

Abstract

BRIGGS, CHRISTOPHER MICHAEL. Multicycle Adaptive Simulation of Boiling Water Reactor Core Simulators. (Under the direction of Paul J. Turinsky).

Adaptive simulation (AS) is an algorithm utilizing a regularized least squares methodology to correct for the discrepancy between core simulators predictions and actual plant measurements [1]. This is an inverse problem that will adjust the cross sections input to a core simulator within their range of uncertainty to obtain better agreement with the plant measurements. The cross section adjustments are constrained to their range of uncertainty using the covariance matrix of the few-group cross sections and in imposing the regularization on the least squares solution. This few-group covariance matrix is obtained using the covariance matrix of the multi-group cross sections and the corresponding lattice physics sensitivity matrix. To perform the adaption, one must also have the sensitivity matrix of the core simulator. Constructing the sensitivity matrix of both the lattice physics code and core simulator would be a daunting task using the traditional brute-force method of computing a forward solve for a perturbation of every input. To avoid this, a singular value decomposition (SVD) is used to construct a low rank approximation of the covariance matrices, thus drastically reducing the number of required forward solves.

Until now, AS has been used on a single depletion cycle to correct for discrepancies resulting from errors introduced by incorrect cross sections only. Adapting to a single depletion cycle means that the cross sections of cycle m were adjusted so that the core simulator better predicts the actual measurements of cycle m (and future cycles if the algorithm is robust). This, however, does not account for the reloaded burnt fuel number density errors at the beginning-of-cycle (BOC) m . By definition a burnt assembly has been used and depleted in a

previous cycle. If adaption changes the cross sections of that burnt assembly in cycle m , those cross sections should have also been changed in any cycle preceding m which would have resulted in different BOC m number densities. This means that the number densities obtained using the original cross sections are not consistent with the newly adapted cross sections.

Hence, the number densities input to a core simulator are not the actual values in the reactor's fuel assemblies for the burnt fuel. This discrepancy in isotopics is another component to the discrepancy between the core simulator and actual observables. This means that the adaption algorithm is adjusting cross sections to account for number density errors.

It is the goal of this research to 1) remove these inconsistencies between the adapted cross sections and the burnt fuel BOC n number densities, and 2) ensure that adjusting cross sections to make up for number density errors does not corrupt the adaption. To do this, we assume that to best predict cycle n (by correcting both cross sections and BOC number densities of cycle n), one must adapt cycles m through $n-1$ simultaneously, where cycle m is the cycle in which the oldest assembly in cycle n is a fresh assembly. After adaption, the cross sections must be used to deplete from cycle m to n . This will remove the number density errors in two ways: 1) burnup healing, and 2) beginning the depletion of fresh assemblies in cycles m through $n-1$ with the correct cross sections. To ensure the cross sections adjustments are not overcompensating for the number density errors, we restrain their adjustment to stay near one standard deviation of their a prior values.

Multicycle Adaptive Simulation of Boiling Water Reactor Core Simulators

by

Christopher Michael Briggs

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements of the Degree of
Master of Science

Nuclear Engineering

Raleigh, North Carolina
2007

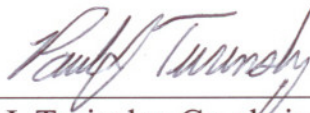
APPROVED BY:



Semyon V. Tsynkov



Hany S. Abdel-Khalik,
Co-chair of Advisory Committee



Paul J. Turinsky, Co-chair of Advisory Committee

Dedication

For my three M's.

Biography

Christopher M. Briggs was born in Houston, TX on February 2nd, 1983. He received his primary education locally in Houston, graduating from Scarborough High School in 2001. He attended Texas A&M and received his Bachelor of Science in May 2005 from the Department of Nuclear Engineering, graduating Magna Cum Laude. After A&M, Chris began his work towards a Master's of Science in Nuclear Engineering at North Carolina State University under the direction of Dr. Paul J. Turinsky. He has accepted a position at Westinghouse Nuclear in Dallas, TX and will begin work there upon completion of his Master's degree.

Acknowledgements

There isn't enough room on this page to properly thank those that made it possible for me to complete my master's work here at N.C. State. I would like to thank my family for the bottomless well of support that kept me going through the hardest and most stressful of times starting from day one. Their continuous self-sacrifice gave me all the time and resources I could have ever needed to be where I am today. I would also like to thank Dr. Turinsky for accepting me as a master's student and teaching more than I ever imagined I could learn in only two years. Next I would like to thank Dr. Paul Keller for his uncanny ability to fix FORMOSA B when I broke it, without which I would have never been able to get my research off the ground. At the opposite end of my research, I also want to thank both Dr. Abdel-Khalik and Mr. Matthew Jessee for the long hours they both contributed to ensure my final product withstood the most intense scrutiny. It is also worth noting the unwavering patience of Dr. Turinsky and Dr. Abdel-Khalik for withstanding the endless stream of mindless mistakes I put them through.

I would also like to thank my peers for their insight, support, and most importantly their constant source of comical relief. This includes Kenny Anderson, Doug DeJulio, Jason Harp, Ross Hays, Tyler Hickie, Matt Jessee, Tracy Stover, and Tim Wright. Lastly, even though things may not have worked out as planned, I want to thank Jen for always being there for me.

Table of Contents

List of Figures.....	vii
1. Introduction	1
1.1 Scope of Work	3
1.2 Core Simulators Overview.....	4
1.2.1 Core Simulator Basics	4
1.2.2 Various Core Simulators	5
1.2.3 Core Simulator Input	6
1.2.4 Simulation Errors	7
1.2.5 Simulator Output	9
1.3 Adaptive Simulation	9
1.3.1 Necessary Traits	10
1.3.2 Adaption Benefits	12
1.3.3 Previous Adaption Methods	12
2. Adaptive Simulation	14
2.1 Least Squares Development.....	14
2.1.1 Regularization Parameter.....	16
2.2 Least Squares Continued	19
2.3 Completing the Adaption.....	20
3. Design and Virtual Cores	25
3.1 Cross Section Perturbations.....	26
3.2 Number Density Perturbation	30
3.2.1 Core Simulator Software and Design Core	31
3.2.2 Introducing the Number Density Errors	32
4. Results	34
4.1 Pre-Adaption Number Density Behavior.....	34
4.1.1 Linear Response.....	34
4.1.2 Burnup Healing and Fresh Fuel Loading.....	35
4.2 Adaption Inputs	37
4.2.1 Regularization Parameter.....	37
4.2.2 keff Weight	38

4.2.3	Cycle Weights	39
4.3	Number Density Decrease vs. Adaption Inputs.....	40
4.3.1	Constant Cycle Weight	42
4.3.2	Constant keff Weight.....	47
4.3.3	Constant Alpha	48
4.3.4	Linearization Error.....	49
4.4	Summary	50
 5. Conclusions.....		65
 6. Future Work.....		67
 References.....		69

List of Figures

Figure 3.1: Singular values vs. principal direction.	28
Figure 4.1: k_{eff} and pcm errors of cycles 15 and 16 resulting from cross section and number density errors	52
Figure 4.2: k_{eff} and pcm errors of cycles 17 and 18 resulting from cross section and number density errors	53
Figure 4.3: $\text{RMS}_{\text{AC}}^{r,s}$ (red) and $\text{RMS}_{\text{AC,ND}}^{r,s}$ (blue) reduction for constant cycle weight.	54
Figure 4.4: k_{eff} (blue) and nodal power (red) misfit for constant cycle weight	55
Figure 4.5: $\langle \text{RMS} \rangle_{\text{AC}}^{r,s}$ (red) and $\langle \text{RMS} \rangle_{\text{AC,ND}}^{r,s}$ (blue) reduction for constant cycle weight	56
Figure 4.6: $\text{RMS}_{\text{AC}}^{r,s}$ (red) and $\text{RMS}_{\text{AC,ND}}^{r,s}$ (blue) reduction for constant k_{eff} weight	57
Figure 4.7: k_{eff} (blue) and nodal power (red) misfit for constant k_{eff} weight	58
Figure 4.8: $\langle \text{RMS} \rangle_{\text{AC}}^{r,s}$ (red) and $\langle \text{RMS} \rangle_{\text{AC,ND}}^{r,s}$ (blue) reduction for constant k_{eff} weight.	59
Figure 4.9: $\text{RMS}_{\text{AC}}^{r,s}$ (red) and $\text{RMS}_{\text{AC,ND}}^{r,s}$ (blue) reduction for constant alpha.	60
Figure 4.10: k_{eff} (blue) and nodal power (red) misfit for constant alpha.	61
Figure 4.11: Number density reduction for constant alpha	62
Figure 4.12: Cycle 18 k_{eff} and pcm error for each core	63
Figure 4.13: Nodal power RMS error: $\text{RMS}_{\text{DC, NP}}^r$ (blue), $\text{RMS}_{\text{AC, NP}}^r$ (green), and $\text{RMS}_{\text{VC}^*, \text{NP}}^r$ (black).	64

Chapter 1: Introduction

In the present day nuclear power industry, a reliable reactor core simulator for light water reactor (LWR) cores is a crucial part of reactor design, economics, and safety. The development of such a crucial tool is no trivial task, requiring design, construction, and testing that can take years. These steps can require an exorbitant amount of time and money before any worthwhile results are produced that justify the effort dedicated. Furthermore, even when the core simulator is complete, it will inevitably have its own limitations that prevent it from completely reproducing actual real plant data. To avoid this scenario, and for multiple reasons to be discussed later, there has been an effort to improve the input data to existing core simulators so that the simulator's prediction of a reactor's behavior are closer to that of the real world reactor. Throughout this paper, this effort is known as adaptive simulation.

Adaptive simulation is a methodology that makes use of both real plant data and core simulator output to change (adapt) the simulator inputs in such a way as to reduce the discrepancy between the two. Crucial to the process, adaption can be executed without changing any of the models used within the core simulator. In fact, adaption can account for different modeling algorithms. It was shown in previous work by Abdel-Khalik and Turinsky that adaptive simulation is capable of improving the agreement between two simulator's that use different thermal hydraulic models [1]. This is evidence that adaption can account for the shortcomings of a model's ability to correctly replicate the actual behavior inside the core. This enables the continued use of currently available core simulators rather than devoting precious resources to altering a currently available simulator or developing a new, superior simulator. Furthermore, note that a preliminary uncertainty analysis study [4] for a boiling water reactor (BWR) indicated that the differences between predicted and measured values of

core reactivity and power distribution are of the same magnitude as the uncertainties in these core attributes that originate due to nuclear data uncertainties, implying that pursuing the development of higher fidelity core simulator models may not prove beneficial.

This improvement in the accuracy of the simulator's output is accomplished by modifying its inputs in such a way as to improve the simulator's agreement with plant data. For core simulators, this includes microscopic cross sections and thermal hydraulic data. Due to the discrete nature of the simulator input and output data, the adaption can be performed by using well developed linear algebra techniques. The methods used to alter the input data prevent any adapted values from changing to a nonphysical value (such as a negative cross section), which would render the method ineffective. Further, adaption of input data must factor the uncertainty associated with this data by assuring the adapted data values are probable. It is important to note that even though the isotopic number densities are also inputs to a core simulator, these values are not adapted since they are not independent of the other inputs. If the number densities were adapted at the same time as the cross sections, there is no guarantee that the adapted number density values will be within the range predicted by the Bateman depletion equations using the adapted nuclear data.

Current adaption capability treats the change in number densities due to the change in adapted input data by solving the Bateman equations. However, if completing adaption on reload cores, which by definition contain partially burnt fuel from earlier cycles, one must address the issue that the isotopic number densities associated with the burnt fuel, which are input data, would be inconsistent with the adapted core simulation. To get the best adaption for cycle n , one would need to go all the way back and adapt cycle one followed by a cycle one depletion with the newly adapted cross sections. Starting at cycle one avoids the issue of

not using consistent number densities for burnt fuel since cycle one only contains fresh fuel. Cycle one adaption is done to update the number densities to correspond with the adapted cross sections, providing consistent number densities for burnt fuel that appears in the cycle two input data. One would successively repeat this process all the way to cycle n using the core simulator to ensure that the cycle n number densities are consistent. Since this is impractical for reactors that have been operating for a substantial time due to the high number of reload cycles and lack of pedigree of experimental data, the focus of this research is on the impact of starting the adaption on a cycle that is not the first cycle of a reactor's life, i.e. cycle m (where $1 < m < n$), and it's impact on the simulation of future cycles greater than m , in particular cycle n . This is accomplished by iterating, if necessary to correct for linearization errors associated with the adaption method, an adaption and depletion sequence starting at a cycle close to n . These iterations should eventually converge to the consistent number densities with respect to the adapted cross sections.

Due to simplicity of the pressurized water reactor (PWR) core relative to the BWR core, the fidelity of PWR core simulators is superior to that of BWRs. Currently, the prediction accuracy of BWR core attributes is such that large design margins are necessary to account for uncertainties, which adversely impacts power plant economics, e.g., cost of electrical energy generated. Therefore, the focus of adaptive simulation has been on BWRs, since these systems have the most room for improvement.

1.1: Scope of Work

The subsequent sections will discuss the general concepts and characteristics of core simulators and adaptive simulation. First the basics of core simulators, including types, inputs,

and sources of errors will be explored. This will be followed by the desired traits of the adaptive method and how we propose to satisfy these traits. The chapter will be concluded by discuss the benefits of cross section adjustment and its history.

The following chapter will present a terse derivation of the mathematics behind adaptive core simulation. Chapter three will describe the virtual approach used to create the measured observables, as opposed to using actual plant data. In Chapter Four, several cases will be investigated to determine the capabilities of adaption. Lastly, Chapter Five and Six will summarize our work and present some ideas for the future of adaptive core simulation.

1.2: Core Simulators Overview

As previously indicated, technological complexity of present day nuclear reactors has made the nuclear power industry heavily reliant on reactor simulators. Small scale tests and experiments are still vital to developing empirical models and collecting data, but these models inevitably serve as the foundation of some sort of computational recreation. This happens largely because repeatedly performing the small scale tests can quickly become impractical due to the time and costs of such procedures, and possible lack of applicability of scaling to the commercial reactor. Due to their importance to the nuclear industry, it is worthwhile to cover the fundamentals of core simulators.

1.2.1 : Core Simulator Basics

In general, the main purpose of a core simulator is to accurately model the neutronic and thermal-hydraulic behavior of a nuclear reactor. The core of a reactor is composed of many structures, including fuel assemblies, control rods, structural support, and monitoring equipment, to name a few. To provide reliable results, the core simulator must account for a

multitude of properties for these materials such as dimensions, compositions, and nuclear properties of the compositions for example. Using this information, the simulator then determines the distribution of neutrons, or neutron flux. The neutron flux, referred to as the flux from here on, is used to determine neutron interaction rates inside various media that make up the reactor core. The flux and reaction rates can then be used to calculate a wide variety of parameters used in design, control, safety, and other reactor fields, such as power distribution and material behavior.

1.2.2 : Various Core Simulators

One type of reactor core simulator is the online simulator. This is most important at a power plant site while the reactor is in operation. The reactor operators take advantage of the online simulator for a wide variety of functions. For example, online simulators aid in the monitoring of the current state of the reactor and predict if the reactor state will be shifting to an unsafe configuration. If the reactor is moving towards an unsafe situation, the simulator can help to correct this by advising the operator to manipulate the control systems. If the control systems aren't enough, the reactor will be shut down by activating the appropriate safety systems. Training simulators, another type of simulator, are used to train future and current reactor operators through the simulations of accidents and other various pedagogues. The other type of reactor simulator is the design simulator. This type performs an innumerable number of functions beyond that of the online and training simulators. The design simulator is used in the design process to determine the core loading pattern (LP), the control rod program (CRP) for the current fuel cycle, interpret physics tests, and evaluate various operational, thermal, and safety design margins.

1.2.3 : Core Simulator Input

Now that we have introduced the concepts of reactor core simulators, it will be beneficial to further elaborate on the type of simulator adapted in this research. This includes the assorted models employed, the input data required, the output obtained, and a brief discussion of the work that must be done to generate these input data. This will lead to where the simulations can go wrong and why. Combining the characteristics of the simulator with the impact and causes of the error will immediately reveal the driving force behind pursuing adaptive techniques.

To determine the flux, a variety of phenomena must be numerically modeled by the simulator. Most importantly, this includes neutron physics and thermal-hydraulics due to the nonlinear behavior of the former with respect to the latter and vice versa. One essential component of the neutron physics is the cross section. This is a fundamental parameter required for any calculation involving neutron interactions. The cross section of a material (or group of materials such as a fuel assembly) can be evaluated so that it describes the probability of neutron interaction (e.g. scatter or absorption) within a media based only on the incident energy of the neutron and current state (such as burnup and temperature) of the interacting media. The evaluation of these cross sections for a core simulator is performed by lattice physics codes. For a core simulator, these codes take the detailed cross section energy dependence and combine it with the complex geometry (from a traversing neutron's perspective) of a fuel assembly to collapse all the cross section information of the entire assembly down to a set of more manageable numbers. These new manageable numbers homogenize the spatial detail of the corresponding assembly that would be impractical to model within the core simulator. For instance, for a given assembly burnup, instead of the

cross sections being energy dependent in an almost continuous fashion, the lattice physics code produces cross sections that are held constant over several energy ranges, but done so as to preserve reaction rates. Also, the cross section is now spatially dependent on an assembly by assembly basis, rather than dependent on its exact location inside a specific lattice position in the core.

The number of cross sections input to a core simulator is extremely large. The exact number will depend on the models used within the simulator. For example, this can depend on the number of isotopes tracked in the fuel assemblies, the number of discrete neutron energy groups used, the number of time steps over the cycle length, and so on. Also, since the state of fuel at a given core location is constantly changing inside the core due to burnup, control rod movement, and thermal-hydraulic conditions, a cross section is evaluated for a variety of states [6]. The core simulator can then interpolate between the state point values to get the cross section that best represents the current state of the fuel at a specific core location. The end result is that the full set of cross sections alone constitute a copious amount of input data.

