The Dissertation Committee for Joshua Loren Peterson Certifies that this is the approved version of the following dissertation:

Expansion of Perturbation Theory Applied to Shim Rotation Automation of the Advanced Test Reactor

Committee:

Erich Schneider, Supervisor

Dave Nigg

Steven Biegalski

Mark Deinert

Graham Carey

Expansion of Perturbation Theory Applied to Shim Rotation Automation of the Advanced Test Reactor

by

Joshua Loren Peterson, B.S.; M.S.

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

August 2011

Expansion of Perturbation Theory Applied to Shim Rotation Automation of the Advanced Test Reactor

Publication No.____________

Joshua Loren Peterson, PhD
The University of Texas at Austin, 2011

Supervisor: Erich Schneider

In 2007, the Department of Energy (DOE) declared the Advanced Test Reactor (ATR) a National Scientific User Facility (NSUF). This declaration expanded the focus of the ATR to include diversified classes of academic and industrial experiments. An essential part of the new suite of more accurate and flexible codes being deployed to support the NSUF is their ability to predict reactor behavior at startup, particularly the position of the outer shim control cylinders (OSCC). The current method used for calculating the OSCC positions during a cycle startup utilizes a heuristic trial and error approach that is impractical with the computationally intensive reactor physics tools, such as NEWT. It is therefore desirable that shim rotation prediction for startup be automated. Shim rotation prediction with perturbation theory was chosen to be investigated as one method for use with startup calculation automation. A modified form of first order perturbation theory, called phase space interpolated perturbation theory, was developed to more accurately model shim rotation prediction. Shim rotation prediction is just one application for this new modified form of perturbation theory. Phase space interpolated perturbation theory can be used on any application where the range of change to the system is known a priori, but the magnitude of change is not known. A cubic regression method was also developed to automate shim rotation prediction by using only forward solutions to the transport equation.

Table of Contents

List of Tables

List of Figures

INTRODUCTION

The Advanced Test Reactor (ATR) (Figure 1), which began operation in 1967, is used by the Navy to test the effects of intense radiation on reactor material samples. In 2007 the Department of Energy (DOE) declared the ATR a national scientific user facility (NSUF) [Allen et al 2010]. This declaration expanded the focus to include diversified classes of academic and industrial experiments. The ATR has nine flux traps each with unique capabilities for high flux irradiation. The separate fuel arrangement achieves a close integration between flux traps and fuel because of the serpentine fuel arrangement (Figure 1). Five of the flux traps are almost completely surrounded by fuel and the remaining four flux traps are surrounded by fuel on three sides. The ATR design enables large power shifts among the flux traps to be obtained. Hence, one of the advantages of the ATR is the precision with which the power density can be adjusted. The ATR uses a combination of control cylinders that rotate and neck shim rods that withdraw vertically to adjust the power. The 16 control cylinders, which operate in eight pairs, are composed of beryllium and are sheathed by plates of hafnium on 120 degrees of their outer surfaces. Rotating the control cylinders away from the core will increase the reactor k-eigenvalue, whereas rotating the control cylinders towards the core will decrease the k-eigenvalue.

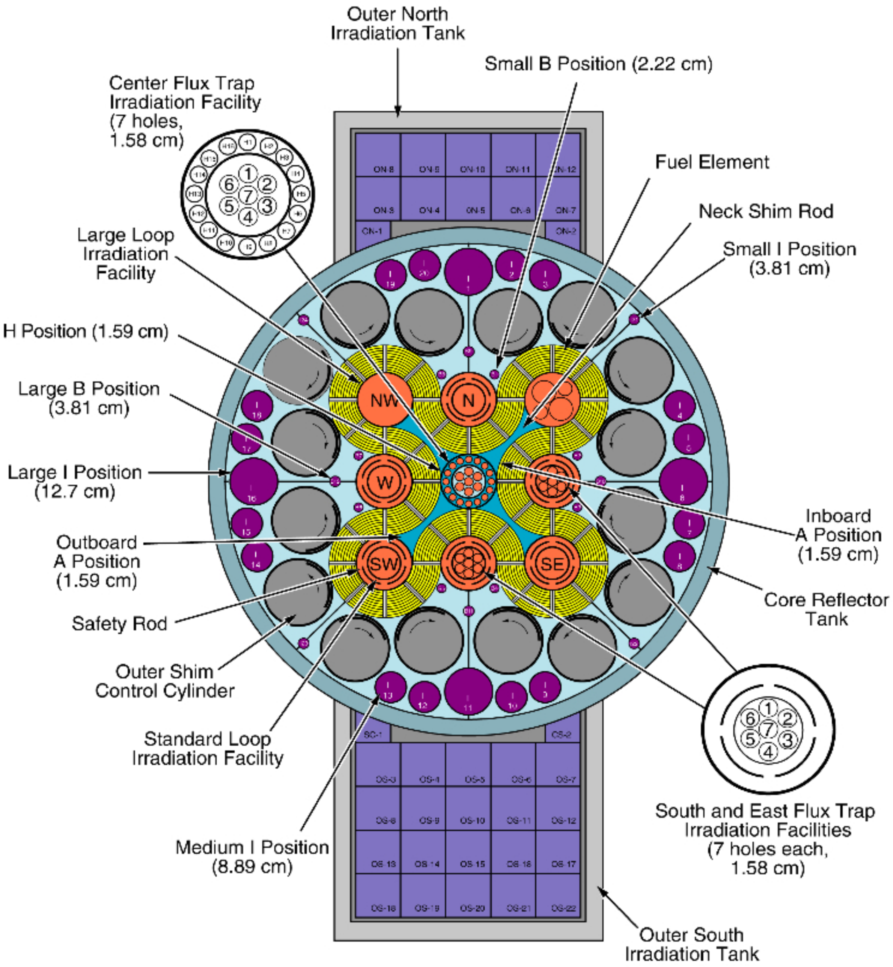


Figure 1 The ATR [Allen et al 2010]

Idaho National Lab (INL) is in the process of upgrading their reactor safety analysis to include more accurate and flexible models of the ATR. INL is currently using a two dimensional four-energy group diffusion code (PDQWS) which is strongly dependent on custom-built cross section sets and has no mesh refinement capability. NEWT, a two-dimensional discrete ordinates (Sn) transport code, has been selected as one of the next generation safety analysis codes for modeling the ATR. Major enhancements must be made before NEWT can accurately model the ATR in all plausible configurations. One of these enhancements provides the motivation for the methods developed in this dissertation.

Problem Statement

The ATR currently uses PDQWS, an in-house code, to perform the neutronic safety analysis [SAR-153]. One of the biggest concerns with PDQWS is the limitations on modeling new shapes or materials because of its fixed grid and highly tuned cross sections. When the ATR became a national user facility, the limited ability to model new materials and shapes significantly constrained the experimental scope of the ATR. Because of the limited ability to model new materials, it is expected that PDQWS will not be able to accurately model the ATR with the new fuel composition.

It is decided that a new suite of codes will be used for the safety analysis of the ATR to replace the outdated PDQWS. One of these, NEWT, is chosen because of its advanced cross section generating capability. NEWT is a two-dimensional discrete ordinates code that uses arbitrary geometry. NEWT is part of the SCALE suite, which the NRC uses for confirmatory runs with commercial nuclear power plants [Dehart 2003]. NEWT, in conjunction with the SCALE suite, can calculate group cross sections, collapse cross sections to smaller groups, homogenize cross sections while conserving reaction rates, and output macroscopic or microscopic cross sections. The cross sections generated from NEWT can then be used in other codes such as Attila, a 3D discrete ordinate transport code.

An essential part of the new suite of codes for safety analysis is their ability to predict how the reactor will behave at startup. One of the main jobs of the reactor physics analyst is to provide the operators a description and associated tolerances to where the outer shim control cylinders (OSCC) should be rotated to obtain criticality with the desired spatial reactor power distribution. At startup, if the location of the shims or required power splits are not within the specified tolerance then the reactor is shut down, and reactor startup is halted until it can be determined why the model and reactor do not match.

The current method used for calculating the OSCC positions during a cycle startup utilizes a heuristic trial and error approach. This trial and error approach has worked well in the past because of the low computational time required to solve a four-energy group course grid diffusion code. However, the new reactor physics tools will require more computational time because of the large energy groups and discrete ordinates treatment of angular dependence. For example, one forward calculation in PDQWS takes around four minutes; however, one forward calculation in NEWT using 44 energy groups, S16 angular discretization, Legendre scattering expansion of order 1, and 80,000 cells takes around 24 hours on one processor. It is therefore desirable that shim rotation prediction for startup be automated.

Shim rotation prediction with perturbation theory is investigated as one method for use with startup calculation automation. However, it is expected that ordinary first order perturbation theory will fail to handle large control system movements. In this work therefore, a modified form of first order perturbation theory, called phase space interpolated perturbation theory, is developed to more accurately model shim rotation prediction. Phase space interpolated perturbation theory uses a modified phase space interpolated adjoint flux in the perturbed regions by solving the adjoint problem for a bounding scenario. The forward solution is the unperturbed or reference case whereas the adjoint, or importance function, is evaluated at an extreme value of a pre-determined domain of changes to geometry or composition. The modified energy and angular dependent flux provides a more realistic representation of the perturbed solution. However, phase space interpolated perturbation theory requires a priori knowledge of the domain of changes being modeled so that they may be bounded by the forward and adjoint solutions.

Phase space interpolated perturbation theory fundamentally improves the reactor physics modeling capability for optimization and automation calculations. For example, phase space interpolated perturbation theory can be used to help optimize BWR control rod sequencing because the movements of the control rods are known beforehand. It can also be used to approximate the reactivity effects of fuel placement in reactor models instead of using the more computational intensive higher order perturbation theory methods. The fuel optimization calculations should work with phase space interpolated perturbation theory because the range of composition of the fuel is known beforehand. Phase space interpolated perturbation theory can be applied to any neutron transport code that can solve for both the forward and adjoint problems. In this study, phase space interpolated perturbation theory capability is demonstrated for shim rotation prediction of the ATR.

In some neutron transport codes the adjoint cannot be easily obtained. These codes would therefore not be able to use perturbation theory for shim rotation automation. A cubic regression method was therefore developed to automate shim rotation prediction without the need for any adjoint solutions.

The next section is a literature review of previous and current methods used for automation of reactor physics calculations. The third chapter describes the development, validation, and implementation of perturbation theory developed specifically for use with NEWT, which has no internal perturbation theory capability. The forth chapter contains the theory, development, validation, and results from shim rotation prediction with phase space interpolated perturbation theory. The fifth chapter describes the theory, method, implementation, and results from a more standard cubic regression method for shim rotation automation. The final chapter contains the conclusion and describes future work needed for improving the ATR physics safety analysis, as well as broader applications of phase space interpolated perturbation theory.

LITERATURE REVIEW

For nearly five decades there has been an effort to automate neutronic codes for optimization and safety analysis, Stover in 1969 being one of the first papers. Most of the effort has been with the optimizations of reloading schemes for BWR, PWR, and CANDU reactors [Turinksy 2005]. Even though the automation of a commercial reactor-reloading scheme is vastly different from calculating shim rotation prediction of the ATR at startup, the theory and tools developed in the past forty years for automation of commercial reactors may be applied to the automation of the ATR shim rotation.

The purpose of the automation of commercial codes for PWR and BWR is to calculate the best fuel loading and control rod withdrawal sequences that will save the utilities the most money while still being able to do the calculations in a relatively short amount of time. However, the nuclear industry is subject to operational and safety restrictions that limit and constrain the optimization calculations. The most difficult part of the entire process is the total number of options available. With the increase of computational power and less expensive parallel computing, the nuclear industry has been able to more effectively optimize the in core fuel loading pattern while staying within the safety and operational constraints. However, such advanced methods for calculating 10^{20} to 10^{1000} possible permutations for each reactor cycle are not needed for shim rotation prediction of the ATR.

For ATR shim rotation prediction, there are only four continuous parameters with around ten constraints. The four variables for shim rotation prediction are the four groups of shim positioning. The constraints include the desired power splits within a given tolerance, the eigenvalue calculation within a given tolerance, and a maximum and minimum physical limitation of the shim positions. In the future, shim rotation automation for the ATR will be coupled with fuel loading optimization that could require a potential of around 40 factorial, or $10^{47,}$ permutations just for the placement of fuel. These future calculations could greatly benefit from more modern fuel management procedures. Until then, simpler methods for automation can be implemented for the ATR reactor physics model.

Before the commercial industry had advanced computational tools and methods, they had to rely on simplified automation and optimization procedures. One of the first commercial refueling optimization codes had only twenty-eight different permutations to choose from [Carter 2002]. The permutations involved different reloading patterns from cycle to cycle.

Initially the fuel engineers used dynamic programming to minimize the cost subject to a few constraints such as power peaking and burnup. Other earlier methods involved direct search methods to maximize the cycle length. Linear programing was used to optimize fuel loading in five instead of three loading zones while minimizing the fresh fuel enrichment. Other authors looked at using variation techniques to locate ideal power distribution; linear programming was used to automate these methods. However many heuristic, or experienced-based, restrictions were placed on the fuel core loading to reduce the total number of possible permutations. [Carter 2002]

As it became desirable to optimize on more parameters, the direct search method became too time-consuming and was typically only used to fine-tune the results after linear programming was used to find an approximate maximum to the solution. Fine-tuning, which also only required small changes, led to evaluations by linear perturbation theory. The use of perturbation theory allowed the evaluation of even more permutations without having to run the time-consuming eigenvalue calculations for every scenario. [Carter 2002]

These older and simpler automation methods appear to be more applicable to modeling shim rotation for the ATR at startup. A more detailed review of three particular methods will be given with a description on how they can be applied to shim rotation automation:

- Dynamic programming
- Linear Programming
- Linear Perturbation Theory

Dynamic programming is an optimization procedure typically used in business or in commerce. It was also one of the first methods for optimization of fuel reloading in commercial nuclear reactors [Stover, 1969]. Dynamic programming solves complex problems by breaking them down into simpler sub-problems that will allow for a direct search for the optimum solution. With dynamic programming, an NxN dimensional problem can be reduced to a set of N by one-dimensional problem. [Downar 1988] Dynamic programming is an accelerated method compared to an exhaustive search but can still be very computationally intensive. Dynamic programming has been applied for certain limited loading schemes, but the complexity of the decision vectors with the general fuel location matrix lead to the abandonment of this approach. [Sauar 1971]

Shim rotation prediction for the ATR presently uses a stepwise approach similar to dynamic programming for the prediction of startup. The operational procedures for shim rotation prediction at startup are broken down in two main steps. First, all sixteen shims are rotated to the same angular positions while maintaining the reactor critical. Then the shims are rotated in groups of four to maintain the criticality while achieving the desired power split in the four lobes. These procedural requirements make it simpler for operators to achieve and maintain criticality.

Another method that was commonly used in initial fuel loading optimization is linear programming. In linear programming, the system and constraint equations are linearized, which allows the implementation of highly effective linear programming algorithms for optimization and automation of fuel reshuffling. This method works well when the problems are slightly nonlinear. Typical an initial guess is provided and successive calculations are made until the solution converges. However, for highly non-linear problems the initial guess can significantly change the converged solution because this method can only find local maximums/minimums and not a global maximum/minimum. [Downar 1988]

In one paper that implemented linear programming for fuel loading optimization the authors found that even with 8000 variables and 2000 constraints, optimal refueling calculations can be obtained within reasonable time. They accomplished this by solving sub problems of minimizing stage wise consumption of the fresh fuel. [Suzuki 1971] Because the shim rotation prediction has only four variables and around ten constraints, linear programming is a desirable method to investigate further to determine if it could be easily implemented for the automation process of shim rotation prediction of the ATR at startup.

Perturbation theory was another method that was initially used in the past for fuel loading optimization. Perturbation theory has been used for many applications besides optimization of fuel loading with the initial application developed by nuclear physicists. Wigner was one of the first scientist to apply perturbation theory to reactor physics in a

1945 report entitled “Effects of Small Perturbation on Pile Period”[Wigner, 1945]. Soodak was the one who showed that the adjoint could be considered an importance function [Soodak, 1965]. Hurwitz applied perturbation theory to reactivity worth [Hurwitz, 1948]. Since this earlier work, eigenvalue perturbation theory has become a standard principle taught to nuclear engineers and is included in many fundamental textbooks such as Lamarsh, Stacey, and Lewis and Miller. [Lamarsh, 2002; Stacey, 2007; Lewis, 1993]

One of the first applications of perturbation theory for optimization of loading patterns was presented at an ANS conference in 1973 [Federowicz]. An initial case was solved with a forward and adjoint solution and then linear programming, in conjunction with perturbation theory, was used to find changes in the reference-loading pattern such that the radial power peaking factor was minimized. The optimization goal was to minimize the peaking factor. The method described in the paper was used for 121 assemblies and was found to calculate a peaking factor of 1.17 with an initial value of 2.01 compared to a manual search method that only decreased the peaking factor to 1.33. [Federowicz 1973]

A more advanced utilization of perturbation theory for fuel optimization was described in a paper by Mingle [1975]. He used a two-part system where he first used linear programming and perturbation theory to minimize cost with a zoned core. He then calculated the shuffling of individual fuel assemblies that would minimize the distribution of the power peaking factor. Even though he used first order perturbation theory that typically does not work well for large changes to a system, he did a final check of the results with a forward calculation eliminating the need for using higher order perturbation theory. He also noted that because perturbation theory assumes the flux profile remains unchanged the procedure could pick some bad arrangements if too large of changes were analyzed. To prevent these bad arrangements he used heuristic rules, such as limiting reactivity changes between incremental steps, to help keep perturbation theory in check. He concluded that perturbation theory allowed for the rapid calculations of various fuel management decisions. [Mingle 1975].

Ho also used perturbation theory for nuclear fuel management optimization. In his paper, he describes an iterative process of using perturbation theory to find the reload pattern that would minimize the peak power. He would then run a forward and adjoint calculation based on optimal solution predicted by perturbation. He would continue this process until he converged on the optimal solution from the forward calculations. However, Ho only used this method with depleted fuel because he felt that new fuel being shuffled introduced too large of an error with perturbation theory. [Ho 1982]

There are some limitations with FOPT that require higher methods or a new type of perturbation theory to obtain reasonable results, as discussed in a paper from Kropaczek et al. In his paper, he used simulated annealing in conjunction with second order accurate generalized perturbation theory. He noted that first order perturbation theory lacked sufficient versatility to handle large perturbations such as shuffling fresh fuel. He did however use FOPT to calculate the changes in reactivity associated with burnup of the fuel once the initial fuel placement was calculated with a combination of simulated annealing and second order GPT. [Kropaczek et al. 1991]

FOPT can be used only to linearly predict the change in reactivity or the eigenvalue of the reactor core. Generalized Perturbation Theory (GPT) can be used to calculate ratios of other performance metrics obtained from a critical reactor. Ratios can include relative power in a specified region, relative power in a fuel pin, and reaction rate ratios [Williams, 1986]. The disadvantage of using GPT versus FOPT is that GPT requires the calculation of the generalized adjoint in addition to the forward and adjoint solution to the transport calculation. The generalized adjoint requires information from the forward and adjoint making GPT less optimal to run on parallel computers. In addition, a different generalized adjoint needs to be calculated for each ratio being analyzed.

Generalized perturbation also was an important tool for fuel cycle optimization. With the advent of faster and more computational power the combination of parallelism and GPT provides a very exhaustive, fast, and accurate way of sampling the solution space for the optimization of fuel cycle reloading patterns as long as the perturbations remain relatively small [Geemert 2001].

In one paper, Kim et al. [2001] were looking at multiple options for future fuel reloading of CANDU reactors such as recycled uranium, slightly enriched, mix oxide fuel, and direct use of spent PWR fuel (DUPIC). With all of the different permutations for the CANDU reactor design, he wanted to find a way of accelerating the adjustments of the zone controller unit (ZCU) after a fuel channel has been refueled. To do this he incorporated a combination of first order perturbation theory, and generalized perturbation theory.

The ZCU are vertical cylindrical compartments that consist of fourteen independent light water compartments that are located in the core. When a fuel channel is refueled the power level of that zone will change and this change is compensated by the ZCU water lever to both maintain the reference zone power and keep the reactor critical. [Kim 2001]

In Kims paper the FOPT was used to adjust the ZCU to maintain the appropriate eigenvalue while GPT was used to renormalize the reference zone power. In his conclusion, he said that the accuracy of the GPT and FOPT method to predict the ZCU change upon refueling was acceptable for an approximation of looking at core fuel management studies. [Kim 2001]

The applications discussed in the Kim et al paper are most closely analogous to the capability required for the ATR. The work considers the maintenance of the eigenvalue corrected for power splits by adjusting the water regions, which act very much like to rotating shims of the ATR. The largest difference between CANDU and ATR in this regards is that the ATR uses hafnium to control the power whereas the CANDU uses light water. The hafnium will make a much larger change in reactivity then the water and first order perturbation theory will most likely not work. A modified form of perturbation theory that can handle larger perturbations will need to be developed before a similar method would work with ATR shim rotation prediction.

Kim found first order GPT to provide sufficiently accurate enough results for his purpose. However, in another study by the author Moore [1996] found that first order GPT was not able to reasonably predict the large perturbations that arising from fresh fuel shuffling conditions, even though it could work for spent fuel shuffling. In his study, he demonstrated that even though GPT works well for small perturbations when large changes occur other methods or higher order perturbation theory needs to be explored. [Moore, 1996]

With the addition of higher order general perturbation theory for small changes, Geemert et al. [2001] found that it was possible to converge on the exact solution. However, he did say that the computational cost of using higher order generalized perturbation theory generally paid off only with the evaluation of several thousands of perturbations. Palmiotti [1987] was also able to use higher order GPT for both power and reactivity reconstruction. However, he found that for large perturbations he needed to calculate around 80 harmonics to get good results.

Generalized perturbation theory could be used to predict the shim rotation needed to obtain the desired power splits. However, this would require four additional generalized adjoint solutions in addition to the unperturbed forward and adjoint solution. In addition, the change from shim rotation prediction might be too large such that higher order perturbation theory would be needed to accurately predict the shim rotation for the desired power split instead of the simplified first order GPT.

According to Palmiotti [1987] over 80 harmonics were needed with higher order perturbation theory to get good results. Even though that was specific for his optimization scheme it still shows that using higher order GPT would not be a valuable option. What also makes higher order GPT less attractive is that the core configuration changes from cycle to cycle so that all of the initial calculations performed for predicting shim rotation would need to be repeated for each reactor cycle. Generalized perturbation theory could be further investigated, but currently does not look as promising as alternative methods such as linear programming.

In conclusion, there has been a lot of research in the area of automation of reactor calculations. The majority of the research has been specifically for commercial reactors but those methods could easily be applied to research reactors. Newer algorithms used for commercial reactor calculation automation work well when the problems are nonlinear and there are an excessive amount of available permutations i.e. 10^{100}. Luckily, the excessive number of permutations in commercial fuel loading designs do not apply to shim rotation prediction. However, some of the older methods for fuel loading appear to be more promising for automating shim rotation prediction, specifically linear programming and first order perturbation theory. These methods worked relatively well when there were few variables and constraints used for optimization. However, first order perturbation theory is expected to fail for shim rotation prediction because of the large change. Handling this large reactivity changes would require either using a higher order method of perturbation theory or developing a modified form of FOPT specifically designed for shim rotation prediction. The higher order method of perturbation theory requires too many coupled moment equations to be practical and computationally efficient; therefore, a modified form of FOPT was investigated for shim rotation prediction along with linear programming. Before a modified form of FOPT can be investigated, the typical form of FOPT was developed and benchmarked for a better understanding of the inherent limitations of FOPT.

STUDY 1: FIRST ORDER PERTURBATION THEORY

The purpose of using perturbation theory for modeling the ATR is to predict the change in reactivity to the system caused by shim rotations. From the literature review, it appears that first order perturbation theory (FOPT) will not be able to treat accurately such a large change to the system. However, quantifying what constitutes a large change in reactivity can be difficult without actually running first order perturbation theory for the system being analyzed. In addition, the results from first order perturbation theory can be compared to exact perturbation theory to determine if the changes in reactivity will be too large.

Theory

The two-dimensional discrete ordinate form of the neutron transport equation can be seen below.

$$\begin{aligned}\left(\Sigma_{tg}(\vec{r})+\mu_n\frac{\partial}{\partial x}+\eta_n\frac{\partial}{\partial x}\right)\varphi_{n,g}(\vec{r}) &= \lambda\chi_g\sum_{k=1}^{G}\left(\upsilon\Sigma_f(\vec{r})\right)_k\varphi_{n,k}(\vec{r})+\\ &\sum_{k=1}^{G}\sum_{l=0}^{L}\sum_{m=0}^{l}\sum_{p=1}^{N}(2-\delta_{mo})w_pY_{lm}^{e}(\hat{\Omega}_n)Y_{lm}^{e}(\hat{\Omega}_p)\left(\Sigma_{sl}(\vec{r})\right)_{k\to g}\varphi_{p,k}(\vec{r})\end{aligned} \tag*{[1]}$$

$\Sigma_{tg}(\vec{r})$ = Total macroscopic cross section (1/cm)

μ_n = Cosine of the scattering angle in the x direction

η_n = Cosine of the scattering angle in the y direction

$\varphi_{ng}(\vec{r})$ = Angular dependent flux $\left(\dfrac{\text{neutron}}{\text{cm}^2\,\text{sec}\,sr\;eV}\right)$

$\vec{r}$ = Directional vector of the neutron

λ = Lambda Eigenvalue

χ_g = Fraction of neutron born in energy group g

υ = Number of neutrons released per fission

$\Sigma_f(\vec{r})$ = Fission macroscopic cross section (1/cm)

G=Number of energy groups

L=Number of lengedre moments

N=Number of angular groups

w_p = Angular weighting

$\Sigma_{sl}(\vec{r})_{k\to g}$ = Legendre moment of order l of the scattering cross section from group k to group g $\left(1/cm\right)$

Y_{lm}^e = Real part of the Laplace Spherical Harmonics

$\hat{\Omega}_n$ = Scattering angle

For simplification, the operators T$_o$ and P$_o$ will be defined as:

$$T_o = \Sigma_{tg}(\vec{r}) + \mu_n \frac{\partial}{\partial x} + \eta_n \frac{\partial}{\partial x} - \sum_{k=1}^{G}\sum_{l=0}^{L}\sum_{m=0}^{l}\sum_{p=1}^{N}(2-\delta_{mo})w_p Y_{lm}^e(\hat{\Omega}_n)Y_{lm}^e(\hat{\Omega}_p)\left(\Sigma_{sl}(\vec{r})\right)_{k\to g}$$

[2]

$$P_o = \chi_g \sum_{k=1}^{G}\left(\upsilon\Sigma_f(\vec{r})\right)_k \quad [3]$$

Therefore, the eigenvalue form of the transport equation can be simplified.

$$\mathrm{T}_o\varphi_o = \lambda P_o\varphi_o \quad [4]$$

The subscript ‘o’ indicates the initial or unperturbed state. The adjoint of the transport equation can also be simplified to Equation 5. Note that the lambda-eigenvalue for the adjoint is the same lambda-eigenvalue used in the forward transport equation.

$$\mathrm{T}^*{}_o\varphi_o{}^* = \lambda P^*{}_o\varphi^*{}_o$$

$$* = \text{adjoint} \qquad [5]$$

The adjoint has the following property:

$$\left\langle \varphi^*, A\varphi \right\rangle = \left\langle \varphi, A^*\varphi^* \right\rangle + \text{boundary term} \quad [6]$$

Where < > represent the inner product or the integral overall all direction, energy, and space. For small changes in the system, the transport equation can be written as:

$$(T_o + \delta T)(\varphi + \delta\varphi_o) = (\lambda_o + \delta\lambda)(P + \delta P)(\varphi_o + \delta\varphi) \quad [7]$$

$$\delta T = \mathrm{T} - \mathrm{T}_o$$
$$\delta\varphi = \varphi - \varphi_o$$
$$\delta\lambda = \lambda - \lambda_o$$
$$\delta P = P - P_o$$

As shown in several textbooks [Lewis, 1993; Lamarsh, 2002; Stacey, 2007] under FOPT $\delta\lambda$ is given by:

$$\delta\lambda = \frac{\left\langle \varphi_o^{\,*} \left(\delta T - \lambda_o \delta P \right) \varphi_o \right\rangle}{\left\langle \varphi_o^{\,*} P_o \varphi_o \right\rangle}$$

[8]

Also shown in several textbooks [Lewis, 1993; Lamarsh, 2002; Stacey, 2007] exact perturbation theory is given by:

$$\delta\lambda = \frac{\left\langle \varphi_o^* \left(\delta T - \lambda_o \delta P \right) \varphi' \right\rangle}{\left\langle \varphi_o^* P' \varphi' \right\rangle} \quad [9]$$

The difference between exact perturbation theory and FOPT is that exact perturbation theory requires the perturbed flux for both the numerator and denominator inner products and requires the perturbed fission cross section instead of the unperturbed fission cross section in the denominator. Exact perturbation theory is not useful for most optimization schemes because it requires a forward solution for every perturbation. However, it can be useful to calculate the change in reactivity for very small changes eliminating the discretization and numerical error [Williams 1986].

For reactor calculations sometimes it is desirable to calculate the change in reactivity associated with the change to the core configuration. The change in reactivity is:

$$\delta\rho = -\delta\lambda = -\frac{\left\langle \varphi_o^{*}\left(\delta T - \frac{\delta P}{k_o}\right)\varphi_o \right\rangle}{\left\langle \varphi_o^{*} P_o \varphi_o \right\rangle}$$

[10]

Equation 10 is the generic form of first order perturbation theory. What separates the discrete ordinate form of the transport equation used in perturbation theory from other forms of the transport equation are the operators T and P. The δP operator for the discrete ordinate form of the transport equation is represented as:

$$\delta P = \chi_g \sum_{k=1}^{G} \left(\left(\upsilon \Sigma'_f(\vec{r}) \right)_k - \left(\upsilon \Sigma_{f_o}(\vec{r}) \right)_k \right)$$ [11]

o = Unperturbed cross section

Σ'_f = Perturbed fission cross section

G = Number of energy groups

The δT operator can be broken up into three parts: delta leakage, delta total, and delta scattering. The delta leakage terms $\mu_n \frac{\partial}{\partial x}$ and $\eta_n \frac{\partial}{\partial y}$ are independent of cross sectional changes and is therefore equal to zero. The delta total term is represented as:

$$\delta T_{\Sigma_t} = \Sigma'_{tg}(\vec{r}) - \Sigma_{tg}(\vec{r})_o$$ [12]

The delta scattering operator is represented as:

$$\delta T_{\Sigma_s} = \sum_{k=1}^{G}\sum_{l=0}^{L}\sum_{m=0}^{l}\sum_{p=1}^{N}(2-\delta_{mo})w_p Y_{lm}^{e}(\hat{\Omega}_n)Y_{lm}^{e}(\hat{\Omega}_p)\left(\left(\Sigma'_{sl}\right)_{i,k\rightarrow g}-\left(\Sigma_{so,l}\right)_{i,k\rightarrow g}\right) \quad [13]$$

Implementation

The perturbation theory code that implemented the above equations is called PertN (Perturbation theory for NEWT). PertN is a perturbation theory wrapper specifically developed for NEWT and uses the forward and adjoint solutions along with cross sections and other data obtained from the output of NEWT to calculate changes to the system. The overall implementation of PertN can be seen in the flowchart in Figure 2.

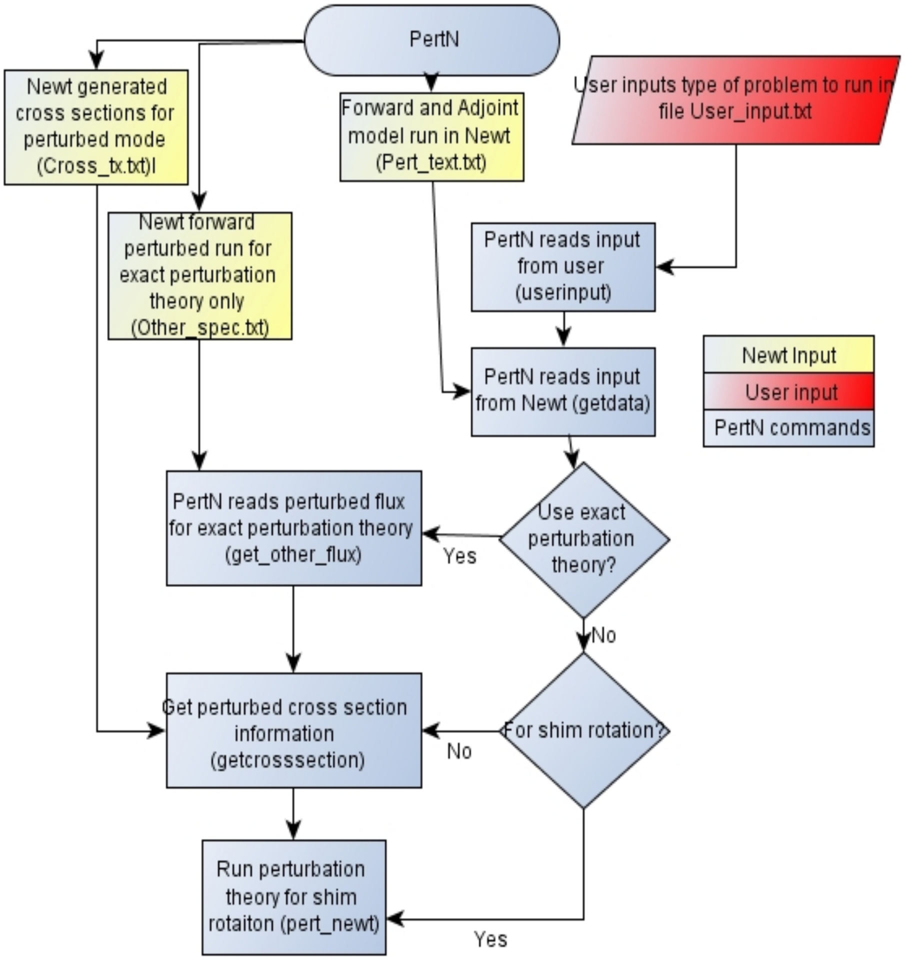

PertN
Newt generated cross sections for perturbed mode (Cross_tx.txt)l
Forward and Adjoint model run in Newt (Pert_text.txt)
User inputs type of problem to run in file User_input.txt
Newt forward perturbed run for exact perturbation theory only (Other_spec.txt)
PertN reads input from user (userinput)
PertN reads input from Newt (getdata)
Newt Input
User input
PertN commands
PertN reads perturbed flux for exact perturbation theory (get_other_flux)
Yes
Use exact perturbation theory?
No
Get perturbed cross section information (getcrosssection)
No
For shim rotation?
Run perturbation theory for shim rotaiton (pert_newt)
Yes

Figure 2 Implementation of PertN

PertN has the ability to solve perturbation theory for FOPT, exact perturbation theory, and phase space interpolated perturbation theory. In addition, PertN can be used to look at changes in material densities, fuel composition, and core configuration in addition to shim rotation prediction. The first step with PertN is to run NEWT to solve for the forward and adjoint solution. NEWT can also be used to generate cross section for the perturbed problem if FOPT is desired or to run forward solutions of the perturbed problem if exact perturbation theory is desired. For shim rotation prediction, PertN does not need any additional results from NEWT. The user also needs to provide an input deck that is used to tell PertN specifically what type of results the user wants. The input deck allows the user to choose if the code is for shim rotation. The user also specifies if exact perturbation theory is needed in addition to FOPT. Finally, the user specifies if the forward and adjoint calculations used for FOPT were calculated in the same input deck or separate input decks.

