University of Tennessee, Knoxville

TRACE: Tennessee Research and Creative Exchange

Doctoral Dissertations

Graduate School

12-2013

Monte Carlo and Depletion Reactor Analysis for High-Performance Computing Applications

Brenden Thomas Mervin

University of Tennessee - Knoxville, bmervin@vols.utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Brenden Thomas Mervin entitled "Monte Carlo and Depletion Reactor Analysis for High-Performance Computing Applications." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Nuclear Engineering.

G. Ivan Maldonado, Major Professor

We have read this dissertation and recommend its acceptance:

Ron E. Pevey, Steve E. Skutnik, Brandon C. Prinns

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

12-2013

Monte Carlo and Depletion Reactor Analysis for High-Performance Computing Applications

Brenden Thomas Mervin

bmervin@utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Brenden Thomas Mervin entitled "Monte Carlo and Depletion Reactor Analysis for High-Performance Computing Applications." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Nuclear Engineering.

G. Ivan Maldonado, Major Professor

We have read this dissertation and recommend its acceptance:

Ron E. Pevey, Steve E. Skutnik, Brandon C. Prinns

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Monte Carlo and Depletion Reactor Analysis for High-Performance Computing Applications

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Brenden Thomas Mervin

December 2013

Acknowledgements

I wish to express my sincere gratitude to the many people who have helped throughout my doctoral studies, as they have been a precious and invaluable guide.

First and foremost I would like to thank my advisor Dr. G. Ivan Maldonado, who's constant support has guided me throughout my graduate studies. I would also like to thank the members of my dissertation committee: Dr. Ron Pevey, Dr. Steve Skutnik, and Dr. Brandon Prins for their support and assistance. I also need to thank my first mentor at Oak Ridge National Laboratory, Dr. A. Lou Qualls, for believing in me and enlightening me on the details of working and being successful within the professional community.

My sincere thanks go out to Dr. John C. Wagner, for creating an active role for me at Oak Ridge National Laboratory, Dr. Scott W. Mosher, for his direction, advice, and always making himself available to answer my questions, Dr. Matt A. Jessee and Dr. William A. Wieselquist, who helped me with some of the more intricate nuances of my research, Dr. Seth R. Johnson, for his patience and guidance enabling me to become a constructive and stable member of a development team, Dr. Thomas M. Evans, for including me in the ongoing development efforts of Exnihilo, Dr. Gregory G. Davidson, for providing guidance and facilitating my needs in the final stages of my research, and last but not least, Shane Hart, whom had I been without I would likely still be trying to change directories in a Linux terminal. I would also like to thank Dr. Bradley T. Rearden, Dr. Robert. E. Grove, Dr. Kevin T. Clarno, Dr. Cihangir Celik, Dr. Germina Ilas, Dr. Chris M. Perfetti, Dr. James Banfield, Dr.

Tara M. Pandya, Aaron Bevill, Jordan A. Lefebvre, Robert A. Lefebvre, and the rest of the Reactor and Nuclear Systems Division for their guidance and support.

Special thanks go out to the support provided by the University of Tennessees Science Alliances Joint Directed Research and Development Fund and Oak Ridge National Laboratory. I would also like to extend my deepest gratitude to the Nuclear Nonproliferation International Safeguards Graduate Fellowship Program for their support.

Most importantly, I would like to thank my family for their continuous encouragement and support, as well as an extra special “thank you” to my fiancée, Holly Spence, who could almost certainly go the rest of her life without seeing me behind another Linux terminal again.

Abstract

This dissertation discusses the research and development for a coupled neutron transport/isotopic depletion capability for use in high-preformance computing applications. Accurate neutronics modeling and simulation for "real" reactor problems has been a long sought after goal in the computational community. A complementary "stretch goal" to this is the ability to perform full-core depletion analysis and spent fuel isotopic characterization. This dissertation thus presents the research and development of a coupled Monte Carlo transport/isotopic depletion implementation with the Exnihilo framework geared for high-performance computing architectures to enable neutronics analysis for full-core reactor problems.

An in-depth case study of the current state of Monte Carlo neutron transport with respect to source sampling, source convergence, uncertainty underprediction and biases associated with localized tallies in Monte Carlo eigenvalue calculations was performed using MCNPand KENO. This analysis is utilized in the design and development of the statistical algorithms for Exnihilo's Monte Carlo framework, Shift. To this end, a methodology has been developed in order to perform tally statistics in domain decomposed environments. This methodology has been shown to produce accurate tally uncertainty estimates in domain-decomposed environments without a significant increase in the memory requirements, processor-to-processor communications, or computational biases.

With the addition of parallel, domain-decomposed tally uncertainty estimation processes, a depletion package was developed for the Exnihilo code suite to utilize the depletion capabilities of the Oak Ridge Isotope GENeration code. This interface was designed to be transport agnostic, meaning that it can be used by any of the reactor analysis packages within Exnihilo such as Denovo or Shift. Extensive validation and testing of the ORIGEN interface and coupling with the Shift Monte Carlo transport code is performed within this dissertation, and results are presented for the calculated eigenvalues, material powers, and nuclide concentrations for the depleted materials. These results are then compared to ORIGEN and TRITON depletion calculations, and analysis shows that the Exnihilo transport-depletion capability is in good agreement with these codes.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Chapter 1 provides a brief background of the research documented in this dissertation. The following sections describe the organization of this document, the overall objectives of this research, and the motivation behind the research presented within this manuscript.

1.1 Organization of Dissertation

The first chapter, Chapter 1, is an introductory chapter which describes the objectives and motivation for this research. Chapter 2 is devoted to introducing the reader to the subjects relevant to the research described herein. This includes presenting a brief background of radiation transport theory with a specific emphasis on Monte Carlo methods, the computational issues associated with full-core reactor analysis, and a brief overview of isotopic depletion analysis.

Chapter 3, presents the research methodologies used in this research in order to reach the goals set forth for this dissertation. A thorough description of methods regarding uncertainty analysis as it applies to localized tally estimation in Monte Carlo eigenvalue problems is described. This is followed by a discussion of Monte Carlo parallel decomposition strategies and the implications of the various strategies on the mechanics for estimating tally statistics. Description of the computational code and tools used throughout this dissertation are presented in Chapter 4 and description of the various benchmark models are available in Appendix A.

Chapter 5 presents the detailed analysis of Monte Carlo eigenvalue calculations and the issues that lead to biases and underprediction in the uncertainty for localized tallies. This is followed by Chapter 6 which presents the research and development of algorithms for calculating tally statistics in domain-decomposed Monte Carlo calculations. This includes results using Shift and a comparison of how the previously presented issues are affected by the mechanics within Shift.

Chapter 7 discusses the development and integration of depletion capabilities into the Exnihilo framework. Finally, the culmination of the research and development efforts presented in Chapters 5 through 7 provide a verification and validation demonstration of the developed capabilities in Chapter 8 by using the Shift Monte Carlo transport code coupled with ORIGEN depletion capabilities.

This document is concluded, in Chapter 9, with an assessment of the progress towards the initial goals set out by this research and the path moving forward as full-core reactor analysis of "real" reactors moves from the older simplified approaches towards the more modern and rigorous approaches as the one illustrated in this dissertation.

1.2 Objective

The overall goal of this research is the enablement of Monte Carlo-based neutronics for "real" reactor analysis, whereby the term "real" refers to what a core designer would require in the industry to accomplish the design, licensing, and operational management of a new core or fuel cycle.

The primary issues which prevent modern reactor analysis tools from performing such analyses are:

1. the prohibitive computation time associated with the level of detail of a real reactor model,

2. the prohibitive memory requirements associated with the number of tallies required to perform full-core analyses,

3. the ability to generate results with high enough precision and accuracy for nuclear reactor design, safety, and operational applications,

4. the ability to efficiently utilize massively parallel systems. Figure 1.1 illustrates an important leap forward with deterministic transport calculations toward a similar goal we present here for stochastic methods.

There are various hybrid methods, source acceleration strategies, and variance reduction methods which enhance Monte Carlo calculations in both efficiency and effectiveness; however the underlying structural design of Monte Carlo codes needs to be reworked in order to fully take advantage of massively parallel architectures. Although modern Monte Carlo codes such as MCNP [2] and KENO [3] are already parallel, their parallel strategy involves replicating the problem geometry on all processors, dividing the number of histories up between the processors, and synchronizing after every cycle in a criticality calculation. However, when the number of available processors is near the order of the number of histories per cycle, much efficiency will be lost by the need to frequently communicate information between the processors. Furthermore, for larger models with hundreds of thousands of material regions, the memory required to store all of this information on each processor far exceed the available memory per core for most Linux clusters. Therefore massively parallel systems cannot be taken advantage of properly using only domain replication strategies.

Domain-decomposition is a parallel strategy that has received much research and development over recent years. However, in domain-decomposed environments where the problem geometry and tallies are divided between multiple processors, the traditional calculation of the sample variance presents communication issues that can severely limit the scalability of the parallel algorithm. Calculating tally statistics in Monte Carlo eigenvalue calculations typically involves calculating the sample variance based on particle histories.

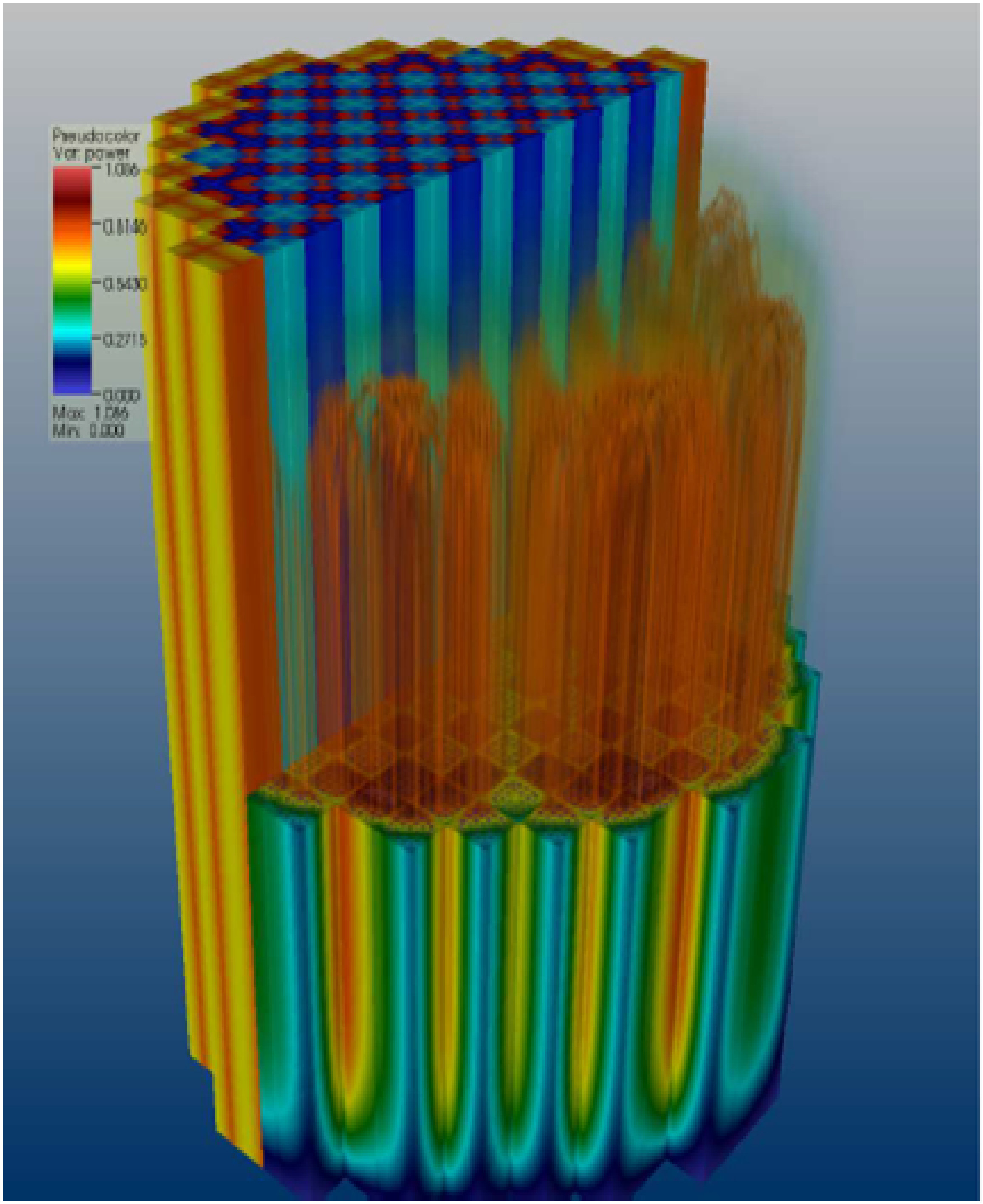


Figure 1.1: Power profile for a generic Westinghouse PWR (PWR-900) [1]

To address these issues, this research involves building the necessary infrastructure to provide coupled transport-depletion capabilities for full-core analysis on massively parallel systems using the Shift hybrid radiation transport code. Accomplishing these goals requires a multistep research and development process which included the following tasks:

1. Investigating the sources of bias and underprediction in the flux and its associated uncertainty in modern Monte Carlo calculations. Since the flux is the fundamental quantity of interest in a depletion calculation, ensuring its accuracy is of the utmost importance. This research will be used in the development of methodologies and algorithms for estimating the flux and statistical uncertainty in Shift. Specific focus will be on pin cell and assembly models with axial burnup distributions to evaluate models similar to develop guidelines and concepts to aid in future development of Shift's statistical algorithms.

2. Developing the algorithms for accurately calculating statistical uncertainties in Shift. This will include developing additional algorithms using the knowledge gained in previous studies to eliminate sources of bias and underprediction in tally results and provide the necessary diagnostics to make a valid assessment of the estimated results.

3. Development and implementation of a methodology capable of efficiently calculating tally statistics in domain-decomposed environments; a task which has not yet been performed in any known production-level Monte Carlo code. This will include researching documented methods for calculating statistical uncertainties and assessing their applicability to domain-decomposed Monte Carlo environments. Investigation of possible alternatives will also be conducted, which will require rigorous testing and verification if chosen as a prime implementation in Shift's statistical analysis.

4. Research and development of an interface to efficiently integrate ORIGEN [4] depletion capabilities into the Exnihilo framework to perform depletion and isotopic tracking analyses following a transport calculation.

5. Testing and validation of all of the above implementations. This includes testing the ability of Shift to efficiently utilize the ORIGEN depletion module for performing coupled transport-depletion calculations using Shift.

Accomplishing these tasks first involves developing the algorithms to accurately calculate localized tally uncertainties in parallel environments both efficiently and effectively. Research will be conducted to identify the common issues associated with current Monte Carlo statistical algorithms in order to aid in the development process of new algorithms. Furthermore, the developed algorithms must be geared towards functioning in parallel Monte Carlo simulations which can utilize either a domain-replicated, domain-decomposed, or a multi-set overlapping domain scheme, noting that calculating tally variances in domain decomposed Monte Carlo is a feat that has not yet been implemented in any available Monte Carlo code. In this regard, a new yet simple methodology has been developed in order to perform tally statistics in domain decomposed environments which requires rigorous testing to ensure its robustness. These algorithms will be coded in the Shift hybrid radiation transport package, which is programmed in C++.

As advancements are made in the world of computational simulation, most notably high-performance computing, the radiation transport community is getting closer to being able to reach the goals for full-core reactor modeling and simulation. The combination of the objectives presented will enable a full-core depletion calculation to be performed by Shift on a massively parallel system such as the Cray XK7 (Titan) machine at the National Center for Computational Sciences [5]. A Monte Carlo simulation of this magnitude is something that has been the goal of computational transport for many decades, and Shift is the ideal platform to carry out such a task.

1.3 Motivation

Nuclear power is a viable and well-established technology for clean energy production on a large-scale, and the nuclear power industry has been performing research and development for some time on nuclear systems which are much improved with respect to cost, safety, and sustainability. However, the pace at which these new technologies can be developed and deployed into viable options and our ability to advance the state-of-the-art for such systems is limited by inherent approximations in our aging computational tools and current approach. There is a definite need for and programmatic opportunities associated with drastic, not incremental, improvements in our Modeling and Simulation (M&S) capabilities. The inability to do significant experimental testing and/or prototyping places increased reliance on M&S. However, current computational tools are ill-suited for predictive, efficient M&S, especially with regard to the design of new and novel energy systems.

Under the support of the University of Tennessee's Science Alliance's Joint Directed Research and Development (JDRD) Fund as well as Oak Ridge National Laboratory (ORNL), a study of the Monte Carlo method for eigenvalue problems was conducted. The specific focal points of this study were on the issues associated with underprediction of localized tally uncertainties in Monte Carlo eigenvalue calculations and on uncertainty estimation in domain-decomposed Monte Carlo environments for massively parallel systems. By developing a thorough understanding of the current issues related to uncertainty underprediction in modern Monte Carlo computational codes, the development of new uncertainty estimation algorithms tailored for domain decomposed Monte Carlo calculations can utilize this knowledge to mitigate and/or eliminate these issues while simultaneously developing new methodologies.

Following this research, under appointment to the U.S. Department of Energy (DOE) Nuclear Nonproliferation International Safeguards (NNIS) Graduate Fellowship Program sponsored by the National Nuclear Security Administration (NNSA)'s

Office of Nonproliferation and International Security, depletion capabilities for full-core Monte Carlo calculations were developed. The ability to decompose the geometry among multiple processors circumvents the prohibitive memory requirements associated with tallying in hundreds of thousands of pin cells and hundreds of axial regions per pin cell, thus making full-core depletion calculations more readily available to the radiation transport community.

Chapter 2

Literature Review

This section contains a review of pertinent literature. Topics include:

1. fundamental issues in Monte Carlo eigenvalue calculations,
2. parallel decomposition strategies in Monte Carlo and tallying in the various decomposed environments,
3. integration of isotopic depletion capabilities, and
4. full-core computational analysis.

2.1 Fundamental Monte Carlo Practice

Modeling and simulation using Monte Carlo methods is the gold standard for nuclear reactor criticality applications, allowing complex geometries to be represented in multidimensional space with minimal approximations. However, obtaining good statistics for localized tallies in eigenvalue calculations takes a large amount of computation time, and several issues exist that may lead to errors in both the average value and the associated uncertainties generated by Monte Carlo algorithms. Developing an understanding of these issues and knowing how to identify and eliminate their effects are not only vital to the proper use of Monte Carlo methods, but vital to the development process of Monte Carlo algorithms.

With respect to general Monte Carlo practice, many issues have been identified in the literature which have no general solutions [6], [7], [8]. For example, it is well-known that statistical estimates obtained from Monte Carlo criticality simulations can be adversely affected by cycle-to-cycle correlations in the fission source, which can lead to estimates of statistical uncertainties that are lower than the true uncertainty by a factor of 5 or more [2]. However, several other more fundamental issues such as adequate source sampling over the fissionable regions and convergence of the fission source distribution can have a significant impact on the uncertainties for the calculated eigenvalue and localized tally means, and these issues may be mistaken for effects resulting from cycle-to-cycle correlations. In worst-case scenarios, the uncertainty may be underpredicted by a factor of 40 or more [7]. The issue of inter-cycle correlations typically presents itself in cases where the dominance ratio is near unity, and no alternate specification of Monte Carlo parameters can correct this issue. The other issues which relate to source sampling and convergence can be dealt with by ensuring that the number of initial cycles skipped and the number of histories per cycle are chosen correctly, and this responsibility remains up to the user, not the developer.

When assessing the validity of a Monte Carlo eigenvalue simulation, the primary concern for the practitioner should be the convergence of the fission source. By observing k_{eff} and the Shannon entropy [9] as a function of cycle, one can generally identify the point at which the fission source has reached a stationary distribution. Depending on the geometry and physics associated with a given model, a large number of histories per cycle may be required to obtain convergence. By paying close attention to these diagnostics one can more easily identify the convergence of the fission source distribution. Furthermore, these diagnostics may aid in ensuring that source regions are adequately sampled, as this is not something that Monte Carlo codes deal with automatically, and statistical checks are not guaranteed to catch these types of issues.

Most production-level Monte Carlo codes print out edits to help identify when the source is actually converged. A much more challenging issue is the impact of cycle-to-cycle correlations on statistical estimates as the effects of these correlations are more difficult to expose and correct for using typical Monte Carlo techniques and can lead to underprediction in the uncertainty that can be as high as a factor of 5 or more [2] in select cases, primarily in cases where the dominance ratio is close to 1 [10]. Most codes provide no information about the effects of cycle-to-cycle correlations on localized tally results, and there are no generic recommendations on how to deal with this issue using only Monte Carlo[*]. Previous results have shown that premature initiation of active cycles can lead to a high correlation between cycles [11]. By ensuring that a stationary distribution has been reached before beginning the active cycles, one can at least rule out this source of correlation. Unfortunately, this is not the only reason cycle-to-cycle correlations may be present.

Since Monte Carlo methods are widely used in criticality safety applications and are increasingly being used for benchmarking reactor analyses, an in-depth understanding of the effects of these issues must be developed in order to support the practical development and usage of Monte Carlo software packages.

2.2 Parallel Computing

In practice, there are substantial limitations on Monte Carlo performance, primarily due to the size and complexity of a full-core reactor model and the slow convergence of the eigenvalue calculation and tallies [10]. Since the mid-1950s, computational technology has seen an average improvement of over 80% in computing power on a yearly basis [12]. Due to the computationally-intensive nature of Monte Carlo, the development and advancement of Monte Carlo computational methods have proceeded synchronously with the advances in computational hardware over

[*] Methods that utilize deterministic calculations are available but are only in the development phase in most Monte Carlo codes.

the course of history. As advancements are made in the world of computational simulation, most notably high-performance computing, the radiation transport community is becoming closer to being able to reach the goals for full-core reactor modeling and simulation, and embedded in this research initiative is the ability to utilize the parallel processing capabilities of today's largest machines in an efficient and effective manner.

The two primary limitations of Monte Carlo calculations are the computational time to obtain statistical uncertainties below a desired threshold, and the amount of memory available to store space- and energy-dependent results.

The first limitation has been overcome to some degree by dividing the number of histories between multiple processors. Furthermore, results have shown that, for full-core Monte Carlo Light Water Reactor (LWR) models, the communication costs represent a small fraction of the total run time [13]. Other issues associated with Monte Carlo calculations which are not strictly dependent on computational time or memory include convergence issues associated with k_{eff}, the fission source distribution, and tally results, as well the biases, underprediction of statistical uncertainties, and inter-cycle correlations. Some, but not all of these issues can be mitigated by simulating more particle histories, making parallel Monte Carlo even more supportive for obtaining accurate, valid results in a reasonable time frame [8].

Thus the major limitation on modern Monte Carlo algorithms for performing full-core analyses is the amount of available memory [8]. This is an especially concerning issue as current research and development is extending focus on utilizing Field-Programmable Gate Arrays (FPGAs) and Graphics Processing Units (GPUs) in addition to multi-core processors due to the fact that GPUs have been showing a sharper increase in computational performance in comparison to Central Processing Units (CPUs) over the past decade. Figure 2.1 illustrates the increase in computational performance for both CPU and GPU from 2003 to 2008.

The reason that GPU-based computing has just recently become a prominent research endeavor in the scientific community is because GPU architectures operate

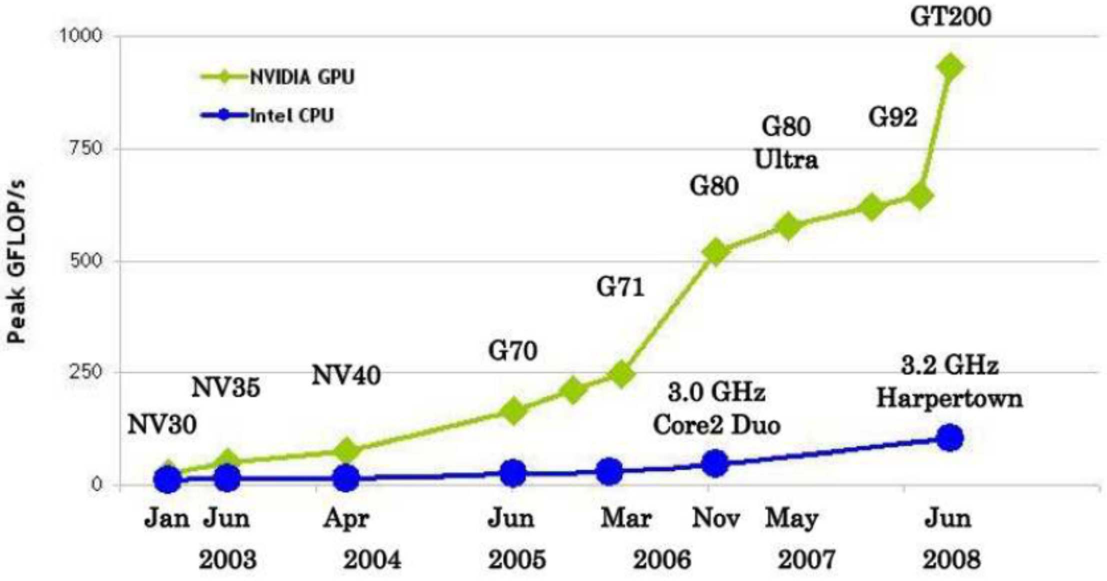


Figure 2.1: GPU vs. PCU performance over time [14]

significantly different than CPU architectures, and vice versa. The CPU is designed for sequential calculations, where the data caches are large, thus reducing overhead from reading and writing data. The GPU on the other hand is designed for parallel calculations where the data caches are small but much faster with respect to transferring data between Random Access Memory (RAM) and the processor* [18]. The issues with employing GPUs in Monte Carlo algorithms are:

1. they have a much lower amount of available memory†,
2. lack of error-checking/correction within the memory,
3. optimization efforts can lead to code that is difficult to read,

*As of 2013, the highest data transfer rates for GPUs are 264 Gigabyte (GB)/s (AMD Radeon 7970) [15] and 288.4 GB/s (NVIDIA GeForce GTX TITAN) [16] while the data transfer rate for CPUs is approximately 25.6 GB/s (Intel Core i7) [17].

†The maximum amount of memory available as of 2013 is 6 GB [16] for 896 cores, which equates to just under 7 Megabyte (MB)/core)

4. can be largely affected by branch statements which Monte Carlo algorithms utilize quite often [18].

Therefore it has taken some time for the computational community to research and develop new methods and algorithms which are capable of efficiently utilizing GPU architectures.

It is expected by 2020 that there will be processor units with over one million CPU-cores, and as Forest Brown stated in 2011, "none of the scientific and engineering software in use today can be scaled to such extreme processor counts" [19]. Thus popular focus areas in Monte Carlo development include various decomposition strategies such as spatial decomposition and data decomposition.

Methods have been developed for Monte Carlo condensed matter simulations using domain decomposition [20], however this methodology requires that the domains be spaced greater than the maximum particle interaction range, which for Monte Carlo neutron transport calculations corresponds to the maximum track length. Unfortunately for reactor physics, the maximum track length cannot truly be calculated.

2.3 Modular Depletion

Similar to neutron transport codes, many codes which perform isotopic depletion or contain isotopic depletion capabilities exist in the computational community. Given the goals of this research, it is undesirable to build a new depletion code from scratch due to the intense validation efforts that would be required. Therefore it would be best to make use of an existing code and simply provide a coupling interface to communicate data to and from the depletion code. Some of the more documented depletion/depletion-capable codes include PEPIN2 [21], BISON-C [22], DEPLETOR [23], CINDER90 [24], MONTEBURNS [25], SARAF [26], Serpent [27], TRITON [28], and ORIGEN [4], and these codes are often used in conjunction with a transport or

Table 2.1: Survey of various isotopic depletion software packages used in neutronics analysis.

Depletion Module	Used With
PEPIN2	APOLLO2/DARWIN [21]
DEPLETOR	PARCS* code [23]
CINDER90	MCNP/MCNPX [25], [24], [29]
SARAF	MCNP4C/BGCore [26]
Serpent	embedded within Serpent [27]
ORIGEN-S	standalone, SCALE (TRITON), Monteburns [4], [28], [30] [25]
BISON-C	standalone [22]

diffusion code for neutronics analysis. Table 2.1 lists each of these depletion modules and the codes they are most-often associated with.

The issue with simply coupling to one of these depletion codes is that they require a large amount of data such as multigroup cross sections, decay constants and fission yield data from multiple libraries such as the Evaluated Nuclear Data File (ENDF) [31] and Joint Evaluated Fission and Fusion (JEFF) [32] libraries, isotopic data such as atomic masses and concentrations, and neutron flux spectrum data. Furthermore many of these codes are either tightly coupled to another code, fully embedded within another code, or can only be coupled via file Input/Output (I/O). In order to evaluate which code would be a best fit for the research being conducted in this dissertation, the modularity of the code must be assessed as well amount of file I/O since reading and writing from a file is highly unfavorable in massively parallel environments.

In the case of DEPLETOR for example, it is built to read PARCS-based input files to perform cross-section processing and group collapsing for depletion. CINDER90 was recently integrated into MCNPX [29] using a file I/O-type interface illustrated in Figure 2.2. Both MONTEBURNS and TRITON utilize file I/O to couple transport to ORIGEN[†]. The flow diagrams for these processes are illustrated in Figures 2.3 and 2.4.

[†] MONTEBURNS recently coupled to CINDER90 as well [26]

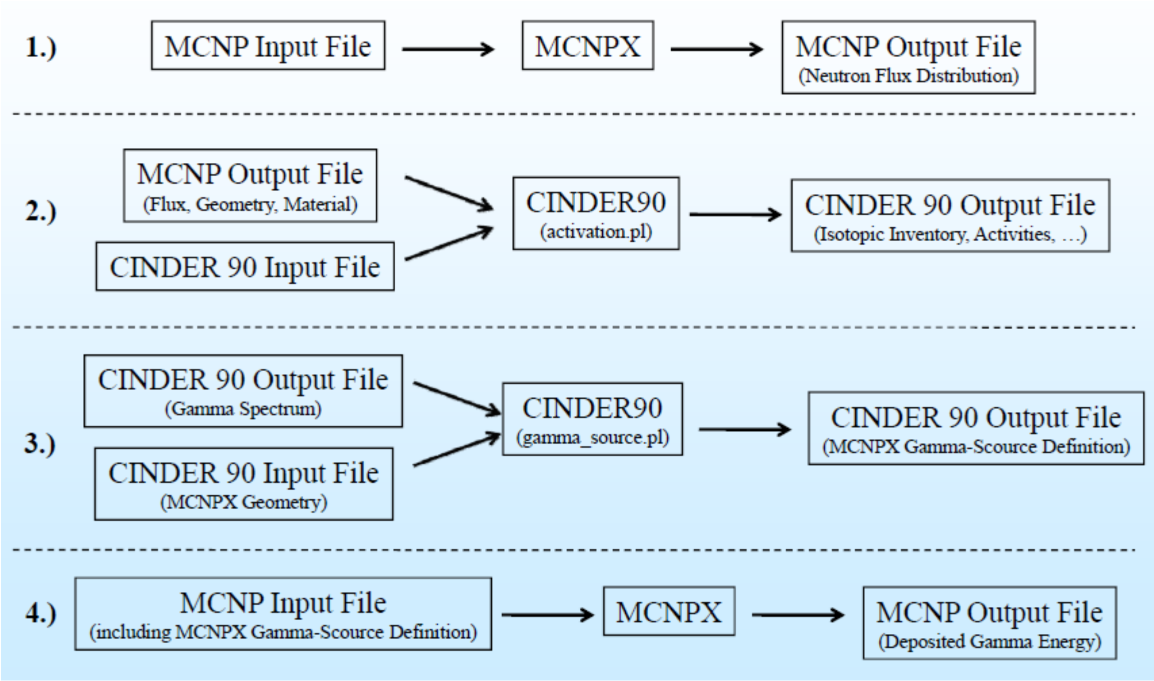


Figure 2.2: Flow diagram for an MCNPX/CINDER90 calculation.

None of the codes evaluated appear to be in a generalized "modular" format which would allow for easy integration into an existing transport code. However, there have been several modular versions of the ORIGEN code developed over the years, most recently for use in the Advanced Multi-Physics (AMP) nuclear fuel performance code [33] as well as the Nodal Eigenvalue, Steady-State, Transient, Le Core Evaluator (NESTLE) advanced nodal diffusion code [34][*]. The drawback of the integration of these standalone variants of ORIGEN in these cases is that the source code itself is simply packaged and built within the respective codes and is therefore not version-controlled or kept up to date with ongoing ORIGEN development.

The modular version of ORIGEN presented in Figure 2.5 was developed such that the input and output variable can be accessed using functions, therefore decrease the

[*] AMP does not use this modular format anymore and instead uses the modular version of ORIGEN distributed with SCALE.

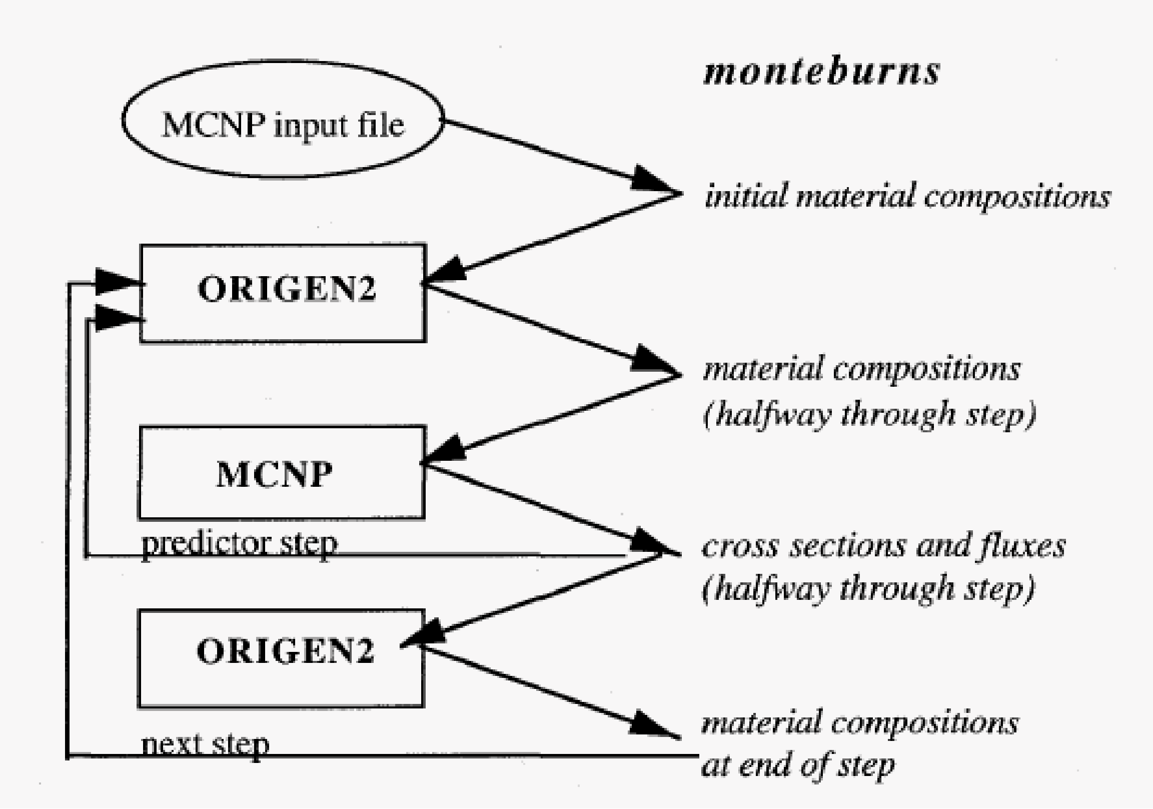


Figure 2.3: Flow diagram for a MONTEBURNS calculation.

amount of file I/O. There are also functions available for problem initialization and to execute ORIGEN.

Fortunately, the ORIGEN code within SCALE has been recently restructured in the latest version of SCALE (6.2+) [30] to provide a single modular version accessible via C++ or Fortran Application Programming Interfaces (APIs) to allow for seamless integration with other codes. The modular APIs have most but not all of the features and capabilities required to perform fully-coupled transport-depletion calculation. The only remaining development to give the APIs full mutable access to all of the nuclide data contained in the ORIGEN library is to provide the capability to modify the fission yield data on-the-fly. Additional development is also required in order to provide access to the raw JEFF neutron activation data [32], which contains cross

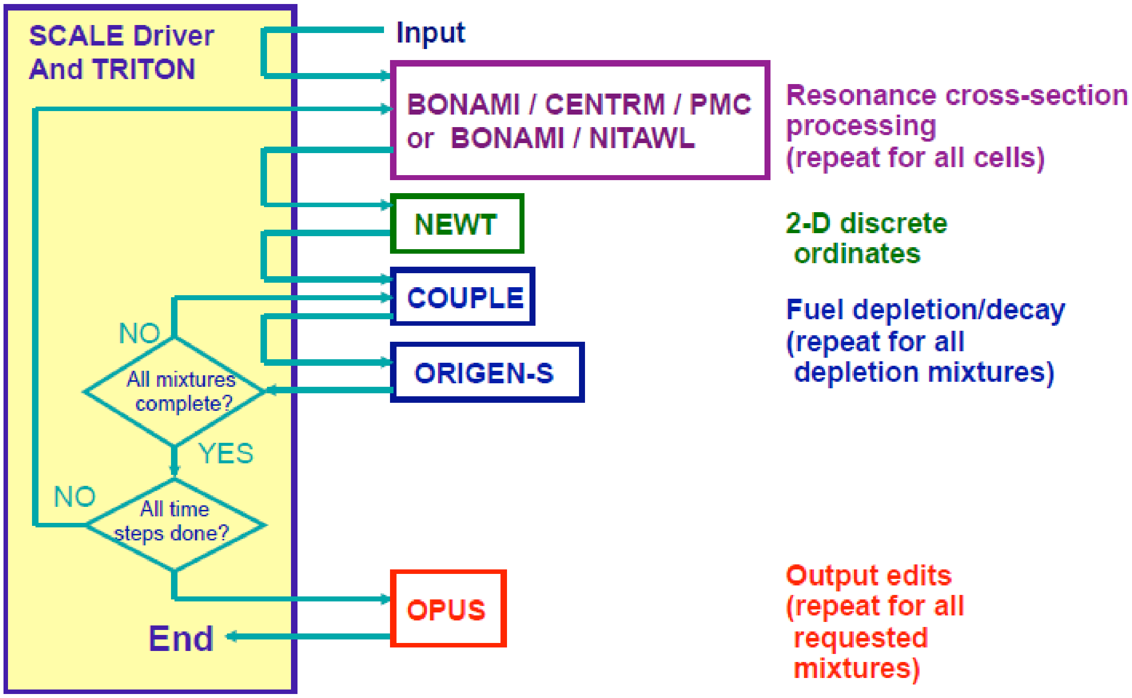


Figure 2.4: Flow diagram for a TRITON calculation.

section data for threshold reactions such as (n, p) and (n, α). However, it is not the responsibility of the ORIGEN package to provide this data, so this development will likely need to take place elsewhere. Modification of both the fission yields and JEFF cross sections were previously performed by the COUPLE [36] module which is used by the TRITON [28] control sequence in SCALE to perform coupled transport and depletion calculations.

2.4 Full-Core Computational Analysis

The ultimate goal of computational reactor analysis tools has always been to be able to predict and evaluate the values of key quantities associated with nuclear reactor design, safety, and operation. In the case of reactor operation, this equates to real-time core monitoring and simulation. The demands of radiation transport methods for

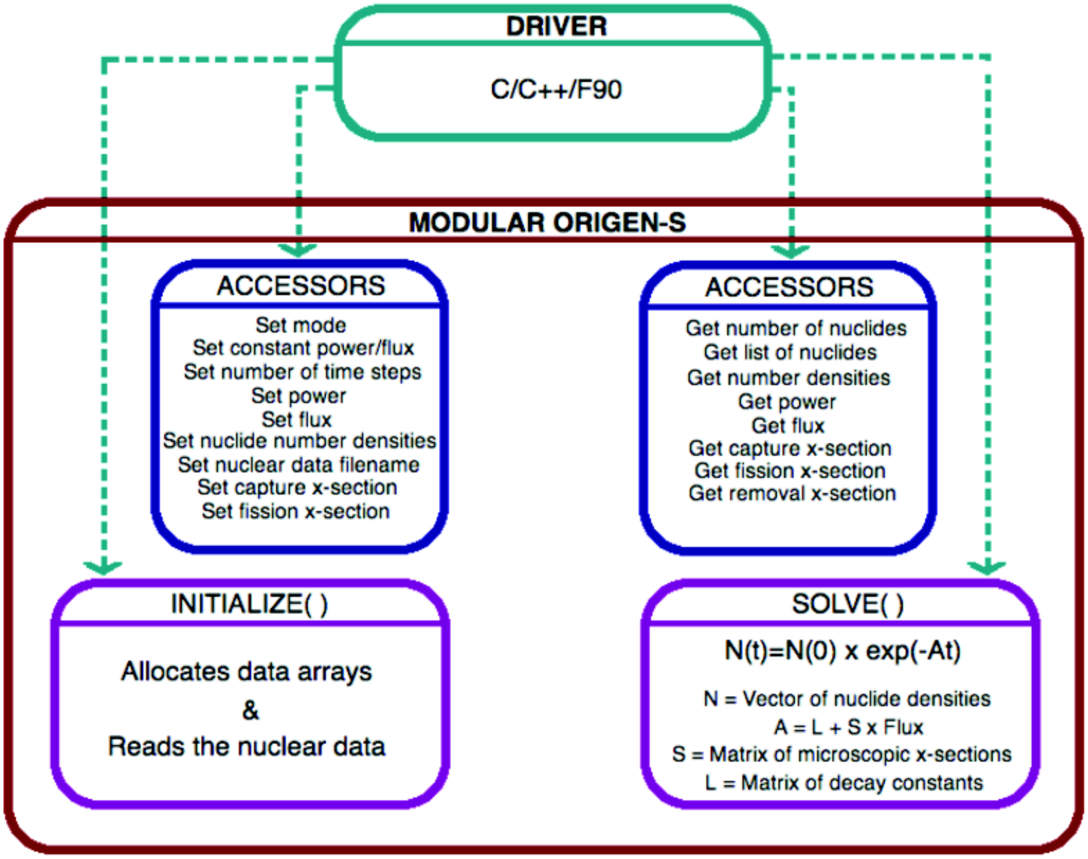


Figure 2.5: Layout of the modular ORIGEN functions used in the version integrated in the AMP and NESTLE codes [35].

complete reactor analysis requires tens of thousands of steady-state three-dimensional core calculations, hundreds of transient core calculations, thousands of operational support calculations, and continuous real-time core monitoring calculations. At an invited lecture at the M&C 2003 conference in Gatlinburg, Kord Smith [37] estimated that it will not be until the year 2030 that such full core computations can be done in under an hour on a single CPU*. Bill Martin also presented a similar analysis at an invited lecture at the M&C 2007 conference in Monterey which estimated that a full-core calculation with 40 000 fuel pins, 100 axial regions, and statistical uncertainties

*This estimate was based on Moore's law.

less than 1% will not be able to be accomplished in under one hour until the year 2019 [38].

To help measure the progress of the computational radiation transport community as they reach towards this goal, Hoogenbroom and Martin [6] have proposed a full-core benchmark model containing 241 17 × 17 assemblies for a total of approximately 70 000 individual pin cells. A full-core Monte Carlo analysis with depletion would recommend approximately 3 radial regions (fuel, cladding, and moderator) per pin cell and 100 axial regions, the total number of material regions totals over 20 million. Although not all of these regions are fuel regions, a complete depletion calculation still requires the number of tally cells to be on the order of 4 million.

Obtaining good statistics for the transport portion of the problem would involve the number of histories per cycle being on the order of 10^7–10^9 or more. This suggests that the number of parallel processes would need to be on the order of 10^3–10^6. The issue here is that, for a parallel Monte Carlo calculation in which 10^8 histories per generation are simulated, the initial broadcast of the source locations involves communicating approximately 2.8 Terabyte (TB) of data[*]. Furthermore consider that, based on a simple estimation presented by Romano and Forget [39], a depletion calculation in which each material contains 20 isotopes that need to be tracked requires approximately 500 bytes per region. Therefore the amount of memory required to store the material and tally data for the fuel regions only requires on the order of tens to hundreds of GB of memory. Adding in the geometry and cross section data, and considering that most high-performance systems have distributed memory environments where the memory is shared between 8 or more processors, each computational node would require at least 80 GB of memory.

The bottom line is that the uncertainties suggested by reactor modeling and simulation largely stem from the multitude of assumptions and approximations used in transport calculations, the uncertainties associated with nuclear data, and the

[*] This estimate assumes that each source location is composed of three double-precision floating point numbers (8 bytes each) to store the physical coordinates and one integer identifier (4 bytes).

statistical uncertainties resulting from stochastic processes such as Monte Carlo calculations. Obtaining these results with the desired level of precision and accuracy sufficient for reactor design and safety analyses is only now becoming a practical research initiative as the computational resources are becoming available. As advancements are made in the world of computational simulation, most notably high-performance computing, the radiation transport community is becoming closer and closer to being able to reach the goals for full-core reactor modeling and simulation and embedded in this research initiative is the ability to utilize the parallel processing capabilities of today's largest machines in an efficient and effective manner.

Chapter 3

Background

This chapter presents a brief discussion of the theory associated with radiation transport (primarily the Monte Carlo method) and time-dependent depletion analysis. Of particular importance to this work is the discussion of Monte Carlo transport and its precision with respect to statistical uncertainties for localized tallies in eigenvalue calculations as well as the impact of different parallel decomposition strategies on traditional Monte Carlo methods and algorithms.

This discussion is not intended to be a complete discussion, but rather a brief review to familiarize the reader with the basic theory examined within. For a more comprehensive discussion, consult one of the many related textbooks available on the subjects [40] [41] [42].

3.1 Radiation Transport Theorey

Solving radiation transport problems in nuclear engineering applications typically involves solving some form of the neutral particle Boltzmann transport equation. In its most basic form, this equation is a conservation of particles where particle inventory is kept using some form of the neutron density distribution throughout the core, $N(\vec{\mathbf{r}}, E, t)d^3rdE$, which represents the expected number of neutrons in d^3r about $\vec{\mathbf{r}}$, energies in dE about E, at time t [40]. Most radiation transport computational calculations however are steady-state and use what is called the angular flux, $\psi(\vec{\mathbf{r}}, E, \hat{\boldsymbol{\Omega}})$, which represents the neutron density at position, $\vec{\mathbf{r}}$, energy, E, and direction, $\hat{\boldsymbol{\Omega}}$, within the phase space $(d\vec{\mathbf{r}}, dE, d\hat{\boldsymbol{\Omega}})$. The time-independent general form of the neutral particle Boltzmann transport equation, relevant for neutron and photon transport through a material, is given by

$$\begin{aligned}
&\vec{\nabla} \cdot \hat{\boldsymbol{\Omega}}\psi(\vec{\mathbf{r}}, E, \hat{\boldsymbol{\Omega}} + \sigma_t(\vec{\mathbf{r}}, E)\psi(\vec{\mathbf{r}}, E, \hat{\boldsymbol{\Omega}}) = \\
&\qquad \frac{\chi(E)}{4\pi} \int_0^{\infty} dE' \int_0^{4\pi} \hat{\boldsymbol{\Omega}}' \nu(E') \sigma_f(\vec{\mathbf{r}}, E' \rightarrow E, \hat{\boldsymbol{\Omega}}' \rightarrow \hat{\boldsymbol{\Omega}}) \psi(\vec{\mathbf{r}}, E', \hat{\boldsymbol{\Omega}}') \\
&\qquad + \int_0^{\infty} dE' \int_0^{4\pi} d\hat{\boldsymbol{\Omega}}' \sigma_s(\vec{\mathbf{r}}, E' \rightarrow E, \hat{\boldsymbol{\Omega}}' \rightarrow \hat{\boldsymbol{\Omega}}) \psi(\vec{\mathbf{r}}, E', \hat{\boldsymbol{\Omega}}') \\
&\qquad + Q(\vec{\mathbf{r}}, E, \hat{\boldsymbol{\Omega}})
\end{aligned} \tag{3.1}$$

where

ψ angular flux at $\vec{\mathbf{r}}$ per unit volume, in direction $\hat{\boldsymbol{\Omega}}$ per unit solid angle, and at energy E per unit energy

σ_t total macroscopic cross section for interaction at $\vec{\mathbf{r}}$ and energy E

χ average number of fission neutrons emitted from fission reactions induced by neutrons with energy E'

ν average number of fission neutrons emitted from fission reactions induced by neutrons with energy E'

σ_s scattering cross section at $\vec{\mathbf{r}}$ from energy E' and direction $\hat{\boldsymbol{\Omega}}$ to energy E and direction $\hat{\boldsymbol{\Omega}}$

In Equation 3.1 the mechanisms for neutron loss are on the left-hand side of the equation and the mechanisms for neutron production are on the right-hand side of the equation [43]. The neutron loss terms account for the streaming rate of loss from the phase space,

$$\vec{\nabla} \cdot \hat{\boldsymbol{\Omega}}\psi(\vec{\mathbf{r}}, E, \hat{\boldsymbol{\Omega}}) \tag{3.2}$$

and the rate of loss from nuclear interactions within the material(s),

$$\sigma_t(\vec{\mathbf{r}}, E)\psi(\vec{\mathbf{r}}, E, \hat{\mathbf{\Omega}}) \tag{3.3}$$

The neutron production terms account for the rate of neutron production from fission reactions,

$$\frac{\chi(E)}{4\pi}\int_0^\infty dE' \int_0^{4\pi} \hat{\mathbf{\Omega}}'\nu(E')\sigma_f(\vec{\mathbf{r}}, E' \to E, \hat{\mathbf{\Omega}}' \to \hat{\mathbf{\Omega}})\psi(\vec{\mathbf{r}}, E', \hat{\mathbf{\Omega}}') \tag{3.4}$$

the rate of neutrons emerging from scattering reactions,

$$\int_0^\infty dE' \int_0^{4\pi} d\hat{\mathbf{\Omega}}'\sigma_s(\vec{\mathbf{r}}, E' \to E, \hat{\mathbf{\Omega}}' \to \hat{\mathbf{\Omega}})\psi(\vec{\mathbf{r}}, E', \hat{\mathbf{\Omega}}') \tag{3.5}$$

and an external source term (if present),

$$Q(\vec{\mathbf{r}}, E, \hat{\mathbf{\Omega}}) \tag{3.6}$$

Reactor analysts are concerned with a number of key quantities such as neutron flux, energy deposition, and other reaction rate quantities as well as the detailed isotopic inventory of various reactor materials over time. Knowledge of the neutron distribution, or more importantly the reaction rate distribution, is important in many aspects of nuclear engineering. Several uses of the neutron distribution in nuclear reactor design include energy deposition in a material, the time-dependent rate of change of nuclide densities (often referred to as burnup), and in calculating k_{eff}.

The two categories in which modern computer codes solve the transport equation are deterministic and stochastic. Deterministic methods in general involve the use of several approximations and discretization of the independent variables (space, energy, and direction) along with the application of one or more numerical methods to solve the neutral particle Boltzmann transport equation (Equation 3.1) for the average particle behavior. Deterministic solution methods include Discrete

Ordinates (S_N), the integral transport method, Method of Characteristics (MOC), and diffusion theory. The most general comparison between these two methodologies is that deterministic solutions provide exact solutions to approximate models whereas stochastic methods give approximate solutions to exact models.

3.1.1 Deterministic Methods

The deterministic detail for full-core reactor modeling requires approximately 10 000 spatial nodes, 30 000 energy points, 200 directions, and 300 time steps. This puts the total number of unknowns for such a calculation on the order of 10^{13}, which is a difficult problem to solve even using the most advanced computational tools and methods. Furthermore, the demands of radiation transport methods for complete reactor analysis require on the order of 10 000 steady-state three-dimensional core calculations, 100s of transient core calculations, 1000s of operational support calculations, and continuous real-time core monitoring calculations [37]. To lessen this undertaking, modern computational processes typically utilize various assumptions and approximations in order to generate results in a timely manner. Unfortunately, reactors operate within the uncertainties estimated by computational simulations, and these approximations make it difficult to accept results from such simulations for safety reasons. This makes deterministic full-core neutron transport analysis quite the enormous task at present, and current computational hardware cannot support these demands for real-time core monitoring.

3.1.2 Stochastic Methods

Stochastic methods for neutron transport include the Monte Carlo method which involves simulating particle transport by randomly sampling the many mathematical distributions or probability density functions that define neutron transport and more specifically, neutron interactions with a material. Results are then attained by recording important information from the individual, independent particle simulations to provide mean values for the average particle behavior. For large-scale applications, the Monte Carlo method is considered the “gold standard”, especially in nuclear reactor and criticality calculations, because it is able to represent complex geometries in multi-dimensional space using continuous-energy data with minimal approximations in comparison to deterministic methods. A more in-depth discussion of Monte Carlo methods is provided in the following section.

3.2 Monte Carlo

To circumvent the restrictions of deterministic methods, Monte Carlo methods are the preferred means for performing 3-dimensional neutron transport analysis. Monte Carlo methods offer several advantages over deterministic methods for full-core high-performance computing applications, namely in being able to avoid complicated meshing for complex geometries and multi-group cross section generation and processing. Furthermore, the Monte Carlo method is much more flexible when it comes to parallelization and adapting the algorithms to High-Performance Computing (HPC) architectures [44]. However, several challenges remain before Monte Carlo methods can be practically applied to full-core computational analysis.

Since the introduction of Monte Carlo methods into the nuclear engineering community in the 1940s at Los Alamos National Laboratory (LANL) [45], they have quickly become the “gold standard” for nuclear reactor criticality applications. The ability to represent complex geometries in multi-dimensional space without the numerous assumptions typically implemented in three-dimensional deterministic calculations gives Monte Carlo methods the fidelity to be used for criticality safety, radiation shielding, and other safety-related aspects of nuclear engineering.

Being a stochastic process, the Monte Carlo method is quite different than deterministic solution methods. Unlike deterministic methods which numerically solve the transport equation to obtain the flux, Monte Carlo methods do not solve the transport equation. Instead, Monte Carlo involves a type of numerical experiment whose expected value corresponds to a desired measure value such as the neutron flux in a particular geometric region or within a given material. The general assumptions made for a Monte Carlo neutron transport simulation are [8]

- static homogeneous media
- time-independent particle behavior
- Markovian (events only depend on current particle state) [46]
- particles do not interact with one another
- no relativistic effects
- particles travel in straight lines

In this experiment, particles follow what is often referred to as a "random walk" process. First a particle is born within the designated source region with an initial position, direction, and energy, which is sampled from an applicable Probability Density Function (PDF). Then the distance to the first interaction is determined using the total cross section in the material and the particle is transported to that location. Next the type of interaction the particle has at the collision site is determined as well as the resulting particle state variables (if the particle is still alive) by sampling from PDFs which are based on nuclear data (i.e. cross sections). The process must also note that certain interactions such as (n, γ) and $(n, 2n)$ result in the creation of new particles which also have to be tracked as they take their own "random walk". This process continues until the particle either leaves the phase space of interest or it is absorbed, after which the progeny born from the interactions of the initial source particle must be tracked in order to finish the particle's complete history within the system. Figure 3.1 provides a breakdown of the physical cross sections used to determine particle interaction.

As a particle is being transported within the phase space, any information of interest is tallied as contributions for the particle history. According to the law of

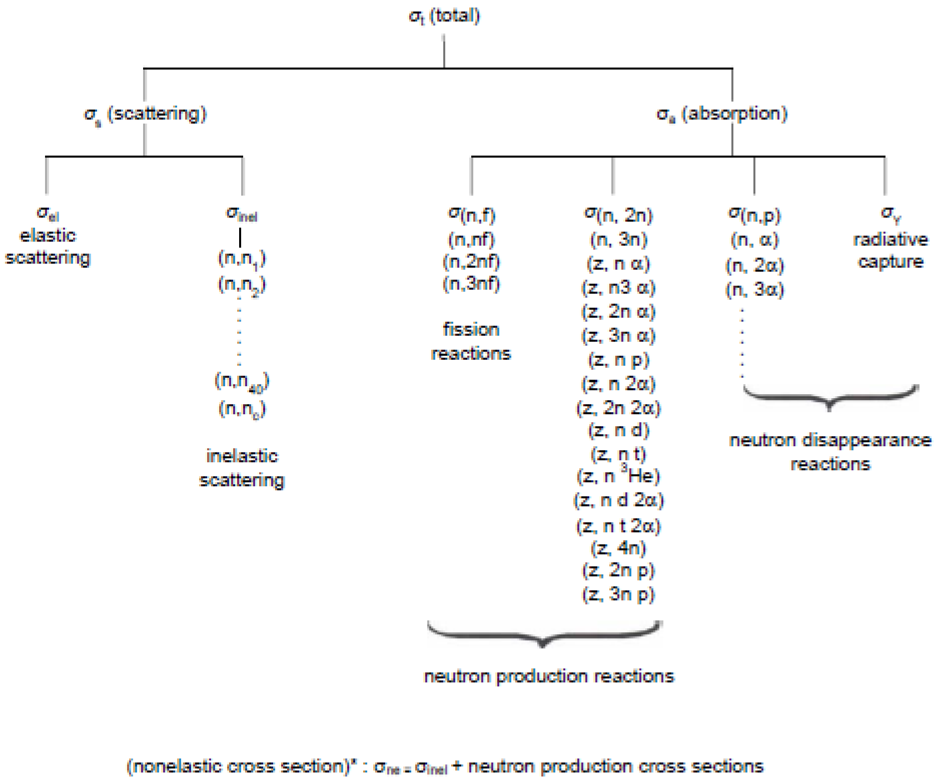


Figure 3.1: Cross section hierarchy for particle interactions [3].

large numbers [47],

$$f = \int_a^b f(x)\pi(x)dx = \lim_{N\to\infty} \frac{\sum_{i=1}^{N} f(x_i)}{N} \tag{3.7}$$

where x_i are chosen using a PDF, $\pi(x)$. Thus by simulating an appropriately large number of particles an estimation of the average particle behavior can be obtained. The Central Limit Theorem (CLT) then states that the sum of a sufficiently large number of identically distributed independent random variables is itself normally distributed. The law of large numbers along with the CLT provides the relationship between the result of a continuous integration and the result of a discrete sampling, and these two theorems construct the foundation of the Monte Carlo calculation.

Thus, the average particle behavior can be inferred using the individual tallied contributions made by the independent particle simulations to determine the quantity of interest within the physical system [47].

This simple example is what is referred to as *analog* Monte Carlo. There is also *non-analog* Monte Carlo, which is what most modern Monte Carlo codes implement. The discussion of *non-analog* Monte Carlo will be presented in more detail in Section 3.2.2.

The way in which Monte Carlo methods solve criticality problems is by simulating M batches (generations) of N neutron histories. The starting locations of the initial histories occur within the fissionable regions according to one or more PDFs typically specified by the user. As the histories are transported, the locations where a fission reaction would take place are stored in a buffer (bank). Following the completion of the first batch of neutrons, the starting locations for the next batch of N neutron histories are sampled from the banked fission site locations from the previous batch. This is done to prevent the neutron population from increasing exponentially or vanishing, however this also introduces a bias into the simulation because each banked location is not guaranteed to be used once and only once. In practice however, a large enough number of histories per generation will leads to a negligible bias in k_{eff}.

3.2.1 Source Convergence

Of particular importance when using this method of banking fission sites is the convergence of the Fission Source Distribution (FSD). The most general method in which the FSD is initialized is by evenly sampling from the fissionable regions within a given model according to their number density of fission cross section. The actual distribution of neutrons however is unlikely flat or directly proportional to the fission cross section for all fissionable regions, and thus fission sites will need to be sampled from some regions more than others. With each subsequent batch of neutrons beyond the first, the fission source distribution will be refined and begin approaching a stationary distribution, at which point the source is said to be converged. It is at this point in the calculation where any important information which will be used to estimate the final results should be tallied.

The primary concern for a Monte Carlo practitioner when assessing the legitimacy of a Monte Carlo calculation is first and foremost the convergence of the fission source. Two parameters are typically used to assess the convergence of the fission source: k_{eff}, and some other quantity or metric specific to source convergence [9], such as the Shannon Entropy or other diagnostic approaches discussed later in this section. By monitoring these two parameters, one can generally identify the point at which the fission source has reached a stationary distribution. Depending on the geometry and physics associated with a given model, the number of histories per cycle, the number of initial cycles skipped, and/or the total number of cycles may need to be much higher than what most Monte Carlo calculations require to obtain convergence. For example, in cases where the dominance ratio is near unity, the FSD converges slowly and may take hundreds of cycles to reach convergence, and even then it is much more difficult to determine whether or not convergence has been achieved [48].

Consequently, monitoring the convergence of the FSD not only helps ensure that the correct eigenvalue is converged upon, but it is also of particular importance to the statistical analysis of the tally estimates. Therefore, as repeatedly inferred, one must be sure that the FSD is converged prior to beginning tally accumulation. Although the exact cycle in which convergence is achieved is difficult to identify in advance, since this requires knowing the fundamental-mode of the FSD, convergence can typically be identified by observing plots of k_{eff} as a function of cycle. However, it should be noted that the convergence of k_{eff} does not guarantee that the FSD is also converged because k_{eff} is an integral quantity and thus it converges much faster than the fission source distribution [9]. Furthermore, plots of single-cycle k_{eff}, cumulative k_{eff}, or multi-dimensional plots of the source distribution versus cycle are sometimes difficult to interpret. Therefore one should also examine an additional parameter that focuses specifically on the convergence of the FSD as a function of cycle.

A series of papers by Taro Ueki and Forrest Brown discussed the stationarity and convergence of the fission source iteration process and the ability to characterize these phenomena using the Shannon entropy, H_{src} [49]. The Shannon entropy is a quantity derived from information theory to characterize the uncertainty associated with a random variable and has become a popular diagnostic tool in Monte Carlo calculations for measuring the convergence of the FSD for each cycle. The Shannon entropy of fission is computed by superimposing a three-dimensional mesh with N_s equal-sized mesh regions over the fissionable regions of a model, tallying the fraction of fission sites in each mesh region for a given cycle (P_j), and then evaluating

$$H_{src} = -\sum_{j=1}^{N_s} P_j \cdot \log_2(P_j) = \frac{\#\ of\ source\ particles\ in\ bin\ j}{\#\ of\ source\ particles\ in\ all\ bins} \tag{3.8}$$

Other diagnostic measures include the Fission Matrix Method (FMM) [50], which was recently shown to accelerate source convergence [51], the Coarse Mesh Projection Method (CMPM), and the Noise Propagation Matrix Method (NPMM) [52]. In addition, multiple convergence acceleration methods have been developed to deal with the issues of source convergence in cases where the dominance ratio is near unity. Cases that are loosely coupled converge slowly and may take hundreds of cycles to reach convergence, and even then it is exceptionally difficult to determine whether or not convergence has been achieved. Figure 3.2 illustrates how the Coarse Mesh Finite Difference (CMFD) accelerates convergence and enables one to obtain a more accurate FSD [48].

The superhistory powering method [53] was designed to eliminate some of the correlation between histories. Figure 3.3 illustrates the application of the forward-weighted Consistent Adjoint Driven Importance Sampling (CADIS) method to aid in source convergence and to increase the precision of the flux tally results. In this example, not only have all of the relative uncertainties been reduced to 1.0–5.0%, but the total run time was also decreased by a factor of 7.1 [54]. Multiple variations of these methods have been developed and each has its ideal application and limitations.

