Implementation and Verification of the Subgroup Decomposition Method in the TITAN 3-D Deterministic Radiation Transport Code

Nathan J. Roskoff

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Nuclear Engineering

Alireza Haghighat, Chair

Mark Pierson

Leigh Winfrey

May 7, 2014

Arlington, Virginia

Keywords: neutron transport theory, multigroup transport, subgroup decomposition method, energy condensation

Implementation and Verification of the Subgroup Decomposition Method in the TITAN 3-D Deterministic Radiation Transport Code

Nathan J. Roskoff

(ABSTRACT)

The subgroup decomposition method (SDM) has recently been developed as an improvement over the consistent generalized energy condensation theory for treatment of the energy variable in deterministic particle transport problems. By explicitly preserving reaction rates of the fine-group energy structure, the SDM directly couples a consistent coarse-group transport calculation with a set of fixed-source "decomposition sweeps" to provide a fine-group flux spectrum. This paper will outline the implementation of the SDM into the three-dimensional, discrete ordinates (S_N) deterministic transport code TITAN. The new version of TITAN, TITAN-SDM, is tested using 1-D and 2-D benchmark problems based on the Japanese designed High Temperature Engineering Test Reactor (HTTR). In addition to accuracy, this study examines the efficiency of the SDM algorithm in a 3-D S_N transport code.

This work was completed with support from the Department of Energy's Nuclear Energy University Program (NEUP) Award Number DE-AC07-05ID14517.

Acknowledgments

I would like to thank my advisor, Dr. Alireza Haghighat, for his guidance and support throughout this project. I also express my gratitude to my fellow graduate students, William Walters and Katherine Royston, whose patience and understanding have been invaluable.

I would also like to thank my family and friends for their continued encouragement and support. I credit much of my success to my mother and father; their love and support throughout my educational endeavors has been truly motivating.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Solving the linear Boltzmann neutron transport equation (LBE) requires knowing the density of neutrons within the phase-space, i.e., at every position, energy, and direction. The LBE cannot be analytically solved for any realistic problem and thus requires a novel numerical strategy which allows the phase-space to be discretized. A sufficiently accurate solution for a large nuclear system (i.e. reactor core or shield problems) requires a finely resolved phase-space which demands an extraordinary amount of computational resources, i.e. memory and time, that can be beyond the capabilities of today's computers.

Methods for reduced model complexity generally address each parameter of the phase-space independently. The spatial variables are addressed by homogenization of material complexities and development of advanced differencing schemes. Angular dependency is handled by the use of the

discrete ordinates approach, S_N, or the use of orthogonal polynomial approaches, such as P_N[1, 2]. The energy dependency is handled almost exclusively using multigroup theory, where the continuous energy spectrum is split into "groups" in which average cross sections

are obtained by preserving reaction rates within each group [1]. The majority of advances in the field have improved the treatment of the spatial and angular variables, with limited activities in the area of energy treatment.

Appropriate selection of group structure is necessary to ensure accurate core calculations; the group structure must be chosen to preserve the physical properties within a given energy group [3]. For the most efficient transport calculation, the fewest number of coarse groups which maintain an acceptable level of accuracy is desirable. This requires that the group structure is carefully selected such that reaction rates are preserved over the condensed or coarse group [4, 5, 6]. Also, in the standard multigroup formulation, the energy-angle coupling is ignored when condensing the total cross-section; the angular flux is considered to be separable in energy and angle [2]. This approximation becomes less valid is the flux is strongly varying [7].

Recently there have been several papers that allow for a detailed flux spectrum to be obtained from a coarse-group calculation [7, 8, 9]. These methods use a set of orthogonal energy expansion functions to capture the energy dependency of the angular flux. The subgroup decomposition method (SDM) shows that, by directly preserving the fine-group reaction rates, detailed energy expansion functions are unnecessary in the multigroup transport environment [10]. This method was found through the derivation of the consistent generalized energy condensation theory [7] which considers that the energy-angle coupling of the condensed cross-section is fully accounted for by the introduction of a "multigroup correction term" [11] The subgroup decomposition method allows for a fine-group flux spectrum to be extracted from a coarse-group transport calculation. This detailed spectrum is obtained by performing a single pass, fixed-source transport sweep, using a set of "subgroup decomposition cross sections" which represent the source in a given fine-group from either fission or scattering from the coarse-group in which they are contained.

Douglass and Rahnema (2012) have implemented and tested the SDM algorithm in a 1-D S_N transport code [10]. They have shown that the algorithm is capable of producing accurate results as well as improved computational efficiency. The SDM was tested on a simplified 1-D BWR core and the 1-D Japanese High Temperature Engineering Test Reactor (HTTR)[12] core. The 1-D calculations showed speedups of around 6.0 for the BWR core and around 1.5 for the HTTR core [10].

This thesis will present the detailed SDM formulations, outline the implementation of the SMD [13] in the TITAN 3-D radiation transport code [14, 15], referred to as TITAN-SDM, and present results from testing TITAN-SDM on a 1-D and 2-D HTTR benchmark problem. The HTTR benchmark problem is of a Gen-IV type reactor; the core and fuel assemblies are comprised of hexagonal geometry. It is first necessary to prove the TITAN code can accurately model hexagonal geometry; therefore, TITAN performance is evaluated for modeling a HTTR 2-D hexagonal fuel assembly [16]. Once the standard TITAN code is proven to accurately model hexagonal geometry, the performance of TITAN-SDM will be evaluated on a one- and two-dimensional HTTR core. The SDM algorithm must be proven capable of working within a three-dimensional environment to be regarded a true asset to the community.

This thesis is organized as follows: Chapter 2 formally introduces the LBE for radiation transport and discusses numerical solution of the equation. Chapter 2 also discusses the power iteration method for solving a criticality problem and introduces the SDM. Chapter 3 shows how the SDM is implemented into the TITAN 3-D radiation transport code. Chapter 4 provides metrics for evaluating the accuracy and performance of the algorithm as well as presents two benchmark problems and reference multigroup calculations. Chapter 5 details the results of testing the TITAN-SDM algorithm on the different benchmark problems. Chapter 6 summaries the findings of this study and suggests further work.

Chapter 2

Theory

2.1 Radiation Transport Methods

Solving for the angular flux requires either the use of a stochastic process, such as Monte Carlo methods, or direct solution of the Linear Boltzmann Equation (LBE). This research will focus on the direct solution of the LBE.

2.1.1 Linear Boltzmann Equation (LBE)

The time-independent LBE, Equation (2.1), is a particle balance equation in phase space $\left(d^3rdEd\Omega\right)$, at position $\underline{r}$, direction $\hat{\Omega}$, and energy E, Figure 2.1 [2]. The left-hand side of the equation represents the total losses and the right-hand side represents the total production.

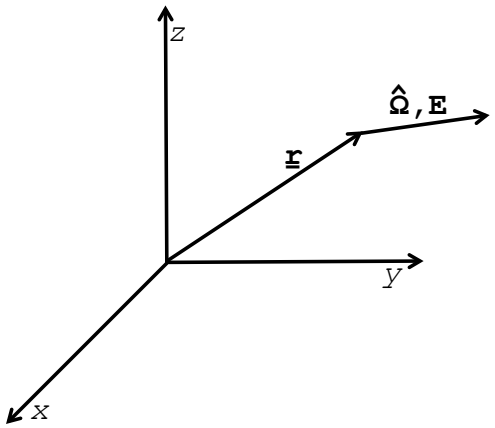


Figure 2.1 Schematic representation of phase space.

$$\underbrace{\hat{\Omega} \cdot \nabla \Psi(\underline{r}, \hat{\Omega}, E)}_{(1)} + \underbrace{\sigma(\underline{r}, E)\Psi(\underline{r}, \hat{\Omega}, E)}_{(2)} = \underbrace{S(\underline{r}, \hat{\Omega}, E)}_{(3)} \tag{2.1}$$

The first term is the particle streaming or leakage within a phase space. The second term defines collisions that remove the particle from the phase space, through either absorption or scattering to another angle or energy. The third term is the total production of particles within the phase space. $\Psi(\underline{r}, E, \hat{\Omega})$ is the angular flux , $\sigma(\underline{r}, E)$ is the total macroscopic cross-section, and $S(\underline{r}, \hat{\Omega}, E)$ is the total source.

Equation (2.2) shows the unit direction $\hat{\Omega}$ as defined in terms of polar (θ) and azimuthal (φ) angles.

$$\hat{\Omega} = \langle \theta, \varphi \rangle = \langle \cos(\theta), \sin(\theta)\cos(\varphi), \sin(\theta)\sin(\varphi) \rangle \tag{2.2}$$

The solid angle is defined as

$$d\Omega = \sin\theta d\theta d\varphi \tag{2.3}$$

or

$$d\Omega = d\mu d\varphi, \tag{2.4}$$

where $\mu = \cos\theta$. The solid angle $d\Omega$ will be normalized on the unit sphere, i.e.,

$$\int_{4\pi} d\hat{\Omega} = \int_{-1}^{1} \frac{d\mu}{2} \int_{0}^{2\pi} \frac{d\varphi}{2\pi} = 1, \tag{2.5}$$

and will hereafter be implicitly assumed.

The source S represents the total production of particles from independent sources, scattering, and fission. Equation (2.6) shows the form of the source

$$S(\underline{r}, \hat{\Omega}, E) = \int_{0}^{\infty} dE' \int_{4\pi} d\Omega' \sigma_s(\underline{r}, E' \to E, \mu_0)\Psi(\underline{r}, \hat{\Omega}, E) + \frac{\chi(\underline{r}, E)}{k} \int_{0}^{\infty} dE' \nu\sigma_f(\underline{r}, E')\phi(\underline{r}, E'), \tag{2.6}$$

where $\sigma_s(\underline{r}, E' \to E, \mu_0)$ is the differential macroscopic scattering cross-section for a particle of energy E' and traveling about $\hat{\Omega}'$ scattering into dE about energy E and solid angle $d\Omega$ about $\hat{\Omega}$, $\mu_0 = \hat{\Omega} \cdot \hat{\Omega}'$, k represents the system multiplication factor, $\nu\sigma_f(\underline{r}, E')$ is the fission cross-section multiplied by the number of fission neutrons produced per fission, and $\chi(\underline{r}, E)$ is the fission spectrum, i.e, the probability density function for fission neutron spectrum. Equation (2.7) defines the scalar flux, $\phi(\underline{r}, E)$.

$$\phi(\underline{r}, E) = \int_{4\pi} d\Omega \Psi(\underline{r}, E, \hat{\Omega}) \tag{2.7}$$

The first term in Equation (2.6) represents particles from all other energies and angles ($dE'd\hat{\Omega}'$) that scatter into the phase space of interest. The second term represents all neutrons born at energy dE' from a fission process that enter into the energy region of interest.

2.1.2 Numeric Solution of the LBE

For numeric solution of the LBE, all independent variables (space, angle, and energy) that define the phase space must be discretized. The following sections will detail the discretization process for each independent variable.

2.1.2.1 Energy, E

Energy is generally discretized using the multigroup approximation [2], which divides the energy spectrum into a set of G energy groups ($g = 1, ..., G$) within which the flux is defined as an integral, over energy E, with quantity and material properties are assumed constant. The groups are defined such that the lowest energy is E_G and the highest energy is E_1. The multigroup formulation is obtained by integrating the LBE over the energy bounds of group g. For brevity, Equation (2.8) presents a notation for this integral that will be used hereafter.

$$\int_g dE \equiv \int_{E_g}^{E_{g-1}} dE \tag{2.8}$$

Equation (2.9) presents the multigroup form of the transport equation

$$\hat{\Omega} \cdot \nabla \psi_g(\underline{r}, \hat{\Omega}) + \sigma_g(\underline{r}) \psi_g(\underline{r}, \hat{\Omega}) = S_g(\underline{r}, \hat{\Omega}), \tag{2.9}$$

where the ψ_g is the in-group angular flux

$$\psi_g(\underline{r}, \hat{\Omega}) = \int_g dE \Psi(\underline{r}, \hat{\Omega}, E), \tag{2.10}$$

σ_g is the total group cross-section

$$\sigma_g(\underline{r}) = \frac{\int_g dE\sigma(\underline{r}, E)\phi(\underline{r}, E)}{\phi_g(\underline{r})}, \tag{2.11}$$

where

$$\phi_g(\underline{r}) = \int\limits_g dE \int\limits_{4\pi} d\Omega\Psi(\underline{r}, \hat{\Omega}, E), \tag{2.12}$$

and S_g is the standard multigroup source

$$S_g(\underline{r}, \hat{\Omega}) = \sum_{g'=1}^{G} \int\limits_{4\pi} d\Omega'\sigma_{s,g'\to g}(\underline{r}, \mu_0)\psi_{g'}(\underline{r}, \hat{\Omega}) + \frac{\chi_g(\underline{r})}{k} \sum_{g'=1}^{G} \nu\sigma_{f,g'}(\underline{r})\phi_{g'}(\underline{r}). \tag{2.13}$$

The multigroup coefficients are defined in Equations (2.14) - (2.16).

$$\sigma_{s,g'\to g}(\underline{r}, \mu_0) = \frac{\int_g dE \int_{g'} dE'\sigma_s(\underline{r}, E' \to E, \mu_0)\phi(\underline{r}, E')}{\phi_{g'}(\underline{r})}, \tag{2.14}$$

$$\chi_g(\underline{r}) = \int\limits_g dE\chi(\underline{r}, E), \tag{2.15}$$

and

$$\nu\sigma_{f,g}(\underline{r}) = \frac{\int_g dE\nu\sigma_f(\underline{r}, E)\phi(\underline{r}, E)}{\phi_g(\underline{r})}. \tag{2.16}$$

2.1.2.2 Multigroup Cross-Section Generation

In order to use the multigroup form of the LBE one must determine the multigroup constants, or cross sections. By observing Equations (2.14) and (2.16) it is clear that the angular flux must be known to evaluate the multigroup cross sections; the angular flux is the specific quantity of interest when solving the LBE and is therefore not known *a priori*. A large amount of time and effort has gone into determining adequate flux spectra to use to

generate condensed cross sections which sufficiently reproduce results of continuous energy calculations. Typically, a three-step approach is used to generate broad group cross-section libraries.

The first step is to obtain the point-wise, "continuous", cross-section library from ENDF/B-VII [17] (Evaluated Nuclear Data File) or others like it. The values in these databases have been obtained by empirical relations and years of experimentation. A processing code, such as NJOY [18], uses weighting factors based on the energy range of the cross-section to collapse the point-wise constants into fine groups; fine-group energy bounds must be defined. The weighting factors used are as follows: for fast neutrons, energy greater than 0.1 MeV, the fission spectrum, $\chi(E)$, is used; for epithermal neutrons, energy below 0.1 MeV and typically above 0.1[1] eV, a $1/E$ spectrum is used; for thermal neutrons, energy less than 0.625 eV, a Maxwellian, $M(E)$ spectrum is used. The fine-group library typically contains 100s of values for shielding calculations and 10s of values for core physics problems. Once fine-group cross-sections are obtained, a simplified model of the problem, for shielding a 1-D radial core model is used and for core physics an assembly model is used, is solved and the calculated fluxes are used to collapse the point-wise cross sections into a broad-group library; broad-group energy bounds must be defined. This final broad-group library typically contains 10s of values for shielding calculations and less than 10 values for a core physics problem.

Once the broad-group library is obtained a more detailed, multidimensional model of the problem is developed and solved, using collision probability method (CPM) for core physics problem and S_N for shielding problem. Reaction rates are calculated using the calculated fluxes and the collapsed broad-group cross sections and compared to experimental results. If the results from this comparison are acceptable, these cross sections are considered accept-

[1]This value depends on the temperature of the system.

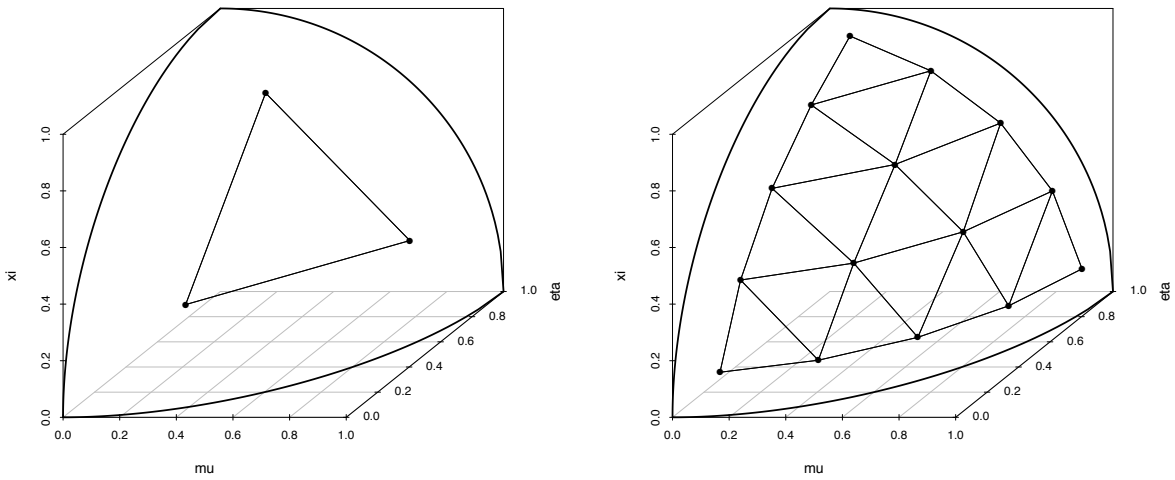
able; if not, the process is started again with a different selection of group energy bounds. It is important to note that this process is very sensitive to the selection of group energy bounds.

2.1.2.3 Angular Quadrature and Scattering Expansion

The angular domain is discretized into a set of M distinct directions $\hat{\Omega}_m$, over which the transport equation is solved. This is known as the discrete ordinates or S_N method [2]. We translate all integrals over $\hat{\Omega}$ into sums over the distinct angles $\hat{\Omega}_m$ as

$$\int_{4\pi} d\Omega f(\hat{\Omega}) = \sum_m \omega_m f(\hat{\Omega}_m) \tag{2.17}$$

where w_m is a weight corresponding to direction $\hat{\Omega}_m$. The combination of $\left\{w_m, \hat{\Omega}_m\right\}$ is referred to as the angular quadrature set. The set is chosen such that integral quantities are preserved. The common approach is to preserve symmetry of directions and integral of directions cosine (directions cosine defined in Equations (2.23) and (2.24)). A technique for generation of a quadrature set is the level-symmetric approach [19], in which N refers to the number of levels and the total number of directions, M, is calculated by $N(N+1)$. The level-symmetric approach is limited to S_{20}, so other techniques such as Legendre-Chebysheve (P_N-T_N) have been developed [20]. Figure 2.2 shows a P_N-T_N quadrature for orders 4 and 10, within a single octant.



(a) S_4, $M = 24$ directions. **(b)** S_{10}, $M = 120$ directions.

Figure 2.2 P_N-T_N quadrature sets for a single octant.

Considering the orthogonality of the Legendre Polynomials, i.e.

$$\int_{-1}^{1} d\mu P_n(\mu) P_m(\mu) = \frac{2}{2n+1}\delta_{nm},$$

where

$$\delta_{nm} = \begin{cases} 1, & n = m, \\ 0, & n \neq m, \end{cases}$$

then the scattering cross section can be expanded in terms of Lengendre Polynomial P_l as

$$\sigma_{s,g'\to g}(\underline{r}, \mu_0) = \sum_{l=0}^{L} (2l+1)\sigma_{sl,g'\to g}(\underline{r}) P_l(\mu_0). \tag{2.18}$$

The moments of the scattering cross-section are obtained by

$$\sigma_{sl,g'\to g}(\underline{r}) = \int_{-1}^{1} \frac{d\mu_0}{2} \sigma_{s,g'\to g}(\underline{r},\mu_0) P_l(\mu_0). \tag{2.19}$$

The Legendre series of Equation (2.18) is truncated at order L and is therefore referred to as the P_L expansion to the scattering cross-section. Equation (2.20) defines the scattering angle

$$\mu_0 = \mu\mu' + \sqrt{1-\mu^2}\sqrt{1-\mu'^2}\cos(\varphi-\varphi'). \tag{2.20}$$

The Legendre polynomial is represented using spherical harmonics $Y_{l,k}$ and $Y^*_{l,k}$ with the Legendre Addition Theorem, Equation (2.21).

$$P_l(\mu_0) = \frac{1}{(2l+1)} \sum_{k=-l}^{l} Y^*_{l,k}(\theta',\varphi') Y_{l,k}(\theta,\varphi) \tag{2.21}$$

The spherical harmonics are defined in terms of the Associated Legendre polynomials P^k_l, therefore

$$P_l(\mu_0) = P_l(\mu)P_l(\mu') + 2\sum_{k=1}^{l} \frac{(l-k)!}{(l+k)!} P^k_l(\mu) P^k_l(\mu') \cos(k(\varphi-\varphi')) \tag{2.22}$$

Also note that there exists a set of direction cosines projected onto the x, y, z axes, represented as $\langle \mu, \eta, \xi \rangle$. These can be expressed in terms of polar and azimuthal angles, as in Equations (2.23) and (2.24).

$$\eta = \sqrt{1-\mu^2}\cos(\varphi) \tag{2.23}$$

$$\xi = \sqrt{1-\mu^2}\sin(\varphi) \tag{2.24}$$

These definitions allow us to expand the streaming operator $\hat{\Omega}\cdot\nabla$ in 3D Cartesian coordinates,

Equation (2.25).

$$\hat{\Omega} \cdot \nabla = \mu_m \frac{\partial}{\partial x} + \eta_m \frac{\partial}{\partial y} + \xi_m \frac{\partial}{\partial z} \tag{2.25}$$

By substituting (2.5),(2.22), and (2.25) in the the LBE (2.1), we can write the Legendre expanded discrete ordinates multigroup form of the LBE in Cartesian geometry considering only fission sources, Equation (2.26).

$$\begin{aligned}
&\left(\mu_m \frac{\partial}{\partial x} + \eta_m \frac{\partial}{\partial y} + \xi_m \frac{\partial}{\partial z}\right) \psi_g(x,y,z,\mu_m,\varphi_m) + \sigma_g(x,y,z)\psi_g(x,y,z,\mu_m,\varphi_m) \\
&= \sum_{g'=1}^{G} \sum_{l=1}^{L} (2l+1)\sigma_{sl,g'\to g}(x,y,z) \left\{ P_l(\mu_m)\phi_{g',l}(x,y,z) + 2\sum_{k=1}^{l} \frac{(l-k)!}{(l+k)!} P_l^k(\mu_m) \cdot \right. \\
&\qquad \left. \left[\phi^k_{C,g',l}(x,y,z)\cos(k\varphi_m) + \phi^k_{S,g',l}(x,y,z)\sin(k\varphi_m)\right] \right\} \\
&\qquad + \frac{\chi_g}{k} \sum_{g'=1}^{G} \nu\sigma_{f,g'}(x,y,z)\phi_{g',0}(x,y,z)
\end{aligned} \tag{2.26}$$

Note that all of the terms for the above discrete ordinates formulation of the transport equation are outlined in Table 2.1.

