Spring 2018

Development and Monte Carlo validation of a finite element reactor analysis framework

Wayne J. Brewster

DEVELOPMENT AND MONTE CARLO VALIDATION OF A FINITE ELEMENT REACTOR ANALYSIS FRAMEWORK

by

WAYNE J. BREWSTER

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN NUCLEAR ENGINEERING

2018

Approved by

Ayodeji Babatunde Alajo, Advisor
Xin Liu
Gary E. Mueller

ABSTRACT

This study presents the development and Monte Carlo validation of a continuous Galerkin finite element reactor analysis framework. In its current state, the framework acts as an interface between the mesh preparation software GMSH and the sparse linear solvers in MATLAB, for the discretization and approximation of 1-D, 2-D, and 3-D linear partial differential equations. Validity of the framework is assessed from the following two benchmarking activities: the 2-D IAEA PWR benchmark; and the 2-D Missouri Science and Technology Reactor benchmark proposed within this study. The 2-D IAEA PWR multi-group diffusion benchmark is conducted with the following discretization schemes: linear, quadratic, and cubic triangular elements; linear and quadratic rectangular elements of mesh sizes 10, 5, 2, 1, 0.5 cm. Convergence to the reference criticality eigenvalue of 1.02985 is observed for all cases.

The proposed 2-D MSTR benchmark is prepared through translation of an experimentally validated 120w core configuration MCNP model into Serpent 2. Validation of the Serpent 2 model is attained from the comparison of criticality eigenvalues, flux traverses, and two 70-group energy spectrums within fuel elements D5 and D9. Then, a two-group 2-D MSTR benchmark of the 120w core configuration is prepared with the spatial homogenization methodology implemented within Serpent 2. Final validation of the framework is assessed from the comparison of criticality eigenvalues and spatial flux solutions of the diffusion and simplified spherical harmonics SP_3 models. The diffusion model resulted in a difference in reactivity of $\Delta\rho = -1673.93$ pcm and the SP_3 model resulted in a difference of $\Delta\rho = -777.60$ pcm with respect to the Serpent 2 criticality eigenvalues.

ACKNOWLEDGMENTS

First off, I would like to thank my advisor Dr. Alajo for giving me the opportunity to pursue research that truly interests me. I would also like to thank Dr. Gary Mueller and Dr. Xin Liu for being members of my committee, the faculty of the Nuclear Engineering Department for providing me an invaluable education, the IT RSS for setting up MCNP on the cluster, and Dr. Xiaoming He for introducing me to the world of finite elements. Finally, I would like to thank my family for their endless support.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

Page

LIST OF TABLES

Page

1. INTRODUCTION

The design and safe operation of a nuclear reactor requires an extensive characterization of its neutronic properties; thereby, allowing the precise manipulation of reactivity configurations and the determination of safe operating limits wherein the delicate balance of criticality is maintained. The most fundamental physical equation which provides a means to characterize the free motion of neutrons in a nuclear reactor is the linear neutron Boltzmann equation in which each physical process neutrons are gained or lost from a seven-dimensional (three in space, two in angle, energy, and time) phase space volume element forms the balance equation describing the expected neutron population. Furthermore, the probability of an interaction per path length (macroscopic cross section) that governs each individual reaction mode (parasitic absorption, scattering, and fission) is subject to change with the evolution of thermal-hydraulic, burn-up, and thermo-mechanical conditions. Consequently, nuclear reactors are multi-physical and multi-scale by nature; therefore, inclusion of all physical models is required for an accurate characterization of a nuclear reactor system [1], [2].

Ultimately, inclusion and simulation of all the governing physical models is non-trivial and a computationally expensive task when one considers full core modelling. In fact, acquisition of high-order, full core, steady-state approximations to the various forms of the neutron Boltzmann equation is computationally prohibitive and typically reserved for: (1) small scale simulations; (2) the preparation of benchmarks and multi-group constants for low-order approximation schemes (spatial homogenization); (3) the academic setting; and (4) final reactor design analysis. However, the foregoing high-order calculations are not used in practical circumstances such as fuel reload design analysis and core relicensing because of the frequency in which they are performed and the high computational cost of these methods. Nevertheless, the high-order solution methods are the bases for the low-order approximation schemes and can be divided into two distinct mathematical solution classes: (1) the Monte Carlo method; and (2) deterministic methods.

The fundamental idea behind the Monte Carlo method is acquisition of expected value of a random variable through the numerical simulation of randomly sampled events. Monte Carlo methods are amenable to neutron transport since the physical processes which govern neutron interactions is inherently stochastic. For neutron transport, the outcome of each individual neutron is randomly sampled and tracked throughout the defined geometry. Tallies are scored in the regions of interest such that various integral estimators provide point, surface, and cell fluxes. Criticality is also estimated by storing the neutrons generated from fission during the current cycle which are subsequently used as the source for the next cycle; therefore, changes of the source sizes over subsequent batches yield the criticality estimate.

The advantage of the Monte Carlo method is the capability to simulate exact physical processes in an arbitrary level of spatial detail. However, it is critical to note that with the expected value comes statistical uncertainty. Therefore, it is imperative to ensure that the conditions of the central limit theorem are met for the results to have significant meaning. When considering large reactor systems, the foregoing condition requires a considerable sample population size and batches which result in increased computational effort. The Monte Carlo method exacerbates the aforesaid computational burden when considering reactor burn-up analyses for large reactor systems because of the additional time component. Nevertheless, implementation of variance reduction techniques improves the precision of integral estimators which results in decreased computational effort such as the case for radiation shielding and general particle transport. The most notable Monte Carlo codes in use today are: (1) MCNP (Los Alamos National Laboratory); (2) OpenMC (Massachusetts Institute of Technology); (3) Serpent 2 (VTT Technical Research Centre of Finland); and (4) TRIPOLI (French Alternatives Energies and Atomic Energy Commission).

In contrast, deterministic methods rely upon the discretization of the independent variables wherein the original differential equation is reduced to a linear system. Further classification of the deterministic methods is based upon the treatment of the angular dependency in which each solution method takes advantage of distinct mathematical properties or numerical methods. The method of characteristics takes a unique approach of reformulation of the integrodifferential form of the neutron Boltzmann equation into an equivalent characteristic form. In short, the frame of reference shifts from an observation of a neutron relative to a fixed point in space as opposed to a reference in space. By projecting characteristic lines over the computational domain, the average value of the angular flux is computed by integrating over each characteristic track divided by the tracks total length.

The discrete ordinates method S_n relies upon discretization of the solid angle (angular component) into discrete direction cosines where a quadrature rule permits integration of the polynomials over the direction cosines. Another method requires expansion of the angular terms as an infinite spherical harmonics series. Truncation of such infinite series results in a set of partial differential equations known as the spherical harmonics P_n equations. Both the discrete ordinates method and spherical harmonics method require the discretization of the spatial component by either the finite difference or finite element method. However, the finite element method is more attractive than the finite difference method because of the former's amenability to irregular domains and capability to obtain higher order approximations with a fixed mesh. The finite difference method requires mesh refinement to improve the order of accuracy and may also produce non-invertible matrices when applied to non-cartesian geometries.

Limitation of the spherical harmonic series to the order of $n = 1$, and the elimination of the odd order moment in the even order equation, provides low-order simplified spherical harmonics equation SP_1 analogous to the diffusion equation derived from the neutron continuity equation and Fick's law. The only difference is the inclusion of the average cosine scattering angle in the proportionality constant in the SP_1 equation. Ultimately, this permits the extension of the SP_1 equation to reactors that exhibit moderate anisotropic scattering. The diffusion/SP_1 equations are the most widely used transport approximations in nodal, full-core analyses. Where, the method relies upon the production of multigroup constants by energy condensation and spatial homogenization of the cross sections using the infinite assembly lattice spectrum obtained by a high-order transport simulation [3]. Then the global homogeneous flux solution is approximated from the construction of the homogenized assemblies into the full core domain for nodal diffusion codes [4], [5].

Typical multi-physics computational paradigms used for production fuel reload analyses and core relicensing rely on the operator splitting method where the non-linear terms are decoupled [1]. Operator splitting permits the use of existing mono-disciplinary codes such that the output of one code is taken as the initial conditions for the next code where solutions are exchanged between the mono-disciplinary codes until the established convergence criteria are met. Since the operator splitting method is explicit in time where the order of accuracy is $O(H^1)$, the method requires time steps on the order of the dynamic time scale of the system [6]. Therefore, the operator splitting method is inefficient when applied to stiff multi-scale systems. This is the case for nuclear reactor systems since the neutronic time scale is on the order of 10^{-6} seconds when neglecting delayed neutrons and the heat transfer time scale is on the order of 10^0 to 10^1 seconds [1].

Recent advances in Jacobian-Free Newton Krylov (JFNK) subspace solvers and physics-based preconditioning has led to increased efficiency of implicit time integration techniques in the simultaneous solution of coupled non-linear equations [7]. However, due to the mathematical rigor and complexity of coupling multiple physics models in a unified framework, the research concerning the implementation of such methods is primarily left to national laboratories, or large university research groups. Where, the main group whose efforts are focused on the application of the JFNK methods to nuclear systems is the Multi-Physics Object-Oriented Simulation Environment (MOOSE) team at the Idaho National Laboratory [8], [9]. Unfortunately, without the ability to readily modify an already established framework, it is virtually impossible to conduct research in multi-physics methods development. Therefore, the aim of this research is to establish the foundation of a general finite element framework where future research can build upon and extend the frameworks capabilities to include non-linear multi-physics modelling.

1.1. RESEARCH OBJECTIVES

The objective of the thesis is the development and Monte Carlo validation of a diffusion and simplified spherical harmonics finite element reactor analysis framework. The objective includes the following relevant issues:

A. perform a preliminary 2-D IAEA PWR benchmark. The initial 2-D IAEA PWR benchmark is the most efficient methodology to obtain initial data to ascertain whether the proposed finite element framework can be correctly implemented to the multi-group neutron diffusion equation;

B. develop Serpent 2 model of the MSTR. Previous experimental MCNP validation of the MSTR model allows the construction of a validation chain between physical reality and multiple computer codes;

C. validate the Serpent 2 MSTR model to the previously validated MCNP model. Without validation of the Serpent 2 model, the link between the finite element framework and physical experiments cease to exist. The foregoing is true because the preparation of the proposed MSTR benchmark relies upon Serpent’s global flux solution to preserve the reaction rates in the process of spatial homogenization and energy condensation of the cross sections and multi-group constants;

D. preparation of stochastic multi-group parameters using the global flux distribution for the proposed 2-D MSTR benchmark. Stochastic generation of multi-group parameters permit the spatial homogenization and energy condensation using: (1) continuous-energy cross section data: (2) the global flux distribution; and (3) incorporation of spatial self-shielding. Pursuant the preparation of the 2-D MSTR benchmark in the foregoing manner results in the minimization of the spatial homogenization errors;

E. validate finite element reactor analysis framework with the 2-D MSTR benchmark. Validation of the finite element framework will hopefully demonstrate the capabilities of the framework allowing its application to reactor analysis. Furthermore, the framework can then serve as a foundation for further research concerning multi-physics simulation.

1.2. FINITE ELEMENT METHODS IN REACTOR PHYSICS

Application of the finite element method (FEM) to reactor analysis dates to the 1970's when diffusion codes were primarily based upon the finite difference method (FDM). Since then, a multitude of papers concerning the application and development of the FEM in reactor analysis have been published; therefore, it is not possible to cover the entirety of the FEMs in reactor analysis in this section. Nevertheless, research and development efforts which highlight the success of the FEM method spanning from the 1970's till the present day are presented.

One of the first papers concerning this matter demonstrated the applicability of the FEM method in a 2-D multigroup criticality code FEND [10]. The FEND code was utilized to approximate criticality eigenvalues and flux eigenvectors for a two-group in-homogenous test problem with Lagrangian linear triangular and bilinear rectangular discretization schemes [10]. Semenza et al. concluded that accurate eigenvalues and eigenvectors were attained with a relatively few nodal points which demonstrates the utility of the FEM method; however, computer memory limitations of the time required auxiliary memory devices for large problems that required a significant amount of nodal points [10].

Demonstration of the finite element methods utility [10] prompted further research concerning the efficiency of the method over the low-order finite difference method for three rector configurations: (1) two-group two-zone reactor; (2) four-group multizone 1000-MW(e) LMFBR mockup; and (3) two-group loosely coupled configuration [11]. Results of the study indicate that high-order FEMs were able to decrease the computational cost of the LMFBR case by a factor of 20 over the finite difference method with a 30% reduction in memory usage [11]. Furthermore, the FEM produced accurate results such that any error can be attributed to the diffusion theory approximation or approximations in the reactor model [11]. If the desired eigenvalue accuracy was to three decimal places, the high-order FEM yielded speed advantages up to a 50:1 ratio in the two-group two-zone reactor [11].

Instead of specifying the degrees of freedom as nodal values (Lagrangian finite elements), Hermitian finite elements specify the degrees of freedom as directional derivatives. The study by Kang and Hansen applied Hermite polynomials to space, energy, and time dependent neutron diffusion problems on rectangular meshes [12]. However, they had issues with the representation of singular points. Hebert solved this issue by utilizing Weierstrass-Erdmann type conditions which permits coupling of the solution over space regardless of singularities [13]. Hebert also implemented a mixed-dual variational formulation using Raviart-Thomas-Schneider elements in 3-D hexagonal geometry. Where, the Raviart-Thomas basis utilizes tensorial products of Legendre polynomials to represent the neutron flux [14]. The formulation was validated with the hexagonal 2-D IAEA benchmark and the 2-D/3-D Monju reactor benchmark [14]. The use of modified Dubiner's polynomials over hexagonal geometry using a fixed triangular mesh was also investigated [15]. Each hexagonal lattice was divided into six equilateral triangles and the order of accuracy was increased by introducing higher-order modified Dubiner's polynomials in the expansion.

More recent efforts have been focused on increasing the computational efficiency of the FEM applied to the multigroup diffusion equation through adaptive mesh refinement [16]. The proposed adaptive algorithm relies on separate meshes for each energy group to take advantage of the smoothness of each energy dependent solution. The calculation starts with a coarse mesh where cell errors are calculated to discern which regions need refinement or coarsening. Numerical results were obtained for the two-group 2-D IAEA PWR benchmark, the two-group 2-D OECD-L-336 fuel assembly benchmark, and a 3-D seven-group problem. Wang concluded that the adaptive refinement algorithm led to faster solutions times for a given order of accuracy over uniform mesh refinement. Wang also concluded that the adaptive mesh refinement led to solution accuracy that was previously impossible, or to the desired accuracy for the first time [16].

2. FINITE ELEMENT METHODS FOR ELLIPTIC PDES

The finite element methods are a mathematical tool which permits the approximation of partial differential equations in variational form over a space V. Through the discretization of the computational domain **Ω** into finite elements and the construction of finite dimensional subspaces V_h of the space V, the approximate discrete solution can be obtained through the linear combination of undetermined coefficients and piece-wise polynomial basis functions $v_h \in V_h$. Typical formulations specify the degrees of freedom of as point values (Lagrange finite elements) or directional derivatives (Hermite finite elements). However, for the purposes of this thesis, only the continuous Galerkin method and Lagrangian type of finite elements are considered. Nevertheless, readers should be aware that other finite element formulations exist, i.e., mixed finite element methods, discontinuous Galerkin methods. In constructing this chapter, it was assumed that the reader has limited exposure to functional analysis, so instead of providing lengthy mathematical proofs, only a summary of their implications is presented. Interested readers may resort to the citations for a deeper understanding of the mathematical proofs.

2.1. HOMOGENOUS DIRICHLET POISSON PROBLEM

Consider the second order elliptic Poisson problem:

$$\begin{cases} -\nabla \cdot c(\vec{r})\nabla u(\vec{r}) = f(\vec{r}) \; in \; \Omega, \\ u(\vec{r}) = g \; on \; d\Omega. \end{cases} \tag{2.1}$$

Where, $f(\vec{r})$ and $c(\vec{r})$ are known functions on Ω, $g(\vec{r})$ is a known function on $d\Omega$, and $u(\vec{r})$ is the unknown solution. The first step in the finite element formulation is to transform the strong problem into an equivalent weak problem. First, multiply both sides of the equation by a test function $v(\vec{r})$ and integrate over the domain Ω. Note: For clarity, the variables spatial dependence has been omitted.

$$-\int_{\Omega} \nabla \cdot (c\nabla u) v \, d\Omega = \int_{\Omega} f v \, d\Omega. \tag{2.2}$$

Applying Green's theorem (multi-dimensional integration by parts) to the differential terms on the LHS.

$$\int_{\Omega} \nabla \cdot (c\nabla u) v \, d\Omega = \int_{d\Omega} (c\nabla u \cdot \vec{n}) v \, ds - \int_{\Omega} c\nabla u \nabla v \, d\Omega\,. \tag{2.3}$$

The Poisson's equation becomes,

$$\int_{\Omega} c\nabla u \nabla v \, d\Omega - \int_{d\Omega} (c\nabla u \cdot \vec{n}) v \, ds = \int_{\Omega} f v \, d\Omega. \tag{2.4}$$

Since the solution u is given on the boundary $d\Omega$ by g, the test function v is chosen such that $v = 0$ on $d\Omega$. Thus, the strong formulation of the Poisson problem is reformulated into an equivalent weak form (Equation 2.5). Essentially, reformulation of the strong problem into the weak form relaxes the derivative requirement. It is no longer required that u be twice differentiable. Instead, weaker requirements have been imposed such that u' exist and be square integrable.

$$\int_{\Omega} c\nabla u\nabla v\, d\Omega = \int_{\Omega} fv\, d\Omega. \tag{2.5}$$

The next step is to find a space V where the derivatives of the functions in this space are square integrable.

2.1.1. Weak Formulation. A space that satisfies the weak form requirements is the Sobolev space $H^m(\Omega)$:

$$H^m(\Omega) = \left\{v \in L^2(\Omega): \frac{\partial^{\alpha_1+\alpha_2} v}{\partial x^{\alpha_1} \partial y^{\alpha_2}} \in L^2(\Omega), \forall \alpha_1 + \alpha_2 = 1, \dots, m\right\}, \tag{2.6}$$

where the Lebesgue $L^p(\Omega)$ space

$$L^p(\Omega) = \left\{v: \Omega \to R: \int_{\Omega} V^p\, dxdy < \infty\right\}. \tag{2.7}$$

Therefore, the functions u and v must belong to the Sobolev spaces [17]. Thus, the weak formulation: find $u \in H^1(\Omega)$ such that $\forall v \in H_0^1(\Omega),\ a(u,v) = (f,v)$. Where, the continuous V-elliptic bilinear and continuous linear form are defined as:

$$a(u,v) = \int_{\Omega} c\nabla u\nabla v\, d\Omega, \tag{2.8}$$

$$(f,v) = \int_{\Omega} fv\, d\Omega \tag{2.9}$$

and are assumed to satisfy the Lax-Milgram lemma [18]. Ultimately, the Lax-Milgram lemma proves that the variational problem (Eq. 2.5) is well-posed and that its solution exists, is unique, and depends continuously on f [18], [19].

2.1.2. Galerkin Formulation. Since an infinite number of test functions v exist in the space V such that u is a weak solution of the PDE, it is necessary to further impose restrictions on the vector space. One such approach is the Galerkin method which characterizes a finite dimensional space U_h to permit approximation of the infinite dimensional abstract variational problem. Let's introduce a triangulation T over the set $\overline{\Omega}$, where Ω is subdivided into finite elements K, that satisfy the following properties: (1) $\overline{\Omega} = \cup_{k \in T} K$; (2) for every element $K \in T$ the interior of K° is non-empty; (3) the intersection of the element interiors is empty; (4) the boundary of ∂K is Lipschitz-continuous; (5) any face of an element K in the triangulation is either a subset of the boundary, or a face of another element [17]. Then for each element within the triangulation, the polynomial function space is defined as $P_k = \{v_{h|K};\ \ v_h \in U_h\}$. Lastly, the space U_h should contain at least one canonical basis where the corresponding basis functions have supports that are small as possible; meaning the set of points in the space U_h where the basis functions are non-zero is minimized.

Assume a finite dimensional subspace $U_h \subset H^1(\Omega)$. Then, the Galerkin formula: find $u_h \in U_h$ such that satisfies the bilinear and linear form $a(u_h, v_h) = (f, v_h)\ \ \forall v_h \in U_h$. Where,

$$a(u_h, v_h) = \int_{\Omega} c \nabla u_h \nabla v_h \, d\Omega, \tag{2.10}$$

$$(f, v_h) = \int_{\Omega} f v_h \, d\Omega. \tag{2.11}$$

Let $\{\phi_j\}_{j=1}^{Nb}$ be a basis of the continuous piecewise function space U_h, where Nb is the total number of basis functions. Since $u_h \in U_h = span\{\phi_j\}_{j=1}^{Nb}$, the finite element solution u_h is a linear combination of the unknown coefficients u_j and known basis functions $\phi_{j.}$

$$u_h = \sum_{j=1}^{Nb} u_j \phi_{j.} \tag{2.12}$$

Due to the finite element space restriction in which the subspace U_h must contain at least one canonical basis, the basis functions ϕ_j are only non-zero on the finite elements that share the node X_k.

$$\phi_j(X_k) = \delta_{jk} = \begin{cases} 0, & if\ j \neq k, \\ 1, & if\ j = k. \end{cases} \tag{2.13}$$

Then,

$$u_h(X_k) = \sum_{j=1}^{Nb} u_j \phi_j(X_k) = u_k. \tag{2.14}$$

Thus, the coefficient u_j is the approximate solution at the node X_j. Next, choose a test function such that $v_h = \phi_i (i = 1, \dots, Nb)$. Hence the finite element formulation,

$$\int_\Omega c\nabla\left(\sum_{j=1}^{Nb} u_j \phi_j\right) \cdot \nabla\phi_i \, d\Omega = \int_\Omega f\phi_i \ d\Omega \tag{2.15}$$

which is equivalent to

$$\sum_{j=1}^{Nb} u_j \left[\iint_\Omega \; c\nabla\phi_j \cdot \nabla\phi_i \; d\Omega\right] = \int_\Omega \; f\phi_i \; d\Omega, \;\; i = 1, \dots, Nb. \tag{2.16}$$

Evaluating the integrals for $i, j = 1, \dots, Nb,$ forms a linear system for the unknown coefficients u_j (finite element solution). In fact, the matrix formed from the inner product on the LHS will be sparse (since most of the integrals will be zero, $i \neq j$) and always invertible due to the original assumption of a V-elliptic bilinear form in the Lax-Milgram lemma [18].

2.1.3. Matrix Formulation. Expression of the finite element formulation in matrix notation will provide the basis for the finite element framework as the code structure will revolve around evaluating and solving for the components of the matrix formulation. The inner product on the LHS is the stiffness matrix, where in matrix notation

$$A = \left[a_{ij}\right]_{ij=1}^{Nb} = \left[\iint_\Omega \; c\nabla\phi_j \cdot \nabla\phi_i \; dxdy\right]_{i.j=1}^{Nb}. \tag{2.17}$$

The RHS load vector

$$\vec{b} = [b_i]_{i=1}^{Nb} = \left[\iint_\Omega \; f\phi_i \; dxdy\right]_{i=1}^{Nb}. \tag{2.18}$$

The unknown vector that contains the finite element solution

$$\vec{X} = \left[u_j\right]_{j=1}^{Nb}. \tag{2.19}$$

Finally, combining all the components results in the linear algebraic system $A\vec{X} = \vec{b}$.

2.2. MIXED BOUNDARY CONDITIONS

Unlike the pure homogenous Dirichlet case, where the boundary conditions are explicitly imposed after the formulation of the linear system. The natural Neumann and Robin boundary conditions are handled implicitly during the transformation of the strong problem into its equivalent weak form. Consequently, extra boundary integrals will be introduced in the formulations where the boundary integrals are surface integrals for three-dimensional domains and line integrals for two-dimensional domains. This section will only demonstrate the derivation of the Dirichlet/Neumann and Dirichlet/Robin formulations for the Poisson problem. However, the same processes are applied to other boundary value problems with any combination of mixed boundary conditions.

2.2.1. Dirichlet/Neumann. Consider the second order Poisson problem from the previous section. Instead of imposing the homogenous Dirichlet boundary condition let's define a split boundary with one portion defined by the essential Dirichlet condition and the other portion with the natural Neumann condition.

$$\begin{cases} -\nabla \cdot c\nabla u = f \ in \ \Omega, \\ u = g \ on \ d\Omega/\Gamma_1, \\ \nabla u \cdot \vec{n} = p \ on \ \Gamma_1 \subset \partial\Omega. \end{cases} \tag{2.20}$$

Recall that the weak formulation for the Poisson equation is

$$\int_{\Omega} c\nabla u \nabla v \, d\Omega - \int_{d\Omega} (c\nabla u \cdot \vec{n}) v \, ds = \int_{\Omega} fv \, d\Omega. \tag{2.21}$$

Since the solution is given by $u = g$ on $\partial\Omega/\Gamma_1$; a test function is chosen such that $v = 0$ on $\partial\Omega/\Gamma_1$. Therefore, the boundary term in the weak formulation becomes

$$\int_{d\Omega} (c\nabla u \cdot \vec{n})v\, ds = \int_{\Gamma_1} (c\nabla u \cdot \vec{n})v\, ds + \int_{\partial\Omega/\Gamma_1} (c\nabla u \cdot \vec{n})v\, ds \tag{2.22}$$

$$= \int_{\Gamma_1} cpvds.$$

Substituting the new boundary term back into the problem

$$\int_{\Omega} c\nabla u \nabla v\, d\Omega - \int_{\Gamma_1} cpv\, ds = \int_{\Omega} fv\, d\Omega \tag{2.23}$$

which is equivalent to

$$\int_{\Omega} c\nabla u \nabla v\, d\Omega = \int_{\Omega} fv\, d\Omega + \int_{\Gamma_1} cpv\, ds. \tag{2.24}$$

Thus, the weak formulation: find $u \in H^1(\Omega)$ such that $a(u, v) = (f, v)\ \forall v \in H^1\ (\Omega)$. Where,

$$a(u, v) = \int_{\Omega} c\nabla u \nabla v\, d\Omega, \tag{2.25}$$

$$(f, v) = \int_{\Omega} fv\, d\Omega + \int_{\Gamma_1} cpv\, ds. \tag{2.26}$$

Without going through the full Galerkin and matrix formulation presented in the homogenous Dirichlet Poisson section (the procedure is the same except for the inclusion of the new boundary term) it is evident that the matrix formulation will include the addition of a new vector to the linear form on the RHS. Assume $U_h \subset H^1(\Omega)$ then the Galerkin formulation: find $u_h \in U_h$ such that $a(u_h, v_h) = (f, v_h)\ \ \forall v_h \in U_h$. Where,

$$a(u_h, v_h) = \int_{\Omega} c\nabla u_h \nabla v_h \, d\Omega, \tag{2.27}$$

$$(f, v_h) = \int_{\Omega} f v_h \, d\Omega + \int_{\Gamma_1} cpv_h \, ds. \tag{2.28}$$

A test function is chosen such that $v_h = \phi_i (i = 1, \ldots, Nb)$. Hence, the additional term in the matrix formulation which results from the Neumann boundary integral

$$\vec{\mathrm{v}} = [\mathrm{v}_i]_{i=1}^{Nb} = \left[\int_{\Gamma_1} cp\phi_i \, ds\right]_{i=1}^{Nb}. \tag{2.29}$$

Modification of the vector results in $\tilde{\vec{b}} = \vec{b} + \vec{v}$ and the linear system of algebraic equations becomes $A\vec{X} = \tilde{\vec{b}}$.

2.2.2. Dirichlet/Robin. Consider the following second order Poisson problem with Dirichlet and Robin boundary conditions:

$$\begin{cases} -\nabla \cdot c\nabla u = f \ in \ \Omega, \\ u = g \ on \ d\Omega/\Gamma_2, \\ \nabla u \cdot \vec{n} + ru = q \ on \ \Gamma_2 \subseteq \partial\Omega. \end{cases} \tag{2.30}$$

Recall the weak formulation for the Poisson problem:

$$\int_{\Omega} c\nabla u \nabla v \, d\Omega - \int_{d\Omega} (c\nabla u \cdot \vec{n}) v \, ds = \int_{\Omega} fv \, d\Omega. \tag{2.31}$$

Since the solution is given by $u = g$ on $\partial\Omega/\Gamma_2$, a test function is chosen such that $v = 0$ on $\partial\Omega/\Gamma_2$; therefore, the boundary term in the weak formulation with $\nabla u \cdot \vec{n} = q - ru$

$$\int_{d\Omega}(c\nabla u\cdot\vec{n})v\,ds=\int_{\Gamma_2}(c\nabla u\cdot\vec{n})v\,ds+\int_{\partial\Omega/\Gamma_2}(c\nabla u\cdot\vec{n})v\,ds \tag{2.32}$$

$$=\int_{\Gamma_2}c(q-ru)vds=\int_{\Gamma_2}cqvds-\int_{\Gamma_2}cruvds\,.$$

Substituting the new boundary term back into the weak formulation gives

$$\int_{\Omega}c\nabla u\nabla v\,d\Omega-\left[\int_{\Gamma_2}cqvds-\int_{\Gamma_2}cruvds\right]=\int_{\Omega}fv\,d\Omega, \tag{2.33}$$

which is equivalent to

$$\int_{\Omega}c\nabla u\nabla v\,d\Omega+\int_{\Gamma_2}cruvds=\int_{\Omega}fv\,d\Omega+\int_{\Gamma_2}cqvds. \tag{2.34}$$

Thus, the weak formulation: find $u\in H^1(\Omega)$ such that $a(u,v)=(f,v)\;\forall v\in H^1\,(\Omega)$.

Where,

$$a(u,v)=\int_{\Omega}c\nabla u\nabla v\,d\Omega+\int_{\Gamma_2}cruv\,ds, \tag{2.35}$$

$$(f,v)=\int_{\Omega}fv\,d\Omega+\int_{\Gamma_2}cqv\,ds. \tag{2.36}$$

Assume $U_h\subset H^1(\Omega)$. Then the Galerkin formulation: find $u_h\in U_h$ such that $a(u_h,v_h)=(f,v_h)\;\;\forall v_h\in U_h$. Where,

$$a(u_h,v_h)=\int_{\Omega}c\nabla u_h\nabla v_h\,d\Omega+\int_{\Gamma_2}cru_hv_h\,ds, \tag{2.37}$$

$$(f, v_h) = \int_{\Omega} f v_h \, d\Omega + \int_{\Gamma_2} cqv_h \, ds. \tag{2.38}$$

A test function is chosen such that $v_h = \phi_i (i = 1, \ldots, Nb)$. As a result of the imposition of the Robin boundary condition, two new integrals have arisen. Hence, the additional terms in the matrix formulation:

$$\vec{\mathrm{w}} = [\mathrm{w}_i]_{i=1}^{Nb} = \left[\iint_{\Gamma_1} cp\phi_i \, ds\right]_{i=1}^{Nb}; \tag{2.39}$$

$$R = \left[r_{ij}\right]_{i,j=1}^{Nb} = \left[\iint_{\Gamma_2} cr\phi_j\phi_i \, ds\right]_{i,j=1}^{Nb}. \tag{2.40}$$

The modified matrix and vector are defined as: $\tilde{A} = A + R$, and $\tilde{\vec{b}} = \vec{b} + \vec{w}$. Thus, the resulting linear algebraic system is $\tilde{A}\vec{X} = \tilde{\vec{b}}$.

2.3. BASIS FUNCTIONS

Recall from section 2.1 that the unknown solution u to the original Poisson equation can be approximated by a function u_h through the linear combination of undetermined coefficients u_j and basis functions ϕ_j. By partitioning the computational domain into nodal finite elements (Lagrangian elements) K and defining a polynomial basis with small supports over the elements, the basis functions are only non-zero when they are evaluated on elements adjacent to the node. Thus, u_j is the approximate nodal solution at the node X_j. For this to be true, the basis functions must be constructed from the elements nodal values. Since the partitioning of the domain into finite elements is completely arbitrary an inverse affine map is utilized to construct and evaluate the local basis functions on an arbitrary element. To demonstrate this idea, the derivation of the linear triangular element will be presented. Since the process is the same for other elements, the higher-order triangular elements, quadrangle elements, and tetrahedral elements are included in Appendix A.

2.3.1. Linear Triangular Element. Figure 2.1 depicts the characterization of the reference linear triangular element by its three vertexes. Before specifying the nodal order let's introduce the following notation to distinguish between the vertexes of the reference element and the local element. The vertexes and coordinates associated with the reference element are denoted by $\hat{A}_i(\hat{x}, \hat{y})$ and the arbitrary local element by $A_i(x, y)$. Where, i is the node number. Ordering the element vertexes are done in a counter clockwise fashion starting from $\hat{A}_1(0, 0)$ since the surface normal vector is chosen to be positive when the vector points out of this page. The next step is to construct the linear Lagrangian reference basis functions $\hat{\psi}_j(\hat{A}_i)$ over the reference element.

The linear Lagrangian interpolation polynomial in two-dimension is defined as:

$$\hat{\psi}_j(\hat{x}, \hat{y}) = a_j\hat{x} + b_j\hat{y} + c_j \tag{2.41}$$

such that

$$\hat{\psi}_j(\hat{A}_i) = \delta_{ij} = \begin{cases} 0, & if\ j \neq i, \\ 1, & if\ j = i, \end{cases} \ for\ i, j = 1, 2, 3. \tag{2.42}$$

By the previous definition of the reference basis function, the following system of equations is obtained for the coefficients of the first reference basis function when $j = 1$ and $i = 1, 2, 3.$

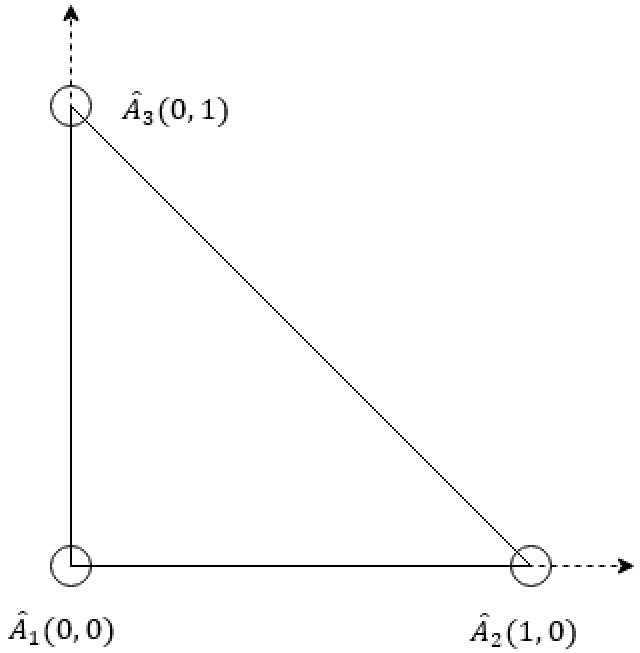


Figure 2.1 Linear triangular reference element

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} a_1 \\ b_1 \\ c_1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}. \tag{2.43}$$

Solving for the coefficients results in $a_1 = -1, b_1 = -1$, and $c_1 = 1$. Thus, the first reference basis function is

$$\hat{\psi}_1(\hat{x}, \hat{y}) = -\hat{x} - \hat{y} + 1. \tag{2.44}$$

Repeating the process to obtain the coefficients for the two remaining basis functions results yields:

$$\hat{\psi}_2(\hat{x}, \hat{y}) = \hat{x}, \tag{2.45}$$

$$\hat{\psi}_3(\hat{x}, \hat{y}) = \hat{y}. \tag{2.46}$$

2.3.2. Affine Mapping. Establishing an invertible affine mapping F_k permits the construction of the local basis functions over an arbitrary element from the previously derived reference basis functions

$$F_k: \begin{pmatrix} x \\ y \end{pmatrix} \in R^2 \rightarrow F_k \begin{pmatrix} x \\ y \end{pmatrix} = M_k \cdot \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} + b_k. \tag{2.47}$$

Where, $M_k(\hat{x}, \hat{y})$ is an invertible matrix and b_k is a vector in R^2. Essentially, the affine mapping preserves the geometric definition of the element when mapping to and from the reference and arbitrary local element. Let's consider the following affine map

$$F_k \begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \cdot \begin{pmatrix} \hat{x}_i \\ \hat{y}_i \end{pmatrix} + \begin{pmatrix} b_x \\ b_y \end{pmatrix}. \tag{2.48}$$

The transformation maps the vertexes of the reference element to the local element

$$\hat{A}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = A_1, \tag{2.49}$$

$$\hat{A}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = A_2, \tag{2.50}$$

and

$$\hat{A}_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} x_3 \\ y_3 \end{pmatrix} = A_3. \tag{2.51}$$

To obtain the complete matrix M_k the map is evaluated for the three cases mentioned above. For the first case when $i = 1$ the mapping of Equation 2.49 yields

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} b_x \\ b_y \end{pmatrix}. \tag{2.52}$$

Thus, the vector b_k becomes

$$\begin{pmatrix} b_x \\ b_y \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}. \tag{2.53}$$

Evaluation of the other two mappings $\hat{A}_2 \rightarrow A_2$ and $\hat{A}_3 \rightarrow A_3$ yields the complete affine mapping

$$F_k \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{pmatrix} \cdot \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} + \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}. \tag{2.54}$$

Inverting the affine map yields the transformation of a point inside the interior of a local element (x, y) to the reference element $(\hat{x}, \hat{y})$.

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \frac{1}{(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)} \begin{pmatrix} y_3 - y_1 & x_1 - x_3 \\ y_1 - y_2 & x_2 - x_1 \end{pmatrix} \cdot \begin{pmatrix} x - x_1 \\ y - y_1 \end{pmatrix} \tag{2.55}$$

Thus, the reference element coordinates in terms of the local element vertexes and interior point coordinates

$$\hat{x} = \frac{(y_3 - y_1)(x - x_1) + (x_1 - x_3)(y - y_1)}{(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)}, \tag{2.56}$$

$$\hat{y} = \frac{(y_1 - y_2)(x - x_1) + (x_2 - x_1)(y - y_1)}{(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)}. \tag{2.57}$$

It is now permissible to define the local basis functions from the preceding definitions of the inverse affine mapping.

2.3.3. Local Basis Functions. The local basis functions defined over an arbitrary element can be derived from the previously established reference basis functions through the affine map and chain rule. Let's consider the n^{th} element of the set of elements $\sum_{n=1}^{N} K_n$ where the vertexes of the n^{th} element are A_{n1}, A_{n2}, and A_{n3}. The coordinates of the three vertexes are $A_{ni} = \binom{x_{ni}}{y_{ni}}$ for $i = 1, 2, 3$. Then the three local basis functions over the n^{th} element are $\psi_{ni}(x_{ni}, y_{ni}) = \hat{\psi}_i(\hat{x}, \hat{y})$ for $i = 1, 2, 3$. Utilizing the chain rule yields the partial derivatives of the local basis functions of the n^{th} element in terms of the reference basis functions and the inverse affine map. Recall from the weak formulation of the Poisson equation that the inner product of the first order derivatives must be evaluated; therefore, the first order partial derivatives of the local basis functions on the n^{th} element are

$$\frac{\partial \psi_{ni}}{\partial x} = \frac{\partial \hat{\psi}_i}{\partial \hat{x}} \frac{\partial \hat{x}}{\partial x} + \frac{\partial \hat{\psi}_i}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial x} = \frac{\partial \hat{\psi}_i}{\partial \hat{x}} \frac{y_{n3} - y_{n1}}{J} + \frac{\partial \hat{\psi}_i}{\partial \hat{y}} \frac{y_{n1} - y_{n2}}{J}, \tag{2.58}$$

and

$$\frac{\partial \psi_{ni}}{\partial y} = \frac{\partial \hat{\psi}_i}{\partial \hat{x}} \frac{\partial \hat{x}}{\partial y} + \frac{\partial \hat{\psi}_i}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial y} = \frac{\partial \hat{\psi}_i}{\partial \hat{x}} \frac{x_{n1} - x_{n3}}{J} + \frac{\partial \hat{\psi}_i}{\partial \hat{y}} \frac{x_{n2} - x_{n1}}{J}. \tag{2.59}$$

Where, $J = (x_{n2} - x_{n1})(y_{n3} - y_{n1}) - (x_{n3} - x_{n1})(y_{n2} - y_{n1})$. The second order partial derivatives of the local basis functions are derived when considering quadratic interpolation polynomials. Such derivations are included in Appendix along with the rectangular and tetrahedral elements.

3. FINITE ELEMENT FRAMEWORK

The objective concerning implementation of the finite element method in computers is to form the linear system of algebraic equations through numerical evaluation of the integrals set forth by the matrix formulation to solve for the unknown coefficients (nodal values). Thus, the framework is broken into five main sub routines: (1) stiffness matrix assembly; (2) load vector assembly; (3) application of Dirichlet boundary conditions; (4) Neumann boundary condition vector assembly; and (5) Robin boundary condition matrix and vector assembly. Application of the foregoing modular approach allows the user to call only the necessary functions required to form the stiffness matrices and load vectors arising from the matrix formulation of a partial differential equation. Modularity also allows the ease of development of new functionalities under the framework. For instance, if a desired problem requires a specific formulation, or new functionality, the framework can be extended without modification of prior developments.

