UNCERTAINTY QUANTIFICATION OF A GENETIC ALGORITHM FOR NEUTRON ENERGY SPECTRUM ADJUSTMENT

A Thesis

by

DANIELLE R. REDHOUSE

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

MASTER OF SCIENCE

Chair of Committee,	Pavel V. Tsvetkov
Co-Chair of Committee,	Ryan G. McClarren
Committee Members,	Charles M. Folden III
	John R. Ford
Head of Department,	Yassin A. Hassan

August 2017

Major Subject: Nuclear Engineering

ABSTRACT

GenSpec is software designed to use a genetic algorithm for neutron energy spectrum adjustment. Currently GenSpec can produce adjusted spectra, but the corresponding covariance matrix is not produced. The uncertainty quantification process implemented includes a parametric sensitivity analysis of the genetic algorithm modifiers for population, generations, gene-sites, polynomial order, and mutation rate. A random perturbation analysis was used to characterize the covariance of the genetic algorithm using multivariate normal random sampling of the characterized input data. The produced 640 by 640 covariance matrix has retained some characteristic features of the sampled covariance. The uncertainty found in the GenSpec program has minimized the covariance present in a calculated trial spectrum.

DEDICATION

I dedicate this thesis to Deborah, Howard, Theala, and Yvette for their unlimited love, support, and laughs.

ACKNOWLEDGMENTS

Additional thanks to my advisors Pavel V. Tsvetkov and Ryan G. McClarren for taking on this project with me. Recognition is extended to both Richard M. Vega for guidance and insight into the GenSpec program and Edward J. Parma for being an outstanding mentor.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Assistant Professor Ryan G. McClarren, Associate Professor Pavel V. Tsvetkov, and Associate Professor John R. Ford of the Department of Nuclear Engineering with Associate Professor Charles M. Folden III of the Department of the Dean of Science.

The experimental analyses depicted in Chapter 3.1 was conducted with the student in conjunction with the Sandia National Laboratories Annular Core Research Reactor (ACRR) Facilities Operators and Laboratory Support Technologists, as well as the Radiation Metrology Laboratory (RML) Scientists and Engineers.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

Graduate study and thesis was supported by Sandia National Laboratories through the Master's Fellowship Program. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

NOMENCLATURE

ACRR	Annular Core Research Reactor
α	alpha particle
DOE	Department of Energy
ENDF	Evaluated Nuclear Data File
eV	electron volt
FREC-II	Fueled Ring External Cavity
γ	gamma particle
GUI	Graphical User Interface
IRDFF	International Reactor Dosimetry Fusion and Fission Nuclear Database
keV	kilo-electron volt
LANL	Los Alamos National Laboratories
MeV	Mega-electron volt
MJ	Megajoule
MW	Megawatt
MCNP	Monte Carlo N-Particle
n	neutron
NAA	neutron activation analysis
QoI	Quantity of Interest
p	proton
SNL	Sandia National Laboratories

UQ	uncertainty quantification
UUR	Unclassified Unlimited Release

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

1. INTRODUCTION

Implementation of uncertainty quantification (UQ) in GenSpec is essential to the completion of the neutron energy spectrum adjustment code. While GenSpec can produce adjusted neutron energy spectra, the question of how uncertainties propagate through the code becomes important in interpreting the merit of the generated data. Therefore, as it stands, GenSpec is an incomplete program without uncertainty quantification and applicable covariance data. Adjustment in GenSpec is a complex process that uses a genetic algorithm to minimize the differences between calculated and measured reaction probabilities. GenSpec relies on neutron activation analysis (NAA) and a trial spectrum that is calculated using a transport code. However, it is not known how if the adjusted spectra have minimized errors, as are expected.

1.1 Background

Prior to introducing the methods for implementing UQ in GenSpec, it is necessary to introduce and explain the concepts of neutron energy spectra, spectrum measurement, spectrum adjustment, GenSpec, and UQ.

1.1.1 Neutron Energy Spectra

When performing a neutron irradiation experiment in a research reactor, experimenters need to know the energy spectrum of the neutron fluence at any given location. Usually this spectrum is desired to a high resolution with associated uncertainty [1]. While calculating a spectrum is a relatively straightforward process, physically measuring a spectrum can only be performed at a low, and often insufficient, resolution. Most often the spectrum is used with a response function in order to calculate integral quantities like displacements per atom, absorbed dose, or fluence. To obtain these quantities, the spectrum is “folded"

with a response function; the mathematical representation can be seen below.

$$I = \int_0^\infty \Phi(E) R(E) \, \mathrm{d}E \tag{1.1}$$

In the above equation, I is the integral quantity, $\phi(E) = d\Phi/dE$ is the differential neutron energy spectrum, Φ is the fluence, and $R(E)$ is an energy-dependent response function. The greater the resolution in each of these functions the greater the accuracy that can be achieved. This is especially important in regions of quickly varying response functions or spectral shape fluctuations due to resonances.

All neutron energy spectra found in fission reactors will have some similar features, due to the nature of the fission process [2]. These features include the Watt fission peak in the fast region, and possibly the Maxwellian thermal peak, depending on the materials within the reactor and their scattering properties. The region between the two peaks is known as the slowing-down region. Throughout a fission reactor's spectrum, there are also resonances present as sharp peaks and dips. These resonance regions directly correspond to escape probabilities in the cross-section of the reactor materials. The quickly varying behavior of the resonance peaks in the neutron cross-sections makes direct measurements of neutron energy spectra at a high resolution a difficult task [2].

1.1.2 Spectrum Measurement

Measuring a neutron energy spectrum in any nuclear reactor can be hindered by many factors. The two common methods of measuring spectra are through detectors or NAA. One detector method is the use of Bonner spheres, which are low-energy neutron detectors surrounded by moderating material. Spheres of varying radii are used, with each detector representing an integral quantity. The spectrum can then be "unfolded" to provide an estimate of the neutron energy spectrum. A major drawback to using Bonner spheres for spectrum measurements is how large they can be, depending on the amount of moderating

material used to enclose the thermal detector [2].

Since the number of detectors will ultimately be limited, NAA analysis can also be used and allows for a larger number of integral quantities to be collected. The NAA method uses a wide variety of specific high-purity material foils. These foils are placed in an irradiation environment for a known period of time and are activated, forming radioactive isotopes. Using both the irradiation and decay time, the activity of the foil can be measured and the reaction-rate integral can be evaluated. Activation foils can also be used with cover materials that alter the spectrum before it reaches the foils. The most common material is cadmium, which eliminates most neutrons below 0.5 eV. This adds additional integral quantities to be evaluated within the spectrum. The process of obtaining a spectrum using only integral quantities in the form of foils or detector responses is known as spectrum "unfolding". Equation 1.1 can be approximated as a summation, shown in below.

$$I = \int_0^\infty \phi(E) R(E)\,\mathrm{d}E \approx \sum_{i=1}^{n} \phi_i R_i \Delta E_i \tag{1.2}$$

In this equation n is the number of energy groups used to represent the spectrum, ϕ_i is the differential neutron fluence in group i, R_i is the response function in group i, and ΔE_i is the energy bin width of group i. It is clear that each integral quantity provides a linear algebraic equation for the unknowns ϕ_i, if the R_i and ΔE_i's are known. The problem then, is that the number of foils or detector responses will likely always be an order of magnitude less than the number of energy groups necessary to obtain a high resolution, making it an under-determined problem. The ultimate goal of spectrum adjustment is to obtain a spectrum with higher resolution than possible through experimental measurements [1].

1.1.3 Spectrum Adjustment

Although many methods exist for spectrum adjustment, two of the most common methods are iterative perturbation and statistical least-squares estimates [3]. These two methods

differ in their outcome. Iterative perturbation relies on matching measured or calculated data, while statistical least squares aims to minimize the uncertainties in the adjusted spectrum only. Both of these methods suffer from the production of “unrealistic spectral shape artifacts [3]." Unrealistic spectral shape artifacts refer to features within the adjusted spectrum such as dips or peaks that have no scientific explanation or physical meaning, and, therefore, exist only as a result of the adjustment method.

For least squares adjustments, a significant deterioration of the spectral shape is seen in favor of a high-resolution spectrum, solely based on achieving a minimized uncertainty [3][4]. This flaw in least-squares is due to the purely mathematical solution method aimed to solve an under-determined problem. Adding more variables to the under-determined problem assumes that all input parameters are uncertain and should thus be adjusted in combination with the input spectrum. This requires knowing covariance matrices and dosimetry cross-sections for the input spectrum, which may not be known at the desired resolution. Re-binning low-resolution data will most often lead to a singular covariance value and render the equations unsolvable [4]. The greatest limitation to any method that focuses on just the reduction of the uncertainties of the adjusted spectrum is the assumption that the covariance matrix for the calculated spectrum is known with any confidence at all.

These unrealistic spectral artifacts provide the basis against focusing on solely matching the measured and calculated data without evaluating uncertainties as well. Through perturbation theory, it is possible to arrive at an unrealistic spectrum simply by ensuring that the predicted probabilities are all incorporated at a high accuracy. In particular, this is the case when the number of iterations is too high; the calculated spectrum has undergone too many perturbations and no longer resembles a familiar form. These known issues were the impetus for the development of a new adjustment code.

1.1.3.1 GenSpec

GenSpec [1] is a new spectrum adjustment code. It uses a genetic algorithm to optimize functions for the relative adjustment factor to a calculated spectrum as a function of energy over the energy range of interest. This greatly reduces the need for costly calculations. The theory behind any genetic algorithm is similar to that of the biologic process seen in human genetics. The goal is to create a method for solving both constrained and unconstrained problems based on a natural selection process that mimics biological evolution [5]. Since genetic algorithms are based on a natural process, several fundamental properties of natural selection are borrowed to solve complex problems.

To begin, most genetic algorithms use a population parameter to initializes the process. The population is either randomly selected, or user selected. Populations can range from a few individuals to a few thousand or more. Once a population is set, an evaluation process must be created to evaluate the ‘fitnessâĂŹ of each individual in the population. Fitness can often differ greatly between each genetic algorithm, but is used to calculate how well each individual fits into the desired system requirements [6]. These system requirements should be simple optimization parameters, such as a local maxima or minima [7]. Once an evaluation of the population can be done, selection of fit individuals should be done. Using a selection algorithm ensures that the populations overall fitness is constantly improved. The selection algorithms typically pass over or discard individuals that do not met certain criterion, and selects the fittest individuals more frequently for mating. ‘MatingâĂŹ, or recombination, takes genes of each individual and combines aspects of both selected individuals. To recombine genes, a crossover point in the genome is selected and swapped between individuals to create or produce offspring. This process is mimicking how reproduction works in nature. This process captures desirable traits, via fitness, from each participating parent individual that is subsequently inherited by child individuals [5].

Crossover can also be a randomly selected point, and can occur across a genome several times if needed. Another process needed to ensure this follows genetics is mutations [7]. For GenSpec, a single point crossover is used, as multiple crossover point did not increase the effectiveness of the crossover algorithm or the program solution. Mutation is a random process in a populations genetic solution. Mutations typically works by creating changes at random across an individual child's genome. Mutation also ensure that solution in the problem are all full explored within the genetic solution space [8]. Lastly, once this entire process is finished, it is repeated through a generational process. Generations are required to evolve a solution in a genetic algorithm, without an adequate number of generation solutions can be sub-optimum [6].

GenSpecs algorithms follows the process of repeatedly modifying a population of solutions. GenSpec creates a population of full energy spectra, using *a priori* characteristics to influence the formation of each initial specimen [1]. A genome or genotype of each spectrum is created by using some N number of gene-sites randomly spaced logarithmically across each spectrum. This means that solutions will be represented by a number of 'genes,' with each gene holding a specific numeric value comparable to a neutron energy. The genetic algorithm selects from the "fittest" individuals within the current population and uses them as parents to produce the children for the next generation. Fitter specimens will be chosen more often for recombination as compared to other less fit specimen. Specimen spectra that don't meet the minimum fitness in the program are eliminated from the mating pool. During recombination in GenSpec, a single point crossover is used as opposed to multiple crossover points. Multiple crossover points did not increase the effectiveness of the recombination algorithm or the program solution. Over successive generations, the population "evolves" toward an optimal solution. In this way, GenSpec behaves like perturbation theory in its aim to match measured reactions [1]. While fitness in a genetic algorithm can vary between problems, fitness in GenSpec will be described

in the Theory section of this thesis. A flow chart of GenSpec's process of optimization can be seen in Figure 1.1. GenSpec is also successful at implementing algorithms for recombination and mutation, which ensures the solution space is sampled thoroughly.

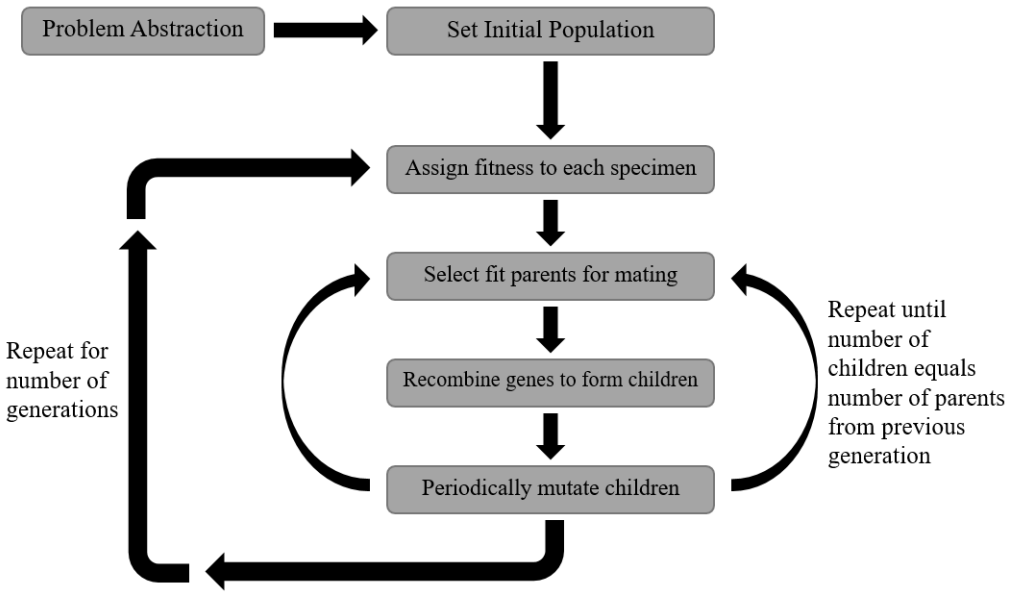

Figure 1.1: Process of a Typical Genetic Algorithm.

As previously noted, GenSpec relies on neutron activation analysis (NAA). Spectrum unfolding aims to reconstruct the spectrum based on experimental data, while adjustment aims to modify a trial spectrum that is supplied through computational means. GenSpec is designed to do both unfolding and adjustment [1]. Herein are the sources of uncertainty that need to be studied, as calculation uncertainties and adjustment uncertainties need to be propagated and assessed.

Since GenSpec's inception and creation, it has performed energy spectrum adjustments for several different environments [1]. A comparison of adjusted spectrum for both GenSpec and *LSL-M2*, a Sandia National Laboratory (SNL) code commonly used for the pur-

pose of spectrum adjustment, can be seen in Figure 1.2 [9]. In Figure 1.2, both adjustments are performed using the 89-energy group structure for *LSL-M2*(left) and GenSpec(right) for the free field environment of the central cavity of the SNL Annular Core Research Reactor (ACRR). In these cases, the y-axis is a measure of differential fluence. The differences have prompted further investigation into the validity of the adjustment that GenSpec performs. It is this initiative that drives for the further expansion of this code, and necessitates the formation of this research and thesis. Currently, the need for an addition of uncertainty quantification is required for GenSpec to be a complete software package.

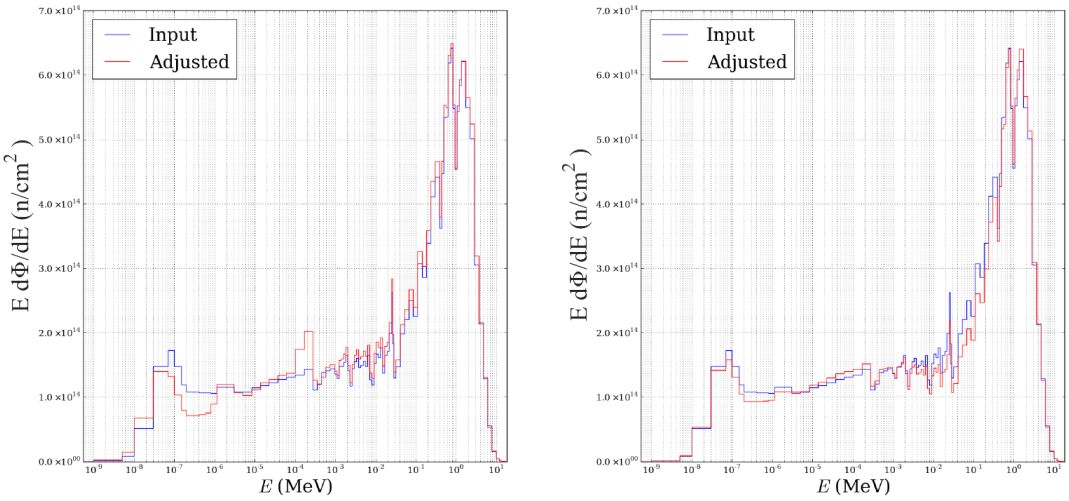

Figure 1.2: Comparison of the Adjustments Between *LSL-M2* and GenSpec.

1.1.4 Uncertainty Quantification and Sources of Uncertainty

Uncertainty quantification (UQ) is the process of quantitative characterization and reduction of uncertainties in both computational and real world applications [10]. It tries to determine how likely certain outcomes are if some aspects of the system are not exactly known or are perturbed. Most often, uncertainty grows with the size and complexity of

simulations and analyses [11].

While computationally calculating energy spectra in a reactor is now a relatively simple task given the capabilities of Monte Carlo transport codes, the problem of uncertainty still exists [12]. Even if the transport code modeled the physics perfectly, parameters such as material composition, densities, and temperature are inputs into the code and can never be known with perfect accuracy. In addition to input parameters, the geometry of the problem can only be known within a measure of accuracy; not perfectly[13]. Therefore, even if it is assumed that the transport code implements the physics of neutral particle transport flawlessly, all the data provided in the form of input parameters must be questioned at some level as most are independent measurements.

1.2 Problem Statement

The overall goals for a neutron adjustment code and of GenSpec, are two-fold [14][12]. First is to bring the reaction probabilities calculated into a better agreement with experimentally measured reaction probabilities. Secondly, an adjusted spectrum should contain smaller uncertainties than those associated with the calculated trial spectrum [14][15]. GenSpec, the program, needs to be expanded to produce its uncertainty data so it may be used as a reliable neutron energy adjustment code. Therefore, the addition of uncertainty quantification to GenSpec's output adjusted spectrum must be included and assessed for validity. This thesis will contain the following results:

- Product of the uncertainty propagation;
- Validation of this implementation;

The thesis will report on the computational process and findings of the uncertainty quantification, as well as possible issues and feasible solutions. In addition, the thesis will

also serve as a Sandia National Laboratories SAND Research Report, aimed for Unclassified Unlimited Release (UUR)[1] to the public.

[1]The Sandia National Laboratories SAND No. 2017-4728 T

2. THEORY

GenSpec as a whole consists of several components including preprocessing components and optimization components. GenSpec borrows components from the *LSL-M2* program to do the preprocessing spectrum unfolding for the NAA analysis. The programming hierarchy can be seen in Figure 2.1.

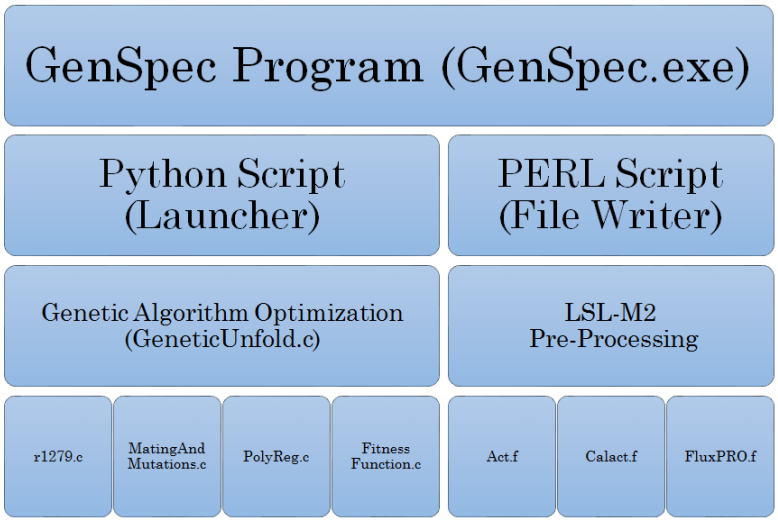

Figure 2.1: Hierarchy of the GenSpec Program, and its Programmed Components.

In the figure above the following sub-programs are:

- Act.f: contains the pre-processing *FORTRAN* code Act, borrowed from the *LSL-M2* code package;
- Calact.f : contains the pre-processing *FORTRAN* code Calact, borrowed from the *LSL-M2* code package;

- FluxPRO.f : contains the pre-processing *FORTRAN* code FluxPRO, borrowed from the *LSL-M2* code package;
- r1279.c : contains the *C* algorithm for the random number generator used by the GenSpec program;
- MatingAndMutation.c : contains the *C* algorithms for parent selection, mating, and mutation used by the GenSpec program;
- PolyReg.c : contains the *C* algorithm for the polynomial regression fit used by the GenSpec program.
- FitnessFunction.c : contains the *C* algorithms for the fitness function, the shift function, and the cross-over function, used by the GenSpec program.

Also noted in Figure 2.1 are the *PERL* and *Python* scripts within GenSpec. The *Python* script is used to launch the GUI and initial file writing of the input file. The *PERL* script reads the input files and data, and modifies the work directory files that either GenSpec will use or the *LSL-M2* components will use. For this thesis, the appended *LSL-M2* components will not be modified, and remain 'as is', simply due to the fact that these are legacy code. GenSpec has been written in *C* so that additions to the code can easily be done by a wide array of individuals and programmers, as *C* has many resources available for reference. While going into the semantics involved with programming in *C* is out of the scope of this research, it is assumed that the readers will have a working knowledge of the *C* programming environment, as well as common commands.

GenSpec orders the energy spectrum as either an 89 or 640 group of energy "bins" and/or solutions. For this case, the optimization is done to each energy groups' 'fitness.' The fitness function in GenSpec is the key Quantity of Interest (QoI) that is used for the spectrum adjustment. Fitness has been defined by GenSpec as:

$$f = C - \sum_{i=1}^{m} \frac{|\sum_{j=1}^{n} \sigma_{(j,i)} \Phi_j - r_i|}{r_i} \tag{2.1}$$

where m is the number of foils used in the NAA analysis, n is the number of energy groups, $\sigma_{j,i}$ is the reaction cross-section for each foil/reaction i in energy group j, Φ_j is the total fluence in energy group j of the specimen spectrum, r_i is the measured reaction probability for foil/reaction i, C is an arbitrary constant, and f is the fitness for the specimen spectrum. The fitness function technically only measures the closeness of the reaction probabilities calculated using the “specimen" to those measured through NAA analysis [1]. In this case, a specimen is an entire spectrum, and its fitness dictates whether it will be selected for recombination, thereby passing on its genes to successive generations. In Equation 2.1 it should be noted that the first part of the second summation can be characterized as the calculated reaction probability. This gives the fitness equation in a slightly simpler form

$$f \approx C - \sum_{i=1}^{m} \frac{|\sum_{j=1}^{n} r_{j,i}^{c} - r_i^z|}{r_i^z} \tag{2.2}$$

where m is the number of foils used in the NAA analysis, n is the number of energy groups, $r_{j,i}^c$ is the calculated reaction probability for each foil/reaction i in each energy group j, and r_i^z is the measured reaction probability for foil/reaction i, C is an arbitrary constant, and f is the fitness for the specimen spectrum. The calculated reaction probability comes from the unfolded spectrum, referencing back to Equations 1.1 and 1.2 from the Introduction. It is apparent that complications arise for even low-resolution spectrum unfolding. The integral quantity I that forms the non-homogeneous part of the linear equations will be an experimentally measured quantity, and will therefore have associated measurement uncertainty. Additionally, the response functions R_i and r_i, are microscopic reaction cross-sections since the integral quantities are reaction probabilities attained from

NAA analysis. These microscopic reaction cross-sections will also have associated uncertainties [10]. From both the unfolding integral in Equation 1.2 and the fitness function in Equation 2.1, it is apparent that the sources of uncertainty arise from both calculation uncertainty and adjustment uncertainty. Calculation uncertainties are those present within the model used to provide the calculated data, such as the trial spectrum attained using the *Monte Carlo N-Particle (MCNP)* transport code. The adjustment uncertainty is due to experimental measurements and reaction cross-sections [2][16].

Based on the previous equations in this section and the introduction sections, we can simply state GenSpec's sources of uncertainty.

- Nuclear Cross-Sections;
- Trial Spectrum from *MCNP*;
- Genetic algorithm and its inputs;

Since the uncertainty of all the input parameters are known or can be calculated, a few assumptions about the possible resulting covariance data can be made. First, the input parameters for nuclear cross-sections, material self-shielding, trial spectrum, and algorithm inputs are all real, non-imaginary values [2][1]. Secondly, these values cannot be negative as this is physically illogical for any of these inputs [2]. However, any of these previously mentioned parameters can have a possible value of zero. An additional adjustment performed during the unfolding of the NAA analysis is the use of self-shielding factors. These self-shielding factors may or may not have realizable standard deviations to them as for this research they are computationally derived [17]. These self-shielding factors were calculated to ensure that their uncertainty was so small that it is essentially zero, and is discussed in the Methodology section of this report [17].

With respect to the trial spectrum, a previous and lengthy analysis has been done to develop its covariance matrix. An assumption of this project is that the covariance matrix

that was previously formed is correct and will be used in this project [15][18]. The process for verifying an *MCNP* model for any reactor could take upwards of a year or two and is outside the scope of this project. Lastly, the genetic algorithm 'modifiers' of population, generation, polynomial order, gene-sites, and mutation rate can be assessed. These modifiers directly alter the genetic algorithm in GenSpec, and are user specified variables. All the modifiers may very well have no upper limit, save for mutation, but ultimately will have no further effects at very high values. A more in-depth discussion of the genetic algorithm modifiers of GenSpec will be done in the Methodology section, under the Local and Global Parametic Sensitivity Analysis subsection.

In theory, in order to quantify the uncertainty of GenSpec, be will propagate the errors using perturbation. Initially, we can assume that the input uncertainty for the both the trial spectrum and nuclear data cross-sections will be covariance matrices. Another assumption is that these covariance matrices will be square, making them easily factorized. It is likely that these matrices will be positive-definite or positive semi-definite, which implies that it is a complex-square matrix whose diagonal elements must be real and must be their own complex conjugate [2][3]. Should the covariance matrices not be positive definite, a multiplication by its first eigenvalue can be done to make it positive definite [11]. Providing these assumptions, a process such as Cholesky decomposition can be implemented. The basic theory of applying Cholesky decomposition in a numeric process is shown below, beginning with an n by n matrix called A.