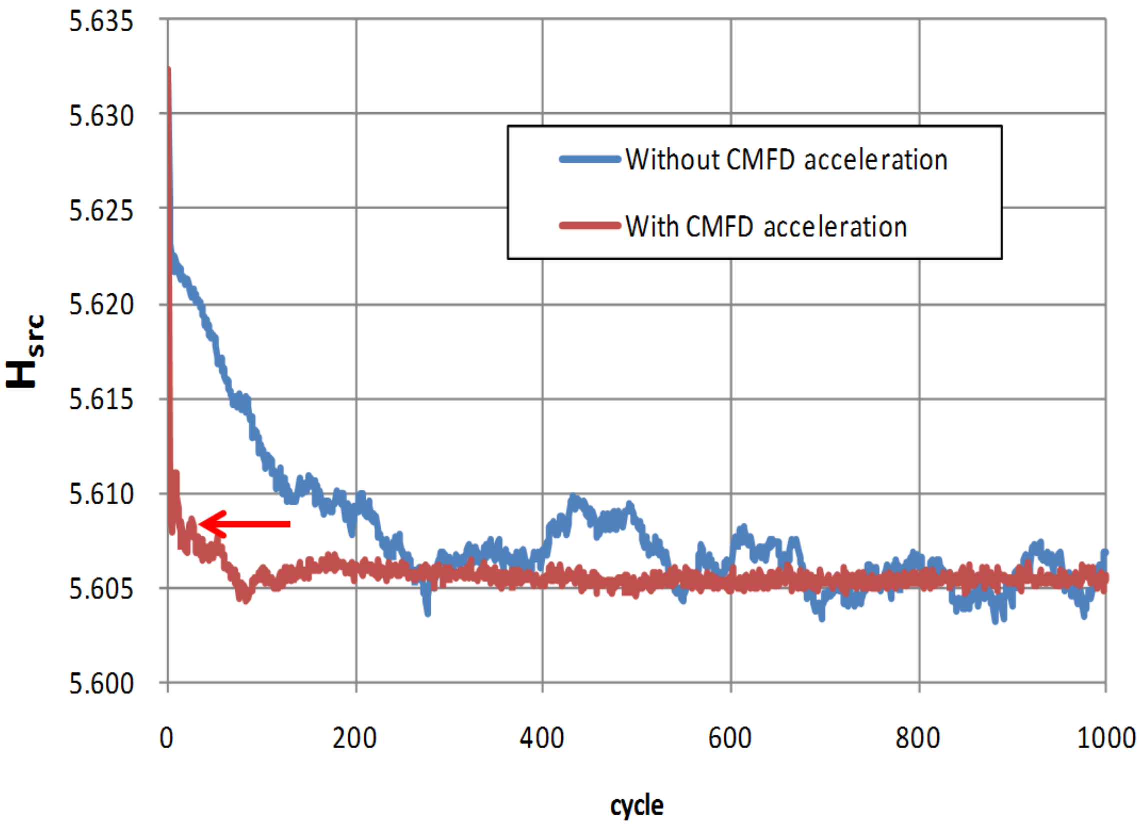
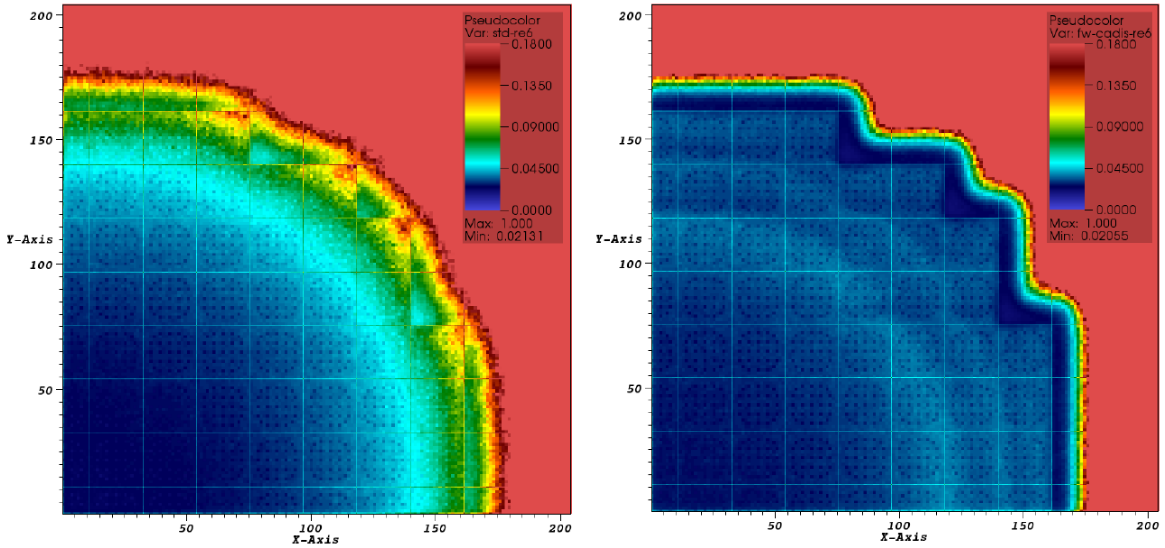


Figure 3.2: Shannon Entropy as a function of cycle both with and without CMFD acceleration (one million histories/cycle) [48].

Other important issues in the Monte Carlo methodology include the biases and uncertainty underprediction issues for both the eigenvalue and localized tally estimated. The biases present in a Monte Carlo eigenvalue calculation are:

1. a bias in k_{eff}, and
2. a bias in the reported variances.

The bias for k_{eff} is of the order M^{-1} and the bias of the variance is of the order $(NM)^{-1}$ where M is the number of cycles while N is the number of neutron histories per cycle [53].



(a) without FW-CADIS

(b) with FW-CADIS

Figure 3.3: Relative uncertainties computed using MCNP with (a) and without(b) FW-CADIS [54].

The bias in k_{eff} from inter-cycle correlations is given as

$$1 + [\frac{2\rho_k}{(1-\rho_k)}] \tag{3.9}$$

where ρ_k is the dominance ratio of the k^{th} eigenvalue mode. Using MacMillian's expression [55], the true standard deviation is given by (for large N)

$$\sigma^2_{R,M} \approx \frac{1}{M}[\sigma^2_R + 2\sum_{k=1}^{K} \frac{\alpha_k}{(1-\rho_k)}] \tag{3.10}$$

3.2.2 Tallies

The feature within a Monte Carlo simulation from which results are often inferred is commonly called a tally. Tallies are used to compute the expected value for a particular result that represents a physical or mathematical quantity within the system. Tallies can be flux-based, event-based, or emergent-based, however in most cases the tallies sought after are ultimately based on reaction rates which stem from the neutron distribution (flux).

Because Monte Carlo is a stochastic process, any result obtained from a Monte Carlo calculation must be accompanied by the statistical uncertainty associated with the estimated result. The behavior of both the mean value and its associated uncertainty, with respect to the number of particle simulations, represents an important analysis parameter which provides a measurement of the quality of the results. In addition, by observing these values as a function of the number of particle simulations, one can gain insight into whether or not the results are statistically well-behaved. This is of particular importance to the user because results that are not well-behaved may not reflect the true uncertainty, and thus the estimated mean value and uncertainty could be erroneous.

Using basic statistical methods, the expected value or mean is calculated as

$$\bar{x} = \int_a^b x p(x) dx \tag{3.11}$$

where $p(x)$ represents the probability density function for x in the range (a, b), such that

$$\int_0^\infty p(x) dx = 1 \tag{3.12}$$

However, Equation 3.11 represents the calculation of the true mean, $\bar{x}$, for a continuous distribution, $p(x)$. Because Monte Carlo is looking at a discrete distribution, Equation 3.11 takes the form

$$\bar{x} = \sum_{i=1}^{M} p_i x_i \tag{3.13}$$

where $p_i \geq 0$, for all i, and

$$\sum_{i=1}^{M} p_i = 1 \tag{3.14}$$

The same holds true for the estimate of the expected squared error. For a continuous distribution

$$\sigma^2(x) = \int_a^b p(x)(x - \bar{x})^2 dx \tag{3.15}$$

while for a discrete distribution

$$\sigma^2(x) = \sum_{i=1}^{M} p_i(x_i - \bar{x})^2 \tag{3.16}$$

Furthermore in a Monte Carlo simulation, the complete distribution is not known since only point-wise samples of the distribution are being taken. Therefore the results being estimated by the simulation are the sample mean and the sample variance of the mean, which are computed by

$$\hat{x}_N = \sum_{i=1}^{N} \frac{x_i}{N} \tag{3.17}$$

and

$$\sigma^2(\hat{x}_N) = \frac{\sigma^2(x)}{N} \tag{3.18}$$

and because the actual variance, $\sigma^2(x)$, is not known, the estimate of the sample variance is used

$$\sigma^2(\hat{x}_N) \cong S^2(\hat{x}_N) = \frac{S^2(x_i)}{N} \tag{3.19}$$

where

$$\S^2(\hat{x}_N) = \sum_{i=1}^{N} \frac{(x_i - \hat{x}_N)^2}{N - 1} \tag{3.20}$$

This gives rise to the inherent drawback of Monte Carlo: each additional digit of accuracy requires that the problem be run with 100 times as many histories, or

$$S(\hat{x}_N) \propto \frac{1}{N} \tag{3.21}$$

To provide an example of how tallies become results, a simple tally works in the following manner:

1. During the course of history i, when a particular event of interest occurs as a particle takes a step, j, through the physical system, a score consisting of a contribution, c_{ij}, multiplied by the particle weight, w_{ij}, is calculated.

2. These contributions are then accumulated in a tally bin, x_i, for that history.

3. At the end of the history, the first and second moments are accumulated

4. Steps 1–3 are then repeated for N histories

5. The Monte Carlo estimate of the tally mean is then computed using Equation 3.17 and the Monte Carlo estimate of the tally variance is computed using the following form of Equation 3.20

$$S^2(\hat{x}_N) = \frac{N}{N-1}(\sum_{i=1}^{N} \frac{x_i^2}{N} - (\sum_{i=1}^{N} \frac{x_i}{N})^2) \tag{3.22}$$

Note that in this simple explanation, the simulation is referred to as an *analog* simulation because the probabilities used in the decision-making process, as a particle is being transported, are explicitly derived from the exact physics (i.e. the transport model is analogous to the naturally occurring transport). Modern Monte Carlo computer codes implement what is called *non-analog* Monte Carlo, as the exact physical probabilities are modified to increase the efficiency of the simulation. Giving the particles a much better chance of contributing to a requested result or tally increases the sample size used for calculating tally statistics and this helps lower the variance more than in the *analog* case where fewer scores are made. To adjust for the modification however, the particles must carry a correction factor, which is defined as the particle's *weight*, in order to ensure that an unbiased result can be obtained from the simulation. The neutron balance for analog Monte Carlo is an exact balance of absorption, leakage, and production terms, whereas in non-analog (weighted) Monte Carlo the neutron balance is made on an on-average basis or an average balance of these terms. In many cases the various non-analog implementations are referred to as variance reduction techniques. By insisting that particles behave in such a way that they have a much better chance of contributing to the requested tallies, a larger sample size is available, which helps lower the variance for that particular tally result.

3.3 Full-Core Neutronics Analysis

Accurate modeling for full-core neutron transport has been a long sought after goal in reactor simulation. Complimentary to this is the ability to perform full-core depletion analysis and spent fuel isotopic characterization. Modern full-core reactor analysis calculations are typically performed using diffusion codes such as PARCS [23] and NESTLE [34] which utilize homogenized nuclear data over large spatial zones and a few-group cross section library [56]. This results in a simplified model, which is exceptionally far from the level of detail required to represent a "real" reactor. However, as computational capabilities advance, it is becoming possible to perform large scale calculations using deterministic and/or Monte Carlo formulations, which offer much higher precision in their statement and representation of the problem to be solved as well as the accuracy of the results.

When neutron transport and isotopic depletion calculations are coupled to one another, they are often grouped together to form what is called *neutronics* analysis. In a multi-physics reaction simulation, the neutronics portion can be expressed quite simply and is illustrated in Figure 3.4. A coupled transport-depletion (neutronics) calculation, which represents what can be termed a quasi-static neutronics calculation, can be divided into three categories:

Cross Section Processing Process ENDF data to create a set of problem-dependent *macroscopic* cross sections for the neutron transport calculation. Processing operations include performing resonance reconstruction, temperature-dependent

Doppler broadening, calculating energy-angle distributions of secondary particles, processing $S(\alpha, \beta)$ data for thermal moderators, and performing multigroup averaging of cross section data to create problem-dependent microscopic cross sections.

Neutron Transport Solving the transport equation to determine the space-energy distribution of neutrons in the reactor. The neutron transport calculation will also determine the region-averaged powers and fluxes for use in the depletion calculation.

Isotopic Depletion Track changes in isotopic concentrations in the relevant reactor materials over time. Isotopic tracking is required in order to determine the amount of fissionable material and the buildup or burnout of neutron poisons and burnable absorbers in the reactor over time, as both of these aspects significantly affect the available reactivity within the reaction.

In the continuous-energy formulation, no cross section processing is required as cross sections are directly sampled from continuous-energy spectra during the transport calculation. In some cases reactor kinetics calculations are also considered a part of the neutronics analysis as they determine important control parameters and predict the dynamic response of the reactor for select scenarios such as design-basis accidents [43].

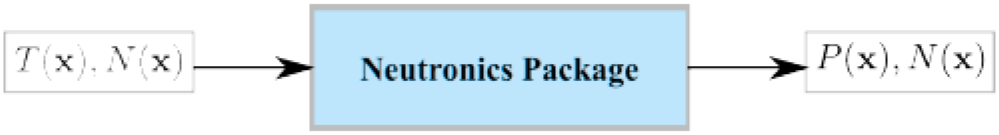

$T(\mathbf{x}), N(\mathbf{x})$
Neutronics Package
$P(\mathbf{x}), N(\mathbf{x})$

Figure 3.4: Input/output for neutronics [57]

Thus the requirements for Monte Carlo neutronics analysis are to be able to significantly decrease tally uncertainties and computational time, deal with the convergence issues associated with LWR problems where the dominance ratios near unity, and overcome the difficulties associated with the massive data storage requirements needed to perform full-core transport and depletion calculations. For example, a Monte Carlo calculation with isotopic depletion analyses for a three-dimensional full-core Pressurized Water Reactor (PWR) would require obtaining local pin powers for approximately 193 assemblies with 264 fuel rods per assembly, and with each pin having 4 radial bins, 4 azimuthal bins, 24 axial bins, and hundreds of energy groups. This results in approximately 4 billion tally cells [58]. This means the number of tally bins would be somewhere on the order of 10^9, which is equivalent to storing GBs of data. Adding in depletion and tracking for approximately 100 different isotopes per tally region brings the memory requirements into the 10s–100s of GBs range solely for storing tally data. This is significantly more memory than the capacity of today's CPUs. Furthermore, considering that modern processor hardware is moving towards GPUs which have a lower amount of available memory per processor [59], having this many tallies per processor is not only impractical, but increasing memory requirements is contrary to the direction the hardware is going.

3.3.1 Parallel Monte Carlo Strategies

In order to efficiently utilize HPC architectures, the parallelization of Monte Carlo algorithms must extend beyond simply dividing the number of particle histories between the processors. Evaluation of parallel algorithms must take into consideration [60]:

1. execution time,
2. memory requirements,
3. communication cost, and

4. implementation cost.

To optimize execution time, one must minimize the time processors spend in an idle state while waiting for information to be sent or received from one or more processors. The objective is therefore to avoid synchronization points and prefer asynchronous communication to synchronous communication wherever possible. To deal with memory constraints one must carefully select what information will be stored globally and passed between processors and what information will be distributed to all processors. Lowering communication costs simply involves minimizing the number of communications performed over the course of a calculation. The physical locations of the processors also plays a role as communication to processors located on the same physical rack is faster than communicating information to processors located on a separate rack. Finally, the implementation cost is something that is specific to the type of application and how robust the programmer wishes to make the implementation. All of these issues ultimately determine the efficiency of the parallel algorithm and how well it scales as the number of processors increases.

The concerns for parallel Monte Carlo with respect to computational speed and algorithmic replication include:

1. having enough independent random number streams to ensure that each history is indeed independent,

2. ensuring that the load is balanced between the processors,

3. the number of global rendezvous points is minimized, and

4. the algorithms scale well with increasing processor counts.

For computations which have both parallel (vector/fast) and serial (scalar/slow) components, the slow components will dominate the execution time and they must be performed sequentially (Amdahl's Law) [61]. The speedup of any given computation can be described by

$$\frac{1}{r_s + \frac{r_p}{n}} \tag{3.23}$$

where r_s and r_p are the fractions of the program that are serial and parallel, respectively, and n is the number of processors.

The concerns regarding computational memory restrictions for parallel Monte Carlo lie in the fact that most computational clusters utilize shared memory, and concurrent components communicate by altering data in shared memory. This typically requires memory locations to be locked and unlocked to prevent multiple processors from attempting to access the memory location simultaneously or contaminating it by changing the data stored at that location before another processor has finished with the data [62]. Using a message passing interface such as the Message Passing Interface (MPI) [63] is generally a much easier and more robust solution. As development of parallel algorithms progresses, the need to develop algorithms which reduce the memory footprint, given the trend of HPC architectures, is becoming more and more evident. The data that consumes the majority of the memory in Monte Carlo calculations are the cross section libraries (several cross sections for 200-400 isotopes, approximately 50 temperature intervals, and 200 000 energy points [44]) and the tally data.

The vast majority of parallel Monte Carlo algorithms are history-based because of the lack of complications in algorithmic development and load balancing. For fixed-source Monte Carlo, only one rendezvous point is necessary at the end of the calculation to gather the results, while for eigenvalue problems a rendezvous point is necessary at the end of every neutron generation in order to preserve normality. This places the restriction on the size of the problem and data because the entire problem must fit on each processor. Obtaining the detailed system-wide solution for large problems with many material regions becomes problematic using domain replication because every processor must allocate storage space for every tally bin in the entire problem. For full-core LWR problems requiring hundreds of millions of tally regions in order to adequately model isotopic depletion and the temperature distribution (for coupling with thermal hydraulics packages), the memory requirements far exceed that of a single processor. Therefore domain replication by itself is not possible regardless of the processor count.

In practice, there are substantial limitations on Monte Carlo performance, primarily due to the size and complexity of a full-core reactor model and the slow convergence of the eigenvalue calculation and tallies [38]. It is only within the past few years that other parallel strategies have been investigated such as domain decomposition and data decomposition. However, as advancements are made in the world of computational simulation, most notably high-performance computing, the radiation transport community is becoming closer to being able to reach the goals for full-core reactor modeling and simulation. Embedded in this research initiative is the ability to utilize the parallel processing capabilities of leadership class computing architectures in an efficient and effective manner.

Domain-Replication

Although Monte Carlo methods are able to provide the detail required for full-core reactor analysis, the time it would take to perform a full-core calculation on a single processor with this level of detail and less than 1% statistical uncertainties would take approximately 5000 hours on a 2.0 Gigahertz (GHz) processor [37]. Furthermore, it would require an enormous amount of memory relative to the amount of memory typically available to a single processor. Fortunately, the speed at which Monte Carlo calculations are executed can be significantly enhanced by utilizing multiple processors.

Monte Carlo simulations are inherently parallel since the transport process is based on generating independent random samples of individual particle histories and the Markov assumption [46] is applied to the Monte Carlo method in general. Parallel Monte Carlo computations are typically performed by replicating the entire problem geometry, tallies, and other global data on all processors. Then the number of histories is divided evenly between the processors and each processor transports one particle at a time. In a fixed-source calculation, only a single rendezvous point is necessary and it takes place at the end of the calculation to compile the results. For eigenvalue problems, synchronization is necessary after every cycle in a criticality calculation in order to preserve normalization. The communication overhead, however, can be made negligible relative to the transport simulation itself if a large enough batch size is used, and this enables parallel, domain-replicated Monte Carlo to scale almost linearly (i.e. the cost of a simulation is reduced approximately by a factor of X when using X processors)[*] [64]. Results have shown that, for full-core Monte Carlo LWR models, the communication costs represent a small fraction of the total run time [13].

Domain-Decomposition

Obtaining detailed, system-wide tally estimates for a large three-dimensional problem such as a full reactor core becomes problematic using domain replication, because every processor must allocate storage space for every tally bin. For full-core LWR problems involving hundreds of millions, or even billions, of material and tally regions due to fuel irradiation, temperature distributions, and the need to couple with multiple physics packages, the memory requirements exceed that of a single processor, and hence domain replication is not possible. In such cases, domain decomposition can be implemented, which involves dividing the spatial domain among a set of processors so that each processor can allocate storage only for the tally cells located on its domain[†] . Although this process does not scale as well as domain-replicated Monte Carlo, the ability to have significantly more material and tally regions than what can be accommodated on a single processor provides an alternative for accurately solving large problems.

Combinations of domain replication and decomposition are also possible to increase the speed and reduce the memory requirements for a given calculation.

[*] In general applications, the number of histories per cycle is typically not large enough to render the cost of communication insignificant, although it is still far less than the cost of the Monte Carlo calculation itself.

[†] This concept can be extended to decompose the material and cross section information. However for continuous-energy reactor core analysis it is likely that each domain will require data for all nuclides in the problem since each domain is likely to contain fuel material.

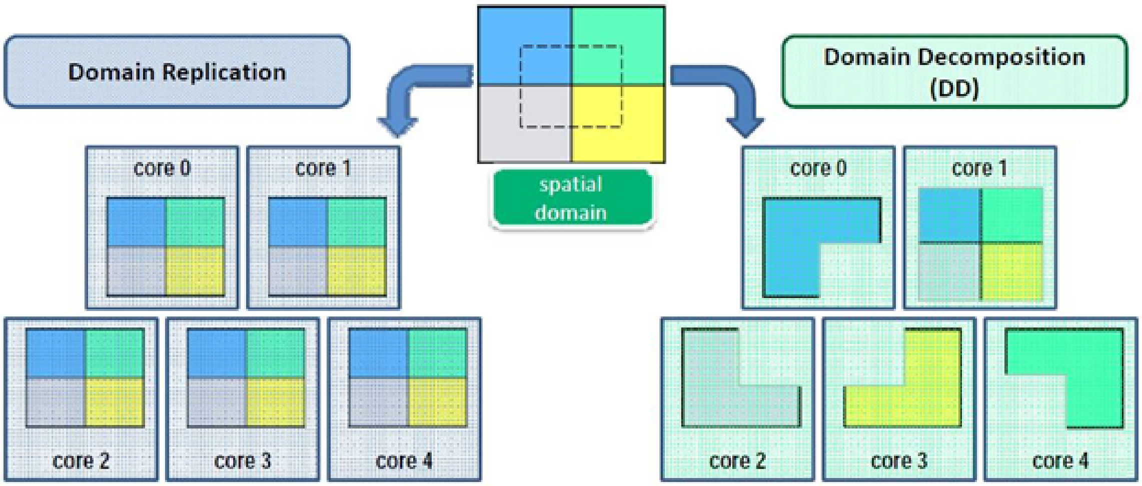


Figure 3.5: Domain-replication versus domain-decomposition.

Figure 3.5 provides an example of domain-replicated and domain-decomposed Monte Carlo. In domain decomposition, each processor core has a subset of the global data. Each processor then simulates particles from the source and particles entering from neighboring domains. For scalability, it is essential to use asynchronous parallel processes and non-blocking communication to minimize the amount of CPU wait time. An algorithm is then implemented to handle the decision making on where to pull the next particle from. Using this strategy there is within-cycle communication, but it is asynchronous so no processor has to wait to send or receive messages during the cycle. A global sync among all processors is still performed at the end of each cycle.

One of the primary disadvantages of domain decomposed Monte Carlo is that the workload may not be evenly distributed over the domains, and the stochastic nature of Monte Carlo transport makes it difficult if not impossible to gain *a priori* knowledge of how the workload will be distributed during a given simulation. Several suggestions on how to gain some idea of the work distribution have been made such as performing a fast preliminary calculation as a check or a mean free path estimation using the scattering cross section data in each domain [65], but these methods will not be able to guarantee an even workload distribution for every conceivable problem geometry. Another disadvantage is that statistical estimates for path-length tallies cannot be performed in the traditional fashion. The issue being that processors need to know when a history has completed in order to calculate the sum of the squared contributions used in calculating the standard deviation. Typically, Monte Carlo codes employ statistical estimation techniques which involve following each particle from the moment it is born until it is "killed". In domain-decomposed environments, this methodology would add a significant amount of idle time among processors and drastically decrease the parallel efficiency of the calculation. So once a particle leaves a domain, that domain cannot finish the history until it knows when the particle has died. This would thus require processors to send signals indicating particle death to the other processors so that they can finalize a given history. The is an example of the kind of synchronous communication that needs to be avoided in order for domain decomposition to be efficient. The only known technique used in cases that suffer from this inconvenience is batch statistics. In some cases, the code will assign a batch identifier to each particle and obtain the statistical uncertainty using the different particle batches [66]. An important aspect of developing scalable and robust domain decomposed algorithms is that asynchronous communication must be used wherever possible, to ensure that the amount of communication work among processors is relatively balanced [67].

3.3.2 Isotopic Depletion

The purpose of performing depletion calculations, from a commercial reactor standpoint, is to monitor how the core composition changes over the life of the reactor. The spatial variation of the isotopic compositions ultimately affects the flux distribution throughout the core, which determines the reactivity increases and decreases observed both in short-term and long-term reactor operation. As the reactor fuel burns up, fissile material is both created and destroyed. Similarly, absorbing materials are also depleted while new high-absorption isotopes are born as fission products, and these fission products poisons build up over time. Therefore to ensure optimal and safe operation of a reactor, frequent adjustment of the control elements must be made in order to maintain criticality. Depletion calculations are also useful for overall fuel cycle analysis and aid in determining optimal fuel reloading strategies to minimize fuel cycle costs. However complete fuel cycle analysis also requires coupling with thermal hydraulics components for time-dependent analysis of the reactor core.

In order to perform a depletion calculation one must first gather the information to solve the Bateman equations for each material composition [68].

$$A_n N_0 \sum_{i=1}^{n} c_i e^{-\lambda_i t} \tag{3.24}$$

Thus for each material region one needs to know the isotopes present and the cross sections for any reaction rates which results in the formation or destruction of any isotope. From Equation 3.24 one can see that depletion calculations introduce a time-dependence when added to a neutron transport calculation. Fortunately, changes in core compositions are relatively slow (i.e. on the order of hours, days, or months), and thus time-dependent neutronics analysis can be performed using a sequence of steady-state transport calculations followed by a depletion calculation. Note however that this treatment is not suitable for nuclear reactor kinetics analysis.

The mathematical formulation for depletion and isotopic tracking involves a system of linear, coupled, first-order Ordinary Differential Equations (ODEs) with constant coefficients of the form [40]

$$\frac{d\bar{N}(t)}{dt} + \bar{\bar{A}}\bar{N}(t) = 0 \tag{3.25}$$

where $\bar{N}(t)$ represents the nuclide number densities and $\bar{\bar{A}}$ represents the transition matrix containing all of the formation, destruction, and decay coefficients as well as the neutron flux. The solution to Equation 3.25 is of the form

$$\bar{N}(t) = \bar{N}(0)e^{-\bar{\bar{A}}t} \tag{3.26}$$

Thus a simple interpretation of Equation 3.25 for a given nuclide is

$$\frac{dN_i}{dt} = \text{FormationRate - DestructionRate - DecayRate} \tag{3.27}$$

In a nuclear reactor environment, formation of new isotopes comes primarily from fission and neutron capture. Likewise, destruction of isotopes comes from fission and neutron absorption. The decay rate simply involves tracking a nuclide until it decays to a stable isotope, therefore encompassing destruction and formation simultaneously. A more in-depth equation for modeling depletion and isotopic tracking is provided in Equation 3.28 [42]

$$\frac{dn_j}{dt} = \sum_i \gamma_j \Sigma_f \phi + (\lambda^{ij} + \sigma^{ij}\phi)n_i - (\lambda^j + \sigma_a^j\phi)n_j \tag{3.28}$$

where

n_i	atom density of nuclide i
γ_j	fraction of fission events that produces fission product species j
λ^{ij}	radioactive disintegration rate of isotope i into isotope j
σ^{ij}	transmutation cross section for the production of isotope j by neutron capture in isotope j
ϕ	neutron flux

The individual terms in Equation 3.28 represent

$\gamma_j \Sigma_f \phi$	the production of n_j from fission events of n_i
$\lambda^{ij} n_i$	the production of n_j from the radioactive decay of n_i
$\sigma^{ij} \phi n_j$	the production of n_j from neutron absorption reactions of n_i
$\lambda^{j} n_j$	the destruction of n_j from radioactive decay
$\sigma_a^{j} \phi n_j$	the destruction of n_j from neutron absorption reactions

Note that the neutron flux is assumed to remain constant over the time interval for which the equation is solved.

$$\phi(t) = \phi_0 | t_i \leq t \leq t_{i+1} \tag{3.29}$$

This is often referred to as the constant flux approximation. The alternative approximation is termed the constant power approximation and assumes

$$P(t) = w_a N(t) \sigma_a \phi(t) = P_0 \mid t_i \leq t \leq t_{i+1} \tag{3.30}$$

where w_a is the energy released per absorption reaction. These two approximations behave quite differently over long periods of time, however they are typically in good agreement over shorter time steps [40]. The accuracy of the solution (in general) is therefore dependent upon the length of the chosen time interval, such that

$$(\lambda^i + \sigma_a^i \phi) \Delta t \ll 1 \tag{3.31}$$

where

$$\Delta t = t_{i+1} - t_i \tag{3.32}$$

for all isotopes [42].

Chapter 4

Computational Codes and Development

The research and development presented in this manuscript involves the use of several computer codes. This chapter provides a brief description of each computer code used to perform the research, development, and benchmarking analyses presented in the chapters to follow.

4.1 MCNP

The Monte Carlo N-Particle transport code (MCNP) is a general-purpose, three-dimensional, continuous-energy Monte Carlo code developed at LANL [2]. MCNP is capable of coupled neutron, photon, and electron transport, and can be used in a variety of applications including radiation protection and dosimetry, nuclear oil well logging, medical physics, criticality safety, fission and fusion reactor design, and decontamination/decommissioning.

MCNP uses point-wise continuous-energy cross-section data from the ENDF [31]. This enables MCNP to account for any reaction type in which data is available. This data is used to generate probability distributions, which are sampled using the

Monte Carlo package within MCNP to solve the transport equation for user-specified parameters. The generalized-geometry implementation represents three-dimensional configurations as geometric cells that are bounded by first- and second-degree surfaces and forth-degree elliptical tori. The flexibility in MCNP's capability to represent a wide variety of source and tally specifications alongside the multitude of variance reduction techniques available to improve calculation efficiency makes MCNP an extremely versatile and powerful tool.

4.2 SCALE

The SCALE nuclear analysis system is a multi-purpose computational suite of codes developed at ORNL and co-sponsored by the Nuclear Regulatory Commission (NRC) and the DOE. SCALE has applications in criticality safety, reactor physics, radiation shielding, spent nuclear fuel and high-level waste characterization, and sensitivity/uncertainty analyses [30]. The modular code system uses automated sequences to perform cross section processing, radiation transport (using either the deterministic or Monte Carlo approach), isotopic depletion, transmutation and decay, and various other reactor analysis operations. SCALE also features an easy-to-use graphical user interface and a multitude of options to aid in model development and visualization of results.

The majority of sequences within SCALE can be executed using SCALE's functional driver module to couple the relevant packages associated with a particular sequence. Alternatively, most packages can be executed as stand-alone calculations as well. The sequences used in this research include KENO, ORIGEN, ARP, AmpxLib, and SCEMPP. The following subsections provide a brief overview of each of these sequences.

4.2.1 KENO

The KENO sequence is a Monte Carlo criticality program developed for use specifically within the SCALE system [3]. KENO-VI is an extension of the KENO-V.a module which uses the SCALE Generalized Geometry Package (SGGP) for increased flexibility in the code's ability to represent a wider array of geometric shapes. The SGGP implemented in KENO-VI is capable of modeling any shape that can be constructed by a quadratic equation, and although the SGGP increases the versatility of the geometric representations available, it does significantly increase the run time compared to KENO-V.a.

The primary function of KENO is to estimate k_{eff} for a three-dimensional system. Unlike MCNP, this Monte Carlo module focuses on neutron transport and is therefore restricted to nuclear reactor and criticality safety applications. KENO can be operated using either continuous energy or multi-group cross section treatment. By being a part of the SCALE code suite, KENO has access to SCALE's Standard Composition Library [69] which provides a flexible and convenient means of generating standardized compositions as well as custom material compositions, compounds, alloys, and mixtures. KENO offers several initial source specification options, however the default of sampling using the fission cross section within each region is typically adequate for criticality safety application. Although KENO does not offer tally specification options, several flags are available to print out relevant information such as energy- and spatially-dependent fluxes, reaction rates, and leakage rates as well as information on the mean neutron lifetime, generation time, average fission energy, mean-free-path data and cross section information for multigroup calculations.

4.2.2 ORIGEN

The Oak Ridge Isotope GENeration (ORIGEN) code is a general-purpose point-depletion and decay code for calculating the isotopic, decay heat, and radiation source terms for a given compositional model [4]. ORIGEN is one of the most robust and well-validated depletion codes available. ORIGEN calculates time-dependent isotopic concentrations, activities, and radiation source terms for the complete list of nuclides produced via irradiation, transmutation, activation, fission, and decay in nuclear reactor environments. The nuclear data libraries within ORIGEN contain nuclear data for a total of 2226 nuclides and the decay data is currently based on ENDF/B-VII.0 [31] cross sections. The version of ORIGEN that is maintained and supported by ORNL is the version that is integrated into the SCALE nuclear analysis system.

ORIGEN calculates the time-dependent rate of change of isotopic concentrations, activities, and radiation source terms for the complete list of nuclides produced in nuclear reactor environments [4]. ORIGEN uses several mathematical methods and numerical approximations to solve the depletion equation

$$\begin{aligned}\frac{dN_j}{dt} &= \sum_k \underbrace{\left(\gamma_k \bar{\sigma}_{\mathrm{f},k} \bar{\phi} N_k + \bar{\sigma}_{\mathrm{c},k} \bar{\phi} N_k + \lambda_k N_k\right)}_{\text{production}} - \underbrace{\left(\bar{\sigma}_{\mathrm{f},j} \bar{\phi} N_j + \bar{\sigma}_{\mathrm{c},j} \bar{\phi} N_j\right)}_{\text{destruction}} - \underbrace{\lambda_j N_j}_{\text{decay}} \\ &= \mathbf{A} N\end{aligned} \tag{4.1}$$

The individual terms in Equation 4.1 represent

$\gamma_k \bar{\sigma}_{\mathrm{f},k} \bar{\phi} N_k$	production of N_j from fission of nuclide N_k
$\bar{\sigma}_{\mathrm{c},k} \bar{\phi} N_k$	transmutation into N_j resulting from neutron capture by nuclide N_k
$\lambda_k N_k$	production of N_j resulting from radioactive decay of nuclide N_k
$\bar{\sigma}_{\mathrm{f},j} \bar{\phi} N_j$	destruction of N_j from fission
$\bar{\sigma}_{\mathrm{c},j} \bar{\phi} N_j$	destruction of N_j from all reactions other than fission
$\lambda_j N_j$	radioactive decay of N_j

When operating in stand-alone mode, ORIGEN simply requires specification of the average specific power or flux, the isotopic concentrations of all nuclides in the material, and the length of time the material will be depleted. In addition, the distributed library file containing the problem-independent multigroup cross sections by isotope which most closely matches the specific reactor assembly

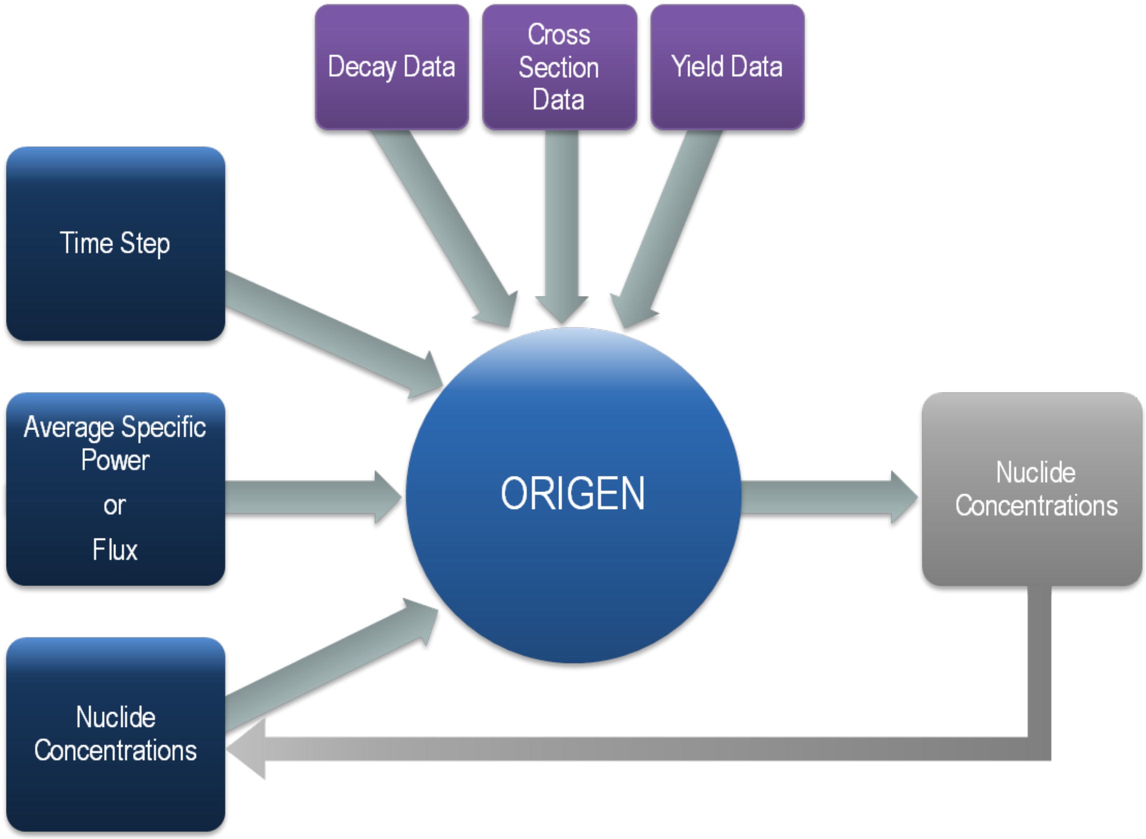


Figure 4.1: Flow diagram for a stand-alone ORIGEN calculation.

configuration. ORIGEN then collects the decay and yield data along with the JEFF cross section data to construct the transition matrix, which solves the depletion equation (Equation 3.25) in matrix form to produce depleted nuclide concentrations. The general flow of data for a standalone ORIGEN calculation is provided in Figure 4.1

When ORIGEN is coupled to a transport calculation, the process is slightly more complex. The decay data, yield data, and problem-independent multigroup cross sections by isotope are combined to form a problem-independent 1-group ORIGEN library. Then the problem-dependent microscopic cross section data for each isotope (ENDF) is obtained directly from either the cross section processing module while the problem-dependent fluxes are obtained from the transport solver. These values are

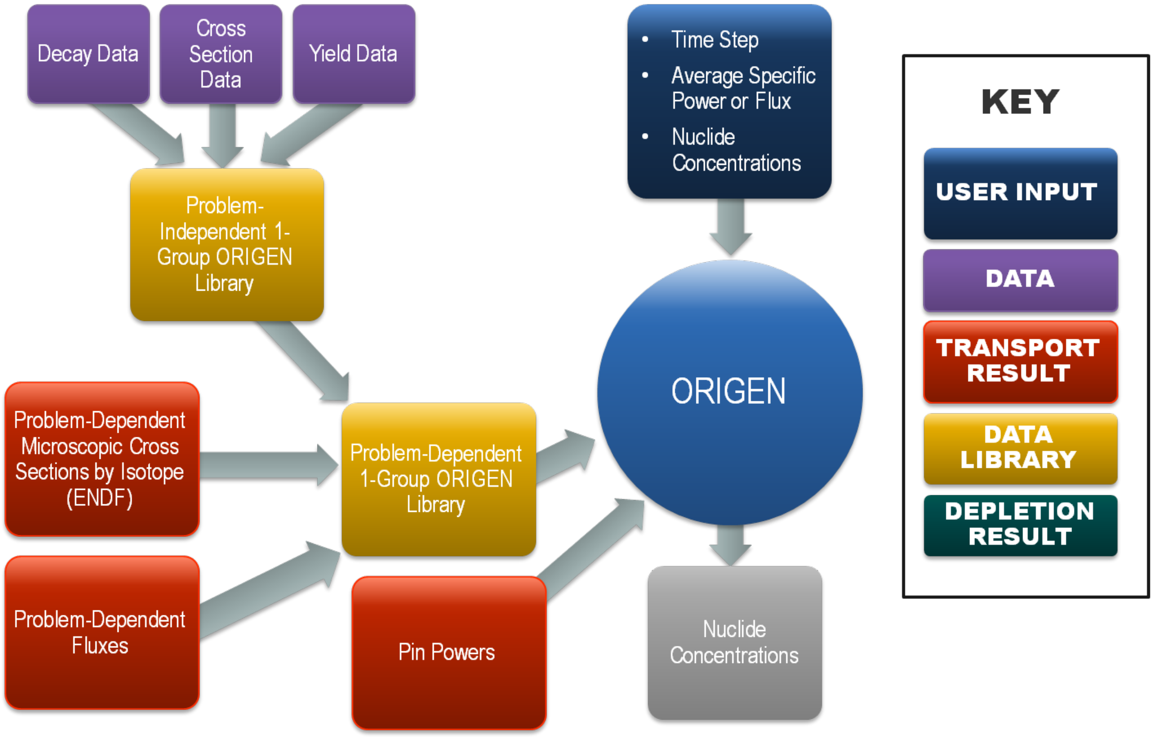


Figure 4.2: Flow diagram for a transport-coupled ORIGEN depletion calculation.

combined to create the problem-dependent 1-group ORIGEN library. The problem-independent and problem-dependent libraries are then combined to create a single binary library which is used to populate the transition matrix. The transport calculation also provides ORIGEN with localized pin powers for each material. The same user input used in a stand-alone ORIGEN calculation (time step size, average specific power or flux, and nuclide concentrations) is required for transport-coupled ORIGEN calculations, and all of these elements come together to provide the data necessary to solve the depletion equation for new nuclide concentrations. A flow diagram for a how ORIGEN works when coupled to a neutron transport calculation is provided in Figure 4.2 .

4.2.3 ARP

The Automated Rapid Processing module in SCALE provides the ability to generate problem-dependent ORIGEN cross section libraries for ORIGEN depletion calculations [70]. The primary purpose of the ARP module is to provide the necessary cross section and decay data which ORIGEN needs perform point depletion and decay calculations for nuclear fuel and other radioactive material. It can also be used in sequence with ORIGEN to provide interpolated libraries for a stand-alone ORIGEN calculation. The ARP module utilizes an interpolation scheme that interpolates between the pre-generated ORIGEN libraries distributed with SCALE. The interpolation parameters include fuel enrichment, burnup, and other operating conditions for a variety of different Uranium Dioxide (UO_2) and Mixed Oxide (MOX) reactor assembly configurations. The list of assembly configurations made available by the pre-generated libraries includes:

BWR 7×7, 8×8–4, 8×8–1, 9×9–8, 9×9–9, 10×10–9, 10×10–8, SVEA–64 and SVEA–100

PWR 14×14, 15×15, 16×16, 17×17

CANDU 28– and 37–element bundle designs

Magnox graphite reaction

Advanced Gas-Cooled Reactor (AGR)

VVER 440 and 1000

RBMK

MOX BWR 8×8–2, 9×9–1, 9×9–9, 10×10–9

MOX PWR 14×14, 15×15, 16×16, 17×17, 18×18

Refer to Reference [70] for detailed descriptions of the different assembly configurations and to view the available interpolation parameters for each configuration.

4.2.4 TRITON

The Transport Rigor Implemented with Time-dependent Operation for Neutronic depletion (TRITON) code is a multipurpose control module within SCALE for performing coupled transport, depletion, and sensitivity and uncertainty analyses [28]. TRITON is capable of performing transport calculations in one and two dimensions using the XSDRNPM [71] and NEWT [72] transport codes. Alternatively, TRITON can also utilize the KENO Monte Carlo code for three-dimensional calculations. Coupled transport-depletion calculations use one of these neutron transport codes in tandem with the ORIGEN depletion module. In version 6.1 of the SCALE code suite, the TSUNAMI-2D sequence was integrated into TRITON to provide automated cross section processing, calculations of sensitivity coefficients, and the uncertainty in k_{eff} and other responses due to the cross section covariance data.

4.2.5 AmpxLib

The AmpxLib package [73] is a resource package within SCALE that processes ENDF cross section data to generate multigroup cross section libraries that can be used by any of the multigroup transport modules within SCALE [74]. Before the AMPX cross section processing system [75] was developed, the only means of processing ENDF data was the NJOY code system [76]. Following the development of the AMPX package, SCALE adopted the AMPX system as the means to create, pass, and store multigroup cross section information.

4.2.6 SCEMPP

Continuous-energy cross section capabilities have recently been added to SCALE 6 for performing Monte Carlo calculations using the KENO sequence. This capability has been extended to apply to sensitivity and depletion analyses. A modular API was also created, the SCALE Continuous-Energy Modular Physics Package API (SCEMPP), to provide modular access via C++ and Fortran APIs, to SCALE's continuous-energy cross section data [74]. SCEMPP provides a collision physics engine for performing neutron and photon transport in a material. Unfortunately, this feature is still in the development/testing phase, and thus there are no additional references or documentation to expand this discussion further.

4.3 Exnihilo

The Exnihilo* transport suite is a collection of packages developed for massively-parallel deterministic, Monte Carlo, and coupled neutronics applications. The package architecture within Exnihilo is designed such that each of its individual packages provides well-defined capabilities; thus following the object-oriented programming paradigm. The package model for Exnihilo is illustrated in Figure 4.3 [78]. The major components of the Exnihilo code suite include:

Nemesis general infrastructure, design-by-contract (DBC) [79], parallel communication (MPI) resources, and testing harnesses

Transcore general components that are used for building particle transport and multiphysics applications (databases, quadratures, fields, etc)

Denovo deterministic transport package for fixed-source and eigenvalue problems for regular-grid S_N and SP_N calculations as well as 2D MOC calculations on combinatorial geometries

Shift hybrid or stand-alone Monte Carlo framework for fixed-source and eigenvalue calculations

Insilico reaction analysis packages for performing neutronics analysis by coupling cross-section processing, isotopic depletion, and/or thermal-hydraulics capabilities with one of Exnihilo's particle transport codes (i.e. Denovo or Shift)

*a Latin phrase meaning "out of nothing" [77]

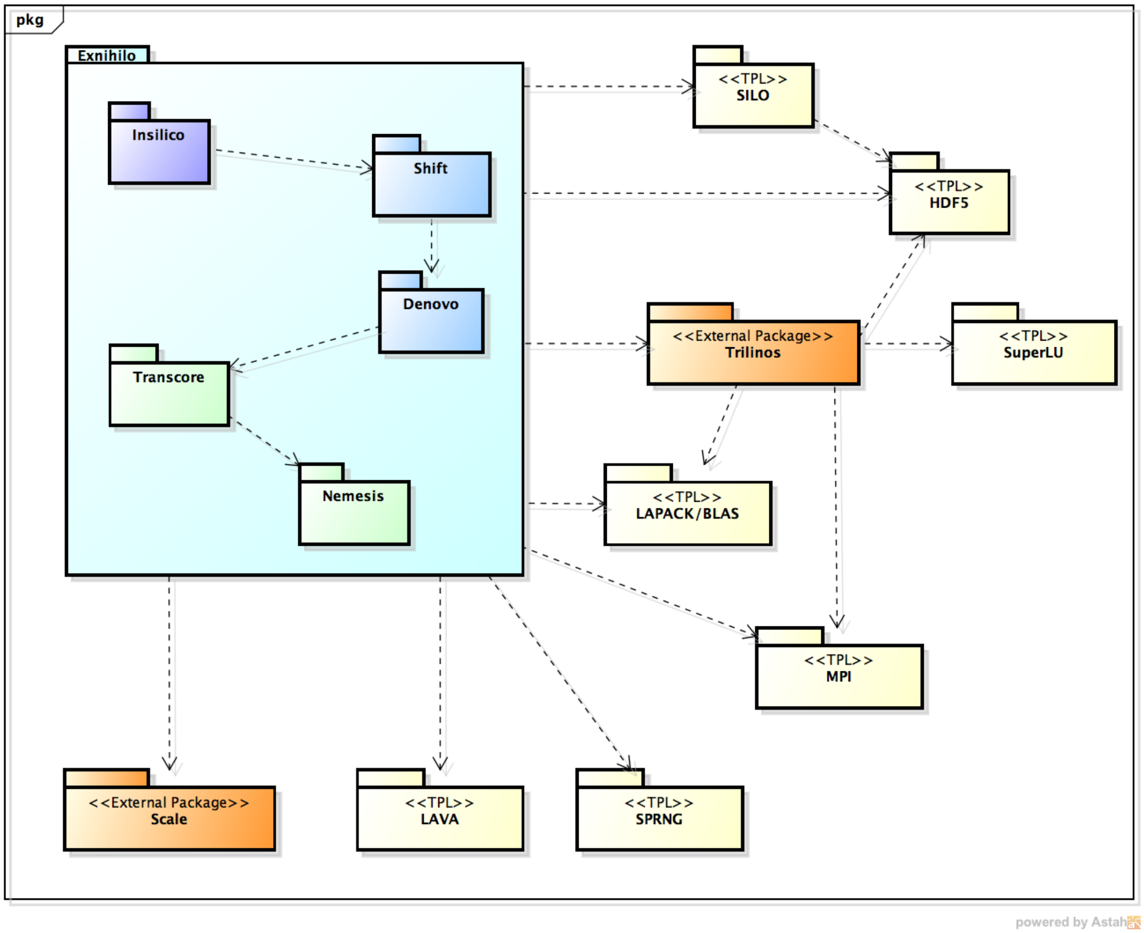


Figure 4.3: Exnihilo package diagram.

The primary transport components, Denovo and Shift, as well as the neutronics package, Insilico, will be discussed in the following sections.

4.3.1 Denovo

The baseline code within Exnihilo is Denovo–a massively parallel deterministic radiation transport code enabling solutions to enormous nuclear energy applications. The modular design of the Denovo package allows easy integration of additional codes/features in order to take advantage of the parallel capabilities, third-party library interfaces, and a powerful Python front-end. Denovo was developed for two reasons:

- to function as the deterministic solver for SCALE's MAVRIC sequence [80], and
- to replace TORT [81] as the chief three-dimensional deterministic radiation transport code at ORNL.

This transport code utilizes state-of-the-art transport methods and nonstationary Krylov methods to solve the within-group equations for fixed-source and eigenvalue problems on massively parallel systems. Denovo is equipped with multiple spatial differencing schemes, modern high-performance solvers, and a Python front-end which makes Denovo an extremely powerful tool that can work efficiently and effectively on high-performance platforms with hundreds of thousands of processors [1].

4.3.2 Shift

The Shift computational package is a new hybrid (deterministic/Monte Carlo) radiation transport framework under development at ORNL. This computational package enhances the radiation transport capabilities within Exnihilo to be applicable to large, complex real-world problems such as reactor simulation, radiation analyses from a nuclear detonation, facility safety and safeguards, and fusion applications.

As a hybrid package, Shift requires both a deterministic and a Monte Carlo package, and while a deterministic package is already available within Exnihilo (i.e. Denovo), a Monte Carlo transport package (source specification, tally specification, and particle tracking) was developed as an exclusive component of Shift. The advantage of Exnihilo developing its own Monte Carlo transport package is that it enables smooth communication between the deterministic and Monte Carlo codes while simultaneously granting Shift access to the pre-existing parallel mechanics and algorithms previously developed for the Denovo framework. Since Shift is initially being developed exclusively for nuclear reactor analysis in HPC applications, it was developed to use domain decomposed and/or domain replicated parallel solution techniques [82]. This implementation is designed to help with both the computational time and memory restrictions that plague large Monte Carlo problems. The overall design of Shift is aimed towards providing detailed system-wide solutions for fission reactor eigenvalue problems on massively parallel systems such as the Cray XK7 (Titan) machine at the National Center for Computational Sciences (NCCS). [5].

In addition to a transport algorithm, a Monte Carlo code requires a geometry package to define the model and a physics package to link cross section and material information to the problem space. Since there are already multiple powerful and well-validated geometry and physics packages available, Shift is designed such that it allows any physics or geometry package to be connected to the code. This categorizes Shift as more of a Monte Carlo *framework* than a Monte Caro code. The geometry packages currently implemented within Shift include KGTLIB (see Section 4.3.4), LAVA (see Section 4.3.4), and an Reactor Tool Kit (RTK) package that is primarily used for internal testing and simple pin cell and array reactor models. The physics packages currently implemented include the SCEMPP continuous-energy package and an internal multigroup physics package based on AMPX working-formatted libraries. The LAVA continuous-energy physics package is currently under development. One of the primary advantages of the Shift framework is that it is both geometry and physics agnostic, meaning that the particle transport processes are the same regardless of which geometry or physics package is being used.

At its current stage of development, Shift's Monte Carlo transport routine has the capacity to perform parallel eigenvalue and fixed-source transport problems using a Python input file or a text-based input file for HPC applications. Shift's parallel capabilities support multiple parallel decompositions including full domain replication, full domain decomposition, domain decomposition with overlapping domains, and Multiple-Set-Overlapping Decomposition (MSOD) decomposition [65]. The variety of parallel capabilities offered enables Shift to be applicable to a wide variety of problems ranging from simple pin cell problems to complex full-core geometries.

4.3.3 Insilico

The Insilico package is Exnihilo's neutronics package which couples multiple core analysis tools for a more in-depth analysis of a nuclear reactor core. Neutronics analysis in the quasi-static case, where the neutron density and/or power is assumed to be constant over a given time step, can be divided into three steps:

Cross Section Processing Processing of cross section data for all compositions present in the model (multigroup formulation only).

Neutron Transport Solution of the transport equation (Equation 3.1) to determine the space-energy distribution of the neutron flux.

Isotopic Depletion Solution of the Bateman equations (Equation 3.28 for each depletable material using the region-averaged average power or flux for a given time interval [83].

Cross section processing within Insilico is performed through an interface to SCALE's cross section processing modules [84]. Insilico also has a depletion package which integrates ORIGEN [4] depletion capabilities into the Exnihilo framework and a Python front-end capable of performing standalone and coupled transport-depletion calculations. Lastly, ongoing development efforts within Insilico include building an automatic coupling between Shift and the COMSOL thermohydraulics package [85] to streamline the transfer of fission heat generation rates into COMSOL-based Computational Fluid Dynamics (CFD) and thermomechanics models. The depletion package, Python front-end, and coupling of the depletion package with the Shift Monte Carlo package make up a large part of the research and development efforts described in this manuscript.

4.3.4 Third Party Libraries (TPLs)

Exnihilo uses a variety of third-party libraries. However, many of these libraries are not relevant to this research. The following sub-sections very briefly discuss the few that are.

KGTLIB

The KENO Geometry Tracking Library provides a C to Fortran API which provides C/C++ access to the tracking functionality used by the three-dimensional Monte Carlo transport modules KENO and Monaco in SCALE [86]. KGTLIB was developed at ORNL for use within the Exnihilo code suite; however it can also be used by other C/C++ radiation transport applications. A vital feature of KGTLIB lies in its ability to be used in parallel C/C++ applications with limited I/O operations.

KGTLIB works by first processing a standard SCALE input file to produce a binary data file that contains all of the geometric information. Then the library grants external access to the geometry information in the binary file so that it can be read into memory and used by a C/C++ radiation transport application. It is beneficial in this case to separate the input processing of the SCALE input file from the geometry tracking algorithms native to KENO to reduce the size and complexity of KGTLIB and to allow KGTLIB to be used in parallel since having multiple processes reading a file from disk is both inefficient and error-prone.

LAVA

LAVA is a software library that provides access to some of the functionality within the MCNP Monte Carlo transport code via a modular API [87]. Currently the LAVA API enables access to the combinatorial geometry, source definition, and collision physics used by MCNP. The combinatoriesl geometry information provided by the LAVA API includes material and cell identifiers as well as particle tracking (distance to boundary/surface/cell). Access to the source definition enables full support for the SDEF card in MCNP input files. Finally, the collision physics provides access to MCNP's continuous-energy neutron physics and multigroup neutron cross section generation capabilities.

Similar to KGTLIB, LAVA consists of a C to Fortran API that is built alongside a set of lightly-patched MCNP Fortran source files, and is accessible by C and C++ applications.

4.4 Coding Standards

The Exnihilo code suite follows the object-oriented paradigm and is developed under a strict set of coding requirements for all code developed within the Exnihilo framework. Exnihilo is primarily written in C++ with a few Fortran/C kernels and Python and Extensible Markup Language (XML)-based front-end executables. The research and development of computational code and algorithms within Exnihilo is a large part of the research presented in this manuscript. Although this aspect of research and development is technical in nature, it is deemed appropriate to discuss the coding standards upheld during the code development processes. The coding requirement for Exnihilo code development include in brief:

- The file structure for C++ classes requires a header file to define the class interface, an implementation file to define the member functions, a template instantiation file for templated classes with templated member functions, and an implementation file for inlined member functions [88].

- Class definitions must make all data members private. This ensures that the underlying data can only be modified by member functions which operate as the class is designed to operate. Encapsulation of private data members also simplifies the interface as clients will not have to decipher all of the requirements and uses for the data members [89].

- All code must be thoroughly commented and documented using Doxygen [90]. This include complete descriptions of each object (class) and every function is provides as well as one-line comments for each line of code that performs a non-trivial operation. A comment to coding ratio of at least 0.3 is desirable for all code developed within Exnihilo.

- Constant-correctness must be enforced whenever possible.

- All code must utilize the DBC paradigm [79].

- Any functions or classes used within Exnihilo but developed externally must be encapsulated to ensure that all coding standards are upheld.

- All classes must be accompanied with a unit test which tests the functionality of each individual function which is made publically accessible.

- New code can only be committed to the master repository if all unit tests pass.

As an aside to these coding standards, special attention was taken for code developed as part of these research and development efforts to optimize data storage due to the intense computational memory requirements of performing coupled transport-depletion calculations for large geometries with thousands or tens of thousands of material regions.

Chapter 5

Analysis of Monte Carlo Uncertainty Estimates

In order to gain an improved understanding of some of the current issues associated with Monte Carlo uncertainty analyses, the pin cell benchmark problems introduced in Section A.1 are used to examine the underprediction in Monte Carlo tally statistics. The primary factors evaluated in connection with underprediction of Monte Carlo uncertainties are source convergence, source sampling, and cycle-to-cycle correlations, and the majority of the ill effects introduces by these factors can be controlled and mitigated via adequate specification of Monte Carlo parameters such as the number of histories per cycle and the number of initial cycles skipped. Each of these issues are addressed in an effort to examine the magnitude and behavior of the underprediction in the uncertainty and to distinguish between the effects of poor sampling, an ill-converged fission source distribution, and the cycle-to-cycle correlations in the fission source. This analysis uses the KENO-VI and MCNP continuous-energy Monte Carlo codes to provide a complete analysis using two independent codes. In addition, differences between the two codes and their impact on the observed underprediction is also presented. The underlying theory which characterized this analysis is presented in Section 5.1 and the results of the analyses performed are presented in Section 5.2 [7].

5.1 Theory and Methodology

To accurately examine and understand situations that lead to underprediction of localized tally uncertainties, some of the fundamental steps associated with "good Monte Carlo practice" must be addressed. The three most general steps include:

1. setting up a proper initial source particle distribution,
2. choosing a sufficient number of histories per cycle, and
3. selecting a sufficient number of initial cycles to skip, prior to accumulating tally information, in order to ensure proper convergence of the fission source distribution.

Each of these issues are herein addressed independently in an effort to examine the magnitude and behavior of the underprediction in the uncertainty and to distinguish between the effects of poor sampling, an ill-converged fission source distribution, and the cycle-to-cycle correlations in the fission source. The analyses use two of the most well-walidated Monte Carlo codes, the continuous-energy version of SCALE/KENO-VI [3] and MCNP5 [2], to determine the characteristics of uncertainty underprediction with respect to the previously mentioned issues.

Examination of the various issues associated with source sampling and convergence that may impact tally uncertainty estimates involves first examining the 3 primary Monte Carlo input parameters:

1. the number of cycles or neutron generations,
2. the number of histories per cycle, and
3. the number of initial cycles skipped.

These analyses involved performing a plethora of simulations of the same case where only one of the control parameters was varied to identify their correlation between underpredictions in tally uncertainties. To identify the amount of underprediction, each case was also run with the same Monte Carlo parameters but using 50 independent random number seeds. Then multiple 50-case sets were simulated where only one of the control parameters was varied so that the correlation between the parameter and the underprediction of tally uncertainties could be identified. This allowed for accurate estimation of the "real" variance and provided a metric which could be constructed to compare the underprediction in the tally uncertainties from multiple different sets.

Some of the underlying theory necessary to carry out these analyses is presented in Section 3.2 and should be kept in mind when performing Monte Carlo eigenvalue calculations, especially in cases where reaction rate tallies are desired. Likewise, the differences between the two Monte Carlo codes herein employed with respect to how they manage (or try to address) each of the above issues and how they may impact localized tally results are discussed in Section 5.1.1.

5.1.1 Comparison of MCNP and KENO-VI

In order to investigate the issue of uncertainty underprediction in modern Monte Carlo calculations both MCNP and KENO-VIare used in these analyses. To make sure that relevant comparisons are made between the two codes, it is important to understand the differences in the "random walk" process in each code as well as the way each code calculates neutron flux tallies and uncertainties. In general, the only major difference between the two codes is how they differ in the weight treatment of particles as they are transported. However, at the point in which the tally results are output, this difference is not relevant as both codes output tallies based on a single source particle. Thus the final results can be compared without the need for any post-processing or normalization.

When computing flux tallies, KENO and MCNP both use a track-length estimator (ξ) to estimate the physical average flux density:

$$\phi_V = \frac{1}{V}\int dE \int dt \int dV \int d\Omega \psi(\vec{\mathbf{r}}, \hat{\mathbf{\Omega}}, E, t) \tag{5.1}$$

where ψ is the angular neutron flux density. For a given history and tally region (assuming no energy or time bins), the track-length estimator is calculated as

$$\xi = \frac{1}{V}\sum_i w_i l_i \tag{5.2}$$

where

V the volume of the tally region

w_i the weight of the particle

l_i the length of the track traveled by the particle

i the history's track index (since a history may generate multiple tracks through an individual tally region)

Estimation of the source distribution in both codes also differs slightly, but as previously noted, the differences lie in the normalization of the particle weight. At each collision where fission is possible, an integer number of fission source particles for the next generation is calculated according to

$$n = INT(W_s + \rho) \tag{5.3}$$

where

$$W_s = \frac{\nu \Sigma_f}{k \cdot \Sigma_t} W \tag{5.4}$$

and ρ is a pseudo-random number between zero and one. The key difference between the two codes is that KENO uses a reduced value for the constant k and gives each source particle a weight of one, whereas MCNP uses a running average of the k_{eff} cycle values and assigns each particle a weight of $\frac{W_s}{n}$. The reduced value for the constant k

is a result of KENO normalizing so that the number of fission sites remains constant while MCNP normalizes so that a constant total weight is preserved [3].

There are significant differences between the standard deviation estimators for the eigenvalue used by each code. KENO recently implemented an iterative approach in version 6.1 that estimates the real variance by taking into account lag covariance data between generations [91]. Thus the difference between the standard deviation estimators may lead to an observable difference in the k_{eff} results.

5.1.2 Uncertainty Underprediction

To quantify the error associated with statistical estimates introduced by accumulating tally data using an ill-converged fission source, inadequately sampling the fission source regions, or cycle-to-cycle correlations, an uncertainty underprediction metric is utilized. This metric is simply the ratio of the real standard deviation to the apparent standard deviation [58]. The real standard deviation is the sample standard deviation calculated from the tally mean values for a set of N cases, denoted as $\hat{\sigma}_r$. The formula is provided in Equation 5.5 where x_i is the reported tally value for each simulation and $\bar{x}$ is the mean of the tally values from all simulations.

$$\hat{\sigma}_r = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{5.5}$$

The apparent standard deviation is the statistical estimate output by the Monte Carlo code. In order to use all of the information available to pinpoint what the underprediction is, an average of the output standard deviations is used as shown in Equation 5.6.

$$\bar{\sigma}_a = \frac{1}{N}\sum_{i=1}^{N}\sigma_i \tag{5.6}$$

The ratio of these two values is denoted as f_σ and is defined in Equation 5.7. This ratio in effect identifies the factor by which the Monte Carlo calculation is underpredicting the standard deviation.

$$f_\sigma = \frac{\hat{\sigma}_r}{\bar{\sigma}_a} \tag{5.7}$$

Using this metric requires each simulation in a given set of N cases to be identical with the exception of having a different initial random number seed. By starting with a set of cases with poorly chosen Monte Carlo parameters, one can observe differences in f_σ as proper source convergence is attained and the fissionable regions are determined to be adequately sampled. Then it may be possible to attribute any additional underprediction in the uncertainty to the inter-cycle correlations in the fission source distribution.

