

# University Research in Support of TREAT Modeling and Simulation, FY 2017

Mark DeHart, Editor

Reactor Physics Design and Analysis Department  
Idaho National Laboratory  
PO. Box 1625, Idaho Falls, ID 83415

September 2017



INL is a U.S. Department of Energy National Laboratory operated by Battelle Energy Alliance

#### **DISCLAIMER**

This information was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness, of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.



# **University Research in Support of TREAT Modeling and Simulation, FY 2017**

**Mark DeHart, Editor**

**September 2017**

**Idaho National Laboratory  
Nuclear Systems Design and Analysis  
Idaho Falls, Idaho 83415**

**<http://www.inl.gov>**

**Prepared for the  
U.S. Department of Energy  
Office of Nuclear Energy  
Under DOE Idaho Operations Office  
Contract DE-AC07-05ID14517**

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>University of Florida</b>	<b>2</b>
<b>3</b>	<b>Massachusetts Institute of Technology</b>	<b>13</b>
<b>4</b>	<b>Texas A&amp;M: Lou and Morel</b>	<b>27</b>
<b>5</b>	<b>Texas A&amp;M: Hammer and Morel</b>	<b>51</b>
<b>6</b>	<b>Idaho State University/Berkeley/INL</b>	<b>64</b>
<b>7</b>	<b>University of Arizona (Visiting Professor)/INL</b>	<b>91</b>
<b>8</b>	<b>Texas A&amp;M University (Visiting Professor)/INL</b>	<b>147</b>
	<b>Appendices</b>	<b>153</b>
<b>A</b>	<b>A Parametric Study of the Adjoint Flux Calculation for TDKENO, UF</b>	<b>153</b>
<b>B</b>	<b>Motivation for Using Generalized Geometry in the Time Dependent Transport Code TDKENO, UF</b>	<b>158</b>
<b>C</b>	<b>Verification of the Time-dependent Transport Code TDKENO with the Monte Carlo Code KENO-VI, UF</b>	<b>162</b>
<b>D</b>	<b>Parallel Performance of the Time Dependent Transport Code TDKENO with Applications to TREAT Simulations, UF</b>	<b>175</b>
<b>E</b>	<b>Continuous Temperature Representation in Coupled OpenMC/MOOSE Simulations, MIT</b>	<b>179</b>
<b>F</b>	<b>A Cumulative Migration Method for Computing Rigorous Transport Cross Sections and Diffusion Coefficients for LWR Lattices with Monte Carlo, MIT</b>	<b>193</b>
<b>G</b>	<b>Comparisons of the Finite-Element-with-Discontiguous-Support Method to Continuous-Energy Monte Carlo for Pin-Cell Problems, TAMU</b>	<b>209</b>

<b>H</b>	<b>Improved Quasi-Static Method Implementation for CFEM Diffusion in Rattlesnake, TAMU</b>	<b>224</b>
<b>I</b>	<b>Neutronic Thermal Hydraulic Benchmark using the SKINATH Point Kinetics Model, UAz</b>	<b>236</b>



# 1 Introduction

The Transient Reactor Test Facility (TREAT) is an air-cooled, thermal-spectrum test facility designed to evaluate reactor fuels and structural materials under simulated nuclear excursions and transient power/cooling mismatch situations in a nuclear reactor. The U.S. Department of Energy Office of Nuclear Energy (DOE/NE) is preparing to resume operation of TREAT, which is located at Idaho National Laboratory (INL), by late 2017.

The INL is currently evolving a modeling and simulation (M&S) capability that will enable improved core operation as well as design and analysis of TREAT experiments. This M&S capability primarily uses MAMMOTH, a reactor physics application being developed under the Multi-physics Object Oriented Simulation Environment (MOOSE) framework. MAMMOTH allows the coupling of a number of other MOOSE-based applications.

This work is being supplemented by related research projects at a number of universities. INL is funding TREAT-related research at North Carolina State University, University of New Mexico, Oregon State University, and MIT under internally funded projects within the INL/Battelle National University Consortium Program. Under NEAMS, research is being performed at MIT (different projects), University of Florida, and Texas A&M University, as well as research being performed at INL by a Joint Appointment staff member from Idaho State University. In addition, Dr. Jim Morel of Texas A&M University and Dr. Barry Ganapol of University of Arizona were funded for summer work at the INL.

This report is a compilation of year-end reports provided by the NEAMS University partners and from the visiting professors. For completeness, also included in the Appendix are conference papers that were submitted as part of this research. However, it is worth noting that some of the research performed by university partners more closely tied to the INL mission under this project. That work was included in reports provided as task deliverables over the year. Specifically, the following reports were provided as NEAMS deliverables in FY16 and reflect work in which university faculty and staff were significant contributors.

Zachary M. Prince, Jean C. Ragusa, Yaqi Wang and Mark D. DeHart, “IQS implementation in Rattlesnake, Final Report,” INL/EXT-16-38059, NEAMS Milestone No. M2MS-16IN0401069, June 2016

Jean C. Ragusa and Mark D. DeHart, “Uncollided Flux Techniques for Discrete-Ordinate Radiation Transport Solutions in Rattlesnake, INL/EXT-16-39796, NEAMS Milestone No. M4MS-16IN0401064, August 2016

Weixiong Zheng, Yaqi Wang and Mark D. DeHart, “Multiscale Capability in Rattlesnake using Contiguous Discontinuous Discretization of Self-Adjoint Angular Flux Equation,” INL/EXT-16-39793, NEAMS Milestone No. M3MS-16IN0401066, Sept. 2016

Leslie Kerby, Mark DeHart and Aaron Tumulak, “Integration of OpenMC methods into MAMMOTH and Serpent,” INL/EXT-16-39874, NEAMS Milestone No. M3MS-16IN04010624, Sept. 2016

# NEAMS Final report FY-16, University of Florida

Zander Mausolff, Zach Weems, Dustin Popp,  
Kristin Smith, Forrest Shriver, Sedat Goluoglu

September 2016

## TDKENO Methodology

TDKENO is a computer program that uses a hybrid method for solving the time-dependent, three-dimensional (3-D) Boltzmann transport equation with explicit representation of delayed neutrons [1-3]. TDKENO utilizes the IQS method, which is a flux factorization method [4]. In flux factorization methods, the flux is assumed to be factored into a purely time-dependent amplitude function, which varies quickly with time and a flux shape function, which varies slowly with time. This relationship can be explicitly represented as seen in Equation (1) where  $\psi(r, E, \Omega, t)$  is the angular flux at position  $r$ , energy  $E$  and time  $t$ ;  $T(t)$  is the amplitude function and  $\Phi(r, E, \Omega, t)$  is the flux shape function with weak time dependence.

$$\psi(r, E, \Omega, t) = T(t) \cdot \Phi(r, E, \Omega, t) \quad (1)$$

The position, energy, and angle dependent neutron flux shape is computed using modified versions of the well-known 3-D Monte Carlo codes KENO V.a [5] or KENO-VI [6]. Solving the shape equation, which is a modified form of the 3-D transport equation, is computationally expensive. However, by taking advantage of the parallel computation capabilities of KENO, computation of shape function is sped up. The flux shape calculated by KENO is used to compute the point kinetics parameters (e.g., reactivity, generation time, effective delayed neutron fractions, etc.). Rapidly varying amplitude equation is solved deterministically many times between shape calculations to obtain a highly accurate solution without the expense of direct integration. TDKENO is a hybrid analysis tool due to use of both a stochastic (KENO computing the flux shape) and deterministic (solving the point kinetics equations) method for determining a solution.

## Calculational Flow

The initial step in a given problem is to perform a steady state adjoint calculation using KENO. Note that this can be turned off if the adjoint flux is already known and provided in the working directory. The steady state adjoint flux is used as a weighting function in the IQS method in calculating the point kinetics parameters. Similarly, the forward calculation is done with KENO to determine the initial flux shape, the effective multiplication factor, and constraint integral. Prior to running TDKENO for a transient analysis, the user needs to prepare multigroup cross section libraries that correspond to the initial state of the system as well as all perturbed states. Perturbation is realized by replacing the actual cross section values for a given material in the cross section library without changing the material number. The cross section files are created by running a KENO calculation with the materials corresponding to that state and saving the KENO-generated Monte Carlo-

formatted cross section library. If the cross sections are provided as a text file, one can use the newly developed cross section generation tool to create an AMPX-formatted cross section library for use in TDKENO. TDKENO linearly interpolates between the user provided cross section sets to determine the system cross sections at each reactivity time step while solving for the amplitude function. However, KENO itself only uses the user-provided cross section files to compute the flux shape, i.e., no new cross section file is created as a result of internal interpolation. Therefore user must provide the cross section files in such a manner that using those files with KENO and linear interpolation inside TDKENO would result in correct representation of the system cross sections at all times.

### Development of Detailed TREAT Models

The desire to model complex cores such as TREAT prompted the modification of TDKENO to support generalized geometry inputs. This is advantageous as the input may be created to exact system parameters. TDKENO was initially developed with KENO V.a: the geometrically simpler version of KENO. While quicker than Monte Carlo codes with generalized geometry, KENO V.a places practical limits on input designs as only combinations of cuboids, spheres, cylinders, etc. are possible. To illustrate this problem, two input strategies for modeling the chamfered corners of a TREAT fuel element are shown in Figure 1 for KENO V.a and KENO-VI.

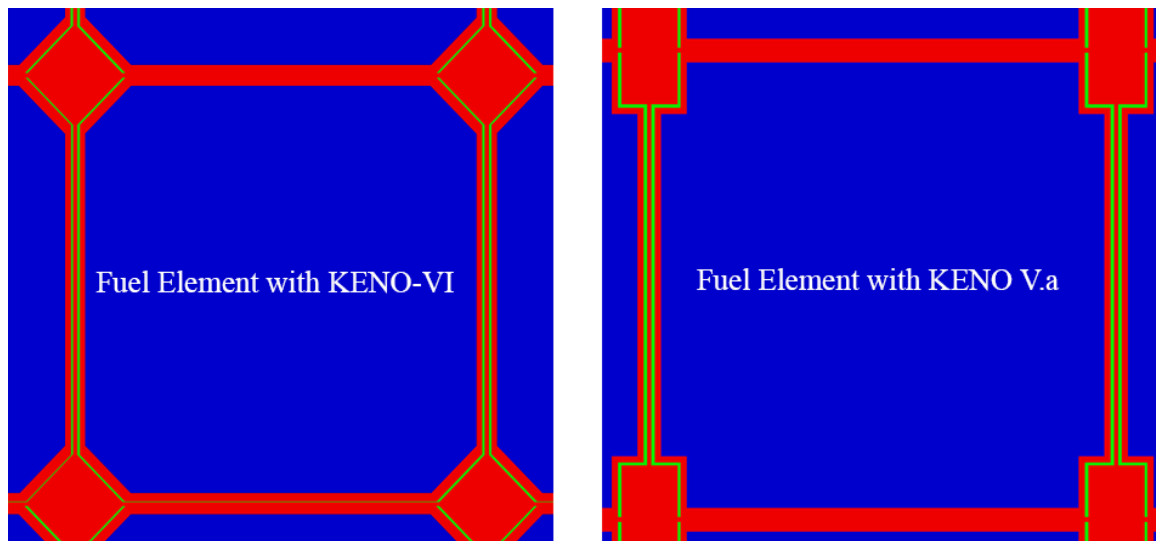


Figure 1. Chamfered corners of fuel elements in KENO V.a and KENO-VI.

Fuel elements have one step and non-fuel elements have a stair-step like structure in KENO V.a. The KENO-VI structure exactly models the chamfered corners for all elements and requires fewer regions. The goal of creating KENO-VI inputs is to better capture the geometry (e.g., corners may be created using planes) with fewer regions.

KENO geometry is comprised of building blocks called units. Units are collection of regions that define a certain geometric body. For example, a LWR pin might be defined as a single unit that has concentric fuel, gap, clad and moderator regions. The same unit may be used in



an array to create a fuel assembly. Different units maybe defined for guide tubes, control rods, etc. When KENO computes fluxes, they are tallied for each unit. If the same unit is used in more than one location as in the case of fuel pins in an assembly, the computed flux is the average of all fluxes in all fuel pins throughout the fuel assembly. This means the computed flux for fuel pins next to control rods will be the same as those that are surrounded by all fuel pins. This maybe acceptable if the control rods don't move. However, for TREAT transients, the transient rods move and therefore it is important to tally the flux for the fuel elements near the transient rods. This is easily accomplished by simply defining each fuel element with a different unit number. Furthermore, the computed fluxes correspond to the entire geometric region. If one is not interested in detailed fluxes, region boundaries may be defined to be the same as the material boundaries. For example, each fuel element meat maybe defined as a single region. In this case, only one spatial flux value for the entire fuel meat is computed. For transient analyses, actual spatial variations for the flux need to be captured especially around the transient rods. To this end, a new input generator has been developed to create detailed TREAT inputs for temperature-limited transients 2855, 2856 and 2857 using KENO-VI geometry. The input generator allows unique unit numbers for each fuel element and also supports either X-axis or Y-axis symmetry in assigning unique unit numbers. In addition, each fuel element and each control, compensation or transient rod can be sub-divided into multiple axial and radial regions to enable detailed spatial flux calculation. Although the input generator currently supports temperature-limited transients 2855, 2856, and 2857 only, it can easily be modified to support other temperature-limited or shaped transient models as well.

The new input generator was used to prepare KENO-VI inputs for pre transient and post transient states of the temperature-limited transients 2855, 2856, and 2857. Previous tests used KENO-VI inputs that were directly converted from KENO V.a, which replaces chamfered corners with stair step edges. New KENO-VI inputs use planes to model the corners, which results in less than 5000 regions for the entire reactor as opposed to previous more than 10000 regions.

### *Analysis of Temperature Limited Transient 2855*

In the temperature-limited transient 2855, transient rods were removed at time zero and travelled a total of 13.6 in. over 0.13 seconds. Reported and computed reactivity insertions for this transient are 1.8%, 1.81% (KENO-VI) and 1.97% (KENO V.a). However, both KENO V.a and KENO-VI calculate about 1.01 as the initial  $k_{\text{eff}}$  of the system that is reported to be just critical. As expected, the KENO-VI calculation agrees better with experiment for reactivity insertion, power and yield. Power vs. time for this transient is shown in Figure 2. Feedback effects for all temperature-limited transients are accounted for by using an empirical formula that defines negative reactivity feedback in terms of total yield for TREAT core:

$$\rho_{fb}(t) = \sum_{i=0}^3 a_i Y^i(t)$$

where  $\rho_{fb}(t)$  is the feedback reactivity,  $Y(t)$  is the total yield in the core at time  $t$  in MJ, and  $a_i$  refers to the empirical coefficients for a third order polynomial.

Of all the transients analyzed, #2855 had the largest discrepancy between simulation and experiment. There are a number of possible reasons for this disagreement. One may be slightly incorrect material definitions resulting in a pre-transient  $k_{eff}$  greater than 1. Another may be the reliance on a simple feedback model as opposed to coupling thermal feedback. Quantifying how far the calculations deviate from experiment is difficult due to apparent inconsistencies in the M8CAL document [7]. For instance, the reported total yield value of  $792 \pm 10\%$  MJ disagrees with the yield of 726 MJ of the experimental plot. Despite the inconsistencies in the M8CAL data, the agreement is sufficient and TDKENOs ability to accept generalized geometry inputs results in better simulation of transients.

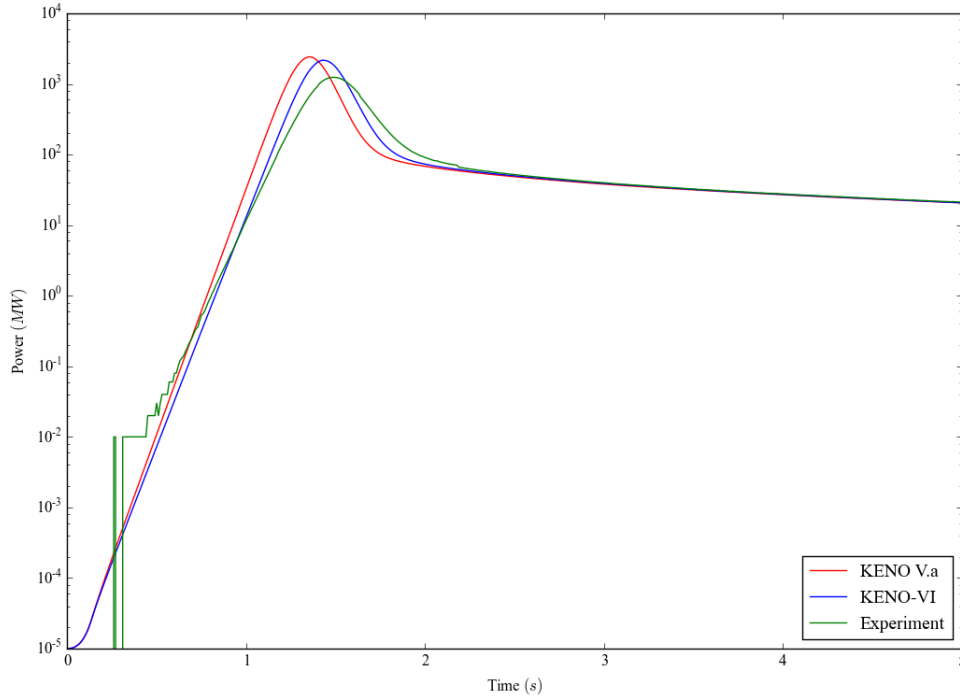


Figure 2. Power vs. Time for Temperature-limited Transient 2855

#### *Analysis of Temperature Limited Transient 2856*

The temperature-limited transient 2856 experiment was carried out by withdrawing the transient rods a distance of 18.60 in. over 0.16 seconds. Reported and computed reactivity insertions for this transient are 3.01%, 3.13% (KENO-VI) and 3.04% (KENO V.a). The experiment was run for 60 seconds and a total of  $1572 \pm 10\%$  MJ was deposited in the TREAT core. Power vs. time for this transient is shown in Figure 3.

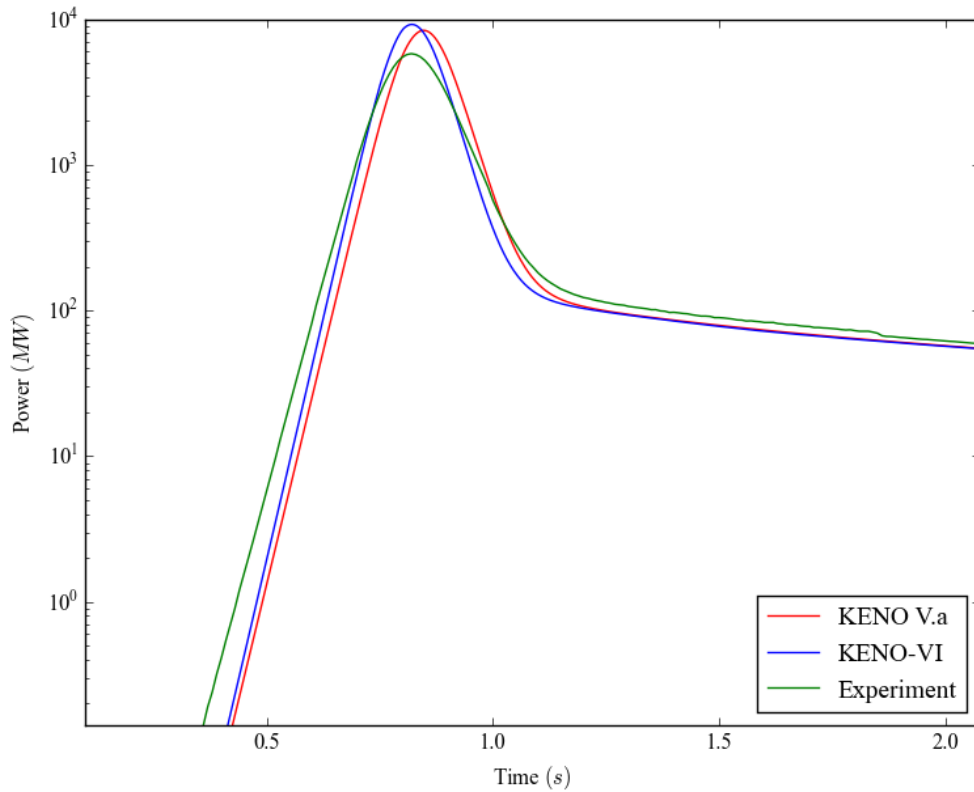


Figure 3. Power vs. Time for Temperature-limited Transient 2856

#### *Analysis of Temperature Limited Transient 2857*

The temperature-limited transient 2856 experiment was carried out by withdrawing the transient rods a distance of 21.5 in. over 0.18 seconds. Reported and computed reactivity insertions for this transient are 3.84%, 3.87% (KENO-VI) and 3.86% (KENO V.a). The experiment was run for 60 seconds and a total of  $2265 \pm 10\%$  MJ was deposited in the TREAT core. Power vs. time for this transient is shown in Figure 4.

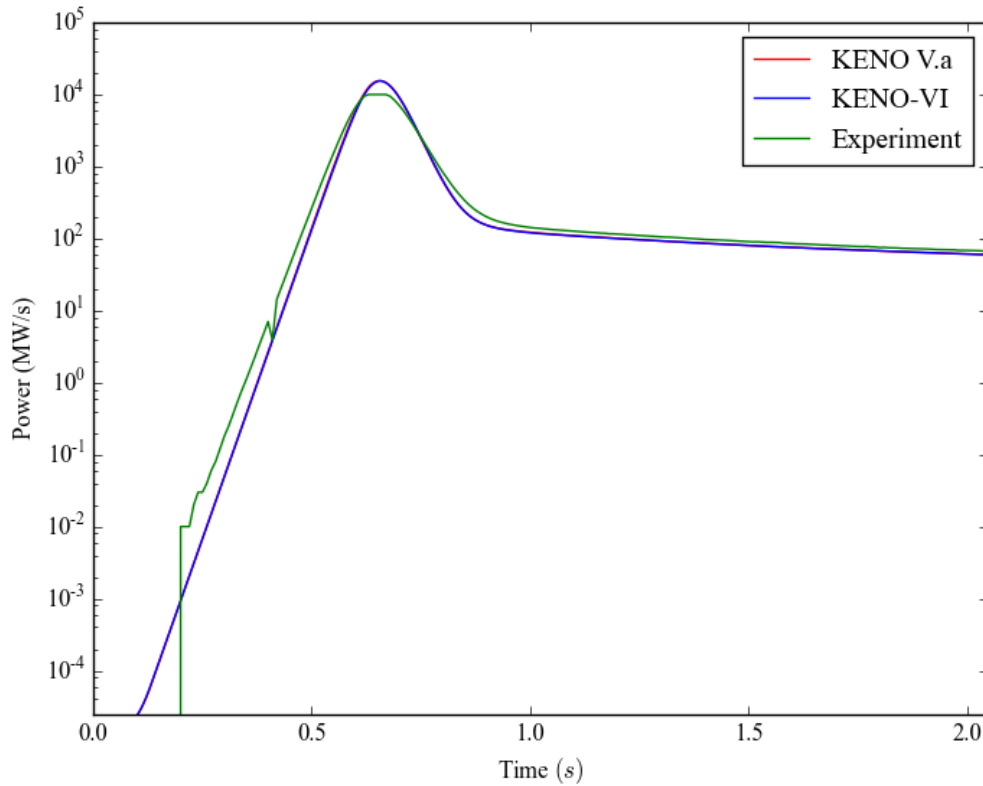


Figure 4. Power vs. Time for Temperature-limited Transient 2857

### Experimental Reported Values for Temperature-limited Transients

M8CAL documentation [7] was reviewed to determine the reason for discrepancies between the cited power values and the values from the detection devices that were used for the temperature-limited transients. It was determined that the cited values are based on RTS energy A data. These data have been digitized for comparison against TDKENO calculated values in the future.

### Effect of Weighting Function on Calculated Power Profiles

IQS methodology uses adjoint flux as the weighting function to equate mathematically derived point kinetics parameters to physical quantities such as system reactivity, generation time and effective delayed neutron fractions. Although the dependence of these parameters on the weighting function is loose, it is worthwhile to determine the effect of accuracy of the adjoint flux solution on the computed power history. To test this effect, temperature-limited transient 2855 has been modeled by varying the number of particles per generation and the total number of generations for adjoint flux calculations. It was determined a minimum of 20000 particles per generation and 5000 active generations are needed to calculate accurate adjoint flux values. Peak power vs. time is shown in Figure 5 for varying number of generations with 20000 particles per generation.

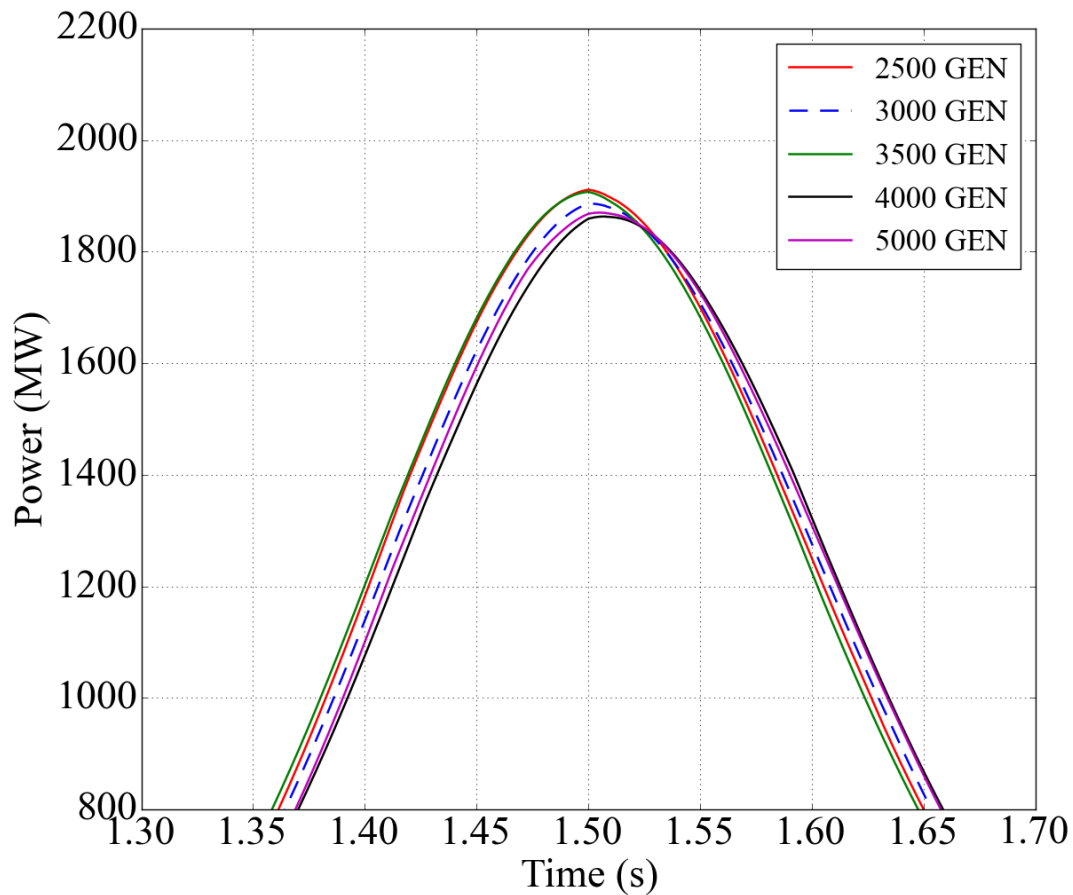


Figure 5. Peak Power vs. Time for Temperature-limited Transient 2855 using Different Number of Adjoint Histories

### Improvements to Reference IQS Solver

Detailed TREAT input with many radial and axial regions contains hundreds of units and tens of thousands of regions for which fluxes are tallied. In order to reduce the statistical uncertainty associated with the flux calculations, tens or hundreds of millions of particle histories must be simulated. To improve performance of TDKENO, latest release of SCALE, which is version 6.2, was acquired and built on both Linux and Mac OS X systems. KENO codes in SCALE6.2 have parallel execution capability, which is needed for detailed full-core transient analyses of TREAT. The KENO codes from SCALE6.2 have been modified and included in TDKENO to perform flux shape calculations using MPI parallelism. Table I below shows the performance of TDKENO for temperature-limited transient 2855 using different number of cores. As seen from these results, optimum utilization is achieved with 16 cores. The modeled core contains about 4000 regions with 21 materials. The calculations were performed with 238 group problem-dependent cross section library based on ENDF/B-VII.0 evaluations. A total of 14 flux shape calculations,

each with 25M histories, were performed with KENO-VI. Total 25M histories were selected because it yielded nearly identical power profile as 300M histories as shown in Figure 6. Note that in these calculations, only KENO-VI is executed in parallel. Parallelization of TDKENO driver itself to compute the delayed neutron concentrations over the phase space is ongoing. Parallelization on GPUs with a directive based approach based on OpenACC is also being investigated. Even naive implementations have shown some speed up.

Table I. Elapsed Time vs. Number of Cores

Total Cores	Elapsed Time (hours)
1	205.8
16	26
32	27
48	28.3
64	17.5
80	16.2
96	15.3
160	13.5

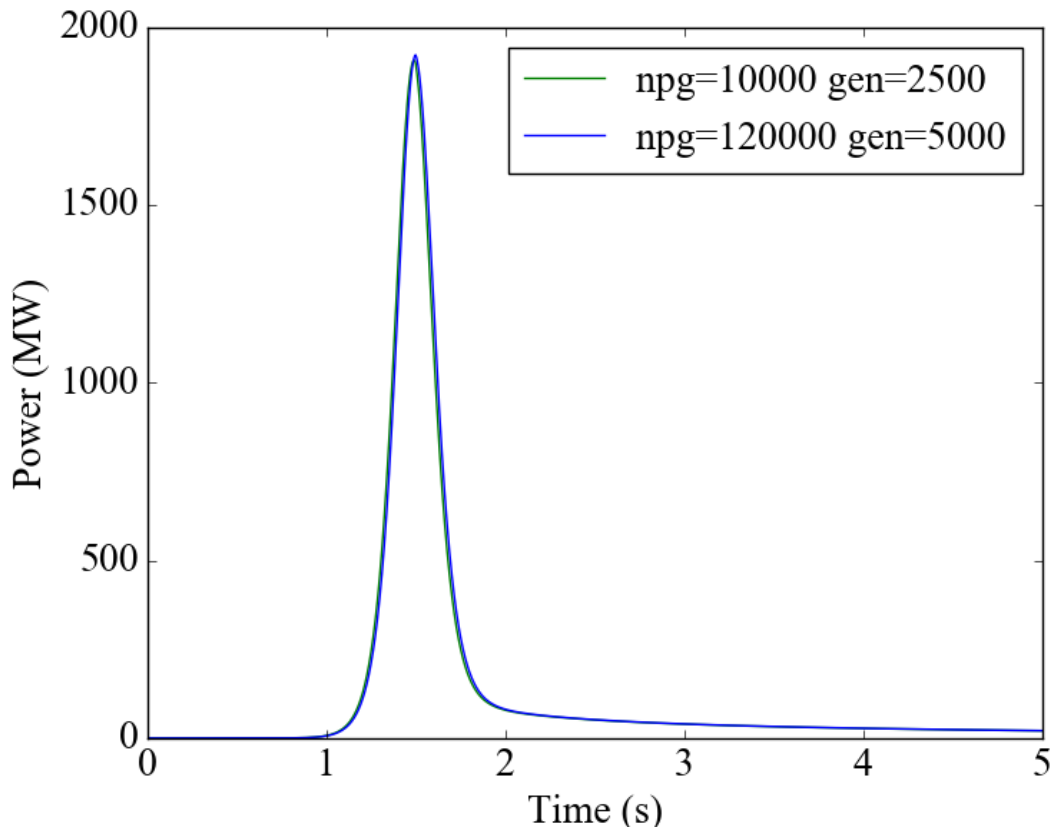


Figure 6. Power vs. Time for 2855 using 25M and 300M histories

### Time Dependent Monte Carlo Solver Coupling to MOOSE

Current version of TDKENO supports some simple feedback mechanisms. In order to couple with a state of the art code such as BISON to compute temperature distribution throughout the TREAT core, coupling of TDKENO to MOOSE has been investigated. When applied to TREAT, KENO outputs are parsed and a 2D representation of the model is generated on a grid. Each grid point represents a fuel element in the TREAT reactor. Associated values are placed on this grid such as power density, thermal conductivity, etc. This information is mapped to a corresponding MOOSE mesh. On the MOOSE side the steady state heat conduction equation is solved and provides a temperature on each nodal point. The complete coupling has not yet been completed as testing on the MOOSE side is ongoing and details of the time scale at which the codes will be coupled is still being worked out. Instead of mapping to a Cartesian mesh on the MOOSE side, mapping the values produced from TDKENO on to a working mesh of TREAT is also being investigated. The mesh for the M8CAL experiments has been taken from the MAMMOTH analyses. It is anticipated that mapping values to this map will be relatively straightforward. Additionally, initial calculations of the temperature using the steady state heat conduction equation over-estimated the temperature compared with historical TREAT data. The historical TREAT data gives the temperature of the core as a function of the energy deposited. The over-estimation is likely in part due to the steady state equation not containing the heat capacity term in it.

### **Cross Section Library Generator**

A cross section conversion utility has been developed to convert cross-section libraries from text or XML format to AMPX working library format. This utility will ensure same cross section sets are used between different codes for modeling TREAT. Initial tests showed that the working cross section library generated with the utility code produces correct library. This utility has been used for creating cross section libraries for TDKENO for the C5G7-TD benchmark study as well.

### **C5G7-TD Computational Benchmarks**

C5G7-TD benchmark models have been developed for 2-D and 3-D cases. New cross section generator was successfully used to convert the provided cross section data into AMPX-formatted cross section libraries for use in TDKENO. In addition, TDKENO was modified to use more than one set of delayed neutron fractions and decay constants as well as neutron velocities. Work is ongoing to model and analyze all C5G7 (2-D and 3-D) transients.

### **REFERENCES**

- [1] S. GOLUOGLU et al., "TDKENO: A Hybrid Time-dependent Transport Analysis Tool", *Proceedings of the international conference on Nuclear Technology*, American Nuclear Society, Anaheim, Ca, Volume 111, pp 1401-1404 (2014).
- [2] K. L. GOLUOGLU, "Revitalization and Status of the Hybrid Stochastic/Deterministic Computer Code for Transient, Three-Dimensional Neutron

Transport, TDKENO,” A practice project for Master of Science degree, The University of Tennessee, Knoxville (2004).

[3] C. L. BENTLEY, “Improvements in a Hybrid Stochastic/Deterministic Method for Transient, Three-Dimensional Neutron Transport,” Ph.D. Dissertation, Nuclear Engineering Department, The University of Tennessee, Knoxville (1996).