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{bmatrix} \tag{2.3}$$

For Cholesky decomposition, assume matrix A has the form:

$$A = LL^T \tag{2.4}$$

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{bmatrix} = \begin{bmatrix} l_{1,1} & 0 & \dots & 0 \\ l_{2,1} & l_{2,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n,1} & l_{n,2} & \dots & l_{n,n} \end{bmatrix} \begin{bmatrix} l_{1,1}^T & l_{1,2}^T & \dots & l_{1,n}^T \\ 0 & l_{2,2}^T & \dots & l_{2,n}^T \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & l_{n,n}^T \end{bmatrix} \tag{2.5}$$

where L is the lower triangular matrix, with real and positive diagonal entries, and L^T is the conjugate transpose of L. In Cholesky factorization it is assumed that every positive definite matrix A can be factored as seen in Equation 2.4, where L is called the Cholesky factor of A and is defined as the “square root" of a positive definite matrix [11].

The next step in developing GenSpec’s uncertainty will include multivariate normal random sampling of the covariance data for the trial spectrum and nuclear cross-sections. This will allow for different realizations of the same distribution to be made.

Multivariate normal distributions are simply a higher-dimensional form of a normal distribution. A collection of variables is jointly distributed according to some mean, or mean vector of value, while the covariance matrix signifies the relation between the variables of each. First, we have a collection of data called X, which in this case is the spectrum distribution that is a mix between a Maxwellian distribution and a Watts-Fission distribution. We will consider this data to be a multivariate normal such that, $X \sim \mathcal{N}(\vec{\mu}, \Sigma)$. This distribution has a vector of mean values μ contained in $\vec{\mu}$ and a covariance matrix Σ, both of some dimension d. In general, a covariance matrix measures how dependent each individual dimension, and/or energy group, is to the other. Below is the mathematical form for a covariance matrix in expectation form and in expanded form:

$$\Sigma = E[(X - \vec{\mu})(X - \vec{\mu}^T)] = \begin{bmatrix} \sigma_{1,1} & \sigma_{1,2} & \dots & \sigma_{1,d} \\ \sigma_{2,1} & \sigma_{2,2} & \dots & \sigma_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d,1} & \sigma_{d,2} & \dots & \sigma_{d,d} \end{bmatrix} \tag{2.6}$$

where X is some number in a collection of data, $\vec{\mu}$ is a vector of discrete mean values μ, $\vec{\mu}^T$ is the transpose of the $\vec{\mu}$, E is the mathematical representation of an expectation calculation, $\sigma_{i,j}$ is a discrete covariance value of some row or column in dimension d, and Σ is a covariance matrix of dimension d. Each discrete covariance value, $\sigma_{i,j}$ is defined as:

$$\sigma_{i,j} = E[(\vec{x_i} - \vec{\mu_i})(\vec{x_j} - \vec{\mu_j})] \tag{2.7}$$

To sample from this distribution, a Cholesky decomposition of the covariance matrix must be done such that $\Sigma = LL^T$, if Σ is positive definite or positive semi-definite [11]. We can then generate d independent samples from a standard normal random variable:

$$\vec{z} = (z_1, ..., z_d)^T, \quad Z_i \sim \mathcal{N}(0, 1) \tag{2.8}$$

To get a sample from X, we then compute

$$x = \mu + LZ \tag{2.9}$$

To demonstrate how this procedure works, we look at a covariance matrix of some normal vector $Z \sim \mathcal{N}(0, 1)$. In terms of the expected value, the covariance of Z is:

$$\Sigma(Z) = E[ZZ^T] = I \tag{2.10}$$

where I is the identity matrix. Now, consider a vector X = LZ. The covariance of a

collection of random variables is

$$E[XX^T] = E[LZ(LZ)^T] = E[LZZ^TL^T] \tag{2.11}$$

$$\therefore E[XX^T] = LE[ZZ^T] = LL^T = \Sigma(X) \tag{2.12}$$

In changing this result to a variable with a non-zero mean, we simply add the desired mean μ [11].

Additionally, to further understand GenSpec, a local and global sensitivity analysis will be done. The motivation of a sensitivity analysis can be widely viewed as quantifying the relative contributions due to individual parameters or inputs and determining how variation in parameters affects the programs response [10]. A local sensitivity analysis will also help to ascertain whether the model is robust or tenuous with regard to parameters [11]. This analysis is also fairly quick and the results will help in determining whether the model can be further simplified by removing or fixing insensitive parameters or by specifying a refined parameter space that optimally impacts the program output and its uncertainties. A study of just the genetic algorithm modifiers should be done to test the genetic algorithm itself, the major processing component of GenSpec for possible errors. This genetic algorithm local sensitivity analysis is a random perturbation of the modifiers of population, generation, gene-sites, polynomial order, and mutation. The local sensitivity analysis will focus on the local parameter of fitness in each generation, while the global analysis will focus on the resulting adjusted spectrum. Studying these inputs will also reduce the epistemic uncertainty that may be overlooked in the code based on untested, user-selected inputs [11].

This theory section provides a broad overview of the statistical analysis required in the uncertainty quantification process. Most of these processes may or may not be utilized in the final project but are presented.

3. METHODOLOGY

The following four programs are utilized in this thesis:

• *GenSpec* - A program that utilizes a genetic algorithm for neutron energy spectrum adjustment and optimization. Written using *C* and *Python* in 2014 with intent to release to the public [1].

• *MATLAB* - A multi-paradigm numerical computing environment and fourth-generation programming language. *MATLAB* allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages [19].

• *C* - A general-purpose, imperative computer programming language, supporting structured programming, lexical variable scope and recursion, while a static type system prevents many unintended operations [20].

3.1 Test Environment

As previously stated, GenSpec aims to adjust neutron energy spectra in nuclear reactors, specifically test reactors that perform experiments. For this analysis we will be assessing the free-field environment of the SNL Annular Core Research Reactor (ACRR) central experimental cavity.

The ACRR is a pulse, transient, and steady-state, pool type reactor. The ACRR can achieve a maximum pulse of 250 MJ with a full-width half-maximum of 6 ms, or can operate at about 4 MW in steady-state mode. In the transient mode, the pulse shape can be altered to the desired requirements allowing for an energy deposition of 300 MJ. It was designed to have an epithermal spectrum, and features a prominent 9.17 inch dry central irradiation cavity at the center of the core. The ACRR is typically used to perform irradiation testing where a high neutron fluence is required for a short period of time. A wide

variety of experimental campaigns include radiation damage in materials testing, nuclear fuel testing, space nuclear thermal propulsion testing, and medical isotope production. As such, it is important to maintain a high degree of resolution in the energy spectrum is needed for both testing and qualification.

It is currently assembled in an annulus configuration in order to accommodate large experiments. The core is composed of 236 fuel elements that feature stainless-steel cladding, and are 1.5 inches in diameter and 21 inches in length. The fuel elements for ACRR are similar in size to TRIGA type fuel, however the fuel is unique in that it is uranium dioxide/beryllium oxide (UO_2-BeO) which can handle a large heat capacity, thus producing larger pulsing abilities. Additionally, coupled to ACRR is the Fueled Ring External Cavity (FREC-II), which was installed to provide additional nuclear environmental testing by increasing the volume of fissile material. FREC-II for these experiments was decoupled and not modeled.

The designed epithermal spectrum allows the fluence to be tailored to the desired specifications, using spectrum modifying “buckets." Different combinations of moderators or absorbers can be used to either thermalize or harden the resulting spectrum within the cavity. For this study none of the spectrum modifying buckets were used as the free-field neutron environment was favored. For an unmoderated/non-absorbed condition, the neutron fluence at the axial centerline of the central cavity is 2.0×10^{13} n/cm^2 per MJ of reactor energy. Approximately 46% of the neutron fluence is above 100 keV and 58% above 10 keV [18]. Figure 3.1 is an image of the coupled ACRR and FREC-II.

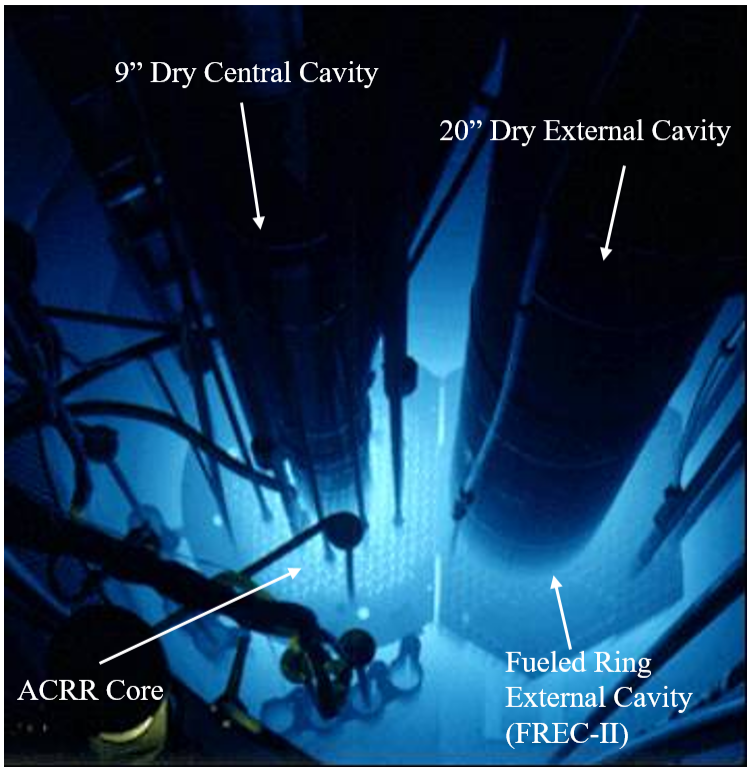

Figure 3.1: The ACRR and FREC-II Operation at 2-MW Steady-State Power.

3.1.1 MCNP Results

A neutronics model of the ACRR was used in the calculation of the trial spectrum and was developed in *MCNP* [14]. A previous verified model of the ACRR *MCNP* model was created using macrobodies geometry descriptions [15]. The file was used for not only the trial spectrum calculation, but all the computational *MCNP* input results. The additions to *MCNP* input that were made to obtain the trial spectrum can be found in Appendix A along with an example of a tally card or the self-shielding factors. A rendered image of the *MCNP* model can be seen below in Figure 3.2.

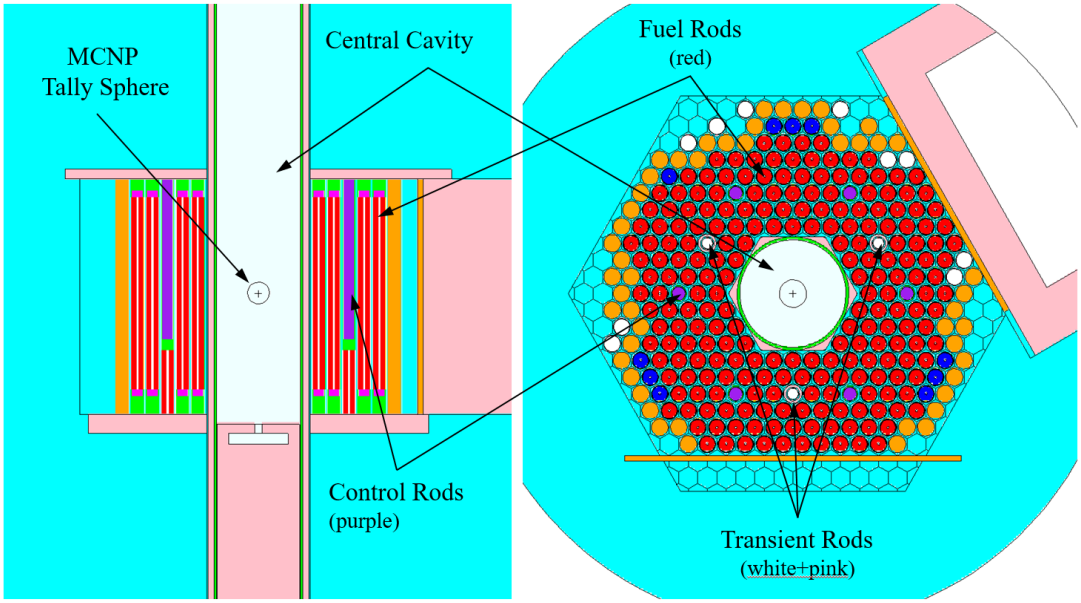

Figure 3.2: Diagrammed MCNP Model of the ACRR.

This model was run with the safety, transient, and control rods in the full-out position. For the tallies used to define the trial spectrum and self-shielding factors, 6 cm diameter tally sphere can be seen in the center. In order for the trial spectrum to attain reasonable statistics in all the energy groups, the ACRR model was run on a parallel machine for 20 billion source particles [18][21]. The covariance for the trial spectrum can be seen plotted below and can be found in Appendix B in raw data form. The neutron energy spectra were calculated for a 640-energy group and an 89-energy group structure using *MCNP5* version 1.60 with the ENDF/B-VII cross-sections and the associated uncertainties. The results from *MCNP* were converted from fluence per source neutron to fluence per fission. Figure 3.3 shows the unadjusted free-field neutron energy spectrum found from *MCNP* for both the 640-group and 89-group structure.

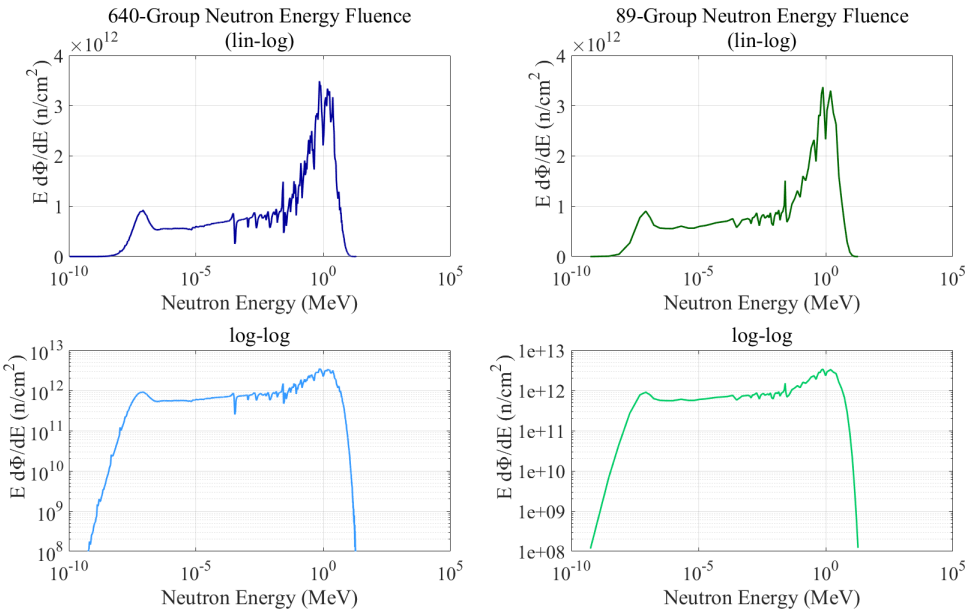

Figure 3.3: *MCNP* 640-Group and 89-Group Neutron Fluence Energy Spectra.

In Figure 3.3, the energy fluence peaks at about 1 MeV, while the thermal peak occurs near 0.07 eV for both energy group structures. The *MCNP* results for both the 640-group and 89-group energy spectra are in agreement as expected. The 640-group energy spectrum shows more structure, particularly in the energy range of around 1 keV to 1 MeV. This structure is considered to be real and not a spectral artifact produced by the code or cross-section [1][16]. This is most likely caused by the resonance structure found in the elastic scattering cross-section of oxygen which is the moderating material for ACRR. Most cross-section covariance data below 0.01 eV has errors above 15%, most likely since 'cold' neutron cross sections are not characterized nor well understood in general [22].

3.1.2 Neutron Activation Analysis

As stated in Section 2.3.1 GenSpec uses NAA analysis. In total for this work, 21 different foil types were irradiated in ACRR at the central axial centerline, and resulted

in 35 reactions. The foils were all irradiated in a total of 10 different pulse operations at 150 MJ. The four fission reaction foils were irradiated individually in both a cadmium cup and boron ball configuration at steady-state operations. The boron ball and cadmium cup configuration can be seen in Figure B.3 of Appendix B. Two additional (n,γ) reaction foils, cobalt and gold, were also irradiated but were not used due to inconsistencies in the results of both in the unfolding process. All the irradiations were performed at the peak axial fast neutron fluence location within the cavity of the core. In order to minimize self-shielding effects, the foils were not stacked, but rather arranged horizontally on a standing aluminum tray, which can be seen in Figures B.1 and B.2 of Appendix B. Several tests were run to test lower power, pulse, and temperature effects of the spectrum, but no statistical differences were determined from these tests [17][18]. Additionally, no variation was seen radially or azimuthally in the fluence near the peak axial fast neutron fluence location [18].

A list of reactions can be found in Figure 3.4, this will also include all of the nuclides used. The selection of foils and activation reactions was chosen based on previous studies that have been conducted over several years [17]. The choice of foil and activation reactions were also chosen based on expert judgment and are not believed to be of issue [16][17][21]. No ideal set of activation foils that allows for the entire neutron energy spectrum to be calculated from dosimetry exists, but enough reactions do exist to result in a high resolution spectrum after energy spectrum adjustment [18]. Dosimetry data for any given neutron environment will statistically vary, but there are some similarities. Typically, neutron activation that results in the emission of protons, neutrons, or alpha particles are often a result of high neutron energy reactions, greater than 1 MeV. Neutron activation that results in the emission of a prompt gamma or a fission reaction will ultimately determine the shape of the thermal and epithermal region of the spectrum. Thus, covering foils with cadmium and boron allows for resonances above the cutoff energies to become conspicuous, allowing for additional information and a higher resolution.

In this figure, the reactions are grouped based on reaction type. The first group represents the high-energy neutron reactions, the second represents the low-energy radiative capture reactions for bare foils. The third group represents the radiative capture with a cadmium cover. Finally, the fourth group represents the fission reaction foils placed in both a cadmium cup and boron ball configuration. Reactions highlighted in dark gray were not used, while light gray colored reactions represent foils that had the results omitted from this report and testing campaigns. The nickel reference foil is highlighted across the figure in group one. Also seen in the fourth column are the counting uncertainty.

Activation Reaction	Half-Life	Activity (Bq/atom-isotope)	Counting Uncertainty (%)
^{58}Ni(n,p)^{58}Co - Reference	70.83 d	9.7808E-18	2.9
^{24}Mg(n,p)^{24}Na	14.957 h *	1.2890E-17	2.3
^{27}Al(n,α)^{24}Na	14.957 h *	6.1585E-18	2.3
^{32}S(n,p)^{32}P Cf-equ	14.284 d	7.0703E+14 n/cm^2	3.6
^{46}Ti(n,p)^{46}Sc	83.788 d	7.8515E-19	2.3
^{47}Ti(n,p)^{47}Sc	3.349 d	3.8055E-17	3.1
^{48}Ti(n,p)^{48}Sc	43.67 h	8.8424E-19	1.4
^{55}Mn(n,2n)^{54}Mn	312.3 d	--	--
^{54}Fe(n,p)^{54}Mn	312.3 d	1.6954E-18	2.2
^{56}Fe(n,p)^{56}Mn	2.579 h *	5.8136E-17	1.8
^{59}Co(n,p)^{59}Fe	44.495 d	1.7965E-19	5.8
^{59}Co(n,2n)^{58}Co	70.83 d	--	--
^{58}Ni(n,2n)^{57}Ni	35.9 h	--	--
^{60}Ni(n,p)^{60}Co	1925.27 d	6.3203E-21	5.3
^{63}Cu(n,α)^{60}Co	1925.27 d	--	--
^{64}Zn(n,p)^{64}Cu	12.701 h *	4.7963E-16	3.1
^{90}Zr(n,2n)^{89}Zr	78.41 h	1.7102E-19	5.1
^{93}Nb(n,2n)^{92m}Nb	10.15 d	2.5526E-19	2.0
^{115}In(n,n')^{115m}In	4.486 h *	--	--
^{23}Na(n,γ)^{24}Na	14.957 h *	1.8587E-15	2.2
^{45}Sc(n,γ)^{46}Sc	83.788 d	7.3161E-16	2.2
^{55}Mn(n, γ)^{56}Mn	2.579 h *	3.4622E-13	1.8
^{58}Fe(n,γ)^{59}Fe	44.495 d	8.3270E-17	2.2
^{59}Co(n,γ)^{60}Co	1925.27 d	5.6707E-17	1.4
^{63}Cu(n,γ)^{64}Cu	12.701 h *	--	--
^{98}Mo(n,γ)^{99}Mo	2.748 d	2.4853E-15	1.7
^{109}Ag(n,γ)^{110m}Ag	249.78 d	8.7524E-17	0.81
^{186}W(n,γ)^{187}W	23.72 h *	2.2104E-13	1.5
^{197}Au(n,γ)^{198}Au	2.694 d	5.6538E-13	3.2
^{23}Na(n,γ)^{24}Na - Cd	14.957 h *	4.2940E-16	2.2
^{45}Sc(n,γ)^{46}Sc - Cd	83.788 d	1.1446E-16	2.2
^{55}Mn(n, γ)^{56}Mn - Cd	2.579 h *	9.1556E-14	1.9
^{58}Fe(n,γ)^{59}Fe - Cd	44.495 d	2.5208E-17	1.8
^{59}Co(n,γ)^{60}Co - Cd	1925.27 d	1.8963E-17	1.4
^{63}Cu(n,γ)^{64}Cu - Cd	12.701 h *	--	--
^{98}Mo(n,γ)^{99}Mo - Cd	2.748 d	2.3502E-15	1.6
^{109}Ag(n,γ)^{110m}Ag	249.78 d	5.7839E-17	0.8
^{197}Au(n,γ)^{198}Au - Cd	2.694 d	4.7342E-13	3.2
^{235}U(n,f)FP - BB	^{140}Ba - 12.752 d	3.1399E-09 #fis	3.5
^{238}U(n,f)FP - BB	^{140}Ba - 12.752 d	2.8531E-10 #fis	3.5
^{239}Pu(n,f)FP - BB	^{140}Ba - 12.752 d	3.1557E-09 #fis	3.5
^{239}Np(n,f)FP - BB	^{140}Ba - 12.752 d	1.5815E-09 #fis	3.5

*half-life less than one day, **half-life less than one hour, #fiss = units are fissions per atom isotope, Cd = foil is located in a standard Cd cup, BB = foil is located in a standard B_4C ball, *Reference* = all activities are normalized to a ^{58}Ni(n,p)^{58}Co activity.

Figure 3.4: Neutron Activation Dosimetry Used by GenSpec.

3.2 Uncertainty Quantification Procedure

Since GenSpec relies on several different inputs, a simple schematic of the errors that need to be accounted for in the uncertainty quantification process is shown in Figure 3.5, below.

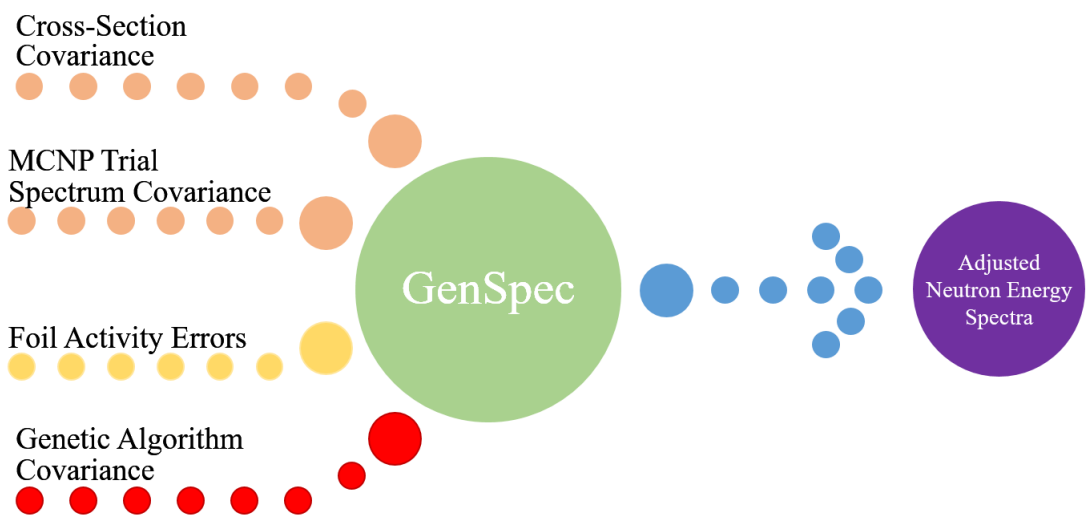

Figure 3.5: Input Data Schematic of GenSpec.

In Figure 3.5, we can see that GenSpec has four sources of error. Both the cross-section library and *MCNP* trial spectrum have covariance, correlation, or uncertainty values in the data library. The cross-section library has correlation matrices that are used by *FluxPRO.f* and the genetic algorithm. *FluxPRO.f* is the *FORTRAN* pre-processing program from *LSL-M2*. The genetic algorithm also uses the cross-sections to quantify the fitness function as seen previously in Equation 2.1 in the theory section of this report. The foil activity errors, are measured data by the user, and therefore have to be manually put in by the user.

For GenSpec most of the uncertainty information for both the calculated or measured data is easily obtained. The nuclear cross-section correlation matrices have been converted

to covariance data, and re-binned to form 640 by 640 matrices. This data is written to the file designated *filename.xsc* within the working file directory.

The process to quantify the uncertainty of GenSpec entails sampling components to develop the uncertainty data as well as test the program for faults. A missing piece of analysis data for this research is the error or covariance associated with just the genetic algorithm and its modifiers, that the user selects when running GenSpec. This process entails not only a random perturbation analysis to quantify its uncertainty, but a local and global parametric sensitivity analysis of the genetic algorithm and its modifiers. *MATLAB* was favored to create simple key files for documentation reasons. Two of the key files can be found in Appendix C and its subsection appendices.

3.2.1 Local and Global Parametric Sensitivity Analysis

For the analysis of the genetic algorithm and its modifiers, a base case was chosen. The base case modifiers are as follows in Table 3.1. These were chosen as the inputs since they were preloaded for the free-field 640-group input file in GenSpec. GenSpec comes with several preloaded test files, so users can confirm the program is working. The base case inputs assume that when a variable is changing, the other parameters remain static.

Table 3.1: Base Case Values of the Genetic Algorithm Modifiers.

Modifier Parameter	Value
Population	200
Number of Generations	600
Polynomial Order	10
Number of Gene-Sites	80
Mutation Rate	0.15

From the original GenSpec report, a few variables could easily have limits identified. Mutation rate is essentially a percentage of the specimen population spectra from 0% to 100% that will randomly undergo a mutation of its gene-sites. Mutation in GenSpec ensures that the solution space is thoroughly explored. Polynomial order is just the order of the fitted polynomial into the gene-sites that creates the shift function after parents have undergone recombination and a child spectrum is being produced. The polynomial order dictates how important each gene-site is to the shift of the original trial spectrum, by either ensuring the polynomial goes through every gene-site or just best fits the gene-site trends. The manual suggests not exceeding a 10th order polynomial, as it can lead to unrealistic spectral artifacts [1].

