DETAILED ANALYSIS OF PHASE SPACE EFFECTS IN FUEL BURNUP/DEPLETION FOR PWR ASSEMBLY & FULL CORE MODELS USING LARGE-SCALE PARALLEL COMPUTATION

A Thesis
Presented to
The Academic Faculty

by

Kevin L. Manalo

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
Nuclear and Radiological Engineering

Georgia Institute of Technology
December 2013

DETAILED ANALYSIS OF PHASE SPACE EFFECTS IN FUEL BURNUP/DEPLETION FOR PWR ASSEMBLY & FULL CORE MODELS USING LARGE-SCALE PARALLEL COMPUTATION

Approved by:

Dr. Glenn E. Sjoden, Advisor
Nuclear and Radiological Engineering
Georgia Institute of Technology

Dr. Farzad Rahnema
Nuclear and Radiological Engineering
Georgia Institute of Technology

Dr. Chaitanya S. Deo
Nuclear and Radiological Engineering
Georgia Institute of Technology

Dr. Bojan Petrovic
Nuclear and Radiological Engineering
Georgia Institute of Technology

Dr. Ce Yi
Nuclear and Radiological Engineering
Georgia Institute of Technology

Dr. Adam N. Stulberg
School of International Affairs
Georgia Institute of Technology

Dr. Robert Smith
Air Force Technical Applications Center

Date Approved: 27 August 2013

To my wife and future children

ACKNOWLEDGEMENTS

There are so many incredible individuals to thank for making the years fly by in graduate school.

I would first like to acknowledge my thesis committee members: Dr. Robert Smith, Dr. Adam Stulberg, Dr. Bojan Petrovic, Dr. Chaitanya Deo, Dr. Ce Yi, and Dr. Farzad Rahnema.

The achievements in this thesis would not have been realized without the aid of: Thomas Plower, Will Walters, Michael Chin, Dr. Tanguy Courau (EDF), and Dr. Ce Yi and his always excellent supporting software. Also, gracious thanks to Dr. Cory Ahrens for outstanding work on advanced icosahedral quadratures that have been applied in this work. A significant foundation of my education goes to the people formerly in the University of Florida Transport Theory Group (UFTTG), Georgia Tech's and Dr. Glenn Sjoden's CRiTCEL group, and the fine folks at PACE High Performance Computing at Georgia Tech.

It goes without saying that my family, Mom, Dad, Carol and her family, and Anne and her family, all have a special place in my heart for their ongoing support.

And of course, the best advisor and advocate for me, Dr. Glenn Sjoden. It's been a wonderful journey so far, and you have been the greatest guiding hand. What started as a small part-time job in undergrad blossomed to my path in graduate school, led by you through the University of Florida and then on to Georgia Tech. As both an advisor and friend, thank you for giving me that push and always providing the right touch of reinforcement and motivation at every step.

But my friend, my teacher, my classmate, my support, and most importantly, my wife and my everything: I would not have braved the journey to a PhD without you

Mimi. I love you and I truly could not have done this without you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS OR ABBREVIATIONS

ANSI	American National Standards Institute.
AR	Angular Resweep.
BC	Boundary Condition.
BR3	Belgian Reactor 3.
BURNDRIVER	The sequence driver controlling PENBURN-MC and PENBURN-SN.
C5G7	C5G7 Benchmark.
CP	Collision Probability Method.
CRAM	Chebyshev Rational Approximation Method.
DD	Diamond Differencing Scheme.
DM	Distributed Memory Model.
DSM	Distributed Shared Memory Model.
DTW	Directional Theta Weighted Differencing Scheme.
DZ	Diamond Zero Differencing Scheme.
EDI	Exponential Directional Iterative Differencing Scheme.
EO	Even-Odd Quadrature.
FOM	Figure of Merit.
HEU	Highly Enriched Uranium.
HMIX	Microscopic and Macroscopic Multigroup Cross Section Homogenization Code.
HPC	High Performance Computing.
IAEA	International Atomic Energy Agency.
IC	Icosahedral Quadrature.
LANL	Los Alamos National Laboratory.
LDFESA	Linear Discontinuous Finite Element - Surface Area Quadrature.

LS	Level-Symmetric.
MC	Monte Carlo.
MCNP	Monte Carlo N-Particle Transport Code.
MESH2MCNP	A Discretized Cartesian Geometry Creator for MCNP5.
MGOPT	Multigroup Cross Section Option Used in MCNP.
MOC	Method of Characteristics.
MOX	Mixed Oxide Fuel.
MPI	Message Passing Interface parallel message-passing software.
NRC	Nuclear Regulartory Commission.
OpenMP	An API for multi-platform shared-memory parallel programming in C/C++ and Fortran.
ORNL	Oak Ridge National Laboratory.
PCM	per cent mille or $10^{-5}\frac{\Delta k}{k}$.
PENBURN	Linear Chain Burnup/Depletion Solver coupled to PENTRAN or Multigroup MCNP.
PENDATA	Parallel Data Output Utility (Flux,Current,Source) for PENTRAN.
PENIMP	Code that converts PENTRAN Adjoint Importance to Weight Window Format in MCNP.
PENPOW	Reaction Rates Code that calculates reaction rates from multigroup flux (Sn or Multigroup MC) to PENBURN.
PENTRAN	Parallel Environment Transport Code.
PnTn	Legendre Chebyshev Quadrature.
PWR	Pressurized Water Reactor.
qfratio	Ratio of source to collision used by PENTRAN's adaptive differencing algorithm.
SCALE	Software Suite by ORNL for Nuclear Safety Analysis and Design.
SM	Shared Memory Model.

Sn	Discrete Ordinates.
TITAN	Hybrid MOC-Sn Transport Code.
TTA	Transmutation Trajectory Analysis.
XSMCNP	Multigroup Macroscopic Cross Section Code Conversion of Deterministic Cross Section Data Into MCNP Multigroup Format.
YGROUP	Multigroup Macroscopic Cross Section Collapsing Code.

CHAPTER I

INTRODUCTION

Burnup calculation, via computer simulation, enables accurate estimates of fuel transmutation in a reactor. In consideration of nuclear safeguards, burnup calculations can provide evaluators and inspectors an ability to esimate the maximum weapons usable material stockpiles that could be generated in a reactor, with changes based on actual history and fuel assemblies in use. Material accountancy could be readily improved with a full understanding of potential nuclear fuel or weapons adaptable and waste material generation for specific reactor types. For precise accuracy, transport computations are much better suited compared with the diffusion codes normally used for modern large core simulations. Transport methods are particularly important as reactor core sizes are reduced, fuel loading patterns vary, or mixed metal or oxide fuels are used. Moreover, because diffusion calculations are not well posed to represent accurate solutions in the vicinity of strong absorbers, material changes, etc., this places extra emphasis on the need for Boltzmann transport driven burnup and transmutation simulations for fuel management and non-proliferation applications. In addition, burnup software is also important in studying unconventional reactor types, new fuels, and next generation designs. This work demonstrates improvements in the timing of 3d transport (Monte Carlo and deterministic Sn) and burnup of nuclear fuel, which has evolved for practical use in 3d reactor simulations, to enable improved assessments of core inventory and isotopic variations in actinides after a specified burnup.

Parallel computing makes large transport simulations feasible. Also, parallel programming advances, that are in current use in outside fields, are considered in context for computational nuclear engineering and its use in enabling speedup of the transport and burnup calculations performed. With the appropriate background research, we present an argument for the feasibility and need for 3d full core burnup modeling with discrete ordinates transport.

If we can accurately track burnup and depletion, we can make accurate, cost-effective assessments of isotopics, for fuel in reactors, spent fuel, and various forms of high-level waste along the fuel cycle. A myriad of topics are involved (and required) in making valid burnup calculations, which this thesis addresses. As burnup is coupled to transport, relevant background is given for Monte Carlo and discrete ordinates methods that determine the solution of the fuel behavior in the reactor. We additionally discuss coupling between burnup and transport, and strategies that most burnup codes use in practice.

Consider how transport-based burnup can augment IAEA safeguards expenditures. With an increasing material stockpile of HEU and separated Plutonium under safeguards which increased from 20 metric tons to 120 metric tons from 1984 to 2004, the budget for IAEA safeguards budgets have not proportionately matched this increase [57]. In 2010, expenditures for IAEA safeguards were at $110 million, with 250 inspectors needing to inspect over 900 facilities spanning 140 countries [30]. There is a saturation limit for the number of inspectors needed in performing physical inspections; as one example, relaying information using 3d burnup calculations applied to specific reactor types with known inventory would give inspectors and evaluators more insight in investigating possible material diversion.

In general, calculations of detailed flux and reaction rates at all points in active fuel are of considerable interest to the engineers and scientists at national laboratories and universities, but all parties with involvement in nuclear nonproliferation and safeguards have a chance to benefit from the effective assessments. Furthermore, “virtual inspection” done with nuclear software, provided the scalable codes are available for parallel computing architectures, do not present significant additional costs relative to other means beyond the high performance computers needed for executing the simulations. Once the computations are concluded, follow-up analyses are then required to engage in valuable information exchange between engineers, scientists, evaluators, inspectors, and policymakers to enhance the expanding efforts in nuclear nonproliferation.

1.1 Objectives

The underlying theme of the thesis was to assess and improve accuracy issues related to space, energy, and angle - commonly referred to as the **phase space** - $(\vec{r}, E, \hat{\Omega})$. Space, energy, and angle define the mathematical phase space that play a significant role in burnup modeling, whether using Monte Carlo or Discrete Ordinates transport solvers. We considered issues of space with Cartesian 3d meshing, energy with multigroup optimization, and angle with advanced quadrature. This thesis studied various impacts on the phase space, with the ultimate goal to improve 3d burnup modeling. In particular, this thesis examines unique impacts in the area of 3d quadrature set for discrete ordinates (Sn) transport, Cartesian mesh and automesh generation, multigroup cross sections, and burnup uncertainties that are driven by transport solver method and discretization choice.

We initially addressed fundamental questions related to 3d full core burnup calculations and achived optimized calculation times or improved accuracy for a prescribed time window.

The questions are listed below:

- What are the merits of discretized mesh multigroup Monte Carlo coupled to burnup? Can we create a system that uses either discretized multigroup Monte Carlo or discrete ordinates transport?

- How does discretized mesh multigroup Monte Carlo compare to discrete ordinates transport for burnup?

- As other fields engage in hybrid parallel programming (using OpenMP/MPI to be discussed), is there is a benefit to using the OpenMP model inside the MPI-based model with 3d discrete ordinates PENTRAN software? Can we benefit from using an OpenMP/MPI hybrid model versus one that uses only MPI, as it pertains to 3d Sn algorithms?

- What are the best criteria for tuning adaptive differencing Sn algorithms in discrete ordinates transport?

Because these questions are somewhat generic and may often be problem dependent, the focus is generally limited to 17x17 PWRs and/or PWR pins available in open literature with benchmark data. In broader terms, the goal of this thesis was to engage in novel research with PWR models to optimize finely meshed model detail as a function of burnup. There is an emphasis on considering the phase space of angle, energy, and space; Subsequently, discretized Cartesian mesh Monte Carlo simulations are developed with a new modeling tool; this enables investigation and relevant cross-comparison of Monte Carlo and deterministic methods for steady-state and quasi-static time driven burnup calculations. Also significant is the parallelization optimization aspect incorporated into this work, which will include investigation of a hybrid distributed shared memory model (OpenMP/MPI), which is to be performed by using a shared memory programming model inside of a distributed memory programming model; this approach fully maximizes the use of current computing architectures, which operate in a distributed-shared memory model.

In the pursuit of computational modeling, our goals were:

1. to provide automated software tools to minimize engineer error and time,

2. to optimize performance of transport models,

3. to demonstrate state-of-the-art capabilities with 3d computational transport,

4. to analyze different solution approaches to burnup with different transport solvers

5. to research state-of-the-art parallel software methods.

1.2 Chapter Organization

Having covered the introduction and objectives this chapter, the following chapters are outlined. The next chapter provides background and literature review. The next three chapters cover issues related to the phase space. The following three chapters cover ongoing computational advances in adaptive difference, hybrid parallel, and burnup coupling advances (in that PENBURN auto-couples to either PENTRAN or MCNP).

The subsequent chapters address interesting subtopics at the forefront of ideal computational strategies that intersect computational discrete ordinates with advances in adaptive difference algorithm tuning and necessary improvements to parallel computation. In particular, the sixth chapter discusses practical adaptive difference tuning of PWR problems. The seventh chapter covers software parallelization of OpenMP into discrete ordinates transport with PENTRAN. The eigth chapter covers burnup benchmarks using PWR-based pins. The final chapter provides an overall summary and recommendations for future work.

1.2.1 Accuracies in Phase Space - Angle, Energy, and Space

Accuracies Related to the Phase Space - Angle, Energy, and Space

The primary theme of the thesis, while focusing on the operation of a burnup/-transport software suite, is the optimization tied to accuracies in the angular, energy, and spatial domains, covering a total of 7 state variables: 2 for angle, 1 for energy (multigroup), 3 for space (in x,y,z), and 1 for time.

Hence, the following three chapters (3-5) address just that: angle, energy, and space.

Angle is treated with the consideration of advanced quadrature implementation. Energy is treated with careful strategies for multigroup collapsing from a basis 238 group structure to a broader group structure. Space is considered by examining the potential issues with Cartesian mesh structures with the creation of an automesh tool and the impacts on eigenvalue variation for PWR-based fuel assemblies.

1.2.2 Adaptive Differencing and Application

In this adaptive differencing chapter, we examine PENTRAN's adaptive differencing algorithm and demonstrate benefits with 1d slab benchmark problem. We also cover the various differencing strategies used with the PENTRAN code particularly: Diamond Zero (DZ), Directional Theta-Weighted (DTW), and Exponential Directional Iterative (EDI).

1.2.3 Hybrid OpenMP-MPI Parallelization Optimization

OpenMP parallelization into the PENTRAN code is discussed in this chapter, further extending parallelization benefits. These benefits are demonstrated on a 64 CPU shared memory node using a PWR pin problem.

1.2.4 Burnup Benchmarking with BR3 Pin and MCNP Coupling to PENBURN

This chapter discusses the implementation of coupling the Monte Carlo MCNP5 solver in a format that is identical to that of discrete ordinates, and performs a validation of the Monte Carlo based burnup (PENBURN-MC) to PENBURN-Sn and also to a reference SCALE calculation.

CHAPTER II

BACKGROUND AND LITERATURE REVIEW

Next, section 2.1 will provide the background for the thesis covering: transport methods, cross sections, parallel programming, coupling transport to burnup, and burnup benchmarks.

2.1 Background

Accurate simulation of particles transporting radiation in a system can be described by the Boltzmann transport equation [36], which describes the behavior of neutral particles in terms of spatial, angular, and energy domains as they interact in a system; the steady-state form of the transport equation is given in Equation 1 using standard notation:

$$\begin{aligned}
&\hat{\Omega}\cdot\nabla\psi(\vec{r},\hat{\Omega},E) + \sigma_t(\vec{r},E)\psi(\vec{r},\hat{\Omega},E) \\
&\quad = \int_0^\infty \int_{4\pi} dE' d\hat{\Omega}' \sigma_s(\vec{r},\hat{\Omega}'\cdot\hat{\Omega},E'\to E)\psi(\vec{r},\hat{\Omega}',E') \\
&\quad + \frac{\chi(\vec{r},E)}{k_0}\int_0^\infty dE'\nu\sigma_f(\vec{r},E')\phi(\vec{r},E') + Q(\vec{r},\hat{\Omega},E)
\end{aligned} \tag{1}$$

ψ is the angular flux, and Q is the source term (that does not depend on ψ). Additionally, $(\vec{r},\hat{\Omega},E)$ are space, direction, energy, and time variables. σ_s, σ_t, and σ_f are scattering, total, and fission cross section values. χ is the fission spectrum, and ν is the fission neutron yield. The left side of Equation 1 represents streaming and collision terms (loss), and the right side represents scattering, fission, and other sources (gain). Since it describes the flow of radiation in a three-dimensional (3d) geometry with angular and energy dependence, this is one of the most challenging equations to solve in terms of complexity and model size. Rendering a deterministic computational solution for a large 3d problem or realistic scale usually requires a robust parallel transport solver and a high performance computing system.

The problem typically faced by researchers involves an ability to efficiently discretize and parallelize Equation 1 among the angular, energy, and spatial variables, although this problem has been solved in the discrete ordinates PENTRAN transport code used in this research. Although much less demanding to initiate and store on a computer system, large scale Monte Carlo simulations with a geometry described by combinatorial surfaces are used to effectively sample the phase space described by the Boltzmann equation with the application of appropriate probability distribution functions. While simple to define, these Monte Carlo models also require significant computer resources to stochastically sample large numbers particle histories with “smart” variance reduction methodologies to render credible, statistically based solutions. The problem with large scale stochastically sampled fission reactor simulations inherent in the Monte Carlo solution is that loose coupling of fission sources can cause significant variances and convergence difficulties not faced by deterministic solution methods [21], since large uncertainties in fission rate lead to large variances in fission yields and transmutation potentially propogate large errors in predicted inventory. The aforementioned variances make highly accurate predictions of burnup and depletion nearly impossible. One solution to be investigated in this work will be a comparison of burnup and transmutation of nuclear fuel between fully discretized deterministic PENTRAN and MCNP5 based Monte Carlo solution methods.

The next subsections briefly summarize the following background topics: Monte Carlo methods, discrete ordinate (Sn) methods, advances in adaptive differencing for discrete ordinates, and burnup/depletion methods.

Section 2.1.8 provides background on parallel programming and its use with transport solvers (Monte Carlo and discrete ordinates). Section 2.1.9 develops Burnup/Depletion coupling considerations, and the final section wraps up with selected burnup benchmarks in the literature that are also performed in this thesis.

2.1.1 Monte Carlo Method and Related Codes

The Monte Carlo method, a statisical sampling technique, was developed by physicists Ulam and von Neumann at the Los Alamos National Laboratory (LANL) in the invesitgation of a neutron-diffusion calculation [23]. Simulation with the Monte Carlo method can be simplified to related a series of generated random numbers aliased to physics interactions of particle transport. To be specific, one type of application of the Monte Carlo method to particle transport is the tracking of a particle history, where the particle is tracked in free flight, and the probability of interaction is sampled to determine the particle's next event, with the eventual end result of the particle being absorbed/killed or escaping from the defined problem boundaries.

We present a few details relating Monte Carlo transport with the time-dependent linear Boltzmann equation (Equation 2):

$$\begin{aligned}\frac{1}{v}\frac{\partial}{\partial t}\Psi(\vec{r},\hat{\Omega},E,t)+\hat{\Omega}\cdot\nabla\Psi+\Sigma_t\Psi-S \\ =\iint\Psi(\vec{r},\hat{\Omega}',E',t)\Sigma_s(\vec{r},\hat{\Omega}'\cdot\hat{\Omega},E'\to E)dE'd\hat{\Omega}'\end{aligned}\tag{2}$$

By application of the method of characteristics, we can cast the transport equation into the form

$$\begin{aligned}&\Psi(\vec{r},\hat{\Omega},E,t)= \\ &\int_0^\infty e^{-\beta}\Big[\iint\Sigma_s(\vec{r}-s\hat{\Omega},\hat{\Omega}'\cdot\hat{\Omega},E'\to E)\Psi(\vec{r}-s\hat{\Omega},\hat{\Omega}',E',t-\frac{s}{v})d\hat{\Omega}'dE'\Big]ds\end{aligned}\tag{3}$$

where β is the optical thickness, defined as

$$\beta = \int_0^s \Sigma_t(\vec{r} - s'\hat{\Omega}, E)ds' \tag{4}$$

The key point to address is that the angular flux can be considered as the integration over all source and particle flight paths to that point with appropriate restrictions of energy, drection and time [22].

If rewritten in operator notation,

$$\Psi = \mathbf{K\Psi} + \mathbf{S}' \tag{5}$$

Defining the following

$$\begin{aligned} \Psi_0 &= S' \\ \Psi_1 &= \mathbf{K\Psi_0} \\ \Psi_{n+1} &= \mathbf{K\Psi_n} \end{aligned}$$

It could be subsequently argued that if the series converges, and a solution to Equation (also known as the Von Neumann series)

$$\Psi = \sum_{n=0}^{\infty} \Psi_n \tag{6}$$

Then in particle transport, presuming the Von Neumann series is valid, we can track a particle in six-phase space: three variables to specify 3d coordinate location, two variables to specify its trajectory, and one variable to specify particle energy.

2.1.2 Exact Geometry vs. Discretized Cartesian Mesh Geometry in Monte Carlo

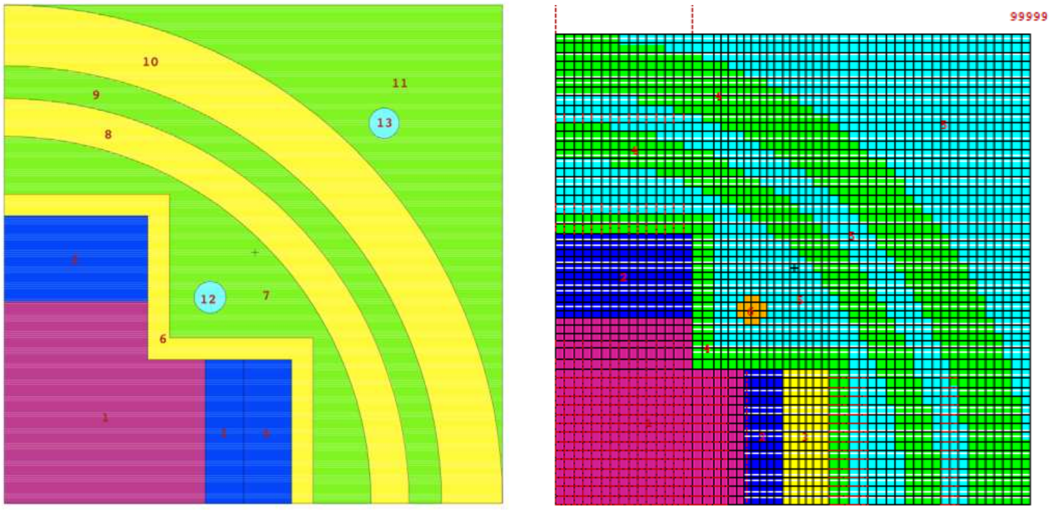
This thesis places additional consideration of the mesh geometry, as there is a need to represent problems with materials defined in complete occupation of a problem boundary, typically within a rectangular parallelpiped. Traditional Monte Carlo codes can track particles in exactly specified surfaces and related cell definitions; the geometry is created via specification of surfaces and the cells that are enclosed by those surfaces. Alternatively, we can discretize the same model in a Cartesian mesh grid, which is a more suitable format for comparison to other transport codes using this format (such as 3d Cartesian discrete ordinates).

As a result of the discretized geometry mesh grid, discrepancies in material balance can occur, when converting a model from “exact” to “discrete”. This material imbalance is the primary focus in Chapter 5, that in particular, examines mesh issues (both Sn and MC) in model conversions of PWR assemblies, which are driven from using automated model conversion from Sn to Monte Carlo.

An example of exact versus discrete mesh geometry is shown in Figure 1, which is a stylized depiction of a quarter core of the VENUS reactor (a zero-power critical facility (dismantled) in Belgium).

2.1.3 Continuous vs. Multigroup Cross Sections

An example of continuous versus multigroup cross sections is given below. Figure 2 was generated using the NJOY cross section processing software [39]. Here Pu-243 is shown in 'continuous' form in dark blue, and the 3 group cross sections are shown in light purple. The calculation of multigroup cross sections is not just simple averaging, and resolving proper group constants is challenging (the extent of which is not addressed here). This is not to undermine the importance of calculating cross sections; it is crucial to have the most accurate set of cross sections, whether continuous or multigroup, as they provide the foundation for valid transport results.



(a) Exact Combinatorial Geometry (b) Cartesian Mesh

Figure 1: Exact vs. Discretized Cartesian Mesh of VENUS Reactor

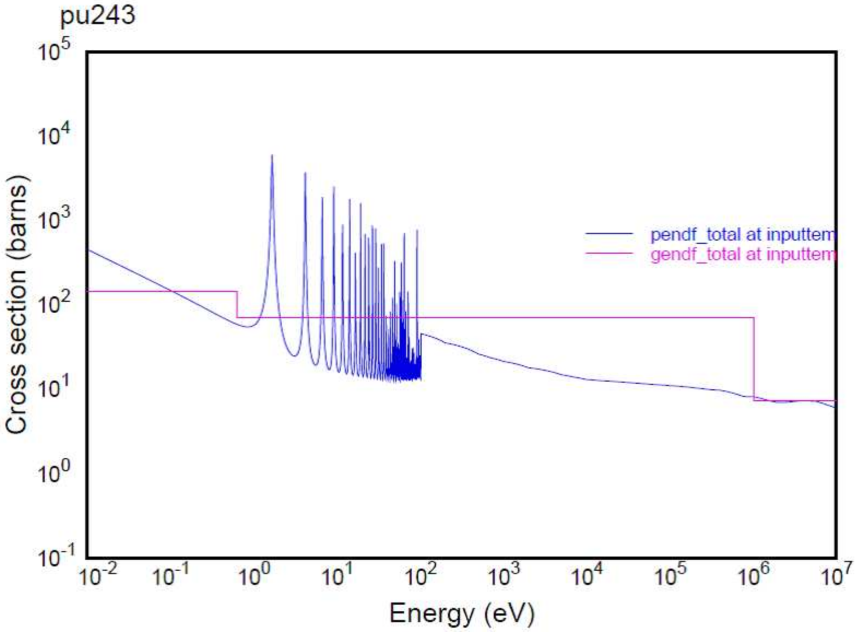


Figure 2: An Example of Continuous (blue) and Multigroup Cross Sections (pink) for Pu243

The code that will be used for Monte Carlo analysis is MCNP5 [60]. This code was coupled to the burnup module PENBURN as part of the thesis effort.

2.1.4 Distinguishing Types of Monte Carlo Simulations

One key background point to establish for this thesis project is the type of Monte Carlo simulations. From the prior sections (2.1.2 and 2.1.3), we can enumerate the following, this convention will be referred for specific calculations in the rest of this thesis document:

Type 1: Exact geometry Monte Carlo with "Continuous" energy cross sections

Type 2: Exact geometry Monte Carlo with multigroup energy cross sections

Type 3: Discretized geometry Monte Carlo with "Continuous" energy cross sections

Type 4: Discretized geometry Monte Carlo with multigroup energy cross sections

Monte Carlo codes, such as MCNP5, can use either type of cross section format; running multigroup cross sections in MCNP5 is typically done for the purpose of benchmarking deterministic transport so that a more applicable comparison can be performed between the two methods. Another suitable code that uses multigroup cross sections is KENO, a multigroup Monte Carlo code available with the SCALE package.

2.1.5 Discrete Ordinates Method and Related Codes

The discrete ordinates method considers a solution to the linear Boltzmann equation, given earlier in Equation 1.

As developed by Carlson in 1958 [11], the angular flux (angular domain) is discretized into a careful choice of discrete directions such that a function can be evaluated at discrete points on a surface. Then, flux moments are preserved with quadrature integration over the discrete set of points. The spatial domain is handled by forming finite difference relationships based on the Boltzmann equation to evaluate the streaming.

The transport code that applies the discrete ordinates method, used with our research group at the Georgia Institute of Technology is PENTRAN [55] , developed by Sjoden and Haghighat. PENTRAN is a parallelized 3d Cartesian Sn neutral particle transport solver with domain decomposition in angle, energy group, and space, written using the MPI Fortran / ANSI Fortran programming standard. We will defer a discussion of parallel architecture to Section 2.1.8.

2.1.6 Adaptive Differening with Automatic Tuning in Sn Codes

Advances in Adaptive Differening with Automatic Tuning in Sn Codes (PENTRAN)

One particular focus in PENTRAN is use of adaptive differencing schemes. PENTRAN allows for mesh density refinement in a block coarse mesh/ fine mesh structure; one method for improving the solver is the application of adaptive mesh refinement; where the mesh densities are changed to adjust to the transport problem. Alternatively, PENTRAN uses a static mesh and adjusts the Sn differencing algorithm during the calculation in a localized coarse mesh to allow for a stable mass representation. We will brief the basic methods of the differencing below. More details are provided in a co-authored reference paper [52] and detailed in Chapter 6.

The adaptive differencing scheme is designed so that more time is spent on the problem changing the finite difference transport operator based on the criteria listed below, in a one way upgrade scheme (steps given) [54, 52]:

1. Diamond Zero (DZ) set as the basis Sn algorithm
2. Upgrade from DZ to Directional Theta Weighted (DTW) if DZ flux fix-ups are required
3. Upgrade from DTW to Exponential Directional Iterative (EDI) if gradients become too steep based on weights/metrics

Upgrading from Step 1 to Step 2 is a very explicit decision. However, the decision to upgrade using the second step criteria from DZ to EDI from step 2 to step 3 requires more careful consideration. One reason is that the exponential treatment in EDI has the highest computational cost because of the cost of computation in using the exponential function, and also because EDI is a fixed point iterative method). There are essential 2 decision criteria in this general statement, which we will discuss.

In order to upgrade from DTW to EDI:

1. Upgrade is not performed if the optical cell thickness is vanishingly thin (usually the value of 0.02 by default, can be changed by user).

2. Upgrade from DTW to EDI is considered if DTW weight factors are approaching unity (a value specified by the user, usually 0.95)

    (a) If DTW weight factors are greater than specification, then determine:

        i. If coarse mesh cell is 'streaming dominated' via source fraction ratio to collision we call *qfratio*, → apply EDI scheme in coarse mesh

        ii. If 'source dominated' retain and → apply DTW scheme

The first step considers the optical cell thickness $\sigma\Delta h_{max}$. If the optical thickness is less than 0.02 where DTW weight factors are nearly unity, and the DTW scheme is not upgraded to the EDI under any circumstance, as the DTW scheme is adequate in resolving the angular flux. The second step considers the use of DTW weight factors. If values of DTW weight factors are approaching unity (and thus a 'step' scheme), this implies that angular fluxes are relatively flat, meaning that the cell is either in a streaming situation with low optical thickness and no or low-level sources, or in a cell location with a strong source. In these situtations, it is desirable to hold the DTW scheme in the strong source case, because the 'step' scheme is effective. To improve the decision criterion, we add in a consideration of *qfratio*. If one divides the multigroup transport equation by the collision density we have

$$\frac{\hat{\Omega} \cdot \nabla \psi_g}{\sigma_g \psi_g} + 1 = \frac{(q_{scat,g} + q_{fiss,g} + q_{indpt,g})}{\sigma_g \psi_g} = qfratio \tag{7}$$

where the group sources are denoted using scatter, fission, and independent source variables. If *qfratio* > 1.00, then clearly a cell is 'source dominated'; in this case we preserve the DTW scheme for use in transport solution. Otherwise, if *qfratio* < 1.00, a cell is 'streaming dominated' and upgrade to EDI occurs. The effectiveness of this will be investigated further.

2.1.7 Transport-Based Burnup and Depletion Methods

The transmutation of fuel in a reactor occurs during operation because of fission. In defining fission in thermal power fission reactor, a neutron is captured by a heavy element, resulting in: energy release, the splitting of a fissile atom usually into binary pieces each having their own kinetic energy, prompt and delayed gamma energy release, neutrino release, and also neutron emission (usually two or three). The fission fragment pieces are defined as fission products. Fission products are also generally radioactive, neutron rich, and subject to decay, neutron capture, and other interactions. How the fuel changes (in the non-kinetics based context, over longer time periods of a day or greater) is governed by the transmutation equations (first order differential equations, solved by Bateman [7]).

There are two major types of methods which solve the burnup/depletion Bateman equations: linear chain methods and matrix exponential methods. Both methods each have a set of advantages and disadvantages and will be discussed briefly.

2.1.7.1 Linear Chain Methods

The Bateman equations can be solved for *batch* and *continuous* production modes. Here, we present the solution of continuous production (when a reactor is operating)

$$\begin{tikzcd}
P \arrow[d] & & & \\
N_1 \arrow[r, "\xi_1"] \arrow[d, "\mu_1-\xi_1"] & N_2 \arrow[r, "\xi_2"] \arrow[d, "\mu_2-\xi_2"] & N_3 \arrow[r, "\xi_3"] \arrow[d, "\mu_3-\xi_3"] & \cdots \\
{} & {} & {} &
\end{tikzcd}$$

Figure 3: Production Linear Bateman Chain with N_1 as a Producer

in Equation 8. This solves for concentration (in atoms or grams or atoms/bn-cm) of N as a function of time, given a production rate P, assuming that concentrations N_1, N_2, $\ldots$, to N_i are initially zero. The values are chain-linking precursor decay rates, and the values are the effective decay rates for each nuclide N. Figure 3 is also provided to enable visualization of the transmutation in action during continuous production. Obviously, in consideration of burnup, P would be the fission product yield attributed to an actinide (for example, uranium and plutonium) times the available actinide concentration, for all actinides that have fission yields.

$$N_i(t) = \sum_{l=1}^{i-1} \left[P_l \xi_l \xi_{l+1} \cdots \xi_{i-1} \sum_{j=l}^{i} \frac{1 - e^{-\mu_j t}}{\prod\limits_{\substack{k=1 \\ k \neq j}} \mu_k - \mu_j} \right] + P_i \frac{1 - e^{-\mu_i t}}{\mu_i} \tag{8}$$

Linear chain methods are direct methods. The solution is already calculated for one chain. The complication of the method is when many chains are involved (an example is given in Figure 4). Each number represents a nuclide being tracked. The figure demonstrates that there are 4 pathways to nuclide N_4 and nuclide N_3. There are 2 pathways in tracking nuclides N_6 and N_2. Nuclides N_1 and N_5 each contribute to the path but can be treated independently. Again, this is a linear chain because all unique contributions to a nuclide are simply superimposed.

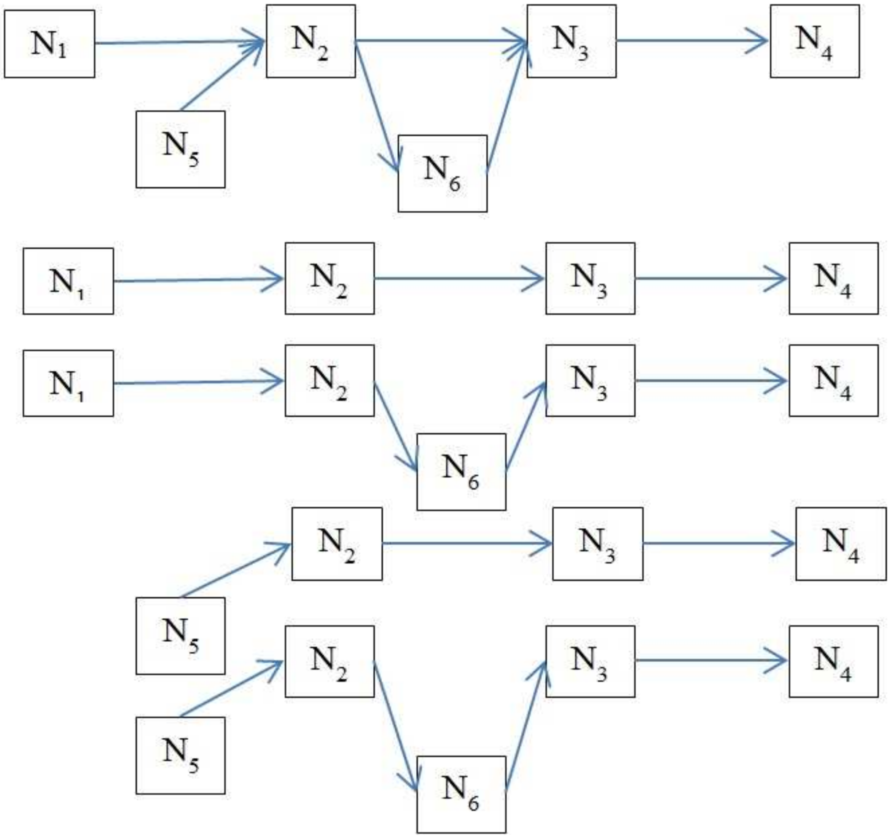


Figure 4: Linear Chain Enumeration - The top chain can be enumerated as 4 different pathways

The advantage that linear chain methods have is that there is little computational overhead because this is a direct solution method. The disadvantage of this approach is the need to explicitly model nuclide chains and enumerate all pathways. This hinders the number of nuclides (fission products and actinides) to the order of 100s.

There have been recent advances in the treatments of Bateman equations, which deal with existing numerical problems, where near-equal decay constants subtract in the denominator of fractional terms, stressing numerical precision. An improvement to the solution of Bateman equations and method by Cetnar[13], referred to as Transmutation Trajectory Analysis (TTA), has remedied this issue.

2.1.7.2 Matrix Exponential Methods for Burnup

Another viable and alternative method is the matrix exponential method. The first order differential equations that govern burnup can be modeled in a system of equations such that

$$\mathbf{n}' = \mathbf{A}\mathbf{n} \qquad \mathbf{n}(0) = \mathbf{n}_0 \tag{9}$$

where $\mathbf{n}$ is a vector of real-valued concentrations and $\mathbf{A}$ is a burnup matrix containing matrix coefficients.

A method for solving the system is performed by using the matrix exponential method where

$$\begin{aligned} \mathbf{n}(t) &= e^{\mathbf{A}t}\mathbf{n}_0 \\ e^{\mathbf{A}t} &= \sum_{k=0}^{\infty} \frac{1}{k!}(\mathbf{A}t)^k \end{aligned} \tag{10}$$

The difficulty in solving the matrix exponential is that the decay constants of the nuclides are widely ranging, resulting a wide eigenvalue spectrum [49]. A recent advance in burnup equation solution methods that is notable is the Chebyshev Rational

Approximation Method (CRAM), used as the default option in the SERPENT code (lattice physics Monte Carlo code coupled to burnup) [48]. A detailed comparison of Matrix Exponential versus CRAM is provided in Appendix A.

2.1.7.3 Burnup Code Development by the Author

The foundation for my Master's thesis work was in writing the PENBURN burnup/depletion solver [40] initially coupled to PENTRAN. PENBURN uses the linear chain method, as discussed in Section 2.1.7.1, and was initially designed to interface with PENTRAN. The code is an essential module of the BURNDRIVER sequence, which manages the coupling of PENBURN and PENTRAN. Future reference to PENBURN coupled to PENTRAN will be referred to as PENBURN-Sn, and coupling to Monte Carlo (MCNP) referred to as PENBURN-MC. We have demonstrated successful use of PENBURN on pin and assembly levels for PWRs [46, 47].

2.1.7.4 Hybrid Use of Computational Methods and General Method for Coupling Burnup and Transport

Two transport methods, Monte Carlo and discrete ordinates, are viable for use in burnup. In particular, the thesis involves weighing the merits of Monte Carlo versus deterministic issues as it pertains in coupling burnup and transport. A general framework for the use of transport and burnup in a code system is given below, that is currently used by BURNDRIVER, a Bash script used in Linux Operating Systems. A more generalized process is given in Figure 5, which effectively details the most basic approach to coupling burnup/depletion and a transport solver.

2.1.8 Parallel Programming and Use with Transport Solvers

An undercurrent to this entire set of research is that the computations would not run in a feasible time without the aid of parallel computers. In the broadest sense, the use of parallel computers allows for a great amount of wall-clock time saved in performing computation.

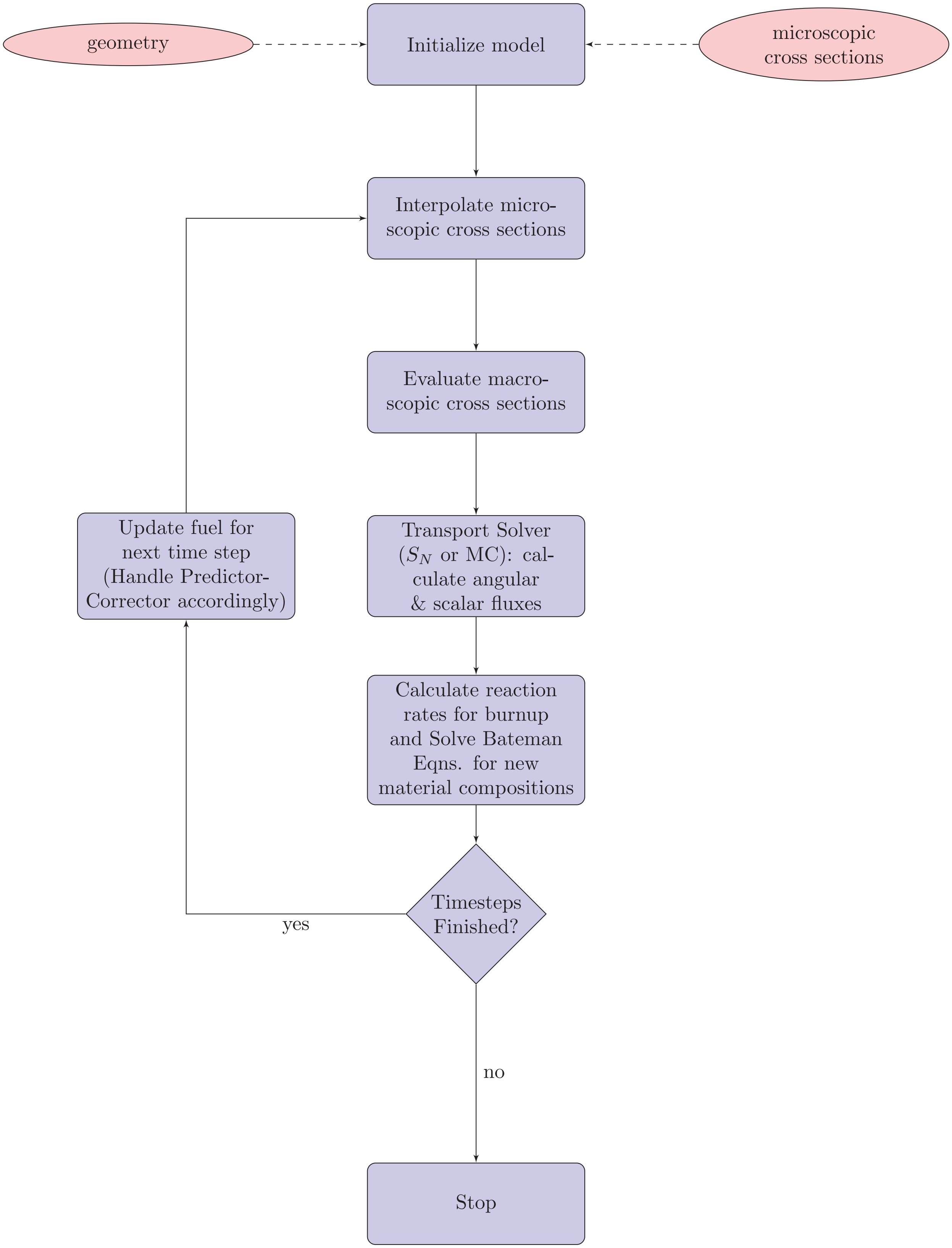

geometry
Initialize model
microscopic cross sections
Interpolate microscopic cross sections
Evaluate macroscopic cross sections
Update fuel for next time step (Handle Predictor-Corrector accordingly)
Transport Solver (S_N or MC): calculate angular & scalar fluxes
Calculate reaction rates for burnup and Solve Bateman Eqns. for new material compositions
Timesteps Finished?
yes
no
Stop

Figure 5: General Methodology for Coupling Transport to Burnup

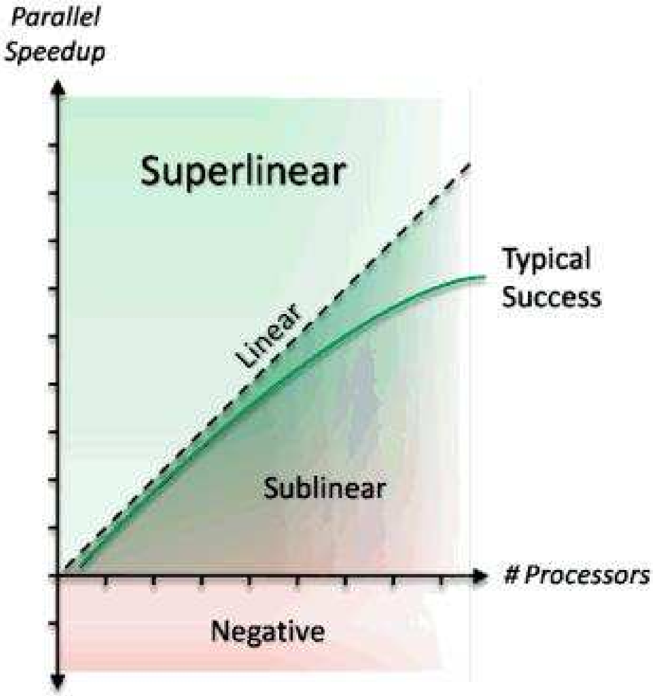


Figure 6: Parallel Speedup Diagram [59]

A clear benefit to parallel computers is that, a typical serial program could experience a speedup that is linear to the number of processors used. Figure 6 presents a graph depicting speedup. In an ideal situation, there is 2x speedup for 2 processors, 3x speedup for 3 processors, and so on. This is termed linear speedup. In certain applications, it is quite possible to achieve in superlinear speedup, for example running an application on 2 processors results in a 3x speedup. However, for most programs, this is rare and atypical. A curve that falls just under linear marked "Typical success" in the figure is a more likely result of parallel scientific software.