[4] S. GOLUOGLU and H. L. DODDS, “Three-dimensional Neutron Transport Methodology,” *Nuclear Science and Engineering*, Volume(139): pp. 248-261 (2001).

[5] R. D. BUSCH, “KENO V.a Primer: A Primer for Criticality Calculations with SCALE/KENO Va Using CSPAN for Input,” ORNL/TM-2002/155. Oak Ridge National Laboratory, Oak Ridge TN (2003).

[6] S. M. BOWMAN, “KENO-VI Primer: A Primer for Criticality Calculations with SCALE/KENO-VI Using GeeWiz,” Oak Ridge National Laboratory, Oak Ridge TN. ORNL/TM-2008/069 (2008).

[7] W. R. ROBINSON and T.H. BAUER, “The M8CAL Power Calibration Experiment,” ANL-IFR- 212, URL <http://www.osti.gov/scitech/servlets/purl/1130413> (1994).



# NEAMS Final report FY2016

Matt Ellis, Samuel Shaner, Will Boyd, Zhaoyuan Liu, Benoit Forget, Kord Smith

September 2016

## 1 Application of Cumulative Migration Method

### 1.1 Directional Cumulative Migration Area and Diffusion Coefficients

In conventional nodal diffusion calculations, material properties are generally assumed isotropic and diffusion coefficients have no variation in different directions. However, the isotropy in practical problems is never true and sometimes can be a very poor assumption, such as in the assembly model of TREAT. TREAT assemblies have air streaming channels in the vertical direction, which makes directional variation a considerable issue for diffusion coefficients. Thus based on the Cumulative Migration Method (CMM) [?] for computing homogenized diffusion coefficients, directional diffusion coefficients can also be computed to improve the precision of diffusion calculations.

Through CMM, directional dependency can be handled naturally by the directional components of the crow flight length. In a Cartesian coordinate system, every track vector can be decomposed into different components by projecting to the according axes.  $\vec{r}$  is the neutron's crow flight vector, and  $\vec{r}_z$  is the vertical component of  $\vec{r}$ . Since  $\vec{r} = \vec{r}_x + \vec{r}_y + \vec{r}_z$  and  $(\vec{r})^2 = (\vec{r}_x)^2 + (\vec{r}_y)^2 + (\vec{r}_z)^2$ . In isotropic materials, it is obvious that  $(\vec{r})^2 = 3 \cdot (\vec{r}_z)^2$ . The directional diffusion coefficients can be computed by using  $\vec{r}_z$  as shown in Equation (??).

$$(M_{z,g}^c)^2 = \frac{1}{6} \overline{(r_{z,g})^2} \quad (1a)$$

$$D_{z,g}^c = 3 \cdot (M_{z,g}^c)^2 \cdot \Sigma_{r,g}^c \quad (1b)$$

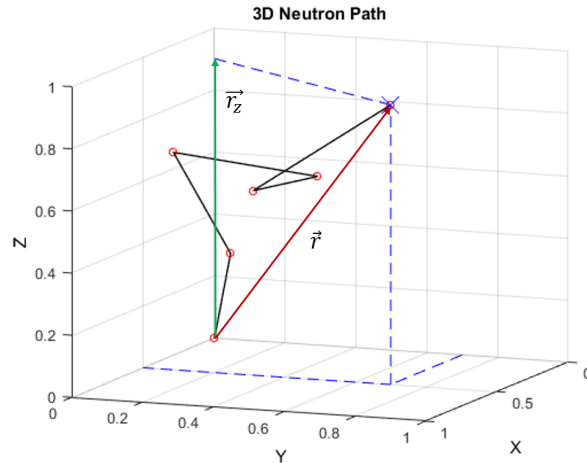


Figure 1: Illustration of directional projection of crow flight vector.

The multiplier factor 3 in Equation (??) ensures that the directional diffusion coefficients  $D_{z,g}^c$  will be the same as the averaged diffusion coefficients  $D_g^c$  in isotropic materials.

To test the effect of directional diffusion coefficients computed by CMM, a 3D Monte Carlo simulation is carried out using OpenMC for the simplified TREAT fuel assembly. Reflective boundary conditions are used in the horizontal plane ( $x$  and  $y$  directions), and 60 cm top and bottom graphite reflectors are used axially ( $z$  direction). Vacuum boundary conditions are employed outside of the top and bottom reflectors.

In addition, the eigenvalue as well as the flux distribution for the same problem is computed using a 1D 11-group diffusion calculation. The 11-group diffusion coefficients and cross sections are tallied and computed using OpenMC for the fuel and reflector separately, each with the actual configuration in 2D horizontal plane and infinite in axial direction, as traditionally done for PWR fuel assemblies.

## 1.2 Four Approaches of Generating Diffusion Coefficients

Using the 3D Monte Carlo result as a reference, it is compared with the 1D 11-group diffusion result with diffusion coefficients generated by four different approaches. The four approaches include:

- $D_g^{os}$  : Out-scatter approximation for computing  $\Sigma_{tr,g}^{os}$ , then  $D_g^{os} = 1/3\Sigma_{tr,g}^{os}$ .
- $D_g^{as}$  : Asymptotic Out-scatter approximation for computing  $\Sigma_{tr,g}^{as}$ , then  $D_g^{as} = 1/3\Sigma_{tr,g}^{as}$ .
- $D_g^{CMM}$  : CMM method for tallying cumulative migration area  $(M_g^c)^2$  using OpenMC, then computing  $D_g^c$  and  $D_g$ .
- $D_{z,g}^{CMM}$  : CMM method for tallying directional cumulative migration area  $(M_{z,g}^c)^2$  using OpenMC, then computing  $D_{z,g}^c$  and  $D_{z,g}$ .

## 1.3 Simulation Results

The results of the eigenvalue for this problem are listed and compared in Table ?? . Compared to the 3D OpenMC result,  $k_{eff}$  from the diffusion calculation with diffusion coefficients from CMM perform much better than those from out-scatter approximations. The accuracy of flux distribution is also improved with CMM, as shown in Figure ?? , by reducing the max relative error of around 6% with out-scatter approximations to about 2% with CMM. The error of group-wise flux distribution is improved in both the fuel region and the reflector region.

It should also be mentioned that it will introduce additional hundreds of pcm difference if the multi-group cross sections in the diffusion problem are replaced by cross sections tallied in the actual 3D geometry instead of 2D infinite length geometry.

Table 1: Comparison of eigenvalues computed by different approaches for the TREAT fuel assembly problem.

Method	$k_{eff}$	Difference
OpenMC	1.44399 ( $\pm 0.00006$ )	(reference)
Diffusion by $D_g^{os}$	1.45750	0.01351
Diffusion by $D_g^{as}$	1.45680	0.01281
Diffusion by $D_g^{CMM}$	1.44738	0.00339
Diffusion by $D_{z,g}^{CMM}$	1.44413	0.00014

## 1.4 Comparison of Diffusion Coefficients from Different Methods

The key of the improvement in diffusion results with CMM lies in the diffusion coefficients of fast groups. The comparison of the 11-group diffusion coefficients computed by different methods is in Table ?? with the energy group structure. It is obvious that the diffusion coefficients of group 1 and group 2 generated by CMM are bigger than those of out-scatter approximations, which impact axial leakage significantly. CMM can improve the accuracy of the diffusion coefficients by better representing the anisotropic property.

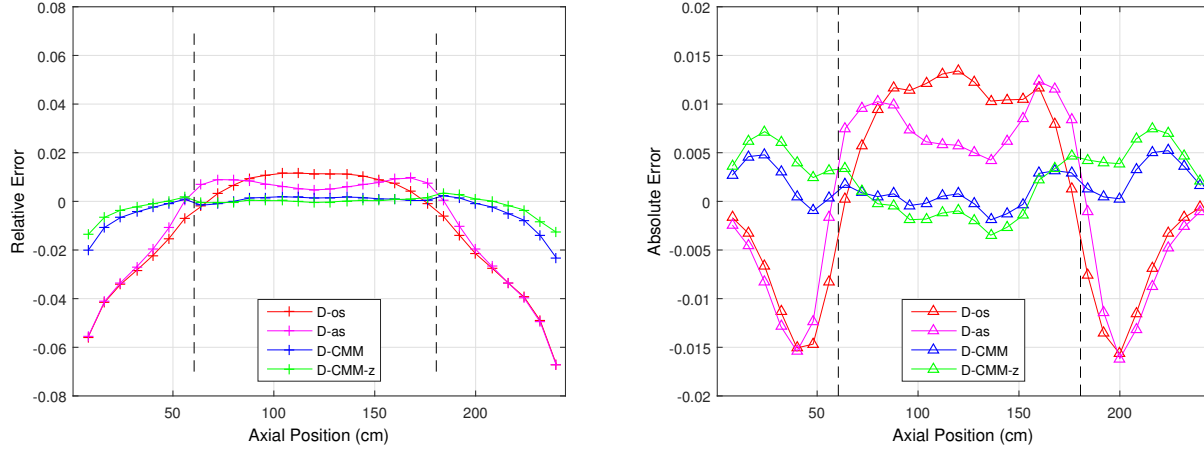


Figure 2: Error distribution of flux computed by 1D diffusion solver for TREAT fuel assembly problem. (Left: Relative error distribution of integrated flux; right: Absolute error distribution of flux in group 5(8.1eV-132.7eV).)

Table 2: Comparison of 11-group diffusion coefficients computed by different methods. (The units of diffusion coefficients in this table are all *cm*.)

Group & Energy (MeV)	$D_g^{os}$	$D_g^{as}$	$D_g^{CMM}$	$D_{z,g}^{CMM}$
1 (3.329E00-2.000E+1)	3.055	2.678	4.068	4.081
2 (1.156E-1-3.329E00)	1.531	1.498	1.748	1.773
3 (3.481E-3-1.156E-1)	0.969	1.008	1.010	1.039
4 (1.327E-4-3.481E-3)	0.944	0.993	0.991	1.020
5 (8.100E-6-1.327E-4)	0.942	0.990	0.987	1.014
6 (6.250E-7-8.100E-6)	0.942	0.989	0.985	1.011
7 (2.096E-7-6.250E-7)	0.927	0.983	0.970	0.997
8 (7.650E-8-2.096E-7)	0.906	0.973	0.951	0.980
9 (4.730E-8-7.650E-8)	0.877	0.956	0.923	0.951
10 (2.001E-8-4.730E-8)	0.836	0.928	0.879	0.908
11 (0.000E00-2.001E-8)	0.697	0.814	0.749	0.777

## 2 Advanced Spatial Homogenization Techniques

### 2.1 Background and Motivation

One of the primary challenges for deterministic core neutronics methods is accurate multi-group cross section (MGXS) generation. Historically, many spectral and geometric approximations have been developed to model the scalar flux to compute multi-group cross sections with tools such as LANL’s NJOY [?] for specific spectral geometries. However, no systematic methodology of approximations to the flux has been developed which is broadly applicable to all reactor types. Monte Carlo (MC) presents a more accurate pathway to generate multi-group cross sections for high-fidelity deterministic calculations since it does not require the use of any approximations to the flux.

In the last decade, MC has been increasingly used to compute few-group constants for coarse mesh diffusion calculations with the advent of the Serpent MC code [?]. Some recent work at MIT to support NEAMS has investigated the use of MC methods to generate MGXS for high-fidelity full core deterministic transport methods. In particular, this work has developed novel spatial homogenization techniques which use **machine learning** to **improve the accuracy** and **accelerate the convergence** of MGXS generated with Monte Carlo simulations.

### 2.2 Methodology

This project’s requirements for “big data” Monte Carlo calculations can be defined along two primary dimensions: **scalable parallel algorithms** for efficient MC tallying, along with **flexible and robust tools** for downstream data processing. In particular, this work has developed of a “simulation triad” encompassing three primary simulation codes. First, the OpenMC Monte Carlo code [?] is utilized to generate multi-group cross sections with a newly developed Python Application Programming Interface for big data analytics [?]. Second, the MGXS are used by the OpenMOC method of characteristics (MOC) code [?] for deterministic multi-group transport calculations. Finally, the OpenCG library [?] enables the processing and transfer of tally data on combinatorial geometry (CG) meshes between OpenMC and OpenMOC. In addition, a significant amount of infrastructural code has been developed to process the results produced by OpenMC and OpenMOC with Python-based tools for scalable machine learning, such as the `scikit-learn` package [?].

Four primary spatial homogenization techniques have been developed to account for inter-pin and intra-pin spatial self-shielding effects, including the following:

- **Null Homogenization** tallies MGXS for each unique fuel pin type using the MC flux from the complete heterogeneous geometry (Fig. ??). This approach is most similar to traditional methods for MGXS generation, but is the least accurate for full core transport simulations.
- **LNS Homogenization** uses OpenCG’s LNS algorithm to identify patterns in a combinatorial geometry to predict spatial self-shielding effects (Fig. ??). This approach is akin to the “geometric templates” used in many common lattice physics codes such as CASMO [?], but is not scalable for full core simulations.
- **Agglomerative Homogenization** uses unsupervised agglomerative clustering to identify trends due to spatial self-shielding effects (Fig. ??). This approach uses machine learning to make tradeoffs between accuracy and speed, and can flexibly adapt to model arbitrary core models without human intervention.
- **Degenerate Homogenization** assigns a unique set of MGXS to each and every fuel pin (Fig. ??). This is the “brute” force approach that best captures spatial self-shielding effects, but is computationally expensive.

These novel spatial homogenization techniques have been applied to various PWR benchmarks due to the overwhelming knowledge base of these reactors, though the methods are equally applicable to other reactor types. Some preliminary results presented here are for a simple PWR model derived from the Benchmark for Evaluation And Validation of Reactor Simulations (BEAVRS) [?]. This model includes a  $2 \times 2$  colorset of fuel assemblies with 1.6% and 3.1% enriched  $\text{UO}_2$  fuel pins, water-filled control rod guide tubes (CRGTs), and burnable poisons (BPs) adjacent to a water reflector. OpenMC was used to generate MGXS in each fuel pin, and each of the four homogenization techniques was applied to construct material/MGXS configurations for OpenMOC as shown in Fig. ??.

Agglomerative homogenization is the most original approach developed as part of this project. Agglomerative homogenization flexibly accounts for spatial self-shielding effects from neighboring CRGTs, BPs and other heterogeneities in reactor models. This scheme analyzes tally data to identify “clusters” in MGXS data induced

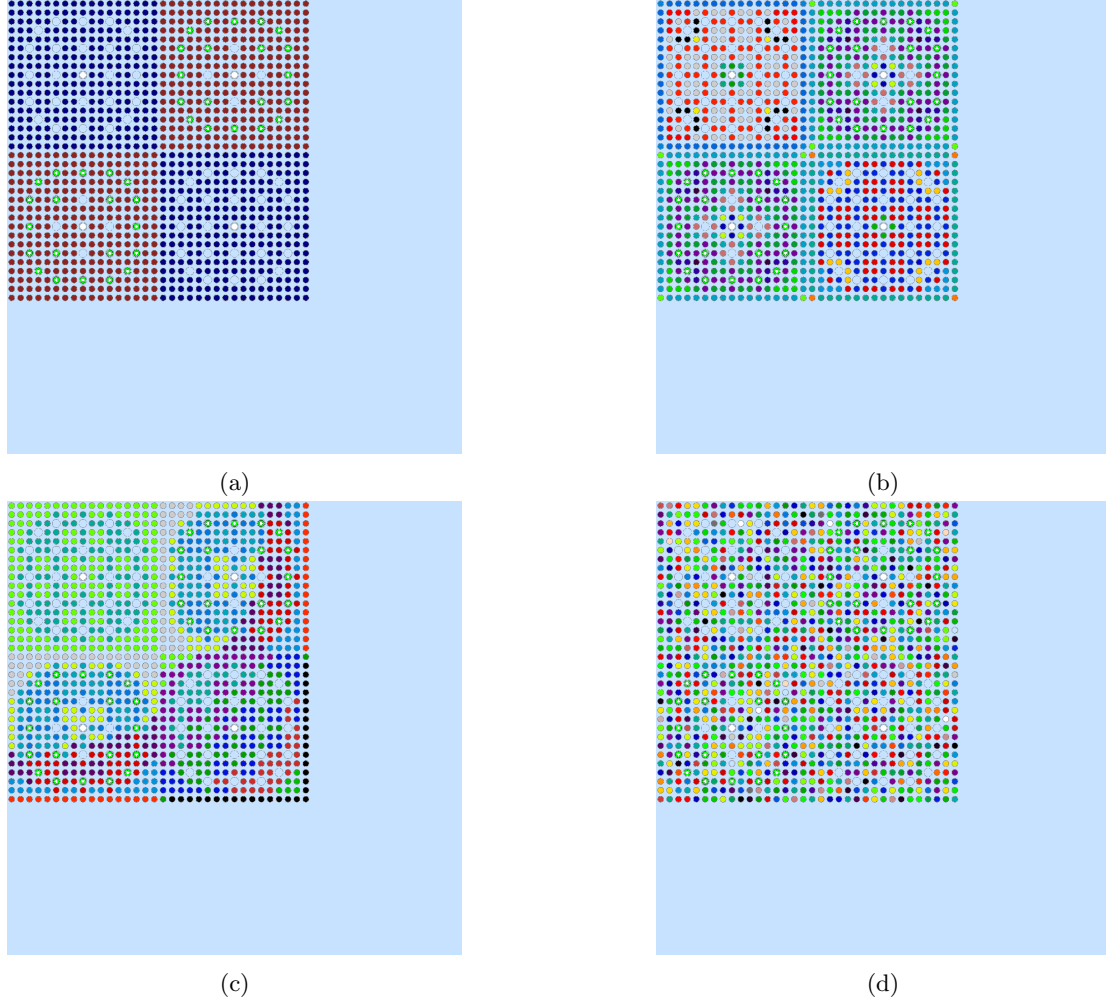


Figure 3: The multi-group material configurations for 2D OpenMOC simulations of a  $2 \times 2$  PWR assembly benchmark using Null (a), LNS (b), Agglomerative (c) and Degenerate (d) spatial homogenization techniques.

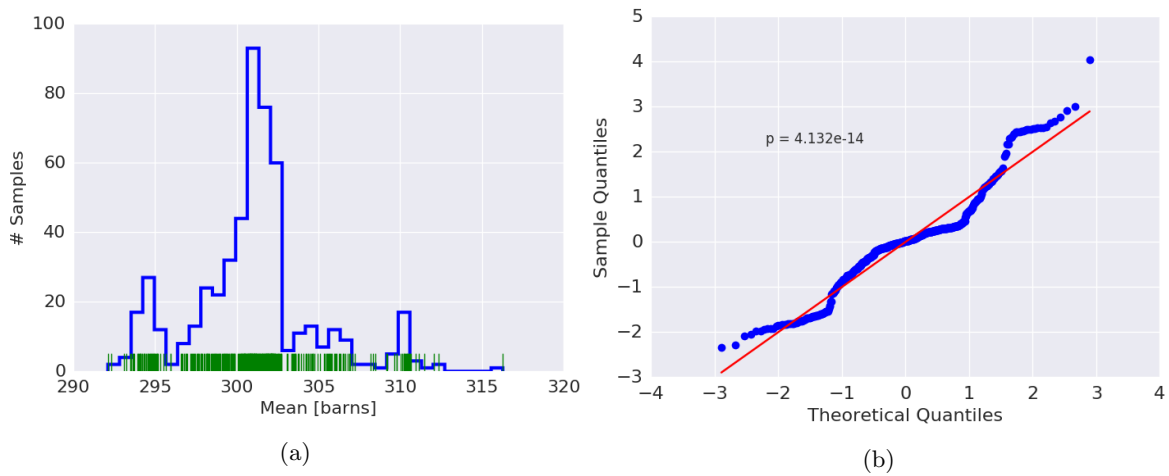


Figure 4: The pin-wise U-235 fission MGXS. A histogram of the MGXS in each fuel pin (a) highlights clustering due to spatial self-shielding effects from CRGTs, BPs and the water reflector. The same data is illustrated in a quantile-quantile plot (b) to demonstrate the deviation from normality and clustering of MGXS data.

by similar spatial self-shielding effects experienced by each pin in a reactor model. The clustering of U-235 thermal fission MGXS in the 1.6% enriched fuel pins in the 2×2 colorset benchmark is depicted in Fig. ???. By averaging the MGXS in each cluster, the convergence rate of Monte Carlo calculations is accelerated for MGXS generation.

## 2.3 Results

The efficacy of each of the four spatial homogenization techniques is evaluated by comparing the results from deterministic multi-group OpenMOC calculations with reference results computed from OpenMC. The spatial homogenization techniques are designed to improve the predictive capabilities for spatial distributions of reaction rates, and in particular, reactions which are especially sensitive to spatial self-shielding. For example, the methods developed here can substantially improve deterministic simulations’ predictive accuracy of the spatial distribution of U-238 capture rates, and hence the production of Pu-239 throughout a reactor core which will have important downstream effects.

The percent relative error for U-238 capture rates are shown in Fig. ???. The null approach is unable to predict U-238 capture for pins at assembly-assembly or assembly-reflector interfaces since it does not distinguish between pins with different neighbors. The LNS approach performs much better, but still poorly predicts capture rates for pins at assembly-reflector interfaces. This approach could be extended to further capture these effects, but would lose some of the automation of the process. The agglomerative and degenerate schemes produce nearly equivalent error distributions that are much more evenly distributed across all pins in the benchmark. These results demonstrate the potential for agglomerative homogenization to achieve the same accuracy as “brute force” degenerate homogenization, but with far fewer MC particle histories needed to generate MGXS.

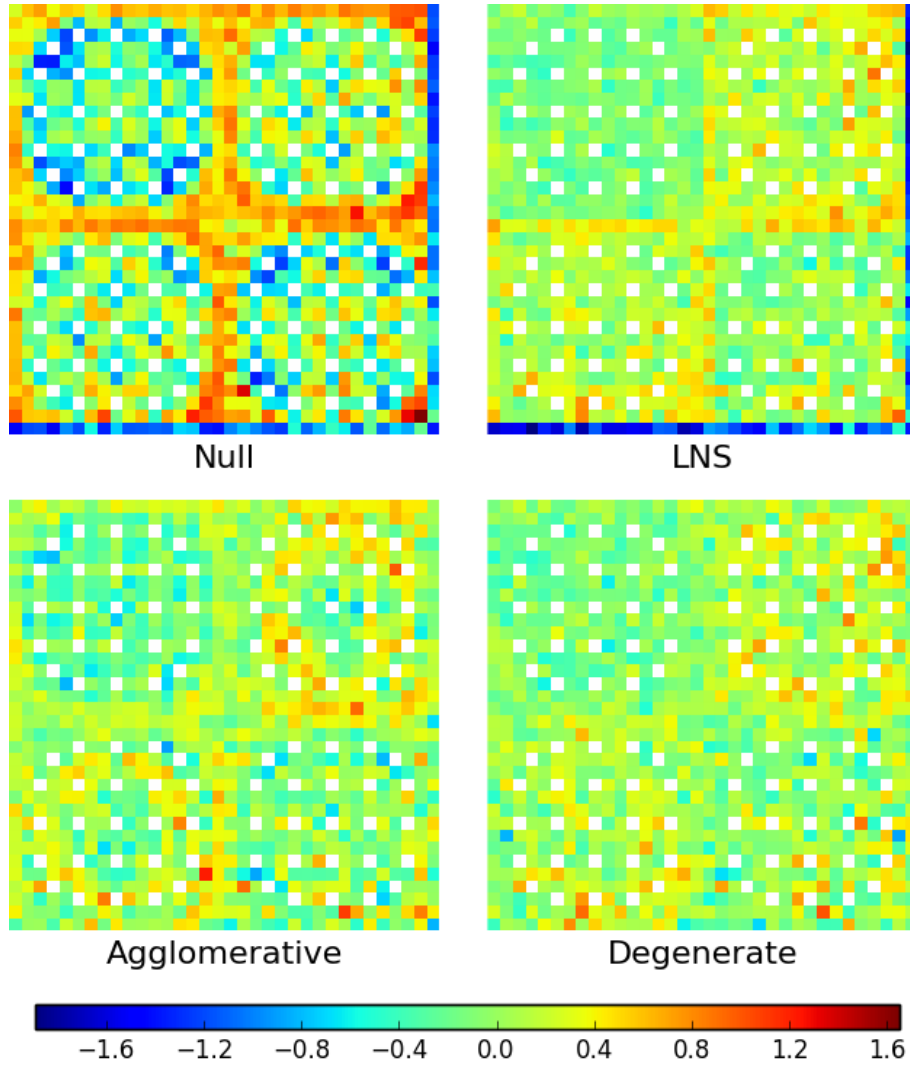


Figure 5: U-238 capture percent relative rate errors for various spatial homogenization techniques.

### 3 Methods for Multiphysics Feedback In Monte Carlo Simulations

One of the primary challenges yet to be fully resolved for Monte Carlo reactor simulations is the efficient and accurate inclusion of multiphysics feedback. To this end, work has been conducted with the support of NEAMS to develop a coupling framework between the open source Monte Carlo code OpenMC and the Multiphysics Object-Oriented Simulation Environment (MOOSE). This coupling framework has been used to investigate the accuracy and efficiency of methods for including multiphysics feedback in the simulation of TREAT and LWRs. In the following sections the coupling framework will be described as well as a novel spatially continuous depletion methodology developed to fully complete the coupling framework.

#### 3.1 Framework for Multiphysics Feedback

The coupling framework between OpenMC and MOOSE must be able to perform two basic tasks. The first task is that tallied reactor physics data in OpenMC must be accurately and efficiently reconstructed on unstructured finite element meshes. The second task, is that multiphysics data on unstructured finite element meshes must be transferred and used on the constructive solid geometry (CSG) model in OpenMC. The primary challenge to accomplishing these tasks is that the OpenMC geometry and MOOSE geometry are discretized in very different manners.

To enable the transfer and reconstruction of tallied data from OpenMC on the MOOSE meshes, Functional Expansion Tallies (FETs) [?] have been implemented in OpenMC [?]. These FETs differ from traditional Monte Carlo tallies by tallying the individual moments of a polynomial expansion over a region. With these higher order moments, detailed distributions for tallied quantities can be reconstructed over a region without the need for discretization in the Monte Carlo simulation. Moreover, a detailed distribution can be passed from OpenMC to MOOSE applications using a small number of expansion coefficients with minimal spatial mapping.

As an example of the FET reconstruction in MOOSE applications, Figure ?? shows the fission power distribution reconstruction on 360 MOOSE multiapps which together make up the TREAT core. The power distribution shown in Figure ?? shows the general power distribution that one expects for the TREAT reactor without control rods inserted into the core. That is, the power distribution is highest in the center of the core and decreases radially and axially outwards from the center. The power distribution shown in Figure ?? was reconstructed from a 2D first order Legendre expansion (four expansion coefficients) in the X and Y directions and a 1D tenth order Legendre expansion (11 coefficients) in the Z direction applied to each assembly.

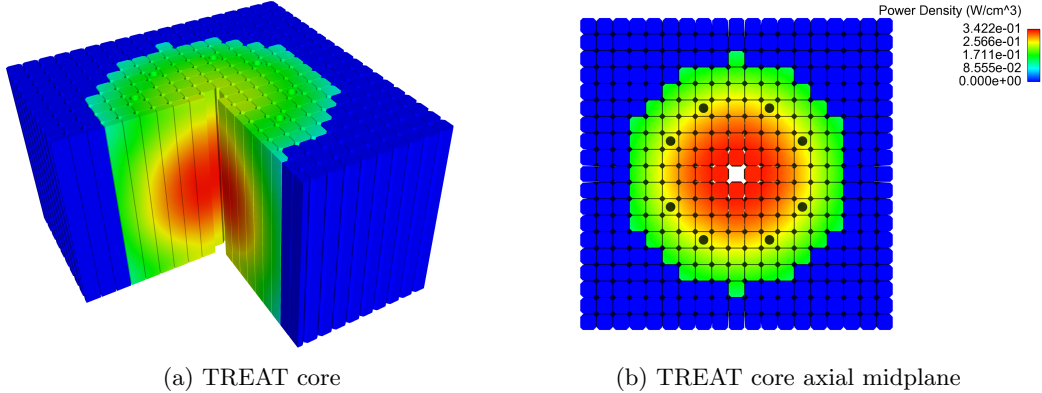


Figure 6: TREAT fission power distribution reconstructed from OpenMC FETs of first order in the X-Y planes and 10th order in the Z plane.

The power distribution reconstruction shown in Figure ?? is worth discussing because embedded in it is one of the unique challenges of the TREAT reactor for multiphysics coupling. More specifically, FETs on the TREAT reactor geometry are challenging because in the X-Y plane the TREAT assembly has chamfered corners. These chamfered corners cause the Legendre expansion domain and the TREAT assembly geometry to not have a simple one-to-one mapping. As a result, expansions above first order will incorrectly excite moments. It is not expected, however, that higher than first order expansions will be needed in the X-Y plane for thermal feedback purposes because graphite has a relatively large thermal conductivity which makes higher order variations in power distribution within assemblies insignificant.



The same solution field expansion methodology that is used to transfer data from OpenMC to MOOSE applications is used to transfer data in reverse. That is, the solution fields in MOOSE are integrated into a single polynomial expansion across the same geometry over which the FET was applied. In the TREAT reactor this means that temperature distributions are passed back to OpenMC for each assembly as a first order Legendre expansion in the X-Y plane and a tenth order Legendre expansion in the Z direction.

To utilize the multiphysics information in OpenMC, a continuous material tracking algorithm first proposed by Brown and Martin has been implemented [?]. This tracking methodology allows neutrons to be transported through a material with a continuously varying total macroscopic cross section. With this method, continuous temperature distributions and density distributions can be used in OpenMC to incorporate detailed multiphysics feedback. A paper published at the PHYSOR 2016 conference shows that the continuous transport methodology accurately predicts Doppler feedback in LWR simulations and shows promising performance characteristics [?]. Optimization of this continuous transport algorithm is ongoing, and application to TREAT transients will be evaluated when the ability to Doppler broaden  $S(\alpha, \beta)$  kernels on-the-fly has been implemented.

### 3.2 Spatially Continuous Depletion for Multiphysics Simulations

One of the primary challenges of moving to a spatially continuous coupling between OpenMC and MOOSE is the ability to perform core depletion calculations. In a traditional calculation, the rate of change of a nuclide's number density is described by the balance equation below [?]:

$$\begin{aligned} \frac{dN_i(t, \vec{r})}{dt} = & \sum_j \left[ \int_0^\infty \gamma_{ji}(E, t) \sigma_{f_j}(E, t, \vec{r}) \phi(E, t, \vec{r}) dE \right] N_j(t, \vec{r}) \\ & + \left[ \int_0^\infty \sigma_{c_{i-1}}(E, t, \vec{r}) \phi(E, t, \vec{r}) dE \right] N_{i-1}(t, \vec{r}) + \lambda_{i'} N_{i'}(t, \vec{r}) \\ & - \left[ \int_0^\infty (\sigma_{f_i}(E, t, \vec{r}) + \sigma_{c_i}(E, t, \vec{r})) \phi(E, t, \vec{r}) dE \right] N_i(t, \vec{r}) - \lambda_i N_i(t, \vec{r}) \end{aligned} \quad (2)$$

where,

$$\begin{aligned} N(t, \vec{r}) &\equiv \text{number density of a nuclide} \\ \gamma_{ji}(E, t) &\equiv \text{fission production yield of nuclide } i \text{ from nuclide } j \\ \phi(E, t, \vec{r}) &\equiv \text{neutron flux} \\ \lambda &\equiv \text{decay constant of a nuclide} \\ \sigma_f(E, t, \vec{r}) &\equiv \text{microscopic fission cross section of a nuclide} \\ \sigma_c(E, t, \vec{r}) &\equiv \text{microscopic capture cross section of a nuclide} \end{aligned}$$

While the continuous material tracking accommodates continuous nuclide number densities, the traditional approach for integrating the nuclide number densities as shown in equation ?? will only yield a spatially averaged value in each cell. In order to complete the coupling framework between OpenMC and MOOSE, a novel approach to nuclide depletion has been developed[?]. In this new approach, higher order moments of nuclide number densities are integrated through time along with the standard cell-average value represented by the 0th order expansion. This means that the number densities are expanded as a linear combination of polynomials and FETs are utilized for the nuclide reaction rates needed for the time integration. The number density expansion and FET expansion are shown in equations ?? and ??, respectively.

$$N_i(t, \vec{r}) = \sum_{\alpha} N_{i,\alpha}(t) P_{\alpha}(\vec{r}) \quad (3)$$

$$\int_0^\infty dE \phi(E, t, \vec{r}) \sigma_x(E, t, \vec{r}) = \sum_{\beta} R_{\beta}^x(t) P_{\beta}(\vec{r}) \quad (4)$$

where  $\alpha$  and  $\beta$  denote the indices for the moments in a three dimensional space.

The expansions of number densities and reaction rates are then used to yield a system of ordinary differential equations with the number of unknowns equal to the total sum of the number of expansion coefficients for all nuclides. This new system of ODEs is shown in equation ??.

$$\begin{aligned}
C_\gamma \frac{dN_{i,\gamma}(t)}{dt} = & C_\gamma \lambda_{i'} N_{i',\gamma} - C_\gamma \lambda_i N_{i,\gamma} \\
& + \sum_{j,\alpha,\beta} R_\beta^{f,j \rightarrow i}(t) N_{j,\alpha}(t) B_{\alpha,\beta,\gamma} + \sum_{\alpha,\beta} R_\beta^{c,i-1,i}(t) N_{i-1,\alpha}(t) B_{\alpha,\beta,\gamma} - \sum_{\alpha,\beta} \left( R_\beta^{f,i}(t) + R_\beta^{c,i}(t) \right) N_{i,\alpha}(t) B_{\alpha,\beta,\gamma}
\end{aligned} \tag{5}$$

Note that the successful implementation of equation ?? requires that integrals of polynomial triple products must be computed before the simulation. This computation can be difficult, but is only required once for a given basis set expansion.

The continuous depletion methodology was tested on a simple problem shown in Figure ?. The test problem consists of nine fuel pellets with accompanying gas gap and cladding, and the domain is surround by reflective boundary conditions. The center fuel pin contains 2% (weight) Gd-157 and will be the pin that is depleted for comparison purposes. Figure ? shows the calculated number density for Xe-135 for Zernike expansions of order

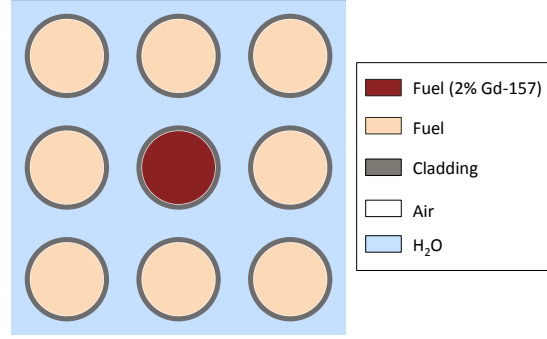


Figure 7: OpenMC geometry model colored by unique material composition

2, 6, and 10 and the L2 error convergence of the expansions. The continuous depletion results were integrated into the same volumes that characterize the discrete case so that a direct comparison of errors between the two solution methods is possible. Note that the abscissa is a cell index instead of a coordinate position. Cell indices are ordered from the origin with sweeps through azimuthal cuts followed by radial rings.

