UNCERTAINTY QUANTIFICATION OF MULTI-COMPONENT ISOTOPE SEPARATION CASCADE MODEL

THESIS

Khoi D. Tran, Capt, USAF

AFIT/GNE/ENP/10-M09

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT/GNE/ENP/10-M09

UNCERTAINTY QUANTIFICATION OF MULTI-COMPONENT ISOTOPE
SEPARATION CASCADE MODEL

THESIS

Presented to the Faculty

Department of Engineering Physics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Nuclear Engineering

Khoi D. Tran, BS

Captain, USAF

March 2010

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT/GNE/ENP/10-M09

UNCERTAINTY QUANTIFICATION OF MULTI-COMPONENT ISOTOPE SEPARATION CASCADE MODEL

Khoi D. Tran, BS

Captain, USAF

Approved:

Kirk A. Mathews (signature)
_____
Kirk A. Mathews, Ph.D. (Chairman)
Professor of Nuclear Engineering

Benjamin R. Kowash (signature)
_____
Benjamin R. Kowash, Capt, USAF (Member)
Assistant Professor of Nuclear Engineering

William P. Baker (signature)
_____
William P. Baker, Ph.D. (Member)
Associate Professor of Mathematics

12 March 2010
_____
Date

15 March 2010
_____
Date

15 Mar 2010
_____
Date

AFIT/GNE/ENP/10-M09

Abstract

Monte Carlo uncertainty quantification (UQ) capability has been added to a code for modeling multi-component steady-state isotope-separation enrichment cascades to characterize the propagation of uncertainties in input data that define the cascade and the feed. Random samples of error for every computational input are drawn from its individual uncertainty distribution and added to the inputs, creating a set of enrichment cascade problems with perturbed inputs. The set of problems is solved using the verified code. The cascade outputs are then characterized using the empirical cumulative distribution. The uncertainty output data are analyzed to gain new insights into the behaviors of enrichment cascades. The UQ capability is an investigative tool that can be used to explore current and new questions of interest regarding enrichment.

Acknowledgments

I would like to express my sincere appreciation to my faculty advisor, Dr. Kirk A. Mathews, for his guidance and support throughout the course of this thesis effort. The insight and experience were certainly appreciated.

Khoi D. Tran

Table of Contents

List of Figures

List of Tables

List of Symbols

1. i, j subscripts denoting stage number

2. a_i depleted flow rate at stage i

3. b_i enriched flow rate at stage i

4. c_i inlet flow rate at stage i

5. f_i feed flow rate at stage i

6. d_{ij} fraction of depleted stream out of stage j going to stage i

7. e_{ij} fraction of enriched stream out of stage j going to stage i

8. u_{ik} feed concentration of component k at stage i

9. x_{ik} depleted concentration of component k at stage i

10. y_{ik} enriched concentration of component k at stage i

11. z_{ik} inlet concentration of component k at stage i

12. n number of stages in a cascade

13. n_e number of enriching stages

14. n_s number of stripping stages

15. D test statistic for a one-sample K-S test

16. n_F stage number where feed is put into the cascade

17. H_0 null hypothesis

18. H_1 alternate hypothesis

19. K the last component in a multi-component cascade

20. KS label used by *Minitab* to denote the K-S test statistic D

21. $K\text{-}S$ shorthand for the Kolmogorov-Smirnov test for normality

22. $P\text{-}Value$ the smallest significance level at which H_0 may be rejected

23. x_W fraction of the lighter component in the waste stream

24. y_P fraction of the lighter component in the product stream

25. z_F fraction of the lighter component in the feed stream

26. z_i fraction of the lighter component at stage i

27. α stage separation factor

28. $\alpha_{k,k'}$ separation factor between components k and k$'$

29. α_{i,k,κ_i} separation factor between components k and κ at stage i

30. β heads separation factor

31. γ tails separation factor

32. $\delta\alpha$ "small" portion of the separation factor equivalent to $\alpha - 1$

33. ζ weight, mole, or atom ratio in the feed stream

34. ζ_i abundance ratio in the feed at stage i

35. η weight, mole, or atom ratio in the heads stream

36. η_i abundance ratio in the product at stage i

UNCERTAINTY QUANTIFICATION OF MULTI-COMPONENT ISOTOPE-SEPARATION CASCADE

I. INTRODUCTION

A multi-component mathematical model of an enrichment cascade for gaseous diffusion technology has been developed. That work has been continued with the development of a multi-component isotope-separation cascade model for gas centrifuge technology. These steady-state models have been implemented in computer code. A new version of the code has been written for the cascade model. The work on this new code has been enhanced through the development of an uncertainty quantification (UQ) component that adds a new capability to the cascade model. The UQ capability uses a Monte Carlo approach to perturb input parameters that define the enrichment cascade. The propagation of uncertainties through the cascade is characterized by analysis of the variation of the resulting output data.

I.A. Objectives

In an enrichment cascade with interconnecting stages, uncertainties in the inputs can propagate through the cascade and cause uncertainties in the outputs. A characterization of these uncertainties is necessary to estimate the size of the errors in the outputs. To accomplish this task, the cascade code should be verified and an UQ capability should be developed and demonstrated.

1. *Verify Cascade Code for Two and Multiple Components*: Verify the accuracy of the cascade code for a two-component ideal cascade and for multi-component cascades.

2. *Develop and Implement Monte Carlo UQ Capability*: Develop the UQ methodology and implement the capability in code.

3. *Demonstrate UQ Capability*: Perturb a variety of input parameters defining an enrichment cascade for several multi-component problems.

4. *Characterize Uncertainty Output Data*: Plot the empirical cumulative distributions and analyze the uncertainty output to assess the propagation of uncertainties.

I.B. Background

Most elements have different isotopes that are chemically similar but that behave very differently in nuclear reactions. Depending on the specific nuclear application, certain isotopes may be more desirable than others. As a result, enrichment methods have been developed and employed to separate isotopes of interest. Enrichment methods use a series of stages to separate one isotope from another of the same element. Regardless of the specific enrichment technology, the separation efficiency for a single stage is generally small. To produce a sufficient purity of a desired isotope, many stages must be tied together in series to form a cascade. If the degree of separation is highly effective per stage, the cascade would require few stages to enrich the preferred isotope to the desired percentage. Likewise, highly inefficient separators would require a greater number of stages to produce that percentage of the desired isotope.

I.B.1 Current Isotope-Separation Models

A number of enrichment models already exist that handle cascades consisting of two components. However, multi-component models, other than a few approximation schemes that have been proposed, are largely an unsolved problem. In addition, solvers for isotope separation cascades are not readily available. Thus, a model that can solve for all the isotopes or components of an element would allow for a greater understanding of enrichment cascades.

A four-component enrichment model has been developed for 1-up, 1-down gaseous diffusion cascades [12]. That work has been extended into a more general mathematical framework that can handle modest-sized cascades with larger separation factors and consisting of four or more components [10]. This math model is called the multi-component isotope-separation cascade model, and it has been implemented in computer code. The cascade code uses a custom-built iterative process to solve the system of nonlinear equations in the cascade model.

A new version of the cascade code has been written with refinements in the equations that allow the solver to converge to a solution in fewer iterations [9].

I.C. Problem Statement

A Monte Carlo UQ capability is developed and implemented in code. The capability is demonstrated through perturbations of input data for a variety of multi-component enrichment cascades and characterization of the uncertainties in output data.

I.D. Scope and Limitations

Because the cascade model assumes a steady-state condition, the Monte Carlo UQ capability will work under the same assumption. As such, the time required for isotope concentrations to move between stages or between other specific parts like mixers, inlets, and outlets is not considered. Secondly, this thesis does not address parameters such as separative work, inventory, operating cost, discrete uncertainties (e.g., flow leakage), or physical uncertainties (e.g., the fabrication tolerances of actual machines). These parameters are beyond the scope of this research effort. The focus of this thesis will be on the input parameters: feed composition, cuts, and separation factors.

I.E. Approach

An enrichment cascade consists of interconnecting stages where uncertainties in the inputs for any one stage will result in errors in the isotope compositions everywhere in the cascade. Because the cascade model is comprised of nonlinear equations, the perturbation in the output is not readily computable. Additionally, standard propagation-of-error formulas generally assume normally-distributed errors and tend to be unworkable with complicated codes. Thus, UQ as applied to the cascade model requires a different method. The alternative is to use a Monte Carlo approach to characterize the propagation of uncertainties.

Generally, Monte Carlo methods use randomly generated numbers as inputs to evaluate a deterministic model in an iterative fashion. Monte Carlo is essentially a sampling method because the inputs required will be generated from

probability distributions. In effect, the process is similar to sampling from a population and is a useful approach for complex or nonlinear models. As a result, this technique is appropriate for the cascade model because the mathematical framework governing the model is a system of nonlinear equations.

The UQ capability quantifies the uncertainties in the inputs and determines the size of the errors in the output data to enhance the fidelity of the existing cascade model. Random error samples for all computational inputs of the cascade are drawn from its individual uncertainty distribution, and the errors are added to each input value to create perturbed values. These perturbed values define an enrichment problem that the cascade code solves. Once the code converges to a solution, the uncertainty output data are written to file. This entire process is repeated a specified number of times. The uncertainty output data are characterized using the empirical cumulative distribution and analyzed.

The Monte Carlo UQ capability is implemented in Fortran-95 code using good modern programming practices. The code is well-documented, flexible, portable, and accurate.

Chapter 2 describes the cascade model in greater detail. The set of equations that make up the cascade model are presented. Sources of uncertainties inherent in the input parameters to the cascade model are discussed. Last, how Monte Carlo propagation of uncertainties is handled in a code environment is explained.

Chapter 3 delves into the implementation of the UQ capability in Fortran-95 code. The organization of the code will be described in parts, starting with the input data. Pseudo-code for the UQ capability and the cascade solver are provided to show how the code takes input data and solves for a solution. The main program and supporting modules are broken down.

Chapter 4 presents the verification of the cascade code for two-component ideal cascade problems using a derived closed-form solution. Verification for multi-component problems is also accomplished by internal checks in the code, by computing the relative difference of two answers, and by checking whether the solution from the code makes good sense.

Chapter 5 gives the results from perturbing a variety of cascade inputs and analyzes the uncertainty output data. The data generated from the perturbed inputs are plotted and analyzed to see if the errors are Gaussian or non-Gaussian. In addition, other input perturbations are made to gain additional insights into enrichment cascades.

Chapter 6 summarizes the research done and gives observations obtained from conducting this thesis research. This chapter also provides conclusions and offers recommendations to continue the research.

II. THEORY

This chapter provides background on the cascade model, explains the sources of uncertainties that exist in an enrichment cascade, and how a Monte Carlo propagation of uncertainties can quantify those uncertainties. First, the equations and terms governing the cascade model are explained in detail. Second, all sources of uncertainties in the model are described. Finally, the UQ methodology is presented.

II.A. The Multi-Component Isotope-Separation Cascade Model

An enrichment cascade is a system of many parts. Individual machines that handle a small portion of the enrichment process are called units. Many units with their flows connected in parallel form a stage. Numerous stages are tied together in series to form a cascade in which feed material enters the inlet of one or more stages. These are called feed points.

Cascades come in a variety of types: simple, countercurrent recycle, square, ideal, and close-separation. This research deals mainly with the countercurrent cascade, and an example of this type of cascade called a 1-up, 1-down cascade is shown in Figure 1.

Figure 1. Example of *1 up-1 down* Cascade

A *1 up-1 down* cascade allows the enriched flow from one stage to be routed to the stage above while the depleted flow is moved to the stage below. The enriched flow from below and the depleted flow from above come together at the mixer and leave the opposite side as a single inlet stream that goes into a separator. In Figure 1, each separator is identified as a particular stage. If feed is introduced at that particular stage, the inlet stream will be comprised of the feed flow in addition to the enriched flow from below and the depleted flow from above. For illustration purposes, only one feed point is shown in Figure 1 although many feed points can exist in a real cascade operation.

All stages above the feed stage are known as enriching stages because their purpose is to enrich the desired isotope to a higher concentration. Likewise, all stages below the feed point are called the stripping stages because they strip the desired isotope out of the stream and send it back up to the feed stage. Given the division of enriching stages and stripping stages, the depleted stream that comes out of stage *1* is called the waste stream. Similarly, the enriched stream that comes out of stage n is called the product stream.

Figure 2 shows stage i of an enrichment cascade. The numbered circles are called mixers. These parts receive material from three sources: the feed stream (coming in the left side in this diagram), the depleted stream coming down one or more stages above (indicated by a dashed line), and the enriched flow coming underneath from the stage below (indicated by a solid line). The enriched stream leaving the stage enters a flow splitter from which portions can be routed back to the stage i inlet, to other stage inlets, or be removed as product. The depleted stream is also split with portions to stage i and other stage inlets or as waste.

The terms a_i, b_i, and c_i represent the mass flow rates of the tails, heads, and the inlet streams at stage i, respectively. The terms "tails" and "heads" are often used synonymously with "depleted" and "enriched", respectively. The term f_i is the mass flow rate of the feed stream at stage i. The d and e factors, each written with two subscripts, represent the fractions of the depleted and enriched streams, respectively. The factor d_{ij} is the fraction of the depleted stream coming out of stage j that is directed to the inlet mixer at stage i. Likewise, e_{ij} represents the fraction of the enriched stream out of stage j into stage i. The product and waste flows at stage i are p_i and w_i, and the fractions from stage i to product and waste are e_{0i} and d_{0i}, respectively.
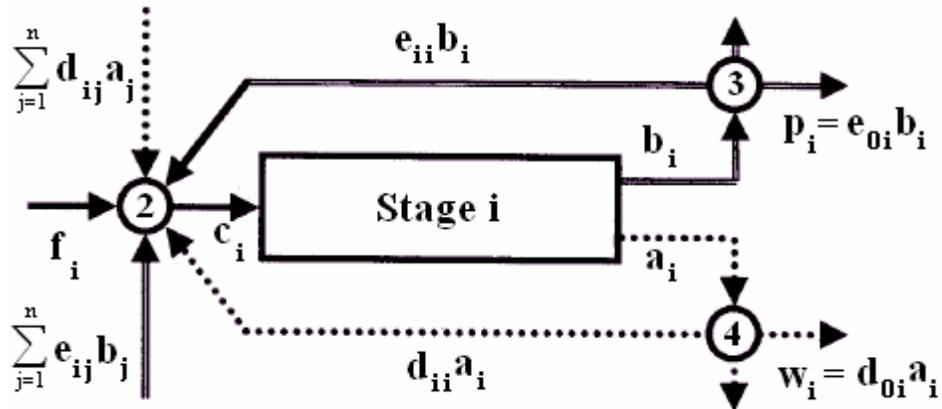

Figure 2. Possible Flow Paths for Stage i

Because the cascade is assumed to be at steady-state in the current model, the inlet mass flow rate must equal the sum of the outlet mass flow rates at each stage. The separator flow balance equation,

$$c_i = a_i + b_i, \tag{1}$$

expresses the conservation of mass. The separator splits the flow into the enriched and depleted streams. The fraction that is in the enriched stream is called the cut:

$$\theta_i \equiv \frac{b_i}{c_i}. \tag{2}$$

The efficiency of a cascade is often very sensitive to the cuts. The equations for flows make up a linear system in the cascade model.

The mass of each component is also conserved; we write the separator component-flow balance equation as

$$c_i z_{ik} = a_i x_{ik} + b_i y_{ik}, \tag{3}$$

where x_{ik}, y_{ik}, and z_{ik} are the fractions of component k in the depleted, enriched, and inlet flows at stage i, respectively.

In the special case of a *1 up-1 down* cascade, the flow balance equation at the inlet mixer is

$$c_i = a_{i+1} + b_{i-1} + f_i. \tag{4}$$

In order to make our model applicable to any design, we generalize the inlet flow balances to account for the outlet flow splitting factors d and e. Thus, accounting for flows that come from all the other stages, the inlet mixer flow balance equation is

$$c_i = \sum_{j=1}^{n} d_{ij} a_j + \sum_{j=1}^{n} e_{ij} b_j + f_i \tag{5}$$

where n is the number of stages.

To account for the component fractions at each stage, the x, y, and z factors are included in the inlet mixer flow balance equation to form the inlet mixer component-flow balance equation

$$c_i z_{ik} = \sum_{j=1}^{n} d_{ij} a_j x_{jk} + \sum_{j=1}^{n} e_{ij} b_j y_{jk} + f_i u_{ik} \tag{6}$$

where u_{ik} is the fraction of component k in the feed flow at stage i.

The separator performance equation, the nonlinear aspect of this system of equations, is

$$\alpha_{i,k,k'} = \frac{y_{ik}/x_{ik}}{y_{ik'}/x_{ik'}} \tag{7}$$

at every stage i for every pair of components k and k'. The performance parameters $\alpha_{i,k,k'}$ must satisfy the relations $\alpha_{i,k,k'}\alpha_{i,k',k''} = \alpha_{i,k,k''}$ for all choices of the subscripts. Thus, equations (7) are redundant. To avoid needless poor-conditioning, the solver applies only the equations for $k \neq \kappa_i$, $k' = \kappa_i$ at stage i where κ_i is the dominant component at stage i: $z_{i\kappa_i} > z_{ik}$ for $k \neq \kappa_i$. The distinction of κ_i is important because the dominant component depends on the stage.