Implementation of the FEM framework pursuant the use of MATLAB results in: (1) simplicity; (2) access to sparse linear solvers; and (3) rapid development time. However, the downside of the decision to use MATLAB is reduced efficiency and scalability. Nevertheless, implementation of the framework in MATLAB demonstrates the framework’s capabilities and potential for further development in a compiled computer language. In terms of future development, the MATLAB code provides a solid foundation upon which future algorithms and framework extensions can be tested before the investment of development time required for their implementation in traditional compiled languages (FORTAN, C++, etc.). Finally, for persons that wish to further the development of the FEM framework, the MATLAB code presents the current state of the framework in the highest possible level thereby minimizing the time required to understand the inner workings of the framework.

The flowchart in Figure 3.1 illustrates the logical flow of the FEM framework where each general constituent represents a collection of functions required to carry out the underlying task. The first step is to prepare the computational domain in the open source GMSH: a 2D/3D meshing software [20]. A parsing function reads the data output from GMSH in ASCII format and processes the data into the correct format required by the FEM framework [20]. Then, the nuclear data is read in from the Serpent 2 output or by manual specification of the nuclear data. The solver that is developed for a specific partial differential equation (based on the matrix formulation) calls the stiffness matrix and load vector assembly routines based upon the number of integrals in the matrix formulation. After the stiffness matrix and load vector assembly, the framework checks each individual boundary condition type to discern which boundary condition functions to call. Lastly, the linear algebraic system is solved for the undetermined coefficients u_j. Presentation of the algorithms initially require that the user of the framework fully understand the data structure upon which the algorithms are built.

3.1.1. Data Structure. Consideration of a simple 2-D square domain (Figure 3.2) with a side length of $l = 1$ that is centered about the point $(0.5, 0.5)$ allows demonstration of the data structure. If the computational domain is discretized into structured triangular elements with $\Delta_{x,y}= 0.5$ whose nodal points are represented by linear interpolation polynomials. For this demonstration the nodes are ordered starting from node #1 at the point (0, 0). The node #2 would correspond to the point at (0, 0.5)

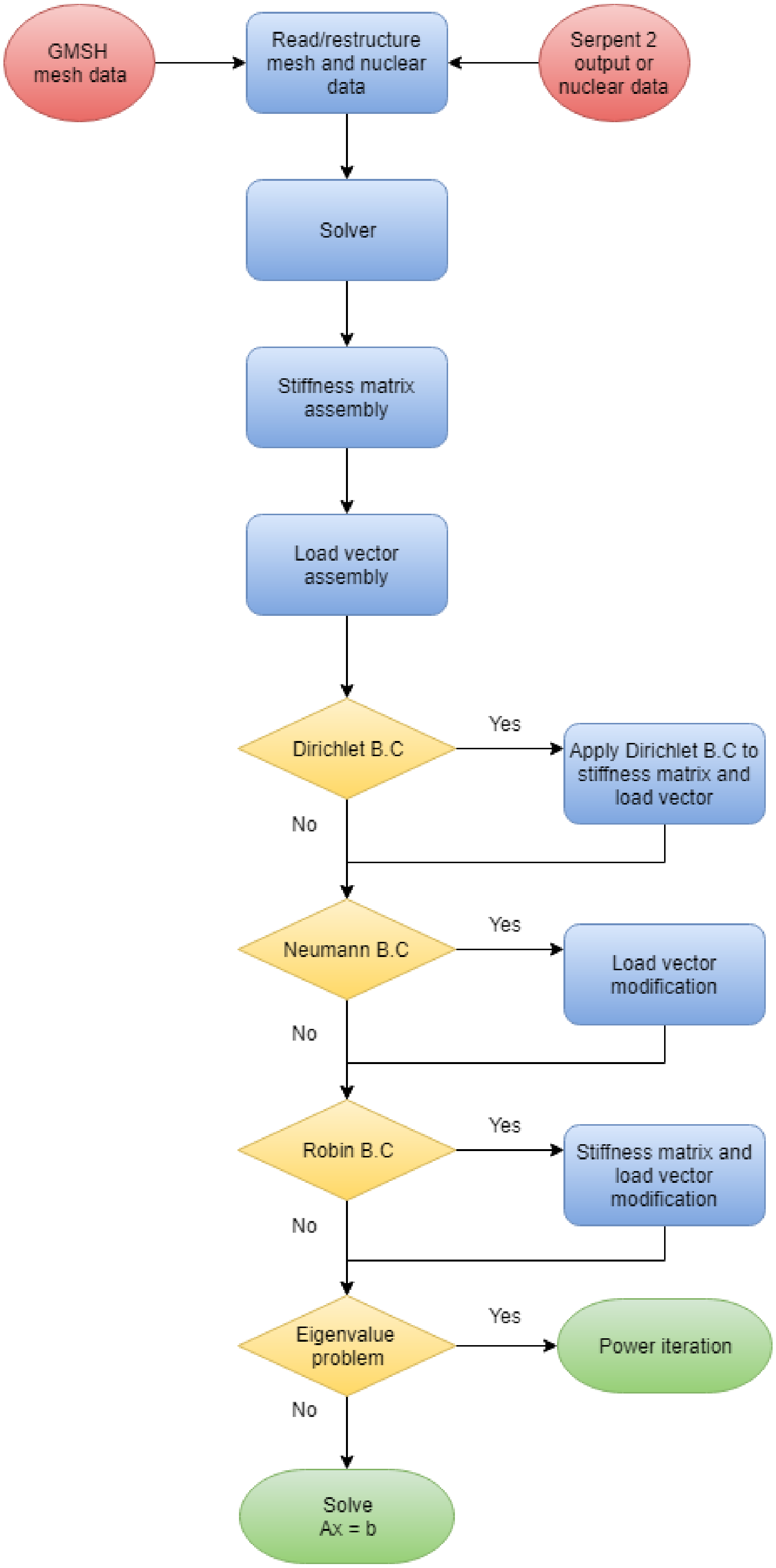

GMSH mesh data
Read/restructure mesh and nuclear data
Serpent 2 output or nuclear data
Solver
Stiffness matrix assembly
Load vector assembly
Dirichlet B.C
Yes
Apply Dirichlet B.C to stiffness matrix and load vector
No
Neumann B.C
Yes
Load vector modification
No
Robin B.C
Yes
Stiffness matrix and load vector modification
No
Eigenvalue problem
Yes
Power iteration
No
Solve Ax = b

Figure 3.1 Finite element framework flowchart for time independent problems

and would continue until all nodes are ordered in a column wise fashion. Although this structured node ordering is chosen for demonstration purposes, the framework does not impose any strict requirements on the node ordering. For instance, node #1 may be the center node at (0.5, 0.5).

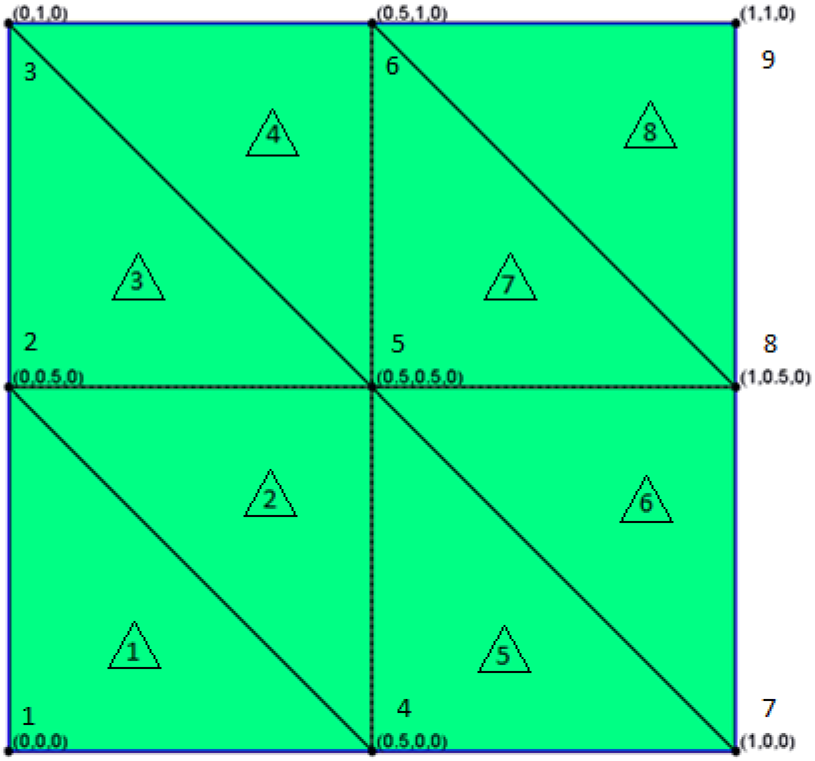


Figure 3.2 Example square mesh with triangular elements

Define two matrices to store the coordinates of all mesh nodes and the global basis function indices of all the mesh elements: (1) *node_coordinates*; (2) *global_indices*. The n^{th} column index of the *node_coordinates* matrix stores the coordinates of the n^{th} mesh node such that the first row stores the x-coordinate and the second row stores the y-coordinate. The j_{th} column of the *global_indices* matrix stores the global basis function indices of the j^{th} mesh element. Recall that the node ordering of the reference triangle is done in a counter-clockwise fashion (see Figure 2.1).; thus, the k^{th} row of the *global_indices* stores the global node index of $A_k(x,y)$ of the j^{th} mesh element. The two information matrices for the mesh in Figure 3.2:

$$node\_coordinates = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0.0 & 0.0 & 0.0 & 0.5 & 0.5 & 0.5 & 1.0 & 1.0 & 1.0 \\ 0.0 & 0.5 & 1.0 & 0.0 & 0.5 & 1.0 & 0.0 & 0.5 & 1.0 \end{pmatrix};$$

$$global\_indices = \begin{pmatrix} 1 & 2 & 2 & 3 & 4 & 5 & 5 & 6 \\ 4 & 4 & 5 & 5 & 7 & 7 & 8 & 8 \\ 2 & 5 & 3 & 6 & 5 & 8 & 6 & 9 \end{pmatrix}.$$

For instance, the 7^{th} mesh element (column 7 in $global\_indices$) would have the node coordinates $A_1(0.5, 0.5)$, $A_2(1.0, 0.5)$, and $A_3(0.5, 1.0)$.

The information regarding the boundary conditions is stored in a vector and matrix: (1) *boundary_nodes*; (2) *boundary_edges*. For the boundary edges that are specified with the Dirichlet boundary conditions, the global boundary node index along those edges are stored in the *boundary_nodes* vector. If all the boundary edges in the mesh in Figure 3.2 are specified as Dirichlet, the *boundary_nodes* matrix is

$$boundary\_nodes = \begin{pmatrix} 1 & 4 & 7 & 8 & 9 & 6 & 3 & 2 \end{pmatrix}.$$

Again, the framework does not require any specific order for which the global node index of the Dirichlet nodes must be stored.

Depending on the dimensionality of a problem, the Neumann and Robin boundary integrals are surface integral for 3-D and line integrals for 2-D; therefore, the information needed to evaluate these integrals are stored differently. For the 3-D case, the information is stored in a matrix *boundary_surface* whose structure is identical to that of the *global_indices* matrix of the 2-D problem. If the mesh in Figure 3.2 was a boundary surface of a cube the *boundary_surface* matrix would be identical to the information in the matrix $global\_indices$.

For the 2-D case where the boundary is an edge, the information is stored in a matrix *boundary_edges*. Thus, the matrix for the mesh in Figure 3.2 where all the boundary edges are Dirichlet except the right-side boundary edge which is specified as Neumann boundary

$$boundary\_edges = \begin{pmatrix} 1002 & 1002 \\ 6 & 8 \\ 7 & 8 \\ 8 & 9 \end{pmatrix}.$$

The first row stores the boundary condition identifier (1002 for Neumann and 1003 for Robin), the second-row stores the mesh element number, and rows three and four store the beginning and ending global node index of the edge. Note: the start and end nodes of a boundary edge are ordered in a counter-clock wise fashion.

To handle interface problems that require material dependent constants or functions; a physical group vector stores the numerical identifier which is used to call the correct mesh element data when evaluating the matrix formulation integrals. For demonstration purposes let’s consider the mesh in Figure 3.2 where the mesh is divided into two regions such that the interface is the line $x = 0.5$. The region to the left of the interface will be region #10 and the region to the right will be region #20. Thus, the physical-group matrix for this problem is

$$physical\_group = \begin{pmatrix} 10 & 10 & 10 & 10 & 20 & 20 & 20 & 20 \end{pmatrix}.$$

Here, the j_{th} column of the $physical\_group$ vector corresponds to the global basis indexing of the j_{th} column of the *global_indices* matrix.

3.1.2. Stiffness Matrix and Load Vector Assembly. Recall the stiffness matrix formulation from the Poisson equation in section 2.1.3. Since most of the integrals will be non-zero, only the integrals for the basis functions that correspond to the local element need to be numerically evaluated; therefore, the central idea behind the matrix assembler is to only evaluate the non-zero integrals and assemble them into their corresponding locations in the stiffness matrix (algorithm 1). For the j^{th} element K_j, there are only N_{lb}^2 non-zero integrals. Where, N_{lb} denotes the number of local basis functions that characterize an element. From the reference linear triangle, recall that a unique basis function characterizes the three vertexes of the element. Thus, for the linear triangular element there will be 9 non-zero local integrals to evaluate and assemble into the matrix. All the information needed to evaluate the integrals and assemble the result into the correct matrix location is contained within the *node_coordinates* and *global_indices* information matrices.

Algorithm 1 is a general 2D matrix assembler that can evaluate and assemble the integrals of the basis functions for any combination of partial derivatives and non-derivatives. To construct the stiffness matrix of the Poisson equation the matrix assembler would be called twice: (1) to assemble the partial derivatives with respect to x ($r = p = 1$ and $s = q = 0$); (2) to assemble the partial derivatives with respect to y ($r = p = 0$ and $s = q = 1$). Assembling the resulting values and matrix indices in vector form reduces the computational complexity of having to reshuffle an already formed sparse matrix after each result is computed; therefore, the sparse command is only called once to construct the complete matrix. Assembly of the load vector (algorithm 2) follows the same process as the matrix assembler minus the terms for the trial function.

Algorithm 1: General 2D Matrix Assembler

```
counter = 1
row = zeros(1,$N_{lb}^2 \times number\_mesh\_elements$); % matrix row index
col = zeros(1,$N_{lb}^2 \times number\_mesh\_elements$); % matrix column index
val = zeros(1,$N_{lb}^2 \times number\_mesh\_elements$); % integral result

for n = 1: number_mesh_elements
     vertices = node_coordinates( : , global_indices ( : , n ) );

    for $\alpha$ = 1: $N_{lb}$
        for $\beta$ = 1: $N_{lb}$
            val(counter) = $\int_{K_n} c \frac{\partial^{r+s}\psi_{n\alpha}}{\partial x^r \partial y^s} \frac{\partial^{p+q}\psi_{n\beta}}{\partial x^p \partial y^q}\ dxdy$
            row(counter) = global_indices($\beta, n$);
            col(counter) = global_indices($\alpha, n$);
            counter=counter+1;

         end for
    end for
end for
end function
```

Algorithm 2: General 2D Vector Assembler

```
b = zeros(number_mesh_nodes,1);

for n = 1: number_mesh_elements
     vertices = node_coordinates( : , global_indices ( : , n ) );

        for $\beta$ = 1: $N_{lb}$
            result = $\int_{K_n} f \frac{\partial^{p+q}\psi_{n\beta}}{\partial x^p \partial y^q}\ dxdy$
            b(global_indices($\beta$,n) , 1) = b(global_indices($\beta$,n ) , 1) + result\

         end for
end for
end function
```

4. NEUTRON TRANSPORT THEORY

The primary objective concerning reactor analysis is to ensure the safe, continuous operation of nuclear reactors subjected to a wide range of operating conditions. By invoking certain assumptions, the simplification of the Boltzmann transport equation (initially derived to characterize the transport of microscopic molecules in a medium) permits its application to the study of neutron transport processes. Ultimately, the mathematical analysis regarding the free motion of a collection of neutrons in a medium, provide reactor physicists a means to characterize neutron distributions and reaction rates. Equipped with this information, reactor physicists can manipulate reactor designs, and reactivity configurations that result in operating limits which maximize efficiency and safety under current licensing regulations. The discussion presented in this chapter is not meant to be exhaustive by any means, but rather serve as an introduction to the fundamentals of neutron transport theory and the necessary approximation methods which result in practical mathematical tools for this work.

4.1. NEUTRON BOLTZMANN EQUATION

In the derivation of the neutron transport equation from the Boltzmann equation, it is necessary to make the following assumptions: (1) neutrons are treated as classical neutral particles; (2) neutrons travel in straight lines between collisions; (3) compared to the density of nuclei in a medium, the neutron density is sufficiently small enough to disregard neutron-neutron interaction, resulting in a linearized scattering term; (4)

material properties are isotropic; (5) only the neutron density (collection of particles) are considered [21]. A phase space volume element $\vec{P} = (\vec{r}, \widehat{\Omega}, E, t)$ (Figure 4.1) which permits the acquisition of the expected number of neutrons in an infinitesimal volume consists of seven independent variables, $\vec{r}$ = spatial position, $\widehat{\Omega}$ = angular direction of motion, E = energy, and t = time.

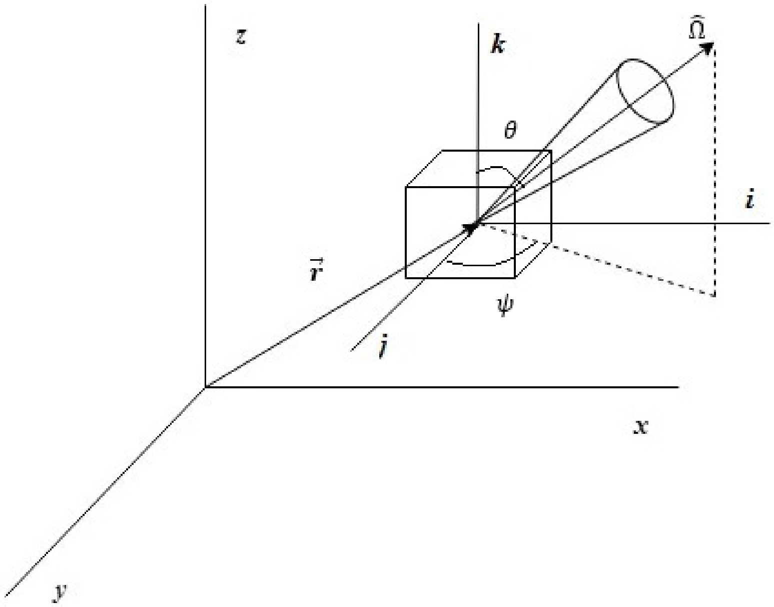


Figure 4.1 Volume and directional element

4.1.1. Angular Neutron Density, Flux, and Current. The expected number of neutrons at a time $t + \Delta t$ in the volume *dr* about *r*, within the energy range *dE* whose direction of motion lie in the differential solid angle $d\Omega$ about Ω is the most general description of the angular neutron density function,

$$N(\vec{r},\widehat{\Omega},E,t)\; d\vec{r}\; d\widehat{\Omega}\; dE\; dt. \tag{4.1}$$

Integration of the angular neutron density over all directions yields the neutron density,

$$n(\vec{r},E,t) = \int_{4\pi} N(\vec{r},\widehat{\Omega},E,t)\; d\Omega. \tag{4.2}$$

The neutron density is the expected number of neutrons at $\vec{r}$, with energy E at time *t*, per unit volume per unit energy. Multiplying the angular density function by the velocity *v* that corresponds to their energy *E* results in the angular neutron flux

$$\Phi(\vec{r},\widehat{\Omega},E,t) = vN(\vec{r},\widehat{\Omega},E,t). \tag{4.3}$$

Integration of the angular neutron flux over all directions yields the total neutron flux,

$$\varphi(\vec{r},E,t) = \int_{4\pi} \Phi(\vec{r},\widehat{\Omega},E,t)\; d\Omega = vn(\vec{r},E,t). \tag{4.4}$$

One can think of the angular neutron flux as the total track length traveled by the neutrons in the phase space volume element per unit time that relates the reaction rate *R*, as neutrons stream through the infinitesimal phase space volume element, to the macroscopic cross section Σ_i (probability of interaction *i* per path length) of the medium.

$$R_i(\vec{r},\widehat{\Omega},E,t) = \Sigma_i(\vec{r},E)\psi(\vec{r},\widehat{\Omega},E,t). \tag{4.5}$$

It is also necessary to account for the scattering reactions in which neutrons scatter from energy E to E′ and direction $\widehat{\Omega}$ to $\widehat{\Omega}'$ through the differential reaction rate.

$$R_s(\vec{r},\widehat{\Omega}\rightarrow\widehat{\Omega}',E\rightarrow E',t) = \Sigma_s(\vec{r},\widehat{\Omega}\rightarrow\widehat{\Omega}',E\rightarrow E')\psi(\vec{r},\widehat{\Omega},E,t) \tag{4.6}$$

where $\Sigma_s(\vec{r}, \widehat{\Omega} \to \widehat{\Omega}', E \to E')$ is the macroscopic double-differential scattering cross section.

Up to this point, only the means to obtain an expected number of neutrons in a volume element is presented; however, it is also necessary to describe the net flow of neutrons streaming into and out of the volume element. The angular neutron current density is the rate that neutrons with energy E and direction $\widehat{\Omega}$ pass through a surface and can be related to the angular flux by

$$j(\vec{r}, E, t) = \widehat{\Omega}\, \Phi(\vec{r}, \widehat{\Omega}, E, t). \tag{4.7}$$

Integrating the angular neutron current density over all directions yields the neutron current density. This is the net number of neutrons of energy E at position $\vec{r}$ and time t crossing a unit area per unit energy and time.

$$J(\vec{r}, E, t) = \int_{4\pi} j(\vec{r}, E, t)\ d\Omega. \tag{4.8}$$

4.1.2. Balance Equation. With the foregoing quantities, it is possible to establish a balance equation which governs the rate of change of the neutron density in an infinitesimal phase space volume element. Let's consider a neutron density whose energy lies in dE about E contained inside the volume element V, about r, at times t and $t + \Delta t$ whose velocity vectors are within $d\Omega$ about Ω. The neutron density balance equation in this phase space volume element $P(\vec{r}, \widehat{\Omega}, E, t)$ is then

$$\frac{\partial}{\partial t}\left[\int_V N(\vec{r}, \widehat{\Omega}, E, t) d^3r\right] dE\ d\widehat{\Omega} = gain\ in\ V - loss\ from\ V. \tag{4.9}$$

The mechanisms that permit the gain of neutrons (1) and loss of neutrons (2) in the phase space volume element are: (1.a) neutron sources inside the volume (fission); (1.b) neutrons streaming into the volume element through a surface; (1.c) neutrons scattering from $E', \widehat{\Omega'}$ into $E, \widehat{\Omega}$; (2.a) neutrons leaking out of the volume element through a surface; (2.b) neutrons that are absorbed by the medium inside the volume element (includes parasitic capture and fission); (2.c) neutrons scattering out of $E, \widehat{\Omega}$.

If the only neutron source inside the volume V are fission neutrons, the source term (1.a) becomes

$$\begin{aligned} &S(\vec{r}, \widehat{\Omega}, E, t) \\ &= \frac{1}{K_{eff}} \frac{\chi(E)}{4\pi} \int_{4\pi} d\widehat{\Omega'} \int_0^\infty dE' \nu'(E')\, \Sigma_f(E') v N(\vec{r}, \widehat{\Omega}, E, t), \end{aligned} \tag{4.10}$$

where $\chi(E)$ is the fraction of fission neutrons born with energy E and $\nu'(E')$ is the number of neutrons emitted from neutron induced fission with energy E'. Combining the neutron streaming gain (1.b) and loss (2.a) terms result in the net leakage over the entire surface:

$$\int_S ds \cdot v\widehat{\Omega}\, N(\vec{r}, \widehat{\Omega}, E, t) dE\, d\widehat{\Omega}. \tag{4.11}$$

Recall the gradient operator,

$$\nabla F = \frac{\partial F}{\partial x} i + \frac{\partial F}{dy} j + \frac{\partial F}{dz} k \tag{4.12}$$

and Gauss’s theorem,

$$\int_S (F \cdot n)\, dS = \int_V (\nabla \cdot F)\, dV. \tag{4.13}$$

Applying Gauss's theorem to leakage term (Eq. 4.11) recasts the surface integral into a volume integral of the divergence inside of the surface.

$$\int_S ds \cdot v\widehat{\Omega}\, N(\vec{r}, \widehat{\Omega}, E, t)\, dE\, d\widehat{\Omega} = \int_V d^3\mathrm{r}\, \nabla \cdot v\widehat{\Omega} N(\vec{r}, \widehat{\Omega}, E, t)\ dE\, d\widehat{\Omega}. \tag{4.14}$$

The gain of neutrons in the volume element resulting from scattering reactions (1.c) from $E', \widehat{\Omega'}$ into $E, \widehat{\Omega}$:

$$\int_V d^3r \int_{4\pi} d\widehat{\Omega'} \int_0^\infty dE' v' \Sigma_s(E' \rightarrow E, \widehat{\Omega'} \rightarrow \widehat{\Omega}) N(\vec{r}, \widehat{\Omega}', E', t)\ dE\ d\widehat{\Omega}. \tag{4.15}$$

The loss of neutrons from the volume element due to out scattering (2.b) and absorption (2.c) reactions:

$$\int_V d^3r\ vN(\vec{r}, \widehat{\Omega}, E, t) \Sigma_t(\vec{r}, E)\ dE\ d\widehat{\Omega}\,. \tag{4.16}$$

4.1.3. Integral-Differential Linear Neutron Boltzmann Equation. Since the volume element is arbitrary, the expression must hold true for any expression inside the integral. Therefore, assembling all the gain and loss terms result in the linearized neutron Boltzmann equation where each term governs a physical process in the system.

$$\frac{1}{v}\frac{\partial \Phi}{\partial t} + \widehat{\Omega} \cdot \nabla \Phi\big(\vec{r}, \widehat{\Omega}, E, t\big) + \Phi\big(\vec{r}, \widehat{\Omega}, E, t\big)\Sigma_t(\vec{r}, E)$$
$$= \int_{4\pi} d\widehat{\Omega'} \int_0^\infty dE' \Phi\big(\vec{r}, \widehat{\Omega}, E, t\big)\Sigma_s\big(\vec{r}, E' \rightarrow E, \widehat{\Omega'} \rightarrow \widehat{\Omega}\big) \quad (4.17)$$
$$+ \frac{\chi(E)}{4\pi} \int_{4\pi} d\widehat{\Omega'} \int_0^\infty dE' \nu(E')\, \Phi\big(\vec{r}, \widehat{\Omega}, E, t\big)\Sigma_f(\vec{r}, E')\,.$$

Readers should be aware that the integral-differential form is only one of the many forms of the linear neutron Boltzmann equation. Where, other forms allow the use of different numerical approximation schemes and mathematical properties.

4.2. SPHERICAL HARMONICS

Expanding the angular flux and scattering terms as a series of basic spherical harmonic functions reduces the form of the neutron Boltzmann equation to a set of differential equations. For the most general cases; a spherical harmonic series represents the angular dependence expansion. However, when considering plane and spherical geometries, the spherical harmonic functions reduce to Legendre polynomials [21]. For the sake of simplicity let’s consider the one-speed (where the cross sections are independent of energy), time-independent, Integro-differential neutron Boltzmann equation for a non-multiplying medium in plane geometry.

$$\widehat{\Omega} \cdot \nabla \Phi\big(x, \widehat{\Omega}\big) + \Phi\big(x, \widehat{\Omega}\big)\Sigma_t(x)$$
$$= \int_{4\pi} \Phi\big(x, \widehat{\Omega}\big)\Sigma_s\big(x, \widehat{\Omega'} \rightarrow \widehat{\Omega}\big)\, d\widehat{\Omega'} + S\big(x, \widehat{\Omega}\big). \quad (4.18)$$

Upon examination of the neutron motion in the plane geometry, it is evident that the angular neutron density is only a function of x and θ, where $\mu = \cos(\theta)$. Simplifying the streaming term based on the physics of neutron motion in plane geometry:

$$\Omega \cdot \nabla \Phi = \frac{d\Phi}{ds} = \frac{\partial \Phi}{\partial x}\frac{\partial x}{\partial s} = \frac{\partial \Phi}{\partial x}\cos\theta = \mu \frac{\partial \Phi}{\partial x}. \tag{4.19}$$

Since the neutron distribution exhibits azimuthal symmetry in plane geometry, integration of the neutron density over all directions **Ω** leads the following definition:

$$\int_{\Omega} N(r,\Omega)d\Omega = 2\pi \int_{-1}^{1} N(x,\mu)d\mu. \tag{4.20}$$

The double-differential scattering cross section is also a function of μ

$$\Sigma_s\left(x,\widehat{\Omega'} \rightarrow \widehat{\Omega}\right) = \Sigma_s(x,\mu_0), \tag{4.21}$$

where $\mu_0 = \cos\left(\widehat{\Omega} \cdot \widehat{\Omega}'\right) = cos\theta_0$ and $\widehat{\Omega}, \widehat{\Omega}'$ are the incident and emitted direction vectors.

Applying the redefined terms into equation yields:

$$\begin{aligned} &\mu \frac{\partial \Phi}{\partial x} + \Phi(x,\mu)\Sigma_t(x) \\ &\qquad = \frac{1}{2\pi}\int_0^{2\pi} d\varphi' \int_{-1}^{1} \Phi(x,\mu')\Sigma_s(x,\mu_0)\ d\mu' + S(x,\mu). \end{aligned} \tag{4.22}$$

Then expand the terms with angular dependence as a series of Legendre polynomials:

$$\Sigma_s(x,\mu_0) = \sum_{l=0}^{\infty} \frac{2l+1}{2}\Sigma_{s,l}(x)P_l(\mu_0); \tag{4.23}$$

$$\Phi(x,\mu) = \sum_{n=0}^{\infty} \frac{2n+1}{2}\Phi_n(x)P_n(\mu); \tag{4.24}$$

$$S(x,\mu) = \sum_{n=0}^{\infty} \frac{2n+1}{2} S_n(x) P_n(\mu). \tag{4.25}$$

From the orthogonality relation of the Legendre polynomials on the interval $-1 \le x \le 1$,

$$\int_{-1}^{1} P_m(\mu) P_n(\mu)\, d\mu = \begin{cases} 0, & n \neq m \\ \dfrac{2}{2n+1}, & n = m \end{cases}. \tag{4.26}$$

Using the addition theorem of the Legendre polynomials allows the Legendre polynomials to be recast in terms of μ and μ'.

$$\begin{aligned} P_l(\mu_0) &= P_l(\mu) P_l(\mu') \\ &\quad + 2 \sum_{m=1}^{l} \frac{(l-m)!}{(l+m)!} P_l^m(\mu)\, P_l^m(\mu') cos\, m(\varphi - \varphi'). \end{aligned} \tag{4.27}$$

Substituting Equation 4.27 into the expansion of the double-differential scattering term yields:

$$\begin{aligned} \Sigma_s(x,\mu_0) = \sum_{l=0}^{\infty} \frac{2l+1}{2} \Sigma_{s,l}(x) \Bigg[& P_l(\mu) P_l(\mu') \\ & + 2 \sum_{m=1}^{l} \frac{(l-m)!}{(l+m)!} P_l^m(\mu)\, P_l^m(\mu') cos\, m(\varphi - \varphi') \Bigg]. \end{aligned} \tag{4.28}$$

Plugging the expanded double-differential scattering cross section into the original one-speed neutron Boltzmann equation while integrating over μ' results in

$$\mu\frac{\partial\Phi}{\partial x}+\Phi(x,\mu)\Sigma_t(x) = \sum_{l=0}^{\infty}\frac{2l+1}{2}\Sigma_{s,l}(x)P_l(\mu)\int_{-1}^{1}\Phi(x,\mu')P_l(\mu')d\mu' + S(x,\mu). \qquad (4.29)$$

Note: $\int_0^{2\pi} cos\, m(\varphi-\varphi')\, d\varphi' = 0$. Next, insert the angular flux and source expansion terms.

$$\mu\frac{\partial\Phi}{\partial x}+\sum_{n=0}^{\infty}\frac{2n+1}{2}\Phi_n(x)P_n(\mu)\Sigma_t(x) = \sum_{l=0}^{\infty}\frac{2l+1}{2}\Sigma_{s,l}(x)P_l(\mu)\phi_l(x) + \sum_{n=0}^{\infty}\frac{2n+1}{2}S_n(x)P_n(\mu). \qquad (4.30)$$

Derivation of the streaming term $\mu\frac{\partial\Phi}{\partial x}$ requires the recursion relation

$$(2n+1)\mu P_n(\mu)=(n+1)P_{n+1}(\mu)+nP_{n-1}(\mu). \qquad (4.31)$$

Therefore,

$$\begin{aligned}\sum_{n=0}^{\infty}\frac{\partial\phi_n(x)}{\partial x}&[(n+1)P_{n+1}(\mu)+nP_{n-1}(\mu)]\\ &+\sum_{n=0}^{\infty}(2n+1)\phi_n(x)P_n(\mu)\Sigma_t(x)\\ &=\sum_{l=0}^{\infty}(2l+1)\,\Sigma_{s,l}(x)P_l(\mu)\phi_l(x)\\ &+\sum_{n=0}^{\infty}(2n+1)\,S_n(x)P_n(\mu).\end{aligned} \tag{4.32}$$

Multiply both sides by $P_n(\mu)$ and integrating μ from -1 to 1 results in the infinite set of P_n equations [21]:

$$\begin{aligned}(n+1)\frac{d\phi_{n+1}(x)}{dx}+n\frac{d\phi_{n-1}(x)}{dx}+\phi_n(x)\big[\Sigma_t(x)-\Sigma_{s,l}(x)\big]\\ =S_n(x),\qquad n=0,1,2,\ldots,N.\end{aligned} \tag{4.33}$$

Since there are only N+1 equations with N+2 unknowns, closure of the set requires setting $d\phi_{N+1}(x)/dx=0$ in the $n=N$ equation.

4.2.1. P_1 Equations and the Diffusion Approximation. Considering only the first two spherical harmonic equations by choosing n = 1 and setting $d\phi_2(x)/dx=0$ in the second equation yields the following system of P_1 equations:

$$\frac{d\phi_1(x)}{dx}+\left(\Sigma_t-\Sigma_{s,0}\right)\phi_0=S_0, \tag{4.34}$$

$$\frac{1}{3}\frac{d\phi_0(x)}{dx}+\left(\Sigma_t-\Sigma_{s,1}\right)\phi_1=S_1. \tag{4.35}$$

Here the zeroth scattering moment $\Sigma_{s,0}$ is equivalent to the total scattering cross section Σ_s and the first scattering moment $\Sigma_{s,1}$ is equivalent to the total scattering cross section multiplied by the average cosine of the scattering angle $\bar{\mu}_0$. Under the assumption of an isotropic source the first order source term becomes zero.

$$\frac{d\phi_1(x)}{dx} + (\Sigma_t - \Sigma_s)\phi_0 = S_0, \tag{4.36}$$

$$\frac{1}{3}\frac{d\phi_0(x)}{dx} + (\Sigma_t - \bar{\mu}_0\Sigma_s)\phi_1 = 0. \tag{4.37}$$

Recall that the first two Legendre polynomials [21] are $P_0 = 1$ and $P_1 = \mu$; therefore, by the orthogonality of the Legendre polynomials

$$\phi_0 = \frac{2n+1}{2}\int_{-1}^{1} \Phi(x,\mu)\, d\mu = \varphi(x), \tag{4.38}$$

$$\phi_1 = \frac{2n+1}{2}\int_{-1}^{1} \Phi(x,\mu)\mu\, d\mu = J(x) \tag{4.39}$$

where $\varphi(x)$ is the scalar flux and $J(x)$ is the neutron current density. With the preceding definitions for ϕ_0 and ϕ_1 the P_1 equations become

$$\frac{dJ(x)}{dx} + (\Sigma_t - \Sigma_s)\varphi(x) = S_0, \tag{4.40}$$

$$\frac{1}{3}\frac{d\varphi(x)}{dx} + (\Sigma_t - \bar{\mu}_0\Sigma_s)J(x) = 0 \tag{4.41}$$

Re-arranging the second equation in terms of the current $J(x)$ yields

$$J(x) = -\frac{1}{3(\Sigma_t - \bar{\mu}_0\Sigma_s)}\frac{d\varphi(x)}{dx}. \tag{4.42}$$

Since the total macroscopic cross section Σ_t is equivalent to the sum of the macroscopic absorption and total scattering cross sections the current term in an equivalent form

$$J(x) = -\frac{1}{3(\Sigma_a + \Sigma_s(1-\bar{\mu}_0))}\frac{d\varphi(x)}{dx} \tag{4.43}$$

where $\Sigma_s(1-\bar{\mu}_0)$ is the macroscopic transport cross section Σ_{tr}. If the medium is more conducive to scattering than absorption ($\Sigma_a \ll \Sigma_s$) the macroscopic absorption cross section can be neglected; therefore, the neutron current simplifies to

$$J(x) = -\frac{1}{3\Sigma_{tr}}\frac{d\varphi(x)}{dx}. \tag{4.44}$$

Substituting the definition for the neutron current J(x) into the first P_1 equation yields

$$-\frac{d}{dx}\left[\frac{1}{3\Sigma_{tr}}\frac{d\varphi(x)}{dx}\right] + \Sigma_a\varphi(x) = S_0. \tag{4.45}$$

The preceding steady-state P_1 equation is nearly identical to the steady-state neutron diffusion equation, which is derived from the neutron continuity equation and Fick's law [22].

$$-\frac{d}{dx}\left[\frac{1}{3\Sigma_s}\frac{d\varphi(x)}{dx}\right] + \Sigma_a\varphi(x) = S. \tag{4.46}$$

Where $D \equiv 1/3\Sigma_s$. Differences between the two formulations are attributed to the treatment of the angular scattering distribution in the neutron current term. In the P_1 equations the angular scattering distribution is accounted for through the macroscopic transport cross section, whereas the diffusion approximation assumes isotropic scattering.

If the angular scattering distribution is forward peaked (meaning that after a scattering event the neutron continues in the general direction it was initially traveling) the average cosine of the scattering angle will be positive. Consequently, the macroscopic transport cross section will be reduced from the purely isotropic case ($\bar{\mu}_0 = 0$) which results in a larger proportionality constant. Thus, the forward scattering of neutrons is somewhat preserved by increasing the proportionality constant in the current to flux gradient relationship (increased net leakage). If the opposite is true, backwards preferential scattering will result in $\bar{\mu}_0 < 0$. Hence, the proportionality constant will be reduced. In the case where the scattering is completely isotropic the average cosine of the scattering angle $\bar{\mu}_0$ will be zero which results in the simplification of the current term in the P_1 equations to Fick's law.

$$J(x) = -\frac{1}{3\Sigma_s}\frac{d\varphi(x)}{dx}. \tag{4.47}$$

Ultimately, accounting for the angular scattering distribution from the P_1 proportionality constant in the definition of the diffusion coefficient permits the extension of the diffusion approximation to systems that exhibit moderate anisotropic scattering; however, one must be mindful of the overall assumptions made in the derivation of the P_1 and diffusion approximations and where these approximations fail. Recall the following assumptions under which Fick's law was derived: (1) the medium is infinite; (2) the medium is uniform (uniform cross sections); (3) no local source or absorbing medium; (4) scattering is isotropic in the laboratory frame; (5) the neutron flux is a slowly varying function of space; (6) the neutron flux is not a function of time [22].

Although these assumptions are quite restrictive and perhaps non-physical; these restrictions under certain cases can be relaxed provided that the relaxation does not result in violation of other assumptions [22]. Despite the assumption of an infinite medium it is possible for Fick’s law to be valid in a finite medium such that the region of interest is sufficiently insulated from the boundary. This permitted since neutron densities further than a few mean free paths from the point of calculation will not affect the current density. The assumption of a uniform medium is not a strict requirement so long as the absorption << scattering, or if the ration Σ_s/Σ_t remains constant over space; however, concentrated regions of high absorption may result in large local flux perturbations which violates the slowly varying spatial flux assumption.

As previously discussed, it is also possible to account for mediums that exhibit moderate anisotropic scattering by using the macroscopic transport correction cross section from the proportionality constant of the P_1 equations. Time dependence is also permitted in cases where the fractional change is sufficiently small enough during the time required for a neutron to travel 3 mean free paths $\left|\frac{1}{\varphi}\frac{d\varphi}{dt}\right| \ll \frac{10^5}{3\lambda_s} sec^{-1}$ [22]. The issue surrounding local sources and absorbing mediums is circumvented by partitioning a heterogeneous region into a set of smaller homogenous regions (spatial homogenization). Essentially, the energy and spatial dependent macroscopic cross sections are averaged over the energy dependent, spatial neutron flux such that the interaction rates are preserved. Thus, removing large deviations in the spatial dependence of the macroscopic cross sections and effectively maintaining the relation $\Sigma_a \ll \Sigma_s$ over space.