1.2.4 : Simulation Errors

In general, all complex numerical computations suffer from the same base set of weaknesses due to finite precision floating point arithmetic and modeling errors. These can be significant sources of error even with the current advances in computer technology. It is impossible to eliminate the effects of round-off error and its propagating effects on computations that require a high precision of accuracy even with employment of iterative correction methods. It is common practice to introduce simplifications to models employed to

reduce the run time to something practical relative to frequency of use. Core simulators are no exception to these issues.

To begin, there are neutron transport models that are capable of describing the flight behavior of individual or ensemble average of neutrons within the reactor core. This can be done stochastically, i.e., Monte Carlo, or deterministically, e.g., S_n method, with time, spatial, angular, and energy dependence of the flux represented with detail. However, this is computationally prohibitive to use in a routine fashion. Nuclear engineers developed a simpler model known as few-group, diffusion theory that uses the energy and angular integrated behavior of the neutrons to determine the flux. The assumptions made to employ the diffusion simplification come with a price however. On top of these diffusion simplifications, there are many numerical methods available to solve the few-group diffusion equations, each with their own strengths and weaknesses.

The simulator must also be able to handle the simplifications of the lattice physics codes. Many of these simplifications are done to aid diffusion theory. The detailed energy dependence of the cross sections is removed so that the few-group diffusion equations can be solved for the few-group flux. The fine spatial detail of the fuel assemblies is also removed from the cross sections by creating assembly averaged cross sections to implement diffusion theory on a coarse mesh. Also, the collapsed cross sections generated by the lattice physics codes are computed using vast amounts of experimentally tabulated data. These data bring along its own experimental uncertainties that propagate through the core simulator. As noted earlier, a preliminary assessment has indicated that for BWR cores, nuclear data uncertainties appear to introduce uncertainties in key core attributes of comparable magnitude to difference

in predicted and measured values of these attributes. This implies that other sources of uncertainties, such as due to modeling and numerical methods, may be of less significance.

1.2.5 : Simulator Output

The outputs of a core simulator that are important to adaption can be divided into two categories: core observables and core attributes. Core observables are reactor quantities measured by in-core instrumentation such as in-core detectors. These detectors are located throughout the BWR core by-pass in flow channel corners that do not contain control rods. It is these instrument readings that the core simulator is adapted to. The cross sections are changed in such a way that the core simulator predicted observables better agree with the plant observables.

Core attributes differ from core observables in that they are not directly measured. Examples of core attributes include local power peaking and thermal margins. It is crucial for an adapted simulator to also be capable of correctly predicting these quantities. These terms will be used to describe the associated output for the remainder of the paper.

1.3: Adaptive Simulation

As has already been indicated, adaptive simulation for core simulators is the process of using real plant data and core simulator output to adjust the simulator input in such a way as to improve the agreement between the two. All of the simulator's inadequacies create a discrepancy between the simulator's output and the collected data of an actual plant. Adaptive simulation is used to minimize the effects of the input data errors of a core simulator without altering any of the simulator's design. To be capable of successfully utilizing an adaptive approach, the simulation must have certain general qualities (fidelity, robustness, and practical

run times). If it does, the potential benefits of adaptive simulation go beyond merely a simulator that better matches real plant behavior. It is the pursuit of these three qualities that is the subject of this research. The following expands on these ideas.

1.3.1 : Necessary Traits

As previously indicated, the nuclear reactor simulator has become a ubiquitous part of nuclear design and operation. The examples given are only a small fraction of the many ways researchers are using simulators to develop and test better ways to overcome the latest obstacles of the nuclear industry. One obvious fact that was never mentioned, but needs discussion, is that all of the research utilizing adaptive simulations is worthwhile only if a capable simulator is available. Before one can reap the benefits of an adapted simulator, there are a number of required traits of the final product. Several important characteristics (among many) of an adapted simulator, for its use to be viable to engineers, include: high fidelity, robustness, and practical run times. The difficulty with globally quantifying each of these terms is that they are all relative to the purpose of the simulator. For the current intentions, fidelity denotes the ability of an adapted simulator to accurately predict the measured observables. Robustness is the ability of the adapted simulator to accurately predict core attributes which are not directly observed and for core operating conditions beyond those for which the core simulator had measurements to adapt to. This could be the measured observables recorded at future times for example [1]. Lastly, one could say that a practical run time would be one that is substantially shorter than the frequency of the simulator's intended use.

It is the last two items of this list (robustness and run time) that brings us to what is driving the current research, recognizing that as a side product improved fidelity should also be obtained. An adapted simulator will be extremely robust if it can accurately predict the core behavior of future cycles. To do this, the isotopic number densities of an adapted cycle must be updated to be within the range of the Bateman equations using the newly adapted cross sections. The Bateman equations are used to compute the change in isotopics as a fuel assembly is depleted. This can be done by starting the adaption at the first cycle, and sequentially adapting and depleting all the way to the cycle of interest, but this will violate the practical run time trait if the cycle of interest is relatively high. This may also be impossible if the required data from previous cycles is unavailable or of questionable pedigree to be adapted for whatever reasons. To resolve both of these issues, it is the subject of this research to attempt starting the adaption at a cycle close to the cycle of interest. To accomplish this, if the cycle of interest is cycle n and the cycle where the adaption will be started is cycle m (where $1 < m < n$), then cycles m through $n-1$ will be simultaneously adapted at one time. To update the number densities, the simulator will then use the adapted cross sections to deplete from cycle m to $n-1$. As will be explained later, the adaption over cycles m to $n-1$ may need to be recomputed, e.g., iterated, since the adaptive method to be utilized is only first-order accurate. Once iterations are completed, these final cross sections and number densities can now be used to predict cycle n and beyond. This process shrinks the run time needed for a robust adaption. Now all three traits have been satisfied.

1.3.2 : Adaption Benefits

The ability to reduce the error between simulation and actual plant measurements has significant impacts on nuclear reactor economics. The more accurate the simulator, the tighter the thermal margins can be that limit important characteristics of the reactor design due to the reduced uncertainty in the safety calculations. This can reduce the conservatism that has to be built into reactor design to compensate for simulator inadequacies. A more accurate simulator can reduce capital, operations and maintenance, and fuel costs. For example, if an adapted simulator is used to simulate a future cycle, one can potentially reduce the conservatism of the thermal margins, allowing a reduction in fuel cycle costs via a more aggressive core design or allow the core to be run at higher powers producing more electric energy to provide consumers.

Adaption is not only limited to changing the input parameters to provide the output with the smallest difference with observables. It can also be used to: estimate the bounds on the range of acceptable model parameters; estimate the formal uncertainties in the model parameters; show the sensitivity of the solution to perturbations in the data; determine the best set of data suited to estimate a certain set of model parameters; and compare different models.

[1]

1.3.3 : Previous Adaption Methods

There have been previous attempts at adjusting input data to get better agreement based on measured data. During the 1970s, researchers tried to manipulate the cross sections needed for fast reactors. The researchers used integral experiments to get the ‘actual’ plant data for which the cross sections would be adjusted to. An integral experiment is a critical configuration of

assemblies that operates at nearly zero power. The configuration of the assemblies is a small scale version of an actual fast reactor core.

There are several fundamental differences between the previous attempts at data adjustment and the current adaption algorithm. For one, the current method uses actual plant data, instead of these integral experiments. The disadvantage to this is the actual plant data may be distorted by feedback effects such as depletion and thermal-hydraulics. The data may also be corrupted if an instrument is out of calibration, or even worse had unknowingly failed. The advantage, however, is that there is a copious amount of core follow data from currently operating plants. Another difference is the way the data adjustment is performed. For the integral experiments, the researchers took advantage of Data Adjustment Techniques (DAT). Due to the discrete nature of the simulator input and output data, this adaption takes advantage of Discrete Inverse Theory (DIT). For ill-posed problems such as this one, DIT is much more robust due to its regularization approach.

Chapter 2: Adaptive Simulation

Since the focus of this research is not on the development of adaption itself, the mathematical details will only be reviewed - for full development see previous work [2]. The first part of this chapter is a mathematical description of the general least squares problem adaption is designed to solve. This will be followed by our multicycle adaption algorithm. In this discussion, it is assumed that the reader has a good understanding of linear algebra, least squares methodology, and inverse theory concepts.

2.1: Least Squares Development

To develop the least squares problem, it will be advantageous to first revisit the single cycle adaption, and then introduce our development of multicycle adaption. To begin, adaptive simulation of a core simulator is a substantial least squares problem in which the goal is to minimize the difference between the predicted and measured observables while simultaneously restricting the adapted parameters to their uncertainty bounds. Let the core simulator be represented as the following vector nonlinear equation:

$$\bar{d}_0^c = \bar{\Theta}(\bar{p}_0) \quad 2-1$$

where \bar{p} is a vector of dimension k whose components are the selected core parameters to be adapted. The 0 in the vector p_0 represents the known vector (i.e., core simulator input vector) that will be adapted to reduce the mismatch between the measured and predicted observables. The \bar{d}^c is a vector of dimension q whose components represent the predicted core observables.

The 0 in the vector \vec{d}_0^C is the calculated observable vector resulting from operating on \bar{p}_0 . The

adaption algorithm adjusts the parameters according to the following minimization problem:

$$\min \left\| \overline{\overline{W}}_d^{-1} (\vec{d}^m - \overline{\Theta}(\bar{p})) \right\|^2 \text{ subject to } \left\| \overline{\overline{W}}_p^{-1} (\bar{p} - \bar{p}_0) \right\| < \varepsilon \quad 2-2$$

where

$$\begin{aligned} \vec{d}^m &= \text{vector of measured core observables} \\ \overline{\overline{W}}_d \overline{\overline{W}}_d^T &= \overline{\overline{C}}_d \text{ (the observables covariance matrix)} \\ \overline{\overline{W}}_p \overline{\overline{W}}_p^T &= \overline{\overline{C}}_p \text{ (the core parameters covariance matrix)} \end{aligned} \quad 2-3$$

The input parameters for which the uncertainty is required in $\overline{\overline{C}}_p$ include the few-group cross sections and other neutronic parameters such as the diffusion coefficient. Since the uncertainty information of these input parameters is not readily available in the required few-group form, it must be calculated before the solution of Eq 2-2 can be found. Starting from scratch, the values are available in the very detailed ENDF point-wise format. The PUFF-III code developed at ORNL is able to propagate the point-wise uncertainties to the multi-group level. This is not sufficient though because the core simulator uses few group parameters. Previous work by Jessee[5] developed the capability using ORNL codes to compute the desired few-group covariance matrix and propagating the few-group uncertainties through the core simulator and its associated preprocessor codes. In this method, the rank deficiency of the covariance matrices is utilized to reduce the computational burden of such a calculation. The rank deficiency makes it beneficial to approximate the action of the covariance matrices by another set of matrices of much smaller size, the SVD factorization. The SVD factors can be calculated directly without ever storing or evaluating the original large matrices.

The covariance matrix for the observables ($\overline{\overline{C_d}}$) is currently an input to the code that performs the adaption. As will be explained later, since plant data are not actually used as the observables, a representative Gaussian noise error is introduced into the simulated measured core observables that are used for adaption (k_{eff} and nodal powers). The magnitude of the Gaussian noise error is reflected in $\overline{\overline{C_d}}$.

With the covariance data of the parameters and observables no longer unknowns, the minimization problem given in Eq 2-2 can be solved using adaptive simulation. The minimization equation can be rewritten as [3]

$$\min \left\{ \left\| \overline{\overline{W_d}}^T (\overline{\overline{d}}^m - \overline{\overline{\Theta}}(\overline{\overline{p}})) \right\|^2 + \alpha^2 \left\| \overline{\overline{W_p}}^{-1} (\overline{\overline{p}} - \overline{\overline{p_0}}) \right\|^2 \right\} \quad 2-4$$

The first term is known as the misfit term and the second is the regularization term. The α in the previous equation is known as the regularization parameter. Before further developing the multicycle least squares problem, it is worthwhile to explore this very important parameter that is used to diminish the adjustment of input parameters that could corrupt the robustness of the adaption.

2.1.1 : Regularization Parameter

It is the goal of adaption to focus on parameters that have a high uncertainty and strong sensitivity. This is so the difference between the measured and calculated observables can be minimized by adjusting important parameters that have room for substantial adjustment within uncertainty bounds. Adaption makes use of α to single out these parameters [8]. To

show the regularization parameter's impact, we must introduce the SVD. If we denote the Jacobian of the $\bar{\Theta}$ operator as \bar{A} ($q \times k$), then the SVD of \bar{A} can be written as

$$\bar{A} = \bar{U}\bar{S}\bar{V}^T \quad 2-5$$

where \bar{U} ($q \times q$) is a matrix composed of the left singular vectors, \bar{S} ($q \times k$) a matrix of the singular values, and \bar{V}^T ($k \times k$) a matrix of the right singular vectors. If we project $\Delta\bar{p}$ and $\Delta\bar{d}$ along the singular vectors as $\Delta\tilde{p}_j = \bar{V}_j^T \cdot \Delta\bar{p}$ and $\Delta\tilde{d}_j = \bar{U}_j^T \cdot \Delta\bar{d}$, then the un-regularized minimum norm least squares solution can be written as

$$\begin{aligned} \Delta\tilde{p}_j &= \frac{[\Delta\tilde{d}^m]_j}{s_j} \quad \text{where } 1 \leq j \leq q \\ \Delta\tilde{p}_j &= 0 \quad \text{where } q < j \leq k \end{aligned} \quad 2-6$$

where s_j is the j^{th} singular value of the \bar{A} operator, $\Delta\tilde{d}^m$ is the difference (in the left singular vector subspace) between the j^{th} measured and predicted observable, and $\Delta\tilde{p}_j$ is the resulting size of the adjustment of the j^{th} input parameter (in the right singular vector subspace). The regularization parameter is employed by modifying the first equation in Eq 2-6 to be

$$\Delta\tilde{p}_j = f_j \frac{[\Delta\tilde{d}^m]_j}{s_j} \quad \text{where } 1 \leq j \leq q \quad 2-7$$

where

$$f(s_j, e) = \frac{s_j^2}{s_j^2 + \alpha^2 e^2} \quad 2-8$$

The e in Eq 2-7 is the standard deviation of the measurement noise. The action of the regularization filtering can be seen by way of the following example:

- Regularization is used to filter out those parameters whose adjustment will be corrupted by the large amplification of inherent noise. First, rewrite the $\Delta \tilde{d}^m$ as $\Delta \tilde{d}^m = \Delta \tilde{d}^f + \Delta \tilde{d}^n$, where f represents the noise free component and n represents the noise term. Equation 2-6 can then be rewritten

$$[\Delta \tilde{p}_j] = \frac{[\Delta \tilde{d}^f]}{s_j} + \frac{[\Delta \tilde{d}^n]}{s_j} \quad \text{where } 1 \leq j \leq q \quad 2-9$$

Here it can be seen that if the singular value is small, the noise of the measurement is severely amplified, which can instantly ruin the adaption's fidelity and robustness. If α is large in the sense $\alpha e \gg |s_j|$, the $f(s_j, \epsilon)$ will be small and thus reduce the parameter adjustment in Eq 2-7.

The effect of a large α in the above example is to make the $\Delta \tilde{p}^m$ in Eq 2-6 as small as possible for parameters with high noise and/or low uncertainty. Adjusting such parameters will ruin both the fidelity and robustness of an adapted simulator. In the current work, the regularization parameter is determined by 'trial and error.' This involves adapting the cross sections with various magnitudes of α and selecting the best results based on whichever metric of interest is most important to the work being done (the RMS cross section adjustment in standard deviations for this work, to be discussed later). One could also produce an 'L-curve' to determine the best α . The L-curve is a plot in which the regularization term is on the ordinate axis and the misfit term is on the abscissa. This curve is shaped like an L, and the optimum α is located in the bend of the L, known as the knee.

2.2: Least Squares Continued

Returning to the least squares development, if one defines \bar{A} as the Jacobian of the core simulator such that

$$\bar{d}^c = \bar{\Theta}(\bar{p}) = \bar{d}_0^c + \bar{A}(\bar{p} - \bar{p}_0) + \text{Higher Order Terms} \quad 2-10$$

$$[\bar{A}]_{i,j} = \frac{\partial d_i}{\partial d_j} \quad 2-11$$

then the minimization problem can be rewritten as

$$\min \left\{ \left\| \bar{W}_d^{-1} (\Delta \bar{d}^m - \bar{A} \Delta \bar{p}) \right\|^2 + \alpha^2 \left\| \bar{W}_p^{-1} \Delta \bar{p} \right\|^2 \right\} \quad 2-12$$

If the regularization parameter is zero, then Eq 2-12 reduces to the standard least squares problem that solely uses the observables to adjust the parameters. If the regularization parameter approaches infinity the misfit term becomes negligible and Eq 2-12 keeps the a prior values. The mathematical methods used to solve this minimization problem can be found in [7]. To extend this to a multi-cycle adaption, the minimization equation is modified to contain the misfit terms for each cycle to be adapted and can be written as

$$\min_{\Delta p} \left\{ \sum_j w_j^2 \left\| \Delta \bar{d}_j^m - \bar{A}_j \Delta \bar{p} \right\|_{C_{d_j}^\dagger}^2 + \alpha^2 \left\| \Delta \bar{p} \right\|_{C_p^\dagger}^2 \right\} \quad 2-13$$

where

- j = cycle number
- w_j^2 = the weight of cycle j
- $\Delta \bar{d}_j^m$ = the difference between the measured and predicted observables for cycle j
- \bar{A}_j = the Jacobian of cycle j
- C_p^\dagger = the generalized inverse of the parameter covariance matrix
- $C_{d,j}^\dagger$ = the generalized inverse of the observables covariance matrix
- α = regularization parameter
- $\Delta \bar{p}$ = vector of cross sections adjustments

2.3: Completing the Adaption

Adaptive simulation solves the minimization problem given in Eq. 2-11. The purpose of formulating such a multicycle problem is because a single cycle adaption is not sufficient to fully update all of the core simulator inputs. This brings up two questions:

- 1) Why can only certain inputs be adapted?, and
- 2) Why are multiple cycles necessary?