PertN can be run once the NEWT results and the user input deck has been copied over into the working directory. PertN will read in the user input deck and then get all of the cell, material, and the forward and adjoint results calculated with NEWT. Then depending on the user's input, PertN will either get other cross sections or perturbed fluxes. PertN will then solve for the desired user input and write the results to an Excel-readable file. The details of how to run PertN including what is in the user input deck can be found in the Appendix B.

Implementing perturbation theory into the computer code requires representing the inner product integrals as summations. Equation 14, Equation 15, and Equation 16, for the delta total, delta fission, and delta scattering term respectively represent the discretization of these inner products into the cell structure used by NEWT.

$$\left\langle \varphi_o^*, \Delta\Sigma_t \varphi \right\rangle = \sum_{i=1}^{I}\sum_{g=1}^{G}\sum_{n=1}^{N} w_n \varphi_{o,i,g,n}^{*} \varphi_{o,i,g,n} \left(\Sigma_{t,i,g}^{'} - \Sigma_{to,i,g} \right) V_i \qquad [14]$$

$$\left\langle \varphi_o^*, \Delta P\varphi \right\rangle = \sum_{i=1}^{I}\sum_{g=1}^{G}\sum_{k=1}^{G}\sum_{n=1}^{N} w_n \varphi_{o,i,g,n}^{*} \varphi_{o,i,k,n} \left(\chi_{i,g}^{'} \nu\Sigma_{f,i,k}^{'} - \chi_{o,i,g} \nu\Sigma_{fo,i,k} \right) V_i \qquad [15]$$

$$\left\langle \varphi_o^*, \delta\Sigma_{sl,,k\to g} \varphi \right\rangle =$$
$$\sum_{i=1}^{I}\sum_{g=1}^{G}\sum_{n=1}^{N}\sum_{k=1}^{G}\sum_{l=0}^{L}\sum_{m=0}^{l}\sum_{p=1}^{N} (2-\delta_{mo}) w_n w_p Y_{lm}^{e}(\hat{\Omega}_n) Y_{lm}^{e}(\hat{\Omega}_p) \varphi_{o,i,g,n}^{*} \varphi_{o,i,k,p} \left(\left(\Sigma_{sl}^{'}\right)_{i,k\to g} - \left(\Sigma_{so,l}\right)_{i,k\to g} \right) V_{i\ i} \qquad [16]$$

To speed up the computer code it is possible to reduce the seven-nested summations associated with the scattering term to a five-nested summation by initially solving for the two equations below beforehand:

$$\left(\phi_{o,l}^{m}\left(\bar{r}\right) \right)_{i,g} = \sum_{l=0}^{Lorder}\sum_{m=0}^{l}\sum_{n=1}^{ndir} Y_{lm,n}^{e} w_n \varphi_{o,i,g,n} \qquad [17]$$

$$\left(\phi_{o,l}^{*m}\left(\bar{r}\right) \right)_{i,g} = \sum_{l=0}^{Lorder}\sum_{m=0}^{l}\sum_{n=1}^{ndir} Y_{lm,n}^{e} w_n \varphi_{o,i,g,n}^{*} \qquad [18]$$

The results from these equations are then substituted back into the scattering cross section resulting in the five-nested summation scattering term.

$$\left\langle \varphi_o^*, \delta\Sigma_{sl,,k\to g} \varphi \right\rangle = \sum_{i=1}^{numcell}\sum_{g=1}^{numem}\sum_{k=1}^{numen}\sum_{l=0}^{Lorder}\sum_{m=0}^{l} (2-\delta_{mo}) \phi_{o,l}^{*m}\left(\bar{r}\right)_{i,g} \phi_{o,l}^{m}\left(\bar{r}\right)_{i,k} \left(\Sigma_{sl,i,k\to g}^{'} - \Sigma_{sl,0,i,k\to g} \right) V_i \qquad [19]$$

The inner product of the P operator found in the denominator of perturbation theory is implemented as:

$$\left\langle \varphi_o^*, P_o \varphi \right\rangle = \sum_{i=1}^{I} \sum_{g=1}^{G} \sum_{k=1}^{G} \sum_{n=1}^{N} w_n \varphi_{o,i,g,n}^* \varphi_{o,i,k,n} \chi_{o,i,g} \nu \Sigma_{fo,i,k} V_i \qquad [20]$$

The above equations are programed into the code PertN. An important part of developing the computer code PertN is to verify the code is working correctly and to validate the code is using the correct equations needed for running both FOPT and exact perturbation theory. This process is called verification and validation.

Verification and Validation

The goal of verification is to establish the numerical accuracy of the code independent of the physical model, or more scientifically stated, "it is the process of determining that a computational model accurately represents the underlying mathematical model and its solution" [Oberkampf, 2002]**.** Verification can be broken up into three parts: software quality engineering, code verification, and solution or calculation verification. A breakdown of the parts of verification for PertN can be seen in Figure 3.

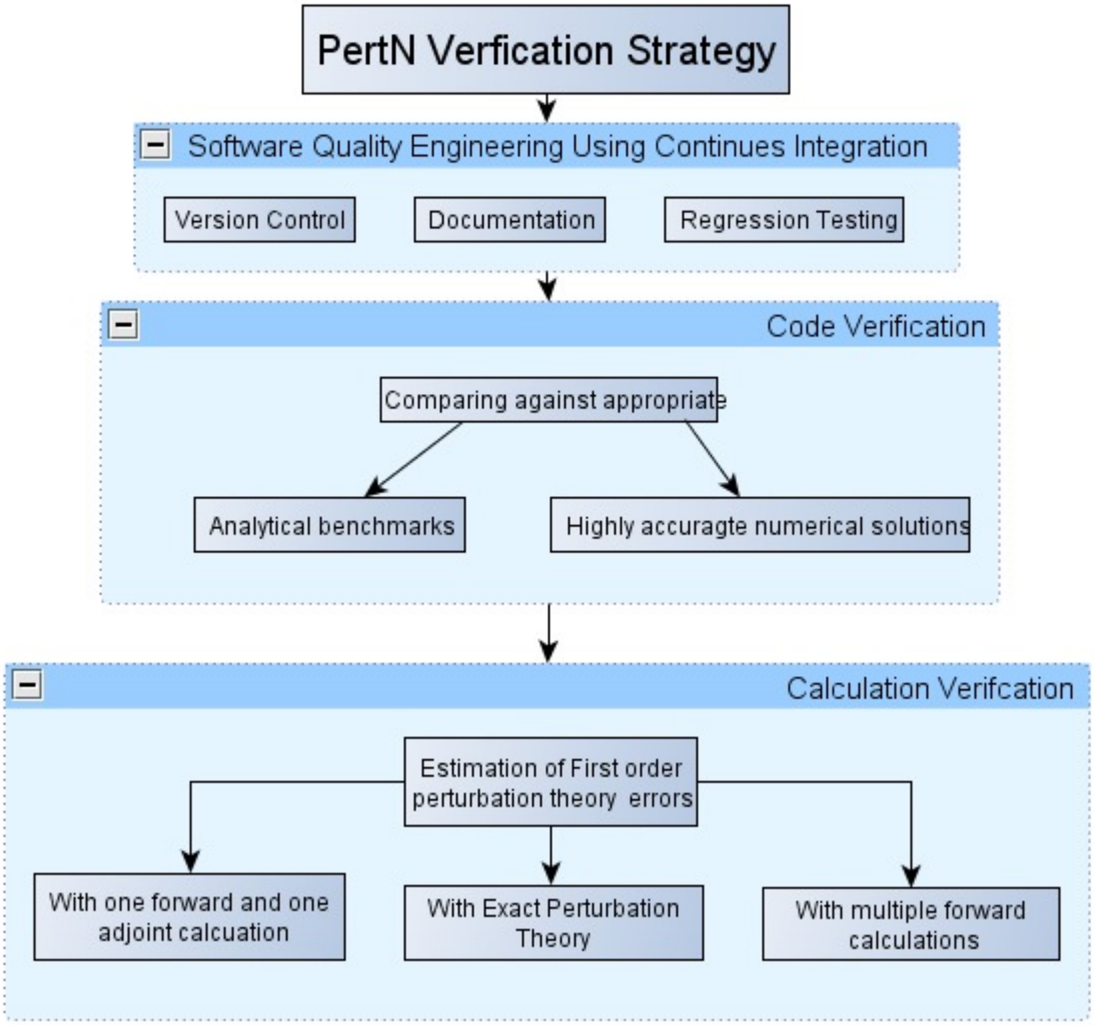

PertN Verfication Strategy
Software Quality Engineering Using Continues Integration
Version Control
Documentation
Regression Testing
Code Verification
Comparing against appropriate
Analytical benchmarks
Highly accuragte numerical solutions
Calculation Verifcation
Estimation of First order perturbation theory errors
With one forward and one adjoint calcuation
With Exact Perturbation Theory
With multiple forward calculations

Figure 3 Breakdown of verification for PertN

Software Quality Engineering

An essential part of verification is to make sure the results are repeatable through documentation and by recording all changes made to the code. Software quality engineering (SQE) uses the integration of tools and procedures to make all of the changes to the code documented concurrently with code development. The main components of SQE for PertN can be seen in Figure 4.

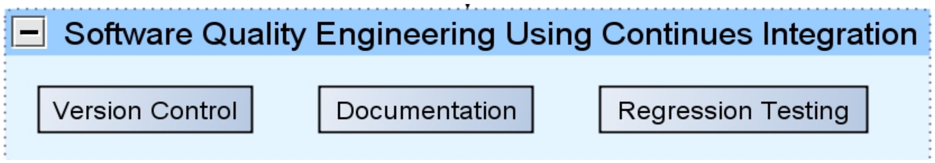


Figure 4 Three main components of software quality engineering used for PertN

Documentation of the PertN code for both users and programmers are important to help others understand the limits and strengths of the code. Documentation for PertN was developed with the software Doxygen [Heesch, 2011]. Doxygen is a documentation generator tool for writing software reference documents. The documents are written within the PertN code making it much easier to keep the document up to date. This also makes the code more readable for future programmers and external verification teams.

Version control is a computer software tool that manages changes to documents, programs, and other electronic information. The version control tool used for PertN is Subversion [Collins-Sussman et al. 2011]. In addition to tracking changes to the code, subversion makes the results accessible to others by allowing them to check out the most current version of the code and to easily update the code as new changes are made.

Regression testing is used to ensure that any changes to the code have not changed the final solution. Regression testing is accomplished by running a test suite and comparing the results to a database of expected values. Regression testing is accomplished by running the test suite every time the code has been modified and committed to the repository. For PertN every time a major change has been made to the code, a python script is run for regression testing. The script runs PertN with multiple benchmarked problems and then compares results from the analytical solutions.

SQE tools typically cannot find errors associated with the code, but is used in conjunction with the other verification methods to prevent new computational and programming errors from being introduced.

CODE VERIFICATION

The second part of the verification plan for PertN is code verification. Code verification is a comparison of the converge solution towards known “benchmarks”. The main task in code verification is to define the appropriate test problems for evaluating accuracy of an algorithm. The test problems run in the code are then compared against the known benchmarks for proper assessment. The benchmark problems need to be very accurate and specific to the type of application used in the code. The types of benchmarking problem in order of desirability are as follows [ASME 2006]:

- Exact analytical solution
- Semi-analytical solution with numerical integration of ODE
- Highly accurate numerical solution to PDE

There are two types of analytical solutions that are used for code verification: plausible but simplified real-world physics problems and manufactured solutions. The method of manufactured solutions is especially useful for a wide variety of highly nonlinear problems. For semi-analytical solutions used for verification benchmarks the analyst must be careful to ensure the numerical error has been significantly reduced. Numerical solutions to PDEs are the least reliable for benchmarking purposes. Extreme caution needs to be used to insure that the solution is highly accurate. In addition, extensive documentation on the verification process of the numerical solutions needs to be available. Benchmarking the results from PertN against MCNP5 [Brown, F. et al. 2002] would be an example of using a numerical solution for benchmarking purposes.

ANALYTICAL BENCHMARK

Multiple analytical benchmarks were chosen for code verification of PertN. The initial selection of benchmark came from a Los Alamos report called "Analytical Benchmark Test Set for Criticality Code Verification." [Avneet 2003]. This paper was a compilation of analytical benchmarks that were used to benchmark MCNP eigenvalue solutions. However, in the documentation and recommended references none of the benchmarks had analytical solutions to an adjoint equation. Therefore, additional work was needed to develop analytical benchmark test sets for perturbation theory verification. The two analytical problems used for perturbation theory verification are as follows:

- One energy group infinite slab isotropic
- Two energy group infinite slab isotropic

One energy infinite slab isotropic:

The first analytical benchmark that was set up for code verification was the flux and eigenvalue solution to an infinite, isotropic, homogenous transport model. The advantage of this scenario is the flux as a function of space is constant so the transport equation can be integrated over all space.

$$\Sigma_t\phi = \Sigma_s\phi + \lambda_\infty \nu\Sigma_f\phi \quad [21]$$

$\phi =$ Energy and space independent flux [n/cm^2 / s]
$\Sigma_t =$ Total macroscopic cross section [1/cm]
$\Sigma_s =$ Scattering macroscopic cross section [1/cm]
$\Sigma_f =$ Fission macroscopic cross section [1/cm]
$\nu =$ Number of neutrons produced per abosrption in the fuel
λ_∞==1/k$_\infty$Infinte medium eigenvalue

The analytical solution for the change in the eigenvalue from the forward transport is:

$$\Delta\lambda_\infty = \frac{\Sigma_t^{'} - \Sigma_s^{'}}{\nu^{'}\Sigma_f^{'}} - \frac{\Sigma_t - \Sigma_s}{\nu\Sigma_f} \quad [22]$$

$N^{'}$ = Perturbed Scenario

The analytical solution for first order and exact perturbation theory can be compared to the analytical solution for the change in the eigenvalue to calculate the error exactly. For the one energy group infinite medium benchmark, the analytical solution to first order and exact perturbation theory can be seen in Equation 23 and Equation 24 respectively.

$$\Delta\lambda_{first} = \frac{\left(\Delta\Sigma_t - \Delta\Sigma_s - \lambda_o \Delta\nu\Sigma_f\right)}{\left(\nu\Sigma_f\right)} \qquad [23]$$

$$\Delta\lambda_{exact} = \frac{\left(\Delta\Sigma_t - \Delta\Sigma_s - \lambda_o \Delta\nu\Sigma_f\right)}{\left(\nu\Sigma^{'}_f\right)} \qquad [24]$$

The error, or difference, between the analytical solution from the forward eigenvalue solution and the analytical solution solved with perturbation theory can be seen in Equation 25 and Equation 26 for first and exact perturbation theory respectively.

$$Error_{First} = \frac{-\Delta\lambda_o \Delta\nu\Sigma_f}{\nu\Sigma_f} \qquad [25]$$

$$Error_{Exact} = \frac{\Delta\nu\Sigma_f\left(\Sigma_t - \Sigma_s - \lambda_o \nu\Sigma_f\right)}{\nu\Sigma_f \nu^{'}\Sigma^{'}_f} = 0 \qquad [26]$$

As expected the error with exact perturbation theory is zero. The error of first order perturbation theory would also be zero if there were no change in the fission cross section. Otherwise, there would be an error that will under-predict positive reactivity and over-predict negative reactivity.

The equations above were used to benchmark PertN. Six different changes were analyzed for 10% increase and decrease for the nu*fission, scattering, and total cross sections. The perturbed cross sections used in this benchmark can be seen in Table 1.

Table 1 Cross section used for perturbation theory analytical benchmark for 1 energy group infinite medium

Changes	$\nu\Sigma_f$ [1/cm]	Σ_s [1/cm]	Σ_t [1/cm]	λ
Standard	1.6575E-02	3.4661E-02	4.1949E-02	4.3970E-01
0.9νΣf	1.4918E-02	3.4661E-02	4.1949E-02	4.8856E-01
1.1νΣf	1.8233E-02	3.4661E-02	4.1949E-02	3.9973E-01
0.9Σs	1.6575E-02	3.1195E-02	4.1949E-02	6.4881E-01
1.1Σs	1.6575E-02	3.8127E-02	4.1949E-02	2.3059E-01
.9Σt	1.6575E-02	3.4661E-02	3.7754E-02	1.8662E-01
1.1Σt	1.6575E-02	3.4661E-02	4.6144E-02	6.9278E-01

The results from the one energy group infinite medium benchmark can be seen in

Table 2. As was expected, the difference between the forward solution and the results with perturbation theory has relatively small error except when the fission cross section is perturbed. The results for the change in reactivity are all in units of pcm, or percent millirho. Pcm is calculated by multiplying the change in reactivity by 10,000.

Table 2. One energy group infinity medium analytical benchmark

Changes	Δλ (Forward)[pcm]	Δλ (FOPT)[pcm]	Diff [pcm]	Δλ (Exact)[pcm]	Diff[pcm]
Standard	0.0000E+00	0.0000E+00	0.000E+00	0.0000E+00	0.000E+00
$0.9\nu\Sigma_f$	-4.8856E+03	-4.3970E+03	-1.000E-01	-4.8856E+03	2.728E-12
$1.1\nu\Sigma_f$	3.9973E+03	4.3970E+03	1.000E-01	3.9973E+03	-9.095E-13
0.9Σs	-2.0911E+04	-2.0911E+04	0.000E+00	-2.0911E+04	0.000E+00
1.1Σs	2.0911E+04	2.0911E+04	3.638E-12	2.0911E+04	3.638E-12
$.9\Sigma_t$	2.5308E+04	2.5308E+04	0.000E+00	2.5308E+04	0.000E+00
$1.1\Sigma_t$	-2.5308E+04	-2.5308E+04	7.276E-12	-2.5308E+04	7.276E-12

Two energy group infinite medium isotropic:

The second analytical benchmark that was used to verify PertN was a two energy group infinite medium with isotropic scattering. Just like the first analytical benchmark, this problem is also an infinite medium benchmark. Therefore, the leakage term in the transport equation is equal to zero and the rest of the transport equation can be integrated over space resulting in Equation 27 and Equation 28 for FOPT and exact perturbation theory respectively.

$$\Sigma_{t1}\phi_1 = \Sigma_{11s}\phi_1 + \Sigma_{12s}\phi_2 + \lambda_\infty \chi_1 \left[\nu_1 \Sigma_{f1}\phi_1 + \nu_2 \Sigma_{f2}\phi_2 \right] \quad [27]$$

ϕ_1 = Fast flux

ϕ_2 = Thermal flux

$$\Sigma_{t2}\phi_2 = \Sigma_{22s}\phi_2 + \Sigma_{21s}\phi_1 + \lambda_\infty \chi_2 \left[\nu_2 \Sigma_{f2}\phi_2 + \nu_1 \Sigma_{f1}\phi_1 \right] \quad [28]$$

These equations can be written in the matrix form.

$$\begin{bmatrix} \left(\Sigma_1^{rem} - \lambda_\infty \chi_1 \nu_1 \Sigma_{f1}\right) & -\left(\Sigma_{12s} + \lambda_\infty \chi_1 \nu_2 \Sigma_{f2}\right) \\ -\left(\Sigma_{21s} + \lambda_\infty \chi_2 \nu_1 \Sigma_{f1}\right) & \left(\Sigma_2^{rem} - \lambda_\infty \chi_2 \nu_2 \Sigma_{f2}\right) \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad [29]$$

$$\Sigma_2^{rem} = \Sigma_{t2} - \Sigma_{22s}$$
$$\Sigma_1^{rem} = \Sigma_{t1} - \Sigma_{11s}$$

The adjoint is easily calculated by taking the adjugate of the matrix.

$$\begin{bmatrix} \left(\Sigma_2^{rem} - \lambda_\infty \chi_2 \nu_2 \Sigma_{f2}\right) & \left(\Sigma_{12s} + \lambda_\infty \chi_1 \nu_2 \Sigma_{f2}\right) \\ \left(\Sigma_{21s} + \lambda_\infty \chi_2 \nu_1 \Sigma_{f1}\right) & \left(\Sigma_1^{rem} - \lambda_\infty \chi_1 \nu_1 \Sigma_{f1}\right) \end{bmatrix} \begin{bmatrix} \phi_1^* \\ \phi_2^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad [30]$$

The eigenvalue solution of both the forward and adjoint equation can be calculated by setting the determinant equal to zero. As expected, the eigenvalue solution for both the forward and adjoint equations are identical.

$$\lambda_\infty = \frac{\Sigma_1^{rem}\Sigma_2^{rem} - \Sigma_{12s}\Sigma_{21s}}{\left(\chi_1\left(\nu_2\Sigma_{f2}\Sigma_{21s} + \Sigma_2^{rem}\nu_1\Sigma_{f1}\right) + \chi_2\left(\nu_1\Sigma_{f1}\Sigma_{12s} + \Sigma_1^{rem}\nu_2\Sigma_{f2}\right)\right)} \quad [31]$$

The relationship between the fast and thermal flux can also be calculated with some algebraic manipulation.

$$\frac{\phi_1}{\phi_2} = \frac{\left[\lambda_\infty \nu_2 \Sigma_{f2} - \Sigma_2^{rem} + \Sigma_{12s}\right]}{\left[\Sigma_1^{rem} - \Sigma_{21s} - \lambda_\infty \nu_1 \Sigma_{f1}\right]} \quad [32]$$

The ratio of the fast and thermal adjoint as, seen in Equation 33, is solved the same way as the forward solution with the exception that the relationship $\chi_1 + \chi_2 = 1$ cannot be used to simplify the results.

$$\frac{\phi_1^*}{\phi_2^*} = -\frac{\left[\Sigma_1^{rem} + \Sigma_{12s} + \chi_1\left(\lambda_\infty \nu_2 \Sigma_{f2} - \lambda_\infty \nu_1 \Sigma_{f1}\right)\right]}{\left[\Sigma_2^{rem} + \Sigma_{21s} + \chi_2\left(\lambda_\infty \nu_1 \Sigma_{f1} - \lambda_\infty \nu_2 \Sigma_{f2}\right)\right]} \quad [33]$$

The three operations for implementing exact perturbation theory are given below; the same equations are also valid for FOPT when the unperturbed flux is used instead of the perturbed flux everywhere, and the unperturbed fission cross section is used in Equation 36 instead of the perturbed fission cross section.

$$\phi^* \delta T \phi = \begin{bmatrix} \phi_1^* \\ \phi_2^* \end{bmatrix} \begin{bmatrix} \left(\Sigma^{'}_{R1} - \Sigma_{R1}\right) + \left(\Sigma^{'}_{12s} - \Sigma_{12s}\right) \\ \left(\Sigma^{'}_{21s} - \Sigma_{21s}\right) + \left(\Sigma^{'}_{R2} - \Sigma_{R2}\right) \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} \quad [34]$$

$$\varphi^* \lambda_o \delta P \varphi = \lambda_o \begin{bmatrix} \varphi_1^* & \varphi_2^* \end{bmatrix} \begin{bmatrix} \left(\chi^{'}_1 \nu^{'}_1 \Sigma^{'}_{f1} - \chi^{'}_1 \nu_1 \Sigma_{f1}\right) + \left(\chi^{'}_1 \nu^{'}_2 \Sigma^{'}_{f2} - \chi_1 \nu_2 \Sigma_{f2}\right) \\ \left(\chi^{'}_2 \nu_1^{'} \Sigma^{'}_{f1} - \chi_2 \nu_1 \Sigma_{f1}\right) + \left({}^{'}\chi_2 \nu^{'}_2 \Sigma^{'}_{f2} - \chi_2 \nu_2 \Sigma_{f2}\right) \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} \quad [35]$$

$$\varphi^* \left(P_o + \delta P\right) \varphi = \begin{bmatrix} \phi_1^* \\ \phi_2^* \end{bmatrix} \begin{bmatrix} \chi_1 \nu_1^{'} \Sigma_{f1}^{'} & \chi_1 \nu_1^{'} \Sigma_{f1}^{'} \\ \chi_2 \nu_1^{'} \Sigma_{f1}^{'} & \chi_2 \nu_2^{'} \Sigma_{f2}^{'} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} \quad [36]$$

The cross sections used in the two-energy group infinite medium benchmark can be seen in Table 3

Table 3 Initial cross section data used for second benchmark

	$\nu\Sigma_f$ [1/cm]	$\Sigma_{s1->1}$ [1/cm]	$\Sigma_{s2->1}$ [1/cm]	Σ_t [1/cm]	χ	Σ_r[1/cm]	Θ[n/cm2/s]
Fast	1.7588E-02	2.5448E-02	5.6515E-03	3.7646E-02	5.6620E-01	1.2197E-02	5.0199E-01
Slow	1.6116E-02	3.5828E-02	0.0000E+00	4.3356E-02	4.3380E-01	7.5283E-03	1.0000E+0
		$\Sigma_{s2->2}$ [1/cm]	$\Sigma s_{1->2}$ [1/cm]				

Twelve different configurations were analyzed in this benchmark. The configurations involved changing one parameter either at ± 10% of the initial data. The configurations looked at the effects of fission, total, group to group scattering, and down scattering. The results from this benchmark can be seen in Table 4.

Table 4 Analytical results used for comparison with PertN

Changes	Δλ Forward [pcm]	ϑt[n/cm2/s]	Δλ (FOPT)[pcm]	pcm off	Δλ (Exact)[pcm]	pcm off
0.9νΣf_1	-1.5907E+03	5.0199E-01	-1.5344E+03	5.6305E+01	-1.5907E+03	-4.0383E-03
0.9νΣf_2	-2.9943E+03	5.0199E-01	-2.8009E+03	1.9345E+02	-2.9943E+03	1.5201E-03
1.1νΣf_1	1.4820E+03	5.0199E-01	1.5344E+03	5.2452E+01	1.4820E+03	1.2812E-03
1.1νΣf_2	2.6309E+03	5.0199E-01	2.8009E+03	1.6997E+02	2.6309E+03	-3.0573E-03
.9Σs,1→1	-4.9849E+03	4.4424E-01	-5.4037E+03	-4.1871E+02	-4.9850E+03	1.4839E-02
.9Σs,2→2	-1.1408E+04	7.4090E-01	-1.3330E+04	-1.9216E+03	-1.1408E+04	3.5610E-03
1.1Σs,1→1	5.8992E+03	5.7702E-01	5.4037E+03	-4.9552E+02	5.8992E+03	2.3860E-03
1.1Σs,2→2	1.6030E+04	2.6309E-01	1.3330E+04	-2.7001E+03	1.6030E+04	4.4265E-03
0.9Σt_1	9.1278E+03	6.2153E-01	7.9936E+03	-1.1342E+03	9.1278E+03	2.5014E-03
0.9Σt_2	2.0260E+04	2.1289E-01	1.6131E+04	-4.1299E+03	2.0260E+04	3.3123E-03
1.1Σt_1	-7.1101E+03	4.2102E-01	-7.9936E+03	-8.8347E+02	-7.1101E+03	9.5705E-03
1.1Σt_2	-1.3399E+04	7.9110E-01	-1.6130E+04	-2.7313E+03	-1.3399E+04	3.8928E-03
.9Σs,2→1	-1.0818E+03	5.2165E-01	-1.0555E+03	2.6340E+01	-1.0818E+03	-2.1090E-03
1.1Σs,2→1	1.0304E+03	4.8376E-01	1.0555E+03	2.5085E+01	1.0304E+03	9.9303E-04

The change in the lambda eigenvalue for the forward, FOPT, and exact peturbation theory were solved exactly. For benchmarking purposes the analyitical solution seen

Table 4 were compared against PertN. The differences between the benchmarked results and PertN are less then 0.07 pcm. This small difference demonsrates that PertN is working correctly for this analyitical benchmark problem. In addition this table shows that the differences between exact perturbation theory and the forward solution are statistically insignificant. As expected the difference between FOPT and the forward solution is significantly different. This demonstrates the limitations to perturbation theory with large differences between the perturbed and unperturbed fluxes. For example, the largest difference in flux, which is when the total cross section is perturbed by 90%, is also the largest error between FOPT and the forward solution, over 400 pcm. The inverse of this is also true as seen with 110% change in the downscattering cross section.

Results

The final component of verification is solution or calculation verification. This portion of the verification process is to estimate the systematic errors associated with the assumptions used for deriving FOPT. One way to calculate the error associated with first order perturbation theory can be seen in the steps below:

1. Solve forward and adjoint transport solution for the unperturbed case.
2. Solve the forward solution for the perturbed case.
3. Apply perturbation theory to the unperturbed flux (results should be tangential to the perturbed solution of step 1).

4. Apply perturbation theory to the perturbed forward flux and unperturbed adjoint (results should be exact to the full transport solution of the perturbed case).
5. Compare both exact and first order perturbation theory to the eigenvalue calculated with NEWT.

Three different reactor models were used for calculation verification on PertN. Each model more accurately represents the ATR core. The first model is a homogenized one-region model. The purpose of this model is to test simple changes in density with PertN in fuel regions. The second model is a three-region model with the dimensions representing the dimensions of the ATR. The goal of the second model is to test changes in density in the un-fueled regions. The last model is a very simplified model of the ATR but has the full core model of the OSCC shims and the correct shape of the ATR serpentine fuel. The purpose of this model is to test PertN's ability to swap materials in non-fueled regions. In addition, this model also is used to demonstrate the capability of OSCC shim rotation prediction with FOPT.

The first simplified model that was used for PertN calculation verification can be seen in Figure 5. It is a one region 100 cm by 100 cm smeared reactor composed of 90 wt% U-235 and 10 wt% aluminum with a total density of 1.6 g/cc.

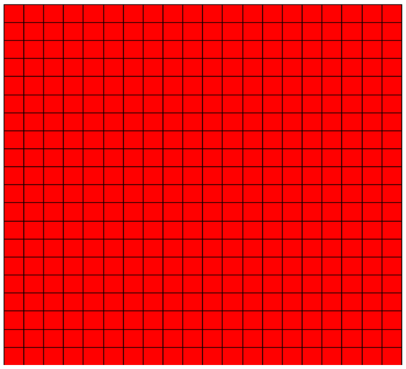


Figure 5 First test geometry

The density of the fuel was perturbed from 1.6 g/cc to 2.5 g/cc in 0.1 g/cc increments. At each step, both the forward and adjoint solutions were calculated with NEWT. The results from NEWT were used with PertN to calculate FOPT and Exact Perturbation theory. For the FOPT calculation the reference solution for both the forward and adjoint calculations were at 1.6 g/cc. The results from these calculations can be seen in Figure 6.

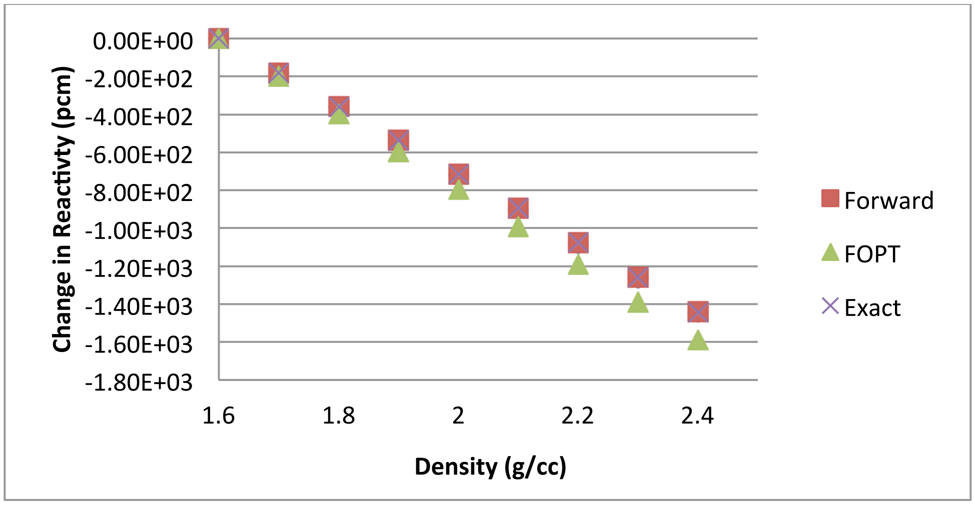


Figure 6 Keff as a function of water density

From Figure 6 it can be seen that first order perturbation theory predicts the slope of the eigenvalue solution at the unperturbed case as expected. The results from this example demonstrate that PertN is working correctly for the above test case. The figure also displays some of the limitations of first order perturbation theory. As the density in fuel departed further from the reference value, the accuracy of FOPT decreased. The relative error associated with perturbation theory for the change in density can be seen in Figure 7.

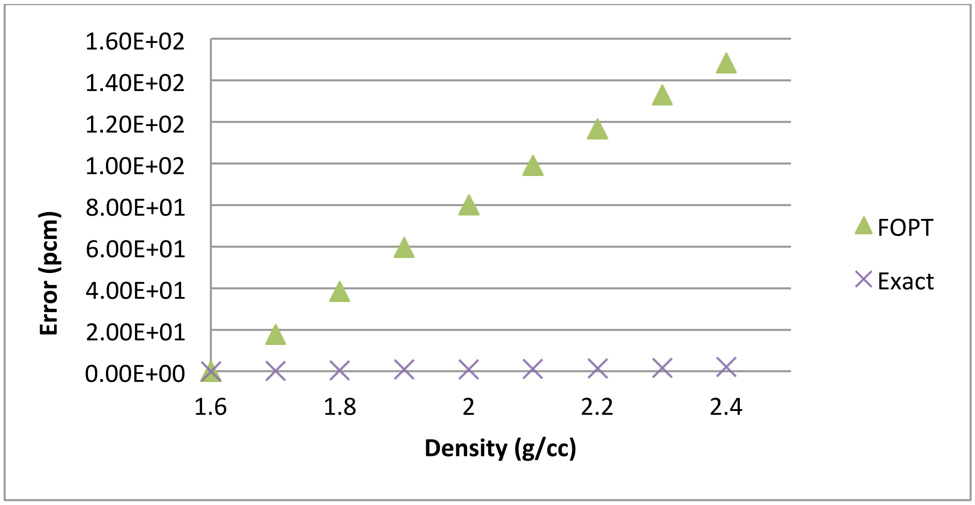


Figure 7 Relative error of the Keff in PCM as a function of variation in the uranium density

The second model used for calculation verification is the simplified three-region model with the inner region composed of water with a density of 1.10 g/cc and an outer radius of 3.266 cm. The middle region was 95 weight percent enriched uranium-235 fuel with a density of 4.2 g/cc. The middle region had an outer radius of 5.804 cm. The outermost region was 100% Be with an outer radius of 25.25 cm and a density of 1.8538 g/cc. A pictorial representation of the simplified model taken from the output of NEWT can be seen in the Figure 8.