With the previous assumptions in mind, the discussion turns to the application of the diffusion/P_1 equations to reactor analysis. Typically, diffusion/P_1 equations with transport correction for hydrogen will provide relatively good global flux approximations to the neutron transport phenomena for large, symmetric, low heterogeneous light water power reactors. For reactors of the light water type, the predominant interaction mode is elastic scattering. Hence, the assumption of $\Sigma_a \ll \Sigma_s$ made in the derivation of Fick's law holds true. Since the fuel pins in light water reactors are distributed in fuel assemblies over the entire domain, spatial homogenization of the assemblies will yield a mostly uniform spatial dependence of the macroscopic cross sections; however, slight heterogeneity may exist from fuel assembly burn-up and varying fuel enrichments.

4.2.2. Simplified Spherical Harmonics Equations (SPn). The simplified spherical harmonics equations, initially discovered by Gelbard, are an ad hoc extension of the higher order planar spherical harmonics equations to the multi-dimensional case [23]. The central idea was to eliminate the odd order moments (in the same manner as the derivation of the diffusion equation from the P_1 equations) from the even order equations, followed by the replacement of the one-dimensional operator by the Laplacian. Since the method lacked mathematical support, the simplified spherical harmonics equations were neglected. In more recent years, several studies by Larsen have been published that indicate the method is in fact an asymptotic correction to the diffusion equation [24], [25]. Brantley and Larsen also derived the simplified SP_3 equations by variational analysis and concluded that the SP_3 equations improved the criticality eigenvalues in MOX assemblies [26]. However, as $n \to \infty$, the simplified spherical harmonics does not approach the transport solution of the spherical harmonics equations. Furthermore, the largest increase in accuracy is attained by the SP_3 equations while the solutions deteriorate after the SP_7 equations.

The advantage of the simplified spherical harmonics equations (mainly the SP_3 equations) is the preservation of transport effects and its rather inexpensive approximation when compared to the traditional spherical harmonics equations and other neutron transport approximation schemes. Since the equations are in a form that is analogous to the multi-group diffusion equation, the method allows the use of existing spatial discretization schemes used for the neutron diffusion equation. As a result, the simplified spherical harmonics equations have been implemented in the existing codes DYN3D [27], and PARCS [5]. Furthermore, the use of the FEM method has also provided successful approximations to the simplified spherical harmonics equations in consideration of a small fast reactor in general geometries [28].

Recall the infinite set of 1-D planar spherical harmonics equations from section 4.2, Equation 4.33. Setting $n = 3$ and the assumption of an isotropic source results in the following coupled system of partial differential equations:

$$\frac{d\varphi_1}{dx} + \Sigma_a \varphi_0 = \frac{1}{k} \nu \Sigma_f \varphi_{0,} \tag{4.48}$$

$$\frac{d\varphi_0}{dx} + 2\frac{d\varphi_2}{dx} + 3(\Sigma_t - \Sigma_{s,1})\varphi_1 = 0, \tag{4.49}$$

$$2\frac{d\varphi_1}{dx} + 3\frac{d\varphi_3}{dx} + 5(\Sigma_t - \Sigma_{s,2})\varphi_2 = 0, \tag{4.50}$$

$$3\frac{d\varphi_2}{dx} + 7(\Sigma_t - \Sigma_{s,3})\varphi_3 = 0. \tag{4.51}$$

Re-arranging the even-order equations (Equations 4.49 and 4.51) in terms of the odd-order flux moments and introduce the pseudo zeroth moment flux $\Phi_0 = \varphi_0 + 2\varphi_2$ yields:

$$\varphi_1 = -\frac{1}{3\left(\Sigma_t - \Sigma_{s,1}\right)} \frac{d}{dx}(\varphi_0 + 2\varphi_2) = -D_0 \frac{d\Phi_0}{dx}, \tag{4.52}$$

$$\varphi_3 = -\frac{3}{7\left(\Sigma_t - \Sigma_{s,3}\right)} \frac{d\varphi_2}{dx} = -D_3 \frac{d\varphi_2}{dx}. \tag{4.53}$$

Then, eliminate the odd-order moments from the even-order equations by substitution of Equations 4.52 and 4.53 into Equations 4.48 and 4.50. Also, the first order flux moment derivative is eliminated in Equation 4.50 by re-arranging and substitution of Equation 4.48.

$$-\frac{d}{dx}\left(D_0\frac{d\Phi_0}{dx}\right)+\Sigma_a(\Phi_0-2\varphi_2)=\frac{1}{k}\nu\Sigma_f\ (\Phi_0-2\varphi_2). \tag{4.54}$$

$$\begin{gathered}-2\Sigma_a\Phi_0-3\frac{d}{dx}(D_3\frac{d\varphi_2}{dx})+(4\Sigma_a+5(\Sigma_t-\Sigma_{s,2}))\varphi_2=\\ \frac{1}{k}\nu\Sigma_f(-2\Phi_0+4\varphi_2).\end{gathered} \tag{4.55}$$

Replacing the 1-D operator by the Laplacian yields the simplified spherical harmonics equations in Equations 4.54 and 4.55:

$$-\nabla(D_0\nabla\Phi_0)+\Sigma_a(\Phi_0-2\varphi_2)=\frac{1}{k}\nu\Sigma_f\ (\Phi_0-2\varphi_2), \tag{4.56}$$

$$\begin{gathered}-3\nabla\ (D_3\nabla\varphi_2)+(4\Sigma_a+5(\Sigma_t-\Sigma_{s,2}))\varphi_2-2\Sigma_a\Phi_0=\\ \frac{1}{k}\nu\Sigma_f(-2\Phi_0+4\varphi_2).\end{gathered} \tag{4.57}$$

Where, the pseudo zeroth order flux moment $\Phi_0=(\varphi_0+2\varphi_2)$, the scalar flux φ_0, the first zeroth order diffusion coefficient $D_0=\frac{1}{3(\Sigma_t-\Sigma_{s,1})}$, and the third-order diffusion coefficient $D_3=\frac{3}{7(\Sigma_t-\Sigma_{s,3})}$.

The preceding SP_3 equations are rewritten in multigroup form for G energy groups with Marshak boundary conditions according to [28]:

$$\begin{bmatrix} -D_{0g}\nabla^2 + \Sigma_{rg} & -2\Sigma_{rg} \\ -2\Sigma_{rg} & -D_{3g}\nabla^2 + 4\Sigma_{rg} + 5\Sigma_{Tg} \end{bmatrix} \begin{bmatrix} \Phi_{0g}(\vec{r}) \\ \varphi_{2g}(\vec{r}) \end{bmatrix} = \begin{bmatrix} q_{0g}(\vec{r}) \\ -2q_{0g}(\vec{r}) \end{bmatrix} \; in\ \Omega. \tag{4.58}$$

Where, the Marshak boundary conditions are

$$\begin{bmatrix} J_{0g}(r) \\ J_{2g}(r) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & -\frac{3}{8} \\ -\frac{3}{8} & \frac{21}{8} \end{bmatrix} \begin{bmatrix} \Phi_{0g}(\vec{r}) \\ \varphi_{2g}(\vec{r}) \end{bmatrix}, \mathrm{r} \in \partial\Omega. \tag{4.59}$$

Here, the isotropic source is

$$q_{0g} = \frac{1}{k}\sum_{g'=1}^{G} \nu\Sigma_{fg'}\varphi_{g'} + \sum_{\substack{g'=1 \\ g'\neq 1}}^{G} \Sigma_{sg'\to g}\varphi_{g}. \tag{4.60}$$

Therefore, when the number of energy groups G = 2; a coupled system of four partial differential equations is formed.

4.3. MULTI-GROUP DIFFUSION EQUATION

Let’s consider the strong formulation of the coupled multi-group critical equation with albedo boundary conditions for G energy groups:

$$\begin{aligned}&-\nabla\cdot D_g(\vec{r})\nabla\varphi_g(\vec{r})+\Sigma_{rg}\varphi_g(\vec{r})\\&\qquad=\sum_{\substack{g'=1\\g'\neq g}}^{G}\Sigma_{s,g'\to g}(\vec{r})\varphi_{g'}(\vec{r})\\&\qquad+\frac{\chi_g}{K_{eff}}\sum_{g'=1}^{G}v_{g'}\Sigma_{f,g'}(\vec{r})\varphi_g(\vec{r})\ \ in\ \ \Omega,\end{aligned}\tag{4.61}$$

$$D_g(\vec{r})\nabla\varphi_g(\vec{r})\cdot\vec{n}(\vec{r})+\frac{1}{2}\frac{1-\beta_g(\vec{r})}{1+\beta_g(\vec{r})}\varphi_g(\vec{r})=0\ \ on\ \ \partial\Omega.\tag{4.62}$$

Where, $\Omega\subset\mathbb{R}^n$ is a bounded domain, D_g is the diffusion coefficient, Σ_i is the macroscopic cross section of the i^{th} reaction type, β_g is the albedo, χ_g is the fraction of the neutrons produced from fission appearing in the g^{th} energy group, $v_{g'}$ is the number of neutrons emitted per fission, $\Sigma_{a,g}+\sum_{\substack{g'=1\\g'\neq g}}^{G}\Sigma_{s,g\to g'}$ is the macroscopic group removal cross section and $\varphi_g=[\varphi_1,\ldots,\varphi_G]^T$ are the unknown multi-group neutron fluxes. Multiply eq. 4.49 by a test function $v=[v_1,\ldots,v_G]^T$ and integrate over the domain Ω.

$$\begin{aligned}&-\int_{\Omega}\left(\nabla\cdot D_g\nabla\varphi_g\right)v\,d\Omega+\int_{\Omega}\Sigma_{rg}\varphi_g v\,d\Omega\\&\qquad=\sum_{g'=1}^{G}\int_{\Omega}\Sigma_{s,g'\to g}\varphi_{g'}v\,d\Omega\\&\qquad+\frac{\chi_g}{K_{eff}}\sum_{g'=1}^{G}\int_{\Omega}v_{g'}\Sigma_{f,g'}\varphi_g v\,d\Omega\ \ in\ \ \Omega.\end{aligned}\tag{4.63}$$

Applying Greens formula (multi-dimensional integration by parts) to the differential leakage term in eq. 4.51,

$$
\begin{aligned}
-\int_{\Omega} \left(\nabla \cdot D_g \nabla \varphi_g\right) v \, d\Omega &= D_g \left[\int_{\Omega} \nabla \varphi_g \cdot \nabla v \, d\Omega - \int_{\Omega} \nabla \cdot \left(\nabla \varphi_g v\right) d\Omega\right] \\
&= D_g \left[\int_{\Omega} \nabla \varphi_g \cdot \nabla v \, d\Omega - \int_{\partial\Omega} \nabla \varphi_g v \cdot \overrightarrow{ds}\right] \qquad (4.64) \\
&= D_g \left[\int_{\Omega} \nabla \varphi_g \cdot \nabla v \, d\Omega - \int_{\partial\Omega} \left(\nabla \varphi_g \cdot \vec{n}\right) v \, ds\right].
\end{aligned}
$$

Substituting the albedo boundary condition into the boundary integral term in eq. 4.52,

$$
\int_{\partial\Omega} \left(\nabla \varphi_g \cdot \vec{n}\right) v \, ds = \int_{\partial\Omega} (0) \, v \, ds - \int_{\Omega} \frac{1}{2D_g} \frac{1-\beta_g}{1+\beta_g} \varphi_g v \, ds. \qquad (4.65)
$$

Hence the general weak formulation of the multi-group critical problem: find

$\varphi_g = [\varphi_1, \dots, \varphi_G]^T \in [H^1(\Omega)]^{\,G}$ such that

$$
\begin{aligned}
&D_g\left(\nabla \varphi_g, \nabla v\right) + \frac{1}{2}\frac{1-\beta_g}{1+\beta_g}\langle \varphi_g, v\rangle + \Sigma_{r,g}\left(\varphi_g, v\right) \\
&\qquad = \sum_{\substack{g'=1 \\ g' \neq 1}}^{G} \Sigma_{s,g' \to g}\left(\varphi_{g'}, v\right) + \sum_{g'=1}^{G} \frac{\chi_g \nu_{g'} \Sigma_{f,g'}}{K_{eff}}\left(\varphi_{g'}, v\right), \qquad (4.66)
\end{aligned}
$$

$\forall v \in [H^1(\Omega)]^G$. Here, $(\cdot,\ \cdot)$ = inner product and $\langle \cdot,\ \cdot \rangle$ = surface/line integral (3D/2D).

4.3.1. Finite Element Formulation. Now, let’s consider the formulation for two energy groups ($G = 2$). The following process is the same for an arbitrary G energy groups, albeit with more finite element spaces. Assume there is a finite dimensional subspace $U_h \times V_h \subset [H^1(\Omega)^2]$. Then the Galerkin formulation: find the approximate flux solution $\varphi_{g,h} = [\varphi_{1h}, \varphi_{2h}]^T \in\ U_h \times V_h$ such that

$$D_g\big(\nabla\varphi_{g,h}, \nabla v_h\big) + \frac{1}{2}\frac{1-\beta_g}{1+\beta_g}\langle\varphi_{g,h}, v_h\rangle + \Sigma_{r,g}\big(\varphi_{g,h}, v_h\big)$$

$$= \sum_{\substack{g'=1\\ g'\neq 1}}^{2} \Sigma_{s,g'\to g}\big(\varphi_{g',h}, v_h\big) \tag{4.67}$$

$$+ \sum_{g'=1}^{2} \frac{\chi_g v_{g'} \Sigma_{f,g'}}{K_{eff}}\big(\varphi_{g',h}, v_h\big)$$

$\forall v_h \in\ U_h \times V_h$. Assume $\varphi_{1h} \in\ U_h = span\{\varphi_j\}_{j=1}^{Nb}$ and $\varphi_{2h} \in\ V_h = span\{\psi_j\}_{j=1}^{Nb}$. Then, $\varphi_{1h} = \sum_{j=1}^{Nb} u_j^{(1)}\varphi_j$ and $\varphi_{2h} = \sum_{j=1}^{Nb} u_j^{(2)}\psi_j$. For the first energy group when $g = 1$, set $v_h = (\varphi_i, 0)^T$. Hence

$$\sum_{j=1}^{Nb} u_j^{(1)}\left[D_1\big(\nabla\varphi_j, \nabla\varphi_i\big) + \frac{1}{2}\frac{1-\beta_1}{1+\beta_1}\langle\varphi_j, \varphi_i\rangle + \Sigma_{r,1}\big(\varphi_j, \varphi_i\big)\right]$$

$$= \sum_{j=1}^{Nb} u_j^{(2)}\, \Sigma_{s,2\to 1}\big(\psi_j, \varphi_i\big) + \tag{4.68}$$

$$\frac{1}{K_{eff}}\left[\sum_{j=1}^{Nb} u_j^{(1)}\left(\chi_1 v_1 \Sigma_{f,1}\big(\varphi_j, \varphi_i\big)\right) + \ \sum_{j=1}^{Nb} u_j^{(2)}\left(\chi_1 v_2 \Sigma_{f,2}\big(\psi_j, \varphi_i\big)\right)\right].$$

For the second energy group when $g = 2$, set $v_h = (0, \psi_i)^T$. Thus,

$$\sum_{j=1}^{Nb} u_j^{(2)} \left[D_2\left(\nabla\psi_j, \nabla\psi_i\right) + \frac{1}{2}\frac{1-\beta_2}{1+\beta_2}\langle\psi_j, \psi_i\rangle + \Sigma_{r,2}\left(\psi_j, \psi_i\right) \right]$$

$$= \sum_{j=1}^{Nb} u_j^{(1)}\, \Sigma_{s,1\to 2}\left(\varphi_j, \psi_i\right) + \qquad (4.69)$$

$$\frac{1}{K_{eff}} \left[\sum_{j=1}^{Nb} u_j^{(1)} \left(\chi_2 \nu_1 \Sigma_{f,1}\left(\varphi_j, \psi_i\right) \right) + \sum_{j=1}^{Nb} u_j^{(2)} \left(\chi_2 \nu_1 \Sigma_{f,2}\left(\psi_j, \psi_i\right) \right) \right].$$

The loss matrix components:

$$L_1 = D_1\left(\nabla\varphi_j, \nabla\varphi_i\right) + \frac{1}{2}\frac{1-\beta_1}{1+\beta_1}\langle\varphi_j, \varphi_i\rangle + \Sigma_{r,1}\left(\varphi_j, \varphi_i\right), \qquad (4.70)$$

$$L_2 = D_2\left(\nabla\psi_j, \nabla\psi_i\right) + \frac{1}{2}\frac{1-\beta_2}{1+\beta_2}\langle\psi_j, \psi_i\rangle + \Sigma_{r,2}\left(\psi_j, \psi_i\right). \qquad (4.71)$$

Thus, the loss matrix:

$$L = \begin{bmatrix} L1 & 0 \\ 0 & L2 \end{bmatrix}. \qquad (4.72)$$

The scattering source matrix:

$$S = \begin{bmatrix} 0 & \Sigma_{s,2\to 1}\left(\psi_j, \varphi_i\right) \\ \Sigma_{s,1\to 2}\left(\varphi_j, \psi_i\right) & 0 \end{bmatrix}. \qquad (4.73)$$

The fission source matrix:

$$F = \begin{bmatrix} \chi_1 \nu_1 \Sigma_{f,1}\left(\varphi_j, \varphi_i\right) & \chi_1 \nu_2 \Sigma_{f,2}\left(\psi_j, \varphi_i\right) \\ \chi_2 \nu_1 \Sigma_{f,1}\left(\varphi_j, \psi_i\right) & \chi_2 \nu_1 \Sigma_{f,1}\left(\psi_j, \psi_i\right) \end{bmatrix}. \qquad (4.74)$$

The finite element flux solution:

$$\vec{X} = \begin{bmatrix} u_j^{(1)} \\ u_j^{(2)} \end{bmatrix}. \tag{4.75}$$

Thus, the two-group critical problem in matrix notation

$$L\vec{X} = \vec{X}\left(\frac{1}{K_{eff}}F + S\right). \tag{4.76}$$

From the preceding matrix formulation, it is evident that coefficient matrices exist on both sides of the equation; therefore, the preceding problem is an eigenvalue problem.

The criticality eigenvalue problem of equation always has the trivial solution where $\overrightarrow{\vec{X}} = 0$; however, the objective is to find the largest value of K_{eff} such that solution $\vec{X}$ is non-zero. It just so happens to be that the only physical solution $\vec{X}$ to the criticality problem corresponds to the largest eigenvalue K_{eff}. In terms of the criticality problem, K_{eff} defines the balance between the neutron fission source and loss terms. If the production of neutrons through the fission source term is greater than the loss terms, the system is supercritical $(K_{eff} > 1)$. Thus, the neutron population will evolve until there are no more fissile atoms. If the fission source term is in balance with the loss terms, the system is at steady-state $(K_{eff} = 1)$. When the fission source term is less than loss terms, the system is sub-critical $(K_{eff} < 1)$. Thus, increases to the fission term are required for the system to achieve steady-state.

4.3.2. Power Iteration. If the largest positive eigenvalue, $K_{eff} > 0$, that is real, unique, and has a non-negative fission distribution, an iterative power iteration scheme can be employed; however, the algorithm may be slow to converge when the dominance ration (K_2/K_{eff}) is close to one. The idea is to provide an initial guess for the eigenvalue K_{eff}, and the eigenvector $\vec{X}$, to solve a fixed source diffusion problem for an updated eigenvector $\vec{X}$. Next, the fission source is updated, and a new eigenvalue is calculated. This process is repeated until specified convergence criteria are met. For the simulations presented in this thesis, the following convergence criteria were used: (1) relative eigenvalue error $\leq 10^{-8}$; and (2) maximum relative flux error $\leq 10^{-4}$.

Algorithm 3: Standard PI

```
Input: $\vec{\varphi}^{n}, k_{eff}^{n}, tol_{k_{eff}},\ tol_{\varphi}$
Result: $\vec{\varphi}, k_{eff}$
while $\varepsilon_{\varphi} > tol_{\varphi} \, || \, \varepsilon_{k_{eff}} > tol_{k_{eff}}$
    1. Solve fixed source problem:
        $\vec{\varphi}^{n+1} = \frac{\left(\frac{1}{{K_{eff}}^{n}} F + S\right)\vec{\varphi}^{n}}{L}.$    (4.77)
    2. Update eigenvalue:
        $k_{eff}^{n+1} = k_{eff}^{n} \frac{F\vec{\varphi}^{n+1}}{F\vec{\varphi}^{n}}.$    (4.78)
    3. Compute max relative flux error: (element wise division)
        $\varepsilon_{\varphi} = \max\left(\frac{|\vec{\varphi}^{n+1} - \vec{\varphi}^{n}|}{\vec{\varphi}^{n+1}}\right).$    (4.79)
    4. Compute relative eigenvalue error:
        $\varepsilon_{k_{eff}} = \frac{\left|k_{eff}^{n+1} - k_{eff}^{n}\right|}{k_{eff}^{n+1}}.$    (4.80)
end while
5. Normalize flux solution: (element wise division)
    $\vec{\varphi} = \left[\frac{1}{V} sum(F\vec{\varphi})\right]^{-1} \vec{\varphi}.$    (4.81)
end function
```

5. SERPENT 2: A CONTINUOS-ENERGY MONTE CARLO CODE

Serpent: A Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code [29] originated from the Ph.D. research conducted by Jaakko Leppanen at the VTT Research Centre of Finland in 2004 [30]. The central idea behind Jaakko's research was to leverage the inherent advantages of stochastic Monte Carlo neutron transport solvers in the development of a novel lattice physics code "Probabilistic Scattering Game", or PSG. Ultimately, PSG would undergo a name change with its public release in 2009, and subsequent development version Serpent 2. Although the expansion of Serpents capabilities in the development version Serpent 2 now include general neutron/photon transport, and multi-physics simulations, the remainder of this chapter presents the spatial homogenization methodologies implemented for the use of multi-group constant generation [31].

5.1. SPATIAL HOMOGENIZATION METHODOLOGY

5.1.1. Reaction Rates. Spatial homogenization is a process that is used to produce multi-group macroscopic cross sections of heterogeneous regions to permit reconstruction of the global homogenous flux solution in full core simulators. Volume averaging the continuous energy macroscopic cross sections over the energy dependent spatial flux results in the homogenous multi-group macroscopic cross section.

$$\Sigma_{i,g} = \frac{\int_{g}^{g-1} \int_{V} \ \Sigma_i(r,E)\varphi(r,E)\ dVdE}{\int_{g}^{g-1} \int_{V} \ \varphi(r,E)\ dVdE}. \tag{5.1}$$

Where, $\Sigma_{i,g}$ is the macroscopic cross section of the i^{th} reaction type of the g^{th} energy group, and φ is the scalar neutron flux. Essentially, this process preserves the reaction rates observed in the heterogeneous transport problem when collapsing the spatial dependence of the macroscopic cross sections.

Instead of utilizing equation 5.1, Serpent takes a different approach by assembling reaction rate estimates into an intermediate energy structure (h that is either pre-defined or user supplied) before generating the few-group cross sections. The Monte Carlo tallies, and analog estimates assembled within each intermediate energy group structure are collapsed into the few-group structure (g) after each criticality source batch [31]. The collection of the group constant estimates at the end of the criticality source simulation form the relative statistical and mean errors. The first steps in the calculation chain are to obtain the scalar flux Φ_h, and macroscopic cross sections $\Sigma_{i,h}$ belonging to the intermediate energy group structure.

$$\Phi_h = \int\limits_{h}^{h-1} \int\limits_{V} \varphi(r, E)\ dVdE. \tag{5.2}$$

$$\Sigma_{i,h} = \frac{\int_h^{h-1} \int_V \ \Sigma_i(r, E)\varphi(r, E)\ dVdE}{\int_h^{h-1} \int_V \ \varphi(r, E)\ dVdE}. \tag{5.3}$$

Then collapsing of the intermediate energy group estimates $\Sigma_{i,h}$ into the final few-group structure $\Sigma_{i,g}$ via flux weighting (equation 5.4).

$$\Sigma_{i,g} = \frac{\sum_{h\in g} \Sigma_{i,h}\Phi_h}{\sum_{h\in g} \Phi_h}. \tag{5.4}$$

Serpent follows this routine to produce the fission, absorption, and total scattering macroscopic cross sections for the selected regions to be homogenized.

5.1.2. Scattering Matrices. Deterministic approximations of the Boltzmann transport equation relies on the discretization of the energy, spatial, and angular dependence of the neutron flux, where the fission, and group to group scattering source terms are responsible for the coupling of the system. A scattering matrix containing the macroscopic group to group scattering cross sections characterizes the transfer of neutrons with energy E to E′. Ideally, one would obtain the macroscopic group transfer cross sections by averaging the differential scattering cross section over incident, and emission energy over the energy dependent spatial flux:

$$\Sigma_{s,g\to g'} = \frac{\int_{E'_g}^{E'_{g-1}} \int_{E_g}^{E_{g-1}} \int_V \ \Sigma_s\left(r, E \to E\,'\right) \varphi(r,E)\ dVdEdE'}{\int_{E_g}^{E_{g-1}} \int_V \ \varphi(r,E)\ dVdE}. \tag{5.5}$$

Since Serpent reads cross section data in ACE format, it is the total scattering cross section, and energy-dependent angular distribution probabilities that are available; therefore, Serpent cannot directly evaluate equation 5.5 [31]. Nevertheless, analog estimates of all sampled scattering reactions from group h to h′ during the transport simulation form the group transfer probabilities:

$$P_{h\to h'} = \frac{\int_{h}^{E'_{h-1}} \int_{E_h}^{E_{h-1}} \int_V \ \Sigma_s\left(r, E \to E\,'\right) \varphi(r,E)\ dVdEdE'}{\int_{E_h}^{E_{h-1}} \int_V \ \Sigma_s(r,E)\varphi(r,E)\ dVdE}. \tag{5.6}$$

Multiplying the total macroscopic scattering cross section (equation 5.3) by the group transfer probabilities generates the P_0 macroscopic group transfer scattering cross sections:

$$\Sigma_{s0,h\to h'} = P_{h\to h'}\Sigma_{s,h}. \tag{5.7}$$

After each criticality source batch Equation 5.8 collapses the intermediate energy group scattering matrix into the few-group structure by flux weighting

$$\Sigma_{s0,g\to g'} = \frac{\sum_{h\in g}\sum_{h'\in g'}\Sigma_{s0,h\to h'}\varphi_h}{\sum_{h\in g}\varphi_h}. \tag{5.8}$$

The final form of the P_0 scattering matrix:

$$\sum_{g=1}^{G}\sum_{g'=1}^{G}\Sigma_{s0,g\to g'} = \left[\Sigma_{s0,g\to g'}\right]_{g,g'=1}^{g,g'=G}. \tag{5.9}$$

Where, results from the criticality source iterations form the associated statistical errors. Weighting the multi-group P_0 group transfer scattering cross sections by the scattering cosine μ forms the P_1 matrix where the scalar product of the incident and emitted neutron vectors provides the scattering angle μ. Note: Although Serpent can obtain the P_n scattering matrices up to the 7^{th} order, Jaakko states “the higher order terms have not been tested” [31].

5.1.3. Diffusion Coefficients. Preparing diffusion coefficients from Monte Carlo transport solvers require the use of various approximations as the diffusion coefficient has no continuous-energy equivalent in transport theory. The approach taken by Serpent relies on the derivation of the diffusion coefficient from the multi-group P_1 equations:

$$\frac{dJ_h}{dx} + \Sigma_{t,h}\Phi_h = \sum_{h'}\Sigma_{s0,h'\to h}\Phi_{h'} + \frac{\chi_h}{k_{eff}}\sum_{h'}\nu\Sigma_{f,h'}\Phi_{h'} \tag{5.10}$$

$$\frac{1}{3}\frac{d\Phi_h}{dx} + \Sigma_{t,h}J_h = \sum_{h'}\Sigma_{s1,h'\to h}J_{h'}. \tag{5.11}$$

Rearranging the first order angular moment in equation 5.11 for the current J_h

$$J_h = -\frac{1}{3}\left(\Sigma_{t,h} - \frac{\sum_{h'} \Sigma_{s1,h'\to h} J_{h'}}{J_h}\right)^{-1} \frac{d\Phi_h}{dx}. \tag{5.12}$$

Notice that equation 5.12 is equivalent to Fick’s law [22]:

$$J_h(r) = -D_h \nabla \Phi_h(r), \tag{5.13}$$

where the diffusion coefficient is defined as

$$D_h = \frac{1}{3}\left(\Sigma_{t,h} - \frac{\sum_{h'} \Sigma_{s1,h'\to h} J_{h'}}{J_h}\right)^{-1}, \tag{5.14}$$

and the transport corrected total cross section,

$$\Sigma_{tr,h} = \Sigma_{t,h} - \frac{\sum_{h'} \Sigma_{s1,h'\to h} J_{h'}}{J_h}. \tag{5.15}$$

The application of the P_1 transport correction can be achieved by either the in-scatter (equation 5.15), or the out-scatter approximation (equation 5.16). Unfortunately, the in-scatter method has limited applicability in Monte Carlo transport solvers due to the current weighting of the P_1 scattering matrix [31]. The work around to this limitation is to replace the neutron current in equation 5.15 with the scalar flux. This is possible under the out-scatter assumption which states “the in-scatter from all groups h' into group h equals the out-scatter from group h to all other groups h'” [31].

$$\sum_{h'} \Sigma_{s1,h'\to h} J_{h'} \approx \sum_{h'} \Sigma_{s1,h\to h'} J_h. \tag{5.16}$$

Substituting equation 5.16 into equation 5.15 yields the following out-scatter transport cross section, and diffusion coefficient:

$$\Sigma_{tr,h} \approx \Sigma_{t,h} - \frac{\sum_{h'} \Sigma_{s1,h\to h'} J_h}{J_h} = \Sigma_{t,h} - \Sigma_{s1,h} \tag{5.17}$$

$$D_h = \frac{1}{3}\left(\Sigma_{t,h} - \Sigma_{s1,h.}\right)^{-1}. \tag{5.18}$$

Recall that the P_1 scattering matrix is obtained by weighting the P_0 scattering matrix by cosine μ both of which are analog estimates. Therefore, only the sampled interactions during the transport simulation will contribute to the calculation of the diffusion coefficients. It is imperative that users ensure that the number of particle histories produce scattering matrices with acceptable statistical errors. Typically, condensing the diffusion coefficients into the few-group structure requires the weighting of the diffusion coefficient by the flux gradient

$$D_g = \frac{\sum_{g\in h} \frac{1}{3\Sigma_{tr,h}} \frac{d\Phi_h}{dx}}{\sum_{g\in h} \frac{d\Phi_h}{dx}}. \tag{5.19}$$

Assuming separable spatial, and spectral flux for all Φ_h with $h \in g$ simplifies equation 5.19, allowing flux weighting of the diffusion coefficient during the energy group condensation.

$$D_g = \frac{\sum_{g\in h} \frac{1}{3\Sigma_{tr,h}} \Phi_h}{\sum_{g\in h} \Phi_h}. \tag{5.20}$$

5.2. HYDROGEN TRANSPORT CORRECTION

Although the out-scatter method is a fundamental approximation to the P_1 equations that effectively removes the current weighting of the transport cross section limitation in Monte Carlo generated diffusion coefficients, recent studies indicate that the out-scatter approximation produces poor macroscopic transport cross sections in lattices containing anisotropic scattering mediums. Proposed solutions include applying an in-scatter equivalent transport correction curve to the transport cross section contributions made by anisotropic scattering mediums, or by relating the diffusion coefficient to the neutron migration area (Cumulative Migration Method). The developers of Serpent were aware of this pitfall and have since included both proposed methods in its current release. Regarding the applicability of each method, the CMM method is only applicable to geometries where the homogenized region represents the entire modeled geometry, whereas the transport correction curve can be utilized for any number of homogenized regions within the model.

5.2.1. Numerical Hydrogen Transport Correction Curve. Herman et al. investigated the methods of diffusion coefficient homogenization in Monte Carlo transport codes, and the out-scatter approximation to the P_1 equations. Herman reported that weighting the fine group transport cross section before the calculation of diffusion coefficients while neglecting a diffusion correction to the out-scatter approximation resulted in tilting of reconstructed pin powers in simple LWR test lattices with a L_2 norm error of 3.6%. The proposed solution relies on the application of a correction curve to the contributions made by H^1 to the transport cross section to account for energy regions dominated by anisotropic scattering.

The proposed NLC correction method preserves the leakage, and spatial flux distribution from the B_1 equations in the diffusion coefficients. Execution of a 70-energy group, 100cm one-dimensional fixed source Hydrogen slab problem in MC21 with a buckled cosine spatial distribution provided the net leakage rate ($J_{right} - J_{eft}$) tallies for a sufficiently insulated sub region of the slab. Substituting the net leakage rate from the transport tallies into equation 5.21 leads to the calculation of the diffusion coefficients, *D*:

$$J_{right} - J_{eft} = DB^2\varphi W. \tag{5.21}$$

Where, W is the slab width.

Conversion of the fine group diffusion coefficients into the macroscopic transport-corrected cross sections permits the acquisition of an energy dependent transport correction curve defined as the ratio of macroscopic transport-corrected to total cross section. Herman concluded that weighting the fine group diffusion coefficients by the flux rather than the fine group transport cross sections reduced the L_2 norm error to .4222%, and the application of the Hydrogen correction curve further reduced the L_2 norm error to .2734% [32].

5.2.2. Analytical Hydrogen Transport Correction Curve. This study presented an analytical method for calculating the NLC B_1 diffusion coefficients in [32]. The method relies on reformulating the energy, and angular dependence of the transport equation with a buckled spatial shape in terms of inverse infinite medium transport operators. Taylor expanding these terms with respect to buckling simplifies the results, providing flexibility in the diffusion coefficients order of accuracy. This led to the following definition of the diffusion coefficient with accuracy $O(B^2)$:

$$D(E) = \frac{1}{3}\frac{\mathcal{L}_1^{-1}\mathcal{L}_0^{-1}\chi(E)}{\mathcal{L}_0^{-1}\chi(E)} + O(B^2). \tag{5.22}$$

Where, the infinite medium operator

$$\mathcal{L}_m h(E) = \Sigma_t(E)h(E) - \int_0^\infty \Sigma_{sm}(E' \to E)\, h(E')dE' \tag{5.23}$$

and the fission spectrum

$$\chi(E) = 0.453e^{-1.036E} sinh\sqrt{2.29E}. \tag{5.24}$$

Results of the study show that the analytical diffusion coefficients are equivalent to the in-scatter method when equation 5.15 utilizes the infinite current spectra, and the buckling $B^2 = 0$. These analytical diffusion coefficients are also identical to those obtained by the NLC method [32] if all the independent variables remain the same between the methods, and a sufficient order of accuracy in the expansion in B^2 [33].

5.2.3. Cumulative Migration Method. A novel homogenized transport cross section, and diffusion coefficient calculation method based on the diffusion migration area (cumulative migration method) is proposed [34]. From diffusion theory, the definition of migration area M^2 is one-sixth of the square of the average distance between the birth of a fast neutron, and its subsequent absorption as a thermal neutron.

$$M^2 = L^2 + \tau_{th}, \tag{5.25}$$

where L^2 = D/Σ_a is the diffusion area, and $\tau_{th}$is the neutron age. The idea is to break up the migration area into cumulative groups where a fast neutron is born and removed from the specified energy ranges.

$$\left(M_g^c\right)^2(E > E_0) = \frac{D_g^c(E > E_0)}{\Sigma_r^c(E > E_0)}. \tag{5.26}$$

Utilizing the one-sixth of the average square of a neutron’s slowing down distance relationship from E to E_0 (equation 5.27) permits the acquisition of the cumulative migration area $M^2(E > E_0)$ using a Monte Carlo tally for the average square of the slowing down distance of sampled particles $\overline{r_g^2}$.

$$M^2(E > E_0) \ = \frac{1}{6}\overline{r_g^2}(E > E_0). \tag{5.27}$$

Unfolding the cumulative diffusion coefficients D_g^c from equation 5.26 via flux weighting results in the desired multi-group diffusion coefficients $D_{g'}$.

$$D_g^c = \frac{\sum_{g'=1}^{g} D_{g'}\varphi_{g'}}{\sum_{g'=1}^{g} \varphi_{g'}}. \tag{5.28}$$

Validation of the CMM method implemented in the Monte Carlo OpenMC code via a pure Hydrogen infinite medium, and an assembly of the BEAVRS benchmark problem suggests that the CMM method generates transport cross sections, and diffusion coefficients that are equivalent to the in-scatter method [34].

6. REACTOR PHYSICS BENCHMARKS

Benchmarking newly developed numerical methods and simulation codes involves the acquisition of an approximate solution to problems with known results (obtained analytically, experimentally, or by validated simulations) to provide a measure of accuracy, precision, and efficiency. Essentially, these performance measures provide the user a basis to gauge the validity of results to problems with unknown solutions. Although the IAEA 2D PWR benchmark [35] serves as a preliminary benchmark due to its stature and frequent use in the reactor physics community, it ultimately lacks a physical counterpart. However, the proposed MSTR benchmark; with a physical counterpart, will provide a more effective validation of the finite element framework.

The proposed MSTR benchmark relies on the MSTR MCNP model developed by Dr. Jeffery King and its validation to experiments performed at the MSTR by Brad Richardson to provide a basis of validation [36], [37], [38]. Replication of the MSTR geometry and material definitions from the MCNP model in the creation of a new Serpent 2 MSTR model allows the stochastic generation of multi-group diffusion constants using the full-core global flux solution. Use of the stochastic full-core spatial homogenization methodology in Serpent 2 over traditional deterministic infinite lattice methods results in the minimization of spatial homogenization errors. Thereby, use of the foregoing methodology allows creation of a benchmark in which the errors are predominantly attributed to the simplified physics of the diffusion approximation and the approximation capability of the finite element method.

6.1. IAEA 2-D PWR

The IAEA 2-D PWR benchmark is a variation of the classical two-group IAEA 3-D PWR benchmark problem, proposed by B. Micheelsen to the IAEA Panel on Reactor-Burnup Physics in 1971, and later included in the Argonne Code Center: Benchmark Problem Book [35]. Figure 6.1 illustrates the multi-region IAEA 2-D PWR benchmark core configuration utilizing quarter symmetry.

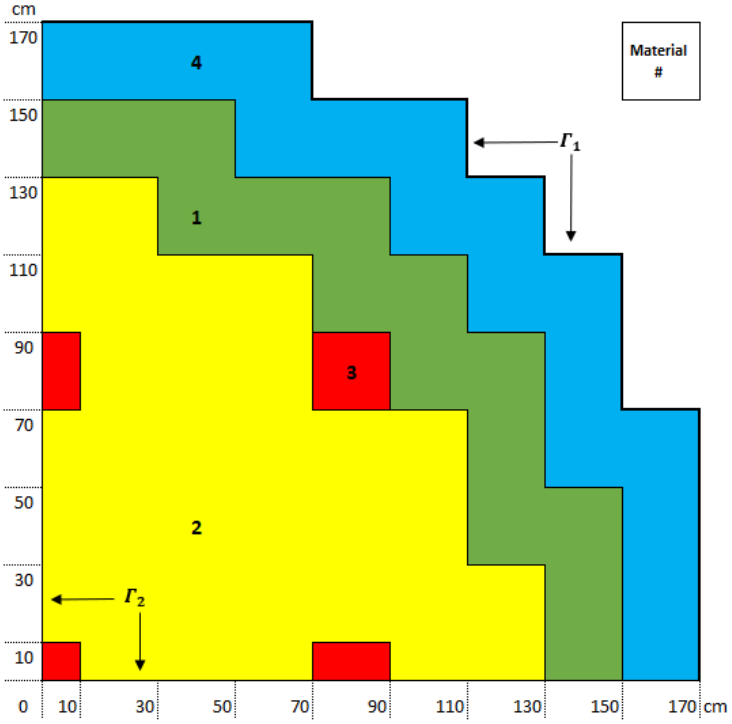


Figure 6.1 IAEA 2-D PWR Benchmark Configuration [35]

The core configuration consists of 20 cm pitched square lattices that are defined by regions containing a smeared fuel/absorber rod assembly (material #3), two varying fuel composition assemblies (material #1-2), and a water reflector at the core periphery (material #4). The pre-computed homogenized parameters associated with each material region are presented in Table 6.1. Upon examination of the multi-group constants presented in Table 6.1, one should expect strong local thermal (group 2) flux perturbations at the absorber rod, and water reflector material interfaces due to the large spatial discontinuities in the multi-group constants.