The answer to the first question is because there are other inputs (number densities) to the core simulator that are dependent on the cross sections (the independent parameters). This dependency is a problem because if the dependent parameters are simultaneously adapted with their independent counterparts, there is no guarantee they will be within the range of their governing equations. The governing equation is the relationship that correlates the dependent parameters to the independent parameters. For example, consider isotope j such that it is only destroyed in a core, e.g. U^{235} . Its number density is given by the associated Bateman depletion equation

$$\frac{dN^{(j)}}{dt} = -\sigma_a^{(j)} \phi N^{(j)} \text{ with the initial condition } N^{(j)}(0) = N_0^{(j)} \quad 2-14$$

where

$N^{(j)}$ = isotopic number density

$\sigma_a^{(j)}$ = isotopic one-group absorption cross section

ϕ = neutron scalar flux

t = time

(This equation has spatial and time (burnup) dependence for all terms appearing in the equation, but we suppress notationally those dependencies for clarity.) Now if nuclear data are adjusted via adaption, $\sigma_a^{(j)}$ is directly changed in the adaption process and ϕ is changed due to its indirect dependence on $\sigma_a^{(j)}$. This will change the time dependence of $N^{(j)}(t)$ in Eq. 2-13, implying that $N^{(j)}(t)$ is dependent upon nuclear data adjustment and cannot be adjusted independently. The correct procedure is to obtain the adapted cross sections alone and then solve the depletion equations using the new cross sections to update the number densities. If the number densities are simultaneously adapted with the cross sections, the resulting number densities may not correspond to the values that would be determined using Eq. 2-13.

To answer the second question, note that if $N^{(j)}(t)$ and other isotopes' number densities change, so do macroscopic cross sections and hence flux; that is, the Bateman equations and neutron diffusion equation are coupled in a nonlinear fashion. In previous work on adaptive simulation, this coupling effect is addressed via a predictor-corrector method. However, previous work was limited to a single cycle. If the adaption is robust, the adapted nuclear data should also apply to earlier reload cycles than the cycle being adapted. The implication is that

the number densities associated with partially burnt fuel loaded into the reload cycle being adapted should be changed to be consistent with the adapted nuclear data. This aspect of adaptation was not addressed in the earlier work.

To put it another way, note that the initial condition $N^{(j)}(0)$, which corresponds to the beginning of cycle condition given in Eq. 2-13, was not updated in the adaptation. All of the subsequent number densities that are updated by using the adapted cross sections are based on the ‘un-updated’ initial condition. Since the initial number densities of fresh fuel are correct, this is only a concern for burnt fuel. Realizing that the initial number densities of one cycle are the final number densities of the previous cycle, to obtain the correct initial number densities, one would need to start adapting from the very first cycle of the reactor’s history, and adapt all the way to the cycle of interest. This would provide correct initial conditions for the cycle of interest. For reactors that have been operating for a substantial amount of time (practically every reactor in the U.S), this would 1) be computationally impractical, and 2) the quality of the old measured data used in the adaptation would be questionable at the very least. This burden is the driving force behind the current research.

Our hypothesis is that only starting several cycles before the cycle of interest, cycle n , is sufficient to get the correct initial number densities in cycle n . This is based on the fact that the number densities of all fresh fuel assemblies in cycle n are correct. This leaves only the burned assemblies in cycle n of interest to be corrected. To account for these burned assemblies, we propose to start the adaptation at the cycle, denoted cycle m , in which the oldest fuel assembly in cycle n was a fresh assembly. This way the adaptation spans all cycles in which every assembly in cycle n was a fresh assembly at some point. In most cases, this only means

going back around three cycles to start the adaption. This is much better than going back fifteen cycles for reactors that have been operating for extended periods of time.

Since the proposed adaption to remove number density errors is only first-order accurate due to linearizing the dependence of observables or parameters, an iterative procedure will be employed. The iterations will proceed as follows:

- 1) Adapt all cycles starting in cycle m and ending in cycle $n-1$ such that burnt fuel assemblies used in cycle n are loaded as fresh assemblies in one of these earlier cycles.
- 2) Using the adapted parameters, deplete all adapted cycles to update the number densities and determine the new calculated observables vector, i.e., $\bar{d}_{\Theta}^c = \bar{\Theta}(\bar{p})$
- 3) Check for convergence (see below for convergence method)
- 4) If converged, stop
- 5) If not converged, relinearize the problem (redetermine \bar{A}) about the updated parameter values and return to step 1.

To determine whether or not to continue the iterations, a stopping criteria must be satisfied. This is done by using the misfit term in Equation 2-12. If the misfit terms of the linear model are negligibly far from the misfit terms of the core simulator, then no iteration would be necessary. This is known as the linearization error. As will be discussed later, we are restricting our average adjustments to be near one standard deviation. If the linearization error is small at the upper limit of adjustments we are comfortable with, then updating the Jacobian would not provide much benefit.

Once this convergence is satisfied, we believe it is acceptable to assume the cross sections and associated number densities have been completely updated. This will allow for a better simulator prediction of cycles n and higher. As discussed in the introduction, a better prediction of future cycles will allow for substantially reducing cost by ways such as reducing fuel enrichment, increasing power density, and reducing control rod use.

Chapter 3: Design and Virtual Cores

As discussed in Chapter 2, the cross sections of cycles m through $n-1$ are adjusted within their uncertainties such that the predicted observables (core simulator output) better agree with the measured observables (plant data). There are two options available for providing the measured observables: 1) real plant data and 2) artificial plant data. Real plant data is actual measurements taken by the in-core detectors while the reactor is operating. For BWRs these detectors include local power range monitors (LPRMs) and traversing in-core probes (TIPs). Such detectors are generally located throughout the core between flow channels without control rods.

Conversely, artificial plant data is ‘measured’ observables generated by perturbing the inputs to a core simulator and simulating the LPRM and TIP readings, or any other measurement to be used in the adaption. For the research at hand, it was chosen to use the artificial plant data approach for multiple reasons. First, by doing this, we know the right answer since we produced it. This can be very beneficial in exploratory research such as this when there still may be undiscovered influences on the adaption results. This helps remove any unknown sources of error. Furthermore, there is no issue with detector drift and/or improper detector calibration. Finally, and most importantly, this allows for the easy introduction of cross section and number density errors. This makes it almost trivial to easily test the fidelity and robustness of the adaptive routine for a wide variety of cases. The set of artificially created measured observables is associated with what we refer to as the virtual core (VC). The set of predicted observables that is created using the original unperturbed cross sections is associated with what we refer to as the design core (DC).

3.1: Cross Section Perturbations

One of the two steps in generating the VC is to perturb the microscopic cross sections of all the lattice types used in cycles m through n . Note that special care must be taken to uniformly perturb any lattices repeatedly used in multiple cycles. This must be observed since the base cross sections and all the branch cases associated with lattice physics calculations of a lattice do not change just because the cycle of depletion has changed. To perturb the cross sections, two criteria must be satisfied:

- 1) The cross sections must be perturbed in a consistent fashion. This is done since it would not be physical to arbitrarily perturb the individual cross sections independent of one another. There is some degree of correlation between the cross sections that must not be ignored.
- 2) The VC must be created by perturbing only those cross sections with a sufficiently large uncertainty.

To fully explain how the cross sections are perturbed, the work of Jessee [5] must be briefly discussed. Part of their work consisted of propagating multi-group cross section uncertainties through the lattice physics code to the few-group cross sections, and then propagating the few-group cross section uncertainties through the core simulator to the core observables. The sheer number of inputs and outputs to both the lattice physics code ($\sim 10^3$) and core simulator ($\sim 10^6$) makes this a daunting task to complete using the traditional method of computing a forward solve for a perturbation in every input. This was overcome using low rank approximations of both the multi-group and few-group covariance matrices. After computing the covariance matrix of the multigroup cross sections (using ORNL PUFF-III), the few-group covariance matrix can be determined by

$$\bar{\bar{C}}_{FG} = \bar{\bar{A}} \bar{\bar{C}}_{MG} \bar{\bar{A}}^T \quad 3-1$$

where $\bar{\bar{C}}_{FG}$ is the few-group covariance matrix, $\bar{\bar{C}}_{MG}$ is the multi-group covariance matrix, and $\bar{\bar{A}}$ is the lattice physics sensitivity matrix. In determining $\bar{\bar{A}}$, advantage is taken of the low effective rank of $\bar{\bar{C}}_{MG}$ making possible the need for only r_{MG} forward lattice physics runs, where r_{MG} is the effective rank of $\bar{\bar{C}}_{MG}$. The effective rank of a matrix is the number of singular values whose magnitude is above a user-defined value. On a side note, $\bar{\bar{C}}_{FG}$ is the covariance matrix used in the regularization term of the minimization problem in Chapter 2.

Both of the criteria for creating the VC can be met utilizing the SVD of $\bar{\bar{C}}_{FG}$,

$$\bar{\bar{C}}_{FG} = \bar{\bar{U}} \bar{\bar{S}} \bar{\bar{U}}^T.$$

To consistently perturb the cross sections, we must utilize the left singular

vectors, i.e. the columns of $\bar{\bar{U}}$, called the principal directions from here on. Just like the covariance matrix itself, these principal directions contain the information that correlates adjustments of one cross section to the adjustment of another. The expression to consistently perturb the cross sections is

$$\bar{x} - \bar{x}_0 = \bar{\bar{U}} \bar{z} \quad 3-2$$

where \bar{x}_0 is the vector of reference cross sections, \bar{x} is the desired vector of perturbed cross sections, $\bar{\bar{U}}$ is the set of principal directions, and \bar{z} is any vector of expansion coefficients.

Given \bar{z} , this satisfies the first condition of cross section adjustment since we are using the

covariance information contained in $\bar{\bar{U}}$ to perturb the reference cross sections. Equation 3-2 will consistently adjust all of the cross sections according to their covariances.

Now that we can adjust the cross sections in a fashion consistent with the real world physics, we must ensure that only cross sections with sufficiently large uncertainties are being perturbed. If we order the singular values of $\bar{\bar{C}}_{FG}$ in decreasing order, a precipitous drop in their magnitude is observed.

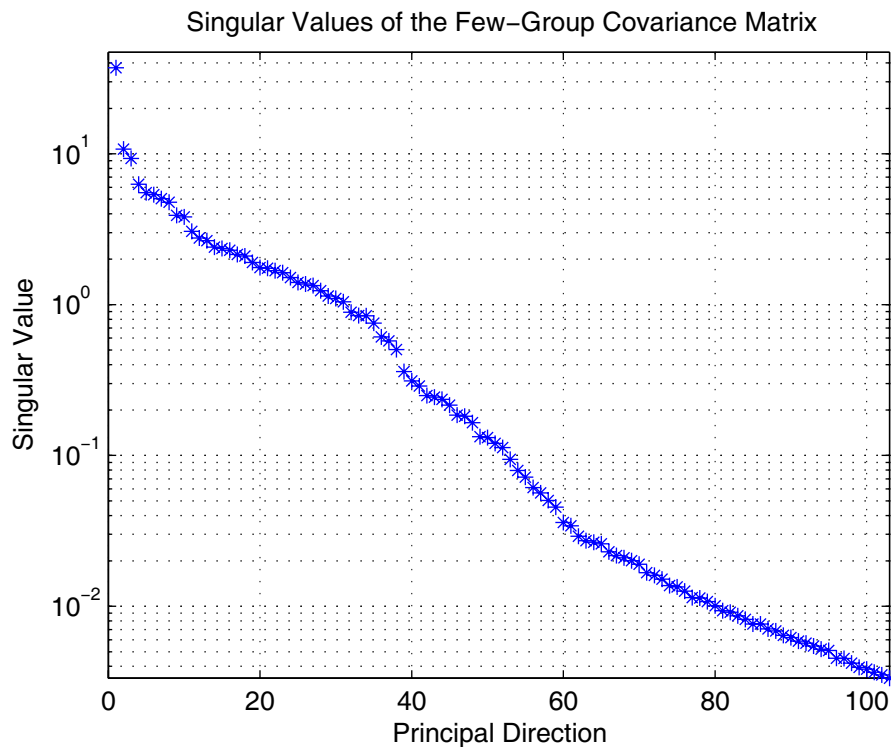


Figure 3.1. Singular Values vs. Principal Direction

Since the singular value of a principal direction is the variance of that direction, a very small singular value represents a very small uncertainty. It doesn't make sense to perturb a cross section that we have confidence in. The goal of adaptive simulation is to correct cross sections that we are uncertain about, so introducing errors in those values we have accurately measured would not test our adaptive algorithm since the regularization term would restrict

adapted values to be very close to a priori values. Thus we elect to only use a subset of $\bar{\bar{U}}$ principal directions in equation 3-2 to completely perturb those cross sections that are uncertain enough to justify adjustment. To accomplish this, we constructed a vector \bar{z}_{VC} composed of perturbed principal directions denoted by the following (the parenthesis have been added to aid the following discussion)

$$\bar{z}_{VC} = \sum_{i=1}^{r_{FG}} (\beta_i \sqrt{s_i}) \left(f \frac{\bar{u}_i}{\|\bar{u}_i\|_{\infty}} \right) \quad 3-3$$

where s_i is the singular value corresponding to the n_i principal direction, β_i is a random number selected from a uniform distribution, r_{FG} is the effective rank of $\bar{\bar{C}}_{FG}$, and f is a user input scaling factor. Within the first set of parenthesis, we randomly scale the singular value of each principal direction where the random number uniformly falls between 1 and -1. Since the singular value of a principal direction is equal to its variance, this randomly scales the uncertainty of each principal direction up to one standard deviation. We then normalize each principal direction with the infinity norm so that its largest element is one. This is then multiplied by the user defined scaling factor f . Normalizing and scaling the vector allows us to control the largest perturbation of any principal direction, equal to f . This produces two parameters available to generate different virtual cores: 1) the random seed used in generating $\{\beta_i\}$, and 2) the scaling factor f . Adjusting these two inputs becomes a game of trial and error to produce a virtual core that has representative perturbation sizes in core observables. Finally, as previously discussed, only the first r principal directions are perturbed because these are the only directions with a large enough uncertainty to warrant adjustment.

After the perturbed cross section set has been constructed through equations 3-2 and 3-3, they are run through the nonlinear core model to create

$$\bar{d}_{\text{VC}}^c = \bar{\Theta}(\bar{p}_{\text{VC}}) = \bar{\Theta}(\bar{p}_0 + \bar{U}\bar{z}_{\text{VC}}) \quad 3-4$$

where \bar{d}_{VC}^c is the set of virtual core observables and $\bar{\Theta}$ is the nonlinear core simulator. In real plants the core observables are measured using in- and out-of-core detectors. These readings are subject to detector noise, drift, incorrect calibration, and even detector failure. In an attempt to account for detector noise, 4% Gaussian noise is added to the core observables \bar{d}_{VC}^c . Currently this includes the nodal power of every node in the core and is represented as

$$\bar{d}_{\text{VC}}^c \big|_j = \bar{d}_{\text{VC}}^c \big|_j + \delta_j \quad 3-5$$

where j corresponds to any component of \bar{d}_{VC}^c subject to noise (nodal powers for this research), and δ_j is selected from a Gaussian distribution with a mean of zero and a standard deviation of 0.04.

3.2: Number Density Perturbation

As previously discussed, the goal of this research is to try to anneal out any errors in burnt fuel number densities in the cycle on interest. Before describing how the number density perturbations are introduced into the problem, it will be necessary to briefly explore the software employed. Once the mechanics of how the simulator transfers the number densities are clear, extending this to perturbing these values will be trivial.

3.2.1 : Core Simulator Software and Design Core

Although we had access to real world cycles that we could use in our core simulator, the data required to introduce realistic number density errors to these cycles were not available. The concept of realistic number density errors will be discussed in the next section. Since these data were unavailable, we improvised by taking advantage of the ‘multicycle restart’ [6] capability of FORMOSA. At the end of a depletion, FORMOSA’s multicycle restart function will generate a new loading pattern and all of the required input files. This new loading pattern can then be depleted, and so on. To create this new loading pattern, FORMOSA stores the beginning-of-cycle (BOC) k_{inf} profile of the current cycle to be depleted. After the cycle is depleted, the new loading pattern is generated by using the end-of-cycle (EOC) fuel and fresh bundle types to best match the BOC k_{inf} profile while retaining core symmetry.

To create the design core for our experiments, we started with a real world loading pattern and used the multicycle restart capability to deplete until all of the burnt fuel that was in the initial cycle had been discharged. This represents our initial design core cycle and initial reference number densities. Using previously established notation, this became cycle m in the multicycle adaption. Cycle n was created by continuing the multicycle depletion four more cycles. At this point, all of the fuel in BOC m was either discharged, or would be discharged by EOC n . Since the real world cycle used to initiate this depletion sequence was a cycle 12 core, cycles m through $m-1$ became cycles 15-17, and cycle n became cycle 18. Cycles 15-17 are used in the multicycle adaption as the design core because no burnt fuel assembly in cycle 18 goes back farther than cycle 15. This was the basic assumption driving this research. These cycle notations will be used in the next chapter when the results are presented.