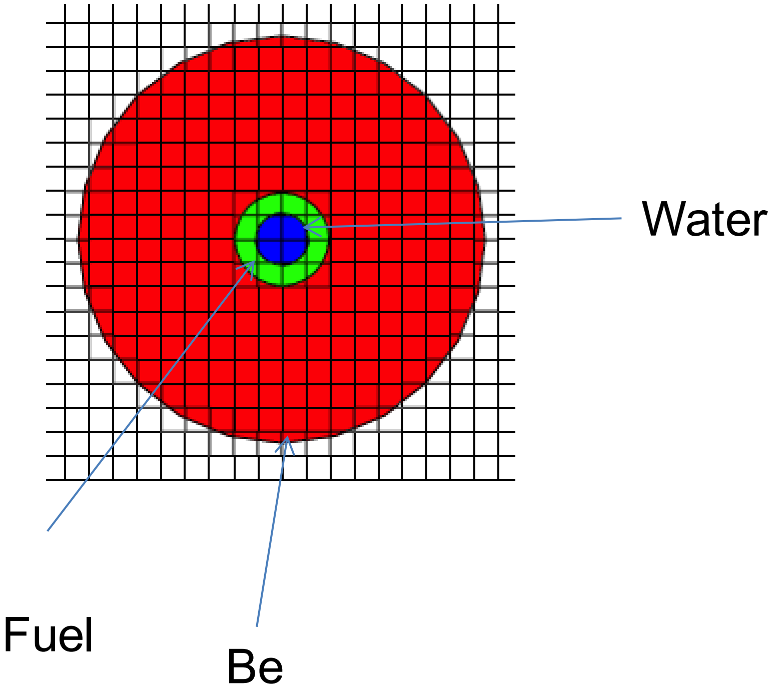


Figure 8 Second test geometry

This model was used to test reactivity predictions for change in the density of water. The density of the water varied from 0.7 g/cc to 1.7 g/cc in 0.1 g/cc increments. The results from the calculation verification of the three-region model can be seen in Figure 9 and Figure 10 for water.

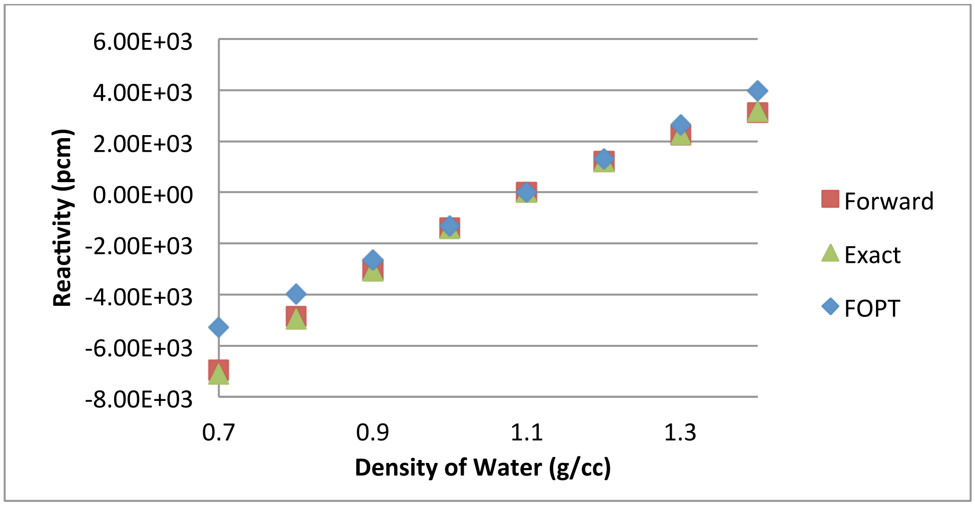


Figure 9 Calculation verification associated with water density perturbations

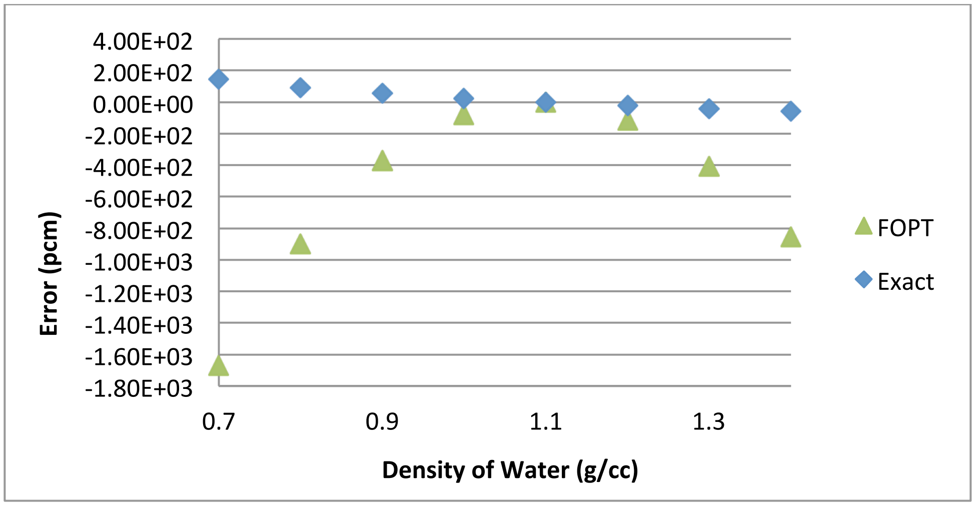


Figure 10 Calculation verification associated with fuel density perturbations

As seen in Figure 10, the error for FOPT gets larger as the changes in reactivity increases. When determining when FOPT would be appropriate to use it is important to consider what type of convergence criteria is needed. In addition, as seen in Figure 10, the error of exact perturbation theory is not exactly zero. A common reason that exact perturbation theory does not match up with forward calculations is because the mesh size has not converged to a stable solution for all perturbations. Therefore, if exact perturbation theory and forward calculations do not line up then a tighter grid tolerance is needed.

The last model that was used for calculation verification was a simplified model of the ATR. The simplified model preserved the same shape and dimensions of the ATR but had some modifications to reduce the run time of the model. The simplifications involved the following (Figure 11):

- All of the flux traps were homogenized and filed with aluminum 6061 density 2.7g/cc except the center flux trap where the H holes were filled with cobalt with a density of 8.9 g/cc.
- The outer beryllium reflector with a density of 1.85 g/cc had no cooling channels.
- The 93% enriched fuel that was homogenized with Al6061 and water was still serpentine shaped but was only one region and had a density 1.14 g/cc.
- The drums were identical to the full core reactor model, containing the inner cooling channel, beryllium body, and 120 degree hafnium segments with a density of 13.31 g/cc.

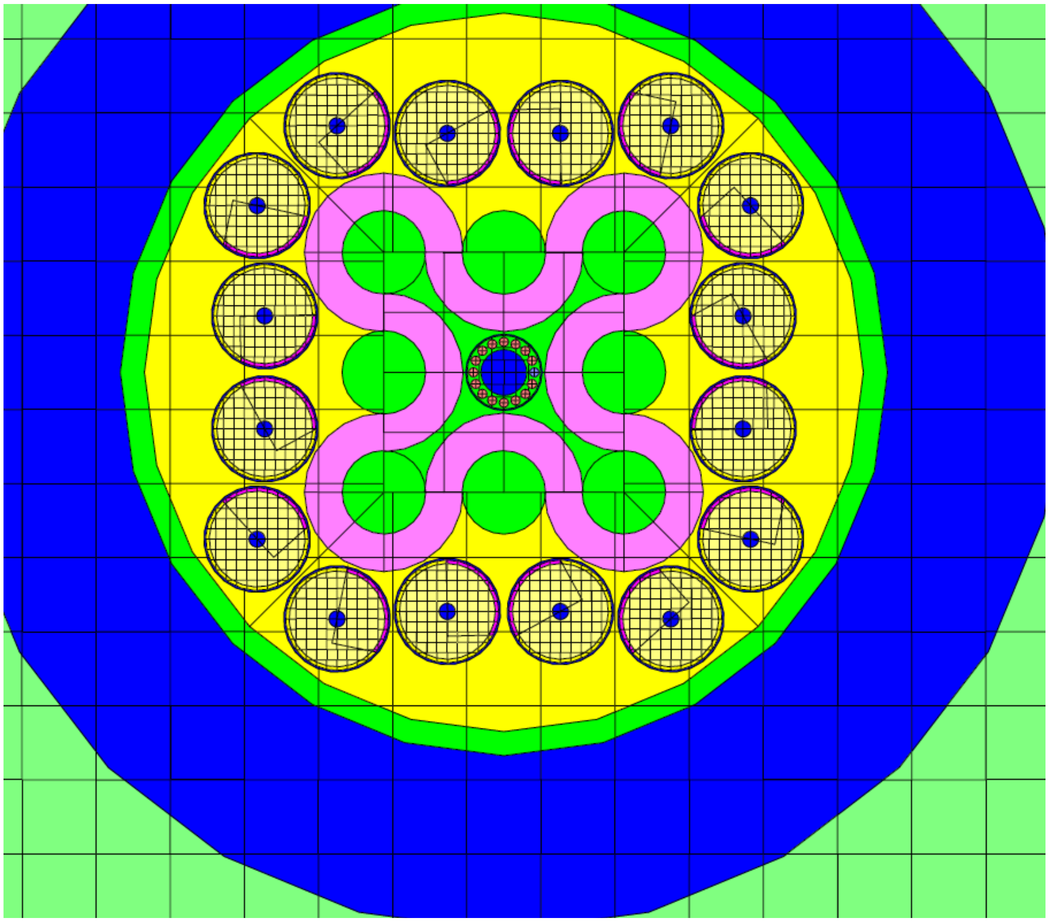


Figure 11 Simplified model used for calculation verification

The perturbation calculations for the third model looked at changes from swapping around material composition of the H5 hole from cobalt to aluminum. The goal this perturbation is to understand how swapping materials in an experimental position affects the ability of FOPT to accurately predict the change in reactivity. The perturbation occurred at the H5 position as seen in Figure 12.

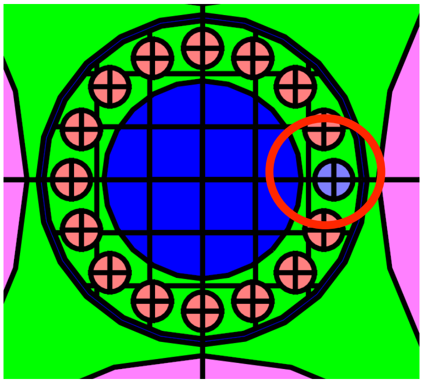


Figure 12 H holes in the simplified ATR model

The reactivity effect of change in composition from cobalt to aluminum within the H5 position can be seen in Figure 13, Figure 14, and Figure 15. The difference between the figures is the unperturbed frame of reference used for calculating reactivity with FOPT. In Figure 13, the unperturbed reference for FOPT is when the material is 50% cobalt and 50% aluminum. In Figure 14, the unperturbed reference is 100% cobalt. In Figure 15, the unperturbed reference is 100% aluminum.

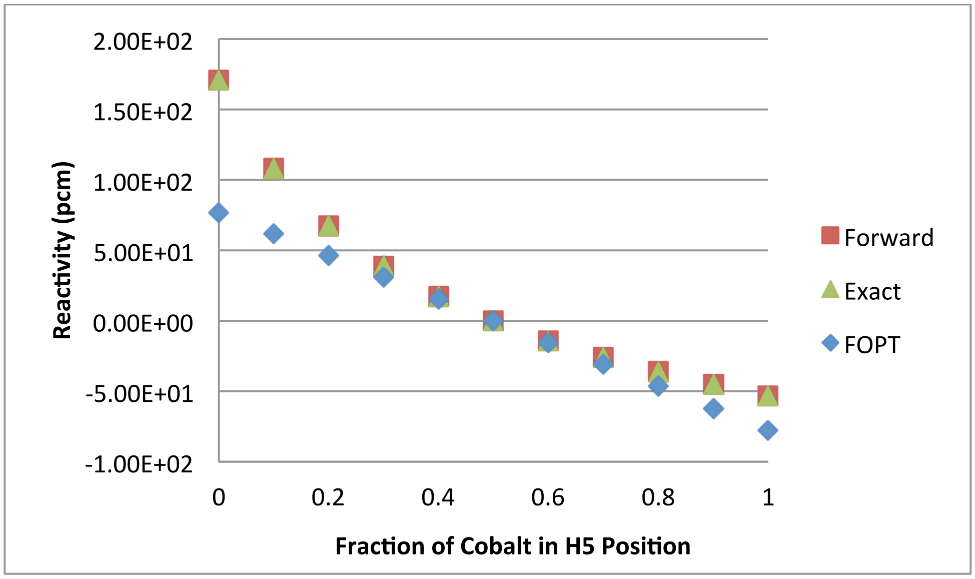


Figure 13 Changing material composition in H hole with unperturbed data at 50% Cobalt 50% Aluminum

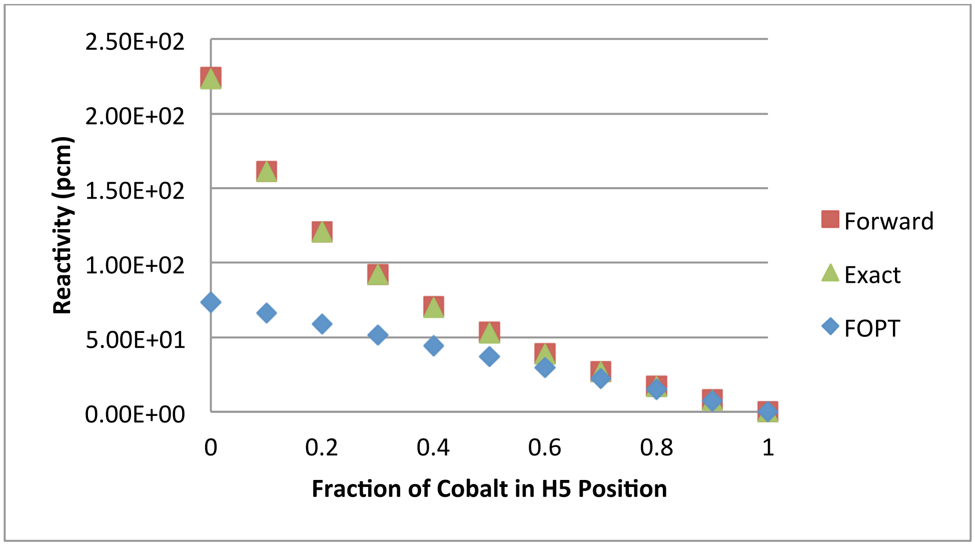


Figure 14 Changing material composition in H hole with unperturbed data at 100% Aluminum

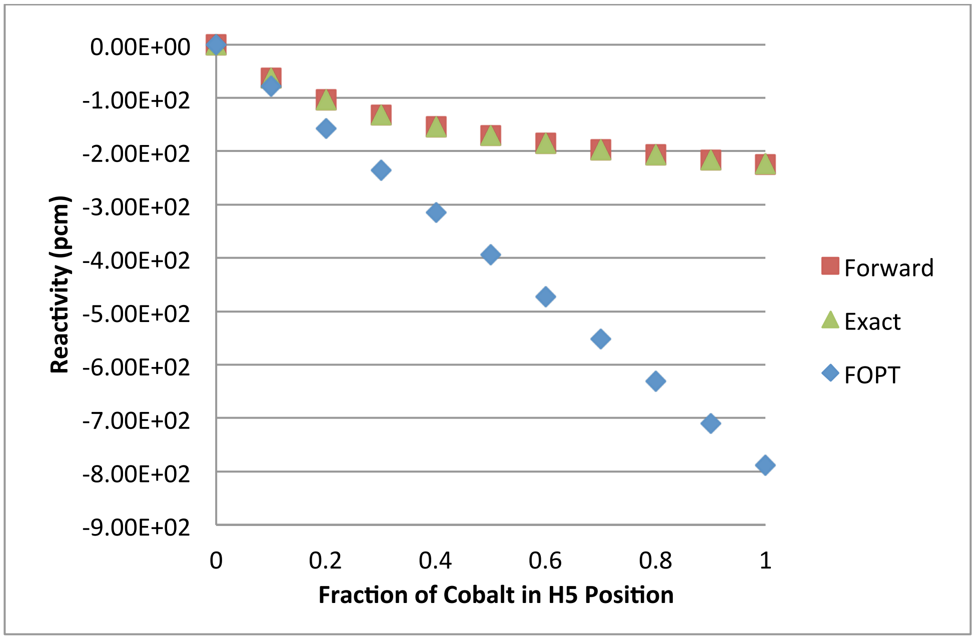


Figure 15 Changing material composition in H hole with unperturbed data at 100% Cobalt

From the figures above it could be concluded that if perturbation theory was applied to an unperturbed experiment that had mostly cobalt and a little aluminum in it, and that experiment was swapped for an experiment with all aluminum FOPT would not introduce a relatively large error. However, if the unperturbed scenario was aluminum and the swapped experiment had a lot of cobalt then FOPT would produce a relatively large error when predicting the change in reactivity. These is because for the 100% aluminum the adjoint, or importance function, has a higher weighting value then the 100% cobalt scenario. This higher weighting value will introduce a larger error when the cross sections are perturbed.

The final study was performed on shim rotation prediction using FOPT. When implementing shim rotation prediction the way to rotate the shims is to assign some regions, or cells, hafnium that were once beryllium and other regions beryllium that were once hafnium (See Appendix C for more information). Figure 16 compares the results with FOPT to forward calculations. The unperturbed reference case for FOPT occurred at 50 degrees.

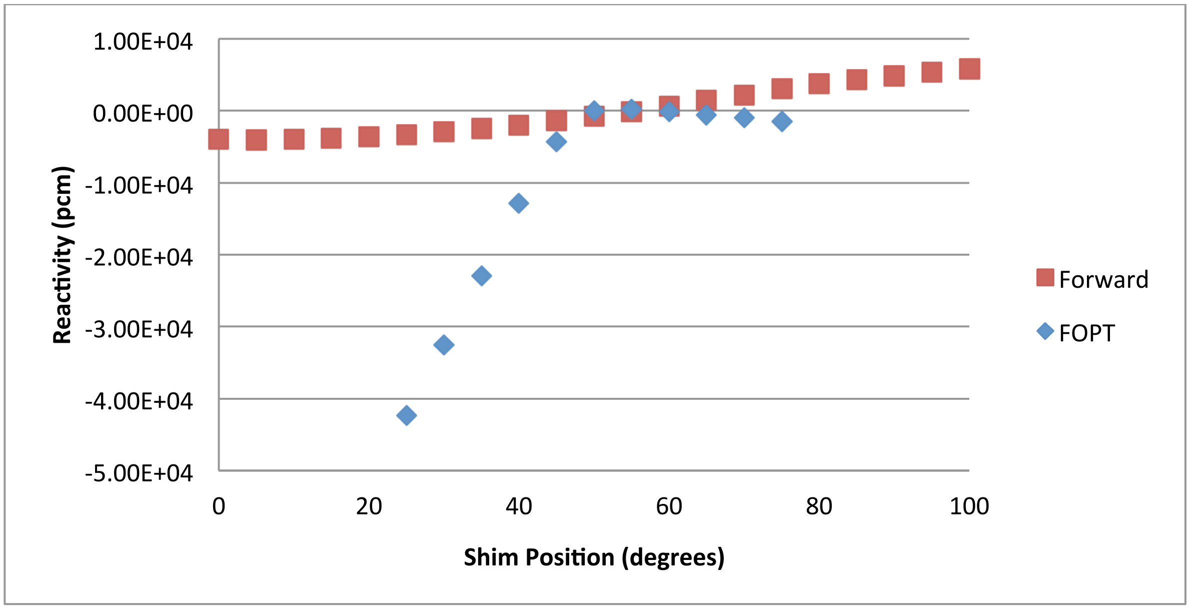


Figure 16 Comparison between FOPT and forward calculations for shim rotation prediction

The results from FOPT over predicted the negative reactivity and under predicted the positive reactivity. This is because going from a low absorbing material to a high absorbing material produces a larger error then going from a high absorbing material to a low absorbing material as seen in Figure 15 and Figure 14 respectively. From the computational results displayed above it appears that FOPT will introduce too large an error into the shim rotation prediction model, as was expected from the literature review.

Conclusion of FOPT

From these results, a few restrictions on using FOPT have been identified. The first restriction is that the domain of applicability of FOPT depends on the required fidelity and needs to be done on a case-by-case basis. The second restriction is FOPT does not work well when swapping one experiment for another if the composition of the material significantly differs, especially if the material had a relatively large absorption cross section. However, perturbation theory works well for swapping out experiments if the composition of the experiments are similar and no new highly absorbing materials are introduced into the experiment location. The final restriction is that FOPT will not work for shim rotation prediction of the Advanced Test Reactor. Either a higher order method for perturbation theory or a different type of perturbation theory is needed to use perturbation theory for shim rotation prediction.

STUDY 2: PHASE SPACE INTERPOLATED PERTURBATION THEORY

As mentioned previously, first order perturbation theory works well for small changes to the system, but can over or under predict changes in reactivity associated with large changes to the energy dependent angular flux. When first order perturbation theory fails, the next step is typically to use higher order perturbation theory [Williams 1986]. However, higher order perturbation theory requires many higher order flux moments to produce accurate results. For example, Palmiotti noted that over 80 harmonics were needed with higher order perturbation theory to get usable results [Palmiotti, 1987]. In addition, NEWT does not have the capability of calculating these higher order flux moments. This would make higher order perturbation theory too computationally intensive for automation of shim rotation. Instead of investigating the use of higher order perturbation theory, a new method was developed that would more accurately predict large changes to the system with only two transport solutions. This new method is called phase space interpolated perturbation theory, because it attempts to interpolated the phase space of the perturbed region.

Theory

With first order perturbation theory, it is assumed that the change to the system is relatively small so that any higher order terms are set equal to zero. However, exact perturbation theory makes no assumption on the amount of change that is being introduced into the system. The error associated with setting the higher order delta terms equal to zero can be calculated by subtracting exact perturbation theory from first order perturbation theory. In the equations below, the subscript 1 is for the unperturbed case and the subscript 2 is for the perturbed case.

$$E_{FOPT} = \frac{\left\langle \varphi_1^* B \right\rangle \left\langle \delta\varphi_{2\to1}^* A \right\rangle - \left\langle \varphi_1^* A \right\rangle \left\langle \delta\varphi_{2\to1}^* B \right\rangle}{\left\langle \varphi_1^* B \right\rangle^2} \left(1 - \frac{\left\langle \delta\varphi_{2\to1}^* B \right\rangle}{\left\langle \varphi_1^* B \right\rangle} + \left(\frac{\left\langle \delta\varphi_{2\to1}^* B \right\rangle}{\left\langle \varphi_1^* B \right\rangle} \right)^2 - \ldots \right) for \left\langle \delta\varphi_{2\to1}^* B \right\rangle < \left\langle \varphi_1^* B \right\rangle$$

[37]

$\varphi^o{}_1$ = Unperturbed Forward flux
φ_1^* = Unperturbed Adjoint flux
φ_2^* = Perturbed Adjoint flux
$\delta T_{2\to1} = T_2 - T_1$
$\delta P_{2\to1} = P_2 - P_1$
$\delta\varphi_{2\to1}{}^* = \varphi_2^* - \varphi_1^*$

For simplification, two more operators A and B are defined as:

$$A = \left(\delta T_{2\to1} - \lambda \delta P_{2\to1} \right) \phi_1$$
$$B = \left(P_1 + \delta P_{2\to1} \right) \phi_1$$

[38]

Substituting A and B into Equation 37 the following equation is obtained:

$$\mathrm{E}_{FOPT} = \frac{\left\langle \varphi^{*}B \right\rangle \left\langle \delta\varphi_{2\to1}{}^{*}A \right\rangle - \left\langle \varphi^{*}A \right\rangle \left\langle \delta\varphi_{2\to1}{}^{*}B \right\rangle}{\left\langle \varphi^{*}B \right\rangle \left\langle \varphi^{*}B \right\rangle + \left\langle \varphi^{*}B \right\rangle \left\langle \delta\varphi_{2\to1}{}^{*}B \right\rangle}$$

[39]

The denominator of Equation 39 can be expanded in a Maclaurin series series about

$$\frac{\left\langle \delta\varphi_{2\to1}^{*}B \right\rangle}{\left\langle \varphi_{1}^{*}B \right\rangle}.$$

$$E_{FOPT} = \frac{\left\langle \varphi_{1}^{*}B \right\rangle \left\langle \delta\varphi_{2\to1}^{*}A \right\rangle - \left\langle \varphi_{1}^{*}A \right\rangle \left\langle \delta\varphi_{2\to1}^{*}B \right\rangle}{\left\langle \varphi_{1}^{*}B \right\rangle^{2}} \left(1 - \frac{\left\langle \delta\varphi_{2\to1}^{*}B \right\rangle}{\left\langle \varphi_{1}^{*}B \right\rangle} + \left(\frac{\left\langle \delta\varphi_{2\to1}^{*}B \right\rangle}{\left\langle \varphi_{1}^{*}B \right\rangle} \right)^{2} - \ldots \right) for \left\langle \delta\varphi_{2\to1}^{*}B \right\rangle < \left\langle \varphi_{1}^{*}B \right\rangle$$

[40]

If the change in the system is relatively far from the fuel such that the change in angular flux within the fuel can be neglected then the error is reduced to:

$$E_{FOPT} = \frac{\left\langle \delta\varphi_{2\to1}^{*} \left(\delta T_{2\to1} \right) \varphi_{1} \right\rangle}{\left\langle \varphi_{1}^{*} \left(P_{1} \right) \varphi_{1} \right\rangle} \quad [41]$$

From Equation 41 it can be inferred that the error introduced with FOPT is related to the change in cross section multiplied by the change in the flux normalized to the fission rate. As long as either the change in cross section or change in angular and energy dependent flux is relatively small then the error from FOPT should also be relatively small. The only variable that is not known is the difference between the perturbed flux and the unperturbed flux for the region of interest. If there were a way to approximate the change in the flux for the non-fueled region of interest and if this change could be incorporated into the equation, then the error would be greatly reduced. PSIPT attempts to approximate the perturbed angular flux without having to solve additional eigenvalue calculations. This is accomplished by solving the unperturbed forward eigenvalue calculation just like first order perturbation theory, but then by solving the adjoint eigenvalue calculation for a bounding case. The bounding case is the expected maximum change the reactor system will undergo during the prediction calculations.

For PSIPT, just as in FOPT and exact perturbation theory, the aim is to calculate the change in the eigenvalue between the perturbed and unperturbed solution. Therefore, PSIPT works best when the system parameters to be perturbed are known beforehand. The perturbed adjoint φ_2^* can be written as a function of the bounding adjoint and the difference between the bounding and perturbed adjoint:

$$\varphi_2^* = \varphi_3^* - \delta\varphi_{3\to 2}^* \; [42]$$

The subscript 3 represents the bounding case that is specific to PSIPT. Substituting Equation 42 in for the perturbed adjoint in the equation for exact perturbation theory, Equation 9, and rearranging results in the following.

$$\delta\lambda_{2\to1} = \frac{\left\langle \left(\varphi_3^* - \delta\varphi_{3\to2}^*\right)\left(T_{2\to1} - \lambda_1 \delta P_{2\to1}\right)\varphi_1 \right\rangle}{\left\langle \left(\varphi_3^* - \delta\varphi_{3\to2}^*\right) P_2 \varphi_1 \right\rangle} \quad [43]$$

The assumption that any higher order delta term $\delta x \delta y$ is sufficently small to be approximated as zero results in the following.

$$\delta\lambda_{2\to1} = \frac{\left\langle \varphi_3^* \left(\delta T_{2\to1} - \lambda_1 \delta P_{2\to1}\right)\varphi_1 \right\rangle}{\left\langle \varphi_3^* P_2 \varphi_1 \right\rangle} \quad [44]$$

This is the equation used for PSIPT to calculate the change in reactivity of the perturbed and unperturbed eigenvalue. The error of PSIPT is calculated by subtracting exact perturbation theory from PSIPT:

$$E_{PSIPT} = \frac{\left\langle \varphi_2^* \left(\delta T_{2\to1} + \lambda_1 \delta P_{2\to1}\right)\varphi_1 \right\rangle}{\left\langle \varphi_2^* \left(P_2\right)\varphi_1 \right\rangle} - \frac{\left\langle \varphi_3^* \left(\delta T_{2\to1} + \lambda_1 \delta P_{2\to1}\right)\varphi_1 \right\rangle}{\left\langle \varphi_3^* \left(P_2\right)\varphi_1 \right\rangle} \qquad [4]$$

E_{PSIPT} = Error from using PSIPT

The expanded form of the error of SCPT is:

$$E_{PSIPT} = \frac{\left\langle \varphi_2^* P_2 \varphi_1 \right\rangle \left\langle \delta\varphi_{3\to2}^* \left(\delta T_{2\to1} + \lambda_1 \delta P_{2\to1}\right)\varphi_1 \right\rangle - \left\langle \delta\varphi_{3\to2}^* P_2 \varphi_1 \right\rangle \left\langle \varphi_2^* \left(\delta T_{2\to1} + \lambda_1 \delta P_{2\to1}\right)\varphi_1 \right\rangle}{\left\langle \left(\varphi_2^* + \delta\varphi_{3\to2}^*\right) P_2 \varphi_1 \right\rangle \left\langle \varphi_2^* P_2 \varphi_1 \right\rangle} \quad [46]$$

Where

$\delta\varphi^*_{3\to2} = \varphi^*_3 - \varphi^*_2$ [47]

Assuming that there is no perturbation in the fuel and that $\delta\varphi^*{}_{2>3}$ in the fuel is small enough that it does not significantly contribute to the overall results, then the following assumptions can be made

$\left\langle \delta\varphi^*_{3\to2} P_2 \varphi_1 \right\rangle = 0$ [48]

$\delta P_{2\to1} = 0$ [49]

$\left\langle \left(\varphi^*_2 - \delta\varphi^*_{3\to2}\right) P_2 \varphi_1 \right\rangle = \left\langle \varphi^*_2 P_2 \varphi_1 \right\rangle$ [50]

Then the error can reduce down to.

$E_{PSIPT} = \dfrac{\left\langle \delta\varphi^*_{3\to2} \left(\delta T_{2\to1} + \lambda_1 \delta P_{2\to1}\right) \varphi_1 \right\rangle}{\left\langle \varphi^*_2 P_2 \varphi_1 \right\rangle}$ [51]

When comparing the error of PSIPT to the error of first order perturbation theory there are two differences. First, the error of PSIPT is related to the difference between the perturbed flux and the bounding flux whereas for FOPT the error is the difference between the perturbed flux and the unperturbed flux. The second difference is that PSIPT is normalized by the inner product of the perturbed adjoint, unperturbed forward, and the perturbed fission operator, whereas the error of FOPT is normalized by the unperturbed adjoint instead of the perturbed adjoint.

It is desirous to know when the error from PSIPT will be less then the error from FOPT. The error of FOPT is only zero when there is no perturbation. However, the error for PSIPT is zero at the two endpoints, i.e. if perturbed state 2 equals unperturbed state 1 or bounding state 3. Because the error of PSIPT is zero at the two end points and the error of FOPT is zero only at one end point, there exists a region within the phase space of intermediate perturbations when the error of PSIPT will be less then the error of FOPT. The region where the error from PSIPT is smaller then the error of FOPT can be calculated by following inequality associated with Equation 41 and Equation 51.

$$\frac{\left\langle \delta\varphi^*_{3\to2}\left(\delta T_{2\to1}+\lambda_1\delta P_{2\to1}\right)\varphi_1\right\rangle}{\left\langle \delta\varphi^*_{2\to1}\left(\delta T_{2\to1}\right)\varphi_1\right\rangle} < \frac{\left\langle \varphi^*_2 P_2 \varphi_1\right\rangle}{\left\langle \varphi^*_1 P_2 \varphi_1\right\rangle} \quad [52]$$

With the assumption that the change in angular adjoint flux in the fuel is insignificant then the term on the left hand side of the equation can be set equal to one. This reduces the equation to:

$$\left\langle \delta\varphi^*_{3\to2}\left(\delta T_{2\to1}+\lambda_1\delta P_{2\to1}\right)\varphi_1\right\rangle < \left\langle \delta\varphi^*_{2\to1}\left(\delta T_{2\to1}\right)\varphi_1\right\rangle \quad [53]$$

From the equation above it can be inferred that for large changes to the perturbed and unperturbed angular adjoint flux the error from FOPT will be much greater then the error from PSIPT. Moreover, for relatively small changes to the angular adjoint flux the error from FOPT would be less than that of PSIPT.   Therefore, as long as the difference between the perturbed flux and the bounded flux is smaller than the difference between the perturbed flux and unperturbed flux, PSIPT will provide results that are more accurate then FOPT.

Phase space interpolated perturbation theory can only be used when the type of change to the reactor is already known, bounding the magnitude of that change can be estimated, and intermediate changes are to be investigated. This new method should work well with shim rotation prediction and control rod removal but would not work well for calculating the reactivity associated with experimental placements, for example, because the type of experiment is not known a priori. Examples of the implementation of phase space interpolated perturbation theory not related to shim rotation prediction can be found in Appendix A

Phase space interpolated perturbation theory was tested on a simplified model to determine if it would perform as expected. The simplified model is the same model as discussed in the previous chapter (see Figure 11 and associated discussion)

Phase space interpolated perturbation theory requires the forward solution to be run at the initial guess but the adjoint solution is run for a range of perturbations being analyzed.

The OSCC drums could be rotated either toward the reactor or away from the reactor without a priori knowledge beforehand. This would require two adjoint calculations instead of just one. The forward and adjoint solutions would be run simultaneously with one adjoint calculation with shims rotating away from the initial starting point and the other adjoint calculation the shims rotating towards the initial starting point as seen in the equations below.

$$x_{i\ away}^{*} = x_i + STEP \quad [54]$$

$$x_{i\ toward}^{*} = x_i - STEP \quad [55]$$

x_i^* = Shim position for the adjoint calculation [Degree]

In order to reduce the total number of eigenvalue calculations from three, as described above, to two, a modified form of phase space interpolated perturbation theory was used with shim rotation prediction. This was accomplished by having the forward calculation still at the initial guess but then the adjoint calculation would also be as the initial guess. However, the difference between the forward and adjoint calculation would be a theoretical bounding guess only for the adjoint in both directions and for both materials, (Be and HF), See Figure 17. In other words, for the single adjoint equation a portion of the hafnium region would be assigned beryllium and a portion of the beryllium region would be assigned hafnium. This would then allow the prediction and automation of shim rotation in both directions.