To simplify notation, the convention of using only the last digit of each component as subscripts will be adopted here. Hence, the separation factor for stage i consisting of ^{235}U and ^{238}U is written as

$$\alpha_{i,5,8} = \frac{y_{i5} x_{i8}}{y_{i8} x_{i5}}. \tag{8}$$

To summarize, the equations that make up the cascade model are the separator flow balance equation, the equation for cuts, the inlet mixer flow balance equation, the separator component-flow balance equation, the inlet mixer component-flow balance equation, and the separator performance equation. They are equations (1), (2), (3), (5), (6), and (7), respectively. Together, these equations allow for the modeling of multi-component enrichment cascades of various sizes, feed points, and connection patterns.

II.B. Sources of Uncertainties in the Cascade Model

Mathematical uncertainties can stem from a loss of precision in calculations. Examples include round-off errors and errors as a result of subtraction between two nearly identical values that can result in a catastrophic cancellation. Many small uncertainties can, over time, accumulate to give unrealistic answers. For any system of equations, identifying the sources of uncertainties in the inputs and understanding how they propagate are necessary to know the reliability of the answers in the outputs. For the system of nonlinear equations that define the cascade model, propagation of uncertainties becomes crucial because small perturbations in the inputs may cause comparatively large changes in the outputs.

Most input parameters that define an enrichment cascade are sources of uncertainties. The inputs with uncertainties are the cuts, the separation factors, and the starting feed composition. Cuts and separation factors are stage dependent, so every stage has its own cut and separation factor. For a small cascade of 50 stages, there would be 100 input uncertainties already if only the cuts and the separation factors were considered. For feed composition, each component is a source of uncertainty because the percentage of each isotope is not an exact value. Because a cascade can have multiple feed points with each feed stage containing a different composition, it is clear that the number of uncertainties can be quite large.

As an enrichment cascade starts, many other uncertainties come into play. Figure 2 shows that the feed stream enters an inlet mixer and is joined by two other streams, a depleted stream coming from the stage above as well as an enriched stream coming from the stage below. The fractions of the depleted and enriched streams, d_{ij} and e_{ij}, are sources of uncertainties. Because the inlet stream is comprised of these three streams, the inlet stream contains many uncertainties as it enters a stage for enrichment. As a stream leaves a stage after separation, the stream splits into an enriched stream and a depleted stream. Every time a flow is split, two additional uncertainties are introduced into the system. For example, suppose a flow is split into two pipes, and 50% of the mass

goes into each pipe. The amount that enters each pipe is a source of uncertainty because there is no guaranteed that exactly 50% will be present in each of the two pipes. One pipe may have slightly more material while the other pipe may have slightly less material. As a result, as material moves along the enrichment process, its numerous uncertainties will propagate throughout the cascade as the stream enters new mixers, combine with other streams, and enter new stages for further separation.

Because each stream is a combination of isotopes, the components introduce new uncertainties as their concentrations change from the start of the cascade to the end. Uncertainties in the separation factors affect uncertainties in component fractions because the separation factor is defined as the quotient of isotopic ratios. Similarly, the uncertainties in the cuts affect the uncertainties in the mass flow rates because the cut is defined as the ratio of the enriched flow to the inlet flow.

II.C. Monte Carlo Propagation of Uncertainties

The first step is to determine all the sources of uncertainties in the enrichment cascade. These uncertainties have been identified as input parameters to the cascade model. For each uncertainty, random samples of error are drawn from its individual uncertainty distribution. The Intel Math Kernel Library (MKL) software [7] has a suite of routines as part of its Vector Statistical Library (VSL) that I use to generate the random error samples. Once the errors are drawn, they are added to the input value to create a set of perturbed problems which are then passed to the cascade code to be solved one at a time.

The code outputs consisting of waste and product flow rates and component fractions for each perturbed problem are written to files. Because natural uranium is used as the example element, each output file will contain either flow rate data or uranium component fraction data for the entire set of perturbed problems. For each perturbation of an input parameter, eight output files get generated with data on product flow rates, waste flow rates, and product and waste fractions for each component. Once all output files are obtained,

Mathematica™ is used to plot the empirical cumulative distributions. Characterizations of the uncertainty output data are done using *Minitab* software.

III. IMPLEMENTATION OF THE UQ MODEL

This chapter explains how the sources of uncertainties and the Monte Carlo UQ capability are implemented in Fortran-95 code. A top-down design was done to deconstruct the problem into logical parts that could be managed, coded, and tested individually. The organization of the code and the input data are discussed before the individual code pieces are explained in detail.

III.A. Organization of the Code

The UQ code is written to draw random samples of error to create perturbed inputs which are then passed to the cascade code to solve. One main program and ten modules make up the entire UQ code and the cascade code. The main program calls on the various modules to perform all calculations and to write all data to file. This organization makes the code flexible and allows the modules to be reusable. The entire program is compiled using Intel Visual Fortran Compiler Version 11.1.048 [8]. Additional routines used in this research include Intel MKL Version 10.2.2.025 software [7] to solve the systems of linear and nonlinear equations in the cascade model and to generate random Gaussian samples of error for the UQ portion.

III.B. Input Data

The main input data to the cascade solver routines are the cuts, the separation factors, and the component concentrations. Additional input data are the number of components, the number of stages, and the feed stage. Because each input can be perturbed individually and also jointly, the number of possible perturbations of the inputs can quickly become daunting. The purpose of this research is to build and demonstrate an UQ capability, rather than to test how every cascade input or combination of inputs can possibly be perturbed. Therefore, I constrained the types of problems that were explored in order to keep the data sets manageable. For this research, enrichment problems were limited to a *1 up-1 down* cascade with a single feed stage. Except where noted,

14

problems were kept to 50 stages using three components with feed composition consisting of natural uranium.

III.C. Main Program and Supporting Modules

The methodology has been described fully in Chapter 2. The pseudo-code for the UQ portion is shown in Figure 3. The descriptions of the actual code follow the pseudo-codes.

UQ Pseudo-code Algorithm

Initialize:

1. The number of random samples to be drawn.
2. The number of problems to be solved.
3. The number of stages.
4. The number of components.
5. The stage number where feed is fed into the cascade.

End of Initialization

6. Create all arrays.
7. Prompt user to provide a starting value for specific cascade input.
8. Initialize uranium component fractions.
9. Set the mean and standard deviation for the cascade input.
10. Draw random samples of error.
11. Add error samples to cascade input to create perturbed inputs.

DO i=1, nProblems

12. Solve the enrichment cascade problem. See Figure 4 for the pseudo-code for this part.

END DO

13. Write output data to file.

Figure 3. UQ Pseudo-code Algorithm

When the perturbed inputs have been created, they are passed to the solver. Figure 4 shows the pseudo-code for how an enrichment problem gets solved in the cascade code.

15

<div align="center">Cascade Code Pseudo-code Algorithm</div>

1. Set cascade type as *1 up-1 down.*

2. Populate d and e matrices and p and w vectors.

3. Set feed stage.

4. Populate u and z matrices.

5. Compute separation factors.

6. Set the cuts.

7. Solve for flow rates c, b, and a, in this order.

8. Solve for relative feed.

9. Populate the *kappa* vector with the dominant component.

DO

10. Iteratively solve for inlet concentration z.

11. Solve for the depleted and enriched fractions, x and y, respectively.

UNTIL converged

Figure 4. Pseudo-code for Cascade Code

Now that the pseudo-codes have been presented, the main program and supporting modules will be discussed in detail. Related computations have been coded in either functions or subroutines and grouped under the same module. Together, they generate the random error samples and solve the enrichment problems.

UncertaintyQuantificationMC Program

The *UncertaintyQuantificationMC* program calls on subroutines to perturb the input parameters set by the user, calls other subroutines to solve the perturbed problems, and writes the uncertainty output data to files. The program starts by allowing the user to initialize input parameters: the number of random samples to be drawn, the number of problems that will be solved, the number of components in the cascade, the number of stages in the cascade, and the feed

stage. Next, all the arrays that will be used to store intermediate computations or to hold output data are created. The program then prompts the user to give an initial value to a particular cascade input or set of inputs. The value is passed to a subroutine that generates the random samples and adds these samples to the user's chosen value. The perturbed inputs are then sent through the cascade solver one at a time to be solved. The solution is written to file each time. Once the entire set of problems is solved, the uncertainty output data are written to text files.

CreateUserInterface Module

This module contains four subroutines. The first three subroutines prompt the user to give a starting value for one of three input parameters: the cut, the separation factor, or a *dAlphaForUnitMassDifference* value which is used by the program to calculate the separation factors for all the stages based on the mass difference of the two constituent components. The fourth subroutine, *SetComponentFractions*, initializes the uranium isotopes to values found in natural uranium, depending on the number of components specified by the user. For two-component problems, ^{235}U is set to 0.007 and ^{238}U is set to 0.993. For three-component problems, ^{234}U is set to 0.0001, ^{235}U is set to 0.007, and ^{238}U is set to 0.9929. For four components, the values are 0.0009, 0.007, 0.0001, and 0.992 for ^{234}U, ^{235}U, ^{236}U, and ^{238}U, respectively. While the code certainly allows for values other than natural uranium percentages, these values have been chosen to constrain the problem for manageability as an initial demonstration of the UQ capability.

RandomSamples Module

This module has two subroutines. The subroutine *GetInputParameters* sets the mean and standard deviation for each input parameter. The second subroutine, *DrawSamples*, receives the mean and standard deviation and passes them to a random number generator to generate normally-distributed error samples. The number of samples drawn is initialized by the user at the outset of

the program. To perturb the cuts and the separation factors, the number of samples drawn is determined by the number of stages multiplied by the number of different problems to be solved. For example, a 50-stage enrichment problem would require 50 perturbed cuts and 50 perturbed separation factors, one for each stage. If one set of data required 200 problems to be solved, 10,000 random samples would need to be generated for the cuts and another 10,000 random samples would be needed for the separation factors.

DataModule Module

This module has numerous subroutines that set up the necessary conditions for an enrichment problem to be solved. First, the module allocates all the arrays necessary to hold the data and initializes them to 0. Second, the subroutine *SetFixedUpDownConnections* initializes the matrices, D and E, to hold the depleted and enriched fractions as well as the product and waste vectors, p and w, respectively. Next, the subroutine *SetOneFeedPoint* initializes the feed flow rate vector, f, the feed concentration matrix, u, and the inlet concentration vector, z. Because only one feed point is used, f is set to 1 at the feed stage and to 0 at all other stages. If used, *SetLinearCuts* can set the cut for all stages to the same value. *SetAlphaByMassDifferences* populates a three-dimension array with *dAlpha* values. In the code, the separation factor α is written in the form

$$\alpha = 1 + \delta\alpha \tag{9}$$

to preserve significant digits during computations. The values of *dAlpha* are just the $\delta\alpha$ portion of equation (9). *FindFlows* solves for the inlet, enriched, and depleted flow rates, c, b, and a, respectively. Finally, *SetSources* computes the relative feed matrix, s, from vectors f and c and matrix u. The equation for s is

$$s_{ik} = \frac{f_i u_{ik}}{c_i}. \tag{10}$$

CascadeRoutines Module

This module contains subroutines that solve for z, x, and y which are the inlet, depleted, and enriched fractions, respectively, for all components at every stage. *CheckSolution* is a subroutine that ensures that all four equations of the cascade model maintain balance at each stage for each component. That is, what is calculated for the left-hand side of the equation must equal the right-hand side of the equation to a specified tolerance.

LinearAlgebraUtilities Module

This module contains several functions that calculate the norms of vectors and matrices, a function that uses Gaussian elimination to solve a system of equations, and a function that solves the SRD between two scalars, vectors, or matrices. The Gaussian elimination function is present but not used to solve the systems of equations in this research. Instead, the Parallel Direct Sparse Solver (PARDISO) from MKL is used. The Gaussian elimination function is maintained for code flexibility in case another user cannot access the PARDISO subroutines.

Other Modules

The modules under this section play secondary support roles to the main program and the primary modules. *WriteToFile* writes all uncertainty output data to text files for analysis. *Kinds* is a module that sets the precision for integers and real numbers used in the program. *FileUnits* assigns each unit number a descriptive name and is used to open and close files for data to be read from or written to file. This naming convention allows for better documentation of code. *SparseSolverRoutines* contains a subroutine to solve systems of equations of the form

$$Ax = b \tag{11}$$

for vector x where A is a matrix of size n by n and b is a vector of length n. The subroutine *PardisoMethod* makes calls to PARDISO to solve for the inlet flow rate, c, and the inlet concentration, z, at each stage. *IdealCascade* contains a

subroutine to solve for z at each stage using a closed-form solution and a function to solve for the corresponding cut at each stage. This module is used as part of the verification process to check whether the answers from the code match the closed-form solution.

IV. VERIFICATION OF THE CASCADE CODE

This chapter verifies the cascade code for the two-component ideal cascade type as well as for multi-component cascades. Validation is not considered at this time because validation requires knowledge of the actual physical measurements of an enrichment plant or specific machines. Such an effort is beyond the scope of this thesis, so verification testing is the focus.

IV.A. Verification of Two-Component Ideal Cascade

The cascade code can be verified for two components by using equations presented in Benedict, Pigford, and Levi [2:660] as well as an additional derived equation. The first step is to define the number of stages in the ideal cascade to be solved. For an ideal cascade, the number of stripping stages, n_s, is given by

$$n_s = \frac{\text{Log}\left[\dfrac{z_F\left(1-x_W\right)}{x_W\left(1-z_F\right)}\right]}{\text{Log}\left(\beta\right)} \tag{12}$$

and the number of enriching stages, n_e, is given by

$$n_e = \frac{\text{Log}\left[\dfrac{y_P\left(1-z_F\right)}{z_F\left(1-y_P\right)}\right]}{\text{Log}\left(\beta\right)}. \tag{13}$$

In equation (12), z_F is the fraction of the lighter component in the feed stream, x_W is the fraction of the lighter component in the waste stream, and β is the heads separation factor. In equation (13), y_P is the fraction of the lighter component in the product stream. We define n_F as the stage where feed is fed into the cascade.

The equation for the heads separation factor is

$$\beta = \frac{y/\left(1-y\right)}{z/\left(1-z\right)} = \frac{\eta}{\zeta} \tag{14}$$

and the equation for the tails separation factor is

$$\gamma = \frac{z/(1-z)}{x/(1-x)} = \frac{\zeta}{\xi} \tag{15}$$

In an ideal cascade, the heads separation factor is equal to the square root of the stage separation factor, α, which is also equivalent to the tails separation factor. The equation can be represented as

$$\beta = \sqrt{\alpha} = \gamma. \tag{16}$$

The equation that describes the relationship between the abundance ratio in the product η_i and the abundance ratio in the feed ζ_i at each stage is given by

$$\eta_i = \beta \zeta_i. \atop i = 1,...,n \tag{17}$$

The equation that gives the concentration of the lighter component at each stage can be written as

$$z_i = \frac{z_F \beta^{(i-F)}}{1 - z_F \left(1 - \beta^{(i-F)}\right)}. \atop i = 1,...,n \tag{18}$$

The derivation for Equation (18) is given in Appendix A.

If the concentration of the lighter component is known for each stage, the cut at each stage of an ideal cascade is computed with the equation

$$\theta_i = \frac{1 + z_i (\beta - 1)}{(\beta + 1)}. \atop i = 1,...,n \tag{19}$$

To verify the cascade code for two components, values were chosen for $x_W, z_F, y_P, \alpha, n_F$, and n to create an ideal cascade problem. The values are listed in Table 1. Using equations (12) and (13), these values give 34 enriching stages and 7 stripping stages for a total of 41 stages. The feed stage is set to 8 which makes it part of the enriching section.

Table 1. Parameters to Verify Two-Component Ideal Cascade Problem

x_W	0.003
z_F	0.007
y_P	0.208
α	1.237
n_F	8
n	41

Using uranium as the element to be separated, the 41-stage problem was put into the cascade code and the inlet concentration at each stage was calculated using equation (18). The cascade code then solved the same problem again using its own algorithm. Next, the symmetric relative difference (SRD) between the closed-form solution and the code solution was computed for each stage. The SRD is the fraction one answer is different than another answer. Therefore, it is a much more stringent test for the convergence of two numbers than absolute difference. The equation for SRD is

$$SRD(x,y) = 2\frac{|x-y|}{|x|+|y|} \tag{20}$$

The largest SRD and the average SRD between the closed-form solution and the cascade code for all stages are given in Table 2.

Table 2. SRDs Between Closed-Form Solution and Cascade Code

Largest SRD	Average SRD
2.56×10^{-11}	7.97×10^{-12}

Table 2 shows that the largest SRD for all stages is 2.56×10^{-11} while the average SRD is 7.97×10^{-12}. These two SRDs illustrate that the inlet concentrations for [235]U computed from the closed-form solution and the cascade

code match to a very high degree. Answers from both methods match to at least ten digits.

IV.B. Verification for Multi-Component Cascades

Verification of the cascade code for multi-component problems is somewhat different because closed-form solutions do not exist when the number of components exceeds two. Hence, it is necessary to consider other verification methods besides exact formulae.