5.2 KENO/MCNP Results

The three pin cell test cases (cases 2.1, 2.2, and 2.3 from Chapter 3 of Reference [92]) are used to identify the magnitude and behavior of the underprediction in the uncertainty with respect to one or more problem characteristics or Monte Carlo parameters. The computations were performed with SCALE 6.0/KENO-VI version 6.0.24 and MCNP version 1.51, both using ENDF/B-VII continuous-energy cross-section libraries on a Linux computational cluster. To identify the underprediction, each benchmark model was run as a series of 50 independent simulations using different initial random number seeds to obtain an unbiased estimate of the statistical errors for equal-volume flux cell tallies that span the axial length of the fuel. The fluxes were tallied on an eight-group energy structure, which is provided in Table A.1 in Appendix A.1. In performing the analyses presented in this Section, it was observed that the behavior was similar for all energy groups. Therefore, all figures that plot values derived from the flux represent the flux in the highest energy group in this eight-group structure.

Figure 5.1 provides baseline axial profiles of the total flux for the three pin cell benchmark cases. Case 2.1 features a symmetric axial burnup distribution, which results in a symmetric flux profile. Since the low-flux region in case 2.1 is in the center of the pin cell and because the flux there is significantly lower than the flux near the ends of the pin cell, the two ends have difficulty exchanging particles, and case 2.1 may therefore be adversely affected by cycle-to-cycle correlations in the fission source. Case 2.2 features a nearly monotonic flux profile due to the asymmetric burnup distribution. The exceptionally low flux near the bottom of the pin cell may require special attention to ensure it is properly sampled relative to the rest of the model in order to produce accurate flux tally results. Case 2.3 is notably similar to case 2.2; however, in case 2.3 the regions where the burnup distribution is asymmetric are much closer to the bottom of the pin cell, leading to a slightly higher flux in the lower regions. Although the flux is higher in case 2.3 than in case 2.2, resolving the flux shape in the lower regions may be more difficult if the pin cell is not sampled properly.

The underprediction in the uncertainty for k_{eff} for all cases examined in this analysis was not greater than 50%. A decrease in the underprediction was observed for cases that simulated more histories per cycle as well as cases where more initial cycles were skipped (i.e., cases exhibiting better source convergence). Using optimal Monte Carlo parameters, the underprediction was 10% or less. Because the underprediction in the uncertainty of the eigenvalue was much lower than that of the tallies, it was not examined in depth in these analyses. Sections 5.2.1 through 5.2.4 contain various analyses of different Monte Carlo parameters to study the magnitude and behavior of the underprediction in localized tally values.

5.2.1 Histories Per Cycle

Case 2.2 features an asymmetrical axial burnup distribution where the flux is higher near the top of the pin cell. It has been asserted that increasing the number of cycles or the number of histories per cycle will not reduce the effects of cycle-to-cycle correlations in the fission source [9]. However, by varying the number of

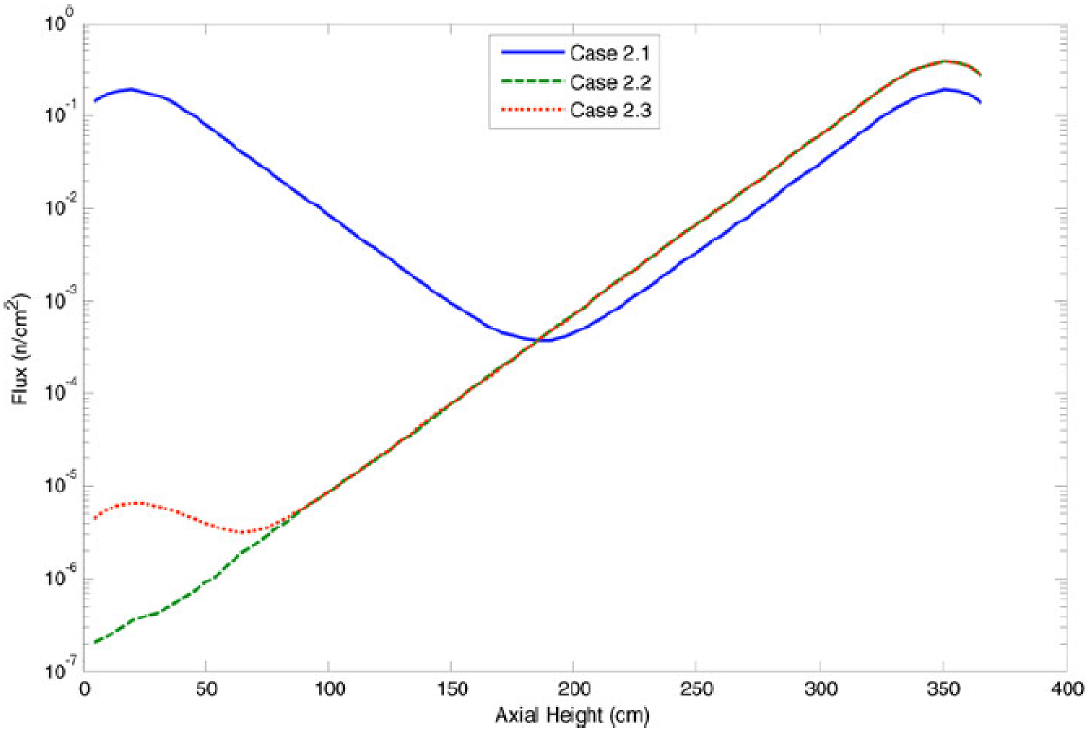
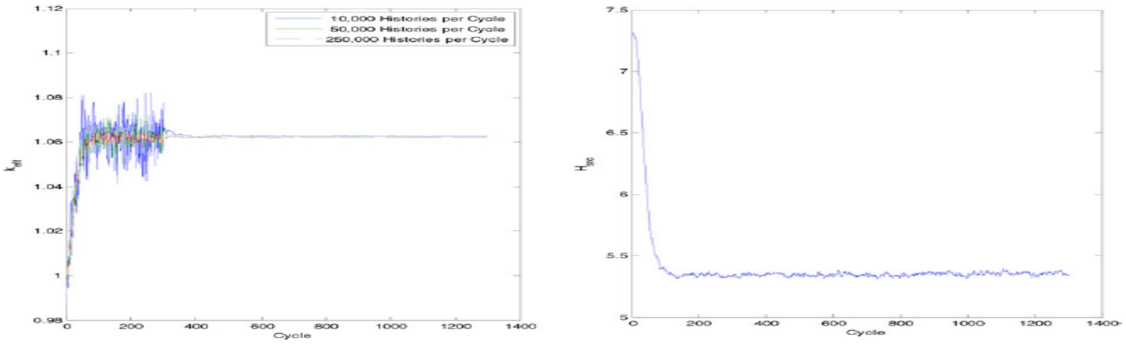


Figure 5.1: Baseline flux results for the OECD benchmark cases.

histories per cycle, an attempt can be made to differentiate between the effects of poor sampling, poor source convergence, and intercycle correlation. Figure 5.2 illustrates the convergence of k_{eff} and the fission source entropy.

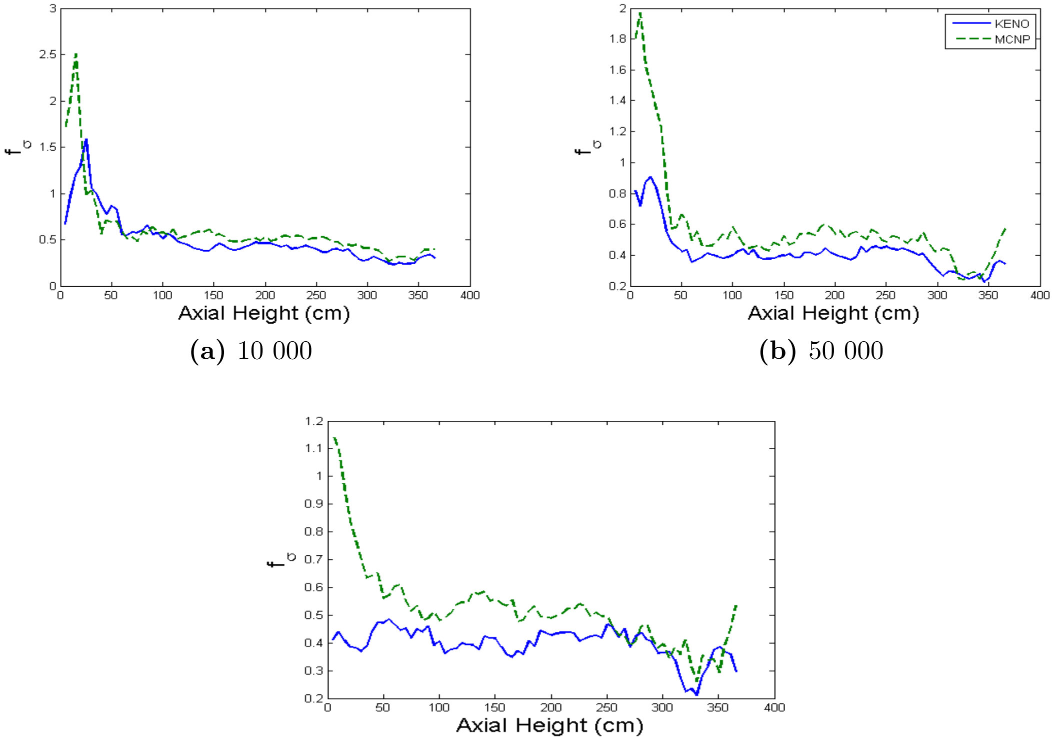
For this study, the convergence of k_{eff} and the fission source distribution appear adequate, although convergence of the entropy does not guarantee source convergence. The only difference between each case is the amount of statistical noise, which decreases as the number of histories per cycle increases, as expected. Figure 5.3 plots f_σ over the low-flux region of the pin cell for sets of cases that use 10 000, 50 000, and 250 000 histories per cycle, 300 skipped cycles, and 1000 active cycles.

For the uncertainty in the flux tallies, the cases that ran 10 000 histories per cycle show that the magnitude of underprediction is relatively high in the low-flux region of the pin cell. When comparing these results to the cases that ran 50 000



(a) k_{eff} **(b)** H_{src}

Figure 5.2: k_{eff} and H_{src} versus cycle for OECD case 2.2.



(a) 10 000

(b) 50 000

(c) 250 000

Figure 5.3: Comparison of f_σ for three different values of the number of histories per cycle.

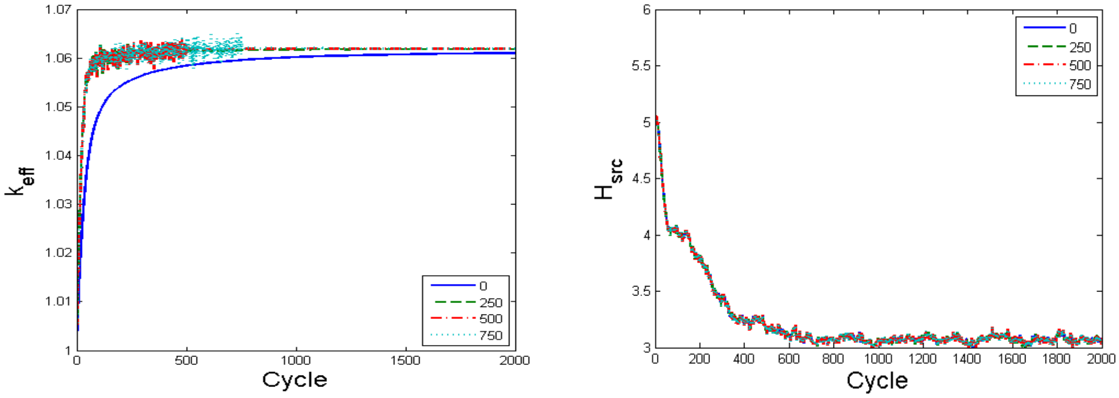
histories per cycle, it is observed that the KENO results are within a much more favorable range in the low-flux regions and the MCNP results are not much worse. Increasing the number of histories per cycle further from 50 000 to 250 000 effectively eliminates any underprediction in the uncertainty as well as the connection between the low-flux regions of the pin cell and a higher magnitude of underprediction. This leads to the conclusion that the underprediction in the uncertainty for case 2.2 is due to poor sampling in the low-flux regions of the pin cell, which can be corrected by simulating more histories per cycle. It is also shown that the differences between MCNP and KENO in this particular analysis are not significant enough to merit further investigation.

5.2.2 Initial Cycles Skipped

The burnup distribution for case 2.3 is similar to that of case 2.2, but convergence of the fission source distribution is much more difficult because of the small increase in the flux profile near the bottom of the pin cell. Therefore, case 2.3 is used to examine the effects of an ill-converged source distribution in terms of f_σ. Figure 5.4 illustrates the convergence of k_{eff} and the fission source entropy.

From Figure 5.4 one notices that skipping zero cycles does not allow k_{eff} to converge within the first 1000 cycles, and the plot of the entropy versus cycle does not indicate convergence even after 500 cycles; hence, it appears that more than 500 skipped cycles are needed for this case. To view the convergence of k_{eff} versus cycle more closely, Figure 5.5 plots k_{eff} versus cycle for 250, 500, 750, and 1000 initial cycles skipped. One should note that the large oscillations observed in Figure 5.5 are because the value for k_{eff} output during the inactive cycles is the cycle value, whereas during the active cycles k_{eff} is output as a rolling average.

From Figure 5.5 it is difficult to identify convergence since steady oscillation about the final value is not observed. However, by observing k_{eff} versus cycle for all 50 cases, as shown in Figure 5.6, it appears that the deviation from the final value is



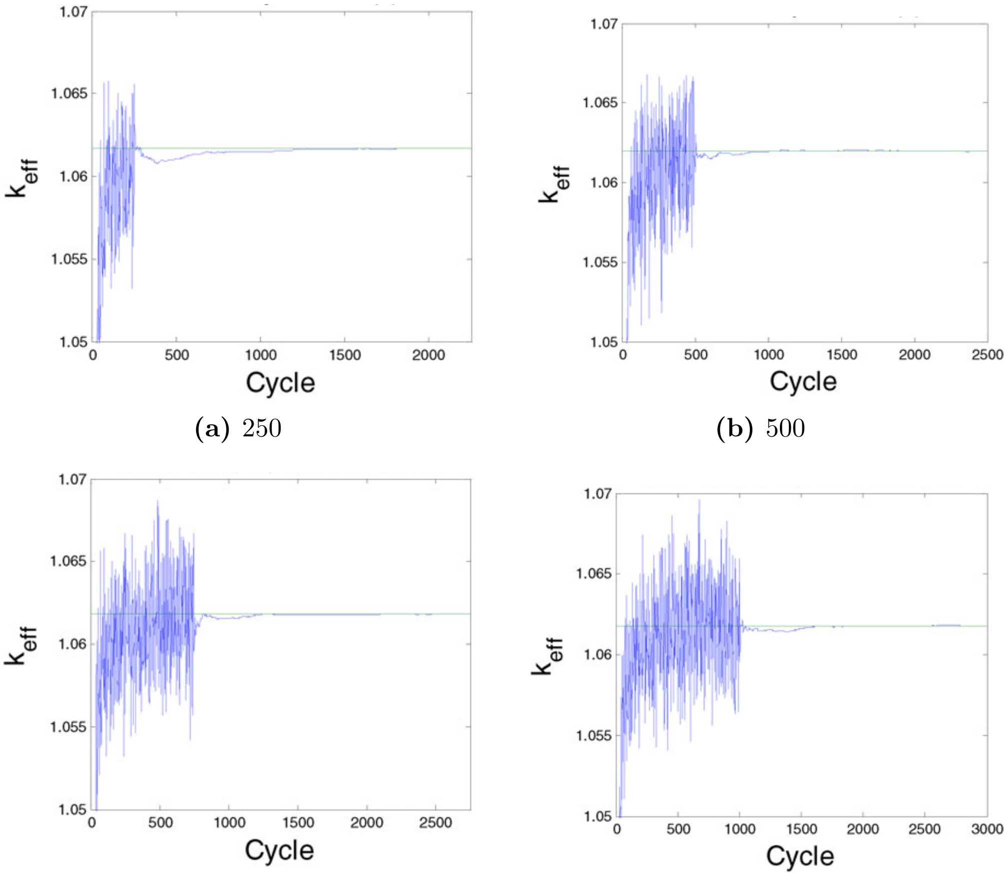
(a) k_{eff}

(b) H_{src}

Figure 5.4: Plot of k_{eff} and H_{src} versus cycle for case 2.3.

nothing of great concern since the final result oscillates evenly about the mean value and no decipherable trend is observed. Therefore, judging from the plot of entropy versus cycle, 1000 cycles is a sufficient number of skipped cycles for case 2.3. The underprediction in the uncertainty for the eigenvalue is again no greater than a factor of 1.5. Figure 5.7 displays a comparison of the value of f_σ as a function of axial height. The number of initial cycles skipped varies from 0 to 1000 while 2000 active cycles were simulated for each case.

From the case that used 10 000 histories per cycle, it can be observed that if the source is not properly converged, the magnitude of the underprediction in the uncertainty can be as large as a factor of 20. A smaller peak is also present near the top of the pin cell, but the peak disappears when an adequate number of initial cycles are skipped. However, even when an adequate number of cycles have been skipped, the results are still being significantly underpredicted. The results for the case with 25 000 histories per cycle show a decrease in the overall magnitude of the convergence metric, but when the number of histories increases further to 100 000, no reduction in the f_σ values is observed. Therefore, cycle-to-cycle correlations are likely the source of the remaining underprediction in the uncertainty.



(a) 250

(b) 500

(c) 750

(d) 1000

Figure 5.5: k_{eff} versus cycle for case 2.3 for four different values of the number of initial cycles skipped.

One other important fact to mention is that increasing the number of active cycles may increase the magnitude of underprediction in the presence of an ill-converged source distribution or cycle-to-cycle correlations. Figure 5.8 illustrates the effect of running more active cycles with and without an adequately converged source. When using an ill-converged source distribution f_σ increases from approximately 9 to greater than 13 when the number of active cycles is increased from 1000 to 2 000. This behavior holds true for any number of cycles skipped that is below the number of cycles required to obtain sufficient convergence of the fission source. If a properly

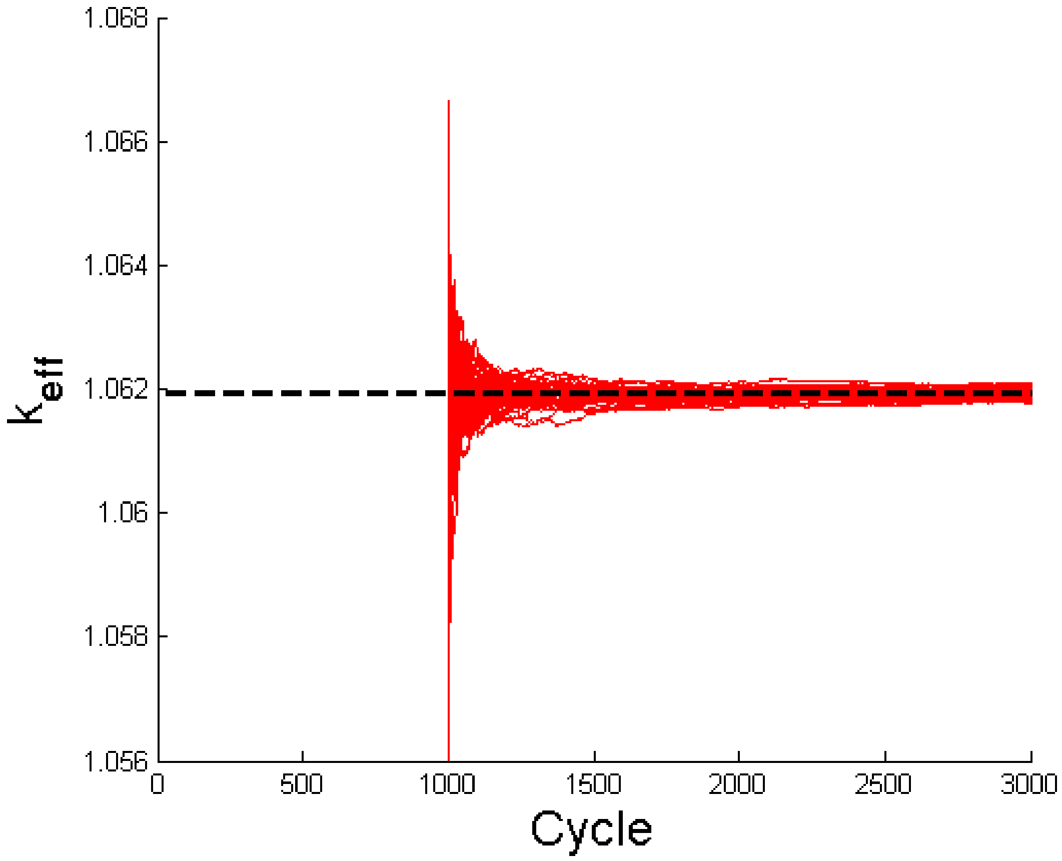
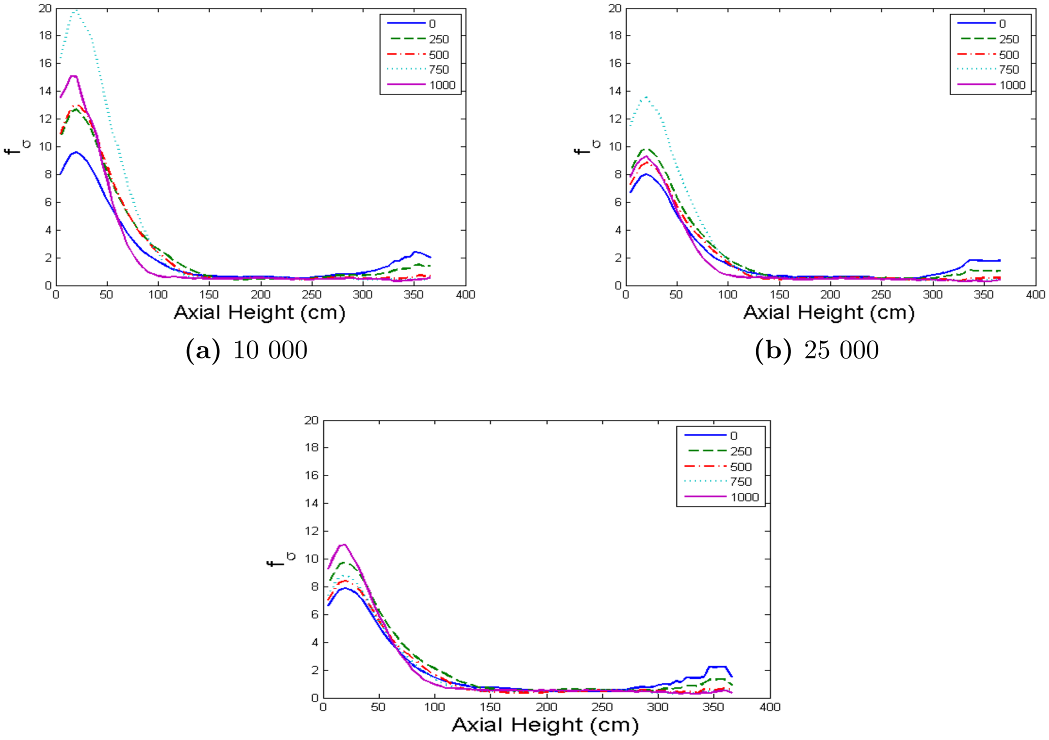


Figure 5.6: Plot of k_{eff} versus cycle for all 50 cases for case 2.3.

converged source distribution is used, the additional active cycles will serve their typical purpose of reducing the uncertainty while their effect on the underprediction of the uncertainty is insignificant.

5.2.3 Cycle-to-Cycle Correlations

Case 2.1 represents a simple pin cell with a symmetric axial burnup distribution. It is anticipated that source convergence will be difficult to assess given the symmetric configuration even if both k_{eff} and H_{src} indicate convergence. Figure 5.9 plots k_{eff} and H_{src} versus cycle for three different values of histories per cycle. Note that a



(a) 10 000

(b) 25 000

(c) 100 000

Figure 5.7: Comparison of f_σ for three different values of the number of initial cycles skipped.

larger number of histories results in a larger value for H_{src} per Equation 3.8. Both cases ran 800 active cycles with 200 initial cycles skipped.

It appears that convergence is easily obtained after approximately 100 skipped cycles in all three cases, and the observed underprediction in the eigenvalue uncertainty is no greater than 20% in each case. Figure 5.10 presents graphs of f_σ for simulations with varying Monte Carlo parameters to test how the underprediction behaves for case 2.1 with respect to both the number of histories per cycle and the number of initial cycles skipped. The results displayed in Figure 5.11 are contrary to what would be expected given the previous analyses performed for cases 2.2 and 2.3 and the indicated convergence shown in Figure 5.9. Increasing the number of histories

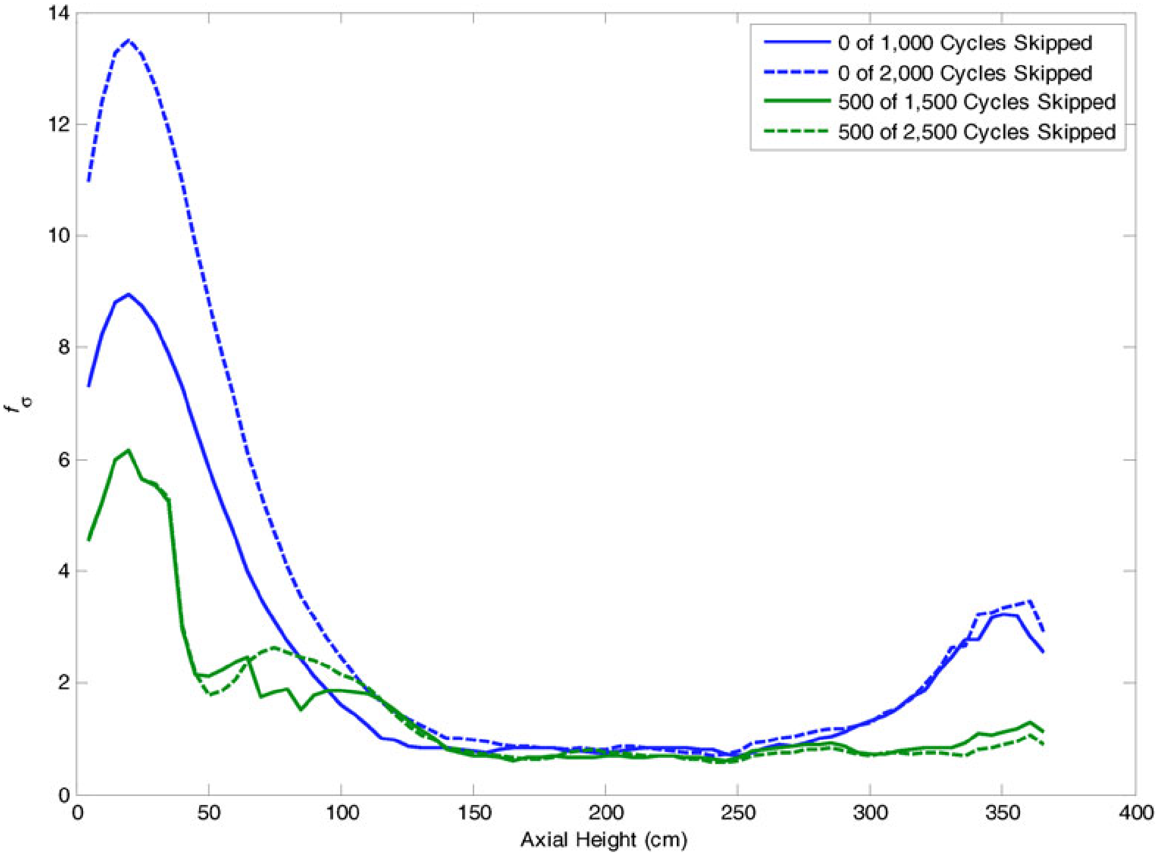
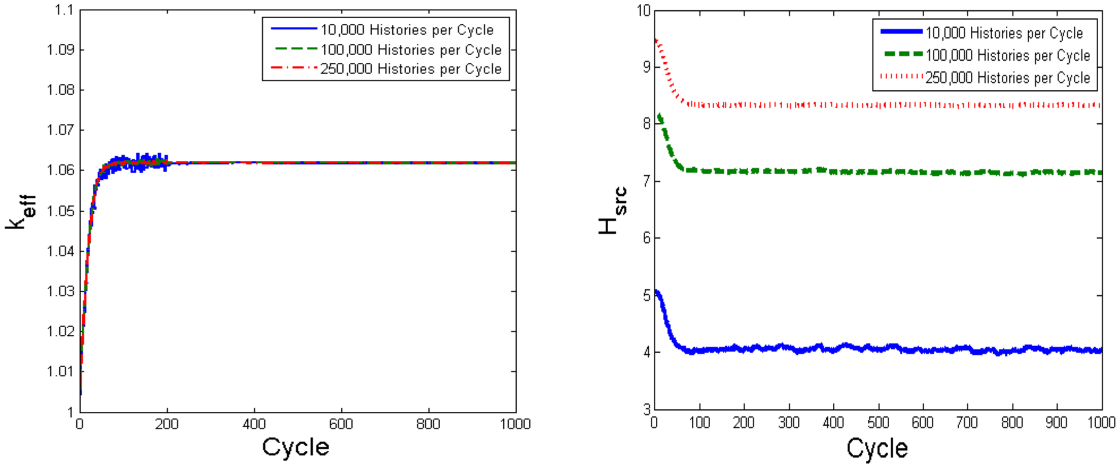


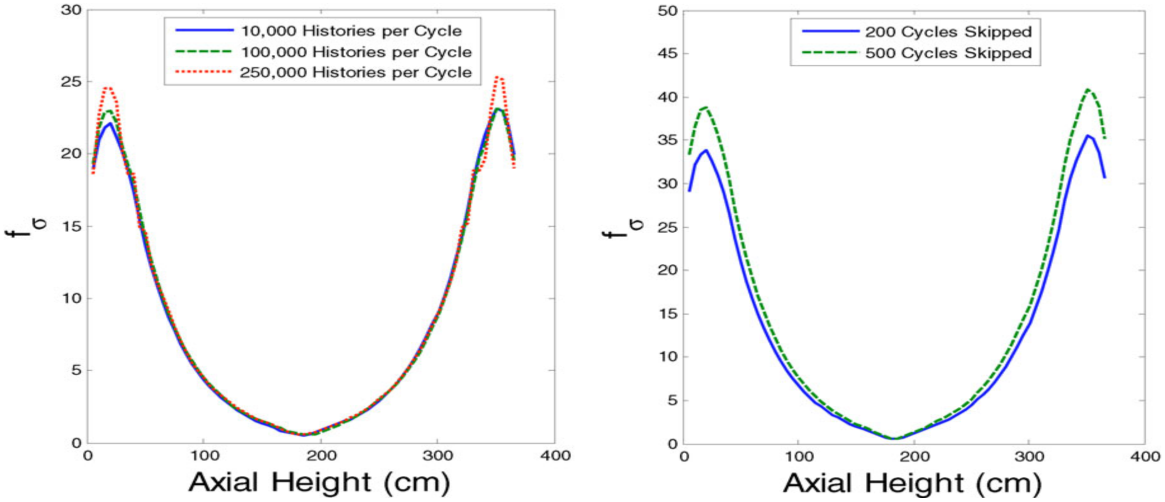
Figure 5.8: Comparison of f_σ for converged and ill-converged source distributions with a varying number of active cycles.

per cycle does not seem to reduce the underprediction in the uncertainty (in fact, it appears to have a small but negative effect for the case that ran 250 000 histories per cycle), which implies that the source regions are being sampled reasonably. The comparison versus the number of cycles skipped (both cases running 10 000 histories per cycle) suggests that skipping more cycles appears to have a negative effect as well. Further investigation reveals that although the source appears converged, the source shape for each of the different random number seeds converges differently, and none of the results are symmetric, contrary to expectations. Plots of the flux for three of the 50 individual cases, denoted as cases A, B, and C, along with the average flux of



(a) k_{eff} **(b)** H_{src}

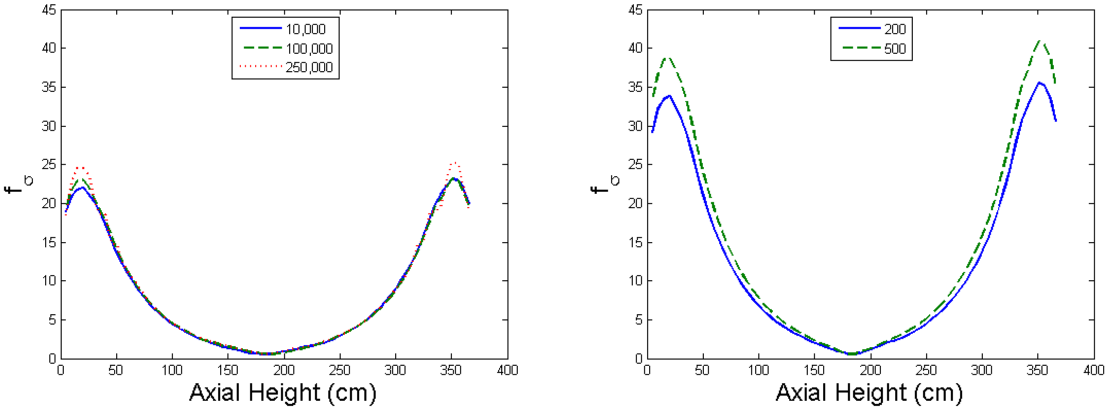
Figure 5.9: k_{eff} and H_{src} versus cycle for OECD case 2.1.



(a) k_{eff} **(b)** H_{src}

Figure 5.10: f_{σ} as a function of axial height for case 2.1.

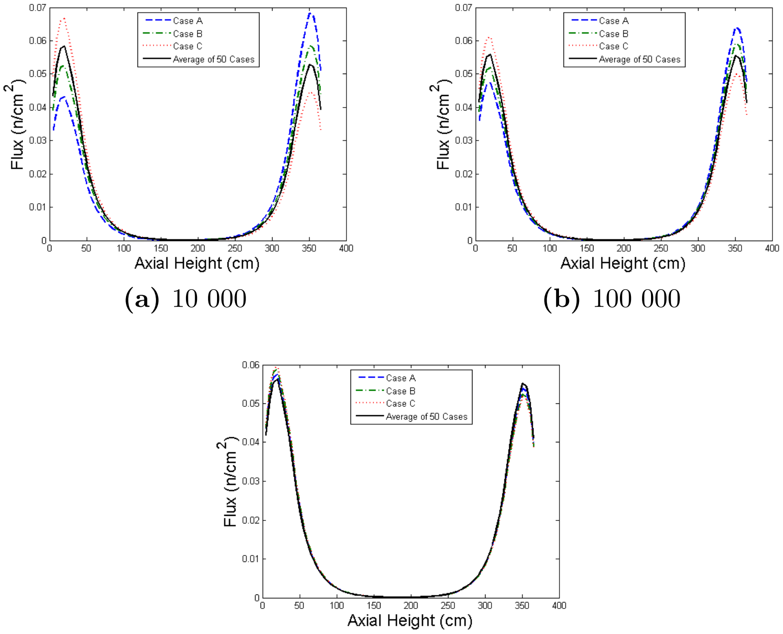
all 50 cases, are presented in Figure 5.12 for the three different values of the number of histories per cycle.



(a) Histories Per Cycle

(b) Initial Cycles Skipped

Figure 5.11: f_σ for select values of the number of histories per cycle and initial cycles skipped for case 2.1.

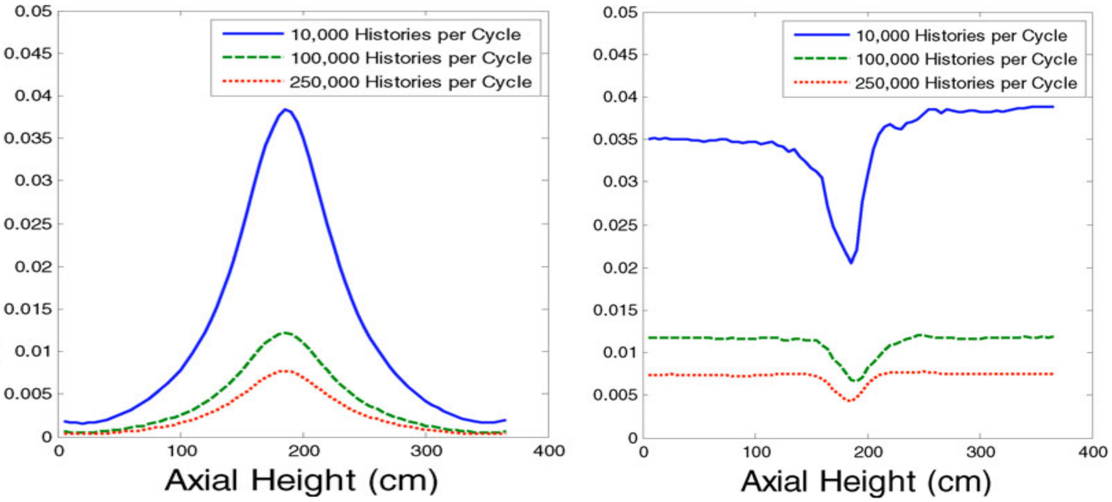


(a) 10 000

(b) 100 000

(c) 250 000

Figure 5.12: Comparison of the flux profile between three individual simulations for case 2.1.



(a) apparent relative error

(b) real relative error

Figure 5.13: Apparent and real relative errors for case 2.1.

For the case with 10 000 histories per cycle, the flux varies significantly between the individual cases; however, the average flux nearly achieves the symmetric shape expected. As the number of histories per cycle is increased, the deviation in the flux results becomes smaller. However, the increase in the number of total neutron histories leads to a lower apparent variance, and the apparent variance is decreasing on the same order as the deviation between the individual flux results. Therefore, the magnitude of underprediction remains relatively unchanged. Plots of the real and apparent variance are provided in Figure 5.13. Figure 5.13 helps to illustrate the issues with case 2.1. Because the majority of simulations converge to an asymmetric source distribution, the apparent variances are relatively low in the high-flux regions of the pin cell where the majority of the source particles are sampled. However, each case does not converge to the same asymmetric source distribution. The source particle population is generally higher at one end of the pin cell than the other. Because of the variation in flux profiles across different simulations, the real variance is much higher than the apparent variance. This is not the case in the low-flux region at the center of the pin cell, because the variation across simulations is relatively small.

Since case 2.1 yields a symmetric flux profile with the lowest flux being in the middle of the pin cell, the high-flux regions at the top and bottom of the pin cell are not able to communicate with each other easily (i.e., the probability that a particle emitted near one end of the pin will make it to the other end is very low); therefore, the source distribution tends to converge to one of many semistable fission source distributions, most of which are significantly asymmetric (i.e., wrong). If a few particles are able to travel from one high-flux region to the other, the source shape may change drastically. To see if this is indeed occurring, the source convergence was monitored out to 10 000 cycles for the case that ran 10 000 histories per cycle. The results shown in Figure 5.14a reveal that the assumption of an adequately converged source was technically incorrect. Since this phenomenon has a chance of occurring multiple times, the Shannon entropy may not be a useful metric for a case as pathologically difficult as case 2.1. As an example, Figure 5.14b plots the entropy for four different simulations that are identical with the exception of a different initial random number seed.

The issue of source convergence illustrated in Figures 5.14 and 5.14b is not evident when looking at the entropy plots for the cases that ran 100 000 histories per cycle in Figure 5.9. Considering the difficulty in communication between the two high-flux regions, it is not guaranteed that a simulated particle will travel from one high-flux region to the other during a given cycle. The results illustrated in Figures 5.14 and 5.14b indicate that the probability of this occurring is on the order of 1 in 10 000. This is why the entropy versus cycle plots in Figure 5.14b are different for multiple instances of the same case, noting that the only difference between cases is the random number seed. If 100 000 histories per cycle are used, communication between the two high-flux regions may occur more often, and thus the source does not appear to have as much difficulty converging within the first 1000 cycles. This can be verified by observing that the entropy versus cycle plot for simulations that run 100 000 histories

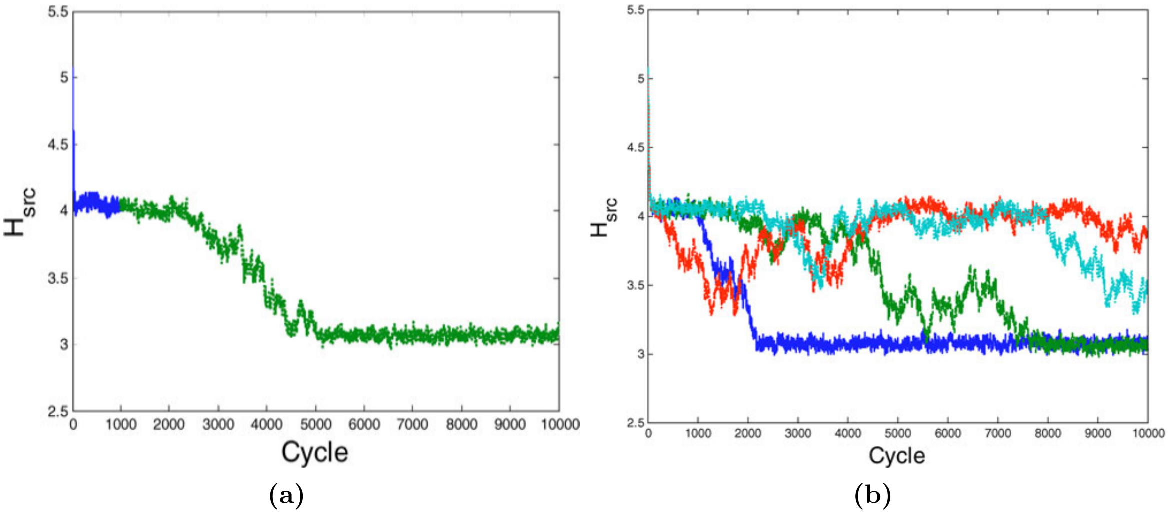


Figure 5.14: Entropy versus cycle for case 2.1 for a case that ran 10 000 histories per cycle and 10 000 total cycle.

per cycle or more show the same convergence trend (i.e., every case converges to the same entropy value after approximately the same number of inactive cycles).

From Figure 5.14b and the analysis of the entropy profiles of the other 46 cases, it appears that there are two distinct semistable states, one at $H_{src} = 3$ and one at $H_{src} = 4$. When $H_{src} = 3$, the source distribution is nearly completely tilted to one end of the pin cell, which is why once H_{src} reaches a value of three for any of the cases, it remains at three and appears converged. At $H_{src} = 4$, the source distribution is much closer to symmetric, which is why every case begins at $H_{src} = 4$ since the initial source distribution is symmetric. However, at $H_{src} = 4$, these cases have a difficult time staying converged since statistical noise can easily tilt the source distribution toward one end of the pin cell or the other, and once the source distribution begins to favor one end of the pin cell, it is very unlikely that it will recover. Thus, one cannot assume that the converged source distribution is correct even if it appears to be so when observing the entropy profile. No matter how many histories are run, the simulation still has a probability of finding the asymmetric state, although using more histories does seem to decrease this probability.

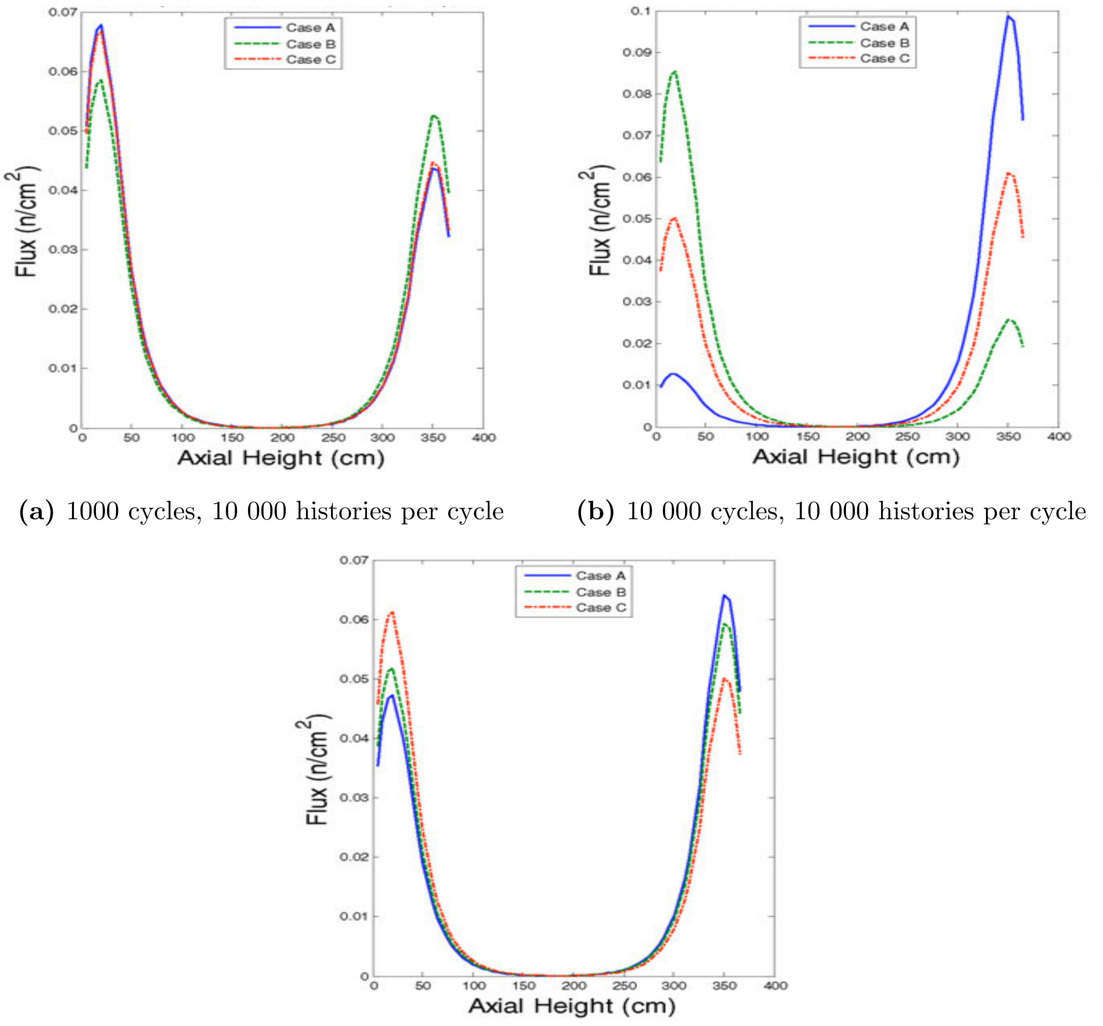
This result also implies that the source particles used in a given cycle depend predominantly on the results from the previous cycle, which are likely not representative of the true source distribution. Hence, the more cycles simulated for a particular case, the larger the underprediction in the uncertainty will be. For example, the case that ran 10 000 cycles underpredicted the uncertainty by a factor of more than 100.

Plots of the axial flux profile are provided in Figure 5.15 to illustrate the issues previously discussed. Three independent simulations using 10 000 histories per cycle and 1000 cycles indicate the flux to be higher in the top of the pin cell, and the relative difference between the two peaks is different for all three cases. Carrying this out to 10 000 cycles yields results that are in complete disagreement. The cases that ran 100 000 histories per cycle seem to be in closer agreement than the cases that ran 10 000 histories per cycle but are still incorrect since a symmetric flux profile is the expected result. It is because a converged source distribution cannot be obtained that the trends observed in cases 2.2 and 2.3 are not observed in case 2.1.

5.2.4 Three-Dimensional Analysis

The GBC-32 cask assembly case is used to illustrate underprediction of tally uncertainties for a three-dimensional case representative of a spent-fuel criticality safety application. For this case, 100 000 histories per cycle were used with 1000 active cycles and 200 skipped cycles. Since this case is relatively large, parametric studies similar to those performed with the pin cell test problems were not executed. Instead, this case will take advantage of the lessons learned from the previous test problems to verify that the underprediction in the uncertainty for a three-dimensional assembly case is negligible when good Monte Carlo practice is observed.

Beginning with the conservative guess that 100 000 histories per cycle will adequately represent the fission source distribution and that 200 initial cycles skipped



(a) 1000 cycles, 10 000 histories per cycle

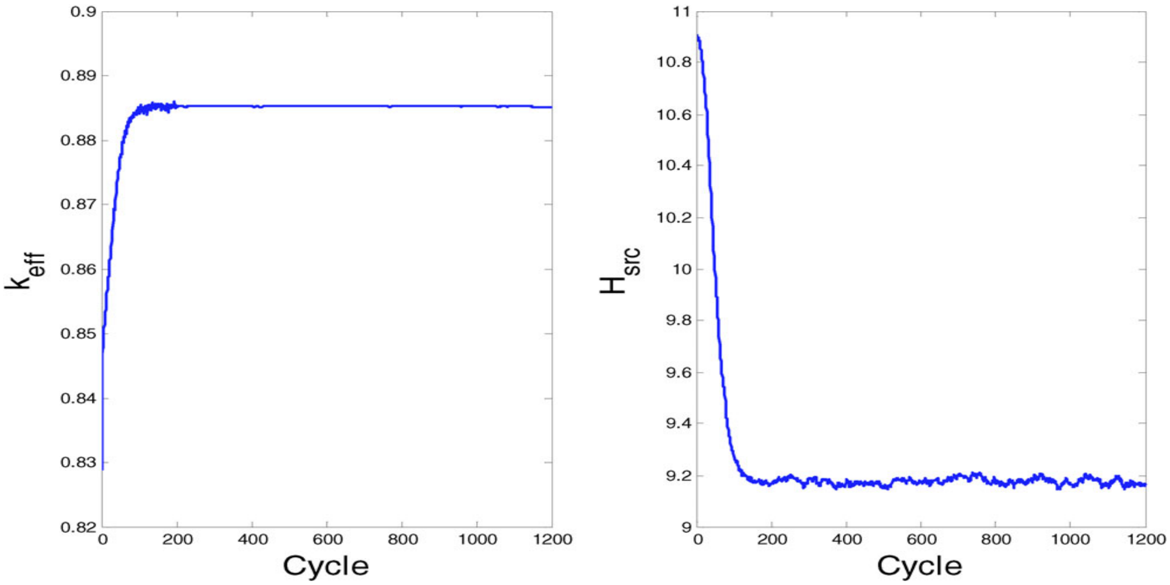
(b) 10 000 cycles, 10 000 histories per cycle

(c) 1000 cycles, 100 000 histories per cycle

Figure 5.15: Flux for three independent simulations for three variations of case 2.1.

will allow the source to converge, a baseline case was ran to observe k_{eff} and the Shannon entropy of the fission source distribution versus cycle. These results are presented in Figure 5.16 and verify that 200 skipped cycles appears to be sufficient for adequate source convergence.

By running a set of 50 independent cases using the parameters deemed appropriate from the baseline case, the difference in the real and apparent uncertainty can



(a) k_{eff} **(b)** H_{src}

Figure 5.16: k_{eff} and H_{src} of the fission source distribution for the baseline case for the GBC-32 cask assembly model.

be evaluated. From these results, it was determined that the f_σ values for the eigenvalue for KENO and MCNP are 1.0346 and 0.9864, respectively, so both codes provided reasonably accurate values for the uncertainty for k_{eff}. To evaluate the underprediction of the flux tallies, the maximum value of f_σ was taken over the 18 axial regions in each pin cell to provide the plot shown in Figure 5.17. Comparable studies have been performed for three-dimensional reactor-type problems and have observed similar results [93].

5.3 Summary of Underprediction Analysis

An analysis of the magnitude and behavior of underprediction in Monte Carlo uncertainties for k_{eff} and localized flux tallies in eigenvalue calculations was presented. The underprediction was found to have at least three components:

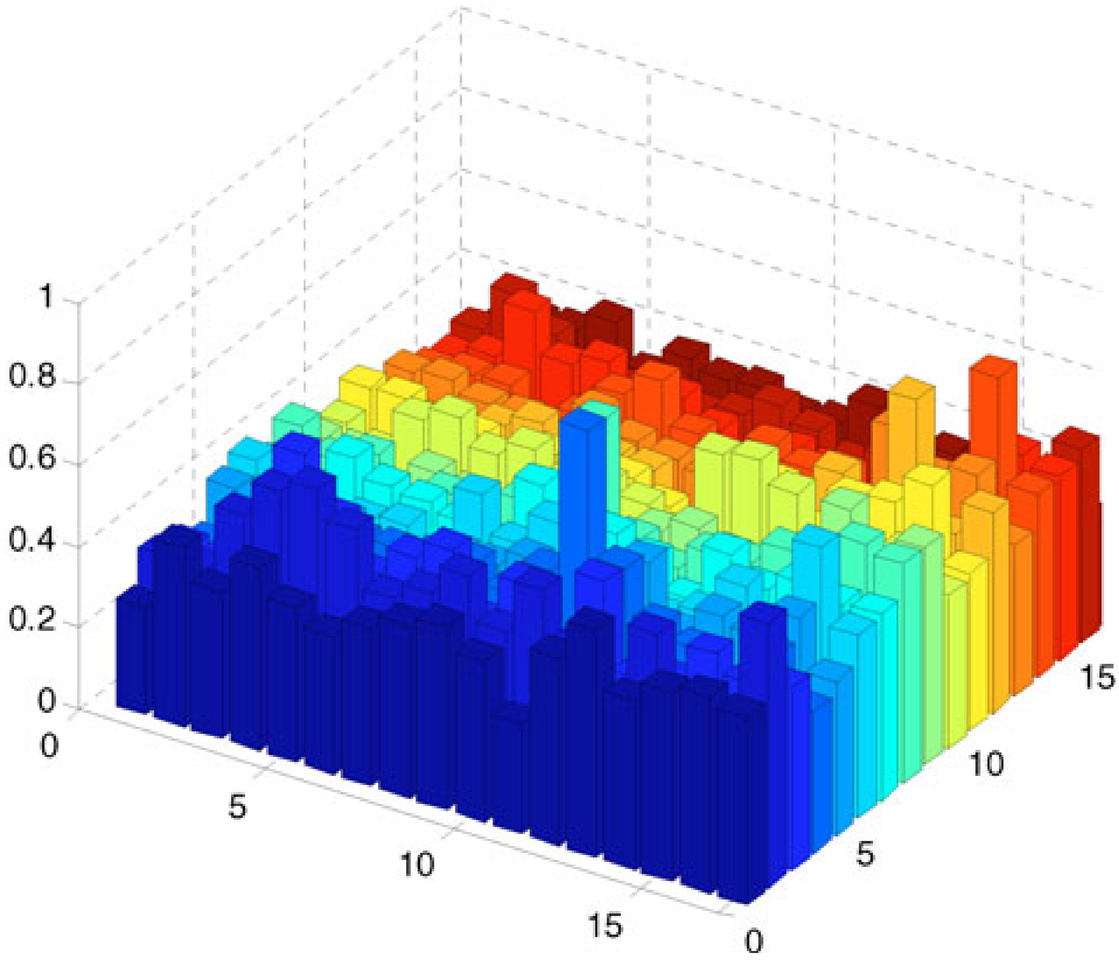


Figure 5.17: Maximum factor of underprediction for each pin cell in the GBC-32 cask assembly model.

one from accumulating tallies using an ill-converged fission source, one from undersampling fission source regions, and one from cycle-to-cycle correlations. An emphasis on differentiating between these components was presented along with their relative effects and behavior with respect to the amount of underprediction that may exist in uncertainty estimates. Suggestions on how to identify each individual component and minimize their negative effects were also discussed.

Studies performed on the number of histories per cycle and the number of initial cycles skipped helped to diagnose the relative effects and behavior of f_σ with respect to these two parameters. In case 2.2 it was shown that increasing the number of histories per cycle ensures adequate sampling of the low-flux region at the bottom of the pin cell and completely eliminates the observed underprediction in the tally uncertainties for this case. The analysis on the number of initial cycles skipped demonstrated the importance of monitoring the convergence of the fission source. An ill-converged fission source distribution is potentially the most significant reason behind underpredicted uncertainties and can cause the predicted uncertainty to be as much as 100 times lower than the true uncertainty, and the magnitude of this underprediction has a dependence on the number of active cycles. Plots of k_{eff} and H_{src} versus cycle can help provide a preliminary measure of how many initial cycles should be skipped. If these two metrics do not appear to converge within a reasonable number of cycles, then using more histories per cycle is recommended.

It was demonstrated that a significant reduction in f_σ is obtained upon reaching convergence, which is particularly relevant in poorly sampled (low-flux) regions. A correlation between f_σ and the number of histories per cycle can identify poorly sampled regions if the magnitude of f_σ increases as the flux decreases.

Identifying underprediction in the uncertainty due to cycle-to-cycle correlations in the fission source involves first dealing with the other two components and then determining if the uncertainty is still being significantly underpredicted. Dominance ratios near unity provide a preliminary indication that cycle-to-cycle correlations may be causing the uncertainty to be underpredicted. This can be accomplished deterministically or by using various Monte Carlo methods such as autoregressive moving average fitting [94] or the noise propagation matrix method [52]; however, most Monte Carlo codes currently lack the implementation to provide results using these methods. Cases where the important regions of the model have difficulty exchanging particles[*] indicate that intercycle correlations in the fission source may be an issue. The negative effects of these intercycle correlations were observed in the case with the symmetric axial burnup distribution (case 2.1) as well as in case 2.3,

[*] These cases tend to represent cases in which the dominance ratio is near unity.

although the effects in case 2.3 were much less significant due to the lower flux near the bottom of the pin cell relative to the symmetric case.

The analysis of the three-dimensional spent-fuel assembly with an axial burnup distribution in a storage cask geometry illustrates that the uncertainty for both the eigenvalue and localized tallies is overpredicted rather than underpredicted. This behavior is also observed in cases 2.2 and 2.3. The overestimation of the standard deviation indicates a negative cycle-to-cycle correlation. Thus, further investigation of additional problem characteristics and calculational parameters is recommended to provide a complete understanding of the behavior of f_σ in Monte Carlo eigenvalue calculations.

Chapter 6

Domain-Decomposed Uncertainty Estimation

This chapter details the research and development of algorithms for calculating tally statistics in domain decomposed Monte Carlo calculations.

6.1 Variance Estimation

One of the primary technical challenges for domain-decomposed Monte Carlo is coming up with a method for efficiently estimating the uncertainty for localized tallies [65]. The process of estimating the sample variance for a tally in a Monte Carlo calculation is a relatively straightforward process in serial and domain-replicated scenarios. In domain-decomposed simulations, however, the movement of particles between domains (processors) introduces significant complexity, especially when a large number of tallies are to be estimated.

Consider a single particle history and the algorithm surrounding track-length estimation of a mesh tally. As the particle is transported across the mesh, the first and second moments must be accumulated in order to provide estimates of the mean and variance for a particular tally result. Thus, the algorithm involves the following

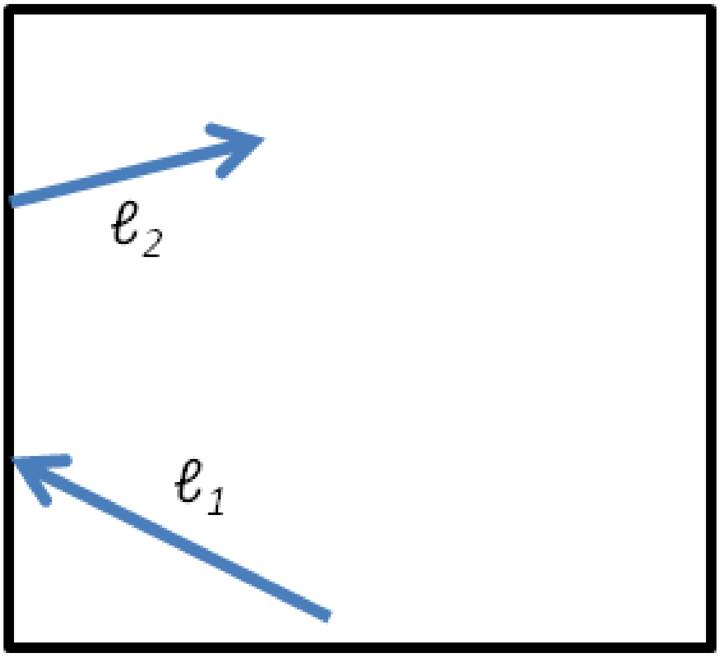


Figure 6.1: Diagram of two tracks made by a single particle in a tally cell.

operations over the course of a history:

$$FirstMoment_j = \sum_{i=1}^{N} \ell_i \tag{6.1}$$

$$SecondMoment_j = (\sum_{i=1}^{N} \ell_i)^2 \tag{6.2}$$

where ℓ_i designates individual track lengths made by the particle in cell j over the course of its history. Now consider a simple example, illustrated in Figure 6.1, where a particle makes two different tally contributions in a tally cell. Using this example, Equations 6.1 and 6.2 are given as:

$$FirstMoment_j = \ell_1 + \ell_2 \tag{6.3}$$

$$SecondMoment_j = (\ell_1 + \ell_2)^2 \tag{6.4}$$

Now consider the fact that this cell and only this cell exists on a given processor and that once the particle hits a boundary it must be communicated to a different processor. After the particle travels the distance ℓ_1, it can be added to the first moment without consequence, but the contribution to the second moment cannot be made since the track length ℓ_2 has not been recorded yet. Therefore in order to correctly estimate the variances using this algorithm, this processor must wait to see if any additional contributions are made by the particle before tallying the second moment for this history. This type of behavior can significantly affect the efficiency of a parallel computation.

Statistical uncertainty estimation for localized tallies in domain-decomposed environments was identified as one of the primary technical challenges associated with domain-decomposed Monte Carlo [65]. Estimating the sample variance involves calculating the sum of the tally contributions as well as the sum of the squared contributions for each history. This is a straightforward process in serial and domain-replicated scenarios. In domain-decomposed simulations, however, the movement of particles between domains (and thus processors) introduces significant complexity, especially when a large number of tallies are to be estimated.

6.2 Tallies

6.2.1 Parallel Algorithms

Many different algorithms can be devised to solve the issue of accurately estimating the tally variance in domain-decomposed problems. However guaranteeing that the variances are calculated correctly is not the primary concern, but the efficiency of the algorithm that poses the larger issue. To get a better idea of the issues affecting efficiency, three options are considered for how to proceed after a particle leaves the processor's domain.

1. The processor can wait for the history to be completed before calculating the second moment. This introduces a synchronization point and often requires the processors to sit idle, which can potentially waste a significant amount of the available processing time.

2. Tally contributions associated with the exiting particle can be stored in a buffer while the processor begins transporting another particle. This approach can potentially require large amounts of memory and introduces additional communication because the processors must know when the history is complete so that the buffered data can be processed and freed.

3. Tally contributions can be communicated with the particle information. The downside to this technique is that particles may end up carrying around large amounts of information, which increases the cost of communication and limits the number of particles that can be buffered at any one time. Moreover, when the history is terminated, the tally information must be communicated back to the processors on which the contributions were scored.

To illustrate the issues associated with adding synchronization points (as described in option 1 above), consider a Monte Carlo simulation on a square geometry decomposed into four domains in which an estimate of the scalar flux on an overlaid mesh is desired. The tracks associated with three hypothetical histories are shown on this geometry in Figure 6.2. Particle A contributes to domain 4 and then is communicated to domain 2, where it makes a contribution and then is subsequently killed. In this particular case, the sum of squares of tally contributions on domain 4 could have been computed when particle A left the domain. Now consider particle B. It makes a contribution on domain 4 and then is communicated to domain 3, where it suffers a collision in which it scatters back to domain 4. This occurs multiple times. In order to calculate the variance accurately, one domain will always be waiting while another domain is transporting the history, and finally domain 4 would have to tell domain 3 that the history was completed in order to calculate the variance accurately.