3.2.2 : Introducing the Number Density Errors

Now that the multicycle approach has been introduced, the method used to introduce number density errors can be presented. To complete the VC, it is necessary to perturb the initial number densities of cycle m . Perturbing the cycle m BOC number densities will then propagate through the subsequent cycles to perturb their number densities. One must take care to perturb the number densities in a manner consistent with the perturbed cross sections. This means the BOC m number densities should be changed relative to the size of the cross sections perturbations. It is tempting to simply randomly perturb the BOC number densities about their initial values; however, this would lead to a nonphysical situation because of the correlation of isotopic cross sections with number densities of other isotopes. To see this, suppose that the cross section perturbation routine described previously leads to a decrease in the U-238 radiative capture cross section. This indirectly implies that the Pu-239 number density should decrease since Pu-239 production is initiated by neutron capture in U-238. This correlation can't be applied via a random number density perturbation. Instead of a random perturbation, the core simulator was used to account for these correlations between the change in the cross section of one isotope with the change in the number densities of other isotopes. The core simulator is capable of capturing these correlations contained in thermal hydraulic and neutronic feedbacks.

Ultimately, to introduce number density perturbations that are consistent with the cross section perturbations, we simply depleted from BOC 12 to EOC 18 with the perturbed cross sections. By the time the multicycle depletion sequence reached BOC 15, the number densities were different from those of the DC. Cycles 15 to 18, with perturbed cross sections and

associated perturbed number densities, served as the VC to generate measurement observables.

Chapter 4: Results

The current adaptive method was constructed to correct for discrepancies between the simulator and some other set of observables induced by cross sections errors only. As discussed in Chapter 3, for this research we introduced both cross section and number density errors so that we could test the abilities of adaption under influences the algorithm is not designed to correct. The following chapter will first discuss the impacts of cross section and number density errors. This will be followed by an examination of the inputs to adaption and their impact on the cross section adjustments. Finally, the results for a the three-cycle adaption will be presented. This means that cycles 15, 16, and 17 are adapted to enhance the simulator's prediction of cycle 18.

4.1: Pre-Adaption Number Density Behavior

Before discussing the abilities of the proposed adaption/depletion sequence, it is necessary to show the core's response to perturbing cross sections and/or number densities. The individual and combined effects of these cross section and number density errors for each cycle can be seen in Figures 4.1 and 4.2.

4.1.1 : Linear Response

For each cycle in the Figures 4.1 and 4.2, there are two graphs; the first shows k_{eff} for each time step of the cycle and the second shows the absolute difference between the design core (DC) and each other core's k_{eff} . The curves labeled 'DC' used the DC cross sections and the DC number densities. The DC curves shown are the actual set of reference responses used to produce the adaption results in the figures that follow. The curves labeled 'VC(XS)' used

the VC cross sections and the DC number densities. The curves labeled ‘DC(ND)’ used the DC cross sections and VC number densities. The curves labeled ‘VC’ used both the VC cross sections and number densities. The VC curves shown are the actual set of reference measurements used to produce the adaption results in the figures that follow. Most importantly these graphs show two things: 1) the combined effect of both sources of error are linear, and 2) the combined effect of burnup healing and fresh fuel loading successfully anneal number density errors. It is crucial to the entire problem that introducing number density perturbations will result in a linear response. If number density perturbations introduced a substantial nonlinear component to the simulator’s observables, the sensitivity matrix of the simulator would be unable to properly adjust the cross sections. Looking at Figures 4.1 and 4.2, we can see that for all cycles the DC(ND) pcm error and the VC(XS) pcm error sum up to the VC pcm error. This means that the number density errors do not introduce any substantial nonlinearities that would render the Jacobian unusable.

4.1.2 : Burnup Healing and Fresh Fuel Loading

The impact of burnup healing and fresh fuel loading can be seen by comparing either ‘DC’ to ‘DC(ND)’ or ‘VC’ to ‘VC(XS).’ In either of these comparisons, the only difference between the cases is the set of number densities used in the depletions. The following discussion will compare ‘DC’ to ‘DC(ND)’ since the pcm error graphs were created with respect to the DC. In the graphs that show the k_{eff} pcm error with the DC for each cycle, it is clear that the number density component of the total error decreases as the cycle depletions progress. In fact, the number density error has essentially disappeared by beginning-of-cycle (BOC) 17 partly because of burnup healing and mostly due to fresh fuel loading. Burnup healing hap-

pens because a core has a given flux shape driven by the characteristics of the core such as the material cross sections and fuel loading pattern. These characteristics generate a given leakage profile and produce a flux shape that is not highly sensitive to changes in the initial number densities. This flux shape will drive the perturbed number densities towards the unperturbed values as the core depletes. To understand this, assume that the number density perturbations cause the U-235 number density at a given node to increase which will cause the U-235 macroscopic cross section to also increase. For a given flux shape and magnitude, the increased cross section will cause the extra uranium introduced as number density errors to rapidly deplete back down towards the original number density.

In addition to burnup healing, the simple act of replacing old fuel assemblies with fresh fuel assemblies at the time of reloading the core has the largest impact on correcting the number density errors. This happens because there are no number density errors in the fresh assemblies. This can be seen by observing that the decrease in error between DC and DC(ND) occurs between each cycle loading. Furthermore, correcting the number density errors in the burnt fuel will improve the flux shape in the burnt assembly. This will cascade to all of the neighboring assemblies and improve those assemblies' flux shapes. This means that correcting burnt fuel isotopics will improve the flux shape of any neighboring fresh assembly, which will improve that assemblies' number density errors. This is cyclic, because 1) improving the fresh fuel power shapes will in turn improve other burnt assembly power shapes, and 2) obviously, after depletion, fresh fuel becomes burnt fuel which will then help correct the next batch of fresh fuel. Notably, it takes only one cycle to reduce the number density induced error by a factor of two. This is the case for the one-, two-, and three-cycle adaptations.

Equally important is the fact that both DC and DC(ND), and VC and VC(XS) converge to DC and VC, respectively. This suggests that depleting with any given cross section set will remove any number densities errors by returning to the number density values to those consistent with the given cross section set. This is very important because it shows that if our adapted cross sections match the VC cross sections, we can be guaranteed the AC number density errors will anneal out over several cycle depletions to their exact VC values.

4.2: Adaption Inputs

To adapt a given core, there are several input options that must be defined by the user to emphasize and/or constrain certain aspects of the adaption algorithm. It was necessary to adjust all of these inputs over a range of values in order to fully investigate the ability of multicycle adaptive simulation to minimize number density errors. To understand their impact on the final results, it is beneficial to review the objective function introduced in Chapter 2.

$$\min_{\Delta\bar{p}} \left\{ \sum_j w_j^2 \left\| \Delta\bar{d}_j^m - \bar{A}_j \Delta\bar{p} \right\|_{C_{d,j}^\dagger}^2 + \alpha^2 \left\| \Delta\bar{p} \right\|_{C_p^\dagger}^2 \right\} \quad 4-1$$

where the terms of equation 4-1 were previously defined in Chapter 2.

4.2.1 : Regularization Parameter

The regularization parameter constrains the adjustment of the DC cross sections. As alpha increases, the objective function becomes highly sensitive to changes in the cross sections and thus restricts their adjustment. As alpha approaches infinity, the adapted cross sections are identical to the DC cross sections. Conversely, a low alpha value allows more freedom for adjustment. An alpha of zero reduces to the standard least squares problem and allows the algorithm to adjust the cross sections irregardless of their uncertainties.

In the figures that will follow, the largest α was selected to keep the RMS of the cross section adjustment around one standard deviation. Two smaller values of alpha were also used in the adaption to show what sort of improvement could be gained by allowing the adaption routine a little more freedom to adjust the cross sections. It was decided to keep the RMS of the adjustment around one standard deviation to keep the cross sections within their range of uncertainty. Allowing too much freedom would not be robust since the cross sections would be adjusted outside of their experimental range of uncertainty. Not allowing enough freedom would not be very useful because the cross sections would be restricted to stay near their DC values.

4.2.2 : k_{eff} Weight

The k_{eff} weight is a multiplier contained within C^\dagger_{dj} . Each row of C^\dagger_{dj} corresponds to a particular observable output by the core simulator. For this research, those observables consist of k_{eff} and core wide nodal powers for each time step of each cycle. In the results that follow, a weight of 1.0 was applied to the nodal powers and the k_{eff} weight was varied. Adjusting the weight on k_{eff} varied the importance the adaption gave to correctly predicting k_{eff} . For example, a large k_{eff} weight would force the adaption algorithm to focus on matching k_{eff} more than the nodal powers in order to optimally minimize the objective function.

To select a sample of k_{eff} weights for use in the figures that follow, a k_{eff} weight that produces good results was found by trial and error. Two more k_{eff} weights were then chosen by increasing and decreasing the good k_{eff} weight by one order of magnitude to ensure a wide range of responses.

4.2.3 : Cycle Weights

The cycle weights of the objective function enable one to weight which cycle's set of measured observables contribute more to the cross section adjustment. This is done by multiplying each cycle's misfit term by a weight constructed from the user defined weights (discussed below). A large cycle weight will increase the sensitivity of the objective function to that cycle's misfit term simply by increasing its magnitude relative to the other cycles. In our research the cycle weights in Equation 4-1 are computed by

$$w_i = \frac{W_i}{\|\bar{W}\|_2} \quad 4-2$$

where $\{W_i\}$ are the user defined weights for cycle i and \bar{W} is a vector of the user defined cycle weights. The weights are computed this way for two reasons: 1) for a constant alpha of a 3-cycle adaption, a user input of $\bar{W} = [a \ a \ a]$ will produce identical results to $\bar{W} = [b \ b \ b]$, and 2) for a constant alpha, a 3-cycle adaption with $\bar{W} = [a \ b \ c]$ will weight the misfit term the same as a 2-cycle adaption with $\bar{W} = [\alpha \ \beta]$ and 1-cycle adaption with $\bar{W} = [\gamma]$. This is true because

$$\sum_{i=1}^j w_i^2 = \sum_{i=1}^j \left[\frac{W_i}{\|\bar{W}\|_2} \right]^2 = \sum_{i=1}^j \frac{W_i^2}{\sum_{i=1}^j W_i^2} = \frac{1}{\sum_{i=1}^j W_i^2} \sum_{i=1}^j W_i^2 = 1 \quad 4-3$$

In the figures that follow, the cycle weights were selected by initially setting the weight for each cycle to 1.0. Keeping the weight of the first cycle at 1.0, the other cycle weights were gradually increased to put more and more weight on the final cycle of the adaption. This is

meant to emphasize 1) the cycle in which we expected the errors caused by number densities to be the smallest, and 2) the cycle which will be transferring its assemblies to cycle 18 (the cycle of interest). Emphasizing the cycle with the smallest number density errors implies that adapted cross sections should be closer to the VC cross sections since they need not also have to correct as much for number density errors. Emphasizing the cycle directly before the cycle of interest will focus on correcting those cross sections significant to predicting the future cycle's behavior.

4.3: Number Density Decrease vs. Adaption Inputs

Two parameters are introduced to quantify the reduction of the error in number densities. Just as with the linearity test introduced in Chapter 2, we will again use k_{inf} to measure how well the number density errors are being corrected over the course of the adaption/depletion sequence. We use k_{inf} because it is sensitive to the number densities that are important to core behavior and ignores those that don't have much impact. This means a decreasing difference in k_{inf} represents a decreasing difference in those number densities with a large influence on core behavior. To develop our metrics, we use k_{inf} computed as a function of the following

$$k_{\text{DC}}^{i,j,r,s} = k_{\text{inf}}(\text{DC}_{\text{ND}}^{i,j,r,s}, \text{DC}_{\text{XS}}^{i,j,r,s}) \quad 4-4$$

$$k_{\text{AC}}^{i,j,r,s} = k_{\text{inf}}(\text{AC}_{\text{ND}}^{i,j,r,s}, \text{AC}_{\text{XS}}^{i,j,r,s}) \quad 4-5$$

$$k_{\text{VC}}^{i,j,r,s} = k_{\text{inf}}(\text{VC}_{\text{ND}}^{i,j,r,s}, \text{VC}_{\text{XS}}^{i,j,r,s}) \quad 4-6$$

$$k_{\text{DC,ND}}^{i,j,r,s} = k_{\text{inf}}(\text{DC}_{\text{ND}}^{i,j,r,s}, \text{VC}_{\text{XS}}^{i,j,r,s}) \quad 4-7$$

$$k_{AC,ND}^{i,j,r,s} = k_{\text{inf}}(AC_{ND}^{i,j,r,s}, VC_{XS}^{i,j,r,s}) \quad 4-8$$

where DC_{ND} is the set of DC number densities, DC_{XS} is the set of DC cross sections, AC_{ND} is the set of AC number densities, AC_{XS} is the set of AC cross sections, VC_{ND} is the set of VC number densities, VC_{XS} is the set of VC cross sections, i is an axial node, j is a fuel assembly, s is a cycle used in the adaption, and r is a time step of cycle s . Using these definitions, our metrics of number density error can be written as

$$RMS_{DC}^{r,s} = \sqrt{\frac{1}{N \cdot z} \sum_{j=1}^N \sum_{i=1}^z (k_{VC}^{i,j,r,s} - k_{DC}^{i,j,r,s})^2} \quad r \in s \quad 4-9$$

$$RMS_{AC}^{r,s} = \sqrt{\frac{1}{N \cdot z} \sum_{j=1}^N \sum_{i=1}^z (k_{VC}^{i,j,r,s} - k_{AC}^{i,j,r,s})^2} \quad r \in s \quad 4-10$$

$$RMS_{DC,ND}^{r,s} = \sqrt{\frac{1}{N \cdot z} \sum_{j=1}^N \sum_{i=1}^z (k_{VC}^{i,j,r,s} - k_{DC,ND}^{i,j,r,s})^2} \quad \text{where } r = 1 \forall s \quad 4-11$$

$$RMS_{AC,ND}^{r,s} = \sqrt{\frac{1}{N \cdot z} \sum_{j=1}^N \sum_{i=1}^z (k_{VC}^{i,j,r,s} - k_{AC,ND}^{i,j,r,s})^2} \quad \text{where } r = 1 \forall s \quad 4-12$$

where N is the number of fuel assemblies and z is the number of axial nodes in a fuel assembly. Our first measure of number density error (Equation 4-10) shows how close the adapted k_{inf} resulting from AC cross sections and AC number densities (Equation 4-5) gets to the VC k_{inf} (Equation 4-6). Our second measure of number density error (Equation 4-12) shows the behavior of k_{inf} due to the difference in VC and AC number densities alone. To compute this metric, the core simulator is depleted using the adapted core (AC) cross sections to obtain the AC number densities. A sort of hybrid k_{inf} is then computed using the VC cross sections and

the AC number densities (Equation 4-8). The node wise RMS of this hybrid k_{inf} is then computed with the VC node wise k_{inf} (Equation 4-12) to single out the difference between the AC and VC number densities. The I/O of the simulator makes it possible to compute Equation 4-12 (and Equation 4-13) only for the first time step of each cycle. In order to see the decrease of the AC number density errors relative to the initial DC number density errors, these metrics are then normalized by the DC k_{inf} error as follows:

$$\langle \text{RMS} \rangle_{\text{AC}}^{r,s} = \frac{\text{RMS}_{\text{AC}}^{r,s}}{\text{RMS}_{\text{DC}}^{r,s}} \quad 4-13$$

$$\langle \text{RMS} \rangle_{\text{AC,ND}}^{r,s} = \frac{\text{RMS}_{\text{AC,ND}}^{r,s}}{\text{RMS}_{\text{DC,ND}}^{r,s}} \text{ where } r = 1 \forall s \quad 4-14$$

It is possible to present our findings now that all metrics used to measure the number density error have been introduced. To examine the response of the number density error, the following subsections will hold one of the input parameters constant and vary the other two.

4.3.1 : Constant Cycle Weight

Due to the same behavior for the constant cycle weight graphs, only case $\bar{W} = [1 \ 1 \ 1]$ will be discussed here (Figures 4-3 through 4-5). The title of each graph tells the alpha used and the resulting RMS cross section adjustment in standard deviations. The left hand side of each row tells the k_{eff} weight used for all three graphs of that row. The numbers in the box at the bottom of the figure are the user defined weights $\{W_i\}$ for every graph in the figure.

Figure 4.3 shows $\text{RMS}_{\text{AC}}^{r,s}$ (Equation 4-10) as the red lines and $\text{RMS}_{\text{AC,ND}}^{r,s}$ (Equation 4-12) as the blue lines. Both curves must be used to decipher if adaption is computing the correct cross sections and number densities. Just looking at only the red or blue line is not enough to interpret the results. We want two things: 1) the blue line to be as low as possible, and 2) the red line as close to the blue line as possible. A low blue lines suggests that the number densities of the AC have converged to those of the VC (by way of k_{inf}) since the only difference in the terms of Equation 4-12 is the number densities used in the AC term. When the red line is close to the blue line, the AC cross sections are near those of the VC since the only difference between Equations 4-10 and 4-12 is the cross sections set used in the AC term. The graph surrounded by a box signifies which set of inputs result in the lowest number density difference (blue line) at BOC 18.

Using these ideas, we can see the sensitivity of the adapted cross sections and number density errors to alpha and k_{eff} weight for a given set of cycle weights. For a k_{eff} weight of 100 or 1,000, decreasing alpha will improve the cross section agreement and reduce the number density error. As alpha is decreased, the increased freedom to adjust cross sections, indicated by the increasing RMS, allows the algorithm to better match the VC cross sections. Naturally, a better set of AC cross sections will improve the reduction in number density errors. This trend does not apply to k_{eff} weight of 10,000, however. Since too much emphasis is being placed on k_{eff} , the cross sections do not have very much freedom for adjustment. This is supported by noticing that the gap between the red and blue line is essentially constant for all three alpha values. This similarity in the cross sections for various alphas produces the same trend and magnitude in the number density error (blue line).