The first verification test for a multi-component cascade is actually part of the code itself. The *CascadeRoutines* module contains a subroutine called *CheckSolution* that ensures that all four equations of the cascade model—the inlet mixer flow balance equation, the separator component-flow balance equation, the inlet mixer component-flow balance equation, and the separator performance equation—are balanced for all components at all stages to specified tolerances. In other words, the left-hand side of each of the four equations must equal the right-hand. In addition, the sum of components must be equal to one for all stages. That is,

$$\sum_{k=1}^{K} x_{ik} = \sum_{k=1}^{K} y_{ik} = \sum_{k=1}^{K} z_{ik} = \sum_{k=1}^{K} u_{ik} = 1 \tag{21}$$

for component k at stage i where K is the number of components.

The inflow mixer equations and separator equations use a tolerance of 10^{-8}, the sum of components use a tolerance of 10^{-10}, and the separation factors use a tolerance of 10^{-12}. If the SRD between the left-hand side and the right-hand side exceeds the tolerance specified, an error message is printed to screen. When testing several well-defined problems with reasonable inputs, none of the error messages were triggered.

The second verification test for multi-component problems is to ensure that the solutions give consistently accurate digits. Again, using uranium isotopes, a four-component 50-stage problem was solved twice, once with double precision arithmetic and a second time with quadruple precision arithmetic. The SRD was then computed between the two answers at every stage to see how

closely the two precisions match. The largest SRDs and average SRDs for all flows and all concentrations of ^{235}U are shown in Table 3.

<div align="center">Table 3. Largest SRDs and Average SRDs of ^{235}U</div>

	Largest SRD	Average SRD
Depleted Flows	4.78×10^{-15}	1.41×10^{-15}
Enriched Flows	4.96×10^{-15}	1.26×10^{-15}
Inlet Flows	2.99×10^{-15}	1.17×10^{-15}
Depleted Fractions	3.55×10^{-14}	3.21×10^{-15}
Enriched Fractions	3.54×10^{-14}	3.01×10^{-15}
Inlet Fractions	3.52×10^{-14}	2.89×10^{-15}

Table 3 shows that the largest SRDs and the average SRDs between the double precision run and the quadruple precision run are consistently small. This indicates that the two answers match to a high degree of accuracy, and the solutions are good to at least 13 significant digits. The small SRDs indicate the problem is well-conditioned and the tolerances are met.

The third and final verification test for multi-component problems is to ensure that the solution to the problem is reasonable. That is, the outputs should make logical sense given a proper understanding of uranium enrichment. For example, the heaviest component, ^{238}U, should decrease while the lighter components, ^{234}U and ^{235}U, should increase as the stage number increases. For the four-component problem used, this is certainly the case as seen in Figure 5. The digits "4", "5", "6", and "8" correspond to ^{234}U, ^{235}U, ^{236}U, and ^{238}U, respectively. The plot also shows that ^{235}U peaks at stage 44 and then its concentration decreases a small amount as it goes to the final stage.
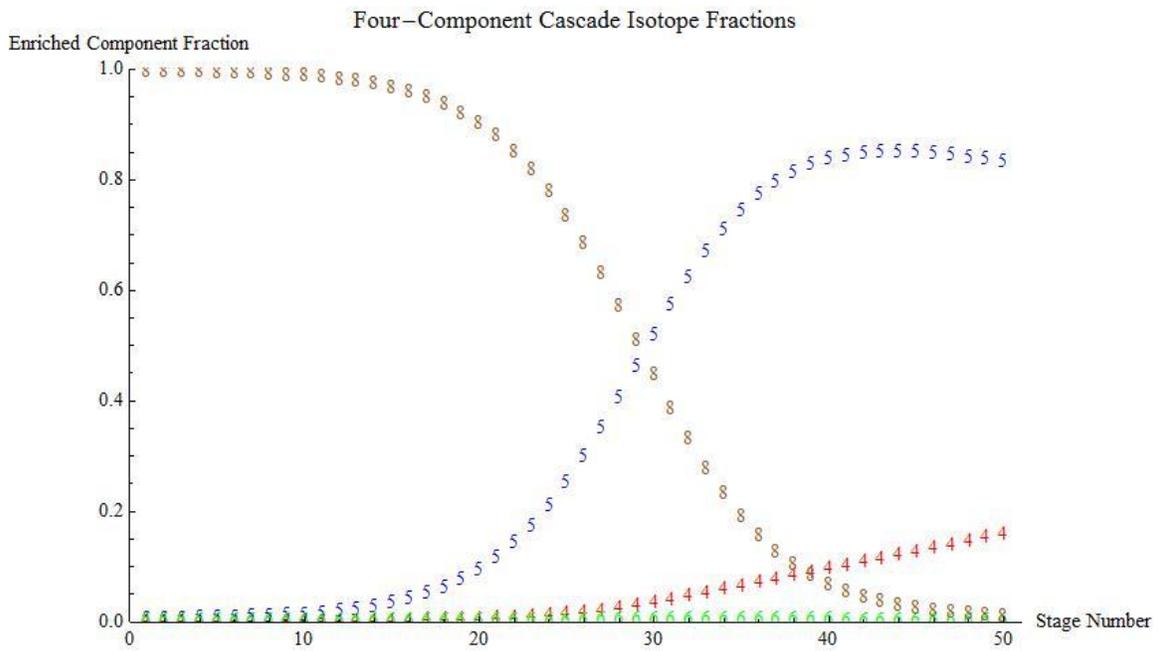
Figure 5. U Component Fractions at Each Stage

26

V. PERTURBATION RESULTS AND ANALYSIS

The following constraints were set to define enrichment problems that were then perturbed to produce the uncertainty output data. These constraints make up a well-defined problem that was used to verify the code for multi-component cascades.

1. Number of Components: 3
2. Number of Stages: 50
3. Cascade Type: *1 up-1 down* cascade
4. Feed Point: Stage 10
5. Feed Composition: Natural feed consisting of 0.01% ^{234}U, 0.7% ^{235}U, and 99.29% ^{238}U
6. Number of Problems Solved: 200
7. Tolerances
    a. Inlet fraction: 10^{-12}
    b. Normalization: 10^{-10}
    c. Inflow mixer and separator balance: 10^{-8}
    d. Separation factor: 10^{-12}

V.A. Characterizing Output Data from Gaussian Inputs

For the first test, the cuts were perturbed while the separation factors were held constant. For the second test, the separation factors at each stage were varied while the cuts were held constant. For the third test, both cuts and separation factors were varied simultaneously.

For each test, two graphical displays are given. The first graphical display is three plots in one. One plot is the empirical cumulative distribution function, *ECDF*, the enriched ^{235}U product at stage *50* for all 200 problems. A second plot labeled *Normal Fit* is the normal distribution with the same mean and standard deviation as *ECDF*. Likewise, the third plot, *ECDF Normal*, is a plot of 200 randomly generated numbers of normal distribution with the same mean and standard deviation as *ECDF*. The x-axis is the amount of enriched ^{235}U at stage

50, and the y-axis is the probability $1/n$ for each of the n problems in the sample.

The second graphical display is the result of a Kolmogorov-Smirnov (K-S) test used to test the normality of the empirical data. A one-sample K-S normality test is a normality test that compares the ECDF of sample data with a distribution that would be expected if the data were normal. *Minitab* [11] was used to conduct all K-S tests. The one-sample K-S test performs a hypothesis test to determine whether the empirical data follow a normal distribution or not (see Appendix B for a full explanation of hypothesis testing). The null and alternative hypotheses are, respectively,

H_0: The data follow a normal distribution.

H_1: The data do not follow a normal distribution.

The K-S test gives a plot with five values as outputs: the mean, the standard deviation, the sample size N, the test statistic KS which is also known as the D statistic, and the probability *P-Value*. The mean and standard deviation are calculated from the empirical data while N is the number of data points in the test sample. The test statistic KS is the maximum vertical distance between the ECDF of the empirical data and the CDF of the Gaussian that has the same mean and variance as the empirical data. The *P-Value* is the smallest significance level at which H_0 may be rejected.

The outputs of the K-S graph provide two ways to test whether the empirical data are normally distributed or not. The first way is to choose an α-level and then use this α-level to compute a decision value. If the KS statistic is greater than the decision value, then the test statistic falls into the critical or rejection region, and H_0 can be rejected. The conclusion would be that the data does not follow a normal distribution. If the KS statistic is less than the decision value, then the test statistic resides within the noncritical or nonrejection region, and H_0 cannot be rejected. A failure to reject H_0 means there is not sufficient reason to conclude the data does not follow a normal distribution. For a two-sided test where the α-level is 0.01, Massey [8:70] gives the equation to compute the decision value as

$$decision\ value = \frac{1.63}{\sqrt{N}} \qquad (22)$$

where N is the sample size greater than 35. For sample sizes of 35 or fewer, pre-calculated decision values can be obtained from tables. Because each data set was generated using 200 problems, the decision value is 0.115 for the K-S tests that will be performed. A decision value good to three decimal places allows for a consistent comparison with the output values given by *Minitab*.

The second way is to use *P-Value*. If this value is less than the chosen α-level, H_0 can be rejected. Otherwise, if this value is greater than the α-level, H_0 cannot be rejected.

Varying All Cuts for All Stages, ^{235}U at Stage 50

Using a 50-stage cascade, each stage was given a different perturbed cut. Each 50-stage cascade represented one problem, and 200 problems were solved. For each problem, the cuts were varied between 0.456 and 0.480 on the order of 0.01. This range of cuts allowed ^{235}U to enrich to a sufficiently high percentage of 80% or higher. While the cuts could theoretically have values between 0 and 1, this particular range was chosen because the effect on the ^{235}U fractions would be easily noticeable from any perturbations. The separation factors were kept constant and were 1.42 for separating ^{234}U from ^{238}U and 1.30 for separating ^{235}U from ^{238}U.

From looking at the *ECDF* plot in Figure 6, it does not appear to be Gaussian. The plot lacks the right shape or symmetry. None of the data points of *ECDF* are close to the data points of *ECDF Normal* except potentially where the two plots intersect. If the output data were normally distributed, the *ECDF* plot would have matched the *ECDF Normal* plot fairly well.
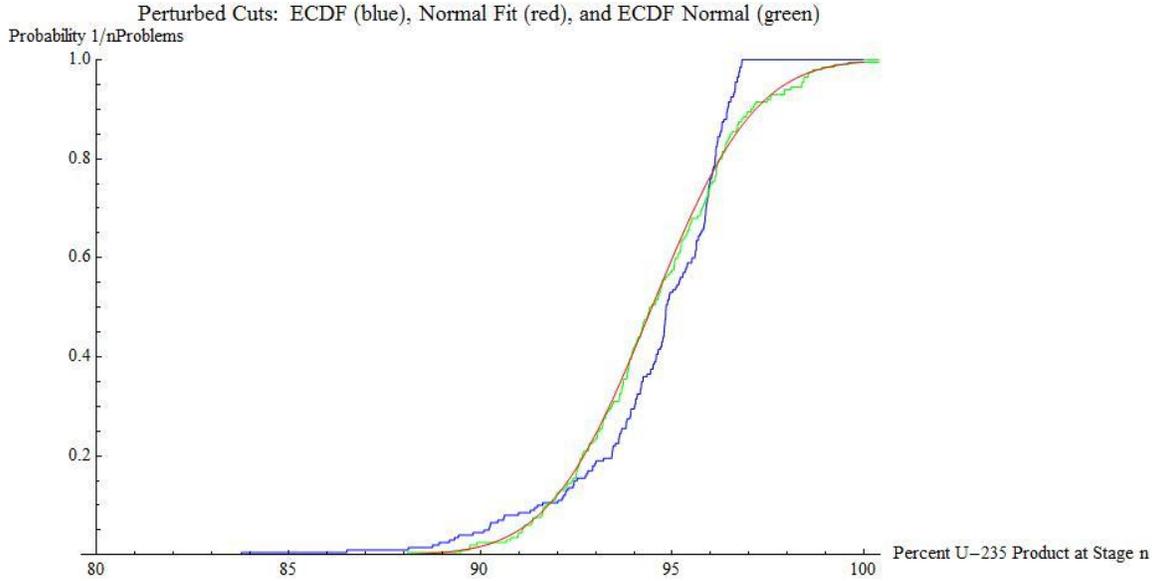
Figure 6. Perturbed Cuts for all Stages, ^{235}U at Stage 50

Figure 7 shows the results of the K-S test on the data obtained from perturbing 50 cuts. The plot in red is the distribution of enriched ^{235}U at stage *50*. The straight line is an estimate of the cumulative distribution function for the sample from which the data are drawn. If the empirical data were normally distributed, it would form a fairly straight line and overlap the blue line. The empirical data clearly do not match the straight blue line. The KS value of 0.132 is greater the decision value of 0.115. The *P-Value* is also less than 0.01 which is less than the α-level. The null hypothesis is rejected, and the conclusion drawn is that the uncertainty output data are not normally distributed when cuts were perturbed for all 50 stages. With this conclusion, there is less than a 1% of making a Type I error by rejecting the null hypothesis.

30
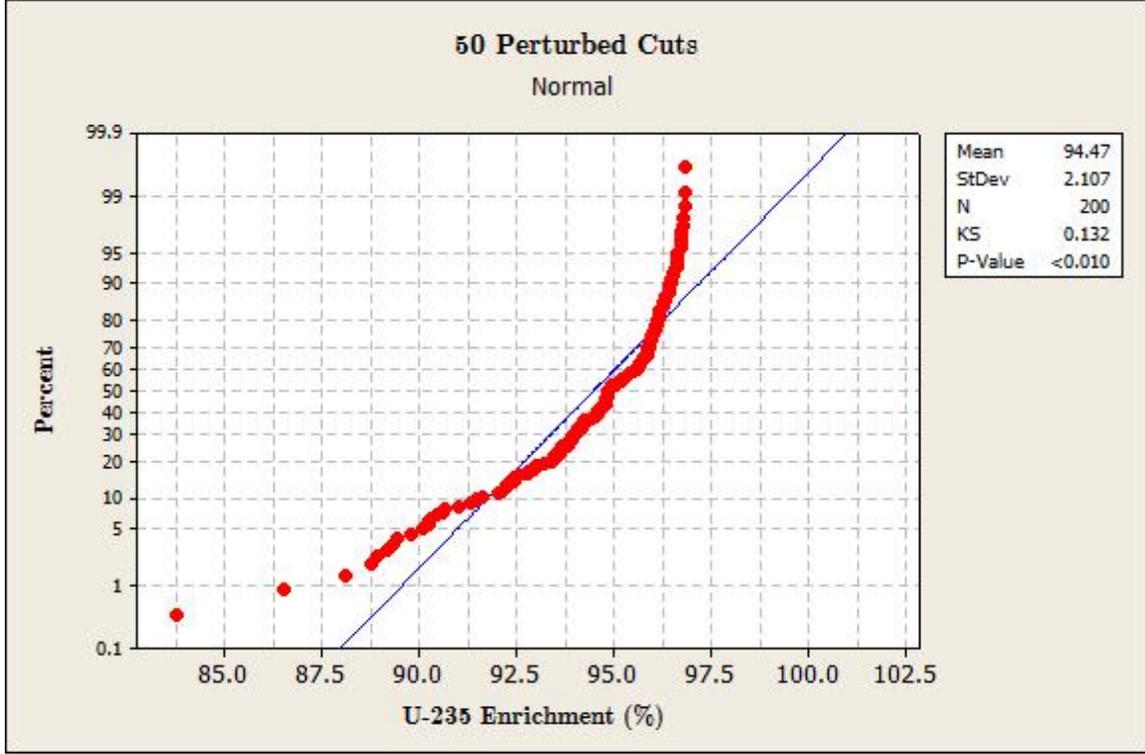
Figure 7. K-S Test for 50 Perturbed Cuts, ^{235}U at Stage 50

Varying All Cuts for All Stages, ^{235}U at Stage 25

As a comparison, the same problem was solved again except the enriched ^{235}U percentage was taken from stage *25*, the middle of the cascade. Figure 8 shows a distribution than is more Gaussian than the enriched ^{235}U at stage *50*. The K-S test in Figure 9 has a test statistic of 0.111 which is less than 0.115. The null hypothesis cannot be rejected at the 0.01-level. However, at the 0.05-level, Massey [8:70] gives the decision value equation as

$$decision\ value = \frac{1.36}{\sqrt{N}}. \tag{23}$$

This gives a decision value of 0.096. We can reject H_0 and say the distribution is not normally distributed at the 0.05-level.

There is a discrepancy between the implications of the test statistic value and the p-value that is worth noting. The test statistic indicates a failure to reject H_0 while the small p-value indicates H_0 should be rejected. Because

Massey's equations for computing decision values date back to 1951, there are good reasons to believe his approximations have limited precision. While these approximations may have been acceptable in 1951, their accuracy would not compare favorably with the approximations computed by modern computers. It is more reasonable to think that *Minitab* has correctly computed the test statistic and p-value for this particular problem.