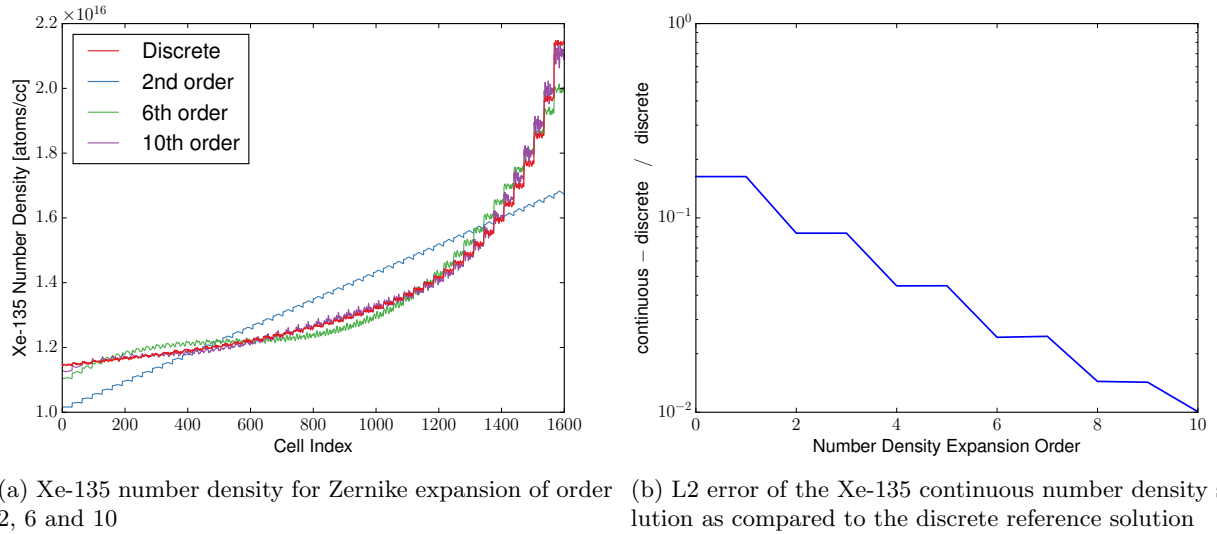


Figure 8: Xe-135 2-month spatially continuous depletion with Zernike expansions compared to a pin discretized with 50 equal volume rings with 32 azimuthal cuts

Figure ?? and Figure ?? show the calculated number density for Gd-157 for 2, 6, and 10th order Zernike expansions with linear and logarithmic scales, respectively. Unlike the Xe-135 number density profile, the Gd-157

spatial distribution is much harder to capture with the Zernike polynomial basis set. This limitation is evident by the negative number densities obtained at the periphery of the pin.

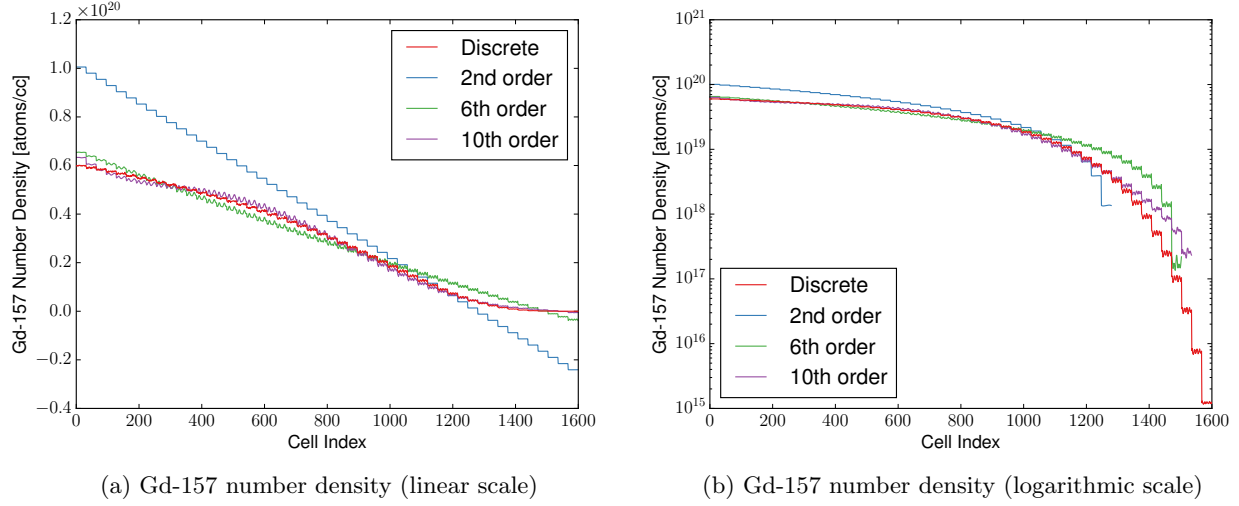
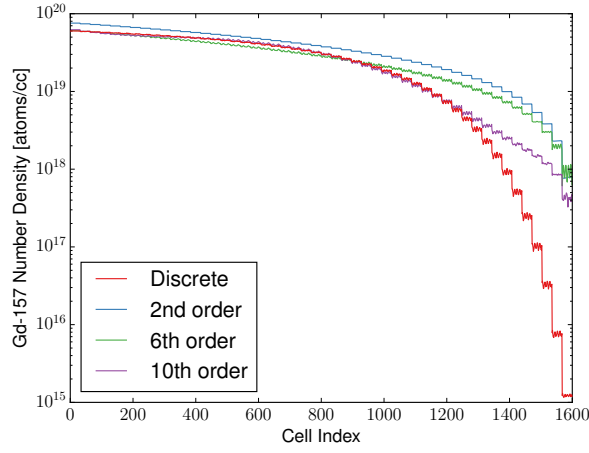


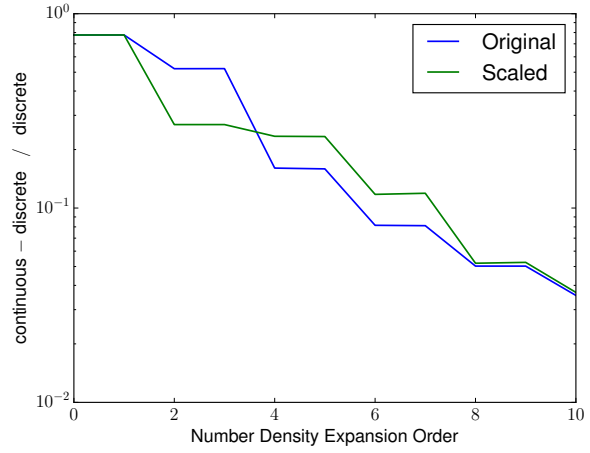
Figure 9: Gd-157 number density with linear and logarithmic scales for Zernike expansion of order 2, 6 and 10, compared to a pin discretized with 50 equal volume rings with 32 azimuthal cuts

The Gd-157 radial number density distribution is one of the most challenging distributions to represent with a polynomial expansion because of the nearly five order of magnitude variation from the center of the pin to a position offset from the pellet surface that shifts inward as the pin is depleted. Figure ?? and Figure ?? show that as the number density expansion order is increased, the prediction of negative number densities is confined to fewer radial rings.

However, even with a higher order expansion, it is impossible to guarantee that the number density will be positive everywhere. As a result, a post processing correction has been implemented to scale all non-zeroth order expansion coefficients of the nuclide number density by a single factor to guarantee that all number densities in the domain remain positive. It is important to note that by only scaling the non-zeroth order expansion coefficient, the total mean nuclide densities are not altered. The Gd-157 number density results after scaling the coefficients is shown in Figure ?. Note that the scaling has minor effects in the interior of the pin where the number density is much larger than on the periphery of the pin. The L2 error for the scaled and unscaled Gd-157 number density solutions are shown in Figure ?.



(a) Gd-157 number density with scaled coefficients



(b) L2 error of the Gd-157 continuous number density solution with scaled and unscaled coefficients

Figure 10: Gd-157 number density for Zernike expansion of order 2, 6 and 10, with scaled coefficients to eliminate negative Gd-157 number density predictions and error convergence for the scaled and unscaled expansions

With the development of the spatially continuous depletion methodology, the coupling framework developed in this work is able to minimize the spatial discretization necessary to couple Monte Carlo codes with multiphysics applications. Ongoing work includes the performance benchmarking of these methods in comparison to the traditional discretization schemes.

## 4 Transient Analysis in Monte Carlo and Method of Characteristics

### 4.1 Background and Motivation

High fidelity methods to precisely describe the behavior of nuclear reactors during transients are critical to their safe operation. The challenge with using transport theory methods such as Monte Carlo (MC) and the Method of Characteristics (MOC) to model reactor transients is the computational expense required to perform fine-grained global calculations. The traditional approaches of using these transport methods to solve transient problems can be separated into two categories:

- **Direct Methods:** The fine-grained flux is propagated forward in time directly within the transport solve. In MC, neutrons and their progeny are propagated until the transient has been completed, which is typically on the order of  $1e5$ - $1e7$  generations for a thermal reactor. In MOC, an approximation is made on the flux derivative term and the transport problem is solved at very fine time step intervals to maintain stability.
- **Indirect Methods:** Directly propagating the neutron flux within the transport solve is very computationally expensive, so indirect methods spatially and energetically condense the cross sections and kinetics parameters to a more computationally-efficient, but highly approximate representation of the problem to be solved. The flux is propagated forward using the spatially and energetically condensed parameters. Periodically, the cross sections and kinetics parameters are updated to account for the changing conditions in the reactor.

Typically, indirect methods spatially condense the cross sections and kinetic parameters down to a single point, resulting in the point reactor kinetics equations (PRKE). While this approach has produced satisfactory results for a wide range of problems, it is not well suited for performing long time steps for transients where the flux shape significantly changes due to the lack of spatial dependence in the PRKE. For this reason, we have developed and implemented a transient analysis method in Monte Carlo (continuous energy and multi-group) and the Method of Characteristics that uses cross sections and kinetics parameters on a coarse-spatial mesh to capture some of the spatial dependence of these parameters in the low-order solve. Fig. ?? shows the geometric meshes that would be used for a direct, indirect coarse-mesh, and indirect point-mesh solves.

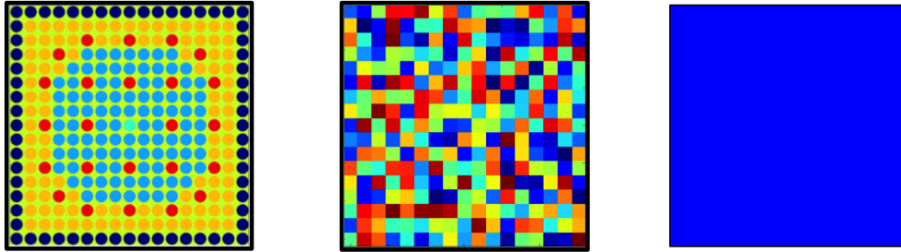


Figure 11: Pictures of the explicit geometry used in direct methods (left), the coarse-mesh representation used in our coarse-mesh indirect method (center), and a point-mesh typically used in indirect methods (right).

### 4.2 Implementation in OpenMC

Recently, we have completed implementation of the transient analysis framework within the OpenMC Monte Carlo code. This required the implementation of numerous features into OpenMC, including the tallying of cross sections, kinetics parameters, and surface currents on a coarse-mesh and adding tallies to calculate the delayed neutron fraction, fission emission spectra for prompt and delayed neutrons, and decay rates for delayed neutron precursors. These tallies have been implemented, tested, and documented for both continuous energy and multi-group MC, allowing for a wide-range of realistic problems. The additional kinetics tallies added to OpenMC will allow others to use our framework to test out other transient methods. Fig. ?? shows the kinetics parameter tallies that were added to OpenMC.

### 4.3 Reactor modeling

With our implementation complete, we are currently focused on analyzing the TREAT reactor as well as the recently released C5G7 time-dependent benchmark. Fig. ?? shows cut-planes of the TREAT reactor with the

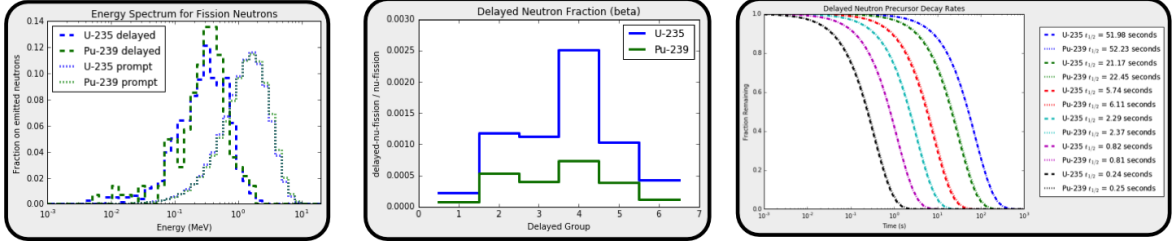


Figure 12: Plots of the fission emission spectra (left), delayed neutron fraction (center), and delayed neutron precursor decay rates (right) for U-235 and Pu-239 in an LWR spectrum.

control rods inserted. In our analysis, we are focusing on modeling the transients in TREAT where the control rods are rapidly withdrawn from the reactor, resulting in a large reactivity insertion within the inner material test location.

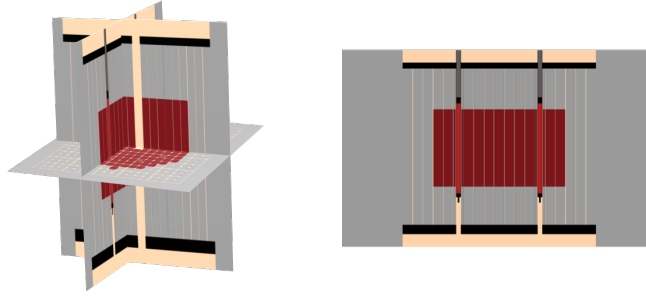


Figure 13: Cut-planes of the TREAT reactor core. The plane on the right shows the control rods inserted into the core along the x-z plane.

Future work will focus on analyzing transient test problems of the TREAT reactor and integrating feedback models that are being developed in our group into the OpenMC transient analysis framework to account for the effects of Doppler broadening, thermal scattering, and materials expansion on the neutron flux during a transient. We anticipate this work will provide a more detailed and accurate representation of the behavior of the TREAT reactor during transient tests. Additionally, our transient analysis framework is designed to be reactor agnostic and therefore extensible to future advanced reactor test cases.

Another important element of our work is that we have implemented the transient analysis framework in continuous-energy Monte Carlo, multi-group Monte Carlo, and multi-group Method of Characteristics. Multi-group cross section libraries can be generated using OpenMC and then directly fed into the transient multi-group Monte Carlo solver within OpenMC. This capability allows multi-group transient test problems to be created using realistic reactor geometries. Additionally, by having transient multi-group Monte Carlo and Method of Characteristic solvers, we can perform cross-method comparisons using the same input data.

# Annual Research Report for Jan 2016 to Aug 2016

Jijie Lou<sup>1</sup> and Jim E. Morel<sup>1</sup>

1.Department of Nuclear Engineering, Texas A&M University

August 22, 2016

## 1 Introduction

During the period from January 2016 to August 2016, the following work has been finished. As a learning exercise, a nonlinear multi-physics application for solving 1D slab thermal radiation diffusion and material equations has been built in MOOSE. Good agreement were shown between the numerical solution and analytical solution. Input files for this 1D slab problem were also merged into Yak. K-eigenvalue calculation for KUCA benchmark[1] problem was performed in Rattlesnake. The results for  $k_{\text{eff}}$  and region-averaged flux from Rattlesnake were consistent with the values in the reference report[1]. Further exploration was performed for LRA benchmark in Rattlesnake. The corresponding input files were merged into Rattlesnake.

Diffusion synthetic acceleration (DSA) has been widely used for accelerating the source iteration process for the Sn equations[2]. However, a challenge arises when treating problems with voids because the diffusion coefficient becomes infinite. The standard approach is to simply use a very large diffusion coefficient. While effective, this can lead to highly ill-conditioned diffusion matrices. A non-local tensor diffusion theory was derived by Morel, et al. [3, 4] that provides a non-zero diffusion coefficient in voids. We applied this theory to generate diffusion coefficients for void in DSA. Decrease in iteration number to solve the linear system in DSA was found. Now we are applying it into discontinuous materials problem to investigate the impact of voids on the effectiveness of DSA. Modification on the code and verification of the result are in process.

Simulation for obtaining quantities of interest (QOIs) of neutron transport need to be accurate and inexpensive. Discretization in energy has been a historical problem due to the resonances. Multi group (MG) method only converges once resonances are resolved, which requires a large amount of energy variables. We are trying to develop a method that will yield convergence with far few degree of freedom (DOF) than standard MG. A comparison of finite element with discontinuous support (FEDS) to SERPENT with continuous-energy monte carlo (MC) for a 2D pin cell was performed. The result showed that FEDS method is a promising alternative to existing energy discretization methods.

## 2 A Non-Linear Multi-Physics Application in MOOSE

### 2.1 Problem specifications

The one group grey thermal radiation diffusion equation is

$$\frac{1}{c} \frac{\partial \phi}{\partial t} - \frac{\partial}{\partial x} D \frac{\partial \phi}{\partial x} = \sigma_a a c T^4 - \sigma_a \phi \quad (1)$$

The material energy equation is

$$C_v \frac{\partial T}{\partial t} = \sigma_a \phi - \sigma_a a c T^4 \quad (2)$$

where  $\phi(x, t)$  is the scalar flux,  $T(x, t)$  is the material temperature,  $c$  is the speed of light,  $a$  is the radiation constant,  $\sigma_a$  is the absorption cross section (opacity),  $D$  is the diffusion coefficient, and  $C_v$  is the heat capacity of the material.

For simplicity in this project, the heat capacity  $C_v$  was set to  $4aT^3$  ( $T$  in unit of keV), and the material was treated as pure absorber with  $D = \frac{1}{3\sigma_a}$ . Then the analytical solution for infinite homogeneous 1D slab can be concluded as:

$$\phi(t) = \frac{\phi_0 + acT_0^4}{2} + \frac{\phi_0 - acT_0^4}{2} \exp(-2\sigma_a ct) \quad (3)$$

$$T(t) = \left\{ \frac{\phi_0 + acT_0^4}{2ac} - \frac{\phi_0 - acT_0^4}{2ac} \exp(-2\sigma_a ct) \right\}^{1/4} \quad (4)$$

where  $\phi(x, 0) = \phi_0$  and  $T(x, 0) = T_0$  are given in the initial condition. Also finite homogeneous 1D slab problem was tested in the application, the solution of which finally came to a steady state of zero due to the leakage of radiation.

## 2.2 Computational Results

The thickness of the slab is 10 cm, and an initial spatially uniform condition is given as below

$$\phi(x, 0) = 12.348 \frac{\text{Jks}}{\text{cm}^2\text{s}}, \quad T(x, 0) = 0.01 \text{ keV} \quad (5)$$

The slab material properties are defined by constant absorption cross section, a null scattering cross section and temperature-dependent heat capacity as indicated in the problem specification. The absorption cross section has a value of  $1 \text{ cm}^{-1}$ . The heat capacity is given by  $4aT^3 \frac{\text{Jks}}{\text{cm}^3\text{keV}}$  with  $T$  in unit of keV. A constant time step sizes of  $\Delta t = 1 \times 10^{-12} \text{ s}$  was applied for the infinite and finite homogeneous 1D slab problem. The calculation was carried out with 200 uniform spatial cells.

### 2.2.1 Infinite homogeneous 1D slab

The numerical results as a function of time from MOOSE along with the analytical solution defined by Eqs. (3) and (4) for the infinite homogeneous 1D slab are plotted in Fig. 1. It can be found from the plots that the numerical solutions show good agreement with the analytical solution.



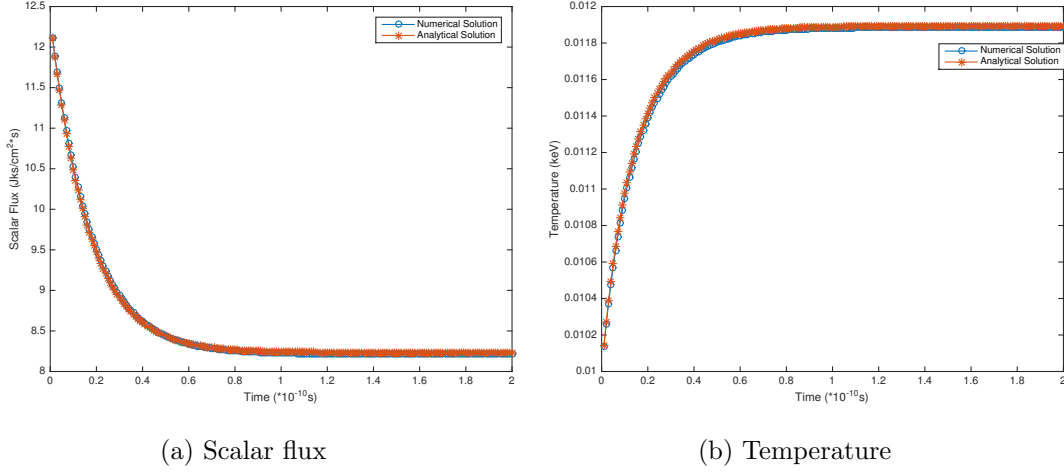


Figure 1: Scalar flux and temperature at arbitrary point for infinite homogeneous 1D slab.

For comparison, we also plotted the material temperature,  $T$ , and the radiative temperature given as  $T_R = (\frac{\phi}{ac})^{1/4}$ . As time increases, they converge to the same value.

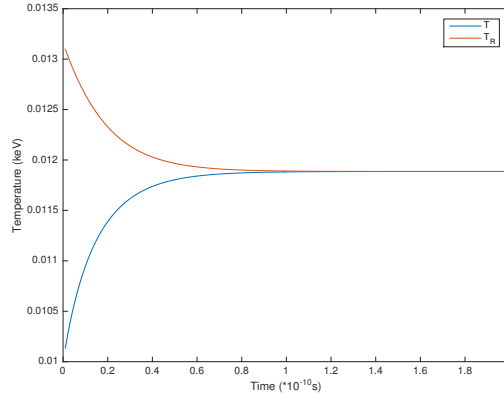


Figure 2: Material and radiative temperature at middle point for infinite homogeneous 1D slab.

### 2.2.2 Finite homogeneous 1D slab

The numerical solution from MOOSE for finite homogeneous 1D slab at several time steps are plotted in Fig. 3. The scalar flux and temperature keep decreasing as time increases, as expected.

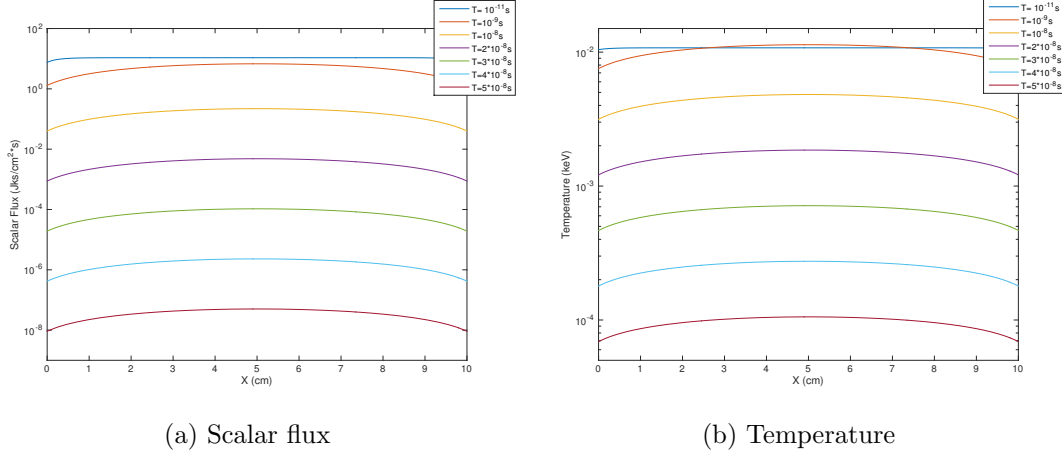


Figure 3: Scalar flux and temperature for finite homogeneous 1D slab.

The material and radiative temperature at the middle point of the slab are plotted in Fig. 4. As time increases, both of them converge to the same value and then decrease to the steady state of zero value due to the leakage of radiation.

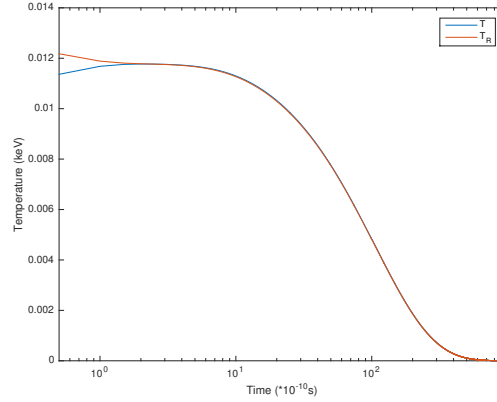


Figure 4: Material and radiative temperature at middle point for finite homogeneous 1D slab.

In this project, an application in MOOSE to solve thermal radiation diffusion and material equations was built. Problem for infinite and finite homogeneous 1D slab was solved using the application. The results, archived from the solving the nonlinear system were compared with the analytical solution. Good agreement are shown. Also, the input files were created and added under yak/tests/actions/radiation\_diffusion\_cfem for the above 1D slab problems.

### 3 K-eigenvalue and Transient Calculation in Rattlesnake

#### 3.1 KUCA benchmark

The description of the KUCA benchmark problem can be found in the reference report [1]. An input file for the KUCA benchmark case 1 was created in Rattlesnake. The macroscopic

cross section input file were written in INSTANT XML format. The problem was solved with SAAF-S8 (transport continuous finite element method (FEM) with S8 Level-Symmetric quadrature) and CD (diffusion continuous FEM) methods, respectively. Region-averaged fluxes were generated using post processors (ElementAverageValue). The results for  $k_{\text{eff}}$  eigenvalue are listed in Table 1.

Table 1: Calculate  $k_{\text{eff}}$  using SAAF-S8 and CD methods

Method	$k_{\text{eff}}$
SAAF-S8	0.979401
CD	0.929341

The region-averaged fluxes are listed in Table 2.

Table 2: Calculate  $k_{\text{eff}}$  using SAAF-S8 and CD methods

Method		Core	Void	Reflector
SAAF-S8	1G	0.00461890	0.0014265	0.000586641
	2G	0.000855297	0.000954291	0.000891332
CD	1G	0.00427129	0.00131496	0.000635652
	2G	0.000815074	0.000834202	0.000941948

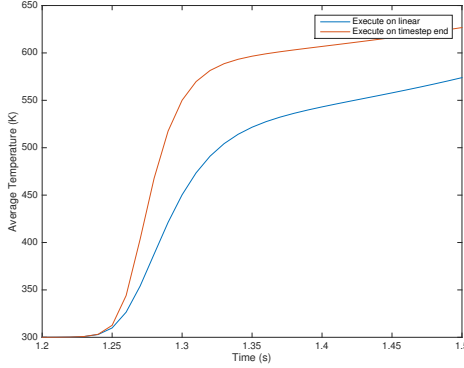
The reported  $k_{\text{eff}}$  using SAAF-S8 is 0.9772, the difference between the results from Rattlesnake and literature is about 0.2%. The cell size used for calculation in the literature is 1 cm  $\times$  1 cm. Due to the calculation time, here we chose the cell size as 1.25 cm  $\times$  1.25 cm, which may account for the difference in the  $k_{\text{eff}}$ . Moreover, in Takeda’s report, it also listed the  $k_{\text{eff}}$  using P1 method : 0.9283 (Fletcher), 0.9234 (Kobayashi). The difference between the results from Rattlesnake using CD method and Kobayashi P1 method is about 0.6%. Comparing the region-averaged flux to the reported values using Sn method, the difference is around 1.3 %-2.7%.

## 3.2 LRA benchmark

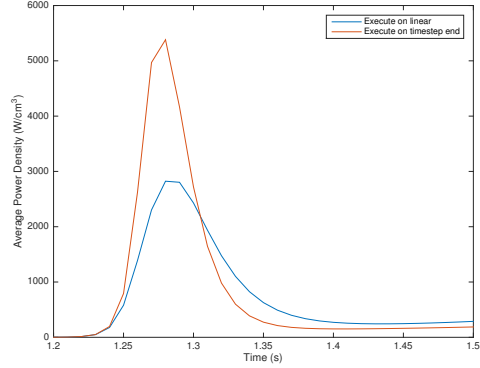
This is a two-group transient benchmark with temperature feedback. Several further exploration were done on the original problems, input files were merged into rattlesnake/tutorials/LRA2D folder.

### 3.2.1 Change the ‘execute\_on’ of the temperature auxiliary kernel from ‘linear’ to ‘timestep\_end’

The execution of the temperature auxiliary kernel was changed from ‘linear’ to ‘timestep\_end’. This modification changed the results: we list result from time 1.2 s to 1.5 s as an example using a constant time step of 0.01 s and zero level of uniform refinement. The reason for the change in result is that when the temperature is computed on ‘linear’, we have up-to-date residual evaluation, and can perform a fully-coupled multi-physics calculation. After using ‘timestep\_end’ execution, we were solving operator split calculation. The maximum relative error in average temperature and average power density can be as much as 22.85% and 115.7%, respectively.



(a) Scalar flux



(b) Temperature

Figure 5: Average temperature and power density using different execution options.

### 3.2.2 Add picard iteration

Picard iteration was added after changing the execution of temperature auxiliary kernel to ‘timestep\_end’. The maximum relative errors in average temperature and power density after adding picard iteration are listed in the following table using the same parameters for calculation:

Table 3: Maximum relative error in average temperature and power to picard iteration

Picard Iteration	Error in Ave. T	Error in Ave. P
1	2.28E-1	1.15E+0
2	5.59E-2	3.30E-1
3	1.84E-2	9.29E-2
4	5.40E-2	2.83E-2
5	2.00E-2	1.08E-2

According to the table, the relative error in average temperature and power decreases with picard iterations. With more Picard iteration, we would be able to reproduce the result of the fully-coupled case.

### 3.2.3 Turn the temperature to a primal variable

The temperature was turned on as a primal variable instead of auxiliary kernel. In this multivariable problem, the residuals of temperature and scalar flux have very different magnitudes before 1 s, which will make the system hard to solve. Therefore, a scaling factor was added to the temperature residual. Also, since the input file had different number of primal variables compared to the ‘lra\_trans\_initial.i’, the ‘MultiAppSystemCopyTransfer’ was no longer applicable. Therefore, ‘Transfers’ type was changed to ‘TransportSystem-VariableTransfer’ in the input file. The results for temperature as auxiliary variable and primal variable converge within iterative tolerance. The maximum relative error in average temperature and power with different scaling on variable temperature is listed in the following table:

Table 4: Maximum relative error using different scaling factors

Scaling	Error in Ave. T	Error in Ave. P
1E-2	5.21E-10	7.51E-09
1E-3	1.66E-11	5.07E-09
1E-4	1.73E-11	5.07E-09

### 3.2.4 Add a conduction term in temperature equation

A conduction term with constant thermal conductivity = 0.01W/cm/K was added to the temperature by including the kernel ‘DiffusionApproximation’. Also, the diffusion coefficient was calculated as 0.00315 cm<sup>2</sup>/s, and added into the input file by adding ‘GenericConstant-Material’. The maximum temperature at time 3 s is listed in Table 5. In order to see the difference, results using diffusion coefficient as 200 cm<sup>2</sup>/s is also included in the table.

