vol = 0

ALLOCATE ( Ye(0:nom,mm) )
Ye = 0

ALLOCATE ( newt_flux(im,0:nom,igm))
newt_flux = 0
```

```
ALLOCATE ( volume_flux(igm,izm) )
volume_flux = 0

ALLOCATE ( scalar_flux(mxx,igm) )
scalar_flux = 0

! Read Record 2 ============
READ(31) ige, ibl, ibr, ibb, ibt, isn, iftg, mmt, nt1, problem_title
! See above

! Read Record 3 ============
READ(31) vol
! vol(i) Volume by mesh cell, i=1..ncells

! Read Record 4 ============
READ(31) weight, Ye
! weight - Angular Quadrature Weights
! Ye - Scattering Constants for Flux Moments
! Ye(i,j),i=0:nmom,j=1:ndir, Y(0,j) = 1 (always) and is not in the file.

! Read Record 5 ============
READ(31) ma, mat, matndx
! ma = zone by mesh (i=1..ncells)
! mat = mixture by zone (i=1..nzones)
! matndx = mixture index by zone (i=1..nzones)

! Read Record 6 ============
READ(31) mix,nuc,matpn,den
! mix(i),i=1,nmix - mixture number in the mixing table (integer)
! nuc(i),i=1,nmix - nuclide id in the mixing table (integer)
! matpn(i),i=1,nummat - the order of the Legendre polynomial being used
(integer)
! den(i),i=1,nmix - atom density in mixing table (single precision)

! Read Record 7 ============
READ(31) matid, chi, fisnu
! matid(i),i=1,nummat - material numbers for each mixture (integer)
! chi(i,j),j=1,numen, i=1,nummat - chi for each mixture (single precision)
! sigfnu(i,j),j=1,numen, i=1,nummat - nubar times fission x-sect for each
mixture (single precision)

! Read Record 8 ============
READ(31) eigenf

! Read Records 9 to 8 + #_ENERGY_GROUPS ============
DO ix_g = 1, igm
      READ(31) newt_flux(:,:,ix_g)
      ! Loop over Mesh Cells
      DO ix_i = 1, im
            ! Get Zone
            ix_j = ma(ix_i)
            ! Calculate Volumetric Flux
```

```
            volume_flux(ix_g, ix_j) = volume_flux(ix_g, ix_j) +
newt_flux(ix_i,0,ix_g)*vol(ix_i)
      ENDDO
ENDDO

! Close File
CLOSE(31)

! Calculate Scalar Flux
DO ix_i = 1, mxx
      DO ix_g = 1, igm
            DO ix_j = 1, im
                  ix_k = ma(ix_j)

                  ! The first mixture is the global mixture
                  IF (ix_i == 1) THEN
                      scalar_flux(ix_i,ix_g) = scalar_flux(ix_i,ix_g) +
newt_flux(ix_j,0,ix_g)*vol(ix_j)

                  ELSE

                      IF (matndx(ix_k) /= ix_i) cycle

                      scalar_flux(ix_i,ix_g) = scalar_flux(ix_i,ix_g) +
newt_flux(ix_j,0, ix_g)*vol(ix_j)
                ENDIF
            ENDDO
      ENDDO
 ENDDO

OPEN(UNIT=32, FILE='FData.out', STATUS='replace')

WRITE(32,'(A20, 1X, F20.14)') 'K-eff: ', eigenf
WRITE(32,*) "Number of Mixtures: ", mxx
WRITE(32,*) "Scalar Flux by Mixture and Energy Group"
WRITE(FC,'(I4)') mxx
WRITE(32,'(A5,' // FC // 'I24)' ) "Group", (ix_i, ix_i=1,mxx)
DO ix_g = 1, igm
       WRITE(32,'(I5, ' // FC // 'E24.14)') ix_g, (scalar_flux(ix_i,
ix_g), ix_i=1,mxx)
ENDDO

CLOSE(32)
END PROGRAM
```

SDFExtract.F90

```
PROGRAM gSDF

! Reads Deterministic SCALE SDF Files

IMPLICIT NONE

! Input Listings

INTEGER :: NJ
CHARACTER(4) :: NJ_STR
INTEGER :: NP
CHARACTER(4) :: NP_STR
CHARACTER(4) :: CPS

INTEGER :: ISO_NUM
INTEGER :: RXN_NUM
INTEGER :: NG
INTEGER :: NF
INTEGER :: NF_RI

INTEGER :: NG_B             ! Group Boundaries
INTEGER :: NG_E             ! Number of Full Line Entries
INTEGER :: NG_LE            ! Number of Last Line Entries

INTEGER :: SMODE

! Temporary Data Storage
CHARACTER(80) :: EMPTY
CHARACTER(20) :: TN1, TN2
CHARACTER(20) :: FILENAME
CHARACTER(20) :: FILEROOT
CHARACTER(20) :: PFILE
CHARACTER(4) :: isz

CHARACTER(32) :: inarg

REAL(KIND=8) :: KFOR                                    ! K-effective
REAL(KIND=8), DIMENSION(:), ALLOCATABLE :: EBOUNDS      ! Energy Group
Bounds

REAL(KIND=8), DIMENSION(:,:), ALLOCATABLE :: S_INT_DATA              !
Energy Integrated Sensitivity Data
REAL(KIND=8), DIMENSION(:), ALLOCATABLE :: S_REG_DATA             !
Region Data Contants
REAL(KIND=8), DIMENSION(:,:), ALLOCATABLE :: S_SEN_DATA              !
GroupWise Sensitivity Data
REAL(KIND=8), DIMENSION(:,:), ALLOCATABLE :: S_SEN_SIGMA
                  ! GroupWise Uncertainty Data
INTEGER, DIMENSION(:,:), ALLOCATABLE :: S_ISO_DATA                   !
Isotope/MT Data