Gene-sites are the number of points of interest distributed in base-10 logarithmic space between the minimum and maximum energy values. These are initially randomly selected traits for the first generation, and inherited traits in subsequent generations. The manual suggests values should ideally be two to three times the value of the polynomial order, but also suggests that large values increase computation time [1]. Generations is the number of iterative generational runs, where the spectra evolve. Each generation runs the exact same population input. While convergence is problem-dependent, the manual states that most cases will achieve convergence in less than 1000 generations [1]. It was difficult to define limits for the last modifier of population. Population defines the number of specimen spectra produced in each generation and is a fixed value, that is reiterated over each generation. The only notes in the manual on values of population are that at higher values, the likelihood of convergence to a sub-optimal solution is reduced, but can greatly increase computation time [1]. The manual also specifies that population values should be at least greater than 200 [1].

Once these limits were identified, a uniform distribution was made to evenly sample the entire parameter solution space to test each parameter's sensitivity on both a local

and global level. Below, in Table 3.2, are the defined limits of this parametric analysis and the number of cases run in-between each. While the same number of cases for each parameter was desired, values such as polynomial order and mutation rate had limits to the input variable characters, such as whole integer values and floating digits, respectively. A smaller number of inputs were also favored, as computation times were not characterized for this program.

Table 3.2: Parametric Sensitivity Analysis Limits and Number of Cases.

Modifier Parameter	Minimum	Maximum	Number of Cases
Population	200	10,000	25
Number of Generations	10	1000	25
Polynomial Order	4	15	12
Number of Gene-Sites	10	1000	19
Mutation Rate	0%	100%	25

MATLAB was used to create all 106 input files, while a batch file was written to execute the GenSpec for each input and store the information of fitnesses, ideal spectrum, and energy midpoints into separate folders. Post analysis was also conducted in *MATLAB* since it allows for matrix manipulations and plotting of functions. *MATLAB* also allows access to the *Statistic and Machine Learning ToolBox* which enables fast analysis of statistical parameters like standard deviations, covariances, and kurtosis.

For the global analysis, GenSpec was run some N number of times, with random input for population, generation, gene-sites, polynomial order, and mutation rate swapped in each time, the resulting matrix B was made in *MATLAB*.

$$B = \begin{bmatrix} \phi_{e1,1} & \phi_{e2,1} & \cdots & \phi_{e640,1} \\ \phi_{e1,2} & \phi_{e2,2} & \cdots & \phi_{e640,2} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{e1,N} & \phi_{e2,N} & \cdots & \phi_{e640,N} \end{bmatrix} \tag{3.1}$$

where ϕ_e represents the solution for the neutron energy spectrum found from GenSpec, the first subscript is the energy group number from 1 to 640, and the second subscript is the N number file run [11]. Once this matrix is formed, *MATLAB*'s covariance function $cov()$ can be simply applied as $cov(B)$. This will result in a 640 by 640 matrix for the covariance of the genetic algorithm modifiers. For this analysis, a goal of 100 file runs will be aimed for, since files may fail. As a confirmation of the possible results of this analysis an additional run may be done to see if the initial results can be reproduced.

3.2.2 GenSpec Random Perturbation Analysis

For the uncertainty propagation of GenSpec, some possible code changes will be made to the *PERL* file writer script and the *GeneticUnfolding.c* script. First the *PERL* file writer script had to be modified. This modification printed out a new file to the working directory for the activity measurement errors. These were printed to a new file called *inputfile-name.aer* to be used by GenSpec. This data is collected directly from the *Python* input file writer which creates the file from the GUI user input data. The activity error data was also restored as *acterr* without a file identifier, so it could be accessed by other programs while GenSpec ran.

For the random perturbation analysis, *MATLAB* was used for most of the file writing and multivariate random sampling. The trial covariance data was loaded into *MATLAB*. During this process it was noticed in the *LSL-M2* code that what was believed to be the covariance matrix was actually the correlation matrix, along with a standard deviation

vector of values, $\vec{\sigma} = (\sigma_1, ..., \sigma_d)$. To find the covariance, the following simple process was followed.

$$cov_{i,j} = corr_{i,j}\sigma_i\sigma_j \tag{3.2}$$

where i and j corresponds to the index of the covariance value by row and column, while it refers to the standard deviation d index value of $\vec{\sigma}$. Once the covariance matrix is formed, *MATLAB*'s built in multivariate normal random numbers function $mvnrnd(\mu,\Sigma)$ can used. This function was used with the nuclear cross-section covariance data, but was less complicated as the data was already in the decomposed covariance form. For each distribution, 300 realizations were made from each covariances. For the trial spectrum, GenSpec requires that the input trial spectrum be a differential fluence and also applied a scaling factor for the correct units [1]. *MATLAB* created and printed all of the perturbation and input files for GenSpec, and these *MATLAB* scripts can be found in Appendix E.

The implementation step in the UQ process was kept relativity fluid in its approach due to the number of component codes and script complexity of the current GenSpec program. Two possible procedures are available for this step. The first will be to add an external C GenSpec component, that can be used when a covariance is needed. Changes will mostly be made in the GenSpec *GeneticUnfold.c* file, where a bulk of the core coding exists. Adjustments would also need to be made to the project header file of GenSpec so it could be used by any subscripts in the program [20]. Two additional code scripts will be needed called *cholesky.c* and *covariance.c*. *Cholesky.c* will contain the working function for the cholesky decomposition and file running for the covariance runs. Both files are located in Appendix section D.1. The *covariance.c* file will build the large resulting spectrum matrix from the random perturbation runs. The covariance script should also print a text file of the covariance to the output folder of GenSpec.

The second procedure that will be used as an alternative, is to write a separate co-

variance script in *MATLAB* which can be run after the optimization is done in GenSpec. While this option will only add to the complexity of GenSpec, it may ultimately surpass the initial option. While making changes to GenSpec's core code guarantees that it remains a cheaper and easily transfered code, it could lead to a slower program and may also be unnecessary. Changes to the code will also lead to more bugs present in the code, and will most likely require changes to the *Python* and *GenSpec.exe* files. This could lead to problems that could be difficult to solve should GenSpec require updates after the UQ results of this report. For this procedure, some changes to the GenSpec code will still be made to simply create text files in the working directory and print variable values into them. A completely separate *MATLAB* script can then be made to carry out the UQ process as a post-processing script. For this step, no additional scripts will be added to the GenSpec *C* project such as the earlier *cholesky.c* and the *covariance.c* scripts, and only a *genspec_cov.m* script will be made.

4. RESULTS

4.1 Genetic Algorithm Covariance

For the global and local sensitivity study, data analysis was done in *MATLAB* so the quick use of the *Statistics and Machine Learning ToolBox* could be utilized. For each of the parameters, two graphs were produced to show the effects each modifier had on fitness, the local parameter. This study of these effects on the main QoI, fitness, should ultimately lead us to understand what modifiers may affect the shape of the output spectra and the variance produced by each parameter. The first graph displays the average (green), maximum (red), and minimum (blue) fitness plotted across each variable's limits. This facilitates the display of convergence or divergence within fitness due to the variations in the parameters [11]. Second is a graph of the variance seen in the fitness over the increase in each parameter. Within this second graph, a fitted linear trend line was added to easily asses a positive or negative variance relationship. This graph aims to display either the increase or decrease in the confidence of the values of fitness. While the modifiers could greatly affect fitness, it may be of use to note if little or no variation is reflected in the output.

From the generation parametric analysis, Figure 4.1 shows the effects incurred to fitness is slightly improved upon with an increase in generation value. This is mostly marked from 0 to 200 generations, and little to no increase after convergence was achieved at roughly 300 generations. This corresponds to a decrease in the fitness variance as seen in Figure 4.2.

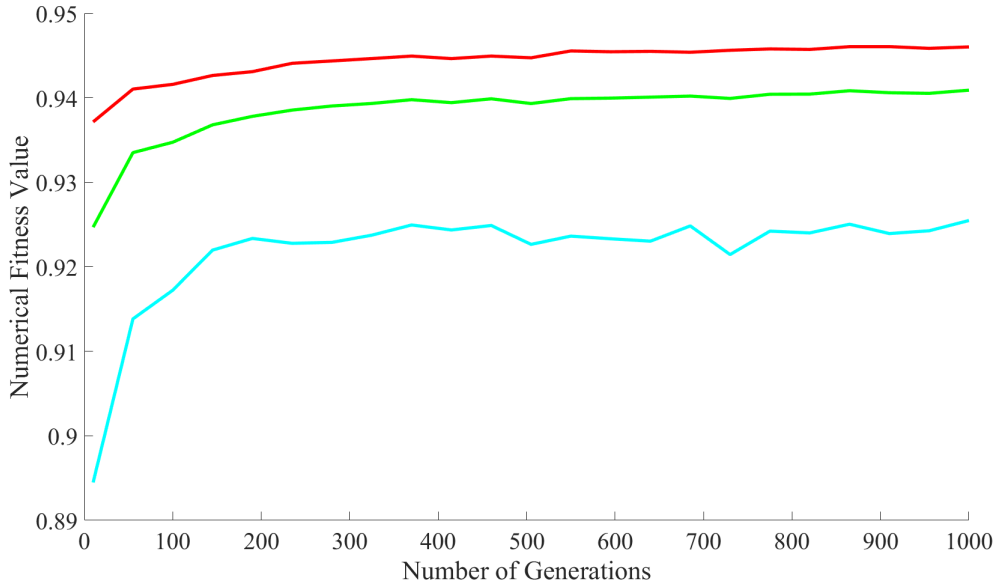

Figure 4.1: Increasing Generation Effects on Fitness.

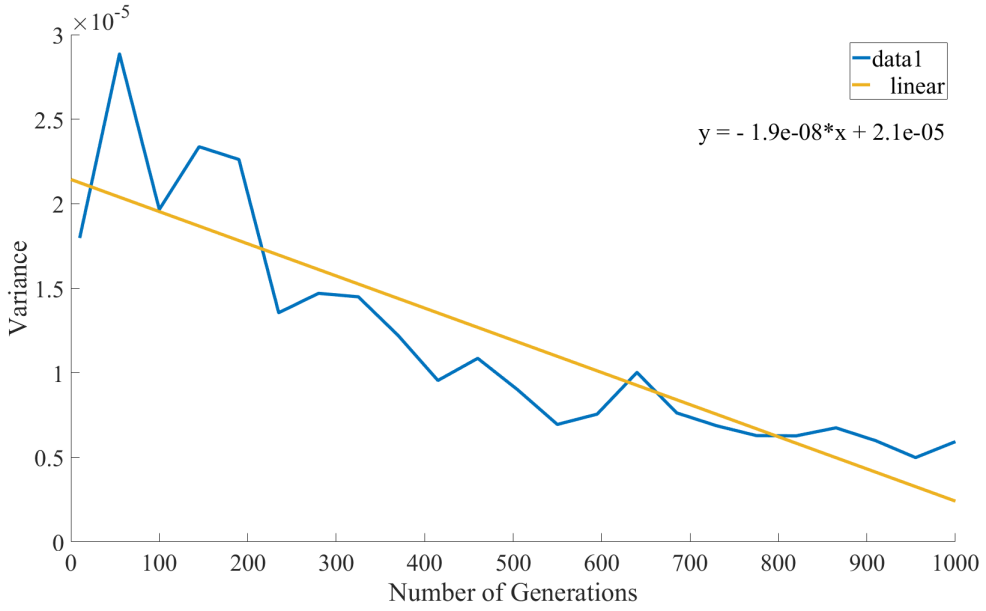

Figure 4.2: Variance of Fitness Due to Increasing Generation Value.

For the polynomial order parameter analysis, it is seen in Figure 4.3 that convergence was achieved after the 12th order. While an increase in fitness is apparent in Figure 4.3, it is seen in Figure 4.4 that an increase in variance also occurs, which negates the validity of the increase in fitness values [11]. It may be the case that convergence is achieved before the 4th order polynomial, and that a divergence and convergence to a suboptimal polynomial order may have occurred.

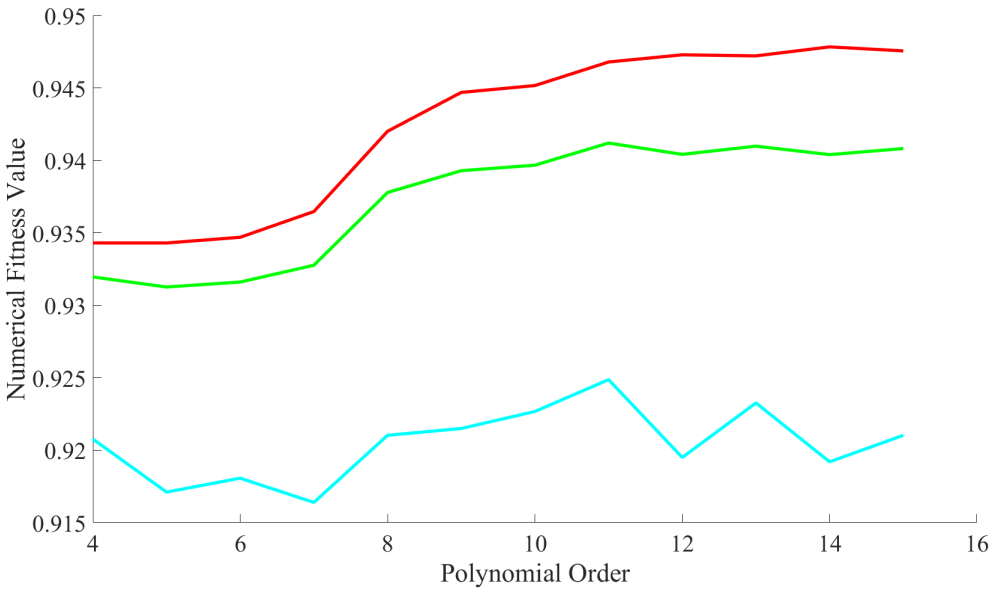

Figure 4.3: Increasing Polynomial Order Effects on Fitness.

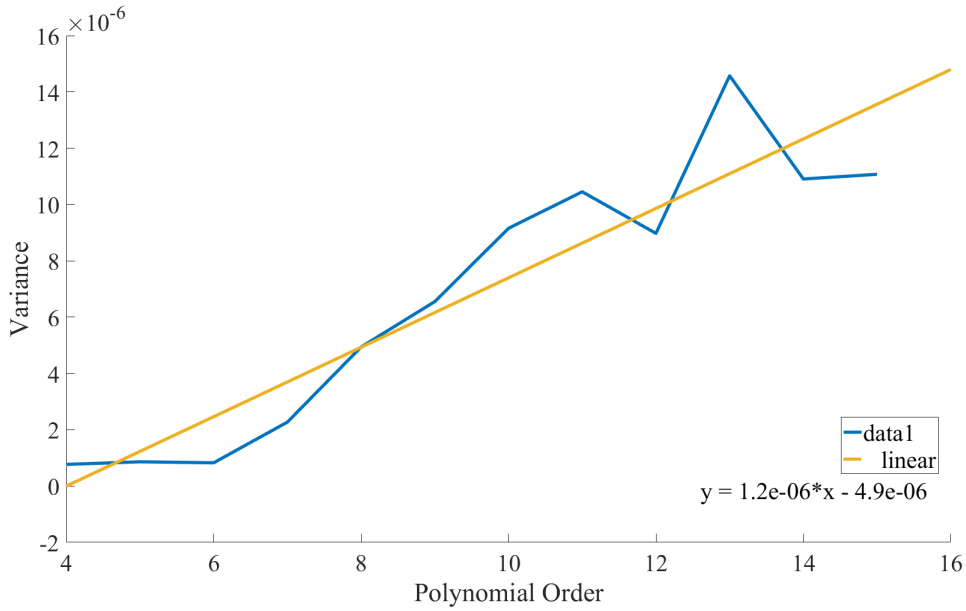

Figure 4.4: Variance of Fitness Due to Increasing Polynomial Order.

For the gene-site parameter analysis, Figure 4.5 shows that convergence is achieved very quickly after just 20 gene-sites are selected. This corresponds to a decrease in variance seen in Figure 4.6. While 10 to 20 gene-sites shows the greatest improvement in fitness, the results plateau and do not noticeably increase further. While these results seem ideal, they may not be relevant at all to an overall knowledge of the effects of gene-sites values on the program. Since the base case polynomial order was 10, these results may only be applicable to this fairly high value. The gene-site modifier will most likely come to be better understood better during the random perturbation analysis was many different values of the polynomial order will test this variables effectiveness as it also differs.

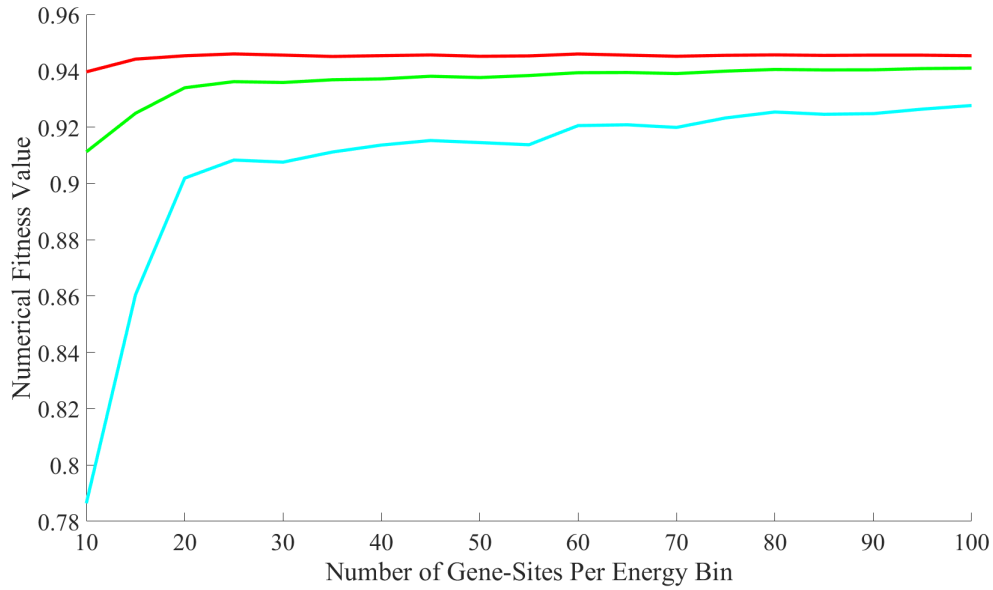

Figure 4.5: Increasing Gene-Sites Effects on Fitness.

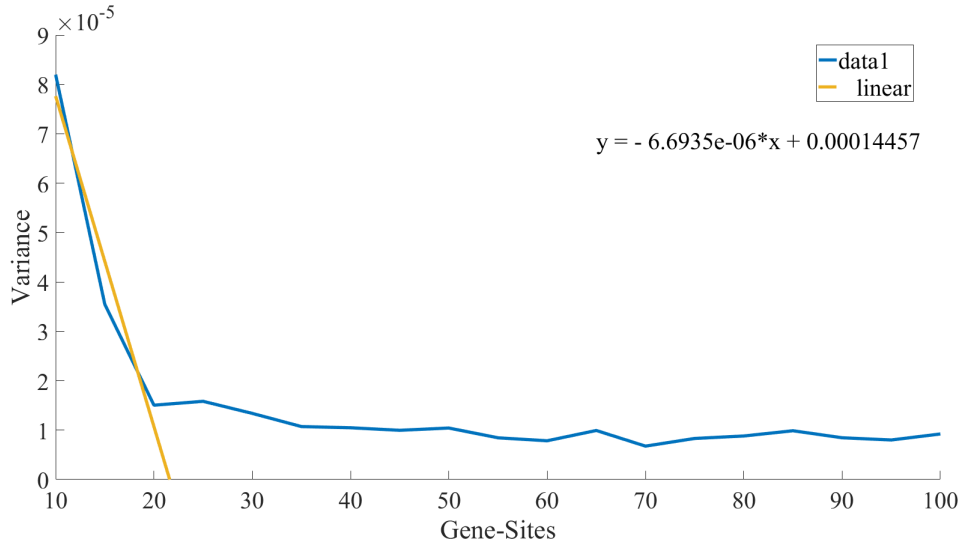

Figure 4.6: Variance of Fitness Due to Increasing Number of Gene-Sites.

Mutation increases lead to a fairly refined ideal spectrum. As seen in Figure 4.7, the maximum remains at a relatively fixed high value, while the average and minimum fitness exhibits volatile behavior and lead to an overall decreased fitness. Even with this behavior, Figure 4.8 shows a decrease in variance. While an increase in mutation may lower the variance in the fitness function, it is likely that since the fitness values actually decrease/diverge, that the range of acceptable specimen spectra is merely increased, leading to a false-positive in the reduction of the variance and indeed the number of “fit" spectra.

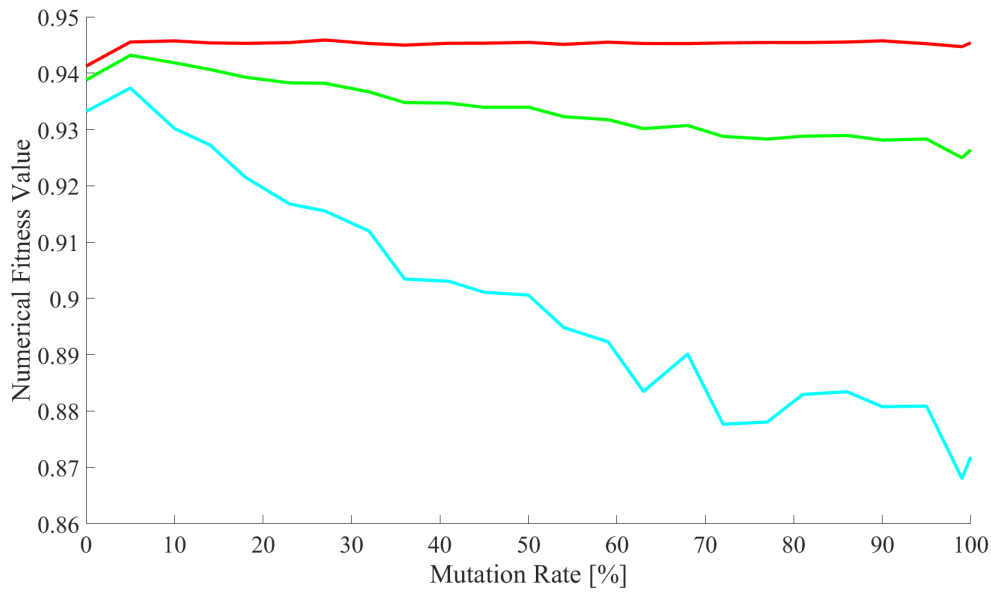

Figure 4.7: Increasing Mutation Rate Effects on Fitness.

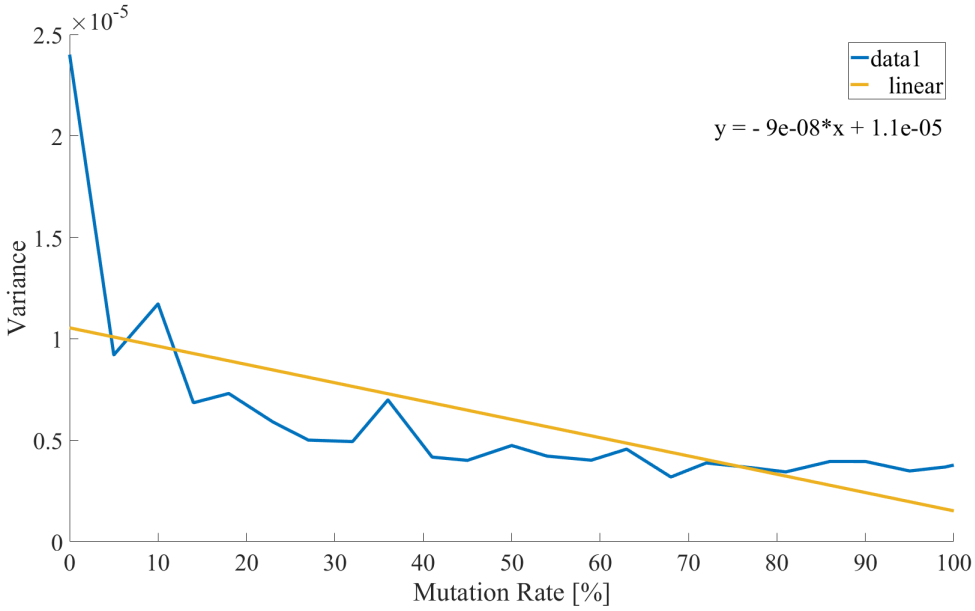

Figure 4.8: Variance of Fitness Due to Increased Mutation Rate.

The last variable to evaluate is population. Many of the graphs within this section suffer from missing components, as some files failed to save and could not be run due to memory issues. From Figure 4.9 it is noted that the population has odd effects on GenSpec. This odd behavior is marked by a slight decrease in average fitness of the specimen spectra. The behavior of the variance exhibits a parabolic shape which is seen is Figure 4.10. This might mean that for this specific base case problem an increase in population will have a slightly negative effect on the fitness, while populations above 4000 will contribute to an increase in the variance of the fitness, meaning these values are questionable in their reliability.

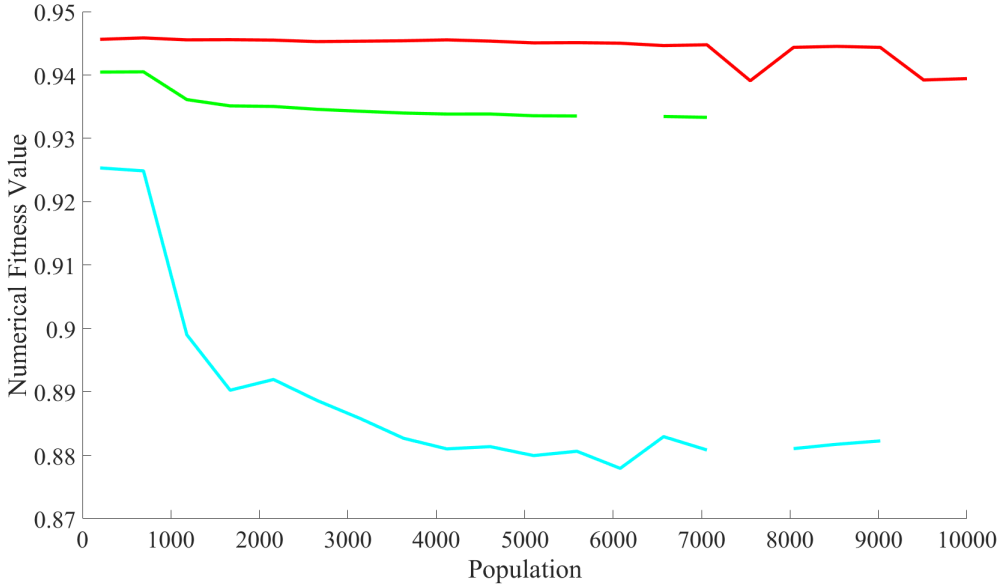

Figure 4.9: Increasing Population Effects on Fitness.