Table 6.1 IAEA 2D PWR Benchmark Homogenized Multi-Group Constants [35]

Region	D_1 (cm)	D_2 (cm)	$\Sigma_{1\rightarrow 2}$ (cm^{-1})	Σ_{a1} (cm^{-1})	Σ_{a2} (cm^{-1})	$\nu\Sigma_{f2}$ (cm^{-1})	Material
1	1.5	0.4	0.02	0.01	0.080	0.135	Fuel 1
2	1.5	0.4	0.02	0.01	0.085	0.135	Fuel 2
3	1.5	0.4	0.02	0.01	0.130	0.135	Fuel 2 + Rod
4	2.0	0.3	0.04	0.00	0.010	0.000	Reflector

The objective concerning the IAEA 2-D PWR benchmark [35] is to obtain the largest eigenvalue k_{eff}, and global flux distributions of a coupled two-group critical problem (equation 6.1, where group one represents the high energy fast neutrons, and group two represents the lesser energetic thermal neutrons) bounded by the assumptions of no incoming neutron current at the outer boundary, and no net current at the symmetry boundary. Accounting for the axial leakage in the 2-D problem requires the addition of a constant to the group removal cross sections defined as the product of the group buckling factor, $B_{z,1,2}^2 = 8 \times 10^{-5}$, and diffusion coefficient, D_g.

$$\begin{cases} -D_1\nabla^2\varphi_1 + \left(\Sigma_{a1} + \Sigma_{1\to2} + D_1B_z^2\right)\varphi_1 = \dfrac{1}{k_{eff}}\nu\Sigma_{f2}\varphi_2 \;\; in\,\Omega, \\ -D_2\nabla^2\varphi_2 + (\Sigma_{a2} + D_2B_z^2)\varphi_2 = \Sigma_{1\to2}\varphi_1 \;\; in\,\Omega, \\ \nabla\varphi_g \cdot \vec{n} = 0 \;\; on\,\Gamma_1 \subset d\Omega, \\ \nabla\varphi_g \cdot \vec{n} + \dfrac{1}{2D_g}\varphi_g = 0 \;\; on\,\Gamma_2 \subset d\Omega, \end{cases} \tag{6.1}$$

Since the reactor is permissible to operate at any arbitrary power rating, the global flux distributions are normalized such that the neutron generation rate over the active fuel volume is one.

$$\frac{1}{V_{core}}\int_{V_{core}}\sum_{g}\nu\Sigma_{fg}\varphi_g dv = 1. \tag{6.2}$$

6.1.1. Reference Solutions. Table 6.2 contains the reference eigenvalues, and maximum inner core thermal flux obtained by the finite difference, finite element, and nodal expansion methods published by the Argonne National Laboratory Benchmark Committee [35].

Table 6.2 IAEA 2-D reference eigenvalues and inner core maximum thermal flux [35]

Method	Grid	Eigenvalue	$\varphi_{2,max}(x,y)$
Mesh centered finite difference	272 x 272 meshes	1.02958	-
Quadratic quadrillateral finite element	36 x 36 meshes	1.0296	11.18 (30,30)
Nodal expansion	h = 3.5 cm	1.029585	11.206 (31,31)

Despite the differences in the mathematical basis of each numerical method, the reference results indicate that the converged maximum eigenvalue is approximately 1.02958. The magnitude, and location of the maximum inner core thermal flux suggests that minimal deviations in the spatial solutions exist. The agreement of results obtained from varying numerical methods indicates that the approximate solutions have converged. Figure 6.2 presents the reference global radial flux traverses obtained by the nodal expansion method with a mesh size of h = 3.5 cm along the x-axis, and the diagonal y = x.

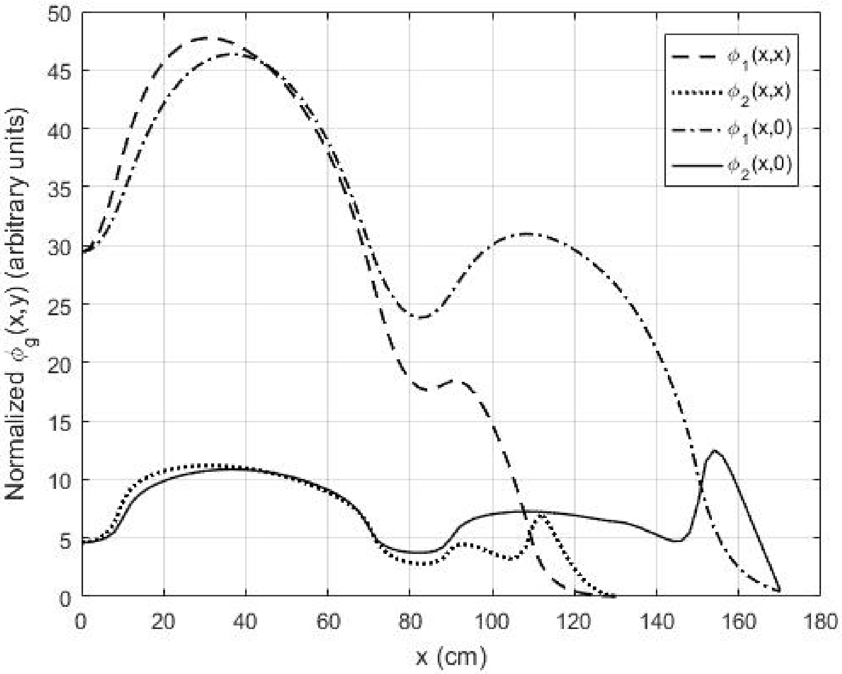


Figure 6.2 IAEA 2-D reference radial flux traverses [35]

It is evident from Figure 6.2 that strong flux perturbations exist at the absorber rod/fuel interface due to the increased parasitic neutron absorption and in the fuel/reflector due to the increased thermalization of fast neutrons. Figure 6.3 (pg. 57) contains the reference normalized assembly average fast flux obtained by the finite element and nodal expansion methods. Note: the FEM results are based on quarter core symmetry, while the NEM results are based on an eighth core symmetry. The assembly with the greatest normalized average fast flux (highlighted in yellow) from the FEM is 46.6185, whereas the NEM resulted in 46.7020. Figure 6.4 (pg. 58) contains the reference normalized assembly average thermal fluxes. The assembly with the greatest average thermal flux corresponds with the same assembly of the greatest average fast flux. For the FEM, the greatest average thermal flux was 10.9427 and for the NEM it was 10.9620.

6.1.2. Benchmark Results. The IAEA 2-D PWR benchmark [35] was carried out for both structured triangular and rectangular elements with mesh sizes $\Delta_{x,y}$ of 10, 5, 2, 1, and 0.5 cm. For the triangular element cases, the element order was also varied using linear, quadratic, and cubic interpolation polynomials. Likewise, the element order was also varied for the rectangular cases; however, only linear, and quadratic interpolation polynomials are used. Table 6.3 (pg. 78) contains the general results of the benchmark that includes: (1) the largest eigenvalue and the change in reactivity from the NEM reference value published in the ANL benchmark book [35]; (2) the maximum normalized thermal flux in the inner core with its corresponding location coordinates (initial spatial convergence check); and (3) the total number of unknowns for each benchmark case. Numerical convergence to the IAEA 2-D PWR eigenvalue is observed (see Table 6.3) for all element shapes and orders of polynomials. The distinguishing factor between the element shapes, polynomial orders, and mesh sizes are their individual rates of convergence and efficiency.

3.3944 -	3.2431 -	2.4578 -	0.7188 -					FEM NEM
20.3675 -	19.8959 -	16.7075 -	5.9875 -	2.4387 -	0.5778 -			
29.432 -	29.8661 -	29.122 -	22.6278 -	14.4753 -	4.0186 -	0.6479 0.6420		
29.9296 -	32.7203 -	33.6976 -	28.5075 -	20.7859 -	14.0517 14.0150	4.0194 3.9930	0.5781 0.5730	
26.5281 -	34.1781 -	37.1833 -	30.8752 -	20.3850 20.3910	20.7890 20.7660	14.4796 14.4390	2.4397 2.4210	
38.6266 -	41.4685 -	42.3909 -	37.6359 37.6770	30.8782 30.8960	28.5131 28.4980	22.6343 22.5870	5.9895 5.5920	0.7190 0.7120
45.7876 -	46.6162 -	46.2972 46.3720	42.3941 42.4480	37.1893 37.2150	33.7055 33.6970	29.1304 29.0890	16.7128 16.6620	2.4586 2.4380
41.8016 -	45.2435 45.2880	46.6185 46.7020	41.4731 41.5350	34.1849 34.2050	32.7299 32.7230	29.8766 29.8420	19.9035 19.8540	3.2444 3.2200
32.3916 32.4670	41.8032 41.8900	45.7911 45.8770	38.6309 38.6950	26.5336 26.5640	29.9397 29.9380	29.4437 29.4110	20.3760 20.3270	3.3959 3.3700

Figure 6.3 IAEA 2-D reference assembly average group 1 flux [35]

8.0138 -	7.6236 -	5.877 -	2.8301 -					FEM NEM
5.6078 -	5.4661 -	5.1497 -	12.4623 -	6.0648 -	2.2525 -			
6.9263 -	7.0461 -	7.2307 -	6.2849 -	4.4419 -	8.3528 -	2.5767 2.5640		
6.9224 -	7.6736 -	7.9302 -	6.7171 -	5.081 -	4.3515 4.3320	8.3545 8.3340	2.2534 2.2380	
4.5176 -	7.9143 -	8.728 -	7.1565 -	3.4904 3.4860	5.0817 5.0780	4.4432 4.4240	6.0674 6.0380	
8.9502 -	9.7246 -	9.9503 -	8.8264 8.8360	7.1572 7.1630	6.7184 6.7140	6.2867 6.2680	12.4665 12.4410	2.8309 2.8120
10.7476 -	10.9421 -	10.8673 10.8850	9.9511 9.9640	8.7293 8.7350	7.9321 7.9290	7.2328 7.2230	5.1513 5.1270	5.8789 5.8390
9.6798 -	10.6093 10.6300	10.9427 10.9620	9.7256 9.7400	7.9158 7.9240	7.6758 7.6750	7.0485 7.0400	5.4682 5.4500	7.6265 7.5790
5.5137 5.5230	9.6801 9.7010	10.7484 10.7680	8.9512 8.9680	4.5185 4.5190	6.9247 6.9260	6.9290 6.9210	5.6102 5.5920	8.0174 7.9680

Figure 6.4 IAEA 2-D reference assembly average group 2 flux [35]

For a given element shape and polynomial order, refining the mesh size (h-refinement) led to decreased total reactivity change against the reference NEM eigenvalue until the solutions converged; however, convergence was not attained for the linear triangle and bilinear rectangle elements until the final mesh size of 0.5×0.5 cm. This is an indication that there are not enough basis functions to efficiently capture the large gradients of the intra element solution unless prohibitively small mesh sizes are utilized. Nevertheless, increased polynomial orders led to faster h-refinement convergence with a lesser number of unknowns when compared to the linear triangular and bilinear rectangular cases, which is consistent with the FEM error convergence proofs.

Figure 6.5 contains two plots of the linear triangle thermal and fast x-axis radial flux traverse h-refinement results. It is evident that h-refinement leads to the overall improvement of the spatial solution. However, as previously discussed, the linear triangle and bilinear rectangle are inefficient as a greater number of unknowns are required to attain numerical convergence. Consequently, h-refinement leads to an increase in computational complexity of sparse operations. By introducing more global basis functions (Nb) with mesh refinement, the size of matrix $A = Nb \times Nb$ increases along with the number of non-zero elements. From MATLAB's documentation on sparse operations, which relies on Tim Davis's sparse Cholesky factorization routine from SuiteSparse (highly optimized LAPACK and level 3 BLAS routines), computational complexity is proportional the number of non-zero elements and linearly dependent on the column size of the sparse matrix [39]. Thus, a better approach would be either p-refinement, or the combination of both h-refinement and p-refinement.

Table 6.3 IAEA 2-D benchmark eigenvalue results

Element shape	Order	$\Delta_{x,y}$(cm)	k_{eff}	$\Delta\rho$ (pcm)[a]	$\varphi_{2,\max}$ (x, y) Fuel region	Total unknowns
Triangular	Linear	10 x 10	1.031233	-155.2	9.60 (40, 30)	552
-	-	5 x 5	1.029841	-24.1	10.80 (30, 30)	2,066
-	-	2 x 2	1.029618	-3.1	11.14 (32, 30)	12,392
-	-	1 x 1	1.029593	-0.7	11.19 (31, 31)	48,882
-	-	.5 x .5	1.029587	-0.2	11.20 (31, 31)	194,162
-	Quadratic	10 x 10	1.029716	-12.3	11.04 (30, 30)	2,066
-	-	5 x 5	1.029593	-0.8	11.18 (30, 30)	7,986
-	-	2 x 2	1.029585	0.0	11.21 (31, 31)	48,882
-	-	1 x 1	1.029585	0.0	11.21 (31, 31)	194,162
-	-	.5 x .5	1.029585	0.0	11.21 (31, 30.8)	773,922
-	Cubic	10 x 10	1.029591	-0.5	11.20 (30, 30)	4,544
-	-	5 x 5	1.029585	0.0	11.20 (31.7, 30)	17,762
-	-	2 x 2	1.029585	0.0	1.21 (30.7, 30.7	109,472
-	-	1 x 1	1.029585	0.0	11.21 (31, 31)	435,842
Rectangular	Bilinear	10 x 10	1.031086	-141.4	9.70 (30, 30)	552
-	-	5 x 5	1.029845	-24.5	10.83 (30, 30)	2,066
-	-	2 x 2	1.029620	-3.3	11.14 (32, 30)	12,392
-	-	1 x 1	1.029594	-0.8	11.19 (31, 31)	48,882
-	-	.5 x .5	1.029587	-0.2	11.20 (31, 31)	194,162
-	Biquadratic	10 x 10	1.029601	-1.5	11.33 (35, 30)	2,066
-	-	5 x 5	1.029585	0.0	11.25 (32.5, 30)	7,986
-	-	2 x 2	1.029585	0.0	11.21 (31, 30)	48,882
-	-	1 x 1	1.029585	0.0	11.21 (31, 30.5)	194,162
-	-	.5 x .5	1.029585	0.0	11.21 (31, 30.8)	773,922

[a] Reference value 1.029585 NEM (ANL, 1977)

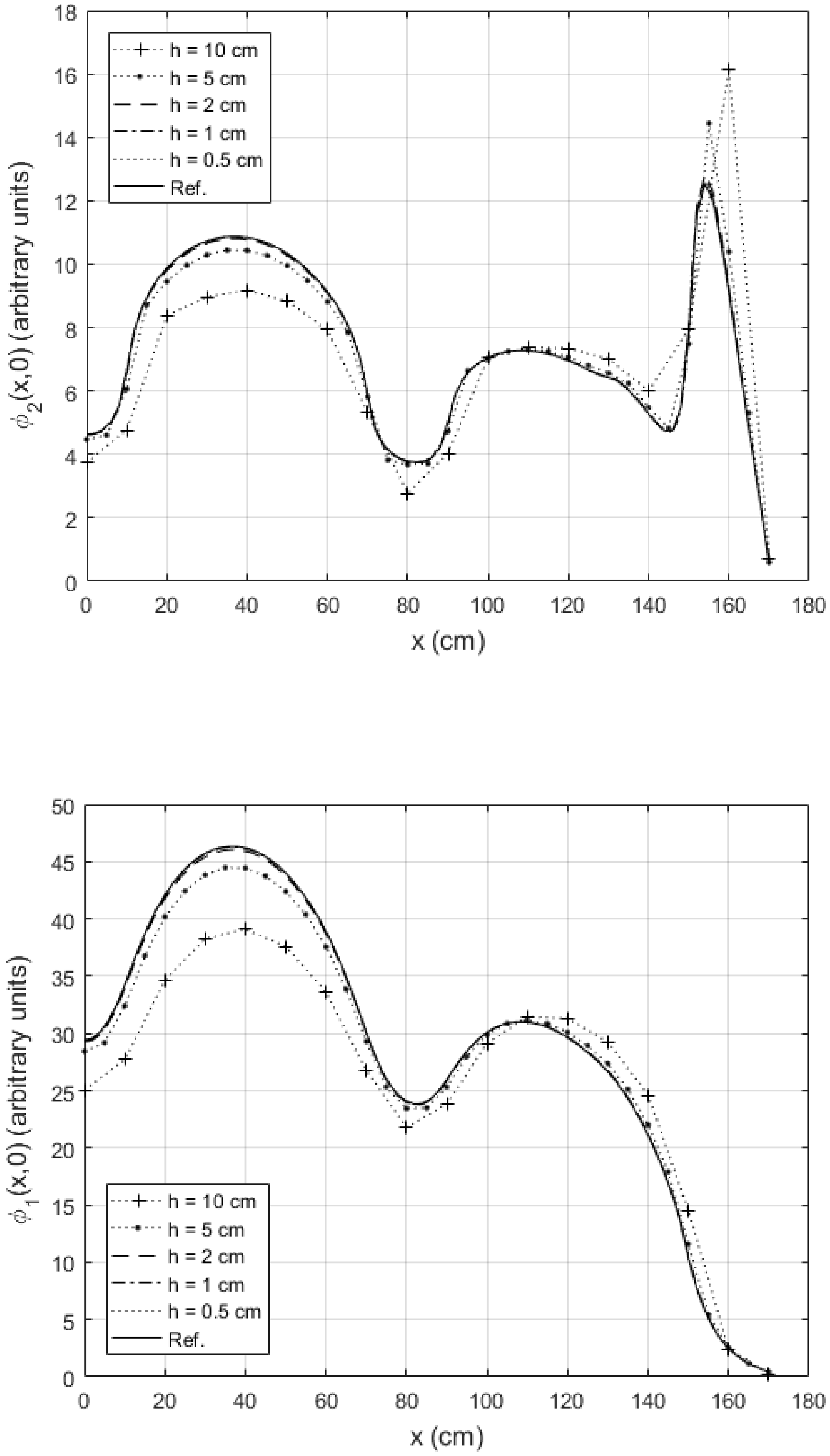


Figure 6.5 Linear triangle h-refinement (top) thermal, (bottom) fast flux traverse [35]

From Figure 6.6, the p-refinement with a 10cm mesh yields similar result as the h-refinement, albeit with a less number of unknowns. Changing only the interpolation polynomial order to quadratic from linear almost attains the converged spatial solution. From Table 6.3 and Figure 6.6, the 10cm mesh cubic triangle case is on the edge of convergence with a ref. reactivity difference of -0.5 pcm. Comparing the 10cm cubic triangle with the 0.5cm linear triangle, it is evident that the former leads to a reduction of the number of unknowns by a factor of 42. Nevertheless, numerical convergence to the published IAEA 2D eigenvalue and spatial flux solution can be obtained with any combination of interpolation polynomials and h-refinement or vice versa. Other h-refinement radial flux traverse plots can be found in Appendix B.

To further the support of spatial convergence, Figure 6.7 contains the assembly average thermal flux for the quadratic triangle with a mesh size of 2 cm. The average RPE over all assemblies regarding the ANL published FEM average assembly fluxes was 0.336%, while the maximum RPE of 1.0824% was in the water reflector [35]. Minute differences between the obtained FEM and reference solutions can be attributed to the reference case using quadratic rectangular elements. The average thermal assembly fluxes can also be compared to the NEM reference values in Figure 6.4. Figure 6.8 is an interpolated normalized thermal flux map using the solution from the cubic triangular mesh of size 1cm. The largest normalized thermal neutron flux of approximately 18.70 (arbitrary units) is in the water reflector, which is adjacent to three fuel assemblies. The largest inner core thermal flux is approximately 11.21 along the radial traverse $y = x$.

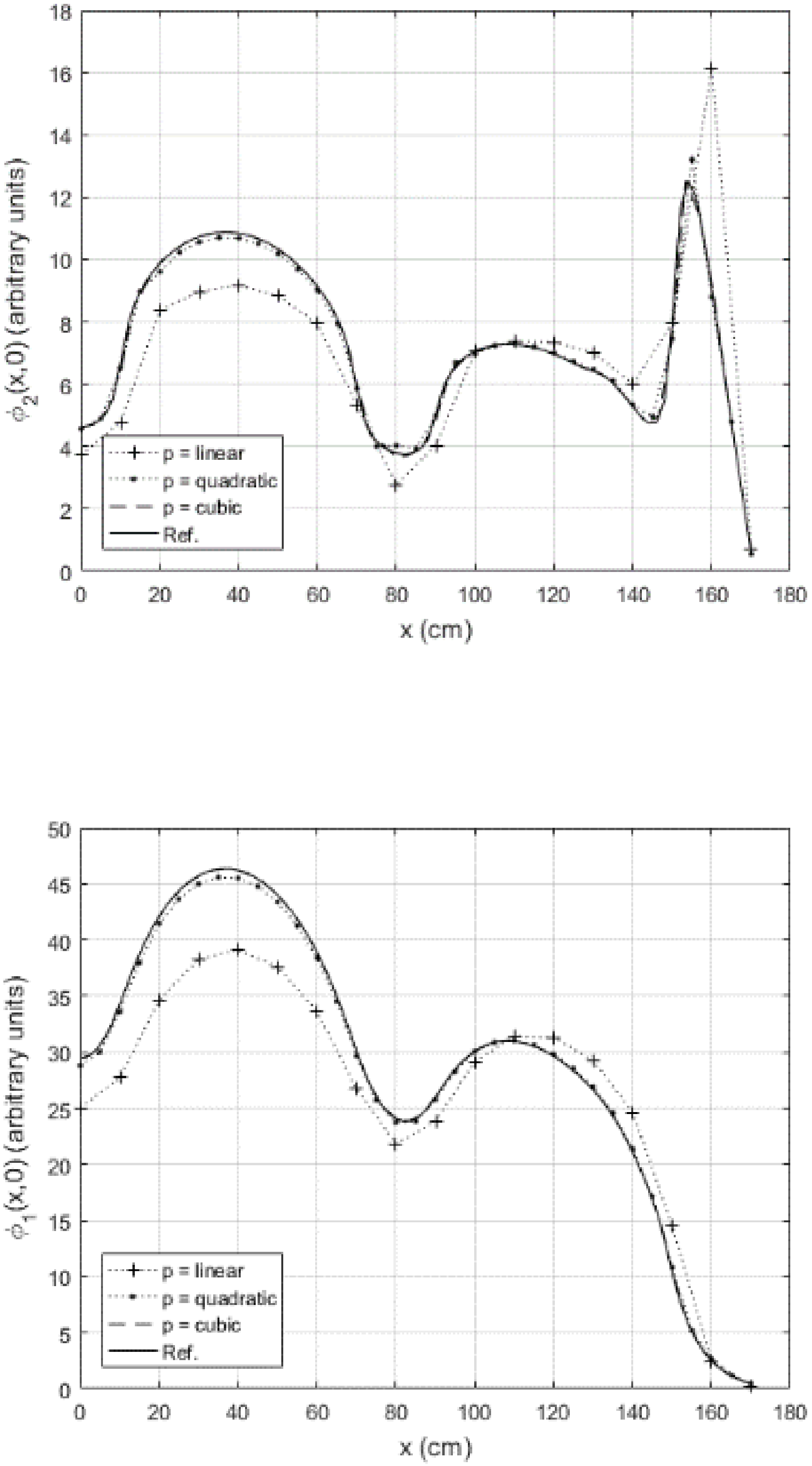


Figure 6.6 Linear triangle 10cm mesh p-refinement (top) thermal (bottom) fast flux [35]

8.0138 7.9377 0.9498	7.6236 7.5499 0.9665	5.877 5.8194 0.9803	2.8301 2.8003 1.0545				Ref. FEM RPE (%)	Avg. RPE (%) 0.3360
5.6078 5.6092 0.0242	5.4661 5.4669 0.0155	5.1497 5.1604 0.2086	12.4623 12.3666 0.7676	6.0648 6.0107 0.8924	2.2525 2.2293 1.0315			
6.9263 6.9210 0.0760	7.0461 7.0405 0.0798	7.2307 7.2203 0.1436	6.2849 6.2887 0.0612	4.4419 4.4524 0.2366	8.3528 8.2882 0.7729	2.5767 2.5518 0.9669		
6.9224 6.9230 0.0093	7.6736 7.6745 0.0123	7.9302 7.9295 0.0094	6.7171 6.7142 0.0434	5.0810 5.0740 0.1376	4.3515 4.3618 0.2372	8.3545 8.2882 0.7931	2.2534 2.2293 1.0711	
4.5176 4.5417 0.5329	7.9143 7.9191 0.0609	8.728 8.7342 0.0712	7.1565 7.1593 0.0398	3.4904 3.5045 0.4050	5.0817 5.0740 0.1513	4.4432 4.4524 0.2073	6.0674 6.0107 0.9349	
8.9502 8.9626 0.1387	9.7246 9.7376 0.1336	9.9503 9.9624 0.1218	8.8264 8.8353 0.1012	7.1572 7.1593 0.0300	6.7184 6.7142 0.0627	6.2867 6.2887 0.0325	12.4665 12.3666 0.8010	2.8309 2.8003 1.0824
10.7476 10.7669 0.1799	10.9421 10.9606 0.1694	10.8673 10.8833 0.1468	9.9511 9.9624 0.1138	8.7293 8.7342 0.0563	7.9321 7.9295 0.0334	7.2328 7.2203 0.1726	5.1513 5.1604 0.1775	5.8789 5.8194 1.0123
9.6798 9.6966 0.1733	10.6093 10.6295 0.1902	10.9427 10.9606 0.1639	9.7256 9.7376 0.1233	7.9158 7.9191 0.0420	7.6758 7.6745 0.0164	7.0485 7.0405 0.1139	5.4682 5.4669 0.0229	7.6265 7.5499 1.0041
5.5137 5.5515 0.6863	9.6801 9.6966 0.1702	10.7484 10.7669 0.1724	8.9512 8.9626 0.1275	4.5185 4.5417 0.5128	6.9247 6.9230 0.0239	6.929 6.9210 0.1149	5.6102 5.6092 0.0186	8.0174 7.9377 0.9943

Figure 6.7 Quadratic triangle 2cm mesh assembly averaged thermal flux [35]

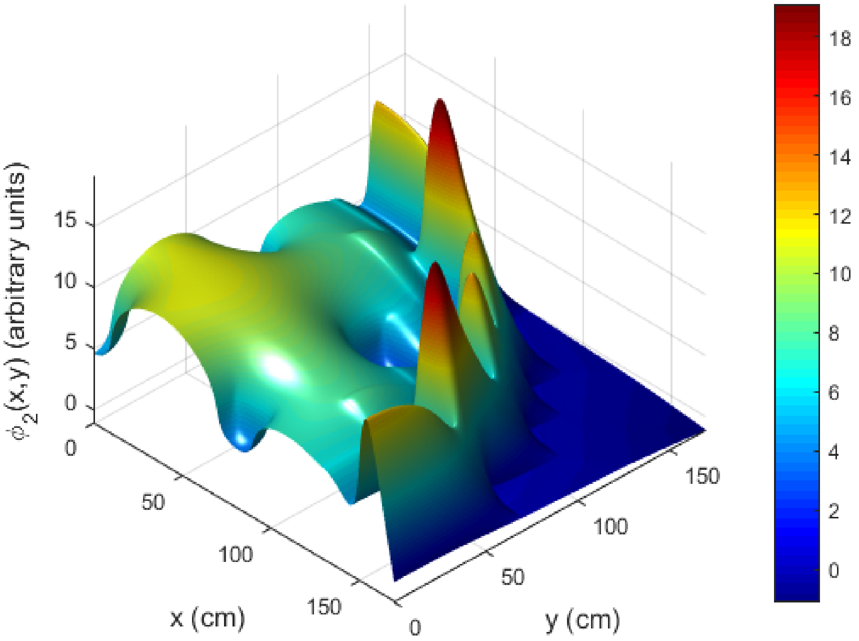


Figure 6.8 Triangle Cubic 1cm mesh: thermal flux map results

Figure 6.9 is the interpolated fast flux map counterpart. The largest fast flux is approximately 47 (arbitrary units) along the radial traverse $y = x$. The solution gradient for the fast flux is smoother when compared to the thermal flux map, as the absorption cross section for the fast group is approximately smaller by a factor of 8. Nevertheless, fast flux perturbations still exist in the fuel/absorber rod regions and the fast flux rapidly approaches zero in the water reflector. Overall, the IAEA 2D benchmark presents some challenges from the inclusion of the fuel/absorber rods and the water reflector, where the capture of the steep thermal flux gradients requires either a combination of h/p-refinement, or major refinement in one category. Another approach would be to use unstructured meshes, which allows the specification of varying mesh sizes in regions with steep solution gradients.

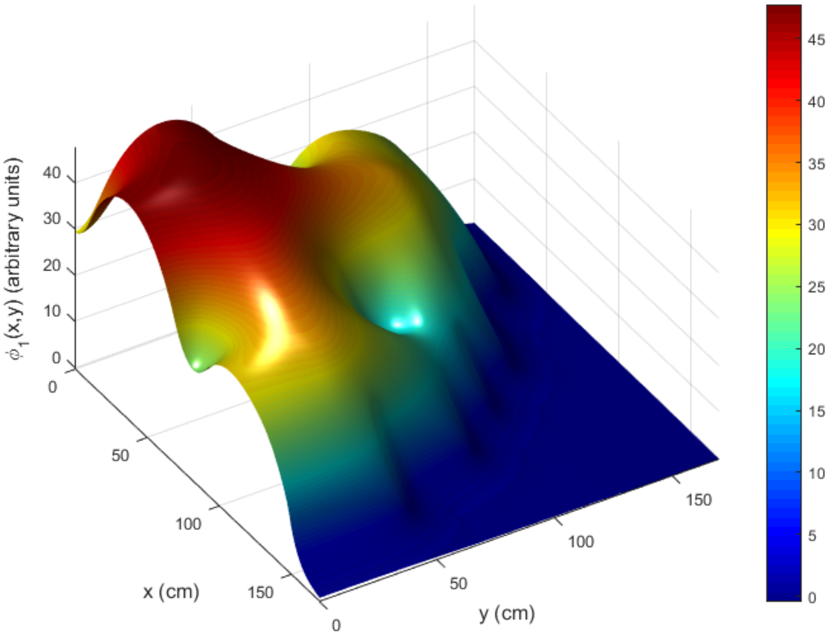


Figure 6.9 Triangle Cubic 1cm mesh: fast flux map results

6.2. MISSOURI S&T REACTOR

The Missouri S&T Reactor (Figure 6.10), formerly known as the University of Missouri-Rolla Reactor, is a light water, open pool reactor designed after the Bulk Shielding Reactor at Oak Ridge National Laboratory. December 9, 1961, marked the reactors first sustained criticality event on an initially licensed maximum core power of 10 kW [40]. In 1966, the MSTR would receive a licensed power uprate to 200 kW [40]. Conversion of the initial highly enriched to low enriched (19.9% ${}^{235}_{92}U$) uranium fuel would take place in 1992 [40], [41]. Ultimately, the MSTR serves as a supplemental educational tool for enrolled students and provides researchers a means to carry out reactor physics and irradiation experiments.

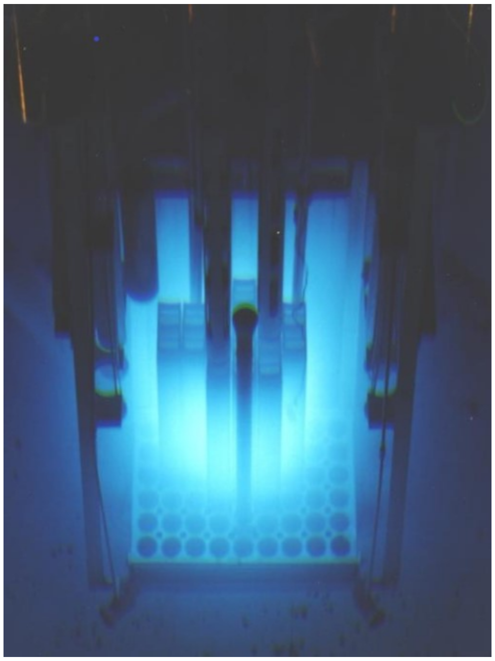


Figure 6.10 MSTR core in operation [42]

A grid plate comprised of 54 lattice locations (indexed by columns 1 through 9 and rows A through F) allow the placement of fuel/control elements, source holder, and experimental apparatuses. Each fuel element (of dimension 3 in. by 3 in. by 3 ft. see Figure 6.11) contains eighteen 0.06-inch-thick curved fuel plates each containing 12.5 grams of low enriched uranium silicide clad in aluminum that extends 24 inches in length [42]. Coolant channels exist in-between each fuel plate, where natural convection removes energy from the fuel plate to the ultimate heat sink. At the top of the fuel element a handle allows the use of hooked tools to assist in the movement of the fuel element, while at the bottom of the fuel element, a hollow cylindrical nose piece allows the secured placement of the fuel element into the grid plate opening.

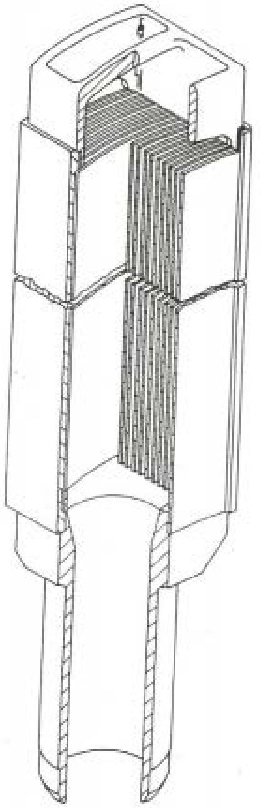


Figure 6.11 Fuel Element Schematic [38]

The control elements are dimensionally identical to the fuel elements, however, the removal of eight middle fuel plates accommodate a control rod guide tube. Of the four control rods, three are comprised of boronated stainless steel 304 used for shutdown, SCRAM, and coarse reactivity manipulation, while the final regulating rod is a hollow stainless steel 304 tube used for fine reactivity manipulation. Two pneumatic rabbit tubes provide small irradiation samples in core access to either the entire reactors neutron energy spectrum, or the epithermal and fast neutrons only, where the later requires use of a cadmium lined rabbit tube to filter out thermal neutrons. Researchers can also irradiate large specimens using hollow aluminum void tubes inserted into empty lattice positions at the core periphery. A source holder along with a removeable plutonium-beryllium neutron source provides signal to the startup instrumentation [42].

The grid plate suspends from a moveable bridge that places the core in a 9 ft. wide by 19 ft. long by 27 ft. deep pool containing 32,000 gallons demineralized water. Movement of the bridge allows repositioning of the core with respect to irradiation instruments located in the pool. The thermal column is a 3.5 ft. by 3.5 ft. by 5 ft. graphite block located behind the core that provides irradiation specimens with a source of thermal neutrons, where the two locations of the core with respect to the thermal column are the W and T configurations [42]. A 6 in. diameter aluminum tube with a lead shield (to shield gammas) positioned on the side of the pool behind the reactor extends to the right side of the grid plate also provides a beam of neutrons for irradiation experiments [42].

6.2.1. MCNP Model Description. Dr. Jeffery King began developing a high-fidelity MSTR MCNP model in 2007 that includes the reactor pool, spent fuel storage pit, thermal-column, beam port, two rabbit tubes, grid plate, and fuel/control elements [36]. Geometrical and material definitions were supplied from the reactors design schematics, material shipping papers, and physical measurements. When selecting fuel elements for a configuration, users have the option of choosing the as-specified fuel element material composition, or the element specific compositions specified by the manufacturer. The model is modular in the sense that the core configuration can be rapidly changed, and the control rods can be individually manipulated through universe fills and transformations. Figure 6.12 is the YZ plane view of the MSTR MCNP model.

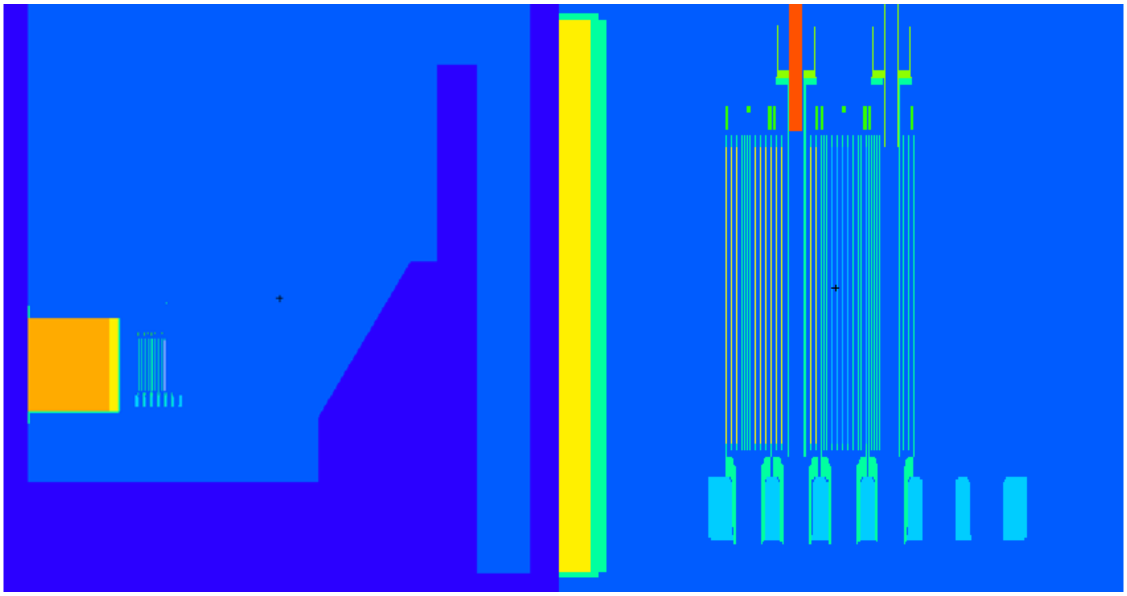


Figure 6.12 MCNP YZ plane (left) entire geometry (right) exploded view of the core

Concerning the continuous-energy neutron data specifications, the cross sections have been updated to 293.6 K ENDF/B-VII.0 .70c and 293.6 K ENDF/B-VII.0 thermal scattering $S(\alpha,\beta)$ for light water (lwtr.10t) and graphite (grah.10t) to be consistent with the Serpent 2 cross section specifications [43]. Material definitions (see Table 6.4 for isotopic compositions) for the nitrogen inside the rabbit tubes and air inside the beam port at normal temperature and pressure have also been added to the MCNP model to eliminate singularities for spatial homogenization in Serpent 2. Table 6.4 lists the individual isotopic compositions and densities defined within the material card of the MCNP and Serpent 2 models. The current geometric layout of the core corresponds to the approach to criticality experiment for the 120W core configuration which will be used to prepare the proposed MSTR benchmark.

Table 6.4 MCNP and Serpent material isotopic compositions [36]

Material	Density	Isotopic composition			
	(atom/b-cm)	(atom %)			
Fuel (as-specified)	0.0543525	U-235: 3.2287 U-238: 12.9533	Si-28: 9.9366 Si-29: 0.5046	Si-30: 0.3326	Al-27: 73.0443
Cladding (wrought 6061 alloy)	0.060004	Al-27: 97.8233 Si-28: 0.6140 Si-29: 0.0312 Si-30: 0.0206	Mg-0: 1.0536 Fe-54: 0.0133 Fe-56: 0.2093 Fe-57: 0.0048	Fe-58: 0.0006 Cr-50: 0.0049 Cr-52: 0.0939 Cr-53: 0.0106	Cr-54: 0.0026 Cu-63: 0.0811 Cu-65: 0.0362
Fuel element handle (cast A356-T6 Al alloy)	0.059363	Al-27: 92.8066 Si-28: 6.1640 Si-29: 0.3130 Si-30: 0.2063	Mg-0: 0.3567 Ti-0: 0.0736 Fe-54: 0.0031 Fe-56: 0.0490	Fe-57: 0.0011 Fe-58: 0.0002 Zn-0: 0.0124 Mn-55: 0.0099	Cu-63: 0.0029 Cu-65: 0.0013
Grid plate (1100 series Al)	0.060441	Al-27: 99.9469	Cu-65: 0.0164	Cu-63: 0.0367	
Keno concrete	0.081776	H-1: 16.8018 H-2: 0.0019 O-16: 56.0969 O-17: 0.0214	Si-28: 18.7429 Si-29: 0.9518 Si-30: 0.6274 Al-27: 2.1343	Na-23: 2.1365 Ca-0: 1.8596 Fe-54: 0.0248 Fe-56: 0.3896	Fe-57: 0.0090 Fe-58: 0.0012
Control rod (1.6 wt% Borated SS304)	0.090540	Fe-54: 3.6869 Fe-56: 57.8772 Fe-57: 1.3366 Fe-58: 0.1779	Cr-50: 0.8237 Cr-52: 15.8843 Cr-53: 1.8011 Cr-54: 0.4483	Ni-58: 5.7165 Ni-60: 2.2020 Ni-61: 0.0957 Ni-62: 0.3052	Ni-64: 0.0777 Mn-55: 1.8887 B-10: 1.5279 B-11: 6.1501
Regulating Rod (SS304)	0.086240	Fe-54: 4.0229 Fe-56: 63.1511 Fe-57: 1.4584 Fe-58: 0.1941	Cr-50: 0.8781 Cr-52: 16.9327 Cr-53: 1.9200 Cr-54: 0.4779	Ni-58: 6.0938 Ni-60: 2.3473 Ni-61: 0.1020 Ni-62: 0.3253	Ni-64: 0.0829 Mn-55: 2.0133
Lead	0.032958	Pb-206: 24.4422	Pb-207: 22.4138	Pb-208: 53.1440	
Cadmium	0.046286	Cd-106: 1.2500 Cd-108: 0.8900	Cd-110: 12.4900 Cd-111: 12.8000	Cd-112: 24.1300 Cd-113: 12.2200	Cd-114: 28.7300 Cd-116: 7.4900
Water	0.033427	H-1: 66.6590	H-2: 0.0077	O-16: 33.3206	O-18: 0.0127
Air (NTP)	4.945200E-05	N-14: 78.0840	O-16: 20.9470	Ar-40: 0.9340	
Nitrogen (NTP)	5.008700E-05	N-14: 99.6300	N-15: 0.3700		
Graphite	9.03E-02	C-0: 100.0			

6.2.2. Experimental MCNP Model Validation. Initial validation of the MCNP model developed by Dr. Jeffery King included the approach to criticality, and axial flux profile experiments for the 120W MSTR core configuration [38], [36]. The prediction of the control rod height whereon the core becomes critical ($K_{eff} = 1$) was determined by extrapolating successive points on a 1/M vs control rod height plot to a value of 1/M = 0, where 1/M is equal to the neutron counts recorded by a fission chamber at the initial control rod height over the counts associated with the current control rod height [22].

$$\frac{1}{M} = \frac{C_0}{C}, \tag{6.1}$$

where

$$M = \frac{1}{1 - K_{eff}}. \tag{6.2}$$

The process of control rod withdrawal to a height in-between the current, and predicted value occurs until the control rod height is within 0.1 inch of the previously predicted critical control rod height. Finally, removal of the control rod to a height which sustains criticality concludes the experiment. Validation of the MCNP model following the preceding experimental procedure at the exact control rod heights resulted in an average model vs experimental predicted control rod height error of 0.59 ± 0.08 inches, where the experimental critical control rod height was 20.0 inches, and the MCNP critical control rod height was 19.3 ± 0.6 inches [38], [36].