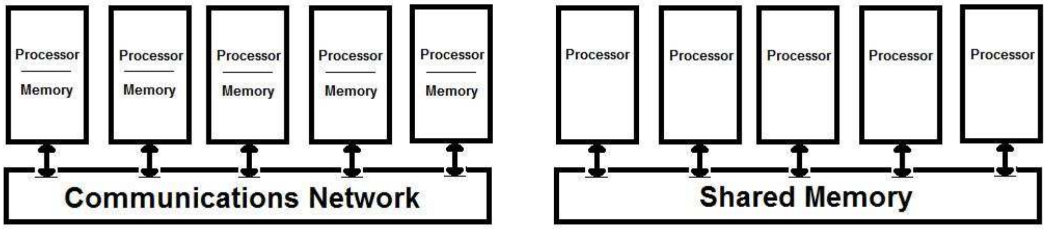
One reason why Monte Carlo is increasingly popular is that it is "embarrassingly parallel"; this is a technical term and compliment to the nature of statistical simulation. As an example, 1 million particles could be split into 10 tasks of 100,000 particles; the results of 10 event simulations are then coalesced together to form the final result. In the past, during the inception of the Monte Carlo method in the 1940s, Monte Carlo was highly impractical because it is inherently required computers for numerical computation. On the other hand, deterministic methods at that time were popular because calculations could be done by hand.

The advancement of computing technologies has not formed a disregard for deterministic methods; currently, deterministic methods have not fallen by the wayside. Deterministic codes, such as PENTRAN are also designed with parallelism in mind [54]. The software was written using MPI Fortran, which comprises a set of programming primitives that are written on top of the base Fortran language. The constructs enables communication and data transfer between each CPU, each having its own memory. This parallel programming paradigm follows a distributed memory model. Another example of a parallelized discrete ordinates code under active development is the Denovo code, developed by ORNL [25].

An emergent architecture common with computer clusters, distributed shared memory (will refer to DSM for convenience), is essentially the norm for most clusters in operation today. However, a majority of software has not yet caught up to fully take advantage of this, simply because of the complicated effort it takes to write a parallel program in the first place.

We can break DSM up into two components. Shared memory (SM) programming assumes that all the CPUs share the same memory. This is prevalent with even consumer CPUs touting dual-core or quad-core architecture. This in turn, affects the parallel programming model, making it a simple process to parallelize code; however, while the writing of code is an advantage, a disadvantage is that the shared memory program can only extend to the span of shared memory, which is usually limited to a node/workstation (a fragment of a full cluster). As of 2013, OpenMP is the most popular standard for shared memory parallel programs.

Distributed memory is a model where each processor with its own memory communicates across a network. An ad hoc "Beowulf" cluster system would be simply to purchase a set of workstations and connect them via a wired router. Similarly, MPI



(a) Shared Memory Model

(b) Distributed Memory Model

Figure 7: Memory Model Diagrams

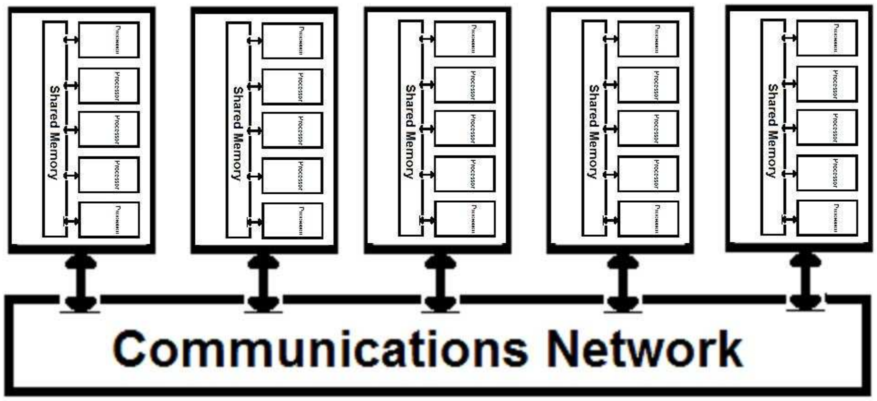


Figure 8: Distributed Shared Memory Model

programming, as was used for PENTRAN, is the standard for distributed memory (DM). Figure 7 demonstrates the DM and SM concepts. Figure 8 is the DSM model. For example, the Reagan cluster in use by our research group has 7 distributed nodes (with 96 GB), each having 12 CPUs or 24 threads with hyper-threading, enabling the potential use of 84 (7x12) total MPI tasks, or 7 MPI tasks of 24 threads used in OpenMP.

Naturally, there is a consideration to integrate both forms of MPI and OpenMP into a hybrid program. Since MPI can run on DSM and DM architectures, why invest in hybrid programs? Hybrid programs can perform faster than sole OpenMP or MPI counterparts. Conveniently, OpenMP is easier to integrate into MPI Programs than vice versa. By easier, we mean that the software design does not require a major restructuring, as code written for the shared memory model can be tucked inside of distributed memory. A more detailed discussion is available as reference [35].

The scope of computational modeling is burdened by the onset of emergent parallel libraries and new parallel computer languages. There are other shared memory models such as Cilk and TBB, that could be used instead of OpenMP. Even new architectures are rapidly explored in the development of scientific programs, such as Graphics Processing Units or GPUs. There is rapid advancement in the Computer Science field in the development of Partitioned Global Address Space (PGAS) languages Unified Parallel C, Co-array Fortran, and Chapel, as some examples. However, even with the traction gained for new languages and parallel programming models, MPI and OpenMP are amongst the most popular in use amongst scientific programmers for parallel programming.

2.1.8.1 Parallel Fraction and Amdahl's Law

Amdahls law is named after Gene Amdahl[6], computer architect, making a simple but effective statement about the theoretical maximum speedup using processors in parallel.

In considering parallel fraction f, and number of processors n, the maximum speedup that can be achieved is

$$\text{Maximum Speedup} = \frac{1}{(1-f) + \frac{f}{n}} \tag{11}$$

As one example, if a parallel program has a fraction 0.75, this implies that $\frac{3}{4}$ of the program can make full use of parallelization of the processors available; for a parallel fraction of 0.75, as the number of processors tends to infinity, the maximum theoretical speedup in the limit is then (by inspection of Equation 11) the inverse

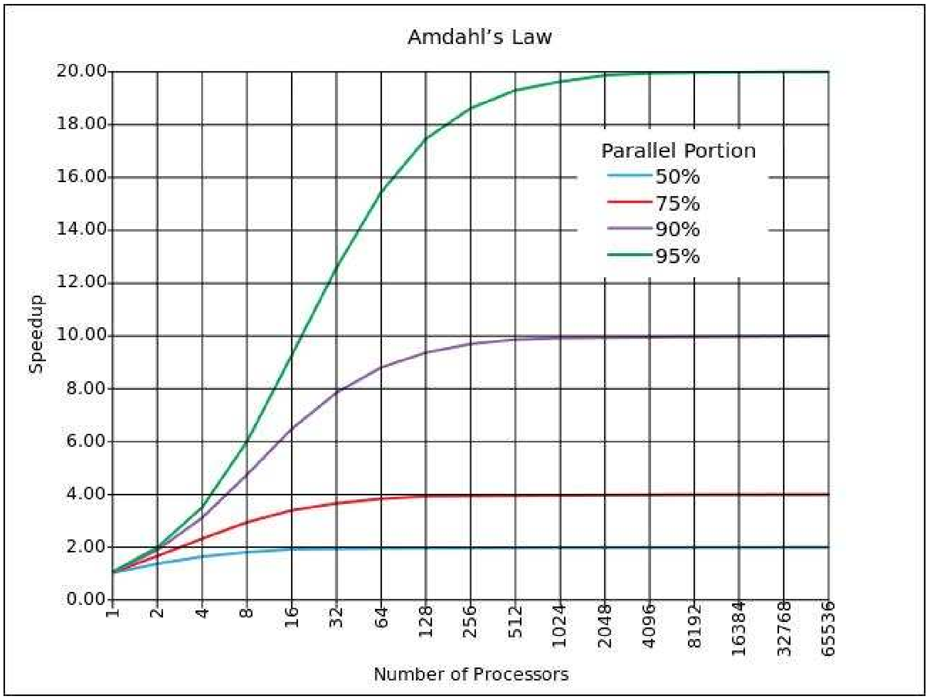


Figure 9: Parallel Fraction and Amdahl's Law [1]

of one minus the parallel fraction f, which in this case is 4. This is a somewhat discouraging result for a parallel program, because if the parallel fraction had been 0.95, the program could have experienced up to a factor of 20 speedup.

A code to compute π to a number of significant digits is simple to achieve high speedup; the integro-differential Sn equations are another matter. Therefore, a parallel fraction must be as near unity as achievable, to take maximum advantage of the parallel computer speedup available. PENTRAN, for example, has a very high estimated parallel fraction of 0.996 [18].

Amdahl's law may take on a pessimistic and unforgiving view towards parallelizable programs. The thesis will explore the application of Amdahl's law to PWR problems solved with OpenMP-MPI parallel software and also discuss other pertinent computer science laws related to simulation on parallel computers.

2.1.9 Burnup/Depletion Considerations - Cross Sections and Coupling

This section discusses the necessary groundwork in creating a burnup solver, and considerations made for burnup/depletion. First, we discuss burnup cross sections.

2.1.9.1 Interpolation of Cross Sections as a Function of Burnup, Fuel, and Moderator Temperature

Multigroup cross sections are not only energy and spatially-dependent (on material). initially defined as $\sigma_x = \sigma_x(\hat{\mathbf{r}}, E) = (\sigma_x)_{mat,g}$, where x is the type of interaction, mat is the material the cross section belongs to, and g is the energy group.

We could also more precisely define cross sections as a function of 5 variables:

$$\begin{aligned}\sigma_x &= \sigma_x\left(\hat{\mathbf{r}}, E, T_{fuel}, T_{mod}, t_{burnup}\right) \\ &= (\sigma_x)_{g,mat}(T_{fuel}, T_{mod}, t_{burnup})\end{aligned} \tag{12}$$

Multigroup cross section collapse and weighting are non-trivial for both deterministic and Monte Carlo codes that rely on multigroup.

When necessary to incorporate into multigroup cross sections, the three variables of fuel temperature, moderator temperature, and burnup time are handled by interpolating against a cross section library for a unit cell pin related to the full core model. More robust spatial treatment is performed with further increase in spatial detail, for example, using $\frac{1}{8}$th core assembly cross sections (with cross sections specified on each pin).

An example of cross section dependence on burnup is shown in Figure 10. This figure demonstrates that for most stages of burnup, isotopic accuracy of Pu-240 would be impacted greatly if only using an average value. The cross sections change by nearly a factor of 3.5 over the course of the burnup to 65 gigawatt days per metric ton heavy metal (GWd/MTHM).

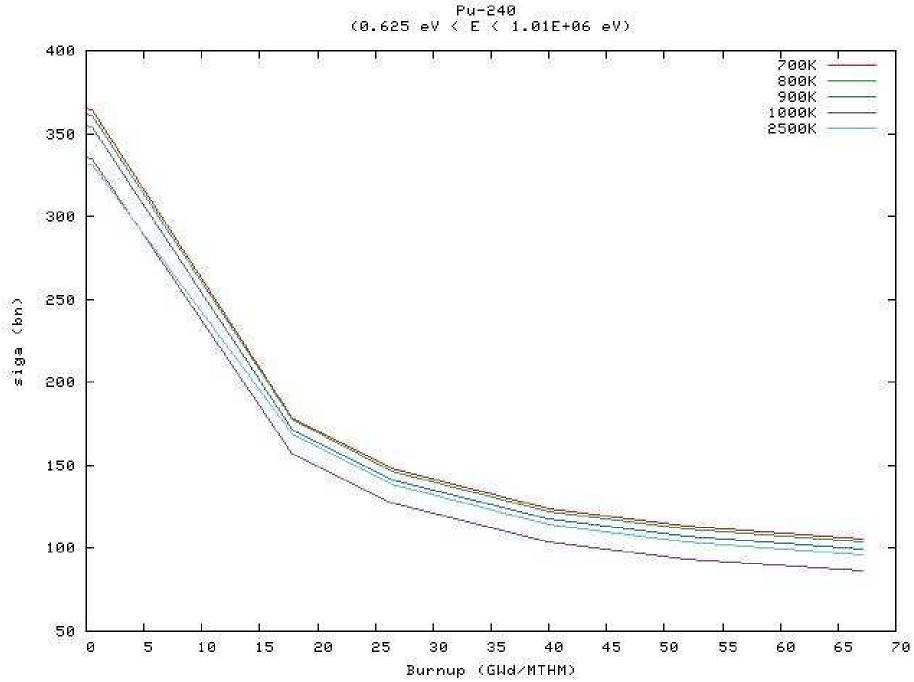


Figure 10: Pu-240 Cross Section as a Function of Burnup, with varying Fuel Temperature [46]

2.1.9.2 Forward-Euler and Predictor Corrector Burnup-Transport Coupling Schemes

The standard option of a quasi-static rotation between transport and burnup, is diagrammed in Figure 12. This is simply termed the Forward-Euler approach, as it is similar to the first order numerical procedure, and is predisposed to small time steps for accuracy. If the time steps are too large, burnup inaccuracies can occur. The problem is inherently similar to problems with the Euler method, where the calculation of a function can diverge from its true solution. See Figure 11 as an example of this function divergence; the blue curve is the true solution, and the Euler method calculates the points in red. This is known as an overshoot, since the time steps are too long to resolve the original function properly.

We can mitigate issues with Forward-Euler by improving the coupling. Since burnup and transport are actually simultaneously occurring within a nuclear reactor, coupling burnup and transport involves a quasi-static rotation, where it is presumed

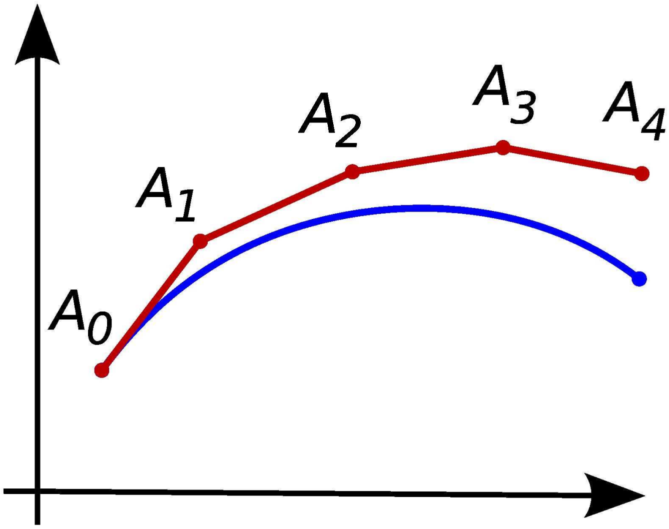


Figure 11: Red curve using Euler method, diverges from blue curve

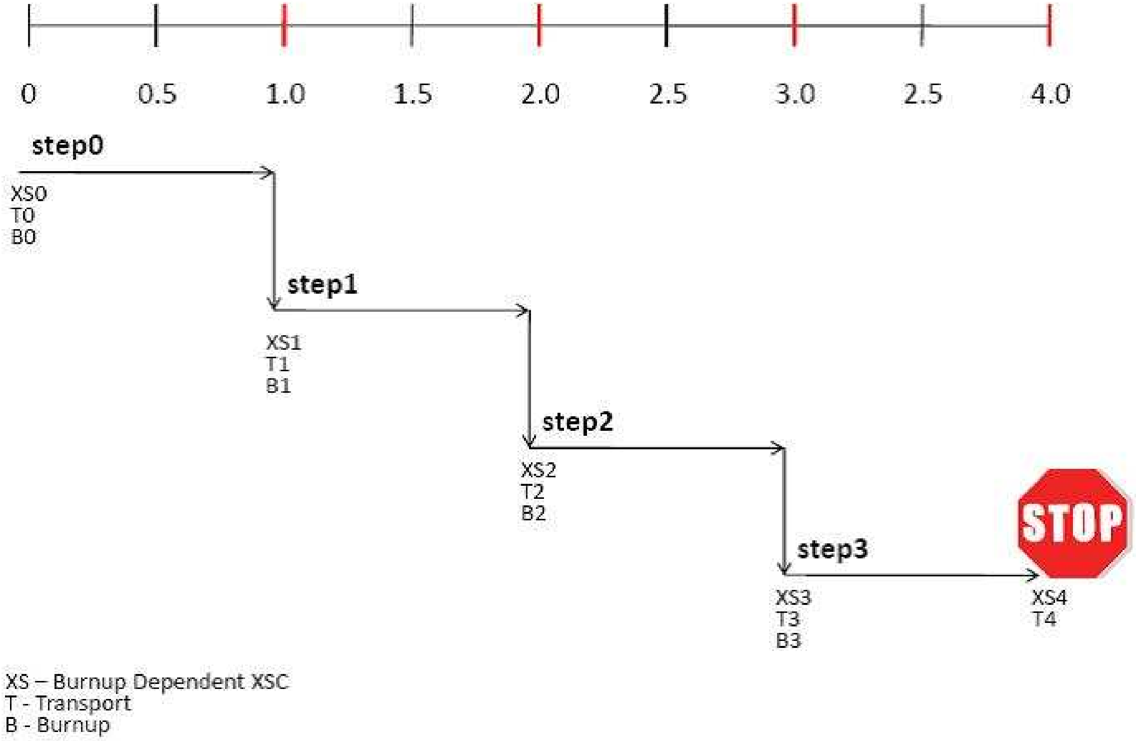


Figure 12: "Forward-Euler" Approach Applied in BURNDRIVER [46]

that the transmutation/burnup is a future projection in time and after each burnup, a new transport calculation is performed which represents the physics exactly at time Δt. This is not necessarily true, as transport was simultaneously occurring during the transmutation. To counter this effect, the burnup calculation is performed to the midway point of a step, where the transport solver is used again to calculate fluxes; the burnup calculation is then re-performed over the entire step using the midpoint fluxes and midpoint cross section information. An example of this is shown in Figure 13 with the BURNDRIVER tool. The application of Predictor-Corrector methods improves burnup solution accuracy, albeit at the expense of additional transport simulation cost.

Most burnup codes have predictor-corrector as an option, with variations on the implementation. The thesis will apply both Forward-Euler and Predictor Corrector schemes (where practical/feasible) and consider their effect on accuracy. We note that PENBURN, SCALE (T-DEPL), and SERPENT are examples of burnup codes that all have a form of predictor-corrector scheme that can be used as an option.

2.1.10 Previous Burnup Benchmarks with 2d/3d Burnup Analysis

There is a large collection of data comparing 2d/3d burnup code simulation offering relevant comparisons to SFCOMPO Takahama data. Previous analysis was performed, on a single PWR pin, in my Master's thesis work [40]. Other analyses were performed by Sanders, DeHart, and Ilas using SCALE [50, 43, 31]. Another analysis was performed by Dalle using MONTEBURNS [19]. The thesis project will provide appropriate comparisons using a designated BR3 benchmark discussed in Chapter 8.

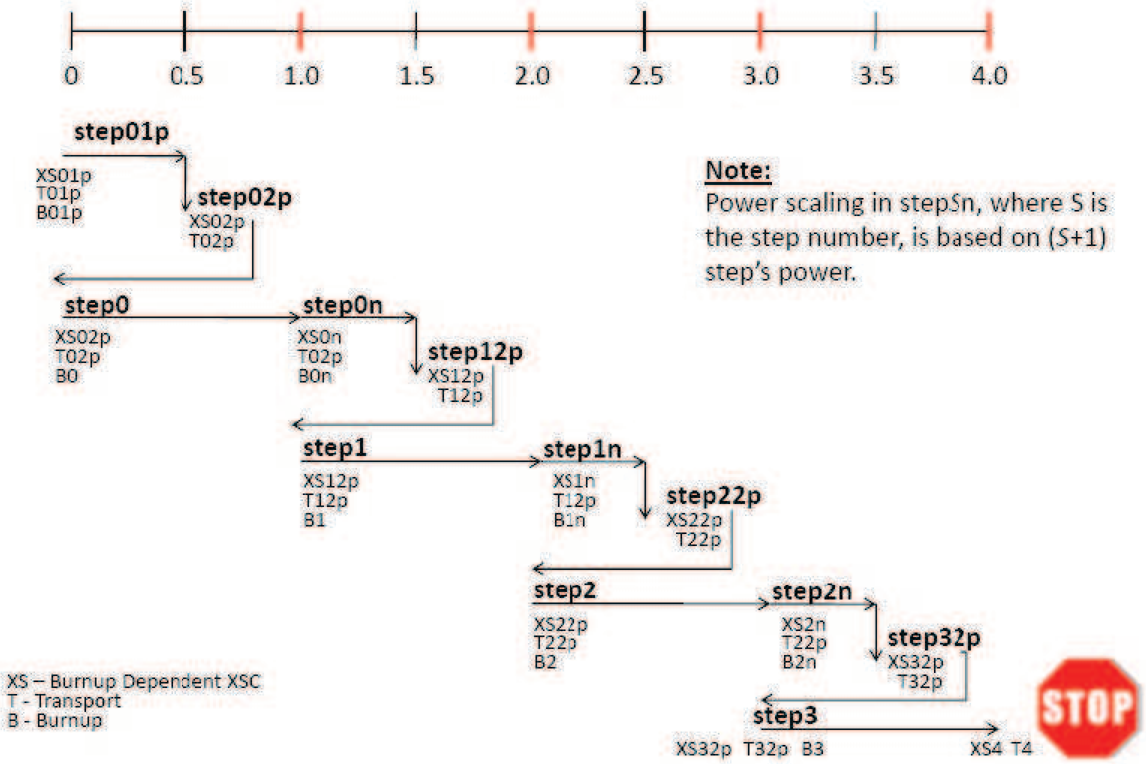


Figure 13: Predictor-Corrector Approach Applied in BURNDRIVER [46]

CHAPTER III

ADVANCED QUADRATURE (ANGLE) OPTIMIZATION

Significant updates in quadratures have warranted investigation with 3d Sn discrete ordinates transport. We show new applications of quadrature departing from level symmetric ($< S_{20}$) and Pn-Tn ($> S_{20}$), investigating 3 recently developed quadratures: Even-Odd (EO), Linear-Discontinuous Finite Element Surface Area (LDFESA), and the non-symmetric Icosahedral Quadrature (IC). We discuss implementation changes to 3d Sn codes (applied to Hybrid MOC-Sn TITAN and 3d parallel PENTRAN) that can be performed to accommodate Icosahedral Quadrature, as this quadrature is not 90-degree rotation invariant. In particular, as demonstrated using PENTRAN, the properties of Icosahedral Quadrature are well suited in the case of periodic boundary conditions.

In addition to implementing periodic BCs for 3d Sn PENTRAN, we implement a technique termed angular resweep which properly conditions periodic BCs for outer eigenvalue loop convergence. As demonstrated by two simple transport problems (3 group fixed source and 3 group reflected/periodic eigenvalue pin cell), we remark that all of the quadratures we investigated are generally superior to level symmetric quadrature, with Icosahedral Quadrature performing the most efficiently.

3.1 Background on Existing Quadratures

The use of level-symmetric quadrature has been a mainstay of 3d Sn transport, and naturally so, given the rotational invariance and integration qualities inherent in these quadratures. However, the past decade has offered a suite of new quadratures from researchers, including, Even-Odd (EO), Linear Discontinuous Finite Element - Surface Area (LDFESA), and Icosahedral (IC) quadratures [24, 32, 3]. Our research aim is purely on the applications of 3d quadratures on the sphere applied to existing discrete ordinates codes in search of enhanced accuracy and solution convergence. In this paper, we limit our focus to investigate the more recently introduced EO, LDFESA, and IC quadratures with comparison to level symmetric quadratures; these quadratures were actively implemented and investigated with 3d Cartesian parallel discrete ordinates PENTRAN [55] and hybrid MOC-Sn TITAN codes [63]. Furthermore, periodic boundary conditions were developed in 3d PENTRAN.

Common limitations of level symmetric quadrature include the fact that negative weights are generated beyond S_{20} (for a suitable reference please see [36]). In PENTRAN, an automatic selection of PnTn quadrature is performed for user requests of quadratures beyond S_{20} when level-symmetric has negative weights ($> S_{20}$) [38]. In this paper, we provide a background on the existing and recently implemented quadratures.

In the next section, we provide a brief discussion of quadratures; however, we limit the scope and offer salient points about each recent quadrature and consider results of the quadratures applied to a selected fixed source and eigenvalue transport problem, both using 3-group PWR type cross sections.

3.2 Level-Symmetric Quadrature and Pn-Tn

Level-symmetric (denoted as LS or LQ) quadrature is widely incorporated in lower order quadratures for 3d Sn, and it is often the primary quadrature discussed for discrete ordinates methods (see [24, 33, 58, 36, 12] for coverage of the method).

One type of level-symmetric quadrature exactly integrates polynomials defined by [58]:

$$\frac{2}{\pi}\int_{0}^{1} d\mu \int_{0}^{\pi/2} d\phi \mu^{k}\eta^{\ell} = \sum_{m=1}^{M} w_m \mu_m^k \eta_m^\ell = \begin{cases} \frac{1}{k+1} & \ell = 0 \\ \frac{1}{k+\ell+1} \prod_{\substack{i=1 \\ iodd}}^{\ell-1} \frac{i}{k+i} & \ell \geq 2 \end{cases} \tag{13}$$

where k and ℓ are even integers, μ and η are direction cosines, M is the number of octant points, and w refers to weight. When $\ell = 0$ and symmetry conditions are applied (a diagonal case) we have 'even moment' conditions:

$$\sum_{m=1}^{M} w_m \mu_m^k = \sum_{m=1}^{M} w_m \eta_m^k = \sum_{m=1}^{M} w_m \xi_m^k = \frac{1}{k+1},\ k \text{ even} \tag{14}$$

It is well known that the above prescriptions will yield negative weights beyond S_{20}, which are unphysical in the solution of the Sn Boltzmann equation. Additionally, "There is no need to include odd-moment conditions since these are automatically satisfied by level-symmetric quadrature sets when the integral is performed over 4π solid angles."[29].

In maintaining fully positive weights, the Legendre-Chebyshev (P_N-T_N) quadrature is not limited to S_{20}. P_N-T_N quadrature is also a form of product quadrature, in which ξ-levels on the z-axis are equal to the roots of the Gauss-Legendre quadrature, and the azimuthal angles are determined from roots of the orthogonal Chebyshev (T_N) polynomials of the first kind [63, 38]. P_N-T_N quadratures have been demonstrated to correctly preserve even moment conditions and can be locally refined [38].

3.3 Background on Recently Developed Quadratures

3.3.1 Even-Odd moment quadratures

Even-Odd (EO) moment quadratures[24] are a natural improvement on level-symmetric quadratures. Even-Odd Moment quadratures apply a set of moment conditions that allow for the preservation of solid angle integration for both the even and the odd moments. The technique for these quadratures is similar to level-symmetric, but instead of Equation 13 it is assumed that angular flux can be expanded by

$$\Psi\left(\overrightarrow{\Omega}\right) = \Psi(\mu, \eta, \xi) = \sum_i \sum_j \sum_k f_{ijk} \mu^i \eta^j \xi^k \tag{15}$$

where f is a coefficient, and i,j,k are exponents of the direction cosines; hence, the following moment condition is used:

$$\begin{aligned}
4\pi \sum_{m=1}^{M} w_m \mu_m^i \eta_m^i \xi_m^i &= \int_{4\pi} \mu^i \eta^j \xi^k d\Omega \\
&= \int_0^{\pi} \sin\theta d\theta \int_0^{2\pi} (\sin\theta\cos\varphi)^i (\sin\theta\sin\varphi)^j (\cos\theta)^k d\varphi \\
&= \left\{ \int_0^{\pi} (\sin\theta)^{i+j+1} (\cos\theta)^k d\theta \right\} \left\{ \int_0^{2\pi} (\cos\varphi)^i (\sin\varphi)^j d\varphi \right\}
\end{aligned} \tag{16}$$

where M is the number of ordinates in one octant, and θ & φ are polar and azimuthal angles, respectively [24]. This method effectively preserves both even and odd moments to satisfy exact integration of (N/2 + 1)th direction cosine moment conditions; Endo and Yamamoto demonstrated improvement of the EO quadratures over level-symmetric because of the new conditions with two benchmark problems, as well as integration of direction cosine moments [24]. Similar to level symmetric, negative weights appear at high quadrature (EO_{18}) so the quadrature is limited for Sn applications to span from EO_8 to EO_{16}. The EO_N notation is the familar Sn notation with N(N+2)/8 ordinates per octant.

One last interesting note on EO_N is the use of the recurrence relation for generation of moment conditions, which appears to be a unique feature of the quadrature that differentiates from LQ_N:

$$\mu^{i+2}\eta^j\xi^k + \mu^i\eta^{j+2}\xi^k + \mu^i\eta^j\xi^{k+2} = \mu^i\eta^j\xi^k \tag{17}$$

3.3.2 Linear Discontinuous Finite Element Surface Area (LDFESA) quadratures

The Linear Discontinuous Finite Element - Surface Area (LDFESA) quadrature by Jarrell and Adams [32] are a form of triangular quadrature constructed by considering equilateral triangles that are subject to the octahedral rotation group (by octant).

The conditions applied in solving for the weights are subject to the linear discontinuous finite element basis functions, and have been demonstrated to be fourth-order accurate, and that “All LDFE sets exactly integrate μ^2, η^2, and ξ^2 but fail to exactly integrate $\mu\eta$ cross-terms or any moments higher than μ^2 over a single octant ”[32]. Despite this accuracy issue, the quadrature has the advantage of local refinement and many-point generation (up to 524,288 points).

3.3.3 Icosahedral Quadratures

The Icosahedral (IC) quadratures, by Ahrens and Beylkin, have a Gaussian basis, and thus are more efficient in integrating spherical harmonics; they are designed to “ . . . exactly integrate subspaces of spherical harmonics from degree 5 to degree 210” that are rotationally invariant under the icosahedral group [2]. However, unlike the other quadratures, this quadrature is symmetric by 60 quaternion rotations and is not invariant under 90-degree rotation, as is the case for LS, EO, and LDFESA quadratures.

With regard to boundary conditions (BC), Ahrens previously noted that transport “in 3d Cartesian geometry with planar boundary conditions introduce planar symmetries that may be difficult to capture accurately with the icosahedral quadratures”[2]; this is a limitation, but we note that periodic BCs are a viable first step in incorporating icosahedral quadrature to relevant unit cell and assembly problems for reactor physics applications. We discuss the implementation later in Section 3. Periodic BCs have been performed before in 2d Sn transport [20], but we now test its application in 3d Sn.

Table 1 (adapted from [2]) gives a measure of the maximum degree subspace that can be integrated by Icosahedral quadrature versus that of level-symmetric quadrature (only matching the degree shown). It is clear that this quadrature, at least on a

Table 1: Icosahedral Maximum Degree and LS Maximum Degree.

Maximum Degree Subspace Integrated	Number of Points in Sphere with IC	Number of Points in Sphere with LS	LS Quadrature Name
5	12	24	S_4
9	32	80	S_8
19	132	480	S_{20}
23	192	672	S_{24}

Table 2: Quadrature Points, Boundary Conditions and Rotations.

Quadrature	Quadratures Available	BC Support	Octahedral Rotation
Level-Symmetric	S_4, S_6, S_8, S_{10}, S_{12}, S_{14}, S_{16}, S_{18}, S_{20}	Reflective & Periodic	Y
P_N-T_N	S_4, S_6, S_8, S_{10} $\ldots$	Reflective & Periodic	Y
Even-Odd	EO_8, EO_{10}, EO_{12}, EO_{14}, EO_{16}	Reflective & Periodic	Y
LDFE-SA	4,16,64,256,1024, 4096,16384,65536	Reflective & Periodic	Y
Icosahedral (IC)	72,192,312,432, 672,912,3432	Periodic	N

per ordinate numerical integration basis, is more efficient than LS. We can also informally argue that IC quadrature, by design of exact subspace integration of spherical harmonics of degree N, are more efficient than the other quadratures presented (LS, P_N-T_N, EO, LDFESA).

The features of the 3 quadratures (along with LS and P_N-T_N) as it relates to practical Sn implementation given in Table 2.

In the next section, we focus on the IC quadrature, since more changes to 3d Sn codes are required to adapt the quadrature. Indeed, IC quadrature is not 90-degree rotation invariant (but symmetric to Icosahedral rotations), and many traditional

Sn codes require adaptation. The EO quadrature and LDFESA quadrature do not require any significant changes to existing Sn codes, other than ensuring that the number of ordinates per octant is changed accordingly.

3.4 Implementing Icosahedral Quadratures in 3d transport

3.4.1 Supporting Icosahedral Quadrature with 3d Parallel Sn PENTRAN and TITAN

Discrete ordinates (Sn) codes are adaptable to support quadratures defined on the entire sphere, and should not be limited to a per-octant basis, especially if there are efficiency gains to be made with more efficient quadrature sets. We take on the challenge to adapt Sn codes to be more state-of-the-art than to assume limitations of the quadrature. IC quadratures, as we will demonstrate, are more accurate on a per-ordinate basis than other quadratures, and should be used when applicable. We explain our adaptation as performed in the PENTRAN and TITAN parallel Sn codes.

Normally, 90-degree rotation invariant quadratures only require a single octant of quadrature information to be read. Icosahedral quadratures are rotationally symmetric through 60 rotations but not 90-degree rotation invariant, and thus information on the entire sphere is read in for direction cosines and corresponding weights.

One of the preferences, with parallel discrete ordinates and related message passing, is that the ordinates per octant are equal. Fortunately, there is some patterning of the Icosahedral quadrature such that points can be subdivided equally to 8 octants. This was the case for the 72, 192, 312, 432, 672, 912, and 3432 point IC quadratures. We have ignored other point sets that do not divide evenly by 8; this is not a limitation, but a preference for an even distribution of points to each octant that is easily adapted by full-scale angular decomposition in our parallel Sn codes. Parallel angular decomposition remains unaffected by communication losses if the quadrature gives an even number of points per octant. We leave the cases for uneven octant point distribution for parallel Sn for future work.

We also remark that the IC quadratures do have direction cosines on the plane, where traditionally, quadrature points are exclusively defined within the octant interior (e.g. no planar or on axis direction cosines). For example, the 192-point IC quadrature has 12 direction cosines that lie on the coordinate planes. We assign greater than conditions to appropriate planes such that the points favor the octant having fewer points. This treatment successfully places the proper number of points per octant equally. For example, the 192-point quadrature has four octants containing 21 points and four octants containing 24 points. Four triplets of points are banked to favor the octants having 21 points, thus preserving an equal number of points per octant. We have found that including planar direction cosines is fairly trivial, and has no impact on any finite differencing transport operators. The original quadrature points are re-sorted by octant, with planar point banking favoring even octant distribution. This is prepared outside of the PENTRAN code.

Another potential issue for implementing IC quadratures into discrete ordinates codes is the fact that quadrature weights are usually a function of an ordinate in an octant $w_i\left(\Omega_i\right) = w\left(|\mu|, |\eta|, |\xi|\right)$, where i is an ordinate in the octant (same for all octants) . This is normally the treatment by Sn codes PENTRAN and TITAN, where appropriate sign changes are made trivially. With the IC quadrature, we add another variable to support weight such that $w_{i,j}\left(\Omega_i, \text{octant } \#\right) = w(\mu, \eta, \xi)$, where i is a local ordinate of the octant, and j is 1 of 8 octants. These changes are propagated throughout the code, as well as the fact that $\mu_{i,j} = \mu(i, j)$, $\eta_{i,j} = \eta(i, j)$, $\xi_{i,j} = \xi(i, j)$; thus, direction cosines are treated as 2d arrays, with the same i (local ordinate in octant) and j (octant) notation.

With proper message passing in parallel programming, the usage of quadratures other than octahedral rotation groups can be implemented with no extra overhead to memory. We note that the use of non-octahedral rotation quadratures has been performed before [61], and that the properties of the quadrature should not be seen

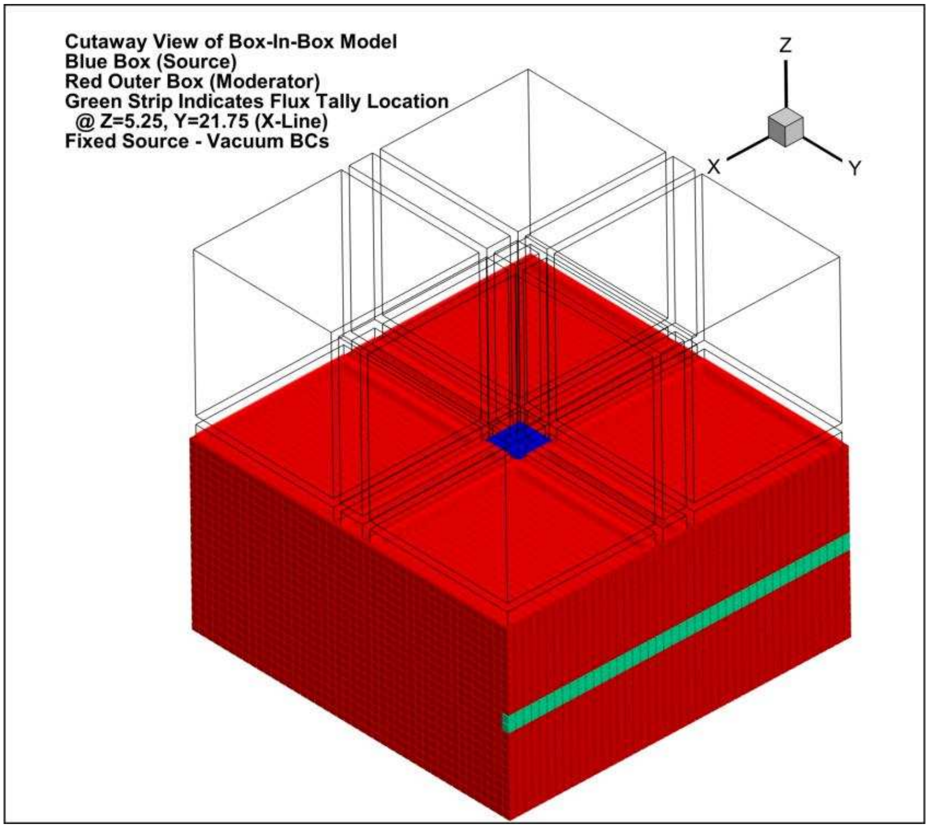
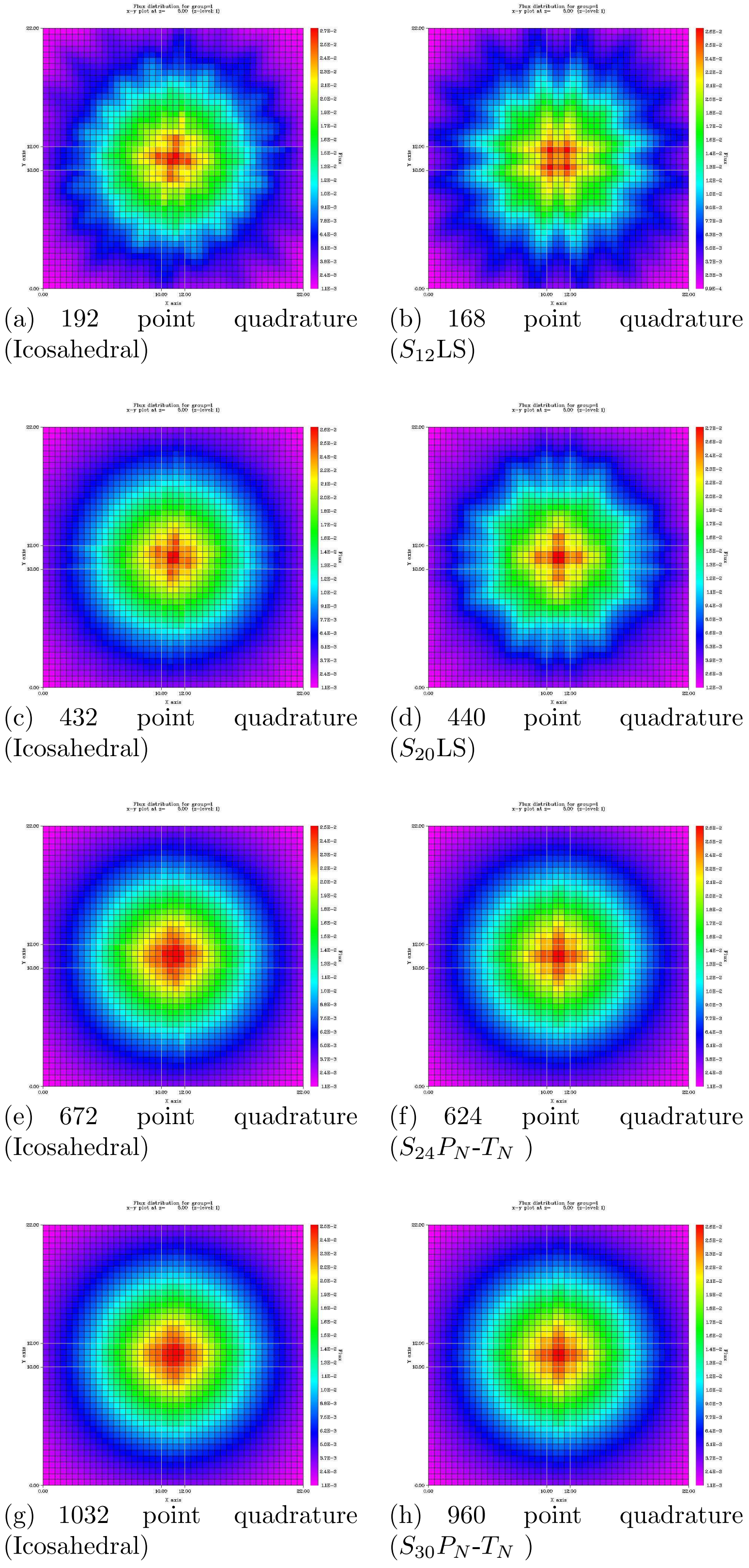

Cutaway View of Box-In-Box Model
Blue Box (Source)
Red Outer Box (Moderator)
Green Strip Indicates Flux Tally Location
@ Z=5.25, Y=21.75 (X-Line)
Fixed Source - Vacuum BCs
Z
X
Y

Figure 14: Box-in-box problem with all vacuum boundaries. Subsequent tallies in Figure 15 consider Group 1 with Z=5.25cm Y=21.75cm (Green Tally Line)

as a limitation, but seen as state-of-the-art. In the next section, we focus on validating the IC quadrature with a simple box-in-a-box problem. Obviously, vacuum boundary conditions are treated trivially with any of the new quadratures. We save the discussion for non-vacuum boundary conditions in Section 3.5.2.