Table 2.1 LBE Discrete Ordinates Notation.

Variable	Description
μ_m	x direction cosine for direction m
η_m	y direction cosine for direction m
ξ_m	z direction cosine for direction m
ψ_g	angular flux for group g (for $g = 1, ..., G$)
φ_m	azimuthal angle constructed from $\arctan(\xi/\eta)$
σ_g	total macroscopic cross-section for group g
l	Legendre expansion ($l = 0, L$), $L = 0$ or odd truncation
$\sigma_{sl,g'\to g}$	l^{th} Legendre moment of the macroscopic differential scattering cross-section from group g' to g
$P_l(\mu)$	l^{th} Legendre polynomial
$\phi_{g',l}$	l^{th} Legendre scalar flux moment for group g'
$P_l^k(\mu)$	l^{th}, k^{th} Associated Legendre polynomial
$\phi_{C,g',l}^k$	l^{th}, k^{th} Cosine Associated Legendre polynomial for group g'
$\phi_{S,g',l}^k$	l^{th}, k^{th} Sine Associated Legendre polynomial for group g'

The flux moments $\phi_{g',l}$, $\phi_{C,g',l}^k$, and $\phi_{S,g',l}^k$ are defined in Equations (2.27),(2.28), and (2.29).

$$\phi_{g',l}(x,y,z) = \int_{-1}^{1} \frac{d\mu'}{2} P_l(\mu') \int_0^{2\pi} \frac{d\varphi'}{2\pi} \psi_{g'}(x,y,z,\mu',\varphi') \tag{2.27}$$

$$\phi_{C,g',l}^k(x,y,z) = \int_{-1}^{1} \frac{d\mu'}{2} P_l^k(\mu') \int_0^{2\pi} \frac{d\varphi'}{2\pi} \cos(k\varphi') \psi_{g'}(x,y,z,\mu',\varphi') \tag{2.28}$$

$$\phi_{S,g',l}^k(x,y,z) = \int_{-1}^{1} \frac{d\mu'}{2} P_l^k(\mu') \int_0^{2\pi} \frac{d\varphi'}{2\pi} \sin(k\varphi') \psi_{g'}(x,y,z,\mu',\varphi') \tag{2.29}$$

The Spherical Harmonics expanded group angular flux, Equation (2.30).

$$\psi_g(x,y,z,\mu,\varphi) \approx \sum_{l=0}^{L}(2l+1)\left\{P_l(\mu)\phi_{g,l}(x,y,z) + 2\sum_{k=1}^{l}\frac{(l-k)!}{(l+k)!}P_l^k(\mu)\left[\phi_{C,g,l}^k(x,y,z)\cos(k\varphi_n)+\phi_{S,g,l}^k(x,y,z)\sin(k\varphi_n)\right]\right\} \tag{2.30}$$

2.1.2.4 Spatial Differencing

Space is discretized based on the finite-volume approach. Here, the cell position is defined by (i,j,k) and volume by $(\Delta x, \Delta y, \Delta z)$ with cell boundaries as defined in Equation (2.31). Note that the g and m subscripts have been dropped on the flux terms for brevity; all of the fluxes presented in the equations of this section represent the flux of group g and direction $\hat{\Omega}_m$.

$$\begin{aligned} \psi_{i-1/2,j,k} \leq x \leq \psi_{i+1/2,j,k} \\ \psi_{i,j-1/2,k} \leq y \leq \psi_{i,j+1/2,k} \\ \psi_{i,j,k-1/2} \leq z \leq \psi_{i,j,k+1/2} \end{aligned} \tag{2.31}$$

For simplicity incoming boundary fluxes, for example in x, will be defined such that $\psi_{in,x} \equiv \psi_{i-1/2,j,k}(x,y,z)$ and similarly for outgoing boundary fluxes $\psi_{out,x} \equiv \psi_{i+1/2,j,k}(x,y,z)$. Also for simplicity, spatial integrals will be defined as in Equation (2.32).

$$\int_{\Delta x} dx \equiv \int_{i-1/2,j,k}^{i+1/2,j,k} dx \tag{2.32}$$

An integration of the LBE over the spatial cells eliminates the derivative term, as in Equation (2.33).

$$\begin{aligned}\frac{\mu_n}{\Delta x_i}\left(\psi_{out,x}-\psi_{in,x}\right)+\frac{\eta_n}{\Delta y_j}\left(\psi_{out,y}-\psi_{in,y}\right)+\frac{\xi_n}{\Delta z_k}\left(\psi_{out,z}-\psi_{in,z}\right)\\ +\sigma_t(x,y,z)\psi_A(x,y,z)=S_A(x,y,z)\end{aligned} \tag{2.33}$$

This integration leads to cell averaged fluxes ψ_A, defined in Equation (2.34). Cell averaged sources S_A are defined in a similar manner.

$$\psi_A=\frac{1}{\Delta x\Delta y\Delta z}\int_{\Delta z}dz\int_{\Delta y}dy\int_{\Delta x}dx\psi_{n,g}(x,y,z)\overbrace{P_0(x)P_0(y)P_0(z)}^{1} \tag{2.34}$$

Using this approximation there are 6 surface fluxes in cell (i,j,k), three of these fluxes are known due to boundary conditions and/or neighboring fluxes, e.g. when sweeping along incoming directions, the incoming fluxes are $\psi_{in,x}$, $\psi_{in,y}$, $\psi_{in,z}$. To obtain the others, three other equations are needed; these equations are referred to as the differencing scheme. For example, one such method is known as Diamond Differencing and is defined in Equation (2.35).

$$\begin{aligned}\psi_A &= \frac{1}{2}\left(\psi_{in,x}+\psi_{out,x}\right)\\ &= \frac{1}{2}\left(\psi_{in,y}+\psi_{out,y}\right)\\ &= \frac{1}{2}\left(\psi_{in,z}+\psi_{out,z}\right)\end{aligned} \tag{2.35}$$

These three additional equations allow one to solve for the three unknown outgoing boundary fluxes. These simple approximations will suffice if the flux gradient across the cell is not severe. If the flux gradient is too large, alternative, more sophisticated methods can be used [21, 22].

2.2 Criticality Calculation

The algorithm described here is used to calculate the system criticality eigenvalue, k .

The LBE, Equation (2.1), may be written in operator form as

$$\mathbf{H}\Psi = \frac{1}{k}\mathbf{F}\Psi, \tag{2.36}$$

where

$$\mathbf{H} = \hat{\Omega}\cdot\nabla + \sigma(\underline{r}) - \underbrace{\int_0^\infty dE' \int_{4\pi} d\Omega'\ \sigma_s(\underline{r}, E' \rightarrow E, \mu_0)}_{\mathbf{M}} \tag{2.37}$$

and

$$\mathbf{F} = \chi(E)\int_0^\infty dE' \int_{4\pi} d\Omega'\ \nu\sigma_f(\underline{r}, E'). \tag{2.38}$$

Note that the total source used to calculate the angular flux is defined as $S_{tot} = S_{scat} + S_{fiss}$, where $S_{scat} = \mathbf{M}\Psi$ and $S_{fiss} = \mathbf{F}\Psi$.

The standard method for solving a criticality problem in a deterministic environment is to use the source-iteration method [23]. This method is described by the following steps: (1) initialize flux, k and source due to fission, (2) update source due to scattering and calculate total source, (3) update the flux with Equation (2.36), (4) check for convergence of flux (if not converged go back to Step 2) , (5) use the converged flux, calculate an updated fission source and k , and (6) check for convergence of k (if k is not converged, calculate total source and go to Step 2). Iterate on this process until the system converges. Note that Steps 2 and 3 are commonly referred to as the *inner* iteration and Steps 4 and 5 are referred to as the *outer* iteration. Figure 2.3 shows the source-iteration scheme.

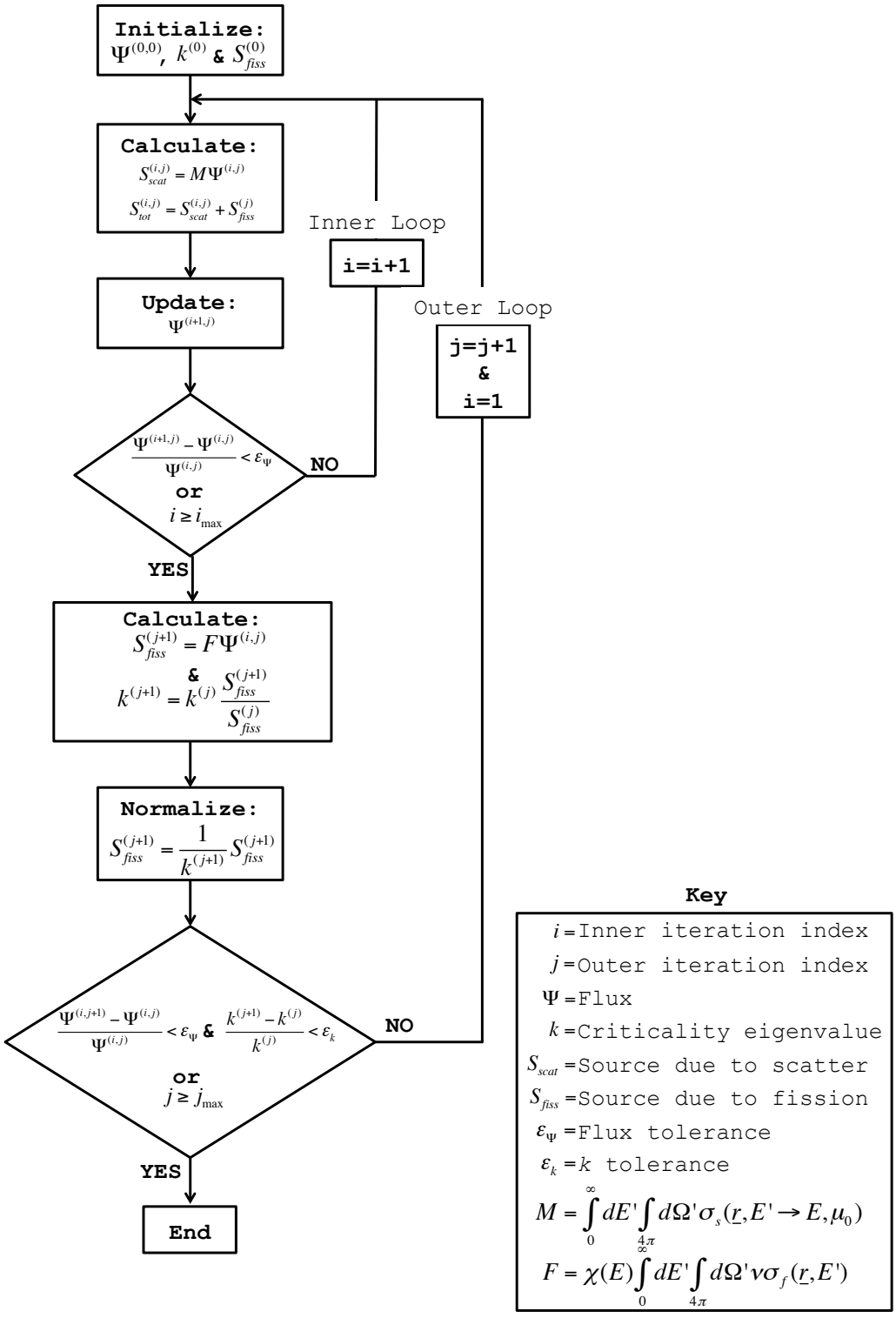


Figure 2.3 Source-iteration method to determine the system criticality eigenvalue.

2.3 TITAN

The TITAN code [14, 15] is a hybrid deterministic transport code, capable of parallel computation, that iteratively solves the multigroup LBE in a 3-D Cartesian environment. The spatial discretization is comprised of coarse and fine meshing. Each coarse mesh is comprised of a uniform set of fine meshes; different coarse meshes can have different sized fine meshes. TITAN is a hybrid code because of its ability to use different LBE solvers in different regions or coarse meshes of the problem. These solvers are (i) the discrete ordinates (S_N) method and (ii) the method of characteristics (CM). In this paper, only the S_N capabilities of TITAN are considered.

2.4 Subgroup Decomposition Method (SDM)

The subgroup decomposition method or SDM is a novel method, developed by Steven Douglass [10], that is used to address the two primary difficulties in the standard group collapsing process. The first difficulty is that the weighting flux used to collapse material cross sections is typically from a lattice-cell calculation that does not necessarily represent the specific core environment. The second is that the group collapsing process is typically a one-way process; when cross sections are collapsed into just a few groups the solution is obtained for this coarse-group structure and a detailed flux spectrum is not available. The SDM can extract detailed fine-group fluxes from a coarse-group transport calculation.

2.4.1 Consistent Multigroup Formulation

An important aspect of the SDM formulation is the fact that it accounts for the energy-angle coupling effect that is typically neglected in standard multigroup transport, Equation (2.9). Therefore, the SDM utilizes an alternative form of the multigroup transport equation known as the consistent multigroup formulation, derived below. Starting with the LBE, Equation (2.1) and integrating over the energy bounds of group g (Equation (2.8))

$$\int_g dE\ \hat{\Omega}\cdot\nabla\Psi(\underline{r},\hat{\Omega},E)+\int_g dE\ \sigma(\underline{r},E)\Psi(\underline{r},\hat{\Omega},E)=\int_g dE\ S(\underline{r},\hat{\Omega},E).$$

The above expression shows that the total collision cross-section should be angular dependent due to the angular flux weighting factor. In standard multigroup transport, Equation (2.9), this angular dependency is typically neglected because the scalar flux is used as the weighting factor. To form a multigroup transport equation which accounts for this angular dependency, the typical multigroup collision term is added and subtracted from the right-hand side as follows

$$\hat{\Omega}\cdot\nabla\psi_g(\underline{r},\hat{\Omega})+\sigma_g(\underline{r},\hat{\Omega})\psi_g(\underline{r},\hat{\Omega})+\sigma_g(\underline{r})\psi_g(\underline{r},\hat{\Omega})-\sigma_g(\underline{r})\psi_g(\underline{r},\hat{\Omega})=S_g(\underline{r},\hat{\Omega}), \tag{2.39}$$

which can be rearranged to

$$\hat{\Omega}\cdot\nabla\psi_g(\underline{r},\hat{\Omega})+\sigma_g(\underline{r},\hat{\Omega})\psi_g(\underline{r},\hat{\Omega})+\sigma_g(\underline{r})\psi_g(\underline{r},\hat{\Omega})=S_g(\underline{r},\hat{\Omega})+\sigma_g(\underline{r})\psi_g(\underline{r},\hat{\Omega})-\sigma_g(\underline{r},\hat{\Omega})\psi_g(\underline{r},\hat{\Omega}). \tag{2.40}$$

Upon further simplification we arrive at the "corrected" multigroup transport equation, one which accounts for the angular dependency of the total group cross-section,

$$\hat{\Omega} \cdot \nabla \psi_g(\underline{r}, \hat{\Omega}) + \sigma_g(\underline{r}) \psi_g(\underline{r}, \hat{\Omega}) = S_g(\underline{r}, \hat{\Omega}) + \delta_g(\underline{r}, \hat{\Omega}), \tag{2.41}$$

where g refers to the fine-group index, $\psi_g(\underline{r}, \hat{\Omega})$ is the angular flux at position $\underline{r}$ and direction $\hat{\Omega}$, Equation (2.10), σ_g is the total macroscopic cross-section, Equation (2.11), S_g is the total multigroup source, Equation (2.13), and δ_g is the multigroup correction term given by

$$\delta_g(\underline{r}, \hat{\Omega}) = \int_g dE \left(\sigma_g(\underline{r}) - \sigma(\underline{r}, E) \right) \psi(\underline{r}, \hat{\Omega}, E) \tag{2.42}$$

The above formulation provides the difference in group reaction rate, the "standard" group cross-section [1] versus the angular-dependent group cross-section.

2.4.2 Strategy

The SDM can be broken into three distinct steps. First, a detailed fine-group flux spectrum is obtained and used to collapse user input fine-group material cross sections into a coarse-group library. Then, a coarse-group transport calculation is completed to obtain coarse-group fluxes. Using these coarse-group fluxes and a set of "subgroup decomposition cross sections," a fine-group "decomposition sweep" is performed to extract the fine-group fluxes. This process is repeated until a converged solution is obtained.

Equation (2.43) shows the fine-group transport equation used to obtain fine-group fluxes

$$\hat{\Omega} \cdot \nabla \psi_g(\underline{r}, \hat{\Omega}) + \sigma_g(\underline{r}) \psi_g(\underline{r}, \hat{\Omega}) = \sum_{g'=1}^{G} \int_{4\pi} d\Omega' \sigma_{s,g' \to g}(\underline{r}, \mu_0) \psi_{g'}(\underline{r}, \hat{\Omega}), + \frac{\chi_g(\underline{r})}{k} \sum_{g'=1}^{G} \nu \sigma_{f,g'}(\underline{r}) \phi_{g'}(\underline{r}) \tag{2.43}$$

where ψ_g is the fine-group angular flux and ϕ_g is the fine-group scalar flux. Constants for the above equation are obtained from the user input fine-group cross-section library.

Once fine-group fluxes are obtained, the coarse-group transport can be calculated using

$$\hat{\Omega}\cdot\nabla\psi_h(\underline{r},\hat{\Omega})+\sigma_h(\underline{r})\psi_h(\underline{r},\hat{\Omega})=\sum_{h'=1}^{H}\int_{4\pi}d\Omega'\sigma_{s,h'\rightarrow h}(\underline{r},\mu_0)\psi_{h'}(\underline{r},\hat{\Omega}) + \frac{\chi_h(\underline{r})}{k}\sum_{h'=1}^{H}\nu\sigma_{f,h'}(\underline{r})\phi_{h'}(\underline{r})+\delta_h(\underline{r},\Omega), \tag{2.44}$$

where ψ_h is the coarse-group angular flux and ϕ_h is the coarse-group scalar flux. The necessary coarse-group constants are generated from the fine-group cross-section library and previously obtained fine group fluxes as follows

$$\sigma_h(\underline{r})=\frac{\sum_{h\in g}\sigma_g(\underline{r})\phi_g(\underline{r})}{\sum_{h\in g}\phi_g(\underline{r})}, \tag{2.45}$$

$$\sigma_{s,h'\rightarrow h}(\underline{r},\mu_0)=\frac{\sum_{h\in g}\sum_{h'\in g'}\sigma_{s,g'\rightarrow g}(\underline{r},\mu_0)\psi_{g'}(\underline{r},\hat{\Omega})}{\sum_{h'\in g'}\psi_{g'}(\underline{r},\hat{\Omega})}, \tag{2.46}$$

$$\chi_h(\underline{r})=\sum_{h\in g}\chi_g(\underline{r}), \tag{2.47}$$

$$\nu\sigma_{f,h}(\underline{r})=\frac{\sum_{h\in g}\nu\sigma_{f,g}(\underline{r})\phi_g(\underline{r})}{\sum_{h\in g}\phi_g(\underline{r})}, \tag{2.48}$$

and

$$\delta_h(\underline{r},\hat{\Omega})=\sigma_h(\underline{r})-\frac{\sum_{h\in g}\sigma_g(\underline{r})\psi_g(\underline{r},\hat{\Omega})}{\sum_{h\in g}\psi_g(\underline{r},\hat{\Omega})}. \tag{2.49}$$

Using the coarse-group fluxes, a fine-group decomposition transport calculation is com-

pleted,Equation (3.10).

$$\hat{\Omega} \cdot \psi_g(\underline{r}, \hat{\Omega}) + \sigma_g(\underline{r})\psi_h(\underline{r}, \hat{\Omega}) = \sum_{h'=1}^{H} \int_{4\pi} d\Omega' \; R_{s,h'\to g}(\underline{r}, \mu_0)\psi_{h'}(\underline{r}, \hat{\Omega}') + \frac{\chi_g(\underline{r})}{k} \sum_{h'=1}^{H} R_{f,h'\to g}(\underline{r})\phi_{h'}(\underline{r}) \tag{2.50}$$

The above fine-group decomposition sweep requires a source constructed by decomposing the coarse-group sources into fine-group sources; this is accomplished using "subgroup decomposition cross sections," Equations (2.51) and (2.52) [11].

$$R_{s,h'\to g}(\underline{r}, \mu_0) = \frac{\sum_{g'\in h'} \sigma_{s,g'\to g}(\underline{r}, \mu_0)\psi_{g'}(\underline{r}, \hat{\Omega})}{\sum_{g'\in h'} \psi_{g'}(\underline{r}, \hat{\Omega})} \tag{2.51}$$

$$R_{f,h'\to g}(\underline{r}) = \frac{\sum_{g'\in h'} \nu\sigma_{f,g'}(\underline{r})\phi_{g'}(\underline{r})}{\sum_{g'\in h'} \phi_{g'}(\underline{r})} \tag{2.52}$$

The above "cross sections" represent the source in fine-group g from scattering or fission, respectively, from coarse-group h. Therefore, if coarse-group fluxes are known they provide a way to decompose the coarse-group source into the respective fine groups.

Chapter 3

SDM Implementation into TITAN, TITAN-SDM

The SDM algorithm has been demonstrated in a one-dimensional deterministic transport code by Douglass and Rahnema [10]. The implementation in a 1-D code is good for a general proof-of-concept, but the method must be proven in a three-dimensional transport environment, such as TITAN, to be deemed of significant value to the community. Haghighat and Sjoden derived a set of equations, i.e Equations (3.1)-(3.13), for implementation of the SDM into a S_N transport code such as TITAN [14, 15]. Then this algorithm was implemented by Yi and Sjoden into TITAN [13].