MSTR facilities lack instrumentation to directly measure the axial flux profile. Nevertheless, an approximate axial flux profile can be determined experimentally. The experiment requires irradiating the copper wire for 10 minutes at a power rating of 500

Watts. After the initial irradiation, an allocation of time was set aside to allow the short-lived Copper-64 isotope to decay. Following the cool down period, cutting the copper wire into 1-inch segments allows the measurement of gamma activity associated with each individual segment. Due to the linear relationship between gamma activity and the in-core neutron flux, a graph of the gamma activities will have the same shape as the axial flux profile [38], [36].

Reproduction of the experiment in the MCNP model included placing a 50-inch copper wire of 0.0225-inch diameter in the same fuel element as the physical experiment, where the modification of 50 1-inch cell flux tallies provides the integral neutron absorption reaction rates. Three separate simulation cases: a critical core, control rods fully withdrawn, and a core divided into top and bottom halves with different temperatures, provide an array of integral absorption reaction rates. Inclusion of the core temperature profile produced the most accurate axial flux profile in MCNP with an average deviation of 10.9% from the experimental values, while the isothermal critical, and control rods fully withdrawn cases produced average deviations of 12.4% and 13.6% [38], [36].

Other MCNP validation efforts included two distinct experiments at the Missouri S&T research reactor that provided modeling of (1) temperature effects; and (2) void effects on reactivity [38], [37]. The measurement of temperature effects on reactivity relied upon the operation of the reactor at the maximum licensed power of 200 kW, so the core and pool could gradually heat throughout the day. To sustain criticality, the reactor’s control system continually withdrew the regulating rod to counteract the temperature feedback effects on reactivity. Utilizing known differential regulating rod reactivity values, together with acquired regulating rod heights and thermocouple temperature readings at 15-minute intervals, permitted calculation of total reactivity change attributable to the heating of the core [38], [37].

Reproduction of the temperature effects on reactivity experiments for three separate temperature profiles validated the MCNP model. The three individual temperature profiles included: (a) isothermal at the upper thermocouple reading; (b) isothermal at the lower thermocouple reading; and (c) a four-region core consisting of (i) linear interpolation of both upper and lower thermocouples readings providing temperatures for the two regions within the core; (ii) the upper thermocouple reading for the region above the core; and (iii) the lower thermocouple reading for the region below the core. Furthermore, each of the three identified temperature profiles included the simulation at every regulating rod height obtained at the 15-minute interval. Due to the minimal deviation of the thermocouple readings with respect to room temperature (293.15 K), it is only necessary to reflect the temperature dependence in the water density, and free-gas thermal treatment on elastic scattering cross sections [38], [37].

Richardson concluded that the replicated temperature effects on reactivity experiment in MCNP resulted in maximum eigenvalues (k_{eff}) within 0.40% of the experimental values; however, in all cases, the model overpredicted criticality such that the core is slightly super-critical when theoretically the values should be one. Richardson further noted that the deviation from unity remained relatively constant throughout the simulations and thereby indicated that the source of error to be likely the result of unaccounted fuel burnup, and limited detail concerning temperature distributions [38], [37].

The criticality simulation regarding previously mentioned temperature profile 1 (an isothermal core at the upper core thermocouple) resulted in eigenvalues that ranged between 1.00234 and 1.00248 with a standard deviation of .00018. For temperature profile 2 (an isothermal core at the lower core thermocouple) resulted in eigenvalues that ranged between 1.00296 and 1.00383 with a standard deviation of .00018. Lastly, the temperature profile 3 (a four-region core) resulted in eigenvalues that ranged between 1.00218 and 1.00302 with a standard deviation of .00018 [38], [37].

Part two of the MCNP validation concerned the placement of a void tube containing water, then air, into multiple lattice positions of a critical core. To maintain criticality after insertion of the void tube apparatus, the tester withdrew the regulating rod to maintain criticality. The experiment compared the difference between the regulating rod heights after the introduction of the void tube (1) filled with water; then (2) filled with air, to known differential regulating rod worth's. This comparison resulted in the determination of the reactivity change associated only with air [38], [37].

The difference between the void tube experiment, and the modeled experiment within MCNP is the elimination of the material properties attributed to the void. Ultimately, the model assumed that a vacuum existed in the void tube apparatus. Richardson determined two void reactivity worth's at identified locations associated with the recorded regulating rod heights required to maintain criticality for both the water filled, and air filled void tube. Richardson stated that he obtained void reactivity worth's at an initial super-critical state. Furthermore, Richardson concluded that his experiment validated the model although he expected some differences between the experimental and predicted void reactivity worth's due to: (1) unaccounted fuel burnup; (2) detailed temperature distributions; and (3) assumption of constant differential regulating rod worth's after core re-configuration. Moreover, he advised that it may be desirable to obtain recalibrated differential regulating rod worth's at each location with the void tube inserted. Nevertheless, Richardson concluded that the void effects experiment validated the MCNP model [38], [37]. Regardless of the foregoing issues concerning the void effects on reactivity experiment, the previously discussed approach to criticality, axial flux profile measurement, and temperature effects on reactivity experiments provide validation of the MSTR MCNP model.

6.2.3. Serpent 2 Model Development and Validation Results. Although the geometric and material definitions (Table 6.3) between the MCNP and Serpent 2 models are identical, claiming that the two models will produce similar results is unsubstantiated. To prove the validity of the Serpent 2 model, the results of core flux profiles, criticality eigenvalues, and flux energy spectrums obtained from both models will be compared using the final approach to criticality control rod height for the 120w configuration. Ultimately, the validation of the Serpent 2 model to the MCNP model allows the previous experimental MCNP validation efforts led by Brad Richardson to be used as support for the proposed MSTR benchmark. However, before the Serpent 2 validation results are presented, it makes sense to disclose the structure of the Serpent 2 model and although minor, the points of deviation with respect to the MCNP model.

Multi-region homogenization in Serpent 2 requires all the defined universes within the model to be contained in a base universe zero and that each specific homogenized region be of the highest possible level. The reason for the latter requirement stems from the way Serpent 2 handles multi-region homogenization. Any Monte Carlo tally that is made in a nested higher-level universe that is contained within a lower level universe will also count towards the homogenization of the lower level universe. Therefore, restructuring of the Serpent 2 universes is required to permit multi-region homogenization of the geometrically nested MSTR model (see Figure 6.13).

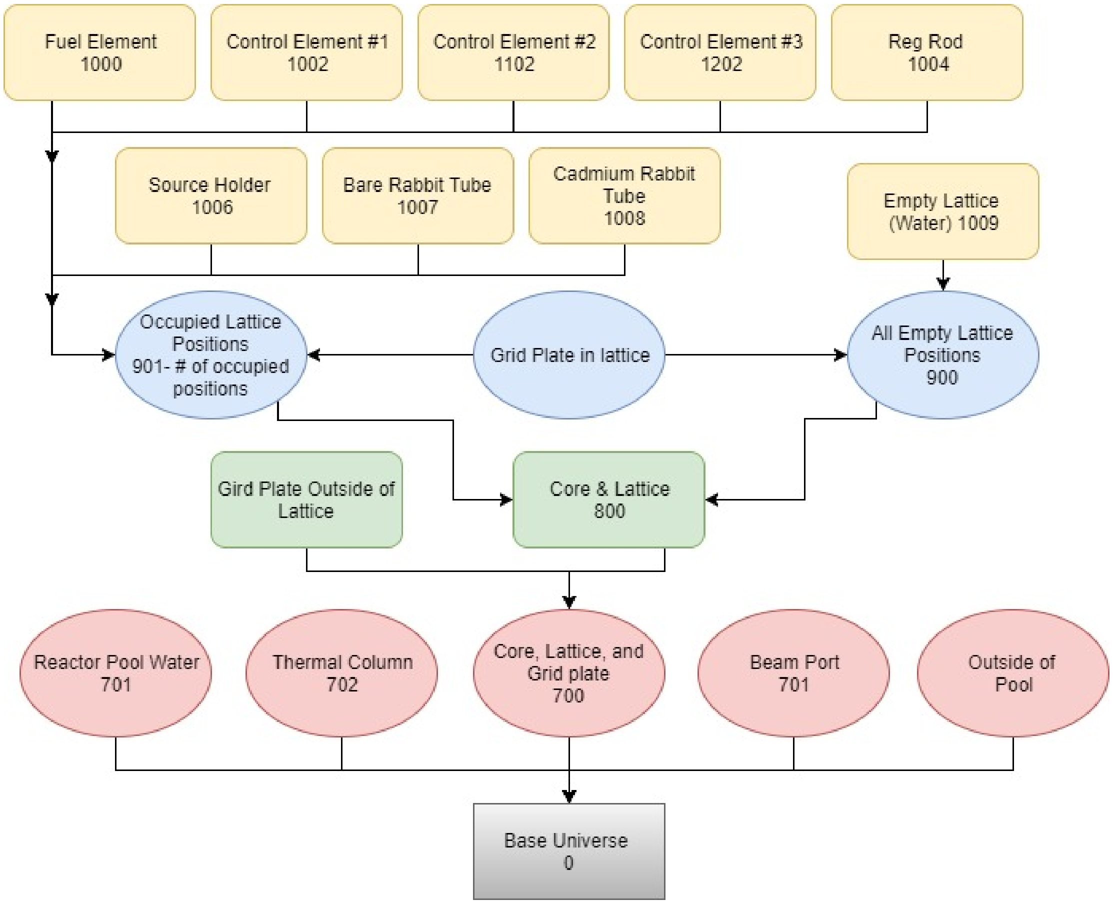
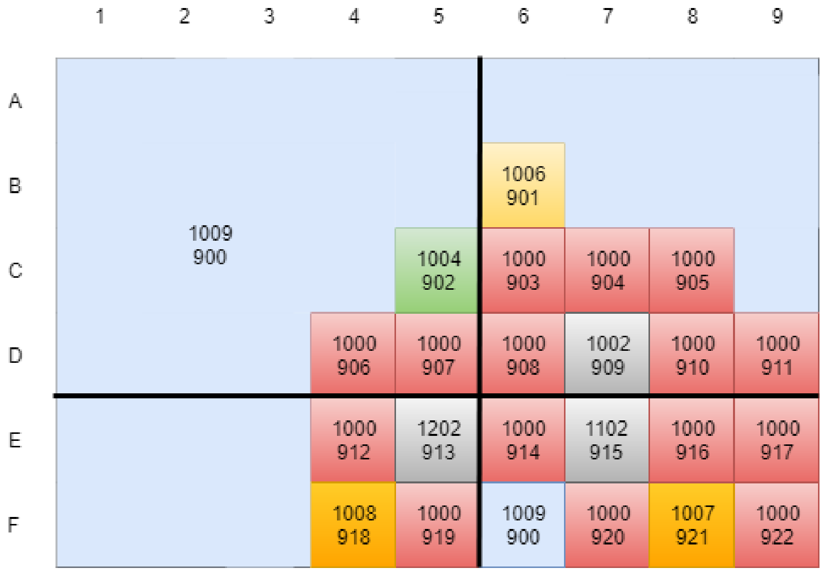


Figure 6.13 Serpent 2 MSTR model universe structure

Figure 6.14 illustrates the universe structure of the MSTR Serpent 2 model. The highest-level universes defined within the model correspond to the fuel elements (1000), control rods (1002, 1102, and 1202), regulating rod (1004), source holder (1006), and rabbit tubes (1007 and 1008) that occupy core lattice positions for a given configuration. The universe 1009 specifies any empty lattice position, which consists of only water. To generate multi-group constants (through flux weighting) that corresponds to a specific lattice position requires each occupied lattice to be defined by its own unique universe identifier; therefore, every lattice position that is occupied by either a fuel element, control rod, regulating rod, source holder, rabbit tube is defined by filling universes starting from 901 through $901 + (n - 1)$, where n = the number of occupied lattice positions. The remaining empty lattice positions are then lumped into a single universe 900.

In the axial direction, each lattice extends from the top to the bottom of the reactor pool to prepare the 2D benchmark. Likewise, the remainder of the reactor pool and experimental instruments are also collapsed. See Figure 6.14 for an illustration of the universe structure that provides lattice specific group constants for the 120W core configuration, where the bottom number in Figure 6.15 is the universe being filled by the top universe. It is imperative that users specify the universes to be homogenized by the Serpent 2 “set gcu 900 901 etc.” command in the order that the regions are defined within Gmesh (meshing software, which starts from 1), as the FE framework calls the mesh information based on the column index. For example, the empty lattice positions (universe 900) will be specified in Gmesh by region #1. Hence, all the group constants for universe 900 will be in column index 1. Since the stiffness matrix and load vector assemblers loop over all the elements in the mesh, the region in which the element belongs to will be passed as input to call the correct multi-group data for the integrals, which for this instance will be column index 1.



	1	2	3	4	5	6	7	8	9
A	1009 900								
B						1006 901			
C					1004 902	1000 903	1000 904	1000 905	
D				1000 906	1000 907	1000 908	1002 909	1000 910	1000 911
E				1000 912	1202 913	1000 914	1102 915	1000 916	1000 917
F				1008 918	1000 919	1009 900	1000 920	1007 921	1000 922

Figure 6.14 MSTR 120W core configuration universe structure

The remainder of the reactor pool and experimental apparatuses outside of the core lattice structure are defined in universes 701 and 702. Where, universe 701 contains the beam port and the reactor pool water and universe 702 contains the thermal column. The homogenized pool region is divided into universes 200 and 300 that can be filled by any combination of universes 701 and 702 depending upon how the user wishes to homogenize the core. This implies that users can change the bounding surfaces that define the cells of the homogenized universe without changing the reactor pool and instrumentation cells for the universes 701 and 702.

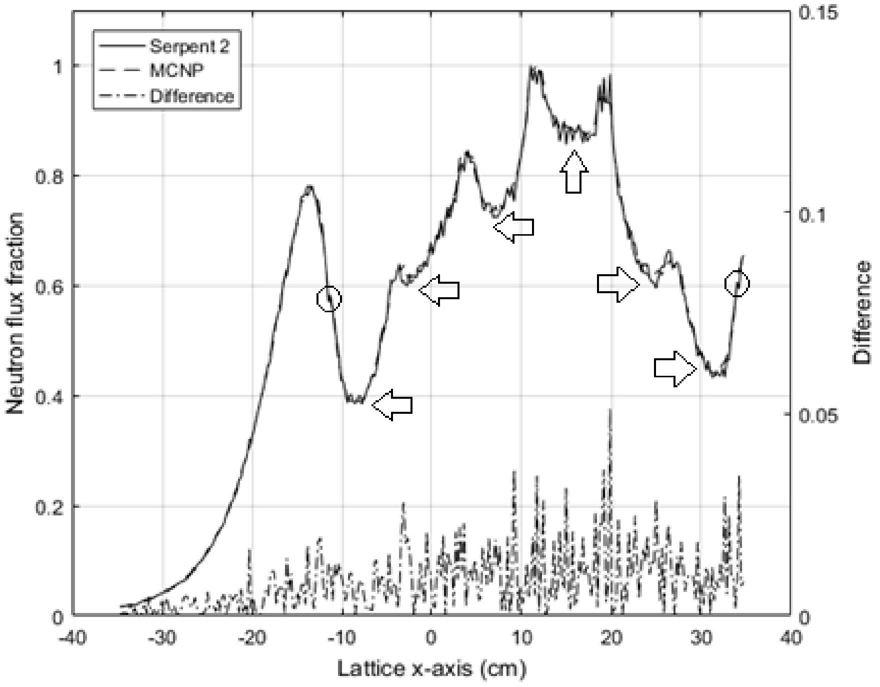
Comparison of the eigenvalues, two-group flux mesh tallies along the lattice dividers of row E/D and column 5/6 (black lines in Figure 6.14); in addition to the 70-group neutron energy spectrum within the element D5 and D9, for 375E+6 neutron histories provide validation of the MSTR Serpent 2 model. If the spatial flux profile solutions between the two models diverge, the mesh size should be small enough to capture these differences. Thus, the mesh tally bin increments for the traverse along the row divider E/D: 300 bins of $\Delta x = 0.232$ cm; one bin of $\Delta y = 0.60$ cm; and one bin of $\Delta z = 1.0$ cm. For the traverse along column divider 5/6: one bin of $\Delta x = 0.60$ cm; 182 bins of $\Delta y = 0.26535$ cm; and one bin of $\Delta z = 1.0$ cm.

The eigenvalues obtained for the 120w critical control rod height configuration were $k_{eff}(MCNP) = 0.99976 \pm 0.00004$ and $k_{eff}(Serpent\ 2) = 0.99959 \pm 0.00006$. Thus, the difference in reactivity between the two models is $\Delta\rho = 17.4 \pm 7.54$ pcm. Table 6.5 contains the tally error statistics for the flux traverses and element energy spectrums where the top values are the thermal group statistics and the bottom numbers are the fast group statistics. The tally bin errors obtained from Serpent 2 are significantly greater than the bin errors obtained from MCNP which indicates that more particle histories are required to produce tally statistics that coincide with MCNP. The increase in variance of the fast group statistics when compared to the thermal statistics is a result of the tally bins extending well into the water reflector where the probability of a fast neutron contributing to those tallies on a consistent basis is low. Thus, the only way to improve those statistics is to increase the sample population of each batch.

Table 6.5 Serpent 2 validation tally statistics

Tally	Position	Avg. Difference (Serpent - MCNP)	Avg. Serpent 2 bin error	Avg. MCNP bin error
Row D/E	mid-plane	0.0085 ± 0.0079 0.0092 ± 0.0093	0.0233 ± 0.0128 0.0391 ± 0.0505	0.0109 ± 0.0080 0.0144 ± 0.0196
Row D/E	top-plane	0.0230 ±0.0233 0.0139 ± 0.0137	0.0403 ± 0.0193 0.0678 ± 0.0854	0.0185 ± 0.0123 0.0245 ± 0.0307
Row D/E	bottom-plane	0.0378 ± 0.0272 0.0128 ± 0.0133	0.0344 ± 0.0181 0.0610 ± 0.0772	0.0161 ±0.0118 0.0220 ± 0.0284
Column 5/6	mid-plane	0.0151 ± 0.0120 0.0122 ± 0.0117	0.0216 ± 0.0070 0.0247 ± 0.0153	0.0068 ± 0.0022 0.0085 ± 0.0071
Column 5/6	top-plane	0.0242 ± 0.0226 0.0180 ± 0.0182	0.0380 ± 0.0127 0.0447 ± 0.0247	0.0118 ± 0.0035 0.0152 ± 0.0120
Column 5/6	bottom-plane	0.0250 ± 0.0224 0.0150 ± 0.0155	0.0320 ± 0.0096 0.0393 ± 0.0228	0.0101 ± 0.0034 0.0135 ± 0.0108
D5	Element	5.31E-4 ± 7.13E-4	0.0012 ± 8.63E-4	8.43E-4 ± 5.81E-4
D9	Element	6.77E-4 ± 9.41E-4	0.0015 ± 0.0010	9.94E-4 ± 7.03E-4

Figure 6.15 is the Serpent 2 and MCNP flux traverse comparison along the row divider E/D at the core mid-plane (0 cm) for two energy groups split at 0.625 eV. The flux profile suppressions (accented by the arrows) in the range of -10 cm to 35 cm are the result of mesh tallies crossing into the top curved fuel plate of each element. It is also evident that strong thermal flux perturbations are present at the fuel/water reflector interface (interface is circled in Figure 6.15) due to the increased neutron thermalization. The average absolute difference between the Serpent 2 and MCNP bin tallies was 0.0085 ± 0.0079 flux fraction. Regarding the flux group bounded from 0.625 eV to 20 MeV (bottom plot in Figure 6.15) the average difference in flux fraction was 0.0092 ± 0.0093. This implies that the thermal and fast spatial flux profiles from the two models agree at the core mid-plane.



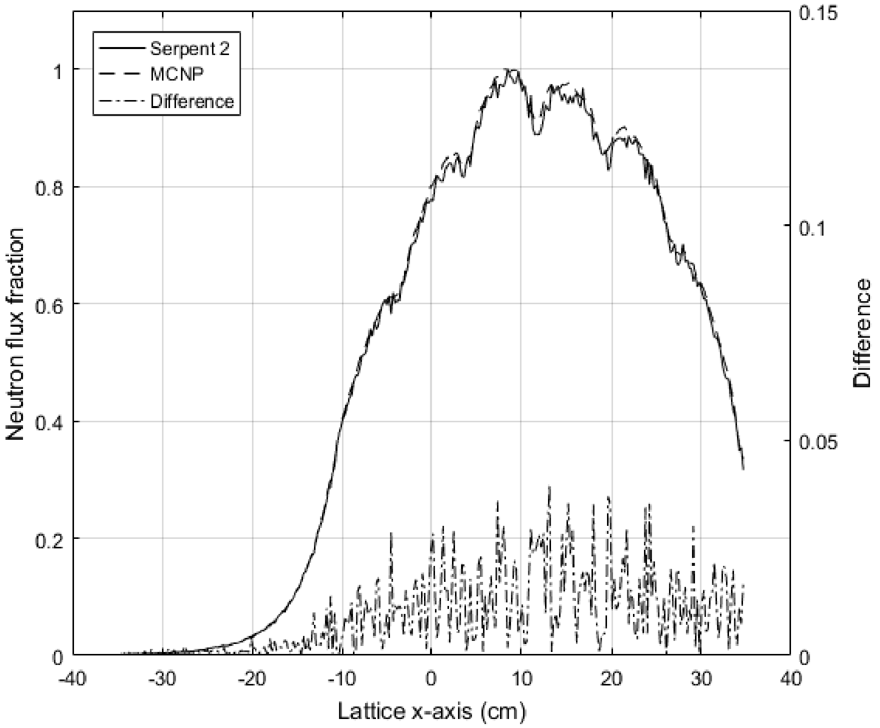
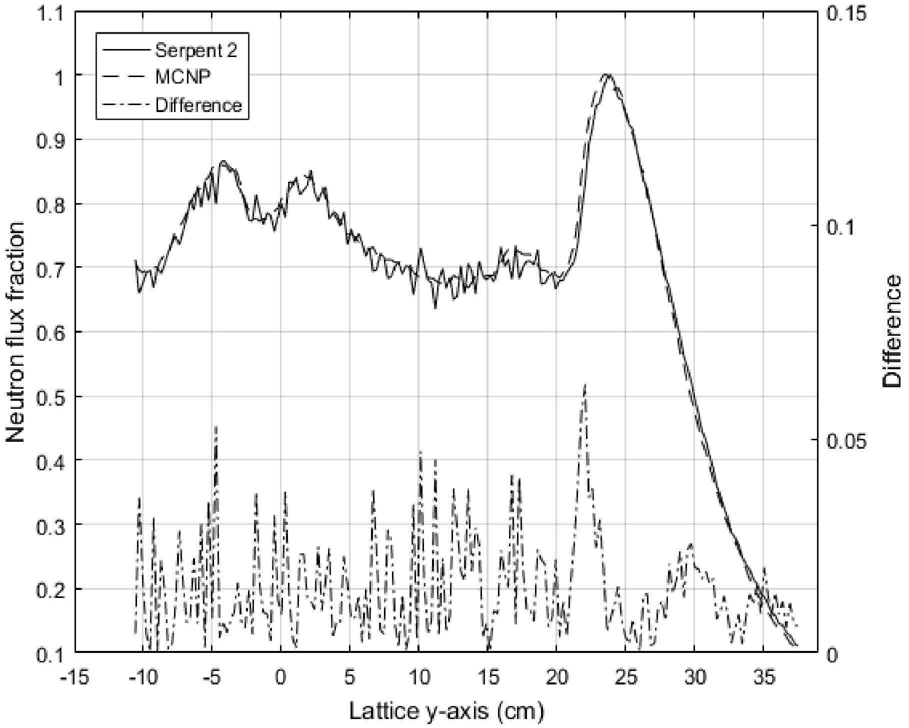


Figure 6.15 Core mid-plane flux traverse along lattice row D/E (top) thermal, (bottom) fast

The Serpent 2 results of the thermal spatial flux profile along the column divider 5/6 at the core mid-plane (Figure 6.16) oscillate between the mesh bins within the active region of the core and get larger for the top and bottom planes (see graphs in Appendix C). However, these fluctuations are nonexistent, or suppressed in the water reflector. This is due to the increased fast flux thermalization in the reflector which provides consistent thermal tallies over all the batches. Likewise, these fluctuations are also absent in the spatial flux profiles obtained from MCNP. Further comparison of the spatial flux profiles along the lattice divider column 5/6 and row D/E indicates that the Serpent 2 flux profiles along the row D/E are more consistent with the results from MCNP. This is also realized when comparing the average difference between the Serpent 2 and MCNP bin tallies in Table 6.5.

Neutrons traveling perpendicular to the fuel plates (y-axis) must travel through more water and high absorption fuel plates than the direction parallel to the fuel plate (x-axis direction). Thus, the total mean free paths a neutron with energy *E* must travel to leak from the system in the parallel direction is lesser than the perpendicular direction. Consequently, less particles per batch are reaching the bins along the column divider 5/6 within the active core when compared to the row divider D/E on a consistent basis. Therefore, increasing the batch sample population in Serpent 2 will decrease the tally variance and provide smooth spatial flux profiles along the column divider 5/6. Nevertheless, the spatial flux profiles along the column divider 5/6 for both the thermal and fast group follow the general trend of the spatial profile obtained from MCNP.



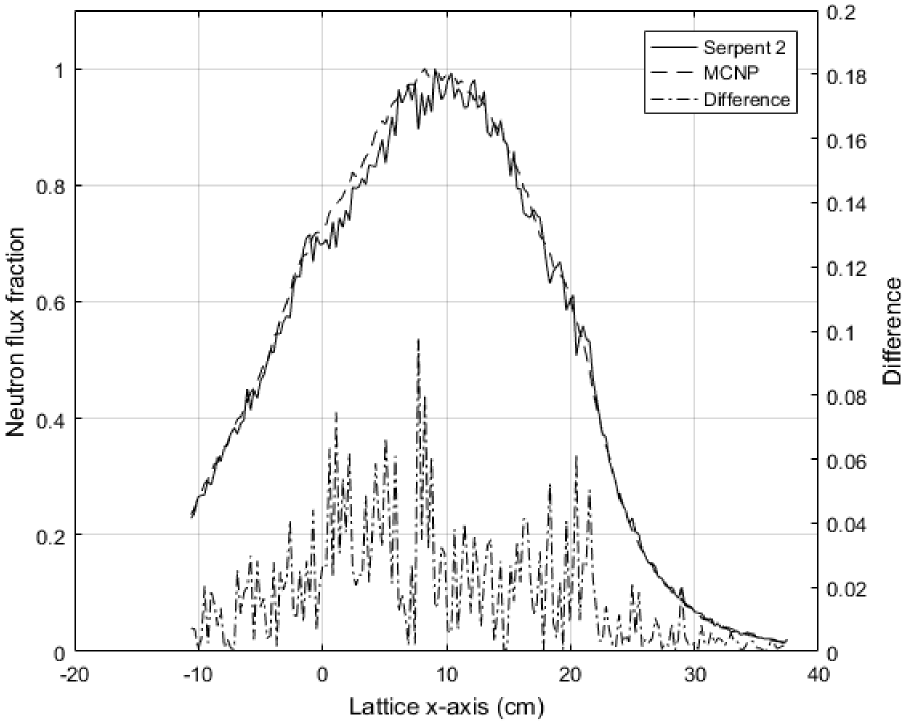


Figure 6.16 Core mid-plane flux traverse along lattice column 5/6 (top) thermal, (bottom) fast

Comparison of the thermal spatial flux profile along the row divider D/E for the top-core and bottom-core planes is provided in Figure 6.17. The effect of the control rods can be seen in thermal spatial flux profile of the top-plane (top plot of Figure 6.17) for the positions in between lattice D7 and E7 (15 cm x-axis coordinate). Ultimately, the inclusion of control rods results in steep flux gradients in the inner-core region such that the flux oscillates from 1.0 to 0.65 flux fraction three times over the span of 30cm. Regarding the difference in the spatial solution between the models, the average bin difference at the top plane was 0.0230 ± .0233 flux fraction; however, it should be noted that two tally bins had differences greater than 0.15 flux fraction.

Of all the spatial flux profiles, the traverse along the row D/E at the bottom-core plane (bottom plot Figure 6.17) had the largest tally differences between the two models. Where, on average the two models differed by 0.0378 ± 0.0272 flux fraction. Ultimately, the difference between the two models comes from the magnitude of the spatial profile as the overall spatial shape is consistent between the models. It is possible this the error is due to an insufficient sample population size, or bad tallies from a non-converged fission source; however, further investigation is needed before a concrete reason for this discrepancy can be given. To support the validation of the MSTR Serpent 2 model; a 70-group energy spectrum tally (Figure 6.18) is obtained for fuel elements in lattice locations D5 (top plot) and D9 (bottom plot). The element locations were chosen to provide spectrums from various locations in the core. Where, element D5 is surrounded by three fuel elements and a regulating rod element and element D9 is surrounded by two fuel elements and the water reflector.

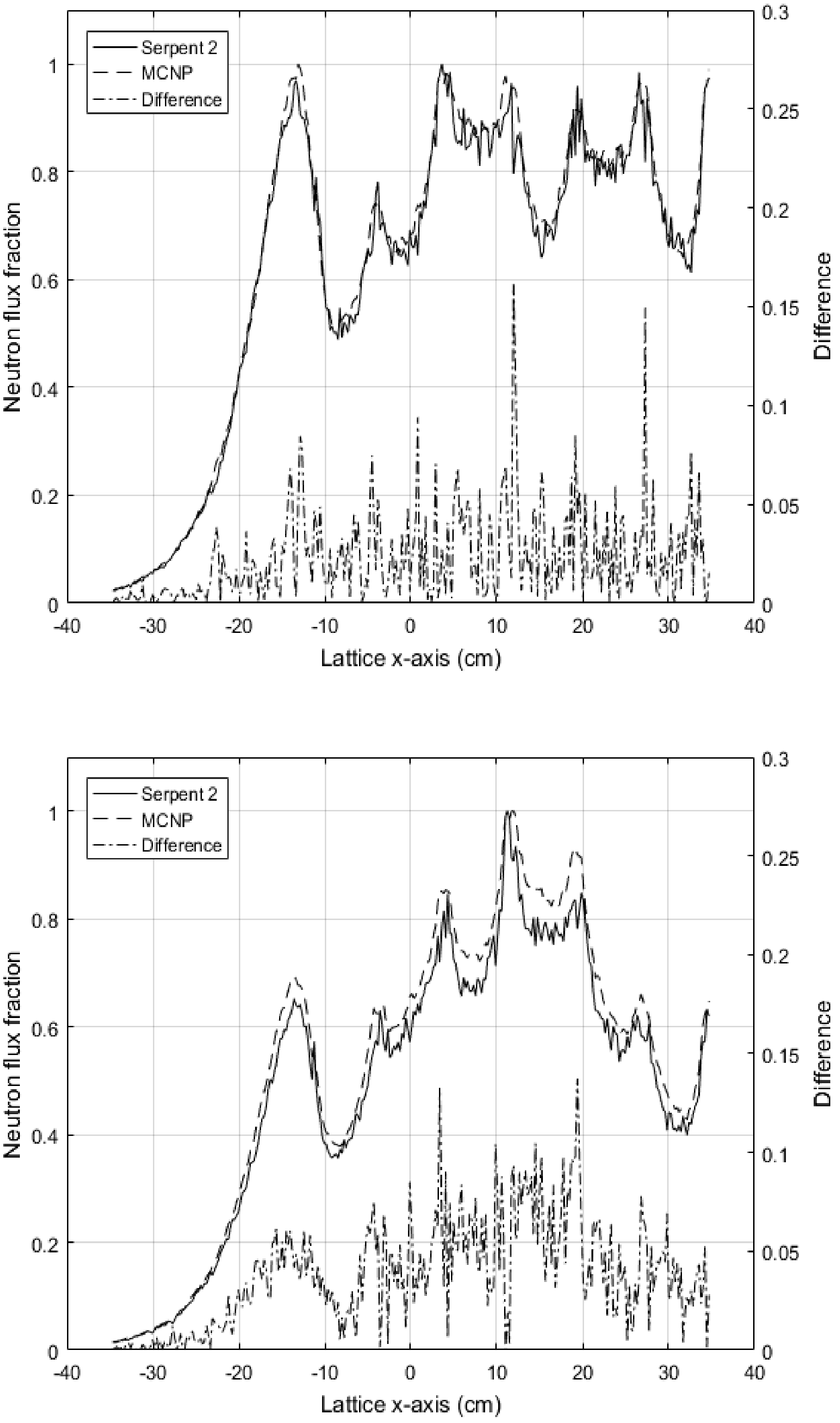


Figure 6.17 Serpent 2 spatial flux profile comparison for the lattice divider along row D/E: (top) top-core plane, (bottom) bottom-core plane

The energy bin structure for the tally follows the default 70-group intermediate energy structure that Serpent 2 uses to assemble spatial homogenization tallies before collapsing them into the few-group structure. Essentially, the energy spectrum, in addition to the spatial flux distribution, ensures that the interaction rates between the Serpent 2 and MCNP model are consistent. The average difference between the Serpent 2 and MCNP model for the energy spectrum in lattice D5 across all bins is 5.3084E-04 ± 7.21275E-04 flux fraction and 6.7746E-04 ± 9.4088E-04 flux fraction for the element in lattice location D9. Therefore, the results of the energy spectrum tallies for both the Serpent 2 and MCNP models suggest that the energy spectrums in these elements are equivalent.

6.2.4. 2-D Benchmark Description. The 2-D MSTR benchmark is prepared through the spatial homogenization of the validated 120w core configuration Serpent 2 MSTR model. Because spatial homogenization routines in Serpent 2 rely upon analog estimators; inclusion of the entire reactor pool results in prohibitively large sample population sizes necessary to obtain tallies throughout the reactor pool. To limit the sample population size, the boundary of the full core model is reduced such that the thickness of the water reflector outside of the core is considered an infinite reflector. Therefore, the total reactivity of the system may be reduced by limitation of the boundary; however, the expected deviation in total reactivity compared to the full reactor pool model should not amount to more than a few pcm. Most of the neutron leakage will occur at the right side of the geometry due to the beam port being located close to the geometric boundary.

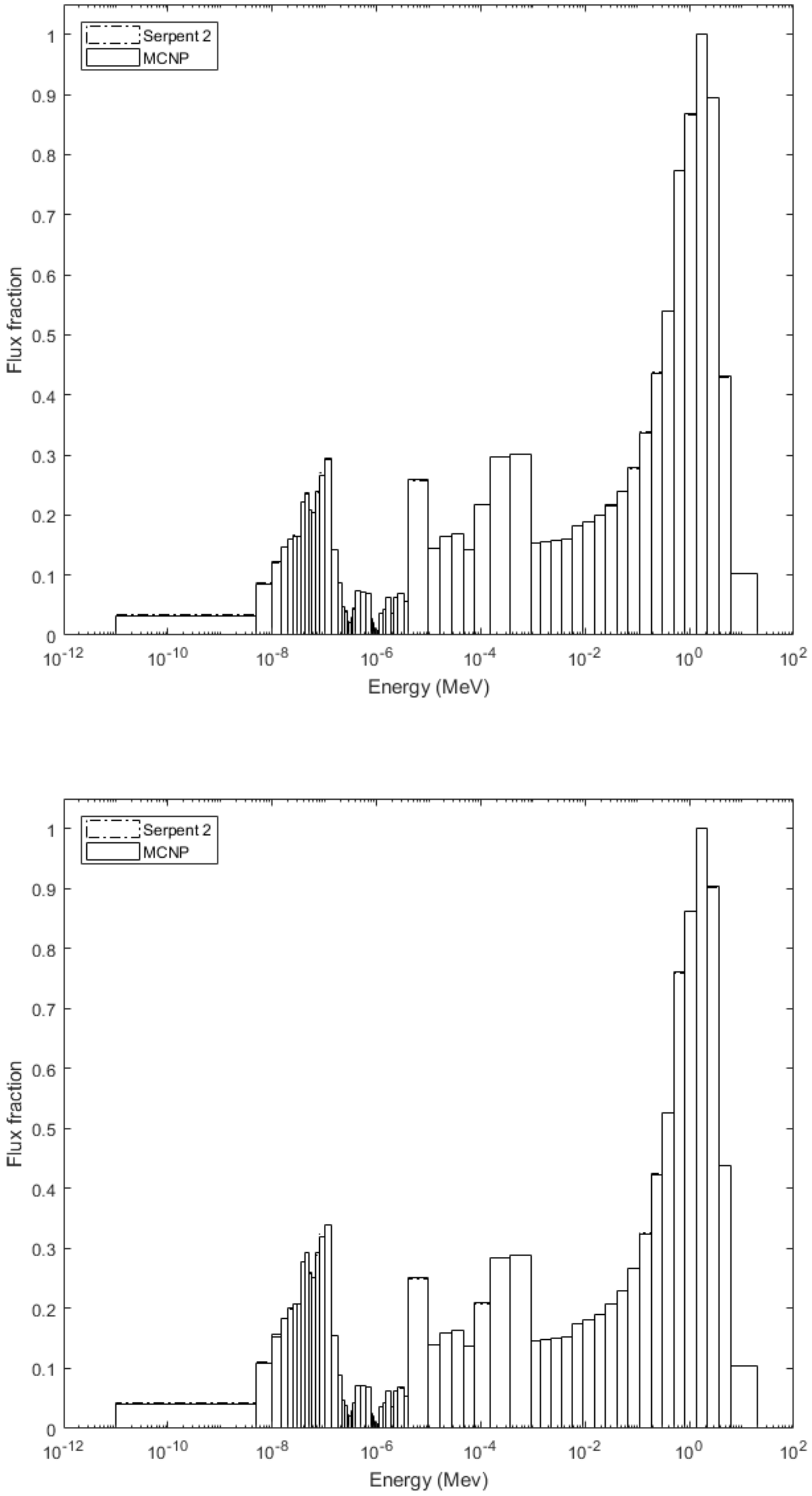


Figure 6.18 70-group neutron flux in elements (top) D5 and (bottom) D9

As mentioned in the previous section, each occupied core lattice is spatially homogenized to provide their corresponding multi-group constants. Outside of the grid plate, the reactor pool is split into two regions that are divided at the front face of the thermal column gamma shield (see Figure 6.19). The region immediately outside of the grid plate contains the reactor pool water and portions of the beam port (region identifier 24, see Figure 6.19). The other region contains the entire thermal column, its surrounding reactor pool water, and portions of the beam port (region identifier 25, see Figure 6.19). The computational mesh corresponding to the defined spatial homogenization regions is prepared with the frontal algorithm in GMESH using a hybrid triangular mesh in which 100 Lloyd smoothing steps are applied. The foregoing results in the improvement of the anisotropy of the unstructured mesh regions. The core lattice within the grid plate is structured to reduce the accumulation of error by the introduction of degenerate elements in this region, whereas the unstructured region contains the geometry outside of the grid plate (region identifiers 24 and 25, see Figure 6.19).

The 2-D MSTR benchmark is prepared for two cases. The first being, no hydrogen transport correction curve; and the second including a hydrogen transport correction curve. The foregoing permits error quantification associated with the out-scatter approximation used in Monte Carlo codes to calculate diffusion coefficients. For each case, the benchmark is conducted with scattering matrices of the zeroth order and up to the third order to demonstrate the effect of anisotropic scattering. The reason for this is that the P_0 matrices tend to under estimate the criticality and the P_1 terms overestimate the criticality eigenvalue [44]. The two tables that contain the multi-group constants prepared by Serpent 2 from 375E+6 neutron histories for the SP_1 and SP_3 transport approximations are in Appendix D. The region number in the tables in Appendix D references to the region number in Figure 6.19. Each table in Appendix D contains both the zeroth and third order scattering cross sections denoted by $\Sigma_{s,order}$. Since preparation of the benchmark results in two energy groups divided at 0.625 eV, the thermal group chi is zero and therefore omitted from the tables.