3.4.2 3d Box-in-a-box Fixed Source Problem to Validate Implementation of IC Quadrature

A 3d box-in-a-box problem is prescribed for initial testing of quadrature problems. Cross sections for the test problem are given in the appendix. The inner source box dimensions are 2cm×2cm×2cm, and the outer moderator dimensions are 22cm×22cm×22cm, with PWR-based 3 group cross sections. In particular, ray effects are known to be strongly present in group 1, so we focus on that group in subsequent figures.



(a) 192 point quadrature (Icosahedral)

(b) 168 point quadrature (S_{12}LS)

(c) 432 point quadrature (Icosahedral)

(d) 440 point quadrature (S_{20}LS)

(e) 672 point quadrature (Icosahedral)

(f) 624 point quadrature ($S_{24}P_N$-T_N)

(g) 1032 point quadrature (Icosahedral)

(h) 960 point quadrature ($S_{30}P_N$-T_N)

Figure 15: 2d Flux Plots for Box-in-box Fixed Source Problem at z=5cm

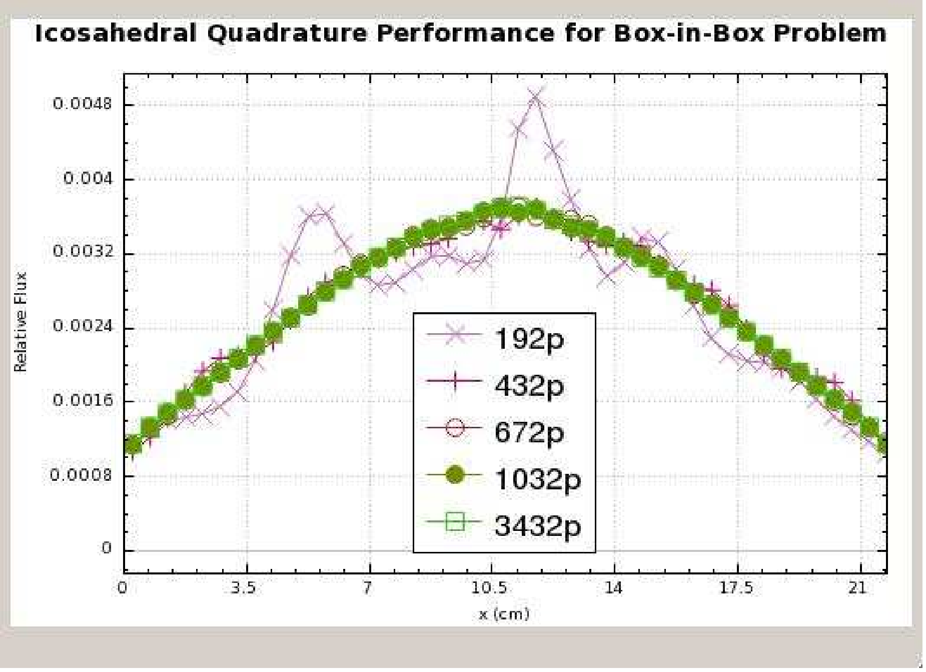


Figure 16: Relative Flux Line Plot with Icosahedral Quadrature

Figure 15 with the left column using Icosahedral quadrature and the right column using level symmetric or P_N-T_N quadrature show 2D flux plots for Z=5.00 cm for Group 1. It should be noted that the closest quadrature point match is given comparing IC and LS. From a qualitative viewpoint, it is clear that the Icosahedral quadrature has a rotating ray effect versus the ray effect exhibited by level-symmetric, seen at the lower quadrature settings. Figure 15(c) shows a remarkable reduction of ray effects using 432 points on the unit sphere, nearly equivalent in the number of directions to LS S20 quadrature, but, upon comparing Figures 15(c) and 15(d), remarkably more physical for similar computational effort.

In order to more properly quantify ray effects beyond a simple density plot inspection, we plot the top edge fluxes derived from Figure. Figure 15 with subfigure (a), (c), (e), and (g) (in addition to 3432 point IC quadrature) are considered with the line flux plot shown in Figure 16. Figure 15 with subfigure (b), (d), (f), and (h) (in addition to $S_{60}P_N$-T_N quadrature) are shown with associated line fluxes in Figure 17. It is clear that ray effects are mitigated more rapidly with increasing quadrature

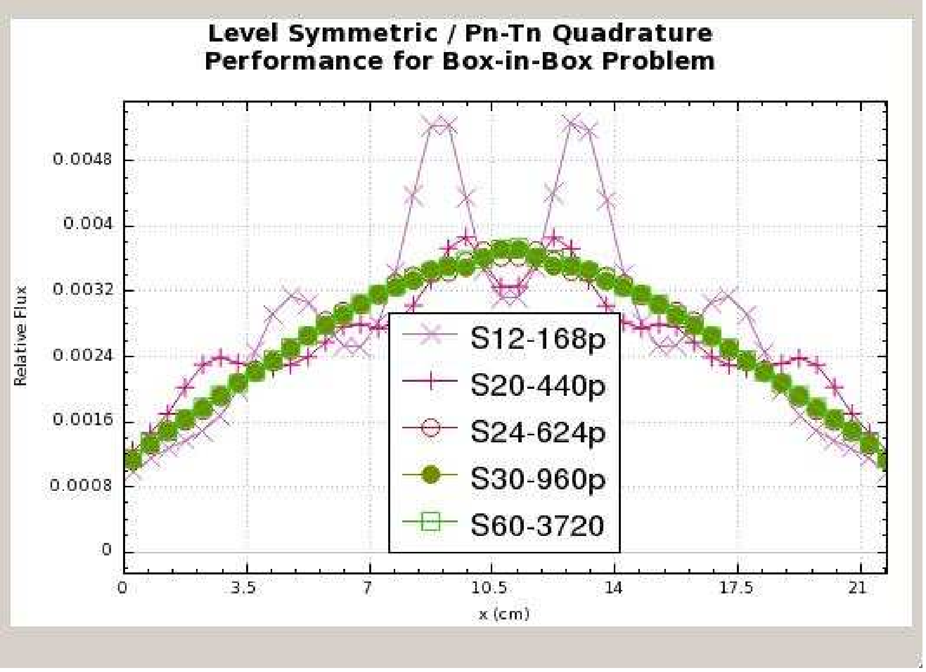


Figure 17: Relative Flux Line Plot with Level Symmetric / (P_N-T_N) Quadrature

using IC quadrature sets with 420 points on the sphere (validating the qualitative plot of Figure 4).

3.5 Adapting Periodic Boundary Condition Support For 3d Sn

It is apparent that many analysis capabilities are lost without adding in a form of non-vacuum boundary conditions to IC quadratures. We have addressed this issue by adding periodic BC support for IC quadratures. We will first depict the results of the Westinghouse PWR 3 group pin problem, and then discuss the implementation of angular resweep on its effect related to periodic and reflective BCs.

3.5.1 PWR Pin / Unit Cell Problem (3d 3 group Eigenvalue Problem with Reflective/Periodic BCs)

In order to move to problems more relatable to reactor physics, we examine a simple eigenvalue PWR pin problem with reflective/periodic BCs using PENTRAN.

For all quadratures, tolerances were set to 10^{-4} and 10^{-6} for inner/ flux tolerance and outer/eigenvalue flux tolerance, respectively. A reference Multigroup MCNP5 calculation provides $k_{\text{eff}} = 1.32384 \pm 0.00038$. All quadratures were with the same tolerances, to explore impacts related purely to quadrature selection. The EDW differencing scheme was used for all pin problems [56].

For 4 quadrature types, we investigate performance at lower orders (roughly between S_8 and S_{16}). Figure 18 provides a snapshot of quadrature performance. Each row corresponds to a specific quadrature, which increases in quadrature points (order) from left to right. Also, each row maintains consistent color scaling in plotting flux. Qualitatively, LS quadrature is shown with 80, 168, and 288 points (corresponding to LS S_8, S_{12}, and S_{16}); edge flux bowing is still present with the 288 point quadrature. EO quadrature (row 2) is shown with the same 80, 168, and 288 point structure. Here, we see that edge flux bowing is strongly mitigated at the 288 point case. The closest equivalent quadratures were chosen with LDFESA found with the 128 and 512 point quadratures; edge flux bowing is also strongly mitigated.

We can move from a qualitative analysis to a quantitative one by examining the edge flux plots (same for any of the sides) in Figure 19. The reference flux (in black) shows that the behavior of the flux is generally flat, with a broad central plateau peak. For characterization of error, we assume the IC-3432 point result as the reference.

To quantify the ray effects in this problem, we calculate the maximum percentage error to the reference value in Figure 20. The error is calculated to be the relative difference between the computed flux from the reference computed flux (using IC-3432 quadratures as the reference - black square plot points).

Figure 20, ranking in order by smallest max percent error from left to right, clearly indicates that IC quadrature is very accurate and minimizes the maximum observed errors to roughly 1 percent using a 192 point quadrature. We note that on the lower orders, EO quadrature does not exceed LS performance (where improvements are noted with EO_{16}).

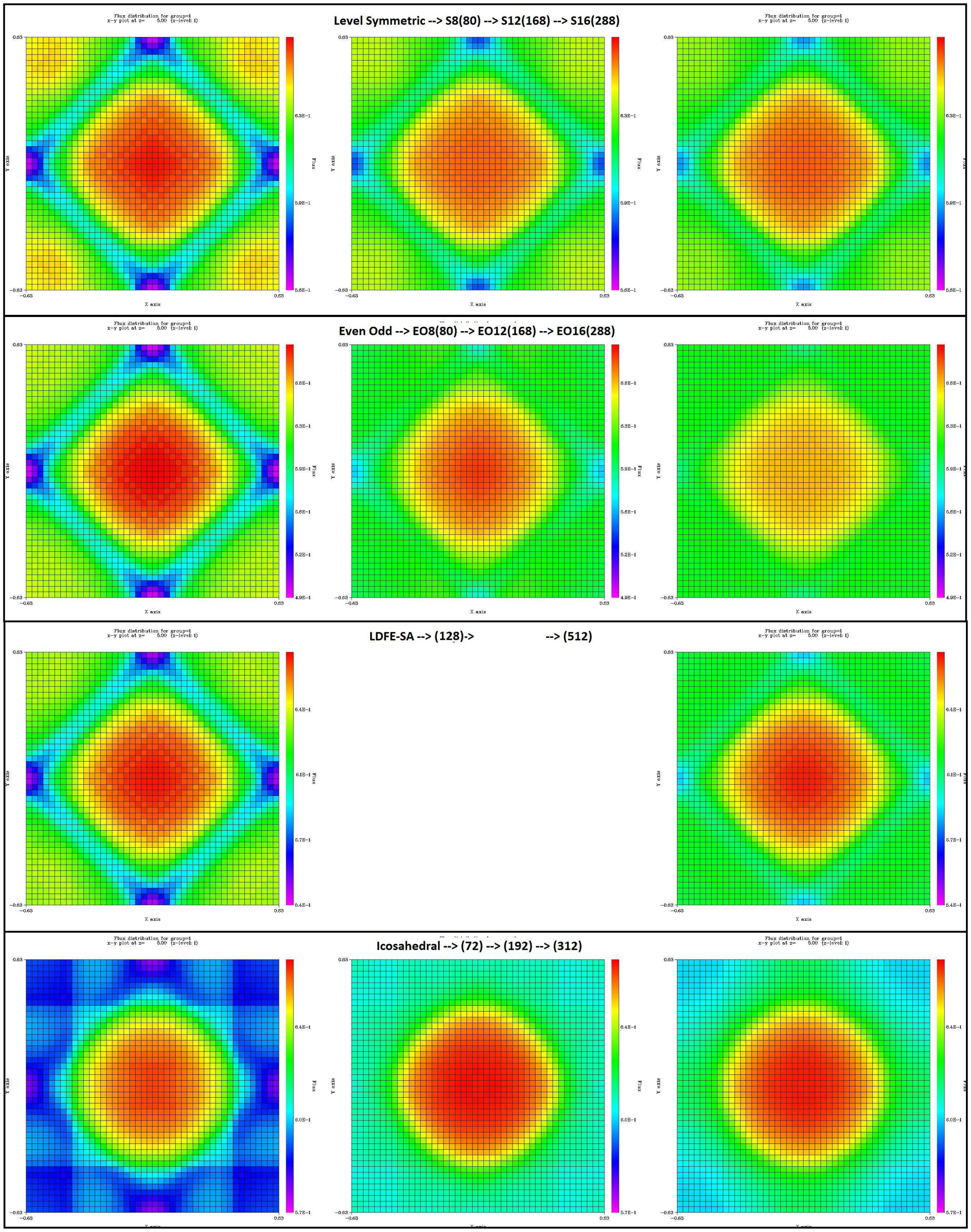


Figure 18: Westinghouse PWR Pin Flux for Various Quadratures: Row 1. LS. Row 2 - EO, Row 3 - LDFE-SA, Row 4 - IC (Increasing Quadrature Points from Left to Right)

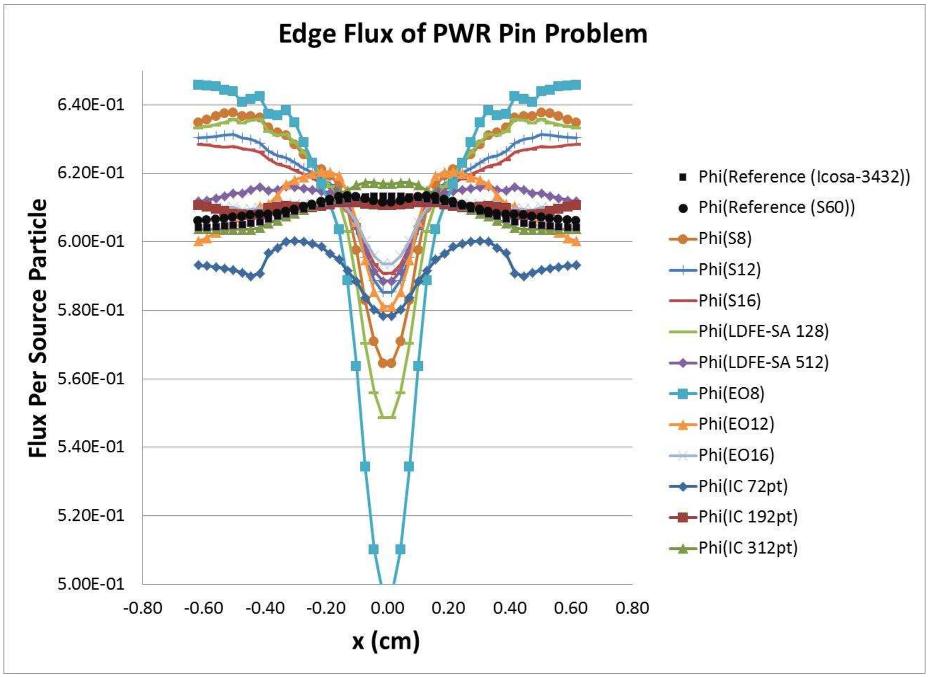


Figure 19: Edge Flux of PWR Unit Cell for Various Quadratures

Another way to visualize Figure 20 is by plotting the error against the number of quadrature points (shown for LS, EO, and IC), shown in Figure 21. It is clear that the IC quadrature improves the accuracy for the same transport parameters when compared to LS and EO quadratures, having a similar number of points.

3.5.2 Angular Resweep Loop in Parallel Sn

Reflective boundary conditions improve coupling between angular sweeps, as information is naturally traded from one octant to another (in four pairs). Because of this, angular flux information is already properly conditioned in parallel. PENTRAN performs angular decomposition by each octant (in multiples of two as 2, 4, and 8), so there is a natural trade of counter current flows of information between inner iterations. Table 3 gives a listing of octants (with octant ordering as designated in PENTRAN).

Since periodic boundary conditions effectively pass information into the same sweep direction, coupling is poor because the angular flux is passed to the same sweep

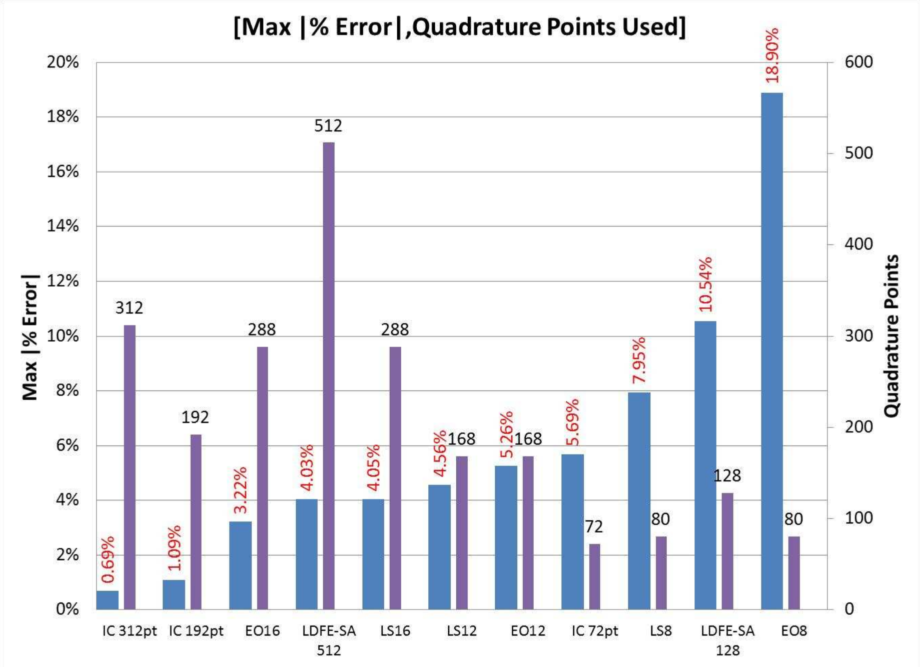


Figure 20: Max |% Error| and Quadrature Points for Various Quadratures. (Two Columns for Each Quadrature - Max % Error in Blue, Number of Quadrature Points in Purple)

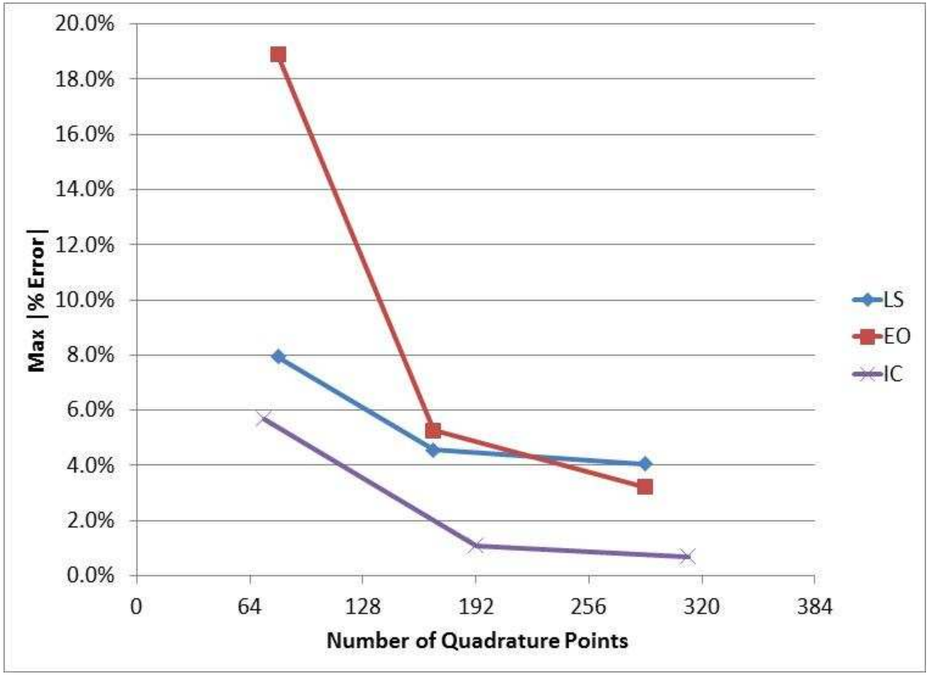


Figure 21: Max |% Error| vs. Quadrature Points for LS, EO, and IC Quadratures for 3 group PWR Eigenvalue Problem

Table 3: PENTRAN Octant Sweep Ordering

Sweep	μ	η	ξ
1	$-$	$-$	$-$
2	$+$	$+$	$+$
3	$-$	$-$	$+$
4	$+$	$+$	$-$
5	$+$	$-$	$-$
6	$-$	$+$	$+$
7	$+$	$-$	$+$
8	$-$	$+$	$-$

(and not to another specular octant partner). In order to improve the coupling, the same finite difference sweep is repeated N times, as designated by the user. This is denoted as angular resweep. We remark that the loop is performed only on the outside of the finite difference sweep and boundary angular flux transfers. Hence, the cost of a resweep action is actually far less than a full inner iteration in the parallel algorithm. The quantification of angular resweep loop in the context of inner iterations remains a future investigation.

It is reported here that even without resweep actions, all eigenvalues are still conditionally convergent, but with inherent oscillation effects (seen in the case of IC and LS quadratures employing periodic BCs in Figures 22 and 23). Resweep actions mitigate this behavior. Another potential reason for the oscillatory behavior is driven because of de-coupling effects in parallel octant, as all cases used a separate MPI processor/task for each octant.

3.5.3 Impact of Angular Resweep Loop in PWR Pin Problem

We tested the impact of angular resweep for 3 cases: Icosahedral quadrature (72 point) with periodic BCs, and LS (S_8) with periodic BCs and then also LS (S_8) with reflective BCs. Also, the pin problem was run using full angular octant decomposition, requiring 10 inner iterations per outer. Our notation for angular resweeps (AR) is

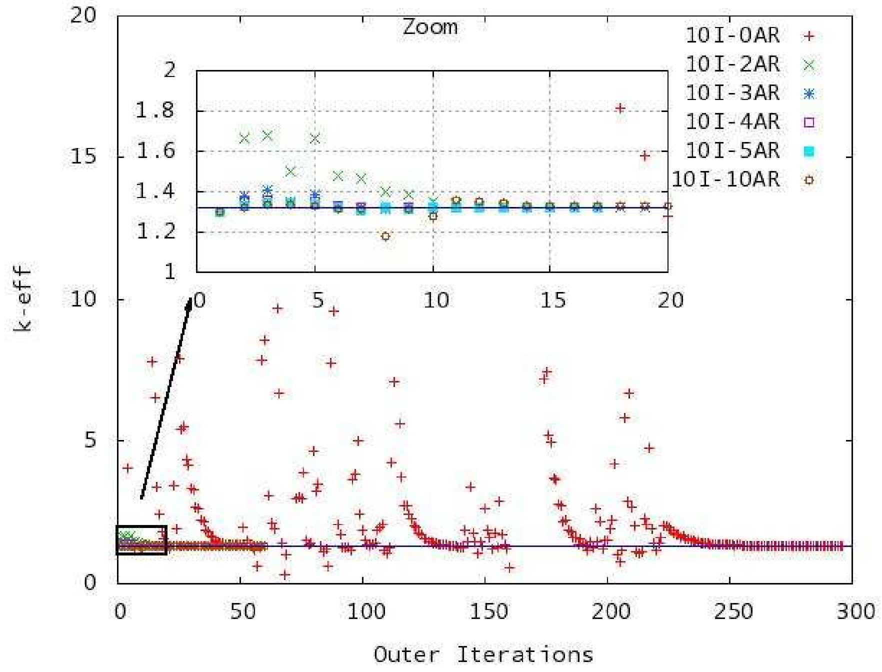


Figure 22: k_{eff} convergence for IC quadrature (10 Inner) with various resweeps using Periodic BCs (PWR Pin)

simply a multiplier. For example, 0AR means that no angular resweeps are performed, and 10AR means that a total of 10 angular resweeps are performed per inner iteration. Hence, 0AR and 1AR are equivalent. Obviously, the cost of an inner iteration goes up with an increased number of angular resweeps.

Figures 22 and 23 show the realized benefit of using angular resweep, especially that even a single use of 2 angular resweeps has the immediate benefit of reducing oscillations from decoupled periodic BCs. The oscillatory effect appears to only be present with periodic BCs and with MPI angular parallel decomposition.

As an additional verification, we checked the performance of level-symmetric using reflective BCs with angular resweeping. Figure 24 (LS quadrature) shows the impact of angular resweep on reflective boundary conditions. It is clear that there is a damping of oscillations, and overall reduction in outer loop iterations. Again, as the test problem is a simple pin, further examination is needed to study the impact of angular resweep on larger problems (assembly and full-core), although it appears that

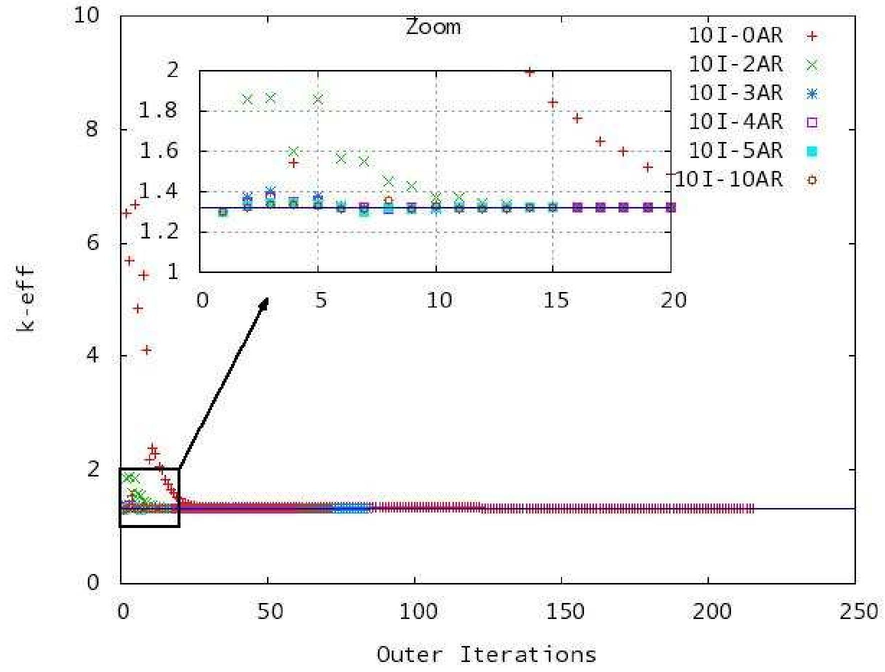


Figure 23: k_{eff} convergence for LS quadrature (10 Inner) with various resweeps using Periodic BCs (PWR Pin)

there is benefit to at least 2 angular resweeps.

3.6 Quadrature Conclusions: Benefits to Efficiency and Accuracy

All new quadratures (EO, LDFESA, and IC) investigated are worthy to implement in 3d discrete ordinates codes, and are generally superior in both qualitative and quantitative results versus that of level-symmetric (LS) sets. In particular, the Icosahedral quadrature (IC) is shown to be extremely efficient and mitigates ray effects more efficiently than the traditional LS sets, for the given spatial mesh and test problems shown.

For the box-in-a-box vacuum BC problem, we validated the successful implementation of the IC quadrature as compared to level-symmetric. We have shown that implementation of IC quadrature is achieved with 3d Sn parallel PENTRAN and 3d hybrid MOC-Sn TITAN codes.

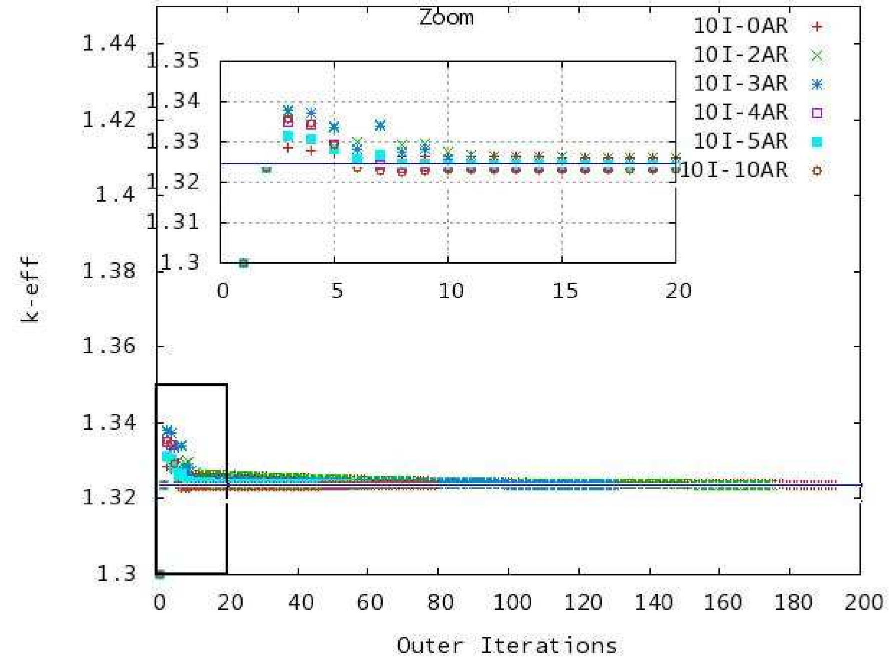


Figure 24: k_{eff} convergence for LS quadrature (10 Inner) with various resweeps using Reflective BCs PWR Pin)

In the case of reflective/albedo boundary conditions, EO and LDFESA quadratures should be favored, as the IC quadratures do not have reflective/albedo boundary condition support. When many ordinate points or local refinement is needed, the use of LDFESA quadrature is recommended.

In the eigenvalue PWR pin problem, the IC quadrature was the most efficient quadrature and should be the standard quadrature for problems that apply vacuum BCs or periodic BCs. In particular, when using periodic BCs, the termed angular resweep (AR) technique should be incorporated to minimize eigenvalue oscillations and improve outer loop convergence.

CHAPTER IV

MULTIGROUP CROSS SECTION (ENERGY) OPTIMIZATION: FAST AND THERMAL

Collapsing cross sections from a master library to a broad group working library for fast reactors, compared to thermal reactors, is more sensitive to both the selection of the multigroup energy bounds as well as the number of groups selected. This plays a particularly important role in eigenvalue problems, where significant changes in the neutron population can result from small changes in cross section [9]. Fine group structures provide a high level of precision and accuracy, but at the cost of increased computational time and effort. It is desirable to collapse the larger master group library to a smaller group bin selection while still maintaining as much fidelity as possible relative to the master group library. Selecting a proper group collapsing strategy is also vital for deterministic transport codes, which rely on multigroup cross sections to prevent systemic errors. Although transport codes globally solve for fluxes precisely across the entire phase space (and the corresponding convergence in the source and the eigenvalue), the results may not be accurate if they are based on non-representative cross sections.

In this chapter, we attempt to address the issue of selecting both a precise and accurate collapsed group structure for a typical fast reactor MOX fuel pin and also for a high enrichment PWR Pin (thermal); we will begin with the general purpose ENDF/BV-II [14] 238 group cross section library available in SCALE6.1. Several codes were used in this analysis, including the 3d deterministic Cartesian parallel transport code PENTRAN [55], the 2d deterministic transport code SCALE6.1 [10], continuous and multigroup MCNP5, a new geometry converter MESH2MCNP[41]

, and the multigroup macroscopic cross section collapsing code YGROUP [62]. In addition, a number of smaller codes and UNIX scripts were used for data formatting across different codes.

The work performed to determine the 10 group cross sections for a fast reactor MOX pin largely derived from the co-authored paper referenced here [17]. The work performed to determine an optimal 3 group structure for the thermal highly enriched PWR pin is newly developed in this chapter.

In the next section, we present a basic theory behind the YGROUP cross section collapsing methodology.

4.1 Theory in Application of Adjoint Fission Sources for Collapsing and Weighting Multigroup Cross Sections

For an eigenvalue problem, the transport equation can be represented by Equation 18, where L is the forward transport operator shown in 19, F is the fission source defined in Equation 20, and ψ is the angular flux $\psi(\hat{\mathbf{r}}, E', \Omega)$.

$$L\psi = \frac{1}{k}F \tag{18}$$

$$L = \hat{\mathbf{\Omega}} \cdot \nabla + \sigma_t(\mathbf{r}, E) - \int_{4\pi} d\Omega \int_0^\infty dE' \sigma_s(\mathbf{r}, E' \to E, \hat{\mathbf{\Omega}} \to \hat{\mathbf{\Omega}}') \tag{19}$$

$$F = \chi(E) \int_{4\pi} d\Omega \int_0^\infty dE' \bar{\nu}\sigma_f(\mathbf{r}, E')\psi(\mathbf{r}, E', \Omega) \tag{20}$$

Note that $\chi(E)dE$ is the average number of fission neutrons emitted with energy from E to $E + dE$ per fission neutron, and is normalized such that $\int_0^\infty \chi(E)dE = 1$. Equations 18 through 20 can be expressed in multigroup and discretized space/angle form. Spatial dependence of the cross sections is implicitly assumed, and using standard energy group (g) notation we obtain Equations 21 to 23.

$$L_g \psi_{i,g} = \frac{1}{k} F_g \tag{21}$$

$$L_g = \hat{\mathbf{\Omega}} \cdot \nabla + \sigma_g(\mathbf{r}) - \sum_{g'=1}^{G} \int_{4\pi} d\Omega' \sigma_{s,g'\to g}(\mathbf{r}, \hat{\mathbf{\Omega}} \to \hat{\mathbf{\Omega}}') \tag{22}$$

$$F_g = \chi_g \int_{4\pi} d\Omega' \sum_{g'=1}^{G} \bar{\nu_{g'}} \sigma_{f,g'} \psi_g(\mathbf{r}, \hat{\mathbf{\Omega}}) \tag{23}$$

Similarly, the adjoint transport operator, in multigroup form, is shown in Equations 24 through 26. Note that the directional vector of neutrons is reversed in the $L_g^\dagger$ adjoint operator, and the scattering component considers neutrons emanating back from from group g up to g'.

$$L_g^\dagger \psi_{i,g}^\dagger = \frac{1}{k} F_g^\dagger \tag{24}$$

$$L_g^\dagger = -\hat{\mathbf{\Omega}} \cdot \nabla + \sigma_g(\mathbf{r}) - \sum_{g'=1}^{G} \int_{4\pi} d\Omega' \sigma_{s,g\to g'}(\mathbf{r}, \hat{\mathbf{\Omega}} \to \hat{\mathbf{\Omega}}') \tag{25}$$

$$F_g^\dagger = \bar{\nu_g} \sigma_{f,g} \int_{4\pi} d\Omega' \sum_{g'=1}^{G} \chi_{g'} \psi_{g'}^\dagger(\mathbf{r}, \hat{\mathbf{\Omega}}) \tag{26}$$

The reciprocity relation for adjoint and forward transport is shown in Equation 27, where <> denotes integration over all independent variables.

$$< \psi_g^\dagger L_g \psi_g > = < \psi_g L_g^\dagger \psi_g^\dagger > \tag{27}$$

In general applications, a forward neutron fixed source of magnitude (q_g) and adjoint importance relative to a detector response cross section $(\sigma_d)_g$ can be determined by independently satisfying the equations, respectively:

$$L\psi_g = q_g \qquad \text{and} \qquad L_g^\dagger \psi_g^\dagger = (\sigma_d)_g \tag{28}$$

Applying 27 and 28 and integrating over all variables results in the useful expression for the reaction rate R:

$$R =< \psi_g (\sigma_d)_g >=< \psi_g^\dagger q_g > \tag{29}$$

We wish to leverage the importance relative to a specific reaction, namely fission, to establish a spectral weighting function. The importance of a neutron to cause fission can be represented by using the fission cross section as the adjoint source:

$$L_g^\dagger \psi_g^\dagger = \sigma_{f,g} \tag{30}$$

Equation 30 will yield a spectral adjoint importance aliased to causing a fission event in the system under consideration.

For our purposes, the forward flux and fission adjoint importance function zeroth angular moments can then be imported into the YGROUP code to determine collapsing and weighting of group structures. Once the group structure has been determined via YGROUP, a cross-comparative analysis of eigenvalue k_∞ (instead of a fission-based fixed source) transport methodologies, with evaluations that include deterministic transport and Monte Carlo based simulations, can be performed to evaluate the accuracy of the group structure.

4.1.1 Overview of YGROUP Cross Section Collapsing

In general, broad group collapsing consists of two steps: (i) the determination of the broad group structure (selection of energy group bounds), and (ii) the subsequent evaluation of broad group macroscopic cross section data. The determination of the broad group structure can be evaluated in several different ways. The YGROUP code offers a collapsing strategy that includes a flat weighted energy spectrum (uniformly dividing the fine group library into broad groups, or weighting the energy spectrum

Table 4: Collapsing Options for Determination of Broad Group Bin Structure in YGROUP

Collapsing Option	Collapsing Strategy	Collapsing Function
1	Flat Collapsing	$C_g = 1$
2	Reaction Rate Collapsing	$C_g = \sum_i V_i \psi_0^g \sigma_t^g$
3	Contributon Collapsing	$\sum_i V_i \sum_\ell^L \sum_{m=-\ell}^{\ell} = \phi_{\ell m}^g (\phi^*)_{\ell m}^g$

Table 5: Weighting Options for Broad Group Cross Sections in YGROUP

Weighting Option	Weighting Strategy	Weighting Function
1	Flat Weighting	$W_g = 1$
2	Flux Weighting	$W_g = \phi_g$
3	Contributon Weighting	$W_g = C_g$
4	Biased Importance Weighting	$W_g = C_g \phi_g$

based on the reaction rate, or applying a contributon strategy (weighting with flux coupled with the adjoint importance function as noted in Table 4) [5]. YGROUP is also capable of having user-specified "reserved" group boundaries [62]. A summary of the collapsing options provided by YGROUP is provided in Table 4.

In Table 4, V_i is the fine mesh per-material volume, ϕ_0^g is the zeroth moment flux for group g, σ_t^g is the total cross section for group g, and $\phi_{\ell m}^g$ and $(\phi^*)_{\ell m}^g$ are the forward and adjoint fine group flux moments, respectively, for a user-specified L Legendre cross section order. For the evaluation of broad group cross section data, YGROUP is capable of weighting the broad group cross sections by flat weighting, flux weighting, contributon weighting, or biased importance weighting [62]. A summary of the cross section data weighting options provided by YGROUP is listed in Table 5.

4.2 Methodology for Fast Reactor MOX Pin

In fast reactors, the group-dependent spectrum is very sensitive due to resonances in the intermediate-energy range from 1-10 keV that are present for both capture and

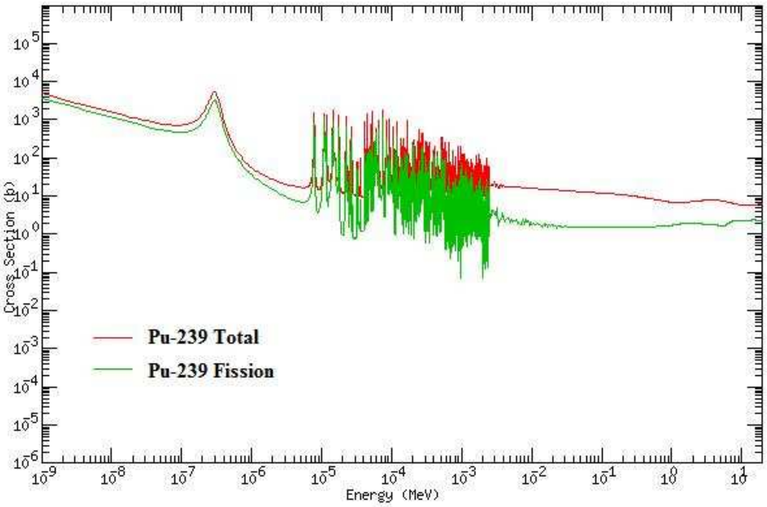


Figure 25: Fission and Total Cross Section for Pu-239 [34]

fission cross sections. This is evidenced by a plot of the pointwise fission cross sections for Pu-239 as given in Figure 25. Additionally, the unresolved and p-wave resonances that exist at high neutron energies can significantly affect the energy spectrum [9]. Since the process of collapsing cross sections involves making the group structure more granular, the contribution of resonances over a broader energy range can subtly impact the group collapsed cross sections. Perhaps more importantly, the influence of fission neutron generation ($\nu\sigma_f$) is largest in the highest energy groups for fast reactors, which if incorrectly collapsed, can average the larger neutron fission contribution across broader energies, overestimating the contribution from fission. This can lead to systemic errors which may not be immediately apparent if an "incorrect" group structure is used. The k-eigenvalue was chosen as a convenient metric for determining the performance of a particular group structure, as it is sensitive to small changes and is easy to localize the effects of a particular perturbation (in this case, the group structure and the corresponding collapsing scheme used to derive them).

The fuel composition selected for a representative fast reactor was a single, central axial segment of a mixed-oxide (MOX) sodium cooled fast reactor fuel pin composed of dual phase 21% $PuO_2 - UO_2$ with liquid sodium coolant, a helium-4 gas gap,

Table 6: Plutonium Isotopics for the MOX Fuel Pin

Nuclide	Atom Percent (%)
Pu-238	0.08
Pu-239	70.32
Pu-240	24.34
Pu-241	4.14
Pu-242	1.12

and stainless steel cladding arranged in a hexagonal lattice. The 21% refers to atom percent composition of standard reactor grade plutonium consisting primarily of Pu-239, with the remainder being Pu-240, Pu-241, Pu-242, and trace amounts of Pu-238 which sum to unity; these are referenced in Table 6. The oxygen and uranium isotopics are assumed to be natural composition. A literature study for fuels of this type, with additional analysis, was used to calculate the various thermodynamic properties associated with the fuel, gap, clad, and coolant. The boundary conditions imposed on this geometry were reflected on all sides.

Fuel pin cross sections were generated in SCALE6.1 using the T-NEWT sequence using a 238 group structure. These 238 group cross sections formed the reference multigroup library for further YGROUP and SCALE group structure analysis. The converged eigenvalues from the T-NEWT sequence were compared with the PENTRAN and MCNP eigenvalues for the 238 group case. Previous fast reactor simulation studies by Hazama et al. show that a broad 70 group collapse is at most 200 pcm from the experimental reference k_∞ values, and a 175 group collapse is within 50 pcm of reference k_∞ values [28]. One objective of this study and report in this paper is to achieve accuracy levels similar to Hazama's results suitable for a fast reactor, but collapsing the 238 group ENDF/B-VII libraries to far fewer groups (e.g. < 24) for accurate fast reactor working libraries using the YGROUP code in various tunable options.