The implementation of the SDM algorithm into a 3-D deterministic transport code is a three step process. These steps are as follows: (i) a fine-group flux distribution is obtained and used to collapse the fine-group cross sections into a coarse-group library; (ii) coarse-group transport calculations are performed; and (iii) fine-group fluxes are obtained through a decompostion sweep. This chapter discusses each step as it is implemented into the three-dimensional transport code. Also, an iteration scheme will be presented and an evaluation

of solution convergence will be discussed.

3.1 Fine-group Transport

This step is skipped on the first iteration of the SDM algorithm, as the initial fine-group flux spectrum is a user input or is simply a flat distribution. This step is referred to as the flux-update in the implementation scheme. The following equation is used to obtain the fine-group flux spectrum.

$$\left(\mu\frac{\partial}{\partial x}+\eta\frac{\partial}{\partial y}+\xi\frac{\partial}{\partial z}\right)\psi_g(x,y,z,\mu,\varphi)+\sigma_g(x,y,z)\psi_g(x,y,z,\mu,\varphi)= \sum_{g'=1}^{G}\sum_{l=0}^{L}(2l+1)\sigma_{sl,g'\rightarrow g}(x,y,z)\left\{P_l(\mu)\phi_{g',l}(x,y,z)+2\sum_{k=1}^{l}\frac{(l-k)!}{(l+k)!}P_l^k(\mu)\cdot \left[\phi^k_{C,g',l}(x,y,z)\cos(k\varphi)+\phi^k_{S,g',l}(x,y,z)\sin(k\varphi)\right]\right\} + \frac{\chi_g}{k}\sum_{g'=1}^{G}\nu\sigma_{f,g'}(x,y,z)\phi_{g',0}(x,y,z) \tag{3.1}$$

All variables in Equation (3.1) have been discussed in Section 2.1.2.3, Table 2.1 and Equations (2.27)-(2.30), and the fine-group cross-sections are obtained from the user input cross-section library.

3.2 Coarse-group Transport

Coarse-group fluxes and flux moments are obtained using the corrected coarse-group transport equation, Equation (2.44). The coarse-group transport equation for a 3-D environment

is expressed by

$$\left(\mu\frac{\partial}{\partial x}+\eta\frac{\partial}{\partial y}+\xi\frac{\partial}{\partial z}\right)\psi_h(x,y,z,\mu,\varphi)+\sigma_h(x,y,z)\psi_h(x,y,z,\mu,\varphi)=$$
$$\sum_{h'=1}^{H}\sum_{l=0}^{L}(2l+1)\left\{\sigma_{sl,h'\to h}(x,y,z)P_l(\mu)\phi_{h',l}(x,y,z)+2\sum_{k=1}^{l}\frac{(l-k)!}{(l+k)!}P_l^k(\mu)\cdot\right.$$
$$\left[\sigma^k_{sl,C,h'\to h}(x,y,z)\phi^k_{C,h',l}(x,y,z)\cos(k\varphi)\right.$$
$$\left.\left.+\ \sigma^k_{sl,S,h'\to h}(x,y,z)\phi^k_{S,h',l}(x,y,z)\sin(k\varphi)\right]\right\}$$
$$+\frac{\chi_h}{k}\sum_{h'=1}^{H}\nu\sigma_{f,h'}(x,y,z)\phi_{h',0}(x,y,z)+\delta_h(x,y,z,\mu,\varphi), \quad (3.2)$$

where the coarse-group correction term is expressed by

$$\delta_h(x,y,z,\mu,\varphi)=\sum_{l=0}^{L}(2l+1)\left\{P_l(\mu)\delta_{h,l}(x,y,z)\phi_{h,l}(x,y,z)+\right.$$
$$2\sum_{k=1}^{l}\frac{(l-k)!}{(l+k)!}P_l^k(\mu)\left[\delta^k_{l,C,h}(x,y,z)\phi^k_{l,C,h}(x,y,z)\cos(k\varphi)+\right.$$
$$\left.\left.\delta^k_{l,S,h}(x,y,z)\phi^k_{l,S,h}(x,y,z)\sin(k\varphi)\right]\right\} \quad (3.3)$$

and the necessary moments are

$$\delta_{h,l}(x,y,z)=\sigma_h(x,y,z)-\frac{\sum_{h\in g}\sigma_g(x,y,z)\phi_{g,l}(x,y,z)}{\sum_{h\in g}\phi_{g,l}(x,y,z)}, \quad (3.4)$$

$$\delta^k_{sl,C,h'\to h}(x,y,z)=\sigma_h(x,y,z)-\frac{\sum_{h\in g}\sigma_g(x,y,z)\phi^k_{C,g,l}(x,y,z)}{\sum_{h\in g}\phi^k_{C,g,l}(x,y,z)}, \quad (3.5)$$

and

$$\delta^k_{sl,S,h'\to h}(x,y,z)=\sigma_h(x,y,z)-\frac{\sum_{h\in g}\sigma_g(x,y,z)\phi^k_{S,g,l}(x,y,z)}{\sum_{h\in g}\phi^k_{S,g,l}(x,y,z)}. \quad (3.6)$$

The total coarse-group cross-section, σ_h, is presented in Equation (2.45), fission spectrum, χ_h, in Equation (2.47), and fission cross-section and $\nu\sigma_{f,h}$ in Equation (2.48). The moments of the differential scattering cross-section are

$$\sigma_{sl,h'\to h}(x,y,z) = \frac{\sum_{h\in g}\sum_{h'\in g'}\sigma_{sl,g'\to g}(x,y,z)\phi_{g',l}(x,y,z)}{\sum_{h'\in g'}\phi_{g',l}(x,y,z)}, \tag{3.7}$$

$$\sigma^k_{sl,C,h'\to h}(x,y,z) = \frac{\sum_{h\in g}\sum_{h'\in g'}\sigma^k_{sl,C,g'\to g}(x,y,z)\phi^k_{C,g',l}(x,y,z)}{\sum_{h'\in g'}\phi^k_{C,g',l}(x,y,z)}, \tag{3.8}$$

and

$$\sigma^k_{sl,S,h'\to h}(x,y,z) = \frac{\sum_{h\in g}\sum_{h'\in g'}\sigma^k_{sl,S,g'\to g}(x,y,z)\phi^k_{S,g',l}(x,y,z)}{\sum_{h'\in g'}\phi^k_{S,g',l}(x,y,z)}. \tag{3.9}$$

3.3 Fine-group Decomposition

Once the coarse-group fluxes and flux moments are calculated, it is desired to obtain fine-group fluxes. This is accomplished by solving a one-pass, fine-group transport equation

$$\begin{aligned}\left(\mu\frac{\partial}{\partial x}+\eta\frac{\partial}{\partial y}+\xi\frac{\partial}{\partial z}\right)\psi_g(x,y,z,\mu,\varphi)+\sigma_g(x,y,z)\psi_g(x,y,z,\mu,\varphi) = \\ \sum_{h'=1}^{H}\sum_{l=0}^{L}(2l+1)\left\{R_{sl,h'\to g}(x,y,z)P_l(\mu)\phi_{h',l}(x,y,z)+2\sum_{k=1}^{l}\frac{(l-k)!}{(l+k)!}P^k_l(\mu)\cdot\right. \\ \left[R^k_{sl,C,h'\to g}(x,y,z)\phi^k_{l,C,h'}(x,y,z)\cos(k\varphi)\right. \\ \left.\left.+\ R^k_{sl,S,h'\to g}(x,y,z)\phi^k_{l,S,h'}(x,y,z)\sin(k\varphi)\right]\right\}+ \\ \frac{\chi_g(x,y,z)}{k}\sum_{h'=1}^{H}R_{f,h'\to g}(x,y,z)\phi_{h'}(x,y,z).\end{aligned} \tag{3.10}$$

These R constants are referred to as "subgroup decomposition cross sections". They represent the source in subgroup g, by either fission, Equation (2.52), or scattering, Equation

(3.11), from coarse-group h'. Therefore if coarse-group fluxes are known they provide a way to decompose the coarse-group source into the respective fine-groups.

The scattering subgroup decomposition cross-section moments, $R_{sl,h'\to g}$, and cosine and sine Associated Legendre moments, $R^k_{sl,C,h'\to g}$ and $R^k_{sl,S,h'\to g}$, are

$$R_{sl,h'\to g}(x,y,z) = \frac{\sum_{g'\in h'} \sigma_{sl,g'\to g}(x,y,z)\phi_{g',l}(x,y,z)}{\sum_{g'\in h'} \phi_{g',l}(x,y,z)}, \tag{3.11}$$

$$R^k_{sl,C,h'\to g}(x,y,z) = \frac{\sum_{g'\in h'} \sigma_{sl,g'\to g}(x,y,z)\phi^k_{C,g',l}(x,y,z)}{\sum_{g'\in h'} \phi^k_{C,g',l}(x,y,z)}, \text{ and} \tag{3.12}$$

$$R^k_{sl,S,h'\to g}(x,y,z) = \frac{\sum_{g'\in h'} \sigma_{sl,g'\to g}(x,y,z)\phi^k_{S,g',l}(x,y,z)}{\sum_{g'\in h'} \phi^k_{S,g',l}(x,y,z)}. \tag{3.13}$$

The fission subgroup decomposition cross-section, $R_{f,h'\to g}$, is given in Equation (2.52)

Note that Equation (3.10) requires only a single iteration because the fission source and scattering source are calculated using the previously obtained coarse-group fluxes, flux moments, and the R coefficients, instead of fine-group fluxes and flux moments. Both source terms are fixed; therefore, only one transport sweep is sufficient to acquire fine-group angular fluxes, because no iteration process is required on convergence of either source term.

3.4 Iteration Scheme

Figure 3.1 describes the iteration scheme of TITAN-SDM. The first step is to initialize the fine-group flux, or in our case use a flat flux, and k. Then, using the fine-group flux distribution, collapse the fine-group cross sections to obtain coarse-group and subgroup constants. Using these coarse-group constants, coarse-group transport calculations are completed to obtain an updated coarse-group flux and k; at this step the source iteration method Section

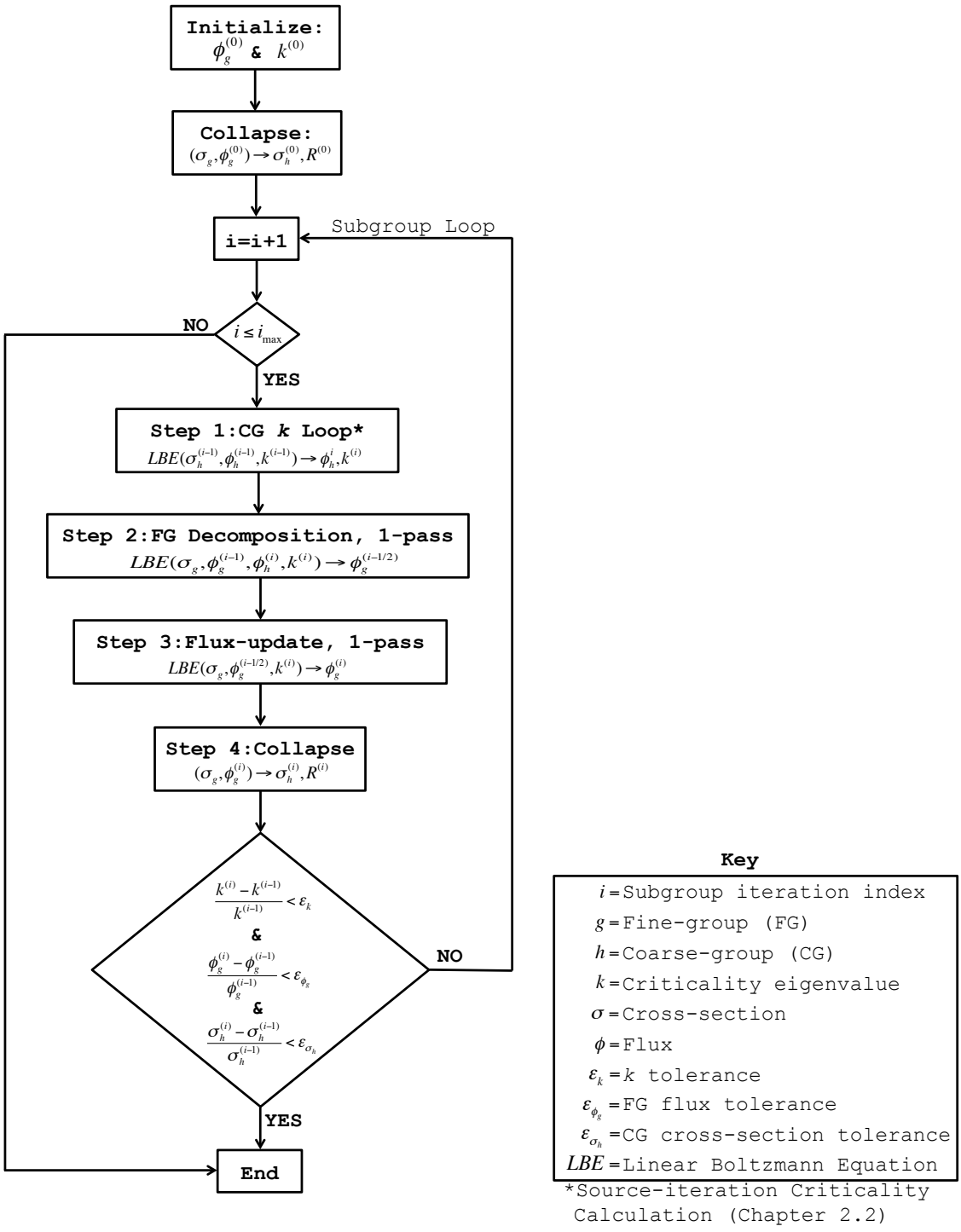


Figure 3.1 TITAN-SDM Iteration Scheme.

2.2 is completed to update coarse-group k and flux. Then a fixed-source, single-pass decomposition sweep is completed using fine-group cross sections, previous iteration fine-group flux, updated coarse-group flux, and k to obtain an updated fine-group flux and k. Then a standard fixed-source, fine-group transport calculation is completed to ensure that particle balance is obtained within the fine-group, this is referred to as the "flux update". Once the fine-group fluxes are updated, the fine-group flux distribution is used to re-collapse the fine-group cross sections to obtain updated coarse-group constants. The coarse-group cross-section data and subgroup cross sections are obtained at the beginning of each iteration using the updated fine-group fluxes as weighting factors, Equations (2.45)-(3.6).

3.4.1 Memory Requirement Estimation

The material-based fine-group cross-section library is a user input and does not change in the iteration process. However, coarse-group cross-section data and R factors are spatially dependent, they are recalculated at the beginning of each subgroup iteration using fine-group flux moments as weighting factors. These fine-group flux moments are spatially dependent, in turn making the coarse-group cross sections and R factors spatially dependent. This means that each fine mesh has it own individual cross-section table and associated R factors.

The additional memory requirement, in bytes, for implementation of the SDM algorithm may be estimated by

$$\text{Memory} = \underbrace{(L+1)N \cdot H^2}_{(1)} + \underbrace{(3L+4)G \cdot H}_{(2)} + \underbrace{3N \cdot H}_{(3)}, \tag{3.14}$$

where N is the number of fine meshes, G is the total number of fine groups, H is the total number of coarse groups, and L is the scattering P_N order. The first term is memory required to store fine-mesh based coarse-group scattering cross-section tables. The second term is the

memory required to store R factors; these values are only calculated once for the single-pass fine-group decomposition and, as such, are calculated on-the-fly for each fine-mesh, thereby eliminating the need to store R factors for every fine-mesh. The third term is the memory required to store total and fission cross sections and $\bar{\nu}$ data.

3.4.2 Algorithm Control Parameters

The implementation of the SDM algorithm introduces two new convergence parameters in addition to the standard k and *flux* tolerances; these tolerances are the coarse-group cross sections (σ_h) and fine-group fluxes (ϕ_g). Table 3.1 summarizes the formulations for the convergence criteria that are examined at the end of each coarse- and fine-group loop.

Table 3.1 TITAN-SDM iterative error calculations.

Coarse-group Loop	**Fine-group Loop**
$RE(k\ ) = \dfrac{k^{\ i-1} - k^{\ i}}{k^{\ i}}$	$RE(k\ ) = \dfrac{k^{\ i-1} - k^{\ i}}{k^{\ i}}$
$RE(\phi_h) = \dfrac{\phi_h^{i-1}(\underline{r}) - \phi_h^i(\underline{r})}{\phi_h^i(\underline{r})}$	$RE(\phi_g) = \dfrac{\phi_g^{i-1}(\underline{r}) - \phi_g^i(\underline{r})}{\phi_g^i(\underline{r})}$
	$RE(\sigma_h) = \dfrac{\sigma_h^{i-1}(\underline{r}) - \sigma_h^i(\underline{r})}{\sigma_h^i(\underline{r})}$

Chapter 4

Methodology for Evaluation of TITAN-SDM Performance

This chapter outlines the methodology used to evaluate TITAN-SDM performance. Performance of the TITAN-SDM code will be evaluated by comparing the SDM with the standard multigroup for two problems: 1-D and 2-D whole-core benchmark models of the High Temperature Engineering Test Reactor (HTTR) will be simulated [16]. This benchmark has been chosen for testing the TITAN-SDM code because it exhibits a strong energy-angle coupling which requires numerous energy groups to achieve an acceptable whole core solution when standard multigroup transport is used.

This chapter presents the reference TITAN solutions used to evaluate the accuracy and performance of the SDM algorithm, as well as the effect of different modeling parameters, including quadrature order, meshing, and tolerances. Section 4.1 presents the metrics used to evaluate the accuracy and performance are presented. Section 4.2 discusses the reference problems and their analysis using the standard TITAN multigroup algorithm.

4.1 Metrics of Performance

This section discusses various metrics used to assess the accuracy and efficiency of TITAN-SDM as compared to the standard mulitgroup TITAN algorithm

4.1.1 Accuracy Analysis

A solution is considered accurate if the eigenvalue and the flux spectrum show adequate agreement with the reference solution. Other integral quantities, such as reaction rates, may also be considered to quantify level of accuracy of a solution.

4.1.1.1 k Eigenvalue

The k relative error ϵ_k is defined in units of percent milli (pcm) reactivity as

$$\epsilon_k = 10^5 \left(\frac{\bar{k} - k}{k} \right), \tag{4.1}$$

where k is the reference eigenvalue and $\bar{k}$ is the calculated eigenvalue. If the SDM algorithm is functioning properly k should agree with the benchmark case and the relative error in k, ϵ_k, should be very small.

4.1.1.2 Scalar Flux

The percent relative error of the scalar flux, ϵ_f, is defined in Equation (4.2), where ϕ is the reference scalar flux and $\bar{\phi}$ is the calculated scalar flux.

$$\epsilon_f = 100 \left(\frac{\bar{\phi} - \phi}{\phi} \right) \tag{4.2}$$

For analysis we will consider the magnitude of the maximum percent relative error, $\max(|\epsilon_f|)$, of the scalar flux within the fuel region and the whole core. We will also look at the distribution of the relative flux error, ϵ_f, across the core.

4.1.1.3 Reaction Rates

For the 1-D problem reaction rates, specifically fission and non-fission capture rates, are calculated for the different regions of the core.

$$R_{x,k} = \sum_{i \in k} \sum_{g=1}^{G} \sigma_{x,g} \phi_{i,g} V_i \tag{4.3}$$

In the above equation, x is the interaction of interest, i.e., fission, absorption, capture, etc., k is the material or region of interest, $\phi_{i,g}$ is the scalar flux of group g within fine mesh i, and V_i is the volume of the fine mesh.

To analyze the main core region, fuel pin fission rates are calculated using the formulation of Equation (4.3). Various methods to quantify the magnitude of the pin-wise fission rate errors are evaluated by the following equations:

percent relative difference

$$re_i = 100 \frac{\bar{R}_{f,i} - R_{f_i}}{R_{f_i}}, \tag{4.4}$$

maximum relative difference

$$MAX = \max(|re_i|), \tag{4.5}$$

and root mean square difference, in percent,

$$RMS = \sqrt{\frac{\sum_i |re_i|^2}{I}}, \tag{4.6}$$

where I is the total number of fuel pins in the assembly.

4.1.2 Efficiency Calculation

A calculated solution is considered efficient if the time required to compute an *accurate* solution is less than the time required to obtain the fine-group reference solution, i.e., speedup greater than 1.0. Computational speedup is defined as

$$\text{Speedup} = \frac{T_{ref}}{T_{SDM}}. \tag{4.7}$$

T_{ref} is the computation time required for the reference case and T_{SDM} is the time required for the calculated solution using SDM.

4.2 Benchmark Problems

Two different problems are considered to evaluate the accuracy and effectiveness of the SDM algorithm as implemented in the TITAN code. The 1-D and 2-D HTTR core models are defined and analyzed in the following sections.

4.2.1 1-D HTTR Core

The 1D Very High Temperature Test Reactor (VHTR) core test problem [24] was derived from a simplified two- and three-dimensional benchmark problem [16] that was based on different High Temperature Test Reactor(HTTR) core design studies [25]. The model specifications and fine-group solution are discussed in the following section.

The HTTR core problem is challenging because it is difficult to obtain a good weighting flux

on the lattice level due to the large number of non-fissionable blocks. The Reflector blocks are of specific concern because of the large flux attenuation in these regions, which is significantly different than the fissionable regions. This means that the lattice spectrum is highly dependent on the spectral characteristics of neighboring blocks, which makes this a good test problem for the SDM algorithm because it is susceptible to spectral core environment errors.

4.2.1.1 Geometry Specifications

The 1-D model includes 11 different materials for which cross-section data are obtained by using the 47-group HELIOS library [26]. Figure 4.1 depicts the 1-D HTTR model. It shows different material regions and indicates that specular reflective and vacuum boundary conditions are prescribed at the left ($-x$) and right ($+x$) boundaries, respectively.

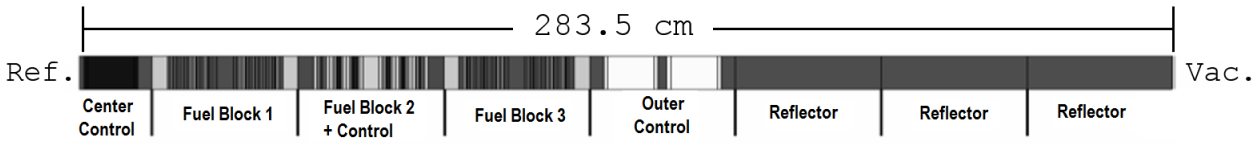


Figure 4.1 1-D HTTR: Geometry, material layout, and boundary conditions.