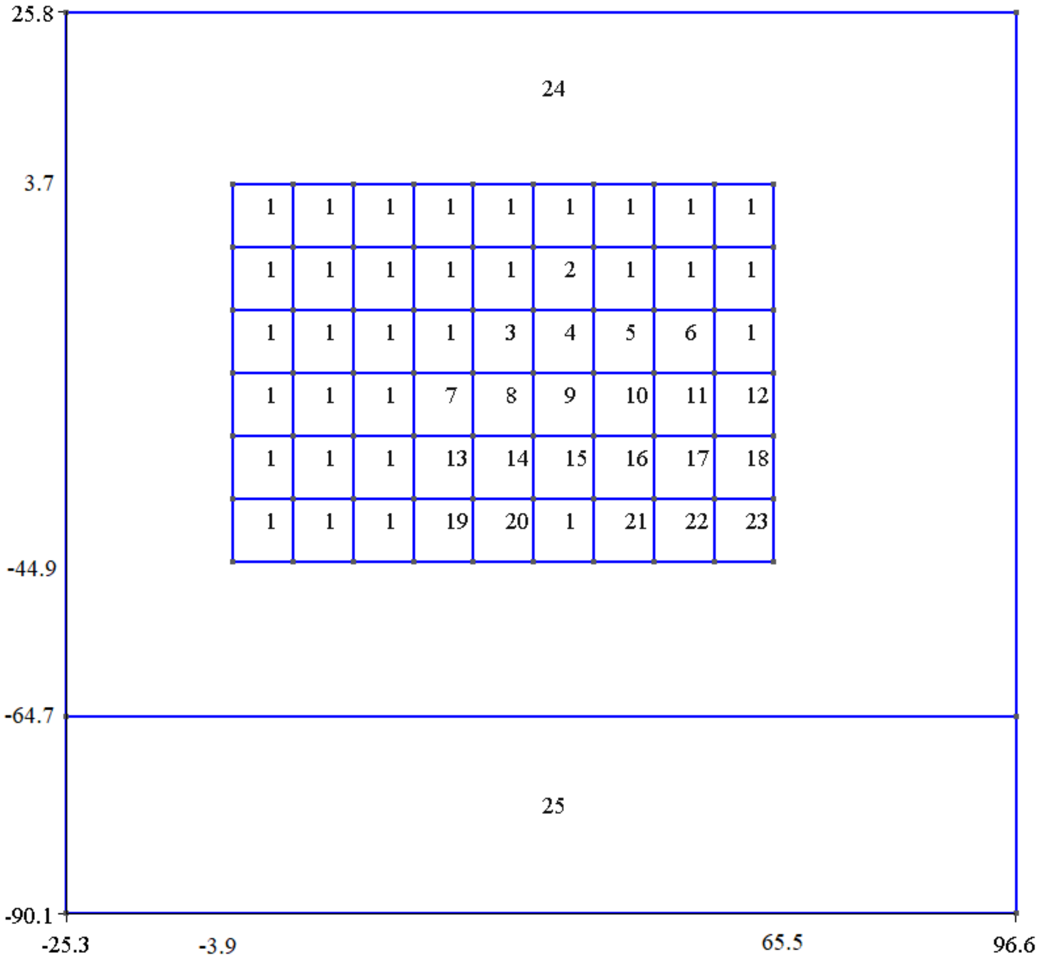


Figure 6.19 MSTR 2D benchmark geometry

6.2.5. Benchmark Results. The 2-D MSTR benchmark was conducted with triangular mesh sizes of $\Delta_{x,y} = 3,\ 2,\ 1,$ and 0.50 cm. Table 6.6 presents the eigenvalue results of both SP_1 and SP_3 approximations in which: (1) no hydrogen transport correction curve is applied; and (2) only the P_0 scattering matrices are considered. The converged eigenvalues for the SP_1 and SP_3 approximations concerning this case were 0.96525 and 0.97387 respectively. Note that none of the linear element cases were able to converge. In regard to the quadratic cases, the mesh size in which the eigenvalue converged was $\Delta_{x,y} = 0.5$ cm. Likewise, the first cubic case that converged was $\Delta_{x,y} = 2$ cm. Application of the SP_3 approximation over the SP_1/diffusion approximation improved the converged eigenvalue result by 885 pcm. However, when considering the reference Serpent 2 eigenvalue of 0.999481 ± 0.00007 the total difference in reactivity was -2629.46 pcm.

Table 6.6 2-D MSTR benchmark eigenvalue results: P0 scattering matrices and no Hydrogen transport correction curve

Element shape	Order	$\Delta_{x,y}$(cm)	SP_1 k$_{\text{eff}}$	SP_1 Δρ (pcm)	Total unknowns	SP_3 k$_{\text{eff}}$	SP_3 Δρ (pcm)	Total unknowns
Triangular	Linear	3 x 3	0.969748	-3066.07	4,306	0.978006	-2195.77	8,612
-	-	2 x 2	0.967446	-3311.28	9,462	0.975883	-2418.13	18,924
-	-	1 x 1	0.965898	-3476.89	35,040	0.974458	-2567.91	70,080
-	-	0.5 x 0.5	0.965429	-3527.15	133,966	0.974119	-2603.54	267,932
-	Quadratic	3 x 3	0.965533	-3516.02	16,894	0.973931	-2623.44	33,788
-	-	2 x 2	0.965327	-3538.05	37,362	0.973931	-2623.44	26,788
-	-	1 x 1	0.965262	-3545.07	139,198	0.973876	-2629.14	278,396
-	-	0.5 x 0.5	0.965255	-3545.81	533,954	0.973873	-2629.47	1,067,908
-	Cubic	3 x 3	0.965267	-3544.55	37,766	0.973884	-2628.36	75,532
-	-	2 x 2	0.965257	-3545.58	83,702	0.973875	-2629.27	167,404
-	-	1 x 1	0.965254	-3545.86	312,476	0.973873	-2629.46	624,952

Table 6.7 reflects the results of the 2-D MSTR benchmark concerning the case in which there is: (1) no Hydrogen transport correction curve; and (2) the scattering macroscopic cross sections now include up to the P_3 order. The results of Table 6.7, reveal convergence as previously discussed in Table 6.6. The converged eigenvalues for both SP_1 and SP_3 approximations concerning the case presented in Table 6.7 were 0.98302 and 0.99176 respectively. Furthermore, the inclusion of scattering matrices up to the P_3 order further improved the converged eigenvalues over the P_0 case by 1807.69 and 1803.86 pcm respectively. Moreover, with respect to the Serpent eigenvalue; the SP_3 approximation that includes up to the P_3 order further reduced the total reactivity difference to -777.620 pcm.

Table 6.7 2-D MSTR benchmark eigenvalue results: up to P3 scattering matrices and no Hydrogen transport correction curve

Element shape	Order	$\Delta_{x,y}$(cm)	SP_1 $\mathrm{k_{eff}}$	SP_1 $\Delta\rho$ (pcm)	Total unknowns	SP_3 $\mathrm{k_{eff}}$	SP_3 $\Delta\rho$ (pcm)	Total unknowns
Triangular	Linear	3 x 3	0.987473	-1216.03	4,306	0.995842	-365.38	8,612
-	-	2 x 2	0.985194	-1450.17	9,462	0.993749	-576.80	18,924
-	-	1 x 1	0.983663	-1608.09	35,040	0.992346	-719.05	70,080
-	-	0.5 x 0.5	0.983198	-1656.08	133,966	0.991924	-761.87	267,932
-	Quadratic	3 x 3	0.983307	-1644.85	16,894	0.992018	-752.32	33,788
-	-	2 x 2	0.983099	-1666.35	37,362	0.991827	-771.74	26,788
-	-	1 x 1	0.983033	-1673.20	139,198	0.991772	-777.29	278,396
-	-	0.5 x 0.5	0.983026	-1673.91	533,954	0.991769	-777.61	1,067,908
-	Cubic	3 x 3	0.983038	-1672.68	37,766	0.991780	-776.52	75,532
-	-	2 x 2	0.983028	-1673.68	83,702	0.991771	-777.41	167,404
-	-	1 x 1	0.983026	-1673.96	312,476	0.991769	-777.60	624,952

Preparation of the 2-D MSTR benchmark with the in-scatter Hydrogen transport correction curve [33] to correct the macroscopic transport cross section for the out-scatter approximation results in an average reduction to the fast group diffusion coefficients of 1.94% ± 0.74%. For the thermal group, the average increase to the diffusion coefficients was 9.37% ± 1.87%. Furthermore, the foregoing changes to the diffusion coefficients increased the difference in total reactivity between the referenced Serpent 2 and the FEM solution to -820.84 pcm.

Figure 6.20 reveals the thermal flux traverses along the lattice divider of row D/E for both SP_1 and SP_3 simulations as discussed in the benchmark description (section 6.2.4). Use of a fine mesh tally in Serpent 2 resulted in the referenced flux traverse solution. The tally bin in the axial direction includes the entire geometry, from the top to the bottom of the reactor pool, thereby providing the means to compare the FEM solutions. Since the lattice divider of row D/E crosses into the fuel plates within the core; increased thermal absorption macroscopic cross sections result in a local perturbation about these points as revealed within Figure 6.20. Notwithstanding the local perturbations, use of the SP_3 approximation results in an improvement in the scalar flux traverse. Furthermore, the SP_3 solution follows the general trend of the Serpent 2 reference flux traverse.

Figure 6.21 is the fast group scalar flux traverse along the lattice divider of row D/E. The fast scalar flux results indicate minimal improvement of the SP_1 and the SP_3 solutions. Both SP_1 and SP_3 solutions have nearly identical maximum points at approximately 38 cm. Furthermore, minimal deviations concerning the Serpent solution occur immediately outside of the core within the water reflector and the locations that correspond to the corners of the fuel element. The peaks between the corners of the fuel elements correspond to the absorption of thermal neutrons in Figure 6.20. Inasmuch as those thermal neutrons result in nuclear fission thereby give rise to the production of fast neutrons.

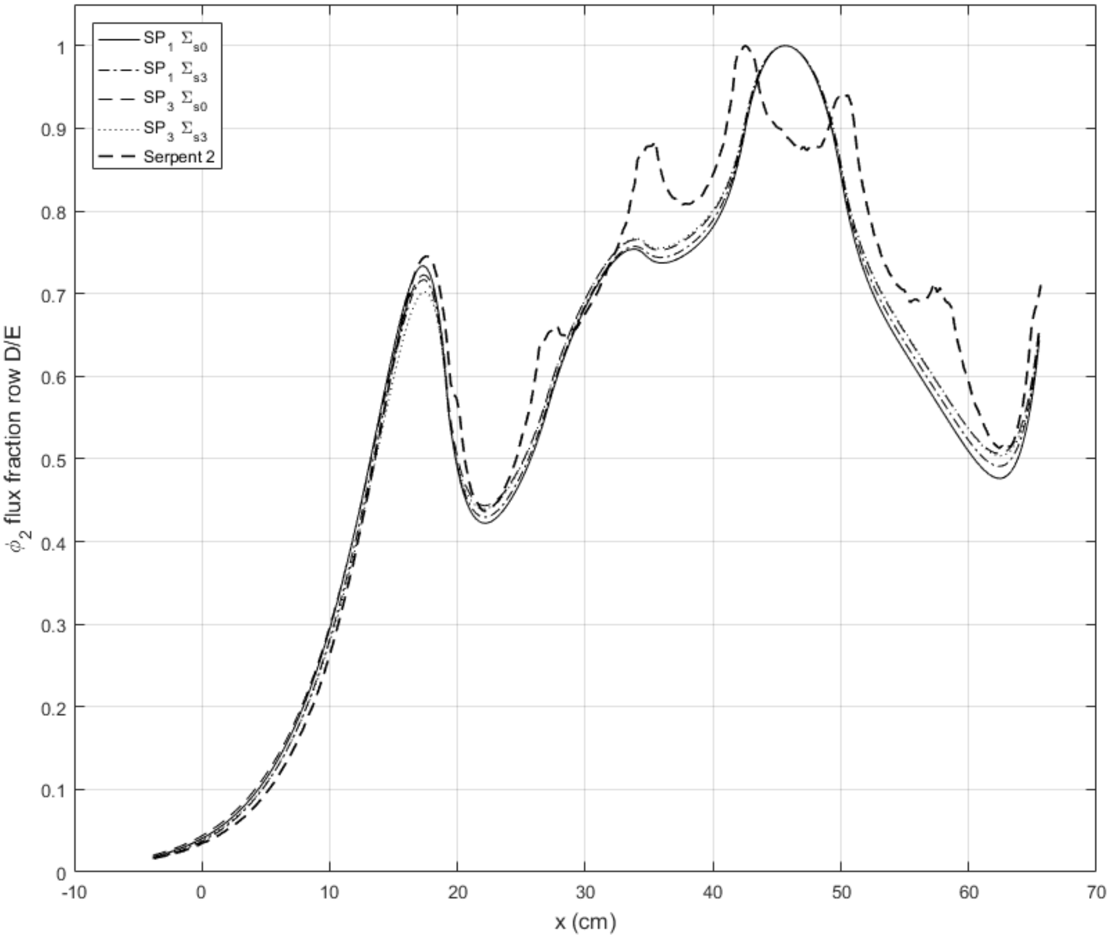


Figure 6.20 2-D MSTR Benchmark: thermal group scalar flux traverses along the lattice divider of row D/E

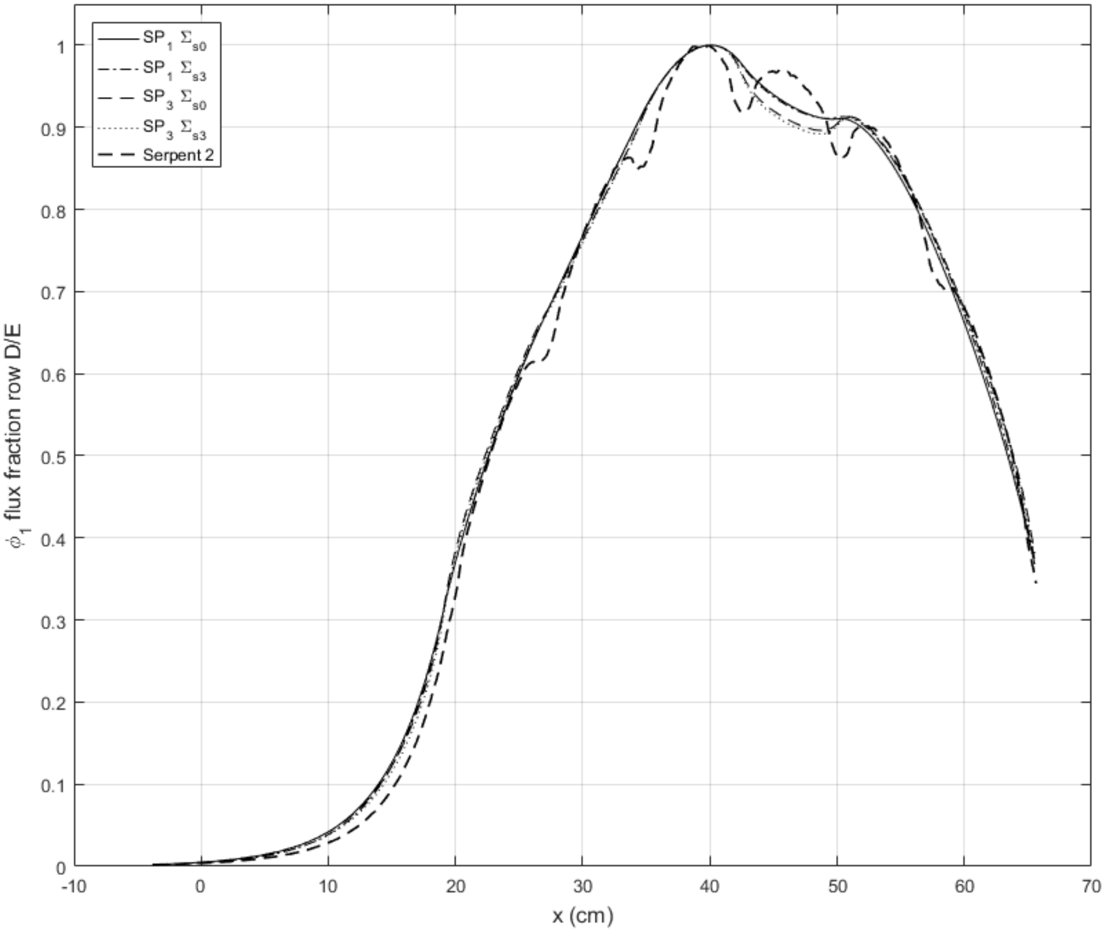


Figure 6.21 2-D MSTR Benchmark: fast group scalar flux traverses along the lattice divider of row D/E

Figure 6.22 demonstrates the thermal flux traverse along the lattice divider 5/6. Application of the SP_3 approximation yields a significant better thermal flux solution as compared to the SP_1 approximation. However, there remains a significant deviation when one compares the Serpent 2 to the best SP_3 solution. It should be noted that validation of the Serpent 2 MSTR model discussed in Section 6.2.3, indicates that anisotropies exist in this direction. Since the SP_3 approximation includes higher-order moments, increased resolution of the flux solution occurs. Furthermore, inclusion of the scattering matrices up to the P_3 order revealed better solutions in both the SP_1 and SP_3 approximations. Moreover, the foregoing provides support for the existence of anisotropies in this direction.

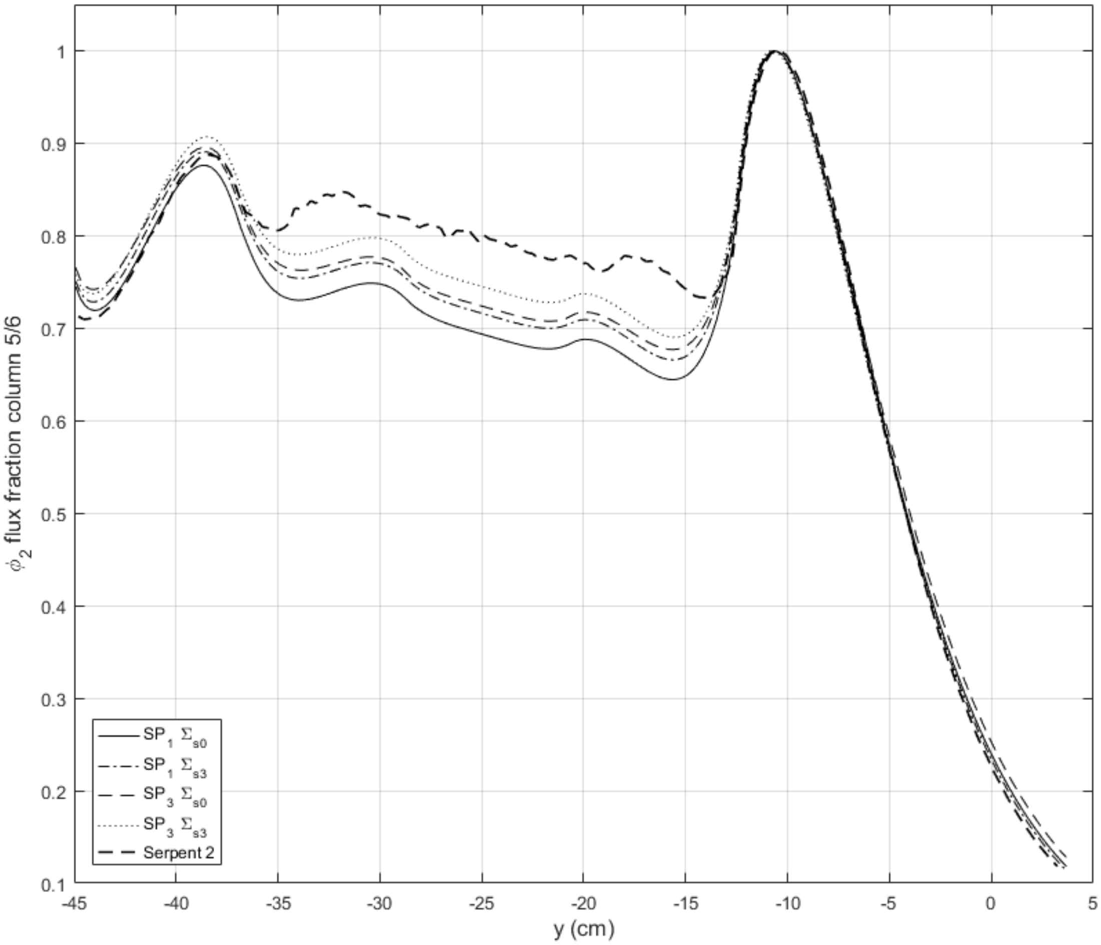


Figure 6.22 2-D MSTR Benchmark: thermal group scalar flux traverses along the lattice divider of column 5/6

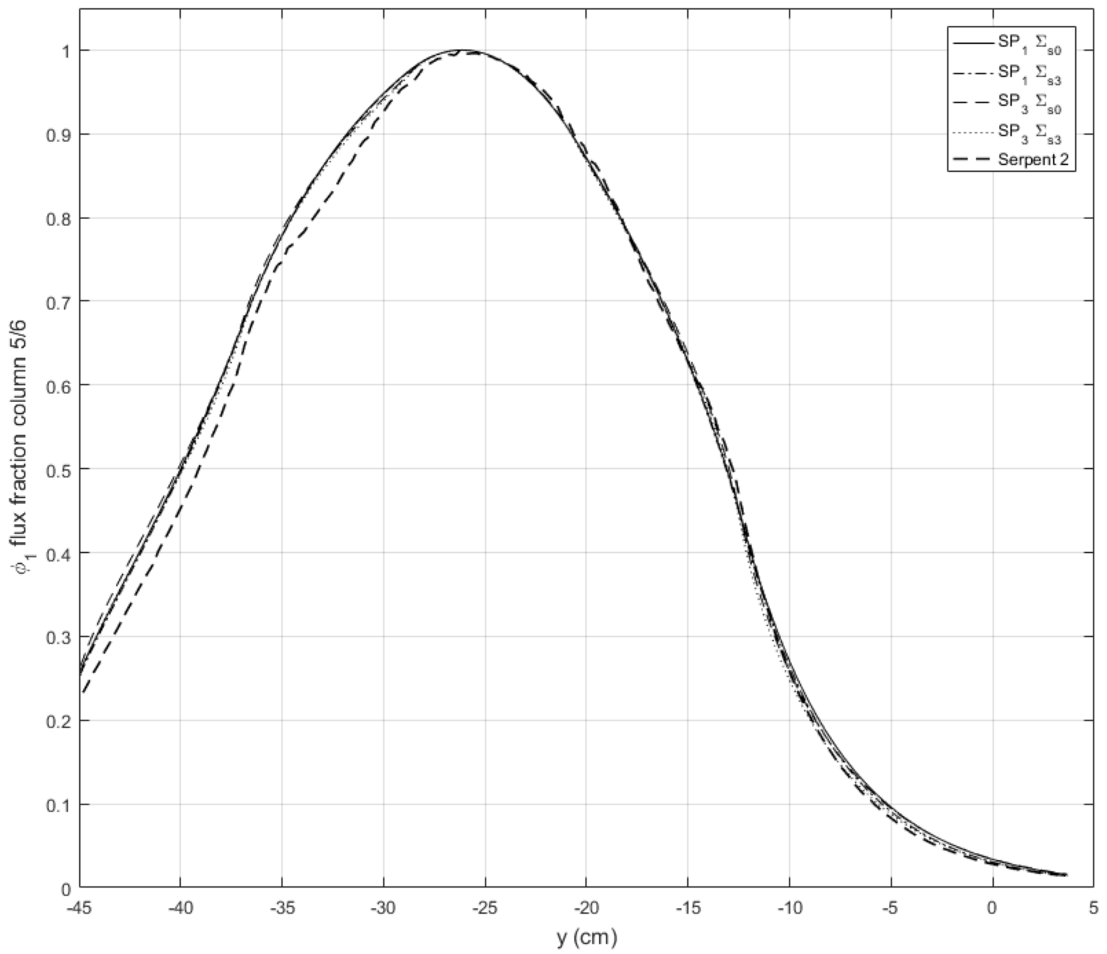


Figure 6.23 2-D MSTR Benchmark: fast group scalar flux traverses along the lattice divider of column 5/6

For the most part, Figure 6.23 indicates that the fast flux solutions along the lattice divider of column 5/6 reveal virtually identical values when evaluating the SP_1, SP_3, and Serpent 2 flux traverses; However, an approximate 3% deviation in the flux fraction exists between the SP_1/SP_3 and Serpent 2 reference solution as the flux approaches the rear side of the grid plate in the direction of the thermal column. No significant deviations exist between the SP_1 and SP_3 solutions.

Construction of a relative thermal flux error map (Figure 6.24) through interpolation of the simulation results of both Serpent 2 (see Appendix E for the Serpent 2 tally relative error map) and the quadratic $\Delta_{x,y}= 0.5$ FEM SP_3 approximation with scattering matrices up to the P_3 order on a structured grid of mesh size $\Delta_{x,y}= 0.1$ cm provides the most accurate means to check the spatial deviations of the thermal flux distribution. Due to spatial homogenization of the cross sections, large deviations will appear in regions with high localized absorbers such as the control rods and the cadmium rabbit tube. Therefore, application of the relative thermal error flux map should be limited to the regions away from the high localized absorbers. In consideration of the foregoing, the relative thermal flux errors are consistently 10% over the entire core lattice.

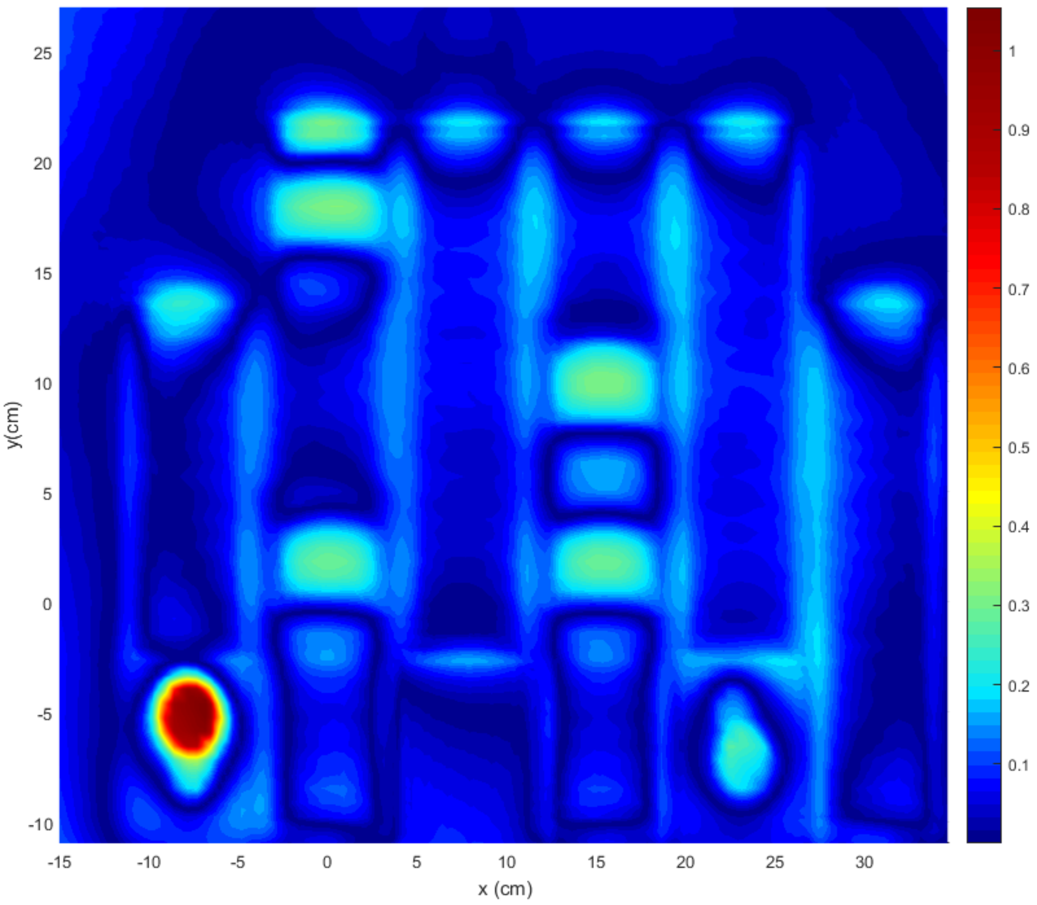


Figure 6.24 2-D MSTR benchmark: relative thermal flux error map between Serpent 2 and FEM quadratic 0.5 cm mesh

7. CONCLUSION

A finite element framework reactor analysis framework was developed so as to include the capability of approximating the multi-group neutron diffusion equation and the simplified spherical harmonics transport equations in arbitrary geometries. The preliminary 2-D IAEA PWR benchmark indicated correct implementation of the framework because convergence to the published criticality eigenvalue of 1.029585 is observed for all element shapes and orders. Additionally, verification of the thermal spatial flux convergence through the assembly averaged fluxes for the quadratic triangular element with mesh size $\Delta_{x,y}= 2$ cm was obtained. The average relative percent error concerning the referenced values over all sub-assemblies was 0.3360%.

Validation of the MSTR Serpent 2 model concerning the approach to criticality experiment for the 120w configuration resulted in agreement of the criticality eigenvalues. The criticality eigenvalues of the MCNP and Serpent 2 model was 0.99976 ± .00004 for the MCNP model and 0.99959 ± 0.00006 for the Serpent 2 model result in an absolute difference of 17.4 ± 7.54 pcm. Spatial flux traverses along the lattice divider of row D/E and column 5/6 at the top, bottom, and mid-core planes are consistent between the two models; however, the traverses for the row D/E were smoother and had less relative errors between the bins. The foregoing suggests that the flux in the MSTR is anisotropic. Furthermore, the average relative difference of the 70-group flux spectrum between the MCNP and Serpent 2 model in element D5 was $5.314\text{E}-04 \pm 7.13\text{E}-04$ flux fraction and $6.77E-4 \pm 9.41E-4$ flux fraction for the element D9. Thus, construction of a validation chain between the physical experiments, the validated MCNP

120w configuration model, and the Serpent 2 model supports the proposed 2-D MSTR benchmark.

Application of the 2-D MSTR benchmark to the neutron diffusion equation while neglecting the scattering matrices greater than the zeroth order resulted in a converged criticality eigenvalue of 0.96525. Further improvements to the criticality eigenvalue were observed when employing the SP_3 equations to the 2-D MSTR benchmark. The SP_3 simulation resulted in a criticality eigenvalue of 0.97387, for an increase in total reactivity of 885 pcm over the diffusion eigenvalue.

Inclusion of the scattering matrices up to the P_3 order resulted in an SP_3 eigenvalue of 0.99176 with a difference of total reactivity with respect to the Serpent 2 reference eigenvalue of -777.60 pcm. Since the MSTR violates the necessary assumptions to derive Fick's law, as seen through the poor eigenvalues and spatial flux solutions, the diffusion equation should not be applied for the analysis of the MSTR. The improvements in the criticality eigenvalue and the magnitude of the spatial flux solution along the lattice divider column 5/6 when considering the SP_3 equations and the scattering up to the P_3 order indicate that the flux is anisotropic and less diffusive in the direction perpendicular to the front face of the thermal column.

Application of the in-scatter Hydrogen transport correction curve to correct the out-scatter approximation resulted in an average increase of the thermal group's diffusion coefficients by 9.37% ± 1.87%. Consequently, the corrected diffusion coefficients result in a larger criticality eigenvalue deviation of -820.84 pcm as compared to the non-corrected diffusion coefficients of -777.60 pcm with respect to the referenced Serpent 2 eigenvalue. The reason for such difference in deviation is due to the in-scatter's smaller ratio of Σ_{tr}/Σ_T in the thermal energy region about 10^{-6} MeV. Thus, Serpent 2 calculates a smaller macroscopic transport cross section than the transport cross section calculated by the out-scatter approximation. Since my work did not include generation of an in-scatter Hydrogen transport correction curve, it is advisable that additional investigation concerning the preparation of a fine Hydrogen transport correction curve, as discussed in section 5.2 to resolve any potential error that may have been introduced using a correction curve from previously published data using a plot digitizer.

Although considerable improvement of the eigenvalue and spatial flux solutions are observed when applying the SP_3 equations to the 2-D MSTR benchmark, the results are not strong enough to suggest that the framework is validated in the case of the MSTR. I believe this to be so, as spatial homogenization with the full core flux solution is a highly idealized case and does not represent the typical methodologies used in practice. Furthermore, it is likely that differences greater than -777.60 pcm with respect to the reference solution will be observed with the traditional spatial homogenization methodology. Nevertheless, results of the 2-D IAEA benchmark indicate that the framework is viable so long as the flux is not strongly anisotropic. It is expected that the application of the SP_5 and SP_7 equations will result in further improvements regarding the 2-D MSTR benchmark and will result in further extension of the framework to a broader class of reactor types.

Future work in connection with this thesis can be broken into two main categories: finite element framework development; and MSTR benchmark improvements. The first matter that should be addressed is the implementation of the framework in compiled computer language (Fortran, C++, etc.). Doing so would permit use of the frameworks 3-D discretization capabilities, its application to time-dependent and large-scale problems. Application of the SP_n equations have shown that unless prohibitively small mesh sizes are employed, the low-order interpolation polynomials will not yield numerical convergence. Therefore, higher-order interpolation polynomials should be implemented to permit the use of larger mesh sizes to further reduce the computational burden in comparison to the small mesh sizes. Future FEM framework development should also include implementation of a hybrid continuous/discontinuous Galerkin formulation where DG-FEM is used at interfaces and CG-FEM is used for the remainder of the computational domain. Implementation of CG/DG-FEM in this manner would limit the increase in computational cost which stems from an increase in the degrees of freedom associated with the DG-FEM formulation.

Following the implementation of the FEM framework in a compiled computer language, a 3-D MSTR benchmark should be prepared for further framework validation. However, it is advisable to investigate the calculation of albedos in Serpent 2 for further domain reduction. Essentially, this would reduce the number of elements and nodal points in the computational mesh and reduce the possible error introduced from inclusion of regions where the tally uncertainty may be large. This investigation can initially be conducted in consideration of the 2-D of the benchmark in the frameworks current state. Since the stochastic generation of the multigroup constants using the full core flux solution in Serpent 2 permits the preparation of an MSTR benchmark, sensitivity analysis concerning the stochastic multigroup parameters should be investigated to quantify the introduction of errors associated with the uncertainty of the stochastic multigroup constants.

APPENDIX A

BASIS FUNCTIONS

Triangular Elements

Local Basis Function Partial Derivatives

$$\frac{\partial^2\psi}{\partial x^2} = \frac{\partial^2\hat{\psi}}{\partial \hat{X}^2}\frac{(y_3-y_1)^2}{J^2} + 2\frac{\partial^2\hat{\psi}}{\partial \hat{X}\partial \hat{Y}}\frac{(y_3-y_1)(y_1-y_2)}{J^2} + \frac{\partial^2\hat{\psi}}{\partial \hat{Y}^2}\frac{(y_1-y_2)^2}{J^2}.$$

$$\frac{\partial^2\psi}{\partial y^2} = \frac{\partial^2\hat{\psi}}{\partial \hat{X}^2}\frac{(x_1-x_3)^2}{J^2} + 2\frac{\partial^2\hat{\psi}}{\partial \hat{X}\partial \hat{Y}}\frac{(x_1-x_3)(x_2-x_1)}{J^2} + \frac{\partial^2\hat{\psi}}{\partial \hat{Y}^2}\frac{(x_2-x_1)^2}{J^2}.$$

$$\begin{aligned}\frac{\partial^2\psi}{\partial x\partial y} = \frac{\partial^2\hat{\psi}}{\partial \hat{X}^2}\frac{(x_1-x_3)(y_3-y_1)}{J^2} + \frac{\partial^2\hat{\psi}}{\partial \hat{X}\partial \hat{Y}}\frac{(x_1-x_3)(y_1-y_2)}{J^2} \\ + \frac{\partial^2\hat{\psi}}{\partial \hat{X}\partial \hat{Y}}\frac{(x_2-x_1)(y_3-y_1)}{J^2} + \frac{\partial^2\hat{\psi}}{\partial \hat{Y}^2}\frac{(x_2-x_1)(y_1-y_2)}{J^2}.\end{aligned}$$

Quadratic Interpolation Polynomial

$$\hat{\psi}_j = a_j\hat{X}^2 + b_j\hat{Y}^2 + c_j\hat{X}\hat{Y} + d_j\hat{Y} + e_j\hat{X} + f_j.$$

Quadratic Reference Node Coordinates

$$\hat{A}_1 = (0, 0),$$

$$\hat{A}_2 = (1, 0),$$

$$\hat{A}_3 = (0, 1),$$

$$\hat{A}_4 = (0.5, 0),$$

$$\hat{A}_5 = (0.5, 0.5),$$

$$\hat{A}_6 = (0, 0.5).$$

Quadratic Reference Basis Functions

$$\hat{\psi}_1(\hat{X}, \hat{Y}) = 2\hat{X}^2 + 2\hat{Y}^2 + 4\hat{X}\hat{Y} - 3\hat{X} - 3\hat{Y} + 1,$$

$$\hat{\psi}_2(\hat{X},\hat{Y}) = 2\hat{X}^2 - \hat{X},$$

$$\hat{\psi}_3(\hat{X},\hat{Y}) = 2\hat{Y}^2 - \hat{Y},$$

$$\hat{\psi}_4(\hat{X},\hat{Y}) = -4\hat{X}^2 - 4\hat{X}\hat{Y} + 4\hat{X},$$

$$\hat{\psi}_5(\hat{X},\hat{Y}) = 4\hat{X}\hat{Y},$$

$$\hat{\psi}_6(\hat{X},\hat{Y}) = -4\hat{Y}^2 - 4\hat{X}\hat{Y} + 4\hat{Y}.$$

Cubic Interpolation Polynomial

$$\hat{\psi}_j = a_j\hat{X}^3 + b_j\hat{Y}^3 + c_j\hat{X}^2\hat{Y} + d_j\hat{X}\hat{Y}^2 + e_j\hat{X}^2 + f_j\hat{Y}^2 + g_j\hat{X}\hat{Y} + h_j\hat{Y} + i_j\hat{X} + j_j.$$

Cubic Reference Node Coordinates

$$\hat{A}_1 = (0,0),$$

$$\hat{A}_2 = (1,0),$$

$$\hat{A}_3 = (0,1),$$

$$\hat{A}_4 = \left(\frac{1}{3},0\right),$$

$$\hat{A}_5 = \left(\frac{2}{3},0\right),$$

$$\hat{A}_6 = \left(\frac{2}{3},\frac{1}{3}\right),$$

$$\hat{A}_7 = \left(\frac{1}{3},\frac{2}{3}\right),$$

$$\hat{A}_8 = \left(0,\frac{2}{3}\right),$$

$$\hat{A}_9 = \left(0,\frac{1}{3}\right),$$

$$\hat{A}_{10} = \left(\frac{1}{3},\frac{1}{3}\right).$$

Cubic Reference Basis Functions

$$\hat{\psi}_1(\hat{X},\hat{Y}) = 1 - 5.5\hat{X} - 5.5\hat{Y} + 18\hat{X}\hat{Y} + 9\hat{X}^2 + 9\hat{Y}^2 - 13.5\hat{X}^2\hat{Y} - 13.5\hat{X}\hat{Y}^2 - 4.5\hat{X}^3 - 4.5\hat{Y}^3,$$

$$\hat{\psi}_2(\hat{X},\hat{Y}) = \hat{X} - 4.5\hat{X}^2 + 4.5\hat{X}^3,$$

$$\hat{\psi}_3(\hat{X},\hat{Y}) = \hat{Y} - 4.5\hat{Y}^2 + 4.5\hat{Y}^3,$$

$$\hat{\psi}_4(\hat{X},\hat{Y}) = 9\hat{X} - 22.5\hat{X}\hat{Y} - 22.5\hat{X}^2 + 27\hat{X}^2\hat{Y} + 13.5\hat{X}\hat{Y}^2 + 13.5\hat{X}^3,$$

$$\hat{\psi}_5(\hat{X},\hat{Y}) = -4.5\hat{X} + 4.5\hat{X}\hat{Y} + 18\hat{X}^2 - 13.5\hat{X}^2\hat{Y} - 13.5\hat{X}^3,$$

$$\hat{\psi}_6(\hat{X},\hat{Y}) = -4.5\hat{X}\hat{Y} + 13.5\hat{X}^2\hat{Y},$$

$$\hat{\psi}_7(\hat{X},\hat{Y}) = -4.5\hat{X}\hat{Y} + 13.5\hat{X}\hat{Y}^2,$$

$$\hat{\psi}_8(\hat{X},\hat{Y}) = -4.5\hat{Y} + 4.5\hat{X}\hat{Y} + 18\hat{Y}^2 - 13.5\hat{X}\hat{Y}^2 - 13.5\hat{Y}^3,$$

$$\hat{\psi}_9(\hat{X},\hat{Y}) = 9\hat{Y} - 22.5\hat{X}\hat{Y} - 22.5\hat{Y}^2 + 27\hat{X}\hat{Y}^2 + 13.5\hat{X}^2\hat{Y} + 13.5\hat{Y}^3,$$

$$\hat{\psi}_{10}(\hat{X},\hat{Y}) = 27\hat{X}\hat{Y} - 27\hat{X}^2\hat{Y} - 27\hat{X}\hat{Y}^2.$$