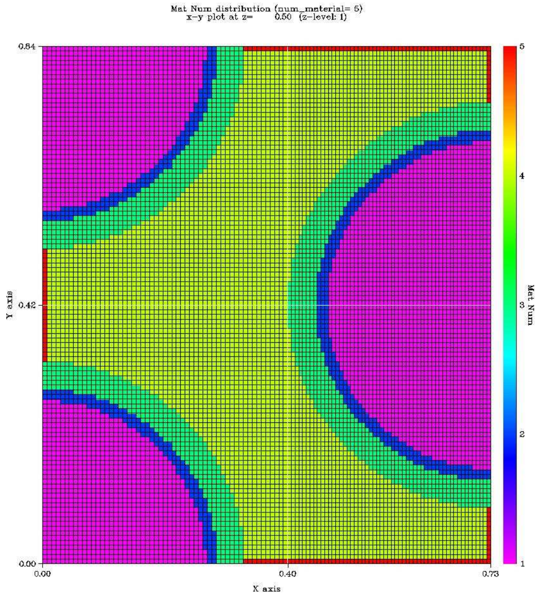

Mat Num distribution (num_material= 5)
x-y plot at z= 0.50 (z-level: 1)
0.84
Y axis
0.42
0.00
0.00
0.40
0.73
X axis
5
4
3
2
1
Mat Num

Figure 26: XY PENTRAN Slice of the MOX Fuel Pin. Colors represent different materials: Red is the adjoint fission source (in coolant), yellow is coolant, green is cladding, blue is gap, and purple is fuel

To accurately represent hexagonal pitch geometry in Cartesian coordinates, the fuel pin was split into three parts separated by the pin pitch, which in aggregate represents a single fuel pin mass. When the fuel pin was voxelized into a fine mesh format, the fuel mass was within 2% of the exact combinatorial geometry mass target. For adjoint computations (as noted in Equation 30), an adjoint fission source was placed on the coolant boundaries of the problem (boundary fine meshes that were not fuel, gap, or clad); when an adjoint transport fixed source simulation is performed, these fine meshes at the edge will stream adjoint "particles" that represent the importance to cause fission across the entire pin using Eq. 24 through 30. In effect, this gives the contribution of neutrons that will fission outside the local pin region. The geometry for adjoint computations described above is shown in Figure 26.

An equivalent discretized geometry for MCNP5 was created using the MESH2MCNP Python code [41]. This code instantly and precisely replicates the fine mesh structure used in the PENTRAN discretized model as voxelized cells and surfaces within the

MCNP code, so that both models have the same geometric basis. Continuous energy MCNP and multigroup MCNP cross sections with the discretized geometry were used to compare eigenvalue calculations. A particular distinction will be made between group-collapsed cross sections derived from the YGROUP code and group-collapsed cross sections rendered from SCALE alone.

Since YGROUP has several options for determining broad group structure and cross-section data evaluation for that group structure, it is not *a priori* evident which method will provide the "optimal" group structure. Various combinations of the options highlighted in Table 4 and Table 5 were used and are shown in Section 4.3 below.

Fast reactors will have a much greater importance among higher energy group ranges since thermalization is not necessary for fission (hence the lowest energy groups should have a comparatively minimal group weight). Therefore, a logical choice for group minimization will be to reserve a large block of the lower energy range as a single group, while preserving some fidelity in the higher (keV to MeV) energy range. A convenient visual aide for roughly determining group boundary cutoff reservations is by graphing the relative flux as a function of energy, and selecting energy ranges that average the flux contribution over a large energy range. Further adjustments to the group structure for an even number of groups are recommended for parallel computation in the event that energy group decomposition of the PENTRAN transport simulation is required. In this paper, a collapsed group structure ranging from 4 groups to 24 groups (even groups only) was considered.

Table 7: Eigenvalue Results for the 238 group MOX Fuel Pin

Number of Groups	Code	k_∞	2σ Standard Deviation (MC) or Tolerance (Deterministic)	Percent Difference to C/E MCNP5
N/A	Continuous Energy MCNP	1.44761	1.20×10^{-4}	N/A
238	Multigroup MCNP (P_3 pdf)	1.44243	2.60×10^{-4}	0.3584 %
238	SCALE6.1 (2d $S_8 - P_3$)	1.44187	1.67×10^{-4}	0.3973 %
238	PENTRAN (3d $S_12 - P_3$)	1.44320	2.00×10^{-6}	0.3051%

4.3 Fast Reactor MOX Pin Eigenvalue Results and Collapse Analysis

4.3.1 Cross-Comparative 238 Group Analysis of PENTRAN and Multigroup MCNP

The initial reference 238 group eigenvalue computations were performed using 2d SCALE6.1, 3d PENTRAN, continuous energy MCNP, and multigroup MCNP. The deterministic runs used S_8 quadrature and P_3 scattering moment cross sections, and the multigroup MCNP cases used equivalently sampled P_3 moment cross sections. The k_∞ for the MOX fuel pin was close to 1.44 in each case, with several hundred PCM deviations between the various codes. The results are tabulated in Table 7.

The relative flux versus energy for the MOX fuel pin is shown in Figure 27. The PENTRAN fluxes were normalized with respect to the fine-mesh, volume-weighted fission source term; the MCNP fluxes were normalized per source particle with error bars shown. The MCNP calculation could not achieve adequate particle statistics to render fluxes below 10 eV; nonetheless, the converged values from the deterministic PENTRAN calculation are shown, and there is reasonable agreement between the two codes. Higher quadratures were warranted for the 3d deterministic calculations due to fast neutron streaming.

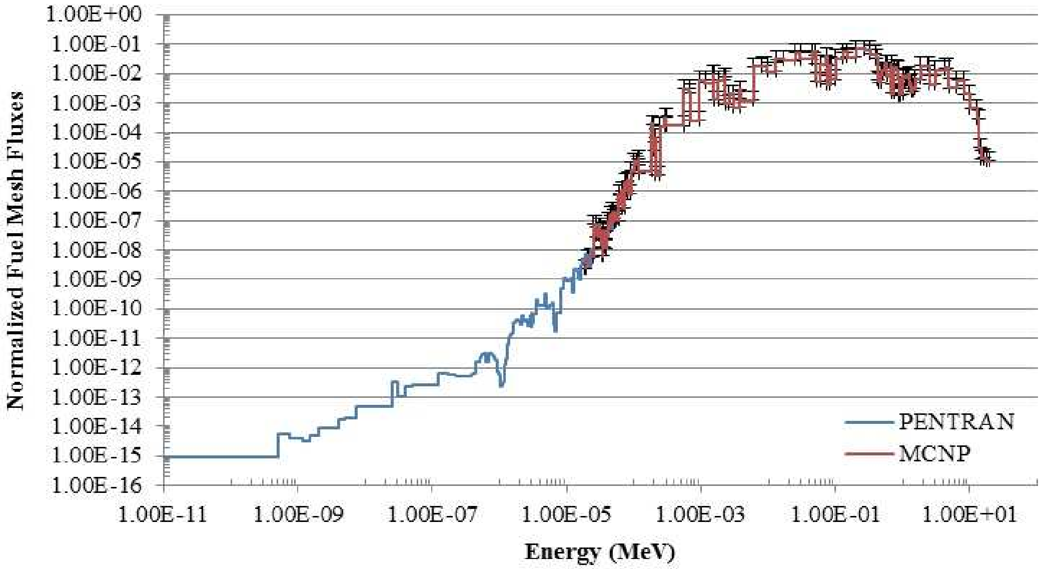


Figure 27: Normalized Fuel Flux Comparison Between Multigroup MCNP and PENTRAN with Same Basis Cross Sections (238 Group Reference from SCALE6.1). At a center fuel cell fine mesh, with error bars for MCNP

The 238 groups in SCALE6 were then collapsed to N number of groups, where N was set as an even number from 4 to 24. In each of these cases, the best results were generated when Groups 1-10 (20 MeV - 4.8 MeV), 61-74 (3.9 keV - 550 eV), and 75-238 (305 eV - $1.04 \times 10^{-4} eV$) were lumped together. The balance of groups is handled internally in YGROUP with corresponding weights assigned. Graphically, these regions are shown for the 10 group YGROUP cross section collapsed case in Figure 28. The broad group weight shows the relative importance of that particular group for the system, and hence any system-wide characteristics such as the k_∞ which rely on group data. Previous analysis by Yi, et al, showed that a broad group keff can be preserved within 100 PCM when collapsing from 26 groups down to 5 groups [62]. Notice that the normalized broad group weight for the 75-238 group region is 6.985×10^{-4}, indicating (as expected) that the thermal regions of a fast reactor are not very important to fission relative to the fast groups (generally > $\tilde{3}0$ keV).

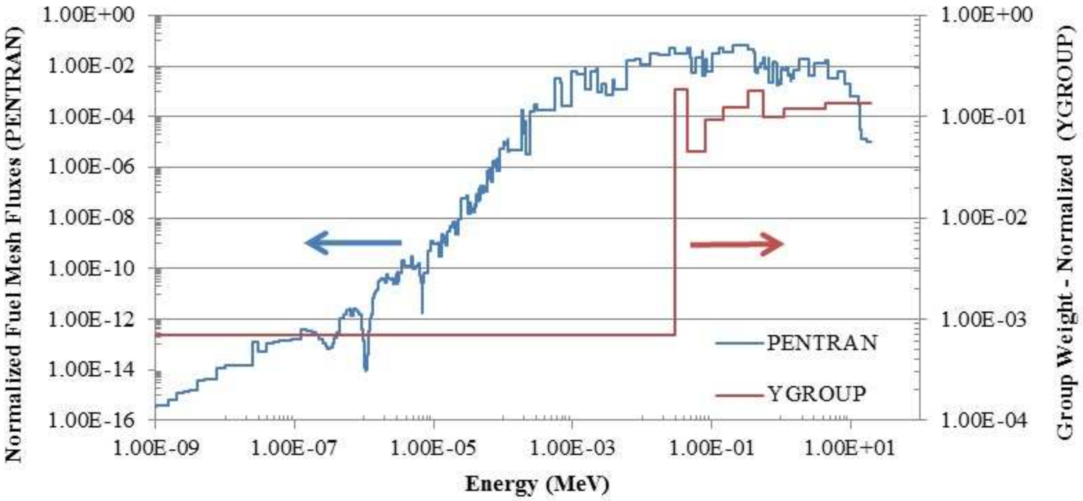


Figure 28: 238 Group Relative Flux and 10 Group Weighting Derived from Fission Adjoint Pin Calculation as a Function of Energy for the 21% MOX Fuel Pin.
Note: PENTRAN fluxes and YGROUP weights are scaled on different ordinate axes, arrows indicate reference scale

Table 8: Eigenvalues for discretized multigroup MCNP5 calculations with YGROUP collapsed cross sections - Fixed group structure

Number of Groups	Collapsing Option	Weighting Option	k_∞	2σ Standard Deviation	Percent Deviation to Cont. E. MCNP
8	Reaction Rate	Flux	1.47803	6.0×10^{-4}	2.079%
8	Contributon	Flux	1.47046	5.8×10^{-4}	1.566%
8	Contributon	Contributon	1.49128	6.2×10^{-4}	2.972%
8	Contributon	Biased	1.51403	6.1×10^{-4}	4.485%

4.3.2 Group Dependent Eigenvalue Calculations

Both multigroup MCNP and PENTRAN with YGROUP basis collapsed cross sections were performed using a variety of collapsing/weighting schemes. It is insightful to see how different collapsing and weighting schemes can impact the k eigenvalue for a fixed collapsed group selection. This is shown in Table 8.

An interesting result from Table 8 is that the contributon collapsed, flux (reaction) weighted cross sections appeared to have the lowest k_∞; also, the contributon

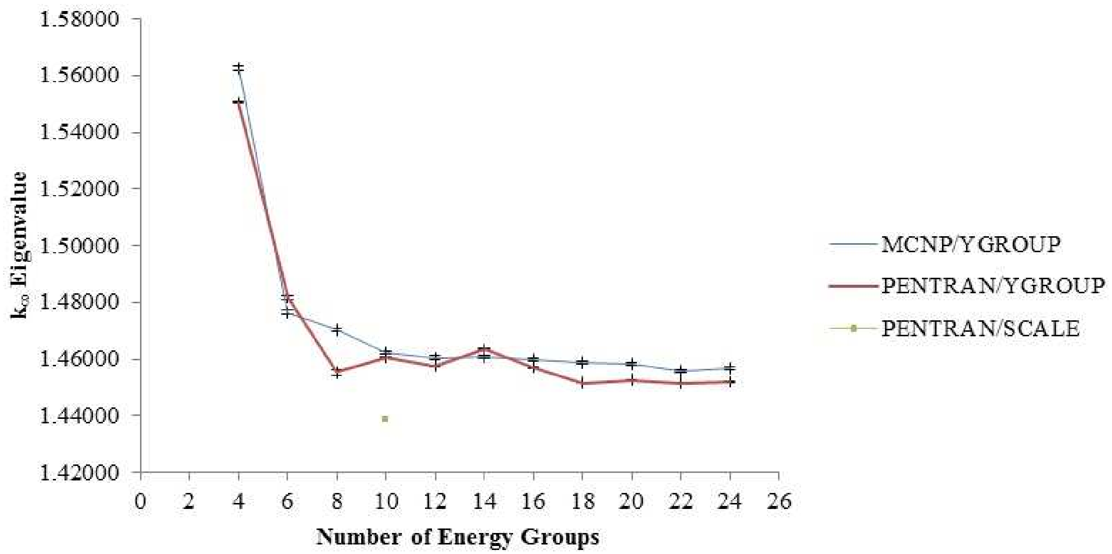


Figure 29: Eigenvalue as a Function of Energy Group for Contributon Collapsed, Flux Weighted YGROUP Multigroup Cross Sections

collapsed, bias weighted cross sections were among the highest eigenvalues observed. This behavior was fairly consistent for several group selections. Since the objective of our analysis was ultimately to choose a group structure that yielded an eigenvalue closest to the reference 238 group case, the contributon collapsed, flux (reaction) weighted option appeared to be the best collapsing/weighting scheme to select.

In order to address the validity of a group scheme, a series of PENTRAN and MCNP runs were performed with the contributon collapsed, flux (reaction) weighted YGROUP cross sections computed in even multiples of 2 spanning from 4 groups to 24 groups. In thermal reactors it is the general consensus to collapse to a fairly small amount of groups, generally from three to six; our primary question was whether or not this could apply to fast reactor geometries. This led to Figure 29, which shows an asymptotic approach of the eigenvalue to the 238 group eigenvalue.

4.3.3 Adjoint Fission Results

As described, an adjoint fission source was painted (as shown in Figure 26) on the boundaries of the problem to determine the importance to fission outside the pin.

Table 9: YGROUP Broad Cross Section Collapse Energies and Weights for a 10 Broad Group Collapse

Broad Group	Fine Group Range	Group Upper Energy (MeV)	Broad Group Weight
1	1 to 10 (Reserved)	2.000×10^{1}	1.77876×10^{-2}
2	11 to 21	4.304×10^{0}	1.35582×10^{-1}
3	22 to 33	1.100×10^{0}	1.18728×10^{-1}
4	34 to 39	5.500×10^{-1}	1.00373×10^{-1}
5	40 to 42	3.300×10^{-1}	1.82944×10^{-1}
6	43 to 46	1.500×10^{-1}	1.21835×10^{-1}
7	47 to 52	8.200×10^{-2}	9.42636×10^{-2}
8	53 to 53	4.500×10^{-2}	4.46333×10^{-2}
9	54 to 74 (Reserved)	3.000×10^{-2}	1.83156×10^{-1}
10	75 to 238 (Reserved)	3.050×10^{-4}	6.98533×10^{-4}

With reflected boundaries, this approximately gives the fraction of neutrons causing fission. The behavior of the adjoint importance as a function of energy group is shown in Figure 30. Note that the contribution to fission is largest in the highest groups as expected for a fast reactor configuration. This behavior also justifies the low corresponding weights in the YGROUP rendered, as well as the ability to lump all groups lower than 300 eV into a single group.

4.3.4 Homogenization Impacts with MOX Fuel

Previous discrete ordinates transport homogenization analysis using the SNT code by Chiba indicates that it is possible to achieve within $0.1\%\Delta k/k$ for small fast assemblies such as the cylindrical 3d sodium-cooled MZA core [16]. The homogenous eigenvalues were relatively invariant when using level-symmetric quadrature, so S_8 quadrature was assumed for eigenvalue calculations. Our results roughly agree with this metric, particularly when all materials are homogenized together.

The MOX fuel for this particular problem has a fairly small pin size (0.66 cm radius) and pin-to-pin pitches. Even though representing this geometry heterogeneously

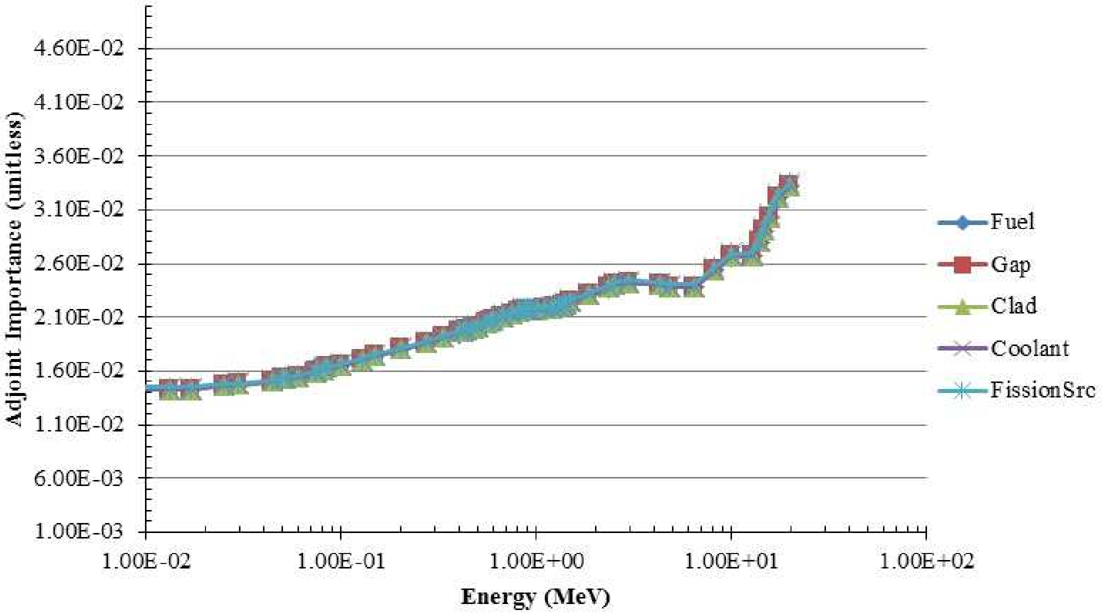


Figure 30: Adjoint Importance Based on Fission as a Function of Energy for the MOX Fuel Pin. Note that the second peak tapers off after below around 3 MeV.

on an assembly and core basis is possible with parallel memory in our deterministic computations, it would take a large allocation of computing power to do so since a very large global mesh would be required to discretize the finely grained fuel-gap-clad region. Homogenization of the collapsed group cross sections using a standard reaction rate weighting scheme with fine mesh fluxes was performed using the HMIX code [42]. Homogenizing the fuel, gap, clad, and coolant for large fast reactors is valid since the mean free path of fast neutrons is on the order of centimeters (i.e. for 4.304 MeV, the mean free path is 3.72 cm), larger than any individual pin size. Table 10 depicts results from a brief eigenvalue calculation resulting from homogenization of different materials.

4.4 High Enrichment PWR Pin Group Collapse Study

In the PWR pin analysis, we adopted a tandem strategy to incorporate the optimal use of both determinstic and Monte Carlo codes. The cross section collapsing strategy is driven by a need for accurate contributon values. One component of the contributon

Table 10: PENTRAN eigenvalue calculation (with level symmetric quadrature) using HMIX cross sections based on 10 group SCALE collapsed XS using the YGROUP group structure

Group	Code	k_∞	2σ Standard Deviation	% $\Delta k/k$ compared to 238 group PENTRAN
10	PENTRAN/HMIX Fuel-Gap-Clad	1.404689	5.70×10^{-5}	-2.667
10	PENTRAN/HMIX Fuel-Gap-Clad-Coolant	1.440745	3.06×10^{-4}	-0.169

is the averaged adjoint by material and the other component is the averaged scalar forward flux by material. For eigenvalue/criticality, we use the fuel fission source as the driver. In order to achieve reasonable fidelity, we perfomed calculations that used the finest group structures allowable. The adjoint importances are obtained using a PENTRAN 49 group calculation, which are then upsampled/extrapolated to 238 groups. The scalar fluxes are derived from multigroup Monte Carlo using the 238 group structure. We detail the strategy and the results of their combined application in the next sections.

4.4.1 Ideal Tandem Use of Monte Carlo and Deterministic Strategies

The effective collapse of cross sections is an integration over the energy variable, which is information loss that cannot be recovered to 100 percent, as there is a smearing of highly detailed flux that would could only be preserved through high order mathematical functional represention, which then would increase storage requirements for cross sections. Despite this loss in flux resolution, the appropriate choice of collapsing and subsequent weighting performed with the fast reactor pin led to a strategy also applied to the PWR pin.

Typically, in handling weighting, the manner in which this is handled is done via a homogenization strategy, of which there are many forms but we will focus on the "Contributon" weighting with applied adjoint fission source. We detail the

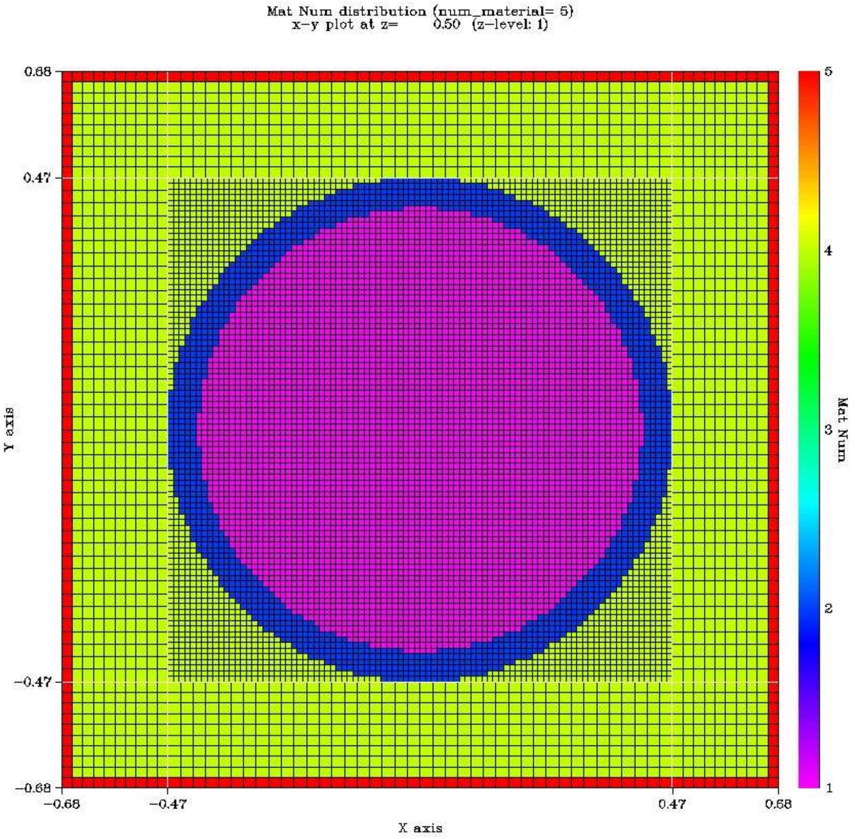

Mat Num distribution (num_material= 5)
x–y plot at z= 0.50 (z–level: 1)
0.68
0.47
–0.47
–0.68
Y axis
–0.68
–0.47
0.47
0.68
X axis
5
4
3
2
1
Mat Num

Figure 31: PWR Pin Model Purple Fuel, Blue Clad, Green Moderator, Red Adjoint Moderator (Used only in conjunction with Adjoint Calculation)

effectiveness of collapse by using measurement of eigenvalue yet again, which we prioritize in retaining accurately. Another reason for using the eigenvalue as a metric for cross section collapse is that, in the case of burnup, it is important to maintain for proper actinide attribution, as the fission source term drives the fission yield which then drives the linear Bateman relationship formed from effective constants that are driven by reaction rates. We apply the contributon-collapse, flux-weight strategy as done for the fast reactor pin, which should trail over as a viable strategy for the thermal PWR case. To support the strategy, we address the narrative by supplying the case of the thermal PWR pin shown in Figure 31, which is an infinite 2d PWR pin in a square lattice, with six-sided reflective BCs. The materials for the transport model are based on the fuel specified by the BR3 Belgian reactor benchmark.

4.4.1.1 Multigroup Monte Carlo Application with 238 Groups

YGROUP starts with the fine 238 group structure (SCALE), and needs averaged material fluxes and adjoint importances. This is important to feature, as local flux resolution is not incredibly important. In the case of 238 groups, Monte Carlo is favored over deterministic, because raw number of groups presents a memory challenge for Sn. Calculations for Sn can be performed, but at present, takes on the order of days without advanced acceleration, whereas the calculations for Monte Carlo can be obtained within hours. We present the material averaged forward flux for the three base materials in Figure 32. Nonsmooth flux profiles are clearly detected at groups 150 and higher (3 eV and lower). The information in the Monte Carlo calculations are reduced to 238 groups, which essentially provides a probability distribution function for the PWR fuel type. The higher flux below 3 eV implies a thermal system.

Also, Figure 32 shows that material averaged flux behavior for all the 3 materials (fuel,clad,moderator) are not separated significantly, where the fuel flux, in the marked red crosses, is more strongly weighted in the thermal region (as it provides the source). Since our end goal is an eigenvalue calculation for a lower group structure with associated reduced multigroup cross sections, compared to a reference 238 group structure, we select the fuel material as the basis material flux used in the the Contributon weighting.

Stochastic issues are ever present and a potential concern, the next Figure 33 shows a subset of fluxes with obvious statistical errors present. Groups 190 and 237 are clearly are very noisy. The problem of local fine mesh flux inaccuracies by reducing error in the fuel flux by tallying a $0.4 \times 0.4\text{cm}^2$ panel over the fuel; this calculation leads to Figure 34, which in our analysis was driven to keep fuel flux error below 1 percent, which is achieved except for the bank of 5 groups at the top and the bottom of the 238 groups, since fluxes are quite small at high, fast energies and extremely thermal energies. This too is reflected in the fuel flux with minimized values in the

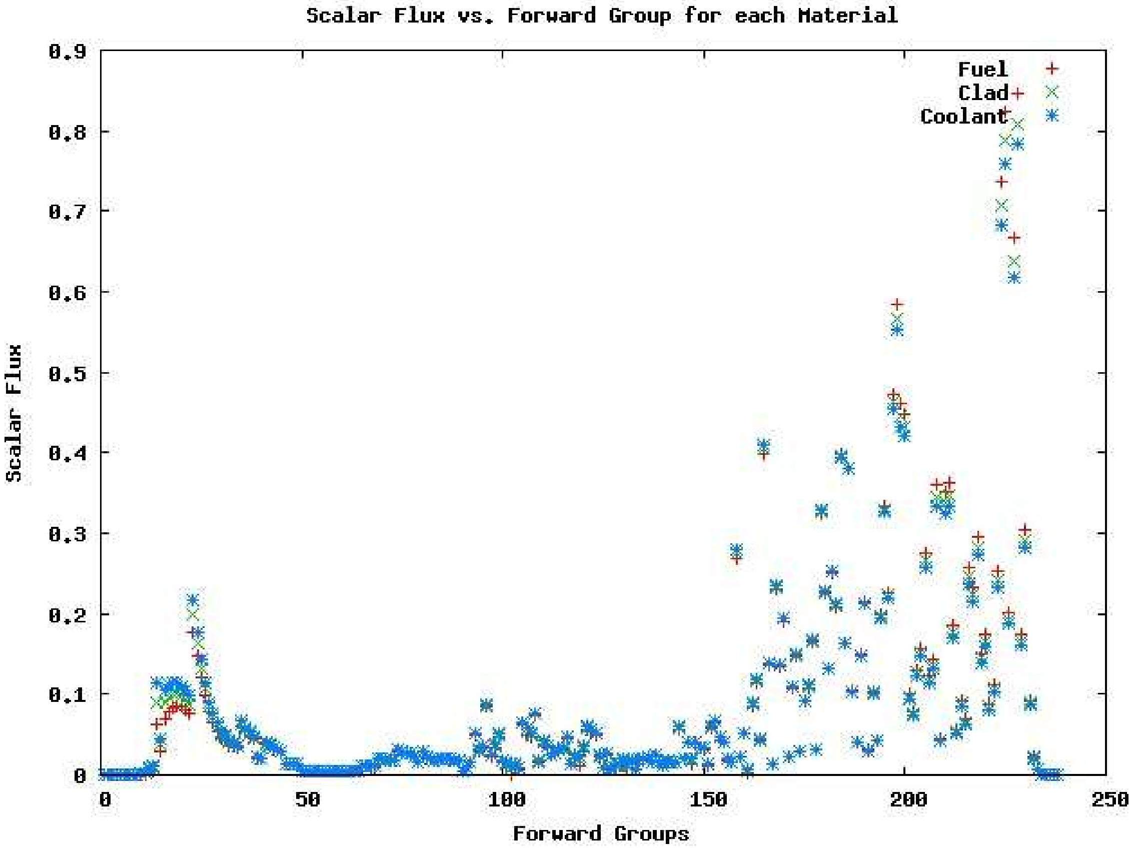


Figure 32: Material Averaged Flux Behavior in 238 group MCNP Model

extreme ends. This figure shows that fuel flux averages are acceptably attainable with a multigroup Monte Carlo calculation.

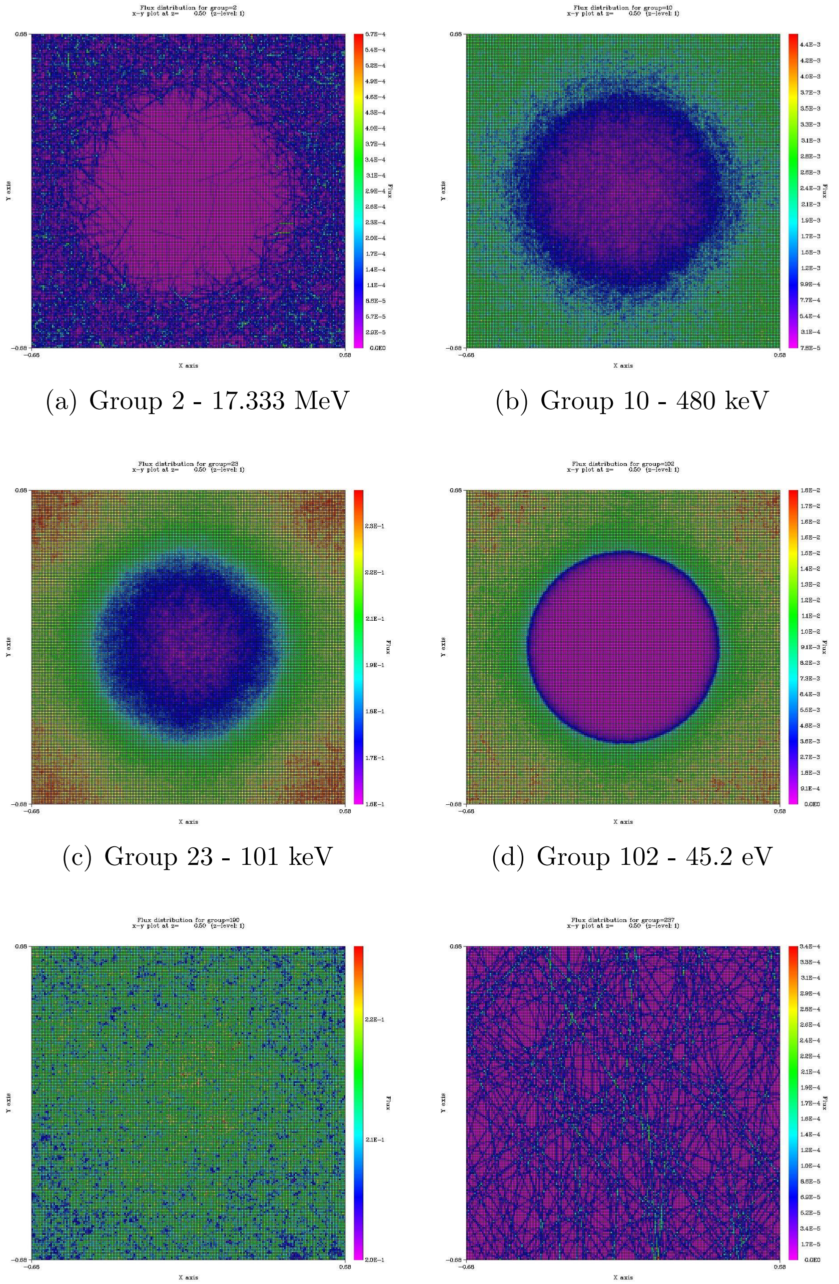
4.4.1.2 Deterministic Adjoint Application with PENTRAN Using 49 Groups

The deterministic Sn calculation with PENTRAN is applied by using a 49 group basis with the adjoint fission source strategy, applied in this case for the thermal PWR pin. Prior studies with 49 group cross sections were used by the NRC for PWR analyses. As is known from analysing the fission cross section, now used as the "source" in adjoint mode, the source term is relatively smooth, in contrast to the widely varying thermal model. This behavior is also expected, as the fast MOX case presented a relatively smooth adjoint importance behavior as a function of energy (Figure 30. We achieve similarly with the thermal PWR adjoint fission source calculation, shown in 35, which shows adjoint importance as a function of the 49 foward groups (see Appendix B for group structure). Note that the marked adjoint fission source (purple square) is referring to the location of the red zone given by 31. For YGROUP, the fuel location is of interest because we need the importance of fuel causing a fission event to the edge of the infinite lattice pin.

The 49 group based adjoint is then extrapolated to 238 groups to match the original SCALE 238-group structure. This is justified in the qualitative assessment of the adjoint behavior being smooth across groups, with a slight bump in groups 20 (8.1 eV) through 35 (0.325 eV) where fuel is slightly more important in the thermal range), but the probability distribution function is very flat except toward the most thermal range.

4.4.2 Evaluation of Group Collapse and Eigenvalue Preservation from 238 Groups to 3 Groups

The last Sections, 4.4.1.1 and 4.4.1.2, resolved the determination of what are effectively material averaged fluxes used in the weighting portion of YGROUP, where the



(a) Group 2 - 17.333 MeV

(b) Group 10 - 480 keV

(c) Group 23 - 101 keV

(d) Group 102 - 45.2 eV

(e) Group 190 - 1 eV

(f) Group 237 - 0.0005 eV

Figure 33: Stochastic Flux Variation for Various Multigroup Tallies in MCNP

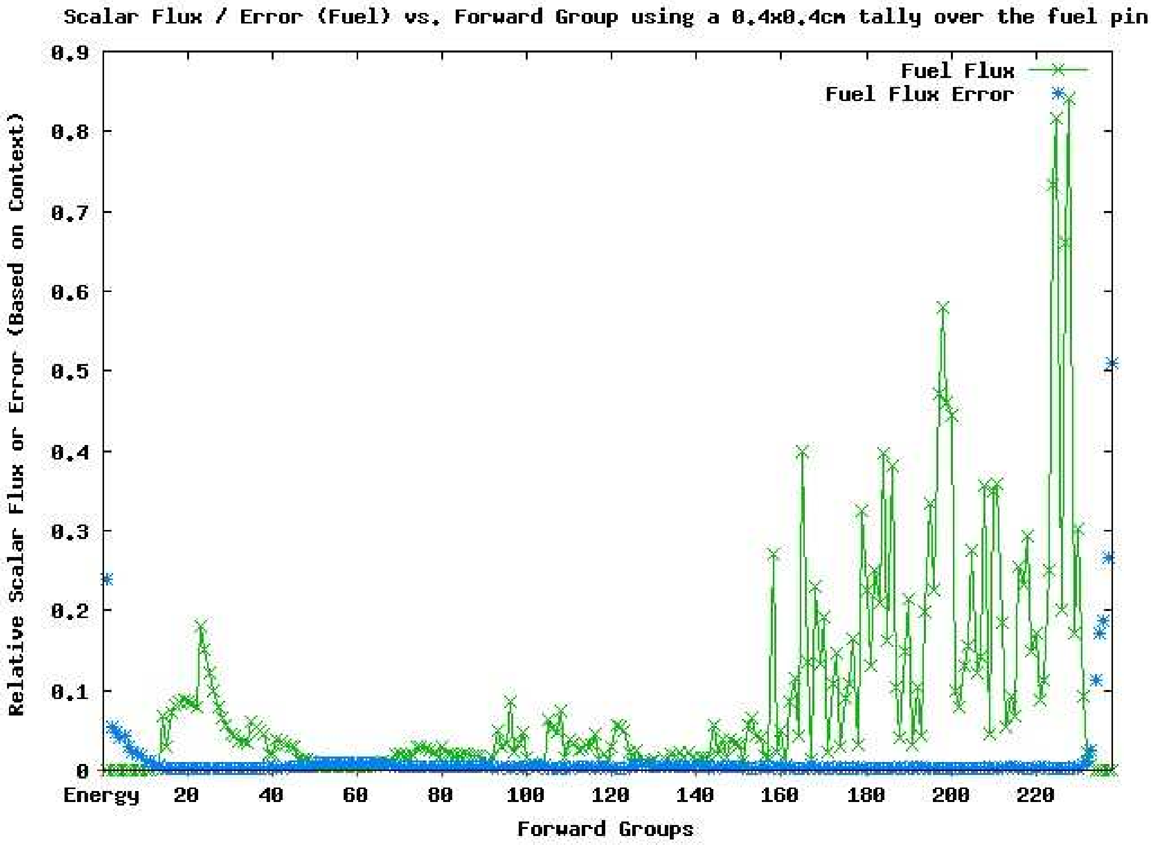


Figure 34: Fuel Flux and Associated Monte Carlo Tally Error Using a Square Tally by Group

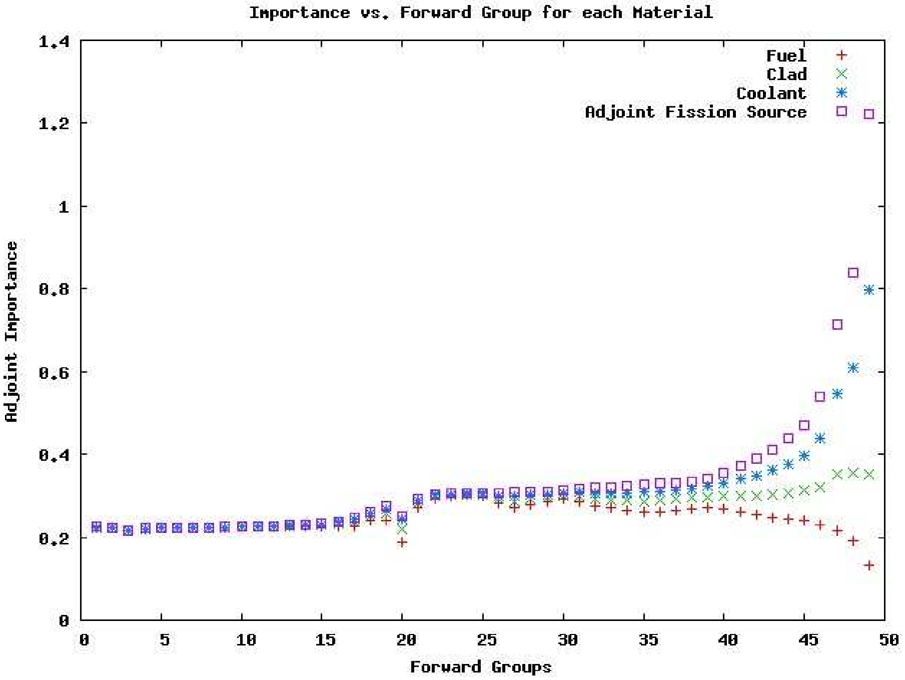


Figure 35: Adjoint Importance as a Function of 49 Groups

contributon is calculation by multiplying both fission adjoint importance times forward fuel flux in the same group, for each of the 238 groups. The material averaged forward fuel flux is applied in the reaction rate collapsing.

The automatic YGROUP entails running deterministically for every possible user requested group structure, in this algorithm, we retain the first unique group structure that is generated. Out of 238 cases starting from user requesting 238 groups all the way to 1 group, this generated 38 unique group structures. There are indeed more possible unique group structures, but to improve the effiencicy taking the first uniquely generated *number* of groups was the primary basis for our automatic selection. The eigenvalues were calculated by using XSMCNP to convert the cross sections for use with multigroup MCNP5 [64], with 'kcode' settings: 20000 particles per cycle, 50 skip cycles, and 250 total cycles.

The eigenvalues for each one of the 38 cases is presented in Figure 36, where the expected contributon weight and reaction rate collapse are compared to a reaction rate weight and reaction rate collapse. The contributon-reaction rate case in red provides is more slighly improved in the representation of the lower group structures compared to the reaction rate - reaction rate case. Ultimately, there are many more insights to gather; the work in this chapter is intended to provide the guiding framework for an automatic algorithm which would run from a single basis transport input.

Some of background work driving this analysis is established by automated conversion tools and scripts, which could lead to a single analysis tool that is run to assign a low order broad group structure suitable for future transport and burnup calculations.

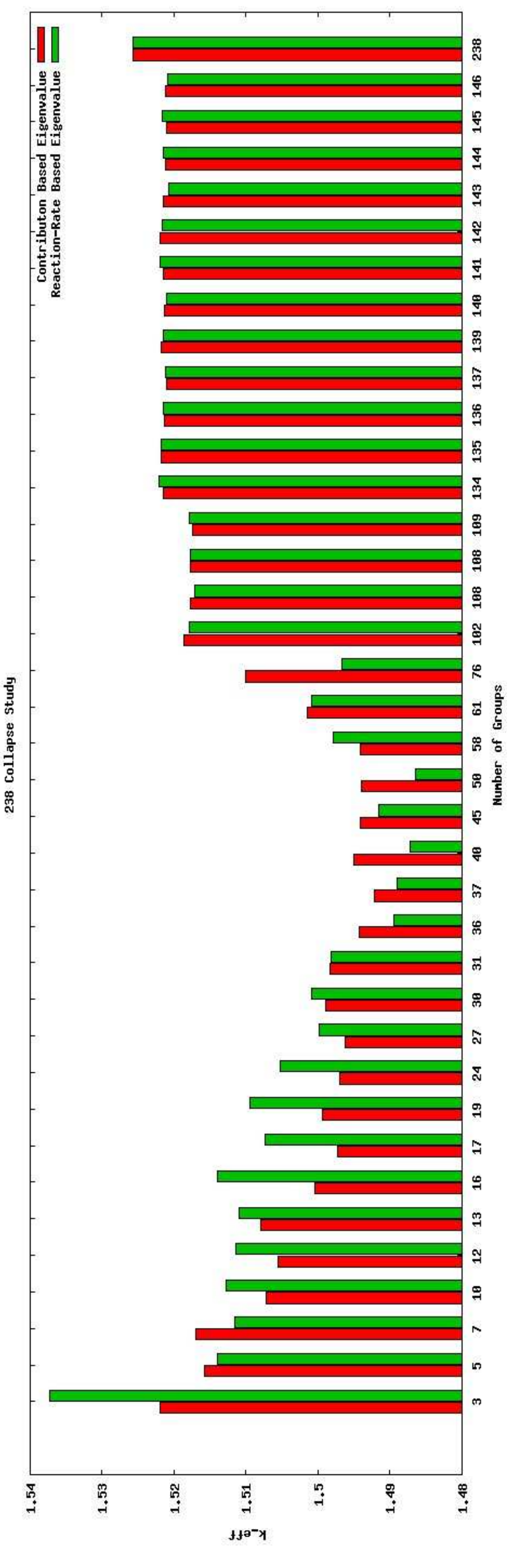


Figure 36: keff versus Unique Multigroup Using Hybrid Deterministic Strategy

CHAPTER V

DISCRETIZED MESH (SPACE) OPTIMIZATION

Hybrid methods of neutron transport have increased greatly in use, for example, in applications of using both Monte Carlo and deterministic transport to calculate quantities of interest, such as flux and eigenvalue in a nuclear reactor. Many 3d parallel Sn codes apply a Cartesian mesh, and thus for nuclear reactors the representation of curved fuels (cylinder, sphere, etc.) are impacted in the representation of proper fuel inventory (both in deviation of mass and exact geometry representation). For a PWR assembly eigenvalue problem, we explore the errors associated with this Cartesian discrete mesh representation, and perform an analysis to calculate a slope parameter that relates the pcm to the percent areal/volumetric deviation (areal corresponds to 2d and volumetric to 3d, respectively). Our initial analysis demonstrates a linear relationship between pcm change and areal/volumetric deviation using Multigroup MCNP on a PWR assembly compared to a reference exact combinatorial MCNP geometry calculation. For the same multigroup problems, we also intend to characterize this linear relationship in discrete ordinates (3d PENTRAN) and discuss issues related to transport cross-comparison. In addition, we discuss auto-conversion techniques with our 3d Cartesian mesh generation tools to allow for full generation of MCNP5 inputs (Cartesian mesh and Multigroup XS) from a basis PENTRAN Sn model.