The discretized model includes 188 coarse meshes with fine mesh dimensions determined by 0.5 material mean-free-path[1] (mfp). Since TITAN is a 3-D Cartesian code system, a reflective boundary condition is prescribed on ±y and ±z boundaries and, to remove numerical effects of the reflective boundary conditions, a large width of 100 cm is considered for both y and z axes. Additionally for the scattering expansion, we consider P0 scattering order.

[1]Mean-free-path is equal to the inverse of the maximum total cross-section, i.e., $mfp = 1/\max(\sigma_g)$

4.2.1.2 Mesh and Quadrature Order Refinement

The effect of mesh size and quadrature order is examined by simultaneously refining the mesh size and quadrature order. This analysis will demonstrate which combination of these parameters provides an acceptable level of accuracy; the relative error of the k-eigenvalue will be calculated versus the most refined case for the evaluation of accuracy.

Table 4.1 shows three different mesh sizes and four different quadrature orders that are considered; mesh sizes are defined as a function of mfp. For every mesh all quadrature orders are tested.

Table 4.1 1-D HTTR: Solution numerical parameters.

(a) Spatial Mesh

Mesh Size
mfp
$mfp/2$
$mfp/4$

(b) Angular quadratures

Quad. Order	**No. of Directions in 3-D**
4	24
6	48
8	80
10	120

Table 4.2 shows the results of the mesh and quadrature order study. Table 4.2a shows the relative errors in pcm compared to the most refined case, a mesh of $mfp/4$ with S_10 quadrature order. Table 4.2b shows the computation time, in minutes, required for each calculation. The presented results are as expected; relative error decreases with a more refined mesh and quadrature order (Table 4.2a), while the required computation time increases (Table 4.2b). A meshing of $mfp/2$ with a quadrature order of S_8 is used for remaining calculations.

Table 4.2 1-D HTTR: Results of the mesh and quadrature order sensitivity analysis.

(a) Absolute relative error, in *pcm*, of k compared to $mfp/4$, S_10.

Quad	Mesh Size		
Order	**mfp**	**mfp/2**	**mfp/4**
4	147	119	85
6	93	66	32
8	63	37	9
10	50	24	0

(b) Computation times, in minutes, for the meshing and quadrature orders.

Quad	Mesh Size		
Order	**mfp**	**mfp/2**	**mfp/4**
4	35.4	40.5	54.4
6	54.3	61.3	77.6
8	74.6	78.3	111.6
10	103.7	111.2	129.3

4.2.1.3 Convergence Parameters

The effect of *flux* and k tolerances is evaluated here. First, the effect of the flux tolerance in the reflector region is examined by setting the k-tolerance and core flux tolerance constant and varying the flux tolerance in the reflector. Table 4.3 shows results from varying the flux tolerance in the reflector while k-tolerance is 10^{-6} and core flux tolerance is 10^{-5}. Relative errors are calculated against the tightest reflector flux tolerance of 10^{-5}, this represents the case with same flux tolerance everywhere. The above table indicates that the tolerance

Table 4.3 1-D HTTR: Analysis of flux tolerance in reflector region.

$\phi_{refl.}$**-Tol**	k_{eff}	**Err**[1] **- [*pcm*]**	**Iterations**	**Time - [min.]**
$\mathbf{10^{-1}}$	1.10303	-12	551	66.1
$\mathbf{10^{-2}}$	1.10303	-12	551	65.8
$\mathbf{10^{-3}}$	1.10305	-10	561	54.2
$\mathbf{10^{-4}}$	1.10299	-15	521	40.3
$\mathbf{10^{-5}}$	1.10316	ref.	577	61.9

[1] Relative error calculated against the case where reflector flux tolerance is 10^{-5}, flux tolerance in the core is 10^{-6}, and k tolerance is 10^{-6}.

in the reflector region has little effect on the overall core solution. This is because the outer control region, made of graphite, essentially acts as the core reflector. As such, if any neutrons make it through the outer control block and are reflected back towards the

core, the outer control region will shield these thermalized neutrons from reaching the fuel blocks. Therefore, the reflector regions are inconsequential to the overall core solution. We will continue to investigate the effect of the other tolerances using a reflector tolerance of 10^{-1}.

Now we will see how the outer (k) and inner (core flux) tolerances affect the solution. Table 4.4 shows the relative errors in k , calculated versus a k tolerance of 10^{-7} and core flux tolerance of 10^{-6}, for various combinations of outer and inner tolerances. The behavior is as expected, the relative errors decrease with the tighter tolerances.

Table 4.4 1-D HTTR: Absolute k relative errors, in *pcm* compared against k-tolerance of 10^{-7} and core flux tolerance of 10^{-6}.

k-	**Core Flux Tolerance**		
Tol.	$\mathbf{10^{-4}}$	$\mathbf{10^{-5}}$	$\mathbf{10^{-6}}$
$\mathbf{10^{-5}}$	285	38	5
$\mathbf{10^{-6}}$	113	30	5
$\mathbf{10^{-7}}$	102	17	0

In order to reduce the difference between the SDM solutions and the reference solution due to effects of convergence criteria of the reference solution, based on the above examination, the reference solution will have very tight convergence tolerances: k tolerance of 10^{-7}, core flux tolerance of 10^{-5}, and the convergence in the reflector region will essentially be ignored with a reflector flux tolerance of 10^{-1}.

4.2.1.4 47-group Reference Calculation Results

Here a 47-group reference solution is presented. All the parameters required for modeling the 1-D HTTR core are outlined in Table 4.5, below.

Table 4.5 1-D HTTR: Modeling parameters in TITAN calculations.

Parameter	**Value**
Scattering Order	P0
Quadrature Order	8
Differencing Scheme	Diamond w/ fixup
k -tolerance	10^{-7}
flux-tolerance[1]	10^{-5} (core) 10^{-1} (refl.)
Max. flux iters.	10
Max. k iters	1000

[1]Regional flux tolerances were implemented.

The reference case k results along with computation time and number of iterations are displayed in Table 4.6.

Table 4.6 1-D HTTR: 47-group reference calculation results.

k_{eff}	**Iterations**[1]	**Time - [hrs]**
1.10330	1318	1.80

[1]Total number of k (Outer) iterations.

Fast (groups 1-35) and thermal (groups 36-47) flux distributions and fuel pin fission rates are shown in Figure 4.2. Note that the pin fission rates have been normalized so that the average pin fission rate is 1.

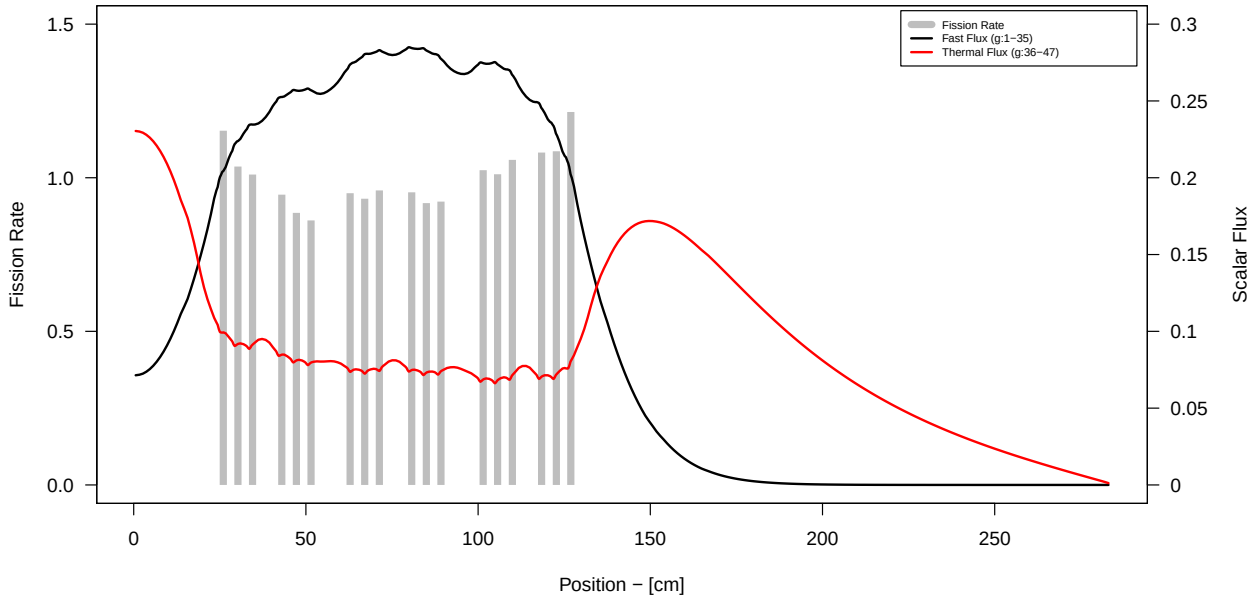


Figure 4.2 1-D HTTR: Fast and thermal flux distribution and pin fission rates.

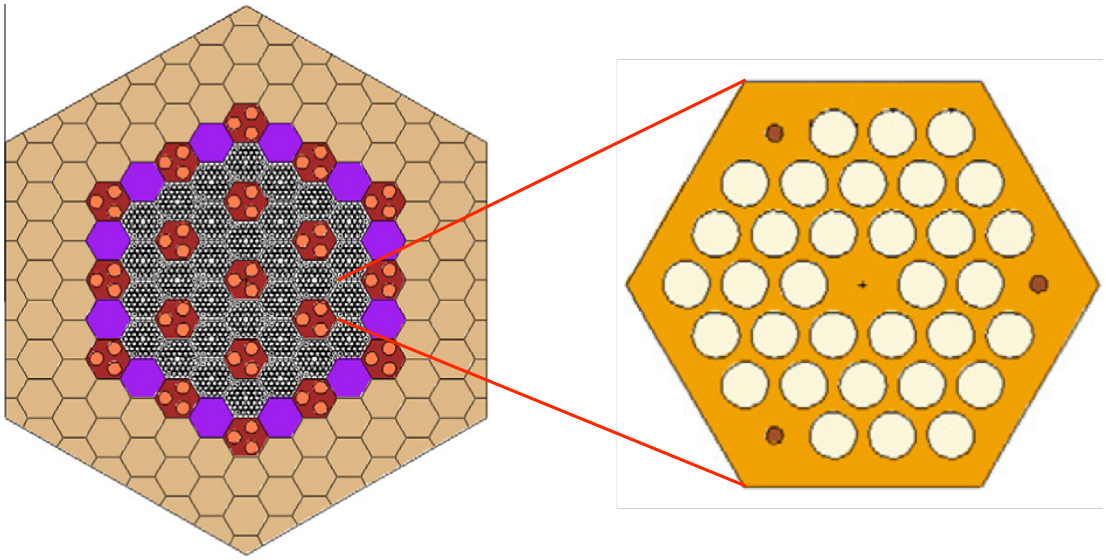
The flux profile is as expected, the thermal flux peaks in the center region while the fast flux drops in this region. In the fissionable region the fast flux is at its maximum and the thermal flux is low in this region; as you approach regions between fuel pins the fast flux drops and the thermal flux increases. In the reflector region the fast flux drops significantly once outside of the fuel region, at the same point the thermal flux increases, after this point the thermal flux slowly attenuates while the fast flux is effectively zero here. The fuel pin fission rates also behave as expected; higher fission rate on the outer fuel pins than in the center fuel pins, more neutrons are absorbed in these pins than in the center pins because they are the closest to the moderator regions.

The fact that the fast flux is effectively zero in the reflector shows that in this region it will be difficult to converge on the fast flux. Large flux errors observed in these regions with nearly zero fluxes may be attributed to roundoff errors due to numerical issues with using single

precision values. For a problem such as this, with significant flux gradients, it is necessary to use regional flux tolerances. This is an option that was not available in TITAN prior to this work, but was implemented to handle these problems.

4.2.2 2-D HTTR Core

The 2-D homogenized HTTR core test problem is derived from the simplified two- and three-dimensional benchmark problem [16] based on different High Temperature Engineering Test Reactor (HTTR) core design studies [25]. The model, shown in Figure 4.3, uses 11 different materials, 7 fuels with varying enrichment, 4 different graphites for placement in different regions, 1 burnable poison, and 1 control rod. The center core region, Figure 4.3a, is composed of a lattice of 61 hexagonal blocks that are either a fuel, control, or reflector block surrounded by reflector blocks. An individual fuel assembly, Figure 4.3b, is comprised of 33 fuel rods in a hexagonal lattice with 3 burnable poison rods.



(a) Heterogeneous model. **(b)** Fuel assembly geometry.

Figure 4.3 2D HTTR: Benchmark problem layout.

The grey cells are fuel assemblies, magnified in Figure 4.3b, comprised of 33 fuel pins and three burnable poison rods, the red cells are control blocks with three removable control rods, the purple cells are fixed control blocks, and the brown cells are reflector blocks.

Prior to completing a whole core model, it is necessary to demonstrate that TITAN can accurately estimate hexagonal geometry using Cartesian geometry. A single fuel assembly will be modeled in TITAN and compared to a MCNP reference solution.

The following sections first evaluate the capability of TITAN to model hexagonal geometry for a 2-D HTTR fuel assembly and then present results from a 2-D full-core HTTR model.

4.2.2.1 Fuel Assembly

This section examines TITAN's capability to accurately model a hexagonal geometry. It must be demonstrated that the cartesian geometry used in TITAN can accurately represent a hexagonal geometry. The conservation of material volume is of primary concern. A reference multigroup MCNP calculation is used to qualify the TITAN model.

TITAN Model

A TITAN model has been prepared using PENMSHXP [27], as shown in Figure 4.4. The fuel assembly contains 3 different materials: graphite is used as a moderator, UO_2 fuel enriched to 3.4 wt.% ^{235}U enrichment, and a burnable poison. There are 33 fuel rods and three burnable poison rods in this assembly model.

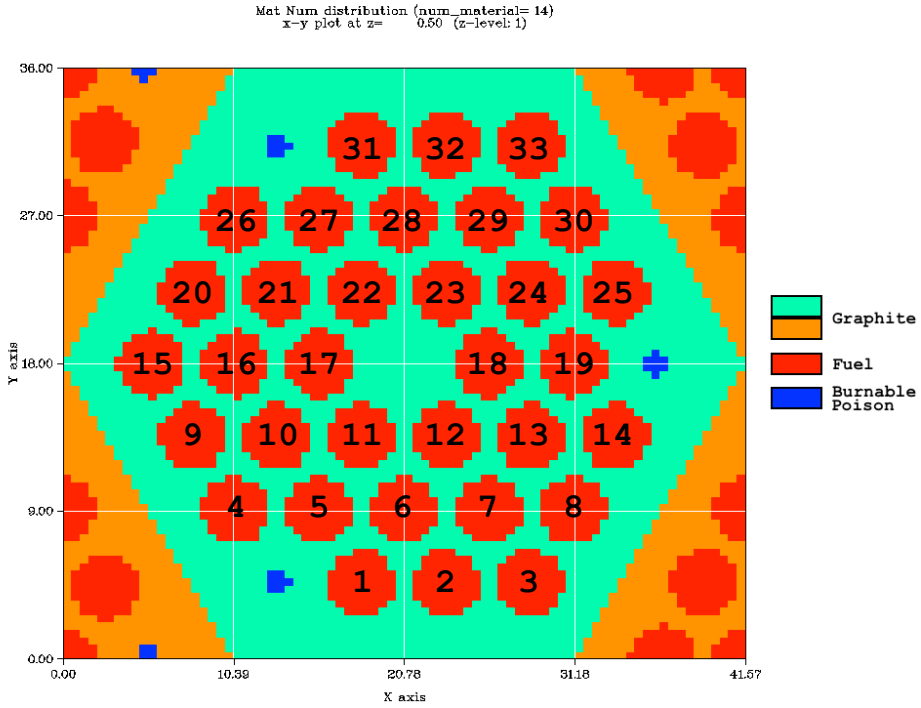


Figure 4.4 2-D HTTR Fuel Assembly: Geometry, material layout, and fuel pin numbers.

All boundaries are modeled as specular reflective for the 2-D fuel assembly in Figure 4.4. Note that, because of TITAN's Cartesian geometry, this model includes segments of the four neighboring assemblies. Of course, this is not a true representation of the fuel assembly, but the purpose of this study is to examine the effectiveness of TITAN in modeling a hexagonal lattice using a Cartesian geometry.

The TITAN model, Figure 4.4, is comprised of 16 coarse meshes, and each coarse mesh contains 20 by 20 fine meshes. A 6-group cross-section library was obtained from Zhang, et al [16]. Table 4.7 shows the 6-group energy structure [28].

Table 4.7 Upper energy of the 6-group library.

Group	Upper Energy (eV)
1	2.00×10^7
2	1.83×10^5
3	9.61×10^2
4	2.38
5	0.65
6	0.105

Appendix A provides a sample TITAN input deck for the 2-D HTTR fuel assembly described here.

TITAN Calculation Results

Table 4.8 presents the parameters used in the base TITAN calculation with 20x20 fine meshes per coarse mesh (FM/CM). All computations were done using a single processor.

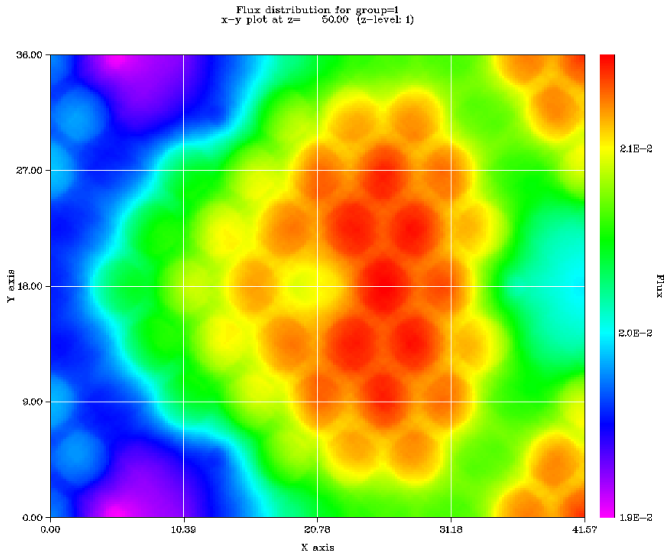
Table 4.8 2-D HTTR Fuel Assembly: TITAN parameters.

Parameter	Value
Scattering Order	P0
Quadrature Order	8
Differencing Scheme	Diamond w/ fixup
k -tolerance	10^{-6}
flux-tolerance[1]	10^{-4}
Max. flux iters.	15
Max. k iters	1000

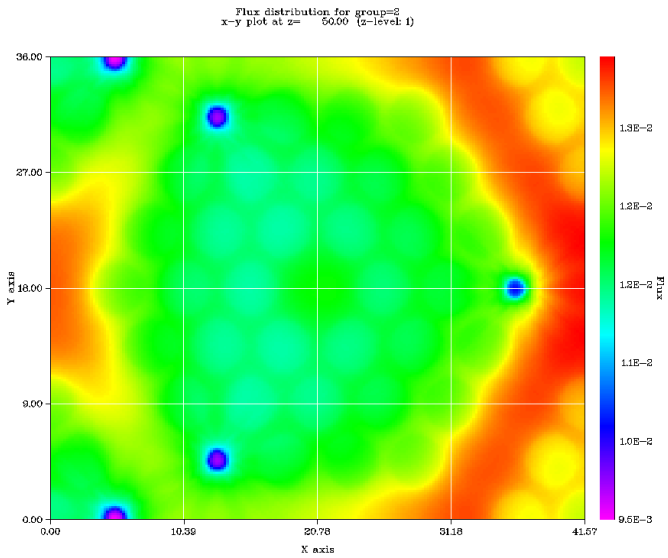
[1]Uniform flux tolerances were used.

The eigenvalue for the base run was k =1.15039. Figure 4.5 shows the fast (groups 1-4) and thermal (groups 5-6) flux distributions calculated by TITAN. These figures show the expected trends, the fast flux peaks in the fuel pins where the thermal flux is low, and

the thermal flux drops in the burnable poison rods. In addition, the flux distribution is asymmetric due to the asymmetric positioning of the burnable poison rods.



(a) Fast (groups 1-4).



(b) Thermal (groups 5-6).

Figure 4.5 2-D HTTR Fuel Assembly: Fast and thermal flux distributions calculated with TITAN.

Effect of Mesh Size and Quadrature Order

The effect of meshing and quadrature order are observed by simultaneously refining the mesh size and quadrature order. This analysis will demonstrate which combination of these parameters provides an acceptable level of accuracy; the relative error of the k-eigenvalue will be calculated versus the most refined case for the evaluation of accuracy.

As presented in Table 4.9 , five different mesh sizes, two coarser (5×5 and 10×10) and two finer (40×40 and 50×50), and four different quadrature orders, two coarser (S_6 and S_8) and two finer (S_{10} and S_{14}), will be considered; for each mesh size, all quadrature orders are tested. Table 4.10 presents results of the meshing and quadrature order study. Table 4.10 shows relative differences in k as compared to the most refined case, 50 × 50 fine-meshes per coarse mesh and S_{14} quadrature order. Table 4.10b shows the computation time, in minutes, necessary for each calculation. Table 4.10a suggests that the quadrature order doesn't significantly affect the solution accuracy, while mesh size is important. If the mesh size is not refined enough TITAN may calculate an inaccurate solution. This analysis suggests it is necessary to use a finely resolved meshing to properly model this hexagonal assembly; a quadrature order of S_6 should suffice to accurately model the assembly.

Table 4.9 2-D HTTR Fuel Assembly: TITAN numerical analysis.

(a) Spatial Meshing

Mesh Size	**No. FM/CM**[1]
mfp	5×5
$mfp/2$	10×10
$mfp/4$	20×20
$mfp/8$	40×40
$mfp/12$	50×50

[1] fine-mesh per coarse-mesh

(b) Angular Quadrature

Quad Order	**No. of Directions in 3-D**
6	48
8	80
10	120
14	224

Table 4.10 2-D HTTR Fuel Assembly: Results of the mesh and quadrature order sensitivity analysis.

(a) Absolute relative error, in *pcm*, of k compared to $mfp/12$, S_{14} case.

Quad	**Mesh Size**				
Order	**mfp**	**mfp/2**	**mfp/4**	**mfp/8**	**mfp/12**
6	1318	3014	162	217	54
8	1352	2975	138	247	23
10	1365	2956	125	265	8
14	1381	2935	114	274	0

(b) Computation time, in minutes, for the meshing and quadrature refinements.