Table 5: Maximum temperature at time 3 s

Diffusion Coefficient	Maximum T (KeV)
0	3884.3
0.00315	3884.1
200	3127.1

## 4 Implementation of Non-Local Tensor Diffusion Theory in DSA

The derivation process of non-local tensor diffusion theory can be described as follows. The detailed documentation can be found in the reference papers[3, 4]. For steady-state mono-energetic Boltzmann transport equation with isotropic scattering and isotropic distributed source, we have:

$$\vec{\Omega} \cdot \nabla \psi(\vec{r}, \vec{\Omega}) + \Sigma_t(\vec{r})\psi(\vec{r}, \vec{\Omega}) = \frac{1}{4\pi}Q(\vec{r}) \quad (6)$$

Here,  $Q$  is the summation of the scattering source and distributed source  $q(\vec{r})$ .

$$Q(\vec{r}) = \Sigma_s(\vec{r})\phi(\vec{r}) + q(\vec{r}) \quad (6a)$$

In a local spherical coordinate system centered at point  $\vec{r}_0$ , an arbitrary point,  $\vec{r}$  can be expressed as:

$$\vec{r} = \vec{r}_0 - s \vec{\Omega}. \quad (7)$$

Then we have:

$$\begin{aligned} \psi(\vec{r}_0, \vec{\Omega}) &= \int_0^\infty \frac{Q(\vec{r}_0 - s \vec{\Omega})}{4\pi} e^{-\int_0^s \Sigma_t(\vec{r}_0 - s' \vec{\Omega}) ds'} ds \\ &= \int_0^\infty e^{-\int_0^s \Sigma_t(\vec{r}_0 - s' \vec{\Omega}) ds'} \frac{1}{4\pi} [\Sigma_s(\vec{r}_0 - s \vec{\Omega})\phi(\vec{r}_0 - s \vec{\Omega}) \\ &\quad + q(\vec{r}_0 - s \vec{\Omega})] ds \end{aligned} \quad (8)$$

$$J(\vec{r}_0) = \int_{4\pi} \vec{\Omega} d\Omega \int_0^\infty e^{-\int_0^s \Sigma_t(\vec{r}_0 - s' \vec{\Omega}) ds'} \frac{1}{4\pi} [\Sigma_s(\vec{r}_0 - s \vec{\Omega}) \phi(\vec{r}_0 - s \vec{\Omega}) + q(\vec{r}_0 - s \vec{\Omega})] ds \quad (9)$$

Here we assume there is no distributed source in the problem domain, and approximate the spatial dependence of  $\phi$  with a linear expression about  $\vec{r}_0$ :

$$\phi(\vec{r}) = \phi(\vec{r}_0) - \nabla \phi(\vec{r}_0) \cdot \vec{\Omega} s. \quad (10)$$

Therefore, for a pure scatter,  $\vec{J}(\vec{r}_0)$  can be expressed as :

$$\begin{aligned} J(\vec{r}_0) &\approx \int_{4\pi} \vec{\Omega} d\Omega \int_0^\infty e^{-\int_0^s \Sigma_t(\vec{r}_0 - s' \vec{\Omega}) ds'} \frac{1}{4\pi} \Sigma_t(\vec{r}_0 - s \vec{\Omega}) \times [\phi(\vec{r}_0) - \nabla \phi(\vec{r}_0) \cdot \vec{\Omega} s] ds \\ &= \frac{1}{4\pi} \int_{4\pi} \vec{\Omega} d\Omega \int_0^\infty e^{-\int_0^s \Sigma_t(\vec{r}_0 - s' \vec{\Omega}) ds'} \Sigma_t(\vec{r}_0 - s \vec{\Omega}) ds \phi(\vec{r}_0) \\ &\quad - \frac{1}{4\pi} \int_{4\pi} \vec{\Omega} \cdot \vec{\Omega} d\Omega \int_0^\infty s e^{-\int_0^s \Sigma_t(\vec{r}_0 - s' \vec{\Omega}) ds'} \Sigma_t(\vec{r}_0 - s \vec{\Omega}) ds \nabla \phi(\vec{r}_0) \end{aligned} \quad (11)$$

The first part in Eq. (11) can be expressed as follows:

$$\begin{aligned} &\frac{1}{4\pi} \int_{4\pi} \vec{\Omega} d\Omega \int_0^\infty e^{-\int_0^s \Sigma_t(\vec{r}_0 - s' \vec{\Omega}) ds'} \Sigma_t(\vec{r}_0 - s \vec{\Omega}) ds \phi(\vec{r}_0) \\ &= -\frac{1}{4\pi} \int_{4\pi} \vec{\Omega} d\Omega [\lim_{s \rightarrow 0} e^{-\int_0^s \Sigma_t(\vec{r}_0 - s' \vec{\Omega}) ds'}] \phi(\vec{r}_0) \\ &= 0 \end{aligned} \quad (12)$$

Then we have:

$$J(\vec{r}_0) \approx -\frac{1}{4\pi} \int_{4\pi} \vec{\Omega} \cdot \vec{\Omega} d\Omega \int_0^\infty s e^{-\int_0^s \Sigma_t(\vec{r}_0 - s' \vec{\Omega}) ds'} \Sigma_t(\vec{r}_0 - s \vec{\Omega}) ds \nabla \phi(\vec{r}_0) \quad (13)$$

Therefore, we can conclude the relationship as follows:

$$D_{ij}(\vec{r}_0) = \frac{1}{4\pi} \int_{4\pi} \Omega_i \Omega_j \int_0^\infty s e^{-\int_0^s \Sigma_t(\vec{r}_0 - s' \vec{\Omega}) ds'} \Sigma_t(\vec{r}_0 - s \vec{\Omega}) ds d\Omega \quad (14)$$

In a finite medium with vacuum boundary condition, it becomes

$$D_{ij}(\vec{r}_0) = \frac{1}{4\pi} \int_{4\pi} \Omega_i \Omega_j \int_0^{s_b} s e^{-\int_0^s \Sigma_t(\vec{r}_0 - s' \vec{\Omega}) ds'} \Sigma_t(\vec{r}_0 - s \vec{\Omega}) ds d\Omega \quad (15)$$

where  $s_b$  denotes the distance to the boundary from point  $\vec{r}_0$  along the direction  $-\vec{\Omega}$ ,

After a comparison of integral transport theory and Eq. (15), we found the function  $f$  satisfies the following transport equation with no incoming fluxes:

$$\vec{\Omega} \cdot \nabla f(\vec{r}, \vec{\Omega}) + \Sigma_t f(\vec{r}, \vec{\Omega}) = 1 \quad (16)$$

## 4.1 Calculation Result

### 4.1.1 Problem with embedded void

Calculation were performed for two different geometries with vacuum condition: a  $10\text{ cm} \times 10\text{ cm} \times 10\text{ cm}$  pure scatter with  $1\text{ cm} \times 1\text{ cm} \times 1\text{ cm}$  void in the center, or on the boundary face, respectively. The scatter has  $\Sigma_t = 1\text{ cm}^{-1}$ . There is isotropic distributed source in the material as  $Q = 1\frac{\text{particles}}{\text{cm}^3}$ .

The diffusion coefficient was calculated for both geometries with reflective or vacuum condition, respectively, plotted in Fig. 6 and 7.

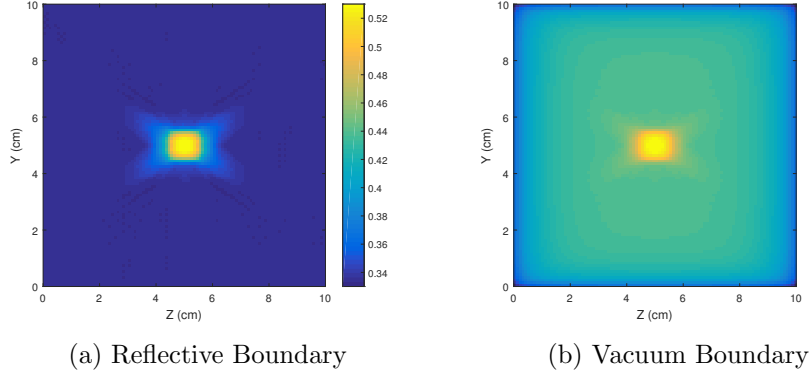


Figure 6: Calculated Result for Diffusion Coefficients at cut plane  $x=5$  in Geometry 1

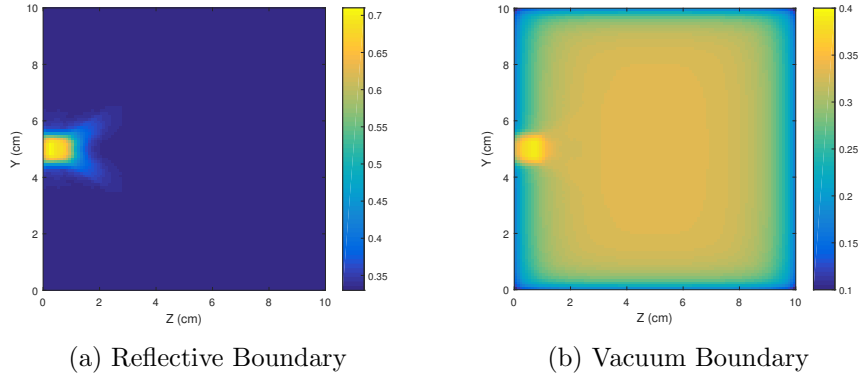


Figure 7: Calculated Result for Diffusion Coefficients at cut plane  $x=5$  in Geometry 2

There were three cases for choosing the diffusion coefficient in DSA: 1) use the diffusion coefficient calculated from Eq. (15) for both material and void; 2) use the diffusion coefficient calculated from Eq. (15) for void, while using  $1/3\Sigma_t$  for material; 3) use 1000 for diffusion coefficient for void, and using  $1/3\Sigma_t$  for material. We recorded the source iteration (SI) number, and iteration number in preconditioned conjugate gradients (PCG) method used in DSA to solve the linear system for the above three cases.

Table 6: Result for geometry 1

Boundary		Case 1	Case 2	Case 3
Reflective	SI	4	4	4
	PCG	1739	1595	3157
Vacuum	SI	4	4	4
	PCG	1944	1594	3157

Table 7: Result for geometry 2

Boundary		Case 1	Case 2	Case 3
Reflective	SI	4	4	5
	PCG	1889	1884	4219
Vacuum	SI	4	4	5
	PCG	2137	1882	4219

Non-local tensor diffusion theory was used to generate the diffusion tensor, which can be approximately given by the pressure tensor associated with the solution to a pure absorber with a constant isotropic source on problem geometry. The effect of various options in the computation of the non-local diffusion coefficients were explored. Reflective condition can preserve the standard diffusion tensor for the material in the current problem, however, with vacuum condition, correction is needed for the neglected boundary term. Based on the result in this work, using the generated diffusion coefficient for voids from reflective condition and standard diffusion coefficient for material in DSA can save the most computation cost, which gives an implication for the future work.

#### 4.1.2 Problem with discontinuous materials

According to the literature, the large discontinuities in cross section can degrade the effectiveness of DSA [5]. The non-local tensor diffusion theory was also applied to investigate the impact of voids on the effectiveness of DSA. We calculated the spectral radius for 2D problem with two discontinuous materials and reflective boundary condition. The problem size is 10 cm  $\times$  10 cm, divided into two equal material blocks. Each material has scattering ratio as 0.999. The spectral radius were calculated for different combination of  $\Sigma_t$  while making the ratio of  $\Sigma_t$  as 1E-6. From Table 8, we can find the degree of degradation changes with different combination of  $\Sigma_t$ , even the ration of cross section across the interface being constant.



Table 8: Spectral Radius for Discontinuous Material Problem

Material 1	Material 2	Spectral Radius	
$\Sigma_t$	$\Sigma_t$	$D=\min(D_{\text{calculated}}, 1/(3\Sigma_t))$	$D=1/(3\Sigma_t)$
1	1	0.2489	0.2485
1	1.00E-06	0.4155	0.4405
10	1.00E-05	0.7617	0.7819
100	1.00E-04	0.8446	0.8486
1000	1.00E-03	0.8618	0.8614
10000	1.00E-02	0.5099	0.5100
100000	1.00E-01	0.1679	0.1882
1000000	1.00E+00	0.2467	0.2469

## 5 Comparison of PDT Result Using the Finite-Element-With-Discontiguous-Support (FEDS) or SERPENT Cross Section with Continuous-Energy Monte Carlo for a 2D Pincell Problem

### 5.1 FEDS method

FEDS is a Petrov-Galerkin finite element method in energy, detailed method description can be found in the references [6, 7]. The only approximation we used in this method is:

$$\psi_{\text{exact}}(\mathbf{r}, E, \mathbf{\Omega}, t) \approx \psi_{\text{FEDS}}(\mathbf{r}, E, \mathbf{\Omega}, t) \equiv \sum_k b_k(\mathbf{r}, E) \psi_k(\mathbf{r}, \mathbf{\Omega}, t) \quad (17)$$

The basis functions are:

$$b_k(r \in V_i, E) = \begin{cases} C_{i,k} f_i(E), & E \in E_k \\ 0, & \text{otherwise} \end{cases} \quad (18)$$

where  $f_i(E)$  is an approximate solution spectrum in region  $i$  and  $C_{i,k} = \frac{1}{\int_{E_k} dE f_i(E)}$ .

The weight functions are:

$$w_k(E) = \begin{cases} 1, & \text{if } E \in E_k \\ 0, & \text{otherwise} \end{cases} \quad (19)$$

We begin with the continuous-energy neutron equation with isotropic scattering and fission:

$$\begin{aligned} \mathbf{\Omega} \cdot \nabla \psi(\mathbf{r}, E, \mathbf{\Omega}) + \Sigma_t(\mathbf{r}, E) \psi(\mathbf{r}, E, \mathbf{\Omega}) &= \frac{1}{4\pi} \int_0^\infty dE' \Sigma_s(\mathbf{r}, E' \rightarrow E) \phi(\mathbf{r}, E') \\ &+ \frac{\chi(\mathbf{r}, E)}{4\pi k_{\text{eff}}} \int_0^\infty dE' \nu \Sigma_f(\mathbf{r}, E') \phi(\mathbf{r}, E') \end{aligned} \quad (20)$$

Then test the transport equation against the weight functions and expand the fluxed into their basis function representations. After algebraic manipulation we get what looks like a standard MG formulation:

$$\begin{aligned} \mathbf{\Omega} \cdot \nabla \psi_k(\mathbf{r}, \mathbf{\Omega}) + \Sigma_{t,k,i} \psi_k(\mathbf{r}, \mathbf{\Omega}) &= \frac{1}{4\pi} \sum_{k'} \Sigma_{s,k' \rightarrow k,i} \phi_{k'}(\mathbf{r}) \\ &+ \frac{\chi_{k,i}}{4\pi k_{\text{eff}}} \sum_{k'} \nu \Sigma_{f,k',i} \phi_{k'}(\mathbf{r}) \end{aligned} \quad (21)$$

Cross sections are now averaged over discontinuous energy domains instead of continuous ones. The cross sections become, for  $r \in V_i$ :

$$\Sigma_{t,k,i} \equiv \int_0^\infty dE b_k(\mathbf{r}, E) \Sigma_{t,i}(E) \quad (22)$$

$$\chi_{k,i} \equiv \int_0^\infty dE w_k(E) \chi_i(E) \quad (23)$$

$$\psi_k(\mathbf{r}, \mathbf{\Omega}, t) \equiv \int_0^\infty dE w_k(E) \psi(\mathbf{r}, E, \mathbf{\Omega}, t) \quad (24)$$

$$\Sigma_{s,k' \rightarrow k,i} \equiv \int_0^\infty dE' b_{k'}(\mathbf{r}, E') \int_0^\infty dE w_k(E) \Sigma_{s,i}(E' \rightarrow E) \quad (25)$$

The energy mesh is determined by solving the minimization problem using clustering, and the basis functions are decided by computing problem-dependent infinite-medium slowing down spectra with analytic escape cross section on a hyper-fine group structure for each material. Then cross section on the contiguous subelements for each material are computed using NJOY, and combined into element cross section for each material. Those cross section can be further applied in a standard MG transport code.

## 5.2 Results for Comparison

### 5.2.1 Problem Description

We solved a k-eigenvalue problem for a 2D infinite lattice of pin cell made of  $\text{UO}_2$  fuel: 10.29769 g/cm<sup>3</sup>, 96% U-238, 4% U-235, and  $\text{H}_2\text{O}$  moderator: 0.740582 g/cm<sup>3</sup>. The radius of fuel is 0.39218 cm (inner fuel radius as 0.32021 cm for later quantities of interest (QOIs) analysis), the length of pitch is 1.25984 cm. Cross section for two material zones (fuel and moderator) were generated from FEDS and SERPENT methods, respectively.

### 5.2.2 FEDS Cross Section

We created the energy mesh hierarchically. We first divided the energy variable into 12 coarse group, listed in Table 9. For the energy range outside the reserved resonance range (RRR), we applied the SHEM-361[8] group boundaries. Inside the RRR, from coarse group 3-10, we applied FEDS method to decide the energy element. We chose our coarse group to be hierarchical to the SHEM-361 group boundaries.

Table 9: Coarse groups used for QOI

Coarse Group	Upper Energy (eV)	Uses Discontiguous Subdomains
I	$2.00000 \times 10^7$	No
II	$1.40000 \times 10^5$	No
III	$2.26994 \times 10^4$	Yes
IV	$9.11881 \times 10^3$	Yes
V	$2.08410 \times 10^3$	Yes
VI	$5.39204 \times 10^2$	Yes
VII	$1.54176 \times 10^2$	Yes
VIII	$5.17847 \times 10^1$	Yes
IX	$2.78852 \times 10^1$	Yes
X	$9.50002 \times 10^0$	Yes
XI	$4.21983 \times 10^0$	No
XII	$6.24999 \times 10^{-1}$	No

### 5.2.3 Cross Section Generated by SERPENT

The same ace files were used by SERPENT and MCNP. Those ace files were generated using ENDF/B-VII.1 cross sections at 293.6 K processed with NJOY 99.364. The macroscopic cross sections for fuel and moderator were obtained using the group constant generation in SERPENT by giving the above materials different universe numbers. The microscopic cross sections for U-235 and U-238 were generated using a detector card. Three different multi group (166, 244, 361) cross sections were produced from SERPENT using 2000 cycles with 50 inactive cycles (used shannon entropy of fission distribution for assessing convergence of MC). A resolution study on the particles/cycle was performed to attribute the difference in k-eigenvalue is below 2 pcm.

### 5.2.4 Parameters used in PDT

For each pincell, 12 rings of cells in the fuel were used: 4 in the center and 8 near its boundary, and 6 rings of cells in the moderator. Within each cell, lumped PWLD was used. A product Gauss-Legendre / Gauss-Chebyshev quadrature was used for the angular quadrature, with the 16 polar angles and 40 azimuthal angles in one quadrant. P3 scattering was used for calculation. For one pincell, the calculation used 5184 spatial unknowns and 2560 angular unknowns in total. The mesh used in PDT is plotted in Fig. 8. Those parameters were decided from a previous resolution study, which allows us to attribute discrepancies of more than 5 pcm between PDT and MCNP to energy discretization effects alone.

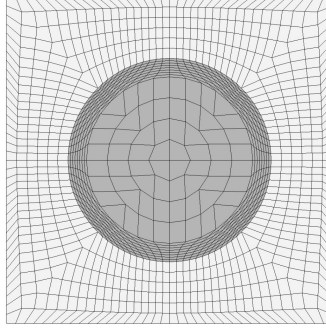


Figure 8: Mesh used in PDT.

### 5.2.5 Result for comparison

The errors in  $k_{\text{eff}}$  are listed in Table 10. The minimum error among the three methods for the same number of energy unknown are color-marked. When comparing FEDS and SERPENT methods for the case of 30 unknowns, SERPENT gives much better performance. However, with increasing number of energy elements, they have almost the same magnitude of errors. We define efficiency such that a method with first-order convergence has a constant efficiency. Errors in the FEDS method are initially larger but convergence is more uniform. The MG-SHEM results are the best for 361-group case, which is perhaps expected since the SHEM sets are highly hand-tuned. However, there is little difference between any of the methods at the highest resolution. Three points is not adequate to establish order accuracy, but other calculations made with FEDSMG have indicated that it begins to converge with first-order accuracy with far fewer unknowns than “standard” multigroup[7]. We can efficiently generate FEDSMG sets with very high energy resolution, but the SERPENT method quickly becomes impractical as the number of groups is increased, and it is not clear as to the cost or feasibility of generating higher resolution SHEM sets.

Table 10: Errors in  $k_{\text{eff}}$  for various energy resolution

Method	Unknowns (RRR <sup>1</sup> )	Keff	Rel. Error (P-M)/M (%)	Error (P-M) (pcm)	Efficiency $\frac{1}{ \text{Error}  \times \text{DOF}}$
FEDS-166	30	1.47075	0.42	609.82	5.5
FEDS-244	108	1.46680	0.15	214.69	4.3
FEDS-361	225	1.46544	0.05	79.44	5.6
SHEM <sup>2</sup> -166	30	1.46875	0.28	409.74	8.1
SHEM-244	108	1.46767	0.21	301.81	3.1
SHEM-361	225	1.46410	-0.04	-55.04	8.1
SERPENT-166	30	1.46073	-0.27	-392.04	8.5
SERPENT-244	108	1.46268	-0.13	-197.14	4.7
SERPENT-361	225	1.46385	-0.05	-79.51	5.6

The coarse-group QOIs are plotted in Fig 9-16. The maximum absolute errors in coarse-group QOIs are listed in Table 11. The SHEM results are much less accurate at low resolution for the QOIs than for the eigenvalue. At highest resolution, FEDSMG is best

for 3 of 7 QOI's and SHEM is best for 4 of 7 QOI's. In some cases the SERPENT error increases with increasing resolution, but the FEDSMG and SHEM errors always decrease.

Table 11: Maximum Absolute Error in coarse-group QOIs

Method	U-238 Abs. (Inner fuel)	U-238 Abs. (Outer fuel)	U-238 Abs (Fuel)	U-235 Abs. (Fuel)	U-235 Fiss. (Fuel)	Abs. (Fuel)	Fiss. (Fuel)
FEDS-166	190.9	330.8	237.6	828.3	706.8	928.0	707.9
FEDS-244	67.2	131.3	81.8	287.2	245.3	332.1	246.4
FEDS-361	29.0	44.6	30.1	162.0	138.6	194.3	139.7
SHEM-166	1759.5	2392.7	486.6	524.4	447.7	593.4	448.8
SHEM-244	472.9	2062.7	372.3	364.6	311.4	417.4	312.5
SHEM-361	443.7	765.0	40.8	50.8	43.7	71.8	44.8
SERPENT-166	1600.1	2672.2	175.9	261.5	218.1	318.8	218.1
SERPENT-244	891.9	1630.2	80.5	252.5	211.1	307.2	211.1
SERPENT-361	438.9	777.2	68.9	313.4	271.7	301.4	271.1

Table 12: Error in total QOIs

Method	U-238 Abs. (Inner fuel)	U-238 Abs. (Outer fuel)	U-238 Abs (Fuel)	U-235 Abs. (Fuel)	U-235 Fiss. (Fuel)	Abs. (Fuel)	Fiss. (Fuel)
FEDS-166	-285.1	-1326.5	-633.5	557.5	543.4	-68.4	522.7
FEDS-244	-149.0	-386.1	-229.3	199.9	207.7	-21.8	187.0
FEDS-361	-57.0	-162.6	-93.5	74.9	92.8	-11.0	72.1
SHEM-166	4397.0	-9981.5	-398.2	401.7	371.0	26.8	334.7
SHEM-244	2240.6	-5297.3	-274.3	291.9	279.4	40.9	243.0
SHEM-361	1140.7	-1944.1	110.2	-65.0	-23.7	68.6	-60.1
SERPENT-166	5175.9	-9421.4	308.4	-192.8	-177.5	54.1	-181.7
SERPENT-244	2665.6	-5006.4	106.6	-11.5	-9.5	33.1	-13.4
SERPENT-361	999.4	-1966.5	9.2	77.1	73.0	23.6	69.0

A comparison of FEDS to SERPENT with continuous-energy MC for a 2D pincell was performed in the current study. The FEDSMG method is a promising alternative to existing energy discretization methods. The current version uses a trivial amount of geometric information (the escape cross section defined as  $S/(4V)$ ), yet gives good results overall. It appears practical to generate very high resolution sets using the FEDSMG method. This does not appear to be the case for SERPENT.

## 6 Future Work

Modification on code for implementation of non-local tensor diffusion theory in 2D DSA with Sn method are in process. It will be applied to mock up the standard model problem to see if we do in fact get the effects predicted by Fourier analysis and help verify the overall result.

We would like to continue the comparison of FEDS and SERPENT methods for TREAT modeling, starting from a 3x3 pin cell problem. We want to evaluate the effect of increasing

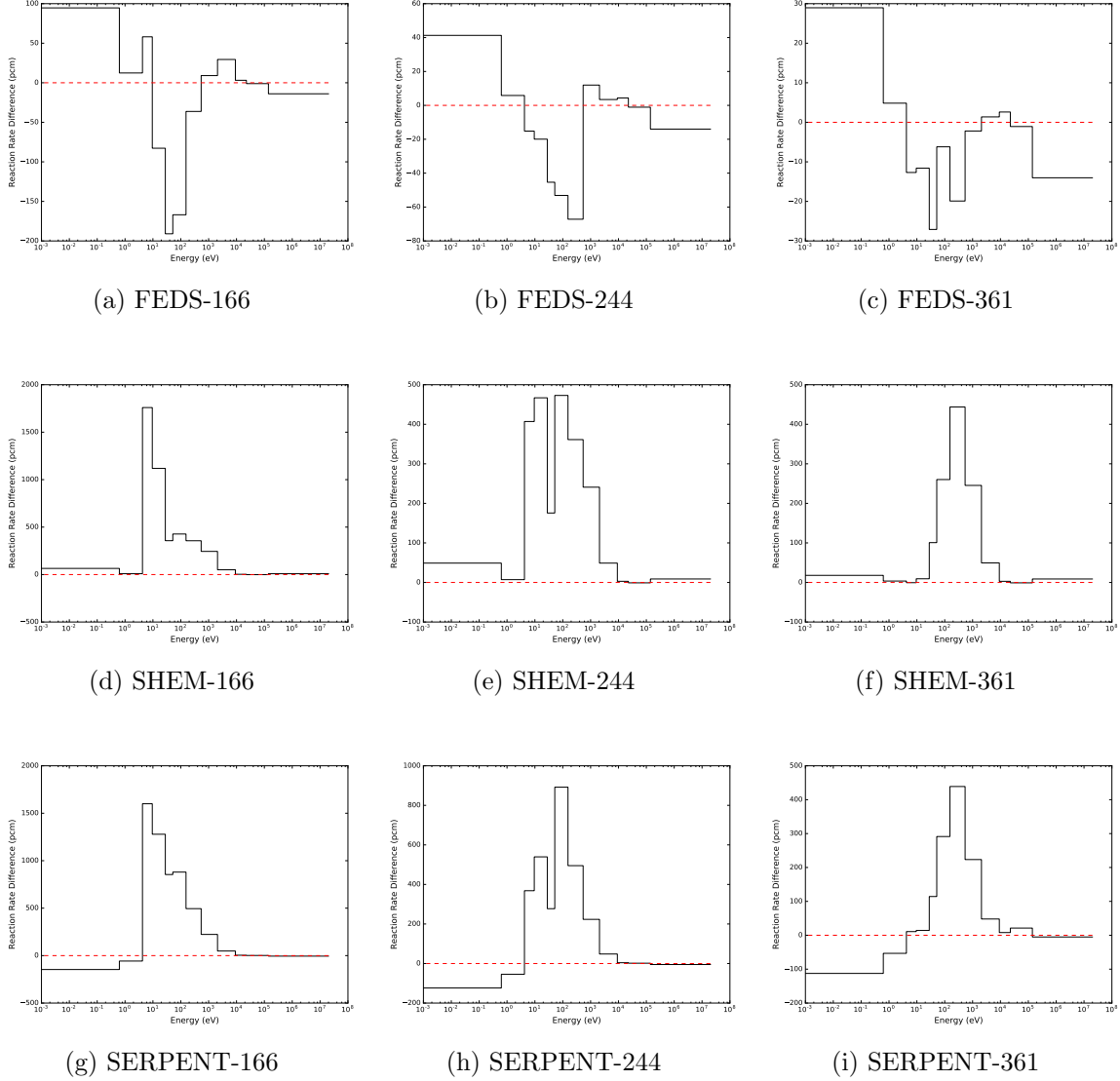


Figure 9: U-238 absorption errors in inner fuel between MCNP and PDT for different energy structures. Columns have the same total energy unknowns in the RRR (30, 108, and 225). Rows have the same mesh family (FEDS, MG, SERPENT)

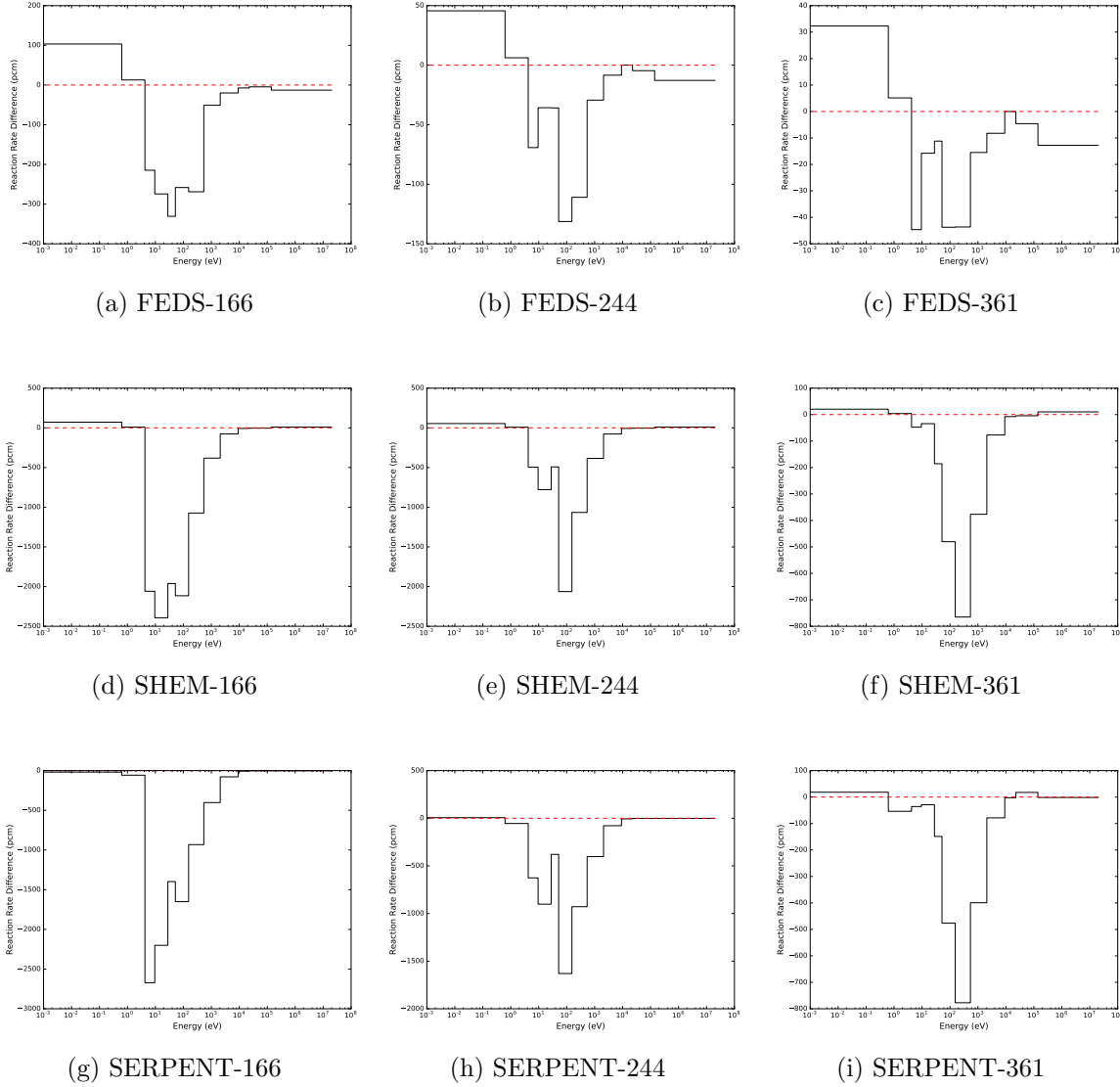


Figure 10: U-238 absorption errors in outer fuel between MCNP and PDT for different energy structures. Columns have the same total energy unknowns in the RRR (30, 108, and 225). Rows have the same mesh family (FEDS, MG, SERPENT)

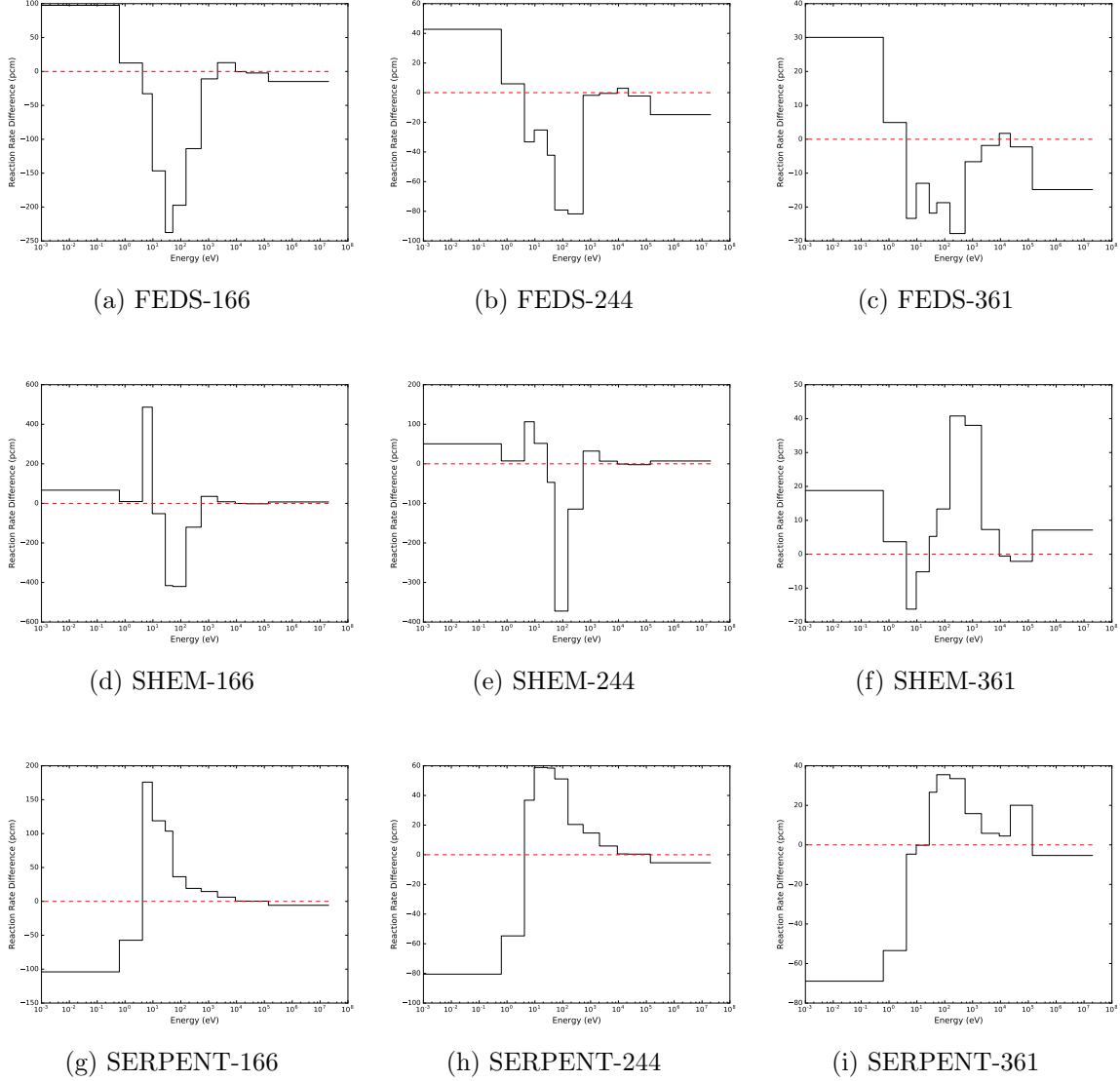


Figure 11: U-238 absorption errors in fuel between MCNP and PDT for different energy structures. Columns have the same total energy unknowns in the RRR (30, 108, and 225). Rows have the same mesh family (FEDS, MG, SERPENT)