Tetrahedral Elements

Affine Mapping

$$\begin{pmatrix}\hat{X}\\ \hat{Y}\\ \hat{Z}\end{pmatrix} = \frac{1}{\det M}\begin{pmatrix} M_{22}M_{33} - M_{23}M_{32} & M_{13}M_{32} - M_{12}M_{33} & M_{12}M_{23} - M_{13}M_{22} \\ M_{23}M_{31} - M_{21}M_{33} & M_{11}M_{33} - M_{13}M_{31} & M_{13}M_{21} - M_{11}M_{23} \\ M_{21}M_{32} - M_{22}M_{31} & M_{12}M_{31} - M_{11}M_{32} & M_{11}M_{22} - M_{12}M_{21}\end{pmatrix} \cdot \begin{pmatrix} x - x_1 \\ y - y_1 \\ z - z_1 \end{pmatrix},$$

$$M_{11} = x_2 - x_1,$$

$$M_{12} = x_3 - x_1,$$

$$M_{13} = x_4 - x_1,$$

$$M_{21} = y_2 - y_1,$$

$$M_{22} = y_3 - y_1,$$

$$M_{23} = y_4 - y_1,$$

$$M_{31} = z_2 - z_1,$$

$$M_{32} = z_3 - z_1,$$

$$M_{33} = z_4 - z_1.$$

$$\begin{aligned}\det M = {} & (x_2 - x_1)\big((y_3 - y_1)(z_4 - z_1) - (y_4 - y_1)(z_3 - z_1)\big) \\ & + (x_3 - x_1)\big((y_4 - y_1)(z_2 - z_1) - (y_2 - y_1)(z_4 - z_1)\big) \ldots \\ & + (x_4 - x_1)\big((y_2 - y_1)(z_3 - z_1) - (y_3 - y_1)(z_2 - z_1)\big).\end{aligned}$$

Local Basis Function Partial Derivatives

$$\frac{\partial \psi}{\partial x} = \frac{\partial \hat{\psi}}{\partial \hat{X}}\frac{\partial \hat{X}}{\partial x} + \frac{\partial \hat{\psi}}{\partial \hat{Y}}\frac{\partial \hat{Y}}{\partial x} + \frac{\partial \hat{\psi}}{\partial \hat{Z}}\frac{\partial \hat{Z}}{\partial x},$$

$$\frac{\partial \psi}{\partial y} = \frac{\partial \hat{\psi}}{\partial \hat{X}}\frac{\partial \hat{X}}{\partial y} + \frac{\partial \hat{\psi}}{\partial \hat{Y}}\frac{\partial \hat{Y}}{\partial y} + \frac{\partial \hat{\psi}}{\partial \hat{Z}}\frac{\partial \hat{Z}}{\partial y},$$

$$\frac{\partial \psi}{\partial z} = \frac{\partial \hat{\psi}}{\partial \hat{X}}\frac{\partial \hat{X}}{\partial z} + \frac{\partial \hat{\psi}}{\partial \hat{Y}}\frac{\partial \hat{Y}}{\partial z} + \frac{\partial \hat{\psi}}{\partial \hat{Z}}\frac{\partial \hat{Z}}{\partial z},$$

$$\begin{aligned}\frac{\partial^2 \psi}{\partial x^2} = {} & \frac{\partial^2 \hat{\psi}}{\partial \hat{X}^2}\left(\frac{\partial \hat{X}}{\partial x}\right)^2 + \frac{\partial^2 \hat{\psi}}{\partial \hat{Y}^2}\left(\frac{\partial \hat{Y}}{\partial x}\right)^2 + \frac{\partial^2 \hat{\psi}}{\partial \hat{Z}^2}\left(\frac{\partial \hat{Z}}{\partial x}\right)^2 + 2\frac{\partial^2 \hat{\psi}}{\partial \hat{X}\partial \hat{Y}}\left(\frac{\partial \hat{X}}{\partial x}\frac{\partial \hat{Y}}{\partial x}\right) \\ & + 2\frac{\partial^2 \hat{\psi}}{\partial \hat{X}\partial \hat{Z}}\left(\frac{\partial \hat{X}}{\partial x}\frac{\partial \hat{Z}}{\partial x}\right) + 2\frac{\partial^2 \hat{\psi}}{\partial \hat{Y}\partial \hat{Z}}\left(\frac{\partial \hat{Y}}{\partial x}\frac{\partial \hat{Z}}{\partial x}\right),\end{aligned}$$

$$\frac{\partial^2\psi}{\partial y^2}=\frac{\partial^2\hat{\psi}}{\partial\hat{X}^2}\left(\frac{\partial\hat{X}}{\partial y}\right)^2+\frac{\partial^2\hat{\psi}}{\partial\hat{Y}^2}\left(\frac{\partial\hat{Y}}{\partial y}\right)^2+\frac{\partial^2\hat{\psi}}{\partial\hat{Z}^2}\left(\frac{\partial\hat{Z}}{\partial y}\right)^2+2\frac{\partial^2\hat{\psi}}{\partial\hat{X}\partial\hat{Y}}\left(\frac{\partial\hat{X}}{\partial y}\frac{\partial\hat{Y}}{\partial y}\right)$$

$$+2\frac{\partial^2\hat{\psi}}{\partial\hat{X}\partial\hat{Z}}\left(\frac{\partial\hat{X}}{\partial y}\frac{\partial\hat{Z}}{\partial y}\right)+2\frac{\partial^2\hat{\psi}}{\partial\hat{Y}\partial\hat{Z}}\left(\frac{\partial\hat{Y}}{\partial y}\frac{\partial\hat{Z}}{\partial y}\right),$$

$$\frac{\partial^2\psi}{\partial z^2}=\frac{\partial^2\hat{\psi}}{\partial\hat{X}^2}\left(\frac{\partial\hat{X}}{\partial z}\right)^2+\frac{\partial^2\hat{\psi}}{\partial\hat{Y}^2}\left(\frac{\partial\hat{Y}}{\partial z}\right)^2+\frac{\partial^2\hat{\psi}}{\partial\hat{Z}^2}\left(\frac{\partial\hat{Z}}{\partial z}\right)^2+2\frac{\partial^2\hat{\psi}}{\partial\hat{X}\partial\hat{Y}}\left(\frac{\partial\hat{X}}{\partial z}\frac{\partial\hat{Y}}{\partial z}\right)$$

$$+2\frac{\partial^2\hat{\psi}}{\partial\hat{X}\partial\hat{Z}}\left(\frac{\partial\hat{X}}{\partial z}\frac{\partial\hat{Z}}{\partial z}\right)+2\frac{\partial^2\hat{\psi}}{\partial\hat{Y}\partial\hat{Z}}\left(\frac{\partial\hat{Y}}{\partial z}\frac{\partial\hat{Z}}{\partial z}\right),$$

$$\frac{\partial^2\psi}{\partial x\partial y}=\frac{\partial^2\hat{\psi}}{\partial\hat{X}^2}\left(\frac{\partial\hat{X}}{\partial x}\frac{\partial\hat{X}}{dy}\right)+\frac{\partial^2\hat{\psi}}{\partial\hat{Y}^2}\left(\frac{\partial\hat{Y}}{\partial x}\frac{\partial\hat{Y}}{dy}\right)+\frac{\partial^2\hat{\psi}}{\partial\hat{Z}^2}\left(\frac{\partial\hat{Z}}{\partial x}\frac{\partial\hat{Z}}{dy}\right)+\frac{\partial^2\hat{\psi}}{\partial\hat{X}\partial\hat{Y}}\left(\frac{\partial\hat{X}}{\partial x}\frac{\partial\hat{Y}}{\partial y}+\frac{\partial\hat{Y}}{\partial x}\frac{\partial\hat{X}}{\partial y}\right)$$

$$+\frac{\partial^2\hat{\psi}}{\partial\hat{Y}\partial\hat{Z}}\left(\frac{\partial\hat{Y}}{\partial x}\frac{\partial\hat{Z}}{\partial y}+\frac{\partial\hat{Z}}{\partial x}\frac{\partial\hat{Y}}{dy}\right)+\frac{\partial^2\hat{\psi}}{\partial\hat{X}\partial\hat{Z}}\left(\frac{\partial\hat{X}}{\partial x}\frac{\partial\hat{Z}}{\partial y}+\frac{\partial\hat{Z}}{\partial x}\frac{\partial\hat{X}}{dy}\right),$$

$$\frac{\partial^2\psi}{\partial x\partial z}=\frac{\partial^2\hat{\psi}}{\partial\hat{X}^2}\left(\frac{\partial\hat{X}}{\partial x}\frac{\partial\hat{X}}{dz}\right)+\frac{\partial^2\hat{\psi}}{\partial\hat{Y}^2}\left(\frac{\partial\hat{Y}}{\partial x}\frac{\partial\hat{Y}}{dz}\right)+\frac{\partial^2\hat{\psi}}{\partial\hat{Z}^2}\left(\frac{\partial\hat{Z}}{\partial x}\frac{\partial\hat{Z}}{dz}\right)+\frac{\partial^2\hat{\psi}}{\partial\hat{X}\partial\hat{Y}}\left(\frac{\partial\hat{X}}{\partial x}\frac{\partial\hat{Y}}{\partial z}+\frac{\partial\hat{Y}}{\partial x}\frac{\partial\hat{X}}{\partial z}\right)$$

$$+\frac{\partial^2\hat{\psi}}{\partial\hat{Y}\partial\hat{Z}}\left(\frac{\partial\hat{Y}}{\partial x}\frac{\partial\hat{Z}}{\partial z}+\frac{\partial\hat{Z}}{\partial x}\frac{\partial\hat{Y}}{dz}\right)+\frac{\partial^2\hat{\psi}}{\partial\hat{X}\partial\hat{Z}}\left(\frac{\partial\hat{X}}{\partial x}\frac{\partial\hat{Z}}{\partial z}+\frac{\partial\hat{Z}}{\partial x}\frac{\partial\hat{X}}{dz}\right),$$

$$\frac{\partial^2\psi}{\partial y\partial z}=\frac{\partial^2\hat{\psi}}{\partial\hat{X}^2}\left(\frac{\partial\hat{X}}{\partial y}\frac{\partial\hat{X}}{dz}\right)+\frac{\partial^2\hat{\psi}}{\partial\hat{Y}^2}\left(\frac{\partial\hat{Y}}{\partial y}\frac{\partial\hat{Y}}{dz}\right)+\frac{\partial^2\hat{\psi}}{\partial\hat{Z}^2}\left(\frac{\partial\hat{Z}}{\partial y}\frac{\partial\hat{Z}}{dz}\right)+\frac{\partial^2\hat{\psi}}{\partial\hat{X}\partial\hat{Y}}\left(\frac{\partial\hat{X}}{\partial y}\frac{\partial\hat{Y}}{\partial z}+\frac{\partial\hat{Y}}{\partial y}\frac{\partial\hat{X}}{\partial z}\right)$$

$$+\frac{\partial^2\hat{\psi}}{\partial\hat{Y}\partial\hat{Z}}\left(\frac{\partial\hat{Y}}{\partial y}\frac{\partial\hat{Z}}{\partial z}+\frac{\partial\hat{Z}}{\partial y}\frac{\partial\hat{Y}}{dz}\right)+\frac{\partial^2\hat{\psi}}{\partial\hat{X}\partial\hat{Z}}\left(\frac{\partial\hat{X}}{\partial y}\frac{\partial\hat{Z}}{\partial z}+\frac{\partial\hat{Z}}{\partial y}\frac{\partial\hat{X}}{dz}\right),$$

$$\frac{\partial\hat{X}}{\partial x}=\frac{M_{22}M_{33}-M_{23}M_{32}}{\det M},$$

$$\frac{\partial\hat{X}}{\partial y}=\frac{M_{13}M_{32}-M_{12}M_{33}}{\det M},$$

$$\frac{\partial\hat{X}}{\partial z}=\frac{M_{12}M_{23}-M_{13}M_{22}}{\det M},$$

$$\frac{\partial \hat{Y}}{\partial x} = \frac{M_{23}M_{31} - M_{21}M_{33}}{\det M},$$

$$\frac{\partial \hat{Y}}{\partial y} = \frac{M_{11}M_{33} - M_{13}M_{31}}{\det M},$$

$$\frac{\partial \hat{Y}}{\partial z} = \frac{M_{13}M_{21} - M_{11}M_{23}}{\det M},$$

$$\frac{\partial \hat{Z}}{\partial x} = \frac{M_{21}M_{32} - M_{22}M_{31}}{\det M},$$

$$\frac{\partial \hat{Z}}{\partial y} = \frac{M_{12}M_{31} - M_{11}M_{32}}{\det M},$$

$$\frac{\partial \hat{Z}}{\partial z} = \frac{M_{11}M_{22} - M_{12}M_{21}}{\det M}.$$

Linear Interpolation Polynomial

$$\hat{\psi}_j\left(\hat{X},\hat{Y},\hat{Z}\right) = a_j\hat{X} + b_j\hat{Y} + c_j\hat{Z} + d_j, \qquad j = 1,2,3,4,$$

Linear Reference Node Coordinates

$$\hat{A}_1 = (0,0,0),$$
$$\hat{A}_2 = (1,0,0),$$
$$\hat{A}_3 = (0,1,0),$$
$$\hat{A}_4 = (0,0,1).$$

Linear Tetrahedral Reference Basis Functions

$$\hat{\psi}_1\left(\hat{X},\hat{Y},\hat{Z}\right) = -\hat{X} - \hat{Y} - \hat{Z} + 1.$$
$$\hat{\psi}_2\left(\hat{X},\hat{Y},\hat{Z}\right) = \hat{X},$$
$$\hat{\psi}_3\left(\hat{X},\hat{Y},\hat{Z}\right) = \hat{Y},$$

$$\hat{\psi}_4(\hat{X},\hat{Y},\hat{Z}) = \hat{Z}.$$

Quadratic Interpolation Polynomial

$$\hat{\psi}_j(\hat{X},\hat{Y},\hat{Z}) = a_j\hat{X}^2 + b_j\hat{Y}^2 + c_j\hat{Z}^2 + d_j\hat{X}\hat{Y} + e_j\hat{X}\hat{Z} + f_j\hat{Y}\hat{Z} + g_j\hat{X} + h_j\hat{Y} + i_j\hat{Z} + j_j.$$

Quadratic Reference Node Coordinates

$$\hat{A}_1 = (0,0,0),$$
$$\hat{A}_2 = (1,0,0),$$
$$\hat{A}_3 = (0,1,0),$$
$$\hat{A}_4 = (0,0,1),$$
$$\hat{A}_5 = (0.5,0,0),$$
$$\hat{A}_6 = (0.5,0.5,0),$$
$$\hat{A}_7 = (0,0.5,0),$$
$$\hat{A}_8 = (0,0,0.5),$$
$$\hat{A}_9 = (0,0.5,0.5),$$
$$\hat{A}_{10} = (0.5,0,0.5).$$

Quadratic Tetrahedral Reference Basis function

$$\hat{\psi}_1(\hat{X},\hat{Y},\hat{Z}) = 2\hat{X}^2 + 2\hat{Y}^2 + 2\hat{Z}^2 + 4\hat{X}\hat{Y} + 4\hat{X}\hat{Z} + 4\hat{Y}\hat{Z} - 3\hat{X} - 3\hat{Y} - 3\hat{Z} + 1,$$
$$\hat{\psi}_2(\hat{X},\hat{Y},\hat{Z}) = 2\hat{X}^2 - \hat{X},$$
$$\hat{\psi}_3(\hat{X},\hat{Y},\hat{Z}) = 2\hat{Y}^2 - \hat{Y},$$
$$\hat{\psi}_4(\hat{X},\hat{Y},\hat{Z}) = 2\hat{Z}^2 - \hat{Z},$$
$$\hat{\psi}_5(\hat{X},\hat{Y},\hat{Z}) = -4\hat{X}^2 - 4\hat{X}\hat{Y} - 4\hat{X}\hat{Z} + 4\hat{X},$$
$$\hat{\psi}_6(\hat{X},\hat{Y},\hat{Z}) = 4\hat{X}\hat{Y},$$
$$\hat{\psi}_7(\hat{X},\hat{Y},\hat{Z}) = -4\hat{Y}^2 - 4\hat{X}\hat{Y} - 4\hat{Y}\hat{Z} + 4\hat{Y},$$

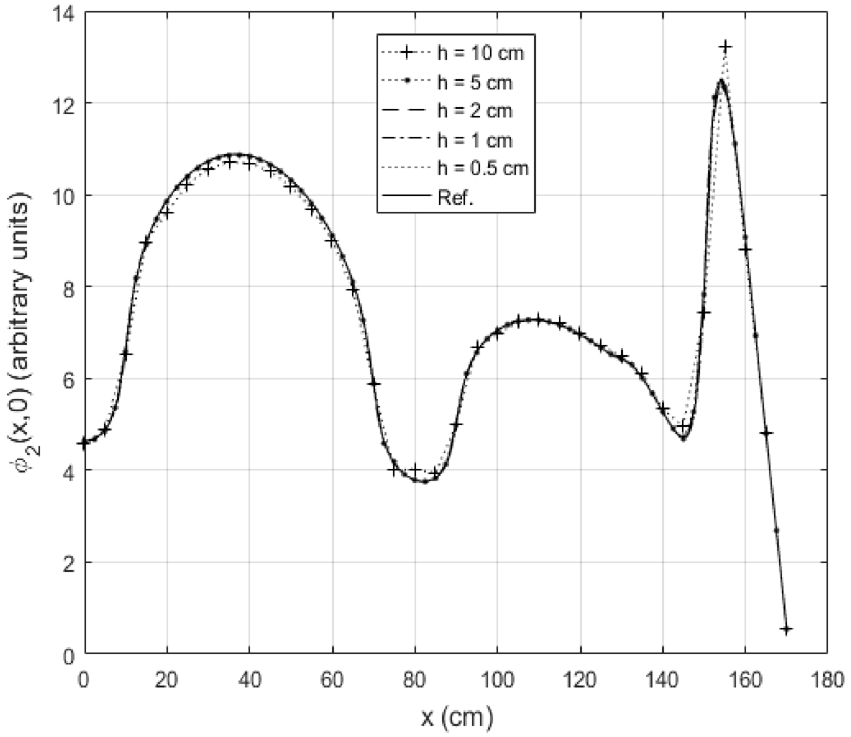
$$\hat{\psi}_8\left(\hat{X},\hat{Y},\hat{Z}\right) = -4\hat{Z}^2 - 4\hat{X}\hat{Z} - 4\hat{Y}\hat{Z} + 4\hat{Z},$$

$$\hat{\psi}_9\left(\hat{X},\hat{Y},\hat{Z}\right) = 4\hat{Y}\hat{Z},$$

$$\hat{\psi}_{10}\left(\hat{X},\hat{Y},\hat{Z}\right) = 4\hat{X}\hat{Z}.$$

APPENDIX B

SUPPLEMENTARY 2-D IAEA PWR BENCHMARK PLOTS



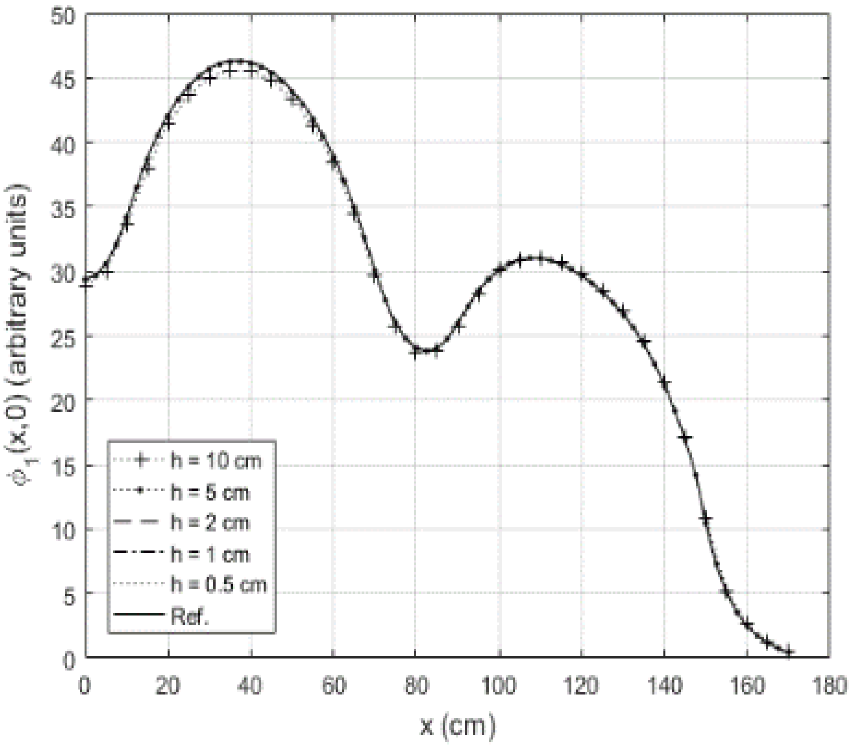


Figure IAEA 2D PWR Benchmark quadratic triangle h-refinement: (top) thermal, (bottom) fast flux traverse along x-axis

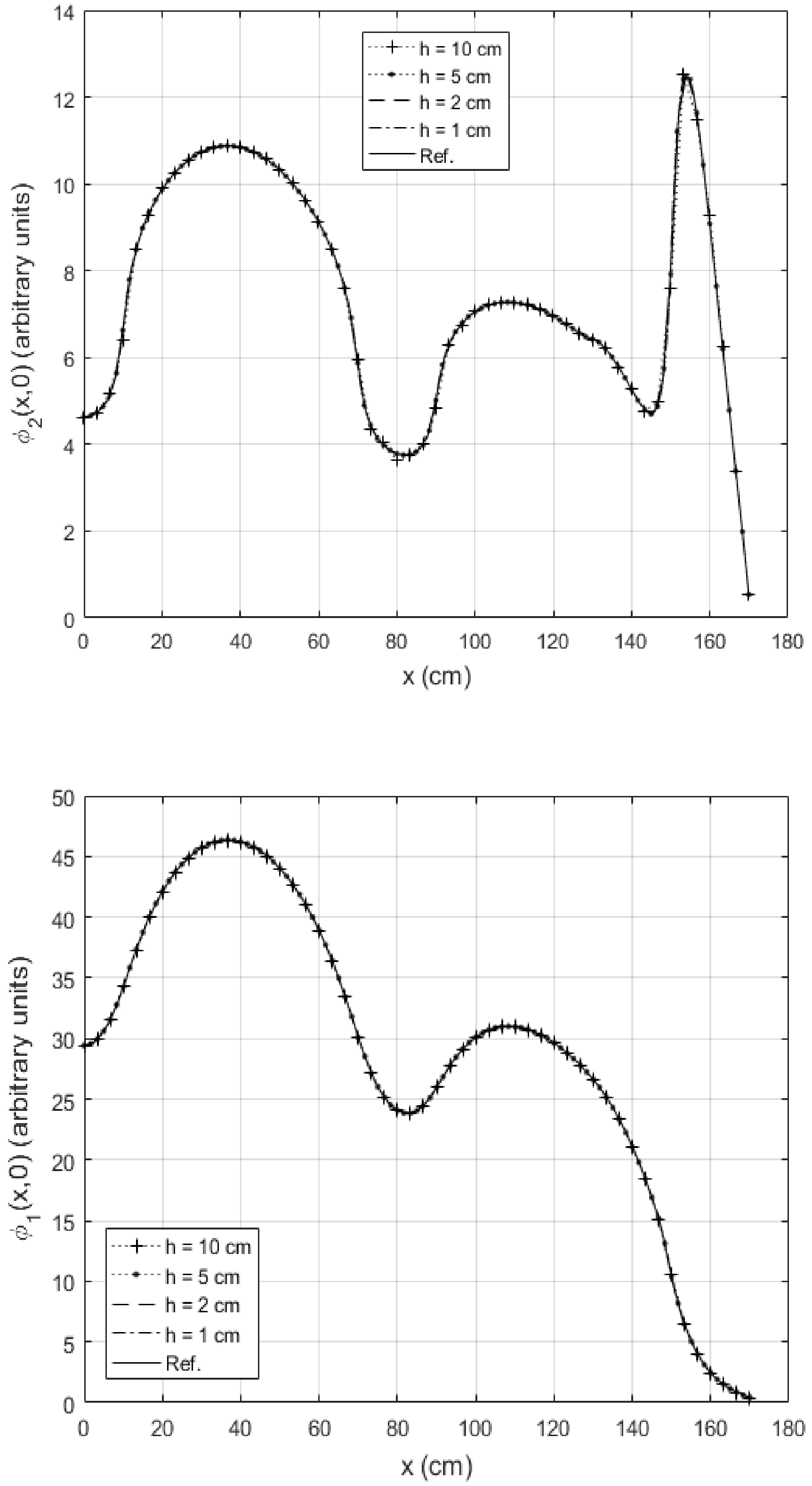


Figure IAEA 2D PWR Benchmark cubic triangle h-refinement: (top) thermal, (bottom) fast flux traverse along x-axis

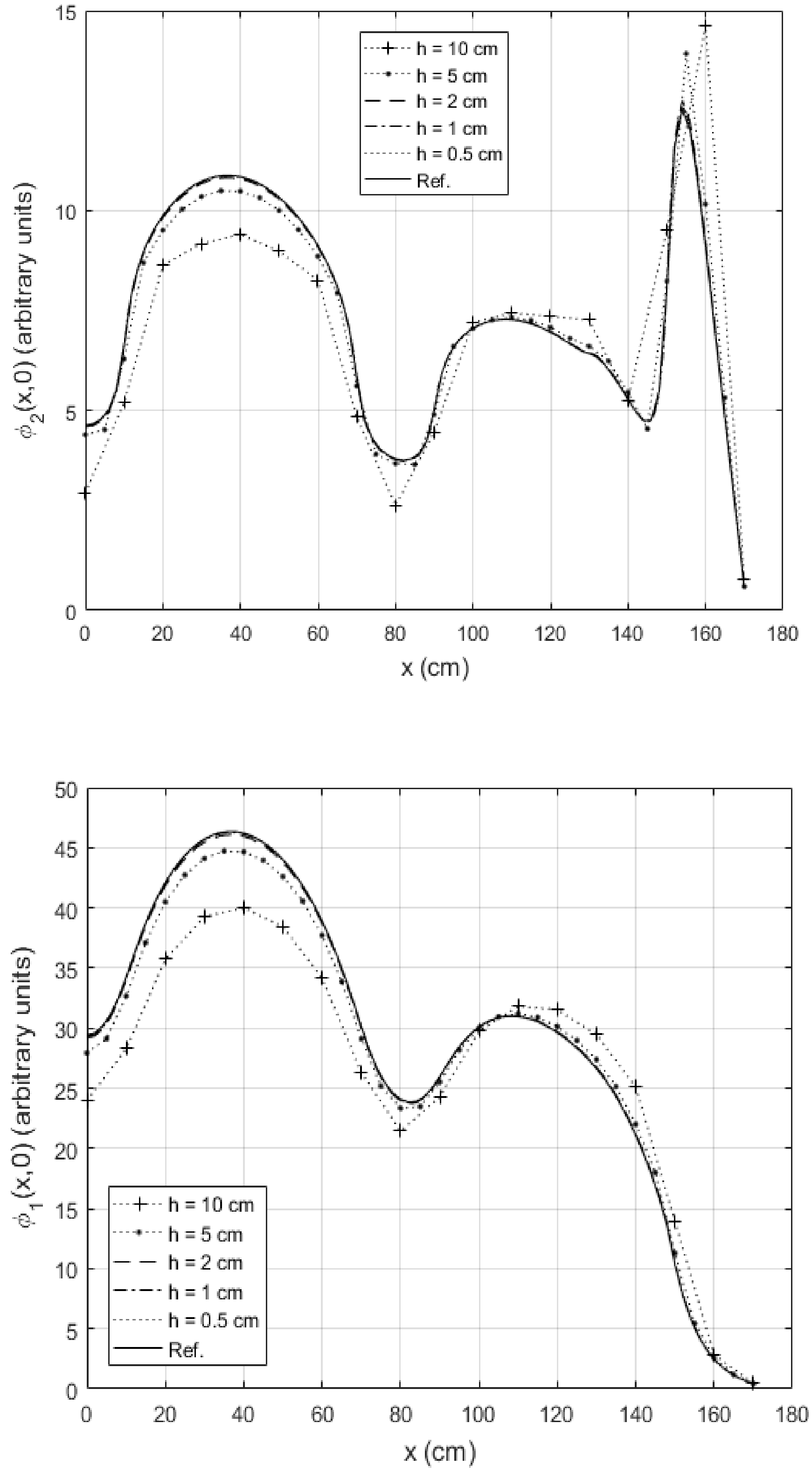


Figure IAEA 2D PWR Benchmark linear quadrangle h-refinement: (top) thermal, (bottom) fast flux traverse along x-axis

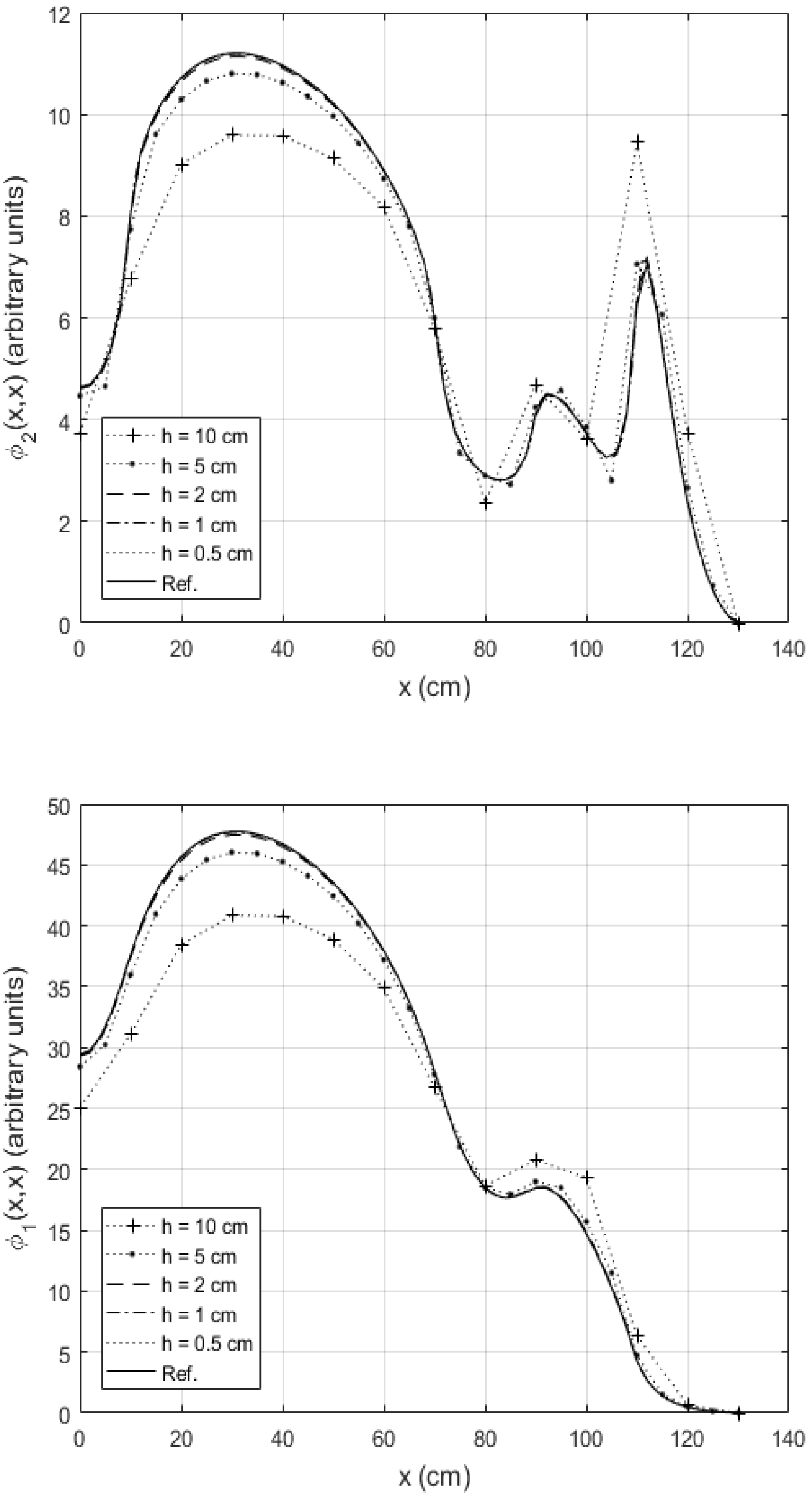


Figure IAEA 2D PWR Benchmark linear triangle h-refinement: (top) thermal, (bottom) fast flux traverse along the line y = x

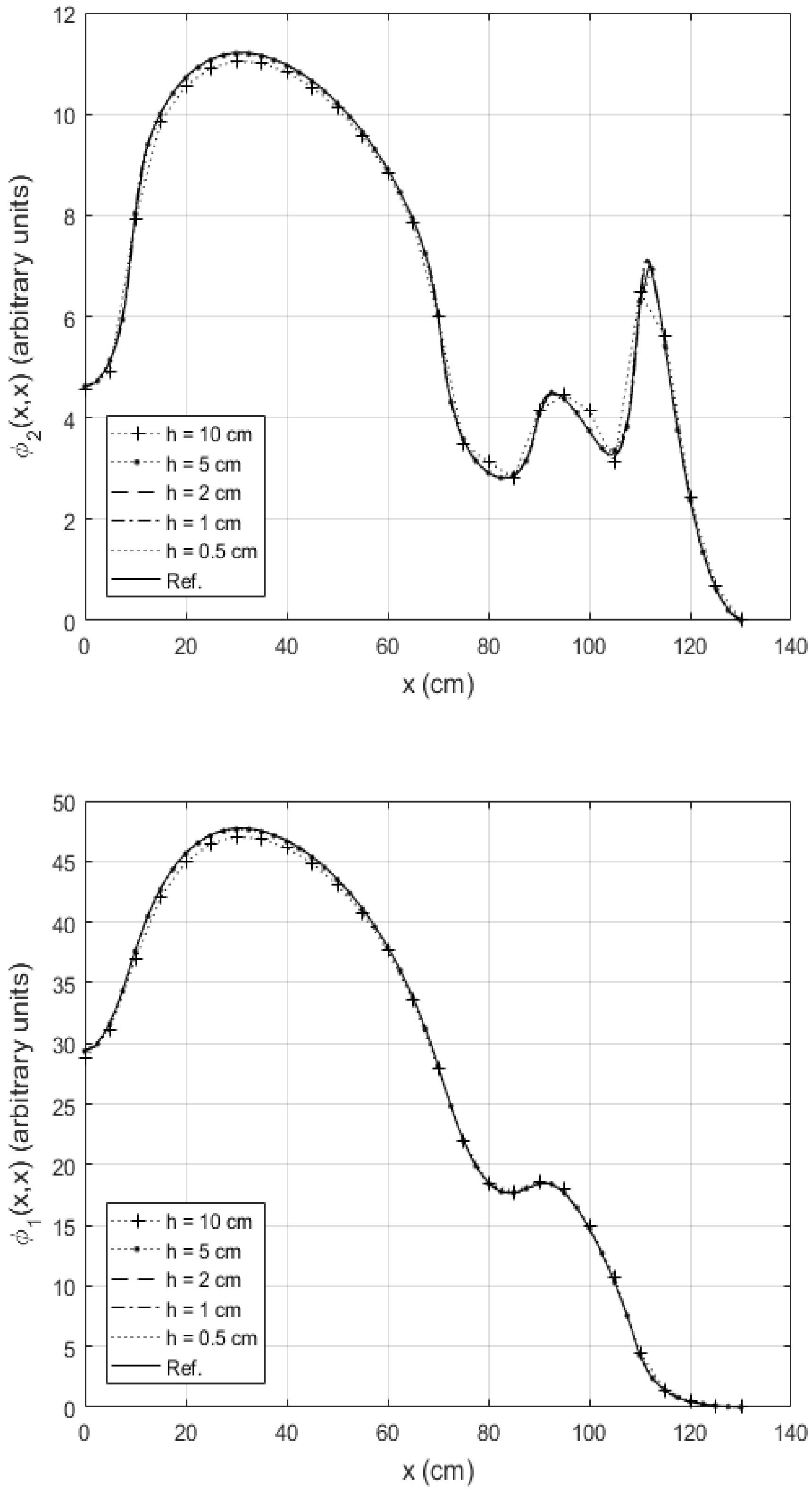


Figure IAEA 2D PWR Benchmark quadratic triangle h-refinement: (top) thermal, (bottom) fast flux traverse along the line y = x

APPENDIX C

SUPPLEMENTARY SERPENT 2 MSTR VALIDATION PLOTS

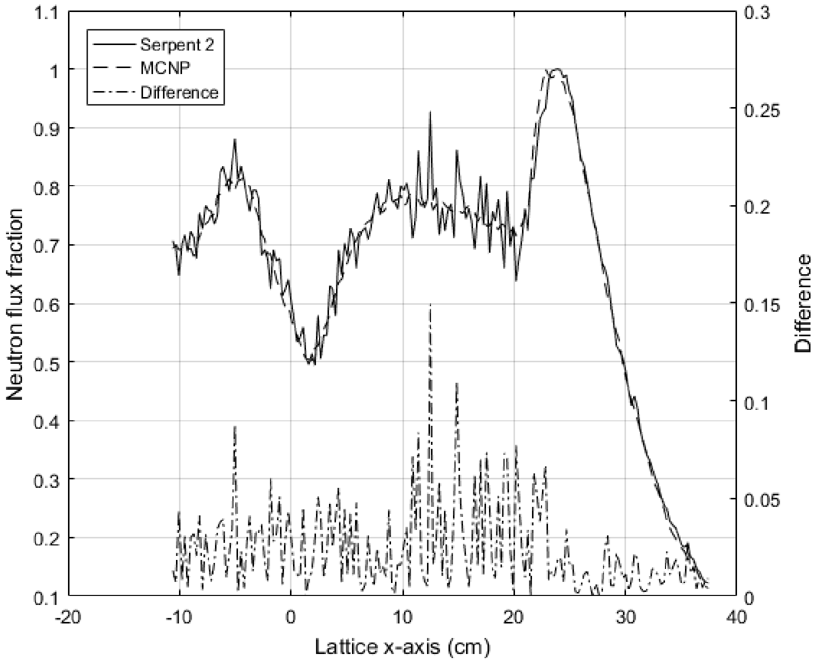


Figure Serpent 2 validation flux comparison along column 5/6 at the core top plane

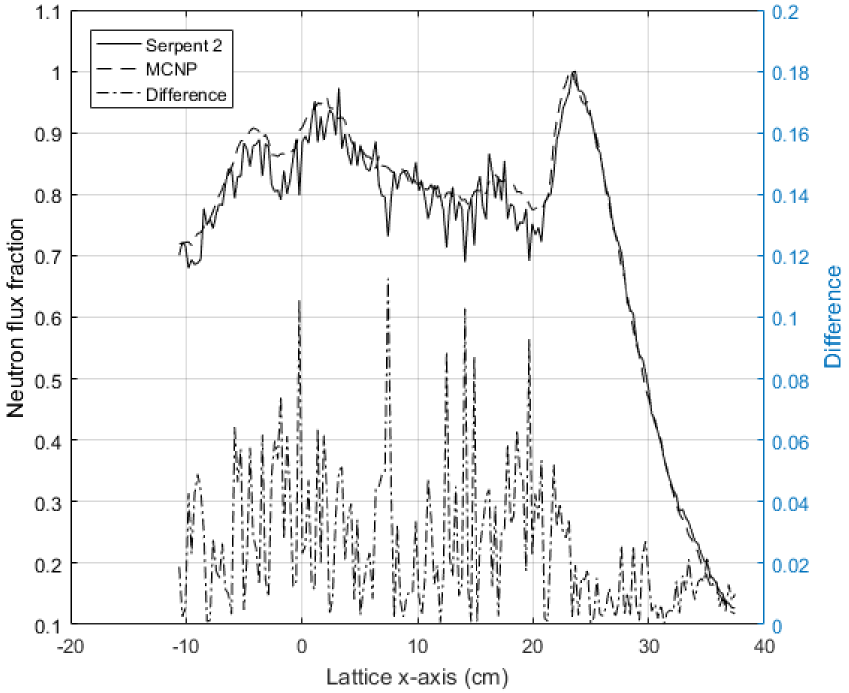


Figure Serpent 2 validation flux comparison along column 5/6 at the core bottom plane