Quad	**Mesh Size**				
Order	**mfp**	**mfp/2**	**mfp/4**	**mfp/8**	**mfp/12**
6	0.7	1.8	5.6	17.8	77.7
8	1.0	2.8	8.6	29.9	107.6
10	1.4	4.0	11.6	43.7	183.7
14	2.5	6.8	19.9	76.9	211.2

Comparison of the TITAN Solution with MCNP Reference Solution

An MCNP model, Figure 4.6, was developed to evaluate the accuracy of the TITAN solution. All boundaries are modeled as specular reflective and the same TITAN 6-group cross-section library is used.

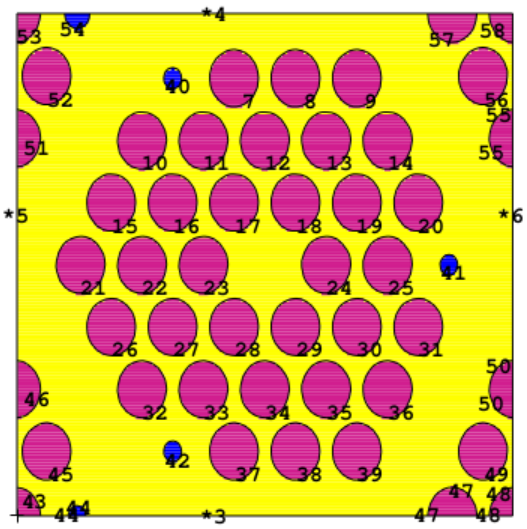


Figure 4.6 2-D HTTR Fuel Assembly: MCNP model geometry.

Table 4.11 shows a comparison of the eigenvalue results from the TITAN model and the MCNP reference solution.

Table 4.11 2-D HTTR Fuel Assembly: Comparison of calculated k-eigenvalues between TITAN calculation and MCNP reference solution.

Method	k	***pcm***
MCNP	1.15108 ($\pm$ 2pcm)	0
TITAN	1.15136	23

The above comparison of calculated eigenvalues indicates that TITAN is capable of calculating an accurate k-eigenvalue, to a level of 20 pcm, using the TITAN parameters defined in Table 4.8. To further evaluate the accuracy of the TITAN model it is necessary to compare integral quantities to the MCNP reference solution, as MCNP is not capable of producing point-wise flux values. We will compare the local peaking factor (LPF) for each fuel pin[2] of the TITAN calulation to the MCNP reference solution. The LPF is a quantity commonly used to classify the shape of the power curve within a reactor.

[2]Fuel pin numbers are labeled in Figure 4.4.

Fuel pin power is expressed by

$$P_i = E_R \cdot R_{f_i} \tag{4.8}$$

and LPF is expressed by

$$LPF_i = \frac{P_i}{\bar{P}}. \tag{4.9}$$

Where E_R is the energy recoverable by fission, 200 MeV, f_i is the fission rate calculated using Equation (4.3), and $\bar{P}$ is the core-average power, $\bar{P} = (\Sigma_i P_i)/N$, where N is the total number of fuel pins. The uncertainty of the LPF_i relative difference calculation, $\sigma_{re_{LPF_i}}$, is

$$\sigma_{re_{LPF_i}} = \sigma_{P_{M_i}} \left[\left(\frac{P_{T,i}}{\bar{P}_T P_{M,i}} \right)^2 + \left(\frac{\bar{P}_M P_{T,i}}{\bar{P}_T P_{M,i}^2} \right)^2 \right]^{1/2}, \tag{4.10}$$

where $\sigma_{P_{M_i}}$ is the uncertainty of the MCNP flux tally, $\bar{P}_M$ is the average MCNP calculated power over all fuel pins, $\bar{P}_T$ is the average TITAN calculated power in all fuel pins, P_{M_i} is the MCNP calculated power in fuel pin i, and P_{T_i} is the TITAN calculated power in fuel pin i.

Figure 4.7 shows the relative difference of the LPFs from the TITAN calculation compared to the MCNP solution along with the associated uncertainties. Note that the relative differences are presented as a percent, i.e., (Rel-Diff)$\times 10^2$. Figure 4.7 shows that the TITAN model accurately captures the LPF trends of the assembly. The maximum relative difference is $\mp 0.46\%$ at fuel pins 7 and 20, respectively; the maximum 1-σ standard deviation of the relative difference is $\pm 0.09\%$ or approximately 20% of the maximum relative difference.

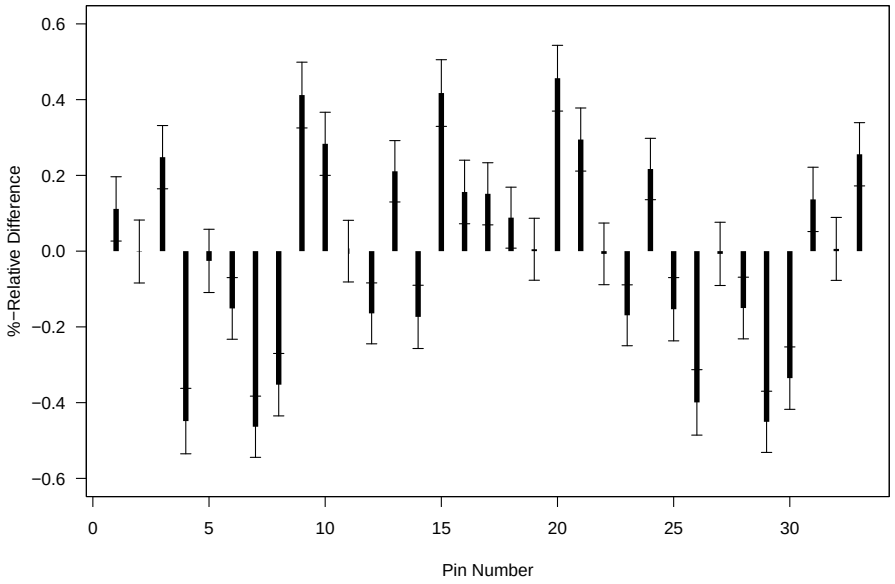


Figure 4.7 2-D HTTR Fuel Assembly: Relative difference of the local peaking factors from the TITAN calculation as compared to the MCNP reference solution, along with associated uncertainties.

This analysis has shown that TITAN is capable of accurately modeling a hexagonal geometry using Cartesian meshing. Care must be taken in selection of mesh size; it is important that a small enough mesh size is used to conserve material volumes, especially when estimating curved or diagonal surfaces.

4.2.2.2 Whole Core

A hexagonal model with the detail of Figure 4.3a would be extremely complicated using Cartesian-based meshing. Therefore, a "block-homogenized" full core model was developed, as pictured in Figure 4.8. The core was homogenized by block type, e.g., fuel block or control block, this leads to 8 different material regions (4 fuel, 2 control, and 2 reflectors) with a void region that must also be modeled due to TITAN's Cartesian meshing. This problem was further simplified to only 3 materials (1 fuel, 1 control, and 1 reflector) in addition to

the void region. The void region was modeled by creating a purely absorbing material, i.e., $\sigma_a = \sigma_t$.

Geometry Specifications

The 2D HTTR core is modeled in TITAN using quarter-symmetry. As previously mentioned, this model has only 3 different materials with cross-section information from the 26-group HELIOS library [26]. A fourth material, a pure absorber, is necessary to place in the void region (top right). Model geometry and material layout are shown in Figure 4.8.

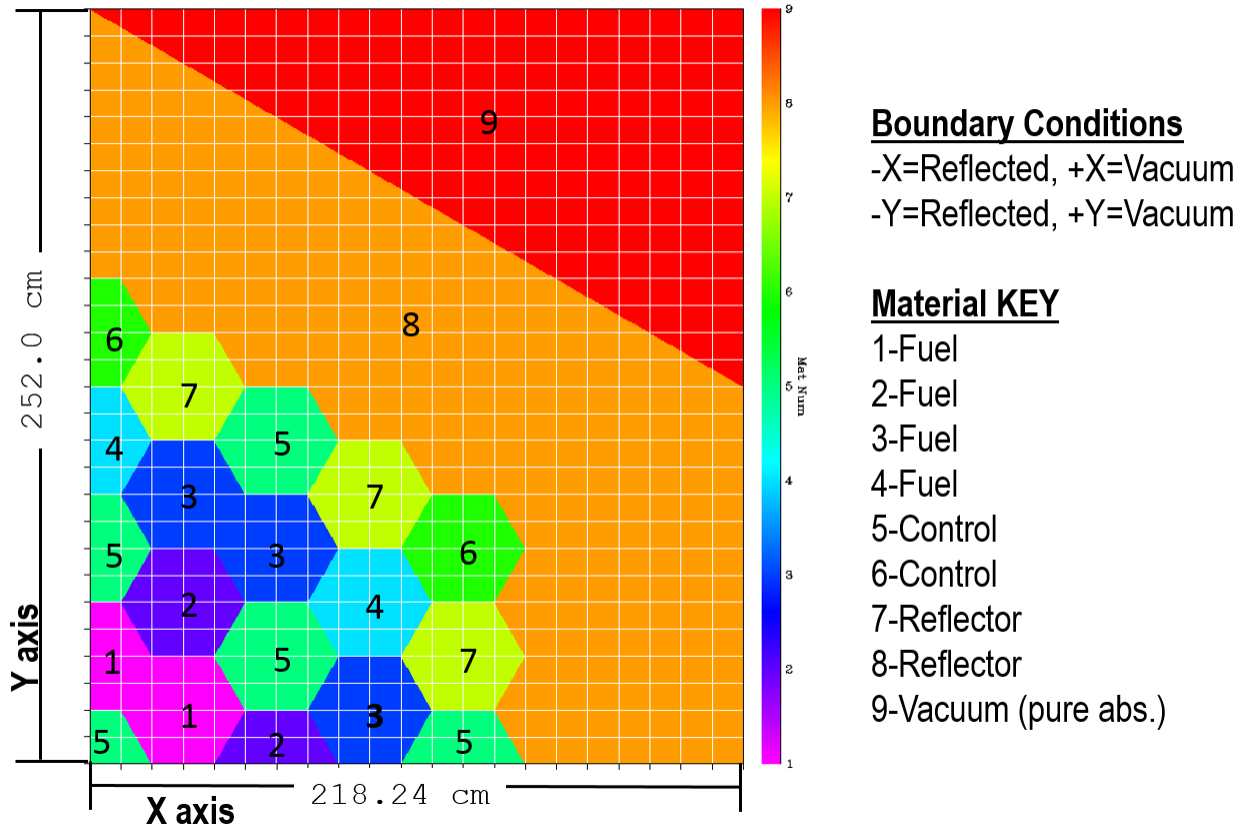


Figure 4.8 2-D HTTR Core: Geometry, material layout, and boundary conditions.

Note that there are 9 different materials labeled in Figure 4.8; this is because the mesh was

developed to model the original core which actually has 9 different material regions [16]. The bottom (-y) and left (-x) boundaries are modeled as specular reflective and the top (+y) and right (+x) are modeled as vacuum.

Mesh and Quadrature Order Refinement

It is necessary to examine the effect of mesh size and quadrature order. This is accomplished by simultaneously refining mesh size and increasing quadrature order. This analysis will determine which combination of these parameters will provide an acceptable level of accuracy in a reasonable time. The k relative error will be calculated against the most refined case as a measure of solution accuracy.

Three different mesh sizes and four different quadrature orders will be considered. For every mesh size all quadrature orders are tested. Table 4.12 lists the different mesh sizes and angular quadrature orders considered.

Table 4.12 2-D HTTR Core: TITAN numerical analysis.

(a) Spatial Meshing

Mesh Size
mfp
$mfp/2$
$mfp/4$

(b) Angular Quadrature

Quad. Order	**Number of Directions**
4	24
6	48
10	120
14	224

Table 4.13 shows the results of the meshing and quadrature order refinement study. Table 4.13a shows the k relative errors in *pcm* calculated against the most refined case, a mesh of $mfp/4$ and quadrature order of S_{14}. Table 4.13b shows the necessary computation time in minutes required.

Table 4.13 2-D HTTR Core: Results from the meshing and quadrature order sensitivity analysis.

(a) Absolute relative error, in *pcm*, of k compared with $mfp/4$, S_{14}.

Quad	**Mesh Size**		
Order	**mfp**	**mfp/2**	**mfp/4**
4	97	2	11
6	119	33	17
10	167	61	9
14	212	124	0

(b) Computation time, in hours, for the meshing and quadrature refinements.

Quad	**Mesh Size**		
Order	**mfp**	**mfp/2**	**mfp/4**
4	0.8	3.5	15.7
6	1.4	5.8	30.6
10	3.2	13.1	67.7
14	5.7	22.0	125.8

At a meshing of $mfp/4$ the relative errors are well below 25 pcm, but the required computation time is substantially more due to the smaller meshing size. We will use a meshing of $mfp/2$ for calculations hereafter because we can get a sufficiently accurate solution for use in this study, $\epsilon_k \leq 60$ pcm , at a much lower computational cost.

For a fixed mesh size, example $mfp/2$, it is noticed that the relative error increases with increasing quadrature order. The fact that the solution is diverging indicates that the meshing is not small enough to resolve all of the directions. This indicates that a finer mesh is required for the given quadrature set. Another issue could be the differencing scheme, Diamond Difference with Fix-up, is not capable of accurately resolving the meshing and quadrature order selected.

Using a model with $mfp/2$, the relative error of the scalar flux is calculated against the

S_{14} model. Table 4.14 shows the maximum percent relative scalar flux errors along with the locations that they occur within the core.

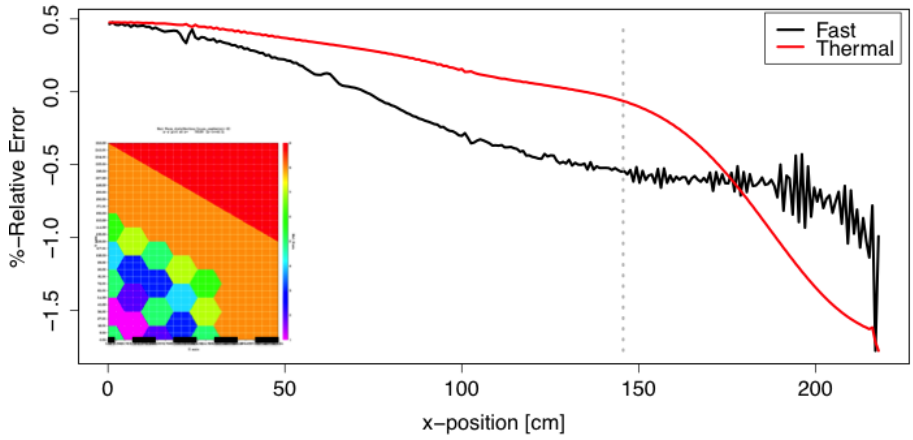
Table 4.14 2-D HTTR Core: Maximum scalar flux relative errors compared to the S_{14} case for meshing of $mfp/2$.

Quad. Order	**Max. ϕ Err - [%]**	**Location - [cm] x**	**y**	**CM No.**
4	62.2	205.3	175.1	419
6	47.2	214.6	144.45	357
10	25.3	217.8	129.4	315

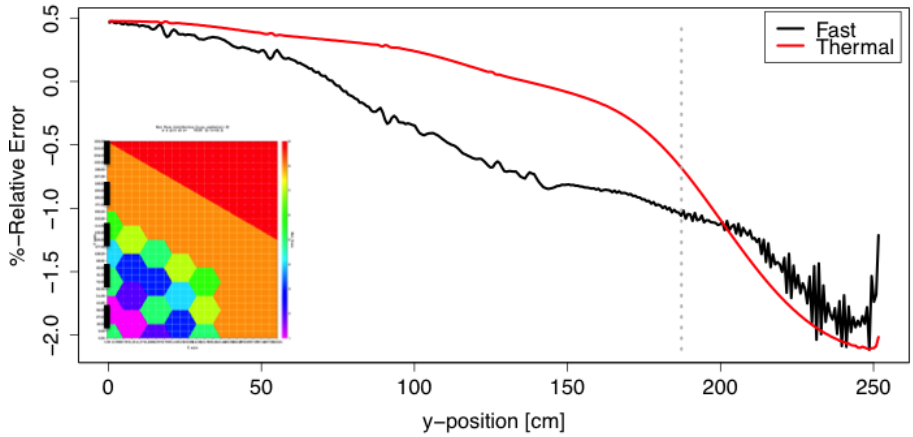
The analysis summarized in Table 4.14 shows that, although the k relative errors seemed to be increasing with quadrature order (see Table 4.13a), the magnitude of the maximum flux error is decreasing with an increasing quadrature order. These errors may seem to be quite large, but the errors occur at the outer edge of the reactor core within the reflector region. At this location the fluxes are extremely small, see Figure 4.12, which causes the error calculations to yield large values even though the actual difference in flux at these locations is negligible. In addition, the flux in this region has a negligible effect on the overall core solution because the magnitude is so small. The magnitude of the flux error is reduced by half when going from S_6 to S_{10}. To reduce the effect of angular discretization errors when examining the SDM algorithm, we will use a quadrature order of S_{10} for all following analysis.

Figure 4.9 shows the flux percent relative error profiles for horizontal (Figure 4.9a) and vertical (Figure 4.9b) slices of the 2D HTTR core. These figures demonstrate how the errors are exaggerated in the reflector regions, where the flux is extremely small, near zero in some cases. Again, note that in these figures the maximum errors are near the periphery of the modeled region. At these locations the magnitude of the flux is very small and the effect on the overall solution is minimal; therefore, these relatively large errors ($\leq$2.0%) are of little

concern.



(a) Horizontal slice at $y = 0.0$cm.



(b) Vertical slice at $x = 0.0$cm.

Figure 4.9 2-D HTTR Core: Scalar flux error distribution for the 1-D slices as compared to the S_{14} case with meshing of $mfp/2$.

Convergence Parameters

This model will produce some very sharp flux gradients in the areas where there is no fissionable material; this means that there will be regions that have near zero fluxes which may lead to some convergence issues. The main issue, that was previously discussed, is the fact that these regions with near zero fluxes may falsely give large errors because of

the roundoff errors associated with single precision values. Therefore regional flux tolerances may be helpful in achieving convergence in a reasonable amount of time. The other tolerance of specific interest in these calculations is k , which is an environmental parameter.

To examine the effect of the regional flux tolerances, three different regions will be considered. These regions will be the "core", comprised of the fuel and control elements, the reflector, and the void region. Figure 4.10 shows the boundaries that will be used between the different regions. Note that in order to apply different tolerances the regions must be defined by coarse meshes.

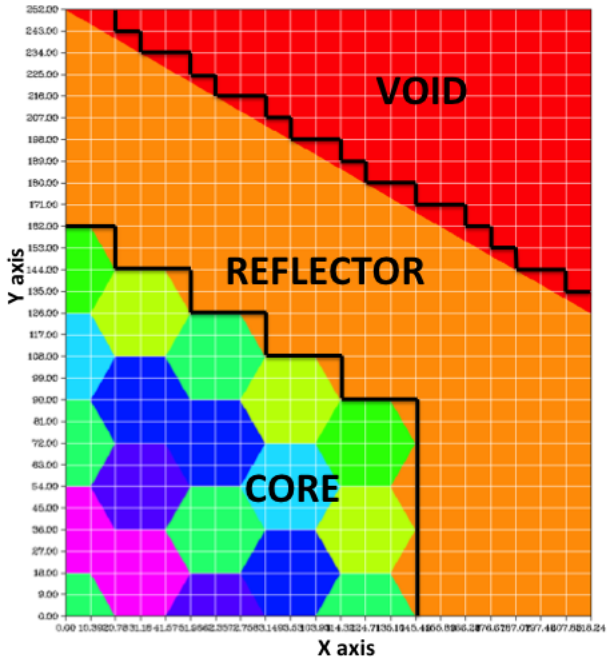


Figure 4.10 2-D HTTR Core: Flux tolerance regions.

First the effect of flux tolerance in the void and reflector regions will be considered; to examine this we must fix the k tolerance and the core flux tolerance. The k tolerance is set to 10^{-5} and the core flux tolerance is set to 10^{-5}. Any cases where the tolerance of the void region is smaller than the reflector region will not be considered. It is realized that the

tolerance in the reflector and void regions does not have a significant effect of the overall core solution. Table 4.15 demonstrates the effect of the tolerances in these regions on the number of iterations required.

Table 4.15 2-D HTTR Core: Number of inner iterations required for different void and reflector region flux tolerances.

ϕ_{void}	$\phi_{refl.}$ **Tolerance**			
Tol.	$\mathbf{10^{-2}}$	$\mathbf{10^{-3}}$	$\mathbf{10^{-4}}$	$\mathbf{10^{-5}}$
$\mathbf{10^{-2}}$	33075	33088	33540	33921
$\mathbf{10^{-3}}$	–	33089	33530	33919
$\mathbf{10^{-4}}$	–	–	34386	34429
$\mathbf{10^{-5}}$	–	–	–	34580

Table 4.15 implies that tighter tolerances in these regions force extra inner iterations in an attempt to converge the fluxes in these regions, though these extra iterations do not significantly effect the computation time, but may become an issue when trying to converge fluxes using the SDM algorithm. Flux tolerances of 10^{-3} will be used for the reflector and void regions hereafter.

With these tolerances set, it is necessary to select k tolerance and core flux tolerance that produce a solution with an acceptable level of accuracy. The outer (k) and inner (core flux) tolerances are simultaneously refined and k relative error is calculated against the most refined case, which has an outer tolerance of 10^{-6} and an inner tolerance of 10^{-6}. Table 4.16 demonstrates the effect of k tolerance and core flux tolerance on solution accuracy.

Table 4.16 2-D HTTR Core: Absolute k relative errors, in pcm, for different combinations of k-eigenvalue and core flux convergence tolerances.

k-	**Core Flux Tolerance**		
Tol.	$\mathbf{10^{-4}}$	$\mathbf{10^{-5}}$	$\mathbf{10^{-6}}$
$\mathbf{10^{-3}}$	213	107	79
$\mathbf{10^{-4}}$	174	53	39
$\mathbf{10^{-5}}$	169	17	1
$\mathbf{10^{-6}}$	169	17	0

Table 4.16 demonstrates that an acceptable level of accuracy ($\epsilon_k \leq 20$ pcm) is obtained with a k tolerance of 10^{-5} and a core flux tolerance of 10^{-5}. The reference solution results will be presented in the following section. The tolerances decided upon in this section will be used when evaluating the SDM algorithm with the 2D HTTR model.