5.1 Introduction

Hybrid transport methods have become vital tools in accurate transport calculation. In particular, using both Monte Carlo (either continuous or multigroup energy) and deterministic transport (Sn, MOC, CP) have become commonplace. However, the burden has increased for the user to create multiple models for both Monte Carlo and Sn transport separately. To address this issue, we have written an auto-conversion tool to convert Cartesian mesh in Sn to Monte Carlo. Also, as Monte Carlo models render exact combinatorial geometry, for discrete ordinates models with discrete meshes, we need to consider the impact of the deviation from the true geometry, which is mostly driven by deviation from exact area or volume.

In this chapter, we discuss auto-conversion of Cartesian meshes in the 3d Cartesian Sn PENTRAN model to the Monte Carlo (MCNP5), using the MESH2MCNP script. This Python software tool, in tandem with XSMCNP, which converts multigroup cross sections in the proper format for MCNP5, can create a complete conversion, save for primary MCNP5 data parameters (e.g. kcode) unrelated to the Sn model.

Our analysis, considers the impacts of Cartesian meshing in MCNP5 (using the MESH2MCNP and XSMCNP tool), applied to a 7-group PWR and MOX assembly (adapted from the C5G7 benchmark[37]) eigenvalue calculation compared to an exact geometry reference calculation using multigroup MCNP5.

5.2 Development of MESH2MCNP Software Tool

The MESH2MCNP tool was developed in order to provide an autoconversion of Sn input decks created for PENTRAN to MCNP, matching the exact Cartesian discretized mesh grid. This tool was also developed, to aid the existing BURNDRIVER framework, as all tools were previously written to support Sn, in particular multigroup fluxes driven obtained from PENPOW, a reaction rate code that feeds data into PENBURN.

What ultimately resulted into a primary goal for the burnup sequence ended up serving as a valuble tool independent of burnup needs; in this chapter, we focus on the issues surrounding the "space" phase space, in particular, the issues related to 3d Cartesian Sn. What is obviously known is that, for PWR systems, the representation of target mass is an important driver of eigenvalue accuracy. However, we quantify a metric that is a characteristic of the transport system.

Early in this chapter, we describe the method of autoconversion of PENTRAN input with MESH2MCNP, via a basic two-level universe scheme. We then follow with existing flux tallying options, and integration with XSMCNP for multigroup cross sections (though if full material specifications are known the use of XSMCNP is not required).

Then, we use a PWR and MOX assembly derived from the C5G7 benchmarks, and focus on spatial mesh grid variation, and a successful non-homogenization technique termed corner fractioning, which is an application of volume weighting given initial overrepresentation of fuel.

5.2.1 Two-Level Universe Scheme

MESH2MCNP segments the same Cartesian grid that is used in the deterministic frame; with PENTRAN there is system of 3d hexahedral fine meshes that are uniform within a designated coarse mesh.

This is done by labeling universes as primary material IDs, and definining basis fine meshes for each coarse mesh. A section of the MESH2MCNP deck is shown in Figure 37. The next figure, Figure 38, shows what the equivalent PENMSHXP generated input deck for PENTRAN, which is the primary input for conversion to MCNP. In both of the preceding figures, only one coarse mesh of nine is shown. The text encoding seen in Figure 38 is also known as FIDO notation and used commonly in many 3d Sn codes, in particular used by ORNL's SCALE package [44].

5.2.2 Flexible Flux Tallying Options

MESH2MCNP was written to support a variety of output flux formats, in particular "fmesh", mctal file output, and also universe fine mesh tallies. Sample output listings are given in Figure 39.

```
c MESH2MCNP MCNP Input Generator (K. Manalo), Date generated:  ↙
   ↳ 2013-07-23
c input deck generated from PENTRAN input: moxm10_out.pen
c
c cell cards -----
c description: each material
c   is assigned to a matching cell number
c   and matching universe (same as matid)
c   inside of the global rpp
1    1   1.0  -99990   u=1     imp:n=1 $ material 1
2    2   1.0  -99990   u=2     imp:n=1 $ material 2
3    3   1.0  -99990   u=3     imp:n=1 $ material 3
4    4   1.0  -99990   u=4     imp:n=1 $ material 4
5    5   1.0  -99990   u=5     imp:n=1 $ material 5
6    6   1.0  -99990   u=6     imp:n=1 $ material 6
7    7   1.0  -99990   u=7     imp:n=1 $ material 7
c
c cells 10001 to 10009  describe a cartesian f.m. lattice in each
c coarse mesh
c
10001  0      -1
         lat=1                       u=10001 imp:n=1 $ lattice
         fill=0:59  0:59  0:0
     7 62R 2 3R 7 5R 2 3R 7 5R 2 3R 7 5R 2 3R 7 5R 2 3R 7 5R 2 3R
     7 4R 2 5R 2 5R 7 3R 2 5R 7 3R 2 5R 7 3R 2 5R 7 3R 2 5R 7 2R
     2 7R 7 1R 2 7R 7 1R 7 1R 2 7R 7 1R 2 7R 7 1R 2 7R 7 1R 2 7R
     7 1R 2 7R 7 1R 2 7R 7 1R 2 7R 2 7R 7 1R 2 7R 7 1R 2 7R 7 1R
     2 7R 7 1R 2 7R 7 1R 2 7R 7 1R 2 7R 7 1R 7 1R 2 7R 7 1R 2 7R
     7 1R 2 7R 7 1R 2 7R 7 1R 2 7R 7 1R 2 7R 7 2R 2 5R
```

Figure 37: MCNP Coarse Mesh Lattice Example Generated by MESH2MCNP

```
/
/  material distribution for zlev=  1
/
    nmattp=1
63R7 4R2 6R7 4R2 6R7 4R2 6R7 4R2 6R7 4R2 6R7 4R2 5R7 6R2 4R7 6R2
4R7 6R2 4R7 6R2 4R7 6R2 4R7 6R2 3R7 8R2 2R7 8R2 2R7 8R2 2R7 8R2
2R7 8R2 2R7 8R2 3Q60 3R7 6R2 4R7 6R2 4R7 6R2 4R7 6R2 4R7 6R2 4R7
6R2 5R7 4R2 6R7 4R2 6R7 4R2 6R7 4R2 6R7 4R2 6R7 4R2 126R7 4R2 6R7
4R3 6R7 4R3 6R7 4R3 6R7 4R3 6R7 4R3 5R7 6R2 4R 6R3 4R7 6R3 4R7
6R3 4R7 6R3 4R7 6R3 3R7 8R2 2R7 8R3 2R7 8R3 2R7 8R3 2R7 8R3 2R7
8R3 3Q60 3R7 6R2 4R7 6R3 4R7 6R3 4R7 6R3 4R7 6R3 4R7 6R3 5R7 4R2
6R7 4R3 6R7 4R3 6R7 4R3 6R7 4R3 6R7 4R3 126R7 4R2 6R7 4R3 6R7 4R3
6R7 4R3 6R7 4R3 6R 4R6 5R7 6R2 4R7 6R3 4R7 6R3 4R7 6R3 4R7 6R3
4R7 6R6 3R7 8R2 2R7 8R3 2R7 8R3 2R7 8R3 2R7 8R3 2R7 8R6 3Q60 3R7
6R2 4R7 6R3 4R7 6R3 4R7 6R3 4R7 6R3 4R7 6R6 5R7 4R2 6R7 4R3 6R7
4R3 6R7 4R3 6R7 4R3 6R7 4R6 126R7 4R2 6R7 4R3 6R7 4R3 6R 4R6 6R7
4R3 6R7 4R4 5R7 6R2 4R7 6R3 4R7 6R3 4R7 6R6 4R7 6R3 4R7 6R4 3R7
8R2 2R7 8R3 2R7 8R3 2R7 8R6 2R7 8R3 2R7 8R4 3Q60 3R7 6R2 4R7 6R3
4R7 6R3 4R7 6R6 4R7 6R3 4R7 6R4 5R7 4R2 6R7 4R3 6R7 4R3 6R7 4R6
6R7 4R3 6R7 4R4 126R7 4R2 6R 4R3 6R7 4R3 6R7 4R3 6R7 4R4 6R7 4R4
5R7 6R2 4R7 6R3 4R7 6R3 4R7 6R3 4R7 6R4 4R7 6R4 3R7 8R2 2R7 8R3
2R7 8R3 2R7 8R3 2R7 8R4 2R7 8R4 3Q60 3R7 6R2 4R7 6R3 4R7 6R3 4R7
6R3 4R7 6R4 4R7 6R4 5R7 4R2 6R7 4R3 6R7 4R3 6R7 4R3 6R7 4R4 6R7
4R4 126R7 4R2 6R7 4R3 6R7 4R6 6R7 4R4 6R7 4R4 6R7 4R6 5R7 6R2 4R7
6R3 4R7 6R 4R7 6R4 4R7 6R4 4R7 6R6 3R7 8R2 2R7 8R3 2R7 8R6 2R7
8R4 2R7 8R4 2R7 8R6 3Q60 3R7 6R2 4R7 6R3 4R7 6R6 4R7 6R4 4R7 6R4
4R7 6R6 5R7 4R2 6R7 4R3 6R7 4R6 6R7 4R4 6R7 4R4 6R7 4R6 63R7
```

Figure 38: FIDO Textual Compression Generated with PENMSHXP for PENTRAN for a Single Coarse Mesh

```
c data card -----
mode n
c material cards
m1  1000.22m  1.0
m2  2000.22m  1.0
m3  3000.22m  1.0
m4  4000.22m  1.0
m5  5000.22m  1.0
m6  6000.22m  1.0
m7  7000.22m  1.0
mgopt f 7
c fmesh tallies that mimic fine meshes in each coarse mesh
fmesh14:n  origin=0.0 0.0 0.0
      imesh= 7.56 iints= 60
      jmesh= 7.56 jints= 60
      kmesh= 1.0 kints= 1
fmesh24:n  origin=7.56 0.0 0.0
      imesh= 13.86 iints= 50
      jmesh= 7.56 jints= 60
      kmesh= 1.0 kints= 1
..
..
c universe tallies perform less efficiently than fmesh
c f14:n  ( ( 1 2 3 4 5 6 7 ) < 10001[0:59  0:59  0:0] < 50001 )
c f24:n  ( ( 1 2 3 4 5 6 7 ) < 10002[0:49  0:59  0:0] < 50002 )
..
..
```

Figure 39: MESH2MCNP Tally Options

5.2.3 Development of Multigroup Support

Multigroup cross-section conversion is aided by XSMCNP developed by Yi at the University of Florida and used at Georgia Tech [64], which was also used for analysis in Chapter 4. This Fortran code converts standard ORNL/LANL based multigroup cross section formats to a suitable “xsdir” file and labeled “mgxs” file which are used with a suitable MCNP input file using the **mgopt** option availble in MCNP.

5.3 Application to C5G7 PWR Assembly

As described in the C5G7 benchmark, a PWR fuel pin is represented by a homogenized fuel-clad (exact cylinder) and moderator. We use MESH2MCNP to generate 10×10 mesh, 11×11 mesh, all the way to 40×40 mesh over the fuel pin.

5.3.1 C5G7 Cropped PWR and MOX Models

A PWR cropped from the C5G7 benchmark is used, shown in Figure 40, which uses a 7 group P_0 Legendre cross section with fuel-clad homogenized as a cylinder. Figure 42 shows a sampling of meshes used, from 10×10 to 40×40 mesh per pin (with black lines removed from 40×40 case). The model is run with reflective BCs.

5.3.2 Results

Using the mesh generation tool, we generated PWR assembly cases from 10×10 pin mesh to a 40×40 pin mesh (31 MCNP inputs and 31 PENTRAN inputs).

In order to reasonably mitigate statistical uncertainty in eigenvalue, the mesh cases were run to satisfy a standard deviation of no worse than 0.00006 with kcode specified to satisfy 100,000 particles per cycle, 50 skipped cycles, with 1050 active cycles.

Figure 43, with the appropriate scaling adjustments shows the plot of negative areal deviation vs. mesh density, with a clearly related correlation between pcm change and areal deviation. In the case of the PWR assembly with homogenized

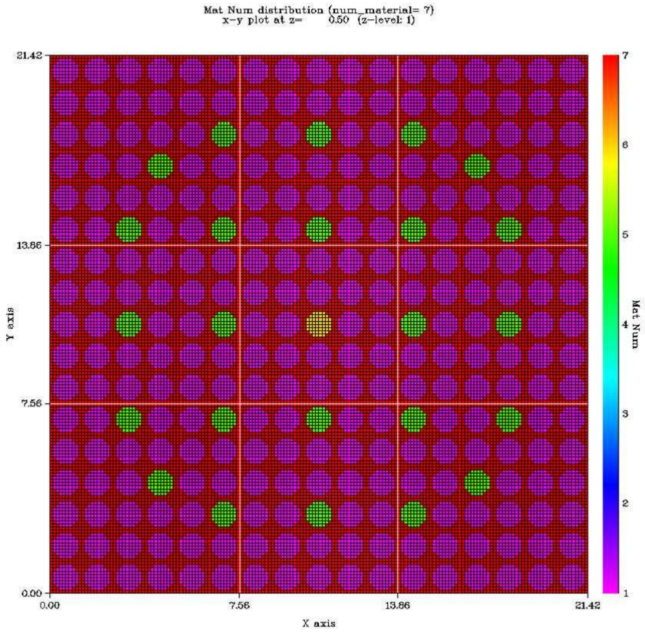

Mat Num distribution (num_material= 7)
x-y plot at z= 0.50 (z-level: 1)
21.42
13.86
7.56
0.00
0.00
7.56
13.86
21.42
Y axis
X axis
Mat Num
1
2
3
4
5
6
7

Figure 40: PWR Assembly Model generated with PENMSHXP

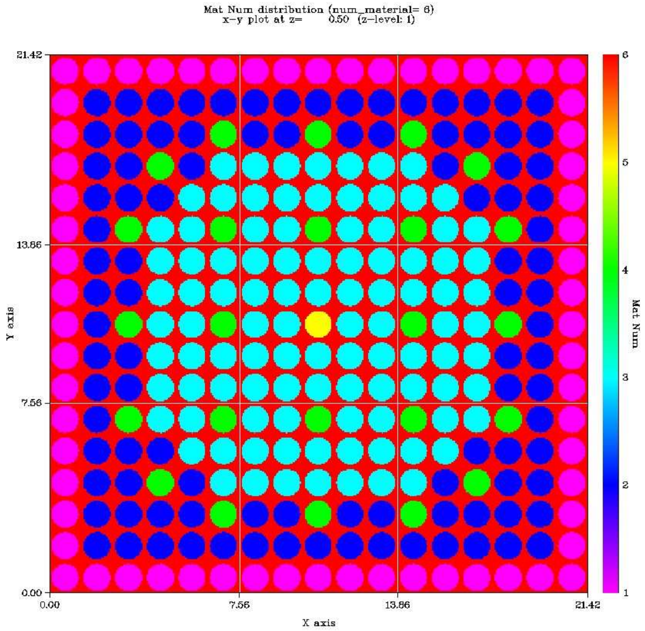
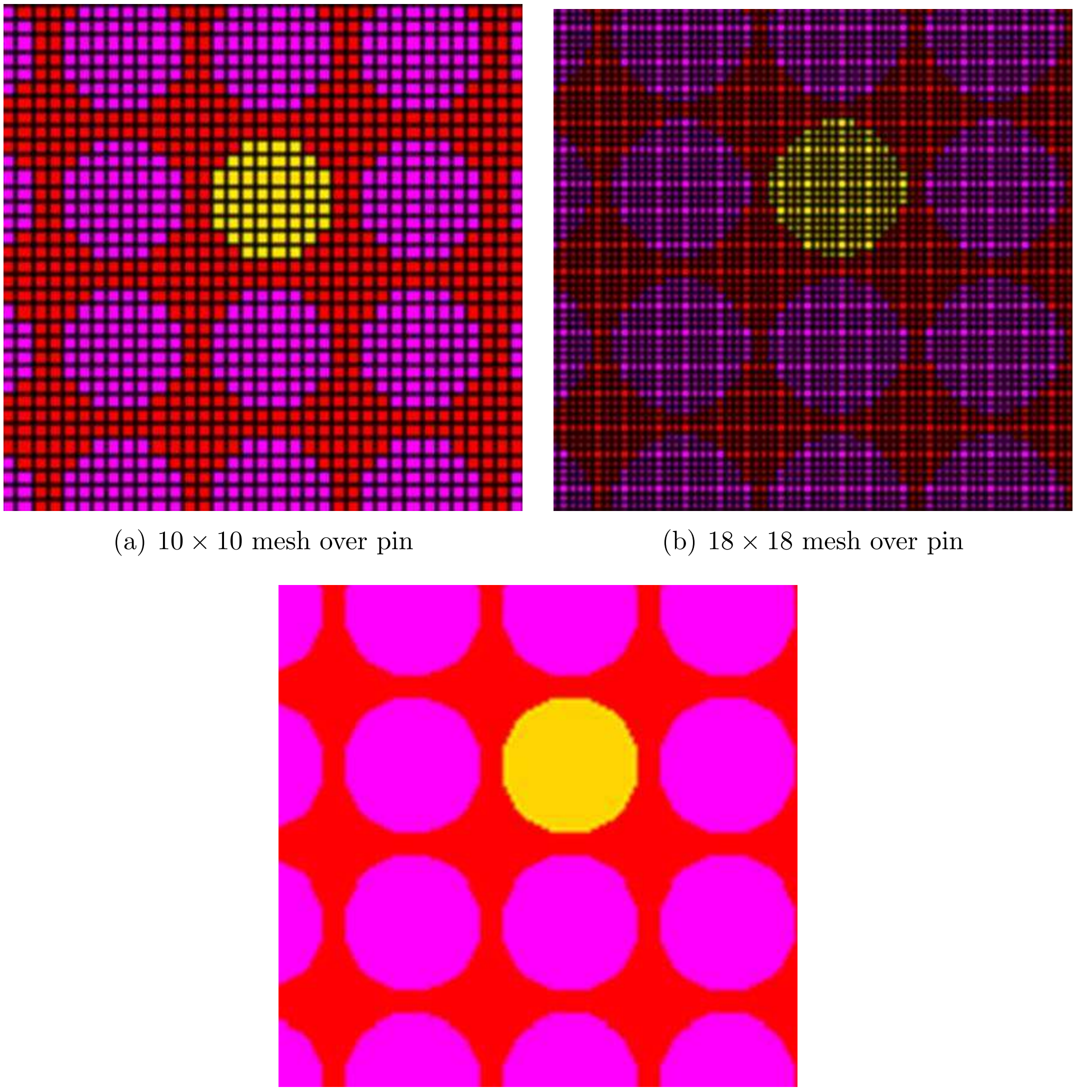

Mat Num distribution (num_material= 6)
x-y plot at z= 0.50 (z-level: 1)
21.42
13.86
7.56
0.00
0.00
7.56
13.86
21.42
Y axis
X axis
Mat Num
1
2
3
4
5
6

Figure 41: MOX Assembly Model generated with PENMSHXP



(a) 10 × 10 mesh over pin

(b) 18 × 18 mesh over pin

(c) 40 × 40 mesh over pin

Figure 42: Various Meshing Per Fuel Pin for the PWR Assembly Case

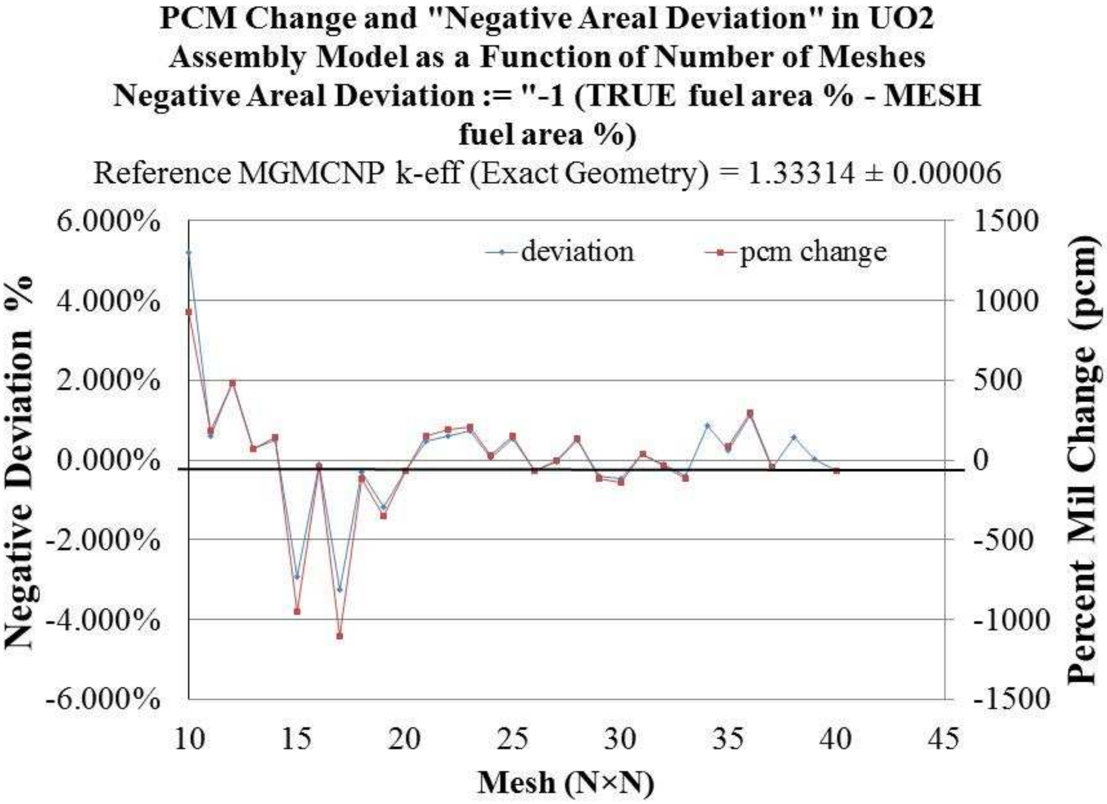


Figure 43: PWR Assembly PENTRAN PCM Deviation vs. Mesh

fuel-clad, a decrease in target mass increases the pcm change, raising the eigenvalue. This plot is effective in demonstrating an optimality of mesh selection; typically the user would want to minimize both mesh size and areal deviation. A 11 × 11 mesh would be optimal here. A caveat of this strategy is the jagged edge (in imaging term as aliased) effect which is more strongly present in small meshes. Our study aims to avoid this aliased effect by starting with a 10 × 10 mesh per pin forthright.

To better quantify this relationship between pcm change and negative areal deviation, we plot pcm change vs. negative areal deviation and demonstrate a linear trend in Figure 44. With the PWR assembly, the slope is calculated to be about 300 pcm / (% areal deviation), which demonstrates the sensitivity of the eigenvalue in deviating from the targeted mass. For example, this implies that 0.30 percent in areal deviation from the true area, can drive a change of up to 100 pcm. Results for two types of quadratures, EO_{16} and S_{16} , are given for PENTRAN to compare against the reference MCNP multigroup. As studied in Chapter , the higher order Even-Odd quadratures are more accurate than their level-symmetric counterparts. This is demonstrated with the UO2 figure, where EO_{16} shifts towards the reference MCNP multigroup result.

Again, we performed thesame analysis with the MOX assembly, and achieve a slopes of 647 and 681/653 for the MCNP and PENTRAN cases seen in Figure 45. This would indicate a pcm sensitivity to volumetric deviation that is a factor of 2.2 times more than the UO2 assembly. In this case, we show that improvements in level-symmetric quadrature from S_8 to S_{16} show an improvement towards the MCNP multigroup eigenvalues.

To retain positivity in slope plots, we deliberately reversed the sign of the slope. There may be other problems beyond this investigation that exhibit a different sign in slope.

5.4 Corner Fraction Technique

It is simple to remark that in general, if one increases the mesh grid, the accuracy in eigenvalue increases, though not monotonically, as evidenced by Figure 43. But the question that could be asked is what would the user do if the mesh can no longer be increased? For deterministic Sn calculations, optimization of calculations in 2d and 3d are necessary; suppose we have a model with $10 \times 10 \times 10 = 1000$ fine meshes, to effectively double the resolution increases the total number of fine meshes cubically to $20 \times 20 \times 20 = 8000$. Hence this implies 8 times the calculational effort assuming a linear scaling at best.

One technique we have developed from the analysis of correlated (negative) linear pcm variation against areal/volumetric deviation is termed *corner fractioning*. It is a simple volume weighting technique that appears to drive Type 4 (Sn-Multigroup)

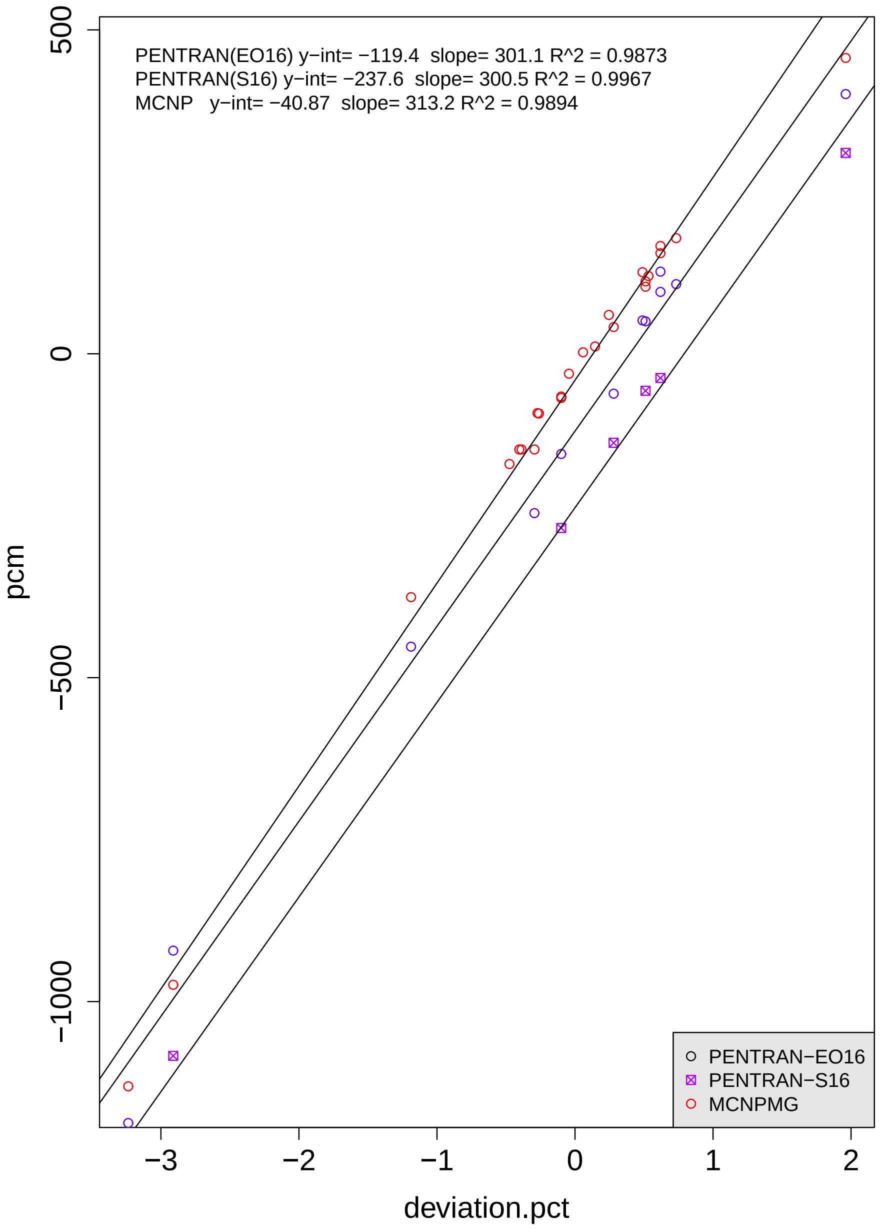


Figure 44: PCM vs. Negative a Areal Deviation for PENTRAN (Blue EO_{16} and Purple S_{16}) and MCNP (Red) for the UO2 assembly

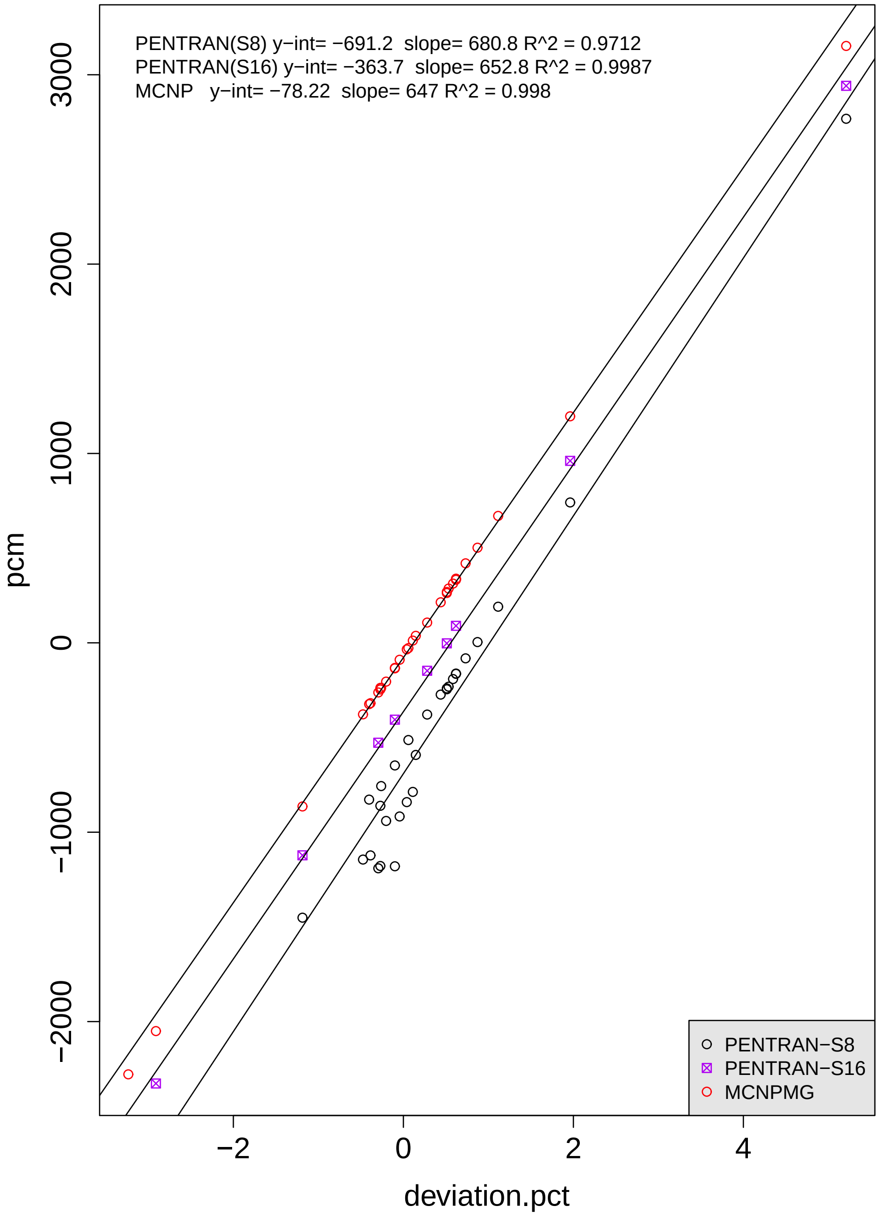


Figure 45: PCM vs. Negative Areal Deviation for PENTRAN (Black S_8 and Purple S_{16}) and MCNP (Red) for the MOX assembly

calculations closely to the reference eigenvalue results generated by Type 2 (MC-Exact-Multigroup)calculations. We avoid Type 3 (MC-Grid-Multigroup) calculations in this *corner fraction* development, because in all likelihood this calculation is not performed except when benchmarking Sn to MC more closely.

The development of the corner (volume) fractioning was done by assuming that the fine meshing was fixed to 10 × 10 over each of the pins. Although we have previously identified that an 11 × 11 is likely a more suitable choice, in 3d assuming the same span in x-y-z coordinates, that would lead to a 33% increase in total mesh. It is more likely the modeler would not increase the z fine mesh extent as there are no curvature issues in z, yet considering just a two-dimensional increase in x-y would still cost a factor of 1.21.

The treatment of homogenization is avoided, since it requires a prior determination or assumption of transport multigroup fluxes used in weighting the cross sections. That is more strongly considered in the prior chapter, but in this chapter we focus on a technique that avoids the use of transport flux, and basically applies a form of volume weighting.

Since it is clear that deviation in eigenvalue is driven by inaccuracies in the mesh grid, we use the corner voxels (in our test problem we have 8) and volume weight the cross sections that compensate for missing target mass. It would be a requirement of this technique that the initial mesh is underreporting the accounting of fuel, since the corner voxels serve as the remainder term to balance between fuel and moderator. Volume weighted cross sections on material interfaces were used and balanced such that total fuel mass and moderator are preserved. Note that again, the C5G7 benchmarks homogenized the fuel and clad together. The corner voxels are shown in Figure 47.

A first attempt with a broader rim around the pin was used shown in Figure 46, but unsurprisingly the transport calculations proved ineffective, since the physics of the mixed cross sections greatly perturbed the system. The full zoom-out view of Figure 47 is given with Figure 48.

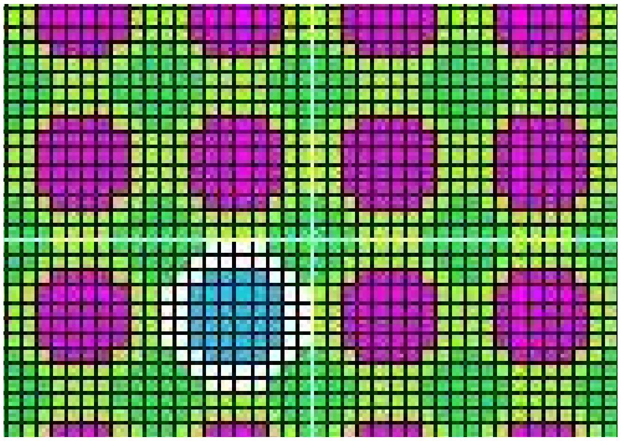


Figure 46: Subset of PWR Assembly with 10×10 Mesh Over Pin, Broad Rim (Not Suitable for Corner Fractioning)

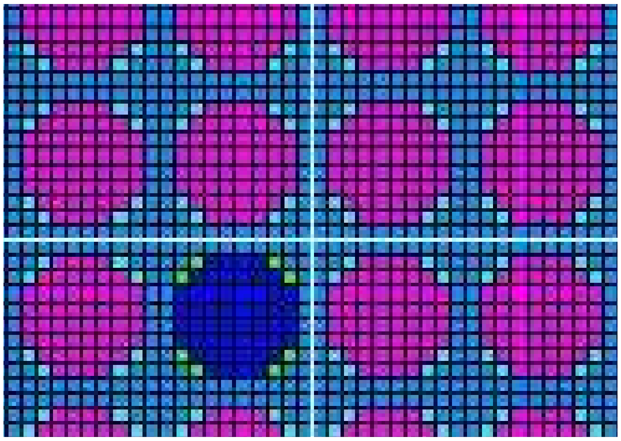


Figure 47: Subset of PWR Assembly with 10×10 Mesh Over Pin, 8 *Corner* Voxels Per Pin

5.4.1 Eigenvalue Results Applied to the UO2 Assembly

The eigenvalue results are tabulated in Tables 11 through 13. In particular, if we presume the dominant material is fuel in the corner pixel (see Figure 47), eigenvalues are lower than the reference, and shown in Table 11. Alternatively, if the dominating corner pixel material is moderator, this raises the eigenvalue above the reference.

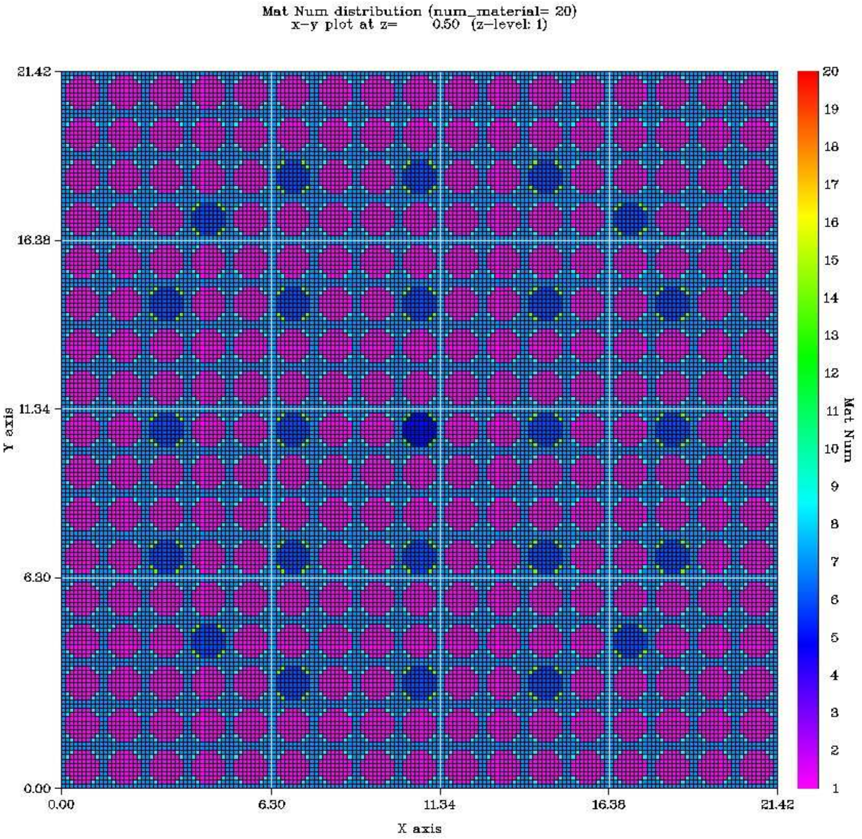


Figure 48: XY View of the Final UO2 Assembly with Corner Fraction Application

When *corner fractioning* is applied, pcm values are significantly reduced to under 100 pcm on average for the cases considered, with a slight positive pcm in all cases; this is conservative and hence our initial testing of this technique is always slightly above the reference eigenvalue.

Table 11: Listing of various PENTRAN Sn Eigenvalue Calculations with *no* Corner Fractioning and Interior Material Dominating Corner Pixel (Reference $k_{\text{eff}} = 1.33314 \pm 0.00006$)

k_{eff}	RBE	CMs	Quadrature	Inner Tol. 10^{-x}	Outer Tol. 10^{-x}	$\Delta k \times 10^5$	pcm
1.324881	4×10^{-4}	1	LS16	3	5	-830	-620
1.325329	4×10^{-5}	4	LS16	3	5	-780	-590
1.329543	4×10^{-3}	16	LS16	3	5	-360	-270

Table 12: Listing of various PENTRAN Sn Eigenvalue Calculations with *no* Corner Fractioning and Exterior/Moderator Material Dominating Corner Pixel (Reference $k_{\text{eff}} = 1.33314 \pm 0.00006$)

k_{eff}	RBE	CMs	Quadrature	Inner Tol. 10^{-x}	Outer Tol. 10^{-x}	$\Delta k \times 10^5$	pcm
1.345784	-2×10^{-4}	1	LS16	3	5	1260	950
1.345989	-1×10^{-4}	4	LS16	3	5	1280	960
1.355348	-1×10^{-4}	16	LS16	3	5	2220	1670

Table 13: Listing of various PENTRAN Sn Eigenvalue Calculations with Corner Fractioning (Reference $k_{\text{eff}} = 1.33314 \pm 0.00006$)

k_{eff}	RBE	CMs	Quadrature	Inner Tol. 10^{-x}	Outer Tol. 10^{-x}	$\Delta k \times 10^5$	pcm
1.336702	1×10^{-4}	1	IC72	4	5	360	270
1.335074	6×10^{-5}	1	IC192	4	5	190	150
1.333821	9×10^{-5}	1	LS16	4	5	70	50
1.333581	2×10^{-4}	1	LS16	4	5	40	30
1.333862	3×10^{-5}	1	LS16	4	6	70	50
1.333840	-2×10^{-5}	1	LS16	5	6	70	50
1.333843	2×10^{-4}	4	LS16	3	5	70	50
1.333914	2×10^{-4}	4	LS16	4	5	80	60
1.334191	4×10^{-5}	4	LS16	5	6	110	80
1.333740	2×10^{-4}	16	LS16	3	5	60	50

CHAPTER VI

ADAPTIVE DIFFERENCING

In this chapter, we discuss tuning of adaptive differencing parameters, for PWR application. We previously discussed adaptive differencing in the introductory chapter; in Section 6.1 we provide further detail of existing differencing schemes in PENTRAN: diamond difference (DD), diamond zero (DZ), directional theta-weighted (DTW), and exponential directional iterative (EDI).

In Section 6.2, we show the benefits of adaptive differencing with a 1d slab problem. The work on the 1d slab benchmarking analysis was performed by myself and contributed to the paper referenced here [52].