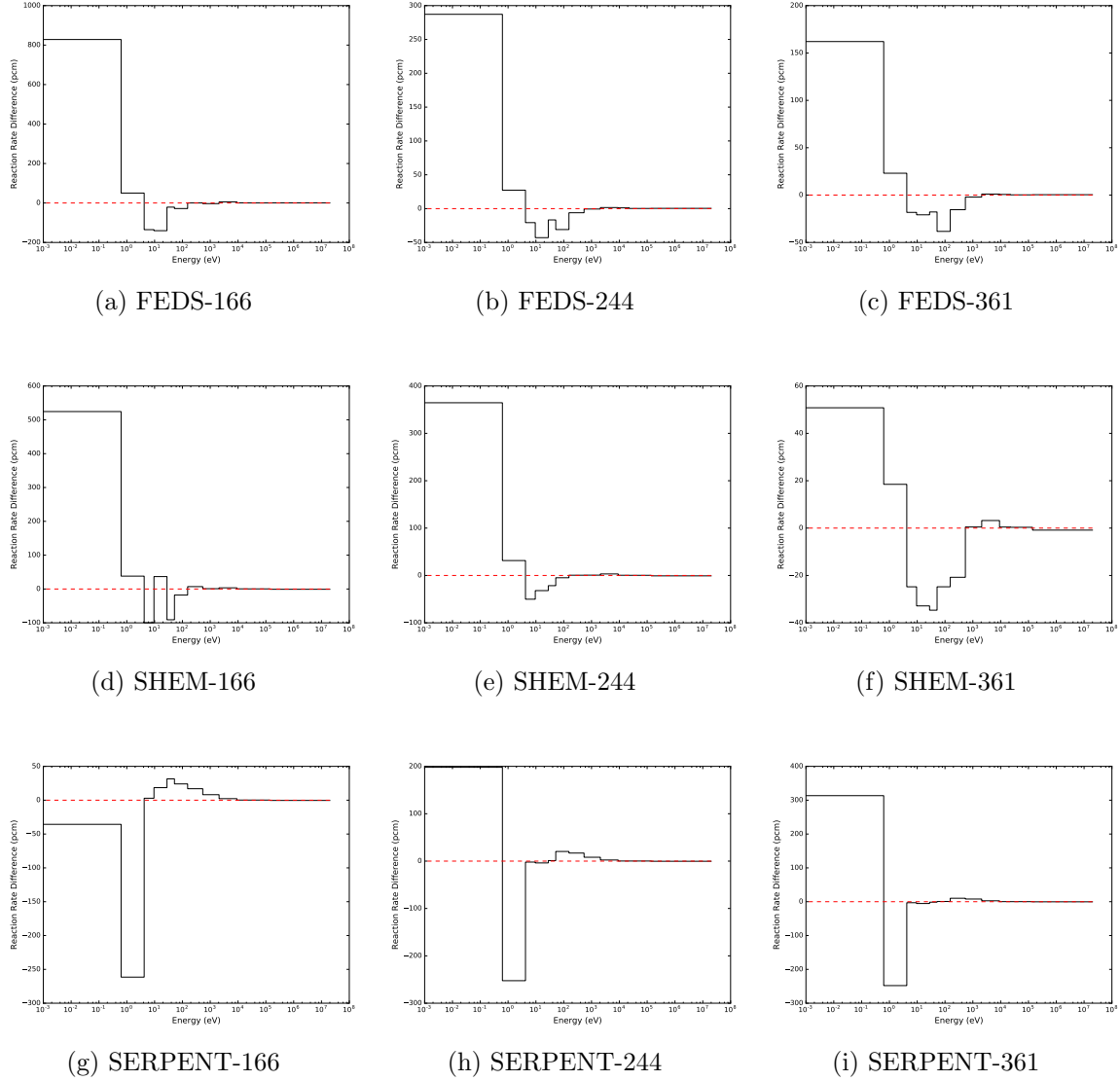


Figure 12: U-235 absorption errors in fuel between MCNP and PDT for different energy structures. Columns have the same total energy unknowns in the RRR (30, 108, and 225). Rows have the same mesh family (FEDS, MG, SERPENT)

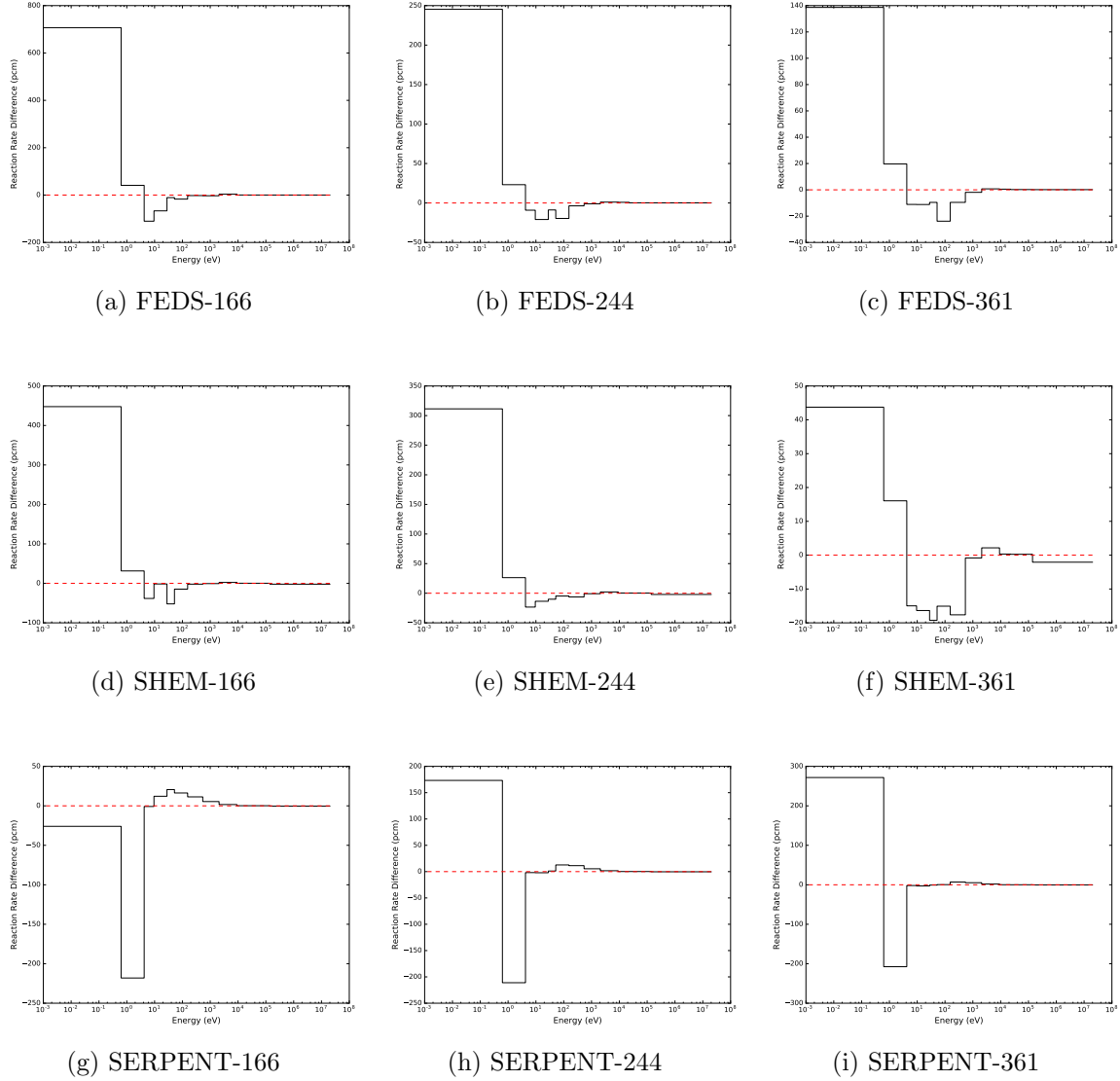


Figure 13: U-235 fission errors in fuel between MCNP and PDT for different energy structures. Columns have the same total energy unknowns in the RRR (30, 108, and 225). Rows have the same mesh family (FEDS, MG, SERPENT)

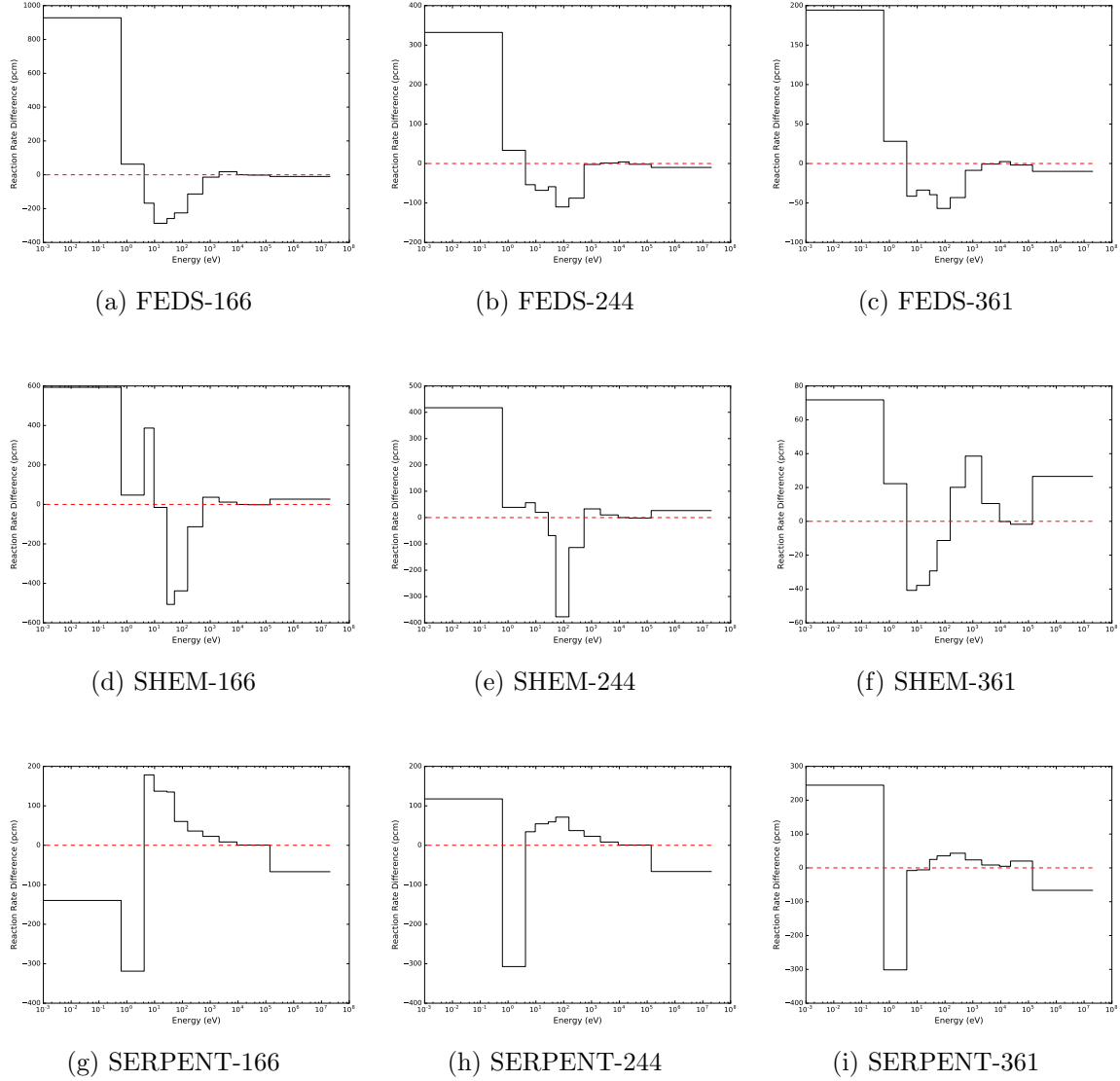


Figure 14: Total absorption errors in fuel between MCNP and PDT for different energy structures. Columns have the same total energy unknowns in the RRR (30, 108, and 225). Rows have the same mesh family (FEDS, MG, SERPENT)

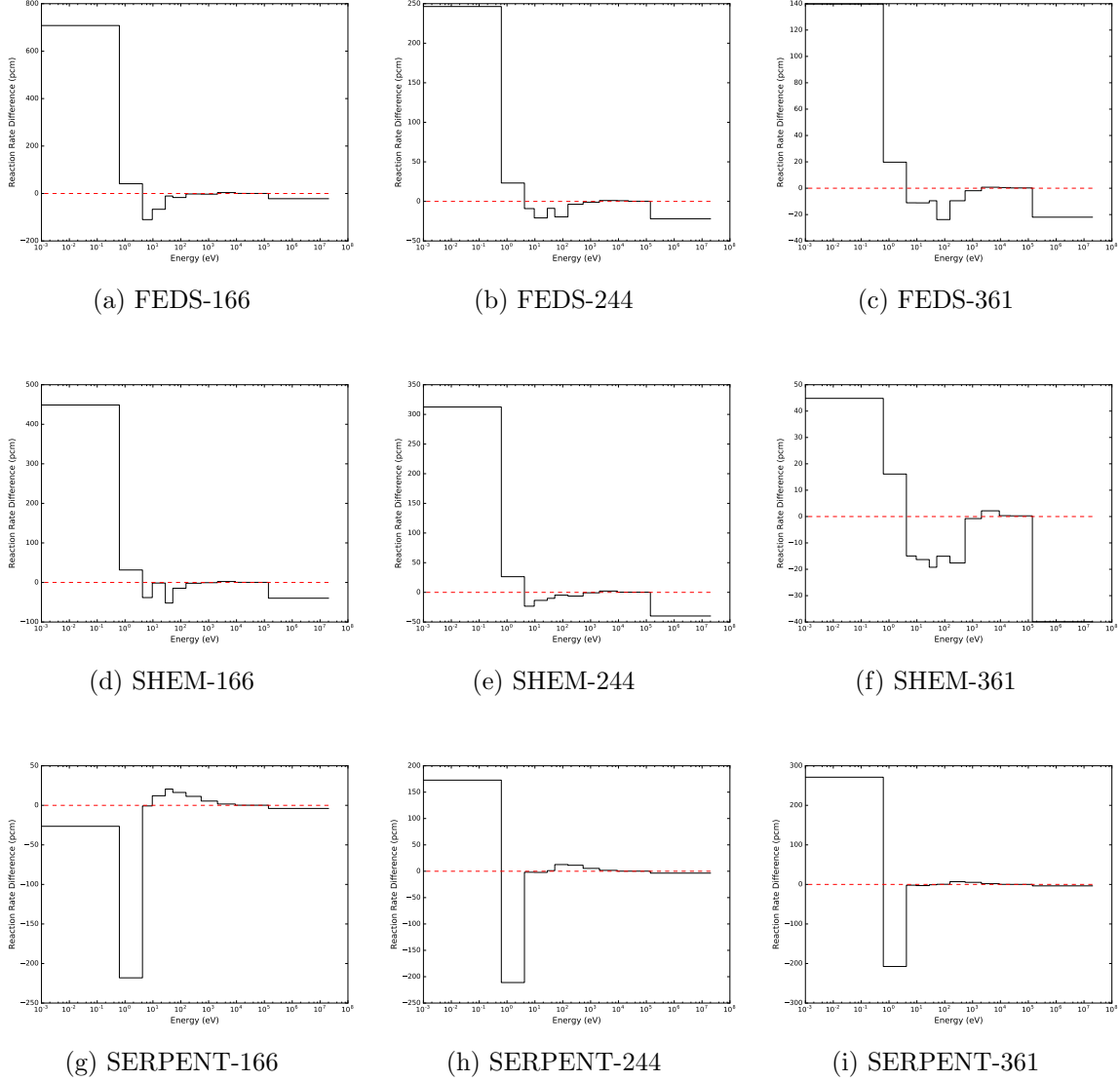


Figure 15: Total fission errors in fuel between MCNP and PDT for different energy structures. Columns have the same total energy unknowns in the RRR (30, 108, and 225). Rows have the same mesh family (FEDS, MG, SERPENT)

the number of regions within the fuel and moderator. We want to investigate space and energy dependent escape cross section for the FEDSMG method via continuous-energy MC with a detailed geometry. Escape cross sections may be much cheaper to generate than full spectra for MG averaging. There are other potential FEDSMG improvements that we want to investigate relating to discontinuous group structure generation.

## References

- [1] T. TAKEDA, H. IKEDA. *3D Neutron Transport Benchmark*. OECD/NEA Committee on Reactor Physics, NEACPR-L-330.
- [2] R. E. Alcouffe. *Diffusion Synthetic Acceleration Methods for the Diamond-Differenced Discrete-Ordinate Equations*. Nuclear Science and Engineering: 64,344-355 (1977).
- [3] J. E. Morel., CCS-2 and Texas A&M University. *A Non-Local Tensor Diffusion Theory*. Los Alamos National Laboratory Report, LA-UR-07-5257 (2007).
- [4] J. E. Morel., CCS-2 and Texas A&M University. *Alternative Generation of Non-Local Diffusion Tensors*. Los Alamos National Laboratory Report, LA-UR-10-08285 (2010).
- [5] J. S. Warsa, T. A. Wareing, and J. E. Morel. *Krylov Iterative Methods and the Degraded Effectiveness of Diffusion Synthetic Acceleration for Multidimensional SN Calculations in Problems with Material Discontinuities* Nuclear Science and Engineering, 147, 218-248 (2004).
- [6] A. T. Till, M. L. Adams, and J. E. Morel. *The Finite Element with Discontiguous Support Multigroup Method: Theory*. Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA), and the Monte Carlo (MC) Method. American Nuclear Society, Nashville, Tennessee (2015).
- [7] A. T. Till. *Finite Elements with Discontiguous Support for Energy Discretization in Particle Transport*. Ph.D. thesis, Texas A&M University, College Station, TX (2015).
- [8] A. Hbert and A. Santamarina. *Refinement of the Santamarina-Hfaiedh Energy Mesh Between 22.5 eV and 11.4 keV*. International Conference on the Physics of Reactors (PHYSOR). American Nuclear Society, Interlaken, Switzerland (2008).

# Annual research report for Oct 2015 to Sep 2016

Hans R. Hammer<sup>1</sup> and Jim E. Morel<sup>1</sup>

<sup>1</sup>Texas A&M University - Department of Nuclear Engineering

September 6, 2016

## 1 Introduction

Our application are detailed calculations of modern nuclear reactors. Current development in modeling and simulation raised the needs for tools, that can handle voids or near voids. While this is definitely possible with the first order transport equation, second order schemes often show singularities, condition or convergence problems for (near) zero total cross sections. However, second order forms of the transport equation offer the advantage of using continuous finite elements (CFEM), which is especially appealing with INL's multiphysics framework MOOSE [1].

Least-squares (LS) forms of the transport equation can avoid these problems, but are non-conservative. This is why they are not commonly used in the nuclear community. A newly developed form of least-squares transport equation is compatible with voids and standard solution techniques, but is non-conservative [2], therefore conservation of particles is only achieved as the numerical solution converges to the analytical solution.

This can be corrected by the use of the nonlinear diffusion acceleration. While this acceleration scheme significantly decreases computational cost, it also can ensure conservation when the conservative low order equation is used. In order to achieve conservation on the low order equation, it must be inconsistent with the high order LS equation. This leads to the convergence of the solutions for the high order and low order solve only in the limit as the spatial mesh is increasingly refined. The basic implementation in YAK of the LS-SN-CFEM-NDA scheme was finished 2014 [7].

While the high order LS equation is valid in voids, the nonlinear diffusion acceleration employs a low order diffusion equation. This equation does not hold in void

due to the definition of the diffusion coefficient. Work was done to find an alternative definition, that holds in voids. Additionally, the drift vector, coupling the low order equation with the high order equation, is also not defined in a void. A Fourier analysis was performed to find a way of describing the drift vector in voids.

Recent calculations showed, that unweighted LS behaves poorly in voids or at strong absorber boundaries, although no problems have been observed in realistic k-eigenvalue calculations. The unweighted LS is influenced by adjacent material regions. Uncoupling by introducing a interior boundary condition solves the problem, but the matrix is not symmetric positive-definite any more. A different approach is the introduction of a weight function into the least-squares transport equation.

## 2 Weighted Least-Squares Transport Equation

The weighted least-squares transport equation addresses some issues of the unweighted LS equation on material interfaces and in voids. Due to the second order nature of the LS equation, downstream information can influence the upstream solution. A optically thick material further downstream the current direction reduces the flux in the unweighted LS for coarse meshes. The introduction of a weight function reduces this problem significantly. The weighted LS equation is defined by:

Given a trial space  $W_{\mathcal{D}}$ , consisting of continuous basis functions, the weak form for a specific direction and group is as follows: find  $\psi_{m,g} \in W_{\mathcal{D}}$  such that

$$b_m^g(\psi_{m,g}, \psi_{m,g}^*) = l_m^g(\psi_{m,g}^*), \quad \forall \psi_{m,g}^* \in W_{\mathcal{D}}, \quad (1)$$

where

$$\begin{aligned} b_m^g(\psi_{m,g}, \psi_{m,g}^*) \equiv & \left( w_{m,g} \vec{\Omega}_m \cdot \vec{\nabla} \psi_{m,g}, \vec{\Omega}_m \cdot \vec{\nabla} \psi_{m,g}^* + \sigma_{t,g} \psi_{m,g}^* \right)_{\mathcal{D}} \\ & + \left( w_{m,g} \sigma_{t,g} \psi_{m,g}, \vec{\Omega}_m \cdot \vec{\nabla} \psi_{m,g}^* + \sigma_{t,g} \psi_{m,g}^* \right)_{\mathcal{D}} \\ & + \left\langle w_{m,g} f_m \psi_{m,g}, \psi_{m,g}^* \right\rangle_{\partial \mathcal{D}_m^-} \end{aligned} \quad (2)$$



and

$$\begin{aligned}
l_m^g(\psi_{m,g}^*) \equiv & \left( w_{m,g} \sum_{g'=1}^G \sum_{l=0}^L \sum_{p=-l}^l \frac{2l+1}{4\pi} \sigma_l^{g' \rightarrow g} \phi_{l,g'}^p Y_{l,m}^p, \vec{\Omega}_m \cdot \vec{\nabla} \psi_{m,g}^* + \sigma_{t,g} \psi_{m,g}^* \right)_{\mathcal{D}} \\
& + \left( w_{m,g} \frac{\chi^g}{4\pi} \sum_{g'=1}^G \nu \sigma_f^{g'} \phi_0^{0,g'}, \vec{\Omega}_m \cdot \vec{\nabla} \psi_{m,g}^* + \sigma_{t,g} \psi_{m,g}^* \right)_{\mathcal{D}} \\
& + \left( w_{m,g} \frac{1}{4\pi} q_g, \vec{\Omega}_m \cdot \vec{\nabla} \psi_{m,g}^* + \sigma_{t,g} \psi_{m,g}^* \right)_{\mathcal{D}} \\
& + \left\langle w_{m,g} f_m \psi_m^{inc,g}, \psi_{m,g}^* \right\rangle_{\partial \mathcal{D}_m^-}.
\end{aligned} \tag{3}$$

with the optional weak boundary condition

$$\left\langle w_{m,g} f_m (\psi_{m,g} - \psi_m^{inc,g}), \psi_{m,g}^* \right\rangle_{\partial \mathcal{D}_m^-}, \quad m = 1..M, \quad g = 1..G, \tag{4}$$

where  $\partial \mathcal{D}_m^-$  is the portion of the boundary for which  $\vec{\Omega}_m \cdot \hat{n} < 0$ . Calculation showed that  $f_m = \sigma_t |\vec{\Omega} \cdot \vec{n}|$  is a suitable choice. For near void problems however the choice

$$f_m = \max \left( \sigma_t, \frac{1}{h} \right) |\vec{\Omega} \cdot \vec{n}| \tag{5}$$

gives a more stable version. Here  $h$  denotes a length constant of the boundary cell. Another choice that gave good convergence in near void problems is

$$f_m = 4 \frac{|\vec{\Omega} \cdot \vec{n}|}{h} + \sigma_t \left( 3 |\vec{\Omega} \cdot \vec{n}|^2 + 2 |\vec{\Omega} \cdot \vec{n}| \right) \tag{6}$$

The weight function is currently defined as

$$w_{g,m} \equiv \frac{1}{\sigma_{t,g}} \tag{7}$$

This choice gives as the self-adjoint angular flux equation (SAAF) [4]. The weight functions were implemented in Yak using the material interface.

## 2.1 Numerical results

For this study we look at a one dimensional problem with two material regions. The left region contains a weak absorber ( $\sigma_t = 0.1$ ), while the right region has a

strong absorber ( $\sigma_t = 10$ ). Each region is 1 cm thick and the problem is surrounded by vacuum. A constant source of  $1 \text{ s}^{-1}$  is added in both regions. We compare the unweighted LS to the weighted LS and the SAAF implementation in Rattlesnake. Additionally, we use two different boundary conditions for the LS formulation as specified in eqs. (5) to (6). We use a  $S_8$  Gauss quadrature for all calculation.

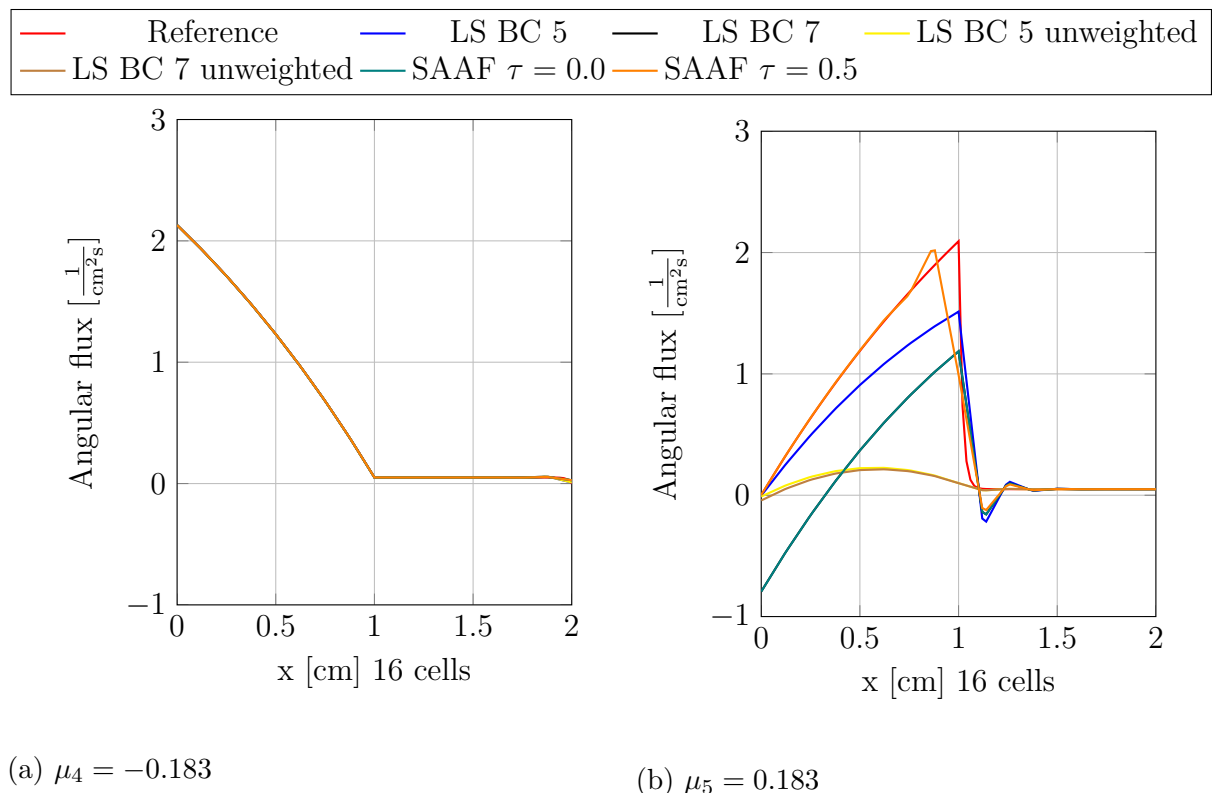


Figure 1: Angular flux for the lowest magnitude angles in positive and negative direction for the two absorber problem

We can see in fig. 1, that the for fluxes from the thick into the thin region all schemes perform well. However, for the other direction, we see large deviations between the schemes. First we note, that the unweighted LS complete misses the material interface. We also see that SAAF without void treatment and LS with the BC eq. (5) have the same result, which proofs our finding, that SAAF is a special case of LS. We also note, that both this schemes have a large negative value at the left boundary. This can be significantly improved by choosing the other boundary condition option (eq. (6)). SAAF with void treatment shows the best results, however

it misses the sharp material interface. All weighted LS and SAAF schemes have a negativity behind the material interface.

## 2.2 Angular dependent weight function

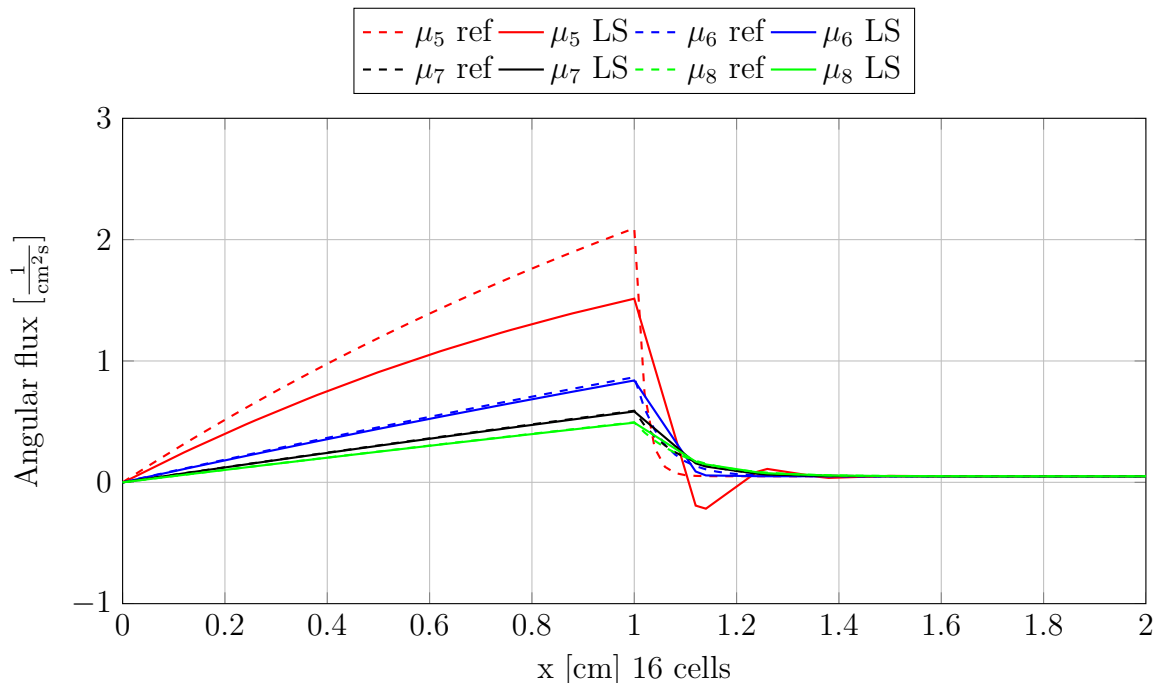


Figure 2: Comparison between different positive angles for the weighted LS with BC 5 and the reference solution

If we take a more detailed look at the all positive angle for the weighted LS (fig. 2), we can see, that more perpendicular angles agree better with the reference solution. The difference grows the lower the magnitude of the angle becomes.

Therefore, if we move further with the idea of weighting with the mean free path, we note, that the weight function must be angular dependent. A simple approach in one dimension is be

$$w_{m,g} \equiv \frac{1}{\mu_m \sigma_{t,g}} \quad (8)$$

Calculations showed, that this modification has no effect on the result. The problem might be, that a full row of the matrix is weighted by the same  $\mu$ . It is lost by matrix

normalization performed by many solvers. Further studies using a transport solution of the auxiliary problem

$$-\vec{\Omega}_m \cdot \vec{\nabla} w_{m,g} + \sigma_{t,g} w_{m,g} = 1 \quad (9)$$

with

$$w_{m,g}(\vec{r}_b) = 0, \quad \forall \vec{r}_b \in \partial\mathcal{D}, \vec{\Omega}_m \cdot \hat{n}(\vec{r}_b) < 0. \quad (10)$$

did not give the desired results.