26-group Reference Calculation Results

Table 4.17 presents the TITAN modeling parameters.

Table 4.17 2-D HTTR Core: Modeling parameters.

Parameter	**Value**
Meshing	$mfp/2$
Quadrature Order	10
Scattering Order	P0
Differencing Scheme	Diamond w/ fixup
k-tolerance	10^{-5}
flux-tolerance[1]	10^{-5} (core) 10^{-3} (refl.) 10^{-3} (void)
Max. flux iters.	10
Max. k iters	500

[1]Regional flux tolerances were implemented.

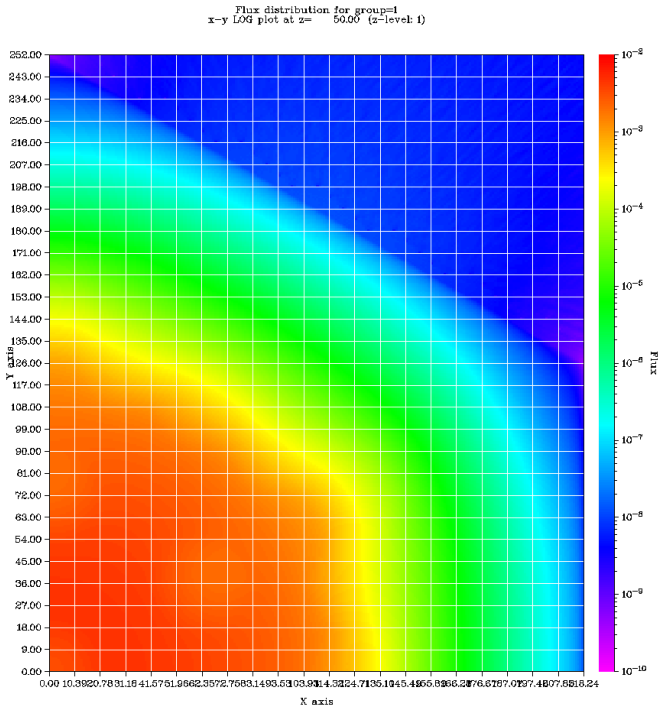
Table 4.18 summarizes the reference solution results.

Table 4.18 2-D HTTR Core: 26-group reference case results.

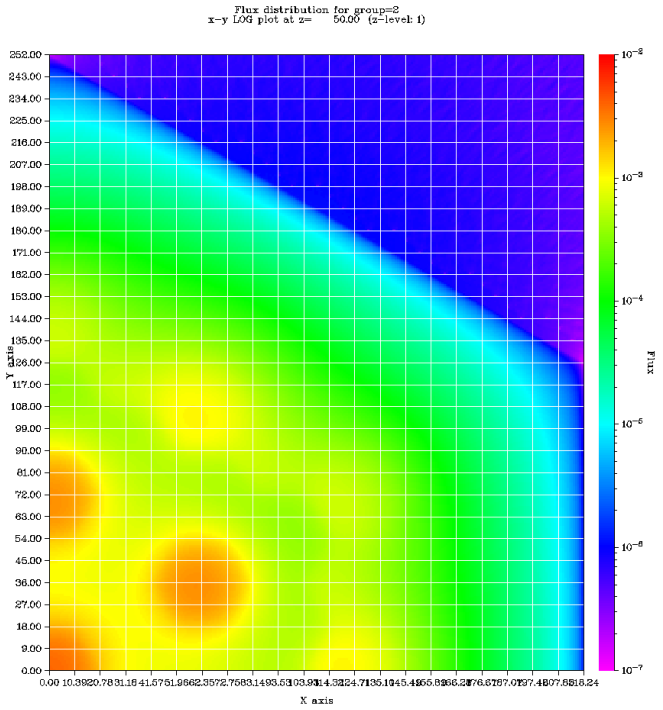
k	**Iterations**[1]	**Time - [hrs.]**
1.04622	133	15.8

[1]Total number of k (Outer) iterations.

Figure 4.11 shows the fast (groups 1-17) and thermal (groups 18-26) flux distributions. The behavior of the flux is as expected with the fast flux peaking in the core, while the thermal flux is lower in this region. Note that in the thermal flux plot there are peaks in the moderator regions, this is where the fast neutrons are slowed down to thermal energies. In both the fast and thermal flux plots the magnitude of the flux significantly decreases, to nearly zero, within the reflector regions and continues to decrease moving outwards to the void region.



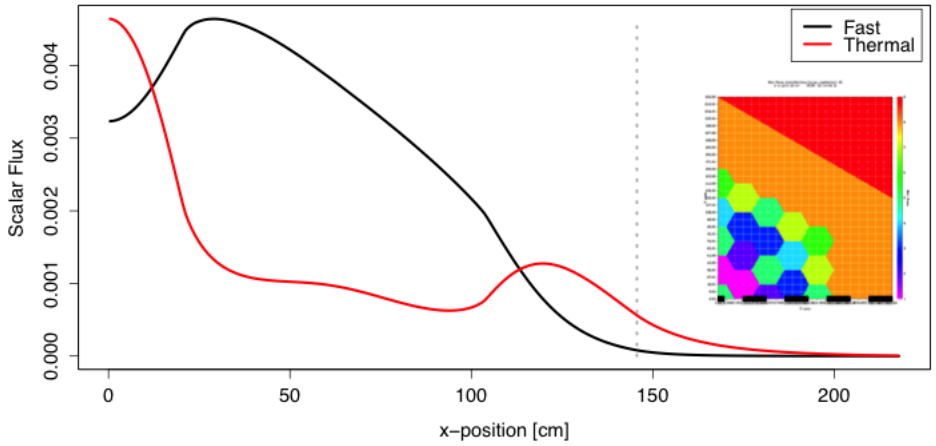
(a) Fast (groups 1-17).



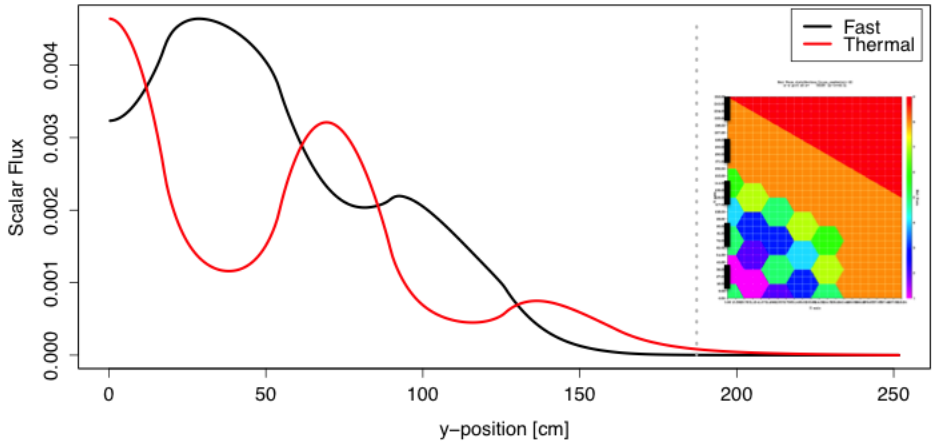
(b) Thermal (groups 18-26).

Figure 4.11 2-D HTTR Core: 26-group reference case flux distribution.

To better illustrate these trends horizontal, Figure 5.4a, and vertical, Figure 5.4b, flux profiles are created along $y = 0.0$cm and $x = 0.0$cm lines, respectively.



(a) Horizontal slice at $y = 0.0$cm.



(b) Vertical slice at $x = 0.0$cm.

Figure 4.12 2-D HTTR Core: 26-group reference 1D flux profiles.

These figures clearly show the expected flux behavior. Thermal flux peaks in the moderator regions while the fast flux is low. In the fuel regions the fast flux peaks and the thermal flux is low. Figure 4.12 shows how the flux drastically decreases once in the reflector region. It demonstrates how low the flux is in this region, this will be an important point when addressing flux errors when evaluating the SDM performance.

Chapter 5

Testing the SDM

Two test problems are presented here for examining the SDM algorithm in TITAN. These two test problems are: i) 1D HTTR core [24], and ii) 2D homogenized HTTR core [16].

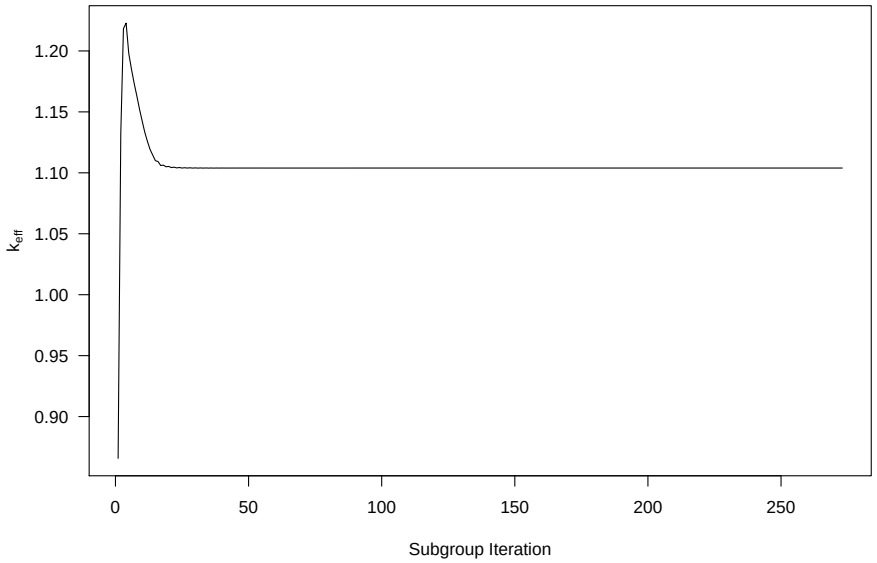
This section outlines the results of testing the TITAN-SDM on the 1-D and 2-D HTTR core benchmark problems discussed in Chapter 4 Sections 4.2.1 and 4.2.2.2. The effect of coarse-group structure, as defined by the number of coarse groups and boundary definition is examined. As previously mentioned, the SDM algorithm introduces three additional possible convergence criteria; the effect of these criteria will be addressed.

5.1 SDM Convergence Criteria

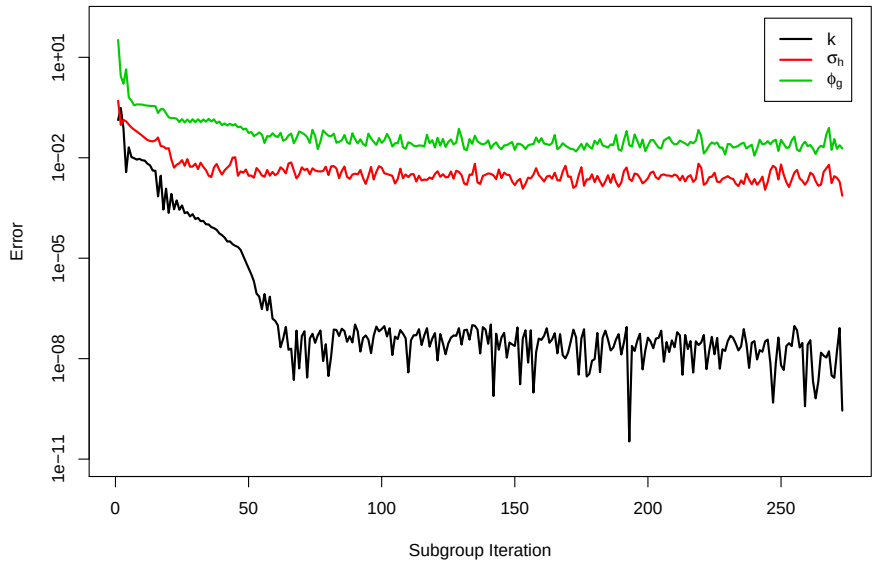
The implementation of the SDM algorithm introduces two new convergence parameters in addition to the standard k and *flux* tolerances; these tolerances are the coarse-group (CG) cross sections and fine-group (FG) fluxes. The errors for these new parameters and eigenvalue k are calculated at the end of every subgroup or recondensation loop, after the fine-group

fluxes are extracted and cross sections are recondensed.

The author of the SDM paper [10] states that the model will be converged once the coarse-group cross sections have met the convergence criteria. After some detailed analysis, it was observed that it is only necessary to converge on the subgroup eigenvalue for the model to be adequately converged. It has been shown that CG cross sections and FG fluxes will never reach below a certain level, regardless of how many subgroup loops are performed. Therefore, if the set tolerance, for either CG cross-section or FG flux, is too low the simulation will continue to run with little to no improvement in the eigenvalue or the flux spectrum. Figure 5.1a shows the typical behavior of k versus subgroup iteration.



(a) k convergence trend.



(b) Subgroup iteration error trends, k , coarse-group cross-section, and fine-group flux.

Figure 5.1 1-D HTTR: 6 coarse-group TITAN-SDM run with convergence parameters k-tol = 10^{-6}, σ_h-tol= 10^{-3}, and ϕ_g-tol=1.0 .

Fiigure 5.1a indicates that k has converged after approximately 30 subgroup iterations, while computation was continued because of apparent lack of convergence of cross sections and

fine-group fluxes. This is demonstrated in Figure 5.1b, which presents the error behavior of different variables including k , CG cross-section, and FG flux as a function of subgroup iterations. Figure 5.1b shows that k was effectively converged after approximately 70 iterations. The fluctuations observed in eigenvalue error, ϵ_k , are purely a result of numerics, as error calculations are carried out with single precision numbers. At this point the CG cross-section error and FG flux error are as low as can be achieved with the given model; it is only by chance that the CG cross-section actually dropped below the specified tolerance and the simulation exited before the maximum number of iterations was met.

All of the tests that are presented hereafter have converged only on the subgroup loop eigenvalue and the standard transport parameters (k and flux) as applied to the coarse-group loops. Note that the same k tolerance is used for the CG loops and the subgroup loops.

5.2 1-D HTTR Core

The same meshing, scattering order, and quadrature order are used as specified for the 47-group reference case. See Chapter 4 Section 4.2.1 for other modeling parameters used. Table 5.1 shows the convergence parameters used for testing TITAN-SDM on the 1-D HTTR core.

Table 5.1 1-D HTTR: TITAN-SDM convergence parameters.

Parameter	Value
k-tolerance	10^{-7}
flux-tolerance[1]	10^{-5} (core) 10^{-1}(refl.)
CG cross-section tol.	10^{-1}
FG flux tol.	1.0
Max. CG flux iters.	10
Max. CG k iters	40
Max. subgroup iters	200

[1]Regional flux tolerances were implemented.

5.2.1 Coarse-group Testing

To effectively examine the performance of the SDM algorithm, we have considered several coarse-group structures by collapsing the 47-group structure. Table 5.2 shows the specific coarse-group structures examined, as defined by fine-group boundaries as the lower boundary of the corresponding coarse group.

Table 5.2 1-D HTTR: SDM coarse-group testing structure.

	Coarse-Group ID #															
Test Case	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	Fine-Group ID # (Lower Energy Bound)															
2cg	35	47														
3cg	10	35	47													
6cg	7	10	24	35	40	47										
8cg-Fast	3	7	9	10	24	35	40	47								
14cg-Epi	7	10	13	16	18	19	20	24	27	30	33	35	40	47		
16cg-Therm	7	10	24	35	36	37	38	39	40	41	42	43	44	45	46	47

The 2cg, 3cg, and 6cg structures are the standard group structure used for HTTR problems [16]. The other coarse-group structures were decided on by independently refining the different energy regions: fast, epithermal, and thermal. Within the epithermal region there was a sharp peak in the cross sections at group 19; when refining the epithermal region we

attempt to capture this peak by isolating it.

Table 5.3 summarizes k and performance parameters for the aforementioned coarse-group structures. The maximum percent relative flux errors are calculated for fast and thermal fluxes within the fuel regions and overall, in the whole core. Note the maximum flux errors alway occurred within the reflector region.

Table 5.3 1-D HTTR: Summary of TITAN-SDM testing results.

			SG[2]	CG[3]		Max Flux Error[4]	
	k	***pcm***[1]	**Iters**	**Iters**	**Speedup**	**Fuel Region**	**Reflector**
47g-Ref	1.10330	ref	n/a	551	1.00	ref.	ref.
2cg	1.10336	5	132	3994	3.14	0.26	2.30
3cg	1.10340	9	91	2272	3.92	0.25	2.20
6cg	1.10341	10	37	1147	6.16	0.31	2.80
8cg-Fast	1.10338	7	36	1107	5.22	0.26	2.20
14cg-Epi	1.10355	23	40	1237	4.17	0.53	5.20
16cg-Thrm	1.10333	3	48	1271	3.56	0.09	0.71

[1]percent milli reactivity error; [2]subgroup; [3]coarse-group; [4]maximum absolute flux error

Table 5.3 shows that all group structures provide good agreement in k , under ± 30 *pcm*, and flux within the fuel region, under 0.5% error. The maximum errors in the whole model always occur in the outer reflector regions, and in general these errors remain under around 3.0%, excluding the epithermal coarse-group structures. The coarse groups with refinement in the epithermal region show errors larger than 3.0% in the reflector region. It is worth noting that these same cases also have the largest *pcm* k differences. This could mean that the SDM algorithm does not work as well when attempting to capture peaks within the resonance region, as was attempted when refining in the epithermal range.

The observed speedups indicate that computation time is dependent on the number of coarse groups chosen, and how they are selected. Too many or too few groups (see 2cg and 16cg-Thrm cases) will result in lower speedups. The maximum speedup was 6.16 with the 6cg

arrangement, which also had very good eigenvalue agreement with 10 *pcm* and very low flux errors in the fuel region and reflector, 0.31% and 2.80% respectively.

Figure 5.2 shows the error in flux and fuel-pin fission density for the 6cg case as compared to the 47-group reference solution. The error distribution is representative of the observed trends. The relatively large errors observed in the reflector region can be attributed to the issue of calculation of relative differences of very small numbers in single-precision.

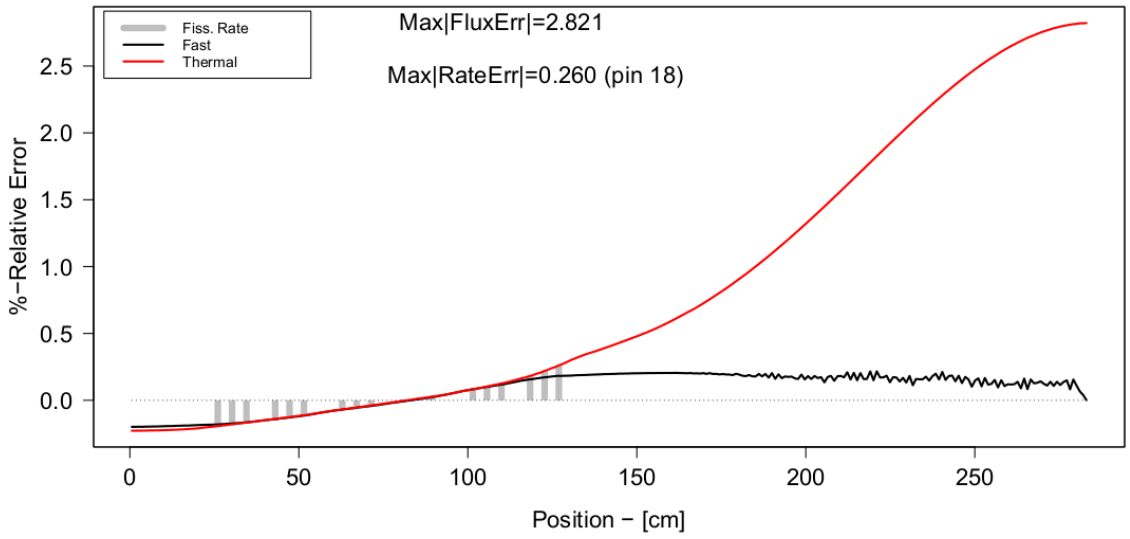


Figure 5.2 1-D HTTR: 6cg flux and fuel pin fission rate errors.

5.2.2 Reaction Rate Analysis

Reaction rates were calculated using Equation (4.3) within the different blocks of the core assembly. Table 5.4 shows the fission reaction rate (fr) and the non-fission capture rate (nfcr) for all 8 blocks of the 47-group reference solution (displayed in bold face text). The table also includes the percent relative errors, fr-%err and nfcr-%err for fission and non-fission capture rates, respectively, for all of the coarse-group SDM cases as compared to the

47-group reference solution.

Table 5.4 1-D HTTR: Reaction rate errors compared to the 47-group reference solution.

	Center Cont.	Fuel 1	Fuel 2	Fuel 3	Outer Cont.	Refl.	Refl.	Refl.
	Reference Reaction Rates - [#/cm^3-s]							
47g								
fr[1]($\times 10^3$)		14.14	13.51	15.54				
nfcr[2]($\times 10^3$)	3.37	14.13	12.30	13.55	3.18	2.77	1.24	0.36
	%-Relative Difference (vs. 47g Reference)							
2cg								
fr-%err[3]		-0.13	-0.02	0.14				
nfcr-%err[4]	-0.17	-0.12	-0.02	0.13	0.43	0.89	1.58	2.15
3cg								
fr-%err		-0.11	-0.01	0.13				
nfcr-%err	-0.16	-0.12	-0.02	0.12	0.41	0.84	1.50	2.04
6cg								
fr-%err		-0.16	-0.02	0.17				
nfcr-%err	-0.22	-0.16	-0.02	0.16	0.52	1.05	1.90	2.59
8cg-Fast								
fr-%err		-0.14	-0.01	0.15				
nfcr-%err	-0.20	-0.14	-0.02	0.13	0.42	0.84	1.51	2.07
14cg-Epi								
fr-%err		-0.23	-0.04	0.26				
nfcr-%err	-0.30	-0.23	-0.06	0.24	0.91	1.93	3.51	4.77
16cg-Thrm								
fr-%err		-0.05	0.00	0.05				
nfcr-%err	-0.07	-0.05	-0.01	0.04	0.14	0.27	0.48	0.65

[1]fission reaction rate; [2]non-fission capture reaction rate; [3]fission reaction rate percent relative error; [4]non-fission capture reaction rate percent relative error

The above reaction rate analysis demonstrates the expected trend within the TITAN 47-group reference solution, the maximum reaction rates are in the outer two fuel regions and the non-fission capture rate decreases moving further away from the core center. For all coarse-group structures, the SDM calculation demonstrates percent relative errors are well below 1% indicating exceptional accuracy with the TITAN solution. The magnitude of the errors increases in moving further away from the core center, in agreement with the observed

trends of the flux. The 16cg-Thrm case shows the lowest relative errors, this is in agreement with the observed trend of the k relative errors.