For a constant α , the behavior is more complicated. Looking at any column of Figure 4.3, we can see the three possible outcomes: 1) wrong cross sections and correct number densities for k_{eff} weight of 100, 2) correct cross sections and correct number densities for k_{eff} weight of 1,000, and 3) wrong cross sections and wrong number densities for k_{eff} weight of 10,000. Furthermore, comparing the boxed in graph in column one with the graph directly below it shows that it is possible for incorrect cross sections to reduce the number density error lower than a better set of AC cross sections. This behavior is possible because there is infinitely many combinations of cross sections that can match the VC responses. In order to pick the best set of inputs, Figure 4.4 shows how to avoid being fooled by this behavior.

Figure 4.4 shows the weighted components of the AC misfit term in Equation 4-1 normalized by the DC weighted misfit components. Note, however, that the misfit values plotted in Figure 4.4 were computed using the core simulator as opposed to the Jacobian. This represents the improvement in AC observables from their original DC values. The misfit term is a combination of the k_{eff} and nodal power misfits, written as

$$w_j^2 \left\| \Delta \bar{d}_j^m - \bar{\Theta} \Delta \bar{p} \right\|_{C_{d,j}^\dagger}^2 = w_j^2 \left[\left\| \Delta \bar{d}_j^m - \bar{\Theta} \Delta \bar{p} \right\|_{C_{k,j}^\dagger}^2 + \left\| \Delta \bar{d}_j^m - \bar{\Theta} \Delta \bar{p} \right\|_{C_{NP,j}^\dagger}^2 \right] \quad 4-15$$

where $C_{k,j}^\dagger$ is the k_{eff} covariance matrix, $C_{NP,j}^\dagger$ is the nodal powers covariance matrix, and $\bar{\Theta}$ is the core simulator. Equation 4-15 can be broken into these terms because the measurements associated with nodal powers and k_{eff} are uncorrelated observables (meaning $C_{d,j}^\dagger$ is block diagonal). If we define the components of the misfit term in Equation 4-15 for the design and adapted core as

$$\text{MF}_{k,j}^{\text{DC}} = \left\| \Delta \bar{d}_j^m \right\|_{C^\dagger_{k,j}}^2 \quad \text{since } \Delta \bar{p} = \bar{p}_{\text{DC}} - \bar{p}_{\text{DC}} \quad 4-16$$

$$\text{MF}_{NP,j}^{\text{DC}} = \left\| \Delta \bar{d}_j^m \right\|_{C^\dagger_{NP,j}}^2 \quad \text{since } \Delta \bar{p} = \bar{p}_{\text{DC}} - \bar{p}_{\text{DC}} \quad 4-17$$

$$\text{MF}_{k,j}^{\text{AC}} = \left\| \Delta \bar{d}_j^m - \bar{\Theta} \Delta \bar{p}_{\text{AC}} \right\|_{C^\dagger_{k,j}}^2 \quad 4-18$$

$$\text{MF}_{NP,j}^{\text{AC}} = \left\| \Delta \bar{d}_j^m - \bar{\Theta} \Delta \bar{p}_{\text{AC}} \right\|_{C^\dagger_{NP,j}}^2 \quad 4-19$$

where $\Delta \bar{p}_{\text{AC}} = \bar{p}_{\text{AC}} - \bar{p}_{\text{DC}}$, then the values shown in Figure 4.4 can be written as

$$\text{MF}_{NP,j} = \frac{\text{MF}_{NP,j}^{\text{AC}}}{\text{MF}_{NP,j}^{\text{DC}}} \quad 4-20$$

$$\text{MF}_{k,j} = \frac{\text{MF}_{k,j}^{\text{AC}}}{\text{MF}_{k,j}^{\text{DC}}} \quad 4-21$$

In Figure 4.4, we can see that it does not make sense to select any cases using a k_{eff} weight of 100. Even though the AC cross section sets were able to reduce the number density errors more than any other k_{eff} weight (in Figure 4.3 for a constant α), the cross sections are incapable of predicting k_{eff} . For the $\alpha = 5$ case, reducing the DC k_{eff} mismatch (blue line) to 40% is equivalent to being around ~700 pcm away from the VC. By cycle 17, reducing the mismatch to 20% is equivalent to being around ~500 pcm away from the VC. However, by increasing the k_{eff} weight to 1,000, you can get number density error reduction comparable to k_{eff} weight 100 and also reduce $\text{MF}_{k,j}$ to nearly zero. This appears to reveal a guideline for

picking a k_{eff} weight will result in correct cross sections and a substantial reduction in number density errors: select the lowest k_{eff} weight that also substantially reduces the k_{eff} mismatch.

Using Figure 4.4 to analyze the power misfit is difficult because of the noise component added to nodal powers. A small reduction in the nodal power misfit is misleading since adaption cannot predict observables any better than the applied noise. In Figure 4.4, the $\text{MF}_{k,j}$ values for a k_{eff} weight of 100 and $\alpha = 5$ successfully reduce the error in nodal powers to the noise. Knowing this, it is clear that all cases shown nearly predict the nodal powers down to the noise level for all adapted cycles. Adding number density errors does not corrupt the ability of the adaption algorithm to correctly predict the nodal powers. Predicting the nodal powers seems to be insensitive to both the k_{eff} weight and alpha.

All of this being said, Figure 4.5 shows that adjusting the cross sections cannot completely remove the number density errors. Figure 4.5 shows the $\langle \text{RMS} \rangle^{r,s}$ (red line) and $\langle \text{RMS} \rangle_{\text{AC,ND}}^{r,s}$ (blue line). These curves represent the AC errors normalized by the original DC error. It is clear that the rate of error reduction over each cycle reduces to almost zero before the error ever reaches zero. This means that introducing number density errors perturbs the core in ways the cross sections cannot correct for. This is actually beneficial, because if number density errors were indistinguishable from cross section errors, then adaption would incorrectly adjust cross sections for both sources of error. This unphysical adaption would not be robust once the number density errors have been annealed.

Figure 4.5 also shows the impact of including multiple cycles in the adaption/depletion sequence. Starting several cycles back allows burnup healing and fresh fuel loading to anneal

down the number density errors. This supports our method of starting the adaptation at the cycle where the oldest assembly in cycle n is fresh.

4.3.2 : Constant k_{eff} Weight

Due to the same behavior in the constant k_{eff} weight graphs, only the case k_{eff} weight=1,000 will be discussed here (Figures 4-6 through 4-8). The title of each graph is the alpha used and the titles of Figures 4-7 and 4-8 also include the resulting RMS cross section adjustment in standard deviations. The left hand side of each row tells the cycle weights $\{W_i\}$ used for all three graphs of that row. The number in the box at the bottom of the figure is the k_{eff} weight used for every graph in the figure.

The rows of Figure 4.6 show what was revealed in the previous subsection; decreasing alpha will result in better AC cross sections. A lower alpha allows more freedom to adjust the DC cross sections. Since decreasing alpha produces cross sections closer to the VC cross sections, it also further reduces the number density error.

Surprisingly, for a constant α , weighting the later cycles more than the earlier cycles does not really have an impact on the number density error. The only noticeable improvement is gained by increasing the weight of any cycle other than 15. The decrease in number density error obtained by increasing the weight of one or more cycles is essentially identical for all cases. This is true for all three values of alpha.

Albeit small, changing cycle weights does impact $MF_{k,j}$. The impact follows the same trend as before. Once any cycle weight other than 15 has been increased, the impact is identical for all combinations of cycle weights. Looking at Figure 4.7, increasing the weight of the higher cycles increases the magnitude of $MF_{NP,j}$ most notably for cycle 15. In fact, the

behavior seen in Figure 4.7 is probably responsible for the behavior in Figure 4.6. Increasing the cycle weight of the higher cycles may be resulting in a marginally smaller $MF_{k,16}$ and

$MF_{k,17}$, which will result in a marginally smaller $RMS_{AC}^{r,s}$ and $RMS_{AC,ND}^{r,s}$.

To bring it full circle, the marginal reductions in Figure 4.6 result in substantial reductions in Figure 4.8. The lower the alpha, the bigger the reduction. For $\alpha = 5$, increasing the cycle weights will earn about a one-third reduction in the cycle 18 number density error.

4.3.3 : Constant Alpha

Again, due to the similar behavior in the constant alpha graphs, only the case $\alpha = 10$ will be discussed here (Figures 4-9 through 4-11). The title of each graph is the k_{eff} weight used and the resulting RMS cross section adjustment in standard deviations. The left hand side of each row tells the cycle weights $\{W_i\}$ used for all three graphs of that row. The number in the box at the bottom of the figure is the α used for every graph in the figure.

Just as we learned from Figure 4.6, Figure 4.9 shows that varying the k_{eff} weight will control whether or not the AC cross sections correspond to the VC cross sections. Along with Figure 4.9, it is necessary to also look at the misfit terms in Figure 4.10 to determine which k_{eff} weight is also correctly matching the VC. Observing the number density error for a constant k_{eff} weight while changing the cycle weights follows the exact behavior when changing cycle weights in the previous section. The only improvement is gained when the cycle weights are compared to $\bar{W} = [1 \ 1 \ 1]$. The results are virtually identical when comparing any other set of the cycle weights. The same holds for Figure 4.11.

4.3.4 : Linearization Error

In Chapter 2, we introduced the comparison used to determine whether or not an iteration was required. This metric compares how well the k_{eff} and nodal power misfits predicted by the Jacobian match the k_{eff} and nodal power misfits predicted by the core simulator. If the misfits from the respective models are close to one another, then the cross sections have not been adjusted outside the first-order approximation of the Jacobian. Since this is the difference between the linear and nonlinear model, this is called the linearization error and was measured by comparing equations 4-20 and 4-21 to

$$\text{MF}_{NP,j}^{\text{LIN}} = \frac{\text{MF}_{NP,j}^{\text{AC,LIN}}}{\text{MF}_{NP,j}^{\text{DC}}} \quad 4-22$$

$$\text{MF}_{k,j}^{\text{LIN}} = \frac{\text{MF}_{k,j}^{\text{AC,LIN}}}{\text{MF}_{k,j}^{\text{DC}}} \quad 4-23$$

where

$$\text{MF}_{NP,j}^{\text{AC,LIN}} = \left\| \Delta \bar{d}_j^m - \bar{\bar{A}}_j \Delta \bar{p}_{\text{AC}} \right\|_{C_{NP,j}^\dagger}^2 \quad 4-24$$

$$\text{MF}_{k,j}^{\text{AC,LIN}} = \left\| \Delta \bar{d}_j^m - \bar{\bar{A}}_j \Delta \bar{p}_{\text{AC}} \right\|_{C_{k,j}^\dagger}^2 \quad 4-25$$

noting that the Jacobian is being used here instead of the core simulator. Since the linear misfits computed using Equations 4-22 and 4-23 are so close to those of 4-20 and 4-21, they will not be plotted. Their values would look identical to all graphs in Figures 4.4, 4.7, and 4.10 meaning the Jacobian ($\bar{\bar{A}}_j$) closely approximates the core simulator ($\bar{\bar{\Theta}}$) for the adjustments of all cases shown. Since the linearization error is negligible at the upper bound of adjust-

ments that we are comfortable with (~ 1 standard deviation), iterating would not provide any substantial gain.

4.4: Summary

In summary we can see that number density errors do not corrupt the fidelity of adaption. The figures which display the misfit values show that adaption is capable of adjusting the cross sections even in the presence of number density errors so that the AC observables approach the VC observables. Since it is possible to get the correct misfit values with the wrong cross sections, comparing the $\text{RMS}_{AC}^{r,s}$ to $\text{RMS}_{AC,ND}^{r,s}$ verified that the AC cross sections approached the VC cross sections with the proper k_{eff} weight. Furthermore, to ensure the resulting cross sections were within their experimental uncertainty, the standard deviation RMS of cross section adjustments was kept near 1.0.

In addition to the sustained fidelity, number density errors do not corrupt the robustness of adaption. This can be seen by using the adapted cross sections to predict cycle 18. Since cycle 18 was not adapted, this will ensure the cross sections can correctly predict core observables under conditions not included in the adaption. The first graph in Figure 4.12 shows k_{eff} of the DC, VC, and AC for cycle 18. The second graph shows the DC-VC and AC-VC pcm error. Figure 4.13 shows the core wide nodal power RMS for each time step of cycle 18. These values were computed as

$$\text{RMS}_{DC, NP}^r = \sqrt{\frac{1}{N \cdot Z} \sum_{j=1}^N \sum_{i=1}^Z (d(\text{VC}_{i,j,r}) - d(\text{DC}_{i,j,r}))^2} \quad 4-26$$

$$\text{RMS}_{AC, NP}^r = \sqrt{\frac{1}{N \cdot Z} \sum_{j=1}^N \sum_{i=1}^Z (d(\text{VC}_{i,j,r}) - d(\text{AC}_{i,j,r}))^2} \quad 4-27$$

$$\text{RMS}_{VC^*, NP}^r = \sqrt{\frac{1}{N \cdot Z} \sum_{j=1}^N \sum_{i=1}^Z (d(\text{VC}_{i,j,r}^*) - d(\text{AC}_{i,j,r}))^2} \quad 4-28$$

where $d(k_{i,j,r})$ represents the nodal power of node i , of assembly j , at time step r of core k .

Here VC^* is the virtual core observables with no noise added. The notation of the other cores follows as previously defined, where DC is the design core, AC is the adapted core, and VC is the virtual core. Figures 4.12 and 4.13 were created using $\bar{W} = [1 \ 1 \ 1]$, a k_{eff} weight of 1,000, and $\alpha = 10$.

Lastly, the graphs that show $\text{RMS}_{AC}^{r,s}$, $\text{RMS}_{AC,ND}^{r,s}$, and their normalized counterparts demonstrate that it is not completely necessary to return to cycle 1 to remove number density errors. Starting the adaption/depletion sequence several cycles before the cycle of interest is all that is needed to reduce the number density error as low as the adapted cross section set will allow. Fortunately, the reduction in number density error is generally greater than 50% and can be up to 75% for a k_{eff} weight of 1,000 (Figure 4.8).