Finally, consider particle C, which begins on domain 3, is communicated to domain 1, where it has multiple scattering interactions before finally being communicated to domain 2, where it is subsequently killed. During this history, domain 3 is doing nothing for the majority of the history, and both domain 3 and 1 experience wait time while the particle is being transported on domain 2. The remaining algorithms for calculating tally variances involve buffering tally information for individual particles. Although these methods allow for asynchronous communication, messages must be transmitted frequently to ensure that the tallies or communication buffers are cleared in a timely fashion. This added communication tends to reduce the scalability of the algorithm. Worse yet, there are no guarantees that memory or communication buffers will not be exhausted. When that happens, a synchronization point is created and the process must wait for a history to end or leave its domain.

Batch Statistics

Another possible solution is to use batch statistics. This process would involve simulating batches of particles and computing variance estimates based on batch mean values. The completion of each batch represents a global synchronization point, which motivates a relatively large batch size. Adding 10s-100s of additional synchronization points does not introduce the efficiency issue which the previously proposed solutions introduced, since those algorithms involved adding millions to billions of additional synchronization points.

The downside of using batch statistics is that the accuracy of the estimation procedure is affected. With batch statistics, the variance of the batch means is not equivalent to the variance of the history scores, unless the batch size is one. A significant consequence of this is the likelihood that large batches will mask the appearance of high-scoring histories. This is cause for concern, because the high-scoring outliers tend to be a symptom of statistical convergence issues (that practitioners should pay particular attention to), especially in hybrid deterministic/Monte Carlo simulations.

Proposed Methodology

Keeping in mind that Shift is currently being designed for LWR full-core simulations, an alternative approach can be considered for estimating tally statistics in domain-decomposed environments. Consider the following assumption: *Once a particle leaves a particular domain, that particle and any progeny associated with the current history do not reenter that domain.* In other words, once a particle leaves a domain, that history is considered to be complete with respect to the tally variance calculation. If a particle does indeed reenter a domain it had previously left, it is considered a new independent history. The consequence of this scheme is that cells which treat a returning particle as a new history will under-predict the variance because:

$$\ell_1^2 + \ell_2^2 < (\ell_1 + \ell_2)^2 \tag{6.5}$$

Thus, this scheme is expected to yield under-predicted variances in tally cells near domain boundaries, where particles are more likely to leave and reenter during the same history. Note that this assumption only affects the variance estimate; the calculated tally means will be identical to those calculated in the conventional way.

Fortunately, the underprediction introduced into the variance calculation can be mitigated using overlapping domains. The domain decomposition algorithm implemented in Shift provides the ability to have domains that overlap each other. The amount that one domain overlaps another domain is specified using a number between zero and one, and this number represents the fraction of the neighboring domain's size that is included in the domain. With overlapping domains, once a particle leaves a domain, it does not start on the boundary of the domain it had recently left because the boundaries for neighboring domains are no longer coincident. Instead, the particle starts some distance, determined by the overlap fraction, inside the other domain boundary. For example, consider a 3 × 3 assembly model where the domain boundaries are defined in between each assembly (i.e. there are 9 domains and each domain owns a single assembly) and the overlap fraction is 0.5. Figure 6.3 displays this configuration and helps illustrate the difference between overlapping and non-overlapping domains for the central assembly. The non-overlapping domain for the central assembly is shaded in red while the outer boundaries of the overlapping domain are represented by the dashed green box. The primary benefit of this decomposition scheme is that it minimizes the amount of domain-to-domain communication during a transport cycle by significantly reducing the probability that a particle will return to the domain it had recently left. In addition, it eliminates the costs (that would be incurred if no overlap was used) associated with particles that scatter back and forth between domains [65]. Using the example provided in Figure 6.3, consider a particle that leaves the central domain by crossing the overlapping boundary. The starting position of this particle in the new domain is such that the particle will have to traverse a distance greater than or equal to the width of an assembly in order to return to the domain from which it came, since this new domain also overlaps the central domain by a factor of 0.5. By considering the probability of this particle both scattering in the direction of the central domain as well as the probability it survives the trip back to the central domain, one can see how overlapping domains can help significantly mitigate the underprediction in the variance introduced by this approximation.

6.3 Variance Estimation Results

To evaluate bias in the estimated variances obtained using this approach, numerical tests were conducted using two variations of the C5G7 model. The first is a modified version of the two-dimensional problem that omits the moderator regions. The second is the full three-dimensional model as defined in the benchmark. Variances from domain-decomposed simulations using the technique described in this paper were compared to those from a single-processor simulation using the conventional variance estimator. Each case features a 2 × 2 decomposition in which all domains are the same size. Then, the amount of overlap is modified in order to examine the effect that various levels of overlap have on the estimated uncertainties. The expectation is that adding overlapping domain boundaries will significantly reduce the errors obtained using non-overlapping domains.

6.3.1 Two-Dimensional C5G7 Model

In this case, the reflector regions of the C5G7 benchmark were omitted. This test case was first run on one processor, and tally variances were estimated in the conventional manner. This establishes the "reference" results for this case. Then the problem was set up and executed in parallel with a 2 × 2 boundary mesh (four domains, each owning one of the assemblies) and a 32 × 32 tally mesh. Both calculations simulated the same number of particle histories in both serial and domain-decomposed simulations so that both cases can be compared directly. Figure 6.4 illustrates the relative difference (with negative values representing underprediction) between the variances estimated for the domain-decomposed case when compared to the serial case. The results show that uncertainty estimates near the domain boundaries are under-predicted by approximately 10-50%, while uncertainties in cells that are at least two pin cells away from a domain boundary are within 5% of the reference results. Considering that the average mean free path of a neutron for this case is approximately 1.5-2.0 cm, and that the mesh tally cells are 1.34 × 1.34 cm in size, it is understandable that the errors introduced using this approximation become small in mesh tally cells further away from the domain boundary as neutrons have a much lower chance of revisiting these cells after leaving and reentering the domain.

6.3.2 Three-Dimensional C5G7 Model

The second example uses the full three-dimensional representation of the C5G7 benchmark case. The decomposition used divides the model into four equally sized domains, and then the flux is tallied in each pin cell. This example tests multiple aspects of the validity of the variance estimation process for domain-decomposed models. The first is to test how much the underprediction will increase when a highly-scattering material is present in the model. Secondly, this model checks the increase in underprediction when the mesh tally cells are roughly the same size as a neutron's average mean free path. In order to provide visualization of the results similar to those presented in Figure 6.4, the middle axial region was used to generate the comparison shown in Figure 6.5. The results in Figure 6.5 show slightly increased underprediction near the domain boundaries relative to the two-dimensional model. In this case, the amount of underprediction is approximately 50% for cells that lie on the domain boundaries versus the 40% shown in Figure 6.4. One can see that significant over-predictions are observed deeper into the moderator region; however the mean values in these tally cells also differ considerably between cases.

6.3.3 Three-Dimensional C5G7 Model with Overlapping Domains

Utilizing overlapping domains provides additional flexibility for decomposing the domains. To evaluate variance estimation in overlapped domains, several different simulations were performed and the effect the amount of overlap has on the variances was observed. Table 6.1 provides the k_{eff} results, and Figure 6.6 illustrates the relative difference in the uncertainty for these cases. The results displayed in Figure 6.6 illustrate the decrease in underprediction of the uncertainty using overlapping domains. The maximum underprediction occurring near the domain boundaries decreases from approximately 50% to below 10% on the case with an overlap fraction of 0.1. As the amount of overlap increases, the underprediction in the variance is further reduced while the remaining underprediction is essentially smeared over a larger number of cells. Because Monte Carlo codes typically report statistical uncertainties in the form of standard deviations or relative errors, Figure 6.7 bins the fraction of the number of tally cells by how much they are under-predicted for each of the different overlapping decompositions.

Table 6.1: Eigenvalue results for cases with various amounts of overlap.

Overlap	k_{eff}	σ	within # standard deviations of serial results
0.0	1.12811	6.658×10^{-5}	0.32
0.1	1.12814	6.652×10^{-5}	0.01
0.2	1.12810	6.419×10^{-5}	0.44
0.3	1.12812	6.226×10^{-5}	0.22
0.4	1.12834	6.674×10^{-5}	2.15
0.5	1.12809	6.325×10^{-5}	0.55

6.4 Summary of Domain-Decomposed Variance Estimation Analysis

The analyses presented in this chapter have shown that assuming histories do not reenter domains they have previously left can provide reasonably estimated tally variances in domain-decomposed problems without adding additional synchronization points, processor communications, or memory allocation other than what would be required by a serial calculation. The bias associated with the assumption that particles do not reenter domains they has previously left has been evaluated for the C5G7 reactor benchmark case and the results appear adequate for uncertainty estimation using domain-decomposed Monte Carlo in LWR eigenvalue problems. A simple preliminary statistical check on the average mean free path of neutrons in a tally cell relative to the size of the tally cell can be used to determine the recommended amount of overlap for a particular problem. This will help mitigate the bias by ensuring that the size of the overlapped region is greater than the average mean free path of a neutron, and will therefore significantly reduce the probability of a neutron making additional tally contributions to cells on a domain it had previously left. This methodology is anticipated to produce favorable results for domain-decomposed full-core simulations because it eliminates the need for additional computational time and resources. Future work should include testing this method using continuous-energy physics and evaluating the limitations, if any, of the approximation.

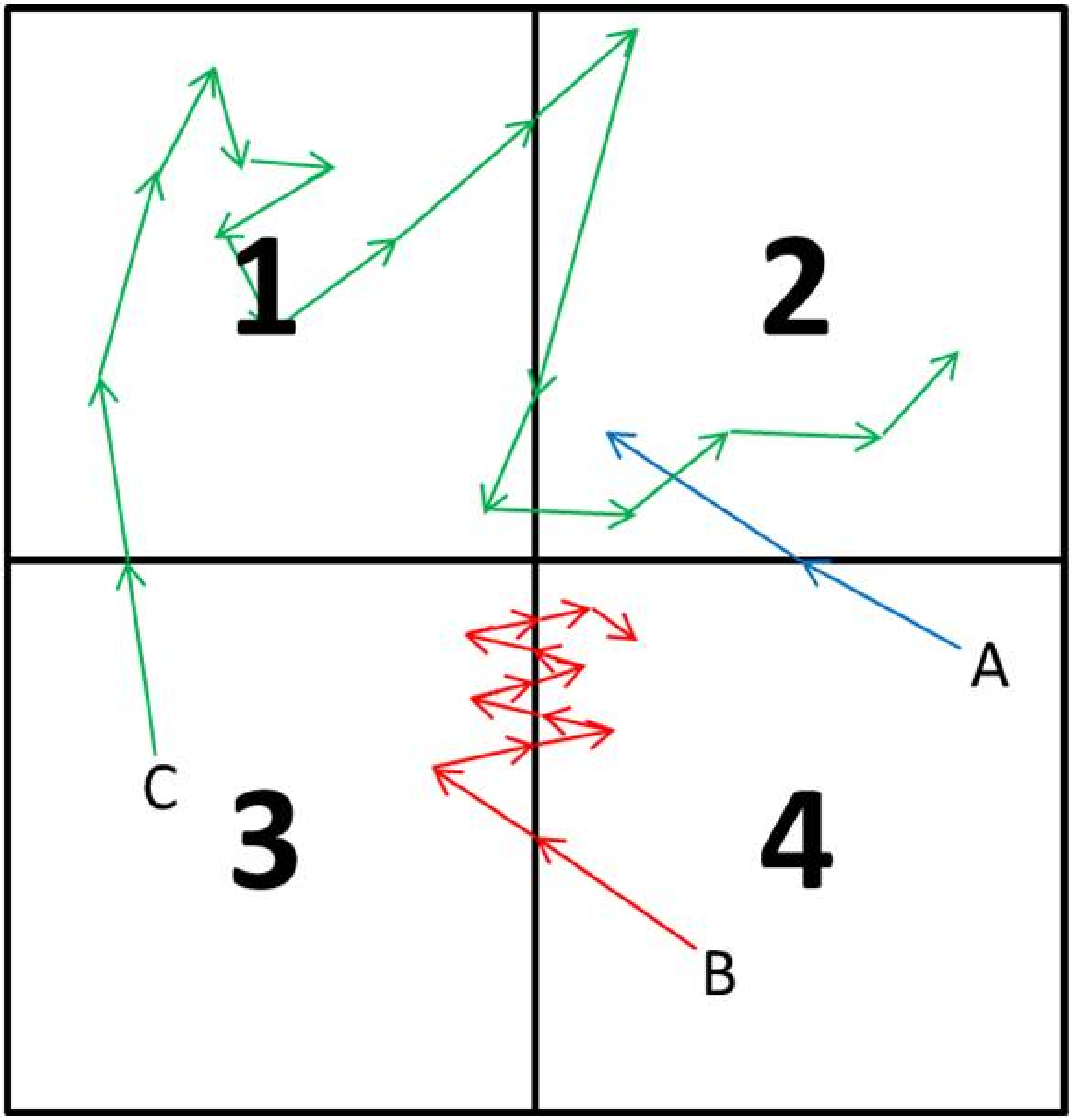


Figure 6.2: Diagram of three individual particle histories on a square geometry decomposed into four individual domains.

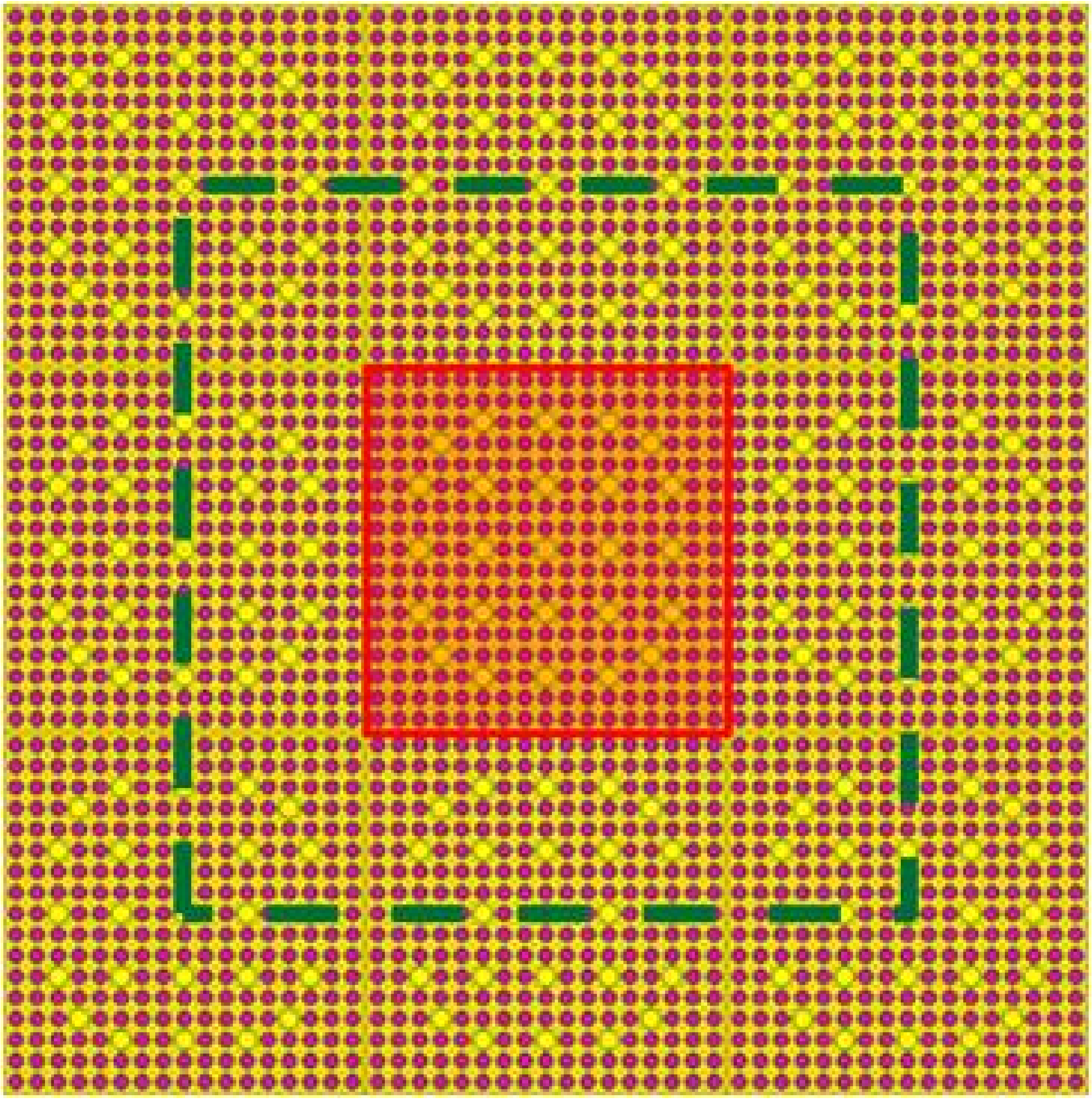


Figure 6.3: Illustration of an overlapping domain.

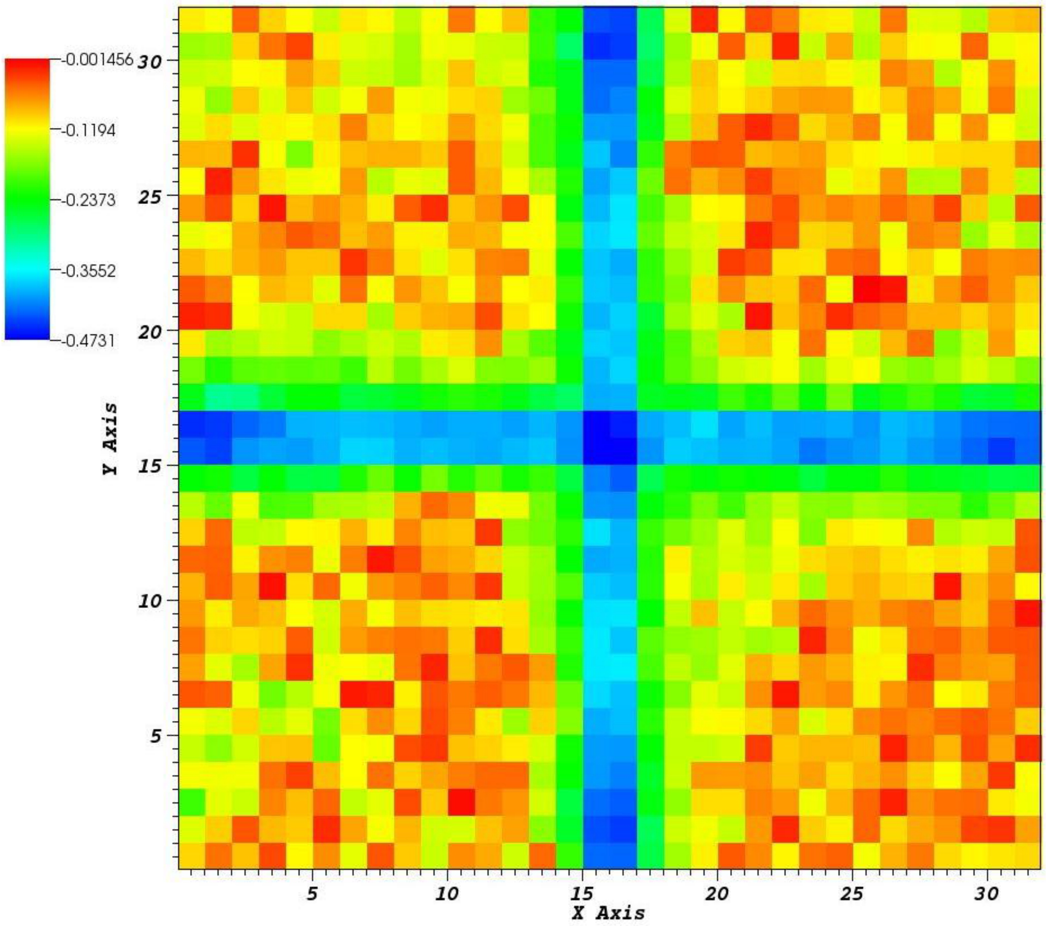


Figure 6.4: Relative difference in mesh tally variance estimates for the 2-D C5G7 domain-decomposed Monte Carlo simulation.

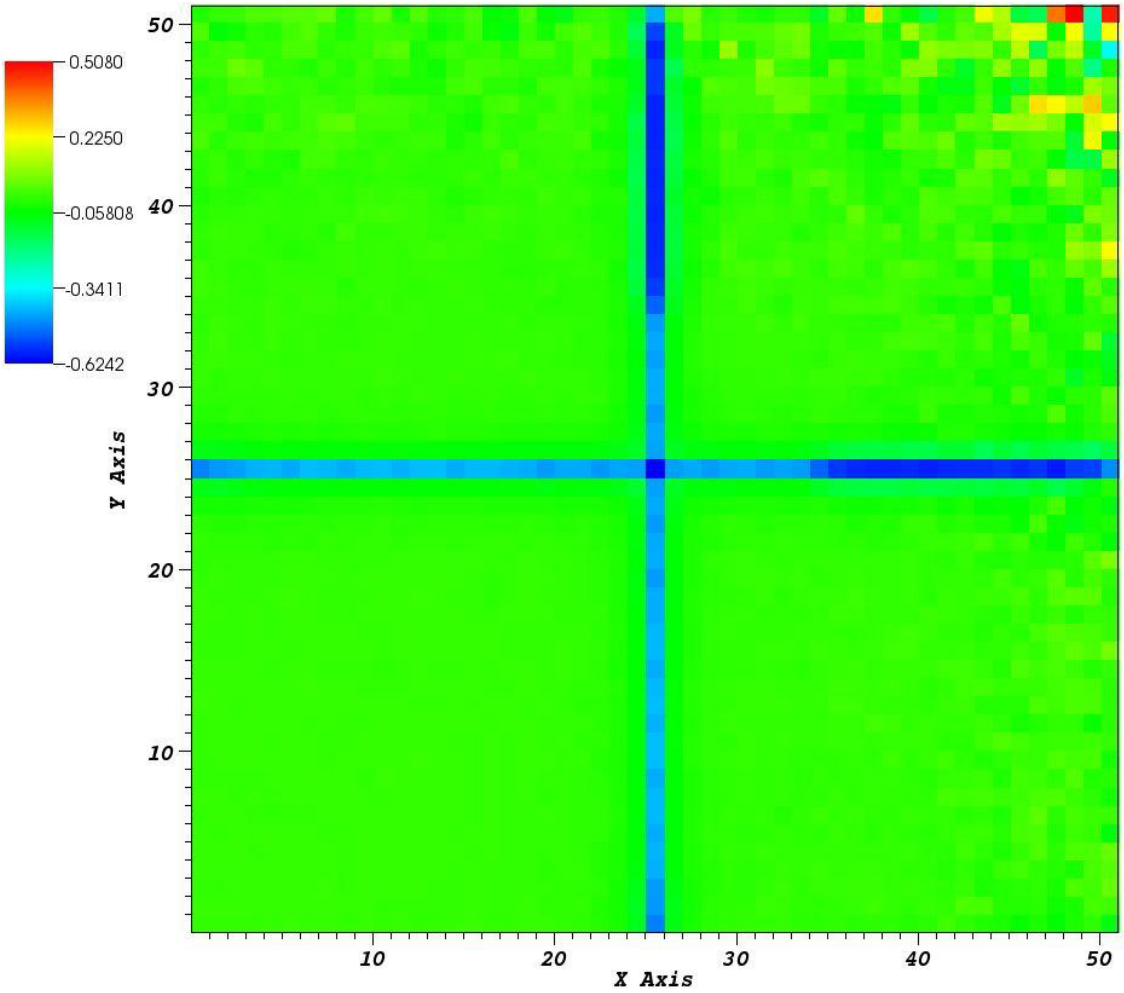
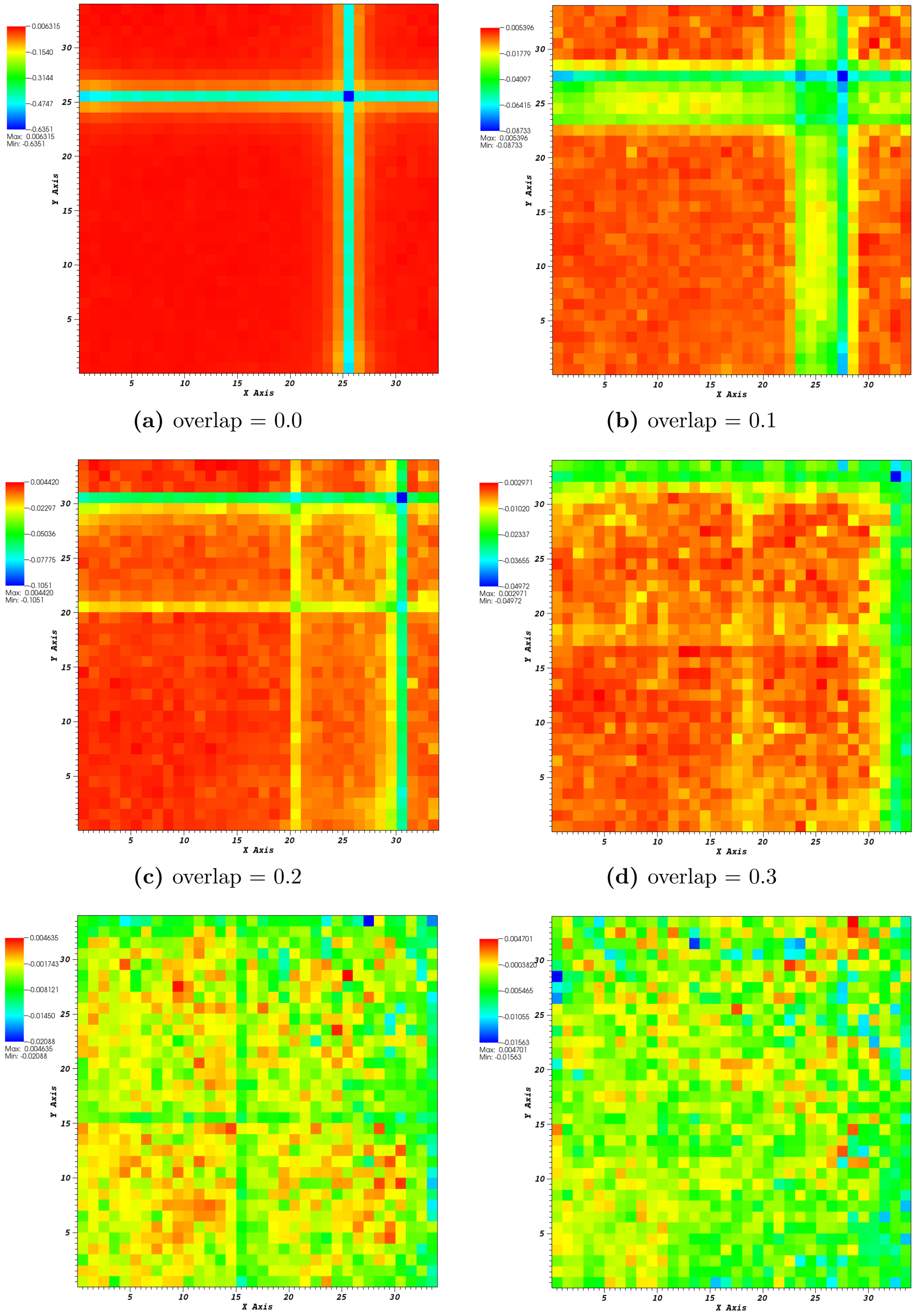


Figure 6.5: Relative difference in the variance for an axial slice of the full 3D C5G7 benchmark model.



(a) overlap = 0.0

(b) overlap = 0.1

(c) overlap = 0.2

(d) overlap = 0.3

(e) overlap = 0.4

(f) overlap = 0.5

Figure 6.6: Relative difference in the variance over the fuel pin cells for six simulations, each with a different overlap fraction.

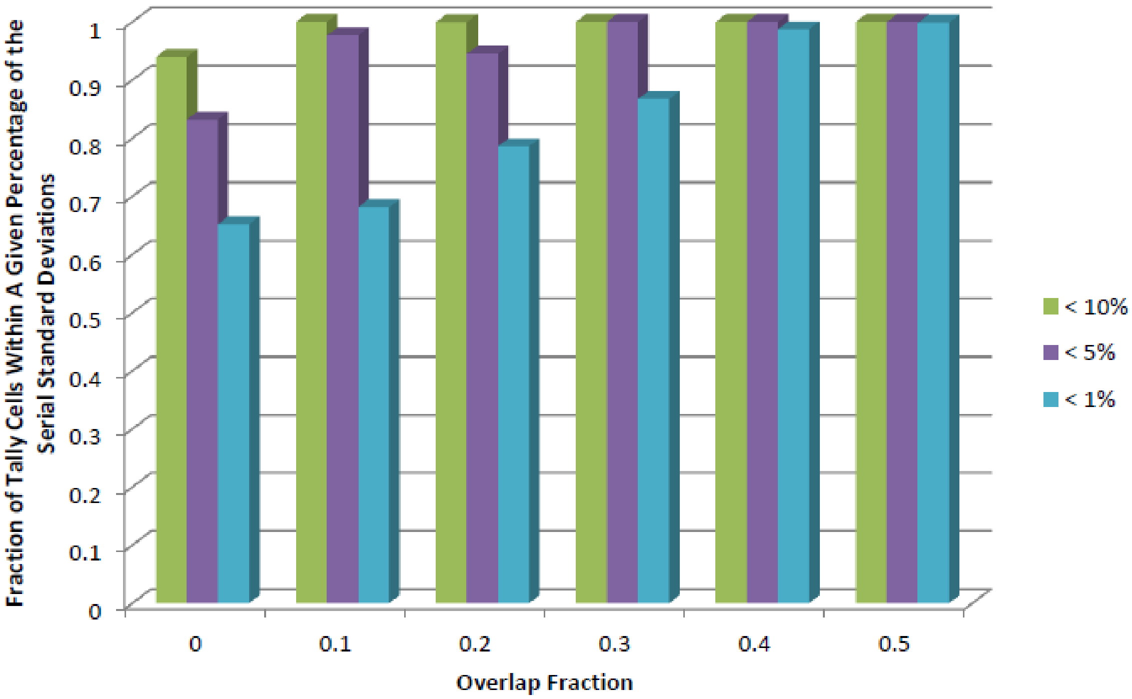


Figure 6.7: Fraction of mesh tally cells within a given percentage of the standard deviation reported by the serial simulation.

Chapter 7

Depletion Integration

In this chapter, the development and integration of depletion capabilities into the Exnihilo code suite are discussed. The discussion begins with a brief introduction and review of the coupled transport-depletion model. This is followed by a description of the recently refactored modular version of ORIGEN and the development work performed within SCALE as part of this research. This chapter concludes with a discussion on the development of the depletion package within Exnihilo, how it interacts with native Exnihilo and SCALE infrastructures, and how the depletion package couples to the Shift Monte Carlo framework to form an HPC-enabled module for integrated neutronics analysis.

7.1 Depletion Model

A major component of neutronics analysis for nuclear reactors involves coupling neutron transport and isotopic depletion capabilities. This coupling has been previously implemented in codes such as TRITON [28] and MONTEBURNS [25] and these codes have been continuously evolving and improving for over a decade. However, as nuclear engineering computational software is beginning to utilize HPC architectures to perform high-fidelity core analyses, many of these codes are ill-suited for scaling up to hundreds of thousands of processor cores, and it is difficult to retrofit entire codes to make them efficient for HPC applications. Although this is primarily directed at the neutron transport component, the depletion component as well as the coupling between the two components must be capable of handling large processor counts. At present, most – if not all – of the current computational packages which offer coupled transport-depletion capabilities cannot effectively utilize leadership class computing infrastructures. Therefore a depletion package was developed within the Exnihilo framework to meet the goal of providing coupled transport-depletion capabilities for HPC applications.

Coupled depletion capabilities were initially embedded within the Shift Monte Carlo package using the same strategy as the geometry and physics packages; an agnostic interface which allows multiple different depletion packages to be utilized transparently by Shift. However, as development progressed there was no reason why depletion capabilities could not be made accessible to Denovo or any other transport routine available within Exnihilo or other neutron transport packages as well. Furthermore, because a depletion package was not a necessary component when the development plan for Shift was designed, a considerable amount of additional infrastructure and conditional statements were required to directly integrate depletion capabilities into Shift. Thus, the depletion package was completely detached from the Shift code and made a primary component of the Insilico package within Exnihilo. A domain model of the fully-integrated transport-depletion capability is provided in Figure 7.1. The ORIGEN *Integrator*, ORIGEN *Aggregator*, and ORIGEN *Solver* in Figure 7.1 are abstract concept objects that generalize the flow of data in a coupled transport-depletion calculation. The resulting calculational steps performed to couple an Exnihilo transport calculation to an ORIGEN calculation are:

1. aggregate the necessary cross section and material data for each material in the system,

2. normalize the initial mass of heavy metals in the system to one metric tonne of heavy metal or a user-specified value,

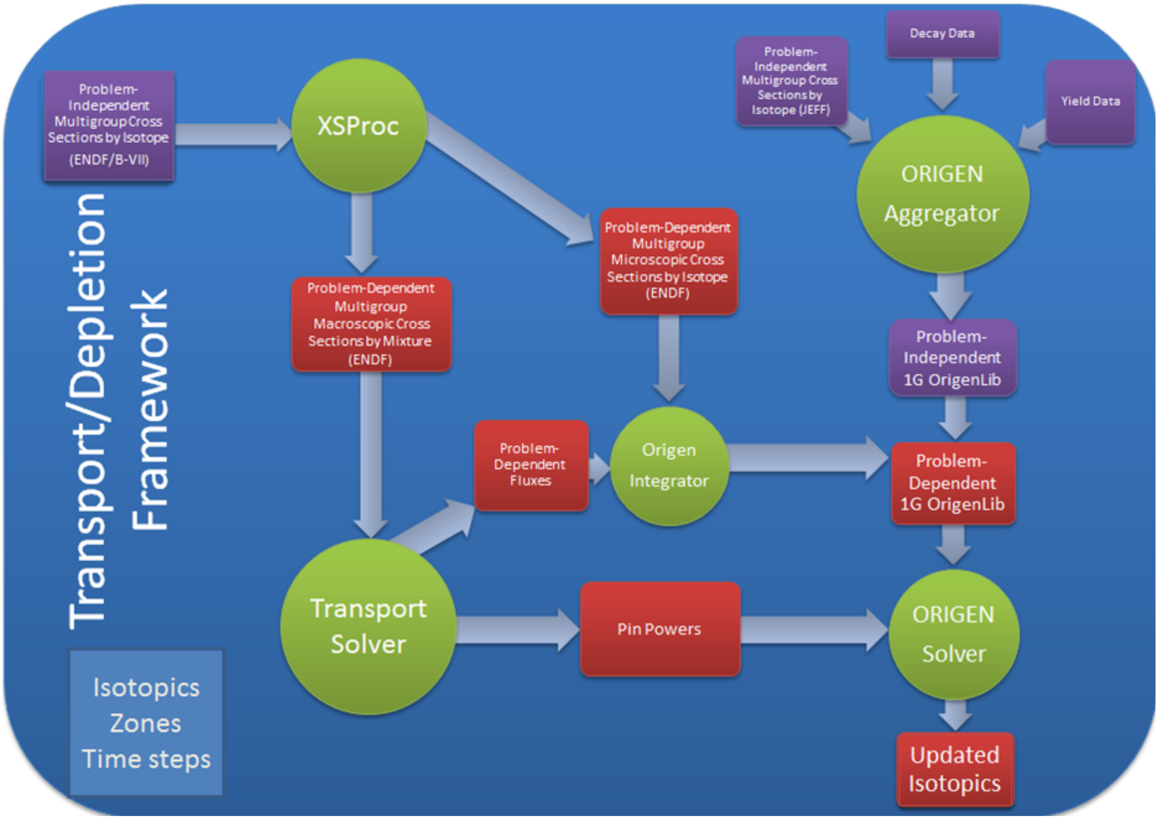

Transport/Depletion Framework
Problem-Independent Multigroup Cross Sections by Isotope (ENDF/B-VII)
XSProc
Problem-Independent Multigroup Cross Sections by Isotope (JEFF)
Decay Data
Yield Data
ORIGEN Aggregator
Problem-Dependent Multigroup Microscopic Cross Sections by Isotope (ENDF)
Problem-Dependent Multigroup Macroscopic Cross Sections by Mixture (ENDF)
Problem-Independent 1G OrigenLib
Problem-Dependent Fluxes
Origen Integrator
Problem-Dependent 1G OrigenLib
Transport Solver
Pin Powers
ORIGEN Solver
Isotopics
Zones
Time steps
Updated Isotopics

Figure 7.1: Generic domain model for a coupled transport-depletion calculation using ORIGEN in Exnihilo.

3. calculate the power in each material using the neutron flux solution provided by the transport calculation and normalize it to a user-specified average specific power in the basis material(s),

4. calculate the collapsed one-group cross sections for each material, using the neutron flux as the weighting function, for each nuclide and reaction which results in a transition from one nuclide to another and use these cross sections to populate the ORIGEN library, and

5. deplete each material using ORIGEN over the given time interval to produce new isotopic concentrations.

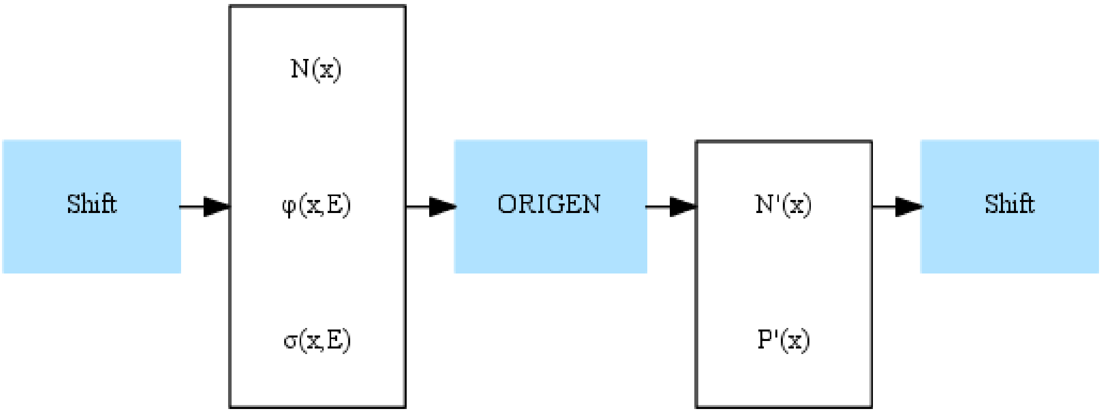


Figure 7.2: Input/Output data flow for a single-pass transport-depletion cycle.

In addition to performing all of the communication with the ORIGEN module, the depletion package is expanded to provide the additional components necessary to calculate the model power distribution and perform a flux-weighted one-group collapse of microscopic cross sections for each nuclide and reaction type relevant to depletion. These additional components allow the depletion package to provide the data necessary to drive the ORIGEN calculation. The depletion package currently supports cross section data in the form of AMPX [74] working-formatted libraries for multigroup depletion calculations or a handle to the SCEMPP (SCALE Continuous Energy Modular Physics Package) API for continuous-energy depletion calculations, however support for additional physics packages can be provided as they become available. Thus, the developed depletion package is structured such that any transport module capable of providing the required information can seamlessly integrate depletion capabilities with minimal additional development.

Two Python front-end user interfaces are available with the Insilico package; one for standalone depletion calculations and one for transport-coupled depletion calculations using Exnihilo's Monte Carlo framework, Shift. A simple I/O diagram of a single-pass Shift-ORIGEN calculation within Exnihilo is provided in Figure 7.2.

The development of an interface to ORIGEN as well as the coupling of ORIGEN and Shift required C++, Fortran, and Python code development in both the SCALE and Exnihilo code suites.

7.2 SCALE Development

The ORIGEN code has long been the premier code for performing isotopic depletion analyses for nuclear fuels. However, as the code was originally written in 1973 [95], the code architecture has been an increasing burden to ongoing development of depletion capabilities as SCALE is moving towards a more object-oriented framework. This has resulted in several modular versions of ORIGEN being developed and used in numerous codes that are not actively supported or maintained by Radiation Safety Information Computational Center (RSICC) [96][*]. Furthermore, the only means of interfacing with ORIGEN in many of these modular versions have been through file I/O, FIDO [97] input files, or global functions made accessible by embedding modified ORIGEN source code within a particular project application [35]. These coupling methods present significant difficulties for software developed for HPC applications such as Exnihilo where accessing data stored on disk becomes prohibitive.

Recent efforts outside of the research presented in this manuscript involved a major refactor of the ORIGEN code in the latest version of SCALE (6.2+) to provide a number of improvements to the code architecture making it more easily maintained and more easily integrated into other software packages. The restructured version of ORIGEN provides a single modular version following the object-oriented paradigm and accessible via C++ or Fortran APIs to allow for seamless integration with other codes [98]. The major constituents of this refactor included building two distinct components – an object which represents the ORIGEN library and an object which represents the ORIGEN solver – in which to house all of the data structures, porting

[*] See Section 2.3 for a discussion of modular version of ORIGEN that have been previously developed

the data structures from Fortran to C++, and establishing two APIs that link the newly created objects to the original Fortran algorithms. This approach succeeds in modularizing ORIGEN and allows it to be accessible from either C++ or Fortran.

7.2.1 Enabling ORIGEN for HPC Applications

Exnihilo is the first code to utilize ORIGEN's new C++ API for coupled transport-depletion calculations. However in order to enable the usage of ORIGEN in HPC applications, additional development was required to limit the amount of on-disk file I/O. In a given ORIGEN depletion calculation, all of the cross section and related data necessary to build the transition matrix ($\bar{\bar{A}}$ in Equation 3.25) is contained in a binary library file which is stored on disk and loaded during the initialization of the ORIGEN calculation. Thus in an HPC process one could have tens to hundreds of thousands of processors attempting to read this one file in off the disk, which is detrimental to parallel efficiency. Therefore, an alternate process for reading in the binary library data had to be developed in order to enable ORIGEN depletion calculations for HPC applications.

The ORIGEN binary library itself can be split into several components:

1. **Header**: scalar quantities such as the number of nuclides, number of transitions, and a list of the nuclide ZAIDs in the library,

2. **Decay Data**: vector data such as radioactive decay constants and recoverable energy values,

3. **Burnup Data**: vector data such as fission cross sections and neutron yields for each burnup position in the library[*],

4. **Transition Data**: vector data related to each transition such as the number of parents and the MT reaction identifier.

[*] A given ORIGEN binary library may contain data for a series of different burnup steps.

In order to enable ORIGEN for HPC applications, this data must be read off the disk by a single processor and then broadcast to the other processors using a parallel communication interface such as MPI. Because messages passed between processors must be native datatypes (i.e., int, char, float, etc.), the classes and the data within each class must be converted to a native datatype and packed into a single container to be broadcast to the other processors. Packing of the data must also follow a strict ordering criteria within the container so that the processors which are receiving the data know how to unpack it and reconstruct the binary library. Thus, the infrastructure for packing and unpacking each of the binary library components was developed along with the logic for packing and unpacking an ORIGEN binary library object. Furthermore, the OrigenLibraryContainer class was adapted to provide the means for loading in a binary library file and loading the different library positions instead of retrieving them directly from disk. These features have been successfully implemented in ORIGEN and will be in the next released version of the SCALE software package.

7.2.2 ARP Interface

Standalone ORIGEN calculations utilize a set of pre-generated ORIGEN libraries that are distributed with the SCALE package to build the transition matrix for the ORIGEN depletion calculation. Each of these distributed libraries is categorized based on assembly configuration, fuel enrichment, burnup, and a variety of other fuel properties and operating conditions [99]. However, these parameters are not a part of the ORIGEN binary library itself. The primary issue with this is that as the fuel is burned, a different library position may need to be used. Furthermore, interpolating between the libraries may be desired to obtain a more accurate cross section library for a given model.

The ability to both detect and interpolate between the distributed library files is available in the ARP package, however this package is written in Fortran and thus

Exnihilo cannot simply use it directly. Thus an ArpInterface package was developed in order to provide the means to obtain information about the different burnup positions in the distributed libraries and to interpolate between distributed libraries. The ArpInterface package modularized some of the features available in the ARP package and provides the C++ to Fortran binding layers such that this functionality can be accessed from either C++ or Fortran applications. Because only a limited usage of ARP is necessary, a functional interface was developed instead of a class interface since any given calculation will only require calling a single ArpInterface function once. This means that each time an ArpInterface function is called, it will create a new instance of ARP, perform the requested calculation, and destroy the ARP object before returning the results of the calculation instead of the leaving the ARP object in memory to be called upon for a subsequent calculation. This implementation aids in safeguarding the client from allocation issues and memory leaks as each new object and variable created when an ArpInterface function is called is destroyed before the function exits.

The two distinct functions that comprise the entirety of the ArpInterface package are *Arp_getBurnups* and *Arp_interpolate*. The *Arp_getBurnups* function is necessary to populate the list of library positions available in the distributed library files. Unfortunately, the ORIGEN package itself does not have any method to obtain the available burnup positions for a given ORIGEN binary library; however ARP has the functionality to obtain these values. A flow diagram of these two C++ functions and how they wrap the native Fortran operations is provided in Figure 7.3. Each function call involves creating an ARP Fortran object, reading a text file (located on disk) which contains all of the available information on the distributed libraries, saving the desired data to be passed back through the interface, and destroying the ARP Fortran object. Note also that the *Arp_interpolate* function is overloaded and will call either *F_arpInterpolateUO2* or *F_arpInterpolateMOX* depending on the number of input arguments, since interpolation between MOX libraries requires an additional argument.

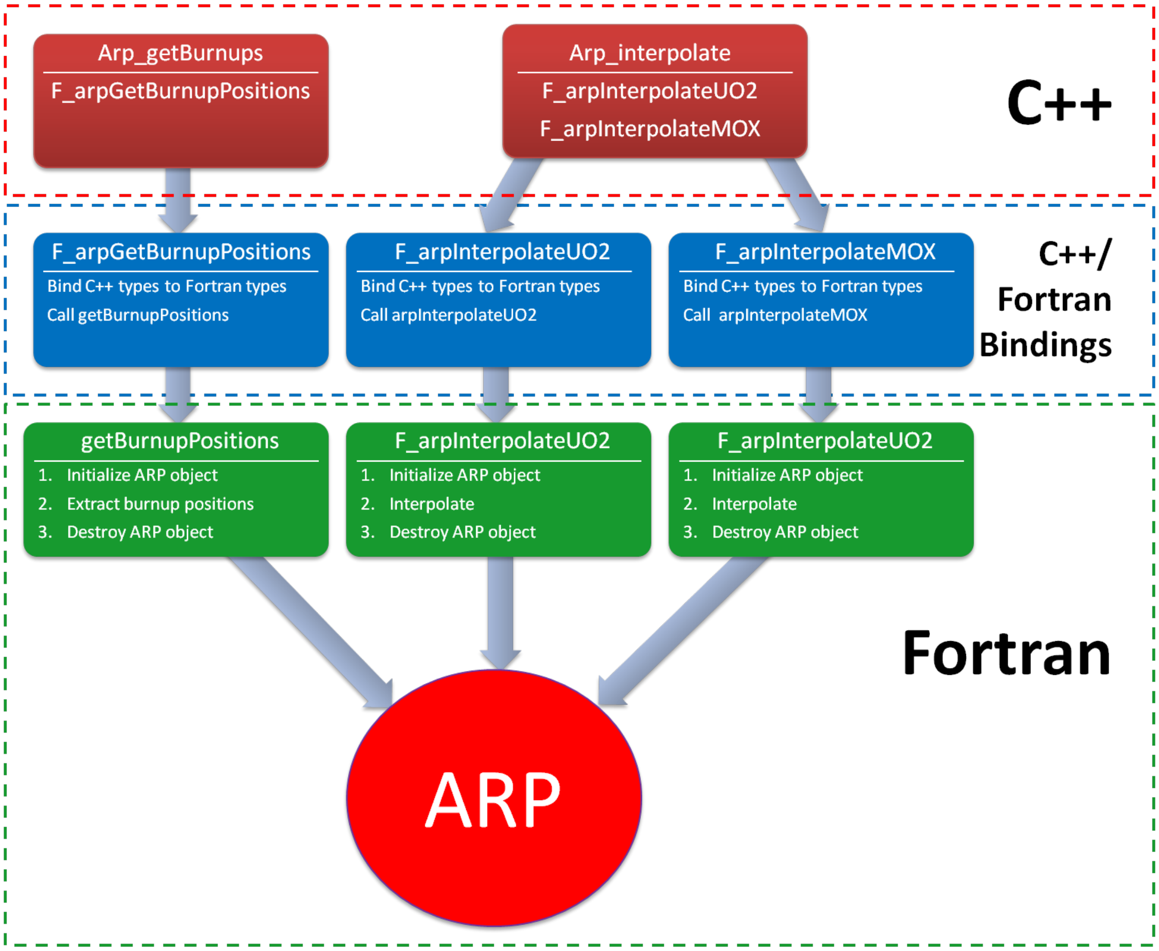


Figure 7.3: Functional diagram of the ArpInterface package developed for use by Exnihilo.

The ArpInterface package has been fully tested and integrated into the development version of SCALE. Since this package is not required by any of the primary control modules in SCALE, it is considered secondary stable code; only useful for developers.Future development will include annexing these functions directly into ORIGEN, therefore obviating the interface problem, as the ArpInterface package by itself has no front end in which the user can interact with.

7.3 Exnihilo Development

This section defines the general design and layout of the depletion package which integrates ORIGEN [4] depletion capabilities into the Exnihilo framework to calculate changes in isotopic concentrations over time. In addition to performing all of the necessary communication with the ORIGEN module, the depletion package calculates model power distributions and performs a flux-weighted collapse of microscopic cross sections for each nuclide and reaction type relevant to depletion. This package is structured such that any transport module capable of providing the required information can seamlessly integrate depletion capabilities with minimal additional development.

The depletion package in Exnihilo is essentially divided into three components:

1. **Material Processor**: collects and stores material data (i.e., power, temperature, volume) and nuclide data (i.e., nuclide ids, number densities, and cross sections) obtained from multiple different libraries and locations in both SCALE and Exnihilo.

2. **ORIGEN Coupling**: interface to the ORIGEN depletion module in SCALE.

3. **Transport Coupling**: interface which analyzes and applies the results from the transport calculation to a subsequent depletion calculation by performing the powers calculation and flux-weighted cross section collapse.

Support is currently available for cross section data in the form of AMPX working-formatted libraries [74] for multigroup depletion calculations or a handle to the SCEMPP (SCALE Continuous Energy Modular Physics Package) API for continuous-energy depletion calculations. However, additional support can be easily provided as additional physics packages become available within Exnihilo. A simplified flow diagram of the depletion package and how it interacts with certain SCALE components is provided in Figure 7.4. Each component and the primary functions performed by each component are explained in more detail in the following sections.

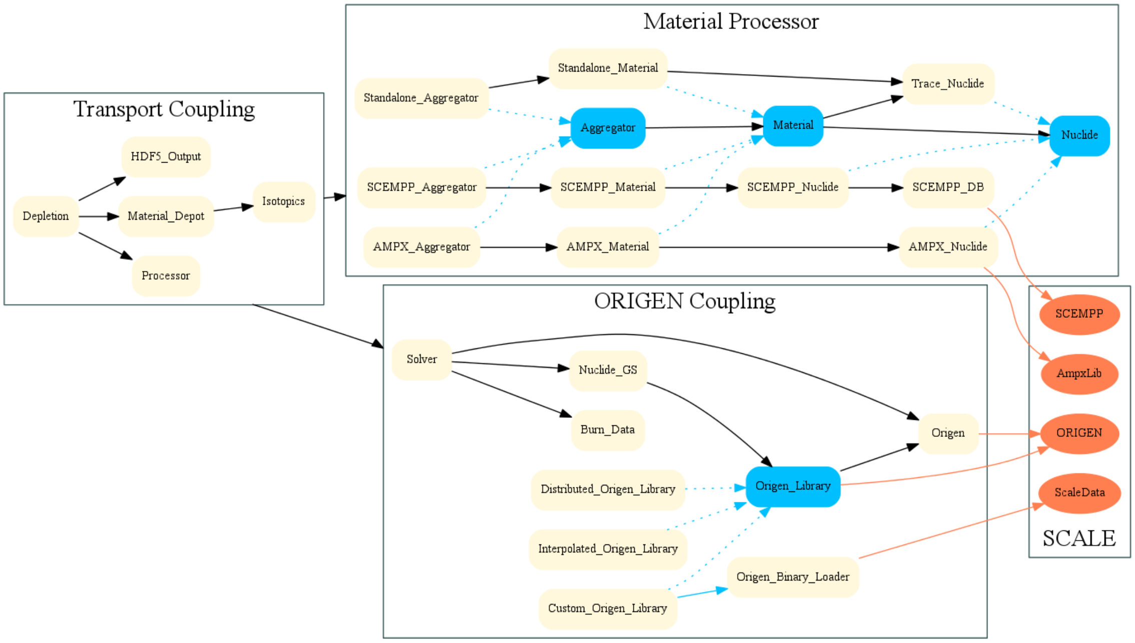


Figure 7.4: Flow diagram of the Exnihilo depletion package.

7.3.1 Material Processor

In order to perform transport post-processing calculations, such as the model power calculation and cross section collapse, a wide range of data for every material in the model must be accessible. These data are likely spread out in multiple different components as dictated by the transport and cross section processing components. Thus within the Exnihilo depletion package there exists a subset of classes designed to aggregate the necessary data from the different sources. Note that this requires the depletion package to have its own definition of what makes up a material and what makes up a nuclide. The primary motivation for this aside from the ease of access of data from a programming standpoint is that depletion calculations require microscopic cross section information, whereas the transport calculation typically uses macroscopic cross sections. Furthermore, this information needs to be accessible on a nuclide basis and not based on a particular material or spatial location as again

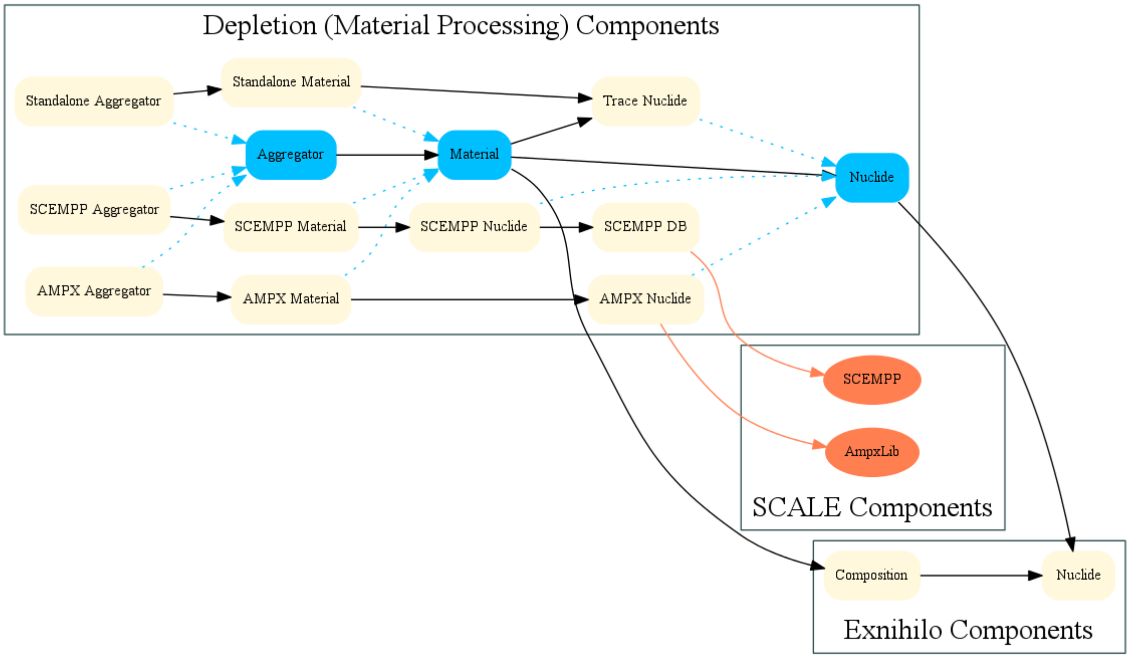


Figure 7.5: Data aggregation components in the depletion package.

is typical for neutron transport processes. A closer look at the data aggregation components and how they interact with the objects outside of the depletion package is provided in Figure 7.5.

Note that a given set of material processing components consists of an aggregator, one or more material definitions, and one or more nuclide definitions for each material. The material definition includes a list of nuclides as well as information related to the geometric cell in which the material resides such as the volume, power, and flux. The nuclide definition within the depletion package is designed to hold the nuclide ID, atomic mass, number density, energy release per fission, energy release per capture, and multigroup microscopic cross section information for the relevant reaction types for that nuclide. The specific set of aggregator, material, and nuclide components used depends on whether the calculation is standalone or coupled to a transport calculation. If it is a transport coupled calculation, the components are further subdivided based on the physics used in the transport calculation. This approach provides the various transport post-processing algorithms with centralized data containers which are both consistent and easily accessible. The consistency lies in the fact that the rest of the depletion package operates irrespective of whether the original cross section data was in a multigroup or continuous-energy format.

Aggregation of material and cross section data is currently available for both multigroup and continuous-energy applications. In the multigroup case, the depletion package requires an AMPX working-formatted library which contains microscopic cross section information by nuclide. Note that this cross section library file differs from the AMPX library used by the transport packages, which contains macroscopic cross sections by mixture. In the continuous-energy case, the depletion package requires a handle to the SCEMPP API. In addition, knowledge of the energy group structure used to perform the flux tally during the transport calculation must be explicitly provided to the depletion package in the continuous-energy case in order to generate multigroup microscopic cross section responses from the continuous-energy spectrum in the same group structure[*]. Alternatively, if the transport calculation is set up to tally all of the necessary one-group reaction rates required to populate the ORIGEN library, knowledge of the group structure is not required since the power calculation and group collapse has already been performed.

The aggregation of cross section data in the continuous-energy case includes an additional step, aside from the generation of the microscopic cross section responses in comparison to the multigroup case. This step involves creating a database of the generated multigroup cross section responses for each nuclide/temperature pair. The reason for this is to minimize the memory requirement for storing the cross section data. Since the microscopic cross section responses are generated directly from the continuous-energy spectrum (i.e. without any additional cross section processing), the generated responses for a given nuclide will be the same regardless of which material they belongs to. Considering that the number of energy groups for the

[*] In the multigroup case, the flux tally uses the same group structure as the AMPX library.

continuous-energy calculation (when using the flux-binning method) will likely be on the order of 10^4, the cross section data for each nuclide in the model are stored in a database as opposed to with each nuclide. Then, as the individual material nuclides are constructed, they are simply given the location of their cross section data in the database. This scheme significantly reduces the memory used to store the cross section data. For example, consider a 17 × 17 assembly model where 94 nuclides are being tracked in each fuel region and the flux tally is using a 43 000-group structure. If the data were simply generated and stored for each individual nuclide instance, the required storage for microscopic cross section data would be approximately 50 GB. By generating and storing the cross section data for each unique nuclide one time only, the memory requirement is reduced by approximately 99.6%. This is why there is an additional SCEMPP aggregation component in Figure 7.5 (i.e. the SCEMPP_DB).

Standalone Aggregation

The aggregation process for standalone depletion calculations is the simplest of the three sequences. Since no manipulation of cross section data is required by the depletion package, no cross section library is required when running in standalone mode. Furthermore, in a standalone calculation there is only one material. These two aspects narrow the aggregation process to simply creating a Material which is in essence identical to the Composition created by the user in the problem input. The primary way in which the aggregation process differs for standalone cases is that every nuclide present in the ORIGEN library is added to the Material definition so that it can be tracked. In coupled transport-depletion calculations only a subset of nuclides are added in order to reduce the memory footprint of the materials. A Universal Modeling Language (UML) diagram of the standalone material processing components is provided in Figure 7.6.

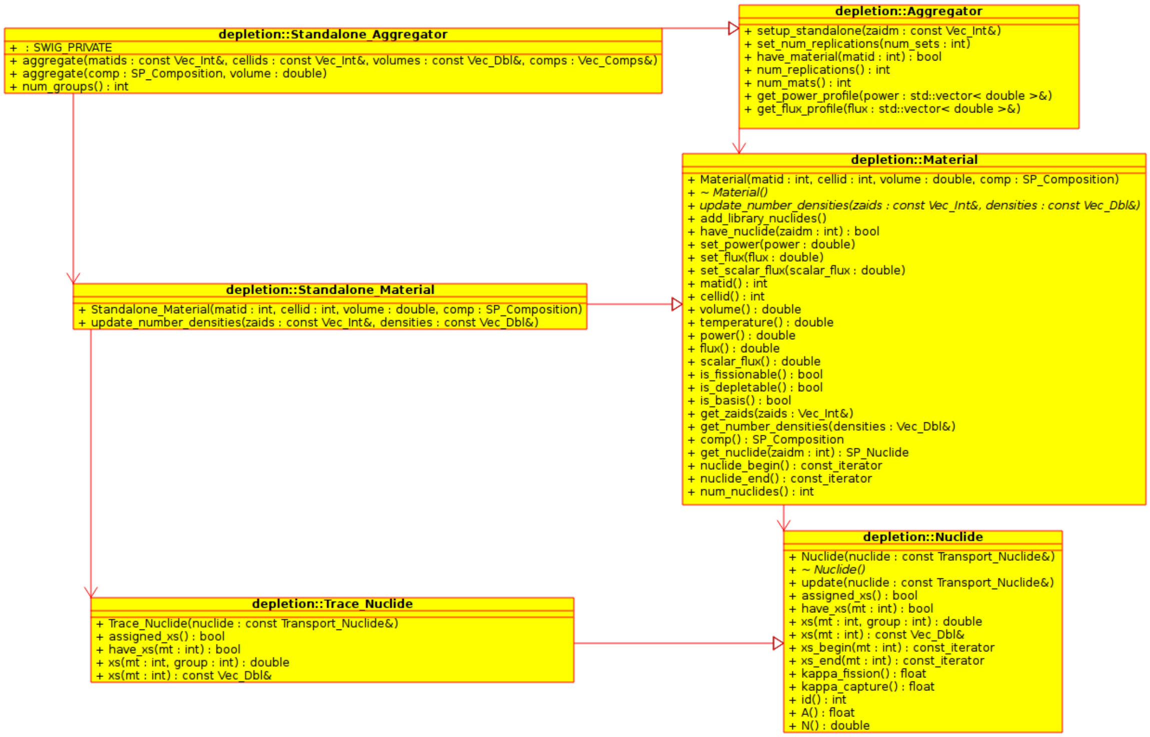

depletion::Standalone_Aggregator
+ : SWIG_PRIVATE
+ aggregate(matids : const Vec_Int&, cellids : const Vec_Int&, volumes : const Vec_Dbl&, comps : Vec_Comps&)
+ aggregate(comp : SP_Composition, volume : double)
+ num_groups() : int
depletion::Aggregator
+ setup_standalone(zaidm : const Vec_Int&)
+ set_num_replications(num_sets : int)
+ have_material(matid : int) : bool
+ num_replications() : int
+ num_mats() : int
+ get_power_profile(power : std::vector< double >&)
+ get_flux_profile(flux : std::vector< double >&)
depletion::Material
+ Material(matid : int, cellid : int, volume : double, comp : SP_Composition)
+ ~ Material()
+ update_number_densities(zaids : const Vec_Int&, densities : const Vec_Dbl&)
+ add_library_nuclides()
+ have_nuclide(zaidm : int) : bool
+ set_power(power : double)
+ set_flux(flux : double)
+ set_scalar_flux(scalar_flux : double)
+ matid() : int
+ cellid() : int
+ volume() : double
+ temperature() : double
+ power() : double
+ flux() : double
+ scalar_flux() : double
+ is_fissionable() : bool
+ is_depletable() : bool
+ is_basis() : bool
+ get_zaids(zaids : Vec_Int&)
+ get_number_densities(densities : Vec_Dbl&)
+ comp() : SP_Composition
+ get_nuclide(zaidm : int) : SP_Nuclide
+ nuclide_begin() : const_iterator
+ nuclide_end() : const_iterator
+ num_nuclides() : int
depletion::Standalone_Material
+ Standalone_Material(matid : int, cellid : int, volume : double, comp : SP_Composition)
+ update_number_densities(zaids : const Vec_Int&, densities : const Vec_Dbl&)
depletion::Nuclide
+ Nuclide(nuclide : const Transport_Nuclide&)
+ ~ Nuclide()
+ update(nuclide : const Transport_Nuclide&)
+ assigned_xs() : bool
+ have_xs(mt : int) : bool
+ xs(mt : int, group : int) : double
+ xs(mt : int) : const Vec_Dbl&
+ xs_begin(mt : int) : const_iterator
+ xs_end(mt : int) : const_iterator
+ kappa_fission() : float
+ kappa_capture() : float
+ id() : int
+ A() : float
+ N() : double
depletion::Trace_Nuclide
+ Trace_Nuclide(nuclide : const Transport_Nuclide&)
+ assigned_xs() : bool
+ have_xs(mt : int) : bool
+ xs(mt : int, group : int) : double
+ xs(mt : int) : const Vec_Dbl&

Figure 7.6: Class diagram of the material processing components for standalone depletion calculations.