An analysis of fuel pin fission rate errors was completed by first calculating the fission rates in each of the 18 fuel pins. The percent relative errors were calculated for each fuel pin i versus the 47-group reference solution. Table 5.5 compares maximum relative error and RMS for the different coarse-group structures. The specific error calculations used are defined in Equations (4.5) and (4.6).

Table 5.5 1-D HTTR: Fuel pin fission rate error analysis.

	RMS	**MAX**
2cg	0.11	0.22
3cg	0.10	0.20
6cg	0.14	0.26
8cg-Fast	0.12	0.22
14cg-Epi	0.21	0.43
16cg-Thrm	0.04	0.08

The fuel pin fission rate analysis shows that, in general, the root-mean-square values are below 0.15%; the exception being the 14cg-Thermal case, which as previously mentioned also shows the largest flux and k errors. This finding reiterates that coarse-group refinement within the epithermal range may not produce the most accurate results. All of the maximum relative differences are below 0.5%, which indicates good agreement with the TITAN reference solution. This analysis shows that the TITAN-SDM is capable of reproducing the TITAN reference solution to an acceptable level of accuracy.

5.3 2-D HTTR Core

The same meshing, scattering order, and quadrature order are used as specified for the 26-group reference case. See Chapter 4 Section 4.2.1 for other modeling parameters used. Table 5.6 outlines the TITAN-SDM convergence parameters for the 2-D HTTR core.

Table 5.6 2-D HTTR Core: TITAN-SDM modeling parameters.

Parameter	**Value**
k-tolerance	10^{-6}
flux-tolerance[1]	10^{-5} (core) 10^{-3} (refl.) 1.0 (void)
CG cross-section tol.	10^{-1}
FG flux tol.	1.0
Max. CG flux iters.	10
Max. CG k iters	20
Max. subgroup iters	200

[1]Regional flux tolerances were implemented.

5.3.1 Coarse-group Testing

We will again consider different coarse-group structures, by collapsing the 26-group structure, to examine the effect on the TITAN-SDM solution. Table 5.7 shows the coarse-group structures, as defined by fine-group boundaries as the lower boundary of the corresponding coarse-group.

Table 5.7 2-D HTTR Core: TITAN-SDM coarse-group testing structure.

Test Case	Coarse-Group ID #												
	1	2	3	4	5	6	7	8	9	10	11	12	13
	Fine-Group ID # (Lower Energy Bound)												
2cg	17	26											
3cg	6	17	26										
6cg	3	6	12	17	21	26							
8cg-Epi	3	6	9	12	15	17	21	26					
13cg-Thrm	3	6	12	17	18	19	20	21	22	23	24	25	26

The cross sections were plotted versus energy groups and the trends were examined to determine appropriate bounds for the fast, epithermal, and thermal energy ranges. These selections were chosen by an informed decision, not bt the actual value of the energy level that defines group bounds (this information was not available). The coarse-group structures were decided on by independently refining the different energy regions: fast, epithermal, and thermal.

Table 5.8 summaries k and performance parameters for the tested coarse-group structures presented in Table 5.7. The maximum percent flux errors are calculated for fast and thermal fluxes within the fuel regions and overall, in the whole core.

Table 5.8 2-D HTTR Core: Summary of TITAN-SDM testing results.

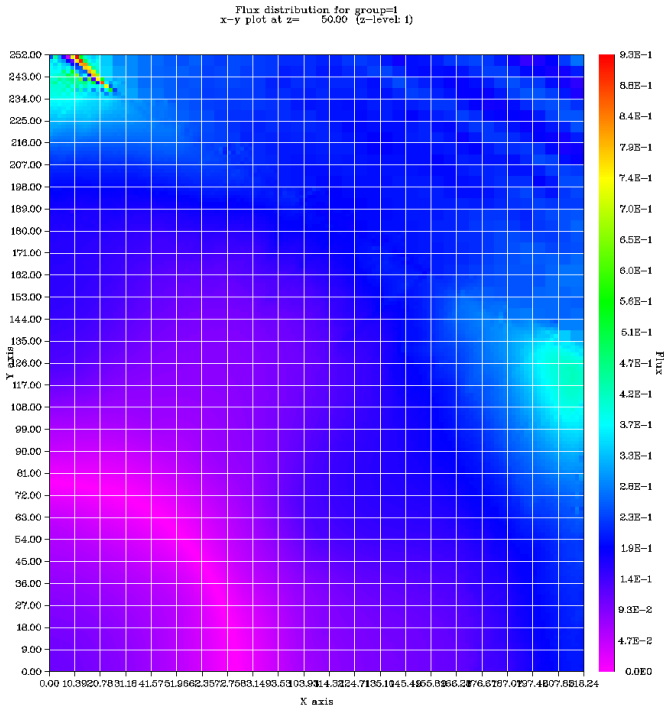
			SG[2]	CG[3]		Flux Error[4]	
	k	***pcm***[1]	**Iters**	**Iters**	**Speedup**	**Fuel Region**	**Reflector**
26g-Ref	1.04472	ref.	n/a	271	1.00	ref.	ref.
2cg	1.04482	10	80	1077	1.08	0.26	6.1
3cg	1.04480	8	33	475	2.05	0.24	6.2
5cg	1.04490	17	21	288	2.44	0.28	6.5
6cg	1.04479	7	24	263	2.24	0.37	7.0
8cg-Epi	1.04483	11	30	328	1.63	0.35	6.9
13cg-Thrm	1.04464	-8	16	252	1.21	0.12	0.5

[1]percent milli reactivity error; [2]subgroup; [3]coarse-group; [4]maximum absolute flux error

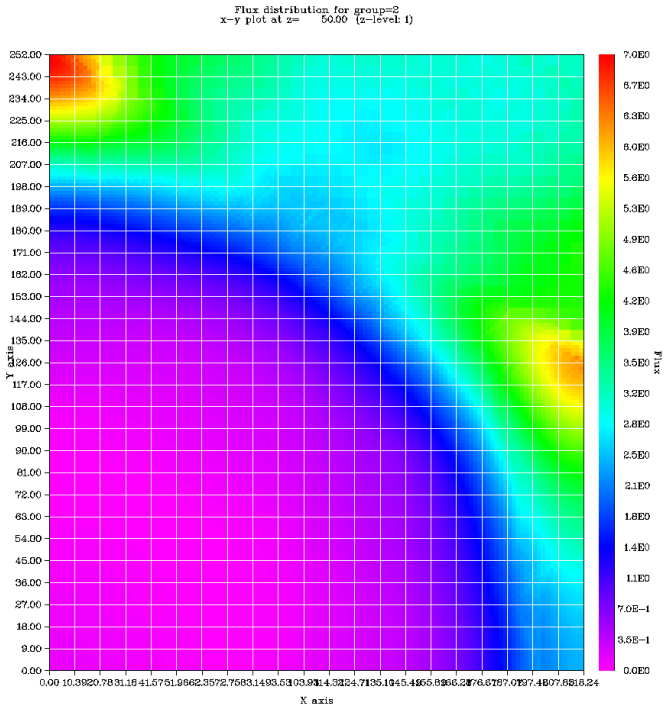
Table 5.8 shows that all group structures give good agreement in eigenvalue, under ± 20 *pcm*, and flux in the fuel region, under 0.4% relative error. Here the maximum errors occurred in the void region, outside of the reflector; these relative errors remain under 7%. These errors may seem large, but they have very little impact on the overall core solution because they are so far away from the fuel region. Therefore these large errors are of little concern.

Speedups for the 2-D homogenized core are not as large as those for the 1-D core. The 5cg group structure shows a speedup of 2.44; considering that the reference model takes over 11 hours to complete, a speedup greater than 2 is quite significant. Again here it can be seen that the coarse-group arrangement affects the computation time; choosing too few or too many groups will cause slower convergence.

The following figures show flux errors of the 5cg case as compared to the 26-group reference solution. Figure 5.3 shows the 2-D flux error distribution; note that in this figure shows the absolute value of the flux relative errors. If you follow the zero line this is where the fluxes go from positive to negative, moving from the core to the reflector. This effect is better observed by viewing the 1-D flux error profiles of Figure 5.4.



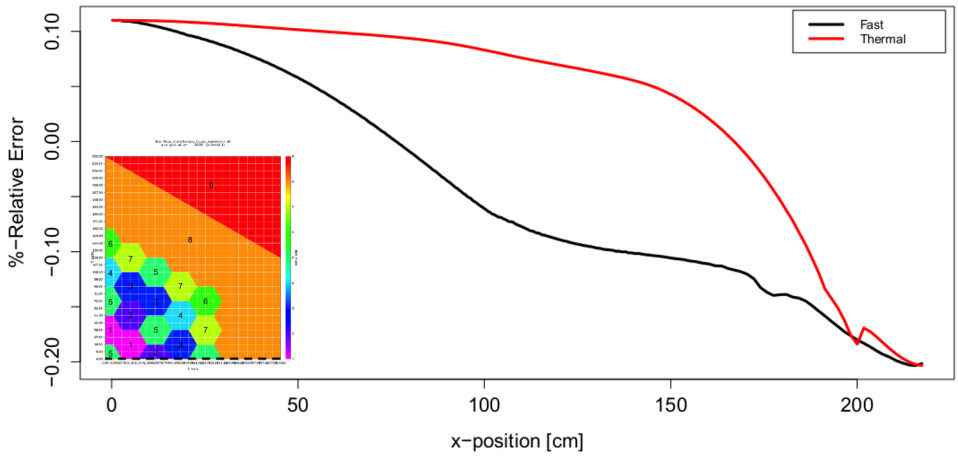
(a) Fast (groups 1-17).



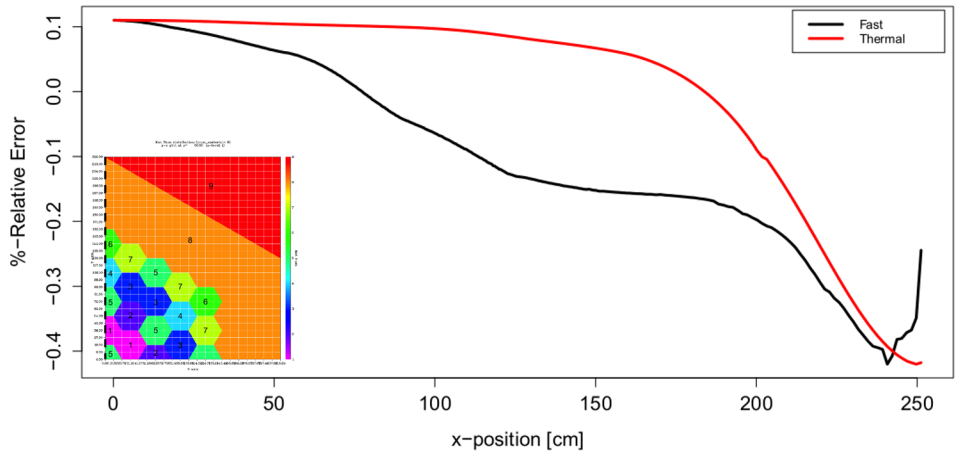
(b) Thermal (groups 18-26).

Figure 5.3 2-D HTTR Core: 6cg TITAN-SDM flux relative error distribution.

The 1-D flux error profiles in Figure 5.4 show much more clearly how the flux relative error transitions from positive to negative when moving from the fuel region to the reflector. It also shows that the maximum relative error is occurring out in the reflector region.



(a) Horizontal slice at y=0.0cm.



(b) Vertical slice at x=0.0cm.

Figure 5.4 2-D HTTR Core: 1-D flux error profiles.

Chapter 6

Conclusions and Future Work

This study has shown that the performance of TITAN-SDM is accurate and efficient in modeling both the 1-D and 2-D HTTR core benchmark problems. The majority of eigenvalue relative differences were below ≈10 *pcm* for both benchmark problems. The only the cases to show k differences above this level were those with epithermal subgroup refinement and are discussed below. The maximum flux relative error within the core for both problems is below 0.4% and the overall flux relative errors remained below 3.0% for the 1-D core and 7.0% for the 2-D core. Again, the cases with epithermal subgroup refinement exceeded this level of error. These overall relative errors may seem large, however, recall, that these errors occurred in the reflector regions where the flux was very small and had negligible impact on the overall core solution and are therefore not of major concern. The 1-D problem showed a maximum speedup of 6.16 for the 6cg case and the 2-D problem showed a maximum computation speedup of 2.44 for the 5cg case.

The SDM algorithm is very sensitive to specified convergence criteria and tolerance levels. As a result, it was necessary to implement regional flux tolerances into the TITAN-SDM code to handle problems which exhibit sharp flux gradients, e.g., HTTR core, specifically in

regions that could potentially produce zero (or very low) fluxes. This is primarily due to the fact that TITAN uses single precision values in calculating errors and fluxes of successive iterations. This issue was not observed by Douglass and Rahnema in their 1-D transport code [10], used to test the SDM algorithm, because double precision values were used for error calculations.

It has been observed that, for a TITAN-SDM solution to be converged, it was *not* necessecary to converge on both the subgroup eigenvalue and coarse-group cross-section, as was stated by the authors of the SDM papers [29, 10]. It was sufficient to solely converge on the subgroup eigenvalue, assuming that the coarse-group fluxes and eigenvalue calculations had converged within the coarse-group transport sweeps.

It was noticed that the selection of number of coarse groups had an effect on calculation time. If too few or too many groups were used the computation required more time to converge. This is due to the following: (i) selecting too few groups causes issues preserving the energy-angle coupling due to the large amount of information that must be preserved in these few groups; (ii) selecting too many groups slows down computation because more coarse-group transport sweeps must be completed, essentially defeating the purpose of the method. The selection of coarse-group structure, defining coarse-group boundaries as a function of subgroups, also affected solution time and accuracy. It was demonstrated that when too many coarse groups were defined in the epithermal range the accuracy of the calculated solution, both flux distribution and eigenvalue, decreased and computation time increased. Due to the presence of resonances in the epithermal energy range, transport calculations can be difficult; therefore, when using SDM it is best to avoid too many coarse groups in the epithermal range.

To avoid issues of false convergence, TITAN-SDM require two consecutive subgroup iterations to fall below the specified tolerance. It is important to note that there are other

approaches that can be used to counter the issue of false convergence that may be considered for use.

Recently an investigation to the extension of the SDM methodology to fixed-source shielding calculations has been initiated [30]. Further work will include a methodology for automatic selection of the number of coarse groups and coarse-group structure based on problem objective.