6.1 Discussion of Differencing Schemes

Many Sn differencing schemes have been formulated over the years since the introduction of the discrete ordinates (Sn) method by Carlson [11]. One of the challenges of solving diverse deterministic transport problems is that the differencing scheme employed must perform well in diffusive as well as streaming regions, both of which can occur in the same local spatial mesh in the case of neutron transport as neutrons down-scatter. This paper presents an adaptive differencing algorithm with automatic tuning for 3d discrete ordinates applications. While this concept has been introduced before [53, 55, 26, 27], the tuning feature enables the adaptive algorithm to correctly apply the most accurate scheme as changes occur in problem physics. The updated adaptive scheme described here, incorporating the new EDI scheme, was implemented into the PENTRAN parallel discrete ordinates Sn solver in Cartesian Coordinates. The spatial mesh distribution in PENTRAN uses a block coarse mesh/fine mesh discretization, enabling different mesh densities to be specified in localized radiation transport problem geometry. Adaptive mesh refinement, where the spatial mesh is refined to adjust to problem physics, is costly for solvers decomposed on parallel computers; alternatively, PENTRAN adjusts the Sn differencing algorithm on a stable spatial coarse mesh/fine mesh grid. Initially, in any one coarse mesh zone, the adaptive algorithm begins with the traditional diamond difference scheme with a zero negative fixup algorithm, hereafter referred to as Diamond Zero (DZ). If fixups are recorded, the Directional Theta Weighted (DTW) algorithm is selected; then, if gradients become too steep using DTW, the recently introduced Exponential Directional Iterative (EDI) scheme [51] is used. Moreover, mechanisms are in place in the adaptive logic sequence so that the optimal application of a differencing scheme is implemented over the phase space of the problem. In this chapter, we briefly present each scheme in the adaptive sequence, and discuss the details of the adaptive algorithm, including the new tuning logic to indicate how each scheme is selected. Following this, we present a classic model shielding test problem used in our evaluation of differencing schemes, comparing our solutions with multigroup Monte Carlo reference solutions.

The particular format of the multigroup, Legendre expanded form of the 3d Cartesian linear Boltzmann equation is given in Equation 31 [36], with variables briefly summarized in Table 14:

$$\begin{aligned}
\left(\mu\frac{\partial}{\partial x}+\eta\frac{\partial}{\partial y}+\xi\frac{\partial}{\partial z}\right)\psi_g\left(x,y,z,\mu,\varphi\right)+\sigma_g(x,y,z)\psi_g\left(x,y,z,\mu,\varphi\right) = \\
\sum_{g'=1}^{G}\sum_{\ell=0}^{L}(2\ell+1)\sigma_{s,g'\to g,\ell}(x,y,z)\Bigg\{\, P_\ell(\mu)\phi_{g',\ell}(x,y,z)+ \\
2\sum_{k=1}^{L}\frac{(\ell-k)!}{(\ell+k)!}P_\ell^k(\mu)\left[(\phi_C^k)_{g',\ell}(x,y,z)\cos(k\varphi)+(\phi_S^k)_{g',\ell}(x,y,z)\sin(k\varphi)\right]\Bigg\}+ \\
\frac{\chi_g}{k_o}\sum_{g'=1}^{G}\nu\sigma_{f,g'}(x,y,z)\phi_{g',0}(x,y,z)
\end{aligned} \tag{31}$$

Table 14: Multigroup, Legendre Expanded Linear Boltzmann Equation Variables

Variable	Description
μ	x direction cosine for angular ordinate
η	y direction cosine for angular ordinate
ξ	z direction cosine for angular ordinate
ψ_g	group g angular particle flux (for groups g=1,G)
σ_g	total group macroscopic cross section
ℓ	Legendre expansion index ($\ell = 0, L$), $L = 0$ or odd truncation
$\sigma_{s,g' \to g,\ell}$	ℓth Legendre moment of the macroscopic differential scattering cross section from group $g' \to g$
$P_\ell(\mu)$	ℓth Legendre polynomial
$\phi_{g',\ell}$	ℓth Legendre sclar flux moment for group g
$P_\ell^k(\mu)$	lth kth Associated Legendre Polynomial
$(\phi_C^k)_{g'\ell}$	lth kth Cosine Assocated Legendre polynomial for group g
$(\phi_S^k)_{g'\ell}$	lth kth Sine Associated Legendre polynomial for group g
χ_g	group fission distribution constant (neutrons)
k_o	criticality eigenvalue (neutrons)
$\nu\sigma_{f,g}$	gropu fission production (neutrons)

The flux moments $\phi_{g',\ell}$, $(\phi_C^k)_{g',\ell}$, and $(\phi_S^k)_{g',\ell}$ are defined in terms of μ' and φ' as:

$$\phi_{g',\ell} = \int_{-1}^{1} \frac{d\mu'}{2} P_\ell(\mu') \int_0^{2\pi} \frac{d\varphi'}{2\pi} \psi_{g'}\left(x, y, z, \mu', \varphi'\right) \tag{32}$$

$$(\phi_C^k)_{g',\ell}(x, y, z) = \int_{-1}^{1} \frac{d\mu'}{2} P_\ell^k(\mu') \int_0^{2} \pi \frac{d\varphi'}{2\pi} \cos(k\varphi') \psi_{g'}\left(x, y, z, \mu', \varphi'\right)$$

$$(\phi_S^k)_{g',\ell}(x, y, z) = \int_{-1}^{1} \frac{d\mu'}{2} P_\ell^k(\mu') \int_0^{2} \pi \frac{d\varphi'}{2\pi} \sin(k\varphi') \psi_{g'}\left(x, y, z, \mu', \varphi'\right)$$

Allowing for only m discrete directions, a spatial approximation to Equation 31 is required by considering a cell volume has parallelepiped dimensions (Δx,Δy,Δz) , and assuming edge and center flux integrals are represented by surface and volumetric averages, the zeroth moment balance equation in 3d Cartesian coordinates is (omitting group subscripts for brevity):

$$\frac{|\mu_m|}{\Delta x}\left(\psi_{out,x} - \psi_{in,x}\right)\frac{|\eta_m|}{\Delta y}\left(\psi_{out,y} - \psi_{in,y}\right)\frac{|\xi_m|}{\Delta z}\left(\psi_{out,z} - \psi_{in,z}\right) + \sigma\psi_A = q_A \tag{33}$$

Equation 33 contains surface averaged terms enter and leave the cell along each axis using 'in' and 'out' subscripts, respectively, dependent upon the propogation of radiation through the cell along a direction $\hat{\mathbf{\Omega}}_m$, and 'A' subscripts denote cell average quantities. Equation 33 is exact, but contains seven unknowns. Three entrant values ('in' surface averages) are known from boundary values or as exiting fluxes from surrounding cells, and that the collective cell averaged volumetric source q_A is assumed to be known from a previous source iteration (in the standard Sn source iteration scheme). As a result, only the cell averaged angular flux ψ_A and the exiting ('out') surface values are unknowns; these values are derived based on a set of auxiliary equations. For weighted spatial differencing schemes, the following auxiliary equations are assumed to hold between cell average and boundary angular fluxes:

$$\begin{aligned}\psi_{out,x} &= \frac{1}{a}\left(\psi_A + \psi_{in,x}(a-1)\right) \\ \psi_{out,y} &= \frac{1}{b}\left(\psi_A + \psi_{in,y}(b-1)\right) \\ \psi_{out,z} &= \frac{1}{c}\left(\psi_A + \psi_{in,z}(c-1)\right)\end{aligned} \tag{34}$$

The Diamond Differencing (DD) scheme results when a=$\frac{1}{2}$, b=$\frac{1}{2}$, and c=$\frac{1}{2}$ in Equations 34; DD is second order accurate, but may lead to negative solutions [36]. In such situations, a “negative flux set to zero fixup” of the Diamond scheme is commonly used, denoted as Diamond Zero (DZ). In addition to being non-physical, the negative flux fixup can also cause load imbalance during parallel processing. Petrovic and Haghighat developed the Directional Theta-Weighted (DTW) scheme [45] that is a modification of Rhoades and Engles strictly positive Theta Weighted (TW) scheme.

For the DTW scheme, to force positivity for the x-axis term, the lower bound of $\psi_{out,x}$ is strictly zero, and we obtain an equation for the a weight in Equation 34:

$$a = 1 - \frac{q_A + \frac{|\mu_m|}{\Delta x}\psi_{in,x} + \theta(\mu_m)\left(\frac{|\eta_m|}{\Delta y}\psi_{in,y} + \frac{|\xi_m|}{\Delta z}\psi_{in,z}\right)}{\left(2\frac{|\eta_m|}{\Delta y} + 2\frac{|\xi_m|}{\Delta z} + \sigma\right)\psi_{in,x}} \tag{35}$$

A similar procedure is applied along the y- and z-axes to yield weights for b and c in Equation 34. The cell averaged angular flux for the DTW scheme is given by [26, 54]:

$$\psi_A = \frac{q_A + \frac{|\mu_m|}{a\Delta x}\psi_{in,x} + \frac{|\eta_m|}{b\Delta y}\psi_{in,y} + \frac{|\xi_m|}{c\Delta z}\psi_{in,z}}{\frac{|\mu_m|}{a\Delta x} + \frac{|\eta_m|}{b\Delta y} + \frac{|\eta_m|}{c\Delta z} + \sigma} \tag{36}$$

The auxiliary equation for the Exponential Directional Iterative (EDI) scheme is formed initially from an exponential auxiliary equation [51]:

$$\psi_m(x,y,z) = a_o \exp\left(\lambda_i \frac{P_1(x)}{|\mu_m|}\right)\exp\left(\lambda_j \frac{P_1(y)}{|\eta_m|}\right)\exp\left(\lambda_k \frac{P_1(z)}{|\xi_m|}\right) \tag{37}$$

where $P_1(u) = (2u/\Delta u - 1)$ are first order spatial Legendre functions (orthogonal to 0^{th} order equations with unit weight) over the widths of a single cell. EDI is a predictor-corrector scheme where angular fluxes from the DTW scheme are used to start a fixed point iteration to refine each exponential constant $\lambda_i, \lambda_j, \lambda_k$ by *successive iterations* $(I-1, I, I+1\ldots)$ of Equations 38:

$$\begin{aligned}
\lambda_{i,I} &= f(\lambda_{i,I-1}) = \frac{(\psi_{out,x} - \psi_{in,x})|\mu_m|}{2\psi_A(\lambda_{i,I-1})} \\
\lambda_{j,I} &= g(\lambda_{i,I-1}) = \frac{(\psi_{out,y} - \psi_{in,y})|\eta_m|}{2\psi_A(\lambda_{j,I-1})} \\
\lambda_{k,I} &= h(\lambda_{i,I-1}) = \frac{(\psi_{out,z} - \psi_{in,z})|\xi_m|}{2\psi_A(\lambda_{k,I-1})}
\end{aligned} \tag{38}$$

These exponential constants are converged through successive iteration, typically two; the fixed point iteration will remain convergent on a finite, nonzero interval $[p, q]$ by adhering to the first derivative criterion in Equations 39:

$$\left|\frac{\partial f(\lambda_{i,I-1}}{\partial \lambda_{i,I-1}}\right| < 1 \qquad \left|\frac{\partial g(\lambda_{j,I-1}}{\partial \lambda_{j,I-1}}\right| < 1 \qquad \left|\frac{\partial h(\lambda_{k,I-1}}{\partial \lambda_{k,I-1}}\right| < 1 \tag{39}$$

The EDI volume averaged angular flux is given in Equations 40 and 41

$$\begin{aligned}\psi_A = &\left(\exp\left(\frac{2\lambda_i}{|\mu_m|}\right) - 1\right)\left(\exp\left(\frac{2\lambda_j}{|\eta_m|}\right) - 1\right)\left(\exp\left(\frac{2\lambda_k}{|\xi_m|}\right) - 1\right) \\ &\cdot \frac{1}{\beta}\left(q_A + \frac{|\mu_m|}{a\Delta x}\psi_{in,x} + \frac{|\eta_m|}{b\Delta y}\psi_{in,y} + \frac{|\xi_m|}{c\Delta z}\psi_{in,z}\right)\end{aligned} \tag{40}$$

with the β term given as:

$$\begin{aligned}\beta = &\frac{2\lambda_i}{\Delta x}\left(\exp\left(\frac{2\lambda_i}{|\mu_m|}\right)\right)\left(\exp\left(\frac{2\lambda_j}{|\eta_m|}\right) - 1\right)\left(\exp\left(\frac{2\lambda_k}{|\xi_m|}\right) - 1\right) \\ &+ \frac{2\lambda_j}{\Delta y}\left(\exp\left(\frac{2\lambda_j}{|\eta_m|}\right)\right)\left(\exp\left(\frac{2\lambda_i}{|\mu_m|}\right) - 1\right)\left(\exp\left(\frac{2\lambda_k}{|\xi_m|}\right) - 1\right) \\ &+ \frac{2\lambda_k}{\Delta z}\left(\exp\left(\frac{2\lambda_k}{|\xi_m|}\right)\right)\left(\exp\left(\frac{2\lambda_i}{|\mu_m|}\right) - 1\right)\left(\exp\left(\frac{2\lambda_j}{|\eta_m|}\right) - 1\right) \\ &+ \sigma\left(\exp\left(\frac{2\lambda_i}{|\mu_m|}\right) - 1\right)\left(\exp\left(\frac{2\lambda_j}{|\eta_m|}\right) - 1\right)\left(\exp\left(\frac{2\lambda_k}{|\xi_m|}\right) - 1\right)\end{aligned} \tag{41}$$

Outbound fluxes can be defined in terms of the average angular flux:

$$\begin{aligned}\psi_{out,x} &= \psi_A \frac{2\lambda_i}{|\mu_m|}\left(1 - \exp\left(\frac{-2\lambda_i}{|\mu_m|}\right)\right)^{-1} \\ \psi_{out,y} &= \psi_A \frac{2\lambda_j}{|\eta_m|}\left(1 - \exp\left(\frac{-2\lambda_j}{|\eta_m|}\right)\right)^{-1} \\ \psi_{out,z} &= \psi_A \frac{2\lambda_k}{|\xi_m|}\left(1 - \exp\left(\frac{-2\lambda_k}{|\xi_m|}\right)\right)^{-1}\end{aligned} \tag{42}$$

Therefore, the EDI scheme is implemented using an initial starting guess from

DTW, with successive applications of Equations 38, 40, 41 and 42 applied in accordance with Equations 39 to yield a stable fixed-point iteration to solve for more accurate values of each exponential constant $\lambda_i, \lambda_j, \lambda_k$. The *adaptive differencing* strategy in PENTRAN, determined individually for each coarse mesh zone, works in the following manner: assume (for illustration) that the DZ scheme is initially assigned in each coarse mesh region containing a number of fine meshes. An automatic differencing scheme upgrade from DZ to DTW takes place if a negative flux fixup is encountered anywhere in the coarse mesh. This is potentially followed later by another transfer from DTW to EDI if any maximum DTW linear weight factor (a,b,c) exceeding a user specified maximum weight factor is detected for DTW within a coarse mesh. This is performed since a high weight factor indicates DTW is being pressed to maintain positivity (at the expense of accuracy) in a severe streaming environment, so that the shift to EDI enables an exponential treatment for cells that are optically thick, since these scenarios are best handled using the exponentially based auxiliary function of the EDI scheme.

A novel “Tuning Feature” handles exceptions to the standard adaptive strategy for the Adaptive differencing strategy, and this feature calls for special provisions for optically thin cells, as well as evaluation of the angular source density relative to the total angular collision density in a parameter called the *qfratio*. First, consider that $\Delta h_{max} = max(\Delta x, \Delta y, \Delta z)$ is computed to evaluate the optical cell thickness $\sigma \Delta h_{max}$; for a given coarse mesh, if this optical thickness is less than a user specified value (e.g. 0.02, used as the default in PENTRAN), then for these “vanishingly thin” cells, the DTW scheme is not upgraded to the EDI scheme for any circumstance, since a very small optical thickness is such that DTW is perfectly adequate to accurately resolve the angular flux, and an exponential treatment is not needed.

Moreover, as mentioned, upgrades from DTW to EDI are based on any DTW weight factor (a,b,c) exceeding a value approaching unity (e.g. a default of 0.95 in

PENTRAN), where in a streaming situation with no or low-level sources, weight factors near unity are required to maintain positivity at the expense of accuracy. However, high weight factors can also occur in any mesh cell that contains a strong source, simply because the angular flux in these situations is shallow or relatively flat (resulting in low angular flux gradients)this is the opposite of the streaming case. Therefore, with a strong source present, this leads to DTW weight factors close to unity, and causes a conflict with the standard upgrade criteria; this is undesirable, since the DTW scheme performs very well in regions where there is a strong source - DTW weight factors are close to unity because the flux is inherently flat. Therefore, if the angular flux is inherently flat due to the presence of a strong source, a step scheme would be very effective - the "step" scheme results algebraically if the weights are set to unity along each direction for DTW. This scenario has been mitigated for the adaptive differencing methodology in PENTRAN through the use of the qfratio parameter.

Considering the group dependent transport equation divided through by the collision density term, with group sources denoted using scatter, fission, and independent source variables (previously referred to in Equation 7):

$$\frac{\hat{\mathbf{\Omega}} \cdot \nabla \psi_g}{\sigma_g \psi_g} + 1 = \frac{(q_{scat,g} + q_{fiss,g} + q_{indpt,g})}{\sigma_g \psi_g} = \mathit{qfratio} \tag{43}$$

The *qfratio* is the computed ratio of the cell total angular source density to the cell angular collision density; a $qfratio > 1.00$ indicates a "source dominated" cell, and a $qfratio < 1.00$ indicates a "streaming dominated" cell, where "source" includes the combined scatter, fission, and independent angular source terms. Note this simple relationship is readily available when solving for angular fluxes within each cell. If the *qfratio* is greater than a user prescribed value (the default for $qfratio = 1.00$ in PENTRAN), then the DTW scheme is automatically selected without regard to the

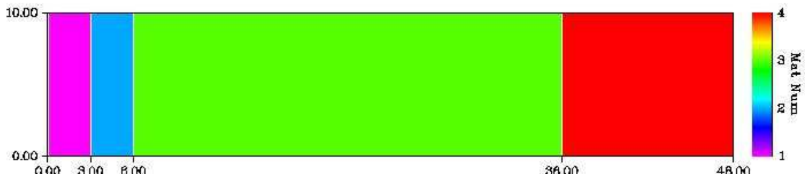


Figure 49: Fixed Source Slab Model Problem Coarse Mesh Cells, Numbered 1 to 4, left to right

DTW weight factors, since in a source dominated cell, the DTW scheme performs optimally, and exponential treatment with EDI is not applicable. Therefore, with the *qfratio* parameter, upgrades will only occur when the fine mesh cell is one that is “streaming dominated” away from source regions.

6.2 Numerical Testing with Slab Problem

To test the adaptive scheme, we employed two model problems. This model applies a slab geometry fixed source differencing scheme test problem originally used in a paper by Alcouffe, et al [4], which contains a mix of both streaming and diffusive regions. This is an excellent problem for evaluating the performance of our adaptive scheme with tuning.

6.2.1 Slab Model Problem: Detailed Description

A schematic of the slab fixed source Model Problem 1 depicting the four coarse meshes is given in Figure 49. This problem is subdivided into 4 coarse meshes, with each coarse mesh zone numbered sequentially from left to right; the first zone is a half-scattering region; the second zone has a source and is a pure absorber; the third zone contains a typical shield; the fourth zone is a diffusive region. Unit density sources are placed in the second (absorbing) and fourth (diffusive) coarse mesh regions; the problem is set up to use reflective boundaries for both y- (vertical) and z- axes (out of the page), with vacuum boundaries on either end (x-axis). The four coarse mesh boundaries span the x-axis from [0, 3.0, 6.0, 36.0, 48.0] cm.

Table 15: Slab Model Problem One-Group Cross Sections (P_0), Source Terms (1/cm units)

Half Scatterer			
σ_a	$\nu\sigma_f$	σ_t	σ_s
0.5	0.00	1.00	0.500
Source (1 n/cc/s) + Pure Absorber			
σ_a	$\nu\sigma_f$	σ_t	σ_s
1.0	0.00	1.00	1×10^6
Shield			
σ_a	$\nu\sigma_f$	σ_t	σ_s
0.95	0.00	1.00	0.500
Source (1 n/cc/s) + Diffusive Region			
σ_a	$\nu\sigma_f$	σ_t	σ_s
0.05	0.00	1.00	0.9500

Cross sections for this problem in each respective zone are given in Table 15. Again, material 1 is a 3.0 cm thick half-scattering material; material 2 is a 3.0 cm thick pure absorber with a source; material 3 is 30.0 cm thick shield material; material 4 is a 12.0 cm diffusive region with a source.

To be consistent with the problem posed, a reference mesh in each region (the number of meshes in the x-direction) was specified (60,60,600,240) for each coarse mesh, with the DZ algorithm, respectively. The quadrature specified for this problem was S_8, with an inner convergence tolerance of 1×10^{-5}. In addition, two single group MCNP5 models were executed, the first with no adjoint weighting, and the second with adjoint weighted weight windows, to yield an independent solution metric for comparison.

In comparison to the aforementioned reference mesh case in PENTRAN, the mean volumetric cell fluxes by material region, normalized to the number of source MCNP particles, were calculated for materials 1 through 4. Table 16 offers the percent error for two cases run in MCNP. The first case is a multigroup MCNP5 run mode

Table 16: Percent error by material/CM Number (PENTRAN reference mesh Sn case is error basis)

MCNP Case	M1 % error	M2 % error	M3 % error	M4 % error	Max Cell Flux R.E.	F4 Tally M3 FOM
Multigroup	-1.216%	0.311%	-0.407%	0.134%	0.001	147104
Multigroup WWINP Adjoint	-1.258%	0.288%	-0.401%	0.138%	0.0023	211977

using the MGOPT (multigroup) option (without adjoint weighting). The second case employs adjoint weighted weight windows using the WWINP option in multigroup mode (MGOPT). The adjoint weighted weight windows were generated with PENTRAN using the reference mesh in adjoint mode, with automated WWINP generation performed with an auxilary code written in Fortran called PENIMP. Running with adjoint weighted weight windows offers the advantage of a higher figure or merit (FOM) and improved overall relative error for a fixed computer time (or even for a fixed number of histories, as shown for 1 × 10^8 histories). The maximum MCNP5 cell/material F4 flux tally error is also reported.

6.2.1.1 1/30th Mesh Solution to Model Problem

The problem was solved on 1/30th of the reference mesh using fixed schemes DZ, DTW, and the Adaptive scheme. In these cases, as shown in Figure 50, the adaptive differencing scheme was the most accurate scheme.

6.2.1.2 1/60th Mesh Solution to Model Problem

The Slab Model Problem was solved on 1/60th of the reference mesh using fixed schemes DZ, DTW, EDI, and the Adaptive scheme; this corresponds to a single fine mesh each in Coarse Mesh zones 1 and 2, with 10 fine mesh cells in Coarse Mesh 3, and 4 fine mesh cells in Coarse Mesh 4. The globally fixed DZ scheme did not

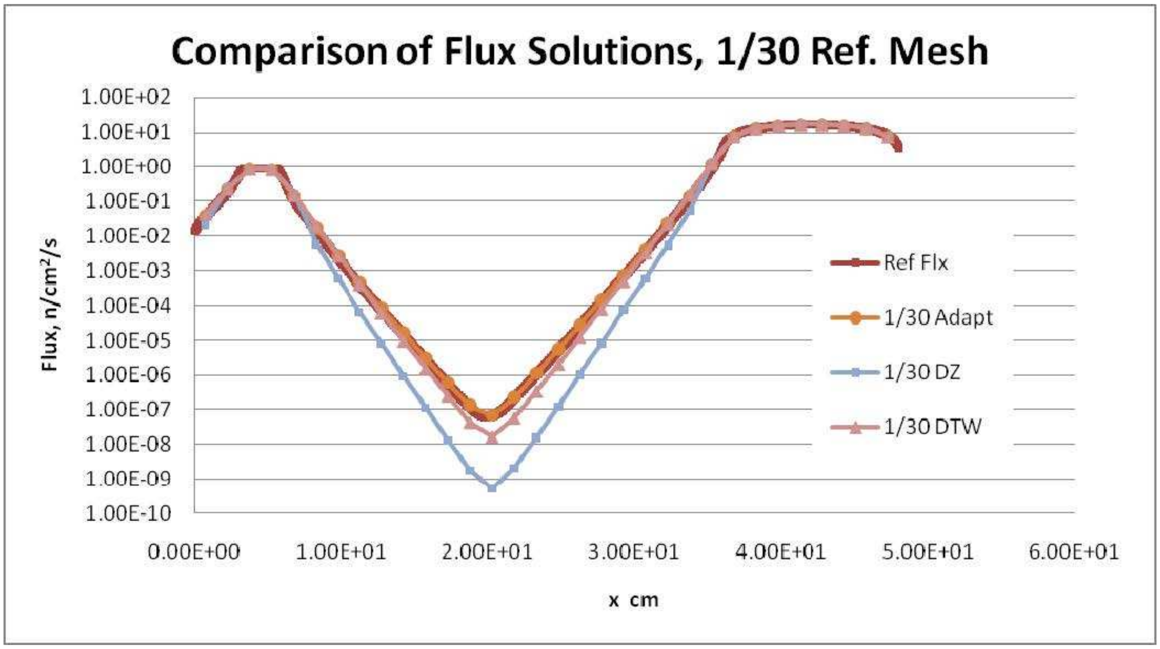


Figure 50: Flux Solutions for 1/30th of the reference mesh

converge at all with this low level of discretization, and had 100% error after hundreds of iterations. The DTW fixed scheme solution converged in 65 iterations, but over-predicted the flux by two orders of magnitude; the EDI-only rendered solution converged in 67 iterations, and upon observation was closest to the Adaptive solution. However, the Adaptive scheme solution was the most accurate, and converged in 66 iterations; for the adaptive scheme, Coarse Meshes 1 and 3 ended up using the EDI scheme, while Coarse Meshes 2 and 4 ended up remaining with the DZ scheme, as no fixups were needed. Again, the ideal nature of this concept is that the scheme applied is via code logic.

To more effectively highlight the differences between the methods on the 1/60th mesh models, it is useful to compare errors in fluxes averaged over each of the Coarse Meshes, achieved by comparing with fluxes from the high resolution reference mesh solution. In doing so, a relative error computed according to

$$\epsilon = \left| \frac{\phi_{TEST} - \phi_{REFERENCE}}{\phi_{REFERENCE}} \right| \tag{44}$$

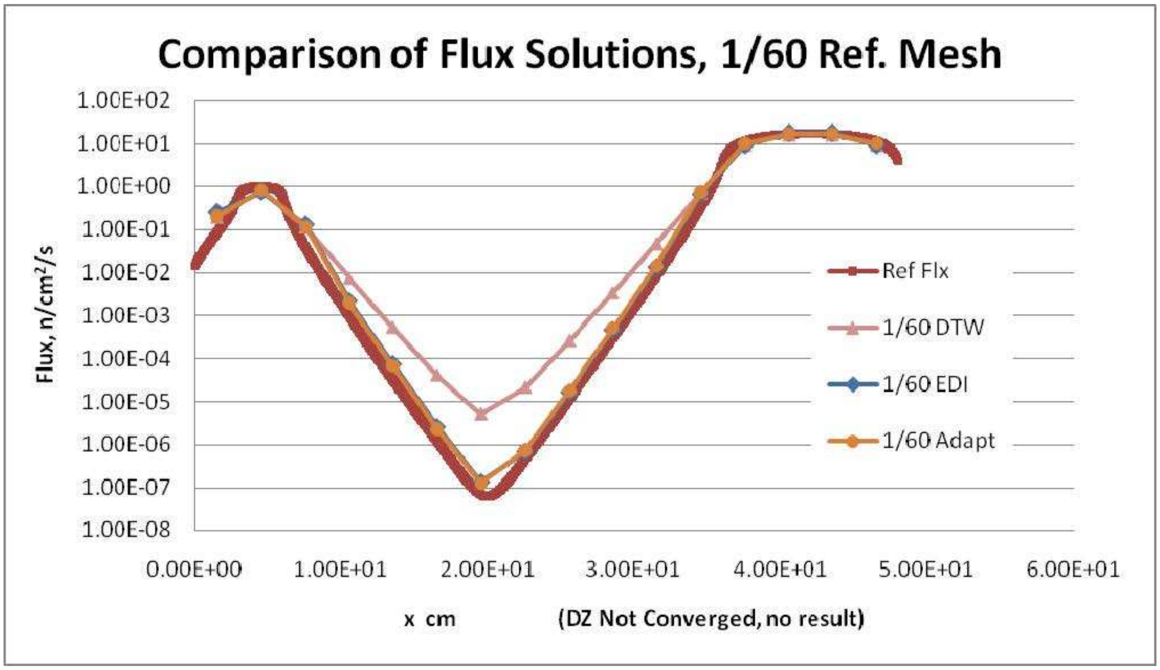


Figure 51: Flux Solutions for 1/60th of the reference mesh

was used to compare Coarse Mesh Average fluxes. In this comparsion, mean relative errors for the various differencing schemes were: Adaptive (11% error), DZ (100% error), DTW (13% error), and EDI (16% error). While the EDI scheme is highly effective in shielding problems due to an exponential based auxiliary formulation, this is not the best scheme to apply unilaterally when strong sources are present, since the natural updward concavity of the EDI scheme does not represent the flux well in source dominated regions.

6.3 Adaptive Differencing Tuning Using Two PWR Assembly Test Problems

In this section, we identify general principles for optimal differencing selection with 3d Sn PENTRAN. Two problems, based on the usual PWR assembly eigenvalue problem (corner fraction PWR mode), each problem having 16 CMs, are considered. One problem applies all reflective BCs in one coarse mesh, and the second problem applies mixed reflective and vacuum BCs. The next paragraphs detail each problem at hand. As it will be demonstrated by this analysis, the best tuning is done by setting locked differencing schemes, because doing so avoids branch condition checking in the PENTRAN code. This may appear to belie the point of adaptive differencing, but locked/preferred difference settings will always be superior because of the increased efficiency in avoidance of 'if' conditions that would check to upgrade the scheme, one-way, from DZ to DTW to EDI. This should not deter the effort in improving default adaptive difference settings, for which we provide a recommendation at chapter's end.

Unless otherwise indicated, all Sn problems were run with default tolerances, inner tolerance 1×10^{-3} and outer tolerance 1×10^{-5}, and level symmetric S_{16} quadrature, with a maximum of 10 inner iterations per outer iteration, 400 maximum total outers unless outer convergence satisfied. Other advanced quadratures, like IC, LDFE-SA, or EO, were not elected for use in this analysis, simply because we wish to fix the quadrature in this case study. The cross sections, again are 7 group P_0.

The first problem and typical Sn results are shown in Figure 52, which is just the corner-fraction PWR assembly model applied towards the end of Chapter 5, but with the removal of all pins in the 11th (of 16) coarse mesh. The Monte Carlo (MCNP multigroup) is taken to be the reference for comparison with various adaptive tuning variations, in particular, variation of two adaptive settings, *dtwmxw* (ranged from 0.5 to 1.0), and *qfratio* (ranged from 0 up to 1). The reference MCNP runs yield 1.30721 ± 0.00004, generated with 100,000 particles per cycle, skipping 50 cycles for a total of 2050 cycles.

The second problem and typical Sn results are shown in Figure 53, which removes all pins from CMs 11, 12, 15, and 16, in addition to applying vacuum boundaries on the +x and +y global sides. This has the effect of bringing the eigenvalue subcritical. Reference MCNP runs yield 0.95531 ± 0.00005, generated with the same parameters as the first problem.

6.3.1 Design Benchmark Results

A myriad of differencing schemes were applied and cases are enumerated with Figures 54 and 55. The intent is to have an all EDI model, an all DTW model, selective EDI in the water region with DTW otherwise, and adaptive (marked with the letter A).

In the experience with running the Water Hole 1 design problem, the comparison between PENTRAN and MCNP yields extremely consistent results, however, upgrade to EDI seemed to occur with default settings, and in fact, may not be the best application given the results in Table 17. The use of locked DTW would be favored, since generally all coarse meshes do not experience significant streaming. Generally, though in all cases pcm error is 80.3 (worst case) and average coarse mesh flux error is no worse than 0.273%. The calculation of average coarse mesh flux error was done by taking tallies of each of the 16 coarse meshes, per group, for a total of 112 points for which the error was tabulated and averaged. By increasing the default dtwmxw weight to 0.99, we withhold the upgrade to EDI, which works well for this problem. The locked DTW scheme also happens to be the best choice for this problem, with near match on pcm and average CM flux error of 0.219%.

With the Water Hole 2 design problem, application of EDI makes more sense, as the presence of the vacuum boundary conditions, in addition to the increased water region. This effect is present with the group flux plots shown in Figure 53. Because of the more significant streaming effect, the minimized pcm error is achieved with the all locked EDI case.

The tabled results indicate that setting locked difference schemes is superior to adaptive difference. This is simply because no time is spent deciding on the upgrade. Also, it is recommended to apply a higher default *dtwmxw* weight, from 0.95, to higher than 0.98, simply to rely more on the use of DTW before upgrading to EDI, especially if it is suspected that the problem is sufficiently meshed, and the particular problem at hand does not have "extreme" gradients. In all cases, PENTRAN compares very

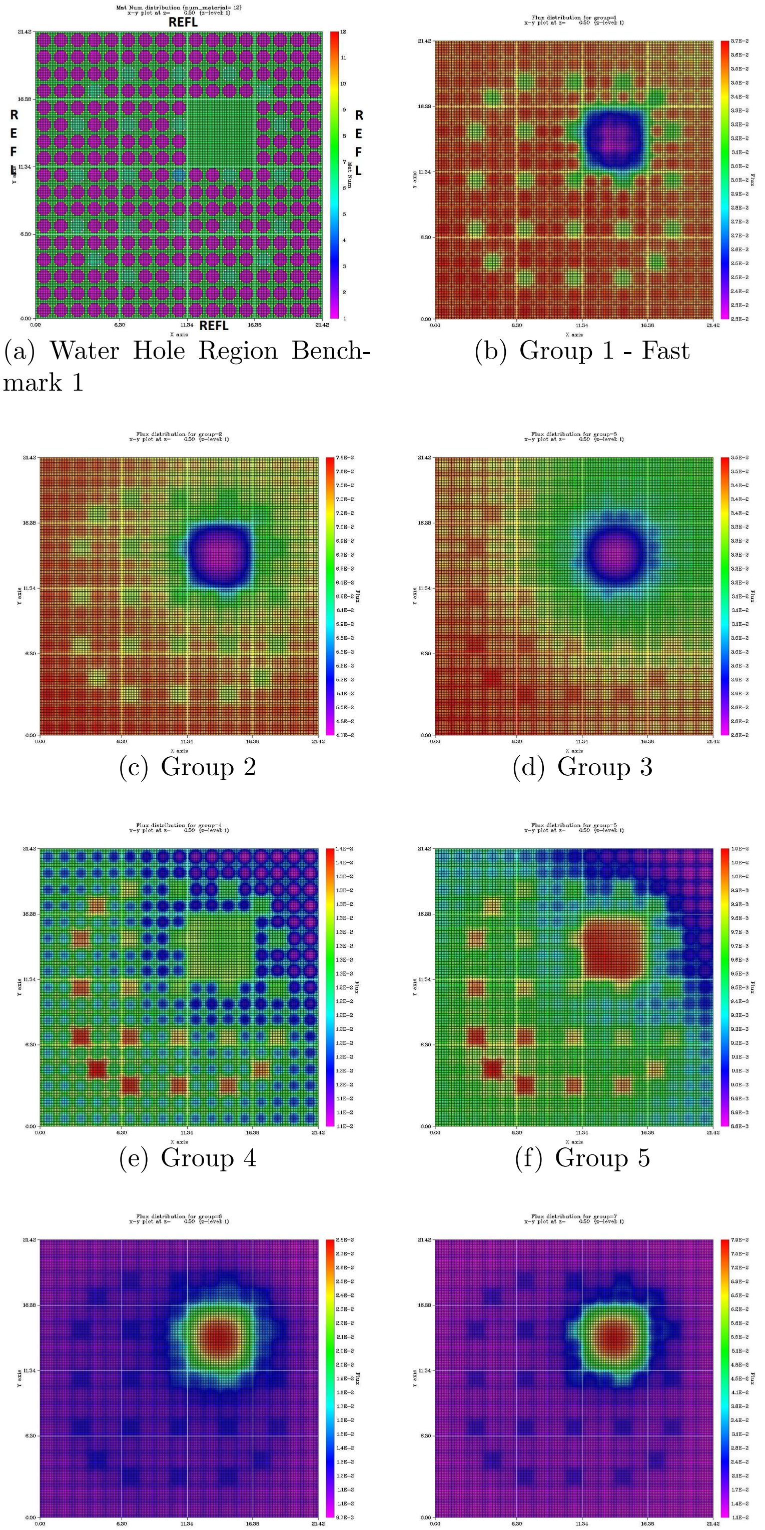
Table 17: Water Hole Benchmark 1 PENTRAN Results (Reference MCNP Eigenvalue 1.30721 ± 0.00004

Case	pcm Error	Inner/Outer Total Iters.	Time (s)	Average CM Flux Error	Note
All Locked EDI	78	2822/225	1349	0.273%	
All Locked DTW	1.3	2887/136	870	0.219%	
Adaptive (A) (dtwmxw=0.95)	80.3	4693/231	1505	0.272%	Upgraded to EDI
Adaptive (A) (dtwmxw=0.99)	34.0	3155/227	998	0.272%	Stayed with DTW
One EDI in water	11.7	3229/205	1313	0.192%	

Table 18: Water Hole Benchmark 2 PENTRAN Results (Reference MCNP Eigenvalue 0.95531 ± 0.00005

Case	pcm Error	Inner/Outer Total Iters.	Time (s)	Average CM Flux Error	Note
All Locked EDI	7.22	2634/232	1265	0.273%	
All Locked DTW	33.29	2581/241	957	0.181%	
Adaptive (A) (dtwmxw=0.95)	9.11	4042/197	924	0.276%	Upgraded to EDI
Adaptive (A) (dtwmxw=0.99)	32.35	2607/237	940	0.269%	Stayed with DTW
Four EDI in Water	22.61	2437/211	850	0.173%	

well to Monte Carlo, with less than 0.3% average coarse mesh flux error and most cases having 35 pcm or smaller error.



(a) Water Hole Region Benchmark 1

(b) Group 1 - Fast

(c) Group 2

(d) Group 3

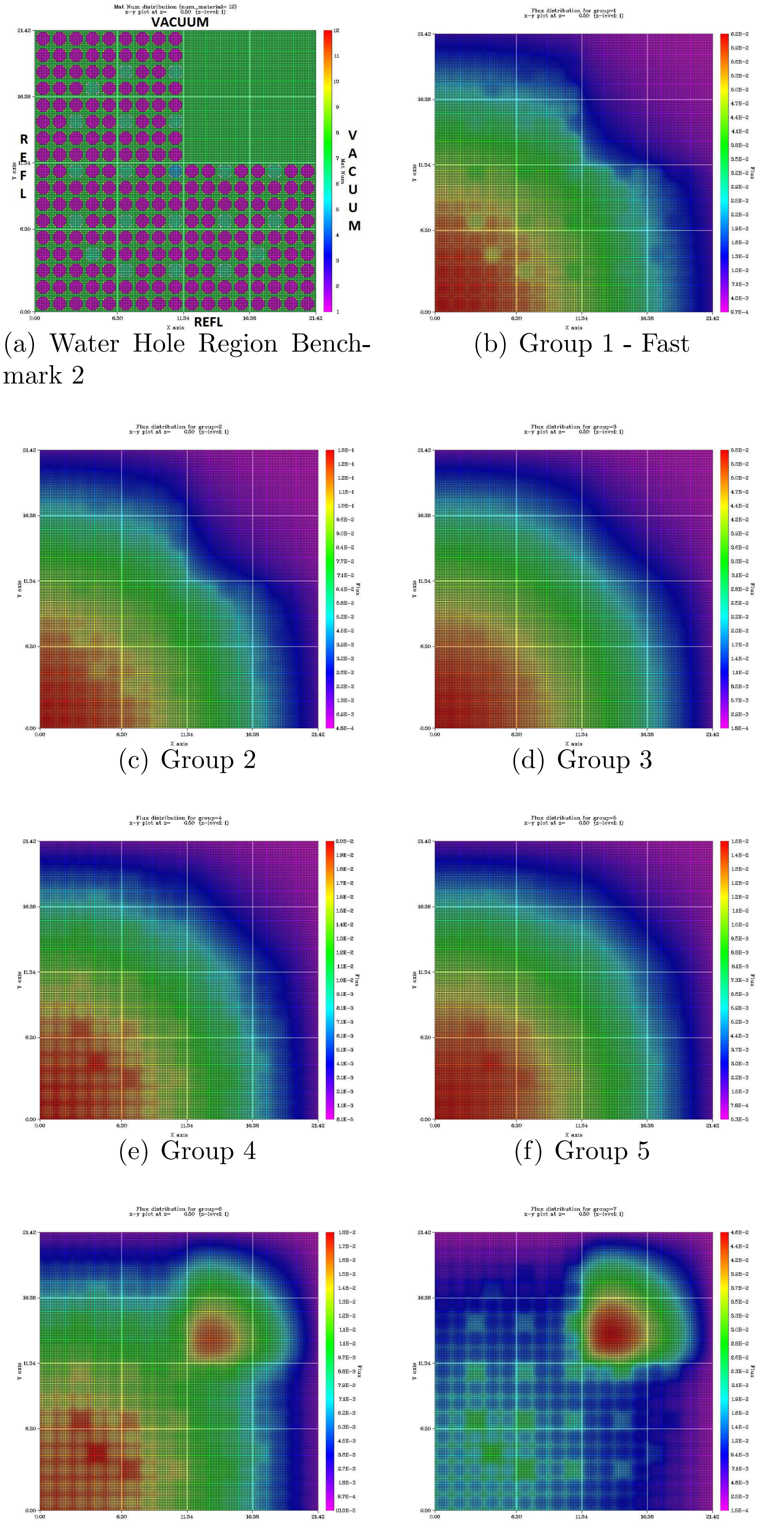
(e) Group 4

(f) Group 5

(g) Group 6

(h) Group 7 - Most Thermal

Figure 52: 2D Material and Group Flux Visualization of Water Hole Benchmark 1 with All Sides Reflective.



(a) Water Hole Region Benchmark 2

(b) Group 1 - Fast

(c) Group 2

(d) Group 3

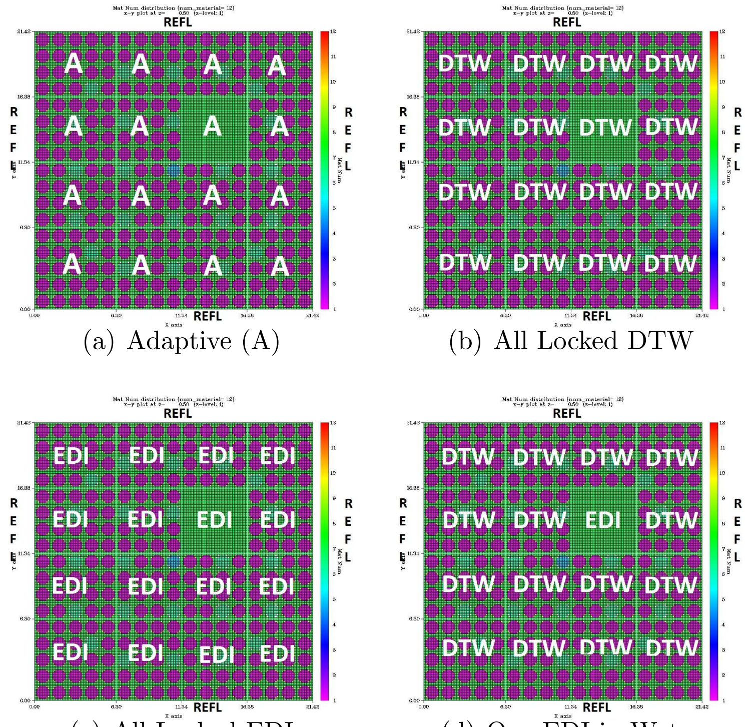
(e) Group 4

(f) Group 5

(g) Group 6

(h) Group 7 - Thermal

Figure 53: 2D Material and Group Flux Visualization of Water Hole Benchmark 2 with Reflective BCs (-x, -y, -z, +z) and Vacuum BCs (+x, +y).