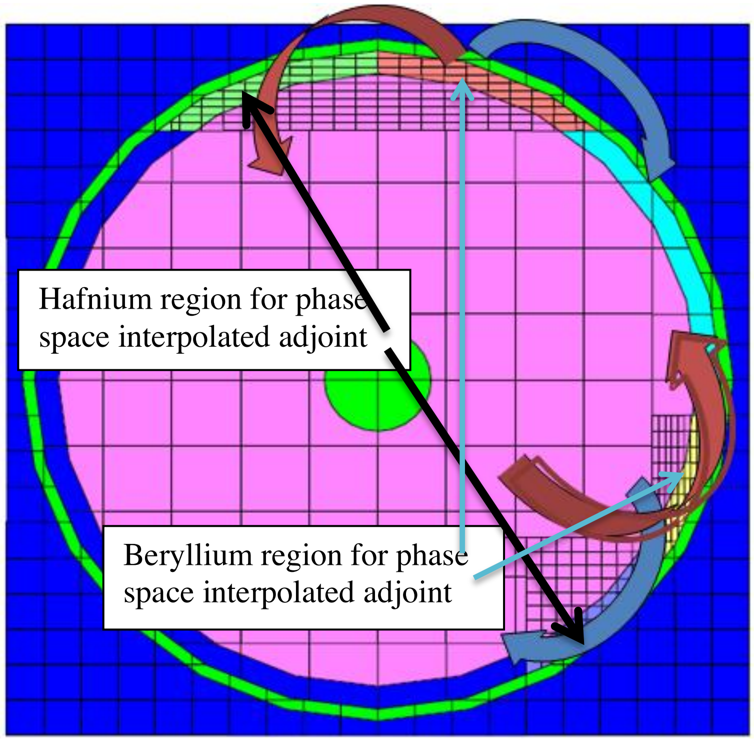

Hafnium region for phase space interpolated adjoint
Beryllium region for phase space interpolated adjoint

Figure 17 Assignment of material for the phase space interpolated adjoint

The results from phase space interpolated perturbation theory obtained from the modified method were compared against the forward eigenvalue calculations and first order perturbation theory. This was to verify that the modified form of phase space interpolated perturbation theory can accurately estimate the change in reactivity of the model as a function of shim rotation, and that phase space interpolated perturbation theory was an improvement to first order perturbation theory. To provide reference data, forward eigenvalue calculations were performed for shim rotation in five-degree increments starting at 31.2 degrees and ending at 71.2 degrees. The initial unperturbed forward eigenvalue solution used with perturbation theory occurred at 51.2 degrees, which was exactly in between the minimum and maximum modeled shim rotation. The adjoint solution also for phase space interpolated perturbation theory used 51.2 degrees as a starting point but assigned 30 degrees of hafnium to beryllium and 30 degrees of beryllium to hafnium. The adjoint solution for FOPT used 51.2 degrees as a starting point with no changes to the hafnium or beryllium region. The results from the eigenvalue calculations, phase space interpolated perturbation theory, and first order perturbation theory calculations are plotted in Figure 18. It took a total of nine eigenvalue forward calculations to obtain the exact reactivity results, but it only took two eigenvalue calculations to obtain the results from both first order perturbation theory and from phase space interpolated perturbation theory.

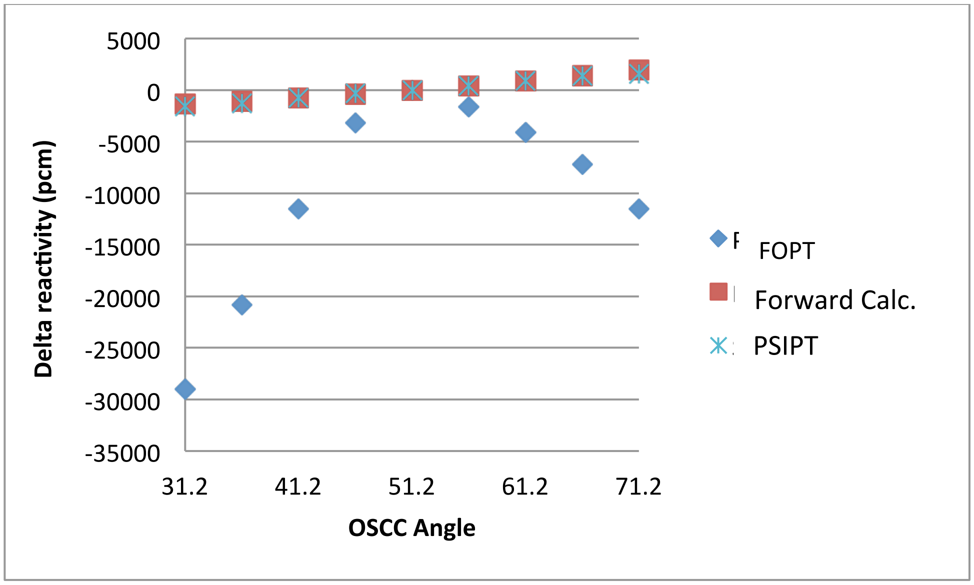


Figure 18 Shim rotation calculations with phase space interpolated perturbation theory compared against forward solutions and first order perturbation theory

As seen above, the results from phase space interpolated perturbation theory line up very well to the results obtained by forward eigenvalue solutions, especially for results close to the unperturbed data, 51.2 degrees. A plot showing the error from the modified phase space interpolated perturbation theory can be seen in Figure 19. The largest error for phase space interpolated perturbation theory was around 375 pcm at 71.2 degrees with an average error of around 260 pcm. An initial guess less than 15 degrees off would allow the new forward calculation to be within 150 pcm of the true value. An initial guess larger than 15 degrees from the final solution would require at least one more run of perturbation theory with new forward and adjoint values before it could converge within a desired tolerance of 150 pcm.

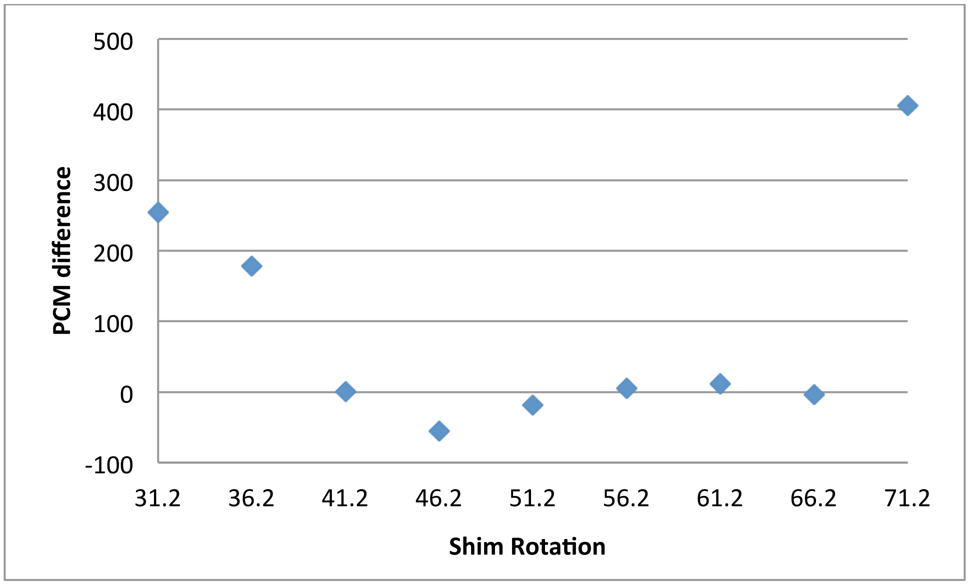


Figure 19 Errors associated with phase space interpolated perturbation theory

For standard phase space interpolated perturbation theory, the error will be the smallest at the bounding range. This is because only a small error is introduced at the minimum condition and for the maximum range phase space interpolated perturbation theory approaches exact perturbation theory

For the modified form of phase space interpolated perturbation theory, that was used with shim rotation prediction, the adjoint solution is at the same location as the forward solution but the materials for the adjoint solution are swapped out for both rotational directions. The problem associated with this type of phase space interpolated perturbation theory is that the solution does not converge towards exact perturbation theory. This results in a more accurate solution near the initial guess and a less accurate solution away from the initial guess. The error associated with this method would be more of a "V" shaped instead of the inverted V shape. This "V" shape can be observed in Figure 19.

This simple comparison demonstrated that phase space interpolated perturbation theory could be used to help predict shim rotation for a desired eigenvalue. Additional fine tuning calculations were used to optimize phase space interpolated perturbation theory specifically for the ATR shim rotation prediction. One optimization parameter was the bounding range of the shims for the adjoint calculation needed with the modified form of phase space interpolated perturbation theory. The geometric beryllium/hafnium ring needs to be divided in to six zones to keep the forward and adjoint meshes the same but to allow different material assignment for the adjoint solution (Figure 20). Of the six zones, one zone will always be beryllium (A), one zone will always be hafnium (D), two zones will be hafnium for the forward calculation and beryllium for the adjoint calculation (C,E), and the last two zones will be beryllium for the forward calculation and hafnium for the adjoint calculation(B,F). If a maximum shim rotation of 60 degrees was used then the shim would be broken up into five pieces with section D having an area of zero.

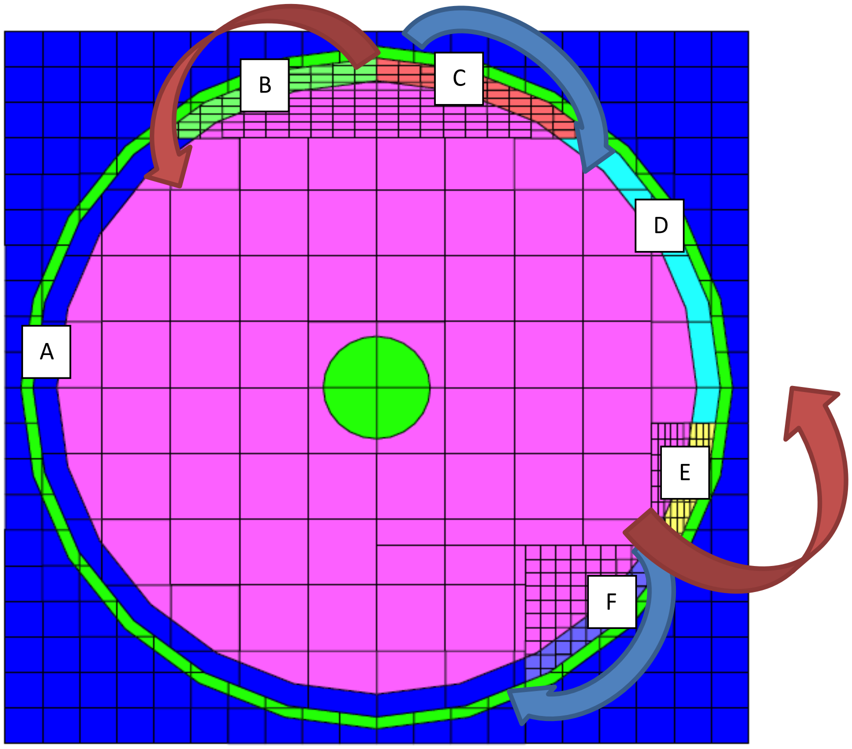


Figure 20 The zone meshing needed for phase space interpolated perturbation theory

One of the difficulties for phase space interpolated perturbation theory specifically within NEWT is with the meshing of the shims for the predicted maximum rotation. A detailed description of how this was handled with NEWT can be found in Appendix D.

To use phase space interpolated-perturbation theory with shim rotation the user needs to know what range the shims will need to be rotated. Two different ranges were analyzed to establish the fidelity tradeoff between rotation ranges covered by two adjoint pairs. The two ranges were 15 degrees and 30 degrees.

To compare between the two ranges, the shims were initially placed at a rotation of 51.2 degrees. Phase space interpolated perturbation theory was used to predict the reactivity for five degree incremental changes. For the 15-degree range rotations 41.2, 46.2 56.2 and 61.2 degrees were analyzed. For the 30-degree range incremental rotations 26.2 to 76.2 degrees were analyzed in increments of five degrees. For both degree ranges the normal equation was used to best fit a line to the data so that the reactivity could be extrapolated beyond the rotated ranges. The extrapolated data is required when the predicted value for shim rotation is beyond the 15 or 30 degree range increments. The plots of exact reactivity calculated with a forward eigenvalue calculation and data for the 15 and 30 degree ranges can be seen in Figure 21.

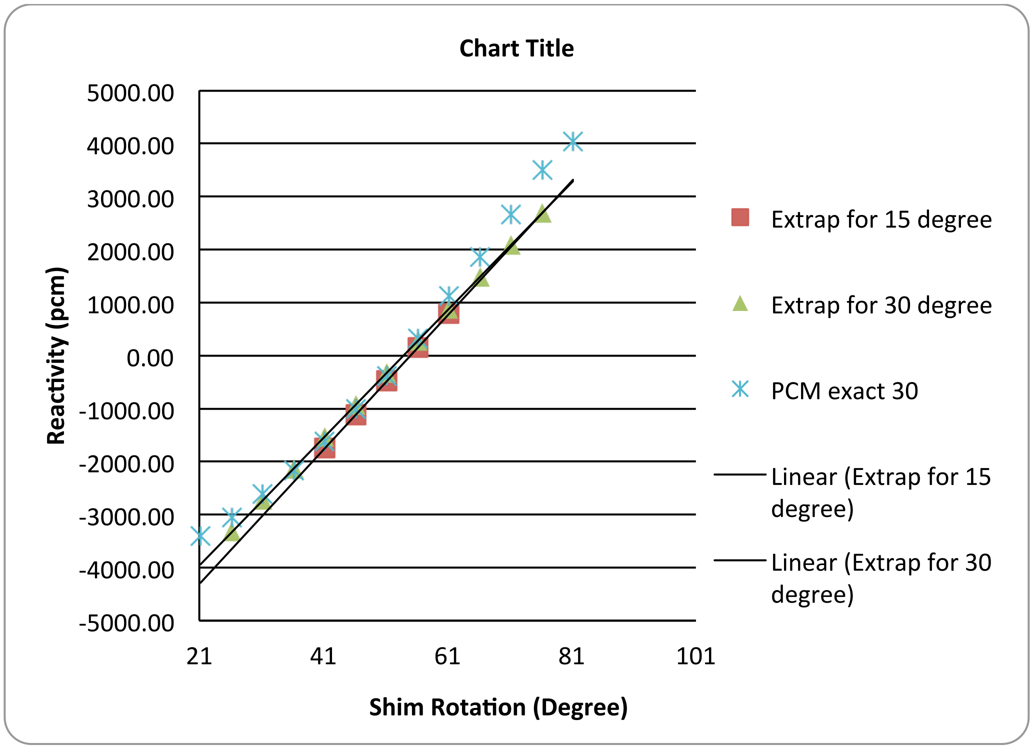


Figure 21 Exact reactivity versus predicted reactivity for a range of 15 and 30 degrees

In Figure 21, the extrapolated equation obtained from the 30-degree range outperformed the extrapolated data for the 15-degree range in the lower ranges of shim rotation. Both of the ranges appeared to give a similar results around 64 degrees to 81 degrees. From this simple study, it appears that a shim rotation range of 30 degrees provides the better fit for this specific example but both ranges would have worked relatively well. It also appears from this study that the results were not sensitive to the range, at least within the 15 to 30 degree window used for phase space interpolated perturbation. Shim rotation range of 30 degrees was used for all of the proceeding results.

Implementation

STEP 1 WITH PHASE SPACE INTERPOLATED PERTURBATION THEORY

The previous computational study demonstrated that phase space interpolated perturbation theory would work well for shim rotation prediction of the ATR with an adjoint range of ± 30 degrees. The goal of the phase space interpolated perturbation theory in this application is to predict the OSCC shim locations needed for startup so that shim rotations can be automated. A flow chart of the two needed steps for shim rotation automation at startup can be seen in Figure 22

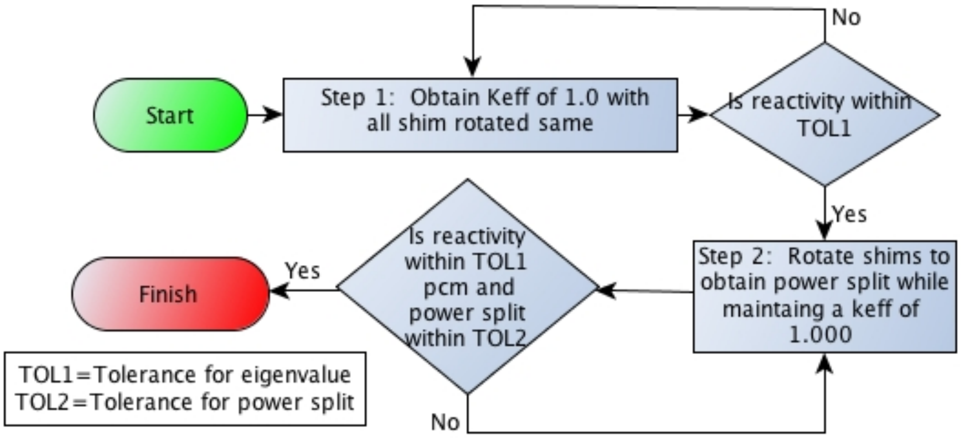

Start
Step 1: Obtain Keff of 1.0 with all shim rotated same
Is reactivity within TOL1
No
Yes
Step 2: Rotate shims to obtain power split while maintaing a keff of 1.000
Is reactivity within TOL1 pcm and power split within TOL2
No
Yes
Finish
TOL1=Tolerance for eigenvalue
TOL2=Tolerance for power split

Figure 22 ATR reactor physics calculations

The first step is to calculate the uniform shim rotation needed for an eigenvalue of one. The flowchart showing the process for step1 of phase space interpolated perturbation theory can be seen in Figure 23.

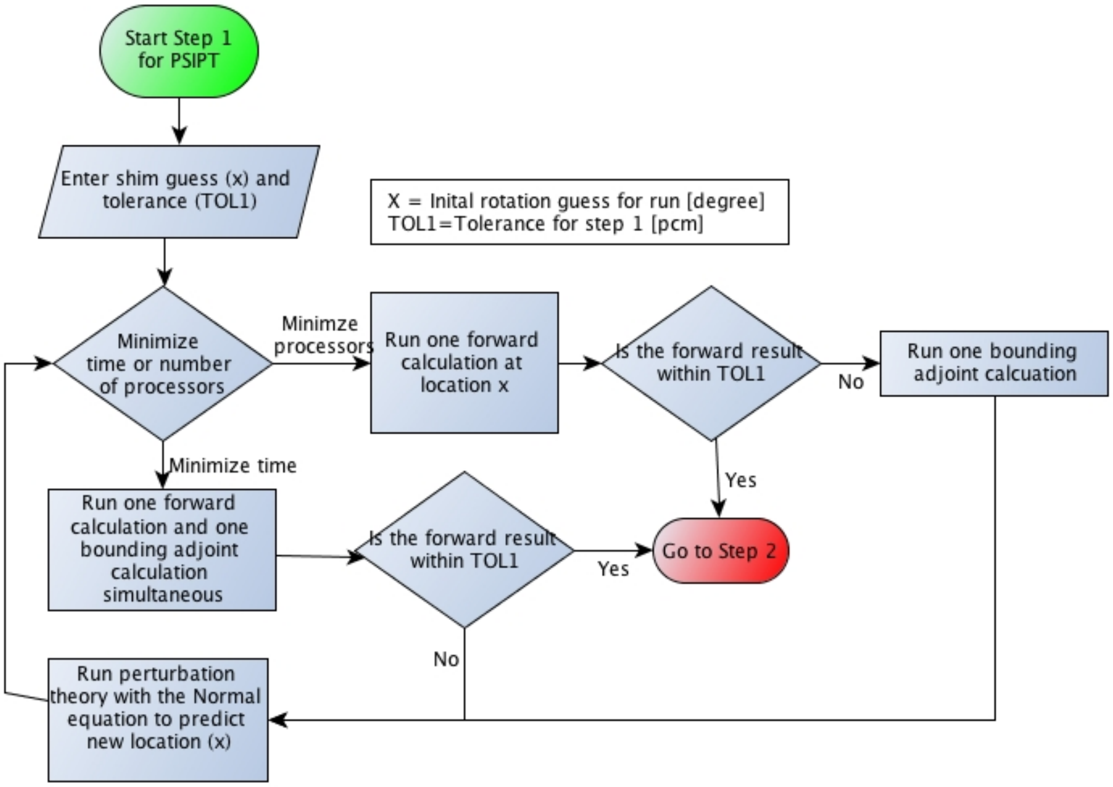

Start Step 1 for PSIPT
Enter shim guess (x) and tolerance (TOL1)
X = Inital rotation guess for run [degree]
TOL1=Tolerance for step 1 [pcm]
Minimize time or number of processors
Minimze processors
Run one forward calculation at location x
Is the forward result within TOL1
No
Run one bounding adjoint calcuation
Minimize time
Yes
Run one forward calculation and one bounding adjoint calculation simultaneous
Is the forward result within TOL1
Yes
Go to Step 2
No
Run perturbation theory with the Normal equation to predict new location (x)

Figure 23 Step 1 of shim rotation prediction

The procedure begins with a guess (X) in degrees of the critical shim position. The shim rotation guess is then inserted into the model and the forward and adjoint solutions are run simultaneously or in series depending on whether time or the number of processors are to be minimized.

Unlike first order perturbation theory, phase space interpolated perturbation theory does not give rise to responses that are linear in the perturbations. Hence, a learning algorithm is used to speed up the process of predicting the correct shim position. Note that in this model implementation of shim rotation can only be used to predict discrete steps that are dependent on the size of the mesh for the model (See Appendix C for further discussion) This makes it impossible to obtain exactly zero for reactivity prediction. Therefore, a tolerance (TOL1), in units of pcm, is needed for the computer algorithm to know it has converged within the desired result. A tolerance of around 150 pcm or, 25 cents, is adopted in all of the following calculations. Convergence to the tolerance is dependent on the size of the mesh used for the shim positioning. The finer the mesh within the shims the tighter the convergence criteria can be, but the longer forward and adjoint solutions will take to converge.

The learning algorithm used for shim rotation prediction with perturbation theory is a supervised regression algorithm called the "Normal Equation," as seen below in a matrix format.

$$\vec{\theta} = \left(X^T X\right)^{-1} X^T \vec{y} \qquad [56]$$

$$h_\theta\left(x\right) = \theta_0 x_0 + \theta_1 x_1 \qquad [57]$$

$$X = \begin{bmatrix} x_0^{(1)} \; x_1^{(1)} \\ x_0^{(1)} \; x_1^{(2)} \\ x_0^{(1)} \; x_1^{(3)} \end{bmatrix}, \vec{y} = \begin{bmatrix} y^{(1)} \\ y^{(1)} \\ y^{(1)} \end{bmatrix} \quad [58]$$

$\vec{\theta}$ = an array of regression coefficent
X=matrix of independent variables
$\vec{y}$=an array of reactivity results [pcm]
$\mathrm{h}_{\theta}\left(x\right) =$ fitted equation used to approximate reactivity (y) as a function of shim rotation [pcm]
x_n^m
n=number of input variables
m=number of training examples
$x_1 =$ Degree for shim rotation
$x_0 = 1$
$y =$ Reactivity

The learning algorithm uses the method of least squares to approximate an equation to a line that will best represent the reactivity as a function of shim rotation. A cubic fit was also used with phase space interpolated perturbation theory, but the extrapolated results from the cubic fit were unrealistic for the first five or six predictions and then would slowly converge on a better curve fit. It was therefore decided that the linear fit with the normal equation provided the best results in the shortest number of iterations.

Once the perturbation theory results had converged on approximately zero, the new shim positions were used to run a confirmatory forward calculation. If time was minimized then an adjoint calculation would be run concurrently with the confirmatory forward calculation in case an addition iterative perturbation calculation was needed. For the processor minimization problem the adjoint calculation would only be run if the forward solution was not within the desired tolerance.

Upon completion of the forward eigenvalue calculation, the results were analyzed to see if the reactivity was within the desired tolerance. If the desired tolerance was not reached then perturbation theory was used with the new forward and adjoint solutions to predict the new shim rotation. If the results were within the needed tolerances then it is on to the second step for shim rotation prediction for reactor startup. The minimum number of eigenvalue calculations needed for this method is three of which the first two can be run concurrently if time is minimized.

Step 2 with Phase Space Interpolated Perturbation Theory

Step 2 for shim rotation prediction requires the combination of linear regression analysis in addition to perturbation theory. Phase space interpolated perturbation theory, which is based upon FOPT, can only be used for predicting changes in reactivity. The perturbation theory used for predicting power splits is known as generalized perturbation theory (GPT). There are two main problems with GPT that make it less optimal to be used for power splits prediction than the linear regression method. The first problem is that in addition to the forward and adjoint calculations needed for calculating perturbation theory, four generalized adjoint calculations are needed to predict the power ratios in the four lobes. This would require a total of six eigenvalue calculations before being able use generalized perturbation theory to predict shim rotation as a function of the power splits. The second issue with GPT is that first order generalized perturbation theory can only be used for small changes in the reactor system. As demonstrated in the study associated with first order perturbation theory, shim rotation is not considered a small change. To get accurate results with GPT, higher order method GPT would need to be used. For these two reasons, it was decided to use a linear regression method in conjunction with phase space interpolated perturbation theory. The theory and implementation of the linear regression method for power splits is discussed in detail in the next chapter entitled "Cubic Regression Method".

For step 2, the shims need to be rotated for the desired power split while maintaining an eigenvalue of 1.00000. In practice, the power splits are considered acceptable if they are within 3 MW for each of the 5 lobes. Achieving a power split convergence to within 1MW, or around 2% for a 50 MW lobe, would ensure an engineering factor of safety of around 3. As in step 1, either time or the number of processors can be minimized. Only the time minimization path for step 2 is shown in the flow chart in Figure 24.

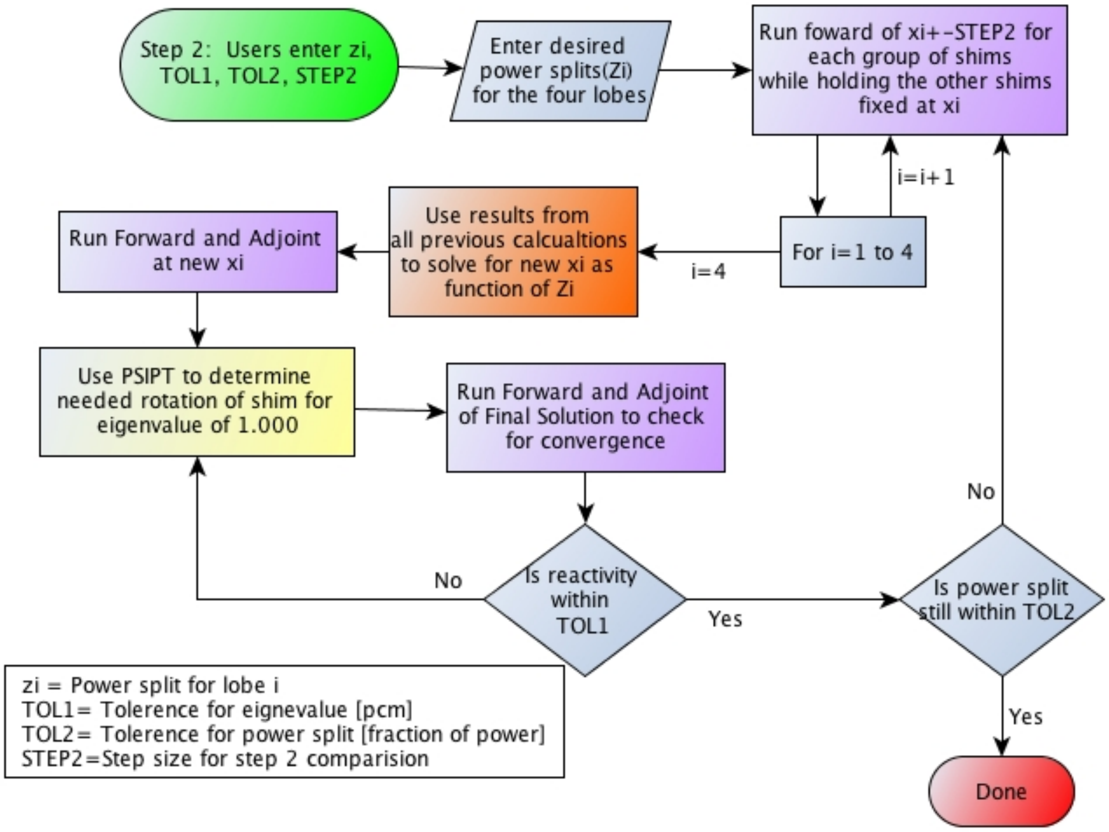

Step 2: Users enter zi, TOL1, TOL2, STEP2
Enter desired power splits(Zi) for the four lobes
Run foward of xi+-STEP2 for each group of shims while holding the other shims fixed at xi
i=i+1
For i=1 to 4
i=4
Use results from all previous calcualtions to solve for new xi as function of Zi
Run Forward and Adjoint at new xi
Use PSIPT to determine needed rotation of shim for eigenvalue of 1.000
Run Forward and Adjoint of Final Solution to check for convergence
No
Is reactivity within TOL1
Yes
No
Is power split still within TOL2
Yes
Done
zi = Power split for lobe i
TOL1= Tolerence for eignevalue [pcm]
TOL2= Tolerence for power split [fraction of power]
STEP2=Step size for step 2 comparision

Figure 24 Tasks to predict shim rotation for step 2

The shim position (xi) obtain from step1 is used as the initial starting point. Shim positions 1, 2, 3, and 4 correspond to the NW, NE, SW, SE shims respectively. The first task for step 2 is for the user to enter the desired percent power split (Zi), the tolerances for reactivity (TOL1) and power split (TOL2), and a recommendation on the step size for calculating the shim power splits (STEP). In Step2, when time is optimized the maximum number of concurrent calculations is nine.

The first part of Step2 is to run forward solutions of xi± STEP for each group of shims while holding the other three shims fixed. If number of processors is minimized then only one shim is rotated away from the reactor if the power split is greater than 25% and toward the reactor if the power split is less than 25% reducing the total number of forward calculations for the initial process from eight to four.

Once the forward runs have finished then the normal equation is used to calculate the needed rotation for the desired power split. A detailed description of the process and theory behind the normal equation can be found in the "Cubic regression" chapter. Once the new shim positions (xi) for the desired power split (Zi) has been calculated then a forward and adjoint calculation with NEWT is run with the new predicted shim location. The results for the forward and adjoint calculations are then used with perturbation theory to predict the needed uniform change in shim rotation to obtain the desired eigenvalue of 1.00000.

A final solution is then run for both forward and adjoint to determine if the solution has converged. If the reactivity is within the desired tolerance then the power split tolerance is checked. If the reactivity is not within the desired tolerance then perturbation theory is used with the final forward and adjoint solution to predict a more accurate shim position. If the power split is not within the desired tolerance then the process is started from the beginning with the final solution being the new starting point (xi) and the STEP size being reduced by half.

Validation of Results

The validation and verification process requires demonstration that for a large range of user inputs the results converge on the correct solution.

STEP 1 WITH PHASE SPACE INTERPOLATED PERTURBATION THEORY

This example verifies the results of automation of shim rotation prediction with phase space interpolated perturbation theory for step 1. In this example an initial guess of 46.2 degrees (xi), an eigenvalue tolerance of 150 pcm (TOL1) were set. For the initial guess a forward and adjoint solution were simultaneously obtained. Both calculations had the shims rotated at 46.2 degrees; however, the adjoint eigenvalue swapped hafnium for beryllium and beryllium for hafnium in two 30 degree arcs on either side of the hafnium beryllium interface (See Figure 20). This allowed for a range of 30 degrees of rotation in both the clockwise and counterclockwise directions.

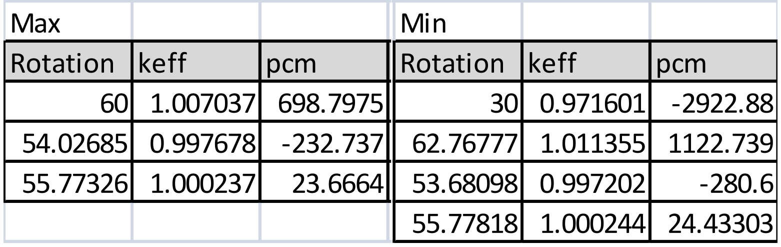
Upon completion of the two eigenvalue calculations, a Python script ran PertN in conjunction with the normal equation to determine the shim rotation to obtain an eigenvalue of 1.00000. The script predicted a shim rotation of 55.5 degrees and ran

NEWT to solve for the forward and adjoint solution at the new predicted shim rotation to confirm that the results were within the desired tolerance.

The forward solution from NEWT for the shim rotation of 55.5 had an eigenvalue of .99977 which was around 22 pcm from critical. Because the new result was within the specified tolerance of 150 pcm the next process was to go to Step 2. Only two sequential runs were needed for this scenario.

The automated shim rotation prediction using phase space interpolated perturbation theory was also tested for the maximum and minimum scenario of 90 and 0.0 degrees respectively. The maximum possible shim rotation is 159.9 degrees. However, the startup procedures have an administrative limit of maintaining the OSCC between 80 and 100 degrees until all neck shims have been withdrawn [SAR- 153]. Given built in bounding values of 30 degrees in PertN, the bounding scenarios chosen were 30 and 60 degrees. It took a total of three and four forward and adjoint runs for the maximum and minimum scenario respectively to converge on a solution that was within 150 pcm of the true value. The results from the convergence study can be seen in Table 5.

Table 5 Maximum and minimum convergence study



Max			Min		
Rotation	keff	pcm	Rotation	keff	pcm
60	1.007037	698.7975	30	0.971601	-2922.88
54.02685	0.997678	-232.737	62.76777	1.011355	1122.739
55.77326	1.000237	23.6664	53.68098	0.997202	-280.6
			55.77818	1.000244	24.43303

From the results obtained from the maximum and minimum case scenario it can be concluded that Step1 of shim rotation automation with phase space interpolated perturbation theory does converge to the correct solution for the given tolerance.

Once the solution has been obtained and has been shown to be within the desired tolerance the next process is Step 2, which is the calculation of the shim rotations needed to maintain criticality while obtaining the desired power splits.

STEP 2 WITH PHASE SPACE INTERPOLATED PERTURBATION THEORY

In this example the starting shim rotation (Xi) obtained from Step 1 was 55.4 degrees. The desired power split (Zi) was selected to be 23%, 24%, 26%, 27% for NW, NE, SW, SE respectively was selected with a convergence tolerance, (TOL2) of 0.2%. The step size (STEP) was chosen to initially be 15 degrees. Eight concurrent forward calculations occurred at an incremental change of ± 15 degrees. The linear regression method for the power splits predicted a shim rotation of 29.0 degrees, 46.5 degrees, 64.4 degrees, and 81.9 degrees. Forward and adjoint calculations were run for these predict shim rotation.