Cuts Varied at All Stages, U–235 Taken at Stage 25: ECDF (blue), Normal Fit (red), and ECDF Normal (green)
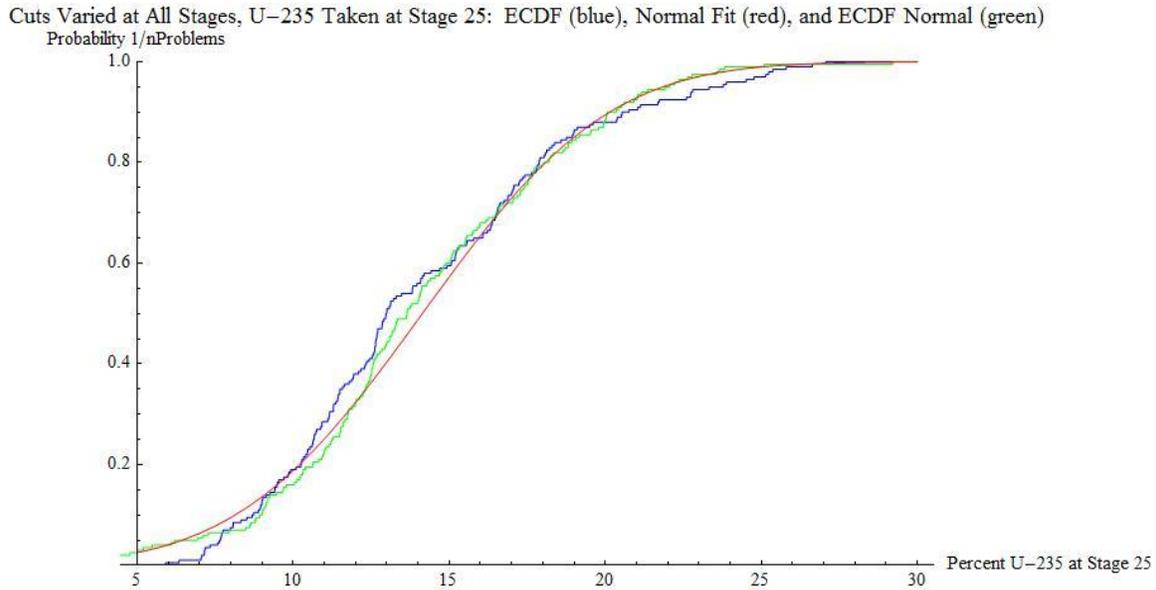
Figure 8. Perturbed Cuts for all Stages, ^{235}U at Stage 25
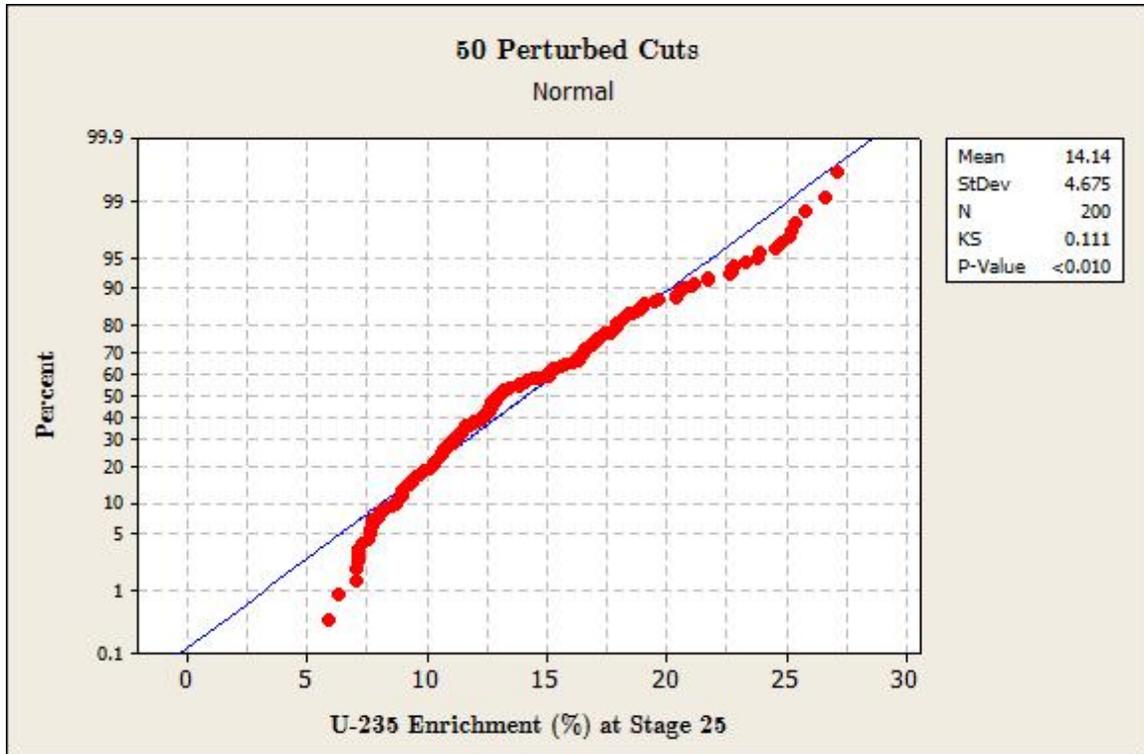
Figure 9. K-S Test for 50 Perturbed Cuts, ^{235}U at Stage 25

Varying Separation Factors for all Stages

For the second set of problems, the separation factors were varied between 1.00261 and 1.917. The *dAlphaForUnitMassDifference* input was set to 0.1 with a mean of 0 and a standard deviation of 0.01 to get this range of values for the separation factors. This gave a good range of alphas to test how varying this input parameter would affect the enriched ^{235}U product. The cut was kept constant at 0.47 for all stages. Figure 10 shows outputs that do not look quite Gaussian. The K-S test in Figure 11 gives a *KS* statistic of 0.179 which is greater than 0.115, and a very small p-value of less than 0.01. Given the results of these two values, the conclusion drawn is that perturbing alphas with Gaussian errors in this instance give non-Gaussian outputs.
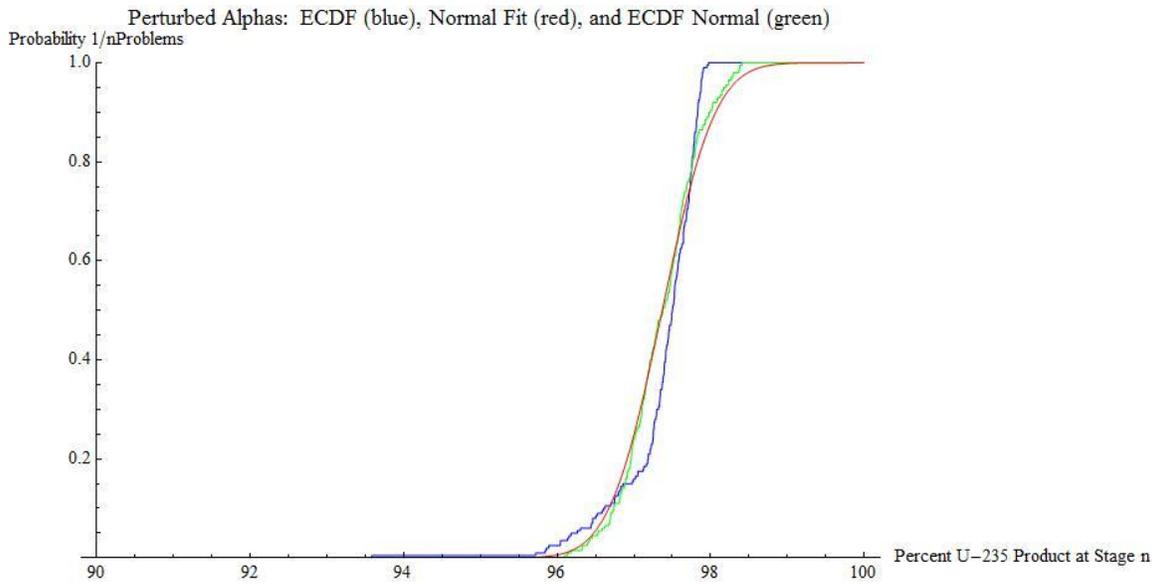
Figure 10. Perturbed Alphas for all Stages
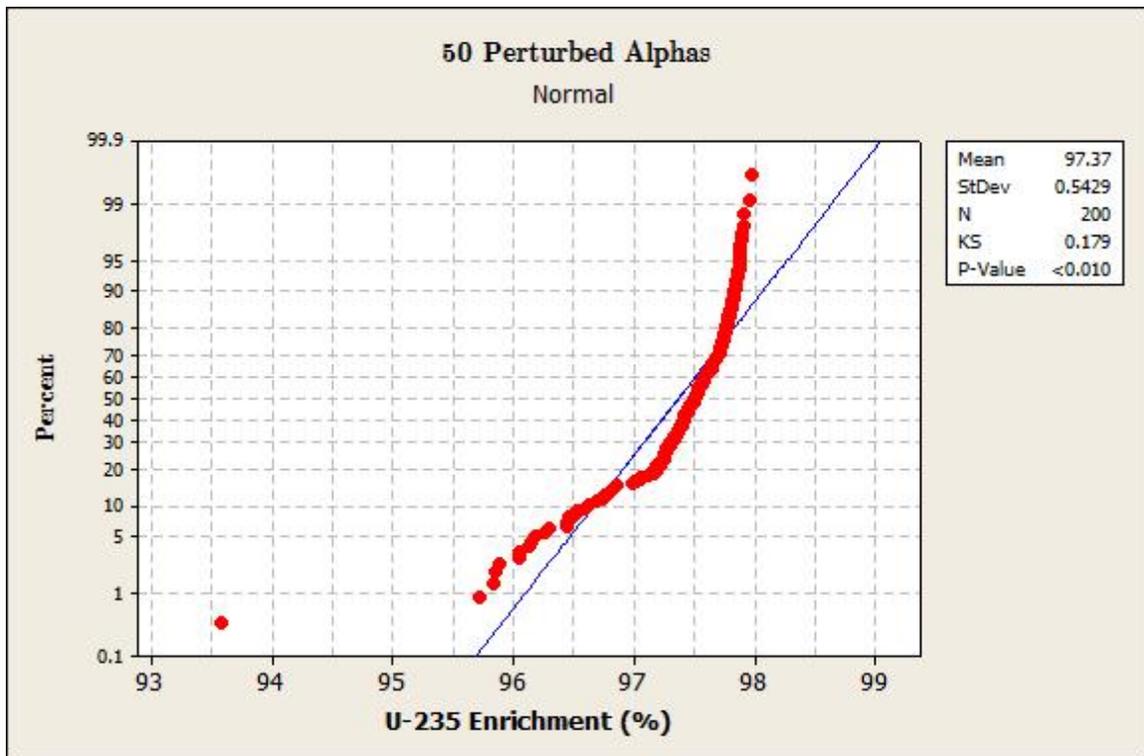
Figure 11. K-S Test for 50 Perturbed Alphas

34

Varying All Cuts and Separation Factors Together for All Stages

In the third set of problems, both cuts and separation factors were varied simultaneously for all stages. Because varying cuts alone and varying separation factors alone gave non-Gaussian outputs, varying both inputs jointly should also produce non-Gaussian outputs. Cuts and alphas were varied over the same range of values used when both input parameters were varied individually. Figure 12 is very similar to the plots generated previously when cuts and alphas were perturbed separately. Here, the KS statistic is larger than both of the KS statistics for perturbing cut and for perturbing alpha individually. The K-S test in Figure 13 also indicates that the data are not Gaussian.
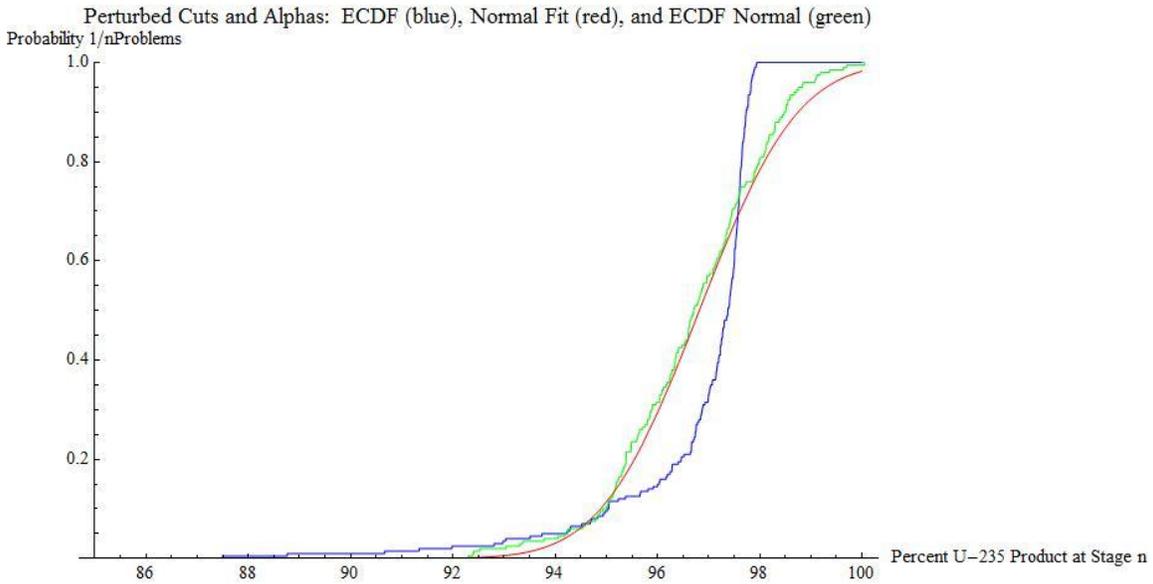
Figure 12. Perturbed Cuts and Alphas for all Stages
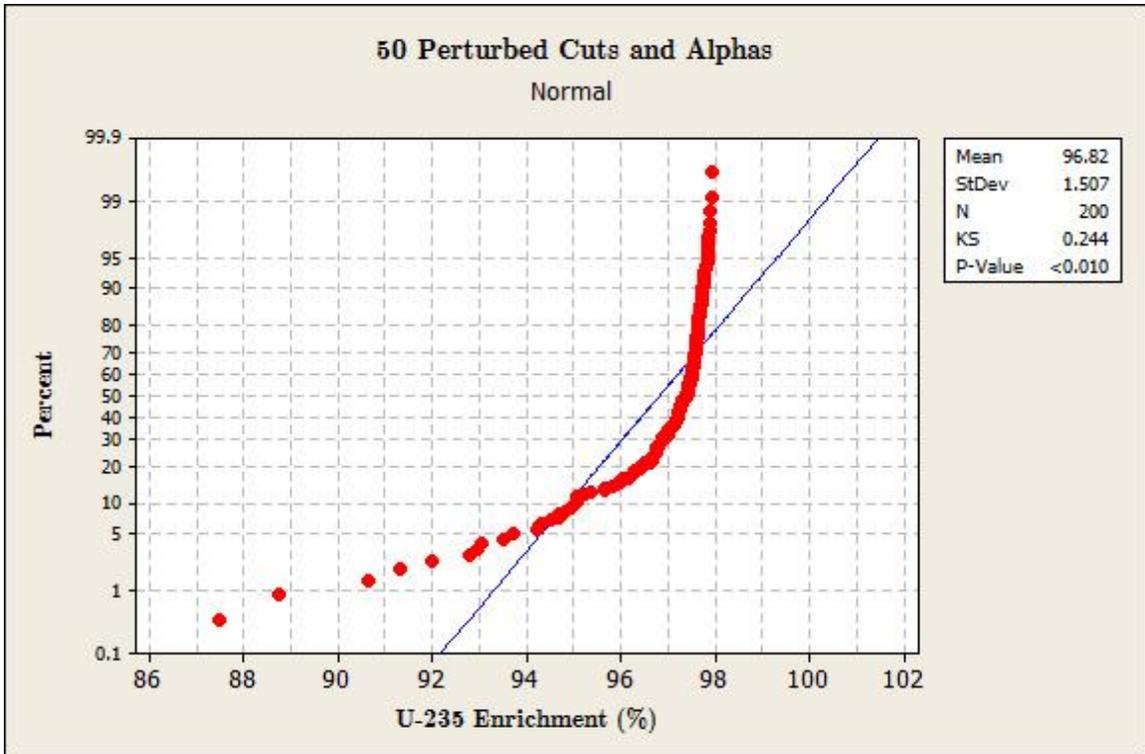
Figure 13. K-S Test for 50 Perturbed Cuts and Alphas

V.B. One Cut for All Stages

For this problem, the cut was varied between 0 and 1. The range between 0 and 1 was broken up into five intervals: 0 to 0.2, 0.2 to 0.4, 0.4 to 0.6, 0.6 to 0.8, and 0.8 to 1. For each interval, 50 random cuts were drawn over that particular interval. The 50 cuts were then used to solve 50 problems. Because each cascade problem only had one cut, all 50 stages in the cascade were given the same cut.

Figure 14 shows the percentage of enriched ^{235}U as a function of cut. The x-axis is the cuts while the y-axis is the percentage of ^{235}U product at stage *50*. For low cut values, ^{235}U does a fairly decent job of enrichment. When the cut starts to approach 0.5, ^{235}U peaks before dropping off steeply. Beyond 0.5 for cuts, ^{235}U hardly enriches at all and peaks at 0.7%.