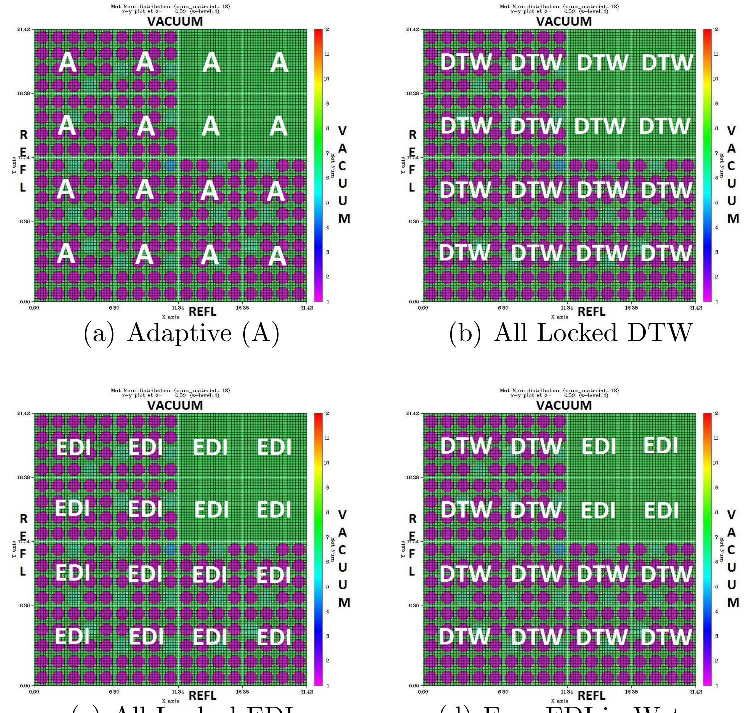
(a) Adaptive (A)

(b) All Locked DTW

(c) All Locked EDI

(d) One EDI in Water

Figure 54: Difference Scheme Selections for Water Hole Benchmark 1



(a) Adaptive (A)

(b) All Locked DTW

(c) All Locked EDI

(d) Four EDI in Water

Figure 55: Difference Scheme Selections for Water Hole Benchmark 2

CHAPTER VII

PARALLELIZATION OPTIMIZATION WITH HYBRID OPENMP-MPI

In this chapter, we implemented and analyzed the use of sub-octant angular domain decomposition by application of OpenMP within an MPI parallel discrete ordinates transport code. We will demonstrate modest gains achieved by use of hybrid MPI-OpenMP for simplified 3-group Pressurized Water Reactor (PWR) pin problem. Preliminary conclusions based on the present data show that speedups of up to 1.31 were achieved, relative to pure MPI, for a small PWR pin problem, and a speedup of 1.25 for a larger PWR pin problem. From a computational and high performance computing (HPC) point-of-view, we start with a highly start with the MPI-based PENTRAN (Parallel Environment Transport), and adapt the code to use OpenMP.

In particular, the angular decomposition in PENTRAN works optimally on a per-octant basis, which can allow for 8 octants/tasks for MPI parallelization (this analysis was limited to angular decomposition, ignoring available features of space and energy decomposition). While further sub-octant handling was treated by initial design, the parallelization benefit stopped with 8 octants, since the MPI communications of angular flux across coarse mesh boundaries were handled by octant only. Thus, further breakdowns in sub-octant gained no additional parallel speedup benefits. As ordinate sweeping has been demonstrated in using roughly 50 percent of the wall clock time (Section 7.3.2.1 and Table 21), additional parallelization could occur because the ordinate loop in the Fortran **SWEEPM** routine is vectorizable and has no data dependencies.

PENTRAN uses Gauss-Seidel sweeping, which has a inherently sequential data dependency that occurs with structured mesh cells along x,y, and z axes, hence, no additonal parallelization was considered here. Prior to OpenMP implementation, the parallelization was limited to 1,2,4, or 8 MPI tasks in parallel in practice. Extension beyond 8 MPI tasks is achieveable with PENTRAN, however, message passing is performed on an octant basis so communication will limit parallel benefit. Because of the identification of the sub-octant parallelization, use of OpenMP was identified as a suitable candidate because of the additional parallelization benefit achieved. Also, present parallel computer clusters are predominantly cache-coherent non-uniform memory access (ccNUMA). Hence, each MPI task in angular decomposition can spawn a number of threads which are typically a divisible factor of the selected quadrature.

Angular quadrature refers to the weights and associated unit direction cosines for use in integration, each ordinate having an associated angular flux. See Figure 56 for a depiction of S_8 quadrature. The number of ordinates is simply determined by the formula N (N + 2) = 8. For example, S_8 quadrature, contains 10 ordinates per octant, making 1,2,5, or 10 natural thread number choices for an MPI task. Therefore, a maximum of 80 CPUs for S_8 quadrature could potentially be used. When the number of threads does not evenly divide, the OMP DYNAMIC option was used. Otherwise, static scheduling was used by default in OpenMP.

In order to reasonably manage the problem scope, we select a PWR Pin with reflective BCs (Figure 57) and vary either mesh sizing (small/large) and quadrature (S_8 / S_{30}) to solve for the eigenvalue as well as the angular group fluxes / scalar fluxes. For this chapter analysis, this resulted in two problem types using the same basis problem: S_8 /small and S_30 /large. In particular, the pin was modeled using a single coarse mesh, hence, there was no investigation in spatial decomposition. Also, since PENTRAN has the capability of using energy decomposition across groups, that option was not used for the OpenMP analysis (no group decomposition). At the time of first OpenMP implementation, the differencing scheme utilized a Directional-Theta

Table 19: PWR pin parameters relevant to parallelization

Case	Quadrature	Ordinates Per Octant	Meshes	Size	Analyzed
I	8	10	9680	Small	X
II	30	10	9680	Small	
III	8	120	106480	Large	
IV	30	120	106480	Large	X

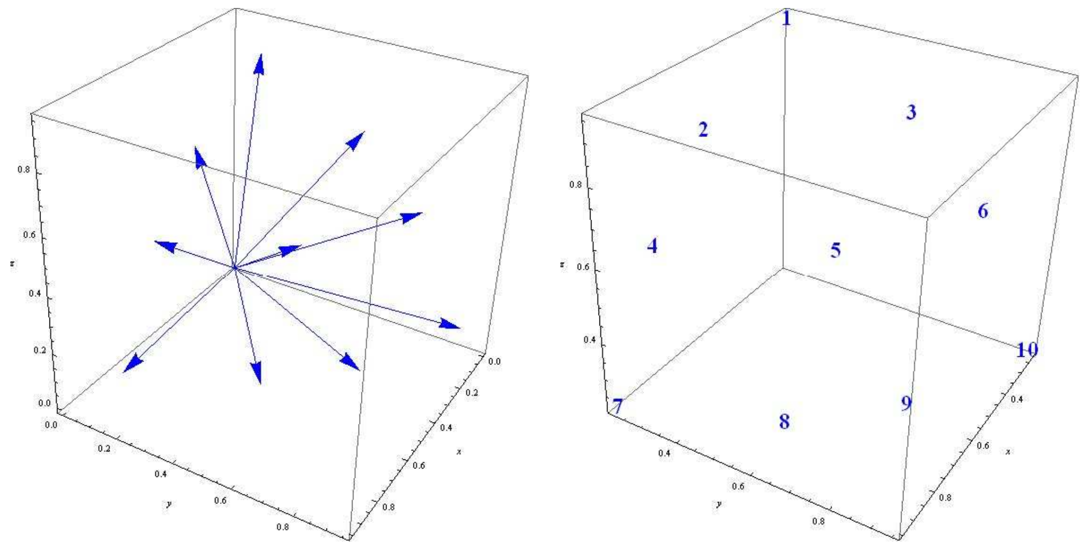
Table 20: Description of Critcel Architecture AMD Opteron 6274

Nodes	Sockets Per Node	CPUs	Total CPUs	Memory
7	4	16	64 × 7	256GB×7

Weighted Scheme (DTW), one of several schemes. If utilizing more than a single coarse mesh, PENTRAN is effectively performing a Block Jacobi iterative scheme. Problem parameters are described in Table 19 and the Critcel cluster hardware information is shown in Table 20.

7.1 Discrete Ordinates Transport in 3D Cartesian Coordinates via Finite Difference and Gauss-Seidel Sweeps: Sweep Algorithm

In order to clarify the algorithm, we provide a brief discussion of ordinate sweeping. The transport problem can be thought of as a Gauss-Seidel sweep following the entering ordinate with incoming known angular fluxes . That is, a transport problem is solved by sweeping across a grid based on the entering ordinate (where all ordinates in an octant have the same directional entry). For example, an ordinate having direction cosines all positive $(+\mu, +\eta, +\xi)$ start with a fine mesh in the most negative corner, sweeping from the origin to the most positive corner. For structured grids, this is done by moving from fine mesh to fine mesh along the x axis, then the y axis, then the z axis, where boundary conditions are transferred in via ghost arrays that hold the results of angular fluxes on the coarse mesh boundaries. Hence, the



(a) S_8 quadrature in one octant with 10 ordinates

(b) OpenMP Threads can be assigned to each ordinate within octant

Figure 56: S_8 Level-Symmetric Quadrature

angular flux is solved for 3 sides of the outgoing hexahedral mesh based on 3 known input angular fluxes on the incoming sides. There are no data dependencies between ordinates, which means that simultaneous finite difference sweeps can occur for each ordinate. Each respective ordinate is given a label from 1 to 10 (as S8 quadrature has 10 ordinates per octant for a total of 80 ordinates, see Figure 56). Figure 58 provides a pseudocode of the SWEEP subroutine.

7.2 Estimate of Parallelization Benefit in Using OpenMP by Application of Amdahl Fraction

The following equation measures speedup, given a parallelizable fraction f, and the number of processors p

$$speedup = \frac{T_{serial}}{T_{parallel}} = \frac{1}{(1-f)+\frac{f}{p}} \tag{45}$$

As an example, with most of our test cases having roughly 50% of time spent in sweeps (see Section 7.3.2.1) , the use of $2, 3, 4,$ and 5 processors results in speedups of

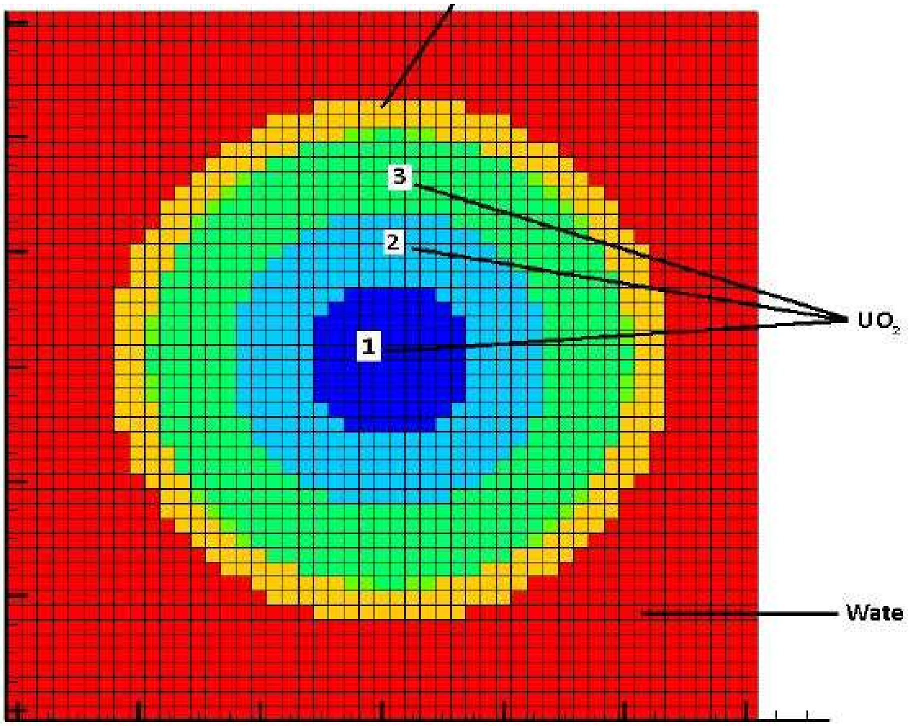


Figure 57: PWR 3-Zone Pin Model Used for MPI-OpenMP PENTRAN

$1.\bar{3}, 1.5, 1.6,$ and $1.\bar{6}$, respectively. Hence, performance gains are not going to follow linear speedup. This bounds our expectations of performance improvement to no better than say, 15% improvement in reduction of wall clock time with the addition of each thread. Also, when taking the limit as p goes to infinity, the speedup goes to 2. So the amount of impactful benefit that could be realized through OpenMP threading is limited in a massively parallel computation scenario. Though a disadvantage, with OpenMP we are still able to realize a parallelization benefit in angle beyond MPI, which is the driving force behind this investigation.

7.3 OpenMP Implementation, Investigation, and Results

7.3.1 Implementation

The Fortran subroutine, entitled SWEEPM, was refactored to support the OpenMP application. This necessitated the inlining of the difference routine, and other auxilary routines that reside within the sweep loop. In order to successfully apply the OpenMP

```
for each coarse mesh:
    determine coarse mesh entry point & set variables based on 1 of 8 sweep octants
for each entering boundary surface:
    determine if adjacent cell flux of boundary condition must be transferred
determine local grid spacing
determine max grid stride
set number of fine meshes for coarse cell
initialize total cross section to homogenize
! The next loop was the result of code refactoring
! to remove if statements from the main sweep loop
for all ordinates within octant:
    set signs of direction cosines for (mu,eta,xi) of ordinate
! OpenMP parallelization on outermost loop
for all ordinates in octant:
! sweep through cells in order of (z(y(x)))
    for kz = kz_start to kz_stop by kz_step
        for jy = jy start to jy stop by jy step
            for ix = ix start to ix stop by ix step
                determine x,y,z entering flux from adjacent cell type/start point
                determine total source: scatter+fission+external
                perform fiite difference given inputs
                    phix in ; phiy in ; phiz in ; mu; eta; xi; dx; dy; dz; sig-tot ;S
                    to determine outputs phix out ; phiy out ; phiz out ; phi average
                store phi average to array
                pass and save/transfer phiout to phiin for next iterate
! OpenMP parallelization ends
for each boundary surface:
  determine if boundary fluxes must be transferred ! not trivial
```

Figure 58: SWEEP Pseudocode

```
OpenMP Directives Used in SWEEPM
!$OMP PARALLEL DO DEFAULT(SHARED)

!$OMP+ FIRSTPRIVATE(a,b,c,top,bot,pA,pxout,pyout,pzout,qa,fmetrc,
!$OMP+ rmu,eta,xi,iploc,jpnt,
!$OMP+ dirxf,diryf,dirzf,adirxf,bdiryf,cdirzf,dx,dy,dz,iomloc,
!$OMP+ thetet,thetmu,thetxi,pxin,pyin,pzin,sigtot,
!$OMP+ psia,  iomega,tid,imloc,imcno,dxm,dym,dzm,
!$OMP+ psixin,psiyin,psizin,psixot,psiyot,psizot,rkeigm,
!$OMP+ imxref,jmyref,kmzref,dx,dxcm,dy,dycm,dz,dzcm,
!$OMP+ iswloc,igploc,iswp,igpg,sigt,jyfac,kzfac,isal,
!$OMP+ kzstrt,kzstop,kzstep,ixm,jym,kzm,icloc,igshft,kmzold,
!$OMP+ jystrt,jystop,jystep,rmu,eta,xi,icg,imxold,jmyold,
!$OMP+ ixstrt,ixstop,ixstep,nomloc,nx,ny,nz,q0fac)

!$OMP PARALLEL END DO DEFAULT(SHARED)
```

Figure 59: OpenMP Directives Used in Fortran SWEEPM subroutine

directives, variables are assumed to be shared by default. Then, in order to minimize the impact of false sharing, a host of variables are labeled private. The specific application of the OpenMP directive call is given next in Figure 59.

7.3.2 OpenMP Results

7.3.2.1 Small Problem Results

The results are given by Table 21, and Figures 60 through 62. In order to 'smooth' results, in the small analysis we performed 10 repeated computations per hybrid MPI/OpenMP combination, and plotted the best, worst, and average data points. This was done to mitigate factors related to system noise and latencies in network communications, and file I/O performance. The calculations were performed utilizing a dedicated node, where the whole node was subscribed for the user.

7.3.2.2 Large Problem Results

The large problem uses S_{30} quadrature and a factor 10 increase in meshing. This was done to increase the time spent in the sweeps, as well as create a more realistic application scenario.

Table 21: Percentage of Wall Clock Time Spent in Transport Sweep (each data point representing the average of 10 runs)

Threads/MPI Tasks	2	4	8
1	48.2%	51.8%	47.4%
2	46.8%	44.6%	39.3%
3	46.6%	51.4%	35.6%
4	34.6%	37.8%	34.9%
5	46.0%	49.5%	30.8%

7.3.3 OpenMP Analysis of Results

In all cases, all results converged to the same exact floating point results, separately for the small and large problem. This was the primary diffculty in achieving success, which was validating that OpenMP would retain the same results. This required the necessary privitization of variables so that proper data writes could be achieved. Otherwise, stochastic results emerged (different inner iterations, slightly different convergence results, similar eigenvalues). As the code is deterministic, variational results are not expected nor are they acceptable!

With the “small” problem, we varied both MPI tasks and threads per MPI task. This was fruitful in demonstrating the sweep algorithm. With 2 MPI tasks, each task takes on 4 octants, calling OpenMP in sequence 4 times per inner sweep iteration. With 4 MPI tasks, each task takes on 2 octants, calling OpenMP twice per sweep iteration. In using 8 MPI tasks, OpenMP is called only once per inner iteration. This appears to be the reason why there is more consistent behavior on 8 MPI tasks vs 2 or 4. In analyzing thread performance with the small problem, best speedups using the average times (with 1 thread as basis) for 2,4, and 8 MPI tasks were calculated to be 1.26 (4 threads), 1.31 (4 threads), and 1.25 (3 threads). Even with the small problem, going beyond 4 threads does not demonstrate a clear performance advantage. The takeaway here, is that there is an optimal choice of threads per MPI task, which may be problem dependent. We can go as far as saying that application of 2 MPI threads is marginally better, but using more than 2 threads may contribute to additional

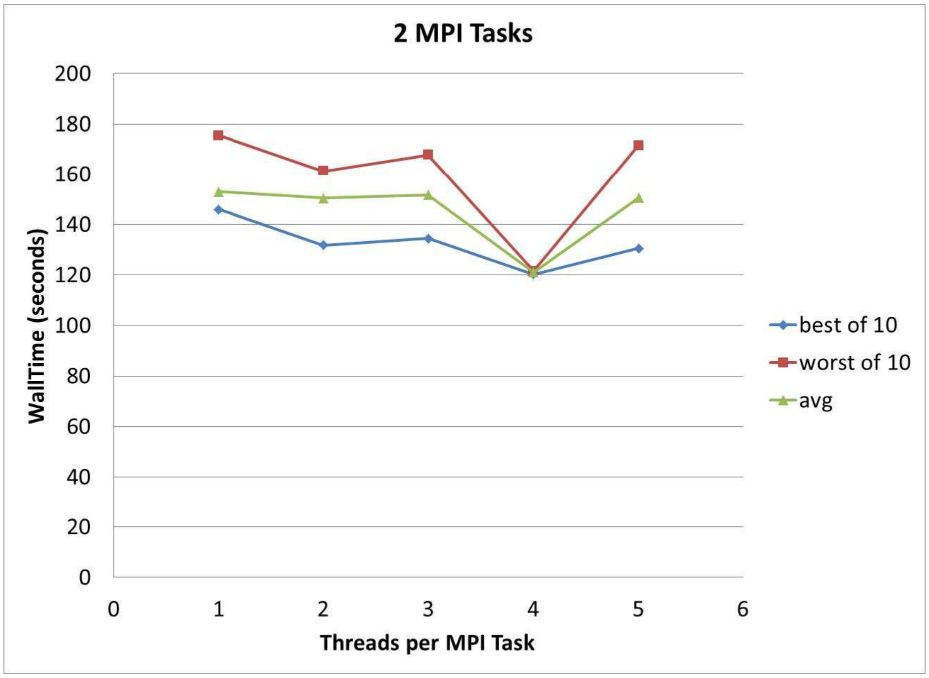


Figure 60: S_8/small with 2 MPI Tasks, for OpenMP Threads 1 to 5

overheads and false sharing. Also, the choice of 8 MPI tasks is favorable, mostly because the wall clock times are overwhelmingly better. The results of this small problem, guided the latter large problem, in which we used 8 MPI tasks.

The large problem is more realistic in size, as PENTRAN users typically run problems on the order of 100K fine meshes per coarse mesh. In order to increase the difficulty of the problem, we increased the quadrature from S_8 to S_{30} (120 ordinates per octant, 960 in all). A sampling of the results are shown in Figure 63. The results show a performance benefit of 12% for 2 threads per MPI task and 21% for 5 threads per MPI task. In practical terms, this took a calculation requiring 2.2 hours down to 1.95 hours. Since some calculations can take on the order of days to weeks, this can potentially drive a savings of 3 hours for every 24 wall clock hours. The benefits are not tremendous, but are worthwhile if the resources are available.

A brief conclusion to be made is that OpenMP parallelization extends the angular parallelization decomposition from 8 CPUs to 8 CPUs times an integer limited to maximum number of ordinates per octant. Again, used in tandem with space and

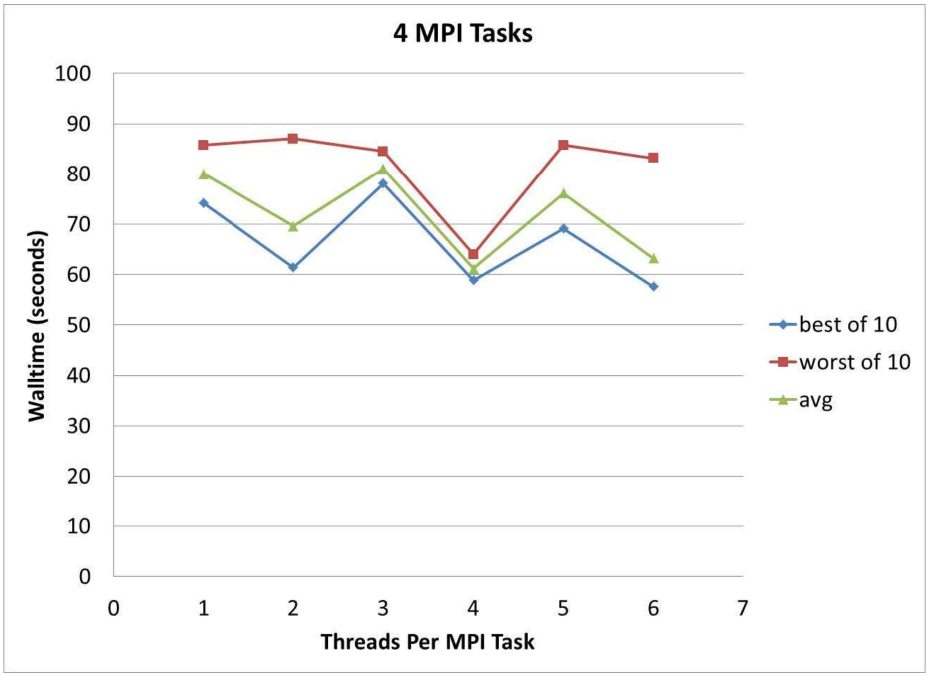


Figure 61: S_8/small with 4 MPI Tasks, for OpenMP Threads 1 to 6

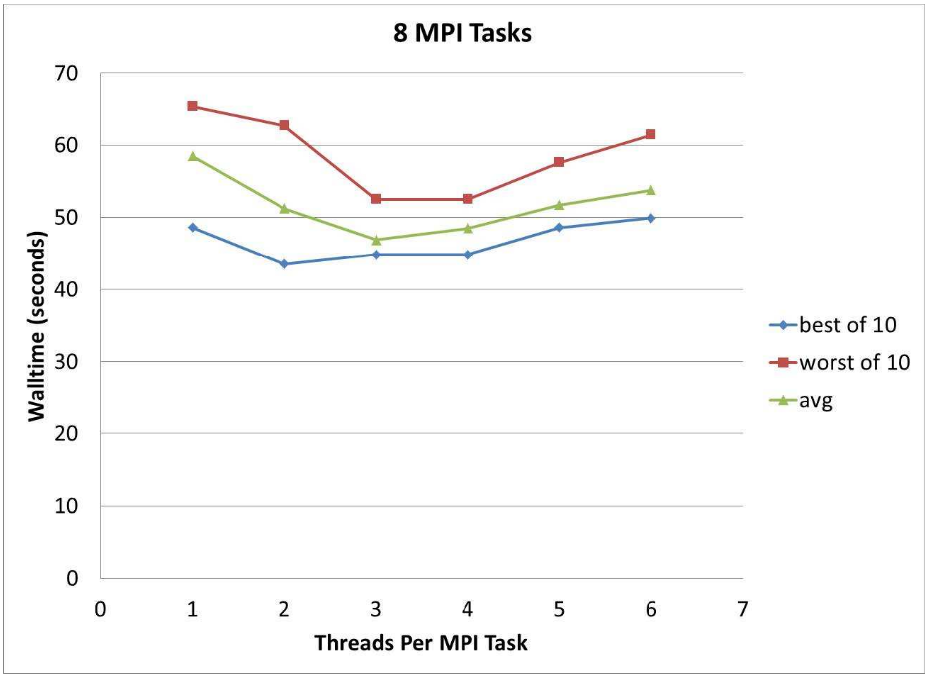


Figure 62: S_8/small with 8 MPI Tasks, for OpenMP Threads 1 to 6

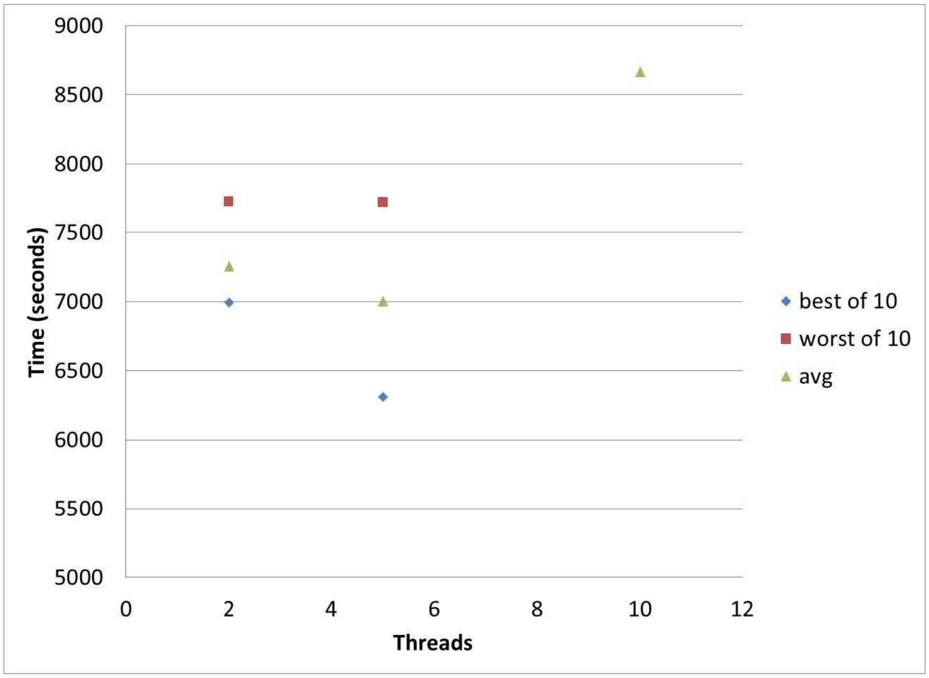


Figure 63: S_{30}/large with 8 MPI Tasks, for OpenMP Threads 2, 5, and 10

energy decomposition allows for a $N * M$ ($N =$ # of OpenMP threads, M=# of MPI tasks) increase in parallelization that may not have been achievable with an MPI-only application. The large problem indicates a speedup of roughly 1.25 given 5 threads used per MPI task, hence a 20% performance improvement. But the drawbacks are noted, in that maximum speedups are theoretically no better than 2, and the additional overhead of calling OpenMP in the sweep loop is an unavoidable cost.

CHAPTER VIII

BURNUP BENCHMARKING WITH BR3 PIN AND ADAPTING PENBURN TO SUPPORT MCNP

This chapter addresses the current state of PENBURN: a burnup solver that couples to both discrete ordinates and Monte Carlo multigroup transport solvers. We first discuss the specific development process created for enabling Monte Carlo support as an option. Then, in order to aid in the validation of the burnup driver framework, we apply analysis using the BR3 2D Pin (discussed in Section 8.2). The novelty of this framework allows for both codes to use the same Cartesian grid geometry, without additional intervention by the user/evaluator. The last section discusses the benchmark used in this thesis, and follows with relevant conclusions regarding the PENBURN results benchmarked to SCALE.

8.1 Adapting Discrete Ordinates Multigroup Burnup to Monte Carlo Multigroup Burnup

In this section, we detail the requirements for enabling burnup with Monte Carlo as an alternative transport solver. This methodology sets the stage for trivially swapping out transport solver components on-the-fly, where one can select their choice of solver, without the need to remodel geometry or cross sections.

In strong adherence to the **BURNDRIVER** framework, this entailed leveraging the MESH2MCNP conversion utility, a conversion script to collect fine mesh flux tallies [41]. Additionally, another supporting script (termed PENDATAMC) in Python was written to convert "meshtal" tallies to multigroup flux file format, labeled as flx files familar to users of PENTRAN, this script is analagous to the use of the

PENDATA utility which generates flux data. Lastly, the burden of cross section conversion is handled using XSMCNP, which converts multigroup Sn-based cross sections to multigroup format in MCNP via the “mgopt” card.

The easiest way to describe the process is to define transport solver module and code components inside each transport module:

1. Sn Transport Module Collection : PENTRAN, PENDATA
2. MC Transport Module Collection : XSMCNP, MESH2MCNP, MCNP, PENDATAMC

A new **BURNDRIVER** Bash script was created to support the MC Transport Module, plugging in Monte Carlo where normally discrete ordinates PENTRAN is called.

The burden to collect highly detailed fine mesh flux in the Monte Carlo transport solver is a current requirement; future development could improve the tally structure so as to tally specific fuel zones, further reducing calculation times; in the present benchmark study, we focus less on the timing reduction here and perform a quality assurance on the validation of the Monte Carlo burnup system.

8.2 BR3 Benchmark

The Belgian Reactor 3 reactor is known as the first PWR developed in Western Europe in 1962 and also the first PWR decomissioned (in region), completed in 2011 [8].

The BR3 was rated to 40.8 MW thermal and 11.5 MW electric, and is a PWR type with 73 *hexagonal* assemblies with four different types of fuel assemblies. Although the assemblies are hexagonal, within 65 of the assemblies the fuel rods were arranged in a square lattice. Reactivity control was performed by the use of “tubular neutron absorbing rods”, “boric acid dissolved in the moderator”, and ”Gd_2O_3 burnable poison blended with UO2 in several fuel rods”. The aforementioned details and further

Table 22: BR3 UO2 Material Composition

ZAID	Weight Percent (%)
92234	0.0555
92235	7.2750
92236	0.0458
92238	80.7736
8016	11.8500

detail are provided from the reference listed here [15].

In particular the benchmark chosen does not focus on the accuracy of the geometric modeling (no hexagonal lattice), but relies on the detail of the burnup schedule. The BR3 benchmark here is simply assumed to be a 2d infinite square lattice pin model.

The analysis of the BURNDRIVER framework is supplemented by studying the following described BR3 benchmark, using SCALE's T-DEPL as the reference benchmark. This code was also chosen as the reference since burnup interpolated cross section data is also derived from the same basis input file. The composition of the fuel material is given in Table 22.

The burnup schedule applied for this analysis is provided in Table 23 in power density - day format. This benchmark is particularly useful in providing a detailed burnup schedule.

In the next table, a variety of test cases were run with PENBURN: interpolation/no interpolation, convergence tolerences (Sn) or histories (Monte Carlo) varied, and choice of Sn or Monte Carlo. These cases and relevant parameters are given in Table 24.

At the time of calculation, the SCALE cross section yielded improper group 2 and 3 microscopic cross section values for hydrogen in water after 1140 days, which led to null results for the PENBURN matching cases after that time. Nonetheless, prior to this cumulative time the matchup is very consistent.

Table 23: Burnup Schedule for BR3 Pin

Power Density (W/g)	Burnup Time (d)	Subsequent Cooling (d)
25.5174	1.	0.0
25.5174	9.	0.0
25.5174	20.	0.0
25.5174	20.	0.0
25.5174	20.	19.
43.3690	2.	0.0
43.3690	20.	0.0
43.3690	100.	0.0
43.3690	100.	0.0
43.3690	100.	0.0
43.3690	100.	0.0
43.3690	100.	0.0
36.7534	27.	433.0
46.2568	1.	0.0
46.2568	14.	0.0
46.2568	30.	24.
43.1065	3.	0.0
43.1065	30.	0.0
43.1065	30.	0.0
43.1065	30.	0.0
43.1065	100.	0.0
43.1065	100.	0.0
43.1065	100.	10957.25

Table 24: Burnup Transport Parameters

Case	Description	MCNP kcode or Sn Criteria
0	SCALE Reference	S_{16} $\epsilon = 10^{-5}$
1	PENBURN-Sn No Interpolation	IC-192, Inner Tol. $= 10^{-4}$, Outer Tol. $= 10^{-5}$
2	PENBURN-Sn Interpolation	IC-192, Inner Tol. $= 10^{-4}$, Outer Tol. $= 10^{-5}$
3	PENBURN-MC No Interpolation	500 particles per cycle, 5 skipped cycles, 35 total cycles
4	PENBURN-MC Interpolation	5000 particles per cycle, 25 skipped cycles, 75 total cycles
5	PENBURN-MC No Interpolation	5000 particles per cycle, 25 skipped cycles, 75 total cycles
6	PENBURN-MC Interpolation	10000 particles per cycle, 100 skipped cycles, 600 total cycles

8.3 BR3 Benchmark Evaluation

In this benchmark evaluation, we specifically consider actinide comparisons, primarily the Pu-240 to Pu-239 ratio, as well as the elemental atom percent of U-235 as a function of time. We specifically wanted to pursue actinide validation of the PENBURN system compared to SCALE.

In particular, the resulting PENBURN cases generally straddle the reference results when examining the Pu-240 to Pu-239 ratio, shown in Figure 64. After 400 days of burnup, differences from the SCALE reference occur. This is more apparent in the next figure, Figure 65, where the Pu-240 to Pu-239 error is plotted. The initial results show that PENBURN-MC cases are above the reference results, whereas the PENBURN-Sn cases are more closely matched. The closest match comes from using the PENBURN-Sn interpolation case, which make sense as transport parameters are closest to SCALE, using discrete ordinates and an interpolated 3-group reference library.

Another comparison is shown in Figure 66, where U-235 elemental atom percent is shown as a function of burnup. Generally, the primary enrichment actinides do not show as much sensitivity as seen in the Plutonium ratio comparison.

The eigenvalue as a function of time in days in shown in Figure 67. The plot shows consistency between all cases. This plot also demonstrates that eigenvalue is not as sensitive, although there is some separation of the Monte Carlo burnup case results towards the end of the burnup schedule.

.

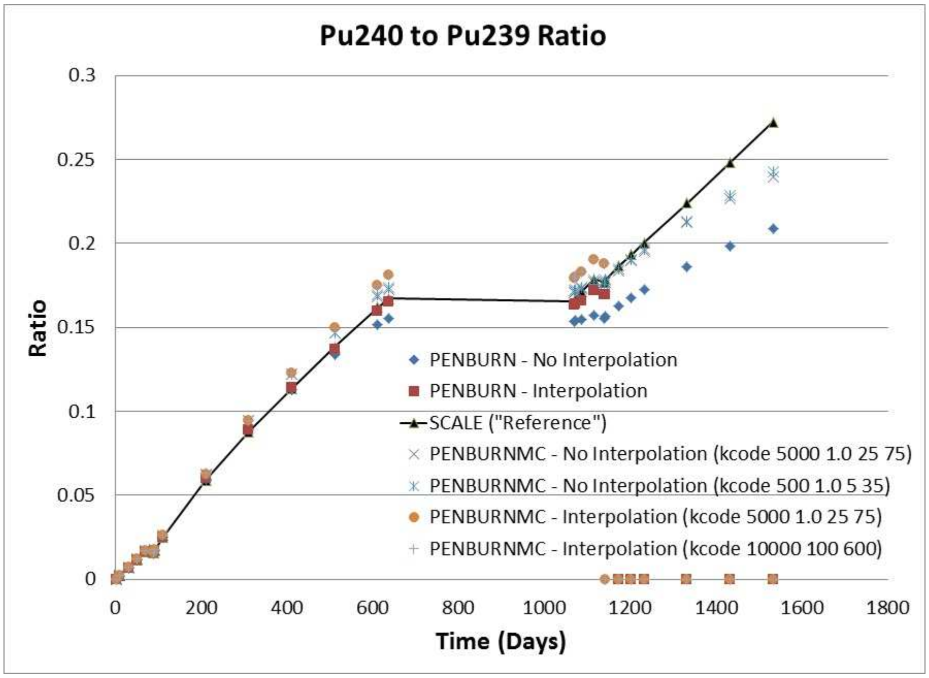


Figure 64: Pu-240 to Pu-239 Ratio Evaluation

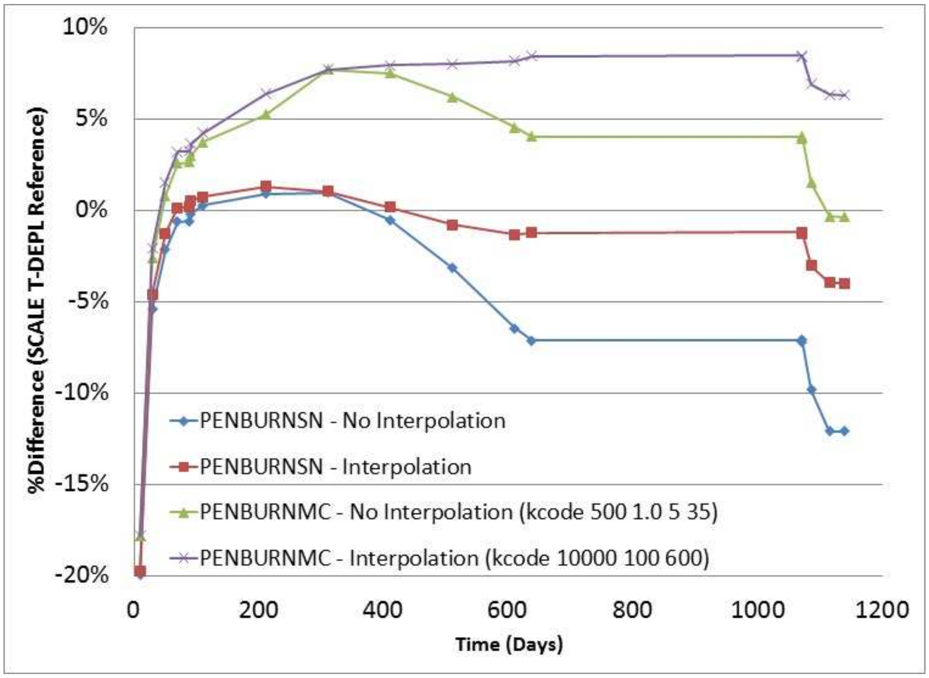


Figure 65: Pu-240 to Pu-239 Error from Reference SCALE Calculation

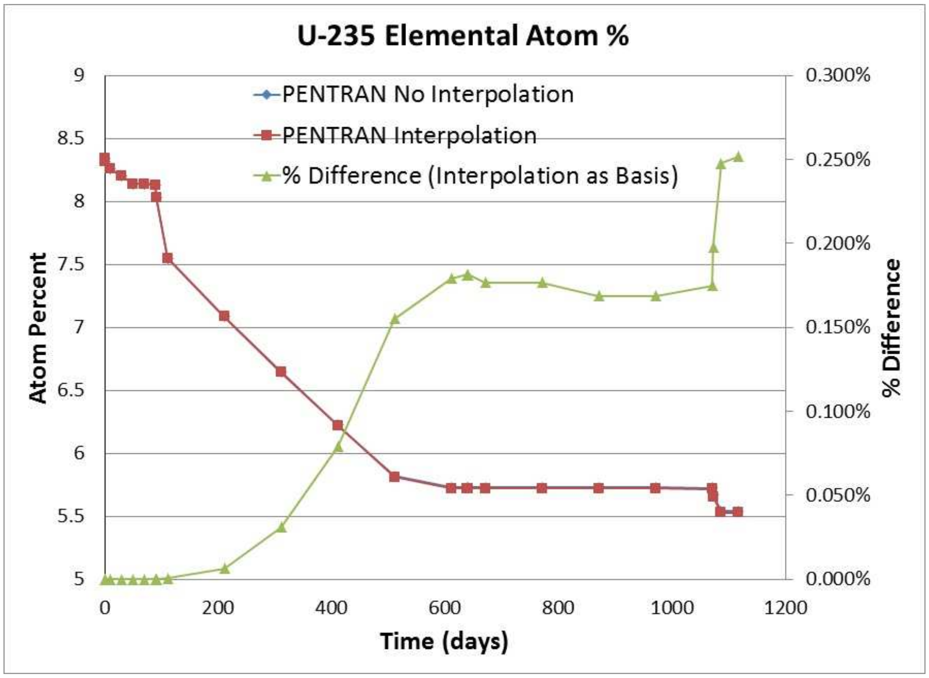


Figure 66: U-235 Enrichment for PENBURN-Sn Cases

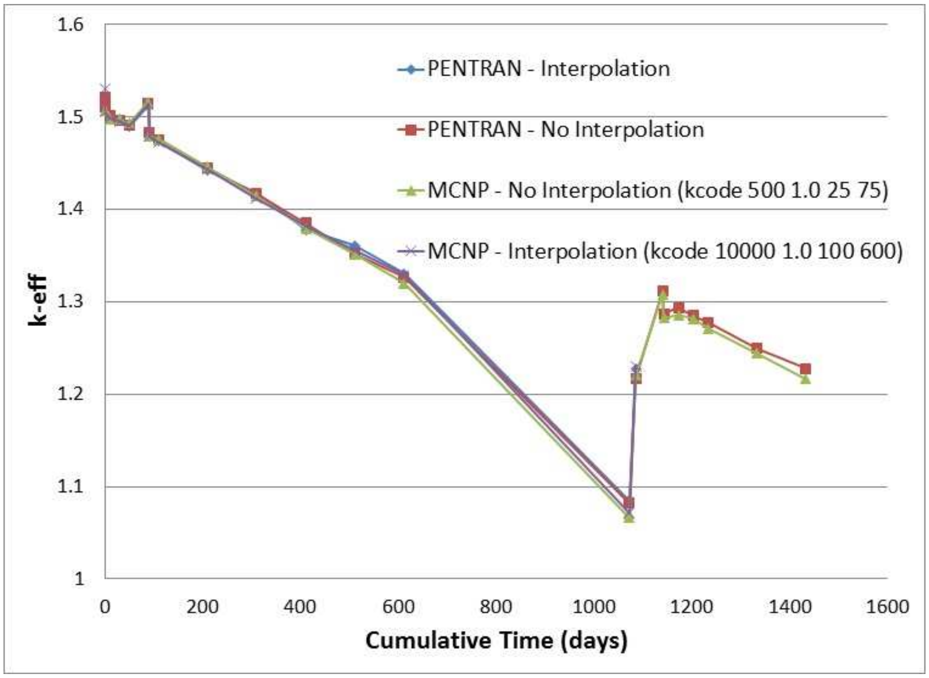


Figure 67: Eigenvalue as a Function of Time for PENBURN Cases

CHAPTER IX

OVERALL SUMMARY AND CONCLUSIONS

This thesis work identifies the transport phase space in the context of 3d Sn quadratures, multigroup (MC/Sn) cross sections, and Cartesian meshing impacts, covered in Chapters 3-5. The following chapters 6 and 7 address adaptive differencing analysis and parallization optimization for Sn with hybrid OpenMP-MPI. The preceding chapter 8 wraps up with a complete benchmark demonstration of hybrid MC-Sn burnup with PENBURN.