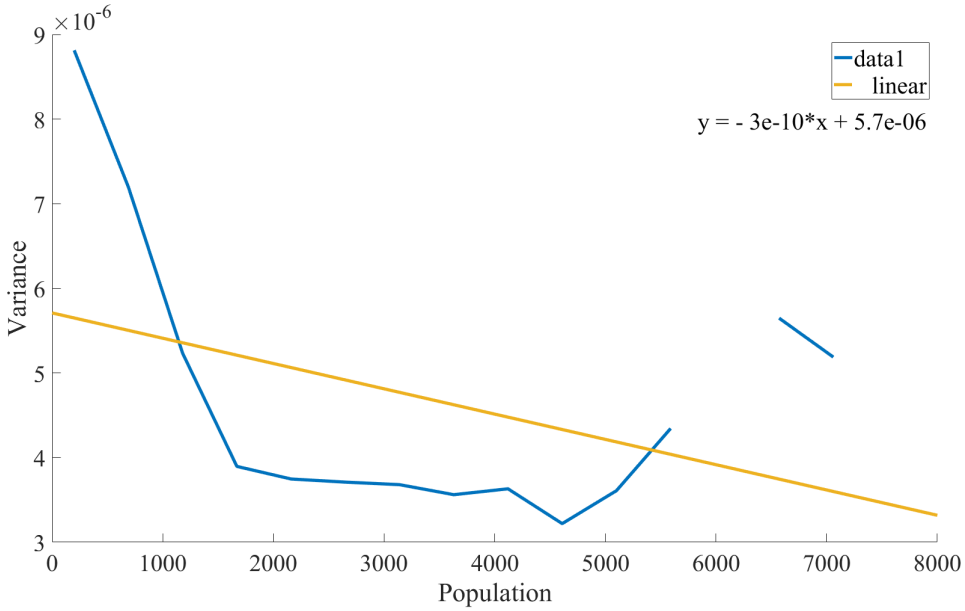

Figure 4.10: Variance of Fitness during Population Increases.

As seen with all the figures, each parameter affected the local variable fitness in different ways. While mutation and population increases saw a minimal production of unrealistic spectral artifacts, some of the population results were unexpected. Exceedingly high population values exhibited an increase in variations of the fitness values, making the confidence in those values low. Additionally, since GenSpec is a fairly new program, it was unknown how long computation times could be. While most simulations easily ran in under 4 minutes on average, it should be noted that some of the high values of population and generation runs often took several hours. It was also observed that due to memory issues, about half of the population runs failed to save ideal spectrum files due to their size. Other issues that were apparent were the production of below zero fitness values, leading to chaotic behavior in GenSpec. The parameters of generation, gene-sites, and mutation all seemed to produce the most positive effects to the produced spectra. All three of these parameters lead to an overall reduction in the variance of the fitness. Population had a somewhat odd effect, and led to a noticeable “window" of ideal population values. Polynomial order, displayed a divergence in the fitness function and also exhibited the greatest increase in the fitness variance.

These differences are mirrored in the resulting spectra, indicating that these local issue lead to global issues as well. Figure 4.11 shows that the variance of the output spectrum for all parameters are quite different from each other.

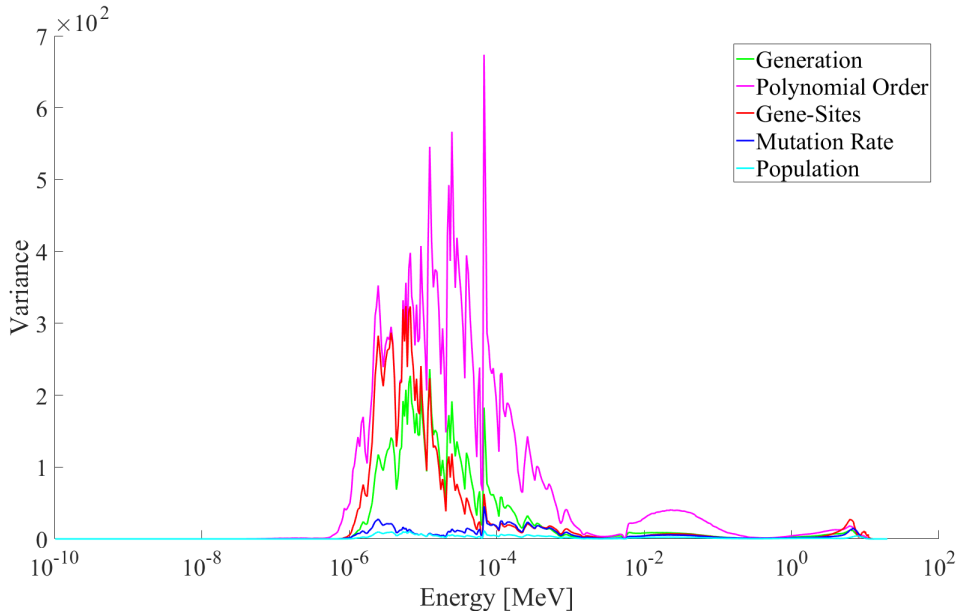

Figure 4.11: Magnitude of the Variance Seen in the Fitness for Each Modifier.

We see that very high variance is attributed to the polynomial order, but that the variance is to the 2nd power. These results could have several different explanations. One, while fitness is a facet of the modifying parameters' effects on the output spectrum, the shift function may also play a large role in changes to the output spectrum as well. The polynomial order is basically the "shift" function in GenSpec, and it is directly applied to the adjusted spectra. Two, this also could mean that large variation in the fitness will not necessarily carry over to the output spectrum, which could be a positive consequence. These could also be underdeveloped results as only 25 cases were run for each parameter. While these effects are important they are only relevant to the base case, but nevertheless assist in redefining the limits of each parameter in order to random sample the genetic algorithm.

For the global sensitivity analysis, a total of 7.6×10^{7} spectra in total were adjusted over the course of 100 input files being run in GenSpec. A method to test for convergence

was developed, and fitness convergence was the simplest. With each new iteration on the fitness parameter done, a cumulative average was produced, and an average among the files was reached. Below in Figure 4.12, the convergence of fitness is at about 0.4 near 60 file runs, which is confirmed at roughly 80 runs.

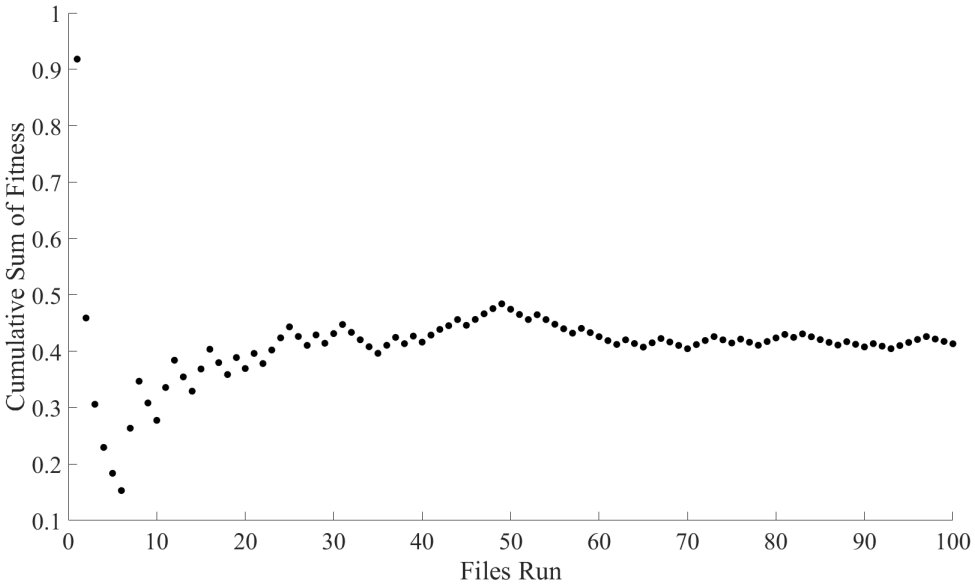

Figure 4.12: Convergence of Fitness for the Random Perturbation Runs.

Once convergence was confirmed, all the spectra produced were plotted to see if variations were apparent. As Figure 4.13 shows, they are most noted in the thermal and resonance regions where they can span several orders of magnitude.

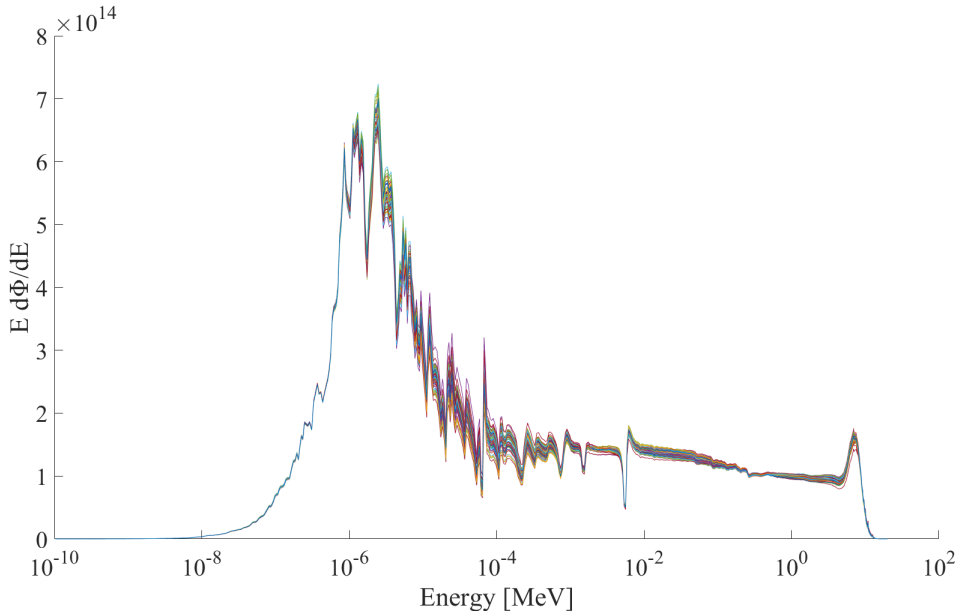

Figure 4.13: Resulting Spectra from the Random Perturbation Runs.

Since variation was obvious in the resulting spectra, these were ordered into a N by 640 matrix in *MATLAB*. Here N represents the number of files run, and in this case it is 100, while 640 is the spectrum energy group structure. Using the simple covariance function $cov()$ from the built-in library, a 640 by 640 covariance matrix was found. A 3D surface plot of the covariance shows that both positive and negative numbers exist. Figure 4.14 shows several views of the covariance as both a 2D and 3D realization.

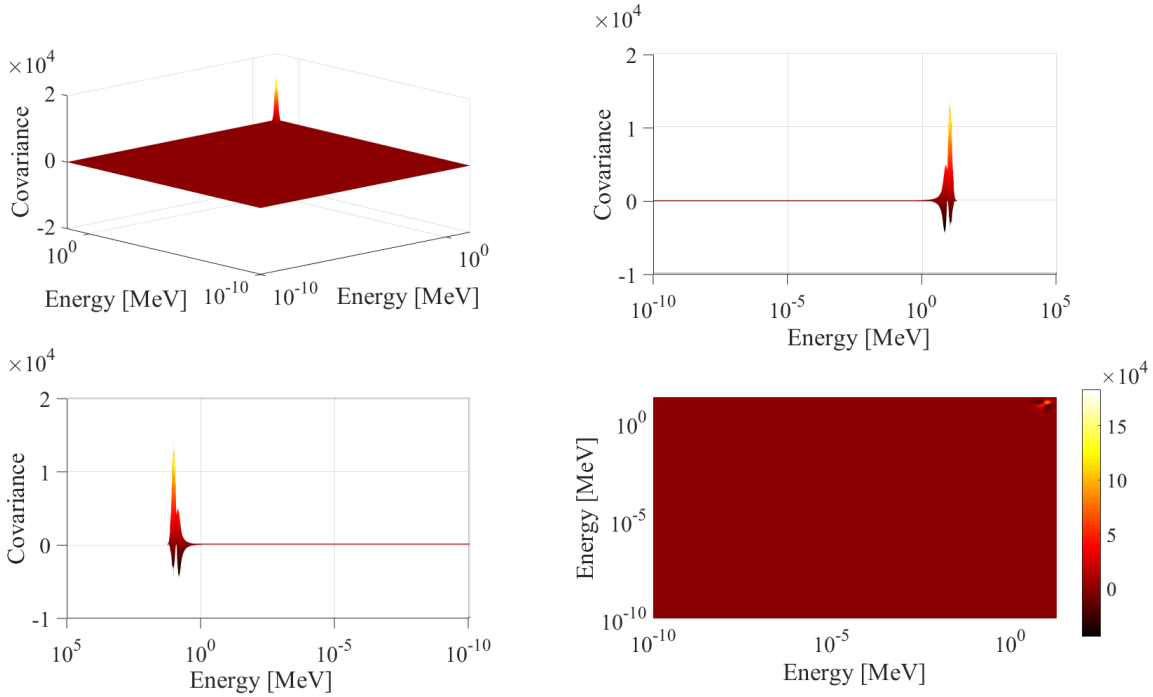

Figure 4.14: 3D and 2D Top and Side Views of the Covariance.

While these numbers do look quite high, they are as expected. Referencing back to Figure 4.11 we see that individually these input modifiers can lead to large variance by themselves, as a variance of all the parameters together, powers to the 4th may be reasonable in this case, especially since the bins represent fluence values which themselves are large numbers as well. A majority of this data is also grouped in the very high energy ranges near 20 MeV, which might mean the uncertainty of this region could be due to cross-sections at high energies [23]. To confirm these initial results, this analysis was run an additional time, since some covariance values where high. This second run had a few conditions added to it. Several files that accounted for a large number of original file, were suspected to have spurious or incorrect results. One of the marked issues in these file runs was the continual production of very low fitness values among several files. Ultimately,

this was a time intensive step as several files would run, including many suspected failed files. While these runs do not necessarily 'fail', they seem to indicate a sub-optimal spectrum is converged too, suggesting these results could be incorrect. Figure 4.15 contains a segment of a suspected failed file for the fitness values. The numbers in the first column are generation, the 2nd, 3rd, and 4th columns are maximum, minimum, and average fitness respectively.

```
0    0.9334541723    0.8677825208    0.9212914230CRLF
1    0.9338322232 -4771488510094854274486851160483155095939171732494378598400.0000000000 -1057979713989989911479
2    0.9338322232 -4771488510094854274486851160483155095939171732494378598400.0000000000 -1057979713989989911479
3    0.9338322232 -4771488510094854274486851160483155095939171732494378598400.0000000000 -1057979713989989911479
4    0.9338322232 -4771488510094854274486851160483155095939171732494378598400.0000000000 -1057979713989989911479
5    0.9338322232 -4771488510094854274486851160483155095939171732494378598400.0000000000 -1057979713989989911479
6    0.9338322232 -4771488510094854274486851160483155095939171732494378598400.0000000000 -1057979713989989911479
```

Figure 4.15: Suspected Failed File showing Fitness Live Feed.

Due to several of these files existing in the first random perturbation analysis of the genetic algorithm, a second run was done after the discussion of results and values were lowered and capped to see if an improvement could be seen in the number of possible failed files. During this second attempt at running the random modifier perturbation, each failed file was noted, and the files that did print a fitness below 0 to the command window were stopped and a new file would start. An additional requirement was that no more than 50% of the files should fail. This 2nd covariance process was attempted two more times, both stopping at roughly 30 perturbations with failure rates around 60%. At first a trend was thought to have been found in the gene-site number, leading to the speculation of a minimum limit existing, but this was not the case. After the 3rd attempt to run, the lead programmer and current researcher concluded that a relationship must exist between the gene-site value and the polynomial order selected that differs from those initially theorized in the GenSpec user manual. Referencing to Appendix B section B.2, file run 12.00

and file run 60.00. This relationship can be noted as unpredictable between the two file run modifier values, making this characterization complex and therefore, non-linear [11]. While this relationship was not characterized, a course of action was developed for future work.

A possible explanation for this issue could be a problem that arises with the freedom of the user selection of the polynomial order, as high polynomial orders can lead to very low fitness values, essentially 0. In Figure 4.16 a simple sine function was fitted with a polynomial at various orders of M. Each fit has some polynomial order M shown in red, fitted to a simple sine function shown in green with gene-site represented as blue circles. The noticeable problem is exhibited in the 9th order polynomial example in the bottom right of Figure 4.16. It is noticeable that towards the end of the r values, a drop off of the polynomial fit below -1 occurs. Issues such as this in GenSpec are most likely leading to the occurrence of fitnesses values below zero. This most likely leads to an acceptance of all the specimen spectra, leading to the production of a possible sub-optimum solution.

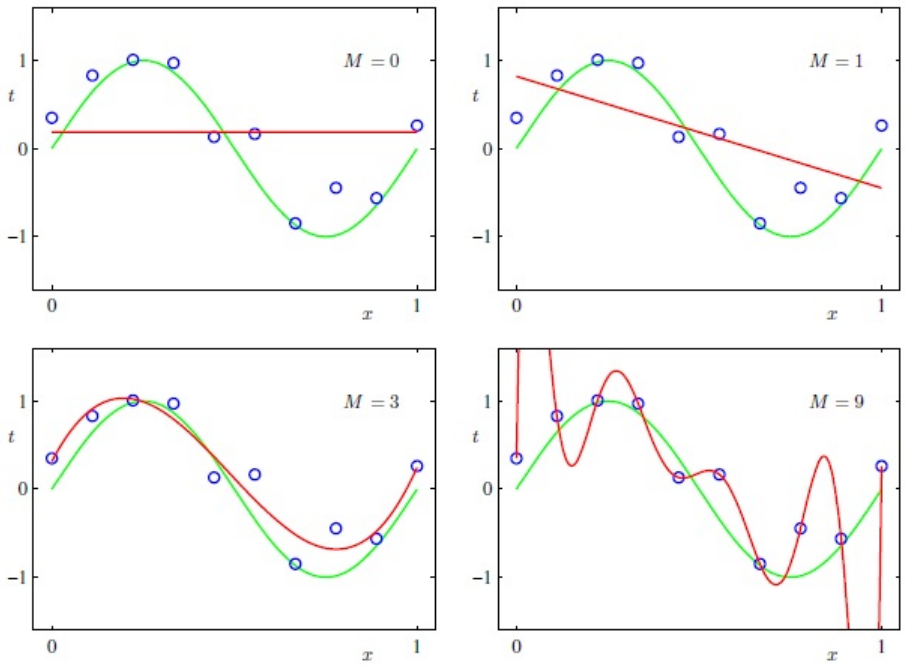

Figure 4.16: Plot of Polynomial Fit Problem.

In total, this step in the random modifier perturbation analysis was attempted 4 different times. While this problem persisted, the analysis process proceeded with a marked reduction in the polynomial order value than previously selected for this run. In the end 211 total files were run for the confirmation covariance analysis, with a failure rate of 53%. Below in Table 4.1 are the limits of each perturbation that was done; a noticeable reduction in polynomial order can be seen.

Table 4.1: Random Perturbation Analysis Limits.

Modifier Parameter	Covariance Run #1	Covariance Run #2
Population Minimum	200	800
Population Maximum	5000	2000
Minimum Number of Generations	200	200
Maximum Number of Generations	800	500
Minimum Polynomial Order	2	3
Maximum Polynomial Order	15	7
Minimum Number of Gene-Sites	5	55
Maximum Number of Gene-Sites	100	85
Minimum Mutation Rate	0%	4%
Maximum Mutation Rate	100%	9%

While an improvement was observed, high rates of failure are still seen due to the possible relationship between gene-sites and polynomial order. As this error could not be reconciled with any input changes, the results from successful file runs were the only ones used. The specific values of each run are shown in the key file with the failed files marked. This key file can be found in Appendix B of this report. Shown in Figure 4.17 is the convergence test for the confirmation covariance random perturbation run.

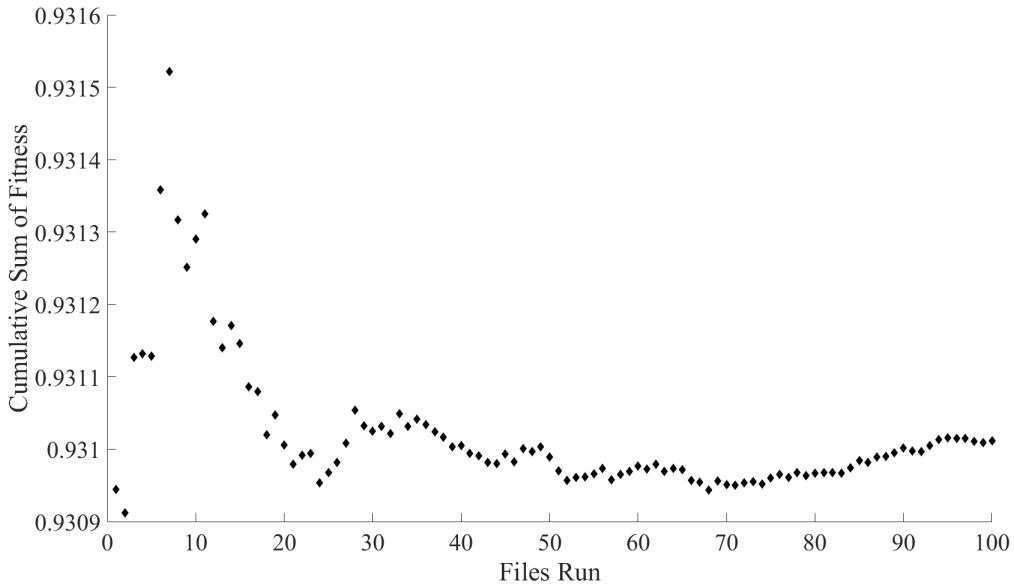

Figure 4.17: Convergence of Fitness for the 4th Random Perturbation Runs.

While this graph does show that convergence was somewhat achieved, the results do differ from the previous convergence graph. While the first convergence runs approached a central average from a lower cumulative fitness' value, this graph looks to approach a cumulative fitness from an initially higher fitness value. These inconsistencies may suggest an issue in present in the code, or one of these random perturbation analysis was conducted incorrectly. In addition, the data in this convergence do not seem to completely converge around a definitive value as previous data did. While these may not be desirable results, time did not permit for more runs to be conducted and with a high corresponding failure rate, an issue is likely to be present in the program. Still the data from all 100 successful runs were collected and the covariance was found and plotted as a surface, below in Figures 4.18.

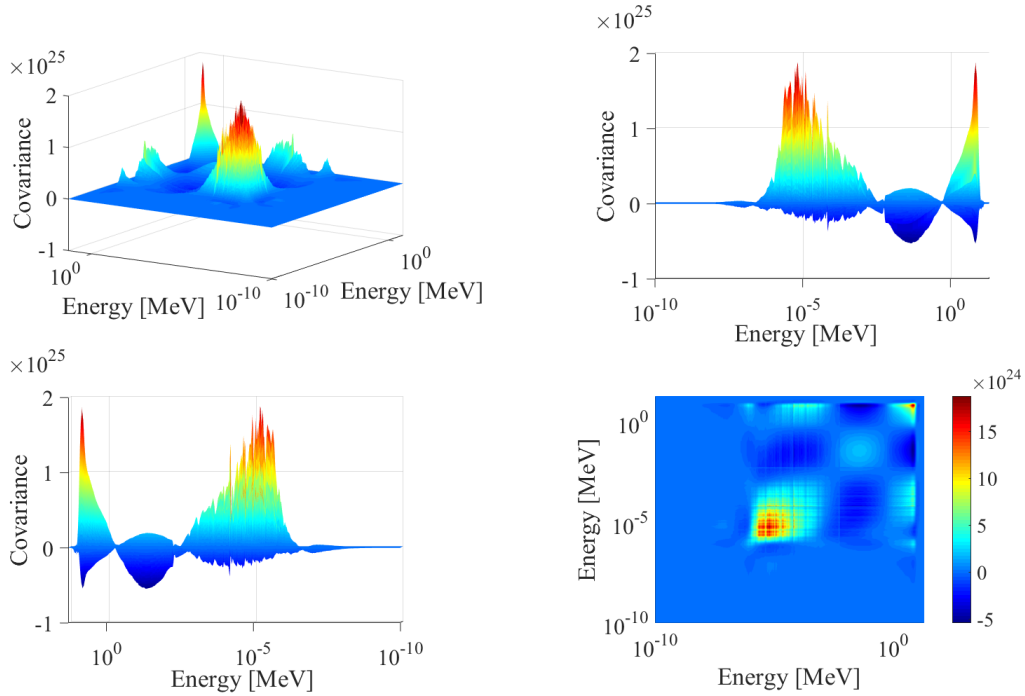

Figure 4.18: 3D and 2D Top and Side Views of the 4th Covariance.

These results are not as expected. While the initial peak in the high energy region is present, other peaks have formed including negative covariance values. The possible explanation of the large peak in the thermal region is the removal of failed files [10]. This will often lead to a covariance looking odd as more data is missing from the analysis. While this second run may not have the best results, it does enlighten that while these files have failed, they should not be removed from the analysis [10]. A greater problem has been encountered in GenSpec in the global sensitivity analysis, that may necessitate it being fixed. This sensitivity analysis has indeed illuminated many issues and positive aspects of GenSpec, and has also highlighted ways to improve and further optimize the solution in future work and in the following covariance work.

4.2 GenSpec Covariance

For the GenSpec covariance random sampling, the suggested *C* component stated in the Methodology section was attempted. These attempted additions to the core code can be found in Appendix D and all its subsection appendices. All of these changes remain in the GenSpec *C* project scripts but have been commented out. The addition to the *PERL* script was left 'as is', since the activity error file will still be used by the external *MATLAB* script.

Bugs in the final program were an overwhelming issue. The additions sparked problems in not only the *C* components of GenSpec, it had a cumulative effect on the *PERL* and *Python* file writing scripts. Problems also occurred where the entire GenSpec program would run up to the last generation, and subsequently fail to save the data due to an error encountered in the *GenSpec.exe* file. This issue could not be solved, and due to the constraint on editing the *FluxPRO.f* components and time, the proposed separate *MATLAB* file was ultimately chosen. This was also a better option as changes now need to be made in GenSpec core code for the polynomial order error.

For the external *MATLAB* script *gencov.m*, the trial covariance was found from the trial correlation matrix that was located in the *filename.cov* file under the working directory folder of GenSpec. Once the covariance was computed, the multivariate normal distribution sampling was done using *MATLAB*'s $mvnrnd()$ function. For the trial spectrum and the cross-section data, 300 different realizations of these distributions were made, and the input files were written and stored in the library folder for GenSpec. Two images of these 300 realizations can be found below. In Figure 4.19, an example of the realizations of the 23Na(n,γ)24Na reaction from the International Reactor Dosimetry Fusion and Fission database (IRDFF) is shown. Figure 4.20 shows the realizations of the trial spectrum for the differential neutron energy fluence. The fluence is now displayed in its differential

form since GenSpec requires the fluence to be input as such. Activities were varied about their standard deviations using a random number generator and scaling to the defined limits. An additional condition that had to be met was all the realizations had to be positive. To over come this issue, the realizations and spectra where normalized to their maximum, then sampled and unnormalized again to ensure the results were positive. A secondary *MATLAB* file was also written to write all of the input files. These *MATLAB* scripts can be found in Appendix E, and will include a sample of the file writing script.