They corresponded to a reactivity of 2677 pcm above the desired tolerance. Therefore, perturbation theory was used to predict the new incremental shim rotation needed to obtain an eigenvalue of one, which was 25.1, 42.5, 60.4, 77.9 degrees. A confirmatory forward calculation occurred at the new predicted locations. From the confirmatory calculation, the new eigenvalue was in error by 23 pcm and the power splits were 0.233, 0.239, 0.257. and 0.271, all of which were less than the desired tolerance (TOL2)for the power split of 2%. For the example problem, Step 2 of phase space interpolated perturbation theory converged within 1 iterations for both the inner and outer calculations.

For additional verification purposes, a range of power splits must be tested for convergence. The three power splits chosen were:

- 50/50=0.25 0.25 0.25 0.25 [Fraction of total power]
- 54/46=0.23 0.24 0.26 0.27 [Fraction of total power]
- 56/44=0.22 0.23 0.27 0.28 [Fraction of total power]

The results from the three power split calculations can be seen in Table 6 and Table 7.

Table 6 Step 2 of phase space interpolated perturbation theory

Scenario	Num procs	X1[Degree]	X2[Degree]	X3[Degree]	X4[Degree]
50-50-Power	8	55.7	54.8	55.5	55.7

50-50-keff	2	55.7	54.8	55.5	55.7
54-46-Power	8	29.1	46.5	64.4	81.9
54-46-keff	2	25.1	42.5	60.4	77.9
56-44 Power	8	17.5	34.1	76.1	94.0
56-44 -keff	2	9.7	24.8	68.9	84.8

Table 7 Step 2 for phase space interpolated perturbation theory

Scenario	P1(NW)	P2	P3	P4	Final Reactivity
	fraction	(NE)	(SW)	(SE)	(pcm)
50-50-Result	0.250	0.249	0.250	0.250	24.2
50-50-Goal	0.250	0.250	0.250	0.250	-
Goal-Result	0.000	-0.001	0.000	0.000	-
54-46-Result	0.233	0.239	0.257	0.271	-23.5
54-46-Goal	0.230	0.240	0.260	0.270	
Goal-Result	0.003	-0.001	-0.003	0.001	
56-44 --Results	0.228	0.230	0.265	0.277	72.7
56-44 Goal	0.220	0.230	0.270	0.278	-
Goal-Result	0.008	0.000	-0.005	-0.001	-

All calculations performed for the three scenarios of STEP2 converged within only one outer and one inner iteration. The outer iteration required a total of eight forward calculations occurring concurrently. A predicted shim rotation for the desired power split was obtained with the eight forward calculations by using the normal equation. The inner iteration required one forward and one adjoint eigenvalue calculation. These two calculations were then used with the normal equation to fine-tune the results for converging within the specified reactivity tolerance (TOL1). A final confirmatory run was performed to ensure both the power split and eigenvalue were within the specified tolerances.

Conclusions

Phase space interpolated perturbation theory has been shown to be a valuable tool for predicting shim rotation for both step1 and step 2. Even if the user made a guess at the maximum or minimum limitations for shim rotation the solutions was able to converge within a maximum of four forward and adjoint pair iterations for step 1 and a maximum of 1 outer and inner iterations for step 2. Even though phase space interpolated perturbation theory works well for shim rotation automation, it requires the transport solver to be able to calculate both the forward and adjoint solutions to the transport equation. For transport solvers that only have the capability to solve for the forward solution it is desirable to develop a method such as cubic regression analysis that can be used to automate shim rotation prediction using only forward solutions.

STUDY 3: CUBIC REGRESSION METHOD

In the previous section, phase space interpolated perturbation theory was shown to work well for shim rotation prediction of the ATR. However, in order to implement phase space interpolated perturbation theory both the forward and adjoint solutions are needed. Obtaining the adjoint solution is relatively simple for a discrete ordinates codes such as NEWT. However, for other codes such as nodal methods or Monte Carlo codes with continuous energy cross sections obtaining the adjoint is not a simple task. When the adjoint solution cannot be obtained then other methods can be employed for shim rotation automation. One method that was chosen to be further investigated for shim rotation automation without the use of the adjoint was the cubic regression method.

Theory

The cubic regression method for automating shim rotation can be applied to all types of neutronic codes, even those that do not solve the adjoint problem. The cubic regression method uses a learning algorithm called the Normal equation to predict the appropriate shim rotations. One advantage of using the cubic regression method is that with enough computational runs it is possible to obtain a confidence interval for the results in addition to other pertinent statistical information.

As with the phase space interpolated perturbation theory method previously discussed, the goal of the cubic regression algorithm is to predict the OSCC shim locations needed for startup. A flowchart and associated discussion showing the overview of the steps needed for shim rotation automation can be seen in Figure 22 located in the previous chapter.

Implementation

STEP 1 FOR THE CUBIC REGRESSION METHOD

The cubic regression method can be optimized for two different parameters depending on the users' needs and computational capability. If the number of processors is not a concern then the cubic regression method can be optimized for time by running multiple calculations simultaneously. However, if the number of processors is a concern then the cubic regression method can be optimized to decrease the total number of forward solutions needed but will require a larger amount of time to obtain the solution. One disadvantage of the optimization process based on computational power is that the small total number of runs used to fit the data to the equation prevents a confidence interval from being obtained. Therefore, a confirmatory run is always needed for the computational minimization option to determine if the solution has converged. The algorithms for Step 1 that are used for both the minimization option based on time or number of processors can be seen in Figure 25

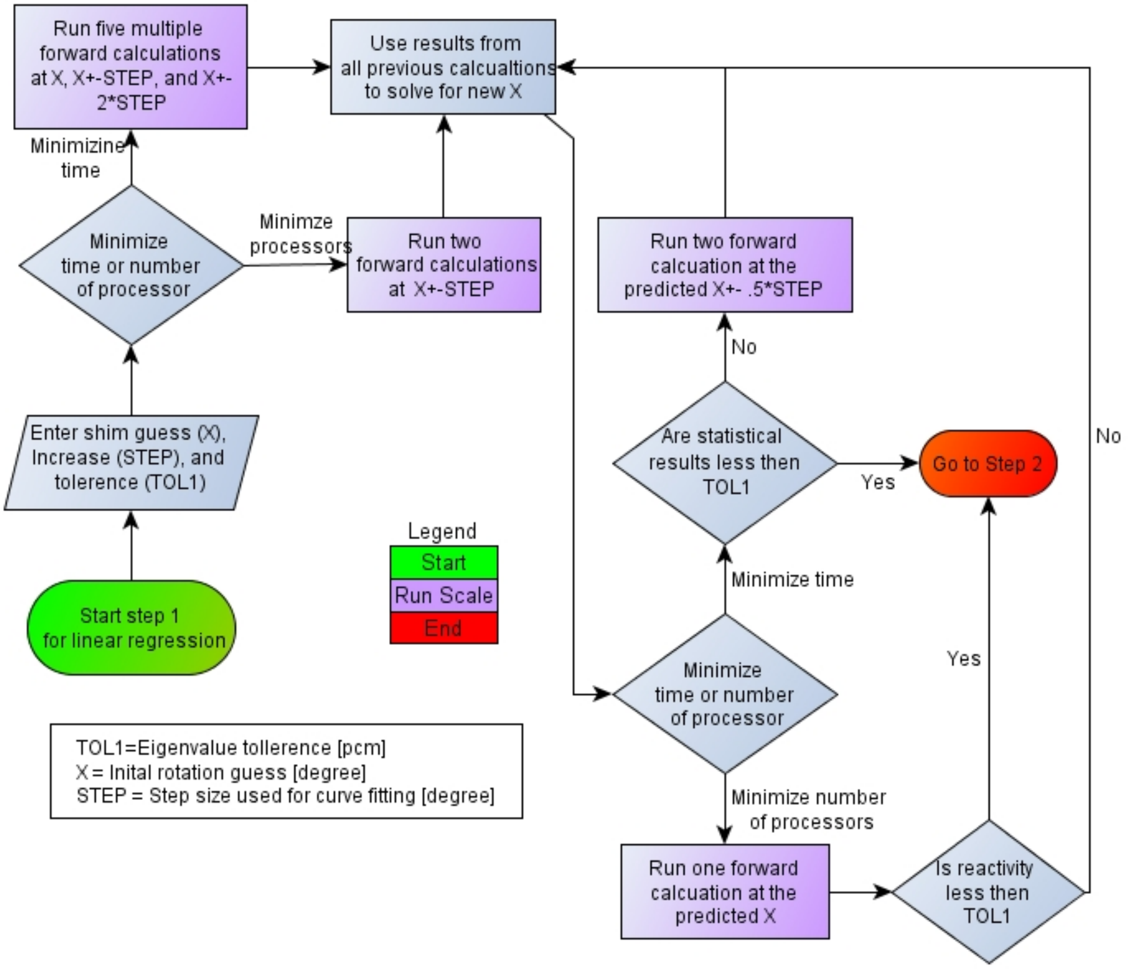

Run five multiple forward calculations at X, X+-STEP, and X+- 2*STEP
Use results from all previous calcualtions to solve for new X
Minimizing time
Minimize time or number of processor
Minimze processors
Run two forward calculations at X+-STEP
Run two forward calcuation at the predicted X+- .5*STEP
No
Are statistical results less then TOL1
Yes
Go to Step 2
No
Enter shim guess (X), Increase (STEP), and tolerence (TOL1)
Legend
Start
Run Scale
End
Start step 1 for linear regression
Minimize time
Yes
Minimize time or number of processor
Minimize number of processors
TOL1=Eigenvalue tollerence [pcm]
X= Inital rotation guess [degree]
STEP = Step size used for curve fitting [degree]
Run one forward calcuation at the predicted X
Is reactivity less then TOL1

Figure 25. Step 1 with cubic regression prediction for shim rotation prediction

The first process in step 1 of the cubic regression method is to enter three parameters: an initial shim position guess in degrees (X), a bounding shim position for the automation rotation in units of degrees (STEP), and a convergence tolerance in pcm (TOL1). If the user chose to minimize time, then five forward calculations are run simultaneously which would include the shims rotated as guessed (X) along with X± STEP and X± 2*STEP. If the user chose to minimize the number of processors, then two forward calculations are sequentially run that bound the initial guess (X±STEP).

When the forward runs are finished, the data is analyzed using the normal equation to predict the shim position. Figure 26 is a flow chart that shows the logical path used with the normal equation to automate the process in Step 1. The normal equation is a supervised machine learning algorithm. As with all supervised machine learning algorithms, multiple training examples are needed before the constants can be chosen to best approximate the relationship between shim rotation and reactivity.

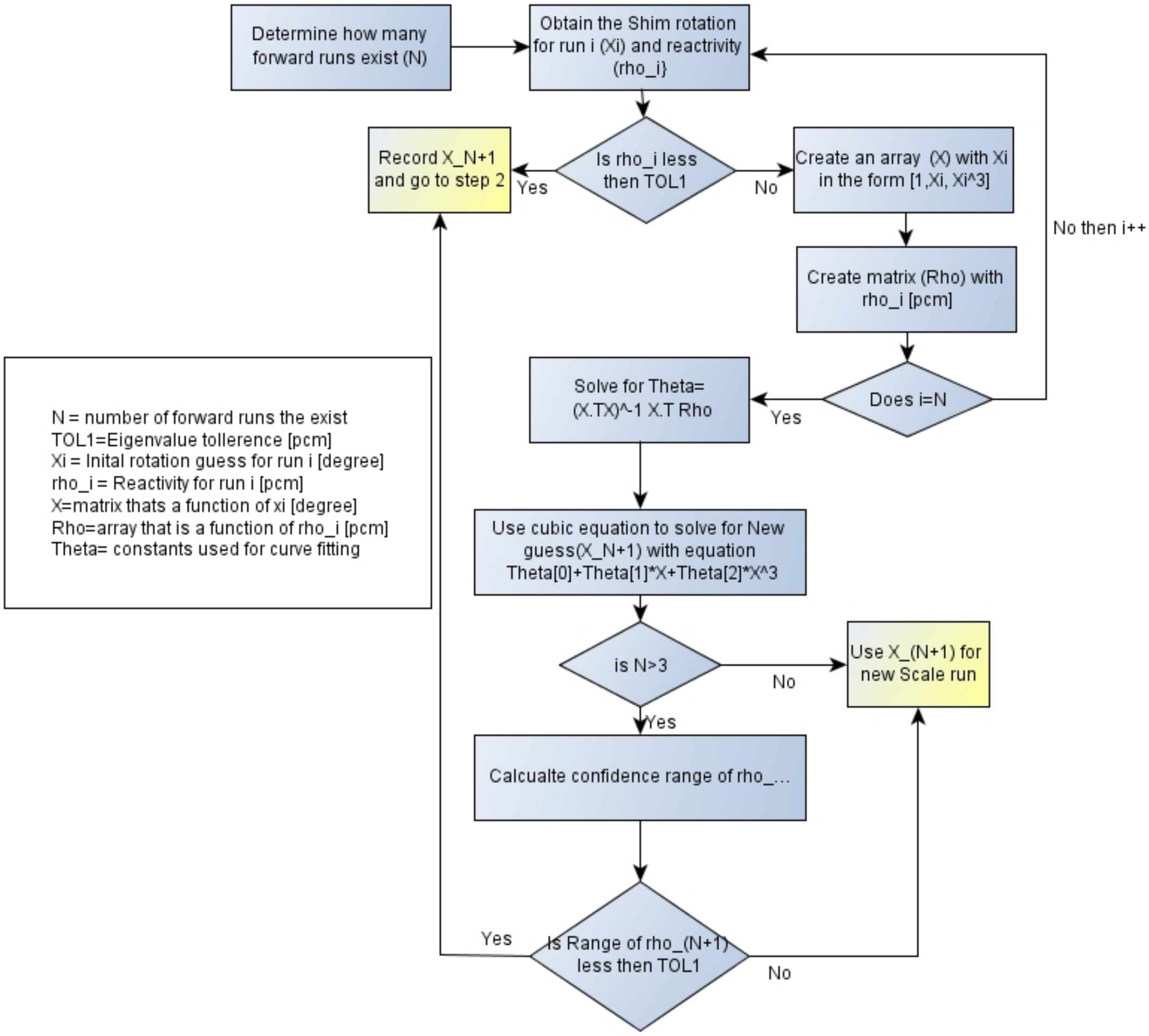

Determine how many forward runs exist (N)
Obtain the Shim rotation for run i (Xi) and reactrivity (rho_i}
Is rho_i less then TOL1
Yes
No
Record X_N+1 and go to step 2
Create an array (X) with Xi in the form [1,Xi, Xi^3]
No then i++
Create matrix (Rho) with rho_i [pcm]
Does i=N
Yes
Solve for Theta= (X.TX)^-1 X.T Rho
N = number of forward runs the exist
TOL1=Eigenvalue tollerence [pcm]
Xi = Inital rotation guess for run i [degree]
rho_i = Reactivity for run i [pcm]
X=matrix thats a function of xi [degree]
Rho=array that is a function of rho_i [pcm]
Theta= constants used for curve fitting
Use cubic equation to solve for New guess(X_N+1) with equation Theta[0]+Theta[1]*X+Theta[2]*X^3
is N>3
No
Use X_(N+1) for new Scale run
Yes
Calcualte confidence range of rho_...
Yes
Is Range of rho_(N+1) less then TOL1
No

Figure 26 Normal equation for step1

The normal equation uses a least square fitting approach, which requires selecting a function that would best represent shim rod worth. For typical control rod worth curves it can be shown mathematically that the reactivity curve can be modeled with a sigmoid function, or S shaped curve (see Equation 49). One major difference between using the sigmoid function for the ATR versus other reactors is that the lower or upper limit of the sigmoid function is not known a priori.

$$y = \frac{a}{1+e^{\left(\frac{x+b}{c}\right)}} + d \quad [59]$$

The sigmoid function would require solving four variables before being able to fit the curve exactly. This would require additional learning scenarios before the final solution would converge on the desired result. The reduction of the number of constants needed to be obtained for curve fitting can be accomplished by taking the Taylor series of the sigmoid function, which results in Equation 55.

$$y = \theta_o + \theta_1 x + \theta_2 x^3 + O\left(h^5\right) [60]$$

The number of constants that are needed to be calculated can be reduced to two or three by using either a linear or a cubit fit as seen Equation 57 and Equation 58 respectively.

$$y = \theta_0 + \theta_1 x_1 \ [571]$$

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_1^3 \quad [61]$$

$y =$ Predicted reactivity [pcm]
$x_1 =$ Shim location for all lobes[degrees]
$\theta_i =$ Constants [pcm/degree] or $\left[\text{pcm/degree}^3\right]$

The matrix format of the linear and cubic equations used with the normal equation can be seen in Equation 62 and Equation 63.

$$\vec{y} = X\vec{\theta} \quad [62]$$

$$X_{line} = \begin{bmatrix} x_0^{(1)} & x_1^{(1)} \\ x_0^{(1)} & x_1^{(2)} \\ \vdots & \vdots \end{bmatrix}, X_{cubic} = \begin{bmatrix} x_0^{(1)} & x_1^{(1)} & x_1^{2(1)} \\ x_0^{(1)} & x_1^{(2)} & x_1^{2(2)} \\ \vdots & \vdots & \vdots \end{bmatrix}, \vec{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \end{bmatrix} \quad [63]$$

$y^m =$ Reactivty for training case m
$x_0^m = 1$ for all training cases
$x_1^{(m)} =$ Shim rotation for training case m
$\vec{\theta} =$ Matrix form of the constants θ_i

The X$_{\text{line}}$ matrix is used for obtaining the parameters, or constants, for the linear equation whereas the X$_{\text{cubic}}$ equation is the matrix equation used for obtaining the parameters for the cubic equation. To obtain the best solution for the constants, Θ, the difference between the left and right hand side of Equation 62 is set equal to zero and the matrix Θ is solved as seen in Equation 64.

$$\vec{\theta} = \left(X^T X\right)^{-1} X^T \vec{y} + Error \quad [64]$$

One useful feature of using the normal equation as compared to phase space interpolated perturbation theory is that standard statistical data can be used to determine if the results are within the desired tolerance to a statistical confidence.

The first statistical test performed on the normal equation was the f-test. The f-test was used to test the regression relationship between the dependent variable and all of the independent variables. The larger the calculated value, the more useful the model. The second statistical test used was called the t-test, which can be obtained from running multiple learning scenarios and is used to determining if the fitting parameters for the linear or cubic equation are statistically significant. Unlike the f-test the t-test looks at the statistical significance of each individual parameter, instead of the entire group. The third statistical analysis was used to calculate a range for the dependent variable for a given confidence interval. The confidence interval calculation was accomplished by multiplying the standard deviation of the dependent variable by the value obtained from the t-distribution. The t-distribution is a function of the degrees of freedom for the specific problem and the confidence interval. For all of the following results a confidence interval of 0.95 was used. More detail on the math and theory behind calculating the parameters in all of the statistical tests discussed above can be found in most statistic handbooks such as the one by Netter. [Neter 1983]

The range obtained from the given confidence interval is used to determine if the results from the new shim position (x^{new}) is statistically within the desired tolerance (TOL1), or if additional runs are needed. If the reactivity range obtained from the third statistical test are outside the specified tolerance (TOL1) then two additional forward calculations are needed. The two new forward value solutions would be calculated with the following equations:

$$x^{N+1} = x^{new} + \frac{STEP}{2} \quad [65]$$

$$x^{N+2} = x^{new} - \frac{STEP}{2} \quad [66]$$

The results from these two new forward calculations in addition to all previous calculations are used with the normal equation to obtain a new prediction (x^{new}) and associated statistical results. When either the result from the forward solution or the calculated statistical range have converged within the specified tolerance (TOL1), it is on to step 2.

STEP 2 FOR SHIM ROTATION PREDICTION

In step2, the shims must be rotated to obtain specified lobe power split while maintaining an eigenvalue of 1.00000. In practice, the power splits are considered acceptable if they fall within three MW, so achieving a power split convergence to within 2.7% for a power cycle of 110 MW would meet this requirement. As in step 1, there are two options for optimizing the method used in step 2: time or processors. The time minimization path can be seen in Figure 27.

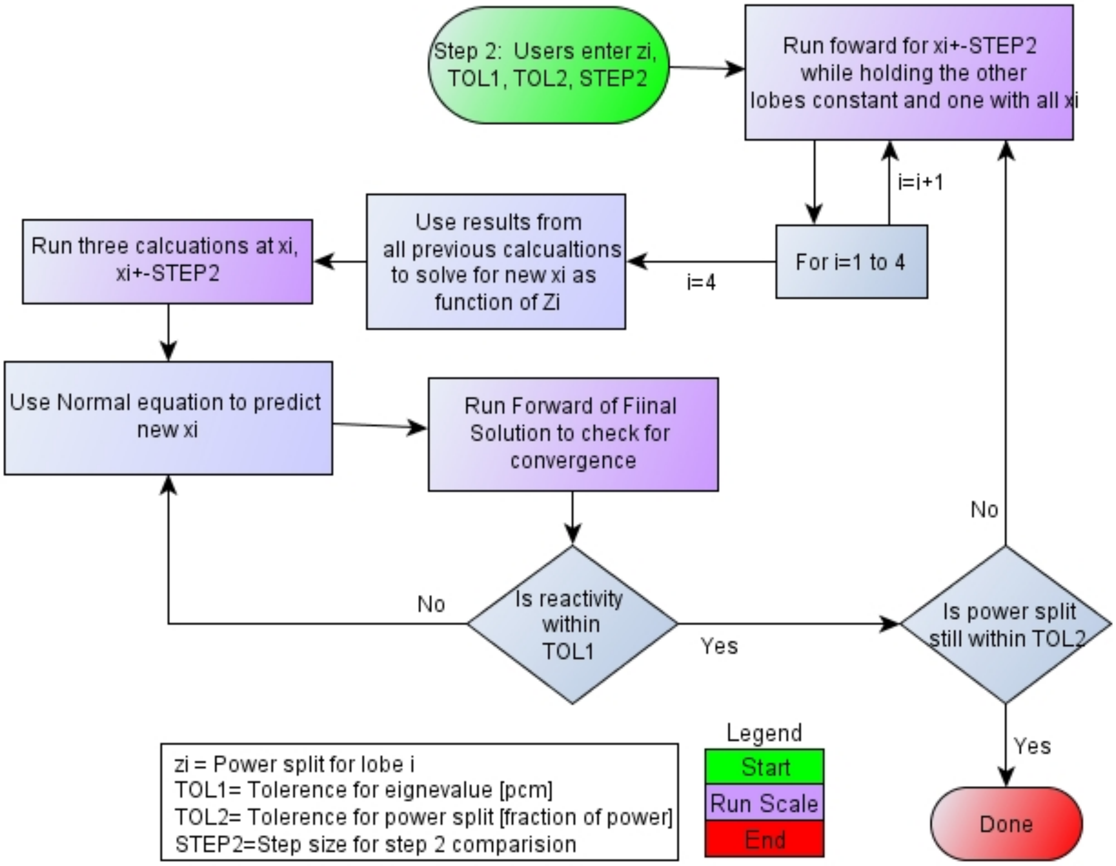

Step 2: Users enter zi, TOL1, TOL2, STEP2
Run foward for xi+-STEP2 while holding the other lobes constant and one with all xi
i=i+1
For i=1 to 4
i=4
Use results from all previous calcualtions to solve for new xi as function of Zi
Run three calcuations at xi, xi+-STEP2
Use Normal equation to predict new xi
Run Forward of Fiinal Solution to check for convergence
No
Is reactivity within TOL1
Yes
No
Is power split still within TOL2
Yes
Done
zi = Power split for lobe i
TOL1= Tolerence for eignevalue [pcm]
TOL2= Tolerence for power split [fraction of power]
STEP2=Step size for step 2 comparision
Legend
Start
Run Scale
End

Figure 27 Time minimization tasks used to predict shim rotation for step 2

The first task for step 2 is for the user to enter the desired power split (zi [% Power]) for the four lobes, the tolerances for reactivity (TOL1) and power split (TOL2), step size for automation script (STEP), and to select the optimization parameter (time or processor).

In step 2 when time is optimized the maximum number of simultaneous calculations is nine with a minimum number of sequential runs being three. If the number of processors is optimized then the minimum number of sequential runs needed for this method is seven.

For the time optimization method, two forward calculations for each lobe rotated xi± STEP while the other three groups of shims are held at the result obtained from step 1. This results in a total of 8 parallel calculations for the time optimization method. For the number of processors minimization method the shim is rotated xi+STEP away from the reactor if the fractional power split is greater than .25 and increment of xi-STEP toward the reactor if the fractional power split is less than 0.25. This rotation also occurs while holding the other three lobes in the fixed position given by step 1. For this portion of step 2 the minimization of processor option requires four sequential forward calculations.

Once the forward runs have finished, the normal equation is used to calculate the needed rotation for the desired power split. A flow chart displaying the logic used with the normal equation can be seen in Figure 28.

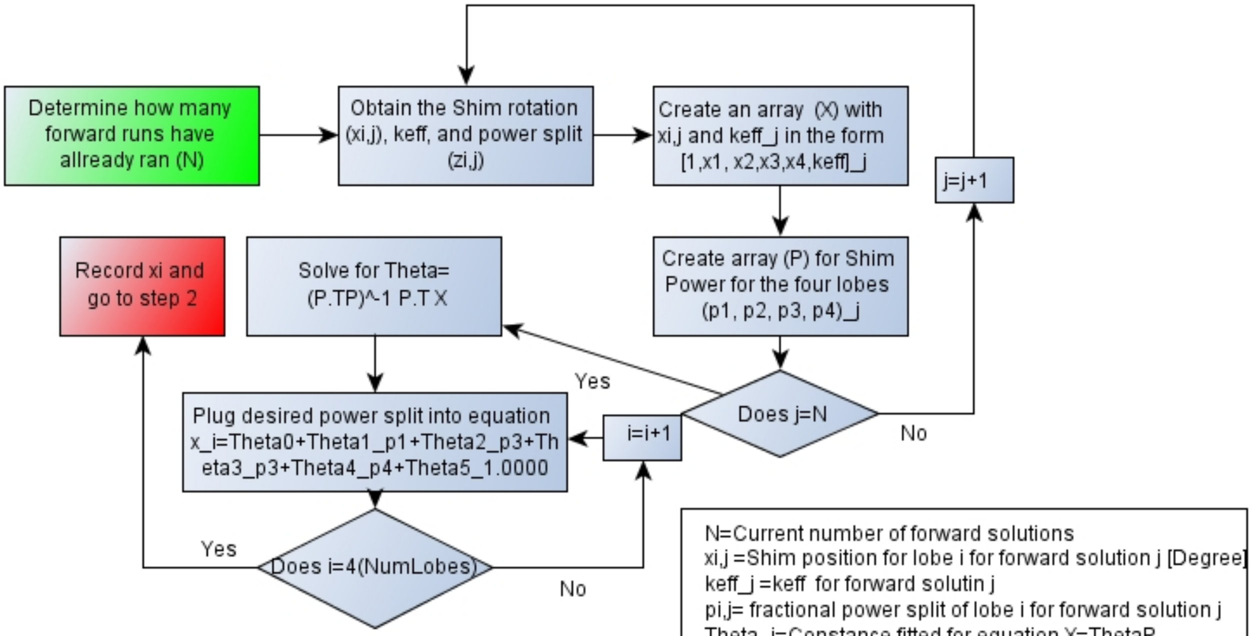

Determine how many forward runs have allready ran (N)
Obtain the Shim rotation (xi,j), keff, and power split (zi,j)
Create an array (X) with xi,j and keff_j in the form [1,x1, x2,x3,x4,keff]_j
j=j+1
Create array (P) for Shim Power for the four lobes (p1, p2, p3, p4)_j
Record xi and go to step 2
Solve for Theta= (P.TP)^-1 P.T X
Yes
Does j=N
No
i=i+1
Plug desired power split into equation x_i=Theta0+Theta1_p1+Theta2_p3+Theta3_p3+Theta4_p4+Theta5_1.0000
Yes
Does i=4(NumLobes)
No
N=Current number of forward solutions
xi,j =Shim position for lobe i for forward solution j [Degree]
keff_j =keff for forward solutin j
pi,j= fractional power split of lobe i for forward solution j
Theta_j=Constance fitted for equation X=ThetaP

Figure 28 Normal equation used in Step2

As seen in Figure 28, the shim prediction automation code obtains the shim rotation for each lobe along with the power split in the four lobes and the keff for each run. The shim rotation and keff data is saved into one matrix whereas the power split data is saved in another matrix. These are used to solve for the normal equation and the solutions are used along with the desired power split to calculate the new shim rotation for each of the four lobes.

As with step 1, the normal equation needs a form that captures changes in power as a function of the shim position. Because of the unique design of the ATR there is no analytical solution for shim position and power split that could be used for curve fitting. Therefore an arbitrary fit was selected for use with the normal equation. For step 2 the arbitrary equation is a linear combination of all of the shim positions plus a scaling factor.

Another important decision in choosing the correct equation to fit the data is to choose the independent and dependent variables. It is desirous to have shim rotation be the dependent variable and the power split become the independent variable. This would allow the user to enter the needed power split and the code would display the correct shim rotation. To provide a unique solution for the independent value of power split (zi), it is required that the eigenvalue of the solution for a given power split to be one. The fitted equations for each eigenvalue “i" and for shim position “j” used with the normal equation for Step 2 can be seen below.

$$x_j^{(i)}(x) = \sum_{n=0}^{5} \theta_{j,n}^{(i)} z_n^{(i)} \quad [67]$$

i represents each eigenvalue soluiton

$z_o^{(i)} = 1, z_1^{(i)} = \text{NW Power split}, z_2^{(i)} = NE \text{ Power split}$

$z_3^{(i)} = SW \text{ Power split}, z_4^{(i)} = SE \text{ Power split}, z_5^{(i)} = \text{Eigenvalue}$

$\theta_{j,n}^{(i)} = \text{regression coefficents for independent variable n for equation describing lobe j}$

$x_j^{(i)}(x) = \text{Shim Position for lobes j}$

As with Step 1 the normal equation is used to minimize the error between the dependent variable, $x_j^{(i)}(z)$ and the independent variable $z_n^{(i)}$ by solving for the regression coefficients $\theta_{j,n}$.

$$\vec{\theta}_j = \left(Z^T Z\right)^{-1} Z^T \vec{X}_j + Error \quad [68]$$

$$Z = \left[\left|\begin{matrix} z_0^{(1)} & z_1^{(1)} & z_2^{(1)} & z_3^{(1)} & z_4^{(1)} & z_5^{(1)} \\ z_0^{(2)} & z_1^{(2)} & z_2^{(2)} & z_3^{(2)} & z_4^{(2)} & z_5^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{matrix}\right.\right], \vec{X}_j = \begin{bmatrix} x_j^{(1)} \\ x_j^{(2)} \\ \vdots \end{bmatrix} [69]$$

The statistical analysis performed for step 1 was also performed for step 2. This included calculation of the, R^2, F-test, T-test, and probability range for the power split for the four lobes.

Once the shims positions for all four lobes have been calculated for obtaining the correct power split, the next step is to run multiple forward calculations to determine what uniform change in shim rotation for all four sets of drums would produce an eigenvalue of one. If the user chose to optimize based on time then additional three forward calculations are ran using the following equations.

$$x_j^{(1)} = x_j^* \quad [70]$$

$x_j^{(2)} = x_j^* + STEP$ [71]

$x_j^{(3)} = x_j^* - STEP$ [72]

$x_j^* =$ New shim position for shim j predicted from normal equation used in step 2

If the user chose to optimize based on processors instead then only two forward runs are used to predict the uniform change in rotation as seen below.

$x_j^{(1)} = x_j^* + STEP$ [73]

$x_j^{(2)} = x_j^* - STEP$ [71]

After the second group of forward calculations have completed, the normal equation used in step 1 is used to determine the new shims rotation positions for obtaining a reactivity of zero pcm. The statistical data was used to calculate the uncertainty in the reactivity for the time minimization routine to determine if the result converged on the correct reactivity. If the result still has not converged on an eigenvalue within the desired tolerance additional runs will be needed beginning from new predicted location. If the statistical data has converged on the correct eigenvalue then a final run is needed to verify that the power splits are also within the desired tolerance.

Rotating the shims uniformly to obtain a reactivity of 0 pcm after obtaining the desired power split previously does not guaranty the power split will still be within the desired tolerance. Therefore, after the final run has finished with the predicted shim rotation and the eigenvalue is within the desired tolerance (TOL1) then the power split needs to be verified that it is still within the desired tolerance (TOL2). If the power split is not within the desired tolerance (TOL2) then the entire process is repeated starting at the last predicted location and having a new STEP size half of the original STEP size. This iterative process continues until both the eigenvalue and power split has converged within the specified tolerances.

Validation of Results

STEP1 OF CUBIC REGRESSION METHOD

For testing and code verification, the cubic regression method was first applied on a simplified model of the ATR. See Figure 11 and the associated discussion found in the previous section for the description of the model used. The first test on this simplified ATR model for both methods was with an initial guess of 55.0 degrees. For the run minimization option based on time, five forward eigenvalue calculations were simultaneously solved with the shims rotated at 35, 45, 55, 65, and 75 degrees.

The normal equation predicted that a rotation of 55.4 degrees would produce a reactivity of 0.0 pcm. The equation used to calculate the needed shim rotation for obtaining a reactivity of zero can be seen in the Table 8 in the row titled "equation". The R^2 for the equation was 0.9998. In addition, a F-test was performed on the data to determine if independent values in the fitted equation would influence the dependent values. The F*-value was calculated to be 6066.59 which is far above the limit for the null hypothesis; the independent values influence the dependent value.

Table 8 Simplified model using cubic regression method

	Theta_0	Theta_1	Theta_2		%Conf
Equation	-5778.85	87.100	.00553	R2=0.99983	95%
t_test	-26.38	13.951	8.5271	T* =4.3026	95%

The t-test was also performed on the results obtained from the five forward calculations for the minimization of time algorithm. For the null hypotheses to be rejected, the t* must be greater than f which was 4.3026. The t-statistic for all three parameters were greater than f. From the t-test it can be concluded that every constant in the cubic equation is significant. A similar statistical test was performed for the linear regression model as seen in Table 9. From the statistical results, it can be concluded that the linear and cubic curve fit can be used to accurately model the reactivity from shim rotation and the parameters calculated for the curve fit are statistically significant.

Table 9 Simplified model using linear regression method

	Theta_0	Theta_1		%Conf
Equation	-7573.9	140.3	R2=0.9962	95%
t_test	-27.6	28.1	T*=3.1824	95%

Using the results for shim rotation obtained for the normal equation, 55.4 degrees, the range of reactivity within of specified confidence interval can be calculated. For this scenario at a confidence of 95% the range in pcm for 55.4 degrees is between -120 and 120 pcm. This range is less than the desired range of under 150 pcm so the automation code assumes this is the correct solution and then moves on to calculate the results needed for step two, using 55.4 degrees as the starting case.