INTEGER :: i1, i2, i3, i4, i5, is
```

```
! ************************************************************
! Command Line Parameters

! Get Input File
CALL getarg(1, PFILE)
is = index(PFILE, '.inp', .FALSE.)
WRITE(isz, '(I3)') is-1
READ(PFILE, '(A' // isz // ')') FILEROOT

CALL getarg(2, inarg)
NJ_STR = inarg
READ(NJ_STR, *) NJ

! Here NP is the current Perturbation
CALL getarg(3, inarg)
NP_STR = inarg
READ(NP_STR, *) NP

! Mode = 0 implies Deterministic, Mode = 1 implies KENO (Uncertainties)
CALL getarg(4, inarg)
READ(inarg, *) SMODE

! Prepare File
FILENAME = TRIM(ADJUSTL(FILEROOT)) // '.sdf'
OPEN (UNIT=(42), FILE=FILENAME)

! File Header
READ(42, *) EMPTY
READ(42, *) NG, EMPTY
READ(42, '(I10,A35,I10)') NF, EMPTY, NF_RI
READ(42, *) KFOR

! Initialize
NG_B = NG + 1
NG_LE = MOD(NG_B, 5)
NG_E = NG_B - NG_LE

! Allocations
ALLOCATE(EBOUNDS(NG_B))
ALLOCATE(S_INT_DATA(NF,5))
ALLOCATE(S_REG_DATA(NF))
ALLOCATE(S_ISO_DATA(NF,4))
ALLOCATE(S_SEN_DATA(NF,NG))
ALLOCATE(S_SEN_SIGMA(NF,NG))

! Energy Boundaries
READ(42, *) EMPTY
DO i1 = 1, NG_E / 5
    READ(42, *) (EBOUNDS(5*(i1-1)+i2), i2=1,5)
ENDDO

IF (NG_LE > 0) THEN
```

```
    READ(42, *) (EBOUNDS(NG_E + i2), i2=1,NG_LE)
ENDIF

! Sensitivity Data Init
NG_LE = MOD(NG, 5)
NG_E = NG - NG_LE

IF (SMODE < 1) THEN

    ! Sensitivity Data (Deterministic)
    DO i1 = 1, NF
        READ(42,'(A12, 1x, A14, 1x, 3I7, ES14.0)') TN1, TN2,
S_ISO_DATA(i1,1), S_ISO_DATA(i1,2), S_ISO_DATA(i1,3), S_REG_DATA(i1)
        READ(42,*) S_INT_DATA(i1, 1), S_INT_DATA(i1, 2), S_INT_DATA(i1, 3)
        DO i2 = 1, NG_E / 5
            READ(42, *) (S_SEN_DATA(i1, 5*(i2-1)+i3), i3=1,5)
        ENDDO
        IF (NG_LE > 0) THEN
            READ(42, *) (S_SEN_DATA(i1, NG_E + i3), i3=1, NG_LE)
        ENDIF
    ENDDO

ELSE

    ! Sensitivity Data (KENO)

    DO i1 = 1, NF
        READ(42,'(A12, 1x, A14, 1x, 2I7)') TN1, TN2, S_ISO_DATA(i1,1),
S_ISO_DATA(i1,2)
        READ(42,*) S_ISO_DATA(i1,3), S_ISO_DATA(i1,4)
        READ(42,*) S_REG_DATA(i1)
        READ(42,*) S_INT_DATA(i1,1), S_INT_DATA(i1,4), S_INT_DATA(i1,2),
S_INT_DATA(i1,3), S_INT_DATA(i1,5)
        DO i2 = 1, NG_E / 5
            READ(42, *) (S_SEN_DATA(i1, 5*(i2-1)+i3), i3=1,5)
        ENDDO
        IF (NG_LE > 0) THEN
            READ(42, *) (S_SEN_DATA(i1, NG_E + i3), i3=1, NG_LE)
        ENDIF
        DO i2 = 1, NG_E / 5
            READ(42, *) (S_SEN_SIGMA(i1, 5*(i2-1)+i3), i3=1,5)
        ENDDO
        IF (NG_LE > 0) THEN
            READ(42, *) (S_SEN_SIGMA(i1, NG_E + i3), i3=1, NG_LE)
        ENDIF
    ENDDO


END IF

! Close File
CLOSE(42)
```

```
! ********************************************************
! Output Data to File

FILENAME = 's' // TRIM(ADJUSTL(NJ_STR)) // 'x' // TRIM(ADJUSTL(NP_STR)) //
'.out'
OPEN(UNIT=40, FILE=FILENAME)

WRITE(40, *) NG, NF

WRITE(40, '(A6,A8,A6,A5,A14, A14)') 'S#', 'ISOTOPE', 'MT', 'GRP', 'Region
Data', 'SensData'
DO i1 = 1, NF
    DO i2 = 1, NG
        WRITE(40, '(I6, I8, I6, I5, 1X, I12, 1X, ES14.6)') i1,
S_ISO_DATA(i1, 1), S_ISO_DATA(i1, 2), i2, abs(S_ISO_DATA(i1,3)),
S_SEN_DATA(i1, i2)
    ENDDO
ENDDO

CLOSE(40)

END program
```

Sort.py

```
#! Utility Sorts output XS Files to be consistent with SDF Files
import numpy as np
import sys

NJ_S = sys.argv[1]

XSP0 = np.loadtxt('XSP_' + NJ_S + '.dat')
s = np.lexsort((XSP0[:,2],XSP0[:,1],XSP0[:,0]))
XSP = XSP0[s]
l = len(XSP0[0])
np.savetxt('XSPS_' + NJ_S + '.dat', XSP, fmt=(3 * '%i ') + ((l-3) *
'%16.10e '))
```