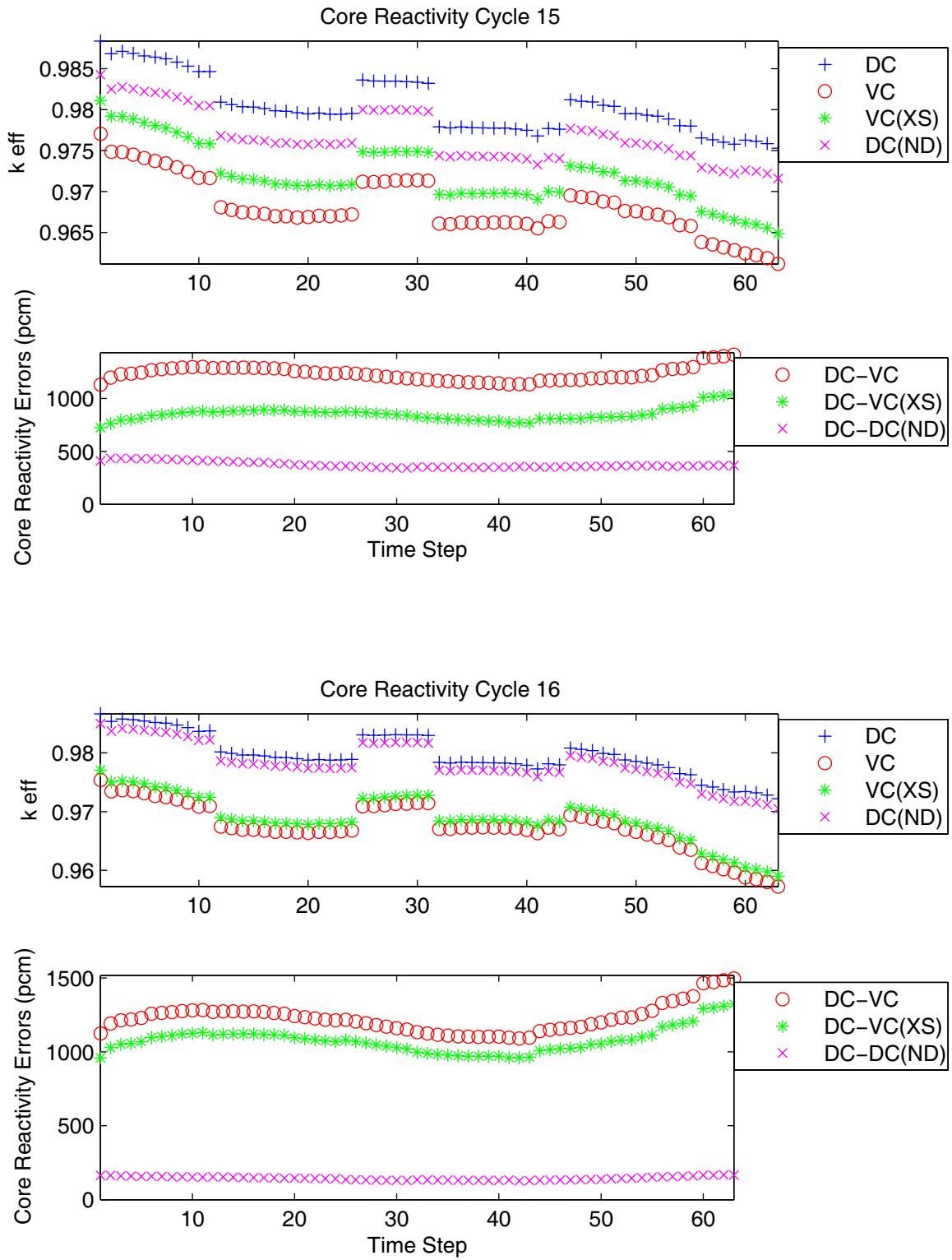


Figure 4.1. Number density and cross section errors vs. depletion

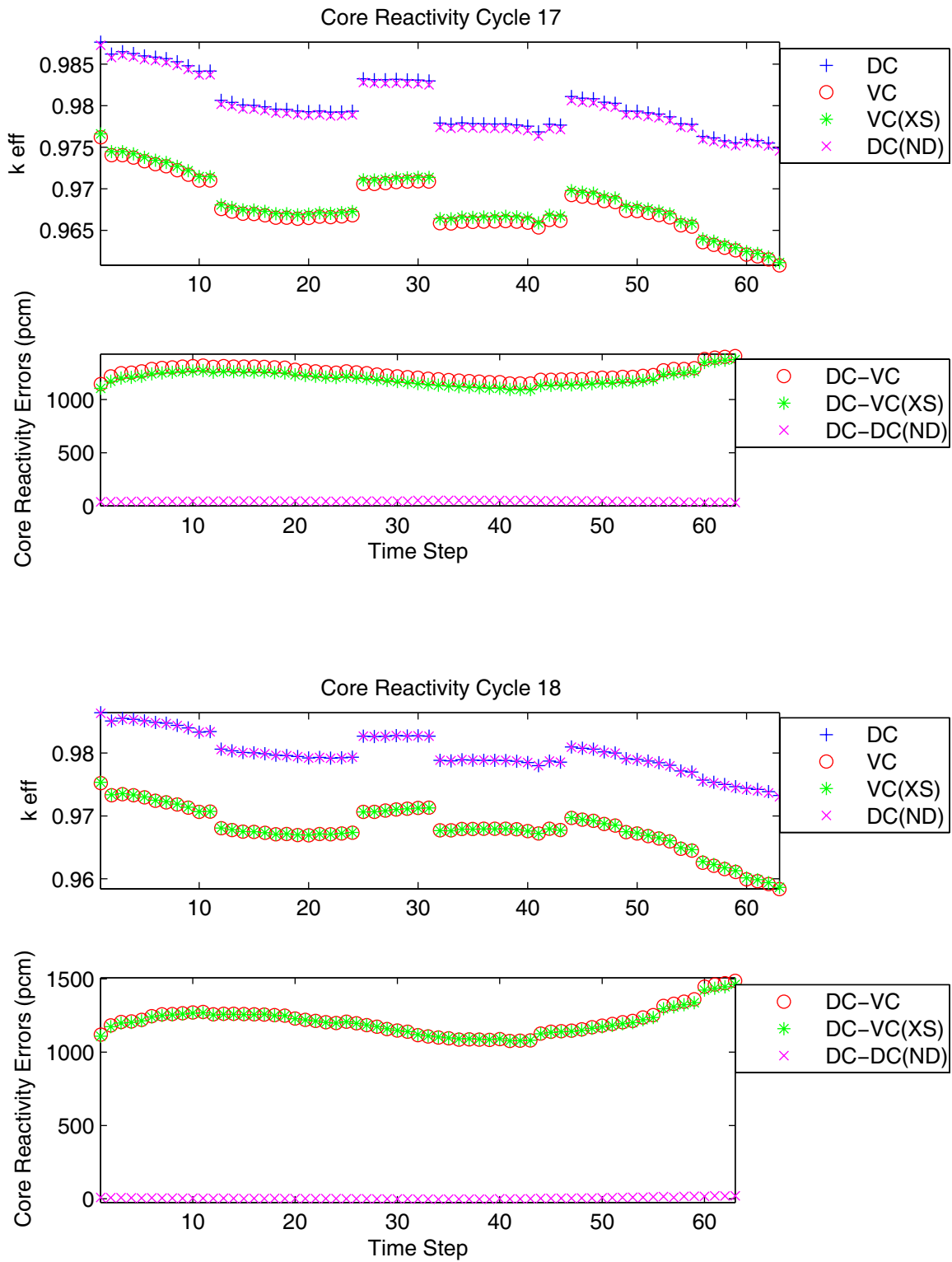


Figure 4.2. Number density and cross section errors vs. depletion

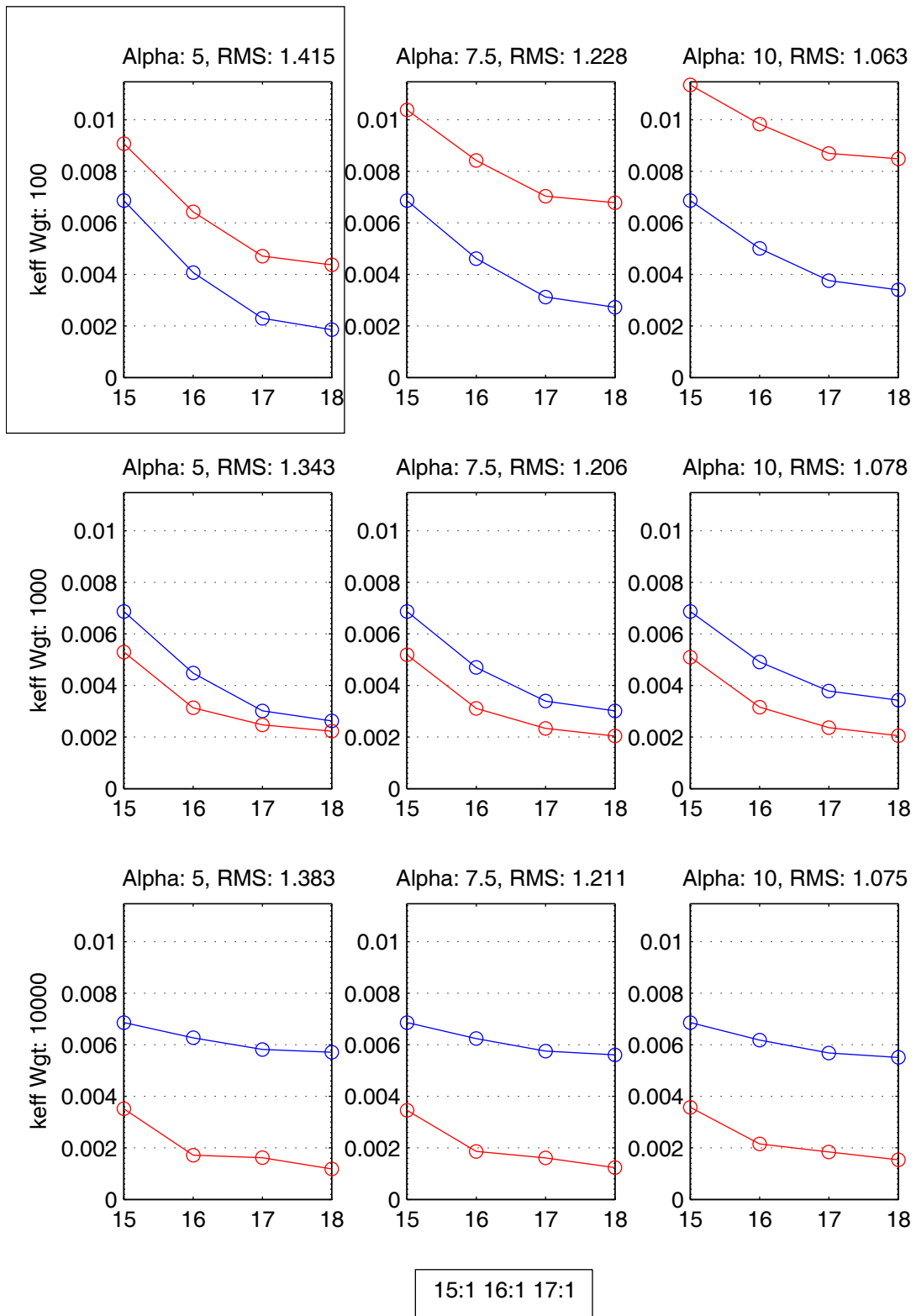


Figure 4.3. $RMS_{AC}^{r,s}$ (red) and $RMS_{AC,ND}^{r,s}$ (blue) reduction for constant cycle weight

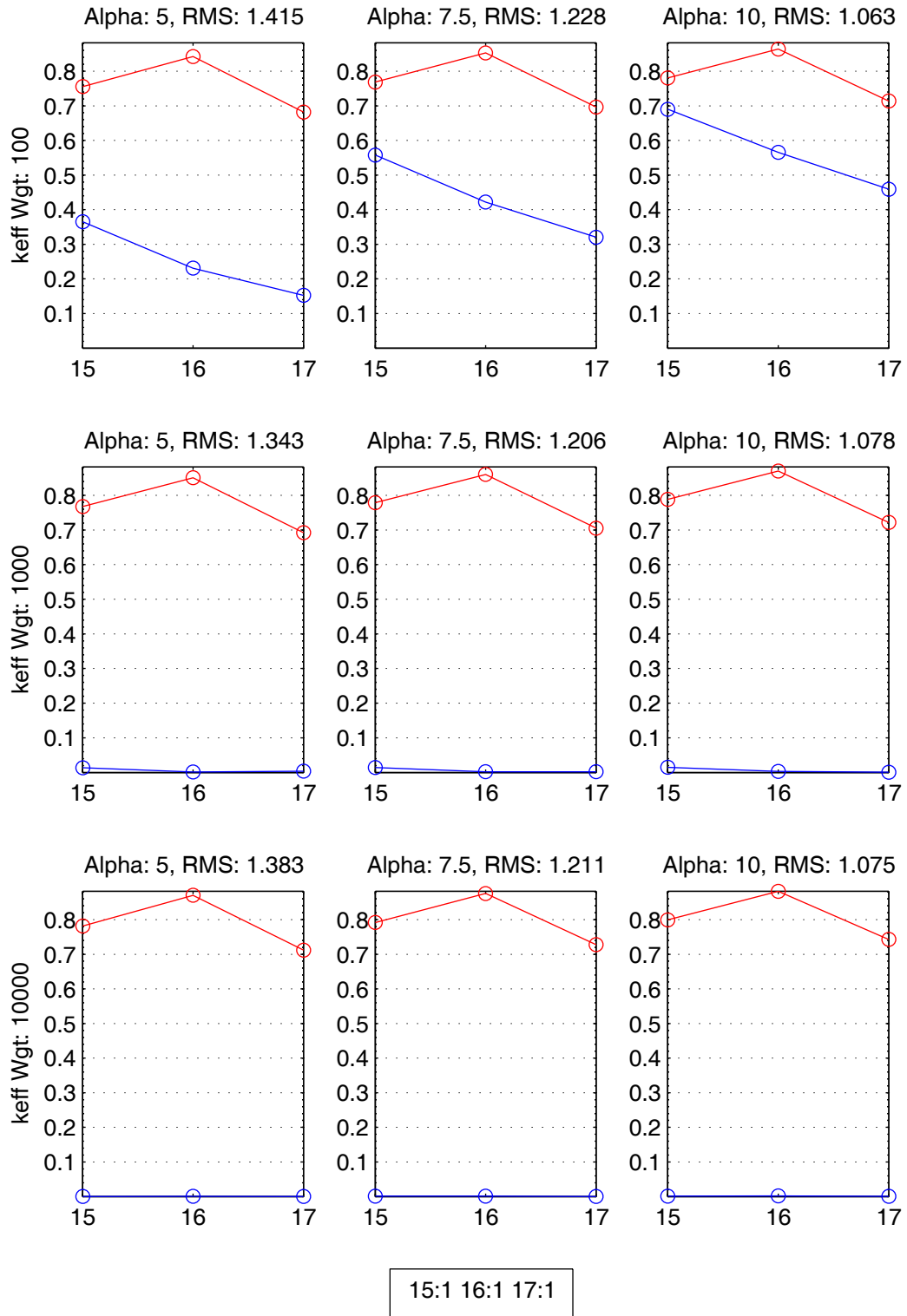


Figure 4.4. k_{eff} (blue) and nodal power (red) misfit for constant cycle weight

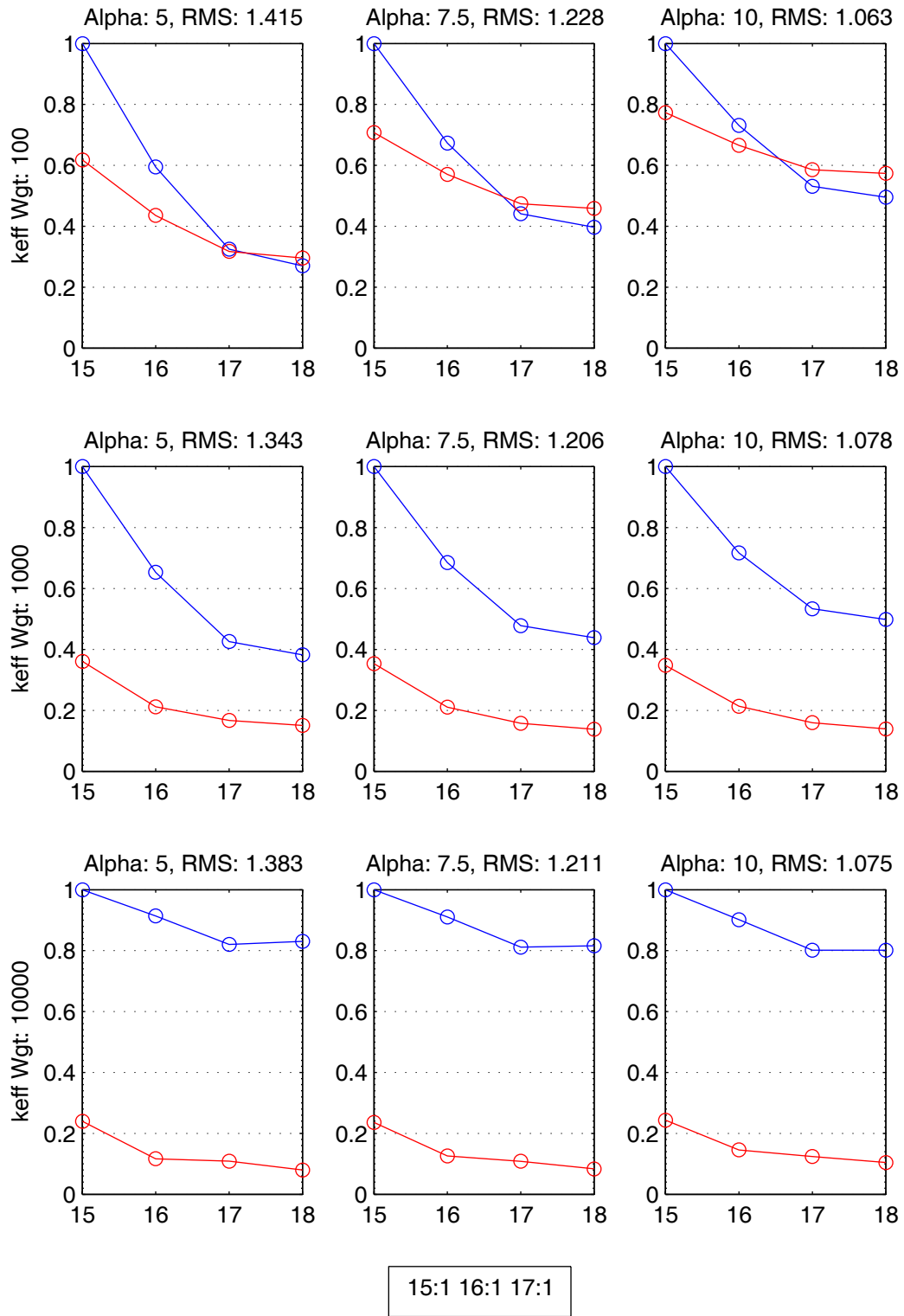


Figure 4.5. $\langle \text{RMS} \rangle_{AC}^{r,s}$ (red) and $\langle \text{RMS} \rangle_{AC,ND}^{r,s}$ (blue) reduction for constant cycle weight

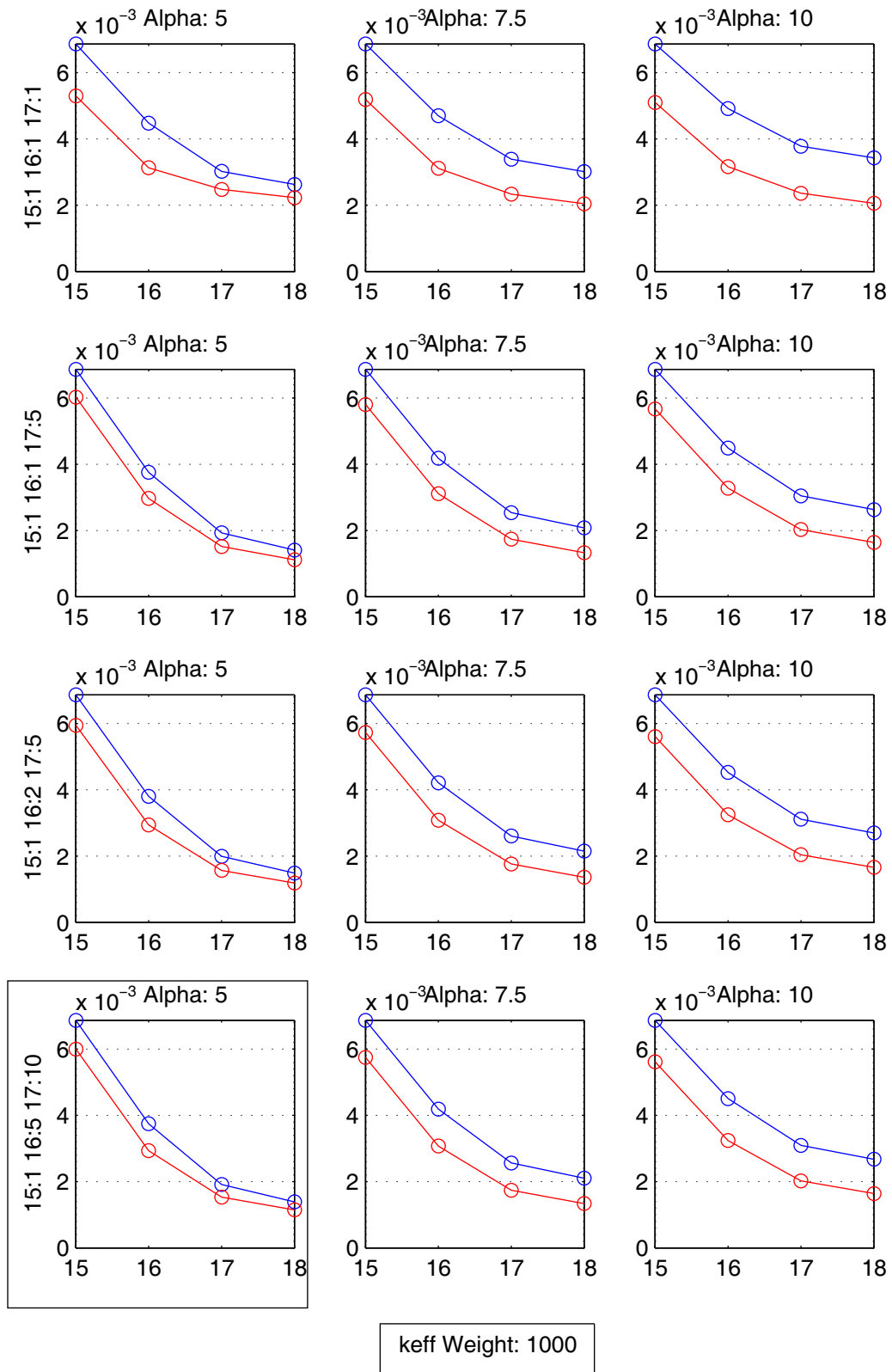


Figure 4.6. $RMS_{AC}^{r,s}$ (red) and $RMS_{AC,ND}^{r,s}$ (blue) reduction for constant k_{eff} weight.