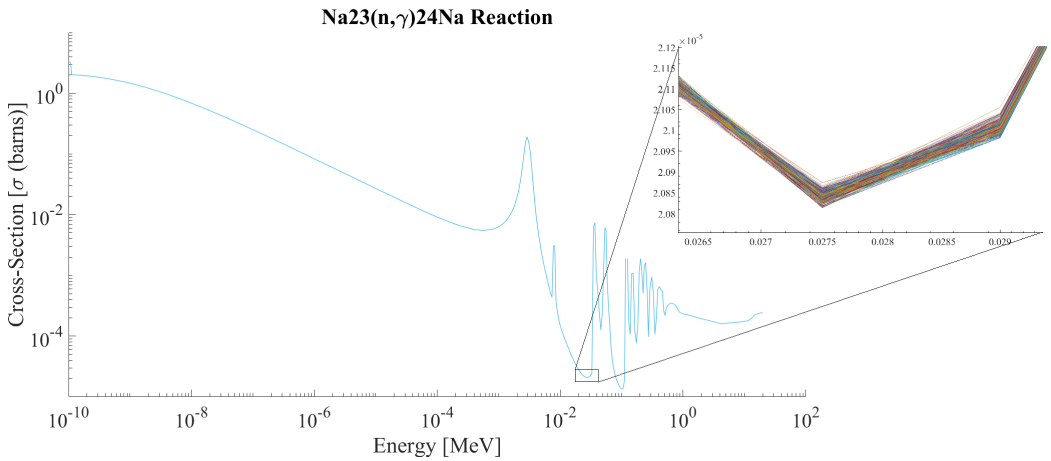

Figure 4.19: Realizations of the IRDFF Na23(n,γ)24Na Reaction Cross-Section.

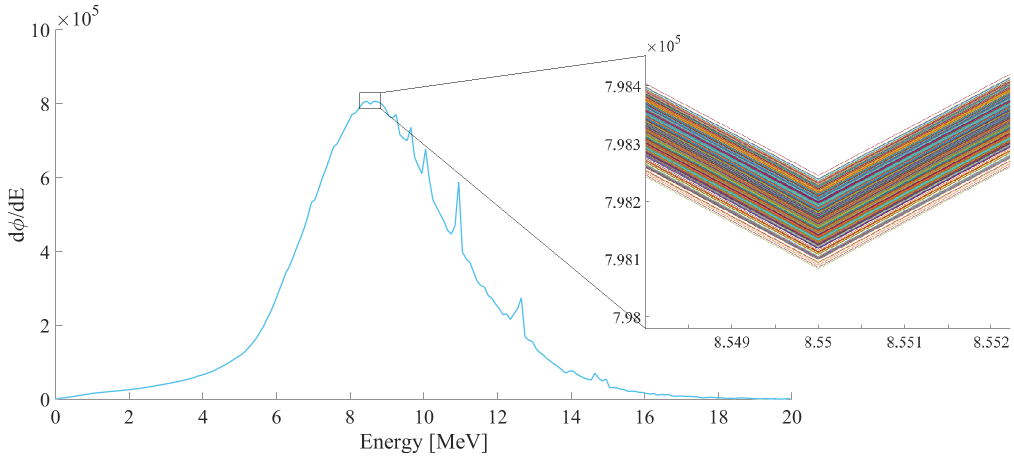

Figure 4.20: Realizations of the Differential Neutron Energy Fluence.

Again, GenSpec underwent another random perturbation process to determine the program's covariance. The goal was to reach at least 100 file runs, and for these runs, data files were not omitted for 'failing' as this can skew the results of the covariance as noted in the previous results section. In total, 1.222×10^9 spectra were adjusted for these results. *MATLAB* was also used for the post processing of the data. All the normalized spectra were loaded into a large matrix, and *MATLAB*'s $cov()$ function was simply applied to this matrix to produce the finalized covariance. Below in Figure 4.21 is a surface image of the GenSpec covariance after the random perturbation. In order to understand the produced covariance, the original trial covariance is also shown below in Figure 4.22.

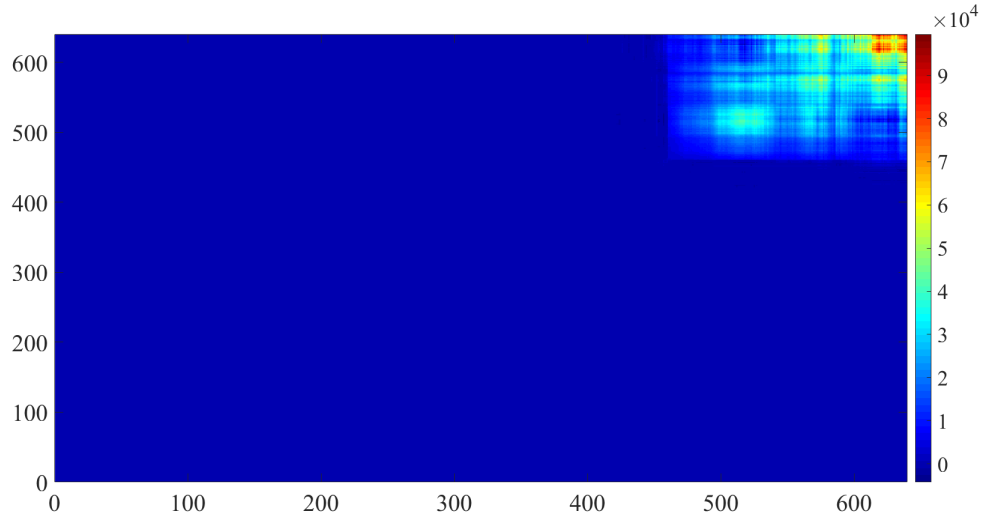

Figure 4.21: The mapped GenSpec Covariance Surface Plot.

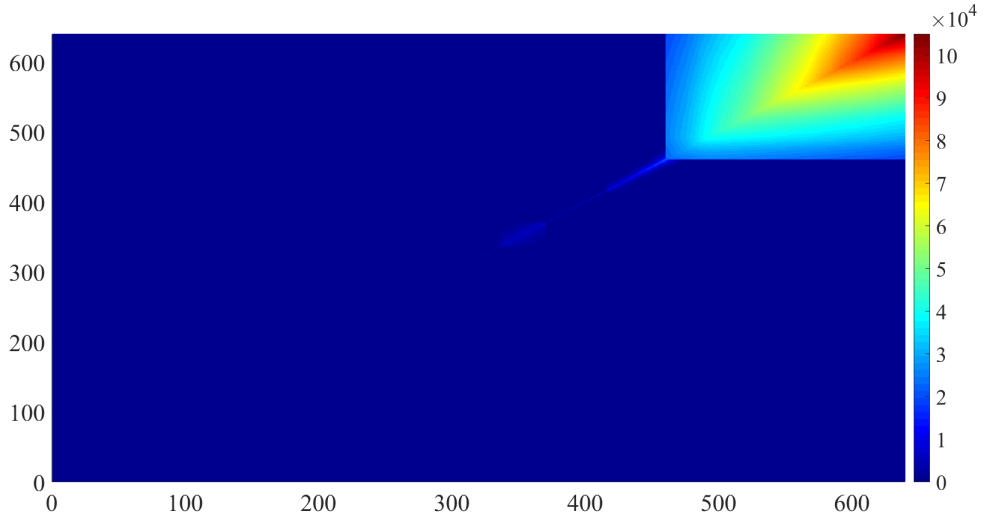

Figure 4.22: The mapped MCNP Trial Covariance Surface Plot.

As previously mentioned in the problem statement, the goals of GenSpec, and adjustment codes are two-fold. One of these goals is that an adjusted spectrum should contain

smaller or minimized covariance values than those associated with the calculated spectrum [3]. In roughly comparing the two above covariances maps, we can see what looks to be minimized covariance values of the GenSpec covariance map compared to the trial spectrum covariance map, as both figures have similar numeric scaling. At around energy group 475, which corresponds to an energy above 15 MeV, the GenSpec spectrum has a patched patten of striped smaller covariance values throughout. Additionally, the GenSpec spectrum should display a reduction in the difference between the calculated and measured reaction rates of the final spectrum.

While the visual structure of the GenSpec covariance map is odd, it is not necessarily a problem. GenSpec's continuous adjustment of the trial spectra though 'evolving' spectra and the use of a polynomial shift function may be the reason behind the results. As continued adjustments take place across any given spectrum, the values change as each generation of spectra evolve, and these adjustments can occur in a staggered fashion across several energy bins. The more important aspect of this comparison is that additional uncertainty or increases in covariance values are not introduced by GenSpec. Both covariances are predominately zero. A covariance of zero does not necessarily mean that the variables are independent [10][11]. A nonlinear relationship can still exist that would result in a covariance value of zero, but such studies are outside the scope of this research. As it stands GenSpec has minimized to occurrence of unrealistic spectral artifacts that were seen in the trial spectra. GenSpec may also reduce uncertainty in the produced spectra, but further statistical analysis is needed to determine this [11]. GenSpec's adjustment of an MCNP trial spectrum through unfolded reaction rates introduces additional data to the problem. The addition of further measured data, taken at a different point in time does affect the uncertainty in the program.

5. CONCLUSION

In conclusion, the covariance map for GenSpec was obtained. While a complete reduction in covariance is not statistically possible, there are marked points across the covariance map were smaller covariance values have been achieved. This final step to the completion of the uncertainty analysis for GenSpec, brings it closer to being ready as a software package. While the covariance map for the program has been found, this does not negate the other issues encountered in the global and local sensitivity analysis.

Future work to this research includes the updating of the polynomial regression fit of the shift function to a natural interpolant spline. In addition, it is desired to improve upon the *LSL-M2* legacy code components used. These sections in GenSpec have proved in this research to be problem areas. The polynomial order may indeed be the greatest fault in this program, as given any random set of inputs, the 'failure' is as high as 53%. Additionally, the *LSL-M2* component *FluxPRO.f* has several hundreds of lines of code doing calculations that are not needed or used by GenSpec. By modernizing this section of GenSpec, it can be streamlined, and make edits across any part of the program easy.

With these noted changes to the code, this work will most likely be revised in the future. An update to the genetic algorithm will require the covariance characterization via random perturbation to be repeated, as a further minimization in the covariance could occur.

REFERENCES

[1] R. M. Vega, "GenSpec: A Genetic Algorithm for Neutron Energy Spectrum Adjustment SAND2014-0024," tech. rep., Sandia National Laboratories., NM (USA), 2014.

[2] G. F. Knoll, *Radiation Detection and Measurement*. John Wiley & Sons, New York, 4th ed., 2010.

[3] W. L. Zipj, "Comparison of Neutron Spectrum Unfolding codes," tech. rep., International Atomic Energy Agency., WIE (AUT), 1978.

[4] F. G. Perey, "Spectrum Unfolding by the Least-Squares methods," tech. rep., Oak Ridge National Lab., TN (USA), 1977.

[5] A. E. Nix and M. D. Vose, "Modeling Genetic Algorithms with Markov Chains," *Annals of Mathematics and Artificial Intelligence*, vol. 5, pp. 79–88, 1992.

[6] M. D. Vose, "Modeling Simple Genetic Algorithms," *Foundations of Genetic Algorithms*, vol. 2, pp. 94–101, 1993.

[7] L. Davis, *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.

[8] L. M. Schmitt, "Theory of Genetic Algorithms," *Theoretical Computer Science*, vol. 259, pp. 1–61, 2001.

[9] F. W. Stallmann, "LSL-M2: A Computer Program for Least-Squares Logarithmic Adjustment of Neutron Spectra," tech. rep., Oak Ridge National Lab., TN (USA), 1986.

[10] J. Taylor, *Introduction to Error Analysis, the Study of Uncertainties in Physical Measurements*. University Science Books, Virginia, 1997.

[11] R. C. Smith, *Uncertainty Quantification: Theory, Implementation, and Applications*. SIAM, North Carolina, 2013.

[12] M. Matzke, "Propagation of Uncertainties in Unfolding Procedures," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 476, pp. 230–241, 2002.

[13] M. Reginatto, "Overview of Spectral Unfolding Techniques and Uncertainty Estimation," *Radiation Measurements and Detection*, vol. 45, pp. 1323–1329, 2010.

[14] F. G. Perey, "Uncertainty Analysis of Dosimetry Spectrum Unfolding," tech. rep., Oak Ridge National Lab., TN (USA), 1977.

[15] K. R. DePriest, P. J. Cooper, and E. J. Parma, "MCNP/MCNPX Model of the Annular Core Research Reactor SAND2006-3067," tech. rep., Sandia National Laboratories., NM (USA), 2006.

[16] E. M. Zsolnay, N. R. Capote, H. J. Nolthenius, and A. Trkov, "Summary Description of the new International Reactor Dosimetry and Fusion File (IRDFF release 1.0)," tech. rep., International Atomic Energy Agency, WIE (AUT), 2012.

[17] P. J. Griffin and J. G. Kelly, "A Rigorous Treatment of Self-Shielding and Covers in Neutron Spectra Determinations," *IEEE Transactions on Nuclear Science*, vol. 42, pp. 1878–1885, 1995.

[18] E. J. Parma, G. E. Naranjo, R. M. Vega, L. E. Lippert, D. W. Vehar, and P. J. Griffin, "Radiation Characterization Summary: ACRR Central Cavity Free-Field Environment with the 32-inch pedestal at the Core Centerline (ACRR-FF-CC-32-cl) SAND2015-6483.," tech. rep., Sandia National Laboratories., NM (USA), 2015.

[19] MathWorks Inc., *MATLAB: the Language of Technical Computing, Desktop Tools and Development Environment*. MathWorks, Massachusetts, 7th ed., 2005.

[20] M. Williams, *Microsoft Visual C/C++ (core reference)*. Microsoft Press, Washington, 2002.

[21] W. N. McElroy, J. S. Berg, T. Crockett, and R. G. Hawkins, "A Computer-Automated Iterative Method for Neutron Flux Spectra Determination by Foil Activation," tech. rep., Kirtland Air Force Base Report (AFWL-TR)., NM (USA), 1967.

[22] F. B. Brown and et.al, "MCNP: A General Monte Carlo N-Particle Transport Code, Version 5," tech. rep., Los Alamos National Laboratory., NM (USA), 2003.

[23] G. Stankunas, P. Batistoni, H. Sjöstrand, S. Conroy, and J. Contributors, "Measurements of Neutron Yields by Neutron Activation Technique: Uncertainty due to the Uncertainty on Activation Cross-Sections," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 788, pp. 168–172, 2015.

APPENDIX A

ACRR MCNP SAMPLE TALLY ADDITIONS

These scripts are the additions to the verified MCNP model what were made for this project.

```
C  Modified by D. Redhouse 05/2016
C ---------------------------------------------------------------
C  EXPERIMENTAL MATERIAL DATA CARDS
C   31 Material Response Functions using IRDFF v1.05 XS-Data
C   Material Card Label = mAAXX# -> AA - Isotope Proton Number (Last Two Digits)
C                                   XX - Periodic Number of Element
C                                    # - Data Library Used;1=IRDFF 1.05
C                                                          2=LLNL-ACTL
C                                                          3=ENDF/B-VII.1
C      EX: Na-23 --> m2311  OR Pu-239 --> m3994
C   23Na(n,gamma)24Na
m23111  11023.10y  1.0
m23112  11023.30y  1.0
m23113  11023.70c  1.0
C
C   24Mg(n,p)24Na
m24121  12024.10y  1.0
m24122  12024.30y  1.0
m24123  12024.70c  1.0
C
C   27Al(n,alpha)24Na & 27Al(n,p)27Mg
m27131  13027.10y  1.0
m27132  13027.30y  1.0
m27133  13027.70c  1.0
C
C   45Sc(n,gamma)46Sc
m45211  21045.10y  1.0
m45212  21045.30y  1.0
m45213  21045.70c  1.0
C
C   46Ti(n,p)46Sc
```

```
m46221  22046.10y  1.0
m46222  22046.30y  1.0
m46223  22046.70c  1.0
C
C   47Ti(n,p)47Sc
m47221  22047.10y  1.0
m47222  22047.30y  1.0
m47223  22047.70c  1.0
C
C   48Ti(n,p)48Sc
m48221  22048.10y  1.0
m48222  22048.30y  1.0
m48223  22048.70c  1.0
C
C   55Mn(n,gamma)56Mn & 55Mn(n,2n)54Mn
m55251  25055.10y  1.0
m55252  25055.30y  1.0
m55253  25055.70c  1.0
C
C   54Fe(n,p)54Mn
m54261  26054.10y  1.0
m54262  26054.30y  1.0
m54263  26054.70c  1.0
C
C   56Fe(n,p)56Mn
m56261  26056.10y  1.0
m56262  26056.30y  1.0
m56263  26056.70c  1.0
C
C   58Fe(n,gamma)59Fe
m58261  26058.10y  1.0
m58262  26058.30y  1.0
m58263  26058.70c  1.0
C
C   59Co(n,p)59Fe & 59Co(n,gamma)60Co & 59Co(n,2n)58Co
m59271  27059.10y  1.0
m59272  27059.30y  1.0
m59273  27059.70c  1.0
C
```

```
C   58Ni(n,2n)57Ni & 58Ni(n,p)58Co
m58281  28058.10y  1.0
m58282  28058.30y  1.0
m58283  28058.70c  1.0
C
C   60Ni(n,p)60Co
m60281  28060.10y  1.0
m60282  28060.30y  1.0
m60283  28060.70c  1.0
C
C   63Cu(n,gamma)64Cu & 63Cu(n,alpha)60Co
m63291  29063.10y  1.0
m63292  29063.30y  1.0
m63293  29063.70c  1.0
C
C   64Zn(n,p)64Cu
m64291  30064.10y  1.0
m64292  30064.30y  1.0
m64293  30064.80c  1.0
C
C   90Zr(n,2n)89Zr
m90401  40090.10y  1.0
m90402  40090.30y  1.0
m90403  40090.70c  1.0
C
C   93Nb(n,gamma)94Nb & 93Nb(n,2n)92mNb
m93411  41093.10y  1.0
m93412  41093.30y  1.0
m93413  41093.70c  1.0
C
C   98Mo(n,gamma)99Mo  **LLNL/ACTL 1983 Library - IRDFF does not have Mo-98**
m98422  42098.30y  1.0
m98423  42098.70c  1.0
C
C   109Ag(n,gamma)110mAg  **LLNL/ACTL 1983 Library**
m19472  47109.30y  1.0
m19473  47109.70c  1.0
C
C   115In(n,gamma)116mIn & 115In(n,nâĂŹ)115mIn
```

```
m15491  49115.10y  1.0
m15492  49115.30y  1.0
m15493  49115.70c  1.0
C
C   186W(n,gamma)187W
m86741  74186.10y  1.0
m86742  74186.30y  1.0
m86743  74186.70c  1.0
C
C   197Au(n,gamma)198Au
m97791  79197.10y  1.0
m97792  79197.30y  1.0
m97793  79197.70c  1.0
C
.
.
.
C ********************
C * NEUTRON TALLIES
C ********************
F14:n 1000
FC14   Neutron 640 Group Fluence (n/cm^2)/MJ
E14   1.000E-10  1.050E-10  1.100E-10  1.150E-10  1.200E-10  1.275E-10
1.350E-10  1.425E-10  1.500E-10  1.600E-10  1.700E-10  1.800E-10
1.900E-10  2.000E-10  2.100E-10  2.200E-10  2.300E-10  2.400E-10
2.550E-10  2.700E-10  2.800E-10  3.000E-10  3.200E-10  3.400E-10
3.600E-10  3.800E-10  4.000E-10  4.250E-10  4.500E-10  4.750E-10
5.000E-10  5.250E-10  5.500E-10  5.750E-10  6.000E-10  6.300E-10
6.600E-10  6.900E-10  7.200E-10  7.600E-10  8.000E-10  8.400E-10
8.800E-10  9.200E-10  9.600E-10  1.000E-09  1.050E-09  1.100E-09
1.150E-09  1.200E-09  1.275E-09  1.350E-09  1.425E-09  1.500E-09
1.600E-09  1.700E-09  1.800E-09  1.900E-09  2.000E-09  2.100E-09
2.200E-09  2.300E-09  2.400E-09  2.550E-09  2.700E-09  2.800E-09
3.000E-09  3.200E-09  3.400E-09  3.600E-09  3.800E-09  4.000E-09
4.250E-09  4.500E-09  4.750E-09  5.000E-09  5.250E-09  5.500E-09
5.750E-09  6.000E-09  6.300E-09  6.600E-09  6.900E-09  7.200E-09
7.600E-09  8.000E-09  8.400E-09  8.800E-09  9.200E-09  9.600E-09
1.000E-08  1.050E-08  1.100E-08  1.150E-08  1.200E-08  1.275E-08
1.350E-08  1.425E-08  1.500E-08  1.600E-08  1.700E-08  1.800E-08
```

1.900E-08	2.000E-08	2.100E-08	2.200E-08	2.300E-08	2.400E-08
2.550E-08	2.700E-08	2.800E-08	3.000E-08	3.200E-08	3.400E-08
3.600E-08	3.800E-08	4.000E-08	4.250E-08	4.500E-08	4.750E-08
5.000E-08	5.250E-08	5.500E-08	5.750E-08	6.000E-08	6.300E-08
6.600E-08	6.900E-08	7.200E-08	7.600E-08	8.000E-08	8.400E-08
8.800E-08	9.200E-08	9.600E-08	1.000E-07	1.050E-07	1.100E-07
1.150E-07	1.200E-07	1.275E-07	1.350E-07	1.425E-07	1.500E-07
1.600E-07	1.700E-07	1.800E-07	1.900E-07	2.000E-07	2.100E-07
2.200E-07	2.300E-07	2.400E-07	2.550E-07	2.700E-07	2.800E-07
3.000E-07	3.200E-07	3.400E-07	3.600E-07	3.800E-07	4.000E-07
4.250E-07	4.500E-07	4.750E-07	5.000E-07	5.250E-07	5.500E-07
5.750E-07	6.000E-07	6.300E-07	6.600E-07	6.900E-07	7.200E-07
7.600E-07	8.000E-07	8.400E-07	8.800E-07	9.200E-07	9.600E-07
1.000E-06	1.050E-06	1.100E-06	1.150E-06	1.200E-06	1.275E-06
1.350E-06	1.425E-06	1.500E-06	1.600E-06	1.700E-06	1.800E-06
1.900E-06	2.000E-06	2.100E-06	2.200E-06	2.300E-06	2.400E-06
2.550E-06	2.700E-06	2.800E-06	3.000E-06	3.200E-06	3.400E-06
3.600E-06	3.800E-06	4.000E-06	4.250E-06	4.500E-06	4.750E-06
5.000E-06	5.250E-06	5.500E-06	5.750E-06	6.000E-06	6.300E-06
6.600E-06	6.900E-06	7.200E-06	7.600E-06	8.000E-06	8.400E-06
8.800E-06	9.200E-06	9.600E-06	1.000E-05	1.050E-05	1.100E-05
1.150E-05	1.200E-05	1.275E-05	1.350E-05	1.425E-05	1.500E-05
1.600E-05	1.700E-05	1.800E-05	1.900E-05	2.000E-05	2.100E-05
2.200E-05	2.300E-05	2.400E-05	2.550E-05	2.700E-05	2.800E-05
3.000E-05	3.200E-05	3.400E-05	3.600E-05	3.800E-05	4.000E-05
4.250E-05	4.500E-05	4.750E-05	5.000E-05	5.250E-05	5.500E-05
5.750E-05	6.000E-05	6.300E-05	6.600E-05	6.900E-05	7.200E-05
7.600E-05	8.000E-05	8.400E-05	8.800E-05	9.200E-05	9.600E-05
1.000E-04	1.050E-04	1.100E-04	1.150E-04	1.200E-04	1.275E-04
1.350E-04	1.425E-04	1.500E-04	1.600E-04	1.700E-04	1.800E-04
1.900E-04	2.000E-04	2.100E-04	2.200E-04	2.300E-04	2.400E-04
2.550E-04	2.700E-04	2.800E-04	3.000E-04	3.200E-04	3.400E-04
3.600E-04	3.800E-04	4.000E-04	4.250E-04	4.500E-04	4.750E-04
5.000E-04	5.250E-04	5.500E-04	5.750E-04	6.000E-04	6.300E-04
6.600E-04	6.900E-04	7.200E-04	7.600E-04	8.000E-04	8.400E-04
8.800E-04	9.200E-04	9.600E-04	1.000E-03	1.050E-03	1.100E-03
1.150E-03	1.200E-03	1.275E-03	1.350E-03	1.425E-03	1.500E-03
1.600E-03	1.700E-03	1.800E-03	1.900E-03	2.000E-03	2.100E-03
2.200E-03	2.300E-03	2.400E-03	2.550E-03	2.700E-03	2.800E-03