9.1 Milestones and Key Accomplishments

In this section, we list notable accomplishments performed by the author for this thesis effort, entailed coverage of analyzing the phase space for accuracy gains that will benefit coupled transport and burnup:

- Identified the potential for the use of Icosahedral quadrature, as well as successfully using EO and LDFE-SA quadratures within PENTRAN
- Adapted Icosahedral quadratures for use in PENTRAN and parallel 3d Sn environments that normally rely on octant-based quadratures
- Implemented periodic boundary condition support in 3d PENTRAN
- Identified an algorithmic improvement for periodic boundary conditions using Icosahedral quadrature, called 'Angular Resweep'
- Developed a hybrid Monte Carlo - Sn strategy for multigroup cross section collapse in fast and thermal reactor systems

- Identified issues in Cartesian grid meshing compared to exact geometry for eigenvalue transport systems; linear relationship between eigenvalue and volume quantified

- Recommended raising the default *dtwmxw* weight to at least 0.98 for PWR-type eigenvalue problems to prevent early upgrade to EDI from DTW differencing schemes as applied in 3n PENTRAN

- Wrote and authored a code, MESH2MCNP, to auto-convert PENTRAN meshing to MCNP as well as a system for autogeneration of mesh cases

- Modified burnup framework code, BURNDRIVER, to support Monte Carlo transport in place of discrete ordinates transport (PENBURN-MC)

- Implemented OpenMP and successful parallelization within PENTRAN, uniquely adapting a 3d Sn transport code to have support for both MPI and OpenMP, noting that no more than 2 OpenMP tasks is likely optimal

APPENDIX A

OVERVIEW OF MATRIX EXPONENTIAL VS. CRAM

First, define the A and n matrices:

```
A = Normal[SparseArray[{{1, 1} → -(σ24p),
      {2, 1} → σ24p, {2, 2} → -(σ25p),
      {3, 2} → σ25p, {3, 3} → -(σ26p),
      {4, 4} → -(σ28p),
      {5, 4} →  σ28p, {5, 5} → -(λ39 + σ39p),
      {6, 5} → λ39, {6, 6} → -1.0 (σ49p),
      {7, 6} → σ49p, {7, 7} → -(σ40p),
      {8, 7} → σ40p, {8, 8} → -(σ41p),
      {9, 8} → σ41p, {9, 9} → -(σ42p)}, {9, 9}]];

MatrixForm[%]
n = ConstantArray[0, {9, 1}];

n // MatrixForm
n[[2, 1]] = 0.01;
n[[4, 1]] = 0.
```

```
⎛-σ24p    0      0      0         0           0        0      0      0    ⎛0
  σ24p  -σ25p    0      0         0           0        0      0      0     0
   0     σ25p  -σ26p    0         0           0        0      0      0     0
   0      0      0    -σ28p       0           0        0      0      0     0
   0      0      0     σ28p   -λ39 - σ39p     0        0      0      0     0
   0      0      0      0        λ39       -1. σ49p    0      0      0     0
   0      0      0      0         0          σ49p    -σ40p    0      0     0
   0      0      0      0         0           0       σ40p  -σ41p    0     0
⎝  0      0      0      0         0           0        0     σ41p  -σ42p  ⎝0
```

Then, set reasonable values for phi, decay constants, and cross section values:

```
phi = 5 * 10^13 * 1. * 10^-24;
λ24 = Log[2] / (7.74723 * 10^12); σ24p = 87 * phi;
λ25 = Log[2] / (2.220976452 * 10^16); σ25p = 87 * phi;
λ26 = Log[2] / (7.39063204 * 10^14); σ26p = 5 * phi;
λ28 = Log[2] / (1.40996 * 10^17); σ28p = 2.4 * phi;
λ39 = Log[2] / 203 602.; σ39p = 33 * phi;
λ49 = Log[2] / (7.60837 * 10^11); σ49p = 972 * phi;
λ40 = Log[2] / (2.0714 * 10^11); σ40p = 263 * phi;
λ41 = Log[2] / (4.52842 * 10^8); σ41p = 1264 * phi;
λ42 = Log[2] / (1.17802 * 10^13); σ42p = 19 * phi;
A = Normal[SparseArray[{{1, 1} → - (σ24p),
     {2, 1} → σ24p, {2, 2} → - (σ25p),
     {3, 2} → σ25p, {3, 3} → - (σ26p),
     {4, 4} → - (σ28p),
     {5, 4} →  σ28p, {5, 5} → - (λ39 + σ39p),
     {6, 5} →  λ39, {6, 6} → -1.0 (σ49p),
     {7, 6} → σ49p, {7, 7} → - (σ40p),
     {8, 7} → σ40p, {8, 8} → - (σ41p),
     {9, 8} → σ41p, {9, 9} → - (σ42p)}, {9, 9}]];
MatrixForm[%]
n = ConstantArray[0, {9, 1}];
n[[2, 1]] = 0.01;
n[[4, 1]] = 0.99;
n // MatrixF
```

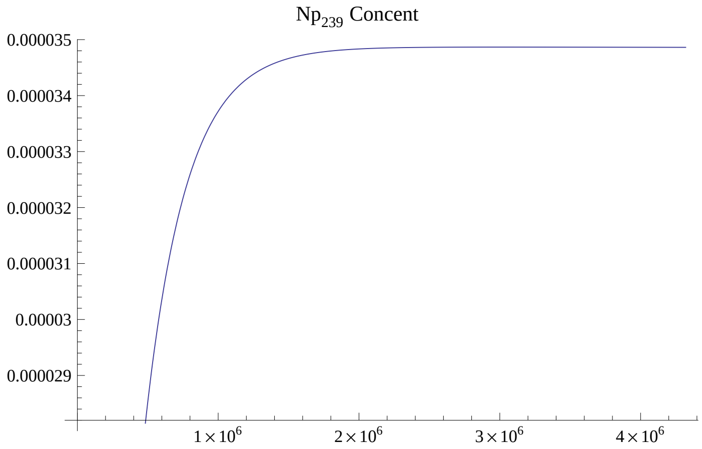
$$
\begin{pmatrix}
-4.35\times10^{-9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
4.35\times10^{-9} & -4.35\times10^{-9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 4.35\times10^{-9} & -2.5\times10^{-10} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1.2\times10^{-10} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1.2\times10^{-10} & -3.40607\times10^{-6} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 3.40442\times10^{-6} & -4.86\times10^{-8} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 4.86\times10^{-8} & -1.315\times10^{-8} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1.315\times10^{-8} & -6.32\times10^{-8} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 6.32\times10^{-8} & -9.5\times10^{-10}
\end{pmatrix}
\begin{pmatrix}
0 \\ 0.01 \\ 0 \\ 0.99 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0
\end{pmatrix}
$$

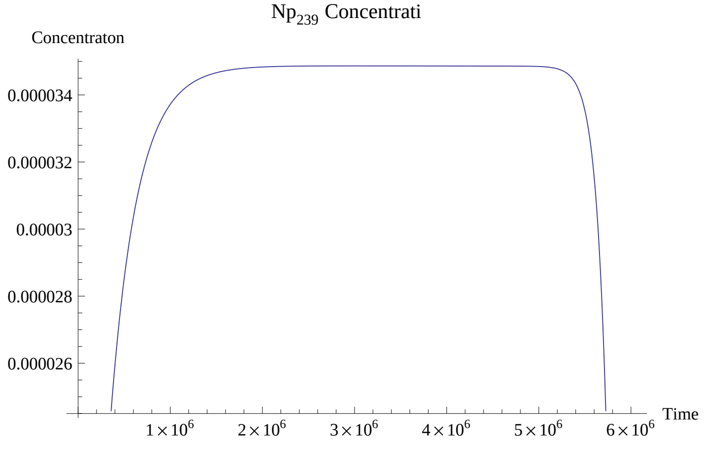
An example of the application of the Matrix Exponential method is shown in the plot of the normalized concentration value of Np_{239}. We can see that when the timestep Δt grows large to where the method fails. This is conveyed with the matrix norm growing large. The matrix norm for depletion systems is large to where it creates unstable behavior in standard Matrix Exponential methods. In the below plots, the norms at 50 and 70 days are 20.8 and 29.1, respectively.

```
result = Sum[MatrixPower[A * t, m]/m!, {m, 0, 50}].n;

Print["Np239 with Norm=", Norm[A * 50 * 86 400], " with Time(days) = ", 50]
Plot[{result[[5]]}, {t, 0, 50 * 86 400}, PlotLabel → "Np239 Concentration"]

Print["Np239 with Norm=", Norm[A * 70 * 86 400], " with Time(days) = ", 70]
Plot[{result[[5]]}, {t, 0, 70 * 86 400},
 AxesLabel → {Time, Concentraton}, PlotLabel → "Np239 Concentrati
```

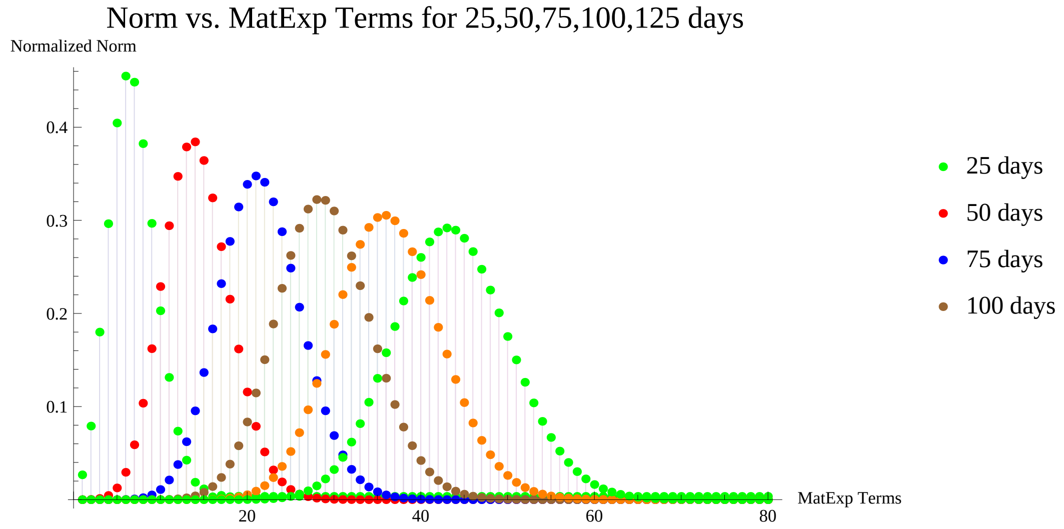




We aim to show here, the 'normalized' norm as a function of Matrix Power terms. Here, there is bell distribution that grows wider as time increases; this means that in order to overcome instability, more Matrix Power terms must be used (the desired number of terms in this graph would be around 70 terms, in order to minimize the norm sufficiently. We move to the next calculation by examining a single norm plot.

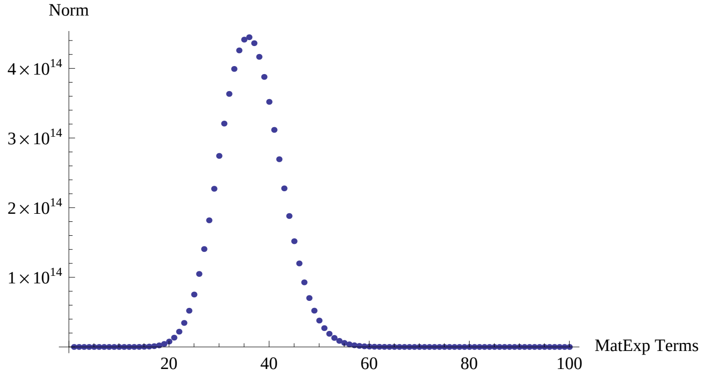
```
                         MatrixPower[A * 86 400 * maxTime, m]
timeN = Hold[Table[Sum[--------------------------------------, {m, 0, maxTerm, 1}] // Norm,
                                          m !
      {maxTerm, 80}] // Normalize];
```

```
ListPlot[Table[ReleaseHold[timeN], {maxTime, 25, 150, 25}],
 PlotRange → Full, Filling → Axis, PlotStyle →
  {{PointSize[Medium], Green}, {PointSize[Medium], Red}, {PointSize[Medium], Blue},
   {PointSize[Medium], Brown}, {PointSize[Medium], Orange}},
 PlotLabel → Style["Norm vs. MatExp Terms for 25,50,75,100,125 days", 16],
 AxesLabel → {"MatExp Terms", "Normalized Norm"},
 PlotLegends → {"25 days", "50 days", "75 days", "100 days", "125 days"}
```



In our experience, the norm of the matrix should be constrained to values less than 1. Consider that using a 100-terms is costly with a much larger matrix system (common depletion systems have 2000 nuclides). With 80, 90, and 100 terms, the norm decreases from approimately 1 million, to 229, to 4. So with a large timestep of 125 days, 100 terms is not sufficient.

```
ListPlot[Table[Sum[MatrixPower[A * 86 400 * 125, m] / m!, {m, 0, maxTerm, 1}] // Norm,
  {maxTerm, 100}], AxesLabel → {"MatExp Terms", "Norm"}]
norm100 = Table[Sum[MatrixPower[A * 86 400 * 125, m] / m!, {m, 0, maxTerm, 1}] // Norm,
   {maxTerm, 100}];
norm100⟦80⟧
norm100⟦90⟧
norm100⟦100
```



This page shows the application of the Chebyshev Rational Approximation Method, termed CRAM, of order 14. The calculations are based on that used in the SERPENT code (CRAM implemented by M. Pusa, VTT Finland).

```
ClearAll[θ, α, k]
k = 14;
θ = {};
θ = Append[θ, -0.562314417475317895 * 10^1 + (0.119406921611247440 * 10^1) I];
θ = Append[θ, -0.508934679728216110 * 10^1 + (0.358882439228376881 * 10^1) I];
θ = Append[θ, -0.399337136365302569 * 10^1 + (0.600483209099604664 * 10^1) I];
θ = Append[θ, -0.226978543095856366 * 10^1 + (0.846173881758693369 * 10^1) I];
θ = Append[θ, 0.208756929753827868 * 10^0 + (0.109912615662209418 * 10^2) I];
θ = Append[θ, 0.370327340957595652 * 10^1 + (0.136563731924991884 * 10^2) I];
θ = Append[θ, 0.889777151877331107 * 10^1 + (0.166309842834712071 * 10^2) I];
θ // MatrixForm
α = {};
α = Append[α, 0.557503973136501826 * 10^2 + (-0.204295038779771857 * 10^3) I];
α = Append[α, -0.938666838877006739 * 10^2 + (0.912874896775456363 * 10^2) I];
α = Append[α, 0.469965415550370835 * 10^2 + (-0.116167609985818103 * 10^2) I];
α = Append[α, -0.961424200626061065 * 10^1 + (-0.264195613880262669 * 10^1) I];
α = Append[α, 0.752722063978321642 + (0.670367365566377770) I];
α = Append[α, -0.188781253158648576 * 10^-1 + (-0.343696176445802414 * 10^-1) I];
α = Append[α, 0.143086431411801849 * 10^-3 + (0.287221133228814096 * 10^-3) I];
α0 = 0.183216998528140087 * 10^-11
α // MatrixForm
```

```
⎛-5.6231441747531790 + 1.19406921611247440 ⅈ
 -5.0893467972821611 + 3.58882439228376881 ⅈ
  -3.9933713636530257 + 6.0048320909960466 ⅈ
  -2.2697854309585637 + 8.4617388175869337 ⅈ
  0.20875692975382787 + 10.991261566220942 ⅈ
   3.7032734095759565 + 13.656373192499188 ⅈ
⎝  8.8977715187733111 + 16.630984283471207 ⅈ
```

```
1.8321699852814009 × 10^-12
```

```
⎛     55.750397313650183 - 204.295038779771857 ⅈ
      -93.866683887700674 + 91.287489677545636 ⅈ
       46.996541555037084 - 11.616760998581810 ⅈ
      -9.6142420062606106 - 2.64195613880262669 ⅈ
     0.75272206397832164 + 0.67036736556637777 ⅈ
  -0.0188781253158648576 - 0.034369617644580241 ⅈ
⎝0.00014308643141180185 + 0.00028722113322881410 ⅈ
```

This next base of calculations creates 3 Tables, at 25, 75, and 125 days. It shows the results of U and Pu actinides in the system using CRAM, 5 term, 20 term, and 80 term Matrix Exponential calculations, and then CRAM Percent Error using the 80 term MatExp as the reference calculation.

For the 25 and 75 day calculations, the errors are extremely small. At 125 days, we expose the problem with the 80 term calculation, where indeed, CRAM should have been the reference calculation. In fact, it would be wise to consider CRAM as the reference calculation in future contexts.

```
Table[cram = α0 n - Sum[α⟦j⟧
      (Inverse[(A t + θ⟦j⟧ IdentityMatrix[9]) /. t → 86400 * time]).n, {j, 1, k / 2}];
  matexp5 = Sum[ MatrixPower[A * time * 86400, m]
                 ------------------------------- , {m, 0, 5}].n;
                               m!
  matexp20 = Sum[ MatrixPower[A * time * 86400, m]
                  ------------------------------- , {m, 0, 20}].n;
                                m!
  matexp80 = Sum[ MatrixPower[A * time * 86400, m]
                  ------------------------------- , {m, 0, 80}].n;
                                m!


  m = Table[{
     Re[cram⟦j, 1⟧],
     matexp5⟦j, 1⟧,
     matexp20⟦j, 1⟧,
     matexp80⟦j, 1⟧,
      (Re[cram⟦j, 1⟧] - matexp80⟦j, 1⟧)
     ---------------------------------- * 100}, {j, 2, 9}];
               matexp80⟦j, 1⟧

  matexp5norm = Norm[Sum[ MatrixPower[A * time * 86400, m]
                          ------------------------------- , {m, 0, 5}]];
                                        m!
  matexp20norm = Norm[Sum[ MatrixPower[A * time * 86400, m]
                           ------------------------------- , {m, 0, 20}]];
                                         m!
  matexp80norm = Norm[Sum[ MatrixPower[A * time * 86400, m]
                           ------------------------------- , {m, 0, 80}]];
                                         m!

  m = Prepend[m, {"CRAM-14",
     "5 TERM MATEXP Norm:"[matexp5norm],
     "20 TERM MATEXP Norm:"[matexp20norm],
     "80 TERM MATEXP Norm:"[matexp80norm],
     "CRAM Error (Basis 80 Term)"}];
  m = MapThread[Prepend, {m, {"Normalized Concs",
      "U235", "U236", "U238", "Np239", "Pu239", "Pu240", "Pu241", "Pu242"}}];
  Print["Time:", time, " days"];
  Grid[m, Frame → All]
  , {time, 25, 150, 50}] // MatrixFor
```

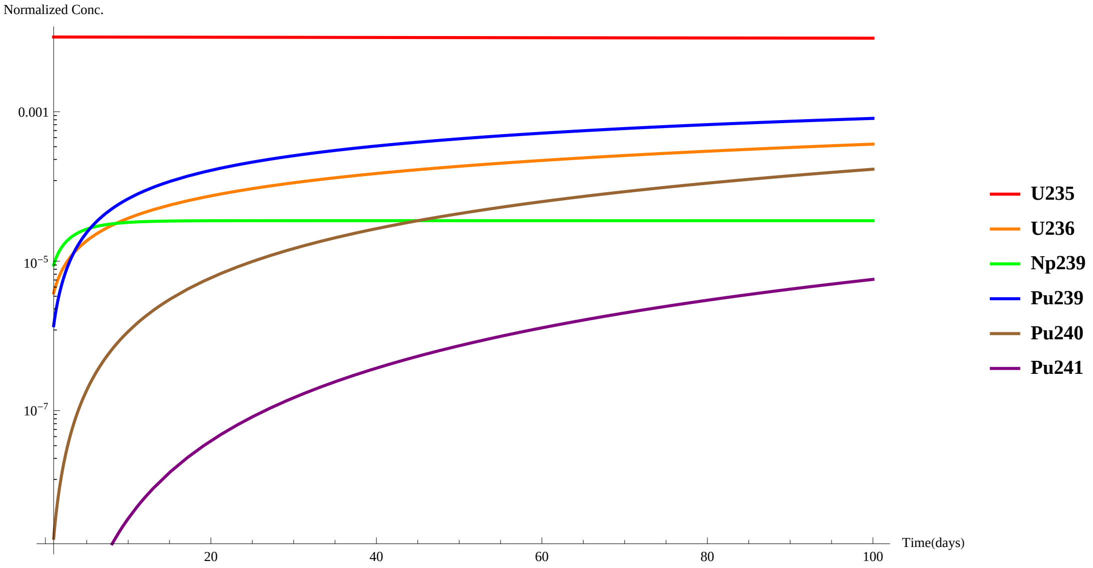
Normalized Concs	CRAM−14	5 TERM MATEXP Norm:[147.647]	20 TERM MATEXP Norm:[1.28094]	80 TERM MATEXP Norm:[1.29671]	CRAM Error (Basis 80 Term)
U_{235}	0.00990648	0.00990648	0.00990648	0.00990648	-3.09454×10^{-10}
U_{236}	0.0000934947	0.0000934947	0.0000934947	0.0000934947	-6.99191×10^{-10}
U_{238}	0.989743	0.989743	0.989743	0.989743	-3.12614×10^{-10}
Np_{239}	0.0000348488	0.0036349	0.0000340387	0.0000348488	-3.12924×10^{-10}
Pu_{239}	0.000211636	−0.00343876	0.000212458	0.000211636	-5.55905×10^{-10}
Pu_{240}	9.89738×10^{-6}	0.0000621854	9.88561×10^{-6}	9.89738×10^{-6}	-4.58892×10^{-9}
Pu_{241}	8.24928×10^{-8}	-1.23186×10^{-7}	8.2539×10^{-8}	8.24928×10^{-8}	-8.91892×10^{-8}
Pu_{242}	2.59064×10^{-9}	6.40069×10^{-9}	2.58978×10^{-9}	2.59064×10^{-9}	-2.572×10^{-6}

Normalized Concs	CRAM−14	5 TERM MATEXP Norm:[50 317.1]	20 TERM MATEXP Norm:[2.2849×10^{8}]	80 TERM MATEXP Norm:[1.18656]	CRAM Error (Basis 80 Term)
U_{235}	0.00972206	0.00972206	0.00972206	0.00972206	-3.03263×10^{-10}
U_{236}	0.000277718	0.000277718	0.000277718	0.000277718	-6.36678×10^{-10}
U_{238}	0.98923	0.98923	0.98923	0.98923	-3.12619×10^{-10}
Np_{239}	0.000034853	1.23233	−5595.91	0.000034853	−0.0000682822
Pu_{239}	0.000634008	−1.24889	5674.16	0.000634008	4.91873×10^{-6}
Pu_{240}	0.0000975662	0.0179957	−81.2762	0.0000975662	3.31561×10^{-7}
Pu_{241}	2.48748×10^{-6}	−0.0000679128	0.319722	2.48748×10^{-6}	3.83755×10^{-9}
Pu_{242}	2.53871×10^{-7}	1.55537×10^{-6}	−0.00593384	2.53871×10^{-7}	-5.03977×10^{-7}

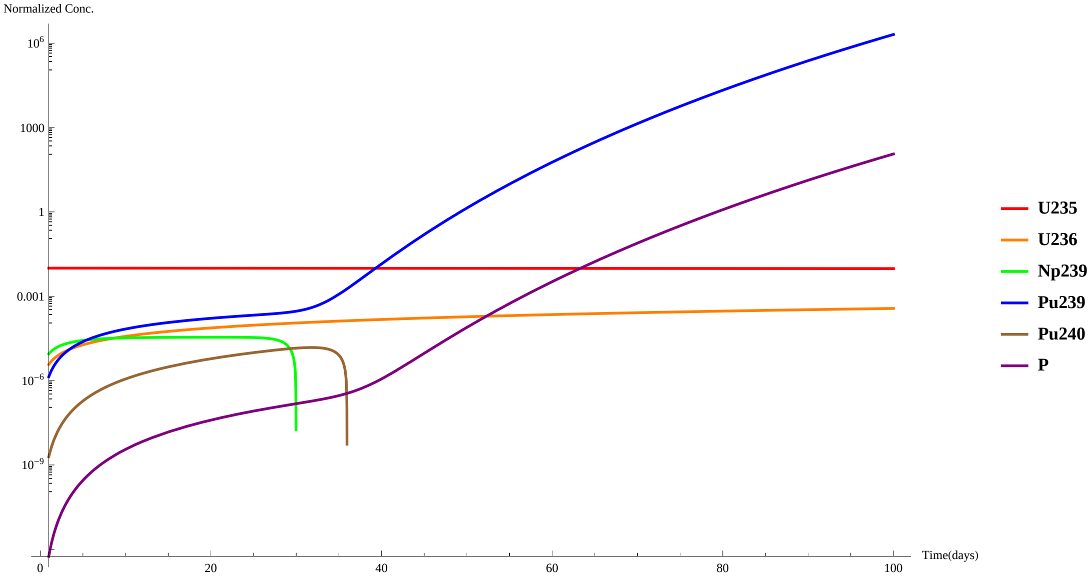
Normalized Concs	CRAM−14	5 TERM MATEXP Norm:[701 673.]	20 TERM MATEXP Norm:[7.75699×10^{12}]	80 TERM MATEXP Norm:[1.1189×10^{6}]	CRAM Error (Basis 80 Term)
U_{235}	0.00954106	0.00954106	0.00954106	0.00954106	-2.99161×10^{-10}
U_{236}	0.000458311	0.000458311	0.000458311	0.000458311	-5.90027×10^{-10}
U_{238}	0.988718	0.988718	0.988718	0.988718	-3.12456×10^{-10}
Np_{239}	0.0000348349	17.1846	-1.89975×10^{8}	−27.4028	−100.
Pu_{239}	0.000976171	−17.4239	1.92632×10^{8}	27.7869	−99.9965
Pu_{240}	0.000258053	0.249851	-2.75925×10^{6}	−0.397746	−100.065
Pu_{241}	0.000010655	−0.000971047	10 854.2	0.0015763	−99.324
Pu_{242}	1.88708×10^{-6}	0.0000200021	−201.456	−0.0000271717	−106.945

Finally, we show the superior results of CRAM with the plot of Normalized Concentration versus time (in days). Here the actinides in the system are plotted. The Matrix Exponential method is shown as the next plot, and it can clearly be seen that numerical instabilities occur at 25 days.

```
cram = α0 n - Sum[
    α⟦j⟧ (Inverse[(A t + θ⟦j⟧ IdentityMatrix[9]) /. t → 86 400 * time]).n, {j, 1, k / 2}];
LogPlot[{Re[cram]⟦2, 1⟧ /. time → myt, Re[cram]⟦3, 1⟧ /. time → myt,
  Re[cram]⟦5, 1⟧ /. time → myt,
  Re[cram]⟦6, 1⟧ /. time → myt, Re[cram]⟦7, 1⟧ /. time → myt,
  Re[cram]⟦8, 1⟧ /. time → myt}, {myt, 1, 100},
 PlotStyle → {{Thick, Red}, {Thick, Orange}, {Thick, Green},
   {Thick, Blue}, {Thick, Brown}, {Thick, Purple}},
 AxesLabel → {"Time(days)", "Normalized Conc."}, PlotLegends → Autom
```



```
Clear[time, m, matexp]
matexp = Sum[MatrixPower[A * time * 86 400, m] / m!, {m, 0, 20}].n;
LogPlot[{
  matexp⟦2, 1⟧ /. time → myt,
  matexp⟦3, 1⟧ /. time → myt,
  matexp⟦5, 1⟧ /. time → myt,
  matexp⟦6, 1⟧ /. time → myt,
  matexp⟦7, 1⟧ /. time → myt,
  matexp⟦8, 1⟧ /. time → myt},
 {myt, 1, 100}, PlotStyle → {{Thick, Red}, {Thick, Orange},
   {Thick, Green}, {Thick, Blue}, {Thick, Brown}, {Thick, Purple}},
 AxesLabel → {"Time(days)", "Normalized Conc."}, PlotLegends → Automatic]
```



Work to pursue using CRAM in the depletion algorithm of PENBURN 2.0. The work is also being pursued with OpenMC Depletion.

APPENDIX B

MULTIGROUP TABLES

Table 25: Multigroup Upper Energies for 49 Group Structure

Group	Upper Energy (MeV)	Group	Upper Energy (MeV)
1	20	26	1.25e-06
2	8.1873	27	1.15e-06
3	6.434	28	1.1e-06
4	4.8	29	1.05e-06
5	3	30	1e-06
6	2.479	31	6.25e-07
7	2.354	32	4e-07
8	1.85	33	3.75e-07
9	1.4	34	3.5e-07
10	0.9	35	3.25e-07
11	0.4	36	2.75e-07
12	0.1	37	2.5e-07
13	0.025	38	2.25e-07
14	0.017	39	2e-07
15	0.003	40	1.5e-07
16	0.00055	41	1e-07
17	0.0001	42	7e-08
18	3e-05	43	5e-08
19	1e-05	44	4e-08
20	8.1e-06	45	3e-08
21	6e-06	46	2.53e-08
22	4.75e-06	47	1e-08
23	3e-06	48	7.5e-09
24	1.77e-06	49	3e-09
25	1.5e-06		

REFERENCES

[1] 2008, "SVG Graph Illustrating Amdahl's Law,", [Online]. Available: http://en.m.wikipedia.org/wiki/File:AmdahlsLaw.svg.

[2] Ahrens, C. D., "Highly Efficient, Exact Quadratures for Three-Dimensional Discrete Ordinates Transport Calculations," *Nuclear Science and Engineering*, vol. 170, no. 1, pp. 98–101, 2012.

[3] Ahrens, C. and Beylkin, G., "Rotationally invariant quadratures for the sphere," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 2009.

[4] Alcouffe, R., Larsen, E., Miller, W. J., and Wienke, B., "Computational Efficiency of Numerical Methods for the Multigroup Discrete Ordinates Neutron Transport Equations: The Slab Geometry Case," *Nuclear Science and Engineering*, vol. 71, pp. 111–127, 1979.

[5] Alpan, F. A. and Haghighat, A., "Advanced Methodology for Selecting Group Structures for Multigroup Cross Section Generation," in *Proceedings of PHYSOR*, pp. 7–23, 2000.

[6] Amdahl, G., "Validity of the single processor approach to achieving large scale computing capabilities," in *AFIPS spring joint computer conference, 1967*, (IBM Sunnyvale, California), 1967.

[7] Bateman, H. *Proc. Cambridge Phil. Soc.*, no. 15, p. 423, 1910.

[8] Belgian Nuclear Research Centre SCKCEN, 2013, "BR3 - Belgian Reactor 3,", [Online]. Available: http://www.sckcen.be/Our-Research/Research-facilities/BR3-Belgian-Reactor-3.

[9] Bell, G. and Glasstone, S., *Nuclear Reactor Theory.* Van Nostrand Reinhold Company, 1970.

[10] Bowman, S. M., "SCALE 6: Comprehensive Nuclear Safety Analysis Code System," *Nuclear Technology*, vol. 174, pp. 126–148, 2011.

[11] Carlson, B., "Solution of the Transport Theory Equation by the Sn Method," Tech. Rep. LA-1891, Los Alamos National Laboratory, 1955.

[12] Carlson, B., "On a More Precise Definition of Discrete Ordinates Methods," in *Proc. Second Conf. Transport Theory*, (Los Alamos, N.M.), pp. 348–390, U.S. Atomic Energyy Commission, 1971.

[13] Cetnar, J., "General solution of Bateman equations for nuclear transmutations," *Annals of Nuclear Energy*, vol. 33, no. 7, pp. 640–645, 2006.

[14] Chadwick, M. B. and others, "ENDF/B-VII.0: Next Generation Evaluated Nuclear Data Library for Nuclear Science and Technology," *Nuclear Data Sheets*, vol. 107, pp. 2931–3060, 2006.

[15] Charlier, A., Gubel, P., Vandenberg, C., and Haas, D., "Experimental Study of the Tritium Inventory in the BR3 and Extrapolation to a PWR of 900 MWe," *Nuclear Science and Technology*, p. 52, 1982.

[16] Chiba, G., "Verification of Homogenization in Fast Critical Assembly Analyses," *Journal of Nuclear Science and Technology*, vol. 43, pp. 1395–1405, 2006.

[17] Chin, M. and others, "Optimization of Multi-Group Cross Sections for Fast Reactor Analysis," *International Conference Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, 2013.

[18] Courau, T. and Sjoden, G., "3D Neutron Transport and HPC: A PWR Full Core Calculation Using PENTRAN SN Code and IBM BLUEGENE/P Computers," in *Proceedings of the ANS Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo*, (Tokyo), American Nuclear Society, 2010.

[19] Dalle, H., "Monte Carlo Burnup Simulation of the Takahama-3 Benchmark Experiment," in *International Nuclear Atlantic Conference - INAC 2009*, (Rio de Janeiro, Brazil), ASSOCIAO BRASILEIRA DE ENERGIA NUCLEAR - ABEN, 2009.

[20] DeHart, M. D., *Discrete ordinates approximation to the neutron transport equation applied to generalized geometries.* PhD thesis, Texas A&M University, 1992.

[21] Dumonteil, E. and Courau, T., "Dominance Ratio Assessment and Monte Carlo Criticality Simulations: Dealing with High Dominance Ratio Systems," *Nuclear Technology*, vol. 172, no. 2, pp. 120–131, 2010.

[22] Dupree, S. and Fraley, S., *Monte Carlo Primer: A Practical Approach to Radiation Transport.* New York, New York: Kluwer Academic / Plenum Publishers, 2002.

[23] Eckhardt, R., "Stan Ulam, John von Newmann, and the Monte Carlo Method," *Los Alamos Science*, no. 15, pp. 131–137, 1987.

[24] Endo, T. and Yamamoto, A., "Development of New Solid Angle Quadrature Sets to Satisfy Even- and Odd-Moment Conditions," *Journal of Nuclear Science and Technology*, vol. 44, no. 10, pp. 1249–1258, 2007.

[25] Evans, T., “Denovo-A New Parallel Discrete Transport Code for Radiation Shielding Applications,” in *American Nuclear Society Radiation Production and Shielding Division 2010*, (Las Vegas, NV), American Nuclear Society, 2010.

[26] Haghighat, A. and Sjoden, G., “Significance of Adaptive Differencing, Variable Grid Density and TPMC for Sn Methods,” in *Proceedings of the 1999 Mathematics and Computation Conference on Reactor Physics and Supercomputing for Nuclear Applications*, vol. 99, (Madrid, Spain), pp. 1775–1783, American Nuclear Society, 1999.

[27] Haghighat, A., Sjoden, G., and Kucukboyaci, V., “Effectiveness of PENTRAN’s Unique Numerics for Simulation of the Kobayashi Benchmarks,” *Progress in Nuclear Energy*, vol. 39, no. 2, 2001.

[28] Hazama, T., Chiba, G., and Sugino, K., “Development of a Fine and Ultra-Fine Group Cell Calculation Code SLAROM-UF for Fast Reactor Analyses,” *Journal of Nuclear Science and Technology*, vol. 43, pp. 908–918, 2006.

[29] Hebert, A., *Applied Reactor Physics*. Canada: Presses inter Polytechnique, 2009.

[30] IAEA, 2010, “Safeguards Statement for 2010,”, [Online]. Available: http://www.iaea.org/OurWork/SV/Safeguards/documents/es2010.pdf.

[31] Ilas, G., Gauld, I., Difilippo, F., and Emmett, M., “Analysis of Experimental Data for High Burnup PWR Spent Fuel Isotopic Validation,” 2011.

[32] Jarrell, J. and Adams, M., “Discrete-Ordinates Quadrature Sets Based on Linear Discontinuous Finite Elements,” in *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011)*, (Rio de Janeiro, RJ, Brazil), Latin American Section (LAS) / American Nuclear Society (ANS), 2011.

[33] Jenal, J., Erickson, P., Rhoades, W., Simpson, D., and Williams, M., “Generation of a computer library for discrete ordinates quadrature sets. [DO-QDP],” Tech. Rep. ORNL/TM-6023, Oak Ridge National Lab, 1977.

[34] Korean Atomic Energy Research Institute, December 25, 1995 2012, “Total and Fission Cross Sections for Plutonium-239,”, [Online]. Available: http://atom.kaeri.re.kr/cgi-bin/endfplot.pl?j=f&d=mcnp&f=mcnp/Pu-239.

[35] Levesque, J. and Wagenbreth, G., *High Performance Computing: Programming and Applications*. Chapman & Hall / CRC Computational Science, 2010.

[36] Lewis, E. and Miller, Jr., W., *Computational Methods of Neutron Transport*. La Grange Park, IL: American Nuclear Society, 1993.

[37] Lewis, E., Smith, M. A., Tsoulfanidis, N., Palmiotti, G., Taiwo, T., and Blomquist, R., "Benchmark for Deterministic 2-D/3-D MOX Fuel Assembly Transport Calculations Without Spatial Homogenization (C5G7MOX)," 2001.

[38] Longoni, G., *Advanced quadrature sets and acceleration and preconditioning techniques for the discrete ordinates method in parallel computing environments.* PhD thesis, University of Florida, 2004.

[39] MacFarlane, R. and Muir, D., "The NJOY Nuclear Data Processing System, Version 99," tech. rep., Los Alamos National Laboratory, 1999.

[40] Manalo, K., "Development, Optimization, and Testing of a 3-D Zone Based Burnup/ Depletion Solver for Deterministic Transport," Master's thesis, University of Florida, 2008.

[41] Manalo, K., "MESH2MCNP: A Discretized Cartesian Geometry Creator for MCNP5," 2012. Georgia Institute of Technology.

[42] Manalo, K., Cerge, R., Wegner, D., and Sjoden, G., "Homogenization of Cross Sections and Related Performance Improvements in PWR Burnup Using the PENBURN/PENTRAN Burnup Sequence," in *PHYSOR 2010 - Advances in Reactor Physics to Power the Nuclear Renaissance*, (Pittsburgh), American Nuclear Society, 2010.

[43] M.D. DeHart, I.C. Gauld, K. S., "Three-dimensional depletion analysis of the axial end of a Takahama fuel rod," in *International Conference on the Physics of Reactors "Nuclear Power: A Sustainable Resource"*, (Interlaken, Switzerland), American Nuclear Society, 2008.

[44] Petrie, L., "FIDO Input System," Tech. Rep. NUREG/CR-0200, Oak Ridge National Laboratory, 1998.

[45] Petrovic, B. and Haghighat, A., "Analysis of inherent oscillations in multidimensional S[N] solutions of the neutron transport equation," *Nuclear Science and Engineering*, vol. 124, no. 1, 1996.

[46] Plower, T., "Fully Automated 3-D Parallel Simulation and Optimization of a Full Scale Pressurized Water Reactor Fuel Assembly with Burnup Corrected Cross Sections," Master's thesis, University of Florida, 2008.

[47] Plower, T., Manalo, K., Rowe, M., Mock, T., and Sjoden, G., "Multi-Zone Based Fuel Burnup Analysis of a PWR Assembly Using PENBURN," in *Transactions of the American Nuclear Society*, vol. 98, pp. 529–532, 2008.

[48] Pusa, M., "Rational Approximations to the Matrix Exponential in Burnup Calculations," *Nuclear Science and Engineering*, vol. 169, no. 2, pp. 155–167, 2011.

[49] Pusa, M. and Leppanen, J., "Computing the Matrix Exponential in Burnup Calculations," *Nuclear Science and Engineering*, vol. 164, no. 2, pp. 140–150, 2010.

[50] Sanders, C. and Gauld, I., "Isotopic Analysis of High-Burnup PWR Spent Fuel Samples From the Takahama-3 Reactor," 2003. Nuclear Regulatory Commission / ORNL.

[51] Sjoden, G., "An Efficient Exponential Directional Iterative Differencing Scheme for 3-D Sn Computations in XYZ Geometry," *Nuclear Science and Engineering*, vol. 155, pp. 179–189, 2007.

[52] Sjoden, G., Courau, T., Manalo, K., and Yi, C., "Automatically Tuned Adaptive Differencing Algorithm for 3-D Sn Implemented in PENTRAN," in *International Conference on Mathematics, Computational Methods & Reactor Physics*, (Saratoga Springs, NY), American Nuclear Society, 2009.

[53] Sjoden, G. and Haghighat, A., "A New Adaptive Differencing Strategy in the PENTRAN 3-D Parallel Sn Code," in *Transactions of the American Nuclear Society*, vol. 75, 1996.

[54] Sjoden, G. and Haghighat, A., "PENTRAN Code System: Parallel Environment Neutral-Particle TRANsport, Parallel Distributed Decomposition of Discrete Ordinates (Sn) in 3-D Cartesian Geometry, Users Guide to Version 9.4X.5 Series," tech. rep., 2008.

[55] Sjoden, G. and Haghighat, A., "PENTRAN- A 3-D Cartesian parallel SN code with angular, energy, and spatial decomposition," in *Proc. Join Int. Conf. on Mathematical Methods and Supercomputing for Nuclear Applications*, vol. 1, (Saratoga Springs, NY), 1997.

[56] Sjoden, G. and Haghighat, A., "The Exponential Directional Weighted (EDW) Sn Differencing Scheme in 3-D Cartesian Geometry," in *Proc. Join Int. Conf. on Mathematical Methods and Supercomputing for Nuclear Applications*, vol. 2, (Saratoga Springs, NY), 1997.

[57] SSI, *Falling Behind: International Scrutiny of the Peaceful Atom.* Strategic Studies Institute, 2008.

[58] Stamm'ler, R. and Abbate, M., *Methods of Steady-State Reactor Physics in Nuclear Design.* London: Academic Press, 1983.

[59] Sutter, H., 2008, "Going Superlinear,", [Online]. Available: http://www.drdobbs.com/cpp/going-superlinear/206100542.

[60] X-5 Monte Carlo Team, "MCNP - A General Monte Carlo N-Particle Transport Code, Version 5," Tech. Rep. LA-CP-03-0245, Los Alamos National Laboratory, 2005.

[61] Yang, W. S., Smith, M. A., Lee, C. H., Wollaber, A., Kaushik, D., and Mohamed, A., "Neutronics Modeling and Simulation of SHARP for Fast Reactor Analysis," *Nuclear Engineering and Technology*, vol. 42, no. 5, 2010.

[62] Yi, C., Sjoden, G., Mattingly, J., and Courau, T., "Computationally optimized multi-group cross section data collapsing using the YGROUP code," in *Proc. Amer. Nucl. Soc. Phys. Reactors Topical Meeting*, 2010.

[63] Yi, C., *Hybrid discrete ordinates and characteristics method for solving the linear Boltzmann equation.* PhD thesis, University of Florida, 2007.

[64] Yi, C., "XSMCNP: Conversion of Deterministic Cross Section Data Into MCNP Multigroup Format," 2009. University of Florida.