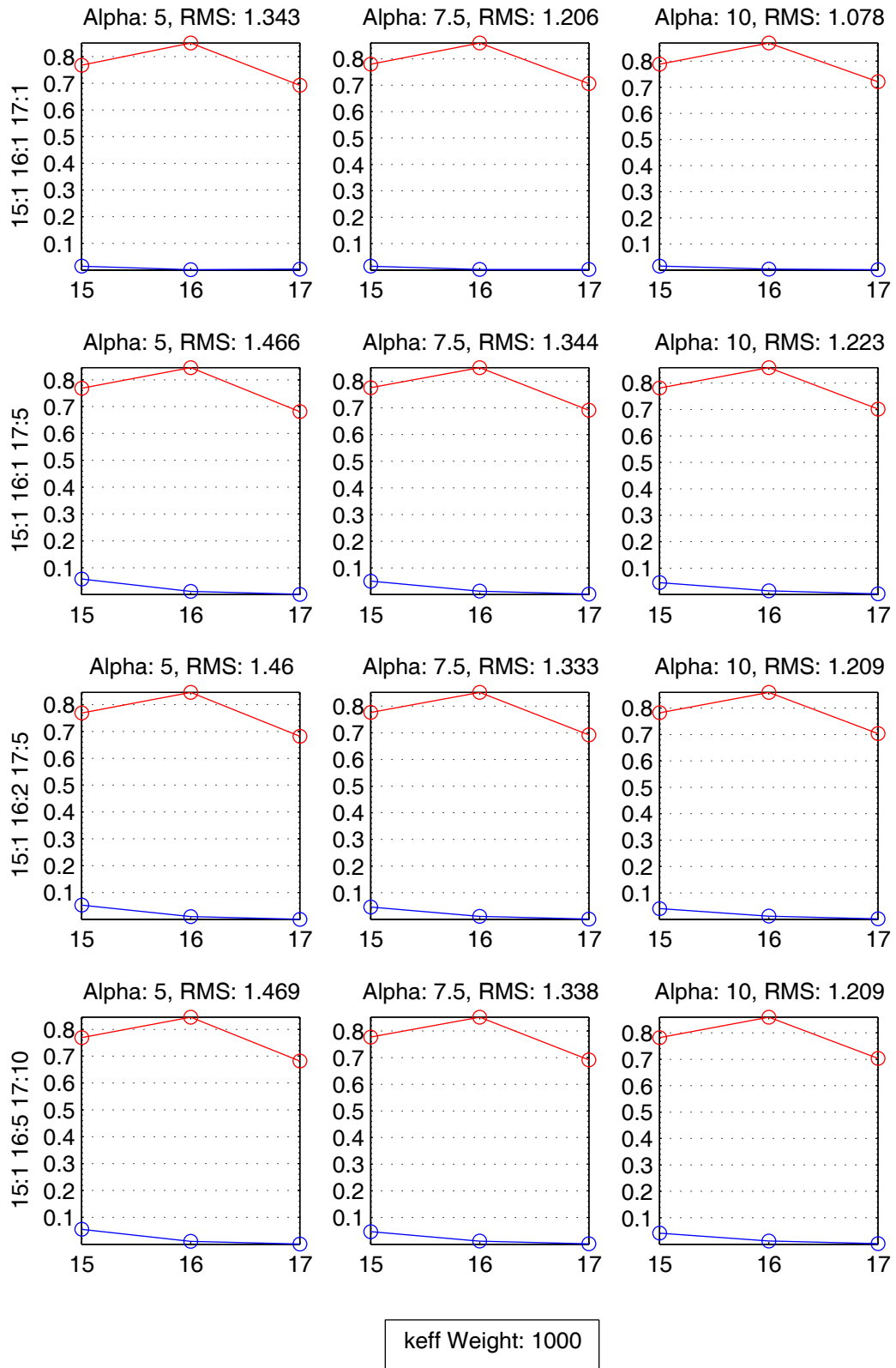


Figure 4.7. k_{eff} (blue) and nodal power (red) misfit for constant k_{eff} weight.

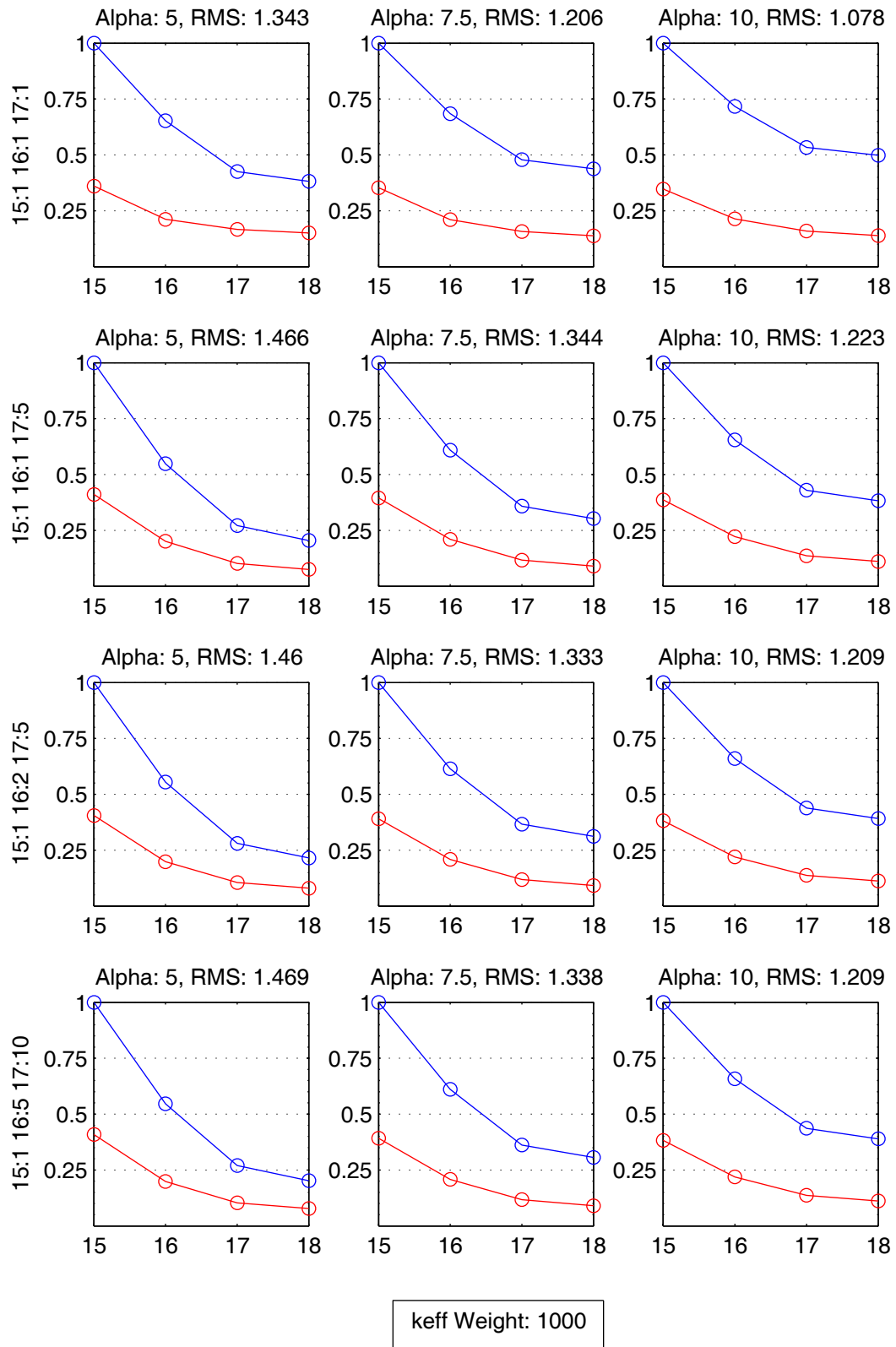


Figure 4.8. $\langle \text{RMS} \rangle_{AC}^{r,s}$ (red) and $\langle \text{RMS} \rangle_{AC,ND}^{r,s}$ (blue) reduction for constant k_{eff} weight

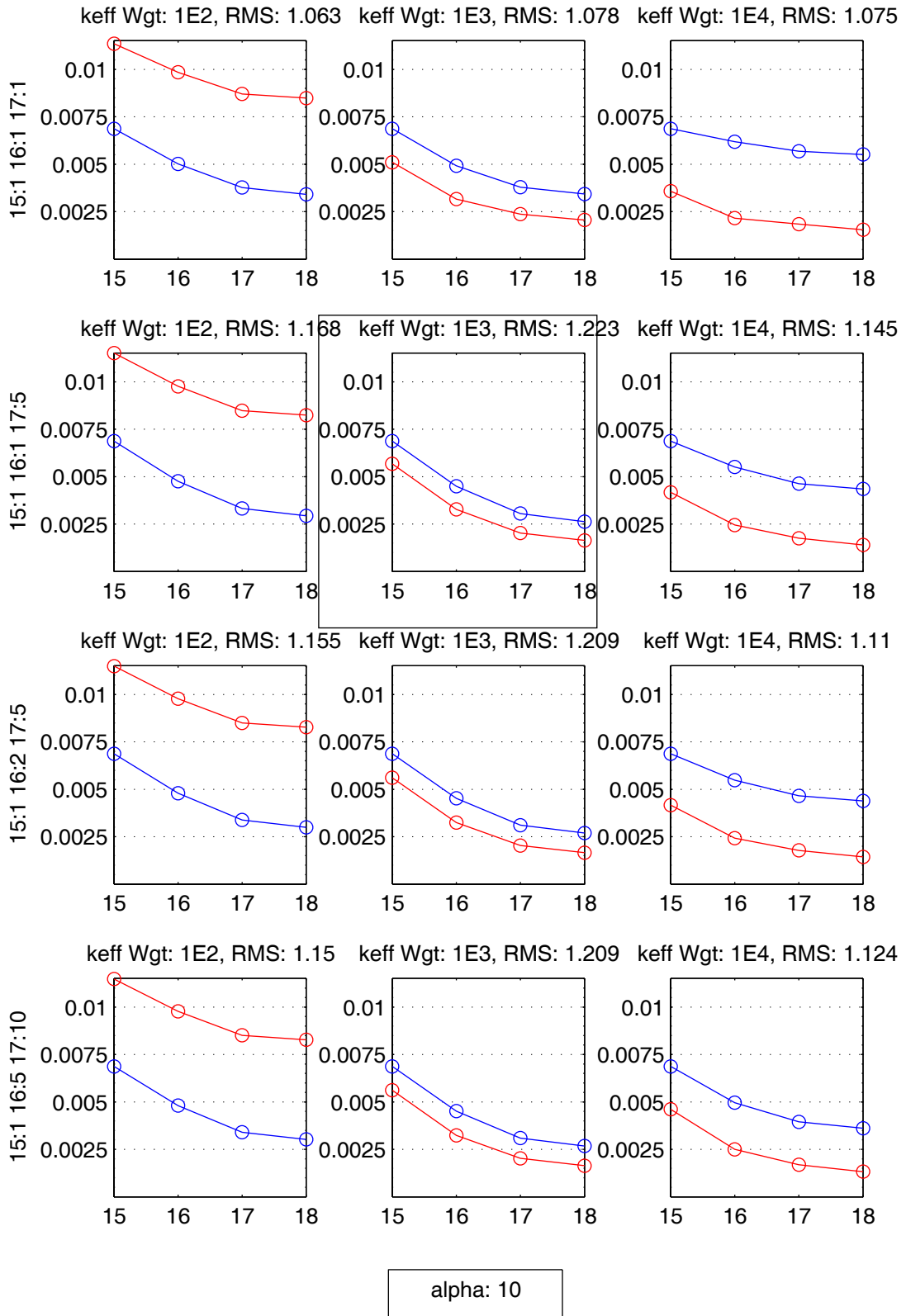


Figure 4.9. $RMS_{AC}^{r,s}$ (red) and $RMS_{AC,ND}^{r,s}$ (blue) reduction for constant alpha.

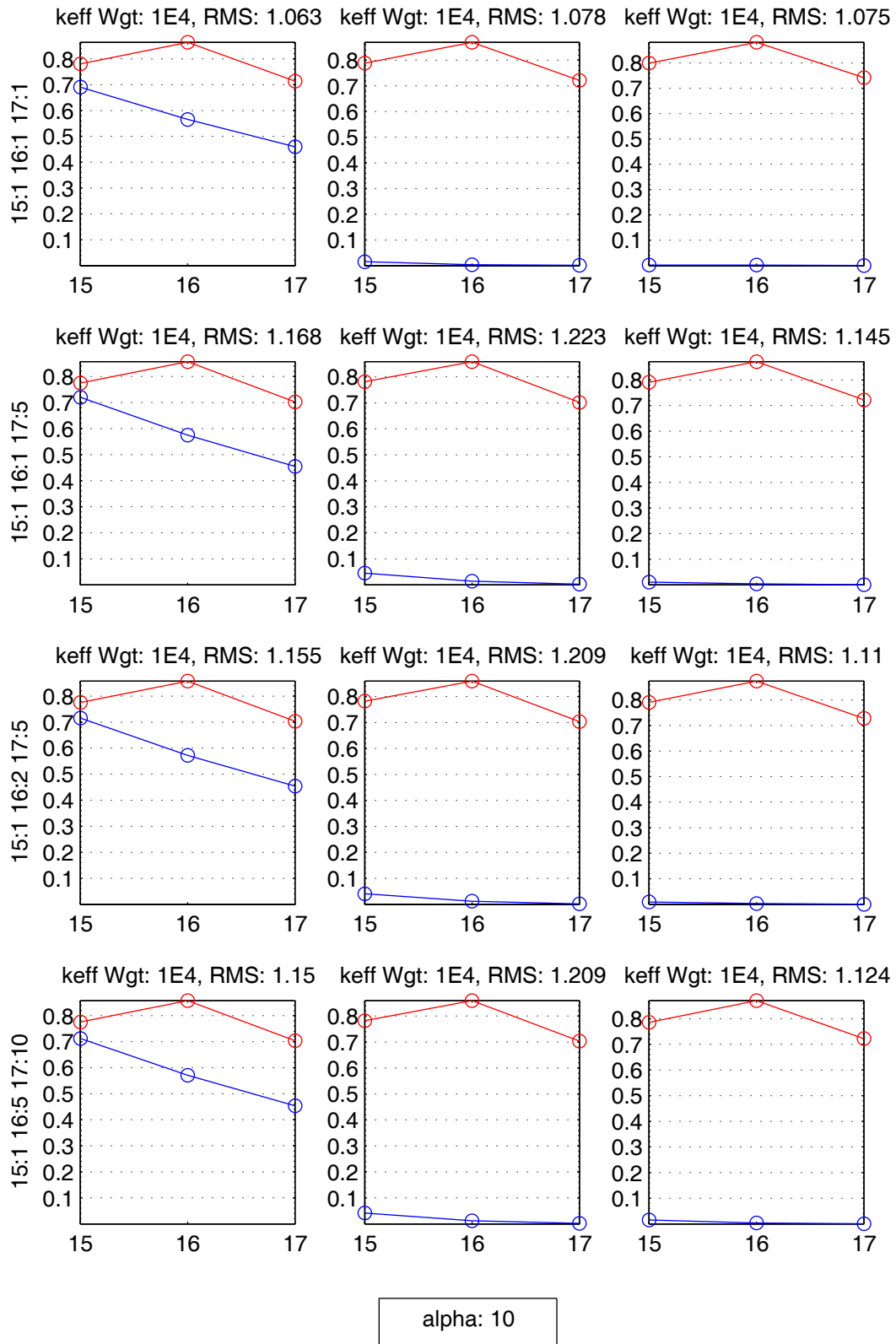


Figure 4.10. k_{eff} (blue) and nodal power (red) misfit for constant alpha.