For the minimization option based on the number of processors the initial guess was also 55 degrees. From this initial guess, the first two sequential forward calculations were with the shims rotated at 45 and 65 degrees. From these initial results analyzed with the normal equation, the next shim location was predicted to be 54.7 degrees. Therefore, another forward calculation was run at 54.7 degrees, which revealed a departure from criticality of greater than 150 pcm. With the three data points obtained from the previous runs, the normal equation was used to predict a new shim rotation at 55.4 degrees. The results from this forward solution was around 32 pcm from critical. Even though the initial guess was very close to the final solution, it took a total of four sequential runs before the solution converged within 150 pcm of critical. However, fewer processors were used with this minimization option than with the minimization option based on time.

Additional runs were performed for bounding cases to study the maximum number of iterations required to converge within 150 pcm of critical. The minimum and maximum the shim positions are 0 to 90 degrees but the minimization of time algorithm chooses 2*STEP lower and 2*STEP above the initial guess. In this scenario the variable STEP was set equal to 10 degrees, therefore, for this test the starting positions were taken as 20 degrees and 70 degrees to prevent calculations of shim rotations beyond the reactor specifications. For the 20 degree scenario, the shim were rotated at 0, 10, 20, 30, and 40 degrees for the minimization option based on time and 10, 30 for the minimization option based on the number of processors.

For the 70-degree scenario, the shims were rotated at 50, 60, 70, 80, and 90 degrees for minimization based on time and 60, 80 degree for minimization based on the number of processors. The convergence results for the maximum bounding shim position guesses can be seen in Table 10 .

Table 10 Maximum Shim Positioning Results for minimization of time and processors

Min	Guess	Increment	Num Proc	Predicted [degree]	Confidence [pcm]	Reactivity Results [Pcm]
Time 1	70	-10	5	55	±293.6	
Time 2	55	-5	3	55	NA	-87.9
Proc 1	70	10	2	55.4	NA	NA
Proc 2	55.4	0	1	NA	NA	-32.5

The upper bounding case required a total of two sequential runs for the time minimization option and three sequential runs for the minimization option based on the number of processors. However, it took a total of eight processors for the minimization of time option but only three for the processors option. The results from the lower initial shim positions can be seen in Table 11.

Table 11 Minimization shim position results for both minimization of time and number of processors

Min	Guess (Time)	Increment	Num Proc	Predicted [degree]	Confidence [pcm]	Reactivity Results [Pcm]
Time 1	70.0	-10	5	55.0	±293.6	
Time 2	51.2	-5	3	56.2	0	94.5
Proc 1	20.0	10	2	86.1	±NA	NA
Proc 2	86.1	0	1	49.3	±3083	4396.0
Proc 3	49.3	0	1	58.0	±1045	-841.6
Proc 4	58.0	0	1	56.7	±423	377.6
Proc 5	56.1	0	1	56.2	0	97.0

From the minimum shim position calculation, it took two sequential runs with a total of eight processors. It took six sequential runs for the processors minimization option.

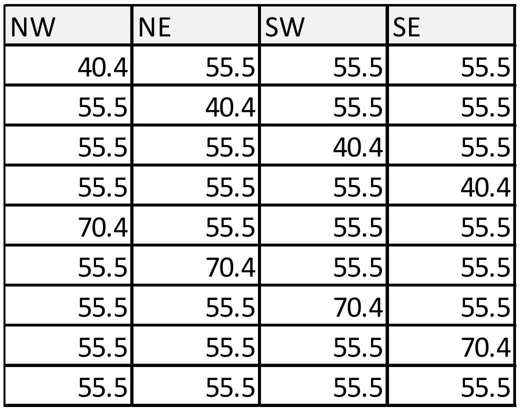
Step 2 of Linear Regression Method

The simplified ATR model described in step 1 was used to demonstrate step2. The first task for shim rotation prediction for step 2 was to take the data from step 1 as a starting point, which was 55.4 degrees in this scenario. Three power splits were chosen for testing the power split algorithm for shim rotation automation. The regression code allows the user to input power splits as a fraction of the total power. However, the code does not attempt to optimize the power split for the center flux trap. This is because the center flux trap power split currently cannot be directly controlled with shim rotation. With these limitations in mind, the three different power splits that were used for testing the power split algorithm can be seen below:

- 50/50=0.25 0.25 0.25 0.25 [Fraction of total power]
- 54/46=0.23 0.24 0.26 0.27 [Fraction of total power]
- 56/44=0.22 0.23 0.27 0.28 [Fraction of total power]

In each case, the initial forward calculations were the same. For the minimization of time option, the forward calculations were run at 40.4 degrees for one lobe while holding all of the other lobes at 55.4 degrees. In addition, forward calculations were also run at 70.4 degrees while holding the other three lobes at 55.4 degrees (see
Table 12). This resulted in a total of eight forward calculations that were run concurrently.

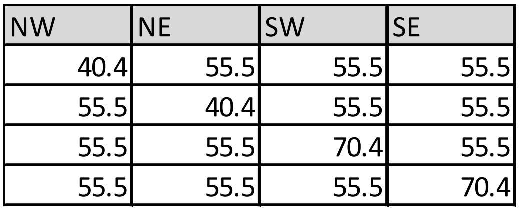
Table 12 Shim positions for the initial power split run for the minimization of time option



NW	NE	SW	SE
40.4	55.5	55.5	55.5
55.5	40.4	55.5	55.5
55.5	55.5	40.4	55.5
55.5	55.5	55.5	40.4
70.4	55.5	55.5	55.5
55.5	70.4	55.5	55.5
55.5	55.5	70.4	55.5
55.5	55.5	55.5	70.4
55.5	55.5	55.5	55.5

For the time minimization problem the NW and NE lobes were rotated 40.4 degrees separately while the other three lobes were held constant and the SE and SW lobes were rotated at 70.4 degrees while the other three lobes were held constant (see Table 13). This required four sequential runs for the number of processors minimization option.

Table 13 Shim position for the initial position for the minimization of processor option



NW	NE	SW	SE
40.4	55.5	55.5	55.5
55.5	40.4	55.5	55.5
55.5	55.5	70.4	55.5
55.5	55.5	55.5	70.4

Once the initial forward calculations were finished, the normal equation was used to predict the shim positions needed to obtain the desired power splits. Table 14 contains the results from the time minimization scenario whereas Table 15 contains the results from the option based on the minimization of processors.

Table 14 Step2 for minimization of time

Scenario	Num procs	X1[Degree]	X2[Degree]	X3[Degree]	X4[Degree]
50-50-Power	8	55	54.7	54.9	54.7
50-50-keff	3	55	54.7	54.9	54.7
54-46-Power	8	28.6	46.8	63.7	82
54-46-keff	3	25	43.3	60.2	78.4
56-44 Power	8	16.6	36.1	75.3	94.6
56-44 -keff	3	7.4	26.8	66.1	85.5

Table 15 Step 2 for Minimization of Processors

Scenario	Num procs	X1[Degree]	X2[Degree]	X3[Degree]	X4[Degree]
50-50-Power	4	48.3	49.3	45.3	43.3
50-50-keff	2	57	58	54	51.9
54-46-Power	4	21.8	42.3	56.6	70.6
54-46-keff	2	25.9	46.4	60.7	74.6
56-44 Power	4	9.6	30	68.1	82.4
56-44 -keff	2	9.6	29.6	67.7	82

On the left column of both tables there is the name of the scenario followed by either power or keff. The power label represents that the normal equation was used to predict the corrected power split. The keff label represents that the normal equation was used to predict a uniform change in shim rotation that would result in a reactivity value of 0.0 pcm. For example in Table 14 for the 54-56-Power row eight forward solutions were used to predict a shim rotation of 28.6, 46,8, 63.7 and 82.0 degrees for the desired power split. Then at this new predicted value (xi) and at xi ±STEP2 three forward solutions were run. From these three forward solutions, the normal equation predicted that a shim position at 25.9, 46.4, 60.7, and 74.6 would result in a reactivity of 0.0 pcm.

Once the new shim positions were predicted a confirmatory run was needed to ensure the eigenvalue result was within the specified tolerance (TOL1) and the power split results were within the specified tolerance (TOL2). The results from the confirmatory run for both minimizations based on time and processors can be seen in Table 16 and Table 17 respectively.

Table 16 Step 2 minimization of time

Scenario	P1(NW) fraction	P2 (NE)	P3 (SW)	P4 (SE)	Final results (pcm)
50-50 Result	0.250	0.250	0.250	0.250	-115.875
50-50 Goal	0.250	0.250	0.250	0.250	-
Results-Goal	0.000	0.000	0.000	0.000	-
54-46 Result	0.233	0.239	0.257	0.271	-1.666
54-46 Goal	0.230	0.240	0.260	0.270	
Results-Goal	0.003	-0.001	-0.003	0.001	
56-44 Results	0.228	0.230	0.263	0.270	-14.600
56-44 Goal	0.220	0.230	0.270	0.278	
Results-Goal	0.008	0.000	-0.007	-0.008	-

Table 17 Step2 minimization based on the number of processors

Scenario	P1(NW) fraction	P2 (NE)	P3 (SW)	P4 (SE)	Final results (pcm)
50-50 Result	0.252	0.253	0.249	0.247	35.200
50-50 Goal	0.250	0.250	0.250	0.250	
Results-Goal	0.002	0.003	-0.001	-0.003	-
54-46 Result	0.233	0.241	0.258	0.268	-31.570
54-46 Goal	0.230	0.240	0.260	0.270	
Results-Goal	0.003	0.001	-0.002	-0.002	
56-44 Results	0.229	0.232	0.264	0.275	-6.326
56-44 Goal	0.220	0.230	0.270	0.280	
Results-Goal	0.009	0.002	-0.006	-0.005	-

Both the minimization based on time and on the number of processors were able to converge to the final solution within one outer iteration.

Conclusion

The cubic regression method is effective for automating shim rotation prediction. One option that was available with the cubic regression method was to optimize the code based on reducing the time or reducing the number of processors used. For Step 1 of shim rotation prediction only two sequential runs were needed for the time optimization method. The processor optimization method was able to converge to the desired solution with a maximum of six sequential runs. For Step 2 of shim rotation automation the time option has a maximum of three sequential runs using at most eight processors. The processor optimization option had a total of seven sequential runs before converging on the desired solution. Because the adjoint solution of the transport equation is not needed for the cubic regression method, this method can be applied to all eigenvalue solvers. Future work for the cubic regression method includes modifying the code to automate shim rotations for new computer codes such as MCNP and Helios.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

In 2007 the DOE declared the ATR a national scientific user facility [Allen et al 2010]. This declaration expanded the focus of the ATR from naval reactor applications to include diversified classes of academic and industrial experiments. For these reasons, the INL is in the process of upgrading safety analysis to include more accurate and flexible models.

An essential part of the new suite of codes for safety analysis is their ability to predict how the reactor will behave at startup. Before any reactor startup, the operators must know where the outer shim control cylinders (OSCC) should be rotated to obtain criticality. In addition, the shims must independently be rotated so that the specified power split for the inserted experiments and reactor fuel composition are obtained. One concern with the new suite of codes such as NEWT is the impracticality of obtaining the needed shim rotation predictions through manual iterations. Therefore, perturbation theory was implemented for automating the shim rotation calculations to speed up the startup analysis, which would otherwise be prohibitively computational-intensive. However, it was shown that ordinary first order perturbation theory failed to handle large control system movements so a modified form of first order perturbation theory, called phase space interpolated perturbation theory, was developed to more accurately model shim rotation prediction. Phase space interpolated perturbation theory uses a modified phase space interpolated adjoint flux in the perturbed regions by solving the adjoint solution for a bounding scenario. The perturbation inner product evaluation wit bounding forward and a modified adjoint phase space interpolated fluxes provide a realistic representation of the perturbed solution. This is especially true for cases intermediated to the bounding configuration.

Phase space interpolated perturbation theory fundamentally improved the reactor physics modeling capability for optimization and automation calculations for the ATR. Phase space interpolated perturbation theory can be applied to any neutron transport code that can solve both the forward and adjoint form of the transport equation. As long as there is a known range of changes to the system that need to be analyzed then phase space interpolated perturbation theory can provide a quick and accurate method for calculating the effect of those changes on reactivity. The application that phase space interpolated perturbation theory was demonstrated in this dissertation for shim rotation prediction of the ATR

For neutronic codes that do not have the capabilities of calculating the adjoint a different method called the cubic regression method was developed and tested for shim rotation automation.

Shim rotation prediction is a two-step process. The first step involves determining the uniform shim position required for the reactor to be critical. The second step involves determining the non-uniform shim positions needed to achieve the specified power splits within each of the four experimental lobes while still maintaining criticality.

Both phase space interpolated perturbation theory and the cubic regression method were used to automate shim rotation prediction for both of the required steps needed for startup. One of the options implemented was to run the calculations so that the number of processors being used was minimized or the amount of time to converge on the correct solution was minimized. Bounding case scenarios were tested to determine that each automating method would converge on the correct solution in a reasonable amount of time and with a reasonable number of processors.

Shim rotation automation for ATR startup using perturbation theory was accomplished but additional work is still needed for shim rotation automation for an entire reactor cycle. The next recommended step involves automating shim rotation for burn up calculations. As the reactor is increased to full power at startup the fuel will begin to fission and the reactivity will decrease. To maintain criticality the OSCC shims will need to be rotated away from the core and the neck shims will eventually need to be removed from the reactor.   All of these steps must occur while maintaining criticality and keeping within the specified tolerances for the power splits.   Shim rotation as a function of burnup can be automated with a combination of phase space interpolated perturbation theory, linear regression, and a depletion code such as Origen. Phase space interpolated perturbation theory could be used to predict where the shims need to be adjusted as the core is being depleted. When the shims are moved at a specified distance then a new forward and adjoint calculation could then be run and the process repeated.

Additional work on ATR physics analysis could also involve using phase space interpolated perturbation theory for ATR fuel placement optimization. One potential advantage of using phase space interpolated perturbation theory instead of FOPT for fuel placement optimization is phase space interpolated perturbation theory may be able to handle both the spent fuel and fresh fuel whereas FOPT has been shown to work only with spent fuel placement.

Phase space interpolated perturbation theory is an original approach to perturbation theory that works well when bounding cases are known. It can be applied to any transport code that provides both the forward and adjoint solutions. Unlike first order perturbation theory, phase space interpolated perturbation theory can even be used for large perturbations such as shim or control rod removal while still only requiring two eigenvalue solutions. It produces results that are more comparable to higher order perturbation theory methods without the need for a significant number of higher order adjoint calculations.

APPENDIX A: ADDITIONAL VALIDATION OF PHASE SPACE INTERPOLATED PERTURBATION THEORY

Three examples will be used to validate phase space interpolated perturbation theory. The first example involves replacing an aluminum experiment for a cobalt experiment in multiple ATR locations. The second example involves calculating the reactivity worth of multiple control shims being withdrawn from the reactor. The final example involves calculating the integral control rod worth curve.

The first example was looking at the change in reactivity from replacing I-hole aluminum fillers with cobalt production targets. The I-holes surround the ATR’s center flux trap and are typically used to irradiate cobalt targets. A simplified model of the ATR, as discussed in Chapter 2, was used for this example with the addition of a more detailed model of the I-holes as seen in Figure 29. Each of the I-hole positions were 0.47325 cm in diameter.

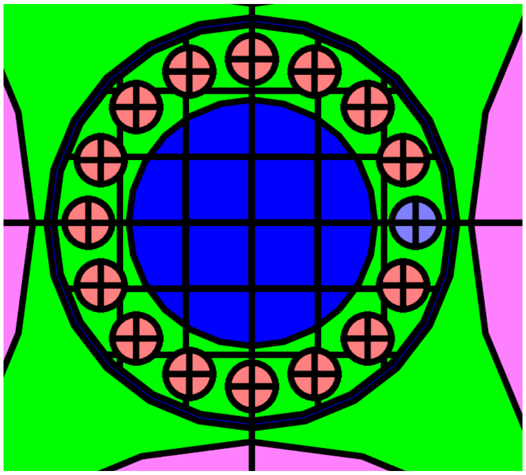


Figure 29 Cobalt experimental placements in the I-holes

The aluminum was incrementally replaced with cobalt starting from the very right I- hole and working clockwise. The forward calculation was computed with all the I-holes filled with aluminum, and the adjoint calculation for phase space interpolated perturbation theory was with all 16 I- holes filled with cobalt. The results from phase space interpolated perturbation theory were compared against results from first order perturbation theory, exact perturbation theory, and the linear regression method as seen in Figure 30.

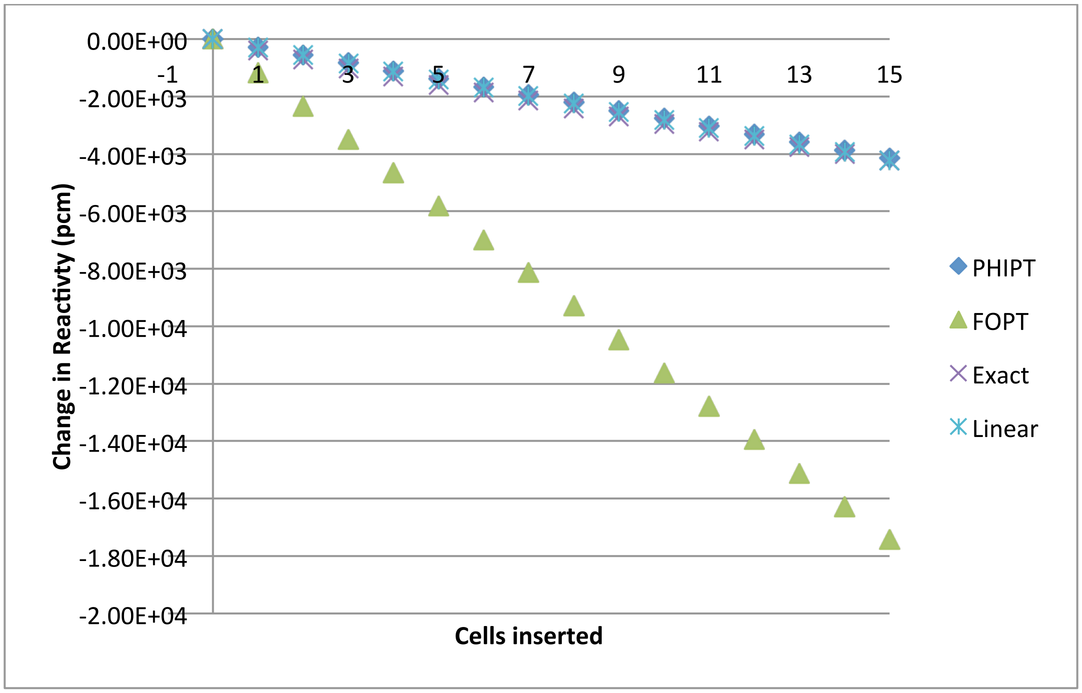


Figure 30 Comparison of FOPT, phase space interpolated and forward solution for cobalt experiment

As seen in the Figure 30 SCPT was a significantly improvement to FOPT.

The second example involves calculating the change in reactivity from removing the neck shims of the ATR. The simplified model of the ATR that was used in the second example was also used for this example with the addition of the twenty-four neck shims as seen in Figure 31. The neck shims were 0.5 cm in diameter and were composed of hafnium (13.31 g/cc). When the neck shims are removed the neck followers, which are composed of 100% Al-27(2.7 g/cc), fill in the void.

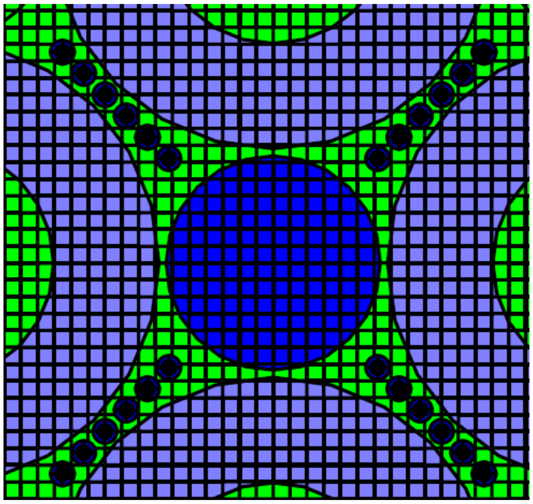


Figure 31 Shim rod removal/insertion scenario

In this example, the forward calculation was performed with all of the neck shims inserted and the adjoint calculation for SCPT was done with all of the neck shims removed. Starting from the four outermost neck shims, in groups of four and for six steps, the reactivity from inserting the shims were calculated (Figure 32.)

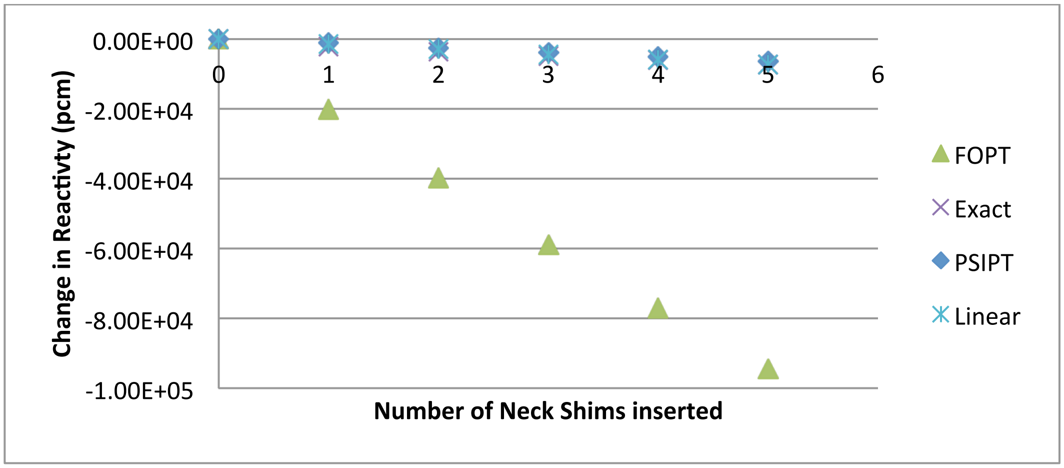


Figure 32 Comparison of results for control rod insertion

As seen in Figure 32, phase space interpolated perturbation theory significantly outperformed first order perturbation theory. As seen in the other example, the errors of phase space interpolated perturbation theory are comparable to the errors of a linear regression method.

The final example with phase space interpolated perturbation theory is an integral control rod worth curve for a simplified reactor that is 100 cm wide and 100 cm high with reflecting boundaries along sides and vacuum boundary along the top and bottom. The reactor is composed of 90 weight percent U235 and 10 weight percent of boron-10 with a density of 2.0 g/cc. In the middle of the reactor is a boron control rod with an aluminum follower that is 2 cm thick. The density of the boron-10 control rod is 2.34 g/cc whereas the density of the aluminum-27 control rod follower is 2.7 g/cc. The control rod worth curve is analyzed for ten equally spaced insertion steps that are 10 cm apart (See Figure 33).

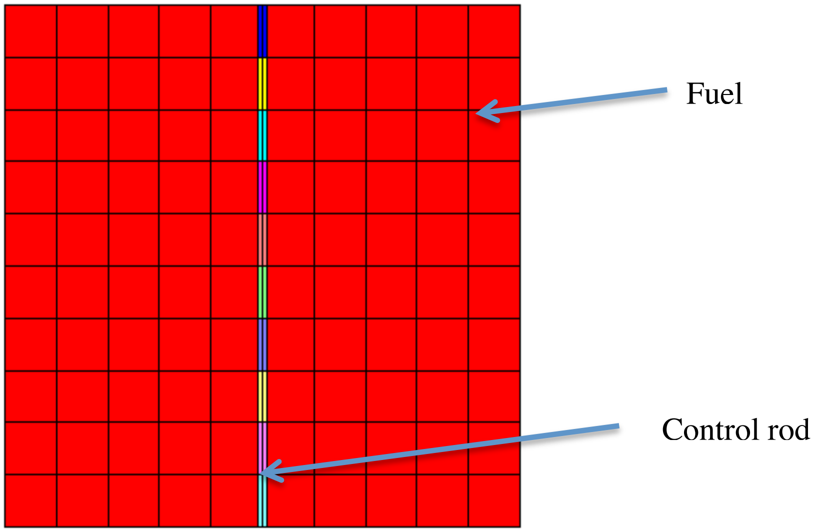


Figure 33 2D control rod insertion geometry

The control rod worth curve was calculated using FOPT, phase space interpolated perturbation theory, exact perturbation theory, and a linear regression method. The linear regression method uses the eigenvalue from the bounding cases to interpolate other results. The linear regression method has been seen to work well for changes that give rise to linear responses, but for nonlinear reactivity effects the linear method will contain larger errors. For FOPT the adjoint and forward eigenvalue solutions were calculated with the control rod removed. For the phase space interpolated perturbation theory, the forward eigenvalue solution was calculated with the control rod removed and the adjoint eigenvalue solution was obtained with the shim completely inserted. This setup for phase space interpolated perturbation theory provides the full range for calculating the integral control rod worth curve. The results from all four methods can be seen in the Figure 34.

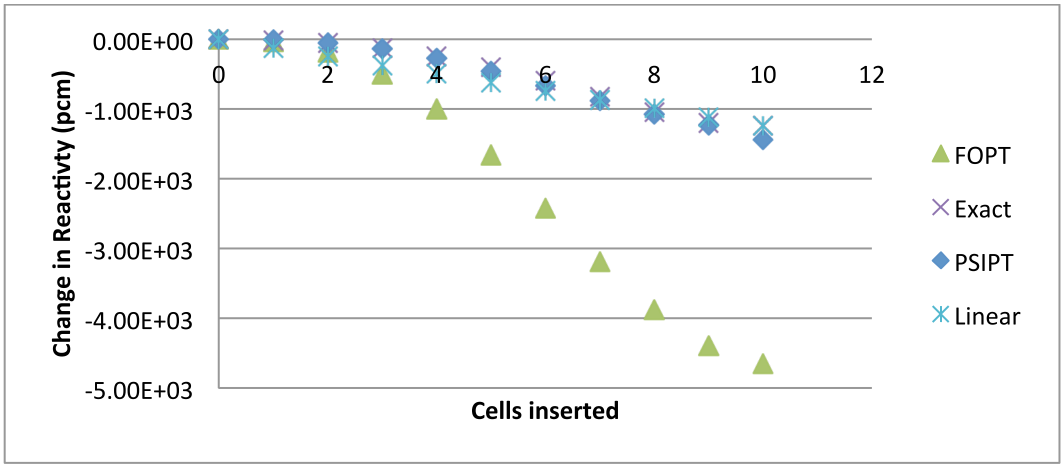


Figure 34 Rod insertion calculations

The phase space interpolated perturbation theory, exact perturbation theory, and the linear regression method all line up relatively well, but as expected the first order perturbation theory over predicts the negative reactivity associated with shim insertion. However, phase space interpolated perturbation theory captures nonlinearities in the rod insertion worth with excellent fidelity. A plot of the error from shim insertion for phase space interpolated perturbation theory and the linear regression method can be seen in Figure 35.

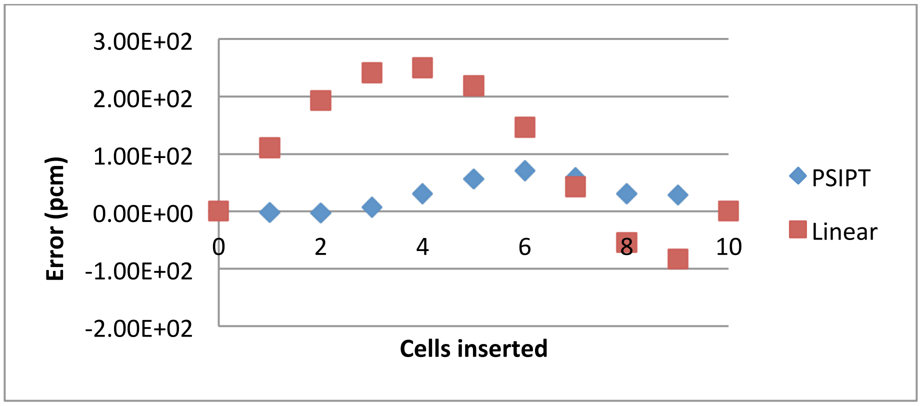


Figure 35 Comparison of the error from phase space interpolated perturbation theory and linear interpolation.

As seen in Figure 35 phase space interpolated perturbation theory on average had smaller errors then the linear regression method. Hence highly nonlinear changes such as control rod worth phase space interpolated perturbation theory will outperform the linear regression method as well as FOPT.

Three different applications for phase space interpolated perturbation theory were analyzed. The results from all three applications demonstrated that phase space interpolated perturbation theory provided a much more accurate calculation than FOPT. The final example also showed that for highly nonlinear results, phase space interpolated perturbation theory outperformed both FOPT and the linear regression method.

APPENDIX B: USERS GUIDE TO PERTN AND ASSOCIATED TOOLS

PertN (Perturbation theory with NEWT) is a computer code that was developed as an external wrapper for NEWT. PertN reads in data obtained from a modified form of NEWT and then performs either first order perturbation theory (FOPT), exact perturbation theory (EPT), or phase space interpolated perturbation theory (SPCT) by looking at the changes in reactivity as a function of the changes in cross section. There is also a subroutine built into PertN that estimates reactivity from shim rotation for the Advanced Test Reactor. Each section of the user guide will include a description of how to run the code and associated tools followed by a detailed walkthrough.

NOTE: This user guide is written with the assumption that the user already knows how to run NEWT.

Using FOPT, EPT, or SCPT for Changes in Cross Sections

PertN is used for post processing the results from NEWT to apply perturbation theory to a reactor model for quick estimations of reactivity from small changes to the core. For all of PertN the user will need to first run Newt multiple times and then assemble the needed output files into a working directory. Then PertN can be run from the working directory and the results can be analyzed from an Excel readable output file. The algorithms used within PertN are discussed in detail in “Study 1” chapter of the dissertation. As a standalone wrapper, PertN can perform perturbation theory on any type of model as long as there is a difference in macroscopic cross sections and other parameters of the model are not changed. Such parameters include the grid and mesh size, the geometry description, the Sn Pn value, and the energy group structure.

NEWT, a neutron transport module integrated in the SCALE suite, has been modified to output four files needed to run PertN. See Appendix E for a detailed description of all of modifications made to Newt. The four files are:

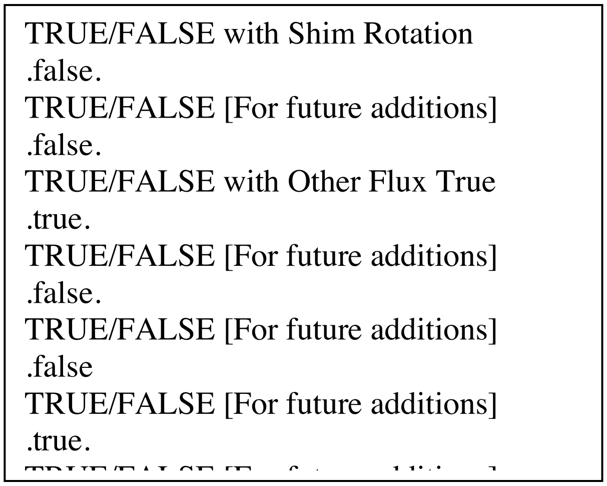
- Cross_text.txt
- Pert_text.txt
- Other_spect.txt [optional]
- Adjoint_spec.txt [optional]

The file “Cross_text.txt” is generated by running the NEWT input deck for a typical forward calculation. The file can also be generated after changing the NEWT parameter flag from “run=yes” to “run=no”. Generating the cross sections with the “run=no” parameter flag should take NEWT less than 3 seconds. With the “run=no” parameter flag the cross sections will first be processed by the SCALE cross section sequence which includes modules such as crawdad, bonami, worker, t-xsec. These modules provide weighting functions for the microscopic cross sections. The weighted microscopic cross sections are then mixed within each zone to produced zone specific macroscopic cross section. The mixed macroscopic cross sections are then used for with NEWT for the discrete ordinate transport code and are also saved as a file “Cross_text.txt” for use with PertN.

The file “Pert_txt.txt” is generated by running the NEWT input deck for a typical forward calculation. The file Pert_txt.txt contains the geometry description of the problem, the unperturbed macroscopic cross sections, the angular flux, and all of the additional data needed for PertN. If the adjoint solution is run in the same input deck as the forward solution then Pert_txt.txt will also contain the adjoint angular flux. If the adjoint calculation is run in the same input deck as the forward calculation then it is essential that the adjoint calculation always occur after the forward calculation.

The file “Other_spect.txt” is produced from running a forward calculation with NEWT and contains only angular flux. The file “Adjoint_spec.txt” is produced by running NEWT with the “forward=no” parameter flag. This parameter flag tells NEWT to solve for the adjoint solution instead of the forward solution. The file “Adjoint_spec.txt” contains only the adjoint angular flux.

Once all of the data obtained from running NEWT multiple times are copied into the working directory, then the user needs to create an input file, “User_input.txt”, to tell PertN the type of perturbation calculation needed. An example of what the “User_input.txt” file looks like can be seen in Figure 36.



```
TRUE/FALSE with Shim Rotation
.false.
TRUE/FALSE [For future additions]
.false.
TRUE/FALSE with Other Flux True
.true.
TRUE/FALSE [For future additions]
.false.
TRUE/FALSE [For future additions]
.false
TRUE/FALSE [For future additions]
.true.
TRUE/FALSE [For future additions]
```

Figure 36 User input file for PertN called "User_input.txt"

The first line of “User_input.txt” is a comment line that typically describes the purpose of the next line that is a Boolean (.true./.false.) flag. This is repeated for a total of eight comments and eight .true./.false. answers. The first set of Boolean flags specifies whether PertN is used for shim rotation. The second, fourth, fifth, sixth, and seventh set of Boolean flags are currently used as place holders for future releases and should not be modified. The third set of Boolean flags are used to turn on (.true.) or off (.false.) the option of running exact perturbation theory in addition to FOPT. If the third Boolean flag is turned on (.true.) then PertN will look for the file “Other_spect.txt” within the working directory.

The last set of Boolean flags allows the user to turn on or off the switch that tells PertN that the unperturbed adjoint was calculated in the same input deck as the forward (.false.) or that the unperturbed adjoint eigenvalue was calculated in a separate input deck (.true.). If the switch is set to true, then PertN will look for the file "Adjoint_spec.txt" within the working directory.

If the adjoint calculation is for the exact unperturbed case as the forward calculation then PertN will calculate the results associated with FOPT. However, if the adjoint calculation is the bounding case instead of the unperturbed case then PertN will calculate perturbations based on phase space interpolated perturbation theory.

All of the data obtained from NEWT needs to be copied into the current working directory where PertN will be run, along with the user input deck that needs to be labeled "User_input.txt". PertN is then run by typing in the command line the word "perturbation". Once PertN has completed all of the calculations specified in "User_input.txt", it will save the results in an Excel readable file called "Excell_output.txt". The output includes the predicted eigenvalue for the perturbed case along with the predicted change in reactivity.