AMPX Aggregation

The AMPX aggregation components aim to provide a pseudo-encapsulation of SCALE's AMPX working library class so that the depletion package has access to all of the necessary cross section information[*]. The AMPX_Aggregator is essentially the depletion packages representation of the AMPX working library. Using an opened AMPX working nuclide library, the AMPX_Aggregator builds an AMPX_Material using a Composition and the corresponding nuclide entry within the AMPX library for each nuclide in the material. The AMPX_Material then constructs an AMPX_Nuclide for each nuclide entry in the AMPX_Material. It is within the AMPX_Nuclide class where

[*] When dealing with data that is contained in an externally developed class or object, it is generally a good idea to copy the data internally and discard the external class/object. This serves to condense the locations where an implementation change in the external class could affect the functionality of the internal class. This is often referred to as pseudo-encapsulation.

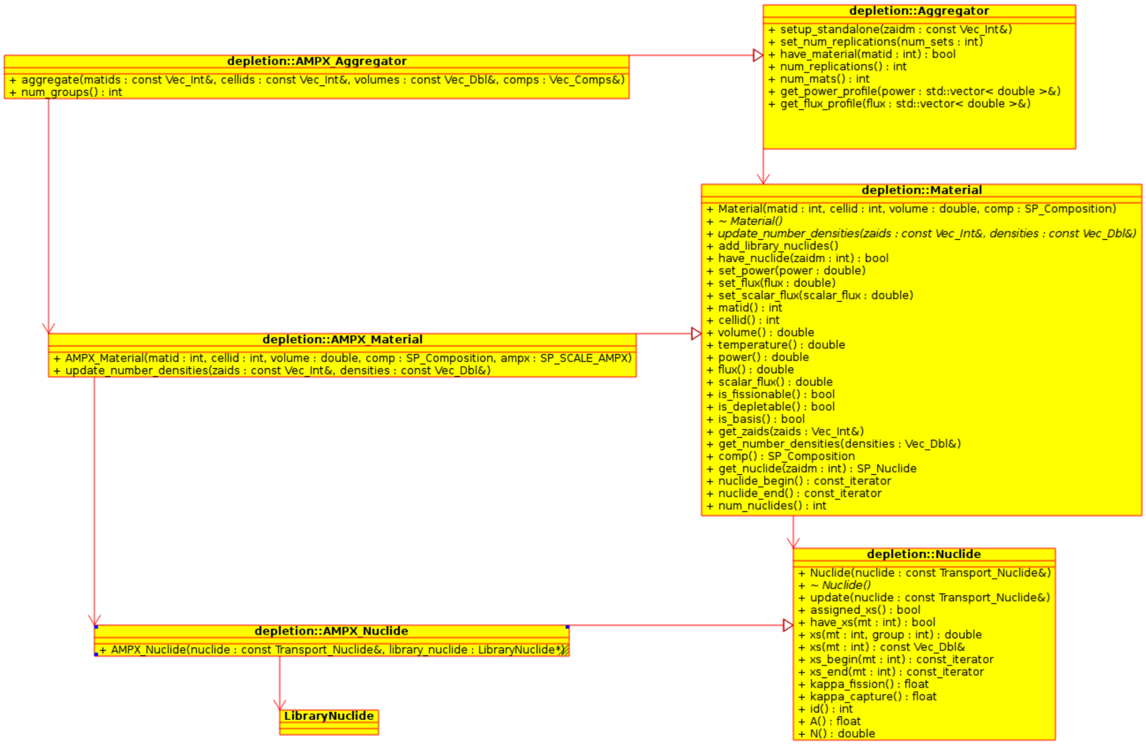

depletion::AMPX_Aggregator
+ aggregate(matids : const Vec_Int&, cellids : const Vec_Int&, volumes : const Vec_Dbl&, comps : Vec_Comps&)
+ num_groups() : int
depletion::Aggregator
+ setup_standalone(zaidm : const Vec_Int&)
+ set_num_replications(num_sets : int)
+ have_material(matid : int) : bool
+ num_replications() : int
+ num_mats() : int
+ get_power_profile(power : std::vector< double >&)
+ get_flux_profile(flux : std::vector< double >&)
depletion::Material
+ Material(matid : int, cellid : int, volume : double, comp : SP_Composition)
+ ~ Material()
+ update_number_densities(zaids : const Vec_Int&, densities : const Vec_Dbl&)
+ add_library_nuclides()
+ have_nuclide(zaidm : int) : bool
+ set_power(power : double)
+ set_flux(flux : double)
+ set_scalar_flux(scalar_flux : double)
+ matid() : int
+ cellid() : int
+ volume() : double
+ temperature() : double
+ power() : double
+ flux() : double
+ scalar_flux() : double
+ is_fissionable() : bool
+ is_depletable() : bool
+ is_basis() : bool
+ get_zaids(zaids : Vec_Int&)
+ get_number_densities(densities : Vec_Dbl&)
+ comp() : SP_Composition
+ get_nuclide(zaidm : int) : SP_Nuclide
+ nuclide_begin() : const_iterator
+ nuclide_end() : const_iterator
+ num_nuclides() : int
depletion::AMPX_Material
+ AMPX_Material(matid : int, cellid : int, volume : double, comp : SP_Composition, ampx : SP_SCALE_AMPX)
+ update_number_densities(zaids : const Vec_Int&, densities : const Vec_Dbl&)
depletion::Nuclide
+ Nuclide(nuclide : const Transport_Nuclide&)
+ ~ Nuclide()
+ update(nuclide : const Transport_Nuclide&)
+ assigned_xs() : bool
+ have_xs(mt : int) : bool
+ xs(mt : int, group : int) : double
+ xs(mt : int) : const Vec_Dbl&
+ xs_begin(mt : int) : const_iterator
+ xs_end(mt : int) : const_iterator
+ kappa_fission() : float
+ kappa_capture() : float
+ id() : int
+ A() : float
+ N() : double
depletion::AMPX_Nuclide
+ AMPX_Nuclide(nuclide : const Transport_Nuclide&, library_nuclide : LibraryNuclide*)
LibraryNuclide

Figure 7.7: Class of the material processing components for multigroup depletion calculations using an AMPX working library.

the cross section data relevant to the depletion package is stored. Once all of the required cross section information has been extracted from the AMPX library, the library is closed and discarded. This is acceptable since the AMPX working nuclide library is only used by the depletion package; the transport calculation uses an AMPX working material library. A UML diagram of the AMPX material processing components is provided in Figure 7.7.

SCEMPP Aggregation

The SCEMPP aggregation process is slightly more complex than the AMPX aggregation components because the depletion package is using a flux-binning methodology and a multigroup algorithm to calculate material powers and collapse cross sections down to one group, and the SCEMPP library only contains point-wise data. The process of building a Material for each Composition and a Nuclide for each nuclide in the Composition is generally the same. However, the Nuclide class itself is not storing any data. Instead, the SCEMPP_DB is storing the data and the Nuclide classes are simply querying for a pointer to the stored data.

The reason for this is because an ultrafine-group "response" is being generated for each relevant reaction for each nuclide in order to provide microscopic cross sections which can be used in the power calculation and in the cross section collapsing algorithms. Noting that these cross sections are microscopic cross sections that are being sampled from the continuous-energy spectra, the generated responses for, say, ^{235}U will be the same no matter what material it is in, provided that the temperatures are similar. Therefore, the SCEMPP_DB builds and stores the responses for each nuclide/temperature pair in the problem in order to prevent data duplication. A UML diagram of the SCEMPP material processing components is provided in Figure 7.8.

7.3.2 ORIGEN Integration

Integrating the ORIGEN module requires the ability to provide the OrigenLibraryContainer and OrigenContainer subpackages with the necessary data to solve the depletion equation (Equation 3.25). Interaction with the OrigenContainer package is a simple task which primarily consists of transferring data input by the user such as burnup intervals, power levels, and isotopic concentrations. The initialization of the OrigenContainer and execution of ORIGEN are then performed using the OrigenContainer's resource API. Interaction with the OrigenLibraryContainer subpackage is relatively more complex in nature; primarily in the case where the depletion calculation is being coupled to a neutron transport calculation. The OrigenLibraryContainer subpackage contains all of the burnup, decay, and other cross section data required to populate the transition matrix. All of this data is stored in a binary library file that must be loaded from disk.

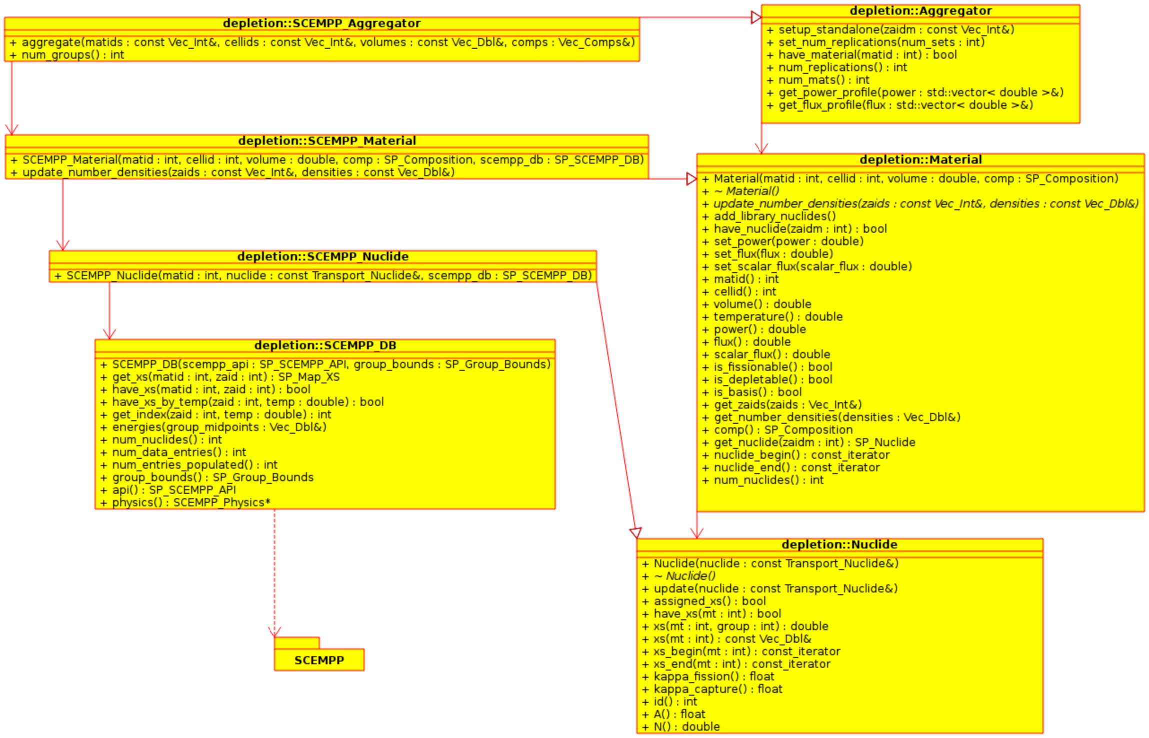

depletion::SCEMPP_Aggregator
+ aggregate(matids : const Vec_Int&, cellids : const Vec_Int&, volumes : const Vec_Dbl&, comps : Vec_Comps&)
+ num_groups() : int
depletion::Aggregator
+ setup_standalone(zaidm : const Vec_Int&)
+ set_num_replications(num_sets : int)
+ have_material(matid : int) : bool
+ num_replications() : int
+ num_mats() : int
+ get_power_profile(power : std::vector< double >&)
+ get_flux_profile(flux : std::vector< double >&)
depletion::SCEMPP_Material
+ SCEMPP_Material(matid : int, cellid : int, volume : double, comp : SP_Composition, scempp_db : SP_SCEMPP_DB)
+ update_number_densities(zaids : const Vec_Int&, densities : const Vec_Dbl&)
depletion::Material
+ Material(matid : int, cellid : int, volume : double, comp : SP_Composition)
+ ~ Material()
+ update_number_densities(zaids : const Vec_Int&, densities : const Vec_Dbl&)
+ add_library_nuclides()
+ have_nuclide(zaidm : int) : bool
+ set_power(power : double)
+ set_flux(flux : double)
+ set_scalar_flux(scalar_flux : double)
+ matid() : int
+ cellid() : int
+ volume() : double
+ temperature() : double
+ power() : double
+ flux() : double
+ scalar_flux() : double
+ is_fissionable() : bool
+ is_depletable() : bool
+ is_basis() : bool
+ get_zaids(zaids : Vec_Int&)
+ get_number_densities(densities : Vec_Dbl&)
+ comp() : SP_Composition
+ get_nuclide(zaidm : int) : SP_Nuclide
+ nuclide_begin() : const_iterator
+ nuclide_end() : const_iterator
+ num_nuclides() : int
depletion::SCEMPP_Nuclide
+ SCEMPP_Nuclide(matid : int, nuclide : const Transport_Nuclide&, scempp_db : SP_SCEMPP_DB)
depletion::SCEMPP_DB
+ SCEMPP_DB(scempp_api : SP_SCEMPP_API, group_bounds : SP_Group_Bounds)
+ get_xs(matid : int, zaid : int) : SP_Map_XS
+ have_xs(matid : int, zaid : int) : bool
+ have_xs_by_temp(zaid : int, temp : double) : bool
+ get_index(zaid : int, temp : double) : int
+ energies(group_midpoints : Vec_Dbl&)
+ num_nuclides() : int
+ num_data_entries() : int
+ num_entries_populated() : int
+ group_bounds() : SP_Group_Bounds
+ api() : SP_SCEMPP_API
+ physics() : SCEMPP_Physics*
depletion::Nuclide
+ Nuclide(nuclide : const Transport_Nuclide&)
+ ~ Nuclide()
+ update(nuclide : const Transport_Nuclide&)
+ assigned_xs() : bool
+ have_xs(mt : int) : bool
+ xs(mt : int, group : int) : double
+ xs(mt : int) : const Vec_Dbl&
+ xs_begin(mt : int) : const_iterator
+ xs_end(mt : int) : const_iterator
+ kappa_fission() : float
+ kappa_capture() : float
+ id() : int
+ A() : float
+ N() : double
SCEMPP

Figure 7.8: Class of the material processing components for continuous-energy depletion calculations using the SCEMPP physics package.

When running a depletion calculation, there are multiple different ways in which the binary library is loaded and used to initialize the OrigenLibraryContainer. By default, if no library file is specified in the problem input description, the depletion package will load a default library which is mutable and thus able to be used for a coupled transport-depletion calculation. Note that this is intended to be the primary mode of operation for coupled calculations. Alternatively, the filename of either a custom ORIGEN library generated from a TRITON calculation or one of the pre–generated ORIGEN libraries distributed with SCALE can be used. The primary difference to note here is that the distributed libraries are not mutable as they do not contain all of the necessary information required to update the cross sections. If a distributed library file is requested when running alongside a transport calculation, the burnup-dependent data (removal and fission cross sections, neutron yields, and the compressed transition matrix) which most accurately represents the model burnup for the midpoint of the current depletion cycle will be used*. For a list of the available libraries distributed with SCALE and the burnup positions available for each material, please consult Reference [70].

The third way in which ORIGEN library data can be initialized is to specify a given assembly configuration type and interpolate between the distributed libraries. This method generates a set of library data which most accurately resembles the burnup and operating conditions of each depletion cycle in comparison to using a single distributed library. Note however that the use of distributed and interpolated libraries is intended primarily for standalone depletion calculations and will be significantly less accurate than using the default ORIGEN library or a problem-specific ORIGEN library generated from a previous TRITON calculation. Figure 7.9 illustrates the relative error over time when using a distributed library, and interpolation between distributed libraries, and the actual problem-specific library generated by coupling to a transport calculation. This figure helps illustrate that while using distributed libraries is acceptable for standalone ORIGEN calculations, an ORIGEN library produced using the spectral results from a transport calculation produces results with far better accuracy.

Note that both of the classes and resource functions which make up the ORIGEN APIs are explicitly wrapped within the depletion package in order to ensure constant-correctness is enforced, provide additional error-checking, and expose only the necessary functions required to integrate ORIGEN depletion capabilities into the Exnihilo transport calculation.

7.3.3 Transport-Depletion Coupling

Coupling of the neutron transport and depletion processes is fundamentally a much more simple task from a mathematical standpoint in comparison to the individual

*The distributed ORIGEN libraries contain data at multiple different burnups. The library position is what is used to determine which set of burnup data to use in a given ORIGEN calculation.

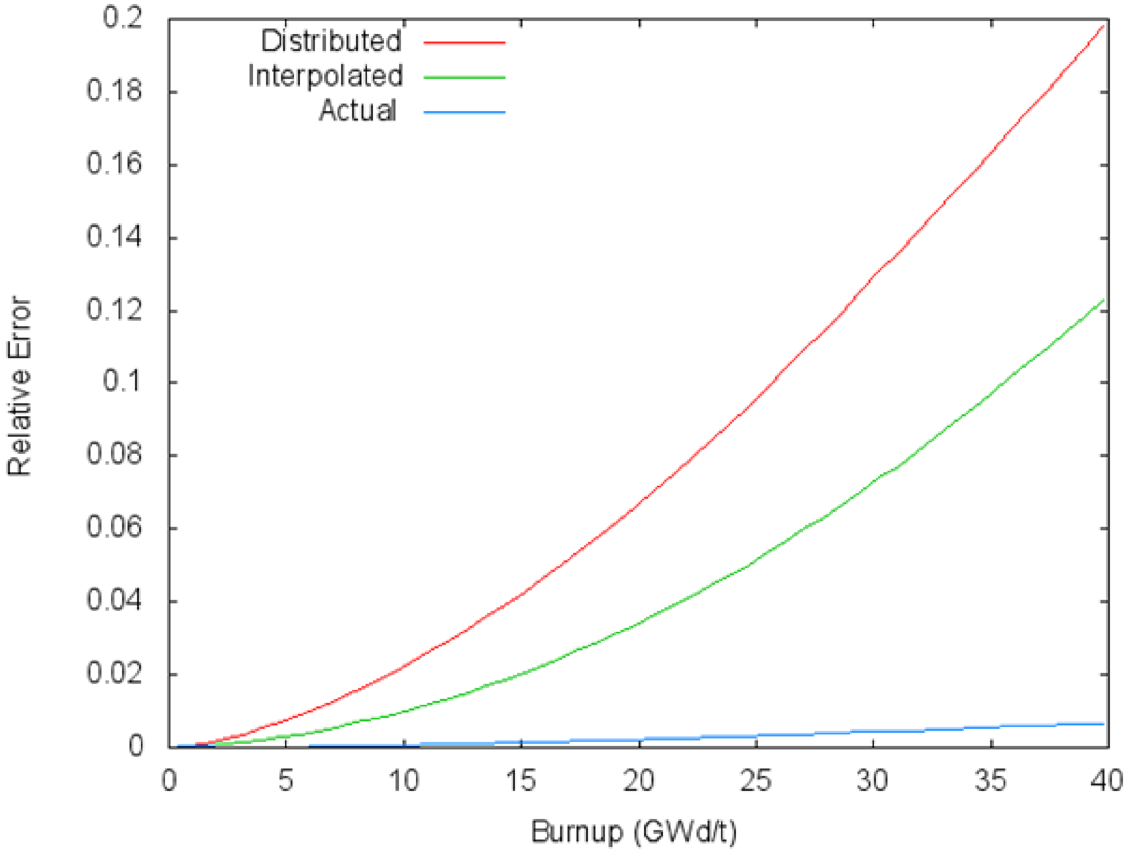


Figure 7.9: Relative error in the ^{235}U concentration using distributed, interpolated and an actual TRITON-generated ORIGEN library for 2.45% enriched 17×17 pin cell model.

processes themselves. However, having to couple multiple computational components that are built under different frameworks (i.e., SCALE and Exnihilo) makes this task difficult from a programming standpoint. A general flow diagram of the inputs and outputs for a single pass coupled transport-depletion calculation is presented in Figure 7.10.

where

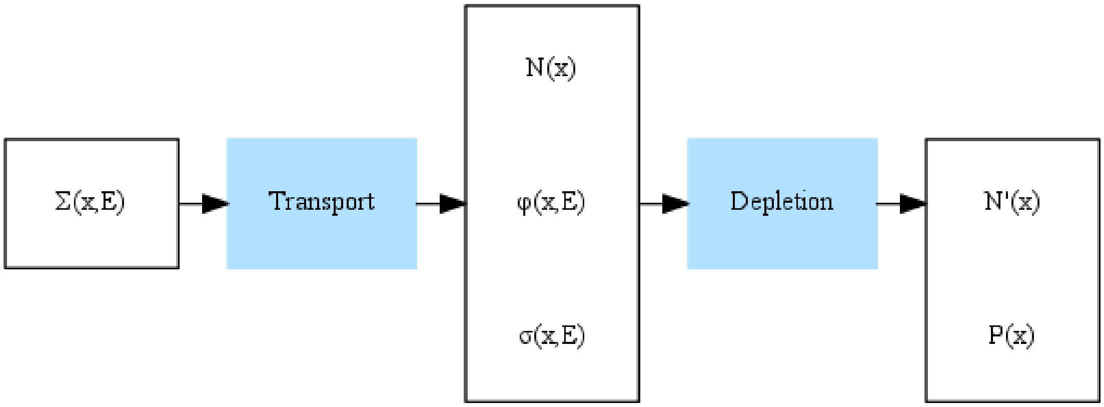


Figure 7.10: Input/Output diagram for a coupled transport-depletion calculation.

$\Sigma(x, E)$	energy-dependent macroscopic cross sections by material
$N(x)$	number densities for each material
$\phi(x, E)$	energy-dependent neutron flux in each material
$\sigma(x, E)$	energy-dependent microscopic cross sections by nuclide
$N'(x)$	updated number densities for each material
$P'(x)$	power in each material

The transport calculation simply supplies the energy-dependent neutron flux in each material region to the depletion calculation in order to calculate the one-group average powers which are used to deplete the materials. The remaining data required to perform the depletion calculation includes the nuclide concentrations, which are generally provided as input parameters to the transport calculation, and cross section data, which is supplied by the cross section processing calculation or directly from a continuous-energy spectrum.

Following a transport calculation, and once the necessary data has been obtained by the material processing components, the power in each material (both depletable and non-depletable) is calculated using

$$P = \sum_{ijk} \int_V d\mathbf{x} N_i \kappa_{i,j} \sigma_{i,j,k}(\mathbf{x}) \phi_k(\mathbf{x}) \tag{7.1}$$

where

N_i number density of nuclide i,

$\sigma_{i,j,j}$ cross section for nuclide i and reaction j in energy group k,

$\kappa_{i,j}$ the recoverable energy release for nuclide i and reaction j, and

ϕ_k neutron flux in energy group k,

The reaction types used to calculate the power include fission reactions and neutron capture reactions such as (n, γ) and (n, α) reactions. The power in the depletion materials is then normalized to a user-specified average specific power which represents the total power in the basis material(s). Since this depletion package is being developed for use in HPC applications, it is also important to note that this is the only location within the depletion package where parallel communication is required for each depletion calculation[*]. This is necessary in order to normalize the power distribution to a given set of materials which may be spread across multiple domains.

The next step involves generation of the one-group cross sections for each nuclide. In order to perform coupled transport-depletion calculations using ORIGEN, the binary-formatted ORIGEN library (which fundamentally represents the transition matrix) must be populated with problem-dependent one-group cross sections for each relevant isotopic transition. The relevant reaction types which make up the transition matrix include: (n, n'), $(n, 2n)$, $(n, 3n)$, (n, f), $(n, n'\alpha)$, $(n, 2n\alpha)$, $(n, n'p)$, $(n, n2\alpha)$, $(n, n'd)$, $(n, n't)$, (n, n'^3He), $(n, 4n)$, $(n, 2np)$, (n, γ), (n, p), (n, d), (n, t), $(n,^3 He)$, (n, α), $(n, 2\alpha)$, $(n, 2p)$, and $(n, p\alpha)$ [100]. This library is responsible for building the transition matrix which contains all of the radioactive disintegration constants and

[*] There is also a single communication performed during the calculation of the total mass of heavy metals in the system, but this calculation is only performed once during the initialization of the first depletion cycle.

branching fractions which result in the production or destruction of a nuclide. The one-group cross sections are collapsed using the neutron flux as the weighting function

$$\bar{\sigma} = \sum_{i=1}^{k} \frac{\phi_i \sigma_i}{\bar{\phi}} \tag{7.2}$$

where

$$\bar{\phi} = \int_0^\infty d\mathbf{E}\phi(\mathbf{E}) \tag{7.3}$$

This approach is available for both multigroup and continuous-energy calculations. As previously mentioned, the transport calculation can tally an ultrafine-group flux using a logarithmic group structure which contains tens of thousands of energy bins, and then responses can be generated on the same energy grid in order to obtain what can be treated as a multigroup cross section library. The alternative option of tallying the individual one-group nuclide cross section in each material instead of a flux tally is currently under development in Shift. Once this capability is available, the depletion package simply has to perform the power normalization and set the cross sections in the ORIGEN library, which significantly reduces the amount of transport post-processing performed by the depletion package.

The final step in coupling the depletion and transport calculations is determining the sequence in which each calculation is performed. The approach of using a predictor-corrector algorithm for isotopic depletion calculations has been implemented in multiple codes such as TRITON, VESTA [101], and CASMO [102]. A predictor-corrector approach involves a three-step process for solving the coupled transport-depletion calculation [103]:

1. A depletion calculation is performed from t_i to $t + \frac{\Delta t}{2}$ to predict the isotopic concentrations at the cycle midpoint.

2. The neutron transport calculation is performed to obtain the flux solution at the cycle midpoint, $t + \frac{\Delta t}{2}$.

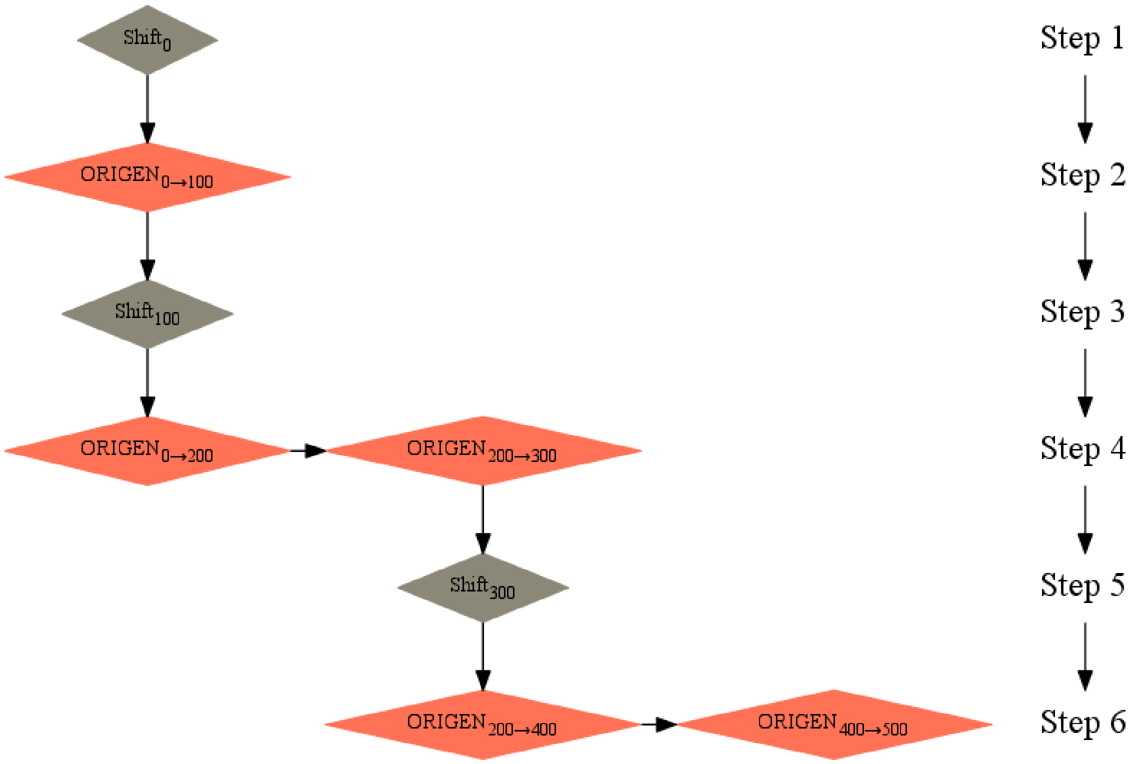


Figure 7.11: Single-cycle predictor-corrector algorithm for a coupled transport-depletion calculation in Exnihilo.

3. The flux and cross sections data at the cycle midpoint are used to perform the depletion calculation from t_i to $t + \Delta t$.

Thus a predictor-corrector approach is implemented in Exnihilo to increase the fidelity of the depletion calculation. This allows for larger time steps as average flux spectrum and cross sections for the cycle are used while the power of flux is being held constant over the time interval. A diagram of the predictor-corrector implementation is provided in Figure 7.11. The depletion calculation offers the ability to bypass the predictor-corrector algorithm if desired by the user, which is acceptable provided that the time steps are small, however the predictor-corrector is enabled as the default behavior for coupled transport-depletion calculations.

7.4 Implementation

The depletion package resides within Exnihilo's Insilico package, which is designed to couple cross-section processing, neutron transport, depletion, and thermal-hydraulics to enable complete neutronics analysis for nuclear reactor power applications. The design of the depletion package is intended for use alongside either Denovo, Exnihilo's deterministic transport package, or Shift, Exnihilo's hybrid/standalone Monte Carlo package.

In addition to the depletion package components illustrated in Figure 7.4, the pydepeltion package been developed to provide the Python front-end user interface to Exnihilo's depletion capabilities. The pydepeltion package provides the Python bindings generated by SWIG [104] that can be used to run a standalone or coupled transport-depletion calculation. In the standalone case there is a Python "runner" class which contains all of the functions the user will call to define and perform a standalone depletion calculation. In the case where depletion is coupled to Shift, a Python wrapper around Shift's Python front-end enables coupled transport-depletion calculations to be performed. This wrapper class is also responsible for setting up the tally or tallies required by the depletion calculation as well as implementing the predictor-corrector approach.

7.4.1 Standalone Depletion

The standalone depletion capability in Exnihilo is fundamentally the same as running a standalone ORIGEN calculation in SCALE. Some of the features available in the ORIGEN API are not yet integrated into Exnihilo's depletion package, such as continuous chemical processing and continuous nuclides feed rates, however these additional features can be added with little to no change in the infrastructure already in place.

One noteworthy feature that the depletion package has is the ability to automatically set the correct library position in the ORIGEN library based on the accumulated burnup over the course of the calculation when using the constant power approximation. Adding this feature involved developing the ArpInterface package: a C++ to Fortran API to SCALE's Automatic Rapid Processing module (ARP) which is designed for performing point depletion calculations for a variety of reactor configurations and operating conditions. ARP has the ability to both read the available burnup positions in the distributed libraries as well as interpolate between libraries to generate a set of problem-dependent ORIGEN libraries to be use in subsequent depletion calculations. The ArpInterface package provides Exnihilo and any other depletion module with a means of accessing this functionality. Note however that this package, in addition to the ORIGEN package, are required in order to perform standalone depletion calculations within Exnihilo.

7.4.2 Coupled Transport–Depletion Calculations in **Exnihilo**

The depletion package is designed such that it can be used by any transport module capable of supplying the following information for each material:

1. material identification number,
2. cell identification number,
3. cell volume,
4. composition data (nuclide ZAIDs and number densities),
5. cross section data (via SCEMPP or an AMPX working-formatted nuclide library), and
6. multigroup flux

The various simulation control flags and depletion-specific parameters such as burnup interval length, decay interval length, and the average specific power (in megawatts per metric tonne initial heavy metal) in the basis material(s) are generally defined by the user in the input specification. Note that the burnup and average specific power in each material to be depleted are the only input parameters required to perform isotopic depletion calculations using this depletion package.

A summary of the available features and implementations which exist within the depletion package include:

- Depletion via the constant power or constant flux approximation.
- Toggle to turn implicit capture off for the model power calculation.
- Five different groups of nuclides to be tracked for each depletable material. For a list of the different sets of nuclides that can be added to those tracked, see Tables T1.3.2 - T1.3.6 in Reference [28][*]. Note that one additional group is present which includes all nuclides recognized by both ORIGEN and SCALE's Standard Composition Library [69].
- Ability to use pre-generated ORIGEN binary libraries (generated using TRITON or one of the libraries distributed with SCALE) or a set of libraries obtained by interpolating between the distributed libraries (use the functionality native to ARP) distributed with SCALE.
- Ability to read in the ORIGEN binary library from disk on a single processor and broadcast the data to all other processors to reduce file I/O in HPC applications.
- Specification of the total mass of heavy metals which the system should be normalized to.
- Specification of the maximum size of the subintervals for ORIGEN burnup calculations (by default each burnup step is divided into substeps of no more than 40 days).
- Specification of the number of subintervals per burn step.

[*] Table D.2 contains the nuclides added too all depletable materials by default.

- Specification of the maximum burn length in coupled transport-depletion calculations (by default, the maximum burn length is 400 days; if a burn step is larger than 400 days, it is N equal-sized subintervals which are all less that 400 days).

- Rule of 3s used for decay calculations using an initial subinterval size less than or equal to 75 days [70]. Decay calculations require at least 3 subintervals regardless of the size of the decay step.

- Supports domain-replicated and domain-decomposed geometries.

- Option to parallelize the depletion calculation by distributing materials that are replicated across multiple processors such that each material is only being depleted once (by default each processor will deplete all materials present on the processor).

- Supports multigroup calculation using cross sections in an AMPX working library format or continuous-energy calculation using cross section data obtained via the SCEMPP API.

- Predictor-corrector approach for performing coupled transport-depletion calculations [28].

- Support for branch calculations to examine the response from perturbing one or more system properties or operating conditions (infrastructure is in place within the depletion package but not implemented in the front-end code).

- Powers, fluxes, and isotopic concentrations for each time step can be output in an Hierarchical Data Format (HDF5)-formatted file [105].

Of these available features, the parallelization of the depletion calculation merits further discussion. The computational time required to perform an ORIGEN depletion calculation for a single material is typically on the order of a second or less, which is significantly smaller than the computational time required to perform the transport calculation. However, as the number of material regions increase so will the total time spent performing depletion calculations. Consider the assembly model presented in Figure 7.12. This model has 264 fuel materials which need to be depleted. Using the predictor-corrector implementation and assuming that it takes one second to deplete a single material, depleting 264 materials requires 8.8 minutes of wall time. Now consider that 24 processor cores are used to perform the transport calculation using domain replication. If the depletable materials are distributed among these 24 processors, that equates to 11 depletable materials per core and thus only 22 seconds performing depletion calculations during each time step. This algorithm does however incur communication costs since each processor must have the updated isotopic number densities for all materials on that processor. Considering that each processor will only have updated isotopics for 24 of the 264 materials, and assuming that each material is composed of 94 different nuclides, each processor will have to communicate approximately 0.4 MBs of data at the end of each depletion cycle.

Further analysis of the communication costs and the scalability of this algorithm are discussed in Section 8.6.2.

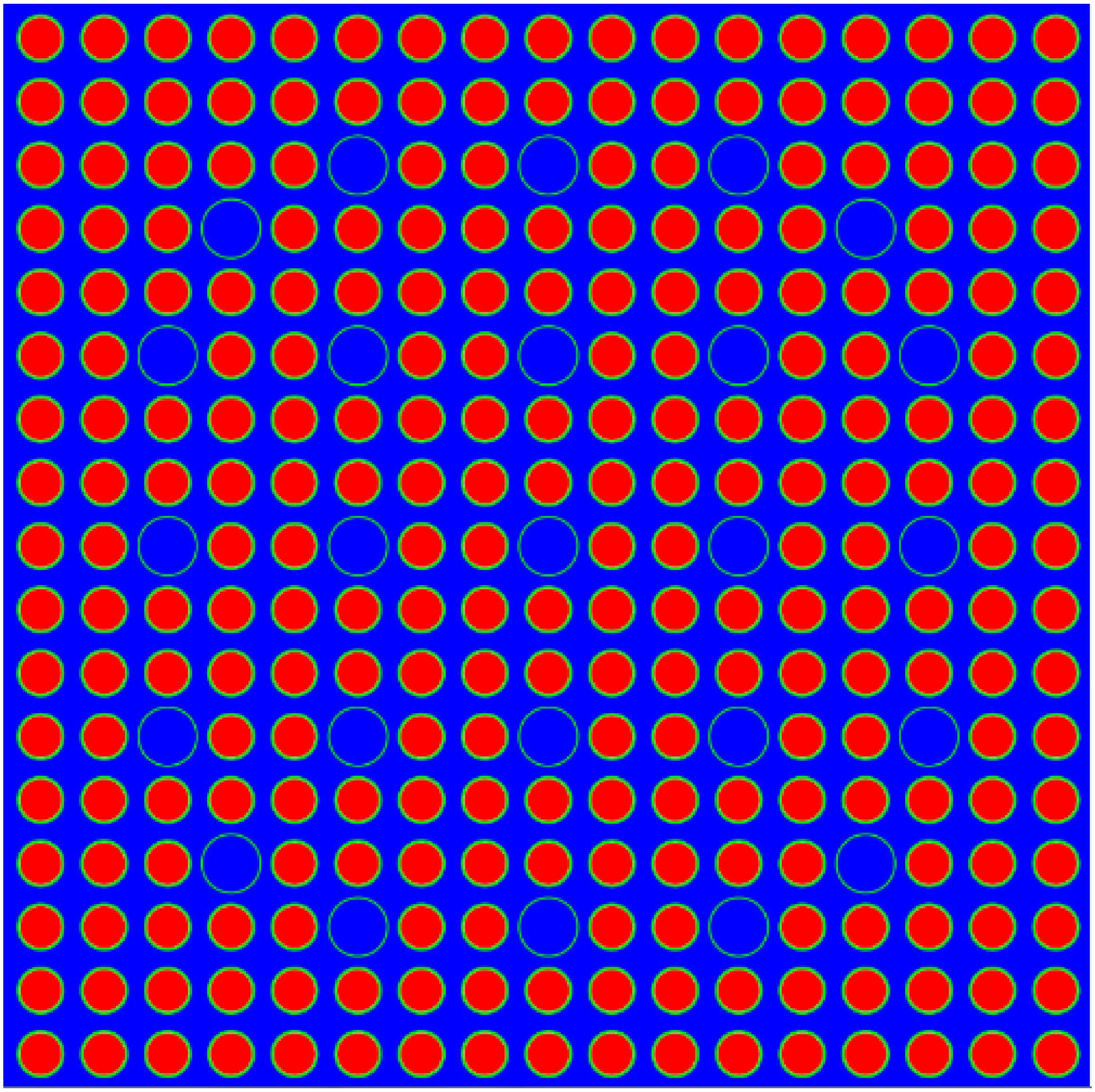


Figure 7.12: Generic Westinhouse 17 × 17 assembly model.

Chapter 8

Depletion Verification and Validation

This chapter presents the results of several benchmark cases which serve as verification and validation of Exnihilo's depletion package. Standalone depletion calculations within Exnihilo are benchmarked against standalone ORIGEN calculations to verify the integration of the ORIGEN module into Exnihilo, and the processes which govern transport-coupled depletion calculations are benchmarked against TRITON to provide verification and validation of coupled Shift-ORIGEN calculations in Exnihilo. Note that the analyses performed in each test category (standalone depletion, multigroup transport-coupled depletion, and continuous-energy transport-coupled depletion) were used as an incremental testing framework during the development of the depletion package. Thus the test cases begin in an overly-simplified state and gradually increase in complexity to test individual aspects of the depletion and coupling processes.

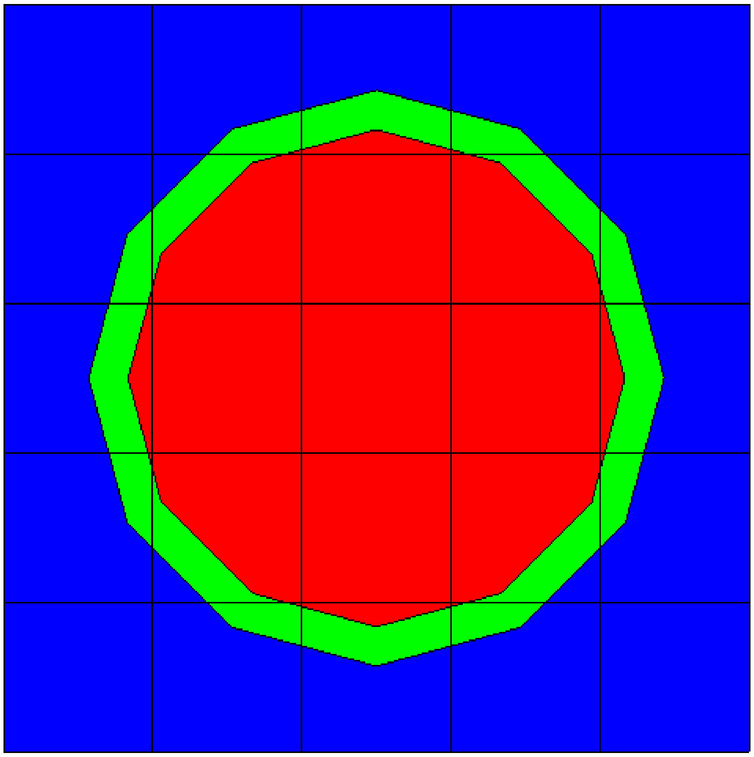


Figure 8.1: Graphical representation of the pin cell model generated by SCALE.

8.1 Verification Models

To analyze and verify the implementation in place for coupled transport-depletion calculations, two benchmark cases were developed: a pin cell model and a 2×2 array model. The pin cell model is illustrated in Figure 8.1 and features a fuel radius of 0.41 cm, cladding outer radius of 0.475 cm, and pitch of 1.26 cm. A temperature of 298 K is used for all materials in the pin cell model. The isotopic specifications for the fuel, clad, and moderator materials along with a graphical illustration of the model can be found in Table B.1 in Appendix B.1.

The second model is a 2×2 array model where each pin cell has a pitch of 1.26 cm but different fuel radii and clad outer radii. A graphical illustration of this model is presented in Figure 8.2. The materials, fuel and cladding radii, and the temperatures

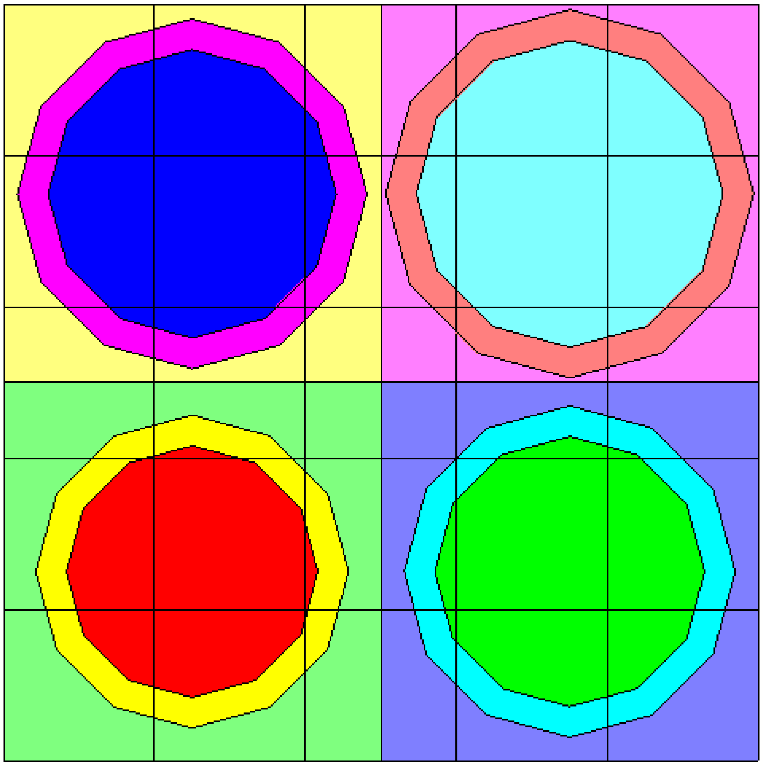


Figure 8.2: Graphical representation of the 2 × 2 assembly model generated by SCALE.

for each of the four pin cells in the 2 × 2 array model are provided in Table B.5 in Appendix B.2 along with a graphical representation of the model and the isotopic concentrations of each of the materials. Note that the material labels referred to in Tables B.5–B.6 are the labels provided in the legend of Figure 8.2.

The pin cell and 2 × 2 array models contain specific anomalies such as trace amounts of ^{242}Am in the fuel and ^{166m}Ho in the moderator in order to test various aspects of the developed implementation and the interactions with Exnihilo modules with SCALE modules. Thus the results presented in Sections 8.3–8.5 are more for verification purposes than validation. Note also that while these models are two-dimensional models, the depletion calculation is not concerned with the geometry or the number of dimensions; only the volume which is used to calculate the total mass of heavy metals in the system.

A third model was developed for performing scaling studies and to analyze some of the implementations of integrated depletion capabilities in Exnihilo. This model is a three-dimensional “block” of pin cells; each pin cell having a fuel radius of 0.41 cm, pitch of 1.26 cm, and height of 1.0 cm. In order to reduce the memory footprint for continuous-energy calculations, the fuel is composed of ^{16}O with a number density of 4.57642×10^{-2} atoms per barn-cm (atom/b-cm) and ^{235}U with a number density of 7.18132×10^{-4} atom/b-cm, and the moderator is pure ^{16}O with a number density of 2.48112×10^{-2} atom/b-cm. The number of pin cells in the x-, y-, and z-directions is varied to produce the desirable number of depletable regions for a given analysis,

Note that all cases simulated as part of the analyses presented in this section were performed on the same computational cluster and both the SCALE calculations and the Exnihilo calculations used the same revision of the development version of SCALE.

8.2 Standalone Depletion

The first set of test cases serves as a benchmark analysis of Exnihilo’s standalone depletion capability using the Python front-end. In the standalone case, the resulting isotopic concentrations produced by Exnihilo are expected to be within the numerical precision of a standalone ORIGEN calculation, which is 5×10^{-5}[*] for nuclides with a concentration above 1×10^{-6} atom/b-cm [4]. The results for five select nuclides which are important for burnup analysis and benchmarking validation–^{90}Sr, ^{131}Xe, ^{145}Nd, ^{235}U, and ^{239}Pu–are used as the basis for comparison in these benchmark analyses. These nuclides were chosen because they all have concentrations above 1×10^{-6} atom/b-cm, therefore guaranteeing that the relative error between the ORIGEN and Exnihilo calculations should be within the 5×10^{-5} tolerance. Although traditionally

[*] In theory, the results should be within the precision of the ORIGEN output since ORIGEN is used in both cases and thus both cases would suffer equally from the uncertainties in the nuclear data and numerical methods.

more nuclides are used to validate a depletion code, this analysis is more concerned with verifying the proper use of ORIGEN within Exnihilo. Since the underlying solver is ORIGEN in either case and because ORIGEN has been extensively validated, both verification and validation of the standalone depletion capabilities within Exnihilo can be supported with only a subset of these nuclides.

The test cases presented in Sections 8.2.1–8.2.5 are fairly simple with respect to the initial isotopic concentrations and burnup cycle parameters, however they fulfill the necessary requirement for properly verifying standalone depletion calculations in Exnihilo by independently examining specific capabilities or features. The first four test cases, 1-1–1-4, use the *w17_e30.arplib* pre-generated ORIGEN library distributed with SCALE, which represents a Westinghouse 17 × 17 PWR assembly configuration with 3% enriched UO_2. The fifth test case interpolates between the available Westinghouse 17 × 17 PWR libraries to match a specific set of operation conditions and burnup parameters. Note that in order to run the depletion calculation in the same manner as standalone ORIGEN, the maximum step size of 40 days is overridden to ensure that the depletion steps were not subdivided into smaller step sizes.

8.2.1 Case 1.1: Depletion by Constant Power

The first test case is designed to test standalone depletion using the constant power approximation. Five 10-day time steps at a power level of 50 MW/MTIHM are used to deplete a simple UO_2 composition. The initial isotopic concentrations for UO_2 are provided in Table 8.1 while Table 8.2 contains the relative differences between resulting number densities from an Exnihilo calculation and a standalone ORIGEN calculation for select nuclides. The results show an agreement that is within the numerical precision of the ORIGEN calculation, which is 1×10^{-5}. This is the expected result since there is no difference in the calculation from an algorithmic standpoint given that ORIGEN is the solver in both scenarios.

Table 8.1: Isotopic composition for case 1.1.

ZAID	Concentration (atom/b-cm)
^{16}O	3.239912×10^2
^{235}U	7.708341×10^1
^{238}U	2.330569×10^3

Table 8.2: Absolute difference (atom/b-cm) between results from a standalone Exnihilo depletion calculation and results from a standalone ORIGEN calculation for select nuclides for case 1.1.

Step	^{90}Sr	^{131}Xe	^{145}Nd	^{235}U	^{239}Pu
1	1.3290e-08	2.7961e-08	1.7345e-08	2.7466e-05	9.5495e-08
2	1.2754e-09	4.4806e-08	3.2444e-08	2.6971e-05	1.0506e-07
3	1.6775e-07	9.5933e-08	3.4562e-07	1.5137e-05	2.1122e-06
4	1.5720e-07	2.4975e-07	1.0973e-07	3.0988e-05	2.8948e-07
5	6.6425e-08	1.3763e-07	1.8657e-07	3.9889e-06	1.7493e-06

8.2.2 Case 1.2: Depletion by Constant Flux

Case 1.2 is identical to case 1.1 with respect to the initial isotopics and time step sizes. The only difference is that the depletion calculation uses the constant flux approximation. A constant flux of 1.0×10^{14} is applied to each of the five 10-day burnup cycles. The relative difference between the Exnihilo and ORIGEN calculations is shown in Table 8.3 and is within the numerical precision of ORIGEN.

8.2.3 Case 1.3: Complex Burnup Cycles

Case 1.3 is designed to test a more complex burnup sequence. In this case, the powers and time intervals vary for the different time steps. The same initial isotopic concentrations as cases 1.1 and 1.2 are used in this test case. The burnup parameters are provided in Table 8.4 and the results for select nuclides are presented in Table 8.5.

Table 8.3: Absolute difference (atom/b-cm) between a results from a standalone Exnihilo depletion calculation and results from a standalone ORIGEN calculation for select nuclides for case 1.2.

Step	^{90}Sr	^{131}Xe	^{145}Nd	^{235}U	^{239}Pu
1	2.5140e-08	1.5443e-10	1.8209e-08	1.6284e-05	2.3415e-07
2	1.4077e-09	4.9511e-07	1.2386e-08	8.9340e-06	2.6461e-07
3	2.4932e-08	9.0372e-07	1.1637e-08	2.2961e-05	1.0466e-07
4	1.6741e-08	1.0959e-06	1.2514e-08	2.4927e-05	2.5410e-07
5	2.7633e-08	1.2166e-06	2.5202e-08	7.5952e-06	2.6771e-07

Table 8.4: Time step data for case 1.3.

Power (MW/MTIHM)	Time Step Size (days)	Decay Step Size (days)
31.12	85.0	0.0
32.51	45.0	0.0
26.20	70.0	0.0
0.00	0.0	29.0
22.12	40.0	0.0

8.2.4 Case 1.4: Long-Duration Burnup

Case 1.4 is designed to test a long-duration burnup event. This test case involves a total of ten time steps. Each burn step is 88 days in length and depletes the material at a constant power of 17.3025 MW/MTIHM, while each decay step in one year in length. The order of the time steps is as follows: 3 burnup, 1 decay, 2 burnup, 1 decay, 1 burnup, 1 decay. The initial isotopic number densities for the depleted material are provided in Table 8.6 and the results for select nuclides are presented in Table 8.7.

Table 8.5: Absolute difference (atom/b-cm) between a results from a standalone Exnihilo depletion calculation and results from a standalone ORIGEN calculation for select nuclides for case 1.3.

Step	^{90}Sr	^{131}Xe	^{145}Nd	^{235}U	^{239}Pu
1	2.7665e-08	1.6561e-09	1.4664e-10	7.6837e-06	1.4659e-08
2	2.9194e-08	3.1837e-07	4.6850e-09	1.6369e-05	2.6810e-07
3	2.8966e-08	5.6690e-07	2.0060e-08	1.5185e-05	5.6850e-08
4	3.4652e-08	7.5084e-07	3.6111e-08	1.9911e-05	2.2815e-08
5	9.7306e-09	8.3715e-07	3.2222e-09	4.8257e-06	3.4541e-07

Table 8.6: Isotopic composition for case 1.4.

ZAID	Concentration (atom/b-cm)
^{234}U	3.011071×10^{-1}
^{235}U	3.781905×10^{1}
^{236}U	1.806642×10^{-1}
^{238}U	1.129754×10^{3}

8.2.5 Case 1.5: ARP-Interpolation

The final standalone test case, case 1.5, is designed to test the use of interpolated libraries in a depletion calculation. The assembly configuration chosen for this case is a Westinghouse 17 × 17 PWR assembly configuration. The ^{235}U enrichment was specified as 4.862% and the moderator density was specified as 0.749 g/cm^3. The test case parameters are provided in Tables 8.8 and 8.9 and the results for select nuclides are presented in Table 8.10.

8.3 Intermediate Calculations

Before performing verification and validation of the full depletion capabilities, some of the individual components of the depletion package can be examined independently to ensure they are functioning properly. The two primary intermediate calculations

Table 8.7: Absolute difference (atom/b-cm) between a results from a standalone Exnihilo depletion calculation and results from a standalone ORIGEN calculation for select nuclides for case 1.4.

Step	^{90}Sr	^{131}Xe	^{145}Nd	^{235}U	^{239}Pu
1	1.0892e-07	2.4176e-07	2.0384e-08	5.6501e-08	1.0690e-07
2	2.4670e-08	1.8422e-07	4.3810e-08	1.3409e-08	3.6814e-08
3	1.9851e-07	4.2448e-07	1.2123e-07	4.3607e-08	1.2123e-07
4	3.2978e-08	1.9612e-07	9.6348e-08	1.1235e-07	2.3975e-09
5	1.3843e-07	5.8948e-07	2.0851e-07	1.3590e-07	2.5752e-08
6	1.2882e-07	9.1007e-07	6.7340e-07	2.0910e-07	3.5086e-08
7	6.6436e-07	2.7231e-07	4.0126e-07	1.1867e-07	1.6375e-07
8	4.2482e-07	1.5918e-06	6.3690e-07	2.9711e-07	5.1792e-08
9	8.7583e-07	7.7711e-07	4.2037e-07	5.4054e-07	9.0884e-08

Table 8.8: Isotopic composition for case 1.5.

ZAID	Concentration (atom/b-cm)
^{16}O	3.318200×10^{2}
^{234}U	8.972991×10^{-1}
^{235}U	1.012322×10^{2}
^{238}U	2.429332×10^{3}

which are useful to validate are the model power calculation and the population of 1-group cross sections in the ORIGEN library. Verification of these functions is performed by comparing to a TRITON calculation. Given that the input variables for both intermediate calculations are the AMPX multigroup cross sections and the group-wise flux, these data are simply extracted[*] from the TRITON calculation and used in the depletion package to perform the same calculation, thus providing a consistent comparison of results that is dependent only on the implementations that exists within the depletion package.

[*] Note that the actual AMPX library used to run the calculation in Exnihilo is generated from a standalone NEWT calculation since it is only possible to copy the AMPX library for the final depletion step. However, there should be little to no differences between this library and the library TRITON generated for the depletion calculation.

Table 8.9: Time step data for case 1.5.

Step	Time Step Size (days)	Power (MW/MTU)
1	25.0	38.285714
2	25.0	38.285714
3	37.5	33.986667
4	37.5	33.986667
5	77.5	15.439560
8	77.5	15.439560

Table 8.10: Absolute difference (atom/b-cm) between a results from a standalone Exnihilo depletion calculation and results from a standalone ORIGEN calculation for select nuclides for case 1.5.

Step	^{90}Sr	^{131}Xe	^{145}Nd	^{235}U	^{239}Pu
1	2.9248e-08	8.1814e-10	2.9812e-08	1.8194e-05	1.7598e-07
2	1.3197e-08	2.4284e-07	1.9529e-09	9.2389e-06	2.1344e-07
3	2.1141e-08	9.5585e-08	4.4053e-08	1.0804e-05	9.9841e-08
4	3.3458e-07	4.9704e-08	2.6361e-08	1.8366e-05	2.9128e-07
5	3.0452e-07	1.0578e-07	5.6978e-08	1.4061e-05	1.5122e-06
6	5.4607e-08	1.4110e-07	1.0590e-07	4.4548e-06	3.2429e-06

Consequently, the comparison of nuclide number densities between Exnihilo and TRITON for cases that are not fully-coupled to a transport calculation are not expected to match up as well as in the standalone case due to the predictor-corrector approach implemented within TRITON to perform the coupled transport-depletion calculation.

8.3.1 Case 2.1: Power Calculation - Pin Cell

Case 2.1 involves a single burnup step of two days with an average specific power of 31.12 MW/MTIHM. This test case is designed to test the power calculation. The model is similar to the pin cell model, except that the material definition is modified to provide a more simple case which should match up with TRITON to within the

Table 8.11: Comparison of the calculated material powers between Exnihilo and TRITON for case 2.1.

Material	Exnihilo	TRITON	$abs(\frac{C}{E}) - 1$
1	9.9724×10^{-1}	9.9724×10^{-1}	3.1661×10^{-6}
2	8.2012×10^{-4}	8.2000×10^{-4}	1.4953×10^{-4}
3	1.9367×10^{-3}	1.9400×10^{-3}	1.6907×10^{-3}

Table 8.12: Comparison of the calculated material fluxes between Exnihilo and TRITON for case 2.1.

Material	Exnihilo	TRITON	$abs(\frac{C}{E}) - 1$
1	2.2204×10^{14}	2.2204×10^{14}	1.6659×10^{-5}
2	2.2155×10^{14}	2.2155×10^{14}	7.3693×10^{-6}
3	2.2251×10^{14}	2.2251×10^{14}	2.0358×10^{-5}

precision of the TRITON power output. Furthermore, the energy group structure for both calculations used the full 238-group structure since it is not possible to calculate powers in TRITON with a collapsed group structure. The materials used in this test case are provided in Table B.3 in Appendix B.1.

The material powers and fluxes from the Exnihilo and TRITON calculations, as well as their corresponding relative differences, are provided in Tables 8.11 and 8.12. Note that the number of significant figures output by TRITON for materials 2 and 3 in Table 8.11 are less than the number of significant figures for material 1. This is the reason for the larger relative differences for materials 2 and 3, however note that the relative differences show that the results do indeed match up to within the precision of the TRITON results.

8.3.2 Case 2.2: Power Calculation - 2×2 Array

Having verified that the material powers and fluxes match up well in the most simplified of cases, analysis of a more complex case is performed for case 2.2. In

Table 8.13: Comparison of the calculated material powers between Exnihilo and TRITON for case 2.2.

Material	Exnihilo	TRITON	$abs(\frac{C}{E}) - 1$
1	6.2219×10^{-1}	6.1930×10^{-1}	4.6724×10^{-3}
2	1.6616×10^{-1}	1.6743×10^{-1}	7.6130×10^{-3}
3	2.0935×10^{-1}	2.1095×10^{-1}	7.5849×10^{-3}
4	2.7149×10^{-4}	2.7300×10^{-4}	5.5418×10^{-3}
5	3.1282×10^{-4}	3.1420×10^{-4}	4.3969×10^{-3}
6	3.2488×10^{-4}	3.2640×10^{-4}	4.6489×10^{-3}
7	1.0793×10^{-3}	1.0909×10^{-3}	1.0598×10^{-2}
8	4.6582×10^{-5}	4.7000×10^{-5}	8.8932×10^{-3}
9	5.6553×10^{-5}	5.7100×10^{-5}	9.5733×10^{-3}
10	4.6721×10^{-5}	4.7200×10^{-5}	1.0144×10^{-2}
11	5.1592×10^{-5}	5.2100×10^{-5}	9.7582×10^{-3}
12	1.1202×10^{-4}	1.1320×10^{-4}	1.0401×10^{-2}

this case the 2 × 2 array model is run for a single time step of 153 days with an average specific power of 31.12 MW/MTIHM. For this test case the expected relative differences will not be as good as in the previous case which examined the pin cell model since the problem has been complicated with many more materials, geometric regions, nuclides, and a much larger time interval and therefore algorithmic differences between the two codes will affect the results. Tables 8.13 and 8.14 provide the comparison of the material powers and fluxes for the 2 × 2 array model.

8.3.3 Case 2.3: One-Group Collapse

Case 2.3 tests the one-group cross-section collapse for each relevant reaction in the initial composition and the subsequent population of the transition matrix elements in the ORIGEN library. This benchmark utilized the 7-group collapsed group structure presented in Table B.2 in Appendix B.1 in order to limit the data transferred from the TRITON output to the Exnihilo input. Performing this benchmark involved first running a NEWT calculating to obtain the 7-group AMPX working formatted library

Table 8.14: Comparison of the calculated material powers between Exnihilo and TRITON for case 2.2.

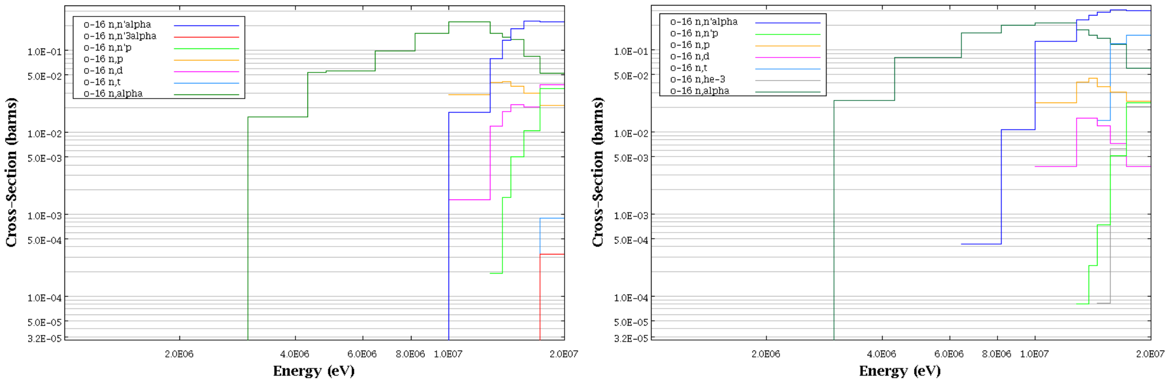
Material	Exnihilo	TRITON	$abs(\frac{C}{E}) - 1$
1	2.6116×10^{14}	2.5949×10^{14}	6.4346×10^{-3}
2	2.6179×10^{14}	2.6011×10^{14}	6.4346×10^{-3}
3	2.6105×10^{14}	2.5938×10^{14}	6.4348×10^{-3}
4	2.6162×10^{14}	2.5994×10^{14}	6.4276×10^{-3}
5	2.6317×10^{14}	2.6149×10^{14}	6.4208×10^{-3}
6	2.6250×10^{14}	2.6082×10^{14}	6.4227×10^{-3}
7	2.6998×10^{14}	2.6827×10^{14}	6.4024×10^{-3}
8	2.6217×10^{14}	2.6050×10^{14}	6.4242×10^{-3}
9	2.6426×10^{14}	2.6258×10^{14}	6.4170×10^{-3}
10	2.6355×10^{14}	2.6186×10^{14}	6.4185×10^{-3}
11	2.6975×10^{14}	2.6803×10^{14}	6.4045×10^{-3}
12	2.7078×10^{14}	2.6906×10^{14}	6.3968×10^{-3}

and the flux spectrum in the fuel material in the 7-group collapsed group structure. Following the NEWT calculation a COUPLE calculation was performed to generate an ORIGEN library using the 7-group flux spectrum and AMPX cross section library provided by the NEWT calculation. Then the fluxes, AMPX cross section library, and the ORIGEN library produced by COUPLE were used as input for an Exnihilo depletion calculation to test the accuracy of the cross section collapse and ORIGEN library population by the depletion package. By using the flux and cross section data from the NEWT calculation, the resulting ORIGEN library produced by Exnihilo should be the same as the ORIGEN library produced by COUPLE. This test case is performed by explicitly recording the cross sections both before and after Exnihilo has updated the library. Thus the cross sections for each nuclide present in the model should remain relatively unchanged after the depletion calculation updates the ORIGEN library. The results for the relevant cross sections both before and after the cross section calculation are presented in Table 8.15*.