36
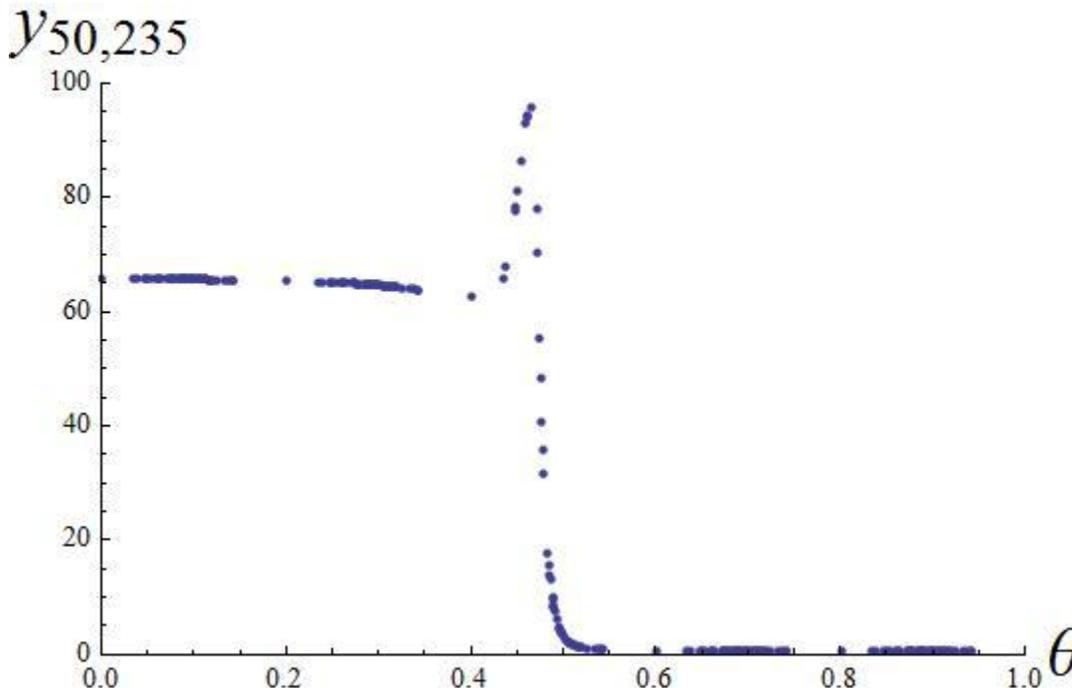
Figure 14. ^{235}U **Product as a Function of Cut**

V.C. Varying Cut at One Stage Only

For the next set of problems, the cut was varied at only one stage while the other 49 cuts remained constant. Varying the cut for one stage only could give some insight into whether the cut is more sensitive at one stage of the enrichment cascade than another. Because a cascade has an enriching section and a stripping section that perform different functions, some variation in enriched ^{235}U product should be seen.

All cuts were initially set to 0.47. The chosen stage to be perturbed was then given a new cut that ranged between 0 and 1. The actual range generated by the random number generator fell approximately between 0.169 and 0.775. The stages chosen to receive the perturbed cut include stage *1*, stage *50*, the stage where feed is fed into the cascade, a stage in the middle of the cascade, a stage between the first stage and the feed stage, and a stage between the feed stage and stage *50*.

A cut of 0.47 at all stages of this particular cascade allows ^{235}U to enrich to a fairly high percentage. By introducing a fairly large perturbation at one stage, the enriched ^{235}U output data could indicate whether the other 49 stages could compensate for this one large deviation. In addition, the ^{235}U output data could indicate whether the perturbation was more sensitive at one stage than another.

Vary Cut at Stage 1

Stage *1* is the first stage in the cascade and part of the stripping section of the cascade. This stage is where the depleted stream leaves the cascade as waste. Figure 15 shows no significant effect on ^{235}U enrichment from perturbing the cut for this stage. This is not altogether surprising because stage *1* should have very little ^{235}U given that the primary purpose of the stripping section is to remove as much ^{235}U as possible to send up to the enriching section. Figure 15 also shows that the outputs for all 200 problems display a non-Gaussian distribution. Likewise, Figure 16 clearly does not show a plot of a straight line for the empirical data. The *KS* statistic is larger than the decision value, and the p-value is smaller than the α-level. As a result, the conclusion is that the output is not normally distributed. This result is also not unexpected because previous examples of varying cuts at all stages gave non-Gaussian outputs.
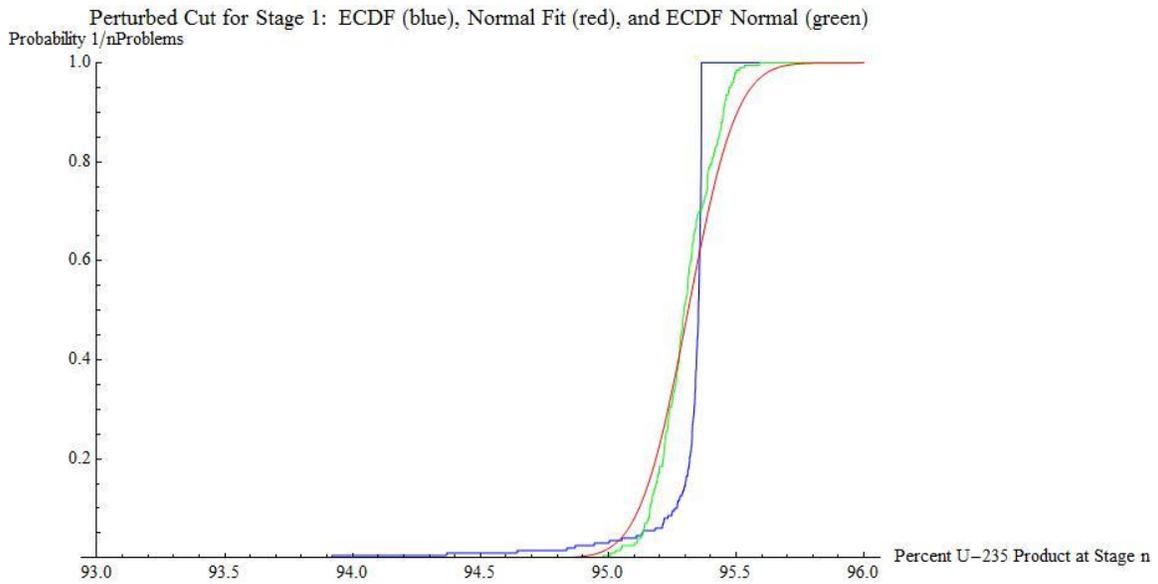
Perturbed Cut for Stage 1: ECDF (blue), Normal Fit (red), and ECDF Normal (green)

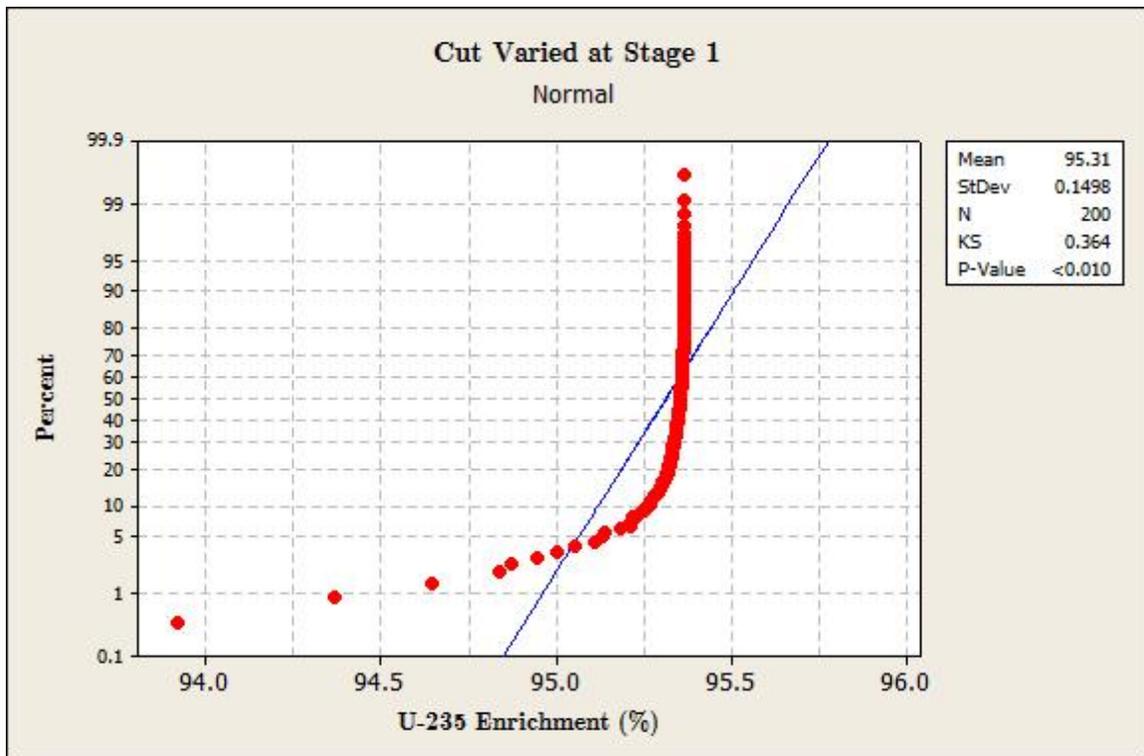**Figure 15. Perturbed Cut at Stage 1**

Figure 16. K-S Test for Varying Cut at Stage 1

39

Figure 17 shows a plot of the enriched ^{235}U product at stage *50* as a function of cut. Even for a wide range of cuts, the final ^{235}U product did not vary by much, less than 2%. The smallest ^{235}U percentage is 93.9% and the largest is about 95.4%. Varying the cut at the first stage really has no significant impact on the final amount of enriched $_{235}$U.

Cut Varied at Stage 1
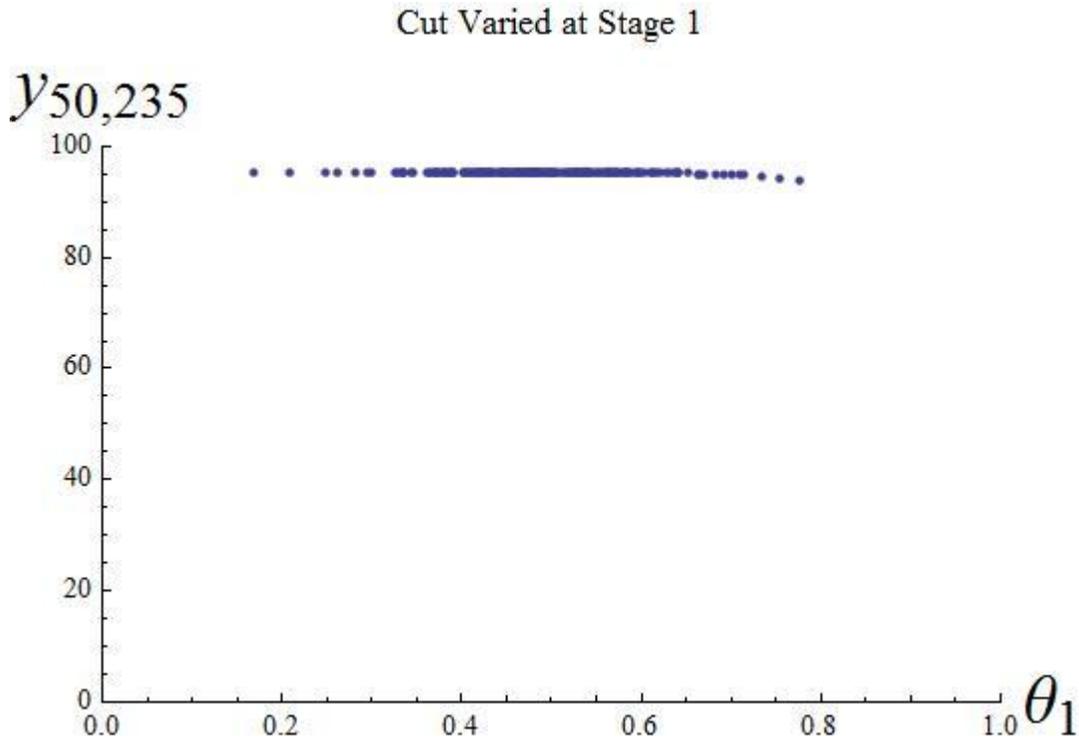
Figure 17. Enriched ^{235}U Product as a Function of Cut for Stage 1

Vary Cut at Stage 25

Stage 25 is in the enriching section of the cascade, so varying the cut at this stage may have a significant effect on the output of enriched ^{235}U. Indeed, Figure 18 shows that the enriched ^{235}U outputs have a wide variance with percentages in the thirties to high nineties. The output shows more non-Gaussian behavior than previous plots. The K-S test in Figure 19 confirms this result.

Perturbed Cut for Stage 25: ECDF (blue), Normal Fit (red), and ECDF Normal (green)
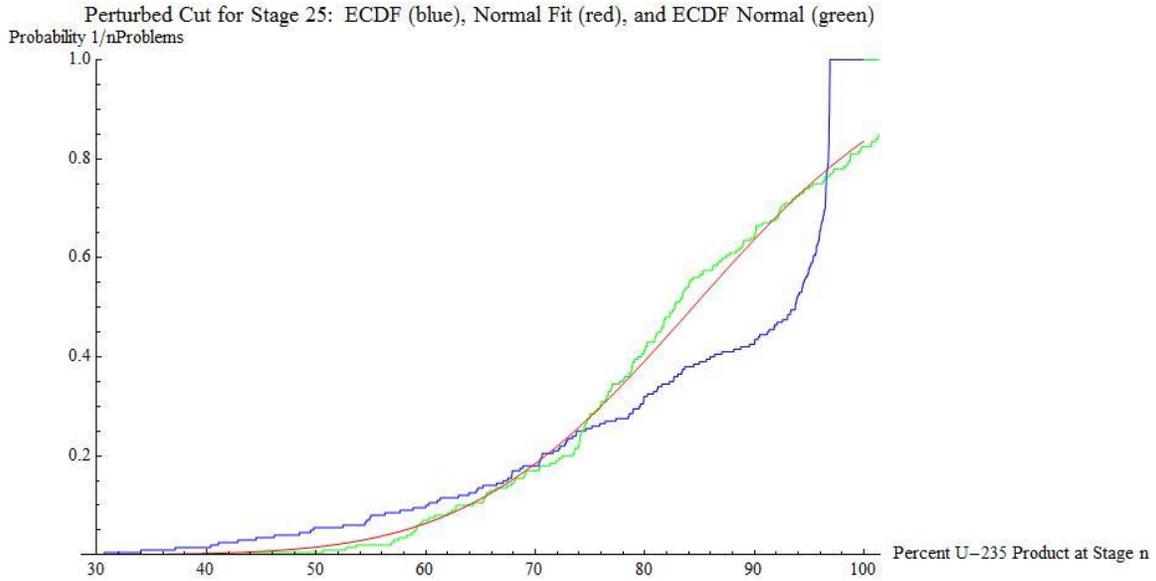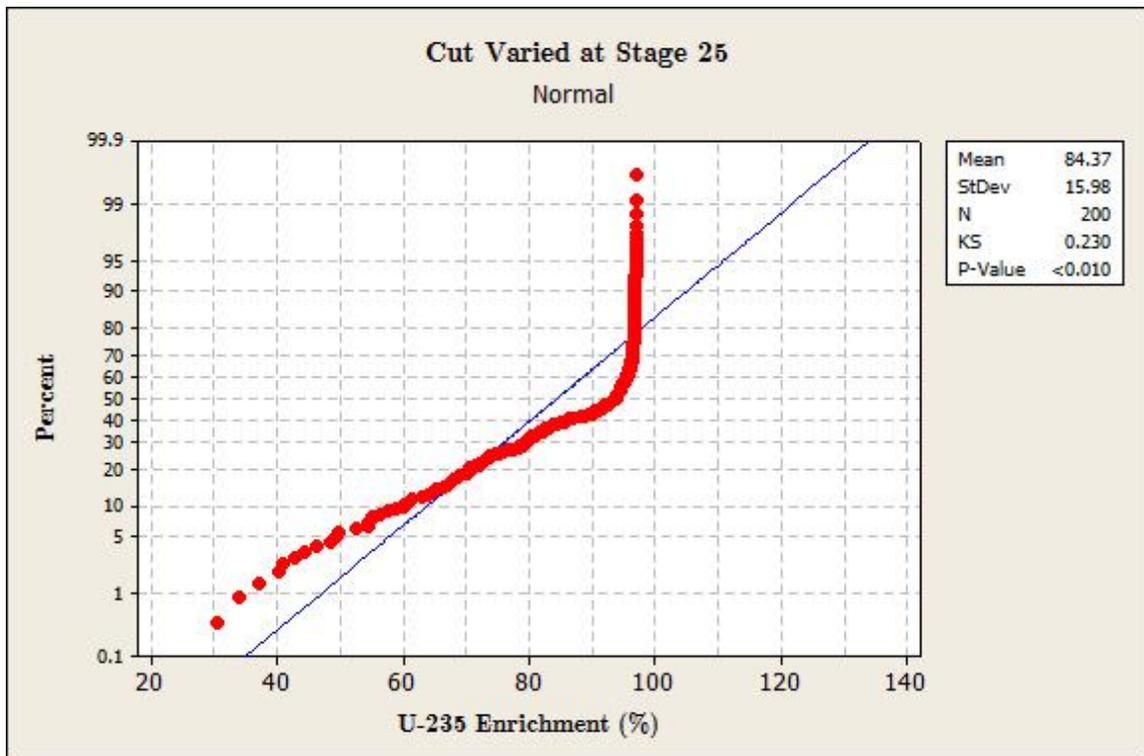