The walkthrough for a specific example with PertN involves looking at the change in reactivity from changing the density of a simplified reactor core. The reactor core is 20 cm long and 20 cm high composed of U-235 with a density of 2.000 g/cc and has vacuum boundary conditions on all four sides. The first step is to run forward calculation with NEWT, which is accomplished with the input deck seen in Figure 37. Once the forward calculation has been run there will be a file created where the forward calculation was run called “Pert_txt.txt”. This file needs to be copied to the working directory.

NOTE: The working directory for PertN should not be the same directory where NEWT is run because NEWT will overwrite files without asking the user for permission.

```
=t-xsec   ]
Create inf. homogeneous cross sections
xn44
read comp

' Side Plates Al-6061 T6
' Fuel Meat
wtptfuelmeat    1   2.000  1
              92235   100
              1.000    310.9   end
end comp
end

=newt
```

Figure 37 Input deck used for walkthrough

The next step is to run the unperturbed adjoint calculation with NEWT. This is also accomplished with the same input deck (Figure 37) with one change. In the parameter line the “forward=yes” command needs to be switched to “forward=no.” This tells NEWT to solve for the adjoint transport equation instead of the forward transport equation. When NEWT has finished calculating the adjoint there will be a file called Adjoint_spec.txt that needs to be copied to the working directory.

The third file that is needed from NEWT is the perturbed macroscopic cross section file, called “Cross_text.txt.” This file is used to compare the unperturbed cross section with the new cross section for performing perturbation theory. The same input deck used for the forward calculation can be used for this calculation (Figure 37) with two modifications. The first modification is in the parameter section of the input deck the “run=yes” command needs to be changed to “run=no” command. The second item that needs to be modified is the density of the fuel. In this case, the density of fuel (2.000 g/cc specified) is changed to a density of 3.000 g/cc. When NEWT has finished computing the macroscopic cross sections for each zone, it will save the results to a file “Cross_text.txt” that needs to be copied into the working director. Next the “User_input.txt” needs to be modified for this specific problem and then PertN can be run. The file “User_input.txt” needs to specify that shim rotation calculation is not being run (with Shim Rotaiton= .false.), that only results from FOPT are desired (Other Flux True =.false.), and that the forward and adjoint results were calculated separately (with Adjoint flux separate = .true.), see Figure 39. The three areas highlighted are the three sections that were changed for this specific problem.

```
TRUE/FALSE with Shim Rotation
.false.
TRUE/FALSE [For future additions]
.false.
TRUE/FALSE with Other Flux True
.false.
TRUE/FALSE [For future additions]
.false.
TRUE/FALSE [For future additions]
.false
TRUE/FALSE [For future additions]
.true.
TRUE/FALSE [For future additions]
.true.
TRUE/FALSE with adjoint flux separate
.true.
```

Figure 38 User_input.txt file used for FOPT calculation with a separate forward and adjoint

Once all four files have been copied into the working directory, the final step is to run PertN by typing in the working directory "perturbation." The results from PertN predicted that changing the density from 2.000 g/cc to 3.000 g/cc cause an increase in the k-eff eigenvalue from 0.503972 to 0.775415 for a total change in reactivity of 69460.5 pcm.

Shim Rotation Calculations with PertN

One of the modules built into PertN is made specifically for shim rotation prediction with the ATR. To run this module the user input file "User_input.txt" , that was also discussed above, needs to be set up as seen below in Figure 39.

```
TRUE/FALSE with Shim Rotation
.true.
TRUE/FALSE [For future additions]
.false.
TRUE/FALSE with Other Flux True
.true.
TRUE/FALSE [For future additions]
.false.
TRUE/FALSE [For future additions]
.false
TRUE/FALSE [For future additions]
.true.
TRUE/FALSE [For future additions]
.true.
TRUE/FALSE with adjoint flux separate
.true.
  Older shims
    [NE]
    [NW]
    [SE]
    [SW]
 New OSCC shims
    [NE]
    [NW]
    [SE]
    [SW]
```

Figure 39 File "User_input.txt" used for shim rotation prediction of the ATR.

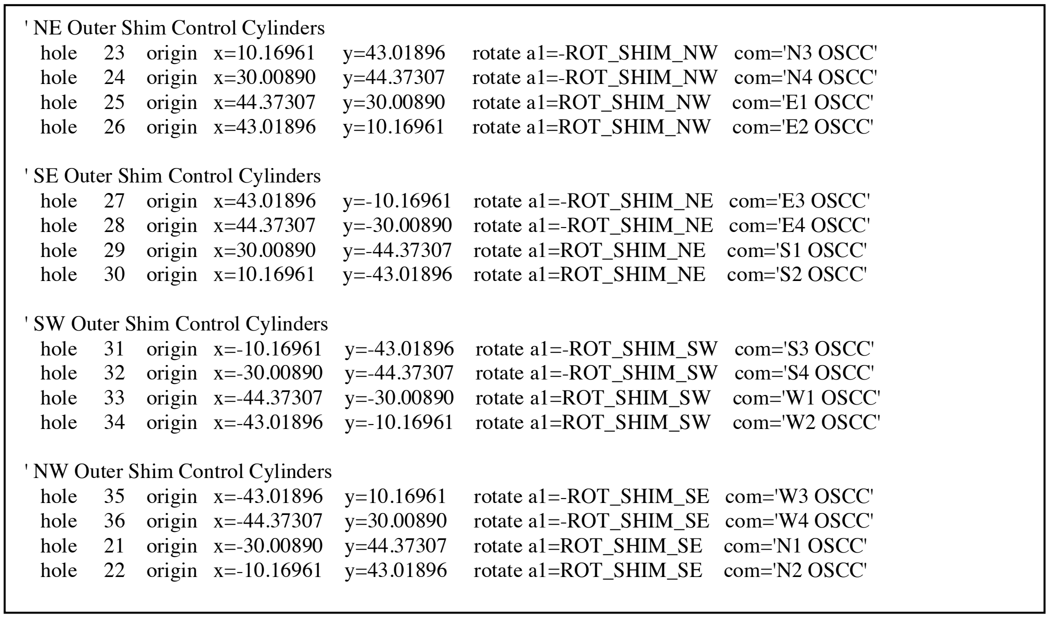
In order to run shim rotation prediction the first set of Boolean flags needs to be turned on (TRUE/FALSE with Shim Rotation =.true.). This tells PertN to use the subroutine for shim rotation prediction. The last set of Boolean flags tells PertN if the adjoint was calculated in separate input decks (.true.) or in the same input deck (.false.). None of the other Boolean flags needs to be changed. The last set of Boolean flags tells PertN if both the forward and adjoint were run in the same input deck (.false), or if the forward and adjoint were run in separate input decks (.true.).

The first part of the new section of the “User_text.txt” is a comment line followed by the initial shim position for the NW, NE, SW, and SE lobes each on a separate line and all in units of degrees. Then another comment is used and then followed by the new shim position for the NW, NE, SW, and SE lobes each on a separate line also in units of degrees.

For shim rotation prediction the files “User_input.txt”, “Pert_txt.txt” and maybe the “Adjoint_spec.txt” ,if the last Boolean flag is set to .true., are needed in the working directory. Once the required two or three files are in the working directory the user runs PertN by typing the word “perturbation” into the command line. Once PertN has finished the results from shim rotation will be saved to an Excel readable file. The output file contains both the unperturbed and perturbed k-eigenvalue solution along with the change in reactivity in units of pcm.

A simple walkthrough to demonstrate shim rotation prediction with PertN looks at the change in reactivity for rotating the shims uniformly from 60.0 degrees to 50.0 degrees. FOPT could be used for shim rotation prediction, but as shown in the first study of the dissertation, FOPT does not work well for shim rotation prediction. Instead, the modified form of phase space interpolated perturbation theory as explained in the second study of the dissertation is used. The input deck that will be used in this walk through is the same input deck used for both the second and third study and can be found in Appendix D.

The first step in shim rotation prediction is to run the forward calculation with Newt. This is accomplished by running the input deck found in Appendix D. To run for 60 degrees the input deck needs to be change at ROT_SHIM_[N,S][E,W] where [N,S][E,W] can represent NW, NE, SW, or SE to 60 degrees as seen in Figure 40 and Figure 41.



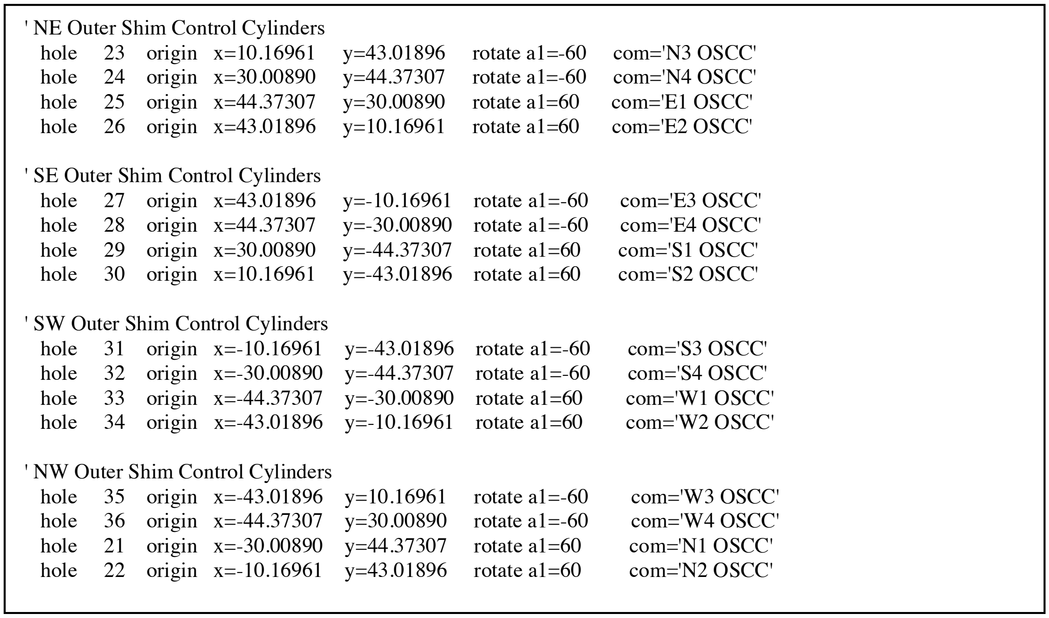
```
' NE Outer Shim Control Cylinders
  hole   23   origin   x=10.16961    y=43.01896    rotate a1=-ROT_SHIM_NW   com='N3 OSCC'
  hole   24   origin   x=30.00890    y=44.37307    rotate a1=-ROT_SHIM_NW   com='N4 OSCC'
  hole   25   origin   x=44.37307    y=30.00890    rotate a1=ROT_SHIM_NW    com='E1 OSCC'
  hole   26   origin   x=43.01896    y=10.16961    rotate a1=ROT_SHIM_NW    com='E2 OSCC'

' SE Outer Shim Control Cylinders
  hole   27   origin   x=43.01896    y=-10.16961   rotate a1=-ROT_SHIM_NE   com='E3 OSCC'
  hole   28   origin   x=44.37307    y=-30.00890   rotate a1=-ROT_SHIM_NE   com='E4 OSCC'
  hole   29   origin   x=30.00890    y=-44.37307   rotate a1=ROT_SHIM_NE    com='S1 OSCC'
  hole   30   origin   x=10.16961    y=-43.01896   rotate a1=ROT_SHIM_NE    com='S2 OSCC'

' SW Outer Shim Control Cylinders
  hole   31   origin   x=-10.16961   y=-43.01896   rotate a1=-ROT_SHIM_SW   com='S3 OSCC'
  hole   32   origin   x=-30.00890   y=-44.37307   rotate a1=-ROT_SHIM_SW   com='S4 OSCC'
  hole   33   origin   x=-44.37307   y=-30.00890   rotate a1=ROT_SHIM_SW    com='W1 OSCC'
  hole   34   origin   x=-43.01896   y=-10.16961   rotate a1=ROT_SHIM_SW    com='W2 OSCC'

' NW Outer Shim Control Cylinders
  hole   35   origin   x=-43.01896   y=10.16961    rotate a1=-ROT_SHIM_SE   com='W3 OSCC'
  hole   36   origin   x=-44.37307   y=30.00890    rotate a1=-ROT_SHIM_SE   com='W4 OSCC'
  hole   21   origin   x=-30.00890   y=44.37307    rotate a1=ROT_SHIM_SE    com='N1 OSCC'
  hole   22   origin   x=-10.16961   y=43.01896    rotate a1=ROT_SHIM_SE    com='N2 OSCC'
```

Figure 40 Location of place holder for shim rotation



```
' NE Outer Shim Control Cylinders
   hole    23   origin  x=10.16961    y=43.01896    rotate a1=-60    com='N3 OSCC'
   hole    24   origin  x=30.00890    y=44.37307    rotate a1=-60    com='N4 OSCC'
   hole    25   origin  x=44.37307    y=30.00890    rotate a1=60     com='E1 OSCC'
   hole    26   origin  x=43.01896    y=10.16961    rotate a1=60     com='E2 OSCC'

' SE Outer Shim Control Cylinders
   hole    27   origin  x=43.01896    y=-10.16961   rotate a1=-60    com='E3 OSCC'
   hole    28   origin  x=44.37307    y=-30.00890   rotate a1=-60    com='E4 OSCC'
   hole    29   origin  x=30.00890    y=-44.37307   rotate a1=60     com='S1 OSCC'
   hole    30   origin  x=10.16961    y=-43.01896   rotate a1=60     com='S2 OSCC'

' SW Outer Shim Control Cylinders
   hole    31   origin  x=-10.16961   y=-43.01896   rotate a1=-60    com='S3 OSCC'
   hole    32   origin  x=-30.00890   y=-44.37307   rotate a1=-60    com='S4 OSCC'
   hole    33   origin  x=-44.37307   y=-30.00890   rotate a1=60     com='W1 OSCC'
   hole    34   origin  x=-43.01896   y=-10.16961   rotate a1=60     com='W2 OSCC'

' NW Outer Shim Control Cylinders
   hole    35   origin  x=-43.01896   y=10.16961    rotate a1=-60    com='W3 OSCC'
   hole    36   origin  x=-44.37307   y=30.00890    rotate a1=-60    com='W4 OSCC'
   hole    21   origin  x=-30.00890   y=44.37307    rotate a1=60     com='N1 OSCC'
   hole    22   origin  x=-10.16961   y=43.01896    rotate a1=60     com='N2 OSCC'
```

Figure 41 The modified input deck to represent rotation of shim at 60 degrees

Once this change is made then the input deck can be run with NEWT. Once NEWT has finished running the file "Pert_text.txt" needs to be moved to the working directory.

The second step for phase space interpolated perturbation theory shim rotation prediction is to run the modified phase space interpolated adjoint. This is accomplished by using the same input deck that was used for the forward calculation with an additional five changes. The first change is with the parameter section of the NEWT input. Here the "forward=true" needs to be replaced with "forward=false". The four other changes occur in the material assignment section in the geometry section of NEWT. The following four lines found in the Newt input deck need to be change from Figure 42 to Figure 43. What these four lines in the input deck to is the swap the hafnium material (17, 18) for the beryllium material (4) and the beryllium material (16, 19) to hafnium (6) as seen in Figure 20 found in Study 2.

```
media     16  1  250   -100
media     17  1  250   -100
media     18  1  250   -100
media     19  1  250   -100
```

Figure 42 Lines to search for when modifying input for calculating the adjoint for shim rotation prediction

```
media     6  1  250   -100
media     4  1  250   -100
media     4  1  250   -100
media     6  1  250   -100
```

Figure 43 The modified adjoint input deck changes

Once the five changes have been made, then NEWT can be run. Once NEWT has finished, the output file “Adjoint_spec.txt” can be copied to working directory.

The final step that is needed for shim rotation prediction is to modify the “User_input.txt” file so that PertN knows to run the subroutine for shim rotation prediction [with Shim Rotation = .true.], that the forward and adjoint or separate [with adjoint flux separate = .true.], and to start at 60 degrees and ending at 50 degrees. (Figure 44)

```
TRUE/FALSE with Shim Rotation
.true.
TRUE/FALSE [For future additions]
.false.
TRUE/FALSE with Other Flux True
.true.
TRUE/FALSE [For future additions]
.false.
TRUE/FALSE [For future additions]
.false
TRUE/FALSE [For future additions]
.true.
TRUE/FALSE [For future additions]
.true.
TRUE/FALSE with adjoint flux separate
.true.
  Older shims
60
60
60
60
 New OSCC shims
50
50
50
50
```

Figure 44 Input deck for sample problem with shim rotation prediction

The results from shim rotation prediction with phase space interpolated perturbation theory include a new eigenvalue approximation of 0.993663 for a total change in reactivity of -1341.4 pcm

Shim Rotation Automation with Spect_Pert.py

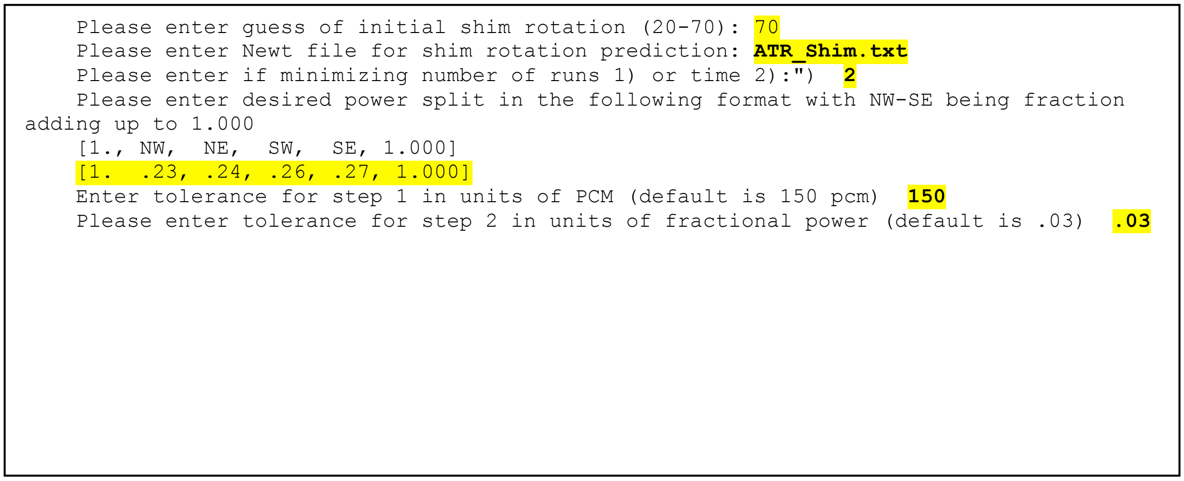
PertN is a computer code written in FORTRAN used for reactivity calculations associated with shim rotation prediction or changes in macroscopic cross sections. However, it does not have the built in ability for shim rotation prediction automation. To accomplish shim rotation automation a python code was written that interacts with PertN. The only file that is needed for shim rotation automation for phase space interpolated perturbation theory is the standard NEWT input file (Appendix D).

The ATR input deck is placed in the working directory and the user enters the command "Spec_Pert.py." The python script will ask the user to enter the following:

- An initial guess for shim rotation [degrees],
- the name of the NEWT file [as described above],
- whether the user wants to minimize based on the number of processors [1] or time [2],
- the desired power split in the NW, NE, SW, SE lobe in units of fractional power,
- the tolerance for the reactivity [pcm],
- the tolerance for the power split [fraction power].

The python script will automatically run forward and adjoint calculations with NEWT, take the data from NEWT and run PertN, and continue this process until it has converged on both the specified reactivity tolerance and power split tolerance.

A walkthrough using the initial input deck found in Appendix D is as follows. The first step for shim rotation automating is to copy the input deck from Appendix D into the working directory. In this case, the input deck was saved as “ATR_Shim.txt”. The next step is to type the following on the command line of the working directory: “Spect_Pert.py.” The following questions will be asked along with the associated answer (Figure 45)



```
    Please enter guess of initial shim rotation (20-70): 70
    Please enter Newt file for shim rotation prediction: ATR_Shim.txt
    Please enter if minimizing number of runs 1) or time 2):")   2
    Please enter desired power split in the following format with NW-SE being fraction
adding up to 1.000
    [1., NW,  NE,  SW,  SE, 1.000]
    [1.   .23, .24, .26, .27, 1.000]
    Enter tolerance for step 1 in units of PCM (default is 150 pcm)  150
    Please enter tolerance for step 2 in units of fractional power (default is .03)   .03
```

Figure 45 Questions and answers for phase space interpolated perturbation theory shim rotation automation

Once all of the questions have been answered, the python script will create the necessary input decks and submit the necessary jobs on the cluster. The python script will then wait for the processors to finish with the scale calculations and then the python script will analyze the data and submit additional calculations as needed. When the python script has converged on with the desired tolerances for both the eigenvalue and power split the results for step1 will be 55.7 degrees and the results for step 2 will be 25.1 degrees, 42.5 degrees, 60.4 degrees, and 78.0 degrees for the NW, NE, SW, SE lobes respectively.

Shim Rotation Automation with Non_pert.py

The exact same procedure used for automating shim rotation prediction for phase space interpolated perturbation theory is used for running shim rotation automation using the cubic regression method with one exception. Instead of typing “Spec_pert.py” the user needs to type “Non_pert.py” in the command line of the working directory. The python script “Non_pert.py” will ask the same questions and automates the entire process until it has converged both on the desired tolerance for reactivity and for the desired tolerance for the power split.

APPENDIX C: METHOD USED FOR ROTATING DRUMS.

The outer shim control cylinders (OSCC) are rotating reflecting and absorbing cylinders located in the beryllium reflector surrounding the fuel. OSCC rotation is the primary method used to control the core. There are four OSCC per quadrant with a total of 16. N1 to N4 in the north quadrant, E1 to E4 in the east quadrant, S1 to S4 in the south quadrant and W1 to W4 in the west quadrant. Each of the OSCC has 120 degrees of its periphery covered by hafnium absorber plates. The hafnium absorber plates are 0.25 inches thick and 46 inches long. The hafnium plates move away from the core as the degree of rotation increases. Therefore increasing the degree of rotation will increase the reactivity of the reactor. The shims are rotated in groups of four. The four groupings are as follows:

- N3, N4, E1, E2
- E3, E4, S1, S2
- S3, S4, W1, W2
- W3, W4, N1 N2

The NEWT model of the ATR is created in such a way that shim rotation is accomplished by changing the angles on the ‘rot=’ card in the standard input deck (Appendix D). This makes rotating the shims relatively simple; however, when applying perturbation theory to shim rotation the mesh for the NEWT model cannot change. To use perturbation theory for shim rotation prediction a subroutine was added to the computer code PertN that could change the assignment of materials within the cells of a zone without having to modify the mesh.

Modifications of the assignment of materials within cells occur when a portion of the hafnium mesh is replaced with beryllium and a portion of the beryllium mesh is replaced with hafnium. This is accomplished by first determining which regions need to be modified, then finding the relationship of each cell within those regions, and finally swapping the material assignments. A picture of the cells within each zone for the OSCC shim can be seen in the Figure 46. The figure on the left hand side is a drawing of the shims meshed with NEWT. The figure on the right hand side is how PertN would change material composition for shims being rotated five degrees clockwise.

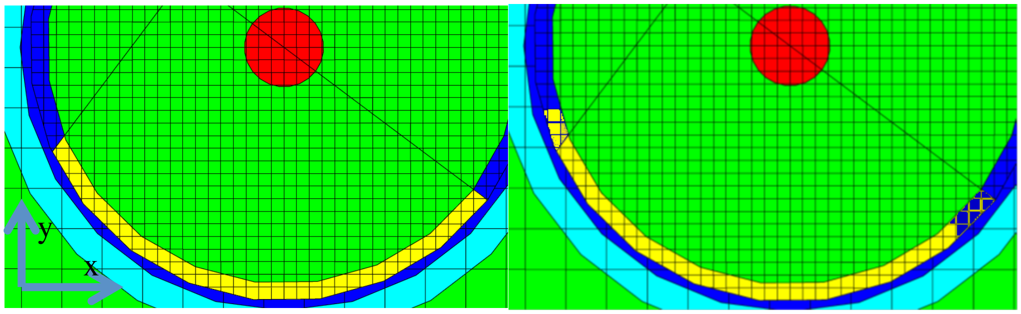


Figure 46 Picture of the OSCC in NEWT before (left) material assignment and after (right) material assignment

NEWT uses an arbitrary polyhedral geometry that can have 12+ sides for each cell and can be concave. The NEWT code saves the x and y coordinates for all of the vertices of each arbitrary polyhedron that make up an individual cell. However, NEWT does not save the relationship between each cell. To determine the relationship of the cells in respect to each other an algorithm was developed that utilized the vertices for each polyhedral cell within a zone.

The first step in the cell relationship algorithm was to find the average x position and y position for the cell by summing all of the vertices for each cell and dividing by the number of vertices. The average values for both the x and y coordinate system is then subtracted from the respective center of the arc that makes up the zone for each OSCC shim. Then the angle with respect to the positive x axis (Figure 46), and by working counter clockwise, is calculated for each cell and is compared to other cells. The cells are then order by incremental angular values to determine the relationship between each cell found within each zone or region. From this simple algorithm, the location of each cell within a zone can be determined and that information can be used to swap the material composition of a cell in a specified zone.

There are two possible directions needed for shim rotation, clockwise or counter clockwise. There are also two steps needed for material replacement, replacing beryllium with hafnium or replacing hafnium with beryllium. The algorithm for shim rotation is therefore broken up into four procedures depending on which OSCC shims are being rotated and direction of rotation. It is important to note that the 1st and 2nd shims in each of the four quadrants (N ,E, S, W) move in the opposite direction of the 3rd and 4th shims in the same quadrature. For example to move the shims away from the reactor in the North quadrant the N1 and N2 shims are moved counter clockwise and the N3 and N4 shims are moved clockwise.

Each of the hafnium absorber plates in the NEWT model occupies an area of 35.4009 cm^2. That is equivalent to 0.098346 cm^2 per degree. The user selects the size of the rotation then the algorithm will calculate that area within the hafnium plate that needs to be converted to beryllium starting with the boundary cell. The amount of area that the newly swapped cell occupies is subtracted from the total amount of area needed and then the next cell is reassigned beryllium. This iterative processes takes place until there is a negative area needed for swapping Hf with Be. Once a negative value has been determine, the final cell just before the negative value is not included in the list of cells that need to have the materials swapped. The code follows the same process for swapping Be with Hf in a similar manner.

As mentioned above, because the mesh cannot be changed for shim rotation there will be only discrete number of shim rotation angles possible. Therefore, improve the accuracy of shim rotation a more resolved mesh is needed. However, the more refined the mesh the longer the eigenvalue calculation will take to run.

One way to determine the mesh refinement for first order perturbation theory is to look at the average reactivity per degree of rotation. This can be approximated with an integral worth curve for shim rotation (Figure 47). The linear slope between 40 and 80 degrees is around $0.15 per degree that corresponds to around 100 pcm per degree. This means that a refinement of less than 1 degree will produce errors that will typically be less than 100 pcm. The refinement of the shim's mesh will depend on the desired tolerance.

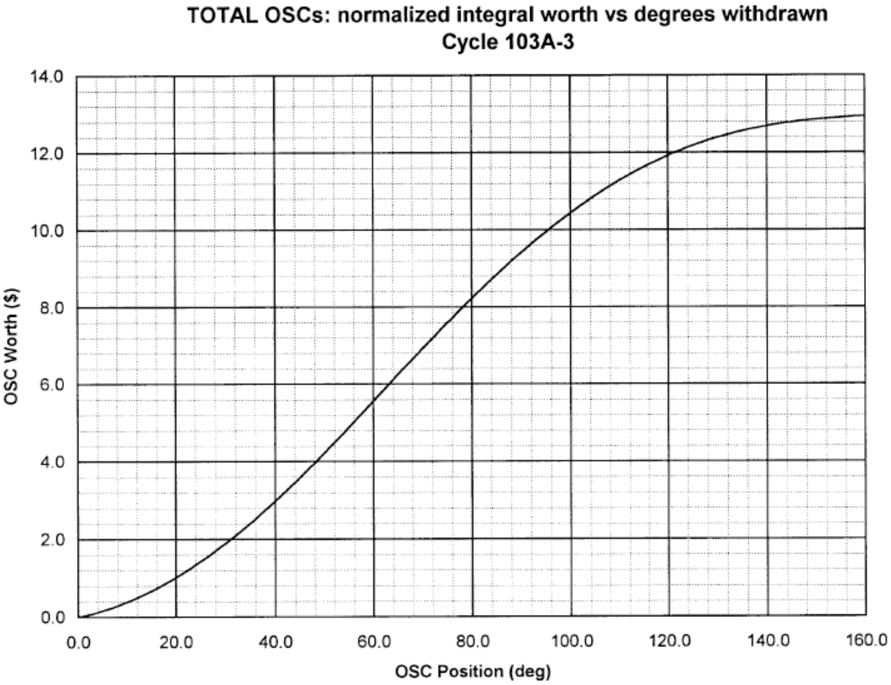


Figure 47 OSCC integral worth versus degree withdrawn [SAR-153]

Once the subroutine has finished reassigning hafnium and beryllium in relation to the desired shim rotation, PertN will continue as it does for other perturbation codes (See Study 1). For example, in one of the subroutines in PertN, it will compare the unperturbed material composition to the perturbed material composition cell by cell, as it does with perturbations from changes in density. PertN will only perform perturbations on the materials that have a change in composition. When the perturbation theory code is comparing the perturbed shims with the unperturbed shims it will see that the material assignments are different in some of the cells and will compute the reactivity effects from changing the material in those cells. The approximation of shim rotation can all be accomplished without having to redefine the mesh.