*Although ^{234}U was left out of the table, the results are similar to the other uranium isotopes.

Table 8.15: Relevant 1-group cross sections both before and after the cross section calculation for Case 2-2.

ZAID	Reaction	Before	After	Diff.
80160	$(n, n'\alpha)$	6.1121×10^{-6}	1.2224×10^{-5}	5.0000×10^{-1}
80160	$(n, n'3\alpha)$	1.1846×10^{-3}	9.3943×10^{-13}	1.2610×10^{9}
80160	$(n, n'p)$	1.4530×10^{-5}	9.3195×10^{-8}	1.5491×10^{2}
80160	(n, γ)	2.9595×10^{-5}	2.9595×10^{-5}	5.1590×10^{-7}
80160	(n, p)	1.3051×10^{-5}	1.3051×10^{-5}	1.2132×10^{-6}
80160	(n, d)	6.4619×10^{-7}	1.2924×10^{-6}	5.0000×10^{-1}
80160	(n, t)	5.6937×10^{-12}	5.6938×10^{-12}	4.5846×10^{-6}
80160	$(n,^3 He)$	2.1966×10^{-8}	2.1966×10^{-8}	0.0000×10^{0}
80160	(n, α)	1.1785×10^{-3}	2.3571×10^{-3}	5.0000×10^{-1}
80160	$(n, 3\alpha)$	5.2729×10^{-12}	1.5819×10^{-11}	6.6667×10^{-1}
922340	$(n, 2n)$	6.7362×10^{-4}	6.7363×10^{-4}	2.4943×10^{-7}
922340	$(n, 3n)$	2.6923×10^{-7}	2.6923×10^{-7}	1.1871×10^{-6}
922340	(n, f)	4.4438×10^{-1}	4.4438×10^{-1}	3.8082×10^{-8}
922340	(n, γ)	2.8406×10^{1}	2.8406×10^{1}	3.2707×10^{-7}
922340	(n, p)	3.7136×10^{-7}	3.7136×10^{-7}	0.0000×10^{0}
922340	(n, α)	7.5807×10^{-8}	7.5807×10^{-8}	0.0000×10^{0}
922350	$(n, 2n)$	3.6251×10^{-3}	3.6252×10^{-3}	1.3593×10^{-6}
922350	$(n, 3n)$	2.2036×10^{-6}	2.2036×10^{-6}	9.6044×10^{-7}
922350	(n, f)	8.7834×10^{1}	8.7834×10^{1}	6.5268×10^{-7}
922350	$(n, 4n)$	1.2931×10^{-12}	1.2931×10^{-12}	9.7184×10^{-6}
922350	(n, n_1)	3.4034×10^{-3}	3.4034×10^{-3}	1.0561×10^{-7}
922350	(n, γ)	1.7220×10^{1}	1.7220×10^{1}	4.4810×10^{-7}
922350	(n, p)	4.0243×10^{-7}	4.0243×10^{-7}	0.0000×10^{0}
922350	(n, α)	1.5618×10^{-5}	1.5618×10^{-5}	0.0000×10^{0}
922360	$(n, 2n)$	1.9921×10^{-3}	1.9921×10^{-3}	7.9125×10^{-7}
922360	$(n, 3n)$	7.7398×10^{-6}	7.7398×10^{-6}	7.6440×10^{-7}
922360	(n, f)	2.9169×10^{-1}	2.9169×10^{-1}	5.0598×10^{-7}
922360	$(n, 4n)$	1.6901×10^{-13}	1.6901×10^{-13}	9.6778×10^{-6}
922360	(n, γ)	8.5634×10^{0}	8.5634×10^{0}	3.6408×10^{-7}
922360	(n, α)	5.0646×10^{-8}	5.0646×10^{-8}	0.0000×10^{0}
922380	$(n, 2n)$	4.6672×10^{-3}	4.6672×10^{-3}	1.3902×10^{-6}
922380	$(n, 3n)$	2.5965×10^{-5}	2.5965×10^{-5}	7.2241×10^{-7}
922380	(n, f)	9.1983×10^{-2}	9.1983×10^{-2}	2.0237×10^{-6}
922380	$(n, 4n)$	6.3113×10^{-12}	6.3114×10^{-12}	9.6706×10^{-6}
922380	(n, γ)	1.0232×10^{0}	1.0232×10^{0}	1.5995×10^{-7}
952421	$(n, 2n)$	1.5799×10^{-3}	1.5799×10^{-3}	1.1882×10^{-6}
952421	$(n, 3n)$	4.5671×10^{-8}	4.5671×10^{-8}	1.3451×10^{-6}
952421	(n, f)	1.1356×10^{3}	1.1356×10^{3}	3.0972×10^{-7}
952421	(n, γ)	2.1851×10^{2}	2.1851×10^{2}	1.2776×10^{-7}
952421	(n, p)	4.6202×10^{-7}	4.6202×10^{-7}	0.0000×10^{0}
952421	(n, α)	8.1861×10^{-8}	8.1861×10^{-8}	0.0000×10^{0}



(a) ENDF

(b) JEFF

Figure 8.3: Select cross sections for ^{16}O [74].

The relative differences presented in Table 8.15 for the actinides are all within 1×10^{-6}, however the cross section updates for ^{16}O are distinctly different from their original value, and in the cases of the $(n, n'3\alpha)$ and $(n, n'p)$ reactions, the difference is many orders of magnitude. To examine these reactions further, a plot of some of these cross sections from the ENDF and JEFF 238-group cross section libraries in SCALE is presented in Figure 8.3 and the flux profile over this same energy range is plotted in Figure 8.4. Since COUPLE utilizes both libraries for generating its cross sections, reaction which exist in both libraries will not be reproduced exactly within Exnihilo. However the larger differences require further investigation. Another important note to make is that reactions (n, p), $(n, ^3He)$ and (n, α) are not present in Table 8.15. This is because these reactions, which correspond to (n, p), $(n, ^3He)$, and (n, α), are high-energy threshold reactions which come from the JEFF nuclear data libraries, and because neither AmpxLib nor ORIGEN have access to the raw JEFF data, the microscopic cross sections are not available and cannot be updated; hence the relative errors of 0.0. The cross sections for these reactions are provided in Figure 8.5.

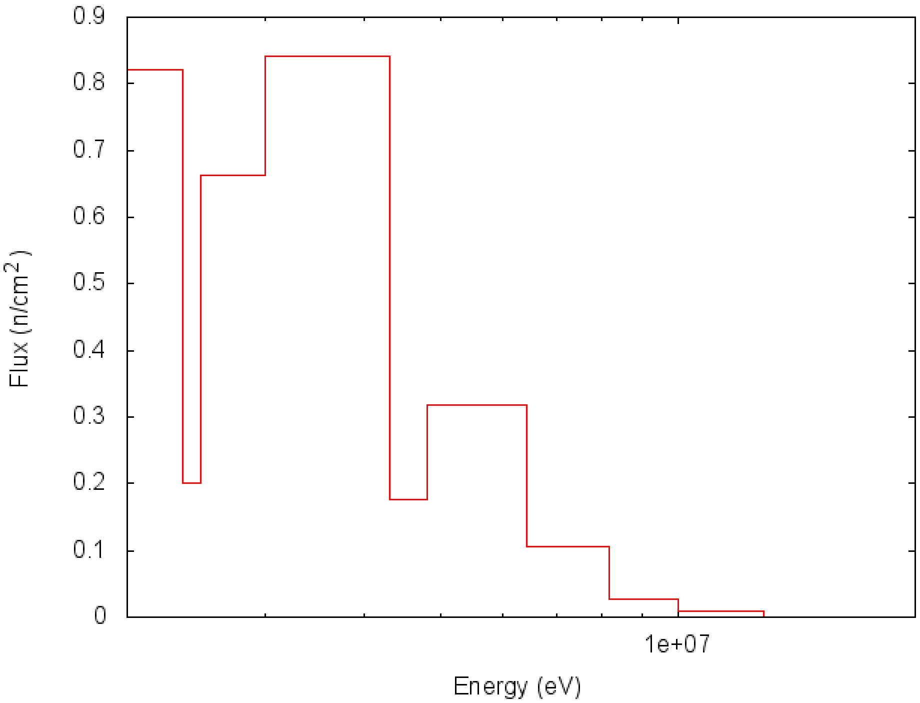


Figure 8.4: High-energy flux profile for Case 2-2.

8.4 Multigroup Depletion

The third set of benchmark cases involves benchmarking multigroup depletion calculations which use an AMPX working-formatted library as the source of cross section data. The AMPX libraries and fluxes used in the Exnihilo depletion calculations are taken from the TRITON calculation in which the Exnihilo depletion is being benchmarked against. Due to the predictor-corrector implementation in TRITON that cannot be bypassed, the number densities estimated by the depletion calculation in Exnihilo do not show an agreement that is within the numerical precision of ORIGEN. Furthermore, although the fluxes and cross sections are being taken directly from the TRITON calculation, there remain some mechanics, such as the way each code divides the burn lengths into subintervals, that will lead to differences in the results from both codes.

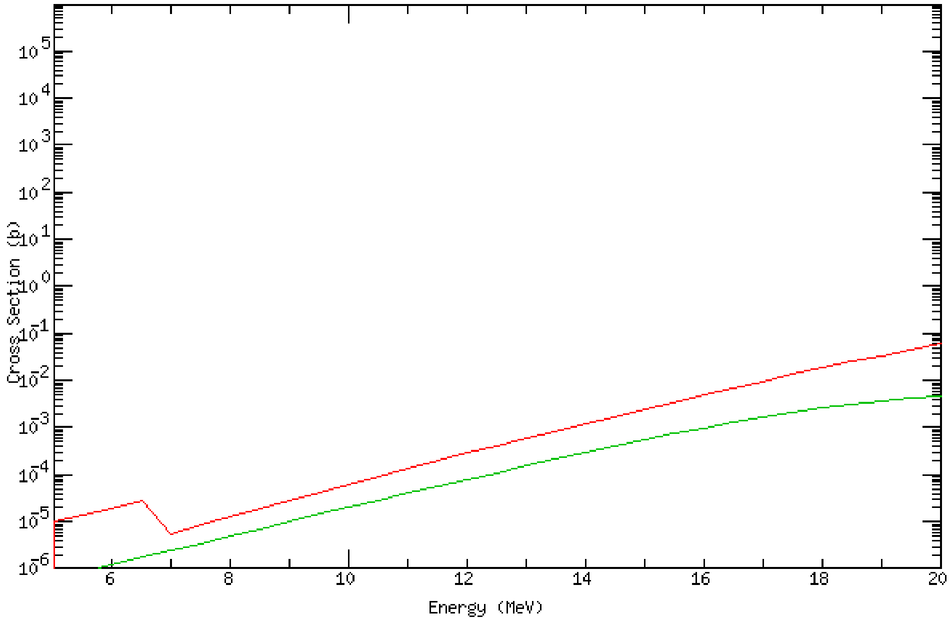


Figure 8.5: (n, p) and (n, α) for ^{235}U.

It should be noted that a complete verification and validation test suite for multigroup depletion calculations cannot be performed at present due to the coupling between Exnihilo and SCALE's multigroup processing modules. Exnihilo currently does not have the ability to use SCALE's resonance processing utilities in the middle of a calculation; only at problem initialization. Furthermore the interface between Exnihilo and SCALE is only set up to provide Exnihilo with an AMPX library that contains macroscopic cross sections by mixture, whereas the depletion package requires an AMPX library containing microscopic cross sections by nuclide. Therefore until this capability is developed, one-way coupling between the transport and depletion packages with no predictor-corrector approach is all that is available for multigroup depletion applications. Note however that the infrastructure is in place within Exnihilo to perform two-way coupling with a predictor-corrector for multigroup depletion calculations in the Exnihilo depletion package.

Table 8.16: Relative differences $(abs(\frac{C}{E}) - 1)$ in the power calculation between Exnihilo and TRITON for cycle 1, case 3.1.

MatID	153 days 4.76 GWd/MTIHM	306 days 14.28 GWd/MTIHM	377 days 14.28 GWd/MTIHM
1	3.1661×10^{-6}	1.0178×10^{-6}	7.4526×10^{-6}
2	1.4953×10^{-4}	8.5959×10^{-4}	8.5996×10^{-4}
3	1.6907×10^{-3}	7.3818×10^{-4}	3.8030×10^{-4}

8.4.1 Case 3.1: 6 Tracked Nuclides

Case 3.1 uses the pin cell benchmark case and is run for a single burnup cycle of 306 days at a constant power of 31.12 MW/MTIHM with a subsequent decay interval of 71 days. In this case, no additional nuclides added to the fuel prior to depletion in both Exnihilo and TRITON calculations, so the only nuclides being tracked are the ones present in the initial material definition. Therefore only the uranium number densities are examined when comparing the resulting number densities from the two codes. The calculated powers for each material for the first burn step are provided in Table 8.16 and the relative differences in the uranium number densities are presented in Table 8.17. Note that a better agreement is observed when comparing the calculated powers in the fuel material (material 1) than in the cladding and moderator materials. The reason for this is because the calculated powers in the cladding and moderator materials are derived from the energy release from neutron capture reactions while the calculate power in the fuel material is dominated by the energy release from fission reactions. While the Exnihilo depletion package is able to fold the flux spectrum into the fission cross sections, some of the capture cross section only exist in the JEFF data libraries and therefore cannot be updated within Exnihilo. Since the energy release per fission is much larger in the fuel material, the effects of this are not observed, but they are observed in the cladding and moderator materials.

Table 8.17: Relative differences $(abs(\frac{C}{E}) - 1)$in uranium number densities between Exnihilo and TRITON for cycle 1, case 3.1.

Nuclide	153 days 4.76 GWd/MTIHM	306 days 14.28 GWd/MTIHM	377 days 14.28 GWd/MTIHM
^{234}U	1.3978×10^{-4}	7.4853×10^{-4}	7.4901×10^{-4}
^{235}U	1.7459×10^{-4}	1.0164×10^{-3}	1.0164×10^{-3}
^{236}U	3.9512×10^{-4}	9.0274×10^{-4}	9.0274×10^{-4}
^{238}U	7.2315×10^{-5}	1.0977×10^{-4}	1.0977×10^{-4}

Table 8.18: Relative differences in the power calculation between Exnihilo and TRITON for burn step 1, case 3.2.

MatID	153 days 4.76 GWd/MTIHM	306 days 14.28 GWd/MTIHM	377 days 14.28 GWd/MTIHM
1	3.2918×10^{-6}	2.9451×10^{-6}	2.0642×10^{-6}
2	3.7597×10^{-5}	3.2657×10^{-3}	1.0311×10^{-3}
3	1.6762×10^{-3}	3.1261×10^{-5}	1.4079×10^{-3}

8.4.2 Case 3.2: 95 Tracked Nuclides

Case 3.2 is run with the same burnup parameters as case 3.1. The difference in this case is that a total of 95 nuclides are being tracked in the fuel in both Exnihilo and TRITON. The calculated powers for each material for the first burn step are provided in Table 8.18 and the relative differences for select nuclides are presented in Table 8.19. The comparison of the powers and isotopic concentrations illustrates an agreement similar to that observed in case 3.1. Thus is can be concluded that the nuclide tracking routines are operating consistently in Exnihilo.

8.5 Continuous-Energy Depletion

Benchmarking of Exnihilo's continuous-energy depletion capability is more challenging than the multigroup or standalone cases since TRITON's continuous-energy depletion

Table 8.19: Relative differences in uranium number densities between Exnihilo and TRITON for burn step 1, case 3.2.

Nuclide	153 days 4.76 GWd/MTIHM	306 days 14.28 GWd/MTIHM	377 days 14.28 GWd/MTIHM
^{95}Mo	9.6743×10^{-4}	6.7118×10^{-4}	9.9794×10^{-4}
^{131}Xe	5.3703×10^{-4}	1.6862×10^{-4}	1.6862×10^{-4}
^{131}Nd	4.1216×10^{-4}	1.9046×10^{-4}	1.9046×10^{-4}
^{235}U	3.4461×10^{-4}	4.8171×10^{-5}	5.0051×10^{-5}
^{239}Pu	3.2174×10^{-5}	1.7708×10^{-3}	1.7652×10^{-3}

capability is also in the development phase. Furthermore, there is a significant difference in the methods used as continuous-energy TRITON tallies one-group reaction rates for each relevant material/nuclide/reaction whereas Exnihilo is tallying only the flux in an ultra fine group structure and then collapsing down to a single energy group. In order to produce a similar comparison as the previous analyses, an ultra-fine group flux cell tally would need to be performed in TRITON to provide the transport results for the depletion calculation. To avoid this sort of monotonous task, testing of continuous-energy depletion using SCEMPP in Exnihilo is performed by coupling to the Shift Monte Carlo package.

8.5.1 Case 4-1: Default **ORIGEN** Library

Case 4-1 tests coupled transport-depletion calculations in Exnihilo for the pin cell model against both two- and three-dimensional multigroup TRITON calculations as well as a three-dimensional continuous-energy TRITON calculation. For the two-dimensional discrete ordinates cases $S_N = 12$ quadrature was used with a 16×16 mesh and a convergence criteria of 1×10^{-5} for all iterative phases of the solution process. For the Monte Carlo calculations, 10 000 histories per cycle were run with 1000 active cycles and 200 initial cycles skipped. The depletion parameters for this case include a single burnup step of 306 days with an average specific power of 31.12

MW/MTIHM and a 71 day decay interval following the burnup interval. Furthermore, a total of 95 nuclides are begin tracked in all calculations.

The first set of results utilize a generic (default) ORIGEN library file as the initial library for the depletion calculation. During the course of the coupled transport-depletion calculation, most of the values in the library will be changed using the problem-dependent flux spectrum. However, the values that cannot be changed include those reactions that are not in the ENDF data set. Therefore an initial library must be loaded in order to populate these reaction cross sections. The library that is considered the default library for Exnihilo depletion calculations was generated to be representative of a generic PWR pin cell and is included with the data libraries distributed with the development version of SCALE. This library is meant to be used as a starting library for codes which are integrating modular ORIGEN depletion capabilities and additional libraries for different reactor types will be added as development progresses.

The k_{eff} results generated by the Exnihilo and TRITON calculations are presented and compared in Table 8.20. The results for the initial transport calculation show a considerable difference depending on which version of TRITON Exnihilo is being compared against. When comparing against continuous-energy TRITON, an absolute difference of 50 pcm (per cent mille) is obtained for the initial criticality calculation, and the difference increases slightly to 61 pcm at 153 days. It is reasonable that a much better agreement is achieved when comparing against continuous-energy TRITON since both codes are in essence using the same physics[*].

As previously stated, there are significant differences in the mechanics between the TRITON and Exnihilo calculations. The most significant of these is the difference in how each code generated the one-group cross sections that are used by ORIGEN to solve the depletion equation. To examine this further the ultrafine-group flux spectrum in the fuel region from the Exnihilo depletion calculation is presented in

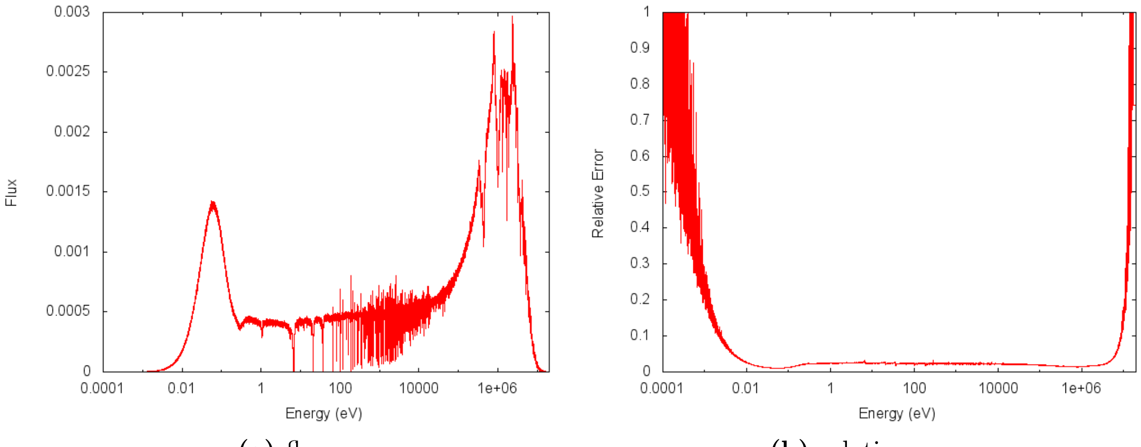
[*] Note that the differences when comparing continuous-energy TRITON to 2D and 3D multigroup TRITON are 342 and 577 pcm, respectively.

Table 8.20: Exnihilo transport results for case 4.1 compared against two-dimensional and three-dimensional multigroup TRITON calculations as well as a three-dimensional continuous-energy TRITON calculation.

Multigroup (2D)					
Burnup (GWd/MTIHM)	Exnihilo		TRITON		Abs. Diff. (pcm)
	k_{eff}	σ (pcm)	k_{eff}	σ (pcm)	
0	1.2544	-	1.2584	24	392
4.76	1.1409	-	1.1431	26	190
Multigroup (3D)					
Burnup (GWd/MTIHM)	Exnihilo		TRITON		Abs. Diff. (pcm)
	k_{eff}	σ (pcm)	k_{eff}	σ (pcm)	
0	1.2521	15	1.2584	24	625
4.76	1.1394	16	1.1431	26	370
Continuous-Energy					
Burnup (GWd/MTIHM)	Exnihilo		TRITON		Abs. Diff. (pcm)
	k_{eff}	σ (pcm)	k_{eff}	σ (pcm)	
0	1.2579	17	1.2584	24	50
4.76	1.1437	20	1.1431	26	61

Figure 8.7a. The primary observation one can make from the plots in Figure 8.7a is that the relative error increases dramatically in the lower energy range below approximately 0.005 eV and in the upper energy range above approximately 5 MeV. However, this is balanced by the fact that the flux also drops off in these same energy ranges. Another concern with such a fine group structure is that some energy groups may not register any tally contributions. In case 4.1, approximately 7% of the flux bins (3100 bins) do not have any contributions. Again however this is balances by the fact that 90% of these bins are below 0.005 eV and the remaining 10% are above 5 MeV.

To get an idea of how the flux spectrum from Exnihilo compares with the flux spectrum from TRITON, the ultrafine-group flux spectrum is collapsed down to the



(a) flux **(b)** relative error

Figure 8.6: Flux spectrum in the fuel region for the two-dimensional multigroup TRITON calculation and the Exnihilo calculation.

238-group structure used in the TRITON calculation*. A comparison of the flux spectrum in the fuel region between Exnihilo and the two-dimensional multigroup TRITON calculation is presented in Figure 8.7. The two-dimensional multigroup TRITON calculation was chosen for this analysis since the calculation itself is deemed the most accurate because the transport solver is deterministic (i.e. the results do not contain any associated uncertainties) and given that the problem is essentially two-dimensional.

The other important result to examine is the predicted number densities of the nuclides tracked in the fuel material. A comparison of the resulting number densities are presented in Table 8.21.

The discrepancies in the isotopic concentrations for the light elements (^{1}H, ^{10}B, ^{11}B, ^{14}N) have a few explanations†. One relevant difference is that TRITON is using JEFF data in addition to ENDF data. Since the important reactions affecting these

*Note that the performed collapse is not a perfect collapse since the group boundaries do not line up exactly, however for the sake of this comparison the minor inconsistencies resulting from this collapse are insignificant.

†The most likely explanation involves a bug which was recently uncovered with respect to the way the ORIGEN API is updating the cross sections for reactions with the same byproducts.

Table 8.21: Comparison of number densities between Exnihilo and a continuous-energy TRITON calculation in a case where Exnihilo is using the default ORIGEN library.

Nuclide	Exnihilo	TRITON	Rel. Diff.	Nuclide	Exnihilo	TRITON	Rel. Diff.
h-1	2.708×10^{-11}	2.588×10^{-09}	0.9895	ce-144	6.868×10^{-06}	6.883×10^{-06}	0.0021
b-10	3.863×10^{-19}	2.070×10^{-19}	0.8662	pr-141	1.249×10^{-05}	1.253×10^{-05}	0.0032
b-11	9.994×10^{-16}	4.660×10^{-16}	1.1446	pr-143	2.214×10^{-08}	2.217×10^{-08}	0.0015
n-14	8.935×10^{-16}	4.168×10^{-16}	1.1437	nd-143	1.109×10^{-05}	1.112×10^{-05}	0.0029
o-16	4.556×10^{-02}	4.557×10^{-02}	0.0001	nd-144	6.501×10^{-06}	6.507×10^{-06}	0.0010
kr-83	1.009×10^{-06}	1.000×10^{-06}	0.0092	nd-145	8.230×10^{-06}	8.245×10^{-06}	0.0018
sr-90	1.105×10^{-05}	1.106×10^{-05}	0.0005	nd-146	7.048×10^{-06}	7.062×10^{-06}	0.0020
zr-91	1.031×10^{-05}	1.030×10^{-05}	0.0008	nd-147	2.837×10^{-09}	2.843×10^{-09}	0.0020
zr-93	1.315×10^{-05}	1.319×10^{-05}	0.0032	nd-148	3.995×10^{-06}	4.005×10^{-06}	0.0024
zr-94	1.364×10^{-05}	1.365×10^{-05}	0.0007	pm-147	3.575×10^{-06}	3.560×10^{-06}	0.0041
zr-95	1.824×10^{-06}	1.827×10^{-06}	0.0017	pm-148	7.277×10^{-11}	5.544×10^{-11}	0.3125
zr-96	1.384×10^{-05}	1.387×10^{-05}	0.0024	pm-149	7.393×10^{-18}	7.579×10^{-18}	0.0245
nb-93	3.423×10^{-13}	3.462×10^{-13}	0.0112	sm-147	5.747×10^{-07}	5.729×10^{-07}	0.0032
nb-95	1.547×10^{-06}	1.550×10^{-06}	0.0016	sm-149	7.852×10^{-08}	8.086×10^{-08}	0.0290
mo-95	1.047×10^{-05}	1.049×10^{-05}	0.0019	sm-150	2.732×10^{-06}	2.791×10^{-06}	0.0212
mo-97	1.352×10^{-05}	1.347×10^{-05}	0.0036	sm-151	1.364×10^{-07}	1.383×10^{-07}	0.0137
mo-98	1.328×10^{-05}	1.335×10^{-05}	0.0053	sm-152	1.493×10^{-06}	1.503×10^{-06}	0.0069
mo-99	2.971×10^{-15}	3.022×10^{-15}	0.0169	sm-153	1.203×10^{-19}	1.363×10^{-19}	0.1172
mo-100	1.471×10^{-05}	1.473×10^{-05}	0.0016	eu-151	2.967×10^{-10}	3.011×10^{-10}	0.0145
tc-99	1.362×10^{-05}	1.370×10^{-05}	0.0059	eu-153	8.215×10^{-07}	8.146×10^{-07}	0.0085
ru-101	1.230×10^{-05}	1.232×10^{-05}	0.0020	eu-154	9.516×10^{-08}	9.931×10^{-08}	0.0418
ru-102	1.112×10^{-05}	1.116×10^{-05}	0.0038	eu-155	4.755×10^{-08}	4.931×10^{-08}	0.0356
ru-103	5.349×10^{-07}	5.372×10^{-07}	0.0044	eu-156	1.222×10^{-09}	1.256×10^{-09}	0.0269
ru-104	6.718×10^{-06}	6.761×10^{-06}	0.0064	gd-152	1.081×10^{-10}	1.023×10^{-10}	0.0570
ru-106	2.157×10^{-06}	2.176×10^{-06}	0.0089	gd-154	3.591×10^{-09}	3.756×10^{-09}	0.0439
rh-103	7.820×10^{-06}	7.857×10^{-06}	0.0047	gd-155	1.511×10^{-09}	1.568×10^{-09}	0.0364
rh-105	1.462×10^{-22}	1.476×10^{-22}	0.0093	gd-156	2.554×10^{-07}	2.604×10^{-07}	0.0194
pd-105	4.288×10^{-06}	4.321×10^{-06}	0.0076	gd-157	8.026×10^{-10}	8.169×10^{-10}	0.0175
pd-107	2.006×10^{-06}	2.028×10^{-06}	0.0109	gd-158	8.729×10^{-08}	8.821×10^{-08}	0.0104
pd-108	1.208×10^{-06}	1.227×10^{-06}	0.0153	gd-160	5.910×10^{-09}	5.974×10^{-09}	0.0107
ag-109	7.433×10^{-07}	7.548×10^{-07}	0.0153	u-234	5.631×10^{-06}	5.993×10^{-06}	0.0604
cd-113	3.270×10^{-09}	3.328×10^{-09}	0.0175	u-235	2.254×10^{-04}	2.255×10^{-04}	0.0007
in-115	3.635×10^{-08}	4.044×10^{-08}	0.1012	u-236	3.759×10^{-05}	3.764×10^{-05}	0.0012
sn-126	2.100×10^{-07}	2.112×10^{-07}	0.0056	u-238	2.219×10^{-02}	2.219×10^{-02}	0.0001
i-127	5.216×10^{-07}	4.795×10^{-07}	0.0877	np-237	1.674×10^{-06}	1.675×10^{-06}	0.0007
i-129	1.687×10^{-06}	1.682×10^{-06}	0.0027	pu-238	1.786×10^{-07}	1.791×10^{-07}	0.0030
xe-131	6.450×10^{-06}	6.459×10^{-06}	0.0014	pu-239	7.527×10^{-05}	7.690×10^{-05}	0.0212
xe-133	3.745×10^{-11}	3.808×10^{-11}	0.0166	pu-240	1.339×10^{-05}	1.369×10^{-05}	0.0221
cs-133	1.486×10^{-05}	1.492×10^{-05}	0.0042	pu-241	5.421×10^{-06}	5.507×10^{-06}	0.0157
cs-134	5.193×10^{-07}	5.215×10^{-07}	0.0042	pu-242	6.539×10^{-07}	6.642×10^{-07}	0.0155
cs-135	2.665×10^{-06}	2.690×10^{-06}	0.0093	am-241	1.058×10^{-07}	1.076×10^{-07}	0.0167
cs-137	1.418×10^{-05}	1.412×10^{-05}	0.0041	am-242	5.916×10^{-15}	6.001×10^{-15}	0.0141
ba-140	1.692×10^{-08}	1.696×10^{-08}	0.0024	am-242m	4.585×10^{-10}	4.651×10^{-10}	0.0142
la-139	1.421×10^{-05}	1.424×10^{-05}	0.0023	am-243	3.304×10^{-08}	3.372×10^{-08}	0.0201
ce-141	4.281×10^{-07}	4.294×10^{-07}	0.0030	cm-242	7.116×10^{-09}	7.218×10^{-09}	0.0141
ce-142	1.292×10^{-05}	1.294×10^{-05}	0.0019	cm-243	1.213×10^{-10}	1.296×10^{-10}	0.0642
ce-143	2.323×10^{-23}	2.327×10^{-23}	0.0019	cm-244	2.249×10^{-09}	2.344×10^{-09}	0.0404

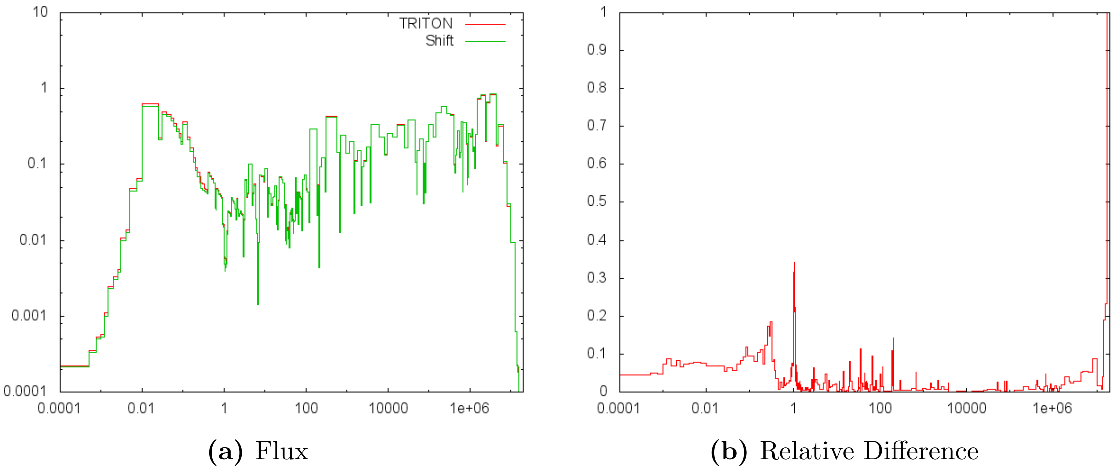


Figure 8.7: 238-group flux spectra in the fuel region calculated by a two-dimensional TRITON calculation and the Shift calculation.

light elements are threshold reactions whose cross sections vary dramatically at higher energies, a small change in the flux or difference in cross section data can result in a much larger change in the isotopic concentrations for these light elements. Recalling Figure 8.3 to use ^{16}O as an example, the relevant energy range for these threshold reactions in between 5 MeV and 20 MeV, and from Figure 8.7 is was observed that the statistics on the ultra-fine group flux tally are poor in this region, and this could also lead to the discrepancies observed in Table 8.21. Furthermore, note that the isotopic concentrations are on the order of 10^{-9} and below, so their relative importance to the global eigenvalue calculation is negligible. For isotopic production applications however, the JEFF data will be required and further investigation will be necessary if the discrepancies in the light elements are not significantly reduced by the integration of JEFF cross section data.

A notable difference of approximately 10% in the concentration of ^{115}In is also present in Table 8.21. By performing a quick comparison of the JEFF and ENDF cross section data for ^{115}In, it is observed that the cross section data for $(n, n'p)$, $(n, n't)$, and (n, d) are equivalent with the exception that the JEFF cross section data has data for a few additional groups, however the cross sections for these groups are on the order of 10^{-15} and should therefore result in a negligible difference in the resulting concentration.

Other isotopes that show notable discrepancies above 5% include ^{127}I, ^{148}Pm, ^{153}Sm, ^{152}Gd, ^{234}U, and ^{243}Cm. For all isotopes except ^{234}U, the concentrations are all less than 1×10^{-10}, so their individual effects on the global eigenvalue calculation are negligible. However a 6% difference in the ^{234}U concentration can have a significant effect on the calculated eigenvalue. Since there are several different sources from which the discrepancy could stem from, each of these sources must be isolated in order to observe their independent effect. This issue is investigated further in the following sections.

Case 4.2: Custom ORIGEN Library

A second set of results for the pin cell benchmark is generated using the ORIGEN library generated by the two-dimensional multigroup TRITON calculation as the initial starting library for the Exnihilo calculation. Because this library was generated for an identical model, it should be a much more representative starting library. The transport results presented in Table 8.22 show an improvement of 7 pcm from using a pre-generated ORIGEN library, which is well within the statistical uncertainty on the k_{eff} calculation. The primary reason that little to no difference is observed is because the fission yield data stored in each library is identical. The only other data that Exnihilo cannot modify is the data for the high-energy threshold reactions which has little to no impact on the eigenvalue results.

A comparison of the resulting number densities for case 4.2 is presented in Table C.1 in Appendix C.1. In comparing Tables 8.21 and C.1, minimal differences are observed when a TRITON-generated ORIGEN library, which was generated for an identical model, is used as the initial ORIGEN library for the Exnihilo calculation. The primary differences between the two results are that the discrepancies for ^{127}I, ^{148}Pm, ^{153}Sm, ^{152}Gd, and ^{234}U all drop below 5%. This either signifies that these isotopes

Table 8.22: Exnihilo transport results for case 4.2 compared against two-dimensional and three-dimensional multigroup TRITON calculations as well as a three-dimensional continuous-energy TRITON calculation.

Multigroup (2D)					
Burnup (GWd/MTIHM)	Exnihilo		TRITON		Abs. Diff. (pcm)
	k_{eff}	σ (pcm)	k_{eff}	σ (pcm)	
0	1.2544	-	1.2584	24	392
4.76	1.1409	-	1.1431	28	223
Multigroup (3D)					
Burnup (GWd/MTIHM)	Exnihilo		TRITON		Abs. Diff. (pcm)
	k_{eff}	σ (pcm)	k_{eff}	σ (pcm)	
0	1.2521	15	1.2581	24	627
4.76	1.1394	16	1.1431	28	377
Continuous-Energy					
Burnup (GWd/MTIHM)	Exnihilo		TRITON		Abs. Diff. (pcm)
	k_{eff}	σ (pcm)	k_{eff}	σ (pcm)	
0	1.2579	17	1.2581	24	50
4.76	1.1437	20	1.1431	28	54

have significant cross sections that exist in the JEFF library and not in the ENDF library, or that this improvement is simply a function of the statistical fluctuation of the Monte Carlo result. The 7 pcm improvement in the eigenvalue is thus likely due to the improved estimation of ^{234}U.

Note that the discrepancies in the light elements, ^{115}In and ^{243}Cm are similar to those presented in Table 8.21, so this eliminates the initial ORIGEN library data loaded at the start of the calculation as the source of these observed discrepancies. This also eliminates the lack of an ability to update cross sections that are only available in the JEFF libraries. Given the difference in methodologies and difference in data availability between TRITON and Exnihilo, further analysis is required to determine whether or not the relative differences in the 1–5% range are acceptable. For a case as simple as Case 4-1, the results are anticipated to be in much better agreement once the JEFF data becomes accessible as the differences in the transport-depletion coupling should have a negligible effect for this problem.

8.6 Additional Analyses

8.6.1 Number of Energy Groups

The method of binning the flux in ultra-fine groups represents a distinct difference between the way Exnihilo and TRITON perform coupled transport-depletion calculations. Thus a study of the effects the number of energy groups has on the calculation as a whole was performed. Impacts on the computational time, memory usage, and accuracy of the calculation are all expected to be significant as the number of energy groups varies. The transport calculation is affected by having to look up the energy bin a particular tally contribution needs to be stored in, so more energy groups leads to a longer lookup time for each tally score. The depletion calculation uses this tally group structure to generate the one-group cross section set used to collapsed the transition cross sections down to one group. Performing this analysis involves testing various attributes of the calculation as the number of energy groups is increased up to 100 000 groups. Recall that the groups are logarithmically spaced and thus no spectral effects are considered when created each group structure. Thus the primary objective of this analysis is to identify at which point the lack of resonance processing becomes insignificant.

First the computational memory as a function of the number of energy groups is analyzed as this analysis can be performed without running any calculations. The estimation of the required memory considers storing the ultrafine-group cell tally as well as the microscopic cross section responses used by the depletion calculation, as these two operations are expected to take up the majoirty of the memory when depletion is added to a Shift transport calculation. As an example, consider a 17×17

assembly which contains a total of 1156 regions, which would result in a total of 1.96 GB in tally and depletion data. Add in 100 axial regions and we reach a total of 115 600 regions and require approximately 160 GB to store the tally data alone. Extend this one step further to a core containing 241 assemblies and the number of regions approaches 28 million which requires approximately 38 000 GB for tally data storage. Thus when running large models such as a full-core depletion calculation, memory management will be a major factor in deciding the size of the physical domains. The Cray XK7 (Titan) machine at the National Center for Computational Sciences [5] for example has 18 688 compute nodes and a total of 598 TB of memory, however if all processors are being used this equates to 2 GB of memory available to each processor. A plot of the memory requirements of the cell tallies used for depletion for a pin cell, two-dimensional assembly, three-dimensional assembly, and full-core case is provided in Figure 8.8.

Figure 8.9 plots the flux and associated uncertainty for several simulations which all used a different number of energy groups for the depletion cell tally. The results show fairly similar flux profiles for the different cases. The primary difference in the flux spectra is that the resonance regions become more defined as the number of energy groups increases which decreases the energy group spacing enough to capture certain cross section resonances. The uncertainties in Figure 8.9b show a general increase in the relative error as the number of energy groups increases. Note that the majority of the relative errors for the 100 000-group case are above 20% while the relative errors for the 1000-group case are mostly below 7%. This means that if more energy groups are used then more particles must be simulated in order to ensure the flux tally estimates are accurate. To take a closer look at the difference in the flux spectrum, the fluxes are collapsed down to a 238-group structure. Then the collapsed flux spectra are compared against the case which used 100 000 energy groups, and the absolute differences are plotted in Figure 8.10. The results of the group collapse clearly illustrate the issue with using only a few thousand energy groups. Some of the flux groups, especially in the mid- to high-energy range, show significant differences

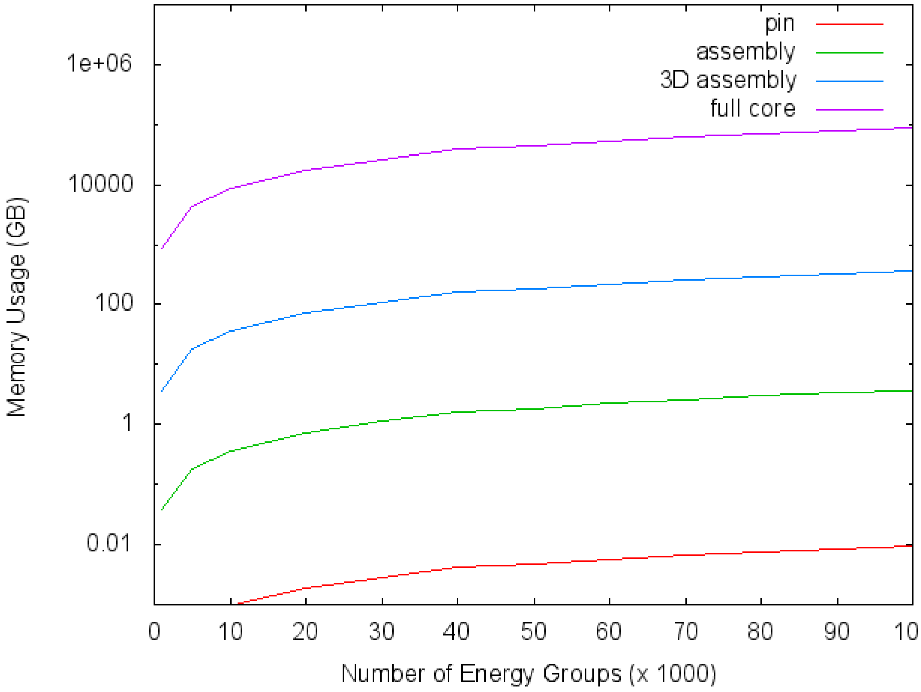
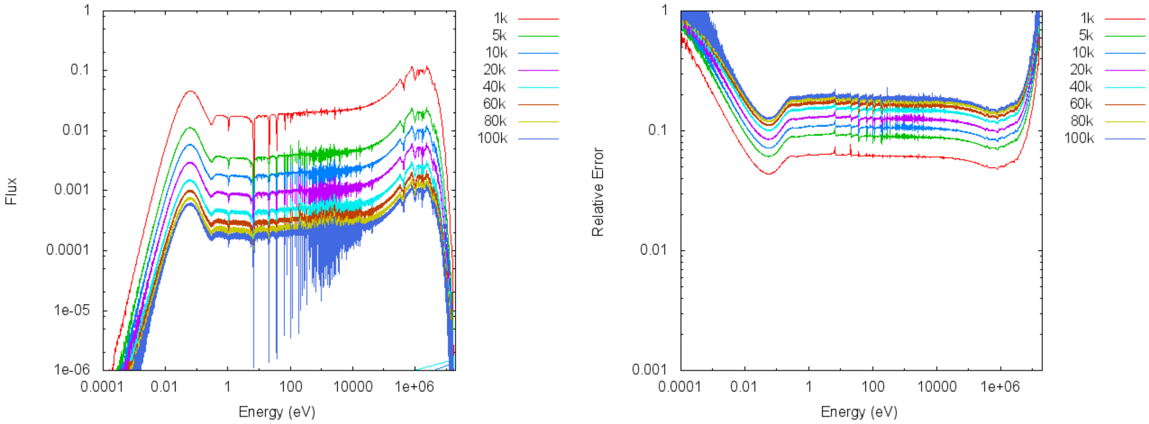


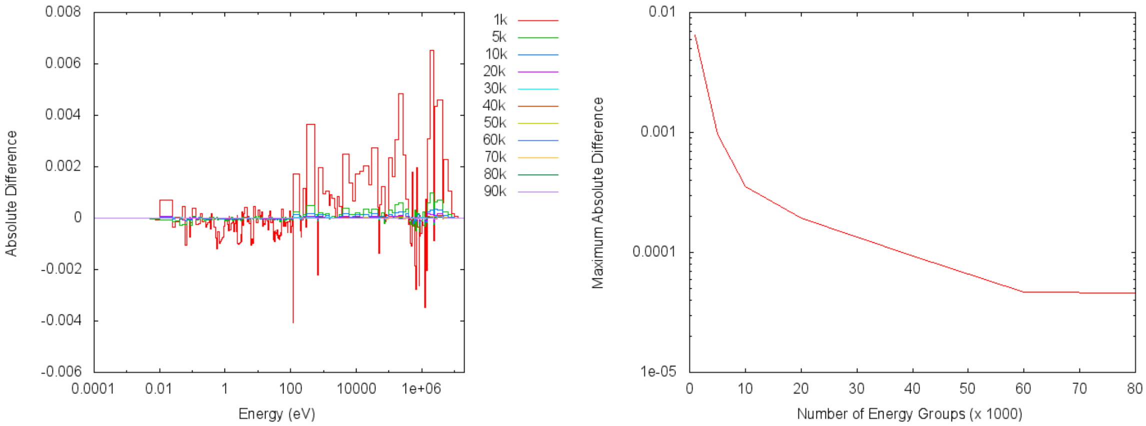
Figure 8.8: Memory required to store data required by the depletion calculation for four different geometry sizes.



(a) flux

(b) relative error

Figure 8.9: Flux spectrum and associated relative error in the pin cell fuel region for cases which used a different number of tally energy groups..



(a) **(b)**

Figure 8.10: Flux spectrum collapsed down to a 238-group structure.

when compared against the 100 000 group case. Figure 8.10b illustrates that, for the case, at least 10 000 energy groups should be used and anything over 60 000 energy groups does not provide much additional benefit[*].

The last and most important parameter to analyze with respect to the number of energy groups used by the depletion calculation is k_{eff}. Deciphering how the number of energy groups affects the overall accuracy of the calculation is the most difficult item to assess. Knowing that the cross sections are not being processed as multigroup cross sections generally are for resonance and self-shielding effects, it can be concluded that too few energy groups will lead to inaccurate cross section representation. The question remains however: how many groups are "too few?" In an attempt to identify a lower bound for the number of energy groups required to mitigate the lack of resonance processing, a plot of the difference in k_{eff} when compared against the reference solutions generated by two- and three-dimensional TRITON calculations is presented in Figure 8.11. The results of the k_{eff} comparison

[*] Note that this is nearly the same number of energy groups that the one-dimensional transport code CENTRM uses to calculate the problem-specific fluxes which are used to generate self-shielded multigroup cross sections for all of the multigroup control sequences in SCALE (CENTRM is the default option in most control sequences) [106].

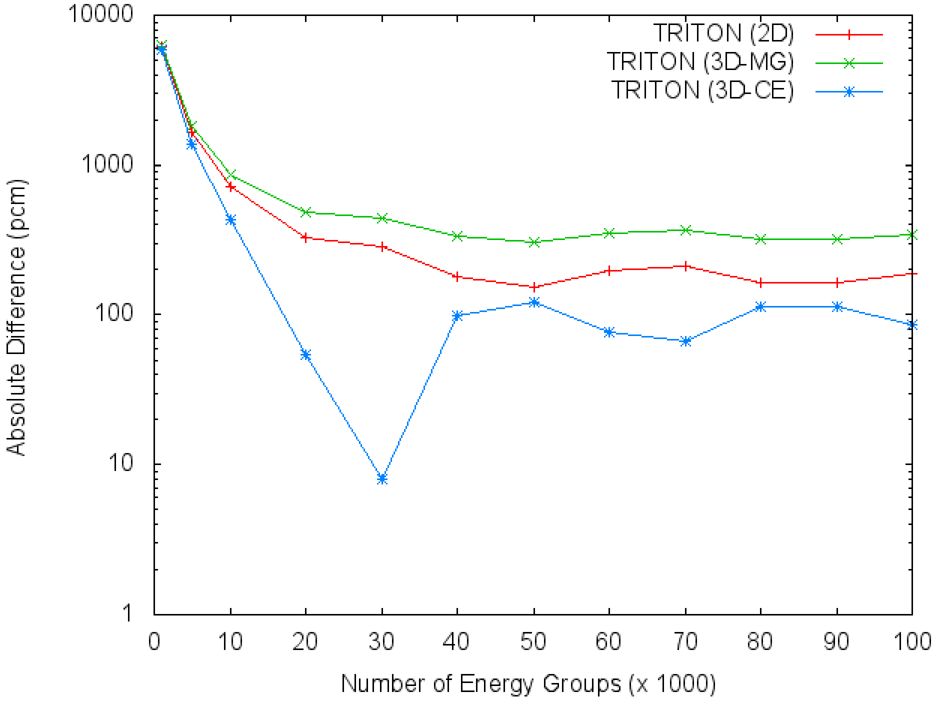
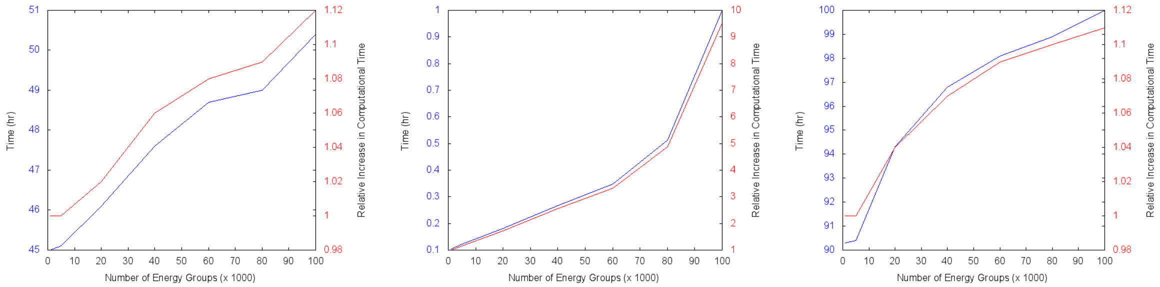


Figure 8.11: Absolute difference in k_{eff} for varying numbers of energy groups compared against several different TRITON calculations.

for all three cases show that the lower bound is somewhere between 20 000 and 40 000 energy groups are required to mitigate the errors associate with using the ultrafine-group flux binning method. Note that the statistical uncertainties in the k_{eff} estimates for all calculations used to generate Figure 8.11 are below 30 pcm.

In analyzing the computational time for these cases, all simulation times were within the range of 43–44 hours, and no discernible trend was observed with respect to the number of energy groups being used. Thus this 3-region model is too simple to perform an analysis on the effect the number of energy groups has on the calculation time. Therefore the pin cell block model discussed in Section 8.1 will be used to make a model with 700 depletable regions ($5 \times 5 \times 28$). This model was run with a varying number of energy groups until the uncertainty in k_{eff} reached approximately 40 pcm.



(a) transport **(b)** depletion **(c)** total

Figure 8.12: Timing results for the Monte Carlo transport and depletion calculations.

The relative increase in computational time* for both the Monte Carlo transport and depletion calculations is provided in Figure 8.12. Although the depletion calculation shows a increase in computational time by a factor of nine using 100 000 energy groups, the relative increase in the Monte Carlo transport calculation is only 12%, which leads to an increase in the total calculation time of approximately 11%. From this analysis it can be concluded that the increase in calculational time is likely not going to be a major factor in deciding the number of energy groups to use for the ultrafine-group flux tally.

Figure 8.13 provides a second comparison of the absolute difference in k_{eff} versus the number of energy groups, however this case is only compared against a single TRITON calculation. The accuracy of the k_{eff} calculation appears to improve up until 40 000 energy groups is reached. However unlike the pin cell results illustrated in Figure 8.11, the accuracy begins to decrease when more energy groups are added beyond 40 000. Considering that this problem has 1400 total tally regions and consider the case where 100 000 energy groups are used as an example. This particular problem will have a total of 140 million tally bins in which enough scores must be registered to provide a reasonable estimation of the flux in each group/region. The

*relative to the case that used 1000 energy groups

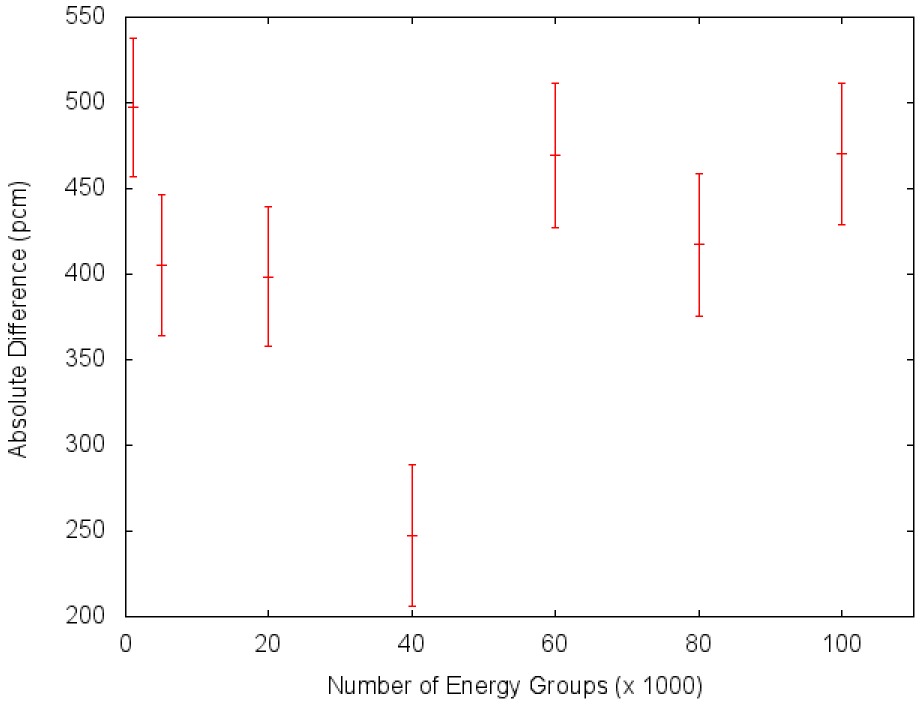


Figure 8.13: Absolute difference in k_{eff} versus a TRITON calculation.

pin cell case only has a total of 300 000 tally bins, which is more than two orders of magnitude smaller.

From this analysis it can be concluded that too many energy groups can also lead to erroneous results. If certain flux groups are not estimated with enough accuracy, and cross sections exist within these groups that are relatively high in comparison to other groups, the one-group collapsed cross section will be incorrect. This could lead to significant differences in the estimated nuclide concentrations for the parent and daughter nuclides for this reaction. Furthermore, the error could propagate further as any nuclides that are relevant to the transmutation-decay chain surrounding this cross section can be affected as well. As a result of this analysis, a statistical check was placed in the depletion package to check that the uncertainties within the 0.005 eV–5 MeV range are below 10%. This energy range was determined using the analysis presented in Section 8.5.1.

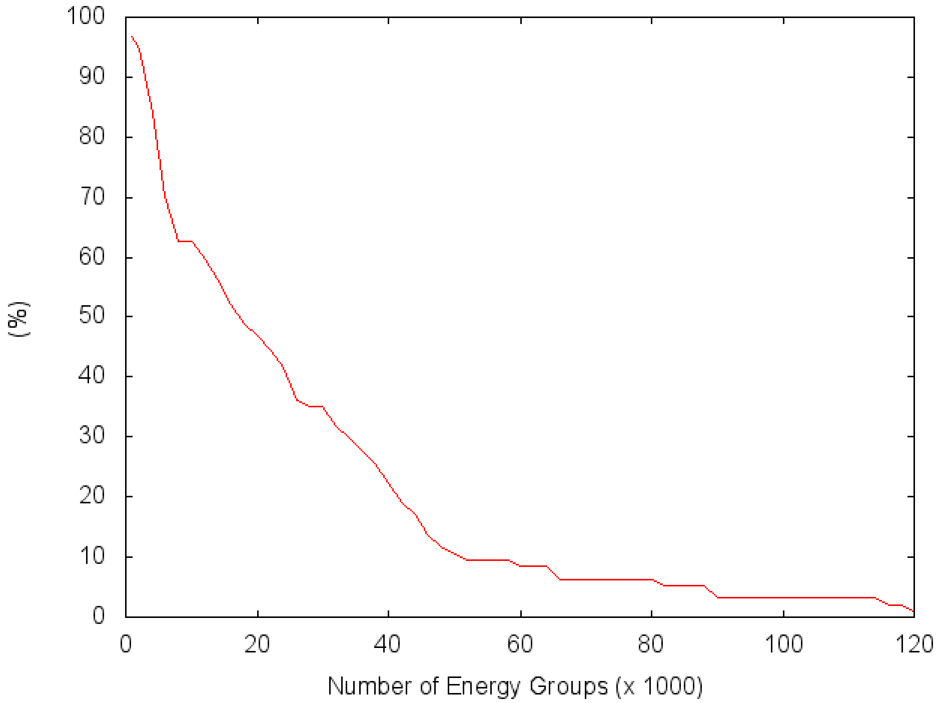


Figure 8.14: Percentage of the default nuclide set (94 total nuclides) typically added to a depletable material that have at least given number of energy points in the continuous-energy library for a temperature of 293 K [107].

The final aspect that is relevant to the choice of the number of energy groups is the number of energy points available in the continuous-energy library. If there are more energy groups than there are cross section data points in the library, multiple groups will end up collapsing the same cross section data. Furthermore, by combining these energy groups one could obtain better statistical estimated for the flux cell tally used to collapse the cross sections down to one group. Figure 8.14 displays what percentage of the 94 nuclides typically tracked in a coupled transport-depletion calculation have at least a given number of energy points. Figure 8.14 illustrates that using 41 000 energy groups results in the flux tally having more energy groups than there are cross section data for 80% of the nuclides in this data set. Using 52 000 groups results in the tally having more groups than 90% of the nuclides. Thus using a number of energy groups within this range provides an ideal efficiency for the flux binning method. Nuclides with fewer cross section groups than the flux tally could theoretically not suffer from the lack of resonance processing if the energy group spacing with similar to the cross section energy points for those nuclides. In this case, the collapsed cross sections would be identical to the cross sections obtained if the individual reaction rates were tallied instead of an ultrafine-group flux. However, every nuclide has a completely independent set of energy points, so providing energy groups that are spaced similarly to all 94 nuclides is not possible.

From the analyses presented in this section, a default value of 43 000 energy groups will be used for the ultrafine-group flux tally within Exnihilo. It is at 43 000 where it begins to take adding an additional 1000 groups just to have as many energy points as one more nuclide. The nuclides within the analyzed set of 94 nuclides that have more than 43 000 energy groups are provided in Table 8.23 along with the required memory to load the nuclide and the number of energy points available for that nuclide within the continuous-energy library. A complete table for all 94 nuclides is presented in Table D.3 in Appendix D.

8.6.2 Parallel Depletion

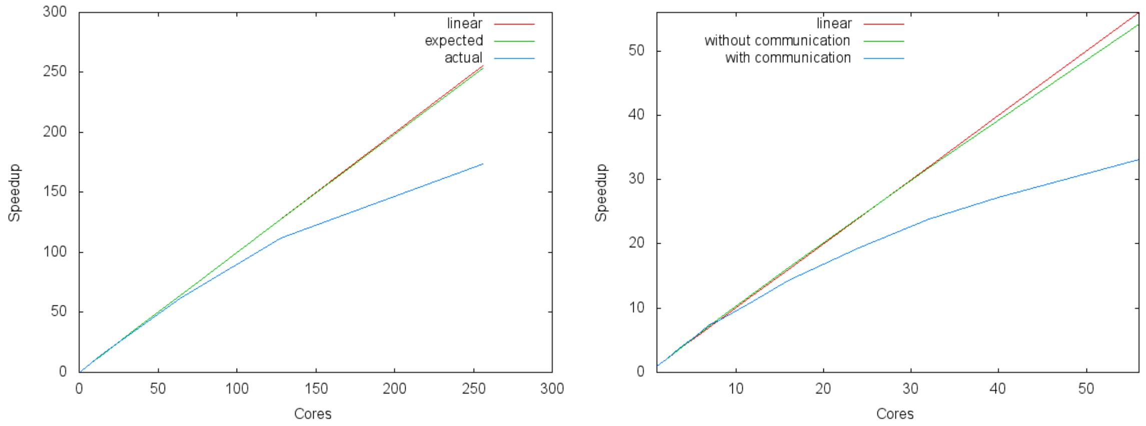
As mentioned in Section 7.4.2, a parallel algorithm was developed to distribute materials across multiple processors. This enables the depletion package to only deplete each material once instead of each processor depleting every material it has knowledge of regardless of whether or not the materials are replicated across multiple processors. Neglecting the communication costs, this algorithm is expected to scale linearly assuming that the number of materials can be divided evenly among all processors. To examine the scalability of this algorithm, a generic array of PWR pin cells was modeled such that is contained 17 500 depletable materials, and this calculation was performed in standalone mode in order to bypass the memory requirements when coupling to a transport calculation. A plot of the speedup for

Table 8.23: Nuclides with more than 43 000 cross section energy points.

Nuclide	Required Memory	Number of Energy Points
^{238}U	281.48	273,303
^{235}U	117.48	118,487
^{239}Pu	127.32	115,278
^{99}Tc	175.42	89,195
^{113}Cd	73.25	89,094
^{240}Pu	54.68	80,353
^{127}I	68.46	65,910
^{109}Ag	96.16	64,124
^{237}Np	61.68	58,680
^{103}Rh	148.34	51,695
^{115}In	30.82	48,798
^{147}Sm	111.18	46,961
^{154}Gd	101.20	46,745
^{145}Nd	92.19	44,913
^{133}Cs	59.52	44,718
^{141}Pr	53.03	44,185

this problem using up to 256 cores is presented in Figure 8.15a. The "expected" profile in Figure 8.15a accounts for the fact that 17 500 materials cannot be divided evenly between all processors, however note that the deviation from the linear profile is small. Figure 8.15a illustrates that the scaling is linear up to 64 cores and then begins to drop off considerably. This is the typical behavior when observing parallel scaling as the amount of work per slave processor begins to become small enough that the communication costs become significant.

The second scaling study involves utilizing the 5 × 5 array model described in Section 8.1 while running coupled to a continuous-energy Shift calculation. Because this calculation is coupled to transport, the global reduction of nuclide number densities now becomes a factor (in the previous case illustrated in Figure 8.15a, this global reduction was not performed), however scaling was only possible up to



(a) 17 500 material model **(b)** 5 × 5 array model

Figure 8.15: Comparisons of the observed speedup obtained by distributing the materials among processors within the same geometric.

56 cores due to the memory requirements to load the continuous-energy data and the computational resources available. Figure 8.15 illustrates the observed speedup and compares it to the speedup observed without the global reduction of nuclide number densities. Thus is can be concluded that a significant speedup can be achieved using this parallel algorithm, however additional scaling studies using more cores, with more available memory, and with more depletable regions needs to be performed in order to properly identify the limitations of this algorithm.