Figure 18. Perturbed Cut at Stage 25

Figure 19. K-S Test for Varying Cut at Stage 25

41

Figure 20 shows a much greater change in the enriched ^{235}U product when the cut was varied at stage *25*, the middle of the cascade. The enriched ^{235}U product peaks at approximately 97% and dips as low as 30%. The cascade is a lot more sensitive to the cut being perturbed at this stage. As the cut increases beyond 0.5, the amount of enriched ^{235}U product drops off dramatically.
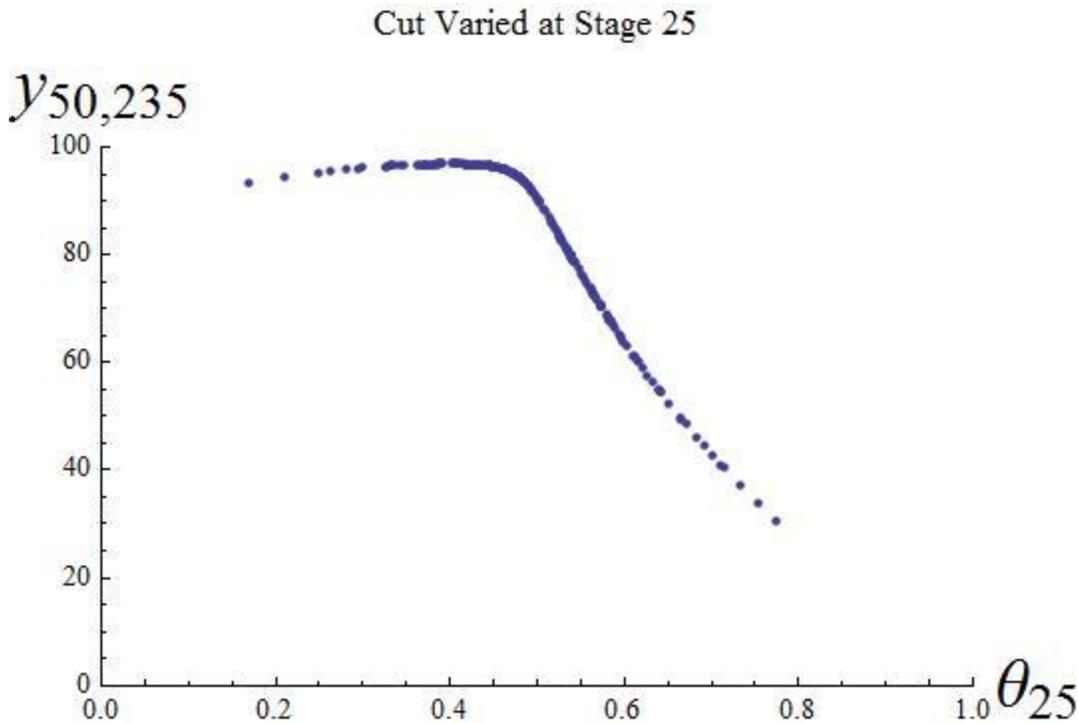
Figure 20. Enriched ^{235}U Product as a Function of Cut for Stage 25

Vary Cut at Stage 50

Stage *50* is the last stage in the cascade. The enriched stream that leaves this stage is the product. Figure 21 shows that varying the cut at this stage produces little effect on the amount of enriched ^{235}U. This is unsurprising because ^{235}U would be expected to have enriched as much as possible already by the final stage. What is surprising, up to this point, is that the uncertainty output data seem to indicate a potential Gaussian distribution by how closely the empirical data seems to match the red normal distribution plot. However, there is still a

good amount of deviation between the data points of the *ECDF* plot and the *ECDF Normal* plot which is what the empirical data should match if the distribution were normal.
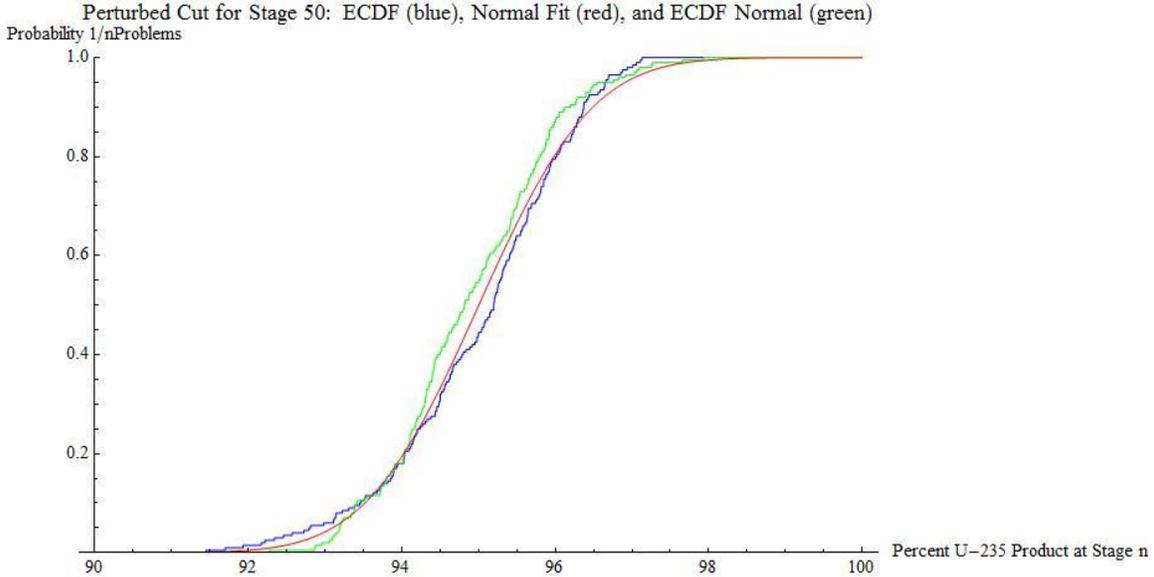

Figure 21. Perturbed Cut at Stage 50

Looking at the results in Figure 22, the majority of the data appears linear except for the tails. The *KS* statistic is quite small. In fact, this is the smallest test statistic seen yet of all the runs. The *KS* statistic of 0.076 is smaller than the decision value of 0.115. This result would mean that H_0 cannot be rejected at the 0.01-level. If the α-level is changed to 0.2, then according to Massey [8:70], the equation for the decision value becomes

$$decision\ value = \frac{1.07}{\sqrt{N}}. \tag{24}$$

This gives a decision value of 0.07566. So, at the 0.2-level, the test statistic is larger than the decision value, and the null hypothesis can be rejected. However, there is now a 20% of having committed a Type I error by concluding that the data are not normally distributed.
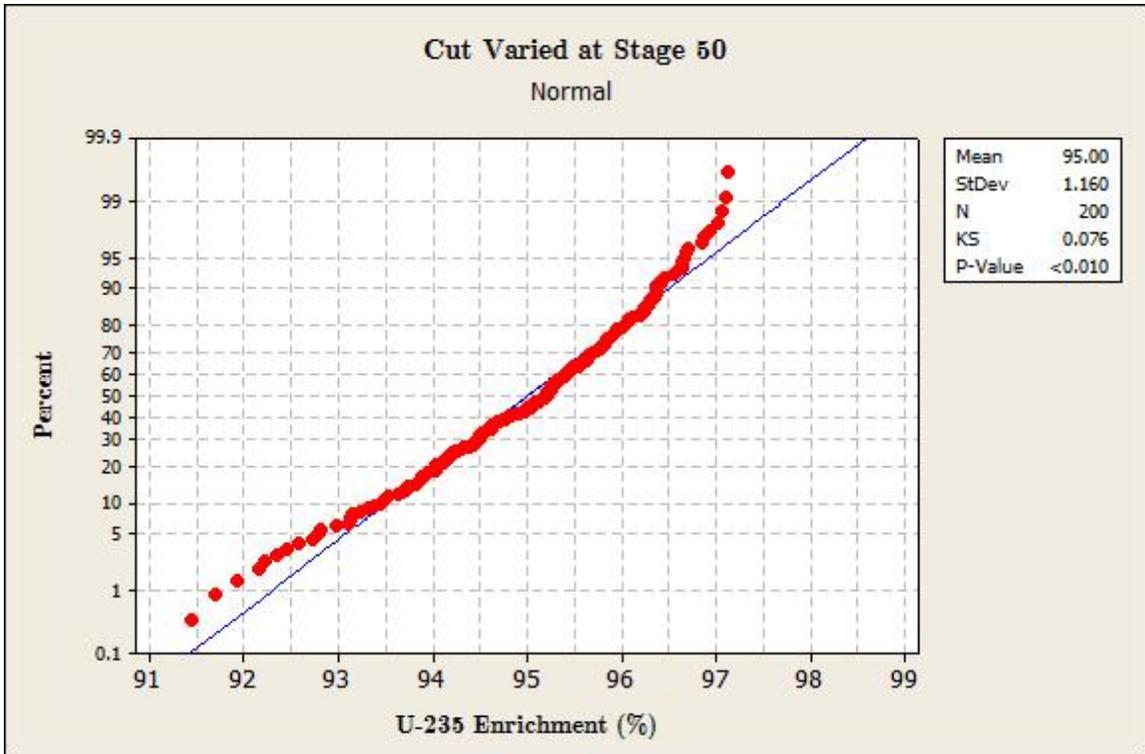
43

Figure 22. K-S Test for Varying Cut at Stage 50

Much like perturbing the cut at the first stage, Figure 23 shows that perturbing the cut at stage *50* has no significant effect on the final enriched ^{235}U product. The enriched ^{235}U product varied by 3% at most, between 93% and 96%.

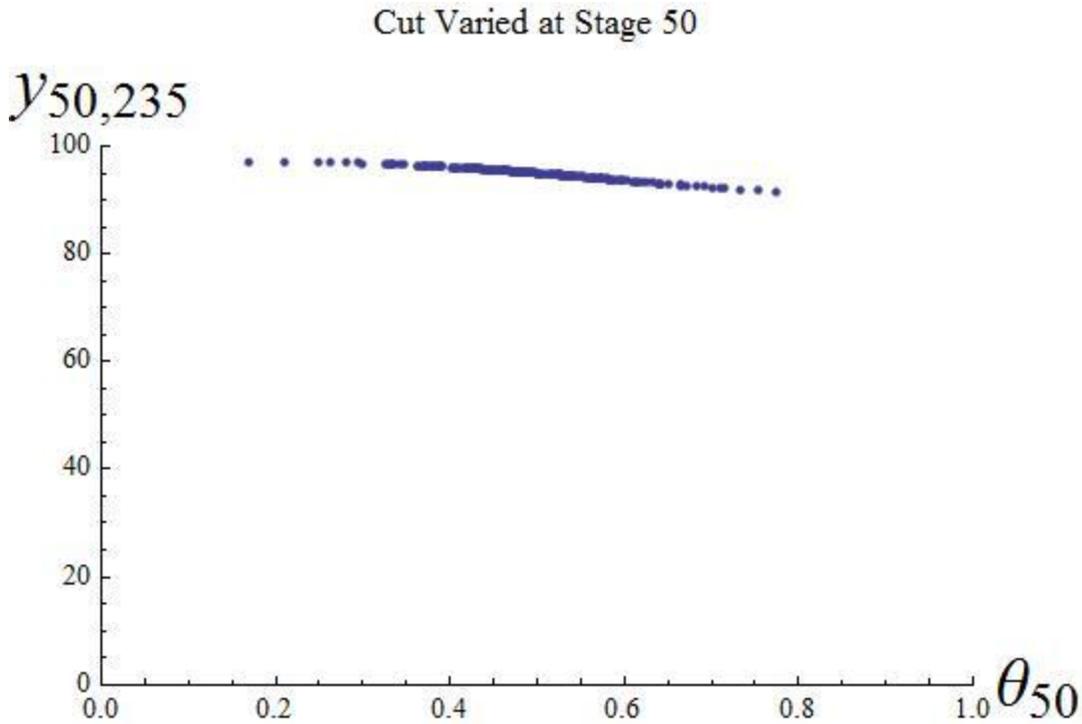Figure 23. Enriched ^{235}U Product as a Function of Cut for Stage 50

Vary Cut at the Feed Stage (Stage 10)

Stage *10* is where feed is put into the cascade normally. Figure 24 is very similar to Figure 18 in that varying the cut at this stage causes the enriched ^{235}U output data to change dramatically. The same non-Gaussian behavior is also seen. Figure 25 clearly shows a plot that is not Gaussian. The KS statistic of 0.227 confirms Figure 24.
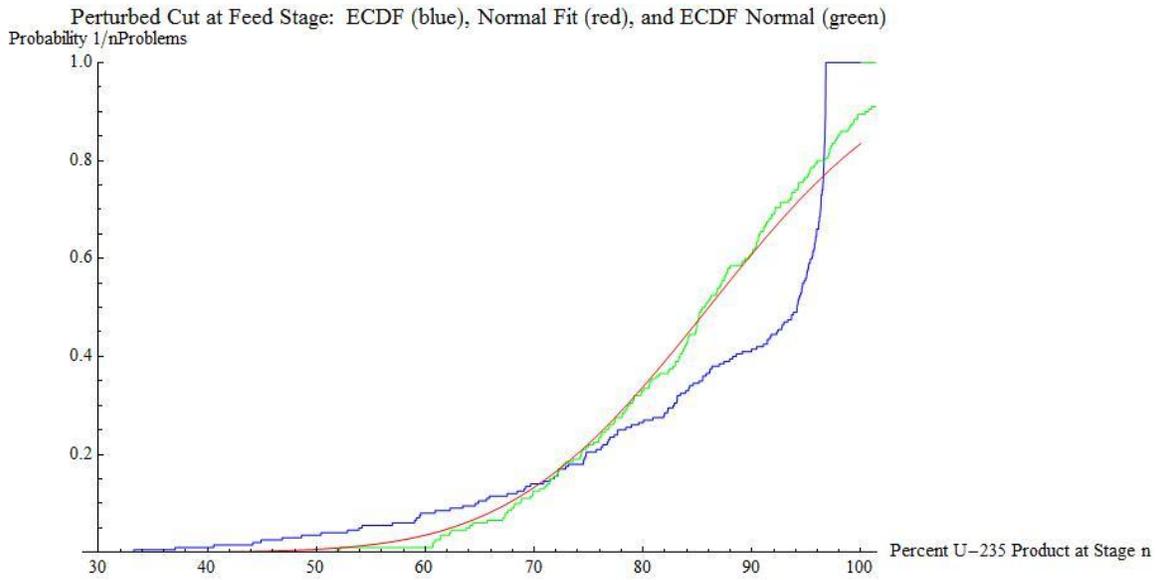
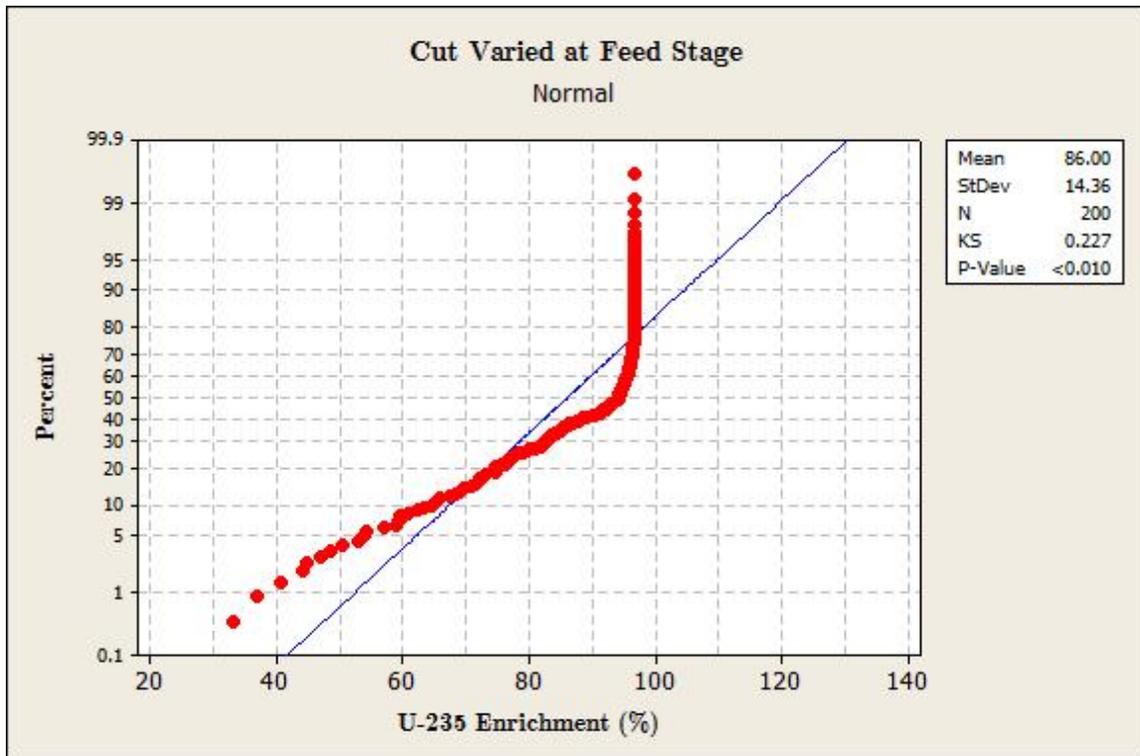**Figure 24. Perturbed Cut at Feed Stage**