APPENDIX D: THE SIMPLIFIED INPUT DECK USED FOR BOTH PHASE SPACE INTERPOLATED PERTURBATION THEORY AND CUBIC REGRESSION METHOD.

```
=t-xsec
Create inf. homogeneous cross sections
44groupndf
read comp

' Side Plates Al-6061 T6
 h2o       1  den=1.1    1.000    310.9    end
 wtptsides  2     2.7   9
          14000    0.64
          26000    0.4
          29000    0.25
          25055    0.08
          12000    0.9
          24000    0.19
'          30000    0.03                              com='Zn not available used Cu'
          29000    0.03
          22000    0.02
          13027    97.49
          1.000    400   end

 h2o       3  den=0.99305    1.000    310.9    end

 be        4  den=1.850      1.000    310.9    end

 h2o       5  den=0.99305    1.000    310.9    end
' Hafnium
 wtpthf    6     13.31    2
          40000    2.0
          72000    98.0
          1.000    300     end

 he        8  den=1.785E-04  1.000    310.9    end
```

```
' OSCC Beryllium
atomoscc_Be  12    1.7916   3
           4009    0.96693
           8016    0.01102
           1001    0.02205
          1.000    310.9    end

' Fuel Meat

hf         16   den=13.31     1.000    310.9    end
be         17   den=1.850      1.000    310.9    end
be         18  den=1.850      1.000    310.9    end
hf         19  den=13.31      1.000    310.9    end

'be         16   den=1.850      1.000    310.9    end
'hf         17   den=13.31     1.000    310.9    end
'hf         18  den=13.31      1.000    310.9    end
'be         19  den=1.850      1.000    310.9    end
  ' Fuel Meat
atomfuelmeat   20   1.14 18
             92234   5.8138E-6
             92235   5.3838E-04
             92236   2.0175E-06
             92238   3.2277E-05
              5010   7.7152E-06
              5011   3.1055E-05
              6000   9.6925E-06
             13027   2.3258E-02
             14000   1.0764E-04
             26000   6.3152E-05
             29000   2.1804E-05
             25055   1.3757E-05
             12000   2.0730E-04
             24000   1.8896E-05
'            30000                                    com='Zn not available used Cu'
             29000   1.9263E-05
             22000   1.5785E-05
              1001   3.8813E-02
              8016   1.9056E-02
             1.000    310.9   end
```

```
 ' Fuel Meat
atomfuelmeat   21   1.14 18
           92234   5.8138E-6
           92235   5.3838E-04
           92236   2.0175E-06
           92238   3.2277E-05
            5010   7.7152E-06
            5011   3.1055E-05
            6000   9.6925E-06
           13027   2.3258E-02
           14000   1.0764E-04
           26000   6.3152E-05
           29000   2.1804E-05
           25055   1.3757E-05
           12000   2.0730E-04
           24000   1.8896E-05
'          30000                              com='Zn not available used Cu'
           29000   1.9263E-05
           22000   1.5785E-05
            1001   3.8813E-02
            8016   1.9056E-02
           1.000    310.9   end
' Fuel Meat
atomfuelmeat   22   1.14 18
           92234   5.8138E-6
           92235   5.3838E-04
           92236   2.0175E-06
           92238   3.2277E-05
            5010   7.7152E-06
            5011   3.1055E-05
            6000   9.6925E-06
           13027   2.3258E-02
           14000   1.0764E-04
           26000   6.3152E-05
           29000   2.1804E-05
           25055   1.3757E-05
           12000   2.0730E-04
           24000   1.8896E-05
'          30000                              com='Zn not available used Cu'
           29000   1.9263E-05
           22000   1.5785E-05
```

```
            1001   3.8813E-02
            8016   1.9056E-02
           1.000    310.9   end
' Fuel Meat
atomfuelmeat   23   1.14 18
           92234   5.8138E-6
           92235   5.3838E-04
           92236   2.0175E-06
           92238   3.2277E-05
            5010   7.7152E-06
            5011   3.1055E-05
            6000   9.6925E-06
           13027   2.3258E-02
           14000   1.0764E-04
           26000   6.3152E-05
           29000   2.1804E-05
           25055   1.3757E-05
           12000   2.0730E-04
           24000   1.8896E-05
'          30000                                    com='Zn not available used Cu'
           29000   1.9263E-05
           22000   1.5785E-05
            1001   3.8813E-02
            8016   1.9056E-02
           1.000    310.9   end

end comp
end

=newt
ATR Model Simplifed 1 Region Fuel

read parameter

accel=no  bf=1.420892 cmfd=no cmfd2g=no collapse=no combine=no
drawit=yes echo=yes epsinner=.00001 epsouter=0.00001 forward=yes height=0.0
html=no inners=4 kguess=1.017 mixtab=92 outers=950 premix=no prtbalnc=yes
prtbroad=yes prtflux=yes prtmxsec=yes prthmmix=yes prtmxtab=yes prtxsec=yes
restart=no run=yes savrest=yes sn=6 solntype=keff timed=yes wtdlib=30 xcmfd=4
xnlib=4 xycmfd=4  ycmfd=4

end parm
```

```
read materials
  1    1    'cold water '     end
  2    1    'side plates'     end
  3    1    'moderator'       end
  4    1    'reflector'       end
  5    1    'hotwater'        end
  6    1    'hafnium'         end
  8    1    'helium'          end
  12   1    'OSCC Be'         end
  16   1    'HF_2_BE_CW'      end
  17   1    'BE_2_HF_CCW'     end
  18   1    'BE_2_HF_CW'      end
  19   1    'HF_2_BE_CCW'     end
  20   1    'NW 1 Region Fuel'   end
  21   1    'NE 1 Region Fuel'   end
  22   1    'SW 1 Region Fuel'   end
  23   1    'SE 1 Region Fuel'   end


end materials

read geom

  global unit 1
  cuboid     2000  4p100.0                            com='pcs'
  cylinder   1990  100.0      sides=24                    com='outer water reflector'
  cylinder   1900  68.58      sides=24                    com='ref tank'
  media   2  1  1900                                  com='Al ref tank'
  media   3  1  1990  -1900                            com='outer water reflector'
  media   8  1  2000  -1990                            com='outer bounds'

' Outer Beryllium Reflector
  hole     50    rotate  a1=45                         com='north quadrant beryllium
reflector'
  hole     50    rotate  a1=315                         com='east quadrant beryllium
reflector'
  hole     50    rotate  a1=225                         com='south quadrant beryllium
reflector'
  hole     50    rotate  a1=135                         com='west quadrant beryllium
reflector'
```

```
' Aluminum Neck Shim Housing
  hole    53                                          com='neck shim housing'

' Fuel
  hole    100   origin   x=0.00000     y=21.59001    rotate a1=270    com='fuel elements
1,2'
  hole    100   origin   x=21.59000    y=21.59000    rotate a1=90     com='fuel elements
3,4'
  hole    100   origin   x=21.59000    y=21.59000    rotate a1=0      com='fuel elements
5,6'
  hole    100   origin   x=21.59000    y=21.59000    rotate a1=270    com='fuel elements
7,8'
  hole    100   origin   x=21.59001    y=0.00000     rotate a1=90     com='fuel elements
9,10'
  hole    101   origin   x=21.59001    y=0.00000     rotate a1=180    com='fuel elements
11,12'
  hole    101   origin   x=21.59000    y=-21.59000   rotate a1=0      com='fuel elements
13,14'
  hole    101   origin   x=21.59000    y=-21.59000   rotate a1=270    com='fuel elements
15,16'
  hole    101   origin   x=21.59000    y=-21.59000   rotate a1=180    com='fuel elements
17,18'
  hole    101   origin   x=0.00000     y=-21.59001   rotate a1=0      com='fuel elements
19,20'
  hole    102   origin   x=0.00000     y=-21.59001   rotate a1=90     com='fuel elements
21,22'
  hole    102   origin   x=-21.59000   y=-21.59000   rotate a1=270    com='fuel
elements 23,24'
  hole    102   origin   x=-21.59000   y=-21.59000   rotate a1=180    com='fuel
elements 25,26'
  hole    102   origin   x=-21.59000   y=-21.59000   rotate a1=90     com='fuel elements
25,26'
  hole    102   origin   x=-21.59001   y=0.00000     rotate a1=270    com='fuel elements
29,30'
  hole    103   origin   x=-21.59001   y=0.00000     rotate a1=0      com='fuel elements
31,32'
  hole    103   origin   x=-21.59000   y=21.59000    rotate a1=180    com='fuel elements
33,34'
  hole    103   origin   x=-21.59000   y=21.59000    rotate a1=90     com='fuel elements
35,36'
  hole    103   origin   x=-21.59000   y=21.59000    rotate a1=0      com='fuel elements
35,36'
```

```
  hole    103   origin   x=0.00000     y=21.59001    rotate a1=180    com='fuel elements
39,40'

'  Flux Trap Baffles
  hole    18    origin   x=0.00000     y=21.59001                  com='north flux trap
baffle'
  hole    18    origin   x=21.59000    y=21.59000                  com='northeast flux trap
baffle'
  hole    18    origin   x=21.59001    y=0.00000                   com='east flux trap baffle'
  hole    18    origin   x=21.59000    y=-21.59000                 com='southeast flux trap
baffle'
  hole    18    origin   x=0.00000     y=-21.59001                 com='south flux trap
baffle'
  hole    18    origin   x=-21.59000   y=-21.59000                 com='southwest flux trap
baffle'
  hole    18    origin   x=-21.59001   y=0.00000                   com='west flux trap
baffle'
  hole    18    origin   x=-21.59000   y=21.59000                  com='northwest flux trap
baffle'

'*********************************************************************
********
' Change the angle of the outer shim control cylinder by changing the 'a1' value
' to the desired rotation angle.
' REMEMBER: Shim numbers 3 and 4 must have a negative value in front of the
' angle to rotate the cylinders the right direction.
'*********************************************************************
********
' NE Outer Shim Control Cylinders
  hole    23    origin   x=10.16961    y=43.01896    rotate a1=-ROT_SHIM_NW
com='N3 OSCC'
  hole    24    origin   x=30.00890    y=44.37307    rotate a1=-ROT_SHIM_NW
com='N4 OSCC'
  hole    25    origin   x=44.37307    y=30.00890    rotate a1=ROT_SHIM_NW
com='E1 OSCC'
  hole    26    origin   x=43.01896    y=10.16961    rotate a1=ROT_SHIM_NW
com='E2 OSCC'

' SE Outer Shim Control Cylinders
  hole    27    origin   x=43.01896    y=-10.16961   rotate a1=-ROT_SHIM_NE
com='E3 OSCC'
```

```
  hole    28   origin   x=44.37307     y=-30.00890    rotate a1=-ROT_SHIM_NE
com='E4 OSCC'
  hole    29   origin   x=30.00890     y=-44.37307    rotate a1=ROT_SHIM_NE
com='S1 OSCC'
  hole    30   origin   x=10.16961     y=-43.01896    rotate a1=ROT_SHIM_NE
com='S2 OSCC'

' SW Outer Shim Control Cylinders
  hole    31   origin   x=-10.16961    y=-43.01896    rotate a1=-ROT_SHIM_SW
com='S3 OSCC'
  hole    32   origin   x=-30.00890    y=-44.37307    rotate a1=-ROT_SHIM_SW
com='S4 OSCC'
  hole    33   origin   x=-44.37307    y=-30.00890    rotate a1=ROT_SHIM_SW
com='W1 OSCC'
  hole    34   origin   x=-43.01896    y=-10.16961    rotate a1=ROT_SHIM_SW
com='W2 OSCC'

' NW Outer Shim Control Cylinders
  hole    35   origin   x=-43.01896    y=10.16961     rotate a1=-ROT_SHIM_SE
com='W3 OSCC'
  hole    36   origin   x=-44.37307    y=30.00890     rotate a1=-ROT_SHIM_SE
com='W4 OSCC'
  hole    21   origin   x=-30.00890    y=44.37307     rotate a1=ROT_SHIM_SE
com='N1 OSCC'
  hole    22   origin   x=-10.16961    y=43.01896     rotate a1=ROT_SHIM_SE
com='N2 OSCC'
  boundary   2000  15 15

'*********************************************************************
********
'*********************************************************************
********
'  Fill Units
'*********************************************************************
********

' Flux Traps
  unit 18
  cylinder   450   7.46125    sides=24                          com='flux trap baffle OR'
  media   2  1    450                                    com='water in flux trap'
  boundary   450
```

```
'Beryllium Reflector Attributes

' Outer Shim Control Cylinder N1
  unit 21
  cylinder   290   9.5250     sides=24                          com='OSCC water OR'
  media   3  1    290
  hole    41                rotate   a1=-99.717                 com='OSCC'
  boundary   290

' Outer Shim Control Cylinder N2
  unit 22
  cylinder   290   9.5250     sides=24                          com='OSCC water OR'
  media   3  1    290
  hole    41                rotate   a1=-113.049                com='OSCC'
  boundary   290

' Outer Shim Control Cylinder N3
  unit 23
  cylinder   290   9.5250     sides=24                          com='OSCC water OR'
  media   3  1    290
  hole    41                rotate   a1=-126.945                com='OSCC'
  boundary   290

' Outer Shim Control Cylinder N4
  unit 24
  cylinder   290   9.5250     sides=24                          com='OSCC water OR'
  media   3  1    290
  hole    41                rotate   a1=219.720                 com='OSCC'
  boundary   290

' Outer Shim Control Cylinder E1
  unit 25
  cylinder   290   9.5250     sides=24                          com='OSCC water OR'
  media   3  1    290
  hole    41                rotate   a1=170.280                 com='OSCC'
  boundary   290

' Outer Shim Control Cylinder E2
  unit 26
  cylinder   290   9.5250     sides=24                          com='OSCC water OR'
  media   3  1    290
  hole    41                rotate   a1=156.945                 com='OSCC'
  boundary   290

' Outer Shim Control Cylinder E3
  unit 27
  cylinder   290   9.5250     sides=24                       com='OSCC water OR'
  media   3  1    290
  hole    41               rotate   a1=143.049                com='OSCC'
  boundary   290

' Outer Shim Control Cylinder E4
  unit 28
  cylinder   290   9.5250     sides=24                       com='OSCC water OR'
  media   3  1    290
  hole    41               rotate   a1=129.717                com='OSCC'
  boundary   290

' Outer Shim Control Cylinder S1
  unit 29
  cylinder   290   9.5250     sides=24                       com='OSCC water OR'
  media   3  1    290
  hole    41               rotate   a1=80.283               com='OSCC'
  boundary   290

' Outer Shim Control Cylinder S2
  unit 30
  cylinder   290   9.5250     sides=24                       com='OSCC water OR'
  media   3  1    290
  hole    41               rotate   a1=66.951               com='OSCC'
  boundary   290

' Outer Shim Control Cylinder S3
  unit 31
  cylinder   290   9.5250     sides=24                       com='OSCC water OR'
  media   3  1    290
  hole    41               rotate   a1=53.055                com='OSCC'
  boundary   290

' Outer Shim Control Cylinder S4
  unit 32
  cylinder   290   9.5250     sides=24                       com='OSCC water OR'
  media   3  1    290
  hole    41               rotate   a1=40.280               com='OSCC'
  boundary    290

' Outer Shim Control Cylinder W1
  unit 33
  cylinder    290   9.5250      sides=24                              com='OSCC water OR'
  media    3  1     290
  hole     41                 rotate   a1=-9.728                   com='OSCC'
  boundary    290

' Outer Shim Control Cylinder W2
  unit 34
  cylinder    290   9.5250      sides=24                              com='OSCC water OR'
  media    3  1     290
  hole     41                 rotate   a1=-23.055                   com='OSCC'
  boundary    290

' Outer Shim Control Cylinder W3
  unit 35
  cylinder    290   9.5250      sides=24                              com='OSCC water OR'
  media    3  1     290
  hole     41                 rotate   a1=-36.951                   com='OSCC'
  boundary    290

' Outer Shim Control Cylinder W4
  unit 36
  cylinder    290   9.5250      sides=24                              com='OSCC water OR'
  media    3  1     290
  hole     41                 rotate   a1=-50.283                   com='OSCC'
  boundary    290

' Outer Shim Control Cylinder Hafnium/Beryllium
  unit 41
  cylinder    120   9.19480     sides=24                              com='OSCC beryllium large
OR'
  cylinder    110   9.19480     sides=24                              com='OSCC hafnium OR'
                          chord +x=0.0   chord +y=-4.4402
  cylinder    100   8.5598      sides=24                              com='OSCC hafnium IR'
  cylinder    090   1.42875     sides=24                              com='OSCC beryllium IR'
  media       3  1   90                                         com='OSCC central water'
  media       4  1  100   -90                                     com='inner beryllium'
  media       4  1  120   -110   -100                               com='outer beryllium'
  media       6  1  110   -100                                     com='OSCC hafnium'
  hole 60
  hole 65
  hole 70
  hole 75
  boundary   120  10 10
'  HF_2_BE_CW
  unit 60
  cylinder   250   9.19480    sides=24  chord +x=0.0   chord +y=7.0
  cylinder   100   8.5598     sides=24
  media     16  1  250   -100
  boundary   250  10 10
'  BE_2_HF_CCW
  unit 65
  cylinder   250   9.19480    sides=24  chord -x=0.0   chord +y=7.0
  cylinder   100   8.5598     sides=24
  media     17  1  250   -100
  boundary   250  10 10
'  BE_2_HF_CW
  unit 70
  cylinder   250   9.19480    sides=24  chord +x=4.0   chord -y=-4.002
  cylinder   100   8.5598     sides=24
  media     18  1  250   -100
  boundary   250 10 10
'  HF_2_BE_CCW
  unit 75
  cylinder   250   9.19480    sides=24  chord +x=7.40000   chord -y=-1.00
  cylinder   100   8.5598     sides=24
  media     19  1  250   -100
  boundary   250 10 10

' Beryllium Reflector
  unit 50
  cylinder   1890  64.29375   sides=24                        com='outer reflector'
             chord  +x=0  chord  +y=0
  media   4  1  1890                                    com='Be outside lobes'
  boundary   1890

' Aluminum Neck Shim Housing
  unit 53
  cuboid     1800  4p21.590798   rotate a1=0                  com='neck shim outer
bounds'
  cylinder   1790  6.83006   sides=24                         com='neck shim inner water'
  media    5  1  1790                                com='neck shim inner water'
  media    2  1  1800  -1790                            com='Al cruciform'
  boundary    1800

' NW Region Fuel
  unit 100
  cylinder    3000  14.18082    chord +y=0.0   chord +x=0.0   sides=24    com='water
channel 20 OR'
  media    20  1     3000                              com='channel 20'
  boundary    3000
' NE Region Fuel
  unit 101
  cylinder    3000  14.18082    chord +y=0.0   chord +x=0.0   sides=24    com='water
channel 20 OR'
  media    21  1     3000                              com='channel 20'
  boundary    3000
' SW Region Fuel
  unit 102
  cylinder    3000  14.18082    chord +y=0.0   chord +x=0.0   sides=24    com='water
channel 20 OR'
  media    22  1     3000                              com='channel 20'
  boundary    3000
' SE Region Fuel
  unit 103
  cylinder    3000  14.18082    chord +y=0.0   chord +x=0.0   sides=24    com='water
channel 20 OR'
  media    23  1     3000                              com='channel 20'
  boundary    3000

end geom

read bounds
  all=vac
end bounds

end
```

APPENDIX E: MODIFICATIONS MADE TO THE NEWT SOURCE CODE FOR USE WITH PERTN

Modification to the newt source code include the following subroutines or modules:

- Collapse.f90
- Ebindump2d.f90
- Input.f90
- Mixem.f90
- Readxn.f90
- Step1.f90

Collapse.f90:

This subroutine uses material fluxes to collapse cross-sections for each nuclide in a material to a broad-group form. To help reading in the collapse cross section the following line was commented out so that zero cross sections were also printed.

```
Before:
                  if(tmpstr(ipos:ipos+6).eq.'0.00000') tmpstr(ipos:ipos+10) = '

After:
!  Modified by Josh Peterson October 2 2009.  I like the clutter
!                 if(tmpstr(ipos:ipos+6).eq.'0.00000') tmpstr(ipos:ipos+10) = '
```

Ebindump2d.f90

This subroutine prints basic problem parameters, fluxes and flux moments to a binary file for the 2D ESC solution, for use by other codes. These parameters were also printed to multiple text files for use with PertN. The modifications made to Ebindump2d.f90 can be seen below:

```
_____________________
Before:
   use module_angle, only: ndir, nmom, r, wt, Ye

After:
   use module_angle, only: ndir, nmom, r, wt, Ye, mu, eta

______________________
Before: line 14
                               fluxmoms, momcount, sideflux, current,
cleakage,zone_aflux,bcflux, fluxrat
After:
                               fluxmoms, momcount, sideflux, current,
cleakage,zone_aflux,bcflux, fluxrat, angularflux
____________________
Before: line 18
   use module_xndata, only: chi, den, iftg, igm, maxpn, mxt, nmix, nuc, numen, sigfnu,
matpn, ebn, sigkaf
After: line 18-20
 !     Added sigs_z and sigma_t so I can obtain the information for perturbation theory
   use module_xndata, only: chi, den, iftg, igm, maxpn, mxt, nmix, nuc, numen, sigfnu,
matpn, ebn, sigkaf, &
                             sigs_z, sigt, mixs
_____________
Before: line 24
   integer :: imom,binfil,icell,i,j,ig,nsides,l,m,imat,lines,oldzones
After: line 26
   integer :: imom,binfil,icell,i,j,ig,k,nsides,l,m,imat,lines,oldzones, out2, out1, n
________________
Before: line 27
   double precision :: expandedflux(numcells,0:nmom), sumflx, sumpwr, volflux

After: line 29c27
   double precision :: expandedflux(numcells,0:nmom), sumflx, sumpwr, volflux, one
___________________
Before:

After: line 42-60

 if (lgpt) then

 elseif(forward)then
```

```
open(unit=20,status='unknown',form='unformatted',action='readwrite',position='rewind',fil
e='Pert_data.txt')

open(unit=21,status='unknown',form='formatted',action='readwrite',position='rewind',file=
'Pert_text.txt')

open(unit=22,status='unknown',form='formatted',action='readwrite',position='rewind',file=
'Other_spec.txt')

open(unit=25,status='unknown',form='formatted',action='readwrite',position='rewind',file=
'Power.txt')

 else

open(unit=20,status='unknown',form='unformatted',action='readwrite',position='append',fil
e='Pert_data.txt')

open(unit=21,status='unknown',form='formatted',action='readwrite',position='append',file=
'Pert_text.txt')

open(unit=23,status='unknown',form='formatted',action='readwrite',position='rewind',file=
'Adjoint_spec.txt')
   endif
147,148d125
     if(forward)
write(25,"(5x,i5,2x,i4,3(2x,1pe12.5))")i,zonemix(i),zoneflx(i),zonepwr(i),zonevol(i)
```

After: line 152

```
   if(forward) close(25)
```

____

Before:

After: line 165-173

```
  out2=out
 out=20;
 out1=21;
 if(forward) write(out) numzones,numcells,nummat,nmix,maxpn,ndir,nmom,numen, oldzones
 if(forward) write(out1,*) numzones,numcells,nummat,nmix,maxpn,ndir,nmom,numen, oldzones
 ! write(out)'Record 1'
 !write(out) 'Zones    Cells   Mixtur  Mixtab &
 !           & Scatter  Angles   Flux m  Energy  '
```

____

Before:

After: line 193-197

```
 if(forward) write(out) xglobal%minbc, &
               &xglobal%maxbc,yglobal%minbc,yglobal%maxbc,iorder,iftg,igm,wrklib
 if(forward) write(out1,*) xglobal%minbc, &
               &xglobal%maxbc,yglobal%minbc,yglobal%maxbc,iorder,iftg,igm,wrklib
```

____

Before:

After: line 200-202

```
 if(forward) write(out)10,-1,-1,-1,-1,iorder,iftg,igm,wrklib,title
 if(forward) write(out1,*)10,-1,-1,-1,-1,iorder,iftg,igm,wrklib,title
```

____

Before:

After: line 227-245

```
  ! if(forward) write(out)'Record 3 volume cells,        material,    numsides,  all
vertices'

  !do j=1,numcells
   if(forward) write(out)cell(:)%volume,cell(:)%material,cell(:)%numsides
 if(forward) write(out1,*)cell(:)%volume,cell(:)%material,cell(:)%numsides

 !   if(forward) write(out)
((vertices(j)%side(i)%x,vertices(j)%side(i)%y,i=1,cell(j)%numsides),j=1,numcells)
 ! if(forward)
write(out1,*)((vertices(j)%side(i)%x,vertices(j)%side(i)%y,i=1,cell(j)%numsides),j=1,numc
ells)

if (forward) then
do j=1,numcells
   do i=1,cell(j)%numsides
       write(out) ((vertices(j)%side(i)%x)), ((vertices(j)%side(i)%y))
       write(out1,*) ((vertices(j)%side(i)%x)), ((vertices(j)%side(i)%y))
   end do
end do
endif
```

____

Before:

After: line 257-263

```
 one=1.0
 ! if(forward) write(out)'Record 4 Weigts in angular qudrature and scattering constand
for flux moments'
!if(forward) write(out) wt(:),(one,(Ye(i,j),i=1,nmom),j=1,ndir) ! R(0,j) is always 1.0
and is not saved
!if(forward) write(out1,*) wt(:),(one,(Ye(i,j),i=1,nmom),j=1,ndir) !
if(forward) write(out) wt(:),(one,(Ye(i,j),i=1,nmom),j=1,ndir)! R(0,j) is always 1.0 and
is not saved
 if(forward) write(out1,*) wt(:),(one,(Ye(i,j),i=1,nmom),j=1,ndir) !
```

____

Before:

After; line 276-279

```
  if(forward) write(out)zonenum(:),zonemix(:),zonemixno(:)
 if(forward) write(out1,*)zonenum(:),zonemix(:),zonemixno(:)
```

____

Before:

After: line 289-291

```
 if(forward) write(out) mxt(:),nuc(:),matpn(:),den(:)
 if(forward) write(out1,*) mxt(:),nuc(:),matpn(:),den(:)
```

____

Before:

After: line 302-325

```
 if(forward) write(out) matid(:),((chi(i,j),j=1,numen),i=1,nummat),&
         ((sigfnu(i,j),j=1,numen),i=1,nummat),((sigt(i,j),j=1,numen),i=1,nummat),ebn(:)
 if(forward) write(out1,*) matid(:),((chi(i,j),j=1,numen),i=1,nummat),&
         ((sigfnu(i,j),j=1,numen),i=1,nummat),((sigt(i,j),j=1,numen),i=1,nummat),ebn(:)

 ! ***
 ! Record 7.5!!!!!!
 !  Sigma_scatter(i,j,k,l) i=1:numat j=1:numen k=1:numen l=1:matpn(imat)
 !  ***
 !if(forward) write(out) 'Record 7.5 SigmaT(mat,energy,enrgy,lorder)'
 if(forward) write(out) mixs
 if(forward) write(out1,*) mixs
 do i=1,mixs
   do j=1,numen
     do k=1,numen
       do l = 0,maxpn
          if(forward)    write(out) sigs_z(i,j,k,l)
          if(forward)    write(out1,*) sigs_z(i,j,k,l)
       end do
     end do
   end do
 end do
```

Before:

After: line 333-346d234

```
 if(forward) then
 ! write(out)'Record 8 keff'
 else
 ! write(out)'Record 8 keff for the Adjoint Flux'
 endif
  write(out) keff
  write(out1,*) keff
 !  This is for other spect when calculating exact perturbaiton
  if (forward) then
      write(22,*) keff
  else
      write(23,*) keff
  endif
```

Before:

```
   end do
```

After: line 373-399

```
      write(out) ((expandedflux(i,j),i=1,numcells),j=0,nmom)
    write(out1,*) ((expandedflux(i,j),i=1,numcells),j=0,nmom)
 !  This is for other spect when calculating exact perturbaiton
  if (forward) then
     write(22,*) ((expandedflux(i,j),i=1,numcells),j=0,nmom)
   else
     write(23,*) ((expandedflux(i,j),i=1,numcells),j=0,nmom)
  endif


 end do

 write(out) (((zone_aflux(n,icell,ig),icell=1,numcells),ig=1,numen),n=1,ndir)
 write(out1,*) (((zone_aflux(n,icell,ig),icell=1,numcells),ig=1,numen),n=1,ndir)



  if (forward) then
        write(22,*) (((zone_aflux(n,icell,ig),icell=1,numcells),ig=1,numen),n=1,ndir)
 !         write(22,*) (((angularflux(n,icell,ig),icell=1,numcells),ig=1,numen),n=1,ndir)

  else
         write(23,*)  (((zone_aflux(n,icell,ig),icell=1,numcells),ig=1,numen),n=1,ndir)
 !          write(23,*)
(((angularflux(n,icell,ig),icell=1,numcells),ig=1,numen),n=1,ndir)

  endif
```

```
Before: line 269
   do ig = 1,numen
After: line 407
!  do ig = 1,numen
```

```
Before: line 271-273
     write(binfil)((current(lines,ig,i), lines=1,numlines),i=1,6)
     write(binfil)(sideflux(lines,ig), lines=1,numlines)
     write(binfil)(cleakage(icell,ig), icell=1,numcells)
After: 409-411c271,273
  !   write(binfil)((current(lines,ig,i), lines=1,numlines),i=1,6)
  !   write(binfil)(sideflux(lines,ig), lines=1,numlines)
  !   write(binfil)(cleakage(icell,ig), icell=1,numcells)
```

```
Before: line 277
   end do
After: line 415c277
 !  end do
```

```
Before: line 289
   write(binfil) bcflux
After:  line 427
 !  write(binfil) bcflux
```

```
Before: line 293
After: line 432-435

   close(out)
   close(out1)
 out=out2
```

Input.f90

This subroutine reads in the parameter input from the NEWT input deck. It has been modified so that it saves the angular flux unless the user specifies not too.

```
Before:

     saveangular = .false.

After:
  !  saveangular = .false.
     saveangular = .true.
```

Mixem.f90

This subroutine prepares mixed macroscopic cross-sections using microscopic cross-section from the working library. This subroutine was modified to save the macroscopic cross sections into an ascii file called Cross_text.txt that can be read in by PertN.

Before: line 46

```
   integer :: ifirst,efirst,elast,pn,lpn,icount
```

After: line 41,42

```
 !  Modifed by Josh Peterson
   integer :: ifirst,efirst,elast,pn,lpn,icount, binary, text
```

Before:

After: line 46-57

```
 !  Changed by Josh Peterson for perturbation
 !
 logical :: perturbs = .true.
 integer :: l

 binary=20;
 text=21
open(unit=binary,status='unknown',form='unformatted',action='readwrite',position='rewind'
,file='Cross_data.txt')

open(unit=text,status='unknown',form='formatted',action='readwrite',position='rewind',fil
e='Cross_text.txt')
```

Before:

After: line 309-314

```
 !  Modified by Josh to work only with forward
         if (forward) then
             write(binary)sigt(n,ien),sigfnu(n,ien),chi(n,ien)
             write(text,*)sigt(n,ien),sigfnu(n,ien),chi(n,ien)
         endif
```

Before:

After: line 317-321

```
         if (forward) then
              write(binary)sigt(n,ien),sigfnu(n,ien),chi(n,ien)
              write(text,*)sigt(n,ien),sigfnu(n,ien),chi(n,ien)
          endif
```

Before: line 378

```
                 if(tmpstr(ipos:ipos+6).eq.'0.00000') tmpstr(ipos:ipos+10) = '
```

```
After: line 379-380
 !  Modified by Josh Peterson October 2 2009.  I like the clutter
!                if(tmpstr(ipos:ipos+6).eq.'0.00000') tmpstr(ipos:ipos+10) = '
'
___________________
Before:
After: line 388-398
!  Modifed by Josh to print Cross section informaotn only when is callled forward
   if (forward) then
      do j=1,numen
        do k=1,numen
          do l = 0,maxpn
             write(binary) sigs_z(n,j,k,l)
            write(text,*) sigs_z(n,j,k,l)
           end do
        end do
      end do
   endif
```

Readxn.f90

This subroutine reads required microscopic cross-sections from an AMPX working library for subsequent mixing into NEWT macroscopic cross-sections. It was modified to save all of the data in an easily read format even if the results were zero.

```
Before:
                  if(tmpstr(ipos:ipos+6).eq.'0.00000') tmpstr(ipos:ipos+10) = '

After:
 !  Modified by Josh Peterson October 2 2009.  I like the clutter
 !                  if(tmpstr(ipos:ipos+6).eq.'0.00000') tmpstr(ipos:ipos+10) = '
'
```

Step1.f90

This subroutine calls routines for reading input and pre-iteration problem setup. It was modified so when the parameter flag is set to not run then it will stop without having to spend time in the geometry section of NEWT. This was a quick way of generating cross sections.

Before:

After: line 38

```
   logical :: perturbs=.true.
```

____

Before

After: line 116-121

```
! *** Josh Peterson.  This just exits after runing for no and after printing cross
section info from mixem
if(perturbs.eq. .true.) then
      if(.not. run) call stopit(-1)
endif
```

REFERENCES

Allen, T. et al. *ATR National Scientific User Facility FY 2009 Annual Report*. Idaho National Laboratory: Idaho Falls, ID. Nov 2010.

ASME. Guide for Verification and Validation in Computational Solid Mechanics. ASME V&V, 10, 2006.

Beizer, B. *Software Testing Techniques*. Van Nostrand Reinhold: New York, 1990.

Brown, F.; Kiedrowski, B.; Bull, J. *MCNP Version 5*. [Software]: Los Alamos National Laboratory: Los Alamos, NM. 2002.

Carter, J. "Genetic Algorithms for In-Core Fuel Management and Other Recent Development in Optimization." *Advances in Nuclear Science and Technology*, 1997, 25.

Chang G.; Ambrosek, R.; Lilio, M. "Advanced Test Reactor LEU Fuel Conversion Feasibility Study: 2006 Annual Report." Idaho National Laboratory: Idaho Falls, ID, 2006.

Collins-Sussman et al. "Version Control with Subversion: For Subversion 1.5 Complied from r3305" [Software]. 2011.

Dahl, O. *Verifiable Programming*. Prentice-Hall: Englewood Cliff, NJ, 1992.

DeHart, M. D. "*NEWT: A New Transport Algorithm for Two-Dimensional Discrete Ordinate Analysis in Non-Orthogonal Geometry*," Vol. II, Sect. F21 of SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation, NUREG/CR-0200, Rev. 7 (ORNL/NUREG/CSD-2/R7), 2003.

Downar, T. "Light Water Reactor Fuel Cycle Optimization: Theory Versus Practice." *Advances in Nuclear Science and Technology*, 1988, 20, 1, pp 71-126.

Federowicz, A.; Stover, R. "Optimization of PWR Loading Patterns by Reference Design Perturbations." *Transactions of the American Nuclear Society,* 1973, 17, pp 308.

Geemert, R.; Hoogenboom, J. "Development of Parallelized Higher-Order Generalized Depletion Perturbation Theory for Application in Equilibrium Cycle Optimization." *Annals of Nuclear Energy,* 2001, 28, pp. 1377-1411.

Hatton, L. The Experiments: Errors in Scientific Software. *IEEE Computational Science & Engineering*, 1997, 4, 2, pp. 27-38.

Heesch, D. *Doxygen*. [Software] Jan 10, 2011

Ho, L. *Perturbation Theory in Nuclear Fuel Management*, Ph.D. Dissertation, Iowa State University: Ames, IA, 1981.

Ho, L.; Rohach, A. "Perturbation Theory in Nuclear Fuel Management Optimization" *Nuclear Science and Engineering*, 1986, 82, pp 151-161.

Hurwitz. H. *Report KAPL-98*. 1948: Knolls Atomic Power Laboratory.

Idaho National Laboratory. "Chapter 4 –Reactor." *Upgraded Final Safety Analysis Report for the Advanced Test Reactor*: Idaho National Laboratory: Idaho Falls, ID. SAR-153 Rev. 14, August 2010.

Kaner, C.; Falk, J.; Nguyen, H. *Testing Computer Software, 2nd edition*. John Wiley: New York, 1999.

Kim, D. el al. "A Generalized Perturbation Theory Program for CANDU Core Analysis" *Annals of Nuclear Energy,* 2001, 28 pp 169-189.

Kropaczek, D.; Turinsky, P. "In Core Nuclear Fuel Management Optimization for Pressurized Water Reactors Utilizing Simulated Annealing." *Nuclear Technology,* 1991, 95 pp 9-32.

Lamarsh, J. *Introduction to Nuclear Reactor Theory*. American Nuclear Society: La Grange Park, IL, 2002.

Lewins, J. *Importance: The Adjoint Function*. Pergamon Press: Oxford, 1965.

Lewis, E. Miller, W. *Computational Method of Neutron Transport*. American Nuclear Society: La Grange Park, IL, 1993.

Mingle, J. "In Core Fuel management Via Perturbation Theory." *Nuclear Technology*, 1975, 27, pp 248-257.

Montgomery, D.; Runger, G. *Applied Statistics and Probability for Engineers: Fifth Edition*. John Wiley & Sons, Inc.: New York, NY, 2010.

Moore, B. Higher Order Generalized Perturbation Theory for BWR In-Core Nuclear Fuel Management Optimization. Ph.D. dissertation, North Caroline State University, Raleigh, NC, 1996.

Moore, B.; Turinsky, P.; Karve, A. "Formosa-B: A Boiling Water Reactor In-Core Fuel Management Optimization Package." *Nuclear Technology,* 1999, 126, pp 153.

Neter, J.; Kutner, M.; Wasserman, W.; Nachtsheim, C. *Applied Linear Regression Models*. McGraw-Hill: Irwin, 1996.

Oberkampf W.; Trucano, T. *Verification and Validation in Computational Fluid Dynamics*. Sandia National Laboratory: Albuquerque, NM, 2002.

Ortiz, J.; Castillio, A.; Montes, J.; Persquia, R. "A New System to Fuel Loading and Control Rod Pattern Optimization in Boiling Water Reactors." *Nuclear Science and Engineering*, 2007, 157, pp 236-244.

Palmiotti, G.; Salvatores, M.; Broccoli, U.; Bruna, G.; Estiot, J; Gomit J. "Perturbation Methods for Power and Reactivity Reconstruction." *International Topical meeting on Advances in Reactor Physics Mathematics and Computation,* April 1987; Paris, France, pp 1413-1420.

Sauar, T. "Application of Linear Programming to In-Core Fuel Management Optimization in Light Water Reactors." *Nuclear Science and Engineering*, 1971, 46, pp 274-283.

Singh, G.; Kothari, L.; Garg, S. "One- and Two-Group Diffusion Theory for Eccentric Control Rods—I." *Annals of Nuclear Energy*, 1985, 12, 8, 1985 pp 391-397.

Sood, A.; Forster, R.; Parsons, D. "Analytical Benchmark Test Set for Criticality Code Verification." *Progress in Nuclear Energy* 2003, 42, 1, pp 55-106.

Stacey, W. *Nuclear Reactor Physics, 2nd*. Wiley-VCH, 2007.

Stevens, J.; Smith, K; Rempe, K. "Optimization of Pressurized Water Reactor Shuffling by Simulated Annealing with Heuristics." *Nuclear Science and Engineering*, 1995, 121, pp 67-88.

Stover, G.; Sesonske, A. "Optimization of BWR Fuel Management Using an Accelerated Exhaustive Search Algorithm." *Journal of Nuclear Energy*, 1969, 23, pp 673-682.

Suzuki, A.; Kiyose, R. "Application of Linear Programming to Refueling Optimization for Light Water Moderated Power Reactors." *Nuclear Science and Engineering*, 1971, 46, pp 112-130.

Turinksy, P. "Nuclear Fuel Management Optimization: A Work in Progress." *Nuclear Technology*, July 2005, 151, pp 3-8.

Williams, M.; Ronen III, Y. "Perturbation Theory for Nuclear Reactor Analysis." *CRC Handbook of Nuclear Reactors calculations Volume III*. CRC Press: Boca Raton, FL. pp. 63-188, 1986.

Wigner, E. P. "Effects of Small Perturbations on Pile Period." *Chicago Report P-G,* 3048, 1945.

Zhong, Z; Downar, T. "Benchmarking the U. S. NRC Neutronic Codes NEWT and PARCS with the Venus-2 MOX Critical Experiments." *Nuclear Mathematical and Computational Sciences: A Century in Review, A Century Anew*. April 6-11, 2003: ANS LaGrange Park, IL