8.7 Pin Cell Demonstrations

To demonstrate the capabilities of coupled transport-depletion calculations using Exnihilo a pin cell model was produced using Problem 1c from the Virtual Environment for Reaction Applications (VERA) benchmark progression models [108]. The differences between the benchmark specification given in Reference [108] for problem 1c and the problem run in this demonstration is that a total of 99 nuclides were tracked in the fuel and the amount of boron in the moderator was decreased from

1300ppm to 600ppm. The burnup cycle data utilized in this demonstration involves five 2.86-day burnup steps at an average specific power of 35 MW/MTIHM with no subsequent decay calculations. A comparison of the eigenvalue results for this problem is presented in Table 8.24, a comparison of the power and flux in the fuel is provided in Table 8.25, and a comparison of the isotopics as a function of burnup for select nuclides which have number densities above 1×10^{-7} is presented in Figure 8.16. The calculation used as the basis for comparison in the plot presented in Figure 8.16 was a two-dimensional TRITON calculation which used the discrete ordinated code NEWT for the transport solution. The relevant parameters for this TRITON calculation include $S_N = 16$, a 16×16 mesh, and convergence criteria of 1×10^{-5} for all iterative phases of the solution process.

The isotopic results presented in Figure 8.16 show percent differences less than 10% for the majority of the nuclides, but the differences are much higher for ^{131}Xe, ^{133}Cs, ^{239}Pu, and ^{240}Pu. Due to the fact that multiple inconsistencies between the two calculations exist which invalidate the comparison, the most important of which is that a comparison between multigroup and continuous-energy calculations is being performed, a comparison versus experimental data would serve as a much better validation the developed depletion implementation. This comparison is provided in Appendix C.3.

8.8 Assembly Demonstration

To demonstrate the capabilities of coupled transport-depletion calculations on a larger geometry, a quarter-assembly model was produced using Problem 2a from the VERA benchmark specifications [108]. A graphical representation of the benchmark model is provided in Figure 8.17. This model represents a single Westinghouse 17 × 17 fuel assembly at Beginning-of-Life (BOL). The primary difference between the model used and the benchmark model in Reference [108] is that a total of 94 nuclides are added to the fuel materials to be tracked during depletion, the amount of boron in the

Table 8.24: Exnihilo transport results for the VERA 1c pin cell model compared against various TRITON calculations.

		Multigroup (2D)			
Burnup (GWd/MTIHM)	TRITON		Exnihilo		Abs. Diff. (pcm)
	k_{eff}	σ (pcm)	k_{eff}	σ (pcm)	
0.00	1.23683	-	1.23890	9	207
0.05	1.19924	-	1.20057	9	134
0.20	1.19303	-	1.19491	9	188
0.45	1.18653	-	1.89070	8	254
0.80	1.18283	-	1.18395	9	113
1.25	1.17853	-	1.17917	9	64
		Multigroup (3D)			
Burnup (GWd/MTIHM)	TRITON		Exnihilo		Abs. Diff. (pcm)
	k_{eff}	σ (pcm)	k_{eff}	σ (pcm)	
0.00	1.23726	5	1.23890	9	164
0.05	1.19957	6	1.20057	9	100
0.20	1.19328	5	1.19491	9	163
0.45	1.18691	6	1.19070	8	216
0.80	1.18313	5	1.18395	9	83
1.25	1.17885	5	1.17917	9	33
		Continuous-Energy			
Burnup (GWd/MTIHM)	TRITON		Exnihilo		Abs. Diff. (pcm)
	k_{eff}	σ (pcm)	k_{eff}	σ (pcm)	
0.00	1.23885	23	1.23890	9	5
0.05	1.20101	27	1.20057	9	44
0.20	1.19438	23	1.19491	9	51
0.45	1.18765	20	1.89070	8	142
0.80	1.18419	23	1.18395	9	24
1.25	1.17953	21	1.17917	9	36

moderator was decreased from 1300ppm to 600ppm, and the 0.04-cm spacer around the lower and right edges of the assembly was not modeled. Within the Exnihilo model, each pin was modeled independently to demonstrate Exnihilo's capability to handle multiple depletion materials. A graphical representation of this is provided

Table 8.25: Comparison of the power and flux in the fuel versus a two-dimensional multigroup TRITON calculation for VERA problem 1c.

FUEL		
Burnup (GWd/MTIHM)	Power (% Diff.)	Flux (% Diff.)
0.05	0.0061	1.3404
0.20	0.0060	1.0943
0.45	0.0071	1.0570
0.80	0.0082	0.7040
1.25	0.0099	0.7691
CLADDING		
Burnup (GWd/MTIHM)	Power (% Diff.)	Flux (% Diff.)
0.05	1.1260	1.4023
0.20	0.9200	1.1532
0.45	1.0301	1.1154
0.80	1.2666	1.2457
1.25	1.5023	1.3326
MODERATOR		
Burnup (GWd/MTIHM)	Power (% Diff.)	Flux (% Diff.)
0.05	0.4756	0.8338
0.20	0.4825	0.5852
0.45	1.9984	0.5496
0.80	1.6490	0.6772
1.25	2.3051	0.7616

in Figure 8.18a. Unfortunately this was not a feasible option for the multigroup TRITON calculation since independent materials requires independent cross section sets for each material which leads to an overflow in memory. Therefore TRITON utilized symmetry to replicated the materials as displayed in Figure 8.18b. The relevant parameters for this TRITON calculation include $S_N = 8$, a 4×4 mesh is used

Table 8.26: Exnihilo transport results for the VERA 2a quarter-assembly model compared against a two-dimensional TRITON calculation.

Burnup (GWd/MTIHM)	TRITON		Exnihilo		Abs. Diff. (pcm)	
	k_{eff}	σ (pcm)	k_{eff}	σ (pcm)		
0.00	1.26664	-	1.27009	10	10	345
4.76	1.16868	-	1.17401	11	11	533

on each pin cell[*], convergence criteria of 1×10^{-4} was applied to the inner and outer iterations of the transport solution, and a convergence criteria of 1×10^{-5} was used for the eigenvalue. This burnup parameters includes a single burnup cycle of 306 days with a 71 day down time and a average specific power of 32.12 MW/MTIHM.

Table 8.26 displays the comparison of the transport k_{eff} results from Exnihilo and TRITON.

Figure 8.19 displays the pin power peaking factors calculated by Exnihilo at 4.76 GWd/MTIHM. Note that no symmetry was used to reduce the number of regions in the quarter-assembly model in order to demonstrate the robustness of the Exnihilo transport-depletion calculation. However, since the TRITON calculation did utilize symmetry, Finally to generate a comparison versus TRITON, the powers are averaged using $1/8^{th}$-assembly symmetry and compared in Figure 8.20.

Because direct comparisons between Exnihilo and TRITON could not be made for this problem (since TRITON is also in the development phase for its continuous-energy depletion capability), additional burnup cycles were not run for this assembly model.

8.9 Core Demonstration

Unfortunately the current development status of Exnihilo does not enable a full-core depletion models to be constructed. Significant reduction of the SCEMPP memory

[*] For half-sized pin cells a 2 × 2, 2 × 4, or 4 × 2 mesh was used depending on the size of the truncated pin cell.

footprint is required to enable continuous-energy transport-depletion calculations for HPC applications using Exnihilo. Table D.3 in Appendix D shows the memory required to store the cross section data for nuclides that are typically tracked in depletable materials. The total amount of memory each processor must reserve to load these nuclides is approximately 2.2 GB [107]. Noting that the amount of memory available per processor core for most shared memory computational clusters is approximately 2 GB, and considering that additional memory must also be available to store geometry, tally, and material data, the memory footprint of a continuous-energy transport-depletion calculation it too large to take full advantage of HPC architectures. It should be noted however that reduction of the SCEMPP memory footprint is currently underway at ORNL and a full-core calculation will likely be possible by 2014.

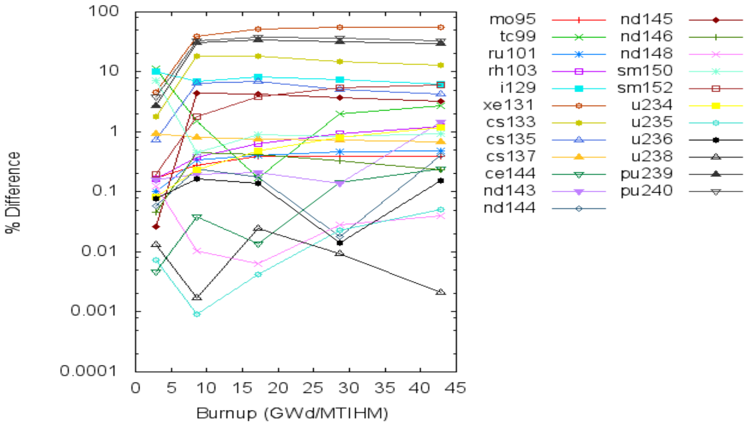


Figure 8.16: Isotopic analysis for VERA case 1c.

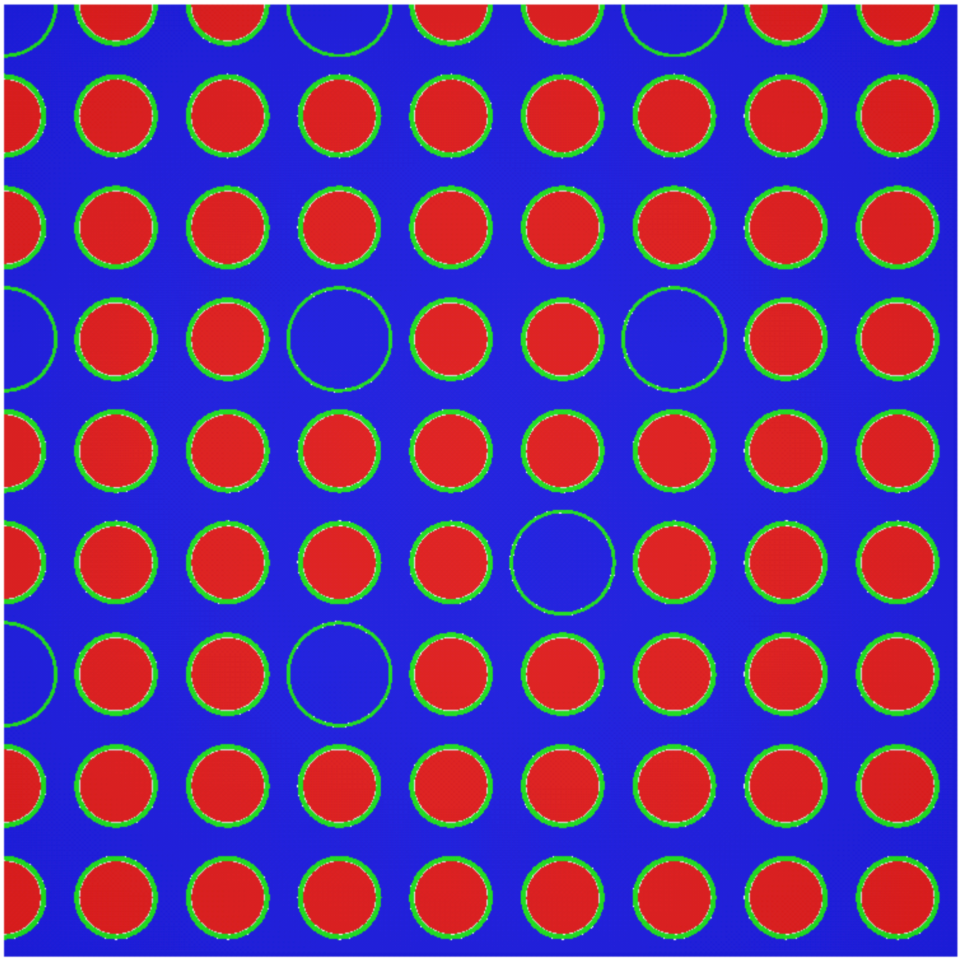
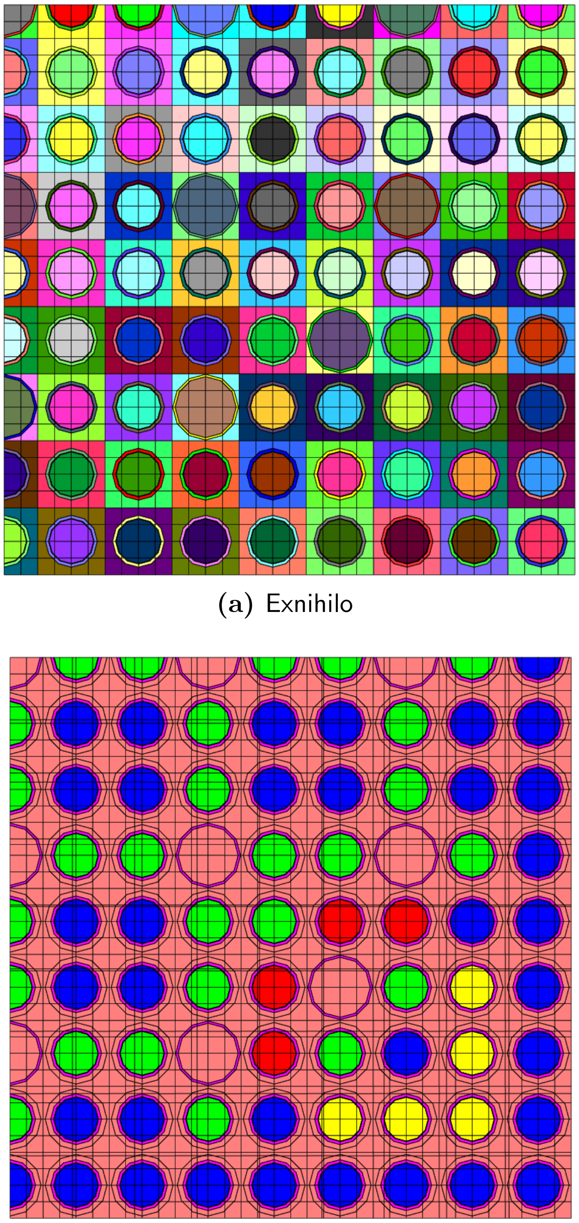


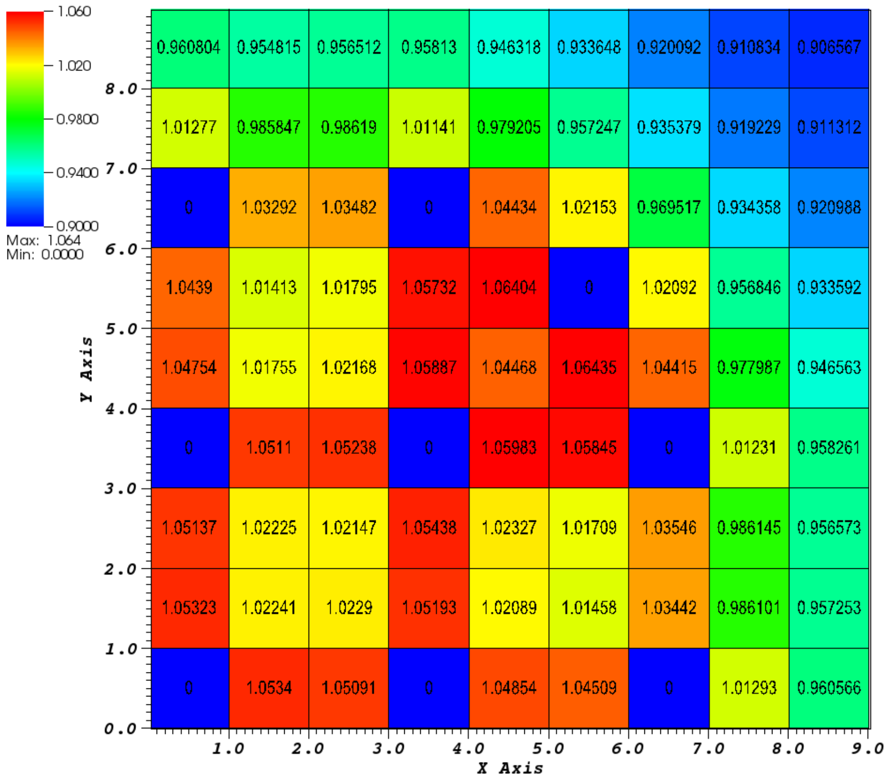
Figure 8.17: Horizontal cross section of the quarter-assembly benchmark model [108].



(a) Exnihilo

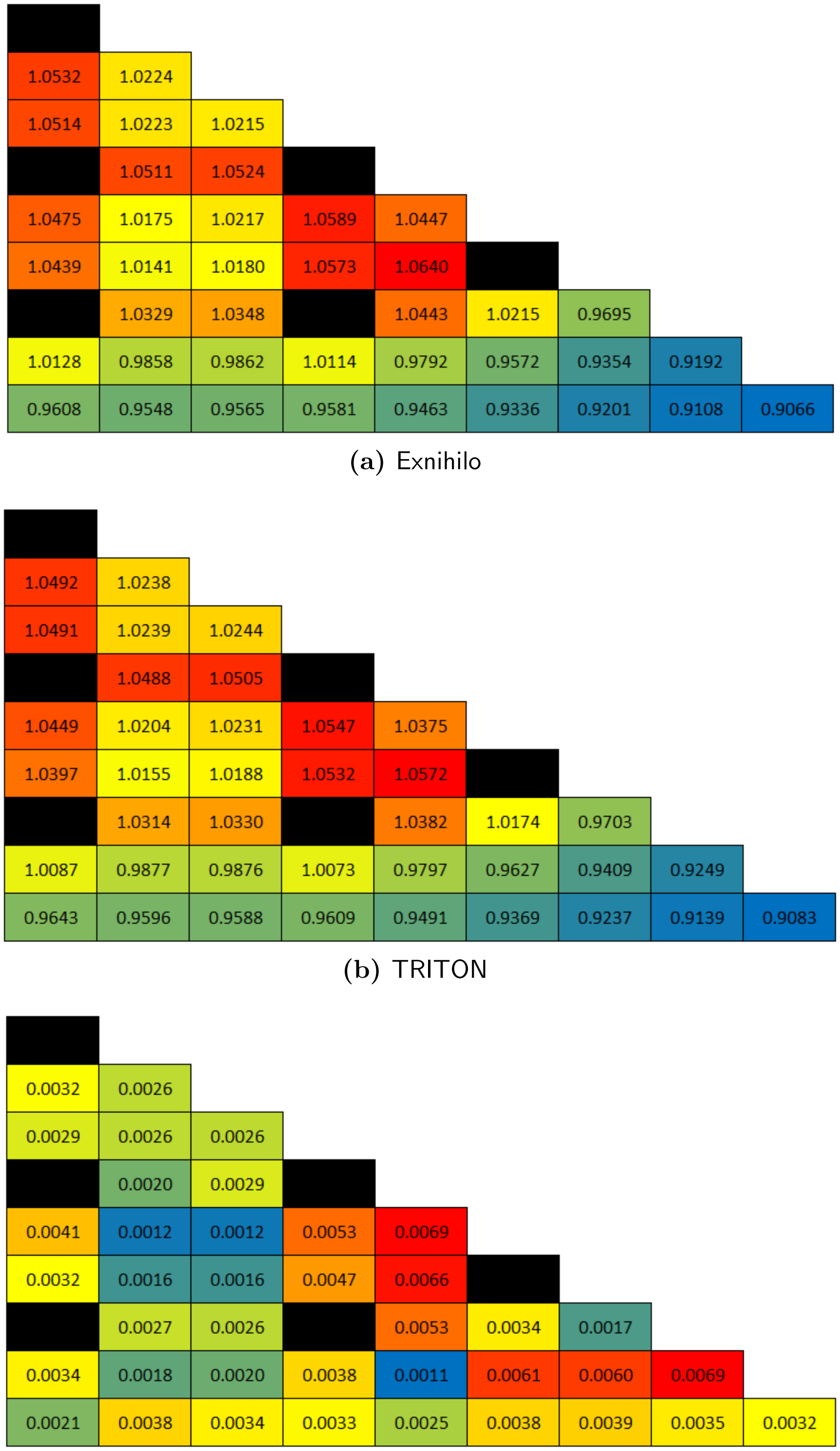
(b) TRITON

Figure 8.18: Exnihilo and TRITON models for VERA problem 2a.



0.960804	0.954815	0.956512	0.95813	0.946318	0.933648	0.920092	0.910834	0.906567
1.01277	0.985847	0.98619	1.01141	0.979205	0.957247	0.935379	0.919229	0.911312
0	1.03292	1.03482	0	1.04434	1.02153	0.969517	0.934358	0.920988
1.0439	1.01413	1.01795	1.05732	1.06404	0	1.02092	0.956846	0.933592
1.04754	1.01755	1.02168	1.05887	1.04468	1.06435	1.04415	0.977987	0.946563
0	1.0511	1.05238	0	1.05983	1.05845	0	1.01231	0.958261
1.05137	1.02225	1.02147	1.05438	1.02327	1.01709	1.03546	0.986145	0.956573
1.05323	1.02241	1.0229	1.05193	1.02089	1.01458	1.03442	0.986101	0.957253
0	1.0534	1.05091	0	1.04854	1.04509	0	1.01293	0.960566

Figure 8.19: Pin power peaking factors calculated by Exnihilo for the VERA 2a quarter assembly model at 4.76 GWd/MTIHM.



■								
1.0532	1.0224							
1.0514	1.0223	1.0215						
■	1.0511	1.0524	■					
1.0475	1.0175	1.0217	1.0589	1.0447				
1.0439	1.0141	1.0180	1.0573	1.0640	■			
■	1.0329	1.0348	■	1.0443	1.0215	0.9695		
1.0128	0.9858	0.9862	1.0114	0.9792	0.9572	0.9354	0.9192	
0.9608	0.9548	0.9565	0.9581	0.9463	0.9336	0.9201	0.9108	0.9066

(a) Exnihilo

■								
1.0492	1.0238							
1.0491	1.0239	1.0244						
■	1.0488	1.0505	■					
1.0449	1.0204	1.0231	1.0547	1.0375				
1.0397	1.0155	1.0188	1.0532	1.0572	■			
■	1.0314	1.0330	■	1.0382	1.0174	0.9703		
1.0087	0.9877	0.9876	1.0073	0.9797	0.9627	0.9409	0.9249	
0.9643	0.9596	0.9588	0.9609	0.9491	0.9369	0.9237	0.9139	0.9083

(b) TRITON

■								
0.0032	0.0026							
0.0029	0.0026	0.0026						
■	0.0020	0.0029	■					
0.0041	0.0012	0.0012	0.0053	0.0069				
0.0032	0.0016	0.0016	0.0047	0.0066	■			
■	0.0027	0.0026	■	0.0053	0.0034	0.0017		
0.0034	0.0018	0.0020	0.0038	0.0011	0.0061	0.0060	0.0069	
0.0021	0.0038	0.0034	0.0033	0.0025	0.0038	0.0039	0.0035	0.0032

(c) $abs(\frac{C}{E}) - 1$

Figure 8.20: Comparison of the power distribution at 4.76 GWd/MTIHM versus TRITON for the VERA 2a quarter assembly model.

Chapter 9

Conclusions

9.1 Summary of Conclusions

The primary goal of this research and development presented in this dissertation is to enable Monte Carlo depletion capabilities for HPC architectures. Accomplishing this goal involved first conducting research and analysis on current tally systems and statistical algorithms in modern Monte Carlo codes. Analysis of general Monte Carlo practice and the issues which lead to underprediction in the uncertainty of localized tallies in Monte Carlo eigenvalue calculations was performed to aid in developing the statistical algorithms for Shift's Monte Carlo transport module. Underprediction of Monte Carlo tally uncertainties was found to have at least three components: one from accumulating tallies using an ill-converged fission source, one from undersampling the fissionable regions, and one from cycle-to-cycle correlations. An emphasis on differentiating between these components was presented along with their relative effects and behavior with respect to the amount of underprediction that may exist in uncertainty estimates.

The knowledge gained from this analysis was applied to the development of Shift's Monte Carlo statistical algorithms. Algorithms were developed to calculate the Shannon entropy of the fission source distribution in domain-replicated, domain-decomposed, and multi-set overlapping domain decomposed environments, as the importance of monitoring the entropy of the fission source distribution in addition to k_{eff} as a function of cycle was made clear by the analyses of the Organization for Economic Cooperation and Development (OECD) benchmark models. Regarding the algorithms for estimating tally statistics, analyses have shown that assuming histories do not reenter domains they have previously left provides reasonably estimated variances in domain-decomposed problems without adding additional synchronization points, communication between processors, or memory allocation. The bias associated with the assumption that particles do not reenter their source domain has been evaluated for the C5G7 reactor benchmark case. Results illustrated that the bias can be mitigated by ensuring the size of the overlapped region is greater than the average mean free path of a neutron, the probability of a neutron making additional tally contributions to cells on a domain it had previously left can be significantly reduced. This methodology is anticipated to produce favorable results for domain-decomposed full-core simulations because it eliminates the need for additional computational time and resources.

Following the integration of these statistical algorithms with the development on Shift's tally system performed by the Exnihilo team at ORNL, integration of ORIGEN depletion capabilities into the Exnihilo framework was performed. This involved first supplementing the revised version-controlled ORIGEN code which had been recently developed at ORNL with some additional capabilities and infrastructure. Development of a modular interface to ARP was performed to provide clients using the modular ORIGEN code with the ability to interpolate between different ORIGEN libraries. This implementation provides Exnihilo and other codes with the ability to replicate the standalone depletion capabilities available within SCALE. Using these two modular interfaces, a standalone depletion capability was developed within Exnihilo which is capable of producing results that are within the limits of the numerical precision quoted by ORIGEN, which is 5×10^{-5}. A Python front-end interface was then developed to make this standalone depletion capability available to the user, and results for several benchmark cases ensured that the estimated nuclide concentrations were indeed within the numerical precision of results output from a standalone ORIGEN calculation.

Loading of the ORIGEN library data from disk was parallelized to enable ORIGEN depletion for HPC architectures. The individual data libraries which make up the ORIGEN binary library were linked with the necessary infrastructure which allows the data to be packed up and sent as a serialized stream to other processors using the Message Passing Interface (MPI). ORIGEN was then modified to utilize this new capability alongside the capabilities provided by the modular API, therefore allowing clients to load their own ORIGEN binary library and use it to populate the data structures within ORIGEN.

Following the development efforts within SCALE, a depletion package was developed within the Exnihilo code suite. This package enables coupled transport-depletion calculations to be performed using any of the transport modules available within Exnihilo or other code suites. The depletion package is capable of handling the various parallel decompositions within the Exnihilo framework including domain replication and domain decomposition. Additional parallel mechanics were developed within the depletion package itself to increase the efficiency of the depletion package in the presence of a domain replicated geometry.

An ultrafine-group flux binning method has been implemented in the depletion package that utilizes a logarithmic group structure. Studies were performed to show the effects of the number of energy groups on the computational time, memory, and accuracy of the calculation. These results were analyzed and a 43 000-group structure was determined to be the ideal number of groups for the ultrafine-group flux tally. This methodology provides the depletion package with the capability of running using multigroup cross sections provided from an AMPX working library or continuous-energy cross sections obtained from the SCEMPP physics package.

A Python front-end was subsequently developed which coupled the depletion package to the Shift Monte Carlo framework to provide a user interface for coupled transport-depletion calculations. Coupled Shift-depletion calculations were performed using a variety of benchmark models to validate the power calculation, one-group cross section generation, and coupling to the transport and depletion processes. Results were presented comparing k_{eff}, power distributions, and the isotopic concentrations for select nuclides, and these results proved to be in reasonable agreement with multigroup and continuous-energy TRITON calculations.

9.2 Suggestions for Future Work

Although the research and development efforts performed and documented within this dissertation were extensive, there remains a significant amount of additional research and analyses that can be performed to enhance and improve upon the developed tally statistical algorithms and depletion capabilities. Select research activities include:

1. Utilize Denovo to calculate the dominance ratio in order to identify problems which may suffer from the negative effects of cycle-to-cycle correlations.

2. The ability to modify ORIGEN library cross sections and data directly needs to be developed. In particular, development of capabilities to modify the fission yields and the cross sections that are only available in the JEFF data libraries need to be integrated into Exnihilo to increase the fidelity of the calculation. Because of this, a starting library must be loaded initially in order to populate the data that Exnihilo cannot calculate. Development of these capabilities will be required in order to improve the accuracy and fidelity of coupled transport-depletion calculations in Exnihilo. Although these capabilities do no greatly affect reactor neutronics calculations, their effect may still be significant. These capabilities however will be required in order to use Exnihilo

for isotopic production analyses or in calculations where the production of secondary particles is of interest.

3. The capabilities made available by the ArpInterface package for retrieving burnup positions and interpolating between the distributed libraries should ideally be integrated directly into the ORIGEN package. By doing this, ORIGEN will have greater control over data storage and can minimize data duplication. Furthermore, the need to allocate and deallocate multiple objects and traverse multiple binding layers can be circumvented via direct integration with ORIGEN.

4. Note that as the Insilico framework is currently in the development stages, it does not contain some of the necessary infrastructure to benchmark depletion calculations against experimental data. The primary feature that needs to be developed within the Insilico package is the ability to modify material and geometric properties at specific time steps. Adding this feature will enable branch calculations as well as the ability to specify time-dependent material property changes such as changes in temperature and boron feed rates in the moderator. These features are required in order to test depletion within Exnihilo against experimental data.

5. Although the use of a predictor-corrector approach increases the fidelity of the depletion calculation, this approach uses the assumption that the absorption rate is constant over the time step. Due to the huge absorption cross sections of isotopes such as ^{155}Gd and ^{157}Gd, the absorption rates vary strongly as the material is depleted, and the classical predictor-corrector does not accurately capture this phenomena [102]. The solution may require a variation of the predictor-corrector algorithm in order to more accurately predict the isotopic concentrations of these isotopes.

6. Sensitivity and uncertainty analyses can be performed on estimated isotopic concentrations to help improve the fidelity of the results. The uncertainties

would be calculated in relation to the ultrafine-group flux-binning methodology employed to perform the depletion calculation. The question remains, however, of whether or not it would be worth the extra computational time and resources to provide this metric for all isotopes, or whether providing this metric for actinides, key absorbers, and select isotopes of interest would be acceptable. Currently however, the ultrafine-group flux tally uncertainties are not used in any subsequent calculation nor are they made available to the user.

7. More in-depth analysis of the number of energy groups to use for the ultrafine-group flux tally. One could examine, for example, the number of resonances captured by different group structures for important cross sections such as the ^{235}U fission cross section, ^{238}U capture cross section, and other cross sections which result in important transitions from one isotope to another. An analysis similar to the one presented in Section 8.6.1 could be carried out for multiple different reactor types to identify if a set of guidelines can be developed for any model or if different reactor types require more energy groups than others.

8. Variance reduction techniques may need to be applied to increase the sample size of neutrons in the low-flux (¡ 0.005 eV) and high-flux (¿ 5 MeV) energy groups. This is especially important since fission cross sections tend to be high at low energies while threshold reactions tend to be high in the a high energies.

9. Improved scaling studies once memory footprint is reduced. Currently the 5×5 array model that contains 700 regions would require approximately 5 GB per core to run, and thus scaling studies are not possible on the distributed memory systems available.

10. Additional development of the coupling between Exnihilo and the cross section processing modules within SCALE is required to enable coupled multistep, multigroup depletion calculations.

11. Additional testing and validation of the different options available within the depletion package is required. These options include, but are not limited to, depletion by constant flux, adding different sets of nuclides to each depletable material, and performing the power calculation without considering the power generated from capture reaction.

12. A significant reduction in the memory footprint of the continuous-energy data is required to ensure the memory on each node will be below 2 GB. This conforms with the amount of memory available per processor on most HPC architectures.

References

[1] T. M. Evans, A. S. Stafford, R. N. Slaybaugh, and K. T. Clarno, "Denovo: A new three-dimensional parallel discrete ordinates code in scale," *Nuclear technology*, vol. 171, no. 2, pp. 171–200, 2010. xv, 4, 59

[2] X-5 Monte Carlo Team, "MCNP–a general purpose monte carlo n-particle transport code," Tech. Rep. LA-UR-03-1987, Los Alamos National Laboratory, April 2003. Version 5. 3, 10, 11, 49, 66

[3] D. Hollenbach, L. Petrie, S. Goluoglu, N. Landers, and M. Dunn, "KENO-VI: A general quadratic version of the keno program," tech. rep., Oak Ridge National Laboratory, Oak Ridge, TN, 2011. xv, 3, 28, 51, 66, 69

[4] I. C. Gauld, "ORIGEN-S: Depletion module to calculate neutron activation, actinide transmutation, fission product generation, and radiation source terms," tech. rep., Oak Ridge National Laboratory, Oak Ridge, TN, June 2011. 6, 14, 15, 51, 52, 61, 117, 141

[5] J. M. Levesque, "Application development for titan-a multi-petaflop hybrid-multicore mpp system," in *High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion:*, pp. 1731–1821, IEEE, 2012. 6, 60, 166

[6] J. E. Hoogenboom, W. R. Martin, and B. Petrovic, "The monte carlo performance benchmark test-aims, specifications and first results," in *International Conference on Mathematics and Computational Methods Applied to*, vol. 2, p. 15, 2011. 10, 20

[7] B. Mervin, S. Mosher, J. Wagner, and G. Maldonado, "Uncertainty underprediction in monte carlo eigenvalue calculations," *Nuclear Science and Engineering*, vol. 173, pp. 276–292, 2013. 10, 65

[8] F. Brown and T. Sutton, "Monte carlo fundamentals," tech. rep., Knolls Atomic Power Lab., Schenectady, NY (United States), 1996. 10, 12, 27

[9] T. Ueki and F. B. Brown, "Informatics approach to stationarity diagnostics of the monte carlo fission source distribution," *Transaction of the American Nuclear Society*, vol. 89, pp. 458–461, 2003. 10, 30, 72

[10] F. B. Brown, "A review of monte carlo criticality calculations–convergence, bias, statistics," in *International Conference on Mathematics, Computational Methods & Reactor Physics (M&C 2009)*, pp. 3–7, 2009. 11

[11] T. Ueki and F. B. Brown, "Stationarity diagnostics using shannon entropy in monte carlo criticality calculation i: F test," *Trans. Am. Nucl. Soc*, vol. 87, pp. 156–158, 2002. 11

[12] K. E. Knight, "Changes in computer performance," *Datamation*, vol. 12, no. 9, pp. 40–54, 1966. 11

[13] A. Siegel, K. Smith, P. Fischer, and V. Mahadevan, "Analysis of communication costs for domain decomposed monte carlo methods in nuclear reactor analysis," *Journal of Computational Physics*, vol. 231, no. 8, pp. 3119–3125, 2012. 12, 43

[14] J. Verschelde, "Introduction to supercomputing," tech. rep., University of Illinois, Chicago, 2012. xv, 13

[15] J. Vilches, "Amd announces radeon hd 7970, claim fastest gpu title," December 2011. Retrieved on May 26, 2013. 13

[16] M. Brown, "Nvidia's geforce gtx 780: a titan for the rest of us," May 2013. Retrieved on May 26, 2013. 13

[17] O. Phatak, "Intel core i3 vs. i5 vs. i7," May 2013. Retrieved on May 26, 2013. 13

[18] A. G. Nelson, *Monte carlo methods for neutron transport on graphics processing units using CUDA*. PhD thesis, The Pennsylvania State University, 2009. 13, 14

[19] F. Brown, "Recent advances and future prospects for monte carlo," in *Joint Int. Conf. Supercomput. Nucl. Appl. Monte Carlo (SNA+ MC2010), Tokyo*, 2010. 14

[20] A. Uhlherr, "Parallel monte carlo simulations by asynchronous domain decomposition," *Computer physics communications*, vol. 155, no. 1, pp. 31–41, 2003. 14

[21] P. Dyck, "Practices and developments in spent fuel burnup credit applications," Tech. Rep. IAEA-TECDOC-1378, International Atomic Energy Agency, April 2002. 14, 15

[22] J. Cetnar and P. Gronek, "Bison-c: A one-dimensional transport and burnup calculation code with consideration of actinides and fission products," *Nuclear science and engineering*, vol. 134, no. 2, pp. 236–237, 2000. 14, 15

[23] T. J. D. D. A. Barber, R. M. Miller, C.-H. L. T. K. Deokjung, L. Y. X. J. Gan, H. G. J. J. Y. Cho, K. Lee, and A. P. Ulses, *PARCS: Purdue Advanced Reactor Core Simulator*. 14, 15, 37

[24] W. Wilson, S. Cowell, T. England, A. Hayes, and P. Moller, *A manual for CINDER90 version 07.4 codes and data*, 2008. 14, 15

[25] D. I. Poston and H. R. Trellue, "Development of a fully-automated monte carlo burnup code monteburns," tech. rep., Los Alamos National Laboratory (LANL), Los Alamos, NM, 1999. 14, 15, 108

[26] E. Bomboni, N. Cerullo, E. Fridman, G. Lomonaco, and E. Shwageraus, "Comparison among mcnp-based depletion codes applied to burnup

calculations of pebble-bed htr lattices," *Nuclear Engineering and Design*, vol. 240, no. 4, pp. 918–924, 2010. 14, 15

[27] J. Leppänen, "Serpent–a continuous-energy monte carlo reactor physics burnup calculation code," tech. rep., VTT Technical Research Centre, August 2012. 14, 15

[28] M. A. Jessee and M. D. DeHart, "TRITON: A multipurpose transport, depletion, and sensitivity and uncertainty analysis module," Tech. Rep. ORNL/TM-2005/39, Oak Ridge National Laboratory, June 2011. Version 6.1, Sect. T1. 14, 15, 18, 56, 108, 134, 135

[29] M. L. Fensin, J. S. Hendricks, and S. Anghaie, "Enhanced monte-carlo-linked depletion capabilities in mcnpx," tech. rep., American Nuclear Society, 555 North Kensington Avenue, La Grange Park, IL 60526 (United States), 2006. 15

[30] Oak Ridge National Laboratory, *SCALE: A Modular Codes System for Performing Standardized Computer Analyses for Licensing Evaluation*, I–III ed., June 2011. Version 6, ORNL/TM-2005/39. 15, 17, 50

[31] M. Chadwick, P. Obložinskỳ, M. Herman, N. Greene, R. McKnight, D. Smith, P. Young, R. MacFarlane, G. Hale, S. Frankle, *et al.*, "Endf/b-vii. 0: Next generation evaluated nuclear data library for nuclear science and technology," *Nuclear Data Sheets*, vol. 107, no. 12, pp. 2931–3060, 2006. 15, 49, 52

[32] A. Santamarina, D. Bernard, P. Blaise, M. Coste, A. Courcelle, T. Huynh, C. Jouanne, P. Leconte, O. Litaize, S. Mengelle, *et al.*, "The jeff-3.1. 1 nuclear data library," *JEFF Report*, vol. 22, no. 10.2, p. 2, 2009. 15, 17

[33] K. T. Clarno, B. Philip, W. K. Cochran, R. S. Sampath, S. Allu, P. Barai, S. Simunovic, M. A. Berrill, L. J. Ott, S. Pannala, *et al.*, "The amp (advanced multiphysics) nuclear fuel performance code," *Nuclear Engineering and Design*, vol. 252, pp. 108–120, 2012. 16

[34] P. J. Turinsky, R. M. Al-Chalabi, P. Engrand, H. N. Sarsour, F. X. Faure, and W. Guo, "Nestle: Few-group neutron diffusion equation solver utilizing the nodal expansion method for eigenvalue, adjoint, fixed-source steady-state and transient problems," tech. rep., EG and G Idaho, Inc., Idaho Falls, ID (United States); Los Alamos National Lab., NM (United States), 1994. 16, 37

[35] G. Yesilyurt, K. T. Clarno, I. C. Gauld, and J. Galloway, "Modular origen-s for multi-physics code systems," in *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011)*, May 8–12, 2011. Rio de Janeiro, Brazil. xv, 19, 112

[36] I. C. Gauld and D. Wiarda, "COUPLE: A nuclear decay and cross section data processing code for creating origen-s libraries," Tech. Rep. ORNL/TM-2005/39, Oak Ridge National Laboratory, June 2011. Version 6.1, Sect. F6. 18

[37] K. Smith, "Reactor core methods plenary presentation," in *PHYSOR 2012 -Advances in Reactor Physics - Linking Research, Industry, and Education,*, (Knoxville, TN), 2012. 19, 25, 42

[38] W. R. Martin, "Advances in monte carlo methods for global reactor analysis," in *Invited lecture at the M&C 2007 International Conference, Monterey, CA, USA*, 2007. 20, 42

[39] P. K. Romano and B. Forget, “Parallel fission bank algorithms in monte carlo criticality calculations,” *Nuclear Science and Engineering*, vol. 170, pp. 125–135, February 2012. 20

[40] J. J. Duderstadt and L. J. Hamilton, *Nuclear Reactor Analysis.* Wiley, 1976. 22, 46, 48

[41] E. E. Lewis, *Computational Methods of Neutron Transport.* Wiley-Interscience, 1993. 22

[42] W. M. Stacey, *Nuclear Reactor Physics (Second edition).* Wiley-VCH, 2007. 22, 47, 48

[43] K. D. Kok, ed., *Nuclear Engineering Handbook (Mechanical and Aerospace Engineering Series).* CRC Press, 2009. 23, 38

[44] A. Siegel, K. Smith, P. Romano, B. Forget, and K. Felker, “The effect of load imbalances on the performance of monte carlo algorithms in lwr analysis,” *Journal of Computational Physics*, vol. 235, pp. 901–911, 2012. 26, 41

[45] H. L. Anderson, “Metropolis, monte carlo, and the maniac,” *Los Alamos Science*, vol. 14, pp. 96–108, 1986. 26

[46] W. Feller, *An Introduction to Probability Theory and Its Applications, Vol. 1, 3rd Edition.* Wiley, 1968. 27, 42

[47] W. Navidi, *Statistics for Engineers And Scientists.* Mcgraw-Hill College, 2007. 28, 29

[48] M. Lee, H. Joo, D. Lee, and K. Smith, “A feasibility study of cmfd acceleration in monte carlo eigenvalue calculation,” in *Trans. of KNS Autumn Meeting, CD-ROM*, 2009. xv, 30, 31, 32

[49] T. Ueki and F. B. Brown, "Stationarity modeling and informatics-based diagnostics in monte carlo criticality calculations," *Nuclear science and engineering*, vol. 149, no. 1, pp. 38–50, 2005. 31

[50] L. Carter and N. McCormick, "Source convergence in monte carlo calculations," tech. rep., Univ. of Washington, Seattle, 1969. 31

[51] T. Kitada and T. Takeda, "Effective convergence of fission source distribution in monte carlo simulation," *Journal of Nuclear Science and Technology*, vol. 38, no. 5, pp. 324–329, 2001. 31

[52] T. Sutton, P. Romano, and B. Nease, "On-the-fly monte carlo dominance ratio calculation using the noise propagation matrix," *Nuclear Science and Technology*, vol. 2, pp. 749–756, 2011. 31, 89

[53] R. Brissenden and A. Garlick, "Biases in the estimation of k¡ sub¿ eff¡/sub¿ and its error by monte carlo methods," *Annals of Nuclear Energy*, vol. 13, no. 2, pp. 63–83, 1986. 31, 32

[54] J. C. Wagner, E. D. Blakeman, and D. E. Peplow, "Forward-weighted cadis method for global variance reduction," *TRANSACTIONS-AMERICAN NUCLEAR SOCIETY*, vol. 97, p. 630, 2007. xv, 31, 33

[55] E. M. Gelbard and R. Prael, "Monte carlo work at argonne national laboratory," tech. rep., Argonne National Lab., Ill.(USA), 1974. 33

[56] S. Langenbuch, A. Seubert, and W. Zwermann, "High accuracy large scale monte carlo and deterministic transport calculations for critical systems," in *Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications (M&C+ SNA 2007)*, pp. 15–19, 2007. 37

[57] T. M. Evans, G. G. Davidson, and J. J., “Design of a neutronics package for multi-physics reactor calculations,” Tech. Rep. RNSD-TN-11-004, Oak Ridge National Laboratory, April 2013. xv, 38

[58] M. Lee, H. G. Joo, D. Lee, and K. Smith, “Multigroup monte carlo reactor calculation with coarse mesh finite difference formulation for real variance reduction,” in *Joint International Conference on Supercomputing in nuclear Applications and Monte Carlo*, pp. 17–21, 2010. 39, 69

[59] C. Gregg and K. Hazelwood, “Where is the data? why you cannot debate cpu vs. gpu performance without the answer,” in *Performance Analysis of Systems and Software (ISPASS), 2011 IEEE International Symposium on*, pp. 134–144, IEEE, 2011. 39

[60] J. Kwiatkowski, “Evaluation of parallel programs by measurement of its granularity,” in *Parallel Processing and Applied Mathematics*, pp. 145–153, Springer, 2006. 39

[61] G. M. Amdahl, “Validity of the single processor approach to achieving large scale computing capabilities,” in *Proceedings of the April 18-20, 1967, spring joint computer conference*, pp. 483–485, ACM, 1967. 40

[62] N. Schramm and A. Sabo, “Concurrent programming method for embedded systems,” in *9th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics*, 2008. 41

[63] D. W. Walker and J. J. Dongarra, “Mpi: a standard message passing interface,” *Supercomputer*, vol. 12, pp. 56–68, 1996. 41

[64] J. Basney, R. Raman, and M. Livny, “High throughput monte carlo,” in *Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing*, Citeseer, 1999. 43

[65] J. C. Wagner, S. W. Mosher, T. M. Evans, D. E. Peplow, and J. A. Turner, "Hybrid and parallel domain-decomposition methods development to enable monte carlo for reactor analyses," *Progress in Nuclear Science and Technology*, vol. 2, pp. 815–820, 2011. 44, 60, 91, 93, 97

[66] P. S. Brantley and L. M. Stuart, "Monte carlo particle transport capability for inertial confinement fusion applications," in *Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Application (M&C+ SNA 2007), American Nuclear Society, Monterey, California*, 2006. 45

[67] T. A. Brunner and P. S. Brantley, "An efficient, robust, domain-decomposition algorithm for particle monte carlo," *Journal of Computational Physics*, vol. 228, no. 10, pp. 3882–3890, 2009. 45

[68] K. S. Krane, *Introductory Nuclear Physics*. Wiley, 1987. 46

[69] L. M. Petrie, R. A. Lefebvre, and D. Wiarda, "Standard composition library," Tech. Rep. ORNL/TM-2005/39, Oak Ridge National Laboratory, June 2011. Version 6.1, Sect. M8. 51, 134

[70] I. C. Gauld, S. M. Bowman, and J. E. Horwedel, "ORIGEN-ARP: Automatic rapid processing for spent fuel depletion, decay, and source term analysis," Tech. Rep. ORNL/TM-2005/39, Oak Ridge National Laboratory, June 2011. 55, 126, 135

[71] N. M. Greene, L. M. Petrie, and M. L. Williams, "XSDRNPM: A one-dimensional discrete-ordinates code for transport analysis," Tech. Rep. ORNL/TM-2005/39, Oak Ridge National Laboratory, June 2011. Version 6.1, Sect. F3. 56

[72] M. Jessee and M. DeHart, "Newt: A new transport algorithm for two-dimensional discrete-ordinates analysis in non-orthogonal geometries," Tech.

Rep. ORNL/TM-2005/39, Oak Ridge National Laboratory, June 2011. Version 6.1, Sect. F21. 56

[73] N. M. Greene and M. E. Dunn, "User's guide for ampx utility modules," Tech. Rep. ORNL/TM-2005/39, Oak Ridge National Laboratory, June 2011. Version 6.1, Sect. M15. 56

[74] S. M. Bowman and M. E. Dunn, "SCALE cross-section libraries," Tech. Rep. ORNL/TM-2005/39, Oak Ridge National Laboratory, June 2011. Version 6.1, Sect. M4. xviii, 56, 57, 111, 117, 152

[75] M. Dunn and N. Greene, "Ampx-2000: A cross-section processing system for generating nuclear data for criticality safety applications," *Trans. Am. Nucl. Soc*, vol. 86, pp. 118–119, 2002. 56

[76] R. MacFarlane and D. Muir, "Njoy99. 0 code system for producing pointwise and multigroup neutron and photon cross sections from endf/b data," Tech. Rep. PSR-480/NJOY99.0, Los Alamos National Laboratory, March 2000. 56

[77] J. Simpson, *The Oxford English Dictionary (20 Volume Set) (Vols 1-20)*. Oxford University Press, USA, 1989. 57

[78] T. Evans, "Exnihilo overview," tech. rep., Oak Ridge National Laboratory, 2012. 57

[79] B. Meyer, "Applying'design by contract'," *Computer*, vol. 25, no. 10, pp. 40–51, 1992. 57, 64

[80] D. Peplow, "MAVRIC : MONACO with automated variance reduction using importance calculations," Tech. Rep. ORNL/TM-2005/39, Oak Ridge National Laboratory, June 2011. 59

[81] W. Rhoades, R. Childs, M. Emmett, and S. Cramer, “Application of the three-dimensional oak ridge transport code,” tech. rep., Oak Ridge National Lab., TN (USA), 1984. 59

[82] N. C. Sly, B. T. Mervin, S. W. Mosher, T. M. Evans, J. C. Wagner, and G. I. Maldonado, “Verification of the shift monte carlo code with the c5g7 reactor benchmark,” in *PHYSOR 2012–Advances in Reactor Physics–Linking Research, Industry, and Education*, 2012. 60

[83] P. C. Robert G., P. T. Nicholas, R. G. Cochran, and W. F. Miller, *Nuclear Fuel Cycle : Analysis and Management.* Amer Nuclear Society, 1993. 61

[84] B. T. Rearden, “Modernization of scale.” NCSP Technical Seminar, Oak Ridge National Laboratory, March 2012. 61

[85] C. Multiphysics, “4.3 user’s guide,” 2012. 61

[86] S. Mosher and D. Peplow, “KGTLIB documentation.” HTML documentation generated by Doxygen. 62

[87] S. Mosher, “LAVA documentation.” HTML documentation generated by Doxygen. 62

[88] T. M. Evans and K. T. Clarno, “C++ coding sstandard for the amp project,” tech. rep., Oak Ridge National Laboratory, September 2009. 63

[89] S. Meyers, *Effective C++: 55 Specific Ways to Improve Your Programs and Designs (3rd Edition).* Addison-Wesley Professional, 2005. 63

[90] D. van Heesch, *Doxygen Manual for version 1.4. 6*, 2004. 64

[91] S. Goluoglu, L. M. Petrie, M. E. Dunn, D. F. Hollenbach, and B. T. Rearden, “Monte carlo criticality methods and analysis capabilities in scale,”

Nuclear Technology, vol. 174, no. 2, pp. 214–235, 2011. Accepted for publication on August 24, 2010. 69

[92] R. N. Blomquist, M. Armishaw, D. Hanlon, N. Smith, Y. Naito, J. Yang, Y. Mioshi, T. Yamamoto, O. Jacquet, and J. Miss, "Source convergence in criticality safety analyses," Tech. Rep. 92-64-02304-6, Nuclear Energy Agency, Organization for Economic Cooperation and Development, 2006. 70, 208

[93] T. Ueki, "Batch estimation of statistical errors in the monte carlo calculation of local powers," *Annals of Nuclear Energy*, vol. 38, no. 11, pp. 2462–2469, 2011. 87

[94] T. Ueki, F. B. Brown, D. K. Parsons, and J. S. Warsa, "Time series analysis of monte carlo fission sources-i: Dominance ratio computation," *Nuclear science and engineering*, vol. 148, no. 3, pp. 374–390, 2004. 89

[95] M. Bell, "Origen: The ornl isotope generation and depletion code," tech. rep., Oak Ridge National Laboratory, 1973. 112

[96] I. C. Gauld, G. Radulescu, G. Ilas, B. D. Murphy, M. L. Williams, and D. Wiarda, "Isotopic and decay methods and analysis capabilities in scale," *Nuclear Technology*, vol. 174, pp. 169–195, August 2011. 112

[97] L. M. Petrie, "Fido input system," Tech. Rep. ORNL/TM-2005/39, Oak Ridge National Laboratory, June 2011. Version 6.1, Sect. M10. 112

[98] S. Skutnik, F. Havluj, D. Lago, and I. Gauld, "Development of an object-oriented origen for advanced nuclear fuel modeling applications," in *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2013)*, May 5–9, 2013. 112

[99] S. M. Bowman and I. C. Gauld, “Origenarp primer: How to perform isotopic depletion and decay calculations with scale/origen,” Technical Report ORNL/TM-2010/43, Oak Ridge National Laboratory, August 2010. 114

[100] A. Trkov, M. Herman, and D. Brown, “Endf-6 formats manual,” tech. rep., Report BNL-90365-2009 Rev. 2, Brookhaven National Laboratory, Upton, New York, 2011. 129

[101] L. Cousin, W. Haeck, and B. Cochet, “Validating the vesta monte-carlo depletion interface using ariane chemical assay data for pressurized water reactor applications,” in *Proceedings of International Conference on the Physics of Reactors (PHYSOR), Pittsburgh, Pennsylvania, USA*, 2010. 130

[102] J. Rhodes, K. Smith, and D. Lee, “Casmo-5 development and applications,” in *Proc. ANS Topical Meeting on Reactor Physics (PHYSOR-2006)*, pp. 10–14, 2006. 130, 189

[103] F. Hoareau, N. Schwartz, and D. Couyras, “A predictor-corrector scheme for the microscopic depletion solver of the cocagne core code,” *Journal of Energy and Power Engineering*, vol. 6, pp. 369–378, 2012. 130

[104] D. M. Beazley *et al.*, “Swig: An easy to use tool for integrating scripting languages with c and c++,” in *Proceedings of the 4th USENIX Tcl/Tk workshop*, pp. 129–139, 1996. 132

[105] M. Folk, A. Cheng, and K. Yates, “Hdf5: A file format and i/o library for high performance computing applications,” in *Proceedings of Supercomputing*, vol. 99, 1999. 135

[106] M. L. Williams and D. F. Hollenbach, “Centrm: A one-dimensional neutron transport code for computing pointwise energy spectra,” Tech. Rep. ORNL/TM-2005/39, Oak Ridge National Laboratory, June 2011. Version 6.1, Sect. F18. 168

[107] C. Celik, March, 28 2013. Private Communication. xiv, xviii, 172, 184, 232

[108] A. T. Godfrey, "Vera core physics benchmark progression problem specifications," Tech. Rep. CASL-U-2012-0131-002, Consortium for Advanced Simulations of LWRs, March 2013. Revision 2. xviii, 175, 176, 177

[109] J. Wagner, "Computational benchmark for estimation of reactivity margin from fission products and minor actinides in pwr burnup credit," tech. rep., ORNL Oak Ridge National Laboratory (US), 2001. 208, 210

[110] E. Lewis, M. Smith, N. Tsoulfanidis, G. Palmiotti, T. Taiwo, and R. Blomquist, "Benchmark specification for deterministic 2-d/3-d mox fuel assembly transport calculations without spatial homogenization (c5g7 mox)," tech. rep., NEA/NSC, 2001. 210

[111] M. D. DeHart, M. C. Brady, and C. V. Parks, "Oecd/nea burnup credit cacalculation criticality benchmark phase i-b results," tech. rep., Oak Ridge National Laboratory, June 1996. NEA/NSC/DOC(96)-06, ORNL-6901. 222, 226

[112] W. Haeck, *VESTA User's Manual–Version 2.0.0.* Institut de Radioprotection et de Surete Nucleaire, 2.1.0 ed., 2008. 225

[113] I. C. Gauld and D. Wiarda, "ORIGEN-S data libraries," Tech. Rep. ORNL/TM-2005/39, Oak Ridge National Laboratory, Oak Ridge, TN, June June 2011. Version 6.1, Sect. M6. xiv, 236

Appendix

A Benchmark Models

This appendix described the four benchmark models used to perform the research and analyses presented in the following chapters. The benchmark models include a set of pin cell models with axial burnup distributions, a cask assembly model with an axial burnup distribution, a sixteen-assembly C5 MOX fuel assembly model with exterior water reflectors, and the Hoogenboom-Martin full-core benchmark model.

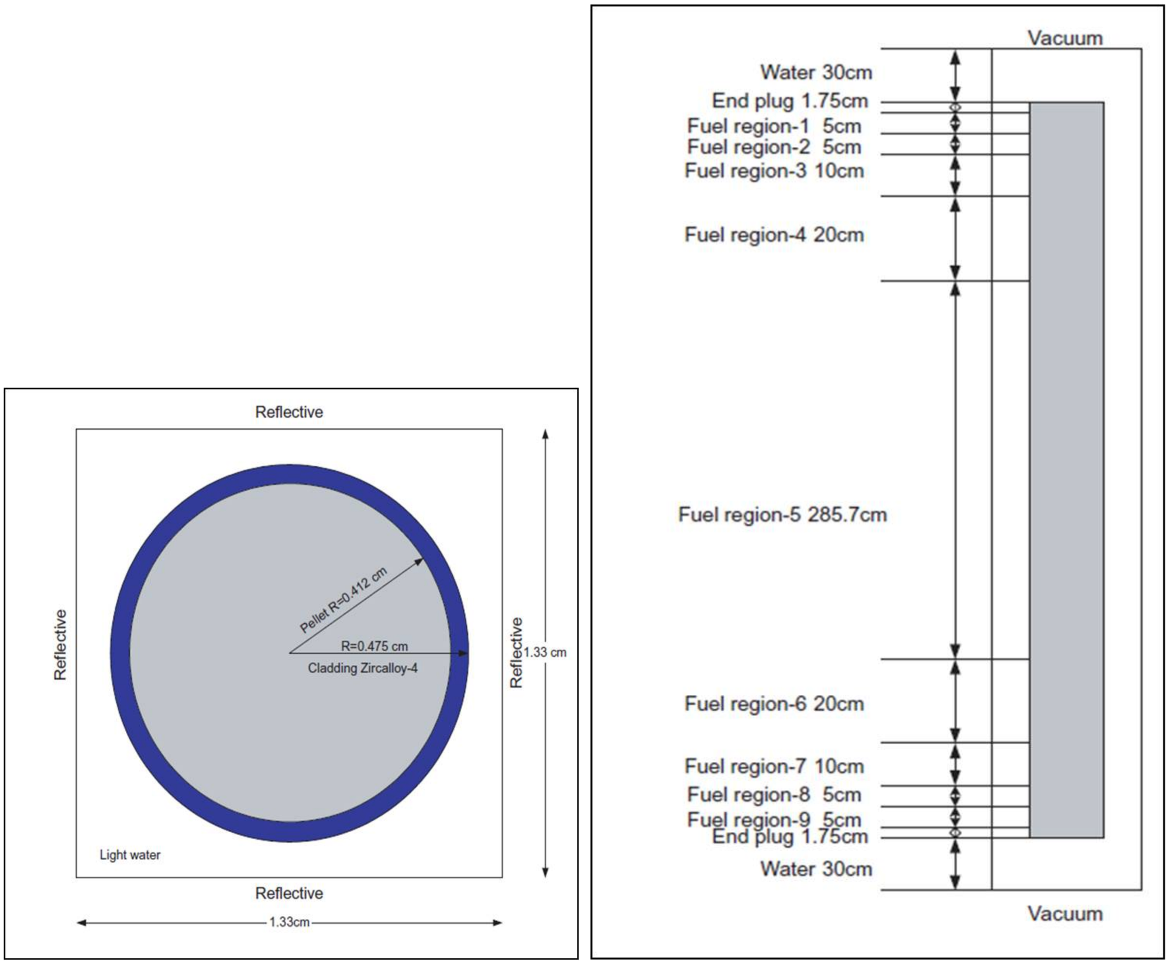
A.1 OECD/NEA Pin Cell Models

The first set of benchmark models, which correspond to cases 2.1–2.3 from Chapter 3 of Reference [92], are test problems devised by the OECD/Nuclear Energy Agency (NEA) Expert Group on Source Convergence in Criticality-Safety Analysis. This particular set of cases represent pin cell arrays with irradiated LWR fuel. A layout of the geometry for the pin cell configuration is given in Figure A.1.

The composition of the LWR spent fuel consists of more reactive, low burnup end regions separated by a long, less reactive, high burnup region. The fuel composition differs among the axial regions for each of the three cases: case 2.1 has a symmetric burnup profile, while cases 2.2 and 2.3 have higher burnups in one or more of the regions at the bottom of the pin cell (regions 6-9). All three benchmark cases used reflected boundary conditions in the horizontal dimensions and vacuum boundary conditions at the top and bottom. The compositions of the different fuel regions as well as their configuration within each pin cell are provided in Reference [92]. For tallying purposes, the model regions were divided into 5-cm axial regions and the fluxes were tallied using the 8-group structure provided in Table A.1.

A.2 GBC-32 Cask Assembly Model

The GBC-32 assembly model is a three-dimensional model of a generic PWR cask assembly (GBC-32) [109]. Similar to the OECD pin cell cases, this test problem is



(a) horizontal

(b) vertical

Figure A.1: Vertical and horizontal cross-sections for the OECD benchmark model.

composed of pins containing burned fuel and thus has an axial burnup distribution which is uniform over all of the fuel pins. The specific model used in these analyses represents an infinite array of 17 × 17 PWR assemblies with boral panels between each assembly. This was accomplished by placing reflected boundary conditions through the center of each boral panel on all horizontal sides. An illustration of a quarter-cask model is provided in Figure A.2 and a cross-sectional representation of a single cask assembly is provided in Figure A.3.

Table A.1: Energy group boundaries for the 8-group structure.

Group	Lower Bound (eV)	Upper Bound (eV)
1	1.0×10^{-5}	5.0×10^{-2}
2	5.0×10^{-2}	1.5×10^{-1}
3	1.5×10^{-1}	2.75×10^{-1}
4	2.75×10^{-1}	6.25×10^{-1}
5	6.25×10^{-1}	3.0×10^{0}
6	3.0×10^{0}	1.7×10^{4}
7	1.7×10^{4}	8.2×10^{5}
8	8.2×10^{5}	2.0×10^{7}

For tallying purposes, each pin cell and axial burnup region was modeled individually for a total of over 5000 tally cells. The same group structure used in the OECD benchmark cases was also used for this case. The compositions of the burned fuel and its configuration within the assembly can be found in Reference [109].

A.3 C5G7 Core Model

The third benchmark model is the OECD/NEA C5G7 benchmark problem [110]. The primary purpose of this benchmark was to test various methods for calculating the sample variance for mesh tallies within Shift. This benchmark case represents a sixteen-assembly C5 MOX fuel assembly model, which utilizes a seven-group set of cross sections. The assemblies themselves represent a 17 × 17 lattice of square pin cells. Cross-sectional diagrams of the model are shown in Figure A.4 and the configuration of each pin cell for all four assemblies is shown in Figure A.5. Models for a two-dimensional and a three-dimensional version of this benchmark were developed and used in these analyses. The two-dimensional model was obtained by taking a 1-cm horizontal slice of the model's axial mid-plane and placing reflected boundary conditions above and below. The three-dimensional model accurately represents the benchmark specification given in Reference [110].