3.000E-03	3.200E-03	3.400E-03	3.600E-03	3.800E-03	4.000E-03
4.250E-03	4.500E-03	4.750E-03	5.000E-03	5.250E-03	5.500E-03
5.750E-03	6.000E-03	6.300E-03	6.600E-03	6.900E-03	7.200E-03
7.600E-03	8.000E-03	8.400E-03	8.800E-03	9.200E-03	9.600E-03
1.000E-02	1.050E-02	1.100E-02	1.150E-02	1.200E-02	1.275E-02
1.350E-02	1.425E-02	1.500E-02	1.600E-02	1.700E-02	1.800E-02
1.900E-02	2.000E-02	2.100E-02	2.200E-02	2.300E-02	2.400E-02
2.550E-02	2.700E-02	2.800E-02	3.000E-02	3.200E-02	3.400E-02
3.600E-02	3.800E-02	4.000E-02	4.250E-02	4.500E-02	4.750E-02
5.000E-02	5.250E-02	5.500E-02	5.750E-02	6.000E-02	6.300E-02
6.600E-02	6.900E-02	7.200E-02	7.600E-02	8.000E-02	8.400E-02
8.800E-02	9.200E-02	9.600E-02	1.000E-01	1.050E-01	1.100E-01
1.150E-01	1.200E-01	1.275E-01	1.350E-01	1.425E-01	1.500E-01
1.600E-01	1.700E-01	1.800E-01	1.900E-01	2.000E-01	2.100E-01
2.200E-01	2.300E-01	2.400E-01	2.550E-01	2.700E-01	2.800E-01
3.000E-01	3.200E-01	3.400E-01	3.600E-01	3.800E-01	4.000E-01
4.250E-01	4.500E-01	4.750E-01	5.000E-01	5.250E-01	5.500E-01
5.750E-01	6.000E-01	6.300E-01	6.600E-01	6.900E-01	7.200E-01
7.600E-01	8.000E-01	8.400E-01	8.800E-01	9.200E-01	9.600E-01
1.000E+00	1.100E+00	1.200E+00	1.300E+00	1.400E+00	1.500E+00
1.600E+00	1.700E+00	1.800E+00	1.900E+00	2.000E+00	2.100E+00
2.200E+00	2.300E+00	2.400E+00	2.500E+00	2.600E+00	2.700E+00
2.800E+00	2.900E+00	3.000E+00	3.100E+00	3.200E+00	3.300E+00
3.400E+00	3.500E+00	3.600E+00	3.700E+00	3.800E+00	3.900E+00
4.000E+00	4.100E+00	4.200E+00	4.300E+00	4.400E+00	4.500E+00
4.600E+00	4.700E+00	4.800E+00	4.900E+00	5.000E+00	5.100E+00
5.200E+00	5.300E+00	5.400E+00	5.500E+00	5.600E+00	5.700E+00
5.800E+00	5.900E+00	6.000E+00	6.100E+00	6.200E+00	6.300E+00
6.400E+00	6.500E+00	6.600E+00	6.700E+00	6.800E+00	6.900E+00
7.000E+00	7.100E+00	7.200E+00	7.300E+00	7.400E+00	7.500E+00
7.600E+00	7.700E+00	7.800E+00	7.900E+00	8.000E+00	8.100E+00
8.200E+00	8.300E+00	8.400E+00	8.500E+00	8.600E+00	8.700E+00
8.800E+00	8.900E+00	9.000E+00	9.100E+00	9.200E+00	9.300E+00
9.400E+00	9.500E+00	9.600E+00	9.700E+00	9.800E+00	9.900E+00
1.000E+01	1.010E+01	1.020E+01	1.030E+01	1.040E+01	1.050E+01
1.060E+01	1.070E+01	1.080E+01	1.090E+01	1.100E+01	1.110E+01
1.120E+01	1.130E+01	1.140E+01	1.150E+01	1.160E+01	1.170E+01
1.180E+01	1.190E+01	1.200E+01	1.210E+01	1.220E+01	1.230E+01
1.240E+01	1.250E+01	1.260E+01	1.270E+01	1.280E+01	1.290E+01

```
1.300E+01  1.310E+01  1.320E+01  1.330E+01  1.340E+01  1.350E+01
1.360E+01  1.370E+01  1.380E+01  1.390E+01  1.400E+01  1.410E+01
1.420E+01  1.430E+01  1.440E+01  1.450E+01  1.460E+01  1.470E+01
1.480E+01  1.490E+01  1.500E+01  1.510E+01  1.520E+01  1.530E+01
1.540E+01  1.550E+01  1.560E+01  1.570E+01  1.580E+01  1.590E+01
1.600E+01  1.610E+01  1.620E+01  1.630E+01  1.640E+01  1.650E+01
1.660E+01  1.670E+01  1.680E+01  1.690E+01  1.700E+01  1.710E+01
1.720E+01  1.730E+01  1.740E+01  1.750E+01  1.760E+01  1.770E+01
1.780E+01  1.790E+01  1.800E+01  1.810E+01  1.820E+01  1.830E+01
1.840E+01  1.850E+01  1.860E+01  1.870E+01  1.880E+01  1.890E+01
1.900E+01  1.910E+01  1.920E+01  1.930E+01  1.940E+01  1.950E+01
1.960E+01  1.970E+01  1.980E+01  1.990E+01  2.000E+01
C
F24:n 1000
FC24   Total Neutron Fluence (n/cm^2)/MJ
C
.
.
.
C ****************************************************************************
C *                                                                          *
C * RESPONSE FUNCTIONS                                                       *
C *  use correct volume of detector to get flux and response correct         *
C *  for a 1 in col one, units are in per atom/b-cm per source neutron       *
C *  use rho * No * 10^-24 / A to get units in reactions/cm^3 per            *
C *  source neutron use No * 10^-24 / A to get units in reactions/gm per     *
C *  source neutrons use 6.0221e23 for No for atomic mass from chart of      *
C *  nuclides C-12 = 12.0000 multiply by ln2/half-life to get activity       *
C *  per source neutron                                                      *
C *                                                                          *
C *  Response Function Labeling Explained                                    *
C *  F1#XX4 - XX spans from 00 to 31 response funtions                       *
C *        # spans from 1 to 3 based on library used (See mcard definition)  *
C *                                                                          *
C *  E card is the 640 group  neutron energy spectrum                        *
C *                                                                          *
C *  EM card is the tally material self-shielding factors or each e-group    *
C *                                                                          *
C ****************************************************************************
```

```
C
C 23Na(n,Îş)24Na
C   Material = m2311
C          A =  22.989
C       T1/2 =  14.9512 hr = 5.38E+04 sec
C  Multiplier =  3.2314E-07
C
F11004:n 1000
FC11004  23Na(n,gamma)24Na (IRDFF) d/s/gm/n
FM11004  3.2314e-07 23111 102
E11004   1.000E-10  1.050E-10  1.100E-10  1.150E-10  1.200E-10  1.275E-10
1.350E-10  1.425E-10  1.500E-10  1.600E-10  1.700E-10  1.800E-10
1.900E-10  2.000E-10  2.100E-10  2.200E-10  2.300E-10  2.400E-10
2.550E-10  2.700E-10  2.800E-10  3.000E-10  3.200E-10  3.400E-10
3.600E-10  3.800E-10  4.000E-10  4.250E-10  4.500E-10  4.750E-10
5.000E-10  5.250E-10  5.500E-10  5.750E-10  6.000E-10  6.300E-10
6.600E-10  6.900E-10  7.200E-10  7.600E-10  8.000E-10  8.400E-10
8.800E-10  9.200E-10  9.600E-10  1.000E-09  1.050E-09  1.100E-09
1.150E-09  1.200E-09  1.275E-09  1.350E-09  1.425E-09  1.500E-09
1.600E-09  1.700E-09  1.800E-09  1.900E-09  2.000E-09  2.100E-09
2.200E-09  2.300E-09  2.400E-09  2.550E-09  2.700E-09  2.800E-09
3.000E-09  3.200E-09  3.400E-09  3.600E-09  3.800E-09  4.000E-09
4.250E-09  4.500E-09  4.750E-09  5.000E-09  5.250E-09  5.500E-09
5.750E-09  6.000E-09  6.300E-09  6.600E-09  6.900E-09  7.200E-09
7.600E-09  8.000E-09  8.400E-09  8.800E-09  9.200E-09  9.600E-09
1.000E-08  1.050E-08  1.100E-08  1.150E-08  1.200E-08  1.275E-08
1.350E-08  1.425E-08  1.500E-08  1.600E-08  1.700E-08  1.800E-08
1.900E-08  2.000E-08  2.100E-08  2.200E-08  2.300E-08  2.400E-08
2.550E-08  2.700E-08  2.800E-08  3.000E-08  3.200E-08  3.400E-08
3.600E-08  3.800E-08  4.000E-08  4.250E-08  4.500E-08  4.750E-08
5.000E-08  5.250E-08  5.500E-08  5.750E-08  6.000E-08  6.300E-08
6.600E-08  6.900E-08  7.200E-08  7.600E-08  8.000E-08  8.400E-08
8.800E-08  9.200E-08  9.600E-08  1.000E-07  1.050E-07  1.100E-07
1.150E-07  1.200E-07  1.275E-07  1.350E-07  1.425E-07  1.500E-07
1.600E-07  1.700E-07  1.800E-07  1.900E-07  2.000E-07  2.100E-07
2.200E-07  2.300E-07  2.400E-07  2.550E-07  2.700E-07  2.800E-07
3.000E-07  3.200E-07  3.400E-07  3.600E-07  3.800E-07  4.000E-07
4.250E-07  4.500E-07  4.750E-07  5.000E-07  5.250E-07  5.500E-07
5.750E-07  6.000E-07  6.300E-07  6.600E-07  6.900E-07  7.200E-07
```

7.600E-07	8.000E-07	8.400E-07	8.800E-07	9.200E-07	9.600E-07
1.000E-06	1.050E-06	1.100E-06	1.150E-06	1.200E-06	1.275E-06
1.350E-06	1.425E-06	1.500E-06	1.600E-06	1.700E-06	1.800E-06
1.900E-06	2.000E-06	2.100E-06	2.200E-06	2.300E-06	2.400E-06
2.550E-06	2.700E-06	2.800E-06	3.000E-06	3.200E-06	3.400E-06
3.600E-06	3.800E-06	4.000E-06	4.250E-06	4.500E-06	4.750E-06
5.000E-06	5.250E-06	5.500E-06	5.750E-06	6.000E-06	6.300E-06
6.600E-06	6.900E-06	7.200E-06	7.600E-06	8.000E-06	8.400E-06
8.800E-06	9.200E-06	9.600E-06	1.000E-05	1.050E-05	1.100E-05
1.150E-05	1.200E-05	1.275E-05	1.350E-05	1.425E-05	1.500E-05
1.600E-05	1.700E-05	1.800E-05	1.900E-05	2.000E-05	2.100E-05
2.200E-05	2.300E-05	2.400E-05	2.550E-05	2.700E-05	2.800E-05
3.000E-05	3.200E-05	3.400E-05	3.600E-05	3.800E-05	4.000E-05
4.250E-05	4.500E-05	4.750E-05	5.000E-05	5.250E-05	5.500E-05
5.750E-05	6.000E-05	6.300E-05	6.600E-05	6.900E-05	7.200E-05
7.600E-05	8.000E-05	8.400E-05	8.800E-05	9.200E-05	9.600E-05
1.000E-04	1.050E-04	1.100E-04	1.150E-04	1.200E-04	1.275E-04
1.350E-04	1.425E-04	1.500E-04	1.600E-04	1.700E-04	1.800E-04
1.900E-04	2.000E-04	2.100E-04	2.200E-04	2.300E-04	2.400E-04
2.550E-04	2.700E-04	2.800E-04	3.000E-04	3.200E-04	3.400E-04
3.600E-04	3.800E-04	4.000E-04	4.250E-04	4.500E-04	4.750E-04
5.000E-04	5.250E-04	5.500E-04	5.750E-04	6.000E-04	6.300E-04
6.600E-04	6.900E-04	7.200E-04	7.600E-04	8.000E-04	8.400E-04
8.800E-04	9.200E-04	9.600E-04	1.000E-03	1.050E-03	1.100E-03
1.150E-03	1.200E-03	1.275E-03	1.350E-03	1.425E-03	1.500E-03
1.600E-03	1.700E-03	1.800E-03	1.900E-03	2.000E-03	2.100E-03
2.200E-03	2.300E-03	2.400E-03	2.550E-03	2.700E-03	2.800E-03
3.000E-03	3.200E-03	3.400E-03	3.600E-03	3.800E-03	4.000E-03
4.250E-03	4.500E-03	4.750E-03	5.000E-03	5.250E-03	5.500E-03
5.750E-03	6.000E-03	6.300E-03	6.600E-03	6.900E-03	7.200E-03
7.600E-03	8.000E-03	8.400E-03	8.800E-03	9.200E-03	9.600E-03
1.000E-02	1.050E-02	1.100E-02	1.150E-02	1.200E-02	1.275E-02
1.350E-02	1.425E-02	1.500E-02	1.600E-02	1.700E-02	1.800E-02
1.900E-02	2.000E-02	2.100E-02	2.200E-02	2.300E-02	2.400E-02
2.550E-02	2.700E-02	2.800E-02	3.000E-02	3.200E-02	3.400E-02
3.600E-02	3.800E-02	4.000E-02	4.250E-02	4.500E-02	4.750E-02
5.000E-02	5.250E-02	5.500E-02	5.750E-02	6.000E-02	6.300E-02
6.600E-02	6.900E-02	7.200E-02	7.600E-02	8.000E-02	8.400E-02
8.800E-02	9.200E-02	9.600E-02	1.000E-01	1.050E-01	1.100E-01

1.150E-01	1.200E-01	1.275E-01	1.350E-01	1.425E-01	1.500E-01
1.600E-01	1.700E-01	1.800E-01	1.900E-01	2.000E-01	2.100E-01
2.200E-01	2.300E-01	2.400E-01	2.550E-01	2.700E-01	2.800E-01
3.000E-01	3.200E-01	3.400E-01	3.600E-01	3.800E-01	4.000E-01
4.250E-01	4.500E-01	4.750E-01	5.000E-01	5.250E-01	5.500E-01
5.750E-01	6.000E-01	6.300E-01	6.600E-01	6.900E-01	7.200E-01
7.600E-01	8.000E-01	8.400E-01	8.800E-01	9.200E-01	9.600E-01
1.000E+00	1.100E+00	1.200E+00	1.300E+00	1.400E+00	1.500E+00
1.600E+00	1.700E+00	1.800E+00	1.900E+00	2.000E+00	2.100E+00
2.200E+00	2.300E+00	2.400E+00	2.500E+00	2.600E+00	2.700E+00
2.800E+00	2.900E+00	3.000E+00	3.100E+00	3.200E+00	3.300E+00
3.400E+00	3.500E+00	3.600E+00	3.700E+00	3.800E+00	3.900E+00
4.000E+00	4.100E+00	4.200E+00	4.300E+00	4.400E+00	4.500E+00
4.600E+00	4.700E+00	4.800E+00	4.900E+00	5.000E+00	5.100E+00
5.200E+00	5.300E+00	5.400E+00	5.500E+00	5.600E+00	5.700E+00
5.800E+00	5.900E+00	6.000E+00	6.100E+00	6.200E+00	6.300E+00
6.400E+00	6.500E+00	6.600E+00	6.700E+00	6.800E+00	6.900E+00
7.000E+00	7.100E+00	7.200E+00	7.300E+00	7.400E+00	7.500E+00
7.600E+00	7.700E+00	7.800E+00	7.900E+00	8.000E+00	8.100E+00
8.200E+00	8.300E+00	8.400E+00	8.500E+00	8.600E+00	8.700E+00
8.800E+00	8.900E+00	9.000E+00	9.100E+00	9.200E+00	9.300E+00
9.400E+00	9.500E+00	9.600E+00	9.700E+00	9.800E+00	9.900E+00
1.000E+01	1.010E+01	1.020E+01	1.030E+01	1.040E+01	1.050E+01
1.060E+01	1.070E+01	1.080E+01	1.090E+01	1.100E+01	1.110E+01
1.120E+01	1.130E+01	1.140E+01	1.150E+01	1.160E+01	1.170E+01
1.180E+01	1.190E+01	1.200E+01	1.210E+01	1.220E+01	1.230E+01
1.240E+01	1.250E+01	1.260E+01	1.270E+01	1.280E+01	1.290E+01
1.300E+01	1.310E+01	1.320E+01	1.330E+01	1.340E+01	1.350E+01
1.360E+01	1.370E+01	1.380E+01	1.390E+01	1.400E+01	1.410E+01
1.420E+01	1.430E+01	1.440E+01	1.450E+01	1.460E+01	1.470E+01
1.480E+01	1.490E+01	1.500E+01	1.510E+01	1.520E+01	1.530E+01
1.540E+01	1.550E+01	1.560E+01	1.570E+01	1.580E+01	1.590E+01
1.600E+01	1.610E+01	1.620E+01	1.630E+01	1.640E+01	1.650E+01
1.660E+01	1.670E+01	1.680E+01	1.690E+01	1.700E+01	1.710E+01
1.720E+01	1.730E+01	1.740E+01	1.750E+01	1.760E+01	1.770E+01
1.780E+01	1.790E+01	1.800E+01	1.810E+01	1.820E+01	1.830E+01
1.840E+01	1.850E+01	1.860E+01	1.870E+01	1.880E+01	1.890E+01
1.900E+01	1.910E+01	1.920E+01	1.930E+01	1.940E+01	1.950E+01
1.960E+01	1.970E+01	1.980E+01	1.990E+01	2.000E+01	

EM11004	2.626E-01	2.681E-01	2.732E-01	2.779E-01	2.840E-01	2.908E-01
2.973E-01	3.035E-01	3.102E-01	3.180E-01	3.253E-01	3.322E-01	
3.389E-01	3.453E-01	3.513E-01	3.571E-01	3.626E-01	3.695E-01	
3.774E-01	3.837E-01	3.905E-01	3.996E-01	4.080E-01	4.161E-01	
4.238E-01	4.311E-01	4.388E-01	4.468E-01	4.544E-01	4.620E-01	
4.690E-01	4.754E-01	4.818E-01	4.879E-01	4.940E-01	5.009E-01	
5.072E-01	5.130E-01	5.198E-01	5.272E-01	5.343E-01	5.410E-01	
5.469E-01	5.530E-01	5.584E-01	5.649E-01	5.713E-01	5.773E-01	
5.832E-01	5.897E-01	5.979E-01	6.050E-01	6.118E-01	6.195E-01	
6.276E-01	6.349E-01	6.422E-01	6.488E-01	6.548E-01	6.608E-01	
6.661E-01	6.716E-01	6.779E-01	6.847E-01	6.900E-01	6.960E-01	
7.035E-01	7.106E-01	7.173E-01	7.230E-01	7.290E-01	7.351E-01	
7.413E-01	7.469E-01	7.522E-01	7.568E-01	7.617E-01	7.662E-01	
7.706E-01	7.744E-01	7.790E-01	7.832E-01	7.867E-01	7.916E-01	
7.957E-01	8.004E-01	8.040E-01	8.078E-01	8.117E-01	8.149E-01	
8.184E-01	8.221E-01	8.257E-01	8.292E-01	8.328E-01	8.370E-01	
8.410E-01	8.447E-01	8.487E-01	8.530E-01	8.569E-01	8.603E-01	
8.634E-01	8.666E-01	8.697E-01	8.723E-01	8.747E-01	8.775E-01	
8.809E-01	8.833E-01	8.863E-01	8.897E-01	8.929E-01	8.960E-01	
8.985E-01	9.011E-01	9.036E-01	9.061E-01	9.085E-01	9.109E-01	
9.130E-01	9.148E-01	9.167E-01	9.186E-01	9.202E-01	9.221E-01	
9.239E-01	9.255E-01	9.270E-01	9.290E-01	9.305E-01	9.321E-01	
9.337E-01	9.349E-01	9.365E-01	9.379E-01	9.389E-01	9.407E-01	
9.417E-01	9.434E-01	9.449E-01	9.463E-01	9.478E-01	9.491E-01	
9.509E-01	9.523E-01	9.537E-01	9.548E-01	9.559E-01	9.571E-01	
9.580E-01	9.587E-01	9.599E-01	9.612E-01	9.621E-01	9.630E-01	
9.643E-01	9.651E-01	9.662E-01	9.673E-01	9.681E-01	9.690E-01	
9.700E-01	9.707E-01	9.716E-01	9.722E-01	9.730E-01	9.736E-01	
9.740E-01	9.748E-01	9.753E-01	9.758E-01	9.763E-01	9.768E-01	
9.773E-01	9.778E-01	9.783E-01	9.788E-01	9.791E-01	9.795E-01	
9.797E-01	9.800E-01	9.803E-01	9.806E-01	9.809E-01	9.814E-01	
9.817E-01	9.823E-01	9.828E-01	9.833E-01	9.842E-01	9.849E-01	
9.858E-01	9.867E-01	9.873E-01	9.879E-01	9.882E-01	9.886E-01	
9.891E-01	9.895E-01	9.900E-01	9.905E-01	9.908E-01	9.908E-01	
9.902E-01	9.897E-01	9.917E-01	9.924E-01	9.928E-01	9.930E-01	
9.933E-01	9.936E-01	9.938E-01	9.938E-01	9.944E-01	9.944E-01	
9.946E-01	9.949E-01	9.951E-01	9.951E-01	9.952E-01	9.953E-01	
9.935E-01	9.950E-01	9.961E-01	9.962E-01	9.964E-01	9.965E-01	
9.966E-01	9.966E-01	9.970E-01	9.972E-01	9.968E-01	9.975E-01	

9.976E-01	9.978E-01	9.980E-01	9.981E-01	9.982E-01	9.982E-01
9.977E-01	9.984E-01	9.985E-01	9.988E-01	9.990E-01	9.988E-01
9.990E-01	9.991E-01	9.993E-01	9.993E-01	9.990E-01	9.997E-01
9.997E-01	9.996E-01	9.997E-01	9.999E-01	9.999E-01	9.998E-01
1.000E+00	1.000E+00	1.000E+00	1.000E+00	9.999E-01	1.000E+00
1.000E+00	9.999E-01	1.000E+00	1.000E+00	1.000E+00	1.000E+00
1.000E+00	1.000E+00	1.000E+00	1.000E+00	1.000E+00	1.000E+00
1.001E+00	1.000E+00	1.001E+00	1.000E+00	1.000E+00	1.000E+00
1.000E+00	1.000E+00	1.000E+00	1.000E+00	1.001E+00	1.000E+00
1.000E+00	1.000E+00	1.000E+00	1.000E+00	1.001E+00	1.000E+00
9.999E-01	9.641E-01	9.967E-01	9.998E-01	9.999E-01	1.000E+00
1.000E+00	9.999E-01	9.999E-01	9.999E-01	9.996E-01	9.998E-01
9.998E-01	9.997E-01	9.996E-01	9.995E-01	9.995E-01	9.993E-01
9.992E-01	9.992E-01	9.992E-01	9.990E-01	9.992E-01	9.988E-01
9.985E-01	9.983E-01	9.979E-01	9.975E-01	9.971E-01	9.965E-01
9.953E-01	9.938E-01	9.915E-01	9.880E-01	9.829E-01	9.739E-01
9.591E-01	9.333E-01	8.648E-01	7.068E-01	5.782E-01	7.112E-01
1.262E+00	1.590E+00	1.530E+00	1.357E+00	1.218E+00	1.127E+00
1.073E+00	1.048E+00	1.035E+00	1.027E+00	1.022E+00	1.019E+00
1.018E+00	1.015E+00	1.014E+00	1.013E+00	1.012E+00	1.008E+00
9.835E-01	1.528E+00	1.240E+00	1.122E+00	1.017E+00	1.014E+00
1.010E+00	1.008E+00	1.007E+00	1.006E+00	1.006E+00	1.006E+00
1.005E+00	1.007E+00	1.005E+00	1.005E+00	1.004E+00	1.004E+00
1.003E+00	1.003E+00	1.003E+00	1.002E+00	1.002E+00	1.002E+00
1.001E+00	1.000E+00	9.998E-01	9.987E-01	9.960E-01	9.716E-01
4.754E+00	4.605E+00	2.695E+00	1.016E+00	9.906E-01	9.859E-01
9.674E-01	9.227E-01	1.110E+00	1.398E+00	1.449E+00	1.207E+00
1.069E+00	1.037E+00	1.021E+00	1.011E+00	1.008E+00	1.006E+00
1.004E+00	1.003E+00	1.002E+00	1.000E+00	9.984E-01	9.886E-01
9.737E-01	3.255E+00	3.673E+00	1.668E+00	9.624E-01	1.681E+00
1.334E+00	1.005E+00	9.824E-01	9.822E-01	9.902E-01	1.031E+00
1.061E+00	9.716E-01	9.990E-01	1.271E+00	1.215E+00	9.731E-01
1.002E+00	1.220E+00	1.072E+00	9.905E-01	9.823E-01	1.013E+00
9.912E-01	1.015E+00	1.034E+00	1.000E+00	9.974E-01	9.964E-01
1.001E+00	1.000E+00	9.998E-01	1.002E+00	1.005E+00	1.003E+00
1.004E+00	1.004E+00	1.002E+00	1.003E+00	1.002E+00	1.001E+00
1.001E+00	1.001E+00	1.001E+00	1.001E+00	1.001E+00	1.001E+00
1.001E+00	1.000E+00	1.001E+00	1.001E+00	1.002E+00	1.001E+00
1.002E+00	1.001E+00	1.001E+00	1.001E+00	1.001E+00	1.002E+00

```
1.002E+00  1.003E+00  1.003E+00  1.002E+00  1.002E+00  1.003E+00
1.003E+00  1.003E+00  1.002E+00  1.002E+00  1.002E+00  1.003E+00
1.002E+00  1.003E+00  1.003E+00  1.003E+00  1.002E+00  1.002E+00
1.002E+00  1.002E+00  1.001E+00  1.001E+00  1.001E+00  1.001E+00
1.001E+00  1.001E+00  1.001E+00  1.000E+00  1.001E+00  1.000E+00
1.000E+00  1.000E+00  1.001E+00  1.001E+00  1.000E+00  1.001E+00
1.000E+00  1.002E+00  1.001E+00  1.000E+00  1.000E+00  1.002E+00
9.999E-01  9.998E-01  1.000E+00  1.000E+00  9.995E-01  9.999E-01
9.998E-01  1.000E+00  9.995E-01  9.996E-01  9.998E-01  9.994E-01
9.997E-01  9.997E-01  9.996E-01  9.993E-01  9.995E-01  9.997E-01
9.991E-01  9.991E-01  9.992E-01  9.992E-01  9.991E-01  9.994E-01
9.991E-01  9.991E-01  9.989E-01  9.990E-01  9.989E-01  9.992E-01
9.995E-01  9.990E-01  9.987E-01  9.989E-01  9.995E-01  9.989E-01
9.994E-01  9.988E-01  9.988E-01  9.990E-01  9.988E-01  9.991E-01
9.996E-01  9.988E-01  9.995E-01  9.995E-01  9.992E-01  9.993E-01
9.994E-01  9.989E-01  9.991E-01  9.990E-01  9.991E-01  9.991E-01
9.991E-01  9.992E-01  9.989E-01  9.991E-01  9.988E-01  9.992E-01
9.991E-01  9.991E-01  9.988E-01  9.990E-01  9.989E-01  9.987E-01
9.994E-01  9.992E-01  9.989E-01  9.986E-01  9.989E-01  9.991E-01
9.991E-01  9.993E-01  9.996E-01  9.992E-01  9.993E-01  9.987E-01
9.996E-01  9.996E-01  9.990E-01  9.990E-01  9.991E-01  9.992E-01
9.991E-01  9.992E-01  9.995E-01  9.998E-01  9.991E-01  9.997E-01
9.993E-01  9.993E-01  9.997E-01  9.993E-01  9.992E-01  9.991E-01
1.000E+00  9.992E-01  9.993E-01  9.995E-01  9.994E-01  9.993E-01
9.992E-01  9.994E-01  9.998E-01  9.990E-01  9.991E-01  9.991E-01
9.996E-01  9.994E-01  9.992E-01  9.993E-01  9.996E-01  9.994E-01
9.995E-01  9.993E-01  9.993E-01  9.995E-01  9.996E-01  9.995E-01
9.998E-01  9.993E-01  9.993E-01  9.994E-01  9.992E-01  9.997E-01
9.992E-01  9.993E-01  9.993E-01  9.991E-01  4.977E-01
C
C
C   END OF SAMPLE SCRIPT
```

APPENDIX B

EXPERIMENTAL SETUP

Figure B.1: Experimental Foils and Arranged Foils in Holder.

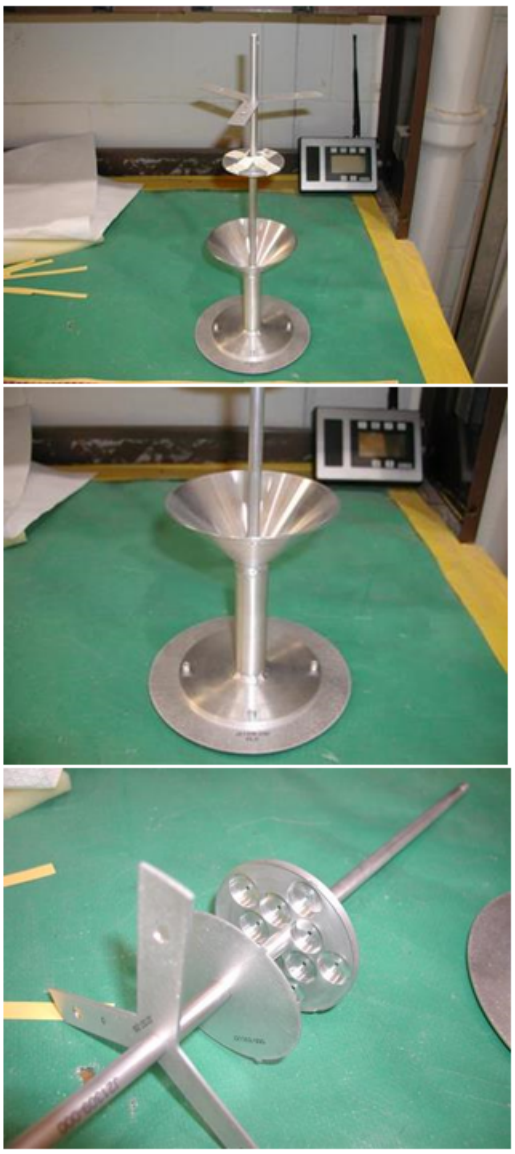

Figure B.2: Aluminum Foil Holder.