Appendices

Appendix A

TITAN Input for 2-D HTTR Fuel Assembly

```
/* TITAN input file hexasy_titan.inp
/* generated by PENMSHXP version 2.67b (Aug 2011)
/* total number of fine meshes: 40000
/* total number of coarse meshes: 16
/* Number of zlevs: 1
/* Number of coarse mesh per z lev: 16
/*************************************************************
#0  /general transport parameter quadrature
/acceptable keywords: nquad, tquad, oquad, splitq(multi), ncmesh, numsrc, nummat, numgrp
/ nquad: number of quadratures.
/ tquad: type of quadratures, NumOfEntry=nquad (0=LevelSym, 1=LegendreCheby)
/ oquad: order of quadratures, NumOfEntry=nquad (even number)
/ splitq: ordinate splitting setup
/   splitq=QuadId, #Splitting, SplitDirIDs, SplitOrders, SplitType, Topnums, alpha
/   QuadId=1, 2,..or nquad, #Splitting=num of splitting directions
/   #Splitting= total num of splitting directions in QuadID
/   SplitDirIDs=splitting direction IDs in the base quadrature,NumOfEntry=#Splitting
```

```
18 /   SplitOrder=splitting order for each splitting direction,↩
     NumOfEntry=#Splitting
19 /   SplitType: 0=rectangular, 1=Pn-Tn, 2=circular, NumOfEntry=#↩
     Splitting
20 /   Topnum: rectangular:unused; Pn-Tn:Num dirs on top level; ↩
     circular:# circles. NumOfEntry=#Splitting
21 /   alpha:rectangular/Pn-Tn:unused(angular range for future);↩
     circular:derail angle.NumOfEntry=#Splitting
22 / ncmesh: # of coarse mesh along x, y ,z . NumOfEntry=3
23 / numsrc: # of sources . NumOfEntry=1
24 / nummat: # of materials . NumOfEntry=1
25 / numgrp: # of groups . NumOfEntry=1
26 ncmesh=4  4  1
27 numsrc=0
28 nummat=14
29 numgrp=6
30 nquad=1
31 tquad=0
32 oquad=6
33 /*****************************************************************
34 #1  /Section 1 : Geometry setup
35 /acceptable keywords: dcpara, xcmbnd, ycmbnd, zcmbnd, cmxfin, ↩
     cmyfin, cmzfin, cmsolv,cmdiff,cmquad
36 / dcpara: spatial parallel factor along x, y, z (future version)
37 / xcmbnd, ycmbnd, zcmbnd: x,y,z coarse mesh boundaries, NumOfEntry↩
     =ncmesh+1
38 / cmxfin, cmyfin, cmzfin: fine mesh number along x,y,z for each ↩
     coarse mesh, NumOfEntry=TotNumOfCM
39 / cmsolv: solver ID for each corase mesh, 0=Sn, 1=Characteristics
40 / cmdiff: Differencing Scheme ID for each corase mesh, 1=DD with ↩
     fixup, 2=DTW
41 / cmquad: Quadrature ID for each coarse mesh, available values↩
     =1,2,... or nquad
42 dcpara=1,1,1
43 xcmbnd= 0.00000E+00  1.03900E+01  2.07800E+01  3.11800E+01  ↩
     4.15700E+01
44 ycmbnd= 0.00000E+00  9.00000E+00  1.80000E+01  2.70000E+01  ↩
     3.60000E+01
45 zcmbnd= 0.00000E+00  1.00000E+02
46 cmxfin=16R50
47 cmyfin=16R50
48 cmzfin=16R1
49 cmsolv=16R0
50 cmdiff=16R1
51 cmquad=16R1
```

```
52 /*****************************************************************
53 #2  /Section 2 : Fine mesh mat. number
54 /acceptable keywords: cmmatn (multi)
55 / cmmatn=CM#, mat # for each fine mesh in this CM
56 cmmatn=1
57 9R3 12R10 7R9 22R10 2Q50 9R3 12R10 6R9 23R10 9R3 14R10 3R9 24R10 9↩
      R3 41R10 8R3
58 42R10 1Q50 7R3 43R10 6R3 44R10 5R3 45R10 3R3 157R10 4R3 43R10 9R3 ↩
      40R10 12R3
59 37R10 14R3 35R10 16R3 1Q50 33R10 18R3 1Q50 31R10 19R3 1Q50 3 4Q50 ↩
      30R10 19R3
60 32R10 2Q50 17R3 34R10 1Q50 10 14R3 37R10 12R3 39R10 10R3 42R10 6R3↩
       135R10 2R3
61 46R10 6R3 42R10 10R3 38R10 13R3 36R10 14R3 1Q50 3 34R10 16R3 1Q50 ↩
      3 32R10 18R3
62 2Q50 32R10 9R3
63 cmmatn=2
64 736R10 4R3 43R10 9R3 40R10 11R3 38R10 13R3 36R10 15R3 34R10 17R3 1↩
      Q50 32R10
65 19R3 13R10 5R9 1Q50 9 12R10 19R3 12R10 7R9 2Q50 12R10 19R3 13R10 1↩
      Q50 5R9
66 13R10 19R3 14R10 3R9 14R10 19R3 32R10 1Q50 17R3 34R10 1Q50 15R3 36↩
      R10 13R3
67 39R10 10R3 42R10 6R3 109R10 3R3 19R10 6R3 19R10 8R3 15R10 10R3 15↩
      R10 12R3
68 12R10 12R3 13R10 14R3 10R10 14R3 11R10 15R3 9R10 16R3 9R10 17R3 8↩
      R10 17R3 8R10
69 17R3 7R10 18R3 7R10 19R3 1Q50 6R10 19R3 5R10 20R3 5R10 20R3 1Q50 5↩
      R10 10R3
70 cmmatn=3
71 710R10 4R3 21R10 4R3 19R10 9R3 16R10 9R3 15R10 11R3 13R10 12R3 13↩
      R10 13R3
72 11R10 14R3 11R10 15R3 9R10 16R3 9R10 17R3 1Q50 7R10 18R3 7R10 19R3↩
       1Q50 6R10
73 19R3 6R10 19R3 5R10 20R3 4Q50 6R10 19R3 6R10 1Q50 18R3 7R10 18R3 7↩
      R10 18R3
74 8R10 17R3 8R10 17R3 8R10 16R3 9R10 16R3 10R10 15R3 10R10 14R3 12↩
      R10 13R3 12R10
75 12R3 14R10 10R3 15R10 10R3 17R10 6R3 19R10 6R3 110R10 3R3 19R10 5↩
      R3 20R10 8R3
76 15R10 10R3 15R10 11R3 13R10 12R3 12R10 14R3 11R10 14R3 10R10 16R3 ↩
      9R10 16R3
77 9R10 16R3 8R10 17R3 8R10 18R3 7R10 18R3 1Q50 6R10 20R3 5R10 19R3 2↩
      Q50 3 5R10
78 10R3
```

```
cmmatn=4
16R10 20R3 5R10 9R3 1Q50 16R10 19R3 6R10 2Q50 9R3 17R10 1Q50 17R3 8R10 8R3
18R10 1Q50 15R3 10R10 7R3 19R10 13R3 12R10 6R3 20R10 11R3 14R10 5R3 22R10 7R3
18R10 3R3 136R10 4R3 44R10 9R3 39R10 12R3 37R10 14R3 35R10 16R3 1Q50 33R10
18R3 2Q50 3 30R10 20R3 4Q50 31R10 1Q50 18R3 32R10 1Q50 10 1Q50 16R3 35R10 14R3
37R10 12R3 39R10 10R3 42R10 6R3 109R10 2R3 46R10 6R3 42R10 10R3 38R10 13R3
36R10 14R3 1Q50 3 34R10 16R3 1Q50 3 32R10 18R3 2Q50 32R10 9R3
cmmatn=5
9R3 32R10 18R3 3Q50 32R10 17R3 34R10 1Q50 15R3 36R10 13R3 37R10 1Q50 10 10R3
42R10 6R3 46R10 2R3 135R10 6R3 42R10 10R3 39R10 12R3 37R10 14R3 35R10 16R3
1Q50 3 32R10 18R3 1Q50 31R10 20R3 5Q50 31R10 18R3 32R10 2Q50 10 16R3 34R10
1Q50 10 14R3 37R10 12R3 40R10 8R3 45R10 2R3 133R10 7R3 18R10 3R3 20R10 11R3
14R10 5R3 19R10 13R3 12R10 6R3 18R10 15R3 10R10 7R3 18R10 16R3 8R10 8R3 17R10
17R3 1Q50 3 6R10 9R3 16R10 19R3 3Q50 3 5R10 9R3
cmmatn=6
10R3 5R10 20R3 5R10 20R3 1Q50 6R10 2Q50 18R3 7R10 18R3 7R10 1Q50 10 17R3 8R10
16R3 9R10 16R3 9R10 15R3 11R10 14R3 11R10 13R3 13R10 12R3 13R10 11R3 16R10 9R3
15R10 8R3 20R10 5R3 20R10 2R3 109R10 7R3 18R10 7R3 17R10 10R3 14R10 11R3 13R10
13R3 12R10 13R3 12R10 14R3 10R10 15R3 10R10 16R3 9R10 16R3 8R10 17R3 8R10 17R3
8R10 18R3 7R10 18R3 7R10 18R3 6R10 19R3 6R10 19R3 1Q50 3 3Q50 5R10 19R3 6R10
19R3 6R10 1Q50 19R3 7R10 1Q50 18R3 7R10 17R3 9R10 16R3 8R10 17R3 9R10 15R3
10R10 15R3 11R10 14R3 11R10 13R3 13R10 12R3 13R10 11R3 16R10 8R3 16R10 9R3
19R10 3R3 22R10 3R3 111R10 4R3 17R10 8R3 21R10 6R3 14R10 11R3 19R10 7R3 12R10
13R3 18R10 8R3 10R10 15R3 17R10 9R3 8R10 16R3 1Q50 3 16R10 9R3 7R10 18R3 16R10
10R3 6R10 19R3 1Q50 15R10 10R3 5R10 20R3 1Q50 15R10
cmmatn=7
```

```
105 10R3 5R10 20R3 5R10 20R3 1Q50 5R10 19R3 6R10 19R3 6R10 1Q50 10 1↩
       Q50 18R3 7R10
106 17R3 8R10 17R3 8R10 17R3 9R10 15R3 10R10 15R3 11R10 14R3 11R10 13↩
       R3 13R10 12R3
107 13R10 11R3 15R10 9R3 16R10 7R3 20R10 5R3 20R10 3R3 109R10 7R3 18↩
       R10 6R3 17R10
108 11R3 14R10 10R3 14R10 13R3 12R10 12R3 12R10 15R3 10R10 14R3 10R10 ↩
       16R3 9R10
109 16R3 9R10 17R3 8R10 17R3 7R10 18R3 7R10 18R3 7R10 19R3 1Q50 6R10 ↩
       19R3 6R10
110 19R3 5R10 20R3 4Q50 6R10 19R3 6R10 1Q50 18R3 8R10 17R3 7R10 1Q50 ↩
       10 16R3 10R10
111 15R3 10R10 15R3 11R10 13R3 11R10 14R3 13R10 11R3 13R10 12R3 15R10 ↩
       9R3 16R10
112 8R3 20R10 3R3 22R10 3R3 132R10 8R3 17R10 4R3 19R10 11R3 14R10 6R3 ↩
       18R10 13R3
113 12R10 7R3 17R10 15R3 10R10 8R3 17R10 16R3 8R10 9R3 16R10 17R3 1Q50↩
        3 6R10 10R3
114 15R10 19R3 2Q50 3 1Q50 5R10 10R3
115 cmmatn=8
116 9R3 32R10 18R3 3Q50 32R10 17R3 34R10 1Q50 15R3 36R10 13R3 38R10 1↩
       Q50 10R3
117 42R10 6R3 46R10 2R3 109R10 6R3 42R10 10R3 39R10 12R3 37R10 14R3 35↩
       R10 16R3
118 33R10 17R3 1Q50 3 1Q50 31R10 19R3 1Q50 3 4Q50 30R10 19R3 32R10 2↩
       Q50 10 16R3
119 34R10 15R3 36R10 1Q50 10 12R3 40R10 8R3 45R10 2R3 137R10 3R3 1Q50 ↩
       2R3 1Q50 3
120 1Q50 3 1Q50 3 2Q50 3 1Q50 13R10 4R9 24R10 9R3 12R10 6R9 1Q50 9 1↩
       Q50 22R10
121 cmmatn=9
122 16R10 20R3 5R10 9R3 16R10 19R3 6R10 2Q50 9R3 17R10 1Q50 17R3 8R10 ↩
       8R3 18R10
123 1Q50 15R3 10R10 7R3 19R10 13R3 12R10 6R3 20R10 11R3 14R10 5R3 22↩
       R10 7R3 18R10
124 3R3 137R10 2R3 45R10 8R3 40R10 12R3 37R10 14R3 35R10 16R3 1Q50 33↩
       R10 18R3 2Q50
125 31R10 20R3 5Q50 31R10 18R3 32R10 1Q50 10 1Q50 16R3 35R10 14R3 37↩
       R10 12R3 39R10
126 10R3 42R10 6R3 109R10 2R3 46R10 6R3 42R10 10R3 38R10 12R3 37R10 14↩
       R3 1Q50 3
127 34R10 16R3 1Q50 3 32R10 18R3 3Q50 32R10 9R3
128 cmmatn=10
129 10R3 5R10 20R3 15R10 1Q50 10R3 6R10 1Q50 19R3 15R10 9R3 7R10 18R3 ↩
       16R10 9R3
```

```
130 8R10 1Q50 16R3 17R10 8R3 10R10 15R3 17R10 7R3 12R10 13R3 18R10 6R3↩
        14R10 11R3
131 19R10 4R3 17R10 8R3 132R10 3R3 22R10 3R3 20R10 8R3 16R10 9R3 15R10↩
        12R3 13R10
132 11R3 13R10 14R3 11R10 13R3 11R10 15R3 10R10 15R3 10R10 16R3 8R10 ↩
       17R3 8R10
133 18R3 1Q50 6R10 19R3 6R10 19R3 2Q50 3 3Q50 5R10 19R3 6R10 19R3 6R10↩
        19R3 7R10
134 1Q50 18R3 7R10 18R3 7R10 17R3 8R10 17R3 9R10 16R3 9R10 16R3 10R10 ↩
       14R3 10R10
135 15R3 11R10 13R3 12R10 13R3 14R10 10R3 14R10 11R3 16R10 7R3 18R10 7↩
       R3 109R10
136 3R3 20R10 5R3 20R10 7R3 16R10 9R3 15R10 11R3 13R10 12R3 13R10 13R3↩
        11R10 14R3
137 11R10 15R3 9R10 16R3 9R10 17R3 8R10 17R3 8R10 17R3 7R10 18R3 7R10 ↩
       19R3 1Q50
138 6R10 19R3 5R10 20R3 1Q50 5R10 20R3 1Q50 5R10 10R3
139 cmmatn=11
140 15R10 20R3 5R10 10R3 1Q50 15R10 19R3 6R10 1Q50 10R3 16R10 1Q50 17↩
       R3 8R10 9R3
141 17R10 1Q50 15R3 10R10 8R3 18R10 13R3 12R10 7R3 19R10 11R3 14R10 6↩
       R3 21R10 8R3
142 17R10 4R3 111R10 3R3 22R10 3R3 19R10 9R3 16R10 8R3 16R10 11R3 13↩
       R10 12R3 13R10
143 13R3 11R10 14R3 11R10 15R3 10R10 15R3 9R10 17R3 8R10 16R3 9R10 17↩
       R3 7R10 18R3
144 7R10 19R3 1Q50 6R10 19R3 6R10 19R3 5R10 20R3 4Q50 6R10 19R3 6R10 1↩
       Q50 18R3
145 7R10 18R3 7R10 18R3 8R10 17R3 8R10 17R3 8R10 16R3 9R10 16R3 10R10 ↩
       15R3 10R10
146 14R3 12R10 13R3 12R10 12R3 14R10 11R3 14R10 10R3 17R10 7R3 18R10 6↩
       R3 110R10
147 2R3 20R10 5R3 20R10 8R3 15R10 9R3 16R10 11R3 13R10 12R3 13R10 13R3↩
        11R10 14R3
148 11R10 15R3 9R10 15R3 10R10 16R3 8R10 17R3 8R10 18R3 7R10 18R3 1Q50↩
        6R10 19R3
149 6R10 19R3 6R10 20R3 1Q50 5R10 20R3 1Q50 5R10 10R3
150 cmmatn=12
151 9R3 12R10 7R9 22R10 1Q50 9R3 12R10 6R9 23R10 9R3 13R10 4R9 24R10 9↩
       R3 41R10 8R3
152 42R10 1Q50 7R3 43R10 6R3 44R10 5R3 45R10 3R3 158R10 2R3 45R10 8R3 ↩
       40R10 12R3
153 37R10 14R3 35R10 15R3 1Q50 3 33R10 18R3 1Q50 31R10 19R3 1Q50 3 4↩
       Q50 30R10 19R3
```

```
32R10 2Q50 17R3 34R10 1Q50 10 14R3 37R10 12R3 39R10 10R3 42R10 6R3 135R10 2R3
46R10 6R3 42R10 10R3 38R10 12R3 1Q50 3 36R10 15R3 34R10 16R3 1Q50 3 32R10 18R3
3Q50 32R10 9R3
cmmatn=13
9R3 32R10 18R3 3Q50 33R10 16R3 34R10 1Q50 10 14R3 36R10 13R3 38R10 1Q50 2R10
8R3 44R10 4R3 107R10 6R3 42R10 10R3 39R10 12R3 37R10 14R3 35R10 16R3 33R10
17R3 1Q50 3 1Q50 31R10 19R3 1Q50 3 4Q50 30R10 19R3 32R10 2Q50 10 16R3 34R10
1Q50 10 14R3 37R10 12R3 39R10 9R3 44R10 4R3 136R10 3R3 1Q50 2R3 1Q50 3 1Q50 3
1Q50 3 2Q50 3 1Q50 14R10 3R9 24R10 9R3 12R10 6R9 1Q50 9 2Q50 22R10
cmmatn=14
10R3 5R10 20R3 5R10 20R3 1Q50 6R10 1Q50 18R3 7R10 18R3 7R10 1Q50 10 17R3 8R10
16R3 9R10 16R3 9R10 16R3 10R10 14R3 11R10 14R3 12R10 12R3 13R10 11R3 15R10
10R3 15R10 8R3 19R10 6R3 19R10 3R3 135R10 6R3 42R10 10R3 38R10 13R3 36R10 15R3
1Q50 3 33R10 17R3 1Q50 3 14R10 3R9 14R10 19R3 13R10 5R9 1Q50 9 12R10 19R3
12R10 7R9 2Q50 12R10 19R3 13R10 1Q50 5R9 13R10 19R3 31R10 1Q50 10 17R3 33R10
1Q50 10 15R3 36R10 13R3 38R10 11R3 40R10 9R3 44R10 4R3 710R10
cmmatn=15
10R3 5R10 20R3 5R10 20R3 5R10 19R3 6R10 1Q50 19R3 7R10 1Q50 18R3 7R10 17R3
8R10 17R3 8R10 17R3 9R10 16R3 9R10 15R3 11R10 14R3 10R10 14R3 13R10 12R3 12R10
12R3 15R10 10R3 15R10 8R3 19R10 5R3 20R10 3R3 109R10 6R3 19R10 6R3 17R10 10R3
15R10 10R3 14R10 13R3 12R10 12R3 12R10 15R3 10R10 14R3 10R10 16R3 9R10 16R3
9R10 17R3 8R10 17R3 7R10 18R3 7R10 18R3 7R10 19R3 1Q50 6R10 19R3 6R10 19R3
5R10 20R3 4Q50 6R10 19R3 6R10 1Q50 18R3 8R10 17R3 7R10 1Q50 10 16R3 10R10 15R3
9R10 16R3 11R10 13R3 11R10 14R3 13R10 11R3 13R10 12R3 15R10 9R3 16R10 9R3
18R10 4R3 21R10 4R3 711R10
cmmatn=16
```

```
9R3 32R10 18R3 2Q50 32R10 17R3 34R10 1Q50 15R3 36R10 1Q50 13R3 38R10 10R3
42R10 6R3 46R10 2R3 135R10 6R3 42R10 10R3 39R10 12R3 37R10 14R3 35R10 16R3
1Q50 3 32R10 18R3 2Q50 3 30R10 20R3 4Q50 31R10 1Q50 18R3 32R10 1Q50 10 16R3
34R10 1Q50 10 14R3 37R10 12R3 40R10 9R3 43R10 4R3 132R10 7R3 18R10 3R3 20R10
11R3 14R10 5R3 19R10 13R3 12R10 6R3 18R10 15R3 10R10 7R3 18R10 16R3 8R10 8R3
17R10 17R3 1Q50 3 6R10 9R3 16R10 19R3 3Q50 3 1Q50 5R10 9R3
/*************************************************************
#3  /Section 3 : src distribution
/acceptable keywords: srcloc, srcmag, srcspm, srcdis (multi), keffin
/ srcloc: source location (CM number where source located), NumOfEntry=numsrc
/ srcmag: source strength (srcmag*srcdis=src density in that fm, #/cm3-sec), NumOfEntry=numsrc
/ srcspm: source spectrum (srcmag*srcdis*srcspm=src density for a group), NumOfEntry=numsrc*numgrp
/ srcdis: source spatial distribution  srcdis=Src#, FineMeshDist
/ keffin: initial Keff guess, NumOfEntry=1, (more entries reserved for future Keff accelaration
keffin=1.000
/*************************************************************
#4  /Section 4 : xs data
/acceptable keywords: xsname, xstype, numcmt, xstihm,  legord, legoxs,xstchi
/ xsname: cross section file name (charater entry)
/ xstype: 0=(2l-1) is not pre-multiplied; 1=(2l-1) pre-multiplied
/ numcmt: number of comment lines in between material xs block
/ xstihm: xs table total length, (SigmaTot@Column 3 always
/   xstihm=3+numgrp:  SigmaSelfScatter@4 , downscattering only
/   xstihm=3+(2*numgrp-1):  SigmaSelfScatter@(3+numgrp) , upperscattering
/ keffin: initial Keff guess, NumOfEntry=1, (more entries reserved for future Keff accelaration
xsname=hexasy.xs
legord=0  legoxs=0
xstype=0
xstihm=14
/# fission chi 86001.96m
xstchi=9.69128E-01   3.08724E-02   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00
```

```
211 /# fission chi 86002.96m
212  9.69156E-01   3.08442E-02   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00
213 /# fission chi 86003.96m
214  9.69190E-01   3.08100E-02   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00
215 /# fission chi 86004.96m
216  9.69200E-01   3.08001E-02   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00
217 /# fission chi 86005.96m
218  9.69224E-01   3.07764E-02   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00
219 /# fission chi 86006.96m
220  9.69231E-01   3.07689E-02   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00
221 /# fission chi 86007.96m
222  9.69246E-01   3.07537E-02   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00
223 /# fission chi 86008.96m
224  9.69265E-01   3.07354E-02   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00
225 /# fission chi 86009.96m
226  0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00
227 /# fission chi 86010.96m
228  0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00
229 /# fission chi 86011.96m
230  0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00
231 /# fission chi 86012.96m
232  0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00
233 /# fission chi 86013.96m
234  0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00
235 /# fission chi 86014.96m
236  0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00
237 numcmt=1
238 /*************************************************************
239 #5  /Section 5 : boundary cond. and tol.
240 /acceptable keywords: tolinn,tolout,maxinn,maxout,xminus,xpluss,
    yminus,ypluss,zminus,zpluss
```

```
241 / tolinn: inner iteration (within-group) tolerance, negative value↩
       : adjustable for keff loop
242 / tolout: outer iteration (keff loop) tolerance, negative value: ↩
       adjustable for keff loop
243 / maxinn: maxium inner iteration number, negative value: ↩
       adjustable for keff loop
244 / maxout: maxium outer iteration number, negative value: ↩
       adjustable for keff loop
245 / xminus,xpluss,yminus,ypluss,zminus,zpluss : Boundary conditions ↩
       at -x,+x, -y,+y, -z,+z
246 /   =0: vaccum;  =1 albedos for each group: reflective
247 xminus=1   6R1
248 xpluss=1   6R1
249 yminus=1   6R1
250 ypluss=1   6R1
251 zminus=1   6R1
252 zpluss=1   6R1
253 tolinn= 1.00000E-04
254 tolout= 1.00000E-06
255 maxout=1000
256 maxinn=10
257 tolicm= 16R1.0000E-04
```

List of References

[1] G. BELL and S. GLASSTONE. *Nuclear Reactor Theory*. Krieger Publish Company (1970).

[2] E. LEWIS and W. MILLER. *Computational Methods of Neutron Transport*. American Nuclear Society, La Grange Park, Illinois (1993).

[3] H. HANSHAW, A. HAGHIGHAT, and J. WAGNER. "Multigroup cross section generation with adjoint weighting and its application to PV dosimetry." In: *Proceedings of the Radiation Protection and Shielding Conference*. Falmouth, MA (1996).

[4] F. ALPAN and A. HAGHIGHAT. "Advanced methodology for selecting group structures for multigroup cross section generation." In: *Proceedings of PHYSOR 2000–ANS International Topical Meeting on Advances in Reactor Physics and Mathematics and Computation into the Next Millenium*, (pp. 7–12) (2000).

[5] F. ALPAN and A. HAGHIGHAT. "Development of the CPXSD methodology for generation of fine-group libraries for shielding applications." *Nuclear Science and Engineering*, **149(1)**: pp. 51–64 (2005).

[6] F. ALPAN. *An advanced methodology for generating multigroup cross sections for shielding calculations*. PhD dissertation, The Pennsylvania State University (2003).

[7] S. DOUGLASS and F. RAHNEMA. "Consistent generalized energy condensation theory." *Annals of Nuclear Energy*, **40**: pp. 200–214 (2012).

[8] F. RAHNEMA, S. DOUGLASS, and B. FORGET. "Generalized energy condensation theory." *Nuclear Science and Engineering*, **160**: pp. 41–58 (2008).

[9] L. ZHU and B. FORGET. "A discrete generalized energy expansion theory." *Nuclear Science and Engineering*, **166**: pp. 239–253 (2010).

[10] S. DOUGLASS and F. RAHNEMA. "Subgroup decomposition method." *Annals of Nuclear Energy*, **48**: pp. 84–101 (2012).

[11] S. DOUGLASS and F. RAHNEMA. "Cross section recondensation method via generalized energy condensation theory." *Annals of Nuclear Energy*, **38**: pp. 2105–2110 (2011).

[12] N. NOJIRI, M. NAKANO, and N. FUJIMOTO. *Benchmark Problem's Data for HTTR's Start-up Core Physics Experiments. JAERI-memo*, JAERI (2008).

[13] N. ROSKOFF *et al.* "Implementation and verification of the SDM in the TITAN 3-D Sn transport code." In: *PHYSOR* (2014).

[14] C. YI and A. HAGHIGHAT. "A three-dimensional block-oriented hybrid discrete ordinates and characteristics method." *Nuclear science and engineering*, **164(3)**: pp. 221–247 (2010).

[15] C. YI. *TITAN: A 3-D Deterministic Radiation Transport Code User Manual Version 1.29 rev.1.* University of Florida, Nuclear and Radiological Engineering Department (2012).

[16] Z. ZHANG *et al.* "Simplified two and three dimensional HTTR benchmark problems." *Annals of Nuclear Energy*, **38**: pp. 1172–1185 (2011).

[17] M. CHADWICK *et al.* "ENDF/B-VII: Next generation evaluated nuclear data library for nuclear science and technology." *Nuclear data sheets*, **107(12)**: pp. 2931–3060 (2006).

[18] R. MACFARLANE and R. BOICOURT. *NJOY: A neutron and photon cross-section processing system. Technical report*, Los Alamos Scientific Lab., NM (1975).

[19] Y. CE and A. HAGHIGHAT. "A hybrid block-oriented discrete ordinates and characteristics method algorithm for solving linear boltzmann equation." In: *Proc. Int. Conf. M&C*, volume 77 (2007).

[20] B. CARLSON. *Transport Theory: Discrete Ordinates Quadrature over the Unit Sphere. Technical report*, Los Alamos Scientific Lab., N. Mex. (1970).

[21] A. HAGHIGHAT, G. SJODEN, and V. KUCUKBOYACI. "Effectivness of PENTRAN's unique numerics for simulation of the kobayashi benchmarks." *Progress in Nuclear Energy*, **39(2)**: pp. 191–206 (2001).

[22] B. PETROVIC and A. HAGHIGHAT, A. "New directional theta-weighted (DTW) differencing scheme and reduction of estimated pressure vessel fluence uncertainty." In: *Proceedings of the 9th International Symposium on Reactor Dosimetry, Prag*, (pp. 746–753) (1996).

[23] J. DUDERSTADT and L. HAMILTON. *Nuclear Reactor Analysis*. John Wiley & Sons, Inc. (1976).

[24] S. DOUGLASS and F. RAHNEMA. "Specification for a 1-dimensional gas-cooled reactor benchmark problem for neutron transport." In: *Transaction of the American Nuclear Society*, 104. American Nuclear Society (2011).

[25] T. TAIWO *et al.* "Evaluation of high temperature gas-cooled reactor physics experiments as VHTR benchmark problems." *ANL-GenIV-059, Argonne National Laboratory* (2005).

[26] T. SIMENOV. *Helios System Version 1.8.* Studsvik Scandpower Report, SSP-03/221 edition (2003).

[27] C. YI. *PENMSH-XP: A 3-D Mesh Generator for PENTRAN, Version 1.5b.* University of Florida (2008).

[28] I. MURATA *et al.* "Evaluation of local power distribution with fine-mesh core model for high temperature engineering test reactor (HTTR)." *Journal of Nuclear Science and Technology*, **31(1)**: pp. 62–72 (1994).

[29] S. DOUGLASS. *Consistent Energy Treatment for Radiation Transport Methods.* PhD dissertation, Georgia Institute of Technology (2012).

[30] N. ROSKOFF, W. WALTERS, and A. HAGHIGHAT. "Application of the subgroup decomposition method for reactor simulation." In: *International Symposium on Reactor Dosimetry* (2014).