### 3 Nonlinear Diffusion Acceleration in Voids

We derive our low order diffusion equation from the first order transport equation as shown by Peterson [7]. This results in a inconsistent, but conservative form of the NDA. Integrating over all angles gives us the zeroth moment equation

$$\vec{\nabla} \cdot \vec{J}_g^{k+1} + \sigma_{t,g} \phi_g^{k+1} = \sum_{g'=1}^G \sigma_{0,g' \rightarrow g} \phi_{g'}^{k+1} + q_{0,g} \quad (11)$$

Multiplying by a test function and integrating over the domain gives the according weak form

$$\left( \vec{\nabla} \cdot \vec{J}_g^{k+1}, \phi^* \right)_{\mathcal{D}} + \left( \sigma_a \phi_g^{k+1}, \phi^* \right)_{\mathcal{D}} = \left( \sum_{\substack{g'=1 \\ g' \neq g}}^G \sigma_{0,g' \rightarrow g} \phi_{g'}^{k+1}, \phi^* \right)_{\mathcal{D}} + (q_{0,g}, \phi^*)_{\mathcal{D}} \quad (12)$$

Applying integration by parts on the current term gives

$$\begin{aligned} & \left( \vec{J}_g^{k+1}, \vec{\nabla} \phi^* \right)_{\mathcal{D}} + \left\langle \vec{n} \cdot \vec{J}_g, \phi^* \right\rangle_{\partial\mathcal{D}} + \left( \sigma_a \phi_g^{k+1}, \phi^* \right)_{\mathcal{D}} \\ &= \left( \sum_{\substack{g'=1 \\ g' \neq g}}^G \sigma_{0,g' \rightarrow g} \phi_{g'}^{k+1}, \phi^* \right)_{\mathcal{D}} + (q_{0,g}, \phi^*)_{\mathcal{D}} \end{aligned} \quad (13)$$

To close eq. (13), we consider the first moment equation

$$\sum_{m=1}^M \omega_m \vec{\Omega}_m \vec{\Omega}_m \cdot \vec{\nabla} \psi_{m,g} + \sigma_{t,g} \vec{J}_g = \sum_{g'=1}^G \sigma_{1,g' \rightarrow g} \vec{J}_{g'} + q_{1,g} \quad (14)$$

with gives the current

$$\vec{J}_g = -\frac{1}{\sigma_{\text{tr},g}} \sum_{m=1}^M \omega_m \vec{\Omega}_m \vec{\Omega}_m \cdot \vec{\nabla} \psi_{m,g} + \frac{1}{\sigma_{\text{tr},g}} \sum_{\substack{g'=1 \\ g' \neq g}}^G \sigma_{1,g' \rightarrow g} \vec{J}_{g'} + \frac{1}{\sigma_{\text{tr},g}} q_{1,g} \quad (15)$$

with

$$\sigma_{\text{tr},g} \equiv \sigma_{\text{t},g} - \sigma_{1,g} \quad (16)$$

We use eq. (15) to construct an additive correction to Fick's law

$$\begin{aligned} \vec{J}_g^{k+1} &= -D_g \vec{\nabla} \phi_g^{k+1} + D_g \vec{\nabla} \phi_g^{k+\frac{1}{2}} \\ &\quad - \frac{1}{\sigma_{\text{tr},g}} \sum_{m=1}^M \omega_m \vec{\Omega}_m \vec{\Omega}_m \cdot \vec{\nabla} \psi_{m,g}^{k+\frac{1}{2}} + \frac{1}{\sigma_{\text{tr},g}} \sum_{\substack{g'=1 \\ g' \neq g}}^G \sigma_{1,g' \rightarrow g} \vec{J}_{g'}^{k+\frac{1}{2}} + \frac{1}{\sigma_{\text{tr},g}} q_{1,g} \\ &= -D_g \vec{\nabla} \phi_g^{k+1} - \vec{D}_g^{k+\frac{1}{2}} \phi_g^{k+1} \end{aligned} \quad (17)$$

where

$$\begin{aligned} \vec{D}_g^{k+\frac{1}{2}} &\equiv \frac{1}{\phi_g^{k+\frac{1}{2}}} \left( \frac{1}{\sigma_{\text{tr},g}} \sum_{m=1}^M \omega_m \vec{\Omega}_m \vec{\Omega}_m \cdot \vec{\nabla} \psi_{m,g}^{k+\frac{1}{2}} - D_g \vec{\nabla} \phi_g^{k+\frac{1}{2}} \right. \\ &\quad \left. + \frac{1}{\sigma_{\text{tr},g}} \sum_{\substack{g'=1 \\ g' \neq g}}^G \sigma_{1,g' \rightarrow g} \vec{J}_{g'} + \frac{1}{\sigma_{\text{tr},g}} q_{1,g} \right) \end{aligned} \quad (18)$$

and the diffusion coefficient is defined as

$$D_g \equiv \frac{1}{3\sigma_{\text{tr},g}} \quad (19)$$

Substituting eq. (17) in eq. (13) gives

$$\begin{aligned} & \left( D \vec{\nabla} \phi_g^{k+1}, \vec{\nabla} \phi^* \right)_{\mathcal{D}} + \left( \vec{D}_g \phi_g^{k+1}, \vec{\nabla} \phi^* \right) + \left\langle \vec{n} \cdot \vec{J}_g, \phi^* \right\rangle_{\partial \mathcal{D}} + \left( \sigma_a \phi_g^{k+1}, \phi^* \right)_{\mathcal{D}} \\ &= \left( \sum_{\substack{g'=1 \\ g' \neq g}}^G \sigma_{0,g' \rightarrow g} \phi_{g'}^{k+1}, \phi^* \right)_{\mathcal{D}} + (q_{0,g}, \phi^*)_{\mathcal{D}} \end{aligned} \quad (20)$$

### 3.1 Nonlocal Diffusion Coefficient

The classical formulation of the diffusion coefficient (eq. (19)) does not hold in voids. However, if we look at eq. (20) we see that in the case of convergence, the diffusion terms cancel. Therefore, the diffusion coefficient is a free parameter in the NDA calculation. We choose to use a nonlocal definition of a diffusion coefficient. The derivation can be found in [3, 5, 6, 8]. This approach gives as a diffusion coefficient tensor  $\underline{\underline{D}}_g$  with

$$D_{ij,g} \equiv \int_{4\pi} (\vec{\Omega} \cdot \vec{e}_i) (\vec{\Omega} \cdot \vec{e}_j) f_g(\vec{\Omega}) d\vec{\Omega} \quad (21)$$

In this equation  $f_g(\Omega)$  is the transport solution to

$$\vec{\Omega} \cdot \vec{\nabla} f_g + \sigma_{t,g} f_g = 1 \quad (22)$$

This equation can be easily solved using any technique to solve a transport equation. Note that we do not have a scattering source, therefore one solve is sufficient. The result is well defined in finite voids. Only in the case of an infinite void along a direction this diffusion coefficient becomes undefined.

### 3.2 Drift Vector in Voids

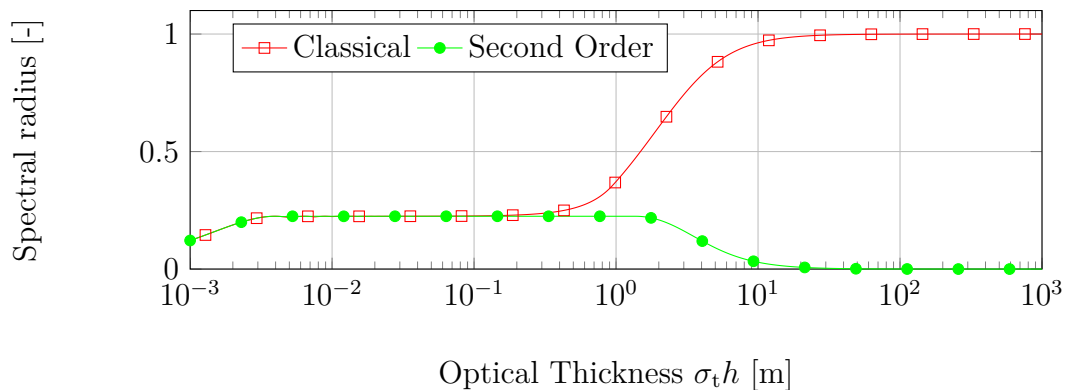


Figure 3: Spectral radius for  $c = 1$  as function of the cell thickness for the two formulation of the current in a homogeneous material. The decrease for thin cells comes from numerical issues.

Additionally to the diffusion coefficient, the current in the drift vector formulation eq. (18) becomes singular in voids. We use eq. (15)

$$\begin{aligned}\vec{J}_g &= \lim_{-\sigma_{\text{tr},g} \rightarrow 0} \left[ \frac{1}{\sigma_{\text{tr},g}} \sum_{m=1}^M \omega_m \vec{\Omega}_m \vec{\Omega}_m \cdot \vec{\nabla} \psi_{m,g} + \frac{1}{\sigma_{\text{tr},g}} \sum_{\substack{g'=1 \\ g' \neq g}}^G \sigma_{1,g' \rightarrow g} \vec{J}_{g'} + \frac{1}{\sigma_{\text{tr},g}} q_{1,g} \right] \\ &= \sum_{m=1}^M \omega_m \vec{\Omega}_m \cdot \psi_{m,g}\end{aligned}\tag{23}$$

to show that we get the classical formulation of the current. A Fourier analysis (fig. 3) showed, that this formulation shows convergence issues for optical thick cells. The reason for this is that the classical formulation results in a bad finite difference discretization scheme, skipping the center node. For high frequencies, this cannot resolve the first derivative correctly. However, voids are optically thin, therefore we choose to use

$$\vec{J}_g^{k+\frac{1}{2}} \equiv \begin{cases} \frac{1}{\sigma_{\text{tr},g}} \sum_{m=1}^M \omega_m \vec{\Omega}_m \vec{\Omega}_m \cdot \vec{\nabla} \psi_{m,g}^{k+\frac{1}{2}} + \frac{1}{\sigma_{\text{tr},g}} \sum_{\substack{g'=1 \\ g' \neq g}}^G \sigma_{1,g' \rightarrow g} \vec{J}_{g'} + \frac{1}{\sigma_{\text{tr},g}} q_{1,g}, & \sigma_{\text{tr},g} h \geq \tau \\ \sum_{m=1}^M \omega_m \vec{\Omega}_m \cdot \psi_{m,g}^{k+\frac{1}{2}}, & \sigma_{\text{tr},g} h < \tau \end{cases}\tag{24}$$

where  $\tau$  is a threshold value for the optical thickness to switch between the two formulations. Looking at fig. 3 we choose  $\tau = 0.01$ .

### 3.3 Implementation

The calculation routine for the nonlocal diffusion tensors was already implemented in Yak. However, it was limited to first order NDA calculation. The routine was moved to make it available for all diffusion calculations including second order NDA calculations. Transfers were added to the transport system to make the new diffusion coefficient available for the calculation of the drift vectors. The LS drift vector calculation was changed to use eq. (24). The SAAF drift vector implementation already worked for void using the SAAF void treatment [9] and the nonlocal diffusion coefficients.

### 3.4 Numerical Results

Our test case is Reed's problem, a well know test problem containing a void region and a highly diffusive region. It reassembles a radial, one dimensional pin in a LWR

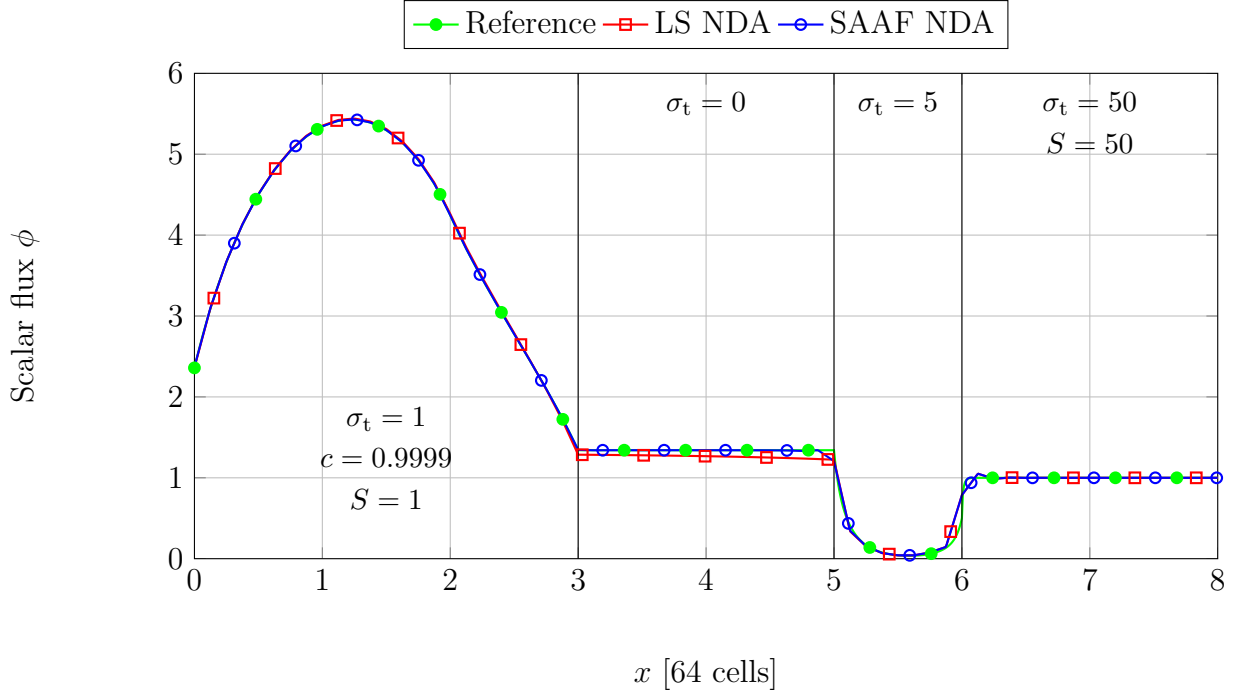


Figure 4: Solution to Reed’s problem with SAAF and LS NDA and comparison to a reference solution. Also included are solutions of the transport update systems.

reactor. The calculations uses the nonlocal diffusion coefficient and the modifications to the LS drift vector to handle the void. The results are shown in fig. 4. The NDA results agree well with the reference transport solutions, however in the void the LS solutions show a non constant slope. This can be seen in more detail in fig. 5. In this figure we also can see, that the SAAF solutions show some small oscillations at the void’s boundary. The according drift vectors are shown in fig. 7. This however, shows that the LS drift vectors are constant and have even a better shape than the SAAF drift vectors. The reason for the slope of the LS solution is not yet understood.

## 4 Conclusion

The implementation of the least-squares transport equation in Rattlesnake was updated with weight functions. These improve the overall performance of the scheme especially at material interfaces from weak to strong absorber and in voids. Attempts



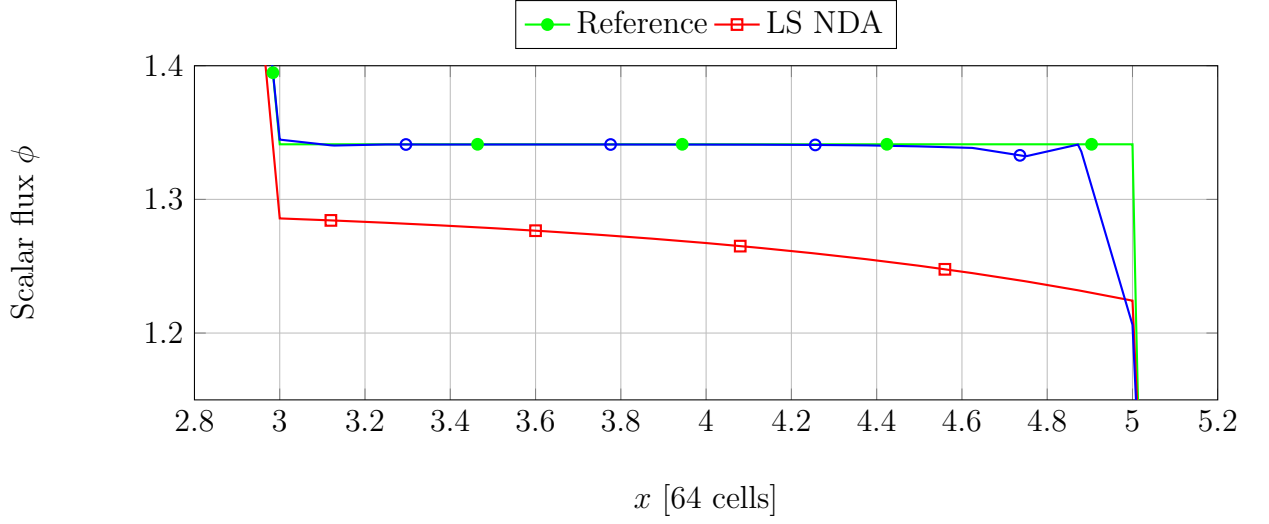


Figure 5: Detailed view on void region for the solution of Reed’s problem with the non constant shape of the LS solution and small oscillations in the SAAF solution

to use an angular dependent weight function to further improve the LS scheme did not yet give good results.

The NDA implementation for the second order schemes in Rattlesnake was changed to handle voids. The nonlocal diffusion coefficient and the modifications to the LS drift vector work. However, the LS result show minor problems with a non constant shape in the void. These need further investigations.

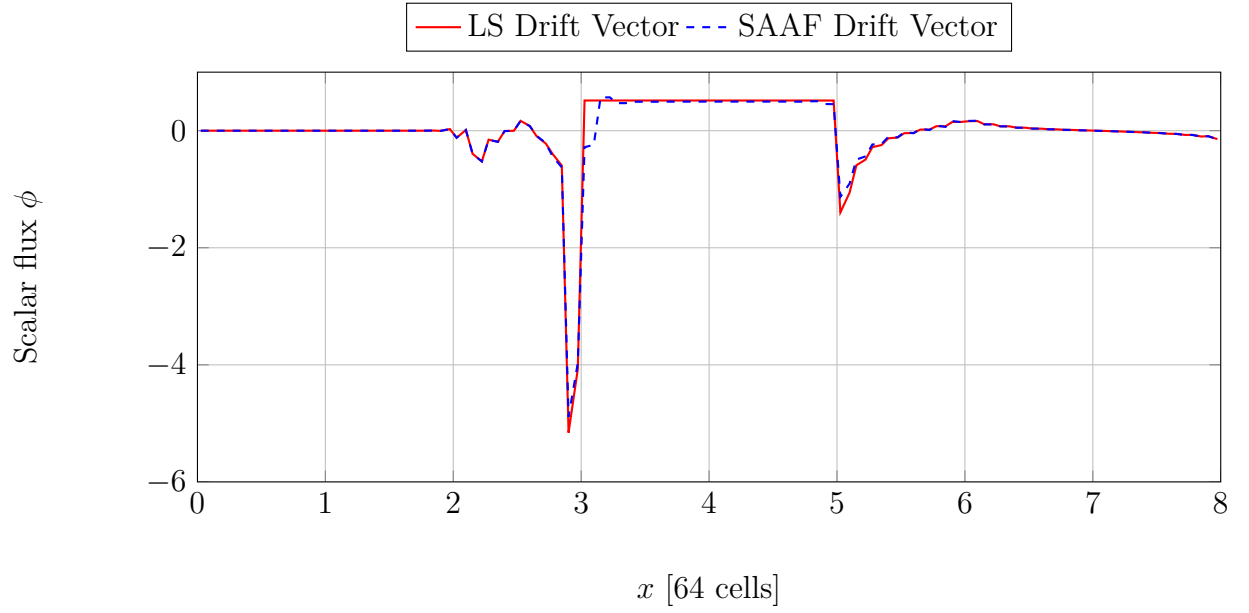


Figure 6: Drift vectors for the LS and SAAF NDA calculation to Reed's problem

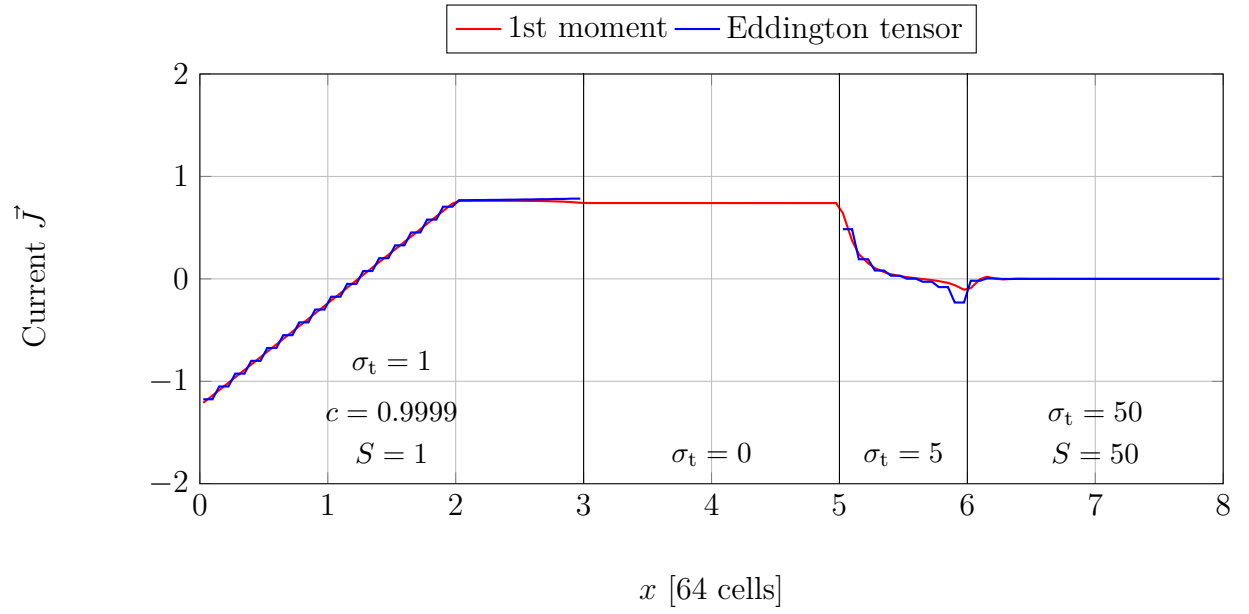


Figure 7: Drift vectors for the LS and SAAF NDA calculation to Reed's problem

## References

- [1] Derek Gaston, Chris Newman, Glen Hansen, and Damien Lebrun-Grandie. MOOSE: A parallel computational framework for coupled systems of nonlinear equations. *Nuclear Engineering and Design*, 239(10):1768–1778, 2009.
- [2] Jon Hansen, Jacob Peterson, Jim Morel, Jean Ragusa, and Yaqi Wang. A least-squares transport equation compatible with voids. *Journal of Computational and Theoretical Transport*, 43(1-7):374–401, 2014.
- [3] Edward W. Larsen and Travis J. Trahan. 2-D anisotropic diffusion in optically thin channels. *Transactions of the American Nuclear Society*, 101:387–389, 2009.
- [4] J. E. Morel and J. M. McGhee. A self-adjoint angular flux equation. *Nuclear science and engineering*, 132(3):312–325, 1999.
- [5] Jim E. Morel. A Non-Local Diffusion Theory. Research Report LA-UR-07-5257, Los Alamos National Laboratory, NM, 2007.
- [6] Jim E. Morel, J. S. Warsa, and K. G. Budge. Alternative Generation of Non-Local Diffusion Tensors. Research Memo, Texas A&M University, 2010.
- [7] Jacob R. Peterson, Hans R. Hammer, Jim E. Morel, Jean C. Ragusa, and Yaqi Wang. Conservative nonlinear diffusion acceleration applied to the unweighted least-squares transport equation in MOOSE. In *Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method*, pages 19–23, 2015.
- [8] Travis J. Trahan and Edward W. Larsen. 3-D Anisotropic Neutron Diffusion in Optically Thick Media with Optically Thin Channels. In *Proc. Intl. Conf. on Math. and Comput. Methods Applied to Nucl. Sci. Eng.(M&C 2011), Rio de Janeiro, Brazil, May 8*, volume 12, 2011.
- [9] Yaqi Wang, Hongbin Zhang, and Richard C. Martineau. Diffusion Acceleration Schemes for Self-Adjoint Angular Flux Formulation with a Void Treatment. *Nuclear Science and Engineering*, 176(2):201–225, 2014.

August, 2016

# Implementation of Weighted Delta-Tracking with Scattering in the Serpent 2 Monte Carlo Code

J. S. Rehak<sup>a,b,\*</sup>  
L. M. Kerby<sup>a,c</sup>  
M. D. Dehart<sup>a</sup>

---

\* E-mail: [jsrehak@berkeley.edu](mailto:jsrehak@berkeley.edu)

<sup>a</sup> Idaho National Lab

<sup>b</sup> University of California, Berkeley

<sup>c</sup> Idaho State University

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Theory</b>	<b>1</b>
2.1	Ray tracing . . . . .	1
2.2	Rejection Sampling . . . . .	2
2.2.1	Woodcock Delta-tracking . . . . .	2
2.2.2	Nonuniform Density Distributions . . . . .	3
2.3	Implicit Events . . . . .	4
2.4	Weighted delta-tracking . . . . .	4
<b>3</b>	<b>Implementation</b>	<b>5</b>
3.1	Weighted delta-tracking scattering . . . . .	6
3.2	Scattering Rejection Sampling . . . . .	6
3.3	Rouletting Routine . . . . .	8
3.4	weighted delta-tracking (WDT) threshold value . . . . .	8
<b>4</b>	<b>Test Case and Results</b>	<b>8</b>
4.1	Performance metrics . . . . .	8
4.2	Homogenized TREAT Fuel Pin . . . . .	9
4.3	Results . . . . .	10
4.4	Boiling Water Reactor (BWR) . . . . .	10
<b>5</b>	<b>Discussion</b>	<b>19</b>
<b>6</b>	<b>Future Work</b>	<b>19</b>
<b>7</b>	<b>Conclusion</b>	<b>19</b>
<b>A</b>	<b>Data Tables for TREAT homogenous fuel pin.</b>	<b>21</b>

## List of Figures

1	Weighted delta-tracking with scattering rejection sampling. . . . .	7
2	Surface tracking, WDT and delta-tracking threshold values. . . . .	8
3	FOM of the infinite flux integral tally by energy group. . . . .	11
4	FOM of the infinite total cross-section by energy group. . . . .	12
5	FOM of the infinite capture cross-section by energy group. . . . .	13
6	FOM of the infinite absorption cross-section by energy group. . . . .	14
7	FOM of the infinite $P_0$ scattering cross-section by energy group. . . . .	15
8	FOM of the infinite $P_1$ scattering cross-section by energy group. . . . .	16
9	FOM of the infinite $P_2$ scattering cross-section by energy group. . . . .	17
10	FOM for the BWR for different WDT threshold values. . . . .	18
11	Average FOM ratio for all cross-sections. . . . .	19

## List of Tables

1	Weighted delta-tracking roulette parameters. . . . .	8
2	Homogenized TREAT fuel pin nuclide composition. . . . .	9
3	Neutron energy group structure for homogenized TREAT fuel pin . . . . .	9
4	Runtime for the TREAT homogenous fuel pin. . . . .	10
5	Criticality parameters for the TREAT homogenous fuel pin. . . . .	10
6	BWR parameters. . . . .	18

7	Infinite group flux for the TREAT homogenous fuel pin. . . . .	21
8	Infinite total reaction cross-section for the TREAT homogenous fuel pin. . . . .	21
9	Infinite capture reaction cross-section for the TREAT homogenous fuel pin. . . . .	22
10	Infinite absorption reaction cross-section for the TREAT homogenous fuel pin. . . . .	22
11	Infinite $P_0$ scattering cross-section for the TREAT homogenous fuel pin. . . . .	22
12	Infinite $P_1$ scattering cross-section for the TREAT homogenous fuel pin. . . . .	23
13	Infinite $P_2$ scattering cross-section for the TREAT homogenous fuel pin. . . . .	23

# 1 Introduction

Simulating the movement of particles is a problem of large and immediate interest. The applications of these simulations are wide-ranging, from medical treatments to analyzing the atmosphere of extra-solar planets. The movement and interaction of particles through a medium can be approached as a statistical problem, doing so allows for simulation using random number generators. Monte Carlo codes and algorithms sample random numbers repeatedly to approximate the desired physical process. One application that has benefitted from the use of Monte Carlo simulations is the design and analysis of nuclear reactors.

Nuclear reactors utilize a sustained chain reaction of the fission heavy elements such as Uranium and Plutonium. This process controlled and maintained by a population of neutrons that cause, and are created in, the chain reaction. The population of neutrons, or flux, can be quite large in most reactor applications. This makes explicitly simulating the position and movement of the neutrons very computationally expensive. The problem is compounded by the complicated material properties and physical layout of modern nuclear reactors. A number of improvements to Monte Carlo algorithms have been developed to improve the computational efficiency of these simulations. Two broad categories that will be discussed in this paper are rejection sampling and implicit events.

A common rejection sampling technique, known as Woodcock delta-tracking, was introduced [7] to address the complicated geometric layout of nuclear reactors. This method is widely used but has a number of drawbacks that make it inefficient in specific cases, such as in the presence of strong absorbers. Morgan and Kotlyar [5] introduced a weighted delta-tracking (WDT) routine to address these issues, replacing the rejection sampling of absorption events with an implicit event.

In this report, we examine the theoretical basis for rejection sampling and implicit events, and how they relate to WDT. We then discuss implementation of a routine that extends the WDT routine to include scattering events. The routine is added to Serpent, a continuous-energy Monte Carlo Reactor physics burnup code developed at the VTT Technical Research Centre of Finland. We examined two test cases, including a homogenous fuel pin from the Idaho National Lab (INL) Transient Reactor Test Facility (TREAT) reactor used to generate cross-sections for deterministic methods.

## 2 Theory

### 2.1 Ray tracing

The basic theory for simulating neutron transport using Monte Carlo is ray tracing. This method explicitly tracks a particle between interaction points and different material boundaries, and forms the basis for the other methods discussed in this paper.

To develop this method, we need to first understand the distance neutrons will travel between interactions, the track length. Neutrons incident on a medium have a total interaction probability  $P$  in a distance traveled  $s$  that is characterized by the material's total-cross section  $\Sigma_t(\mathbf{r})$ :

$$\frac{dP}{ds} = \Sigma_t(\mathbf{r}) \quad (1)$$

The medium therefore attenuates an incident neutron flux  $\phi_0$  as a function of distance:

$$\phi(s) = \phi_0 e^{-s\Sigma_t(\mathbf{r})} \quad (2)$$

The probability that a neutron has its first interaction in differential distance  $ds$  after traveling a distance  $s$  is found by dividing the flux at that position  $\phi(s)$  by the total flux:

$$f(s)ds = \frac{\phi(s)}{\int_0^\infty \phi(s)ds} ds \quad (3)$$

The basis of Eq. (3) is clarified by an example: if the flux after a distance  $s$  is half the total flux, then half of the neutrons have undergone an interaction. Using Eqs. (2) and (3), and assuming that the total cross-section is constant over  $\mathbf{r}$ : [4]:

$$f(s)ds = \Sigma_t e^{-s\Sigma_t} ds \quad (4)$$

This is the probability density function (PDF) of the distance traveled before the first collision,  $s$ . As the distance a neutron travels increases, the value of  $f(s)$  decreases; it is less probable that a neutron will travel further without collision. As the PDF gives a probability at a point  $s$  that a neutron undergoes a collision exactly there, integrating over the distance  $s$  provides us the probability that a neutron undergoes a collision between 0 and  $s$ . This is the cumulative density function (CDF):

$$F(s) = \int_0^s f(s') ds' = 1 - e^{-s\Sigma_t} \quad (5)$$

As expected, increasing  $s$  causes the probability of any interaction,  $F(s)$ , to approach unity. As the probability ranges from zero to unity, it can be replaced by a uniformly distributed random variable  $\xi \in [0, 1)$ . Inverting the function  $F(s)$  allows us to then sample the path length  $s$  based on this random variable.

$$s(\xi) = F^{-1}(\xi) = -\frac{1}{\Sigma_t} \ln(\xi) \quad (6)$$

After sampling the path length of the neutron, its position is updated, based on its original position and direction.

The sampled path length is a function of the material region through which the neutron is propagating, as it determines  $\Sigma_t$ . If the neutron reaches the boundary between two material regions, a new path length must be sampled using the material properties of the region it is entering. This is the process of ray tracing. Each time a path length is sampled, the distance to the nearest boundary in the direction of motion is determined, and the neutron is moved to the boundary and path length is resampled if appropriate. This can become computationally expensive in complicated geometries and when the probability of crossing boundaries with each sample path length is high.

## 2.2 Rejection Sampling

As discussed in Section 2.1, the value of  $\Sigma_t$  is a piece-wise discontinuous function that varies arbitrarily with position and the geometry of the problem [3]. Neutrons must stop at boundaries to sample a new path length in a new material region. This simplifies the problem by ignoring the complicated nature of  $\Sigma_t(\mathbf{r})$  by only sampling in regions where it is a constant value. To avoid the computational inefficiency that arises in geometrically complicated regions, a rejection sampling technique known as Woodcock delta-tracking was developed [7].

### 2.2.1 Woodcock Delta-tracking

Woodcock delta-tracking introduces the concept of the majorant cross-section, chosen to be the maximum of all material total cross-sections in the region of interest.

$$\Sigma_{\text{maj}} \equiv \max_{\mathbf{r} \in \mathbb{V}} \{\Sigma_t(\mathbf{r})\} \quad (7)$$

Where  $\mathbb{V}$  is the volume of interest. The majorant cross-section can also be represented as the summation of the total cross-section and a delta cross-section:

$$\Sigma_{\text{maj}} = \Sigma_\delta(\mathbf{r}) + \Sigma_t(\mathbf{r}) \quad (8)$$

Following from the definition of  $\Sigma_{\text{maj}}$  in Eq. (7), the function  $\Sigma_\delta(\mathbf{r})$  is chosen such that  $\Sigma_{\text{maj}}$  is constant for the entire region of interest. At the position  $\mathbf{r}$  where the maximum value of  $\Sigma_t(\mathbf{r})$  occurs, the delta cross-section is zero.

Following the same derivation in Section 2.1, the PDF of the first collision occurring after  $s$  using the majorant cross-section is given by:

$$f_{\text{maj}}(s) = \Sigma_{\text{maj}} e^{-\Sigma_{\text{maj}} s} \quad (9)$$

$$= (\Sigma_\delta(\mathbf{r}) + \Sigma_t(\mathbf{r})) e^{-\Sigma_{\text{maj}} s} \quad (10)$$



This formulation of the artificial majorant cross-section PDF can be used to sample the real physical collision PDF through the use of rejection sampling without limiting ourselves to regions in which  $\Sigma_t$  is a constant. As described by Lux and Koblinger [4], rejection sampling requires a PDF of interest,  $f(x)$ , and a second PDF  $g(x)$  for which:

$$f(x) \leq Mg(x), \forall x \quad (11)$$

Sampling from  $Mg(x)$  and accepting these samples with probability:

$$P = \frac{f(x)}{Mg(x)} \quad (12)$$

replicates sampling directly from  $f(x)$ .

The PDF of interest is the real collision PDF, Eq. (4); so a PDF  $g(x)$  that satisfies the inequality of Eq. (11) must be chosen. In Woodcock delta tracking, the second PDF  $g(x)$  is chosen to be the majorant PDF, Eq. (9). These functions are both maximized at  $x = 0$ , where the inequality of Eq. (11) is satisfied by setting  $M = 1$ :

$$\Sigma_t(\mathbf{r}) \leq \Sigma_{\text{maj}}(\mathbf{r}), \forall \mathbf{r} \in \mathbb{V} \quad (13)$$

Therefore, by sampling path length using the constant majorant cross-section:

$$s_{\text{maj}} = -\frac{1}{\Sigma_{\text{maj}}} \ln(\xi) \quad (14)$$

The path length for the more complicated real collision probability can be retrieved by accepting these samples with the probability given in Eq. (12) by substituting portion of the PDF for real collisions from Eq. (9):

$$P_{\text{real}} = \frac{f(x)}{Mg(x)} = \frac{\Sigma_t(\mathbf{r})e^{-\Sigma_{\text{maj}}s}}{\Sigma_{\text{maj}}e^{-\Sigma_{\text{maj}}s}} \quad (15)$$

$$= \frac{\Sigma_t(\mathbf{r})}{\Sigma_{\text{maj}}} \quad (16)$$

This clarifies the interpretation of Eq. (9) as the sum of two mutually exclusive PDFs: one representing real physical collisions, one representing virtual collisions that should be rejected by rejection sampling.

$$\begin{aligned} f_{\text{maj}}(s) &= \Sigma_{\text{maj}}e^{-\Sigma_{\text{maj}}s} \\ &= (\Sigma_{\delta}(\mathbf{r}) + \Sigma_t(\mathbf{r}))e^{-\Sigma_{\text{maj}}s} \\ &= f_{\text{virt}}(s) + f_{\text{real}}(s) \end{aligned}$$

Now, the path length can be sampled across multiple material regions of varying  $\Sigma_t$  without explicitly stopping the neutron at a given boundary. This method can become computationally inefficient in regions where the total cross-section is much less than the majorant cross-section, leading to oversampling of virtual collisions. This is seen in geometries that include localized absorbers, such as control rods. Another downside is that the track-length estimator (TLE) for flux cannot be used. The TLE requires calculating the track-lengths within a particular material cell, and therefore does not work when the neutron path length can cross one or more material boundaries. The collision flux estimator (CFE) can be used in its place, but often results in inferior statistics as not every track length sampled ends in a collision [3].

### 2.2.2 Nonuniform Density Distributions

Rejection sampling can also be used when the total cross-section is not constant within a material region. As discussed by Leppänen [3], Serpent 2 conducts a rejection sampling routine similar to Woodcock delta-tracking in these regions. Instead of sampling from a majorant cross-section across multiple materials, a maximum cross-section for the single material region is used. Unlike delta-tracking, this requires stopping the neutron at boundaries and resampling path lengths.