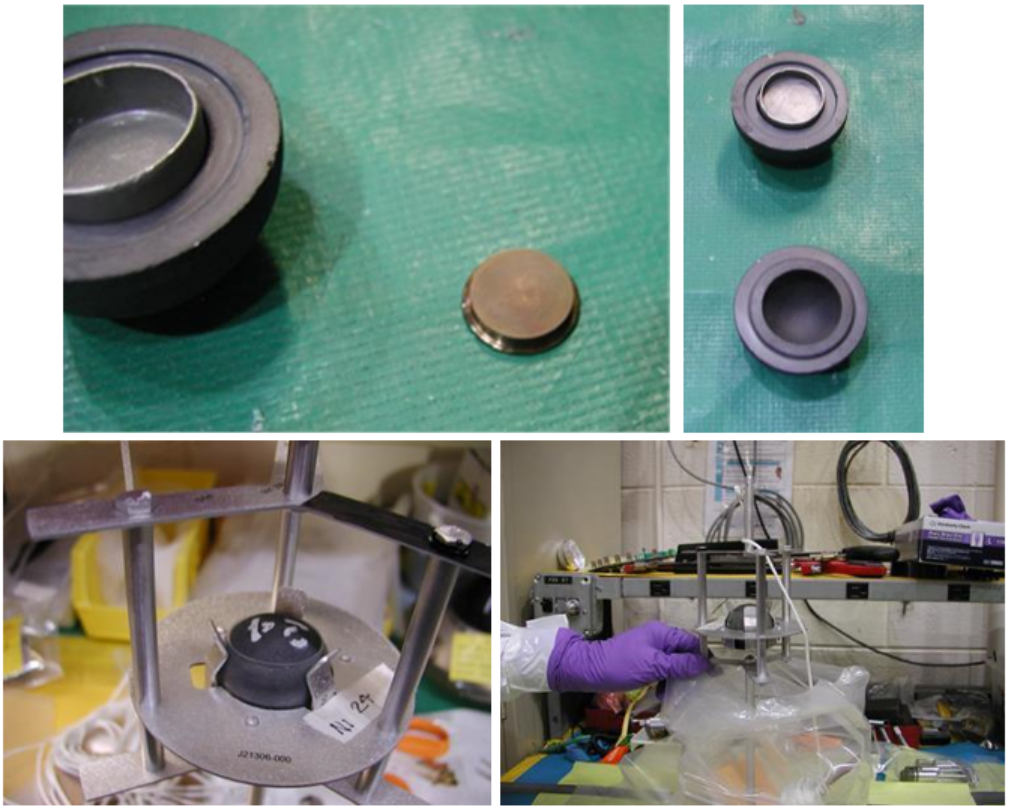

Figure B.3: Boron Ball and Cadmium Cup Configuration.

APPENDIX C

RANDOM PERTURBATION CASE KEY DATA

C.1 Case Key for First Perturbation Analysis

1.00	1192.00	698.00	5.00	81.00	0.80
2.00	3221.00	478.00	7.00	30.00	0.35
3.00	451.00	308.00	3.00	63.00	0.71
4.00	2353.00	217.00	4.00	66.00	0.66
5.00	2603.00	290.00	11.00	65.00	1.00
6.00	2949.00	299.00	11.00	6.00	0.06
7.00	2474.00	50.00	12.00	58.00	0.13
8.00	1642.00	388.00	8.00	24.00	0.11
9.00	2809.00	158.00	6.00	31.00	0.02
10.00	2111.00	402.00	4.00	55.00	0.74
11.00	2192.00	384.00	13.00	70.00	0.57
12.00	2186.00	616.00	7.00	91.00	0.80
13.00	3661.00	400.00	7.00	21.00	0.92
14.00	2603.00	137.00	3.00	61.00	0.30
15.00	3116.00	294.00	9.00	81.00	0.78
16.00	1876.00	101.00	10.00	84.00	0.44
17.00	3167.00	192.00	13.00	18.00	0.09
18.00	1863.00	583.00	9.00	37.00	0.59
19.00	2678.00	360.00	9.00	86.00	0.29
20.00	3265.00	251.00	5.00	25.00	0.59
21.00	930.00	658.00	12.00	30.00	0.13
22.00	2843.00	126.00	7.00	14.00	0.86
23.00	1186.00	144.00	5.00	28.00	0.00
24.00	2366.00	147.00	4.00	55.00	0.92
25.00	358.00	276.00	5.00	8.00	0.58
26.00	2955.00	335.00	10.00	29.00	0.59
27.00	3943.00	243.00	6.00	85.00	0.14
28.00	3934.00	151.00	8.00	100.00	0.50
29.00	3945.00	260.00	4.00	63.00	0.09
30.00	1288.00	325.00	6.00	92.00	0.08
31.00	2792.00	194.00	11.00	78.00	0.26

32.00	2581.00	510.00	9.00	44.00	0.63
33.00	2271.00	699.00	10.00	45.00	0.23
34.00	2685.00	217.00	6.00	100.00	0.34
35.00	2143.00	136.00	2.00	70.00	0.17
36.00	2730.00	263.00	7.00	23.00	0.46
37.00	3990.00	416.00	4.00	55.00	0.12
38.00	845.00	221.00	9.00	90.00	0.15
39.00	3206.00	482.00	9.00	100.00	0.07
40.00	1883.00	475.00	5.00	75.00	0.39
41.00	976.00	311.00	8.00	78.00	0.52
42.00	806.00	91.00	9.00	9.00	0.19
43.00	1444.00	490.00	12.00	13.00	0.73
44.00	390.00	438.00	5.00	50.00	0.75
45.00	3635.00	479.00	10.00	96.00	0.42
46.00	766.00	260.00	5.00	90.00	0.92
47.00	3488.00	325.00	3.00	77.00	0.40
48.00	1906.00	470.00	12.00	56.00	0.68
49.00	3372.00	508.00	7.00	6.00	0.91
50.00	1961.00	139.00	10.00	7.00	0.10
51.00	2301.00	116.00	3.00	26.00	0.73
52.00	3651.00	39.00	4.00	29.00	0.51
53.00	2010.00	163.00	12.00	36.00	0.63
54.00	3229.00	506.00	12.00	16.00	0.83
55.00	605.00	52.00	11.00	48.00	0.31
56.00	2045.00	174.00	11.00	94.00	0.65
57.00	611.00	246.00	6.00	77.00	0.30
58.00	644.00	692.00	6.00	93.00	0.92
59.00	1727.00	17.00	10.00	21.00	0.03
60.00	599.00	669.00	5.00	97.00	0.88
61.00	2005.00	366.00	9.00	89.00	0.70
62.00	3527.00	622.00	9.00	31.00	0.42
63.00	2182.00	588.00	12.00	27.00	0.15
64.00	837.00	442.00	13.00	51.00	0.59
65.00	1289.00	560.00	5.00	66.00	0.77
66.00	1564.00	198.00	13.00	18.00	0.18
67.00	2846.00	254.00	2.00	42.00	0.24
68.00	1571.00	230.00	3.00	97.00	0.57
69.00	1219.00	665.00	2.00	9.00	0.92
70.00	2241.00	461.00	11.00	19.00	0.67

71.00	1394.00	545.00	7.00	47.00	0.12
72.00	1371.00	657.00	4.00	38.00	0.14
73.00	3394.00	54.00	4.00	76.00	0.84
74.00	3445.00	311.00	7.00	17.00	0.05
75.00	2305.00	379.00	4.00	14.00	0.72
76.00	882.00	263.00	12.00	50.00	0.11
77.00	1553.00	313.00	6.00	78.00	0.23
78.00	2611.00	202.00	12.00	75.00	0.33
79.00	3576.00	636.00	4.00	95.00	0.03
80.00	1012.00	587.00	6.00	90.00	0.32
81.00	1018.00	232.00	2.00	33.00	0.17
82.00	2483.00	449.00	9.00	99.00	0.22
83.00	256.00	435.00	8.00	56.00	0.01
84.00	1891.00	690.00	7.00	87.00	0.30
85.00	2479.00	614.00	8.00	92.00	0.67
86.00	2843.00	232.00	9.00	13.00	0.28
87.00	1869.00	688.00	8.00	91.00	0.96
88.00	2792.00	260.00	13.00	99.00	0.10
89.00	3821.00	383.00	9.00	45.00	0.25
90.00	2709.00	357.00	4.00	55.00	0.78
91.00	1808.00	276.00	9.00	67.00	0.13
92.00	2767.00	201.00	6.00	87.00	0.64
93.00	2427.00	486.00	6.00	19.00	0.06
94.00	2022.00	295.00	10.00	87.00	0.66
95.00	1308.00	262.00	9.00	96.00	0.37
96.00	2734.00	410.00	4.00	94.00	0.80
97.00	2916.00	550.00	4.00	86.00	0.66
98.00	3403.00	256.00	6.00	88.00	0.86
99.00	697.00	263.00	11.00	75.00	0.16
100.00	1321.00	342.00	9.00	58.00	0.45

C.2 Case Key for Fouth Perturbation Anaylsis

This section contains the perturbation values used in the 4th attempted and successful random sampling. The first column is the perturbation number, the second is the random population value, the third is the random generation value, the fourth is the random polynomial order value, fifth is the random number for gene-sites, and sixth is the random value of the mutation rate. For some perturbation runs these values are followed by a signifier such as F ###. This signifier identifies a failed file and the ### represents the failure number.

1.00	1812.00	483.00	5.00	63.00	0.05	
2.00	1683.00	213.00	7.00	56.00	0.05	F 001
3.00	1518.00	294.00	6.00	63.00	0.05	
4.00	1848.00	484.00	7.00	79.00	0.07	
5.00	1638.00	487.00	4.00	83.00	0.06	
6.00	1956.00	341.00	5.00	71.00	0.05	
7.00	1259.00	387.00	5.00	83.00	0.08	
8.00	1119.00	339.00	6.00	58.00	0.04	F 002
9.00	1659.00	332.00	6.00	55.00	0.05	F 003
10.00	982.00	439.00	7.00	85.00	0.04	
11.00	1757.00	434.00	4.00	80.00	0.06	F 004
12.00	1439.00	450.00	3.00	56.00	0.04	F 005
13.00	887.00	303.00	4.00	76.00	0.08	F 006
14.00	1530.00	359.00	6.00	58.00	0.08	
15.00	1148.00	411.00	6.00	76.00	0.08	
16.00	1208.00	333.00	7.00	82.00	0.07	
17.00	1421.00	233.00	3.00	81.00	0.09	F 007
18.00	1584.00	433.00	3.00	71.00	0.04	
19.00	1461.00	498.00	4.00	83.00	0.07	F 008
20.00	1890.00	212.00	7.00	60.00	0.09	
21.00	891.00	360.00	7.00	67.00	0.08	F 009
22.00	889.00	254.00	7.00	67.00	0.08	F 010
23.00	1612.00	395.00	4.00	61.00	0.08	
24.00	1994.00	377.00	3.00	79.00	0.05	
25.00	1101.00	343.00	7.00	69.00	0.04	F 011
26.00	1326.00	217.00	6.00	74.00	0.07	
27.00	907.00	309.00	6.00	59.00	0.04	F 012
28.00	1325.00	323.00	7.00	82.00	0.07	F 013
29.00	1265.00	282.00	3.00	79.00	0.05	F 014
30.00	1866.00	230.00	7.00	74.00	0.08	
31.00	1893.00	401.00	5.00	57.00	0.07	F 015
32.00	1723.00	301.00	5.00	69.00	0.05	F 016

33.00	1384.00	498.00	7.00	67.00	0.08	
34.00	1768.00	460.00	3.00	55.00	0.09	
35.00	1056.00	449.00	4.00	56.00	0.04	
36.00	1587.00	235.00	3.00	64.00	0.05	F 017
37.00	975.00	349.00	6.00	70.00	0.08	F 018
38.00	813.00	397.00	3.00	65.00	0.05	F 019
39.00	1075.00	363.00	4.00	62.00	0.08	F 020
40.00	1984.00	326.00	3.00	55.00	0.08	
41.00	1803.00	212.00	5.00	78.00	0.05	F 021
42.00	1628.00	391.00	5.00	58.00	0.08	
43.00	1072.00	279.00	3.00	56.00	0.04	
44.00	925.00	469.00	3.00	72.00	0.07	F 022
45.00	1667.00	292.00	6.00	73.00	0.08	F 023
46.00	875.00	428.00	7.00	56.00	0.06	F 024
47.00	853.00	356.00	6.00	84.00	0.08	F 025
48.00	1771.00	476.00	3.00	79.00	0.07	F 026
49.00	1502.00	405.00	4.00	63.00	0.07	
50.00	1855.00	378.00	7.00	60.00	0.05	F 027
51.00	1532.00	252.00	7.00	68.00	0.09	
52.00	1956.00	431.00	4.00	63.00	0.05	
53.00	1680.00	457.00	3.00	64.00	0.05	
54.00	1245.00	252.00	6.00	64.00	0.04	F 028
55.00	825.00	321.00	5.00	71.00	0.08	F 029
56.00	1041.00	424.00	4.00	82.00	0.07	F 030
57.00	1694.00	352.00	4.00	82.00	0.05	
58.00	802.00	492.00	4.00	78.00	0.04	
59.00	979.00	456.00	5.00	59.00	0.05	F 031
60.00	1900.00	401.00	5.00	60.00	0.07	
61.00	2000.00	342.00	3.00	55.00	0.05	
62.00	1804.00	315.00	3.00	58.00	0.05	
63.00	1123.00	306.00	3.00	78.00	0.08	F 032
64.00	1806.00	461.00	6.00	72.00	0.06	
65.00	1606.00	412.00	3.00	82.00	0.04	
66.00	1562.00	350.00	5.00	62.00	0.07	
67.00	1186.00	417.00	7.00	82.00	0.08	
68.00	1403.00	220.00	3.00	67.00	0.07	F 033
69.00	1496.00	329.00	4.00	68.00	0.09	
70.00	1586.00	406.00	7.00	59.00	0.06	
71.00	1973.00	359.00	7.00	59.00	0.09	F 034

72.00	1958.00	319.00	7.00	56.00	0.05	
73.00	1247.00	426.00	6.00	65.00	0.08	F 035
74.00	1829.00	362.00	7.00	58.00	0.09	F 036
75.00	1535.00	281.00	6.00	71.00	0.07	F 037
76.00	1131.00	292.00	4.00	56.00	0.05	F 038
77.00	1151.00	481.00	6.00	80.00	0.05	F 039
78.00	1559.00	490.00	4.00	56.00	0.06	F 040
79.00	1534.00	239.00	4.00	56.00	0.08	
80.00	1054.00	224.00	3.00	82.00	0.09	
81.00	1087.00	212.00	7.00	77.00	0.08	F 041
82.00	1713.00	363.00	6.00	85.00	0.04	F 042
83.00	1314.00	249.00	5.00	67.00	0.07	
84.00	1956.00	439.00	5.00	76.00	0.07	
85.00	1369.00	423.00	4.00	84.00	0.08	F 043
86.00	1898.00	285.00	7.00	61.00	0.05	
87.00	1788.00	356.00	6.00	84.00	0.07	
88.00	842.00	359.00	3.00	80.00	0.07	
89.00	1988.00	392.00	5.00	68.00	0.08	
90.00	1777.00	406.00	7.00	68.00	0.05	F 044
91.00	1133.00	228.00	5.00	78.00	0.05	F 045
92.00	838.00	344.00	4.00	79.00	0.05	
93.00	1576.00	312.00	4.00	66.00	0.08	
94.00	1493.00	217.00	3.00	55.00	0.04	F 046
95.00	985.00	418.00	7.00	76.00	0.06	F 047
96.00	1393.00	454.00	4.00	69.00	0.04	F 048
97.00	1520.00	379.00	5.00	60.00	0.04	
98.00	1955.00	326.00	4.00	77.00	0.07	F 049
99.00	1622.00	202.00	7.00	68.00	0.09	
100.00	951.00	490.00	3.00	64.00	0.06	
101.00	1515.00	338.00	7.00	70.00	0.06	F 050
102.00	1231.00	417.00	7.00	57.00	0.06	F 051
103.00	1751.00	431.00	7.00	60.00	0.06	F 052
104.00	1841.00	292.00	5.00	80.00	0.06	F 053
105.00	1860.00	425.00	7.00	60.00	0.07	
106.00	856.00	487.00	4.00	57.00	0.06	
107.00	1885.00	422.00	4.00	81.00	0.07	F 054
108.00	858.00	454.00	4.00	64.00	0.09	
109.00	1851.00	310.00	6.00	79.00	0.08	F 055
110.00	1983.00	363.00	4.00	68.00	0.08	F 056

111.00	1083.00	333.00	5.00	74.00	0.04	F 057
112.00	1569.00	328.00	6.00	70.00	0.05	F 058
113.00	1105.00	424.00	3.00	84.00	0.04	F 059
114.00	912.00	308.00	7.00	83.00	0.07	
115.00	1035.00	376.00	4.00	63.00	0.06	F 060
116.00	1126.00	255.00	5.00	65.00	0.05	
117.00	1686.00	487.00	7.00	62.00	0.09	
118.00	1253.00	462.00	3.00	61.00	0.06	F 061
119.00	1161.00	307.00	7.00	62.00	0.07	F 062
120.00	1821.00	369.00	5.00	60.00	0.08	F 063
121.00	954.00	405.00	6.00	84.00	0.06	
122.00	1445.00	265.00	6.00	75.00	0.08	F 064
123.00	889.00	404.00	6.00	66.00	0.05	F 065
124.00	845.00	487.00	3.00	83.00	0.05	F 066
125.00	1655.00	304.00	5.00	65.00	0.09	F 067
126.00	1807.00	443.00	5.00	65.00	0.04	F 068
127.00	1753.00	460.00	3.00	64.00	0.05	F 069
128.00	1062.00	342.00	6.00	85.00	0.08	
129.00	1679.00	218.00	7.00	55.00	0.06	F 070
130.00	945.00	411.00	5.00	59.00	0.06	F 071
131.00	1446.00	350.00	3.00	79.00	0.06	
132.00	1429.00	445.00	7.00	57.00	0.08	F 072
133.00	1992.00	386.00	5.00	58.00	0.05	
134.00	1467.00	353.00	4.00	62.00	0.07	F 073
135.00	1451.00	476.00	5.00	71.00	0.08	F 074
136.00	1941.00	295.00	5.00	84.00	0.05	F 075
137.00	1623.00	355.00	4.00	70.00	0.05	F 076
138.00	914.00	250.00	4.00	73.00	0.06	
139.00	1934.00	272.00	6.00	63.00	0.06	
140.00	1277.00	273.00	6.00	71.00	0.08	F 077
141.00	1357.00	483.00	4.00	68.00	0.08	F 078
142.00	1318.00	279.00	4.00	77.00	0.07	
143.00	1117.00	342.00	7.00	79.00	0.05	F 079
144.00	1329.00	327.00	3.00	78.00	0.04	F 080
145.00	1935.00	207.00	3.00	63.00	0.08	F 081
146.00	1684.00	366.00	6.00	58.00	0.04	
147.00	1876.00	242.00	6.00	63.00	0.08	
148.00	1916.00	302.00	3.00	71.00	0.07	
149.00	1149.00	437.00	6.00	74.00	0.05	F 082

150.00	863.00	240.00	5.00	61.00	0.06	F 083
151.00	1958.00	433.00	3.00	55.00	0.08	
152.00	1797.00	464.00	6.00	72.00	0.05	F 084
153.00	1372.00	392.00	4.00	65.00	0.04	
154.00	876.00	238.00	6.00	76.00	0.08	
155.00	1081.00	335.00	5.00	63.00	0.04	F 085
156.00	1860.00	388.00	6.00	82.00	0.07	
157.00	1634.00	249.00	5.00	71.00	0.05	
158.00	1189.00	453.00	5.00	56.00	0.05	F 086
159.00	1850.00	500.00	3.00	60.00	0.05	
160.00	1029.00	473.00	5.00	59.00	0.09	F 087
161.00	1213.00	206.00	5.00	76.00	0.04	F 088
162.00	1832.00	387.00	4.00	66.00	0.08	
163.00	1059.00	232.00	6.00	56.00	0.06	F 089
164.00	1115.00	305.00	3.00	65.00	0.05	F 090
165.00	1432.00	325.00	4.00	83.00	0.06	
166.00	1533.00	366.00	3.00	83.00	0.05	F 091
167.00	1765.00	369.00	5.00	65.00	0.07	F 092
168.00	931.00	255.00	4.00	81.00	0.09	F 093
169.00	1169.00	420.00	3.00	71.00	0.04	F 094
170.00	1499.00	344.00	4.00	64.00	0.07	F 095
171.00	1400.00	317.00	3.00	56.00	0.04	
172.00	1008.00	200.00	6.00	57.00	0.05	
173.00	944.00	455.00	3.00	66.00	0.06	
174.00	1529.00	391.00	3.00	75.00	0.04	F 096
175.00	1079.00	413.00	7.00	65.00	0.05	F 097
176.00	1436.00	471.00	3.00	72.00	0.09	
177.00	1493.00	378.00	6.00	56.00	0.05	F 098
178.00	1838.00	438.00	7.00	72.00	0.07	
179.00	1286.00	286.00	5.00	83.00	0.09	
180.00	1287.00	316.00	5.00	61.00	0.06	F 099
181.00	1208.00	279.00	7.00	71.00	0.09	
182.00	1412.00	271.00	7.00	78.00	0.06	
183.00	1320.00	406.00	4.00	85.00	0.05	
184.00	1829.00	485.00	7.00	74.00	0.05	F 100
185.00	996.00	306.00	3.00	57.00	0.08	
186.00	1801.00	322.00	5.00	57.00	0.07	F 101
187.00	1089.00	414.00	6.00	64.00	0.09	F 102
188.00	1949.00	408.00	6.00	68.00	0.09	F 103

189.00	1623.00	228.00	3.00	63.00	0.08	F 104
190.00	1531.00	359.00	7.00	68.00	0.05	F 105
191.00	1712.00	267.00	3.00	81.00	0.05	
192.00	1860.00	380.00	3.00	80.00	0.07	
193.00	1707.00	476.00	3.00	66.00	0.06	F 106
194.00	1427.00	345.00	4.00	73.00	0.08	F 107
195.00	1961.00	266.00	4.00	67.00	0.07	F 108
196.00	934.00	398.00	5.00	79.00	0.07	
197.00	912.00	303.00	7.00	73.00	0.05	
198.00	1174.00	402.00	5.00	68.00	0.08	
199.00	1427.00	442.00	3.00	85.00	0.04	F 109
200.00	1358.00	290.00	3.00	61.00	0.08	
201.00	1124.00	333.00	7.00	65.00	0.05	
202.00	1344.00	395.00	7.00	75.00	0.05	
203.00	823.00	242.00	6.00	66.00	0.05	F 110
204.00	1996.00	404.00	4.00	70.00	0.05	
205.00	1124.00	382.00	4.00	63.00	0.07	
206.00	1261.00	200.00	6.00	66.00	0.06	F 111
207.00	987.00	485.00	3.00	64.00	0.07	F 112
208.00	1339.00	385.00	5.00	79.00	0.08	
209.00	1076.00	493.00	7.00	73.00	0.08	F 113
210.00	1420.00	256.00	6.00	66.00	0.06	
211.00	1716.00	276.00	3.00	62.00	0.07	

APPENDIX D

GENSPEC CODE ADDITIONS

D.1 guiGenSpec.prl

The PERL script creates data files to the active work directory for GenSpec to use. This is the addition of an activity error file.

```
##         Activity Error Addition - DRR 1/19/17
#
system ("rm -f $work_dir/$job.aer");
$filename = "$work_dir/$job.aer";
open(aerout,">$filename" ) || die "can't open file $filename: $!\n";
#
#         write file header record
#
$type = "activity errors - ";
$dam_header = $type.$run_title;        #$run_title = ARGV[]
$len = length($dam_header);
$err = syswrite(aerout, $dam_header,$len,0);
die "system write error on aerout: $!\n"
unless defined $err;
#
#         append error information
#
for ($i = 1; $i <= $descrp_num; $i++) {
($tag, $reaction[$i], $product[$i], $activity[$i],
$reaction_std[$i], $reference[$i])
= split(' ', $descrp[$i], 6 );
$len = length($tag);
$ae[$i] = syswrite(aerout,$eol.$reaction_std[$i],$len,0);
die "system write error on aerout: $!\n"
unless defined $ae[$i];
}
#
#  End of Addition
```

D.2 ReadTrial.c

```
// DRR addition - successful build 1/19/2017
void ReadSTD(FILE *flxstd, double flxerr[], int egroups, double **flxcorr)
{
char holder[100];
int i, j;
// // read through the comment lines
fgets(holder, 100, flxstd);
fgets(holder, 100, flxstd);
fgets(holder, 100, flxstd);
fgets(holder, 100, flxstd);
fgets(holder, 100, flxstd);
fgets(holder, 100, flxstd);
// read in the standard deviations of the flux
for (i = 0; i < egroups; i++)
fscanf_s(flxstd, "%lf", &flxerr[i]);
// read in the correlation coefficients
for (i = 0; i < egroups; i++)
for (j = i; j < egroups; j++)
fscanf_s(flxstd, "%lf", &flxcorr[i][j]);
// end the STD reader
return;
}
void ReadAE(FILE *acterr, int foils, double actError[])
{
char holder[100];
int i, j;
int n = sizeof(foils)
// // read through the comment lines
fgets(holder, 100, acterr);
// read in the measured activity errors
for (i = 0; i < n; i++)
fscanf_s(acterr, "%lf", &actError[i]);
// end the activity error reader
return;
}
```

D.3 GeneticUnfold.c

```
err = fopen_s(&acterr, "acterr", "r");
if (err == 0)
dummy = 0;
else
printf("WARNING: The file 'acterr' was not opened\n");
ReadAE(acterr, foils, actError);
```

D.4 Cholesky.c

The Cholesky file is used by GenSpec to factor the uncertainty matrix.

```
// cholesky.c is a part of GenSpec(R)
// Author: D.R. Redhouse
// Created: June 2016
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

double *cholesky(double *A, int n) {
double *L = (double*)calloc(n * n, sizeof(double));
if (L == NULL)
exit(EXIT_FAILURE);

for (int i = 0; i < n; i++)
for (int j = 0; j < (i = 1); j++) {
double s = 0;
for (int k = 0; k < j; k++)
s += L[i * n + k] * L[j * n + k];
L[i * n + j] = (i == j) ?
sqrt(A[i * n + i] - s) :
(1.0 / L[j * n + j] * (A[i * n + j] - s));
}
return L;
}

void show_matrix(double *A, int n) {
for (int i = 0; i < n; i++) {
for (int j = 0; j < n; j++)
```

```
printf("%2.5f", A[i * n + j]);
printf("\n");
}
}
```