Figure 25. K-S Test for Varying Cut at Feed Stage

46

The plot in Figure 26 is very similar to Figure 20. The same trend is seen where increasing the cut up to 0.5 allows the enriched ^{235}U product to peak before decreasing quickly as the cut increases. The enriched ^{235}U product peaks at approximately 97% but goes as low as 33%.
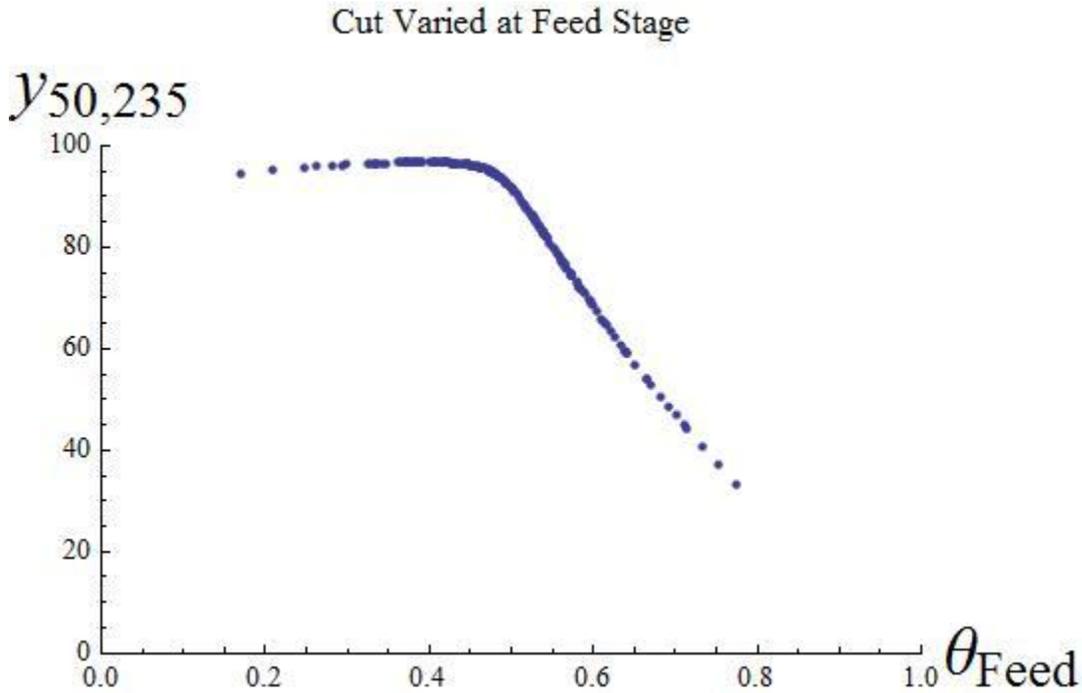

Figure 26. Enriched ^{235}U Product as a Function of Cut for Feed Stage

Vary Cut at Stage 5

Stage *5* was chosen as another test stage to vary the cut because it lies between the first stage and the feed stage. Figure 27 clearly shows the effect on ^{235}U enrichment. For some problems out of the 200 problems, the enrichment percentage ends in the sixties although there is a wide range all the way up to the nineties. There also is a wide deviation from the Gaussian curve. This result is echoed in Figure 28 with a large *KS* statistic of 0.283. The distribution of the data is not Gaussian.
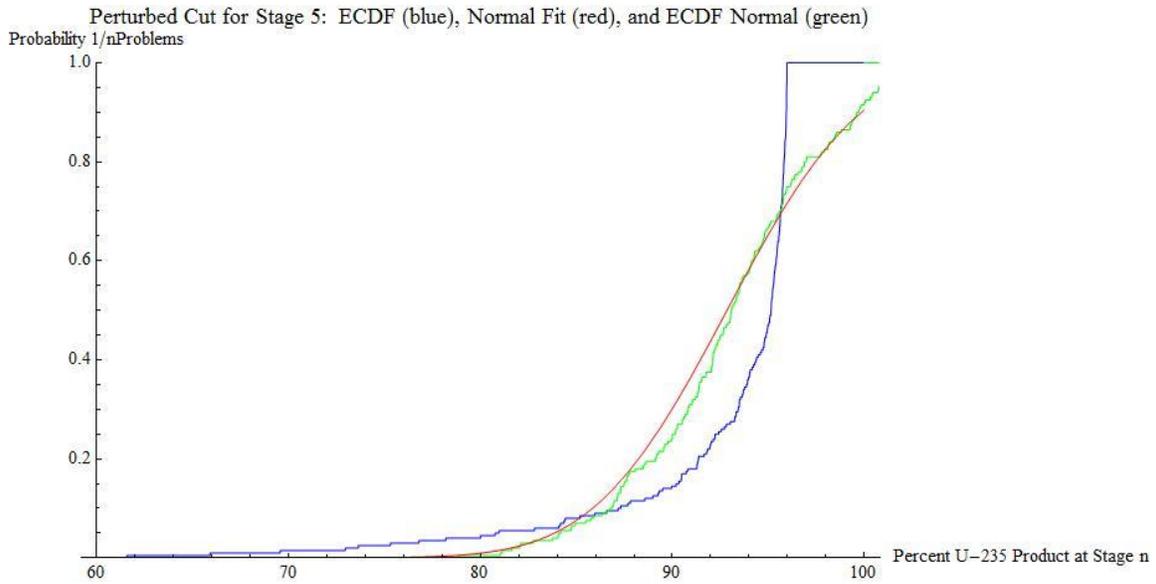
Perturbed Cut for Stage 5: ECDF (blue), Normal Fit (red), and ECDF Normal (green)
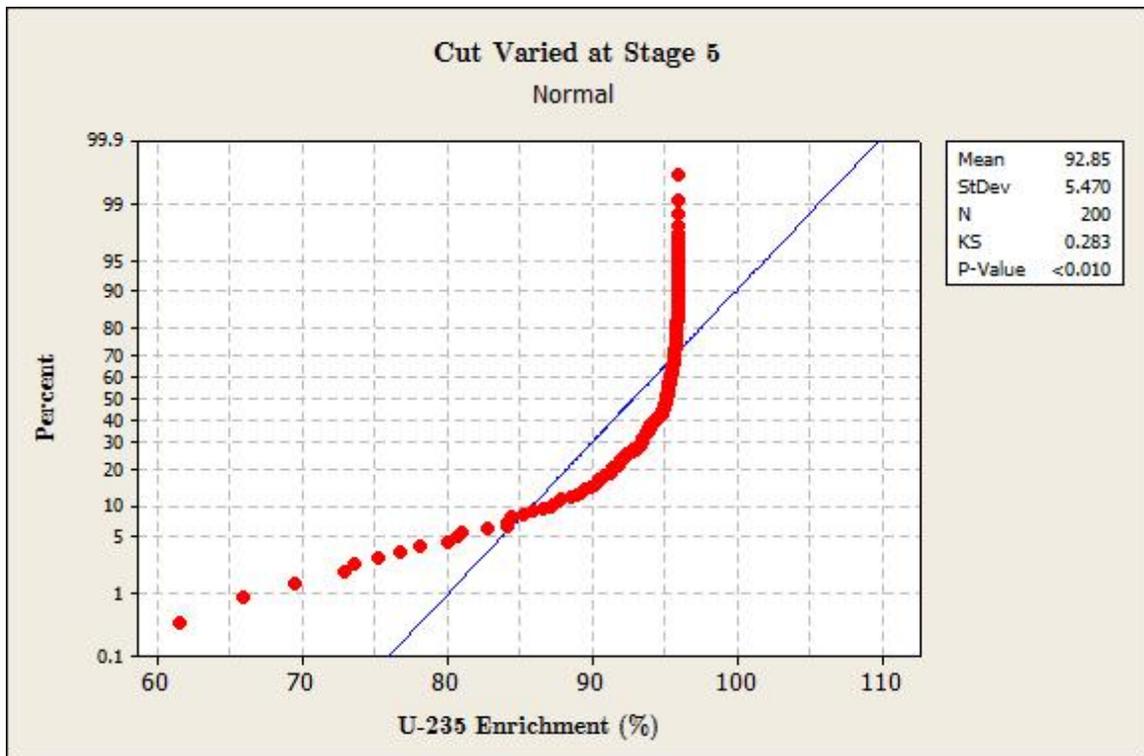

Figure 27. Perturbed Cut at Stage 5

Figure 28. K-S Test for Varying Cut at Stage 5

48

Figure 29 shows the effects of perturbing the cut at stage *5*, in between stage *1* and the feed stage. The enriched ^{235}U product stays fairly constant between cuts of 0.2 and 0.55, varying at most by 5% between 91% and 96%. Beyond cuts of 0.55, the enriched ^{235}U product drops as low as 61.5%
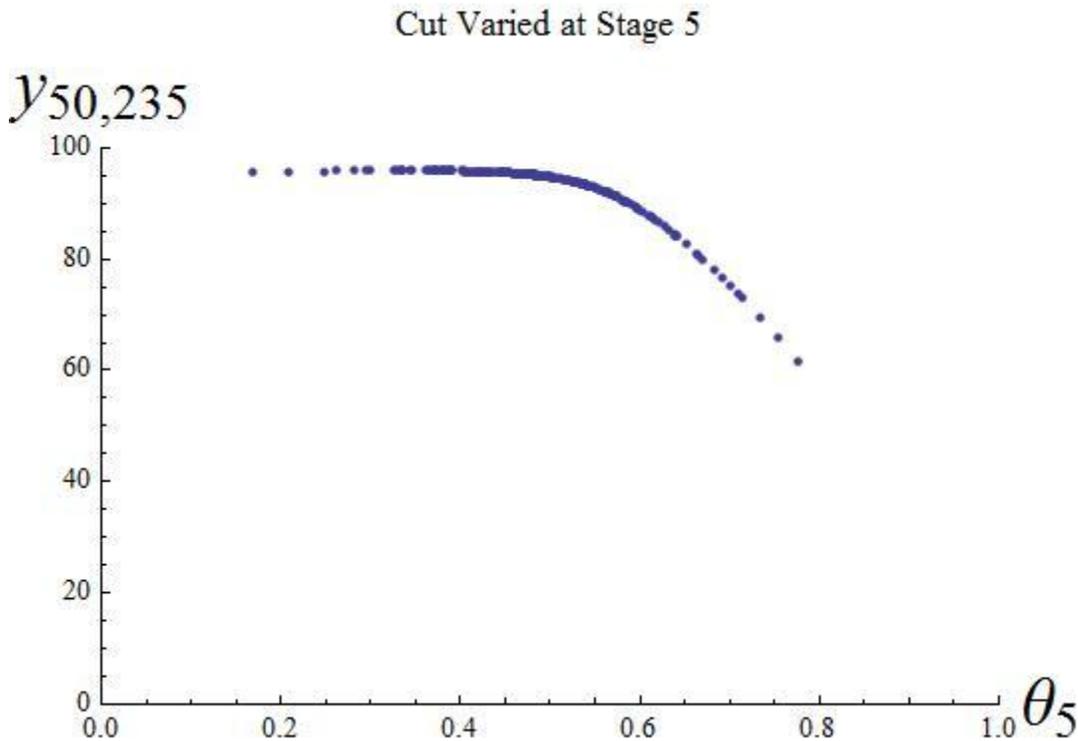

Figure 29. Enriched ^{235}U Product as a Function of Cut for Stage 5

Vary Cut at Stage 30

The last stage to be perturbed was stage *30*. It was chosen as a midpoint between the feed stage, stage *10*, and the last stage or stage *50*. This plot is not significantly different than the plots shown in Figure 18 and Figure 24. The percentages of ^{235}U enrichment here are very similar to those two plots. The similarities indicate that the cascade is most sensitive to a change in cut starting at the feed stage and above. In other words, the variation in cut is more significant in the feed stage and enriching stages than it is for the stripping stages in regard to the enriched ^{235}U product percentage. Figure 30 shows the results of varying cut at this stage, a non-Gaussian. The K-S test in Figure 31

comes to the same conclusion with a *KS* statistic of 0.23 which is larger than the decision value.
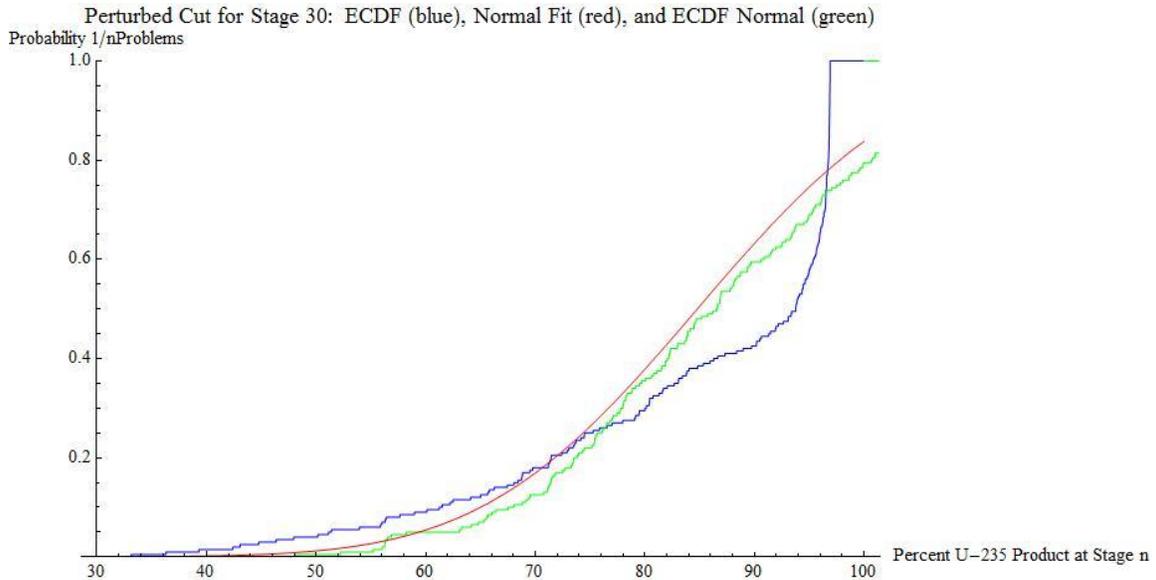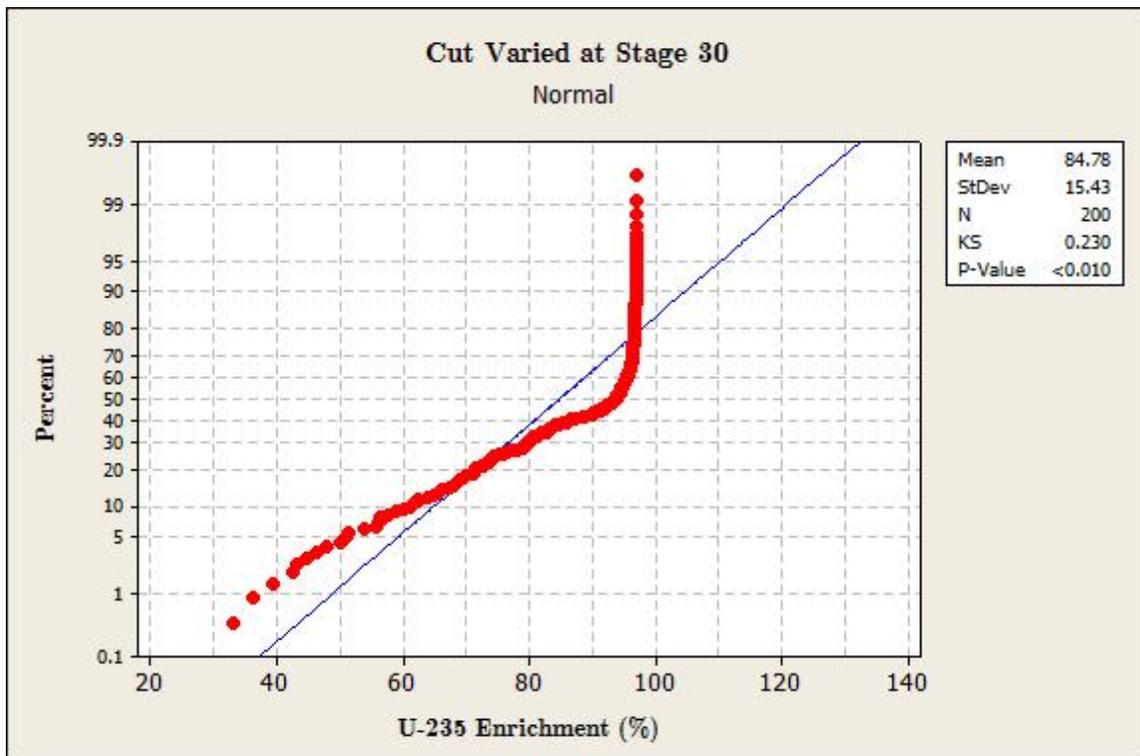

Figure 30. Perturbed Cut at Stage 30

Figure 31. K-S Test for Varying Cut at Stage 30

Figure 32 displays a plot that is very similar to the plots of Figure 20 and Figure 26. Stage *30* sits is in the middle of the feed stage and stage *50*. Figure 20, Figure 26, and Figure 32 indicate that there exist a range of stages between 10 and 30 where perturbing the cut at any one of these stages would not change the distribution of enriched ^{235}U product.