## 2.3 Implicit Events

At each interaction point ending in a collision, the type of collision is determined. Depending on the material, this can result in scattering, capture, fission, or other reactions. Absorption reactions result in the loss of the particle, which diminishes overall statistics. It would be beneficial for the particle to continue colliding or generating track lengths to improve the statistics of the CFE or TLE. A common method to accomplish this is the use of implicit capture, in which the capture collision is replaced by a weight reduction at all collisions.

Proper replacement of a statistical process by introducing a weight reduction is determined by the expected value of the random process [4]. The expected value of a random variable  $x$  that can take values  $x_1 \dots x_n$  with probabilities  $p_1 \dots p_n$  is given by:

$$E[x] = x_1 p_1 + x_2 p_2 \dots + x_n p_n \quad (17)$$

An incident neutron with weight  $w_i$  can scatter, resulting in no weight change, or be absorbed, resulting in a weight of zero. Therefore, the expected value of the final weight  $w_f$  is:

$$\begin{aligned} E[w_f] &= w_{f,\text{scattering}} P_{\text{scattering}} + w_{f,\text{absorption}} P_{\text{absorption}} \\ &= w_i P_{\text{scattering}} \end{aligned}$$

By replacing capture events with a weight reduction, every collision is assumed to be a scattering event. This allows the neutron to survive longer, sampling more path-lengths and causing more collisions; this process is also referred to as survival biasing. The capture score is determined the same way, using the expected value:

$$\begin{aligned} S_{\text{capture}} &= E[w_i - w_f] \\ &= E[w_i] - E[w_f] \\ &= w_i - w_i P_{\text{scattering}} \\ &= w_i (1 - P_{\text{scattering}}) \end{aligned}$$

Without capture events, the loss of neutrons is entirely reliant on leakage from the problem or fission events. Tracking low-weight neutrons can become computationally inefficient, especially in infinite geometries. The general algorithm is shown in Algorithm 1. Following a collision, if the weight of the colliding

---

### Algorithm 1 Rouletteing Routine

---

```

1: if weight < weight threshold then
2:    $\xi \leftarrow$  random number  $\in [0, 1)$ 
3:   if  $\xi <$  roulette probability then
4:     Kill particle
5:   else
6:      $w_f \leftarrow w_i / (\text{roulette probability})$ 
7:   end if
8: end if

```

---

particle is below a defined weight threshold, a random number is sampled. If this number is below a defined rouletteing probability, the particle is killed. If the particle survives rouletteing, its weight is increased proportional to the rouletteing probability. By killing low weight particles, or increasing their weight, the inefficiency of tracking low-weight particles can be reduced.

## 2.4 Weighted delta-tracking

Morgan and Kotlyar [5] introduced a method to improve the inefficiencies of Woodcock delta-tracking in the presence of large absorbers. The method, WDT, replaces the rejection sampling of delta-tracking with a weight reduction. This is similar to the process of implicit capture discussed in Section 2.3.

The WDT method samples the particle path length in the same fashion as Woodcock delta-tracking. As described in Section 2.2.1, after each path length sampled, the delta-tracking method accepts the collision

as real with the probability shown in Eq. 12. The WDT method bypasses this decision tree by accepting all collisions as real with a subsequent reduction in weight. As discussed in Section 2.3, replacing a statistical event requires calculation of the expected value. In this case, the two events are a real collision and a virtual collision.

$$E[w_f] = w_{f,\text{real}}P_{\text{real}} + w_{f,\text{virt}}P_{\text{virt}} \quad (18)$$

Morgan and Kotlyar examine a 1D test case with absorption. As an absorption event removes the particle,  $w_{f,\text{real}} = 0$ , and a virtual collision leaves the weight unchanged. Inserting the appropriate values into Eq. (18) gives the expected value of the final weight for an absorption event.

$$\begin{aligned} E[w_f] &= w_{f,\text{real}}P_{\text{real}} + w_{f,\text{virt}}P_{\text{virt}} \\ &= 0 + w_iP_{\text{virt}} \\ &= w_i(1 - P_{\text{real}}) \\ &= w_i \left( 1 - \frac{\Sigma_t}{\Sigma_{\text{maj}}} \right) \end{aligned}$$

The particle that is left following the collision continues propagating as if it underwent a virtual collision. In this case, the absorption is then scored using the expectation value of the score.

$$\begin{aligned} S_{\text{absorption}} &= E[w_i - w_f] \\ &= E[w_i] - E[w_f] \\ &= w_i \left( \frac{\Sigma_t}{\Sigma_{\text{maj}}} \right) \end{aligned}$$

This is implemented by Kotlyar and Morgan in a 1D problem and the results are verified with the analytical solution. The authors point out that a rouletting routine should be implemented when this is used, to prevent the tracking of low-weight neutrons.

### 3 Implementation

The Serpent 2 Monte Carlo Code uses a combination of surface tracking, Woodcock delta-tracking, and rejection sampling for non-uniform density distributions. Serpent 2 selects between surface tracking and delta-tracking by examining the ratio of total cross-section to majorant cross-section [2]. In regions where many virtual collisions would occur, the code preferentially switches to surface tracking. This is determined by a constant  $c$  and the inequality in Eq. (19).

$$\frac{\Sigma_t}{\Sigma_{\text{maj}}} > 1 - c \quad (19)$$

By default, the value of  $c$  is 0.1, as this was determined to produce the best improvement in run time [2]. Prior to sampling path length, the code tests this ratio for the current neutron position and determines if surface tracking or delta-tracking should be used. If delta-tracking is used, the code then determines if the collision is virtual or real.

To implement WDT in Serpent 2, the delta-tracking routine was modified to always return a real collision. As described in Section 2.4, the weight change for an absorption or capture event is straightforward. Following sampling of the type of collision, the collision is scored using the real portion of the weight. The neutron weight is then adjusted and a new path length sampled. The WDT method developed by Morgan and Kotlyar does not address scattering collisions, so it must be extended.

### 3.1 Weighted delta-tracking scattering

In a scattering event, the weight of the incident particle does not change. Therefore, application of the expectation value as in Section 2.4 results in an expected value of the final weight equal to the initial weight.

$$\begin{aligned}
E[w_f] &= w_{f,\text{real}}P_{\text{real}} + w_{f,\text{virt}}P_{\text{virt}} \\
&= w_iP_{\text{real}} + w_iP_{\text{virt}} \\
&= w_i(P_{\text{real}} + 1 - P_{\text{real}}) \\
&= w_i
\end{aligned}$$

This does not match what is occurring physically when we take these events to be implicit. The WDT method splits the weight of the colliding particle into a virtual portion and a real portion (dependant on the  $P_{\text{real}}$  from Eq. (15)). The real portion of the weight undergoes the collision; this is the portion that was used to score absorption. The virtual portion of the weight is left with the particle that continues propagating as if no collision had occurred at all. In the event of absorption, the real portion of the weight is attributed to a particle that is then immediately killed, but this is not the case in a scattering event.

Extension of this methodology to scattering requires duplication of the particle at the point of collision. The virtual portion of the weight is carried away by a particle that propagates as if no collision has occurred, and the real portion is carried away by a particle that undergoes scattering. In problems with scattering, this results in a rapid multiplication of neutrons. When implemented into Serpent 2, this multiplication very quickly filled any available neutron buffer in simulations of a Boiling Water Reactor (BWR), ending the simulation.

### 3.2 Scattering Rejection Sampling

As described in Section 3.1, the WDT method quickly resulting in an intractable simulation when applied to scattering. To maintain proper statistics, scattering must take into account the possibility of a virtual collision while using delta-tracking. To achieve this goal, the delta-tracking rejection sampling that had been supplanted by WDT was moved into the scattering subroutine. Therefore, the new routine uses both rejection-sampling and implicit events to account for the possibility of real and virtual collisions. The algorithm is shown in Alg. 2 and a flow chart of the routine is shown in Fig. 1 Note that there are two

---

#### Algorithm 2 Weighted delta-tracking with scattering

---

```

1: Sample path length
2: Sample collision type
3: if collision type == (capture or fission) then
4:   Score capture or fission  $\leftarrow w_i P_{\text{real}}$ 
5:   Score collision  $\leftarrow w_i P_{\text{real}}$ 
6:    $w_f \leftarrow w_i(1 - P_{\text{real}})$ 
7:   Execute virtual collision
8: else
9:   Sample random number  $\xi \in [0, 1)$ 
10:  if  $\xi < P_{\text{real}}$  then ▷ Collision is real
11:    Score scattering  $\leftarrow w_i$ 
12:    Score collision  $\leftarrow w_i$ 
13:    Execute scattering collision
14:  else ▷ Collision is virtual
15:    Execute virtual collision
16:  end if
17: end if

```

---

separate scoring events in each collision subroutine: scoring of the actual collision type for calculating specific reaction rates, and scoring of collision itself used by the collision flux estimator. In addition, scoring the fission reaction also encompasses generation of fission neutrons.

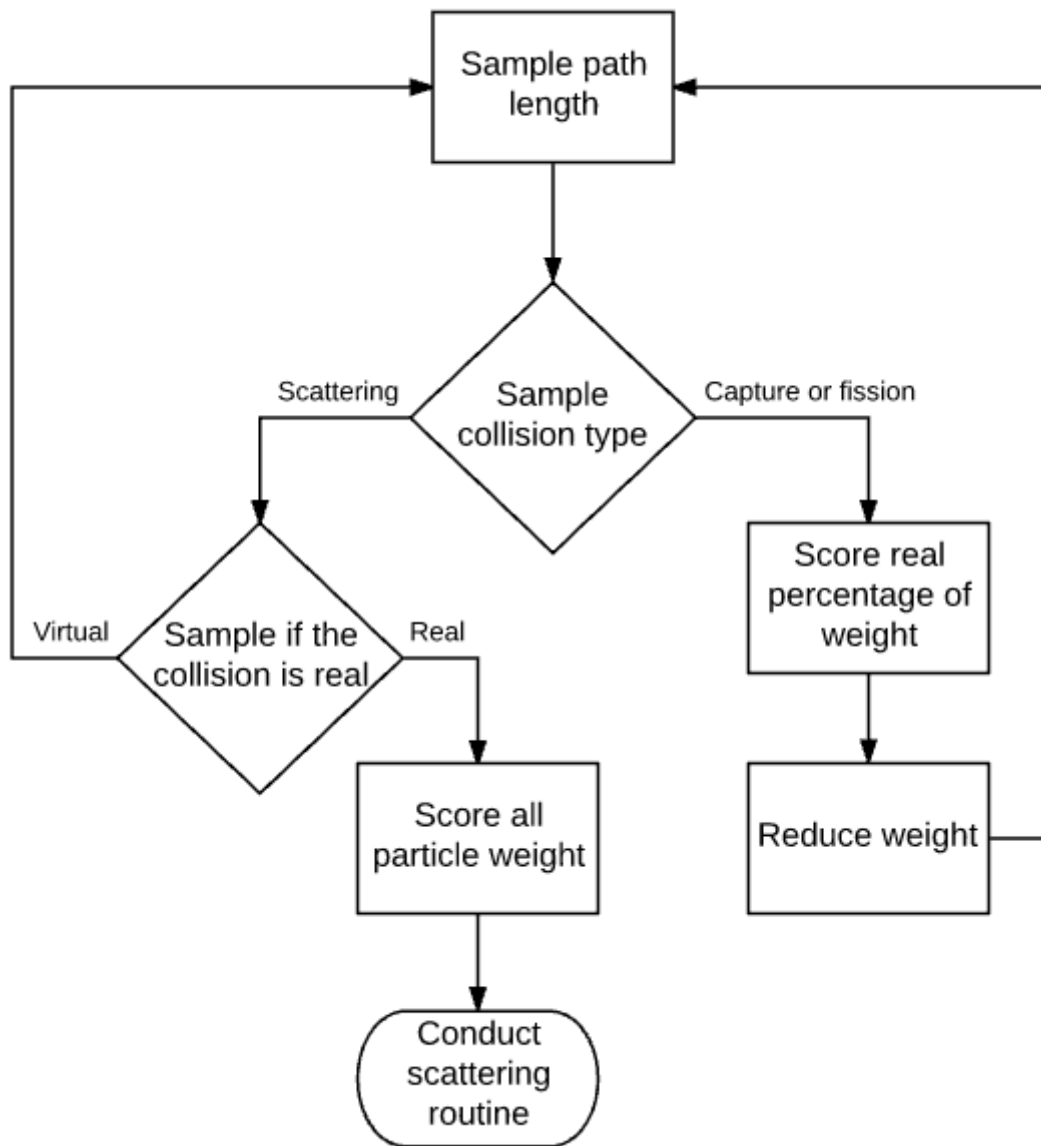


Figure 1: Weighted delta-tracking with scattering rejection sampling.

In the original delta-tracking routine, the rejection sampling takes place prior to collision type sampling, and collision scoring can occur between the two. This is the routine that Serpent 2 used, prior to modification for the above scheme. By moving the rejection sampling after the collision type sampling, the collision score is no longer agnostic of the type of collision that will occur. Therefore, in the implementation of this routine in Serpent 2, the collision scoring was moved after the collision type sampling.

It is also important to note that the addition of WDT does not affect the decision tree that determines if surface tracking is used; surface tracking is still used when the inequality of Eq. (19) is met.

### 3.3 Rouletting Routine

As discussed in Section 2.3, replacing a statistical process with a weight reduction requires introduction of a rouletting routine. This prevents the simulation from tracking low weight neutrons and near infinite loops that may result when there are few neutron removal mechanisms. A rouletting routine was added to Serpent 2 that is called automatically if WDT is used. The algorithm is the same as presented in Alg. 1, and arbitrary values were chosen for the weight cutoff and rouletting probability. These values are summarized in Table 1; tuning of these values to improve performance is a possible direction for future work, as discussed in Section 6.

Parameter	Value
Weight cutoff	0.1
Roulette probability	0.5

Table 1: Weighted delta-tracking roulette parameters.

### 3.4 WDT threshold value

As discussed earlier in this section, Serpent 2 selects between surface tracking and delta-tracking based on a threshold value of  $P_{\text{real}}$ , as shown in Eq. (19). A similar process may provide some benefit when applied to WDT. At high values of  $P_{\text{real}}$ , a majority of the weight of the incoming particle is scored. This leaves the particle that undergoes a virtual collision with a very low weight, relying on the rouletting routine to prevent causing computational inefficiency. An alternate method to avoid the creation of these low weight particles would be establishment of an upper boundary for application of WDT as shown in Figure 2.

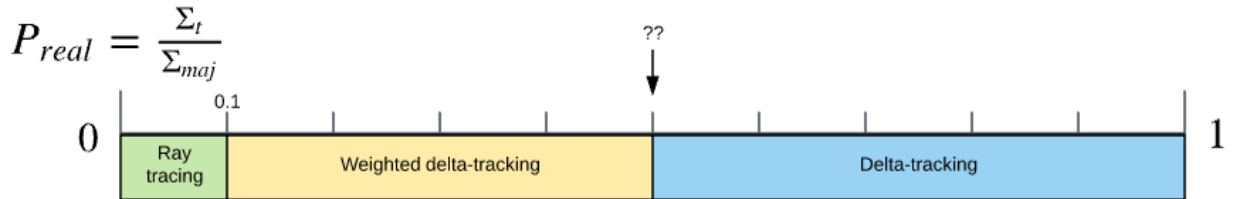


Figure 2: Surface tracking, WDT and delta-tracking threshold values.

## 4 Test Case and Results

### 4.1 Performance metrics

Monte Carlo codes such as Serpent 2 repeat simulations multiple times to calculate the sample mean of a value of interest,  $\hat{x}$ . Generally, the variance of the simulation result  $\sigma^2(\hat{x})$  can be improved through longer computation effort. Therefore, to determine if a method is useful, a figure of merit (FOM) must

be formulated to provide a measure of accuracy per computation time. The standard FOM used by most simulations, described by Lewis and Miller [1], is shown in Eq. (20):

$$\text{FOM} = \frac{1}{\sigma(\hat{x})^2 T} \quad (20)$$

where  $T$  is the runtime of the simulation. A higher FOM indicates higher accuracy per computation time, and therefore a more efficient algorithm. Serpent 2 collects scores for each cycle (or batch of cycles) and outputs the sample mean and standard deviation [6]. This enables comparison of Serpent 2 running with and without WDT to determine the efficiency of the algorithm.

## 4.2 Homogenized TREAT Fuel Pin

Researchers at INL are using Serpent 2 to generate cross-sections for the TREAT Reactor. These cross-sections are used by various deterministic codes that model the neutronics of the reactor. Generation of cross-sections with the current methods low error in a reasonable amount of time requires billions of simulated particles. Improved variance reduction methods, such as WDT, may provide a means to improve the ability of the code to quickly generate cross-sections with low error. The test case chosen to assess the ability of the WDT method to improve cross-section generation is a homogenized TREAT fuel pin. The homogenous fuel pin material composition and the neutron groups used in the simulation are shown in Tables 2 and 3, respectively.

Nuclide	Density ( $10^{24}/\text{cm}^3$ )
$^{10}\text{B}$	$1.4495 \times 10^{-7}$
$^{11}\text{B}$	$5.8343 \times 10^{-7}$
$^{12}\text{C}$	$8.5353 \times 10^{-2}$
$^{16}\text{O}$	$1.8323 \times 10^{-5}$
$^{54}\text{Fe}$	$6.5286 \times 10^{-7}$
$^{56}\text{Fe}$	$1.0239 \times 10^{-5}$
$^{57}\text{Fe}$	$2.3659 \times 10^{-7}$
$^{58}\text{Fe}$	$3.1248 \times 10^{-8}$
$^{235}\text{U}$	$8.6879 \times 10^{-6}$
$^{238}\text{U}$	$6.2967 \times 10^{-7}$

Table 2: Homogenized TREAT fuel pin nuclide composition.

Group	Energy (MeV)
1	$2.00100 \times 10^{-8}$
2	$4.73020 \times 10^{-8}$
3	$7.64970 \times 10^{-8}$
4	$2.09610 \times 10^{-7}$
5	$6.25000 \times 10^{-7}$
6	$8.10003 \times 10^{-6}$
7	$1.32700 \times 10^{-4}$
8	$3.48110 \times 10^{-3}$
9	$1.15620 \times 10^{-1}$
10	$3.32870 \times 10^{-0}$

Table 3: Neutron energy group structure for TREAT fuel pin simulation. The values in the second column indicate the upper limit of the energy bin for that group.

### 4.3 Results

The energy bins were divided into eleven groups, as identified above. The calculation was run with 100 000 source neutrons for 5 inactive and 100 active cycles. The simulation was repeated with identical seeds both with and without using the WDT method. The runtime of both simulations are shown in Table 4, the WDT method causes a slight increase in runtime. Most of the extra processing time is due to sampling the type of collision after every delta-tracking path length, instead of only sampling for real collisions. Table 5 presents the criticality parameters for the simulation, the FOM when using the WDT method falls short or is comparable to running the simulation without WDT.

$T$ (min)	
No WDT	118.883
WDT	121.212

Table 4: Runtime for the TREAT homogenous fuel pin.

Parameters	No WDT			WDT			Ratios	
	$\hat{x}$	$\sigma^2(\hat{x})$	FOM	$\hat{x}$	$\sigma^2(\hat{x})$	FOM	$\sigma^2(\hat{x})$	FOM
Implicit $k_{\text{eff}}$	1.77209	0.000008	$1.382582 \times 10^8$	1.77208	0.000009	$1.018520 \times 10^8$	1.15	0.74
Analog $k_{\text{eff}}$	1.77272	0.000240	$1.460353 \times 10^5$	1.77254	0.000230	$1.559548 \times 10^5$	0.96	1.07

Table 5: Criticality parameters for the TREAT homogenous fuel pin. The ratios column is expressed as the value with WDT divided by the value without WDT.

Fig. 3 shows a plot of the infinite flux and FOM by energy group; data is contained in Table 7 in Appendix A. With the exception of energy group 10, the WDT method returns a larger FOM than Serpent without the method. The second figure shows the ratio of the FOM values, showing that all groups except group ten show an improvement of at least 1.5 times.

Fig. 4 shows the FOM for the infinite total reaction cross-section by energy group. The WDT method returns consistently lower FOM for all but a few groups. In six of the eleven groups, the FOM while using WDT is less than half the non-WDT run.

Figures 5 and 6 show the FOM for the calculated infinite capture and absorption reaction cross-sections by energy group. In both cases, the WDT method shows an improvement in FOM for the mid-range energy groups, as well as the highest energy group. With the exception of the lowest energy group, the largest gains in FOM when using WDT occur when the FOM is depressed, in the mid-energy (4–6) and highest-energy (11) groups.

Scattering cross-section data for  $P_0$ ,  $P_1$  and  $P_2$  are shown in Figs. 7, 8, and 9 respectively. The  $P_0$  scattering cross-section shows the similar pattern to absorption and capture, with a FOM increase in the mid-range energy groups (5–7) and two higher energy groups (9 and 11). WDT also increases FOM in some cases for  $P_1$  and  $P_2$  scattering. In  $P_1$  scattering, WDT shows a reduction in error in two high energy groups (9 and 11).

### 4.4 BWR

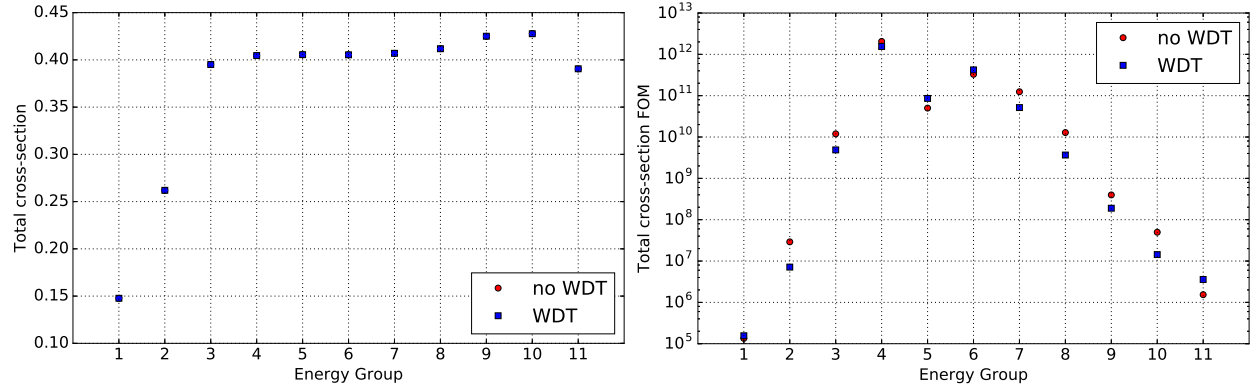
As discussed in Section 3.4, a cutoff threshold value for  $P_{\text{real}}$  may improve performance. A simulation was run on a BWR with parameters summarized in Table 6. Various values of a WDT threshold as described in Eq. 21 were tested on the BWR model.

$$\frac{\Sigma_t}{\Sigma_{\text{maj}}} \leq t_{\text{WDT}} \quad (21)$$

The surface tracking routine was not modified, so when  $t \leq 0.10$ , the normal surface tracking and delta-tracking routines are used. To assess the impact of threshold, we can use the total FOM for all cross-sections of interest. This total FOM is the sum of the figure of merit from all the cross-sections discussed in the

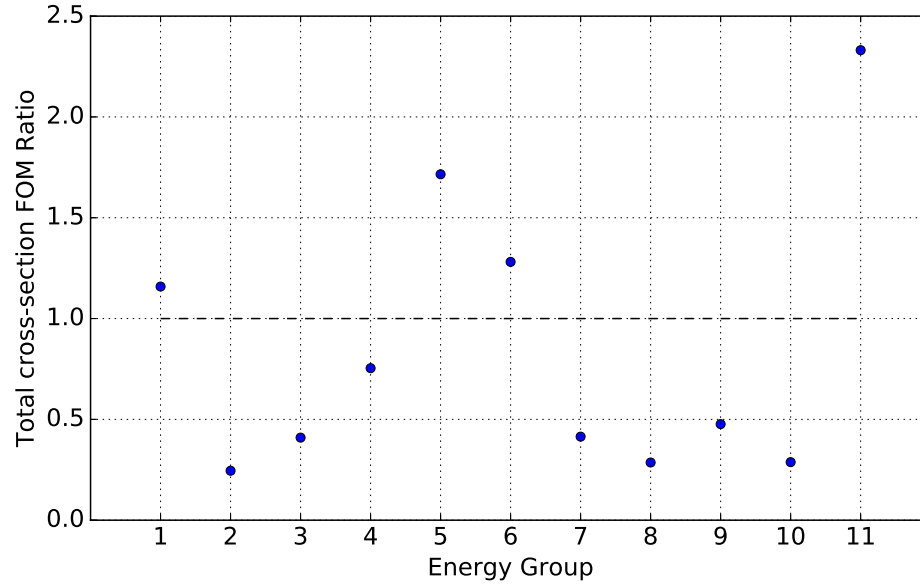






(a) Infinite total cross-section by energy group

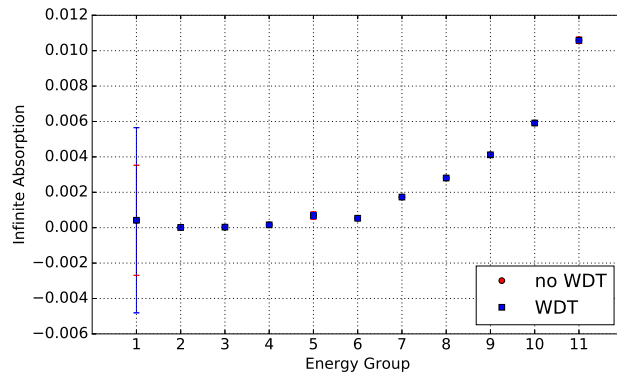
(b) FOM by energy group



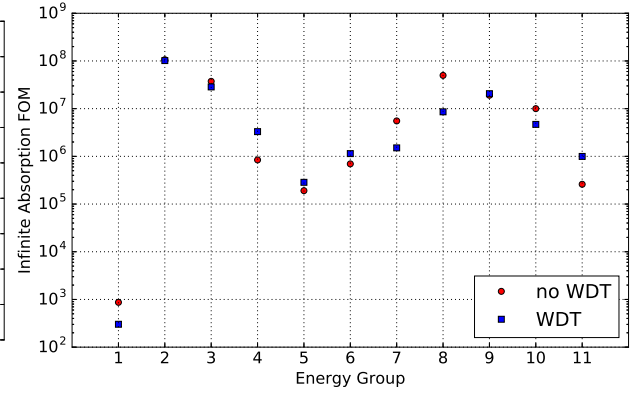
(c) Ratio of WDT FOM to no WDT FOM

Figure 4: FOM of the infinite total cross-section by energy group.

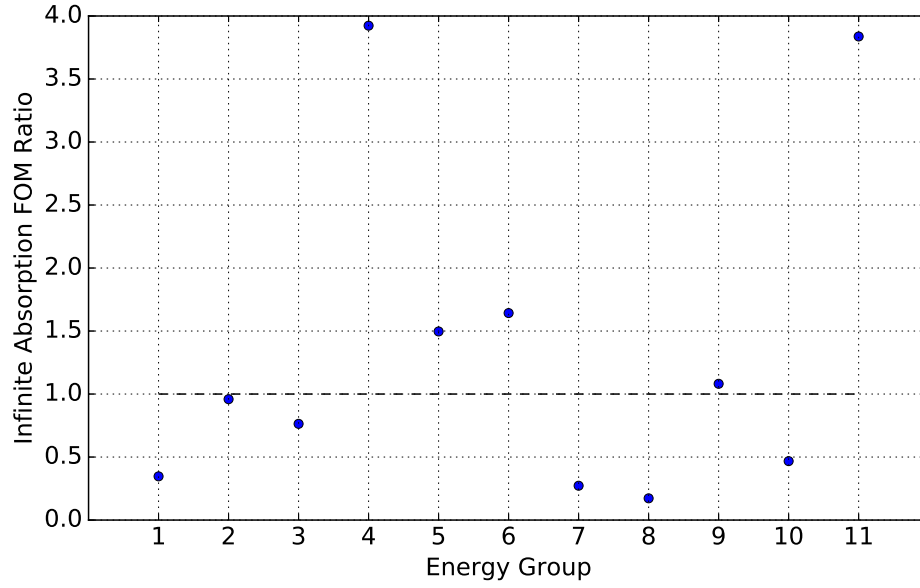




(a) Infinite absorption cross-section by energy group



(b) FOM by energy group



(c) Ratio of WDT FOM to no WDT FOM

Figure 6: FOM of the infinite absorption cross-section by energy group.







previous section. The results are shown in Fig. 10. The total FOM is scattered but shows a decreasing trend as the threshold value rises.

Parameter	Value
Pitch	1.295
Fuel	LEU
Void Fraction	25%
Poison	Gd pins

Table 6: BWR parameters.

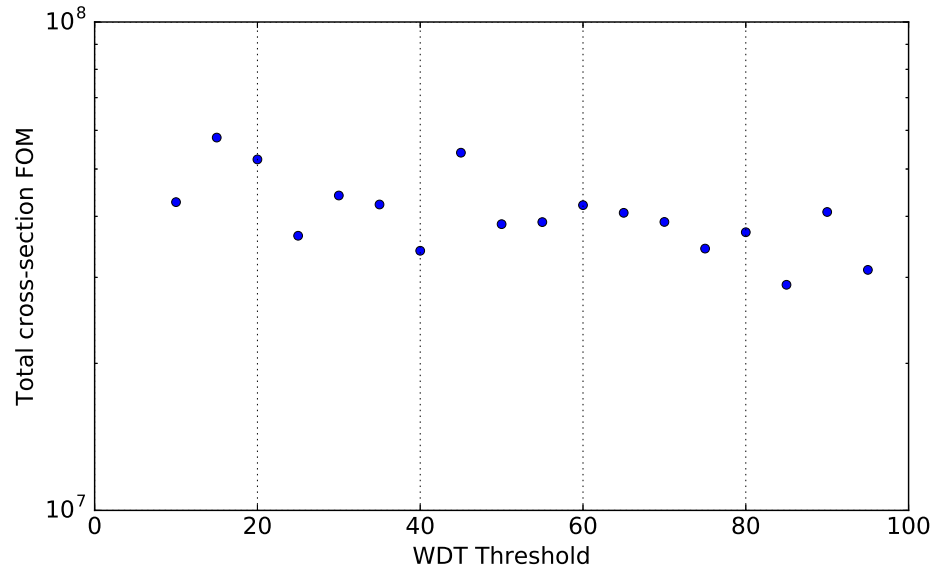


Figure 10: FOM for the BWR for different WDT threshold values.



## 5 Discussion

As discussed in the last section, the main goal of implementing new methods in Serpent 2 at INL is to improve the development of group cross-sections for deterministic codes. Section 4.3 provided the changes in FOM per energy group in some of the cross-sections of interest. As all cross-sections are important to correctly modeling the TREAT reactor, these values must be aggregated. Fig. 11 shows the result of averaging the FOM ratio from each cross-section: total, absorption, capture, and the  $P_0$ ,  $P_1$ , and  $P_2$  moments of scattering. This ratio shows the average FOM improvement when using the WDT method.

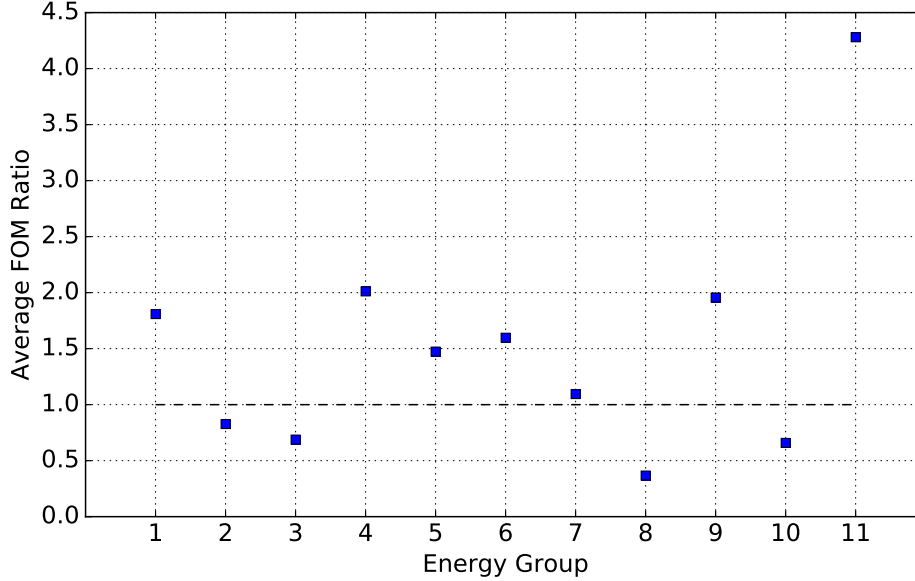


Figure 11: Average FOM ratio for all cross-sections.

On average, the WDT method improves FOM for the mid-energy range (4–7) and a large improvement for the highest energy group. More tests need to be run to determine if this is statistical noise, or shows an actual trend based on energy groups. This is discussed in Sec. 6.

## 6 Future Work

Many of the FOM values and results presented need to be evaluated through multiple batches of runs to determine if they are consistently observed. Importantly, different energy groups need to be used and evaluated, as the top of the eleven energy groups used for the TREAT reactor consists of the entire resonance region. We anticipate that WDT will provide better results in this region, where there can be many virtual collisions that do not contribute to the statistics. This may explain the improvement seen in the highest energy group FOM shown in Sec. 5, but this needs to be verified. Smaller energy bins will give more insight into the possible benefits of the method.

The rouletting routine and WDT threshold values require tuning. Although some testing was done by changing the WDT threshold value, this needs to be examined closer. In addition, the rouletting routine uses an arbitrary weight cutoff, which could be tuned to provide better results.

## 7 Conclusion

The WDT replaces the normal rejection sampling of Woodcock delta-tracking with an implicit weight reduction for absorption events. We have shown that this method can be extended to scattering by reintroducing the rejection sampling after sampling collision type. This method has the potential to make up for a major disadvantage of delta-tracking, that virtual collisions offer no statistical benefit. Implementation of WDT

with scattering requires the introduction of a rouletting routine and a threshold value, which both need to be tuned.

Preliminary results show that the WDT method is as good as normal weighted delta-tracking, returning a higher FOM in some cases. Therefore, with tuning and adjustment, the method may return better statistical results in future studies.