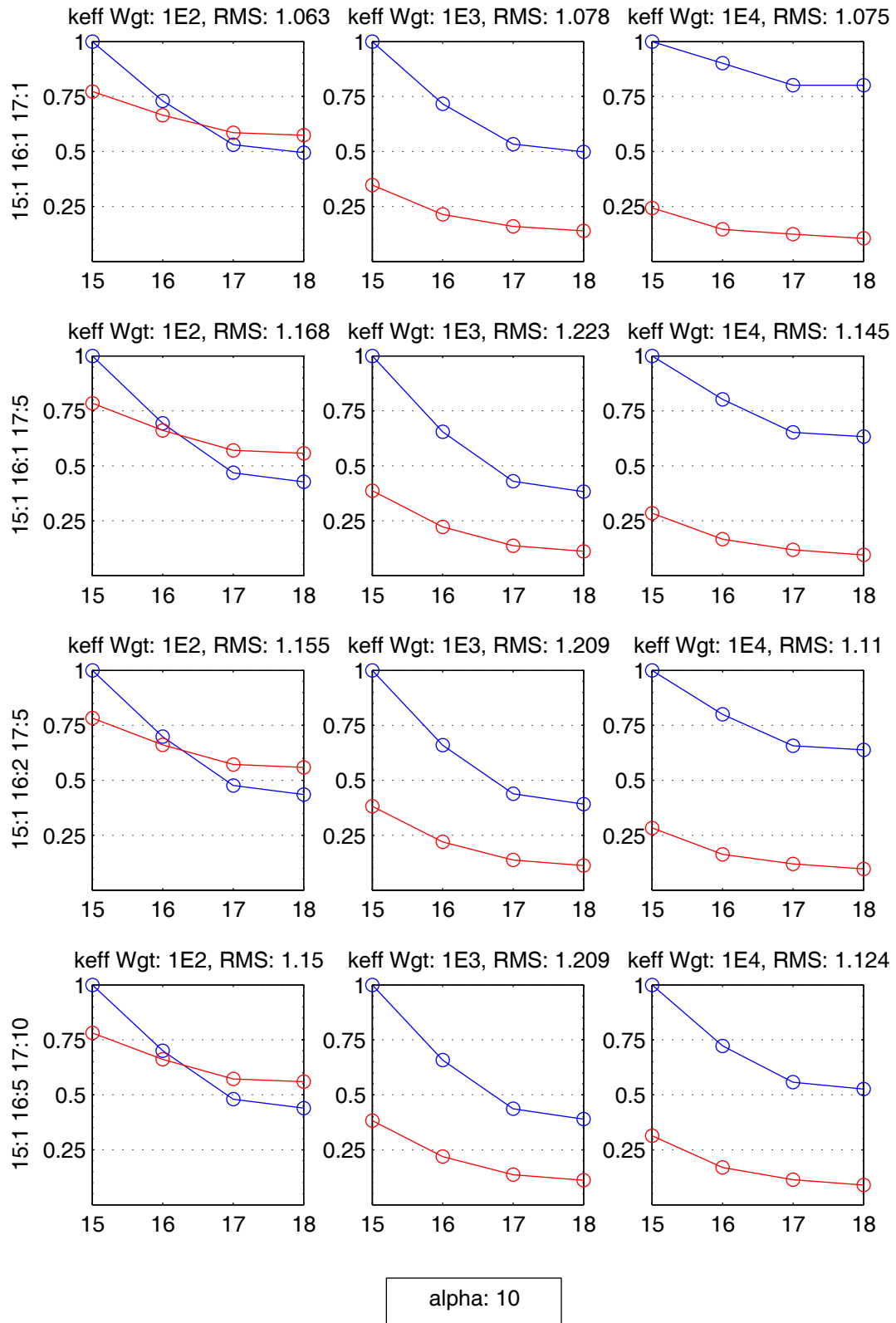


Figure 4.11. Number density reduction for constant alpha

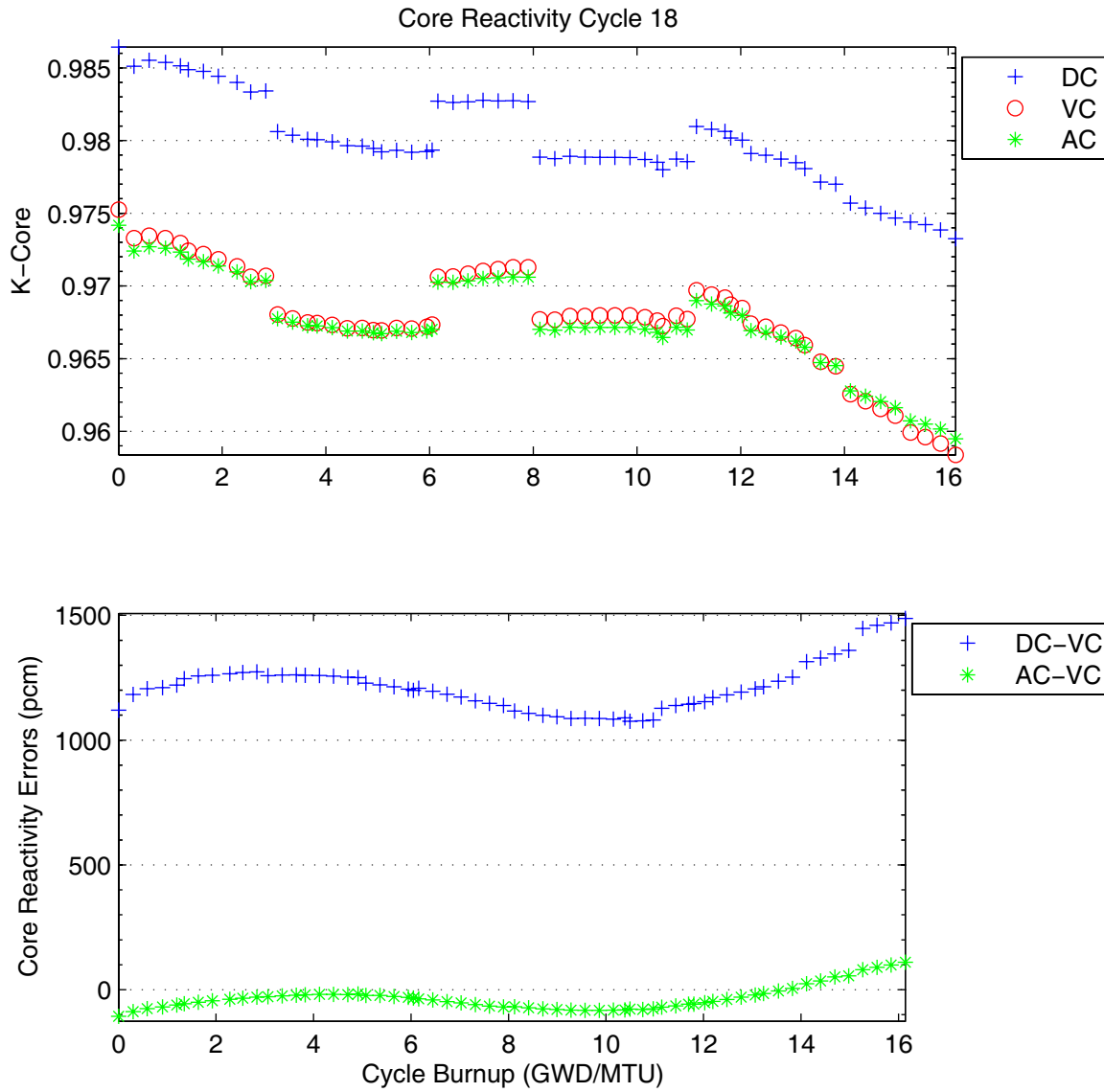


Figure 4.12. Cycle 18 k_{eff} and pcm error for each core

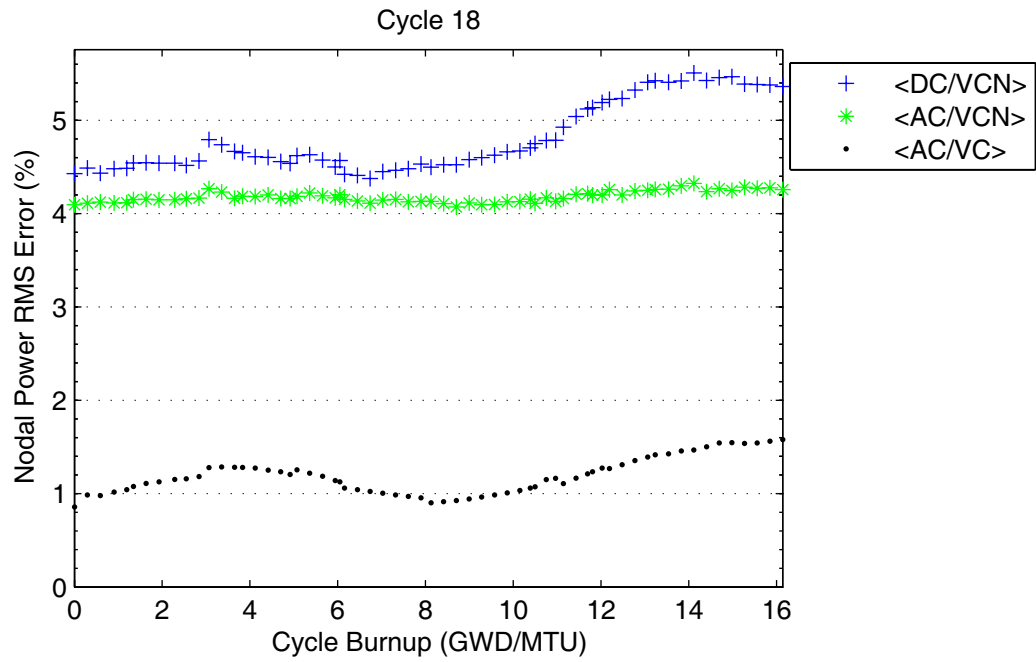


Figure 4.13. Nodal power RMS error: $RMS_{DC, NP}^r$ (blue), $RMS_{AC, NP}^r$ (green), and $RMS_{VC^*, NP}^r$ (black)

Chapter 5: Conclusions

To summarize the findings, multicycle adaptive simulation is capable of partly removing the burnt fuel number density errors induced by adjusting cross sections. There are two sources of number density errors:

- 1) Errors that arise from the simulator's inability to correctly predict the isotopics of the actual plant fuel assemblies. The number densities input to any core simulator will never be identical to those of the actual fuel in the real world core if the simulator's cross sections are wrong. These errors in number densities influence the adaption since the algorithm is using cross section adjustments to account for discrepancies caused by differences in isotopics.
- 2) Errors that arise because the burnt fuel isotopics of any assembly are inconsistent once that assemblies' cross sections are modified via adaption. This is because number densities are dependent on the cross section set used to deplete. Once the set changes at cycle n , it should have also changed in the previous cycles' depletions. The induced error exists because re-depleting with the adapted cross sections will result in different number densities than the number densities corresponding to the original cross section set.

To remove the first source of error and better predict any future cycle n , thus enhancing robustness, the induced number density errors were reduced by performing a multicycle adaptation. This was accomplished by simultaneously adapting cycles m through $n-1$, where cycle m is the cycle in which the oldest assembly in cycle n was fresh.

To update inconsistent set of number densities with the adapted cross sections, the adapted cross sections were used to deplete from cycle m to n . By the time the depletion sequence reaches cycle 18, the number densities have been corrected to match the set of adapted cross sections. This happens through burnup healing and fresh fuel loading.

If adaption were able to exactly predict the VC cross sections, both sources of error would be removed simultaneously during the depletion using the adapted cross sections. Updating the number densities to be consistent with the adapted cross sections would provide the exact number densities of the real fuel. However, with the current capabilities of adaption, this can never happen because the presence of instrument noise, the nonlinear components of the core simulator, and the number density errors that exist in burnt fuel at BOC where adaption begins prevent adaption from ever obtaining the exact cross sections.

In conclusion, although the exact cross sections are not obtained, the algorithm does adjust the cross sections in a fashion that reduces both the discrepancy between the measure and predicted observables, and reduces the error in number density. It was found that using multiple cycles further improves the reduction in number density errors. This is crucial to completing the necessary traits of an adapted core simulator: fidelity, robustness, and short run times. The fidelity was confirmed by using the adapted cross sections to nearly reproduce the observables of the virtual core. Robustness was justified by using the adapted cross sections to predict cycle 18, a cycle that was not included in the adaption. Lastly, it was shown that only several cycles were required to reduce the errors in number densities as low as possible. This means that starting the adaption at cycle 1 of a core, where there is only fresh fuel and hence no number density errors, is unnecessary. To ensure the cross sections were not overadjusted because of the number density errors, a regularization parameter was selected to constrain the adjustments to stay near one standard deviation.

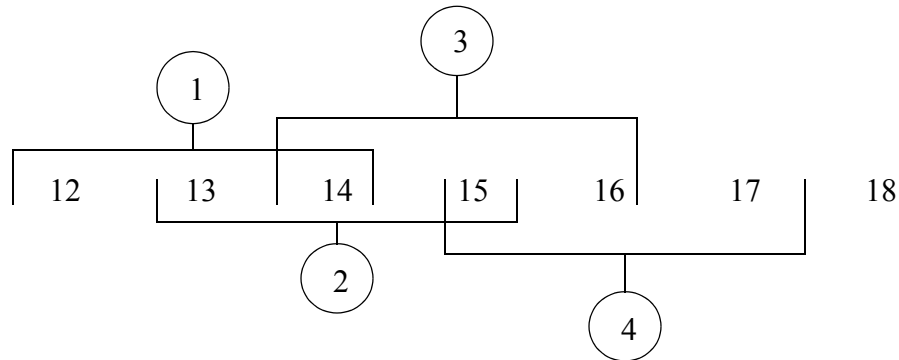
Chapter 6: Future Work

Although this research provided further insight into the capabilities of adaptive simulation, there were several assumptions and simplifications made that should be removed to further understand the limits of the algorithm. The most obvious simplification to remove is the method of perturbing a simulator to produce the virtual core observables. The true test of adaption will come when actual plant data is used to adjust the cross sections used in the simulator. Furthermore, we used k_{eff} and nodal powers as our set of observables. In reality, nodal powers are not observables. Before actual plant data are utilized, one should repeat the work reported upon here using k_{eff} , linear power range monitors (LPRM), and traveling in core probes (TIP) as core observables of the VC. Lastly, the background cross sections were ignored in this work due to the nonphysical nature of their adjustment. The simple solution to correct this problem is to modify the simulator to track the isotopes contained within the background cross section.

Also, over the course of this work, there have been several new ideas of how to remove the number density error component. The first is to correlate a change in number densities with a change in cross sections. With this information, one could use the difference in observables to adjust both the cross sections and number densities of burnt fuel input to the core simulator in the first cycle adaption is being completed on.

The other idea is to remove the number density errors is called ‘sliding adaption.’ To perform sliding adaption, one would start the adaption at a cycle before m , using the notation

consistent with the preceding chapters. This adaption would start at an arbitrary cycle k , such that $k < m$. If we take $k=12$, then sliding adaption would be executed as follows



where each bracket in the above figure represents a multicycle adaption. The number of cycles included in each bracket abides by the same rule used in this work: the oldest assembly in the first cycle outside the bracket was a fresh assembly in the first cycle of the adaption. This sequence of adaptations will use the adapted cross sections to continuously update the beginning-of-cycle (BOC) number densities in each subsequent adaption. This means that the BOC 13 number densities computed in the first adaption would become the BOC 13 number densities used in the second adaption, and so on. By the time adaption step 4 is reached, we expect that the BOC number densities would be much closer to the VC number densities than if the fourth adaption was the only adaption (as in this research).

Finally, this work used only two fuel bundle designs in all the cycles examined. Real power plants have multiple bundle designs, introducing new designs nearly every reload cycle. This situation is believed to present a more demanding test for adaptive simulation, so definitely should be studied as done in this work.

References

- [1] Hany Samy Abdel-Khalik, "Inverse Methods Applied to Adaptive Core Simulation," Master Thesis, North Carolina State University, Department of Nuclear Engineering, Raleigh, NC (2002).
- [2] Hany Samy Abdel-Khalik, "Adaptive Core Simulation," PhD Thesis, North Carolina State University, Department of Nuclear Engineering, Raleigh, NC (2004).
- [3] Hany Samy Abdel-Khalik and P. J. Turinsky, "Adaptive Core Simulation Employing Discrete Inverse Theory - Part I: Theory," *Nuclear Technology*, (2005).
- [4] Hany Samy Abdel-Khalik and P. J. Turinsky, "Inverse Method Applied to Adaptive Core Simulation," *Transactions of American Nuclear Society*, 86, 366, (2002).
- [5] Matthew A. Jessee, Hany S. Abdel Khalik and Paul J. Turinsky, "Evaluation of BWR Core Attributes Uncertainties Due to Multi-Group Cross-Section Uncertainties," *Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications (M&C+SNA 2007)*, Monterey, California, April 15-19, 2007, American Nuclear Society, CD-ROM.
- [6] Brian Randolph Moore, "Higher Order Generalized Perturbation Theory for BWR In-Core Nuclear Fuel Management Optimization," PhD Thesis, North Carolina State University, Department of Nuclear Engineering, Raleigh, NC, (1996).
- [7] A. N. Tikhonov, "Numerical Methods for the Solution of Ill-Posed Problems," Kluwer Academic Publishers, 1995.
- [8] Heinz W. Engl, Martin Hanke, and Andreas Neubauer, "Regularization of Inverse Problems," Kluwer Academic Publishers, 1996.