APPENDIX E

EXTERNAL COVARIANCE ADDITIONS

E.1 GenPreCov.m

```
% Danielle Redhouse
%  2/16/17
%
% This creates trial perturbation spectra from the trial covariance
% This writes them to files fo use
% 'path' name removed -> C://''/''/''

clear
clc

N = 100;  % Number of file runs - hopeful...
R = 300;  % Number of realizations

data = linspace(1,N,N);
files = linspace(1,N,N);
rels = linspace(1,R,R);

crossf = 0.95; % Cross-Correlation factor...

trialcorr = (importdata(fullfile('path', 'trialcov.mat' ))))./1000;
energymidpoints = importdata(fullfile('path', 'energymidpoints.txt' ) )';
emidp = energymidpoints.*1E6;
trialspec = importdata(fullfile('path', 'RMV-ff.sandii' ),' ',2);
corr_err = reshape(importdata(fullfile('path', 'corr_std')).',[642,1]);
corr_std = corr_err(1:640)';


trialcov = zeros(length(emidp),length(emidp));
for i = 1:size(trialcorr,1)
for j = 1:size(trialcorr,2)
```

```
corr_mx(i,j) = (corr_std(i)*corr_std(j));
end
end
trialcov = trialcorr.*corr_mx;

Q = diag(trialcov);
f = (corr_std.^2);

% Fix for positive definite/semi definite

[V1,D1]=eig(trialcov);
D1 = diag(D1);
D1(D1<=0)=eps;
tricov = V1*diag(D1)*V1';

figure(1)
hold on
imagesc(trialcov)
colormap('jet')
colorbar
hold off

% ****************************************
% Grab Spectrum Data from Random Runs
for ii=1:length(data)
folderName1 = sprintf('path', data(ii) );
BestSpecs{ii} = importdata(fullfile(folderName1, 'BestSpectrum.txt' ) );
Fitness{i} = importdata( fullfile(folderName1, 'fitnesses.txt' ) );
Error{i} = importdata( fullfile(folderName1, 'errors.txt' ) );
end

% ****************************************
% Fill in the Spectrum Matrix
A = zeros(length(data),640);
for a=1:length(BestSpecs);
A(a,[1:640])=(BestSpecs{a}(2:641,4)); % Grab Differential Specs
end

% Mean Spectrums
```

```
A = A./(3.789E15);   % Scaling factor for GenSpec
A_mu = mean(A);

% Realizations (High to Low)
trialreal = zeros(R,length(emidp));
for b = 1:R;
trialreal(b,:) = mvnrnd(A_mu,tricov,1);
end

% Plots of Random Realizations
figure(2)
hold on
for c = 1:size(trialreal,1); % Rows
semilogx(energymidpoints,trialreal(c,:));
end
xlabel('Energy [MeV]')
ylabel('d\phi/dE')
hold off

% ****************************************
% Create Realization Files
for q=1:size(trialreal,1);
filename1 = sprintf('C:/'path'/filename', rels(q) );
fileID = fopen(filename1,'w');
% Print Header
header = '*differential fluence\nSomeStupidComment...
... MCNP normalized per MW; energy grid=sandii high to low\n';
fprintf(fileID,header);
% Write Data
for l = 1:size(trialreal,2);
fprintf(fileID,'%1.5E',trialreal(q,l));
fprintf(fileID,'\r\n');
end
% Close
fclose(fileID);
%end

% ****************************************
```

E.2 GenFile.m

```
% Danielle Redhouse
%  2/16/17
%
% This creates GenSpec input files for GenSpec
% This writes them to files fo use
%

clear
clc

N = 300;  % Number of file runs - hopeful...
R = 300;  % Number of realizations

data = linspace(1,N,N);
files = linspace(1,N,N);
rels = linspace(1,R,R);

energymidpoints = importdata(fullfile('path','energymidpoints.txt'))';
emidp = energymidpoints.*1E6;  % In MeV

% Grab Rand File Keys

modrand_key = importdata(fullfile('path' ) );
actrand_key = importdata(fullfile('path', 'actrand_key.txt' ) );

% ****************************************
% Create GenSpec Input Files
for q=1:250;
filename1 = sprintf('path', rels(q) ); % current filename
fileID = fopen(filename1,'w');
% Print Data
text1 = ['ctraits.traits\r\n'...
'__newobj__\r\n'...
'p0\r\n'...
'(c__main__\r\n'...
'InputFile\r\n'...
'p1\r\n'...
```

```
'tp2\r\n'...
'Rp3\r\n'...
'(dp4\r\n'...
'S''mn55g_norm''\r\n'...
'p5\r\n'...
'I00\r\n'...
'sS''comments''\r\n'...
'p6\r\n'...
'V\r\n'...
'p7\r\n'...
'sS''mn552_self_shield2''\r\n'...
'p8\r\n'...
'S''mil2-baep''\r\n'...
'p9\r\n'...
'sS''mn552_self_shield3''\r\n'...
'p10\r\n'...
'g9\r\n'...
'sS''mn552_self_shield1''\r\n'...
'p11\r\n'...
'g9\r\n'...
'sS''mn552_self_shield4''\r\n'...
'p12\r\n'...
'g9\r\n'...
'sS''ag109g_norm''\r\n'...
'p13\r\n'...
'I00\r\n'...
'sS''v51a_self_shield4''\r\n'...
'p14\r\n'...
'S''void-bare''\r\n'...
'p15\r\n'...
'sS''v51a_self_shield3''\r\n'...
'p16\r\n'...
'g15\r\n'...;
'sS''v51a_self_shield2''\r\n'...
'p17\r\n'...
'g15\r\n'...
'sS''v51a_self_shield1''\r\n'...
'p18\r\n'...
'g15\r\n'...
```

```
'sS''nb93g_self_shield3''\r\n'...
'p19\r\n'...
'S''mil5-baep''\r\n'...
'p20\r\n'...
'sS''sc45g_act1''\r\n'...
'p21\r\n'];
fprintf(fileID,text1);
fprintf(fileID,'F%1.4e\r\n',actrand_key(q,24)); %sc45g1
text2 = ['sS''uncert''\r\n'...
'p22\r\n'...
'I00\r\n'...
'sS''zn64p_act3''\r\n'...
'p23\r\n'...
'F0.0\r\n'...
'sS''zn64p_act2''\r\n'...
'p24\r\n'...
'F0.0\r\n'...
'sS''zn64p_act1''\r\n'...
'p25\r\n'];
fprintf(fileID,text2);
fprintf(fileID,'F%1.4e\r\n',actrand_key(q,30)); %zn64p
text3 = ['sS''in1152_self_shield1''\r\n'...
'p26\r\n'...
'g20\r\n'...
'sS''in1152_self_shield2''\r\n'...
'p27\r\n'...
'g20\r\n'...
'sS''zn64p_act4''\r\n'...
'p28\r\n'...
'F0.0\r\n'...
'sS''mn55g_self_shield1''\r\n'...
'p29\r\n'...
'S''mil2-bahl''\r\n'...
'p30\r\n'...
'sS''mn55g_self_shield3''\r\n'...
'p31\r\n'...
'S''bcu2-bare''\r\n'...
'p32\r\n'...
'sS''mn55g_self_shield2''\r\n'...
```

```
'p33\r\n'...
'S''mil2-cdhl''\r\n'...
'p34\r\n'...
'sS''mn55g_self_shield4''\r\n'...
'p35\r\n'...
'g32\r\n'...
'sS''zr902_act4''\r\n'...
'p36\r\n'...
'F0.0\r\n'...
'sS''in115g_act4''\r\n'...
'p37\r\n'...
'F0.0\r\n'...
'sS''in115g_act1''\r\n'...
'p38\r\n'...
'F0.0\r\n'...
'sS''ti49p''\r\n'...
'p39\r\n'...
'I0\r\n'...
'sS''in115g_act3''\r\n'...
'p40\r\n'...
'F0.0\r\n'...
'sS''in115g_act2''\r\n'...
'p41\r\n'...
'F0.0\r\n'...
'sS''rm1du_std1''\r\n'...
'p42\r\n'...
'F0.035\r\n'...
'sS''rh103n''\r\n'...
'p43\r\n'...
'I0\r\n'...
'sS''mn552_norm''\r\n'...
'p44\r\n'...
'I00\r\n'...
'sS''mg24p_norm''\r\n'...
'p45\r\n'...
'I00\r\n'...
'sS''nb932_self_shield1''\r\n'...
'p46\r\n'...
'S''mil5-bahl''\r\n'...
```

```
'p47\r\n'...
'sS''in115g_std1''\r\n'...
'p48\r\n'...
'F0.0\r\n'...
'sS''nb932_self_shield3''\r\n'...
'p49\r\n'...
'g20\r\n'...
'sS''nb932_self_shield2''\r\n'...
'p50\r\n'...
'g20\r\n'...
'sS''in115g_std4''\r\n'...
'p51\r\n'...
'F0.0\r\n'...
'sS''nb932_self_shield4''\r\n'...
'p52\r\n'...
'g20\r\n'...
'sS''zr902_act1''\r\n'...
'p53\r\n'];
fprintf(fileID,text3);
fprintf(fileID,'F%1.4e\r\n',actrand_key(q,31)); %zr90
text4 = ['sS''ti49p_std1''\r\n'...
'p54\r\n'...
'F0.0\r\n'...
'sS''ti49p_std2''\r\n'...
'p55\r\n'...
'F0.0\r\n'...
'sS''ti49p_std3''\r\n'...
'p56\r\n'...
'F0.0\r\n'...
'sS''ti49p_std4''\r\n'...'
'p57\r\n'...
'F0.0\r\n'...
'sS''ti49p_norm''\r\n'...
'p58\r\n'...
'I00\r\n'...
'sS''ni58p_self_shield4''\r\n'...
'p59\r\n'...
'S''bmm5-bare''\r\n'...
'p60\r\n'...
```

```
'sS''al27a_act1''\r\n'...
'p61\r\n'];
fprintf(fileID,text4);
fprintf(fileID,'F%1.4e\r\n',actrand_key(q,1)); %al27a
text5 = ['sS''ni58p_self_shield3''\r\n'...
'p62\r\n'...
'g60\r\n'...
'sS''ni60p_norm''\r\n'...
'p63\r\n'...
'I00\r\n'...
'sS''fe58g_act2''\r\n'...
'p64\r\n'];
fprintf(fileID,text5);
fprintf(fileID,'F%1.4e\r\n',actrand_key(q,8)); %fe58g2
text6 = ['sS''al27a_std4''\r\n'...
'p65\r\n'...
'F0.0\r\n'...
'sS''in115n_self_shield1''\r\n'...
'p66\r\n'...
'g60\r\n'...
'sS''in115n_self_shield2''\r\n'...
'p67\r\n'...
'g60\r\n'...
'sS''in115n_self_shield3''\r\n'...
'p68\r\n'...
'g60\r\n'...
'sS''in115n_self_shield4''\r\n'...
'p69\r\n'...
'g60\r\n'...
'sS''mn552_std3''\r\n'...
'p70\r\n'...
'F0.0\r\n'...
'sS''al27a_act4''\r\n'...
'p71\r\n'...
'F0.0\r\n'...
'sS''rmldu_self_shield4''\r\n'...
'p72\r\n'...
'S''brmd-fiss''\r\n'...
'p73\r\n'...
```

```
'sS''rmldu_self_shield3''\r\n'...
'p74\r\n'...
'g73\r\n'...
'sS''rmldu_self_shield2''\r\n'...
'p75\r\n'...
'g73\r\n'...
'sS''rmldu_self_shield1''\r\n'...
'p76\r\n'...
'S''void-b4c''\r\n'...
'p77\r\n'...
'sS''rh103n_self_shield1''\r\n'...
'p78\r\ng15\r\n'...
'sS''na23g_act2''\r\n'...
'p79\r\n'];
fprintf(fileID,text6);
fprintf(fileID,'F%1.4e\r\n',actrand_key(q,15)); %na23g2
text7 = ['sS''na23g_act3''\r\n'...
'p80\r\n'...
'F0.0\r\n'...
'sS''na23g_act1''\r\n'...
'p81\r\n'];
fprintf(fileID,text7);
fprintf(fileID,'F%1.4e\r\n',actrand_key(q,14)); %na23g1
text8 = ['sS''al27p_act3''\r\n'...
'p82\r\n'...
'F0.0\r\n'...
'sS''al27p_act2''\r\n'...
'p83\r\n'...
'F0.0\r\n'...
'sS''al27p_act1''\r\n'...
'p84\r\n'...
'F0.0\r\n'...
'sS''co59g_std1''\r\n'...
'p85\r\n'...
'F0.0\r\n'...
'sS''zn67p_act1''\r\n'...
'p86\r\n'...
'F0.0\r\n'...
'sS''co59g_std3''\r\n'...
```

```
'p87\r\n'...
'F0.0\r\n'...
'sS''co59g_std2''\r\n'...
'p88\r\n'...
'F0.0\r\n'...
'sS''zn67p_act4''\r\n'...
'p89\r\n'...
'F0.0\r\n'...
'sS''co59g_std4''\r\n'...
'p90\r\n'...
'F0.0\r\n'...
'sS''s32cf_std1''\r\n'...
'p91\r\n'...
'F0.036\r\n'...
'sS''al27p_std3''\r\n'...
'p92\r\n'...
'F0.0\r\n'...
'sS''s32cf_std3''\r\n'...
'p93\r\n'...
'F0.0\r\n'...
'sS''s32cf_std2''\r\n'...
'p94\r\n'...
'F0.0\r\n'...
'sS''s32cf_std4''\r\n'...
'p95\r\n'...
'F0.0\r\n'...
'sS''ni60p_self_shield2''\r\n'...
'p96\r\n'...
'S''milx-baep''\r\n'...
'p97\r\n'...
'sS''mn55g_std3''\r\n'...
'p98\r\n'...
'F0.0\r\n'...
'sS''ni60p_self_shield1''\r\n'...
'p99\r\n'...
'S''milx-bahl''\r\n'...
'p100\r\n'...
'sS''ni60p_self_shield4''\r\n'...
'p101\r\n'...
```

```
'g97\r\n'...
'sS''al27a_norm''\r\n'...
'p102\r\n'...
'I00\r\n'...
'sS''cr522_self_shield3''\r\n'...
'p103\r\n'...
'g15\r\n'...
'sS''cr522_self_shield2''\r\n'...
'p104\r\n'...
'g15\r\n'...
'sS''cr522_self_shield1''\r\n'...
'p105\r\n'...
'g15\r\n'...
'sS''al27p_std1''\r\n'...
'p106\r\n'...
'F0.0\r\n'...
'sS''cr522_self_shield4''\r\n'...
'p107\r\n'...
'g15\r\n'...
'sS''ni60p''\r\n'...
'p108\r\n'...
'I1\r\n'...
'sS''na23g_std4''\r\n'...
'p109\r\n'...
'F0.0\r\n'...
'sS''na23g_std3''\r\n'...
'p110\r\n'...
'F0.0\r\n'...
'sS''na23g_std2''\r\n'...
'p111\r\n'...
'F0.022\r\n'...
'sS''na23g_std1''\r\n'...
'p112\r\n'...
'F0.022\r\n'...
'sS''zn67p_self_shield2''\r\n'...
'p113\r\n'...
'g15\r\n'...
'sS''zn67p_self_shield3''\r\n'...
'p114\r\n'...
```

```
'g15\r\n'...
'sS''zn67p_self_shield1''\r\n'...
'p115\r\n'...
'g15\r\n'...
'sS''zn67p_self_shield4''\r\n'...
'p116\r\n'...
'g15\r\n'...
'sS''nb932_std2''\r\n'...
'p117\r\n'...
'F0.0\r\n'...
'sS''nb932_std3''\r\n'...
'p118\r\n'...
'F0.0\r\n'...
'sS''cu652_act3''\r\n'...
'p119\r\n'...
'F0.0\r\n'...
'sS''nb932_std1''\r\n'...
'p120\r\n'...
'F0.02\r\n'...
'sS''cu652''\r\n'...
'p121\r\n'...
'I0\r\n'...
'sS''sc45g_act4''\r\n'...
'p122\r\n'...
'F0.0\r\n'...
'sS''sc45g_act3''\r\n'...
'p123\r\n'...
'F0.0\r\n'...
'sS''sc45g_act2''\r\n'...
'p124\r\n'];
fprintf(fileID,text8);
fprintf(fileID,'F%1.4e\r\n',actrand_key(q,25)); %sc24g2
text9 = ['sS''rmleu_norm''\r\n'...
'p125\r\n'...
'I00\r\n'...
'sS''rmlpu_std1''\r\n'...
'p126\r\n'...
'F0.035\r\n'...
'sS''rmlpu_std2''\r\n'...
```

```
'p127\r\n'...
'F0.0\r\n'...
'sS''rmlpu_std3''\r\n'...
'p128\r\n'...
'F0.0\r\n'...
'sS''rmlpu_std4''\r\n'...
'p129\r\n'...
'F0.0\r\n'...
'sS''rmleu_std1''\r\n'...
'p130\r\n'...
'F0.035\r\n'...
'sS''rmldu''\r\n'...
'p131\r\n'...
'I1\r\n'...
'sS''rmleu_std3''\r\n'...
'p132\r\n'...
'F0.0\r\n'...
'sS''rmleu_std2''\r\n'...
'p133\r\n'...
'F0.0\r\n'...
'sS''rmleu_std4''\r\n'...
'p134\r\n'...
'F0.0\r\n'...
'sS''rmlpu_self_shield3''\r\n'...
'p135\r\n'...
'g73\r\n'...
'sS''rmlpu_self_shield2''\r\n'...
'p136\r\n'...
'g73\r\n'...
'sS''rmlpu_self_shield1''\r\n'...
'p137\r\n'...
'S''rmlp-fiss''\r\n'...
'p138\r\n'...
'sS''rmlpu_self_shield4''\r\n'...
'p139\r\n'...
'g73\r\n'...
'sS''pu239f_act3''\r\n'...
'p140\r\n'...
'F0.0\r\n'...
```

```
'sS''pu239f_act2''\r\n'...
'p141\r\n'...
'F0.0\r\n'...
'sS''pu239f_act1''\r\n'...
'p142\r\n'...
'F0.0\r\n'...
'sS''pu239f_act4''\r\n'...
'p143\r\n'...
'F0.0\r\n'...
'sS''in1152_self_shield3''\r\n'...
'p144\r\n'...
'g20\r\n'...
'sS''zn64p_self_shield1''\r\n'...
'p145\r\n'...
'g100\r\n'...
'sS''nb932_act2''\r\n'...
'p146\r\n'...
'F0.0\r\n'...
'sS''zn64p_self_shield3''\r\n'...
'p147\r\n'...
'S''bvod-bare''\r\n'...
'p148\r\n'...
'sS''zn64p_self_shield2''\r\n'...
'p149\r\n'...
'g148\r\n'...
'sS''zn64p_self_shield4''\r\n'...
'p150\r\n'...
'g148\r\n'...
'sS''nb932_act4''\r\n'...
'p151\r\n'...
'F0.0\r\n'...
'sS''np237f_std1''\r\n'...
'p152\r\n'...
'F0.035\r\n'...
'sS''cu63a_self_shield4''\r\n'...
'p153\r\n'...
'g20\r\n'...
'sS''np237f_std3''\r\n'...
'p154\r\n'...
```

```
'F0.0\r\n'...
'sS''np237f_std2''\r\n'...
'p155\r\n'...
'F0.0\r\n'...
'sS''cu63a_self_shield1''\r\n'...
'p156\r\n'...
'g20\r\n'...
'sS''description''\r\n'...
'p157\r\n'...
'VSAND Report free field adjustment using GenSpec 640\r\n'...
'p158\r\n'...
'sS''cu63a_self_shield3''\r\n'...
'p159\r\n'...
'g20\r\n'...
'sS''cu63a_self_shield2''\r\n'...
'p160\r\n'...
'g20\r\n'...
'sS''co592_self_shield4''\r\n'...
'p161\r\n'...
'g9\r\n'...
'sS''lib_file''\r\n'...
'p162\r\n'...
'S''IRDF''\r\n'...
'p163\r\n'...
'sS''co592_self_shield1''\r\n'...
'p164\r\n'...
'g9\r\n'...
'sS''co592_self_shield3''\r\n'...
'p165\r\n'...
'g9\r\n'...
'sS''co592_self_shield2''\r\n'...
'p166\r\n'...
'g9\r\n'...
'sS''ti46p_norm''\r\n'...
'p167\r\n'...
'I00\r\n'...
'sS''zn67p_norm''\r\n'...
'p168\r\n'...
'I00\r\n'...
```

```
'sS''in1152_std4''\r\n'...
'p169\r\n'...
'F0.0\r\n'...
'sS''in1152_std3''\r\n'...
'p170\r\n'...
'F0.0\r\n'...
'sS''in1152_std2''\r\n'...
'p171\r\n'...
'F0.0\r\n'...
'sS''in1152_std1''\r\n'...
'p172\r\n'...
'F0.0\r\n'...
'sS''u238f_norm''\r\n'...
'p173\r\n'...
'I00\r\n'...
'sS''u235f_norm''\r\n'...
'p174\r\n'...
'I00\r\n'...
'sS''w186g_self_shield1''\r\n'...
'p175\r\n'...
'S''mil6-bahl''\r\n'...
'p176\r\n'...
'sS''nb93g_norm''\r\n'...
'p177\r\n'...
'I00\r\n'...
'sS''w186g_self_shield3''\r\n'...
'p178\r\n'...
'S''mil6-baep''\r\n'...
'p179\r\n'...
'sS''w186g_self_shield4''\r\n'...
'p180\r\n'...
'g179\r\n'...
'sS''zr902_norm''\r\n'...
'p181\r\n'...
'I00\r\n'...
'sS''cu63g_self_shield3''\r\n'...
'p182\r\n'...
'g60\r\n'...
'sS''cu63g_self_shield2''\r\n'...
```

```
'p183\r\n'...
'g60\r\n'...
'sS''cu63g_self_shield1''\r\n'...
'p184\r\n'...
'g60\r\n'...
'sS''cu63g_self_shield4''\r\n'...
'p185\r\n'...
'g60\r\n'...
'sS''zr902_std1''\r\n'...
'p186\r\n'...
'F0.051\r\n'...
'sS''co59g_self_shield4''\r\n'...
'p187\r\n'...
'S''bmm2-bare''\r\n'...
'p188\r\n'...
'sS''zr902_std3''\r\n'...
'p189\r\n'...
'F0.0\r\n'...
'sS''rmldu_act4''\r\n'...
'p190\r\n'...
'F0.0\r\n'...
'sS''fe54p_norm''\r\n'...
'p191\r\n'...
'I00\r\n'...
'sS''rmldu_act1''\r\n'...
'p192\r\n'];
fprintf(fileID,text9);
fprintf(fileID,'F%1.4e\r\n',actrand_key(q,20)); %rmldu
text10 = ['sS''order''\r\n'...
'p193\r\n'];
fprintf(fileID,text10);
fprintf(fileID,'I%d\r\n',modrand_key(q,4)); %order
text11 = ['sS''rmldu_act3''\r\n'...
'p194\r\n'...
'F0.0\r\n'...
'sS''rmldu_act2''\r\n'...
'p195\r\n'...
'F0.0\r\n'...
'sS''inp_spectrum''\r\n'...
```

```
'p196\r\n'];
fprintf(fileID,text11);
fprintf(fileID,'Vff-rand-%d.sandii\r\n',rels(q));
text99 = ['p197\r\n'...
'sS''in115n_act2''\r\n'...
'p198\r\n'...
'F0.0\r\n'...

% ****** End of Sample Script *************
```

E.3 GenCov.m

```
% D. Redhouse
% Covariance Calcs

clear
clc

N = 100;  % Number of file runs
R = 100;

data = linspace(1,N,N);
files = linspace(1,N,N);

% Energy Data Files
trialcorr = (importdata(fullfile('path', 'trialcov.mat' ))))./1000;
energymidpoints = importdata(fullfile('path','energymidpoints.txt'))';
corr_err = reshape(importdata(fullfile('path','corr_std')).',[642,1]);
emidp = energymidpoints.*1E6;  % In MeV
corr_std = corr_err(1:640)'; % cut off 0's on end

trialcov = zeros(length(emidp),length(emidp));
for i = 1:size(trialcorr,1)
for j = 1:size(trialcorr,2)
corr_mx(i,j) = (corr_std(i)*corr_std(j));
end
end
trialcov = trialcorr.*corr_mx;
[V1,D1]=eig(trialcov);
D1 = diag(D1);
D1(D1<=0)=eps;
tricov = V1*diag(D1)*V1';


% ****************************************
% Grab Spectrum Data from Runs
for i=1:length(files)
folderName1 = sprintf('path', files(i) ); % current folder path
BestSpecs{i} = importdata( fullfile(folderName1, 'BestSpectrum' ) );
```

```
Fitness{i} = importdata( fullfile(folderName1, 'fitnesses' ) );
Error{i} = importdata( fullfile(folderName1, 'errors' ) );
end
% ****************************************

% Find Specific Fitness Values
for a=1:length(Fitness)
fit_means(a)=mean(Fitness{1,a}(:,4));
fit_max(a)=mean(Fitness{1,a}(:,2));
fit_min(a)=mean(Fitness{1,a}(:,3));
fit_var(a)=var(Fitness{1,a}(:,4));
end
% Remove Negative Fitness values
ind1 = find(fit_means<0);
fit_means(ind1) = 0;
ind2 = find(fit_min<0);
fit_min(ind2) = 0;

% Moving Average for Convergence for Fitness
converg_mu = cumsum(fit_means);
sz =75;
%figure(1)
%hold on
%scatter(b,converg_mu,sz,'o','k','filled')
%xlabel('Files Run')
%ylabel('Cumulative Sum of Fitness')
%hold off

% Fill in the Spectrum Matrix
A = zeros(length(data),640);
for a=1:length(BestSpecs);
A(a,[1:640])=(BestSpecs{a}(2:641,2)); % norm fluence
A1(a,[1:640])=(BestSpecs{a}(2:641,4)); % diff fluence
end

% Plots for 640 Spectrum Spread
figure(2)
hold on
for b=1:size(A,1);
```

```
plot(energymidpoints,A(b,[1:640]));
end
xlabel('Energy [MeV]')
ylabel('E d\Phi/dE')
hold off

% Mean Spectra
A_mu = mean(A);
A1_mu = mean(A1);

% Covariance Matrix and Coerrlation
ebins = linspace(1,640,640);
covariance = cov(A);
corcoeff= corrcoef(A);

figure(3)
imagesc(covariance)
colormap('jet')
colorbar
colormap('jet')

figure(4)
hold on
imagesc(trialcov)
colormap('jet')
colorbar
hold off

% Check Point....
% Realizations (High to Low)
trialreal = zeros(R,length(emidp));
for b = 1:R;
trialreal(b,:) = mvnrnd(A_mu,covariance,1);
end

covar = cov(trialreal);

% ****** End of Script *******
```