APPENDIX D

2-D MSTR BENCHMARK MULTIGROUP CONSTANTS

Table MSTR 2D benchmark: Hydrogen transport corrected diffusion coefficients

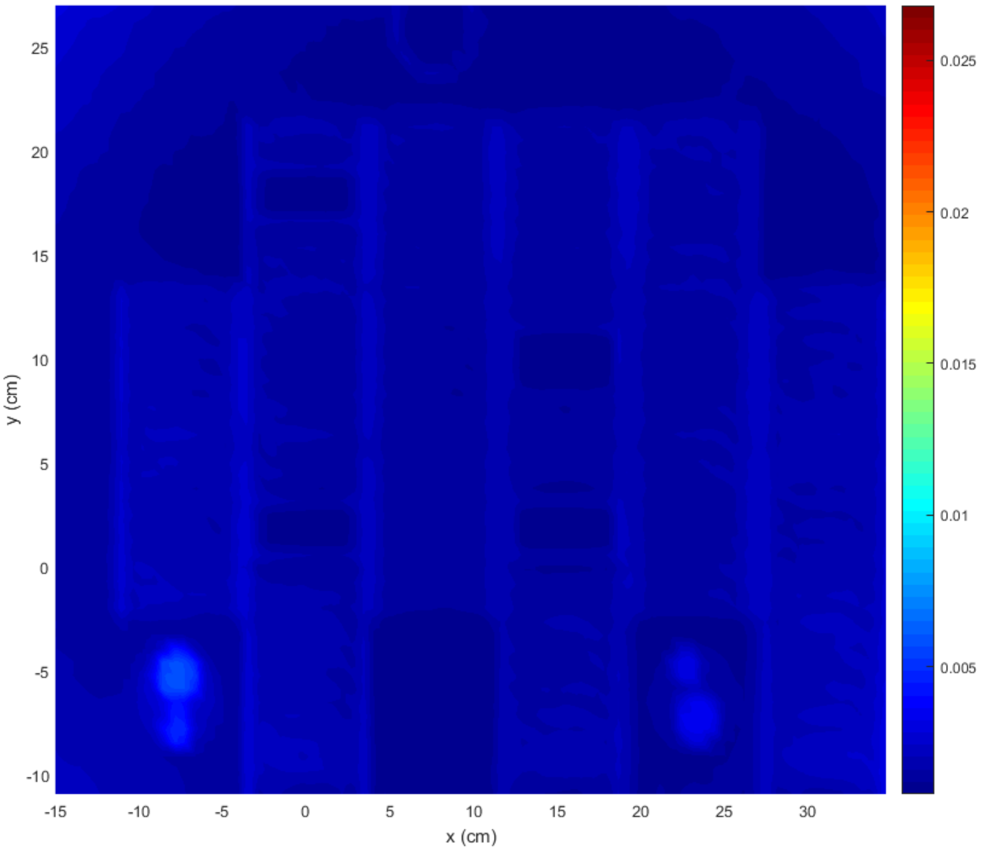
Region	$D_{1,trc}$	$D_{2,trc}$	$\Sigma_{s0,1\to2}$	$\Sigma_{s0,2\to1}$	$\Sigma_{s0\to3,1\to2}$	$\Sigma_{s0\to3,2\to1}$	$\Sigma_{a,1}$	$\Sigma_{a,2}$	$\Sigma_{T,1}$	$\Sigma_{T,2}$	$\Sigma_{f,1}$	$\Sigma_{f,2}$	$\bar{\nu}_1$	$\bar{\nu}_2$	χ_1
1	1.4430	0.1657	5.764E-02	8.662E-05	6.296E-02	2.065E-04	5.116E-04	1.889E-02	7.736E-01	2.656E+00	0	0	0	0	0
2	1.4657	0.1825	5.277E-02	9.433E-05	5.757E-02	2.221E-04	4.952E-04	1.823E-02	7.311E-01	2.416E+00	0	0	0	0	0
3	1.6416	0.2600	2.960E-02	1.645E-04	3.217E-02	3.658E-04	2.544E-03	4.733E-02	5.416E-01	1.717E+00	8.082E-04	2.472E-02	2.476	2.437	1
4	1.6747	0.2780	2.628E-02	2.270E-04	2.852E-02	5.087E-04	3.846E-03	6.885E-02	5.255E-01	1.629E+00	1.361E-03	4.542E-02	2.474	2.437	1
5	1.6755	0.2782	2.623E-02	2.336E-04	2.845E-02	5.238E-04	3.843E-03	6.913E-02	5.250E-01	1.630E+00	1.359E-03	4.567E-02	2.474	2.437	1
6	1.6801	0.2761	2.616E-02	2.073E-04	2.840E-02	4.655E-04	3.829E-03	6.858E-02	5.233E-01	1.632E+00	1.356E-03	4.511E-02	2.475	2.437	1
7	1.6815	0.2754	2.611E-02	2.068E-04	2.832E-02	4.640E-04	3.829E-03	6.880E-02	5.227E-01	1.636E+00	1.355E-03	4.528E-02	2.475	2.437	1
8	1.6684	0.2829	2.634E-02	2.544E-04	2.856E-02	5.721E-04	3.870E-03	6.767E-02	5.283E-01	1.612E+00	1.370E-03	4.457E-02	2.473	2.437	1
9	1.6653	0.2839	2.650E-02	2.577E-04	2.873E-02	5.776E-04	3.898E-03	6.796E-02	5.299E-01	1.607E+00	1.378E-03	4.483E-02	2.473	2.437	1
10	1.6280	0.2645	2.924E-02	2.096E-04	3.172E-02	4.692E-04	3.169E-03	4.858E-02	5.460E-01	1.710E+00	8.087E-04	2.458E-02	2.473	2.437	1
11	1.6671	0.2831	2.643E-02	2.529E-04	2.866E-02	5.660E-04	3.890E-03	6.780E-02	5.291E-01	1.609E+00	1.377E-03	4.467E-02	2.473	2.437	1
12	1.6799	0.2754	2.618E-02	2.111E-04	2.841E-02	4.735E-04	3.840E-03	6.868E-02	5.234E-01	1.636E+00	1.359E-03	4.519E-02	2.475	2.437	1
13	1.6769	0.2776	2.654E-02	2.107E-04	2.882E-02	4.746E-04	3.876E-03	6.853E-02	5.252E-01	1.623E+00	1.372E-03	4.511E-02	2.475	2.437	1
14	1.6311	0.2639	2.910E-02	2.051E-04	3.157E-02	4.591E-04	3.173E-03	4.861E-02	5.448E-01	1.711E+00	8.094E-04	2.448E-02	2.473	2.437	1
15	1.6717	0.2788	2.650E-02	2.304E-04	2.877E-02	5.186E-04	3.896E-03	7.022E-02	5.271E-01	1.625E+00	1.379E-03	4.660E-02	2.474	2.437	1
16	1.6329	0.2637	2.910E-02	2.015E-04	3.158E-02	4.505E-04	3.151E-03	4.881E-02	5.441E-01	1.711E+00	8.077E-04	2.482E-02	2.473	2.437	1
17	1.6695	0.2794	2.663E-02	2.399E-04	2.889E-02	5.366E-04	3.907E-03	6.945E-02	5.281E-01	1.626E+00	1.380E-03	4.598E-02	2.473	2.437	1
18	1.6747	0.2791	2.625E-02	2.247E-04	2.849E-02	5.071E-04	3.857E-03	6.678E-02	5.262E-01	1.620E+00	1.368E-03	4.368E-02	2.474	2.437	1
19	1.6637	0.2003	4.193E-02	1.566E-04	4.562E-02	3.481E-04	1.181E-03	3.296E-02	6.173E-01	2.218E+00	0	0	0	0	0
20	1.6891	0.2734	2.610E-02	1.804E-04	2.835E-02	4.046E-04	3.822E-03	6.914E-02	5.205E-01	1.636E+00	1.361E-03	4.549E-02	2.478	2.437	1
21	1.6886	0.2734	2.609E-02	1.811E-04	2.834E-02	4.062E-04	3.821E-03	6.937E-02	5.204E-01	1.635E+00	1.360E-03	4.568E-02	2.478	2.437	1
22	1.6482	0.2085	4.267E-02	1.493E-04	4.645E-02	3.316E-04	6.853E-04	2.391E-02	6.298E-01	2.134E+00	0	0	0	0	0
23	1.6885	0.2725	2.610E-02	1.800E-04	2.835E-02	4.012E-04	3.816E-03	6.790E-02	5.207E-01	1.640E+00	1.356E-03	4.441E-02	2.478	2.437	1
24	1.5274	0.1681	5.662E-02	6.254E-05	6.195E-02	1.492E-04	5.247E-04	1.845E-02	7.313E-01	2.608E+00	0	0	0	0	0
25	1.5297	0.6405	2.932E-03	5.784E-05	2.917E-03	5.645E-05	1.693E-04	6.206E-03	2.631E-01	5.867E-01	0	0	0	0	0

Table MSTR 2D benchmark: no Hydrogen transport corrected diffusion coefficients

Region	D_1	D_2	$\Sigma_{s,1\to2}$	$\Sigma_{s,2\to1}$	$\Sigma_{s0\to3,1\to2}$	$\Sigma_{s0\to3,2\to1}$	$\Sigma_{a,1}$	$\Sigma_{a,2}$	$\Sigma_{T,1}$	$\Sigma_{T,2}$	$\Sigma_{f,1}$	$\Sigma_{f,2}$	$\bar{\nu}_1$	$\bar{\nu}_2$	χ_1
1	1.4437	0.1541	5.765E-02	8.691E-05	6.297E-02	2.070E-04	5.116E-04	5.116E-04	9.288E-01	3.131E+00	0	0	0	0	0
2	1.4433	0.1694	5.276E-02	9.405E-05	5.762E-02	2.212E-04	4.952E-04	4.952E-04	8.715E-01	2.856E+00	0	0	0	0	0
3	1.6026	0.2382	2.962E-02	1.654E-04	3.219E-02	3.671E-04	2.543E-03	2.543E-03	6.320E-01	2.069E+00	8.066E-04	2.470E-02	2.476	2.437	1
4	1.6402	0.2518	2.628E-02	2.262E-04	2.852E-02	5.072E-04	3.850E-03	3.850E-03	6.118E-01	1.992E+00	1.361E-03	4.539E-02	2.474	2.437	1
5	1.6404	0.2517	2.623E-02	2.302E-04	2.846E-02	5.172E-04	3.844E-03	3.844E-03	6.113E-01	1.996E+00	1.359E-03	4.571E-02	2.474	2.437	1
6	1.6507	0.2511	2.614E-02	2.071E-04	2.838E-02	4.646E-04	3.831E-03	3.831E-03	6.091E-01	1.981E+00	1.356E-03	4.513E-02	2.475	2.437	1
7	1.6525	0.2505	2.611E-02	2.057E-04	2.835E-02	4.629E-04	3.826E-03	3.826E-03	6.082E-01	1.986E+00	1.356E-03	4.528E-02	2.475	2.437	1
8	1.6298	0.2544	2.633E-02	2.543E-04	2.856E-02	5.700E-04	3.869E-03	3.869E-03	6.152E-01	1.988E+00	1.370E-03	4.455E-02	2.473	2.437	1
9	1.6253	0.2553	2.650E-02	2.570E-04	2.874E-02	5.757E-04	3.898E-03	3.898E-03	6.172E-01	1.979E+00	1.378E-03	4.484E-02	2.473	2.437	1
10	1.5787	0.2396	2.925E-02	2.130E-04	3.172E-02	4.776E-04	3.170E-03	3.170E-03	6.376E-01	2.097E+00	8.089E-04	2.458E-02	2.473	2.437	1
11	1.6283	0.2549	2.643E-02	2.521E-04	2.866E-02	5.644E-04	3.888E-03	3.888E-03	6.163E-01	1.979E+00	1.376E-03	4.471E-02	2.473	2.437	1
12	1.6496	0.2505	2.620E-02	2.068E-04	2.843E-02	4.645E-04	3.842E-03	3.842E-03	6.094E-01	1.988E+00	1.360E-03	4.518E-02	2.475	2.437	1
13	1.6471	0.2525	2.653E-02	2.109E-04	2.880E-02	4.728E-04	3.878E-03	3.878E-03	6.113E-01	1.970E+00	1.373E-03	4.510E-02	2.475	2.437	1
14	1.5842	0.2394	2.910E-02	2.068E-04	3.158E-02	4.644E-04	3.176E-03	3.176E-03	6.364E-01	2.093E+00	8.092E-04	2.450E-02	2.473	2.437	1
15	1.6374	0.2524	2.650E-02	2.288E-04	2.876E-02	5.125E-04	3.895E-03	3.895E-03	6.136E-01	1.985E+00	1.379E-03	4.661E-02	2.474	2.437	1
16	1.5863	0.2393	2.910E-02	2.025E-04	3.158E-02	4.556E-04	3.152E-03	3.152E-03	6.354E-01	2.092E+00	8.072E-04	2.484E-02	2.473	2.437	1
17	1.6325	0.2524	2.663E-02	2.401E-04	2.889E-02	5.378E-04	3.906E-03	3.906E-03	6.151E-01	1.994E+00	1.379E-03	4.594E-02	2.473	2.437	1
18	1.6422	0.2531	2.625E-02	2.266E-04	2.848E-02	5.110E-04	3.854E-03	3.854E-03	6.127E-01	1.974E+00	1.368E-03	4.369E-02	2.474	2.437	1
19	1.6322	0.1847	4.193E-02	1.555E-04	4.562E-02	3.470E-04	1.182E-03	1.182E-03	7.469E-01	2.664E+00	0	0	0	0	0
20	1.6700	0.2505	2.610E-02	1.844E-04	2.836E-02	4.156E-04	3.827E-03	3.827E-03	6.056E-01	1.964E+00	1.361E-03	4.551E-02	2.478	2.437	1
21	1.6692	0.2506	2.609E-02	1.803E-04	2.834E-02	4.063E-04	3.822E-03	3.822E-03	6.057E-01	1.964E+00	1.360E-03	4.566E-02	2.478	2.437	1
22	1.6139	0.1920	4.267E-02	1.504E-04	4.645E-02	3.336E-04	6.864E-04	6.864E-04	7.611E-01	2.569E+00	0	0	0	0	0
23	1.6688	0.2498	2.610E-02	1.778E-04	2.836E-02	3.996E-04	3.811E-03	3.811E-03	6.059E-01	1.970E+00	1.355E-03	4.442E-02	2.478	2.437	1
24	1.5688	0.1571	5.662E-02	6.255E-05	6.195E-02	1.486E-04	5.249E-04	5.249E-04	8.787E-01	3.056E+00	0	0	0	0	0
25	1.5259	0.6281	2.940E-03	5.847E-05	2.930E-03	5.751E-05	1.693E-04	1.693E-04	3.043E-01	6.273E-01	0	0	0	0	0

APPENDIX E

SERPENT 2 THERMAL FLUX MAP RELATIVE ERROR


25
20
15
10
5
0
-5
-10
y (cm)
-15
-10
-5
0
5
10
15
20
25
30
x (cm)
0.025
0.02
0.015
0.01
0.005

APPENDIX F

GMSH INPUT: 120W MSTR CONFIGURATION

```
// MSTR CORE 120W CONFIGURATION
// Wayne J. Brewster

SetFactory("OpenCASCADE");

//--------------------------------------------------
// set mesh options
//--------------------------------------------------

x = 3.0;                    // mesh size
Mesh.ElementOrder = 1;   // element order
Mesh.Lloyd = 100;          // Lloyd smoothing steps
Mesh.Algorithm = 6;        // Frontal algorithm

//--------------------------------------------------
// Lattice Geometric Definitions
//--------------------------------------------------

Point(1) = {-3.85445, 3.66903, 0, x};
Point(2) = {-3.85445, -4.43103, 0, x};
Point(3) = {-3.85445, -12.53109, 0, x};
Point(4) = {-3.85445, -20.63115, 0, x};
Point(5) = {-3.85445, -28.73121, 0, x};
Point(6) = {-3.85445, -36.83127, 0, x};
Point(7) = {-3.85445, -44.93133, 0, x};
Point(8) = {3.85445, 3.66903, 0, x};
Point(9) = {3.85445, -4.43103, 0, x};
Point(10) = {3.85445, -12.53109, 0, x};
Point(11) = {3.85445, -20.63115, 0, x};
Point(12) = {3.85445, -28.73121, 0, x};
Point(13) = {3.85445, -36.83127, 0, x};
Point(14) = {3.85445, -44.93133, 0, x};
Point(15) = {11.56335, 3.66903, 0, x};
Point(16) = {11.56335, -4.43103, 0, x};
Point(17) = {11.56335, -12.53109, 0, x};
Point(18) = {11.56335, -20.63115, 0, x};
Point(19) = {11.56335, -28.73121, 0, x};
Point(20) = {11.56335, -36.83127, 0, x};
Point(21) = {11.56335, -44.93133, 0, x};
Point(22) = {19.27225, 3.66903, 0, x};
Point(23) = {19.27225, -4.43103, 0, x};
Point(24) = {19.27225, -12.53109, 0, x};
Point(25) = {19.27225, -20.63115, 0, x};
Point(26) = {19.27225, -28.73121, 0, x};
Point(27) = {19.27225, -36.83127, 0, x};
Point(28) = {19.27225, -44.93133, 0, x};
Point(29) = {26.98115, 3.66903, 0, x};
Point(30) = {26.98115, -4.43103, 0, x};
Point(31) = {26.98115, -12.53109, 0, x};
Point(32) = {26.98115, -20.63115, 0, x};
Point(33) = {26.98115, -28.73121, 0, x};
Point(34) = {26.98115, -36.83127, 0, x};
Point(35) = {26.98115, -44.93133, 0, x};
Point(36) = {34.69005, 3.66903, 0, x};
Point(37) = {34.69005, -4.43103, 0, x};
Point(38) = {34.69005, -12.53109, 0, x};
Point(39) = {34.69005, -20.63115, 0, x};
Point(40) = {34.69005, -28.73121, 0, x};
Point(41) = {34.69005, -36.83127, 0, x};
Point(42) = {34.69005, -44.93133, 0, x};
Point(43) = {42.39895, 3.66903, 0, x};
Point(44) = {42.39895, -4.43103, 0, x};
Point(45) = {42.39895, -12.53109, 0, x};
Point(46) = {42.39895, -20.63115, 0, x};
Point(47) = {42.39895, -28.73121, 0, x};
Point(48) = {42.39895, -36.83127, 0, x};
Point(49) = {42.39895, -44.93133, 0, x};
Point(50) = {50.10785, 3.66903, 0, x};
Point(51) = {50.10785, -4.43103, 0, x};
Point(52) = {50.10785, -12.53109, 0, x};
Point(53) = {50.10785, -20.63115, 0, x};
Point(54) = {50.10785, -28.73121, 0, x};
Point(55) = {50.10785, -36.83127, 0, x};
Point(56) = {50.10785, -44.93133, 0, x};
Point(57) = {57.81675, 3.66903, 0, x};
Point(58) = {57.81675, -4.43103, 0, x};
Point(59) = {57.81675, -12.53109, 0, x};
Point(60) = {57.81675, -20.63115, 0, x};
Point(61) = {57.81675, -28.73121, 0, x};
Point(62) = {57.81675, -36.83127, 0, x};
Point(63) = {57.81675, -44.93133, 0, x};
Point(64) = {65.52565, 3.66903, 0, x};
Point(65) = {65.52565, -4.43103, 0, x};
Point(66) = {65.52565, -12.53109, 0, x};
Point(67) = {65.52565, -20.63115, 0, x};
Point(68) = {65.52565, -28.73121, 0, x};
Point(69) = {65.52565, -36.83127, 0, x};
Point(70) = {65.52565, -44.93133, 0, x};
Line(1) = {1, 8};
Line(2) = {8, 15};
Line(3) = {15, 22};
Line(4) = {22, 29};
Line(5) = {29, 36};
Line(6) = {36, 43};
Line(7) = {43, 50};
Line(8) = {50, 57};
Line(9) = {57, 64};
Line(10) = {2, 9};
Line(11) = {9, 16};
Line(12) = {16, 23};
Line(13) = {23, 30};
Line(14) = {30, 37};
Line(15) = {37, 44};
Line(16) = {44, 51};
Line(17) = {51, 58};
Line(18) = {58, 65};
Line(19) = {3, 10};
Line(20) = {10, 17};
Line(21) = {17, 24};
Line(22) = {24, 31};
Line(23) = {31, 38};
Line(24) = {38, 45};
Line(25) = {45, 52};
Line(26) = {52, 59};
Line(27) = {59, 66};
Line(28) = {4, 11};
Line(29) = {11, 18};
Line(30) = {18, 25};
Line(31) = {25, 32};
Line(32) = {32, 39};
Line(33) = {39, 46};
Line(34) = {46, 53};
Line(35) = {53, 60};
Line(36) = {60, 67};
Line(37) = {5, 12};
Line(38) = {12, 19};
Line(39) = {19, 26};
Line(40) = {26, 33};
Line(41) = {33, 40};
Line(42) = {40, 47};
Line(43) = {47, 54};
Line(44) = {54, 61};
Line(45) = {61, 68};
Line(46) = {6, 13};
Line(47) = {13, 20};
Line(48) = {20, 27};
Line(49) = {27, 34};
Line(50) = {34, 41};
Line(51) = {41, 48};
Line(52) = {48, 55};
Line(53) = {55, 62};
Line(54) = {62, 69};
Line(55) = {7, 14};
Line(56) = {14, 21};
Line(57) = {21, 28};
Line(58) = {28, 35};
Line(59) = {35, 42};
Line(60) = {42, 49};
Line(61) = {49, 56};
Line(62) = {56, 63};
Line(63) = {63, 70};
Line(64) = {1, 2};
Line(65) = {2, 3};
Line(66) = {3, 4};
Line(67) = {4, 5};
Line(68) = {5, 6};
Line(69) = {6, 7};
Line(70) = {8, 9};
Line(71) = {9, 10};
Line(72) = {10, 11};
Line(73) = {11, 12};
Line(74) = {12, 13};
Line(75) = {13, 14};
Line(76) = {15, 16};
Line(77) = {16, 17};
Line(78) = {17, 18};
Line(79) = {18, 19};
Line(80) = {19, 20};
Line(81) = {20, 21};
Line(82) = {22, 23};
Line(83) = {23, 24};
Line(84) = {24, 25};
Line(85) = {25, 26};
Line(86) = {26, 27};
Line(87) = {27, 28};
Line(88) = {29, 30};
Line(89) = {30, 31};
Line(90) = {31, 32};
Line(91) = {32, 33};
Line(92) = {33, 34};
Line(93) = {34, 35};
Line(94) = {36, 37};
Line(95) = {37, 38};
Line(96) = {38, 39};
Line(97) = {39, 40};
Line(98) = {40, 41};
Line(99) = {41, 42};
Line(100) = {43, 44};
Line(101) = {44, 45};
Line(102) = {45, 46};
Line(103) = {46, 47};
Line(104) = {47, 48};
Line(105) = {48, 49};
Line(106) = {50, 51};
Line(107) = {51, 52};
Line(108) = {52, 53};
Line(109) = {53, 54};
Line(110) = {54, 55};
Line(111) = {55, 56};
Line(112) = {57, 58};
Line(113) = {58, 59};
Line(114) = {59, 60};
Line(115) = {60, 61};
Line(116) = {61, 62};
Line(117) = {62, 63};
Line(118) = {64, 65};
Line(119) = {65, 66};
Line(120) = {66, 67};
Line(121) = {67, 68};
Line(122) = {68, 69};
Line(123) = {69, 70};
Line Loop(1) = {10, -70, -1, 64};
Plane Surface(1) = {1};
Line Loop(2) = {11, -76, -2, 70};
Plane Surface(2) = {2};
Line Loop(3) = {12, -82, -3, 76};
Plane Surface(3) = {3};
Line Loop(4) = {13, -88, -4, 82};
Plane Surface(4) = {4};
Line Loop(5) = {14, -94, -5, 88};
Plane Surface(5) = {5};
Line Loop(6) = {15, -100, -6, 94};
Plane Surface(6) = {6};
Line Loop(7) = {16, -106, -7, 100};
```

```
Plane Surface(7) = {7};
Line Loop(8) = {17, -112, -8, 106};
Plane Surface(8) = {8};
Line Loop(9) = {18, -118, -9, 112};
Plane Surface(9) = {9};
Line Loop(10) = {19, -71, -10, 65};
Plane Surface(10) = {10};
Line Loop(11) = {20, -77, -11, 71};
Plane Surface(11) = {11};
Line Loop(12) = {21, -83, -12, 77};
Plane Surface(12) = {12};
Line Loop(13) = {22, -89, -13, 83};
Plane Surface(13) = {13};
Line Loop(14) = {23, -95, -14, 89};
Plane Surface(14) = {14};
Line Loop(15) = {24, -101, -15, 95};
Plane Surface(15) = {15};
Line Loop(16) = {25, -107, -16, 101};
Plane Surface(16) = {16};
Line Loop(17) = {26, -113, -17, 107};
Plane Surface(17) = {17};
Line Loop(18) = {27, -119, -18, 113};
Plane Surface(18) = {18};
Line Loop(19) = {28, -72, -19, 66};
Plane Surface(19) = {19};
Line Loop(20) = {29, -78, -20, 72};
Plane Surface(20) = {20};
Line Loop(21) = {30, -84, -21, 78};
Plane Surface(21) = {21};
Line Loop(22) = {31, -90, -22, 84};
Plane Surface(22) = {22};
Line Loop(23) = {32, -96, -23, 90};
Plane Surface(23) = {23};
Line Loop(24) = {33, -102, -24, 96};
Plane Surface(24) = {24};
Line Loop(25) = {34, -108, -25, 102};
Plane Surface(25) = {25};
Line Loop(26) = {35, -114, -26, 108};
Plane Surface(26) = {26};
Line Loop(27) = {36, -120, -27, 114};
Plane Surface(27) = {27};
Line Loop(28) = {37, -73, -28, 67};
Plane Surface(28) = {28};
Line Loop(29) = {38, -79, -29, 73};
Plane Surface(29) = {29};
Line Loop(30) = {39, -85, -30, 79};
Plane Surface(30) = {30};
Line Loop(31) = {40, -91, -31, 85};
Plane Surface(31) = {31};
Line Loop(32) = {41, -97, -32, 91};
Plane Surface(32) = {32};
Line Loop(33) = {42, -103, -33, 97};
Plane Surface(33) = {33};
Line Loop(34) = {43, -109, -34, 103};
Plane Surface(34) = {34};
Line Loop(35) = {44, -115, -35, 109};
```

```
Plane Surface(35) = {35};
Line Loop(36) = {45, -121, -36, 115};
Plane Surface(36) = {36};
Line Loop(37) = {46, -74, -37, 68};
Plane Surface(37) = {37};
Line Loop(38) = {47, -80, -38, 74};
Plane Surface(38) = {38};
Line Loop(39) = {48, -86, -39, 80};
Plane Surface(39) = {39};
Line Loop(40) = {49, -92, -40, 86};
Plane Surface(40) = {40};
Line Loop(41) = {50, -98, -41, 92};
Plane Surface(41) = {41};
Line Loop(42) = {51, -104, -42, 98};
Plane Surface(42) = {42};
Line Loop(43) = {52, -110, -43, 104};
Plane Surface(43) = {43};
Line Loop(44) = {53, -116, -44, 110};
Plane Surface(44) = {44};
Line Loop(45) = {54, -122, -45, 116};
Plane Surface(45) = {45};
Line Loop(46) = {55, -75, -46, 69};
Plane Surface(46) = {46};
Line Loop(47) = {56, -81, -47, 75};
Plane Surface(47) = {47};
Line Loop(48) = {57, -87, -48, 81};
Plane Surface(48) = {48};
Line Loop(49) = {58, -93, -49, 87};
Plane Surface(49) = {49};
Line Loop(50) = {59, -99, -50, 93};
Plane Surface(50) = {50};
Line Loop(51) = {60, -105, -51, 99};
Plane Surface(51) = {51};
Line Loop(52) = {61, -111, -52, 105};
Plane Surface(52) = {52};
Line Loop(53) = {62, -117, -53, 111};
Plane Surface(53) = {53};
Line Loop(54) = {63, -123, -54, 117};
Plane Surface(54) = {54};

//-------------------------------------------------
// Transfinite all plane surfaces to
// create structured mesh in lattices
//-------------------------------------------------

Transfinite Surface(1);
Transfinite Surface(2);
Transfinite Surface(3);
Transfinite Surface(4);
Transfinite Surface(5);
Transfinite Surface(6);
Transfinite Surface(7);
Transfinite Surface(8);
Transfinite Surface(9);
Transfinite Surface(10);
Transfinite Surface(11);
Transfinite Surface(12);
Transfinite Surface(13);
Transfinite Surface(14);
Transfinite Surface(15);
Transfinite Surface(16);
Transfinite Surface(17);
Transfinite Surface(18);
Transfinite Surface(19);
Transfinite Surface(20);
Transfinite Surface(21);
Transfinite Surface(22);
Transfinite Surface(23);
Transfinite Surface(24);
Transfinite Surface(25);
Transfinite Surface(26);
Transfinite Surface(27);
Transfinite Surface(28);
Transfinite Surface(29);
Transfinite Surface(30);
Transfinite Surface(31);
Transfinite Surface(32);
Transfinite Surface(33);
Transfinite Surface(34);
Transfinite Surface(35);
Transfinite Surface(36);
Transfinite Surface(37);
Transfinite Surface(38);
Transfinite Surface(39);
Transfinite Surface(40);
Transfinite Surface(41);
Transfinite Surface(42);
Transfinite Surface(43);
Transfinite Surface(44);
Transfinite Surface(45);
Transfinite Surface(46);
Transfinite Surface(47);
Transfinite Surface(48);
Transfinite Surface(49);
Transfinite Surface(50);
Transfinite Surface(51);
Transfinite Surface(52);
Transfinite Surface(53);
Transfinite Surface(54);

//----------------------------------------------------
// Reactor Pool Geometry Definitions
//----------------------------------------------------

Point(71) = {-25.25, -90.09223, 0, x};
Point(72) = {96.56865, -90.09223, 0, x};
Point(73) = {96.56865, -64.69223, 0, x};
Point(74) = {96.56865, 25.77173, 0, x};
Point(75) = {-25.25, 25.77173, 0, x};
Point(77) = {-25.25, -90.09223, 0, x};
Point(76) = {-25.25, -64.69223, 0, x};
```

```
Line(124) = {71, 72};
Line(125) = {72, 73};
Line(126) = {73, 74};
Line(127) = {74, 75};
Line(128) = {75, 76};
Line(129) = {76, 71};
Line(130) = {73, 76};

Line Loop(55) = {124, 125, 130, 129};
Plane Surface(55) = {55};
Line Loop(56) = {130, -128, -127, -126};
Line Loop(57) = {55, 56, 57, 58, 59, 60, 61, 62, 63, -123, -122, -121, -120, -119, -118, -9, -8, -7, -6, -5, -4,
-3, -2, -1, 64, 65, 66, 67, 68, 69};
Plane Surface(56) = {56, 57};

//--------------------------------------------------
// Physical Surface Definitions
// Corresponds to spatial homogenization regions
// Order must be the same in the Serpent set gcu
// command and the .res output
//--------------------------------------------------

Physical Surface(1) = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 27, 28, 29, 30,
37, 38, 39, 46, 47, 48, 51};
Physical Surface(2) = {15};
Physical Surface(3) = {23};
Physical Surface(4) = {24};
Physical Surface(5) = {25};
Physical Surface(6) = {26};
Physical Surface(7) = {31};
Physical Surface(8) = {32};
Physical Surface(9) = {33};
Physical Surface(10) = {34};
Physical Surface(11) = {35};
Physical Surface(12) = {36};
Physical Surface(13) = {40};
Physical Surface(14) = {41};
Physical Surface(15) = {42};
Physical Surface(16) = {43};
Physical Surface(17) = {44};
Physical Surface(18) = {45};
Physical Surface(19) = {49};
Physical Surface(20) = {50};
Physical Surface(21) = {52};
Physical Surface(22) = {53};
Physical Surface(23) = {54};
Physical Surface(24) = {56, 57};
Physical Surface(25) = {55};

// Vacuum boundary line
Physical Line(1003) = {124, 125, 126, 127, 128, 129};
```

BIBLIOGRAPHY

[1] A. G. Mylonakis, M. Varvayanni, N. Catsaros, P. Savva and D. G. E. Grigoriadis, "Multi-physics and multi-scale methods used in nuclear reactor analysis," *Annals of Nuclear Energy,* vol. 72, pp. 104-119, 2014.

[2] B. Chanaron, "Advanced multi-physics simulation for reactor safety in the framework of the NURESAFE project," *Annals of Nuclear Energy,* vol. 84, pp. 166-177, 2015.

[3] K. Smith, "Spatial Homogenization Methods for Light Water Reactor Analysis," *Ph.D. Thesis. Massachusetts Institute of Technology,* 1980.

[4] K. Smith, "An analyic nodal method for solving the two-group multidimensional, static and transient neutron diffusion equations," *Thesis. Massachusetts Institute of Technology,* 1979.

[5] T. Downar, Y. Xu and V. Seker, Theory Manual for the PARCS Kinetics Core Simulator Module, Department of Nuclear Engineering and Radiological Sciences University of Michigan, USA , 2009.

[6] D. A. Knoll, L. Chacon, L. G. Margolin and V. A. Mousseau, "On balanced approximations for time integration of multiple time scale systems," *J. Comput. Phys.,* vol. 185, pp. 583-611, 2003.

[7] D. A. Knoll and D. Keyes, "Jacobian-free Newton-Krylov methods: a survey of approaches and applications," *J. Comput. Phys.,* vol. 153, pp. 357-397, 2004.

[8] D. Gaston, C. Newman, G. Hansen and D. Lebrun-Grandie, "MOOSE: A parallel computational framework for coupled systems of nonlinear equations," *Nuclear Engineering and Design: 239,* pp. 1768-1778, 2009.

[9] H. Park, D. A. Knoll, D. R. Gaston and R. C. Martineau, "Tightly Coupled Multiphysics Algorithms for Pebble Bed Reactors," *Nuclear Science and Engineering: 166,* pp. 118-133, 2010.

[10] L. A. Semenza, E. E. Lewis and E. C. Rossow, "The Application of the FInite Element Method to the Multigroup Neutron Diffusion Equation," *Nuclear Science and Engineering: 47,* pp. 302-310, 1972.

[11] H. G. Kaper, G. K. Leaf and A. J. Lindeman, "A Timing Comparison Study for some High Order Finite Element Approximation Procedures and a Low Order Finite Difference Approximation Procedure for the Numerical Solution of the Multigroup Neutron Diffusion Equation," *Nuclear Science and Engineering: 49,* pp. 27-48, 1972.

[12] C. M. Kang and K. F. Hansen, "Finite Element Methods for Reactor Analysis," *Nuclear Science and Engineering: 51,* pp. 456-495, 1973.

[13] A. Hebert, "Application of the Hermite Method for Finite Element Reactor Calculations," *Nuclear Science and Engineering 91,* pp. 34-58, 1985.

[14] A. Herbert, "A Raviart-Thomas-Schneider solution of the diffusion equation in hexagonal geometry," *Annals of Nuclear Energy: 35,* pp. 363-376, 2008.

[15] S. Gonzalez-Pintor, D. Ginestar and G. Verdu, "High Order Finite Element Method for Lambda modes problem on hexagonal geometry," *Annals of Nuclear Energy 36,* pp. 1450-1462, 2009.

[16] Y. Wang, W. Bangerth and J. Ragusa, "Three-dimensional h-adaptivity for the multigroup neutron diffusion equations," *Progress in Nuclear Energy,* vol. 51, pp. 543-555, 2009.

[17] C. Phillipe, The Finite Element Method For Elliptic Problems, Amsterdam, New York, Oxford: North-Holland Publishing Company, 1978.

[18] P. Lax and A. Milgram, "Parabolic Equations," *Annals of Mathematics Studies,* vol. 33, pp. 167-190, 1954.

[19] J. Lions and G. Stampacchia, "Variational Inequalities," *Comm. Pure Appl. Math.,* vol. 20, pp. 493-519, 1967.

[20] C. Geuzaine and J. F. Remacle, "Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities," *International Journal for Numerical Methods in Engineering,* vol. 79, no. 11, pp. 1309-1331, 2009.

[21] G. Bell and S. Glasstone, Nuclear Reactor Theory, New York: Van Nostrand Reinhold Company, 1970.

[22] J. R. Lamarsh and A. Baratta, Introduction to Nuclear Engineering, 3, Ed., Upper Saddle River, NJ: Prentice Hall, 2001.

[23] E. Gelbard, "Application of the Simplified Spherical Harmonics Equation in Spherical Geometry," *WAPD-TM-294,* 1962.

[24] E. Larsen, J. Morel and J. McGhee, "Asymptotic derivation of the simplified Pn equations," in *Proceedings of Joint International Conference on Mathematical Methods and Supercomputing in Nuclear Applications*, Portland, Oregon , 1993.

[25] E. Larsen, J. Morel and J. McGhee, "Asymptotic derivation of the muligroup P1 and simplified Pn equations with anisotropic scattering," *Nuclear Science and Engineering* , vol. 123, no. 3, pp. 328-342, 1996.

[26] P. Brantley and E. Larsen, "The simplified P3 approximation," *Nuclear Science and Engineering,* vol. 134, pp. 1-21, 2000.

[27] C. Beckert and U. Grundman, "Development and verification of a nodal approach for solving the multigroup SP3 equations," *Annals of Nuclear Energy,* vol. 35, pp. 75-86, 2008.

[28] E. H. Ryu and H. G. Joo, "Finite element method solution of the simplified P3 equations for general geometry applications," *Annals of Nuclear Energy,* vol. 56, pp. 194-207, 2013.

[29] J. Leppanen, "The Serpent Monte Carlo code: Status, development and applications in 2013," *Annals of Nuclear Energy, 82,* pp. 142-150, 2015.

[30] J. Leppanen, Development of a New Monte Carlo Reactor Physics Code, Helsinki: VTT Publications, 2007.

[31] J. Leppanen, M. Pusa and E. Fridman, "Overview of methodology for spatial homogenization in the Serpent 2 Monte Carlo code," *Annals of Nuclear Energy 96,* pp. 126-136, 2016.

[32] B. R. Herman, B. Forget, K. Smith and B. Aviles, "Improved diffusion coefficients generated from Monte Carlo codes," in *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering,* Sun Valley, ID, 2013.

[33] B. Yee, E. Larsen and B. Kochunas, "An analytical derivation of transport-corrected P0 cross sections and diffusion coefficients," in *PHYSOR: Unifying Theory and Experiments in the 21st Century*, Sun Valley, ID, 2016.

[34] Z. Liu, K. Smith and B. Forget, "A cumulative migration method for computing rigorous transport cross sections and diffusion coefficients for LWR lattices with Monte Carlo," in *PHYSOR: Unifying Theory and Experiments in the 21st Century*, Sun Valley, ID, 2016.

[35] Argonne Code Center: Benchmark Problem Book, Vols. ANL-7416, Supp. 2 , Argonne National Laboratory, 1977.

[36] B. Richardson, C. H. Castano, J. King, A. Alajo and S. Usman, "Modeling and validation of approach to criticality and axial flux profile experiments at the Missouri S&T Reactor (MSTR)," *Nuclear Engineering and Design 245,* pp. 55-61, 2012.

[37] B. Richardson, J. King, A. Alajo, S. Usman and C. H. Giraldo, "Modeling and validation of Temperature and Void Effects on Reactivity Experiments at the Missouri S&T Reserach Reactor," *Nuclear Science and Engineering 187,* pp. 100-106, 2017.

[38] B. P. Richardson, "Verification of a Monte Carlo model of the Missouri S&T Reactor," Masters Theses. 6950, 2012. [Online]. Available: http://scholarsmine.mst.edu/masters_theses/6950.

[39] MATLAB, "Sparse Matrix Operations," MathWorks, 2018. [Online]. Available: https://www.mathworks.com/help/matlab/math/sparse-matrix-operations.html. [Accessed 19 February 2018].

[40] W. Bonzer, "Annual Progress Report 2016-2017 for the Missouri Science and Technology Reactor (License R-79, Docket No. 50-123)," 2017.

[41] L. J. Covington, "Neutronics study of the conversion of the University of Missouri-Rolla reactor to low enriched uranium fuel," Masters Theses. 718, 1989. [Online]. Available: http://scholarsmine.mst.edu/masters_theses/718.

[42] "Missouri S&T Nuclear Research Reactor," [Online]. Available: http://nuclear.mst.edu/research/reactor/. [Accessed 21 February 2018].

[43] J. L. Conlin, "Listing of Available ACE Data Tables," 30 January 2017. [Online]. Available: https://laws.lanl.gov/vhosts/mcnp.lanl.gov/pdf_files/la-ur-17-20709.pdf. [Accessed 22 February 2018].

[44] A. Yamamoto, "Simplified Treatments of Anisotropic Scattering in LWR Core Calculations," *Journal of Nuclear Science and Technology,* vol. 45, no. 3, pp. 217-229, 2012.

[45] D. Calic, G. Zerovnik, A. Trkov and L. Snoj, "Validation of the Serpent 2 code on TRIGA Mark II benchmark experiments," *Applied Radiation and Isotopes 107,* pp. 165-170, 2016.

[46] D. Ferraro and E. Villarina, "Full 3-D core calculations with refueling for the OPAL Research Reactor using Monte Carlo Code Serpent 2," *Annals of Nuclear Energy 92,* pp. 369-377, 2016.

[47] D. J. Siefman, G. Girardin, A. Rais, A. Pautz and M. Hursin, "Full Core modeling techniques for research reactors with irregular geometries using Serpent and PARCS applied to the CROCUS reactor," *Annals of Nuclear Energy 85,* pp. 434-443, 2015.

[48] G. Baiocco, A. Petruzzi, S. Bznuni and T. Kozlowski, "Analysis of a small PWR core with PARCS/Helios and PARCS/Serpent code systems," *Annals of Nuclear Energy 107,* pp. 42-48, 2017.

[49] V. P. Alferov, A. I. Radaev, M. V. Shchurovskaya, G. V. Tikhomirov and N. A. Hanan, "Comparitive validation of Monte Carlo codes for the conversion of a research reactor," *Annals of Nuclear Energy 77,* pp. 273-280, 2015.

VITA

Wayne J. Brewster was born in St. Louis, Missouri. In May 2016, he received his B.S. in Nuclear Engineering from the Missouri University of Science and Technology, Rolla, Missouri. In May of 2018, he received his M.S. in Nuclear Engineering from the Missouri University of Science and Technology, Rolla, Missouri.