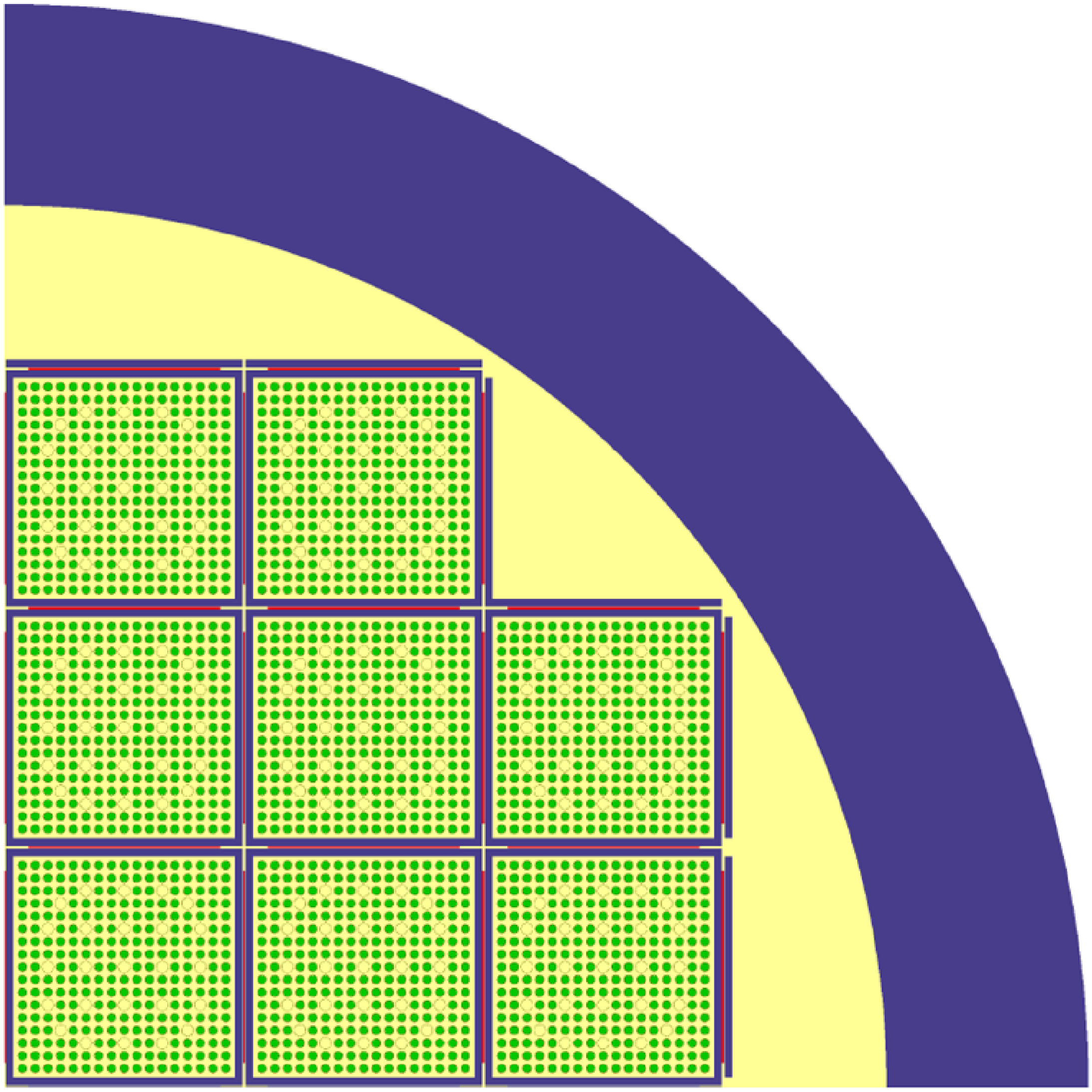


Figure A.2: Radial cross section for one quarter of the GBC-32 cask model.

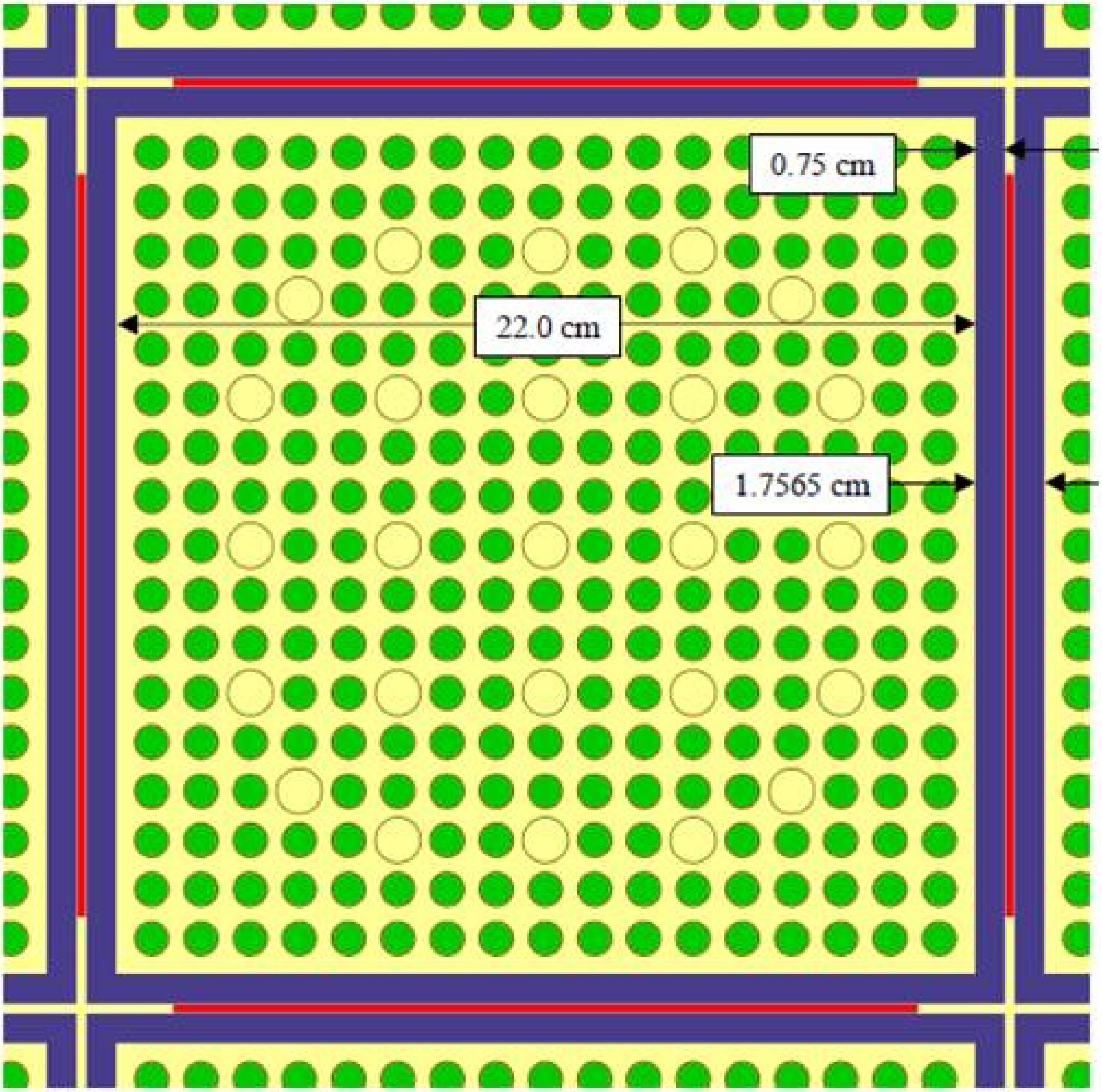
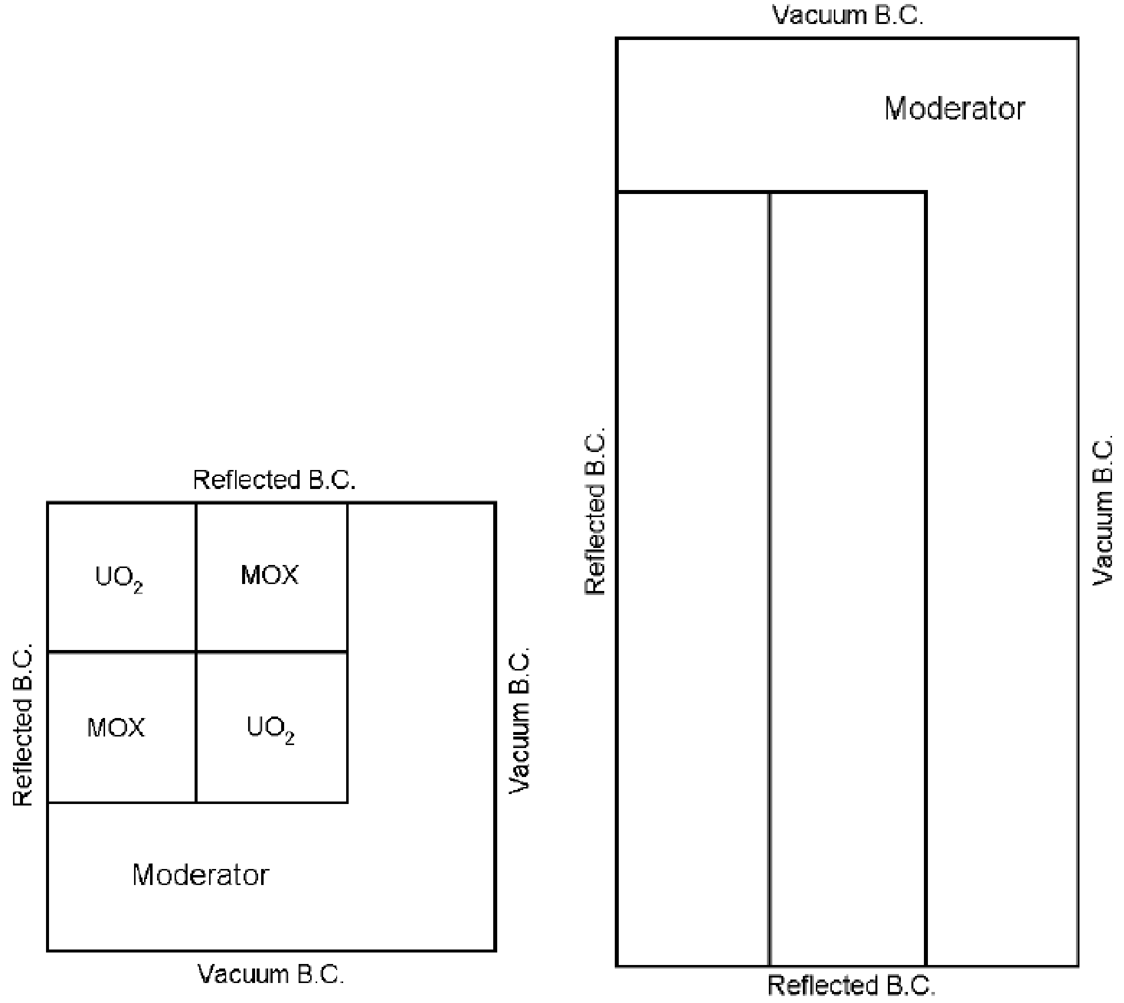


Figure A.3: Cross-sectional view of an assembly storage cell in the GBC-32 cask model.



(a) horizontal

(b) vertical

Figure A.4: Vertical and horizontal cross-sections for the C5G7 benchmark model.

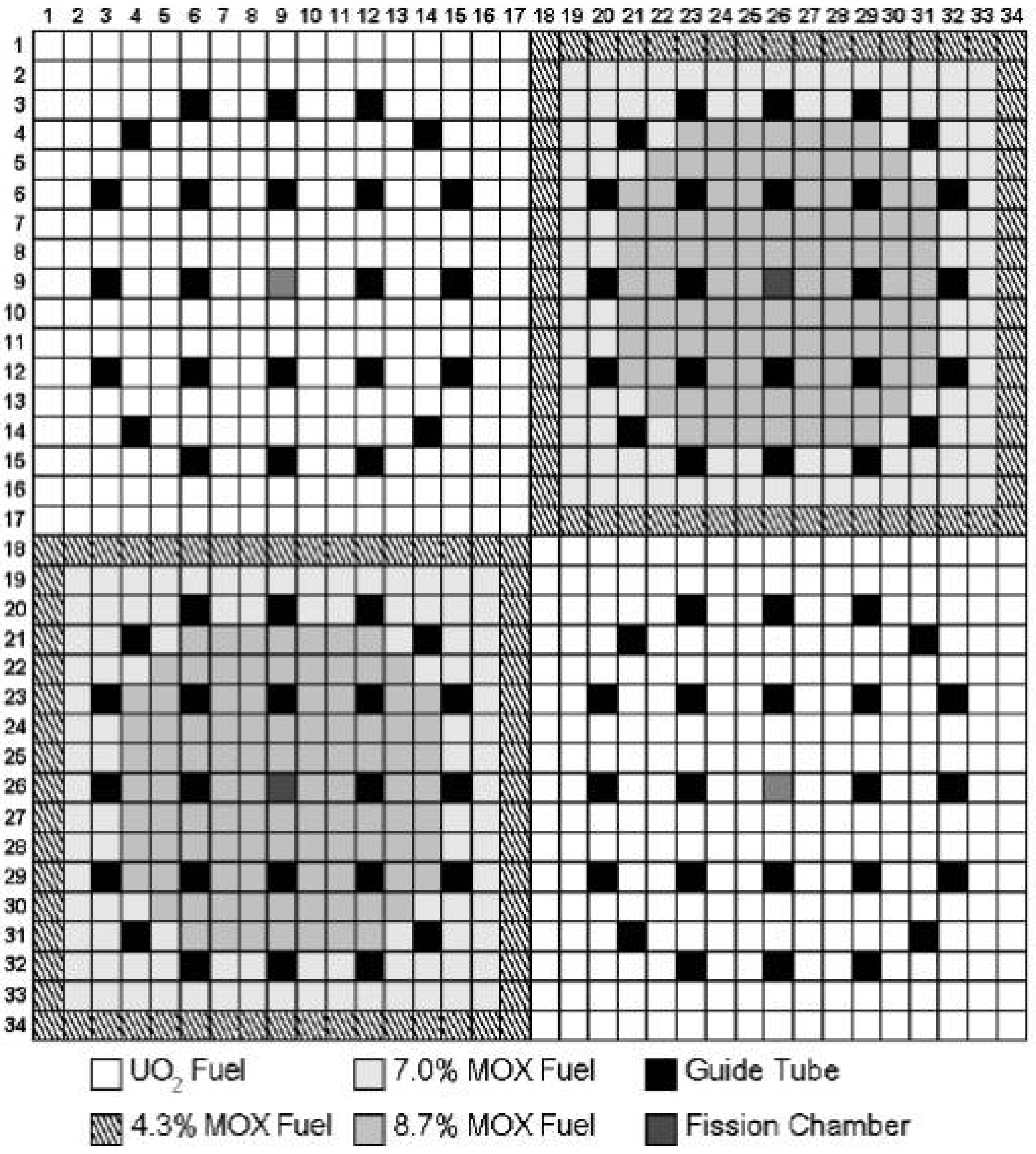


Figure A.5: Pin cell configuration for the C5G7 model.

B Verification Models

B.1 Pin Cell Model

Table B.1: Material specification for the pin cell model.

Region	ZAID	Concentration (atom/b-cm)
Fuel	^{16}O	4.55653×10^{-2}
	^{234}U	7.17988×10^{-6}
	^{235}U	4.27556×10^{-4}
	^{236}U	5.28177×10^{-6}
	^{238}U	2.23426×10^{-2}
	^{242m}Am	1.00000×10^{-9}
Clad	^{50}Cr	3.29746×10^{-6}
	^{52}Cr	6.35883×10^{-5}
	^{53}Cr	7.21041×10^{-6}
	^{54}Cr	1.79482×10^{-6}
	^{54}Fe	8.67281×10^{-6}
	^{56}Fe	1.36145×10^{-4}
	^{57}Fe	3.14417×10^{-6}
	^{58}Fe	4.18432×10^{-7}
	^{90}Zr	2.21142×10^{-2}
	^{91}Zr	4.82258×10^{-3}
	^{92}Zr	7.37141×10^{-3}
	^{94}Zr	7.47027×10^{-3}
	^{96}Zr	1.20350×10^{-3}
Moderator	^{1}H	6.66580×10^{-2}
	^{16}O	3.33290×10^{-2}
	^{166m}Ho	1.00000×10^{-9}

Table B.2: Collapsed 7-group structure used to test the cross section collapse and populations of the ORIGEN library.

Group	Upper Bound (eV)	Lower Bound (eV)
1	2.0×10^{7}	1.01×10^{6}
2	1.01×10^{6}	4.9952×10^{5}
3	4.9952×10^{5}	3.0×10^{0}
4	3.0×10^{0}	6.25×10^{-1}
5	6.25×10^{-1}	1.0×10^{-1}
6	1.0×10^{-1}	$.53 \times 10^{-2}$
7	2.53×10^{-2}	1.0×10^{-5}

Table B.3: Material specification the pin cell benchmark case with simplified isotopics.

Region	ZAID	Concentration (atom/b-cm)
Fuel	^{235}U	4.27556×10^{-4}
	^{238}U	2.23426×10^{-2}
Clad	^{90}Zr	2.21142×10^{-2}
Moderator	^{16}O	3.33290×10^{-2}

B.2 2×2 Array Model

Table B.4: Fuel material specification for the 2×2 array model.

Material	ZAID	Concentration (atom/b-cm)
fuel23	^{16}O	4.5491×10^{-2}
	^{234}U	4.7008×10^{-6}
	^{235}U	5.2968×10^{-4}
	^{236}U	3.4083×10^{-6}
	^{238}U	2.2208×10^{-2}
	^{242m}Am	1.0000×10^{-3}
fuel30	^{16}O	4.5494×10^{-2}
	^{59}Ni	3.0000×10^{-7}
	^{107}Pd	2.0000×10^{-7}
	^{234}U	6.2468×10^{-6}
	^{235}U	6.9087×10^{-4}
	^{236}U	4.3570×10^{-6}
	^{238}U	2.2046×10^{-2}
fuel36	^{16}O	4.5497×10^{-2}
	^{234}U	7.5720×10^{-6}
	^{235}U	8.2904×10^{-4}
	^{236}U	5.1701×10^{-6}
	^{238}U	2.1907×10^{-2}

Table B.5: Geometric parameters for the four pin cells in the 2 × 2 array model.

Material Label			Radius (cm)		Temperature (K)		
Fuel	Clad	Moderator	r_{fuel}	$r_{clad_{outer}}$	T_{fuel}	T_{clad}	T_{mod}
fuel23	clad1	mod1	0.41	0.51	900.0	539.0	549.0
fuel30	clad2	mod2	0.44	0.54	900.0	549.0	549.0
fuel36	clad3	mod3	0.47	0.57	900.0	559.0	549.0
mod5	clad4	mod4	0.50	0.60	449.0	569.0	549.0

Table B.6: Moderator material specification for the 2 × 2 array model.

Material	ZAID	Concentration (atom/b-cm)
mod1	^{1}H ^{16}O	2.92237×10^{-2} 1.46118×10^{-2}
mod2	^{1}H ^{16}O	2.98924×10^{-2} 1.49462×10^{-2}
mod3	^{1}H ^{16}O	3.05612×10^{-2} 1.52806×10^{-2}
mod4	^{1}H ^{16}O	3.12299×10^{-2} 1.56149×10^{-2}
mod5	^{1}H ^{16}O	3.65798×10^{-2} 1.82899×10^{-2}

Table B.7: Cladding material specification for the 2 × 2 array model.

Material	ZAID	Concentration (atom/b-cm)
all cladding materials	^{50}Cr	3.23579×10^{-6}
	^{52}Cr	6.23990×10^{-5}
	^{53}Cr	7.07555×10^{-6}
	^{54}Cr	1.76125×10^{-6}
	^{54}Fe	8.51099×10^{-6}
	^{56}Fe	1.33604×10^{-4}
	^{57}Fe	3.08551×10^{-6}
	^{58}Fe	4.10625×10^{-7}
	^{90}Zr	2.14528×10^{-2}
	^{91}Zr	4.67834×10^{-3}
	^{92}Zr	7.15093×10^{-3}
	^{94}Zr	7.24684×10^{-3}
	^{96}Zr	1.16750×10^{-3}
	^{112}Sn	4.58790×10^{-6}
	^{114}Sn	3.12166×10^{-6}
	^{115}Sn	1.60813×10^{-6}
	^{116}Sn	6.87712×10^{-5}
	^{117}Sn	3.63248×10^{-5}
	^{118}Sn	1.14556×10^{-4}
	^{119}Sn	4.06289×10^{-5}
	^{120}Sn	1.54097×10^{-4}
	^{122}Sn	2.18989×10^{-5}
	^{124}Sn	2.73855×10^{-5}
	^{174}Hf	3.47120×10^{-9}
	^{176}Hf	1.14116×10^{-7}
	^{177}Hf	4.03528×10^{-7}
	^{178}Hf	5.91840×10^{-7}
	^{179}Hf	2.95486×10^{-7}
	^{180}Hf	7.61062×10^{-7}
(added to *clad4*)	^{235}U	4.00000×10^{-6}

C Additional Depletion Results

C.1 Continuous-Energy Depletion

Table C.1: Comparison of number densities when using a custom-generated ORIGEN library.

Nuclide	Exnihilo	TRITON	Rel. Diff.	Nuclide	Exnihilo	TRITON	Rel. Diff.
h-1	2.755×10^{-11}	2.588×10^{-09}	0.9894	ce-144	6.873×10^{-06}	6.883×10^{-06}	0.0014
b-10	3.737×10^{-19}	2.070×10^{-19}	0.8052	pr-141	1.250×10^{-05}	1.253×10^{-05}	0.0022
b-11	1.047×10^{-15}	4.660×10^{-16}	1.2466	pr-143	2.214×10^{-08}	2.217×10^{-08}	0.0015
n-14	1.010×10^{-15}	4.168×10^{-16}	1.4241	nd-143	1.109×10^{-05}	1.112×10^{-05}	0.0028
o-16	4.556×10^{-02}	4.557×10^{-02}	0.0001	nd-144	6.505×10^{-06}	6.507×10^{-06}	0.0004
kr-83	9.998×10^{-07}	1.000×10^{-06}	0.0002	nd-145	8.231×10^{-06}	8.245×10^{-06}	0.0017
sr-90	1.106×10^{-05}	1.106×10^{-05}	0.0001	nd-146	7.050×10^{-06}	7.062×10^{-06}	0.0017
zr-91	1.030×10^{-05}	1.030×10^{-05}	0.0002	nd-147	2.838×10^{-09}	2.843×10^{-09}	0.0018
zr-93	1.318×10^{-05}	1.319×10^{-05}	0.0011	nd-148	3.994×10^{-06}	4.005×10^{-06}	0.0028
zr-94	1.363×10^{-05}	1.365×10^{-05}	0.0018	pm-147	3.577×10^{-06}	3.560×10^{-06}	0.0047
zr-95	1.824×10^{-06}	1.827×10^{-06}	0.0016	pm-148	5.327×10^{-11}	5.544×10^{-11}	0.0391
zr-96	1.384×10^{-05}	1.387×10^{-05}	0.0021	pm-149	7.520×10^{-18}	7.579×10^{-18}	0.0078
nb-93	3.428×10^{-13}	3.462×10^{-13}	0.0099	sm-147	5.750×10^{-07}	5.729×10^{-07}	0.0038
nb-95	1.548×10^{-06}	1.550×10^{-06}	0.0015	sm-149	7.986×10^{-08}	8.086×10^{-08}	0.0124
mo-95	1.047×10^{-05}	1.049×10^{-05}	0.0018	sm-150	2.775×10^{-06}	2.791×10^{-06}	0.0056
mo-97	1.344×10^{-05}	1.347×10^{-05}	0.0023	sm-151	1.367×10^{-07}	1.383×10^{-07}	0.0115
mo-98	1.332×10^{-05}	1.335×10^{-05}	0.0024	sm-152	1.494×10^{-06}	1.503×10^{-06}	0.0057
mo-99	3.013×10^{-15}	3.022×10^{-15}	0.0030	sm-153	1.360×10^{-19}	1.363×10^{-19}	0.0021
mo-100	1.469×10^{-05}	1.473×10^{-05}	0.0025	eu-151	2.973×10^{-10}	3.011×10^{-10}	0.0126
tc-99	1.367×10^{-05}	1.370×10^{-05}	0.0021	eu-153	8.208×10^{-07}	8.146×10^{-07}	0.0076
ru-101	1.230×10^{-05}	1.232×10^{-05}	0.0019	eu-154	9.512×10^{-08}	9.931×10^{-08}	0.0422
ru-102	1.111×10^{-05}	1.116×10^{-05}	0.0045	eu-155	4.765×10^{-08}	4.931×10^{-08}	0.0337
ru-103	5.342×10^{-07}	5.372×10^{-07}	0.0055	eu-156	1.224×10^{-09}	1.256×10^{-09}	0.0258
ru-104	6.713×10^{-06}	6.761×10^{-06}	0.0071	gd-152	1.013×10^{-10}	1.023×10^{-10}	0.0095
ru-106	2.151×10^{-06}	2.176×10^{-06}	0.0115	gd-154	3.588×10^{-09}	3.756×10^{-09}	0.0446
rh-103	7.816×10^{-06}	7.857×10^{-06}	0.0053	gd-155	1.514×10^{-09}	1.568×10^{-09}	0.0346
rh-105	1.461×10^{-22}	1.476×10^{-22}	0.0100	gd-156	2.555×10^{-07}	2.604×10^{-07}	0.0189
pd-105	4.284×10^{-06}	4.321×10^{-06}	0.0086	gd-157	8.011×10^{-10}	8.169×10^{-10}	0.0193
pd-107	2.003×10^{-06}	2.028×10^{-06}	0.0122	gd-158	8.712×10^{-08}	8.821×10^{-08}	0.0124
pd-108	1.207×10^{-06}	1.227×10^{-06}	0.0163	gd-160	5.894×10^{-09}	5.974×10^{-09}	0.0134
ag-109	7.434×10^{-07}	7.548×10^{-07}	0.0151	u-234	5.997×10^{-06}	5.993×10^{-06}	0.0007
cd-113	3.277×10^{-09}	3.328×10^{-09}	0.0153	u-235	2.253×10^{-04}	2.255×10^{-04}	0.0007
in-115	3.564×10^{-08}	4.044×10^{-08}	0.1187	u-236	3.760×10^{-05}	3.764×10^{-05}	0.0010
sn-126	2.099×10^{-07}	2.112×10^{-07}	0.0060	u-238	2.219×10^{-02}	2.219×10^{-02}	0.0001
i-127	4.762×10^{-07}	4.795×10^{-07}	0.0069	np-237	1.661×10^{-06}	1.675×10^{-06}	0.0081
i-129	1.674×10^{-06}	1.682×10^{-06}	0.0050	pu-238	1.754×10^{-07}	1.791×10^{-07}	0.0209
xe-131	6.434×10^{-06}	6.459×10^{-06}	0.0039	pu-239	7.520×10^{-05}	7.690×10^{-05}	0.0221
xe-133	3.797×10^{-11}	3.808×10^{-11}	0.0029	pu-240	1.344×10^{-05}	1.369×10^{-05}	0.0185
cs-133	1.488×10^{-05}	1.492×10^{-05}	0.0024	pu-241	5.416×10^{-06}	5.507×10^{-06}	0.0165
cs-134	5.204×10^{-07}	5.215×10^{-07}	0.0022	pu-242	6.538×10^{-07}	6.642×10^{-07}	0.0156
cs-135	2.671×10^{-06}	2.690×10^{-06}	0.0071	am-241	1.057×10^{-07}	1.076×10^{-07}	0.0172
cs-137	1.408×10^{-05}	1.412×10^{-05}	0.0028	am-242	5.857×10^{-15}	6.001×10^{-15}	0.0240
ba-140	1.693×10^{-08}	1.696×10^{-08}	0.0020	am-242m	4.539×10^{-10}	4.651×10^{-10}	0.0240
la-139	1.421×10^{-05}	1.424×10^{-05}	0.0020	am-243	3.313×10^{-08}	3.372×10^{-08}	0.0174
ce-141	4.286×10^{-07}	4.294×10^{-07}	0.0019	cm-242	7.101×10^{-09}	7.218×10^{-09}	0.0163
ce-142	1.291×10^{-05}	1.294×10^{-05}	0.0025	cm-243	1.211×10^{-10}	1.296×10^{-10}	0.0655
ce-143	2.323×10^{-23}	2.327×10^{-23}	0.0018	cm-244	2.263×10^{-09}	2.344×10^{-09}	0.0346

C.2 VERA Problem 2a

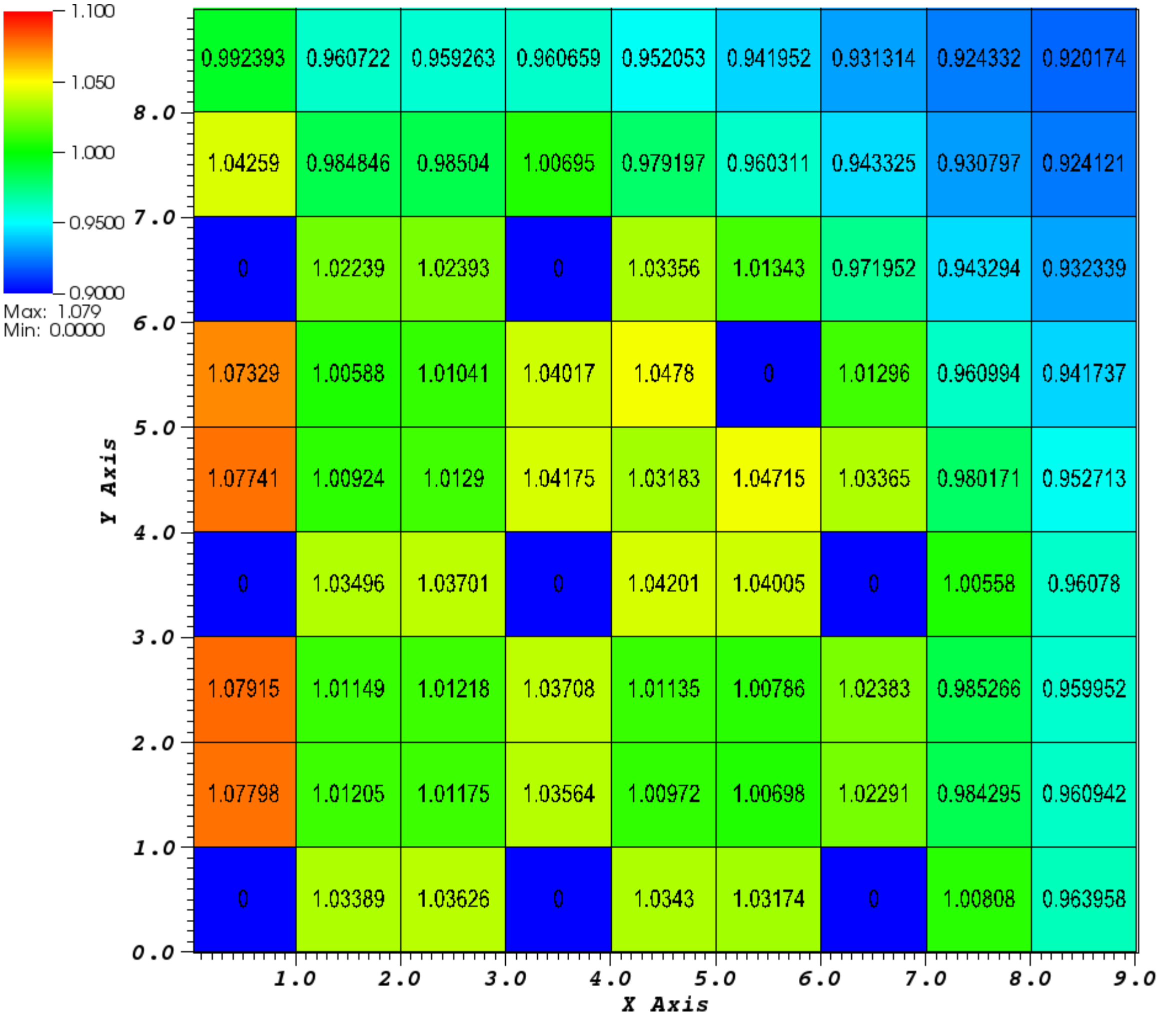
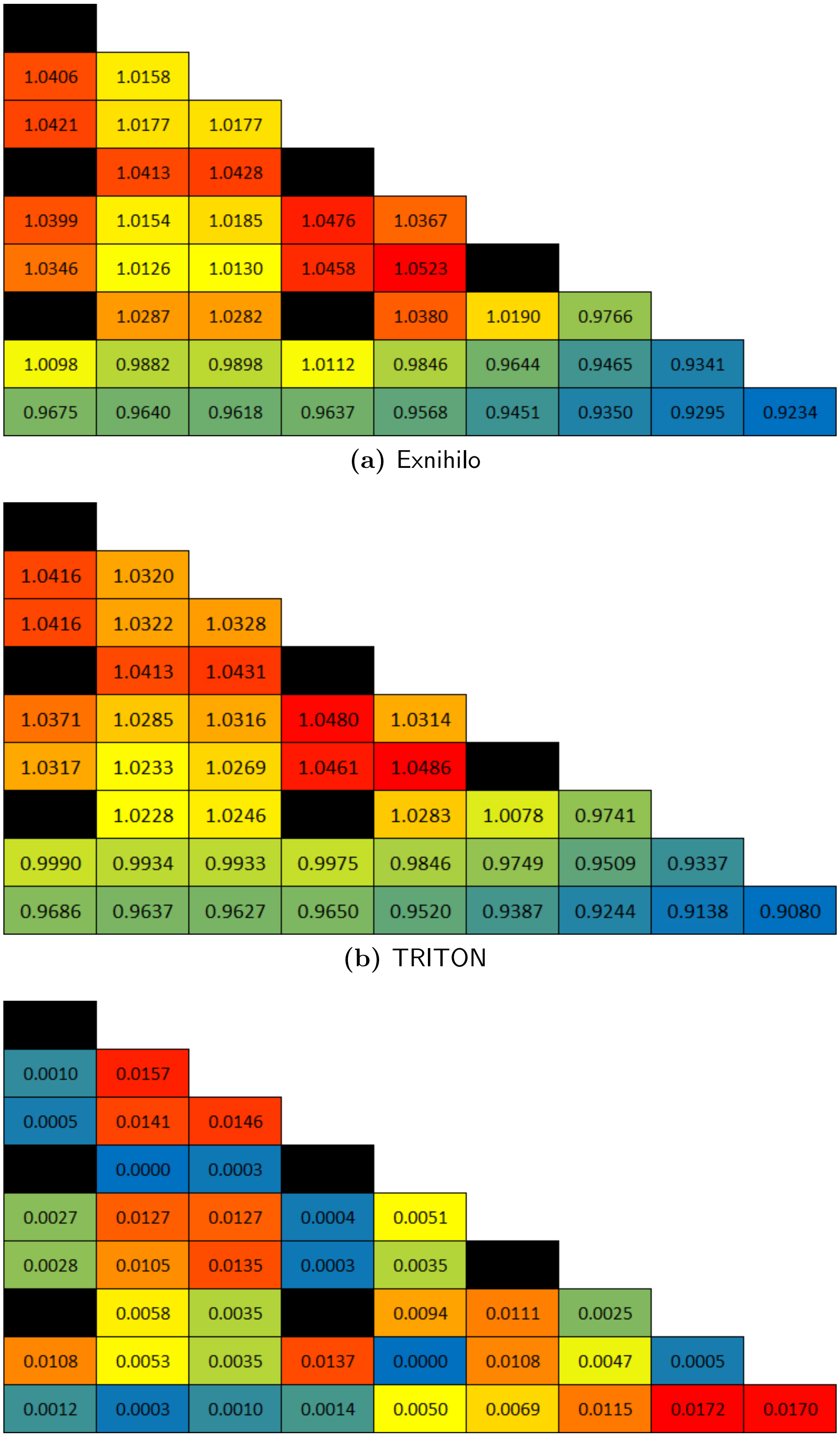


Figure C.1: Pin power peaking factors calculated by Exnihilo for the VERA 2a quarter assembly model at 15.73 GWd/MTIHM.

C.3 OECD/NEA Burnup Credit Calculation

Reference [111] presents a pin cell model which includes experimental (measured) data for select isotopes at the end of a four-cycle operating history. The parameters for each



1.0406	1.0158							
1.0421	1.0177	1.0177						
	1.0413	1.0428						
1.0399	1.0154	1.0185	1.0476	1.0367				
1.0346	1.0126	1.0130	1.0458	1.0523				
	1.0287	1.0282		1.0380	1.0190	0.9766		
1.0098	0.9882	0.9898	1.0112	0.9846	0.9644	0.9465	0.9341	
0.9675	0.9640	0.9618	0.9637	0.9568	0.9451	0.9350	0.9295	0.9234

(a) Exnihilo

1.0416	1.0320							
1.0416	1.0322	1.0328						
	1.0413	1.0431						
1.0371	1.0285	1.0316	1.0480	1.0314				
1.0317	1.0233	1.0269	1.0461	1.0486				
	1.0228	1.0246		1.0283	1.0078	0.9741		
0.9990	0.9934	0.9933	0.9975	0.9846	0.9749	0.9509	0.9337	
0.9686	0.9637	0.9627	0.9650	0.9520	0.9387	0.9244	0.9138	0.9080

(b) TRITON

0.0010	0.0157							
0.0005	0.0141	0.0146						
	0.0000	0.0003						
0.0027	0.0127	0.0127	0.0004	0.0051				
0.0028	0.0105	0.0135	0.0003	0.0035				
	0.0058	0.0035		0.0094	0.0111	0.0025		
0.0108	0.0053	0.0035	0.0137	0.0000	0.0108	0.0047	0.0005	
0.0012	0.0003	0.0010	0.0014	0.0050	0.0069	0.0115	0.0172	0.0170

(c) abs(C/E) - 1

Figure C.2: Comparison of the power distribution for the VERA 2a quarter assembly model at 15.73 GWd/MTIHM.

Table C.2: Operating history for the benchmark pin cell problem.

Cycle	Burntime (days)	Downtime (days)	Boron Concentration (ppm)
1	306.0	71.0	331.0
2	381.7	83.1	469.7
3	466.0	85.0	504.1
4	461.1	1870.0	492.5

of the four cycles is presented in Table C.2. Because the capability to modify the boron concentration over time is not available at present within Exnihilo, four individual calculation were performed where the nuclide number densities were extracted from the output of the previous calculation and used as the initial concentrations for the subsequent calculation. Also note that each cycle divided the burntime into three equal-length subcycles to enhance the fidelity of the calculation. A comparison for all nuclides in which experimental data is available is presented in Figure C.5.

The results presented in Figure C.5 show an improvement in the ^{133}Cs, ^{239}Pu, and ^{240}Pu estimates* versus the comparison for the VERA 1c pin cell model presented in Section 8.7. This is especially encouraging given that this pin cell benchmark problem was more complex and burned for a much longer duration than the the VERA 1c pin cell model. However, the estimates for ^{149}Sm, ^{238}Pu, and ^{234}U are notably high. Furthermore, the results for the remaining samarium and plutonium isotopes, as well as for ^{235}U, as also higher than expected.

The poor predictions of several isotopes in this comparison as well as the comparison presented in Section 8.7 led to additional analysis of the ultrafine group structure used to perform the flux tally and cross section collapse. Although a constant-lethargy approach provides the optimal binning methodology given the physics in a neutron transport environment, it is reasonable to assume that some regions of the energy spectrum are more important than others. Take the ^{238}U (n, γ)

*Experimental data was not available for ^{131}Xe.

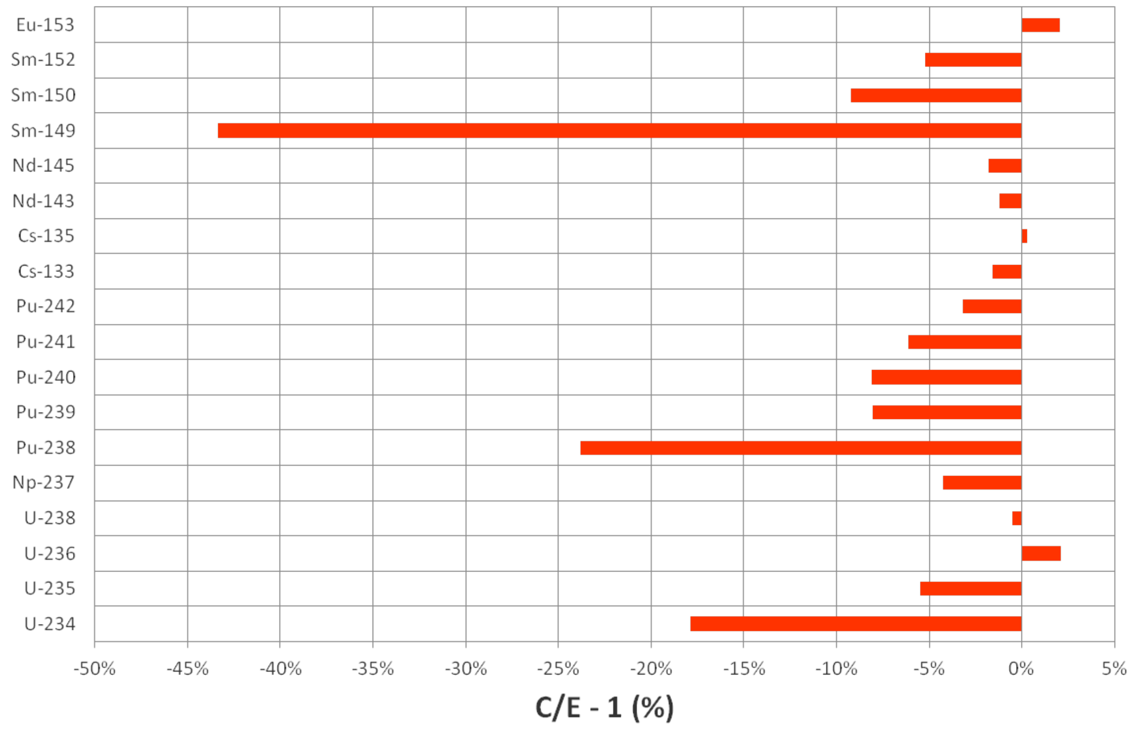


Figure C.3: Comparison of Exnihilo versus experimental data.

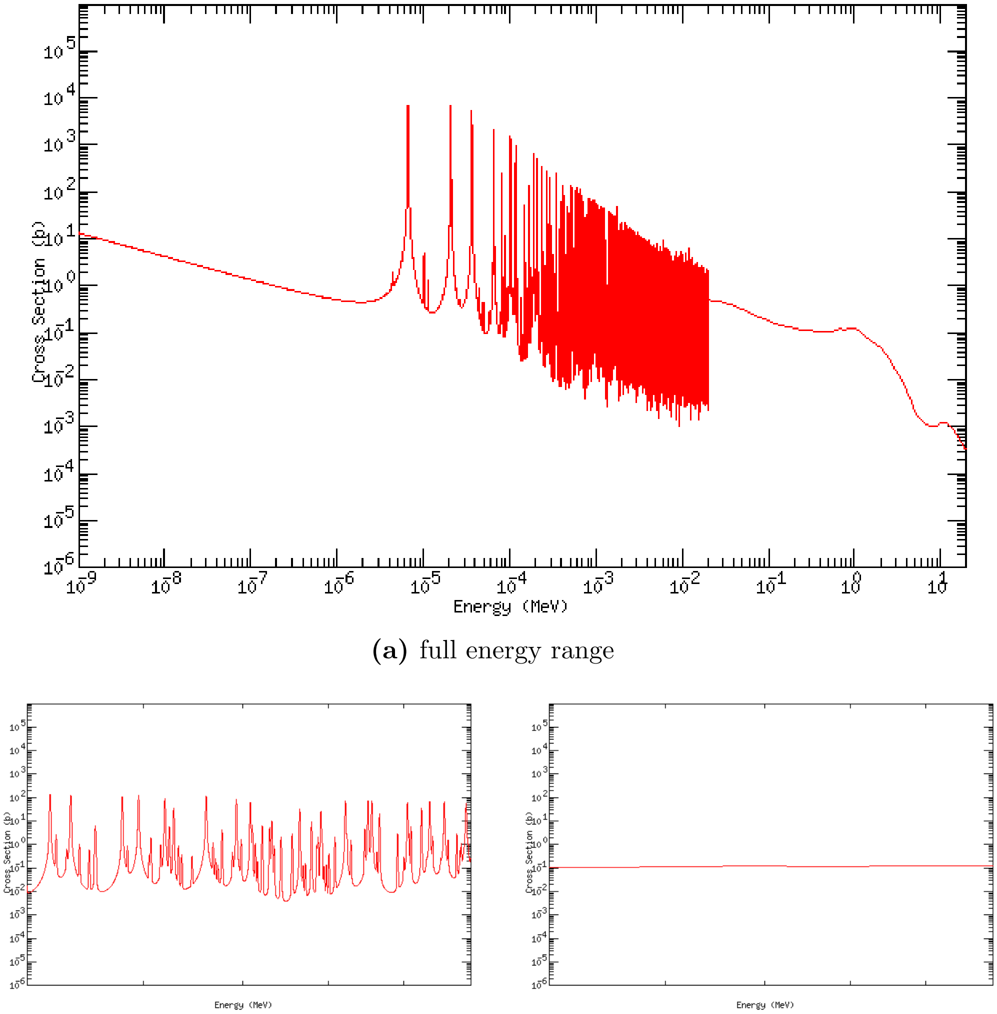
cross section for example, which is provided in Figure C.4. Using a group structure of 43 000 equilethargy groups, approximately 1000 energy groups would be used to represent the energy ranges from 0.5 keV–1.0 keV as well as from 0.5 MeV–1.0MeV. From Figure C.4, one observes that 1000 energy groups is unnecessary in the 0.5 MeV–1.0MeV energy range (Figure C.4c) while more than 1000 energy groups would likely increase the accuracy of the flux tally in the 0.5 keV–1.0 keV energy range (Figure C.4b). Although this same behavior is not guaranteed to be true for every relevant cross section and every nuclide, it is typical for cross sections to change much more dramatically over much smaller energy ranges in the resonance region of the energy spectrum compared to the low and high energy ranges. Using this analysis an optimized energy group structure was taken from the VESTA Monte Carlo depletion code [112]. This energy group structure is provided in Figure C.3.

Table C.3: Ultrafine-group structure utilized by the VESTA Monte Carlo depletion code.

Upper Energy Bound (eV)	# of Equilethargy Groups (× 1000)
10^{-4}	1
10^{-3}	1
10^{-2}	1
10^{-1}	1
10^{0}	1
10^{1}	4
10^{2}	4
10^{3}	10
10^{4}	10
10^{5}	4
10^{6}	4
10^{7}	1
2×10^{7}	1

Following the implementation of an alternate group structure in Exnihilo, the new optimized group structure was used to generated new results for the pin cell benchmark model from Reference [111]. The results are presented alongside the results which used a constant lethargy over the entire energy range in Figure C.5. Significant improvements are observed in the plutonium concentrations with the exception of ^{238}Pu as well as the ^{235}U concentration. Noting that Reference [111] also contains results for each nuclide from several different organizations which each used a different computation code, the differences observed by the other codes is observed alongside the differences observed by Exnihilo to determine whether or not Exnihilo is producing reasonable estimates. Figure C.6 presents the results from the different organizations for ^{149}Sm, ^{234}U, ^{235}U, ^{238}Pu, and ^{239}Pu plotted against. The comparison of results from different codes provides some additional explanation for some of the more poorly estimated nuclides as well as some additional validation for the accuracy of the estimation in general. In the case of ^{149}Sm and ^{238}Pu, the relative* standard deviation of the results from the different codes is approximately 15%. Furthermore, in the case of ^{149}Sm, the number density for ^{149}Sm is on the order of 7×10^{-8}, which means that ORIGEN will not be able to predict its concentration as well as it does for nuclides with number densities above 10^{-5}. In the case of ^{238}Pu, it is important to remember that a multitude of transition to and from ^{238}Pu exists in the transition matrix, and therefore this nuclide is relatively difficult to predict under any circumstances. The analysis of ^{235}U and ^{239}Pu, arguably two of the most important nuclides in depletion analysis, show that Exnihilo is performing very well in comparison to the other codes. Lastly, the comparison of results for ^{234}U show that Exnihilo is producing a result that is much worse than the majority of the results presented for this nuclide. Further analysis of ^{234}U reveals a possible cause for the observed discrepancy. Figure C.7 plots the difference in the fission yield from 0.05 to 1.25 GWd/MTIHM for all fission products. Considering that the difference in the fission yield for many of the other fissile nuclides is less that 5%, a difference of 15 000 000% is obviously a cause for concern a points to a bug in either the ORIGEN code or the fission yield data.

*Relative to the average result.



(a) full energy range

(b) 0.5 keV–1.0keV

(c) 0.5 MeV–1.0MeV

Figure C.4: ^{238}U (n, γ) cross section.

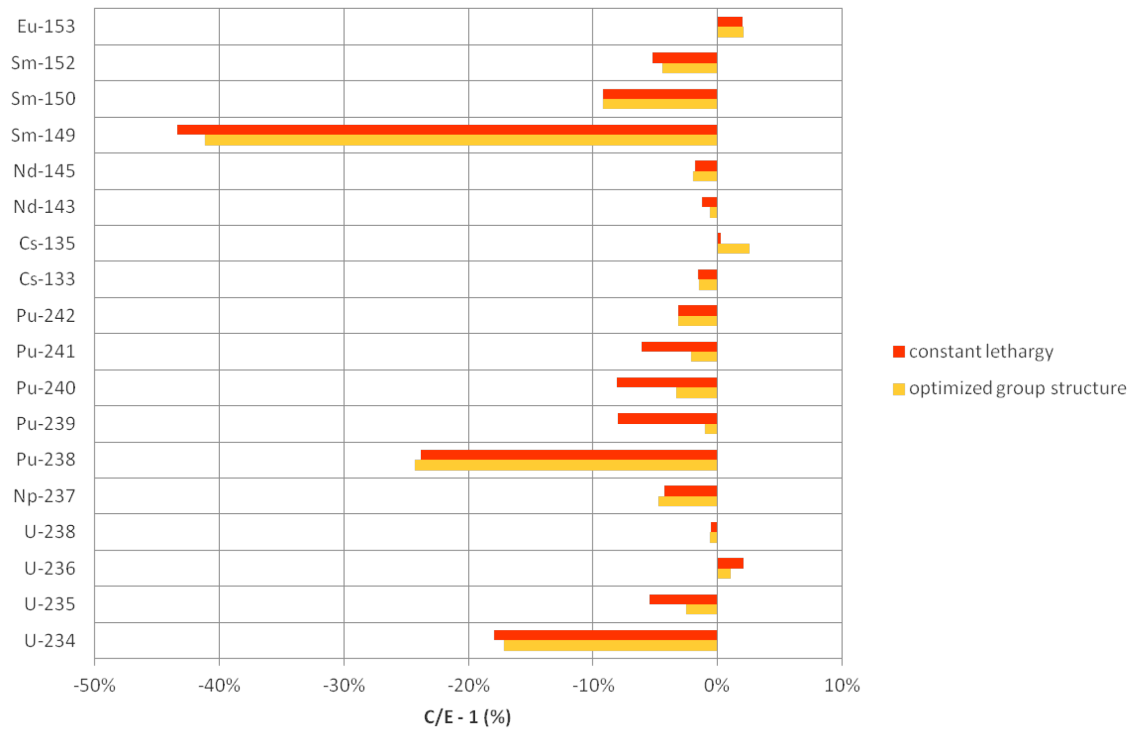
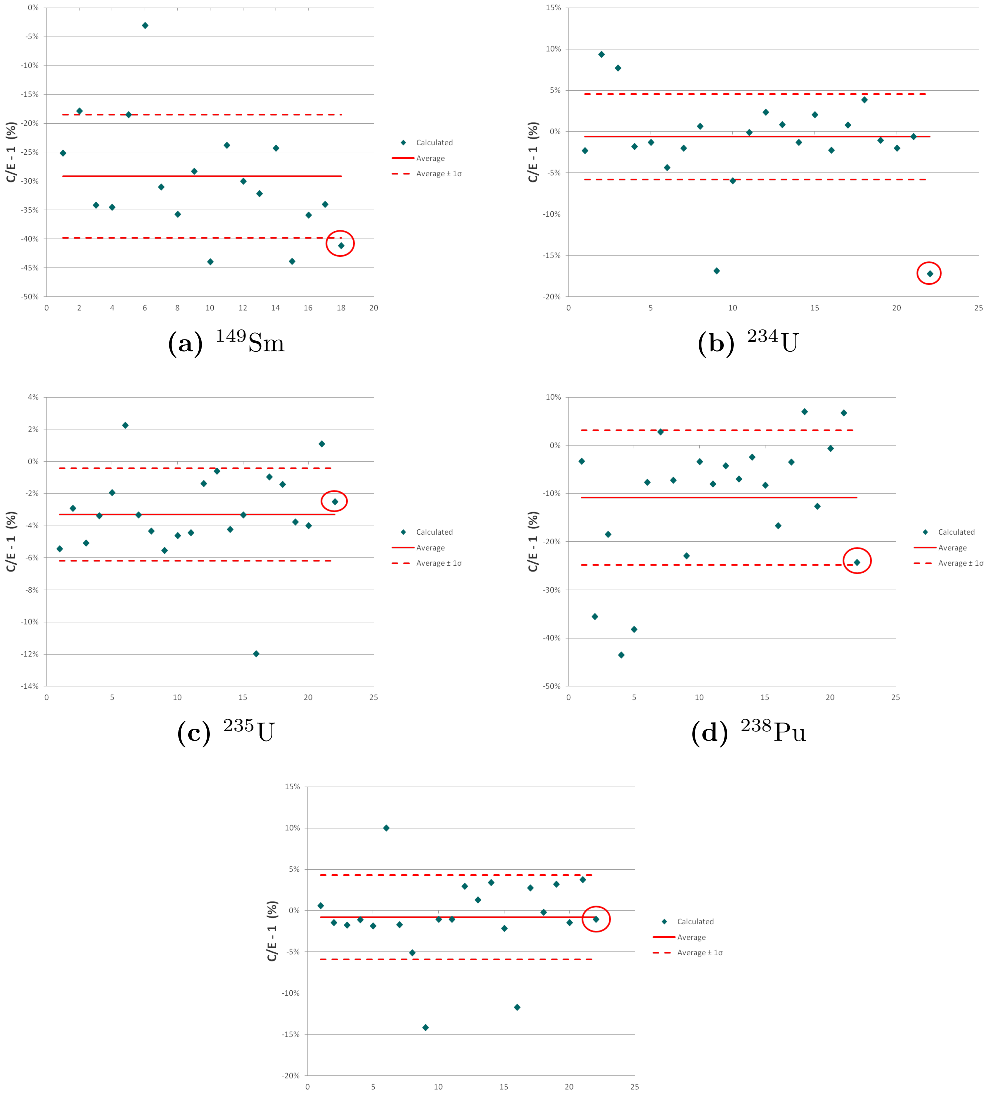


Figure C.5: Comparison of Exnihilo versus experimental data for two different group structures.



(a) ^{149}Sm

(b) ^{234}U

(c) ^{235}U

(d) ^{238}Pu

(e) ^{239}Pu

Figure C.6: Plots of the relative errors from several different codes for select nuclides.

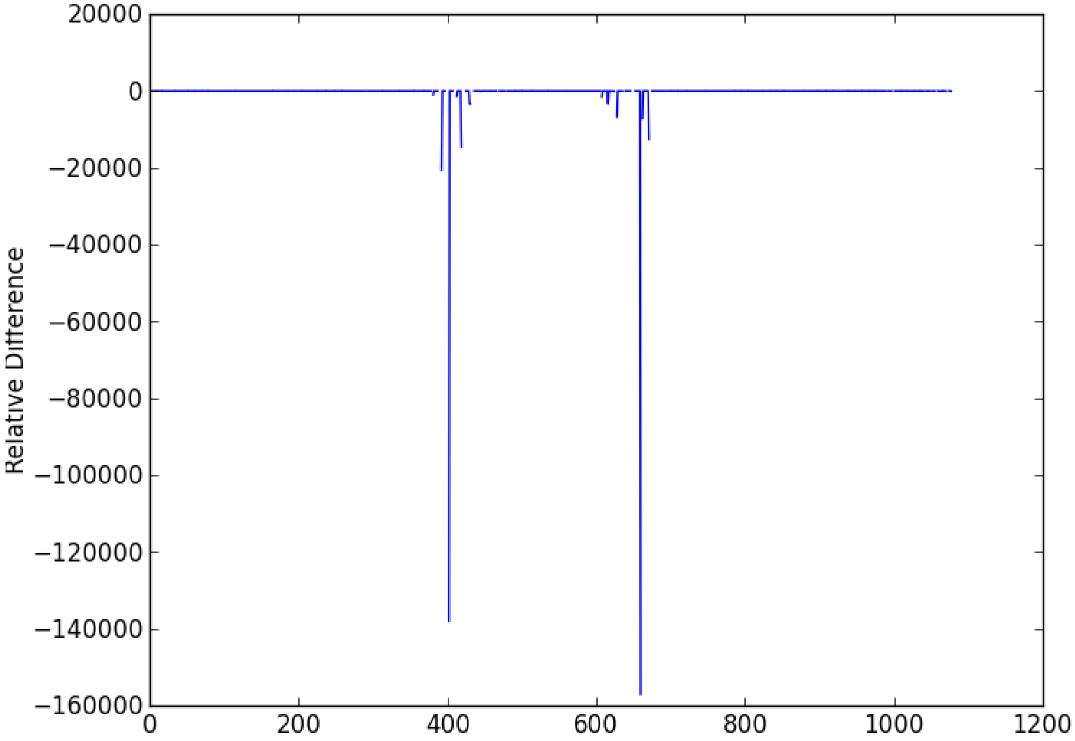


Figure C.7: Difference in the ^{234}U fission yield from 0.05 to 1.25 GWd/MTIHM.

D Supplemental Data

Table D.3: Memory required to load specific nuclides [107].

Nuclide	Required Memory	Number of Energy Points
h1	175.61	580
b10	8.33	1386
b11	71.70	3550
n14	42.64	2313
o16	205.14	4271
kr83	0.00	1599
zr91	29.88	26963
zr93	14.15	19978
zr94	35.36	25502
zr95	8.05	3903
zr96	25.09	14215
nb93	51.45	42510
nb95	8.17	4985
mo95	71.14	16164
mo97	26.82	20926
mo98	49.40	42511
mo99	7.84	4272
mo100	26.74	39690
tc99	175.42	89195
ru101	70.00	12807
ru102	0.00	39932
ru103	16.95	3270

Continued on next page

Table D.3 Continued from previous page

Nuclide	Required Memory	Number of Energy Points
ru104	21.15	33188
ru106	4.02	3839
rh103	148.34	51695
rh105	0.00	991
pd105	53.71	40095
pd107	19.37	19751
pd108	43.75	24603
ag109	96.16	64124
cd113	73.25	89094
in115	30.82	48798
sn126	4.31	3727
i127	68.46	65910
i129	28.72	37404
i135	2.98	785
xe131	75.91	16352
xe133	0.00	4646
xe135	0.00	4843
cs133	59.52	44718
cs134	13.31	7633
cs135	6.47	6936
cs137	5.76	4961
ba140	2.73	5460
la139	32.09	41492
ce141	12.21	3032
ce142	11.99	15324

Continued on next page

Table D.3 Continued from previous page

Nuclide	Required Memory	Number of Energy Points
ce143	56.12	4920
ce144	0.00	4785
pr141	53.03	44185
pr143	7.05	6222
nd143	82.72	34763
nd144	70.24	15920
nd145	92.19	44913
nd146	51.53	16626
nd147	56.37	4171
nd148	61.78	31465
pm147	0.00	10314
pm148	0.00	5111
pm149	7.96	4700
sm147	111.18	46961
sm149	67.53	30352
sm150	61.91	10680
sm151	82.49	20699
sm152	62.93	33035
sm153	50.34	3258
eu151	0.00	12773
eu153	61.02	11231
eu154	0.00	3345
eu155	3.34	2156
eu156	5.43	5057
gd152	96.63	40444

Continued on next page

Table D.3 Continued from previous page

Nuclide	Required Memory	Number of Energy Points
gd154	101.20	46745
gd155	68.84	14805
gd156	87.03	25806
gd157	63.29	13689
gd158	106.46	34712
gd160	70.01	22141
u234	78.03	38749
u235	117.48	118487
u236	81.48	36328
u238	281.48	273303
np237	61.68	58680
pu238	5.05	6417
pu239	127.32	115278
pu240	54.68	80353
pu241	10.64	22385
pu242	16.55	22936
am241	46.93	24170
am242	68.78	7104
am243	24.99	31486
cm242	1.32	6094
cm243	8.21	7193
cm244	14.71	24314

Concluded

Table D.1: Recoverable Energy Release (MeV) [113]

Nuclide	$\kappa_{fission}$	$\kappa_{capture}$	Nuclide	$\kappa_{fission}$	$\kappa_{capture}$	Nuclide	$\kappa_{fission}$	$\kappa_{capture}$
^{1}H	——	2.2246	^{147}Pm	——	5.9	^{237}Np	195.1	5.49
^{10}B	——	2.79	^{148}Pm	——	7.266	^{239}Np	200	4.97
^{16}O	——	4.143	^{147}Sm	——	8.1402	^{238}Pu	197.8	5.55
^{56}Fe	——	7.6	^{149}Sm	——	7.9824	^{239}Pu	200.05	6.533
^{28}Ni	——	9.02	^{150}Sm	——	5.596	^{240}Pu	199.79	5.241
^{90}Zr	——	7.2026	^{151}Sm	——	8.258	^{241}Pu	202.22	6.301
^{91}Zr	——	8.6351	^{152}Sm	——	5.867	^{242}Pu	200.62	5.071
^{92}Zr	——	6.758	^{153}Eu	——	6.444	^{243}Pu	200	6.02
^{96}Zr	——	5.571	^{154}Eu	——	8.167	^{241}Am	202.3	5.529
^{95}Mo	——	9.1542	^{155}Eu	——	6.49	^{242m}Am	202.29	6.426
^{95}Tc	——	7.71	^{230}Th	190	5.01	^{243}Am	202.1	5.363
^{101}Ru	——	9.2161	^{232}Th	189.21	4.786	^{244}Cm	200	6.451
^{103}Rh	——	6.9993	^{233}Th	190	6.08	^{245}Cm	200	6.11
^{105}Rh	——	7.0941	^{231}Pa	190	5.66			
^{109}Ag	——	6.825	^{233}Pa	189.1	5.197			
^{131}Xe	——	8.9363	^{232}U	200	5.93			
^{135}Xe	——	7.88	^{233}U	191.29	6.841			
^{133}Cs	——	6.7044	^{234}U	190.3	5.297			
^{134}Cs	——	6.55	^{235}U	194.02	6.5451			
^{143}Nd	——	7.8174	^{236}U	192.8	5.124			
^{145}Nd	——	7.5654	^{238}U	198.12	4.804			

Table D.2: Default nuclides added to depletable materials.

^{1}H	^{10}B	^{11}B	^{14}N	^{16}O
^{83}Kr	^{91}Zr	^{93}Zr	^{94}Zr	^{95}Zr
^{96}Zr	^{93}Nb	^{95}Nb	^{95}Mo	^{97}Mo
^{98}Mo	^{99}Mo	^{100}Mo	^{99}Tc	^{101}Ru
^{102}Ru	^{103}Ru	^{104}Ru	^{106}Ru	^{103}Rh
^{105}Rh	^{105}Pd	^{107}Pd	^{108}Pd	^{109}Ag
^{113}Cd	^{115}In	^{126}Sn	^{127}I	^{129}I
^{135}I	^{131}Xe	^{133}Xe	^{135}Xe	^{133}Cs
^{134}Cs	^{135}Cs	^{137}Cs	^{140}Ba	^{139}La
^{141}Ce	^{142}Ce	^{143}Ce	^{144}Ce	^{141}Pr
^{143}Pr	^{143}Nd	^{144}Nd	^{145}Nd	^{146}Nd
^{147}Nd	^{148}Nd	^{147}Pm	^{148}Pm	^{149}Pm
^{147}Sm	^{149}Sm	^{150}Sm	^{151}Sm	^{152}Sm
^{153}Sm	^{151}Eu	^{153}Eu	^{154}Eu	^{155}Eu
^{156}Eu	^{152}Gd	^{154}Gd	^{155}Gd	^{156}Gd
^{157}Gd	^{158}Gd	^{160}Gd	^{234}U	^{235}U
^{236}U	^{238}U	^{237}Np	^{238}Pu	^{239}Pu
^{240}Pu	^{241}Pu	^{242}Pu	^{241}Am	^{242}Am
^{243}Am	^{242}Cm	^{243}Cm	^{244}Cm	

Vita

Brenden Mervin was born in Perth Amboy, NJ, to parents Keith and Andrea Mervin. He is the oldest of three children; Brittany his sister and Matthew his brother. He attended elementary school in Sayreville, NJ before moving to Maryville, TN where he finished junior high school and attended William Blount High School. After graduation Brenden chose to attend the University of Tennessee, choosing to major in Nuclear Engineering. After obtaining a Bachelor of Science degree he was recruited by Dr. G. Ivan Maldonado to remain at the University of Tennessee to pursue his Masters of Science in nuclear engineering and became heavily involved in many research and development efforts at Oak Ridge National Laboratory. Due to this heavy involvement Brenden decided to remain at the University of Tennessee to pursue his Doctor of Philosophy degree in nuclear engineering with research focusing on coupling neutron transport to depletion for massively parallel architectures.