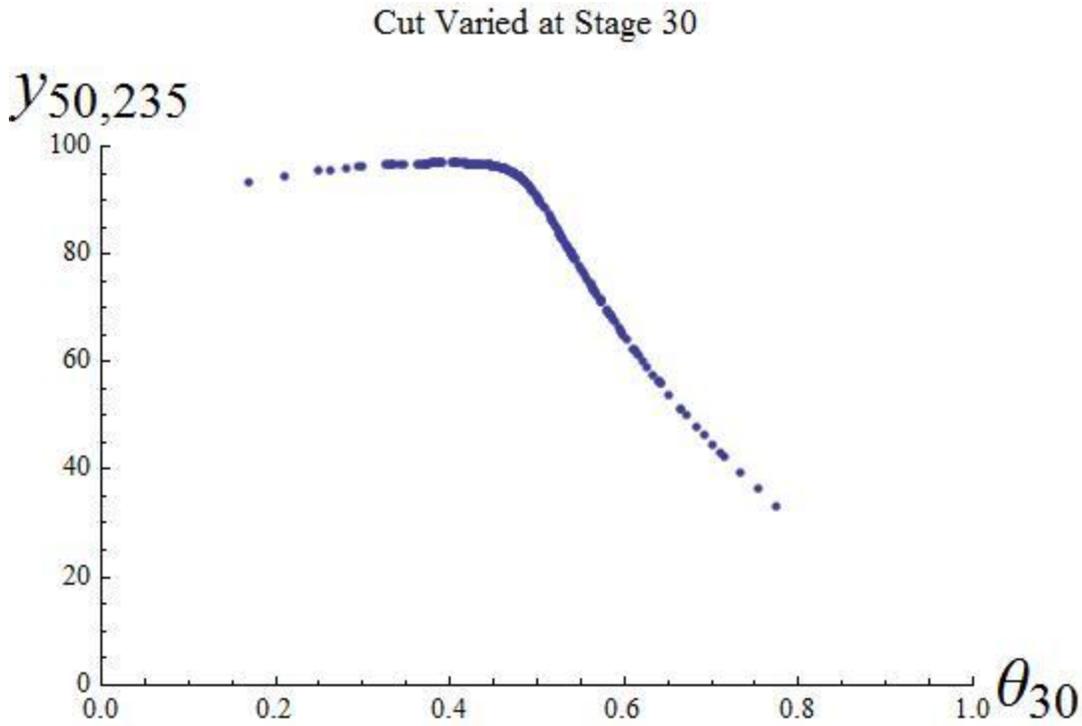Figure 32. Enriched ^{235}U Product as a Function of Cut for Stage 30

VI. SUMMARY

The purpose of this research was to develop a Monte Carlo UQ capability and to implement the capability in code. The UQ tool was built and integrated with the cascade code to handle the propagation of errors for enrichment cascades. Additionally, the UQ capability was demonstrated on a variety of multi-component cascade problems. Individual input parameters as well as combinations of parameters were varied to look at the effects on the percentage of enriched ^{235}U product. Inputs that were perturbed with Gaussian errors were shown to give non-Gaussian errors in the ^{235}U output data. While this result was not altogether surprising, the importance of this result is that the common assumption of normally distributed errors may not always be valid.

It bears emphasizing that the test cases in this research were somewhat limiting in that the cascades chosen to be studied were small, the perturbations to the inputs were large, and only the ^{235}U fraction was analyzed in the final stage of the cascade. Analyzing ^{235}U at different points along the cascade such as the middle, looking at other component fractions, making the perturbations small instead of large, or varying the cascade parameters to enrich ^{235}U to 50% or less may give much different results. As a consequence, the results of non-Gaussian behavior in the output data should not automatically be extended to all cases. Many more test cases can be done. Gathering more data with the UQ tool will provide greater understanding of how enrichment cascades operate with different inputs.

VI.A. Achievement of Objectives

The objectives of this research were to verify the cascade code and to build and demonstrate an UQ capability to handle propagation of errors. All the objectives were achieved. However, much more work can be done to improve upon the UQ capability and to use the tool to conduct a deeper statistical analysis of all uncertainties of an enrichment cascade.

Verify Cascade Code for Two and Multiple Components: Verify the accuracy of the cascade code for a two-component ideal cascade and for multi-component cascades.

The cascade code was verified for the two-component ideal cascade by calculating ^{235}U inlet concentrations that matched the closed-form solution to at least 10 decimal places for every stage of an enrichment cascade. The largest SRD calculated was 2.56×10^{-11} and the average SRD was 7.97×10^{-12}. For this level of accuracy, plots of both the cascade solution and the closed-form solution would appear identical.

The cascade code was also verified for multi-component problems through several checks. The first check ensured that the solution generated was correct to a rigorous, preset tolerance. This was accomplished internally within the code by checking for balance between the left-hand side and the right-hand side of each of the cascade equations for every stage of the cascade. None of the problems tested triggered any balance violations. The second check looked at the precision of the solution. Two runs were conducted with each one set at a different precision. The largest SRD and the average SRD computed were on the order of 10^{-14} and 10^{-15}, respectively. This demonstrated that the code can consistently compute solutions that differ by a negligible amount regardless of the precision used. The third and final check assessed whether the solution generated was reasonable. The problems tested showed the percentage of ^{235}U increasing and the percentage of ^{238}U decreasing as the stage number increased. This is exactly what is supposed to happen in an enrichment cascade.

Develop and Implement Monte Carlo UQ Capability: Develop the UQ methodology and implement the capability in code.

The UQ methodology was developed and implemented in code. The code used MKL routines to generate random normal errors, added the errors to the input parameters, solved the perturbed problem, and wrote the uncertainty output data to file. The code used good modern programming practices by putting the computations into specific modules, subroutines, and functions that are portable and reusable.

Demonstrate UQ Capability: Perturb various cascade inputs for several multi-component problems.

The UQ capability was demonstrated by perturbing cuts and alphas at all stages of an enrichment cascade. Cut was also perturbed at one stage at a time. Other demonstrations of the UQ tool included varying the feed compositions, the feed stage, and the number of stages.

Characterize Uncertainty Output Data: Plot the empirical cumulative distributions and analyze the uncertainty output data to assess the propagation of uncertainties.

The uncertainty output data were plotted using the empirical cumulative distributions. For comparison, normal distributions and random normal distributions were also plotted using the same mean and standard deviation computed from the empirical data. The K-S test for normality was applied, and the conclusion drawn was that Gaussian inputs in a nonlinear system of equations resulted in non-Gaussian outputs.

VI.B. Observations and Conclusions

One observation of interest is that the cascade code suffered from slow convergence, particularly for two-component ideal cascade problems. Sample problems tested required thousands of iterations to converge to a solution with a tolerance of 10^{-12}. If UQ were done for these types of problems, computer run-time could be an issue for large problem sets. Other than verification of the code, only multi-component problems were used in this research, so slow convergence was not an issue. The multi-component problems were solved in fewer than 200 iterations. A multi-component problem could conceivably exist that would require thousands of iterations to reach a solution.

A second noteworthy observation is that while the cascade code was used to solve uranium enrichment problems, the code can handle all elements with multiple components. This flexibility is crucial in order to apply the code to real-world applications where the input parameters may take on a variety of values.

A third observation is that a normality test should always be conducted if data are thought be normally distributed. The assumption of normality is fairly common in classical statistical testing. To determine whether this assumption is valid, a K-S test or other suitable test should be done.

A fourth observation is that well-defined problems are necessary to generate meaningful data. Input parameters can be randomly chosen in such a fashion that the cascade code does not produce any data of interest. That is, poorly chosen inputs can result in outputs where the desired isotope does not enrich to any significant amounts. Time spent on creating reasonable problems would pay off with useful output data.

The cascade code and UQ capability are a useful tool for exploring the behavior of various enrichment cascades. The examples used in this research are only a small demonstration of the types of questions that the UQ tool can begin to answer. The UQ capability has much more utility than what was addressed in this thesis because the main objective was only to build and demonstrate a tool for propagation of errors. Additional utility of this capability will depend on what aspects of enrichment the user will want to explore and the types of questions that are asked.

This research also helped to quantify a level of uncertainty that may be beneficial to people who build enrichment plants or work on nuclear reactors. If the limitations of cascade performance are known beforehand, designers can be cognizant of these uncertainties and incorporate them into the planning stage long before actual construction begins. Knowing ahead of time what performance parameters a plant will not exceed will help save time and funding. In addition, this knowledge would encourage designers not to attempt to create the perfect design on paper that seeks to remove all uncertainties when such a design construct cannot be reasonably achieved. Instead, designers can focus on working with the existing uncertainties and improve along the way.

REFERENCES

1. Acton, Forman S. *Real Computing Made Real.* Princeton, NJ: Princeton University Press, 1996.

2. Benedict, Manson, Thomas H. Pigford, and Hans Wolfgang Levi. *Nuclear Chemical Engineering (Second Edition).* New York: McGraw-Hill Inc., 1981.

3. Burden, Richard L. and J. Douglas Faires. *Numerical Analysis (Eighth Edition).* Pacific Grove, CA: Brooks/Cole, 2005.

4. Chapman, Stephen J. *Fortran 95/2003 for Scientists and Engineers (Third Edition).* New York: McGraw-Hill Inc., 2008.

5. Halle, E. Von. "Multicomponent Isotope Separation in Matched Abundance Ratio Cascades Composed of Stages with Large Separation Factors". Presented at the *Workshop on Separation Phenomena in Liquids and Gases*, Technische Hochschule, Darmstadt, Germany: July 20-23, 1987.

6. Intel MKL Version 10.2.2.025.

7. Intel Visual Fortran Compiler Version 11.1.048.

8. Massey, F. J., Jr. "The Kolmogorov-Smirnov Test of Goodness of Fit". *Journal of the American Statistical Association.* Vol. 46(253): 68-78. 1951.

9. Mathews, Kirk. Private communication. 2009.

10. Miller, Ty E. *Modeling Multiple Component Uranium Enrichment Cascades.* MS thesis, AFIT/GNE/ENP/04-04. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 2004.

11. Minitab 15 Statistical Software English, 2009.

12. Murphy, Michael J. *Modeling Four-Component Uranium Gaseous Diffusion Enrichment Cascades.* MS thesis, AFIT/GAP/ENP/99J-01. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, June 1999 (0726028).

13. Olander, Donald R. "Two-Up, One-Down Ideal Cascades for Isotope Separation". *Nuclear Technology.* Vol. 29: 108-112. October 12, 1975.

14. Yamamoto, Ichiro and Akira Kanagawa. "Multicomponent Isotope Separating Cascade Composed of Elements with Large Separation Factors". *Journal of Nuclear Science and Technology*, 15[8]: 580-584. August 1978.

APPENDIX A

DERIVATIONS OF EQUATIONS

The closed-form solution used to verify the cascade code for two-component ideal cascades is derived from equations *(12.85)* and *(12.96)* given in Benedict, Pigford, and Levi [2:659-660]. The equation for the abundance ratio in the product for each stage is given by

$$\eta_i = \beta \zeta_i$$
$$\text{for } i = 1,...,n \tag{25}$$

where β is the heads separation factor and ς_i is the abundance ratio in the feed at stage i. If F is the stage number where the cascade receives feed, then

$$\eta_F = \beta \zeta_F. \tag{26}$$

The abundance ratio in the product for stage F is also given by

$$\eta_F = \beta^{F-i} \eta_i.$$
$$\text{for } i = 1,\ldots,n \tag{27}$$

Because equations (20) and (21) are equivalent,

$$\beta \zeta_F = \beta^{F-i} \eta_i = \beta^{F-i} \left(\beta \zeta_i \right). \tag{28}$$

When β is cancelled on both sides, the equation becomes

$$\zeta_F = \beta^{F-i} \zeta_i. \tag{29}$$

The abundance ratio in the feed is given by

$$\zeta_F = \frac{z_F}{1 - z_F} \tag{30}$$

and

$$\zeta_i = \frac{z_i}{1 - z_i} \tag{31}$$

for stages F and i, respectively, where z is the fraction of the lighter component. Substituting equations (30) and (31) into equation (29) gives

$$\frac{z_F}{1 - z_F} = \beta^{F-i} \frac{z_i}{1 - z_i}. \tag{32}$$

Cross multiplication gives

$$z_F - z_F z_i = \beta^{F-i} z_i - \beta^{F-i} z_i z_F. \tag{33}$$

Collect all z_i terms on one side and factor out z_i to get

$$z_i \left(z_F + \beta^{F-i} - \beta^{F-i} z_F \right) = z_F. \tag{34}$$

Solving for z_i, the equation for the lighter component at each stage simplifies to

$$z_i = \frac{z_F}{z_F + \dfrac{1}{\beta^{i-F}} - \dfrac{z_F}{\beta^{i-F}}} = \frac{z_F \beta^{i-F}}{z_F \beta^{i-F} + 1 - z_F} = \frac{z_F \beta^{i-F}}{z_F \left(\beta^{i-F} - 1 \right) + 1} = \frac{z_F \beta^{i-F}}{1 - z_F \left(1 - \beta^{i-F} \right)}. \tag{35}$$

APPENDIX B
HYPOTHESIS TESTING

Hypothesis testing is a statistical method in which the probability of a test result is computed to determine whether the result could have reasonably occurred by chance. Two hypotheses are set up, the null hypothesis and the alternate hypothesis. The null hypothesis, denoted by H_0, is the hypothesis proposed by the person conducting the testing. The alternate hypothesis, denoted by H_1, is the negation of the null hypothesis. The purpose of hypothesis testing is to determine if there is sufficient evidence to reject the null hypothesis. If there is insufficient evidence, then it is a failure to reject the null hypothesis.

The decision to reject or not reject the null hypothesis comes from computing a test statistic of the sample data and observing the likelihood of observing this statistic given the probability distribution under the null hypothesis. A decision value is computed that separates the probability distribution into two regions, a critical region and a noncritical region. The critical region is range of values in which the null hypothesis can be rejected. A test statistic that falls within the critical region indicates a low probability that the null hypothesis is true. Similarly, the noncritical region is the range of values in which the null hypothesis would not be rejected. A test statistic within the noncritical region shows a high probability that the null hypothesis is true. Thus, we would fail to reject the null hypothesis if this were the case.

The decision value in hypothesis testing is computed by determining the level of significance of the test or α-level: the probability of committing a Type I error. The α-level is the probability of falsely rejecting H_0 when it is in fact true.

There are four possible outcomes to a hypothesis test.

1. H_0 gets rejected when it is actually true. There is a risk of committing a Type I error.

2. H_0 gets rejected and it is actually false. The decision is the correct decision.

3. There was a failure to reject H_0 and it is true. This is a correct decision.

4. There was a failure to reject H_0 and it is false. There is a risk of committing a Type II error.

An alternative to hypothesis testing is significance testing. H_0 and H_1 are set up exactly as before. However, there is no need to specify the critical and noncritical regions. Instead, a probability called the p-value is computed. The p-value is the smallest significance level at which H_0 may be rejected. If the p-value is smaller than the chosen α-level, H_0 is rejected.

1. REPORT DATE (DD-MM-YYYY) 25 March 2010	2. REPORT TYPE Master's Thesis	3. DATES COVERED (From – To) June 2009 – March 2010

4. TITLE AND SUBTITLE

Uncertainty Quantification of Multi-Component Isotope-Separation Cascade Model

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Tran, Khoi D., Captain, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Building 640
WPAFB OH 45433-8865

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GNE/ENP/10-M09

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Intentionally left blank

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Monte Carlo uncertainty quantification (UQ) capability has been added to a code for modeling multi-component steady-state isotope-separation enrichment cascades to characterize the propagation of uncertainties in input data that define the cascade and the feed. Random samples of error for every computational input are drawn from its individual uncertainty distribution and added to the inputs, creating a set of enrichment cascade problems with perturbed inputs. The set of problems is solved using the verified code. The cascade outputs are then characterized using the empirical cumulative distribution. The uncertainty output data are analyzed to gain new insights into the behaviors of enrichment cascades. The UQ capability is an investigative tool that can be used to explore current and new questions of interest regarding enrichment.

15. SUBJECT TERMS

Enrichment Cascades; Isotope Separation; Computer Modeling; Uranium; Propagation of Uncertainties; Propagation of Errors; Monte Carlo

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Kirk A. Mathews, PhD, (ENP)
a. REPORT	b. ABSTRACT	c. THIS PAGE	UU	76	19b. TELEPHONE NUMBER (Include area code) (937) 255-6565, ext 4508 (kirk.mathews@afit.edu)
U	U	U			