## A Data Tables for TREAT homogenous fuel pin.

Parameters	No WDT			WDT			Ratios	
	$\hat{x}$	$\sigma^2(\hat{x})$	FOM	$\hat{x}$	$\sigma^2(\hat{x})$	FOM	$\sigma^2(\hat{x})$	FOM
1	3.4145	0.00101	$8.245889 \times 10^3$	3.4082	0.00077	$1.391467 \times 10^4$	0.76	1.69
2	63.1888	0.00045	$4.153892 \times 10^4$	63.1755	0.00032	$8.056649 \times 10^4$	0.71	1.94
3	55.7202	0.00043	$4.549287 \times 10^4$	55.7222	0.00032	$8.056649 \times 10^4$	0.74	1.77
4	50.4557	0.00041	$5.003945 \times 10^4$	50.4562	0.00027	$1.131688 \times 10^5$	0.66	2.26
5	42.4235	0.00043	$4.549287 \times 10^4$	42.4237	0.00031	$8.584816 \times 10^4$	0.72	1.89
6	40.5290	0.00038	$5.825230 \times 10^4$	40.5304	0.00030	$9.166676 \times 10^4$	0.79	1.57
7	23.9886	0.00048	$3.650882 \times 10^4$	23.9839	0.00038	$5.713302 \times 10^4$	0.79	1.56
8	68.2463	0.00049	$3.503387 \times 10^4$	68.2439	0.00023	$1.559548 \times 10^5$	0.47	4.45
9	48.0293	0.00034	$7.276498 \times 10^4$	48.0477	0.00026	$1.220415 \times 10^5$	0.76	1.68
10	52.8539	0.00020	$2.102908 \times 10^5$	52.8318	0.00038	$5.713302 \times 10^4$	1.90	0.27
11	18.2887	0.00084	$1.192125 \times 10^4$	18.2994	0.00061	$2.217148 \times 10^4$	0.73	1.86

Table 7: Infinite group flux for the TREAT homogenous fuel pin. The ratios column is expressed as the value with WDT divided by the value without WDT.

Parameters	No WDT			WDT			Ratios	
	$\hat{x}$	$\sigma^2(\hat{x})$	FOM	$\hat{x}$	$\sigma^2(\hat{x})$	FOM	$\sigma^2(\hat{x})$	FOM
1	0.147690	0.00025	$1.345861 \times 10^5$	0.14770	0.00023	$1.559548 \times 10^5$	0.92	1.16
2	0.261935	0.00002	$2.910599 \times 10^7$	0.26194	0.00003	$7.136685 \times 10^6$	2.00	0.25
3	0.395038	0.00000	$1.192125 \times 10^{10}$	0.39504	0.00000	$4.881662 \times 10^9$	1.55	0.41
4	0.404556	0.00000	$2.053621 \times 10^{12}$	0.40456	0.00000	$1.548134 \times 10^{12}$	1.14	0.75
5	0.405524	0.00000	$5.003945 \times 10^{10}$	0.40552	0.00000	$8.584816 \times 10^{10}$	0.76	1.72
6	0.405373	0.00000	$3.285794 \times 10^{11}$	0.40537	0.00000	$4.209188 \times 10^{11}$	0.88	1.28
7	0.406940	0.00000	$1.244324 \times 10^{11}$	0.40694	0.00000	$5.156255 \times 10^{10}$	1.54	0.41
8	0.411804	0.00000	$1.282065 \times 10^{10}$	0.41180	0.00000	$3.666670 \times 10^9$	1.85	0.29
9	0.425011	0.00000	$3.975251 \times 10^8$	0.42501	0.00001	$1.893941 \times 10^8$	1.43	0.48
10	0.427695	0.00001	$4.977297 \times 10^7$	0.42768	0.00002	$1.432293 \times 10^7$	1.85	0.29
11	0.390440	0.00007	$1.536091 \times 10^6$	0.39057	0.00005	$3.580733 \times 10^6$	0.65	2.33

Table 8: Infinite total reaction cross-section for the TREAT homogenous fuel pin. The ratios column is expressed as the value with WDT divided by the value without WDT.

Parameters	No WDT			WDT			Ratios	
	$\hat{x}$	$\sigma^2(\hat{x})$	FOM	$\hat{x}$	$\sigma^2(\hat{x})$	FOM	$\sigma^2(\hat{x})$	FOM
1	0.00041	0.00319	$8.266066 \times 10^2$	0.00041	0.00536	$2.871606 \times 10^2$	1.68	0.35
2	0.00000	0.00003	$1.244324 \times 10^7$	0.00000	0.00004	$6.365747 \times 10^6$	1.38	0.51
3	0.00001	0.00002	$2.330092 \times 10^7$	0.00001	0.00002	$1.704547 \times 10^7$	1.16	0.73
4	0.00006	0.00016	$3.285794 \times 10^5$	0.00006	0.00007	$1.683675 \times 10^6$	0.44	5.12
5	0.00030	0.00028	$1.072912 \times 10^5$	0.00030	0.00020	$2.062502 \times 10^5$	0.71	1.92
6	0.00026	0.00011	$6.951762 \times 10^5$	0.00026	0.00009	$1.018520 \times 10^6$	0.82	1.47
7	0.00049	0.00004	$5.257270 \times 10^6$	0.00049	0.00008	$1.391467 \times 10^6$	1.92	0.26
8	0.00077	0.00001	$8.939985 \times 10^7$	0.00077	0.00002	$1.432293 \times 10^7$	2.47	0.16
9	0.00106	0.00002	$2.330092 \times 10^7$	0.00106	0.00002	$2.546299 \times 10^7$	0.95	1.09
10	0.00150	0.00003	$9.346257 \times 10^6$	0.00150	0.00004	$4.461876 \times 10^6$	1.43	0.48
11	0.00274	0.00018	$2.596183 \times 10^5$	0.00274	0.00009	$9.747174 \times 10^5$	0.51	3.75

Table 9: Infinite capture reaction cross-section for the TREAT homogenous fuel pin. The ratios column is expressed as the value with WDT divided by the value without WDT.

Parameters	No WDT			WDT			Ratios	
	$\hat{x}$	$\sigma^2(\hat{x})$	FOM	$\hat{x}$	$\sigma^2(\hat{x})$	FOM	$\sigma^2(\hat{x})$	FOM
1	0.00042	0.00311	$8.696800 \times 10^2$	0.00042	0.00523	$3.016137 \times 10^2$	1.68	0.35
2	0.00001	0.00001	$1.061941 \times 10^8$	0.00001	0.00001	$1.018520 \times 10^8$	1.01	0.96
3	0.00003	0.00002	$3.738503 \times 10^7$	0.00003	0.00002	$2.854674 \times 10^7$	1.13	0.76
4	0.00016	0.00010	$8.411632 \times 10^5$	0.00016	0.00005	$3.300003 \times 10^6$	0.50	3.92
5	0.00069	0.00021	$1.907399 \times 10^5$	0.00069	0.00017	$2.854674 \times 10^5$	0.81	1.50
6	0.00053	0.00011	$6.951762 \times 10^5$	0.00053	0.00009	$1.141870 \times 10^6$	0.77	1.64
7	0.00173	0.00004	$5.530330 \times 10^6$	0.00173	0.00007	$1.506576 \times 10^6$	1.90	0.27
8	0.00281	0.00001	$4.977297 \times 10^7$	0.00281	0.00003	$8.584816 \times 10^6$	2.38	0.17
9	0.00412	0.00002	$1.907399 \times 10^7$	0.00412	0.00002	$2.062502 \times 10^7$	0.95	1.08
10	0.00591	0.00003	$1.000194 \times 10^7$	0.00591	0.00004	$4.676875 \times 10^6$	1.45	0.47
11	0.01059	0.00018	$2.596183 \times 10^5$	0.01059	0.00009	$9.962575 \times 10^5$	0.51	3.84

Table 10: Infinite absorption reaction cross-section for the TREAT homogenous fuel pin. The ratios column is expressed as the value with WDT divided by the value without WDT.

Parameters	No WDT			WDT			Ratios	
	$\hat{x}$	$\sigma^2(\hat{x})$	FOM	$\hat{x}$	$\sigma^2(\hat{x})$	FOM	$\sigma^2(\hat{x})$	FOM
1	0.14728	0.00024	$1.460353 \times 10^5$	0.14727	0.00025	$1.320001 \times 10^5$	1.04	0.90
2	0.26192	0.00002	$2.910599 \times 10^7$	0.26193	0.00003	$7.136685 \times 10^6$	2.00	0.25
3	0.39501	0.00000	$5.841411 \times 10^9$	0.39501	0.00000	$4.881662 \times 10^9$	1.08	0.84
4	0.40439	0.00000	$9.519728 \times 10^9$	0.40439	0.00000	$4.881662 \times 10^9$	1.38	0.51
5	0.40484	0.00000	$1.737940 \times 10^9$	0.40484	0.00000	$2.854674 \times 10^9$	0.77	1.64
6	0.40484	0.00000	$5.003945 \times 10^8$	0.40484	0.00000	$1.432293 \times 10^9$	0.59	2.86
7	0.40521	0.00001	$2.500485 \times 10^8$	0.40521	0.00000	$4.907798 \times 10^8$	0.71	1.96
8	0.40900	0.00000	$3.975251 \times 10^8$	0.40899	0.00001	$3.171860 \times 10^8$	1.11	0.80
9	0.42089	0.00001	$9.519728 \times 10^7$	0.42090	0.00001	$2.630742 \times 10^8$	0.60	2.76
10	0.42178	0.00001	$4.977297 \times 10^7$	0.42177	0.00002	$3.222659 \times 10^7$	1.23	0.65
11	0.37985	0.00007	$1.716660 \times 10^6$	0.37998	0.00006	$2.452440 \times 10^6$	0.83	1.43

Table 11: Infinite  $P_0$  scattering cross-section for the TREAT homogenous fuel pin. The ratios column is expressed as the value with WDT divided by the value without WDT.

Parameters	No WDT			WDT			Ratios	
	$\hat{x}$	$\sigma^2(\hat{x})$	FOM	$\hat{x}$	$\sigma^2(\hat{x})$	FOM	$\sigma^2(\hat{x})$	FOM
1	0.02891	0.00170	$2.910599 \times 10^3$	0.02887	0.00073	$1.548134 \times 10^4$	0.43	5.32
2	0.02439	0.00056	$2.682280 \times 10^4$	0.02436	0.00040	$5.156255 \times 10^4$	0.71	1.92
3	0.02376	0.00058	$2.500485 \times 10^4$	0.02376	0.00057	$2.539245 \times 10^4$	0.98	1.02
4	0.02271	0.00069	$1.766778 \times 10^4$	0.02270	0.00070	$1.683675 \times 10^4$	1.01	0.95
5	0.02263	0.00066	$1.931045 \times 10^4$	0.02266	0.00065	$1.952665 \times 10^4$	0.98	1.01
6	0.02231	0.00061	$2.260584 \times 10^4$	0.02235	0.00067	$1.837828 \times 10^4$	1.10	0.81
7	0.02157	0.00066	$1.931045 \times 10^4$	0.02165	0.00172	$2.788672 \times 10^3$	2.61	0.14
8	0.01523	0.00072	$1.622614 \times 10^4$	0.01524	0.00133	$4.663920 \times 10^3$	1.85	0.29
9	0.00176	0.01084	$7.158494 \times 10^1$	0.00175	0.00550	$2.727275 \times 10^2$	0.51	3.81
10	0.00530	0.00191	$2.305757 \times 10^3$	0.00532	0.00314	$8.367488 \times 10^2$	1.64	0.36
11	-0.02152	0.00280	$1.072912 \times 10^3$	-0.02157	0.00120	$5.729172 \times 10^3$	0.43	5.34

Table 12: Infinite  $P_1$  scattering cross-section for the TREAT homogenous fuel pin. The ratios column is expressed as the value with WDT divided by the value without WDT.

Parameters	No WDT			WDT			Ratios	
	$\hat{x}$	$\sigma^2(\hat{x})$	FOM	$\hat{x}$	$\sigma^2(\hat{x})$	FOM	$\sigma^2(\hat{x})$	FOM
1	0.03781	0.00148	$3.840226 \times 10^3$	0.03779	0.00088	$1.065342 \times 10^4$	0.59	2.77
2	0.00517	0.00177	$2.684935 \times 10^3$	0.00517	0.00169	$2.888557 \times 10^3$	0.95	1.08
3	0.00073	0.00777	$1.393279 \times 10^2$	0.00072	0.01257	$5.221362 \times 10^1$	1.62	0.37
4	0.00056	0.02148	$1.823105 \times 10^1$	0.00057	0.02380	$1.456466 \times 10^1$	1.11	0.80
5	0.00058	0.03260	7.914893	0.00059	0.03146	8.335602	0.97	1.05
6	-0.00040	0.04173	4.830404	-0.00039	0.03350	7.351311	0.80	1.52
7	-0.00122	0.02570	$1.273544 \times 10^1$	-0.00122	0.01359	$4.466995 \times 10^1$	0.53	3.51
8	-0.00465	0.00172	$2.843304 \times 10^3$	-0.00463	0.00241	$1.420432 \times 10^3$	1.40	0.50
9	0.00230	0.00711	$1.663953 \times 10^2$	0.00228	0.00444	$4.184932 \times 10^2$	0.62	2.52
10	-0.01856	0.00066	$1.931045 \times 10^4$	-0.01853	0.00050	$3.300003 \times 10^4$	0.76	1.71
11	-0.01535	0.00339	$7.319490 \times 10^2$	-0.01533	0.00112	$6.576856 \times 10^3$	0.33	8.99

Table 13: Infinite  $P_2$  scattering cross-section for the TREAT homogenous fuel pin. The ratios column is expressed as the value with WDT divided by the value without WDT.

## References

- [1] E. E. Lewis and W.F. Miller, Jr. *Computational Methods of Neutron Transport*. American Nuclear Society, 1993.
- [2] Jaakko Leppänen. Performance of Woodcock delta-tracking in lattice physics applications using the Serpent Monte Carlo reactor physics burnup calculation code. *Annals of Nuclear Energy*, 37:715–722, 2010.
- [3] Jaakko Leppänen. Modeling of Nonuniform Density Distributions in the Serpent 2 Monte Carlo Code. *Nuclear Science and Engineering*, 174:318–325, 2013.
- [4] Iván Lux and László Koblinger. *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations*. CRC Press, 1991.
- [5] L.W.G. Morgan and D. Kotlyar. Weighted-delta-tracking for Monte Carlo particle transport. *Annals of Nuclear Energy*, 85:1184–1188, 2015.
- [6] Toni Kaltiaisenaho. Statistical Tests and the Underestimation of Variance in Serpent 2. Technical Report VTT-R-00371-14, VTT Technical Research Centre of Finland, 2014.
- [7] E.R Woodcock et al. Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry. *ANL-7050. Argonne National Laboratory.*, 1965.

**INTERIM REPORT  
ON  
MULTIPHYSICS SEMI-ANALYTICAL BENCHMARK TEST HARNESS  
HEAVY TESTING QNA SUITE  
FOR THE RATTLESNAKE TRANSPORT CODE**

**B. Ganapol  
INL Visiting Staff  
Ganapol@cowboy.ame.arizona.edu**

**SUMMARY**

We all know how important V&V of multiphysics applications in nuclear engineering is; however, at the same time, the lack of available true multiphysics benchmarks is severely limiting. In their absence, we must be satisfied with unit testing, which is generally inconclusive when simulating physical phenomena at a variety of time scales.

What we present here is the first installment (Su2016) of the Multiphysics Test HARNESS for verification of the Rattlesnake transport code. Because generating multiphysics benchmarks is not particularly straightforward, we begin simple with neutronics described by point kinetics. In this way, we concentrate on coupling as much physics as possible in a meaningful way. Now, before you say point kinetics is not particularly informative when it comes to V&V, read to the end of this document and then make your judgement. One would be surprised on how well point kinetics can capture important physics. In any case, one can numerically solve the Point Kinetics Equations (PKEs) relatively easily to provide highly precise semi-analytical benchmarks.

The class of benchmarks, called semi-analytical, represents highly precise numerical solution evaluations, either directly, or by convergence acceleration of a numerical algorithm. Usually the most precise semi-analytical benchmarks result from evaluations in closed form. Unfortunately, closed form solutions are infrequent and one must therefore contend with infinite series and Pickard iterative forms. Recently, with the emergence of convergence acceleration, one can accelerate solutions based on discretizations to the true solution as a limit. While not always guaranteed to converge to the correct result, more often than not, they

indeed do. In this way, a highly precise solution is obtained simply by rearranging iterates in a systematic way to capture the asymptotic behavior of a numerical solution sequence. In the kinetic benchmarks to follow, liberal use is made of two accelerations, Richardsons and Wynn-epsilon extrapolations.

The benchmarks we report involve Doppler feedback in increasing order of physical complexity. Two methods of solution provide the high order solutions—usually to six or more digits of precision. We now summarize the three benchmarks in the Semi-Analytical Rattlesnake Test Harness.

### **1. Keepin Doppler Multiphysics Point Kinetics Benchmark**

The first benchmark is a well-known benchmark, first published by R.G. Keepin, hence the name. There are several versions of this Doppler limited excursion. The one we consider is reactivity ramp initiation from fast to slow insertion. Reactor heating is adiabatic with several prescribed Doppler feedback coefficients. Reactivity insertion is proportional to the fission energy deposited within the core, which goes into sensible heating. The Keepin benchmark particularly lends itself to an analytical solution in the form of Taylor series for the reactor power and precursor concentrations, as well as the fission energy. Most importantly, this benchmark introduces the **Converged Accelerated Taylor Series (CATS)** algorithm as an appropriate method for performing multiphysics benchmarks and a superior method for solving the PKEs. As the name implies, convergence acceleration plays a central role in the high precision solution.

### **2. SKINATH Multiphysics Point Kinetics Benchmark**

SKINATH is a 1984 point kinetics solver much in the form of the MATLAB™ ODE solvers of today. The solver, called **LSODE**, originated at LNL. **SKINATH** developers Lee Dodds and Mike Westfall solve the PKEs with Doppler insertion based on instantaneous fuel temperature difference from initial temperature. The reactor, simulating a waste canister, is a homogeneous fuel-water mix characterized by a uniform (lumped) temperature. The canister is air cooled with a non-trivial temperature dependent heat transfer coefficient. The air coolant remains at a fixed temperature. We cannot apply the **CATS** algorithm here because the heat transfer coefficient does not possess a Taylor series. For this reason, we use the **Backward Euler Finite Difference (BEFD)** algorithm. This algorithm handles nearly any form of reactivity insertion through iteration of the Jacobian coupled with convergence acceleration. Thus, we showcase the **BEFD** algorithm, including liberal use of convergence acceleration, with this benchmark.

### **3. HTR-10 Multiphysics Point Kinetics Benchmark**



An informative article in an obscure journal, *Nuclear Technology & Radiation Protection*, from an unknown author, suggests a transient analysis for China's HTR10 pebble bed reactor. We conveniently find most of the parameters for the PKE analysis in the article. Part of the challenge here is to reproduce the results of the article with the **CATS** and **BEFD** methods. The pebble bed core is doubly heterogeneous with 8335 fissile kernels in TRISO fuel embedded in 16988 graphite pebbles in a continuously moving arrangement through the core. Flowing Helium cools the pebbles and remains at a constant temperature during the transient. Doppler feedback comes in the form of instantaneous temperature difference from initial temperature for both fuel and graphite moderator. Heat is transferred from the fuel kernel to the graphite and then to the Helium. The model considers the transient response of a single representative pebble to a uniform reactivity insertion. **CATS** provides the solution for the constant specific heat case. To add more realism to the model and more physics, we allow the specific heats to be temperature dependent, which **CATS** algorithm cannot accommodate, but the **BEFD** algorithm can. Hence, we apply both PKE solution methods and in so doing show that they give identical results for constant specific heats.

## Interim Progress Report A1 (Su16)

### Keepin Doppler Multiphysics Point Kinetics Benchmark

B. Ganapol

INL Visiting Staff

[Ganapol@cowboy.ame.arizona.edu](mailto:Ganapol@cowboy.ame.arizona.edu)

M. DeHart, F. Gleicher, J. Ortensi, S. Schunert and R. Martineau

INL Staff

#### I. Benchmark Scope

We consider a fundamental benchmark found in the literature, the Keepin benchmark[1], which was one of the first (limited) Doppler multiphysics benchmarks. One considers this a minimal multiphysics benchmark since neutronic heating produces only Doppler feedback from deposited fission energy. The only model required, in addition to the neutronics model, is how the Doppler feedback adiabatically introduces reactivity. There are no explicit models for either the reactor or coolant temperature. One of the primary reasons for featuring this case is to introduce the **CATS**[2] solution of the Reactor Point Kinetics Equations (RKEs) as one of two the high precision methods.

This work is part of a semi-analytical multiphysics benchmarking effort recently initiated at INL to be part of the heavy testing QNA suit for the RATTLESNAKE transport code.

#### I.1 The Reactor Kinetics Model

The reactor kinetics model begins with the PKEs in the form

$$\begin{aligned}\frac{dN(t)}{dt} &= \left[ \frac{\rho(t, N) - \beta}{\Lambda} \right] N(t) + \sum_{l=1}^m \lambda_l C_l(t) + q(t) \\ \frac{dC_l(t)}{dt} &= \frac{\beta_l}{\Lambda} N(t) - \lambda_l C_l(t), \quad l = 1, \dots, m,\end{aligned}\tag{1a,b}$$

where  $N(t)$  and  $C_l(t)$  are the reactor power and precursor concentrations for  $m$  delayed neutron groups, which will be six here. Initial power is  $N(0)$  to give the initial precursor concentrations

$$C_l(0) = \frac{\beta_l}{\lambda_l \Lambda} N(0), \quad l = 1, \dots, m. \quad (1c)$$

$\rho(t, N)$  is the reactivity and all other symbols have their usual meaning

$\Lambda$	Neutron generation time
$\beta_l$	Delay neutron yield for group $l$
$\beta$	$\sum_{l=1}^m \beta_l$
$\lambda_l$	Delayed group decay constant.

Note that the reactivity is, in general, a strong function of power.

The proposed solution is not new, but its implementation is.

## I.2 Mathematical Formulation of the Solution

For a reactivity in the form

$$\rho(t) = \rho_0(t) + \hat{\rho}(t, N(t)), \quad (2a)$$

where the first term is an imposed insertion and the second represents Doppler feedback, we recast the model equations as a set of vector equations

$$\frac{d\mathbf{y}(t)}{dt} = \mathbf{A}(N(t))\mathbf{y}(t) + \mathbf{q}(t), \quad (2b)$$

to be solved, with

$$\begin{aligned} \mathbf{y}(t) &\equiv \begin{bmatrix} N(t) & C_1(t) & \dots & C_m(t) & \rho(t, N) \end{bmatrix}^T \\ \mathbf{q}(t) &\equiv \begin{bmatrix} 0 & 0 & \dots & 0 & \frac{d\rho_0(t)}{dt} \end{bmatrix}^T. \end{aligned} \quad (2c,d)$$

The Jacobian is

$$\mathbf{A}(t, N(t)) \equiv \begin{bmatrix} (\rho(t, N(t)) - \beta) / \Lambda & \lambda_1 & \lambda_2 & \dots & \lambda_m & 0 \\ \frac{\beta_1}{\Lambda} & -\lambda_1 & 0 & \dots & \dots & 0 \\ \frac{\beta_2}{\Lambda} & 0 & -\lambda_2 & & & \dots \\ \dots & \dots & & \dots & & \dots \\ \frac{\beta_m}{\Lambda} & 0 & \dots & 0 & -\lambda_m & 0 \\ 0 & 0 & \dots & & \dots & \frac{d\hat{\rho}(t, N(t))}{dt} \end{bmatrix}, \quad (2e)$$

and the initial conditions are

$$\mathbf{y}(0) = \begin{bmatrix} N(0) & \frac{\beta_1}{\lambda_1 \Lambda} N(0) & \dots & \frac{\beta_m}{\lambda_m \Lambda} N(0) & 0 \end{bmatrix}^T. \quad (2f)$$

The proposed solution is the following Taylor series in the time interval  $t_{j-1} \leq t \leq t_j$ :

$$\mathbf{y}(t) = \sum_{n=0}^{\infty} \mathbf{y}_{n,j-1} (t - t_{j-1})^n. \quad (3a)$$

If, in addition, one expresses the Jacobian in a Taylor series (TS)

$$\mathbf{A}(t, N(t)) = \sum_{n=0}^{\infty} \mathbf{a}_{n,j-1} (t - t_{j-1})^n, \quad (3b)$$

when introduced into the product term of the RHS of Eq(2b), there results

$$\mathbf{A}(N(t)) \mathbf{y}(t) = \sum_{n=0}^{\infty} \sum_{l=0}^n \mathbf{a}_{n-l} \mathbf{y}_{n,l} (t - t_{j-1})^n.$$

Assuming a Taylor series for the source vector, the series coefficients are

$$\mathbf{y}_{n+1,j-1} = \frac{1}{n+1} \left[ \sum_{l=0}^n \mathbf{a}_{n-l} \mathbf{y}_{n,l} + \mathbf{q}_n(t) \right], \quad (3c)$$

simply found by equating the coefficients of the powers of  $(t - t_{j-1})^n$ .

Expressing the Jacobian as a Taylor series is not particularly limiting since this assumption includes power series function representations; however, is not as general as the **BEFD** formulation[3].

The solution is remarkable in several ways. First, the solution is a true analytical solution, not in closed form, but analytical nevertheless. Second, we can make the solution as numerically precise as desired. In particular, the true numerical solution is the limit

$$y(t) = \lim_{N \rightarrow \infty} \sum_{n=0}^N y_{n,j-1} (t - t_{j-1})^n. \quad (4)$$

Using the concept of convergence acceleration, in the form of Wynn-epsilon (*W-e*) and Richardsons (*R*) extrapolations, one can realized the limit to high precision.

## **I.3 Numerical Implementation and Verification**

### **a. Numerical Implementation**

#### **a.1. Grid Refinement**

Figure 1 gives the computational flow of the **KEEPIN/CATS** algorithm. To begin, edit intervals  $[t_{j-1}, t_j]$ , called original intervals, are chosen, where  $t_j$  is a desired edit entered at input as are the initial conditions (ICs). To enable better performance and to generate sufficient plot resolution, we can further partition the interval between the original edits into *nel* subintervals at the start of the calculation.

The algorithm begins with the known initial conditions specified at  $t_0 = 0$ , or from a restart, to find the densities (neutron and precursor) at time edit  $t_1$  (at the end of the first edit interval). We then construct the TS solution for the first edit interval (called refinement  $l_0$ ) to give a first estimate of the solution vector called  $y(t_1; l_0)$ . Then divide the first edit interval by 2 to form two subintervals. We label this refinement  $l_1$ -- the first grid refinement. One then evaluates the TS for the densities at the end of the first subinterval as above. The solution serves as the initial condition for the TS of the second subinterval (initial term in TS) and gives a second estimate of the solution for refinement 1,  $y(t_1; l_1)$ , at the edit  $t_1$ . Subsequent

grid refinement of the subintervals continues to generate a sequence of solution estimates at  $t_1, y(t_1; l_j)$ , for

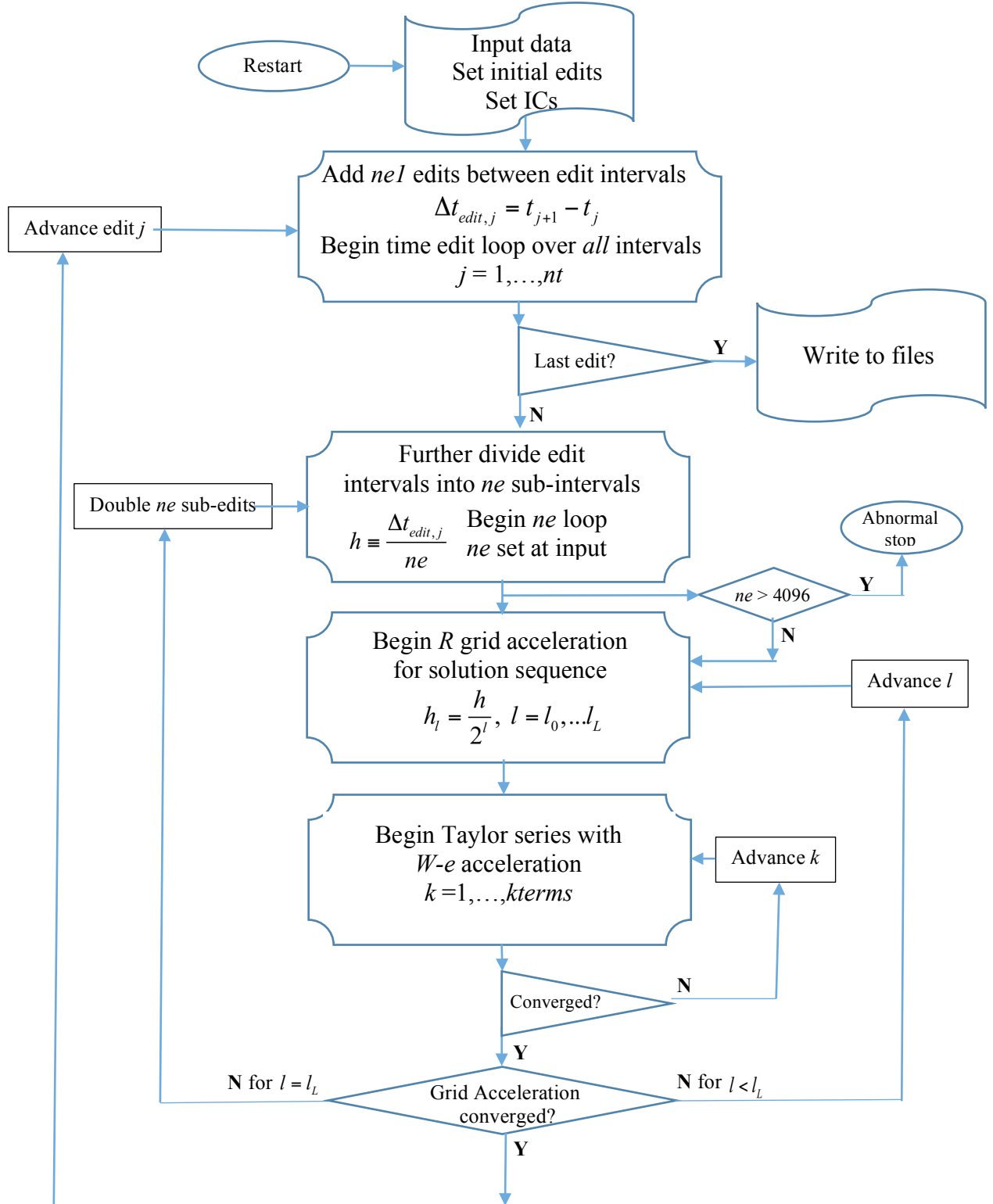


Fig. 1. Computational Flow Chart for **KEEPIN/CATS**.

refinements  $j=2,\dots,r$ . Since the sequence has a limit, we apply convergence acceleration in the forms of Wynn-epsilon ( $W-e$ ) and Richardsons ( $R$ ) accelerations[4] to estimate the limit precisely. Next, we treat the second edit interval  $[t_1, t_2]$  exactly as the first but with the newly converged solution  $y(t_1; l_\infty)$  as the second interval initial condition. The process continues until all original edits have converged or a limit on the number of subdivisions is exceeded. In this way, we reduce propagation error since each original interval begins with a highly precise (converged) initial condition.

### a.2. TS Acceleration

To further reduce error, the sequence of partial Taylor series sums, whose limit is the TS solution, is also accelerated by a  $W-e$  acceleration and required to converge within 25 ( $=K$ ) terms. If the TS does not converge within  $K$  terms, the calculation, at the current grid refinement, ceases and moves on to the next grid refinement ignoring the previous one entirely. In addition, if any TS for density converges negatively, the current interval is abandoned and the calculation moves on to the next refinement. Eventually, either the TS converges as the subintervals become smaller with further grid refinement and the sequence of solutions at each  $t_j$  is fully established or the maximum number of grid partitions is exceeds  $2^{12}$ .

### a.3. Adaptivity

If however, the TS fails to converge, say after  $r = 12$  grid refinements at a particular edit, the calculation restarts from the last converged edit, but now the next interval is partitioned into two sub-intervals on each failure. This introduces additional edits into the original interval and is the adaptive element of the algorithm. Partitioning of the original interval continues until a maximum number of 12 partitions. Note that all edits from any partition are converged through acceleration; Hence, with adaptivity, we introduce additional precision through precise initial conditions for each sub-edit interval.

As an initial test, the algorithm was applied to a 0.1\$/s ramp for Thermal Reactor I whose kinetic properties are in Table 1a. The results, given in Table 1b, agree to the highly precise results of the **BEFD**[3] algorithm, which uses a completely different solution method. Included in the table is the unaccelerated result with the incorrect digits emboldened. The advantage of the acceleration is clear if one desires highly precise results. This verification, give us confidence in the **KEEPIN/CATS** code with regard to the programming and precision of the algorithm.

## b. KEEPIN/CATS Benchmarks



A specific purpose **FORTRAN** program evaluates the classic Keepin benchmarks[1]. For these benchmarks, Doppler is proportional to the fission energy

**Table 1a. Thermal Reactor I.**

<i>i</i>	$\beta_i$	$\lambda_i$
1	0.000266	0.0127
2	0.001491	0.0317
3	0.001316	0.115
4	0.002849	0.311
5	0.000896	1.40
6	0.000182	3.87
$\beta = 0.007$		

**Table 1b. Ramp \$0.1/s.**

<i>t</i>	<i>Nacc</i>	<i>Nunacc</i>
2.0	1.338200050E+00	1.338200050E+00
4.0	2.228441897E+00	2.228441903E+00
6.0	5.582052449E+00	5.582052446E+00
8.0	4.278629573E+01	4.278629569E+01
10.0	4.511636239E+05	4.511636213E+05
11.0	1.792213607E+16	1.792213575E+16
14.0	5.540879076+132	5.540878472+132

deposition, initiated by ramp insertion

$$\rho(t) = at + B \int_0^t dt' N(t'). \quad (5a)$$

Taking a derivative,

$$\frac{d\rho(t)}{dt} = a + BN(t) \quad (5b)$$

gives the Taylor series coefficients as

$$\begin{aligned} \rho_{1,j} &= a \\ \rho_{k,j} &= -BN_{k-1,j}, \quad k = 2, \dots \end{aligned} \quad (5c)$$

Figure 2 shows the power variation in the adiabatic system for three ramp rates, each for a given Doppler coefficient for Thermal Reactor II, whose kinetics

parameters are in Table 2a. The oscillatory behavior of the rapid insertion rates is clear as well as the return to steady state. Table 2b provides benchmark powers expected to be correct to one unit in the last digit.

**Table 2a. Thermal Reactor II.**

$i$	$\beta_i$	$\lambda_i$
1	0.00021	0.0124
2	0.00141	0.0305
3	0.00127	0.111
4	0.00255	0.301
5	0.00074	1.13
6	0.00027	3.00

$\beta = 0.00645$

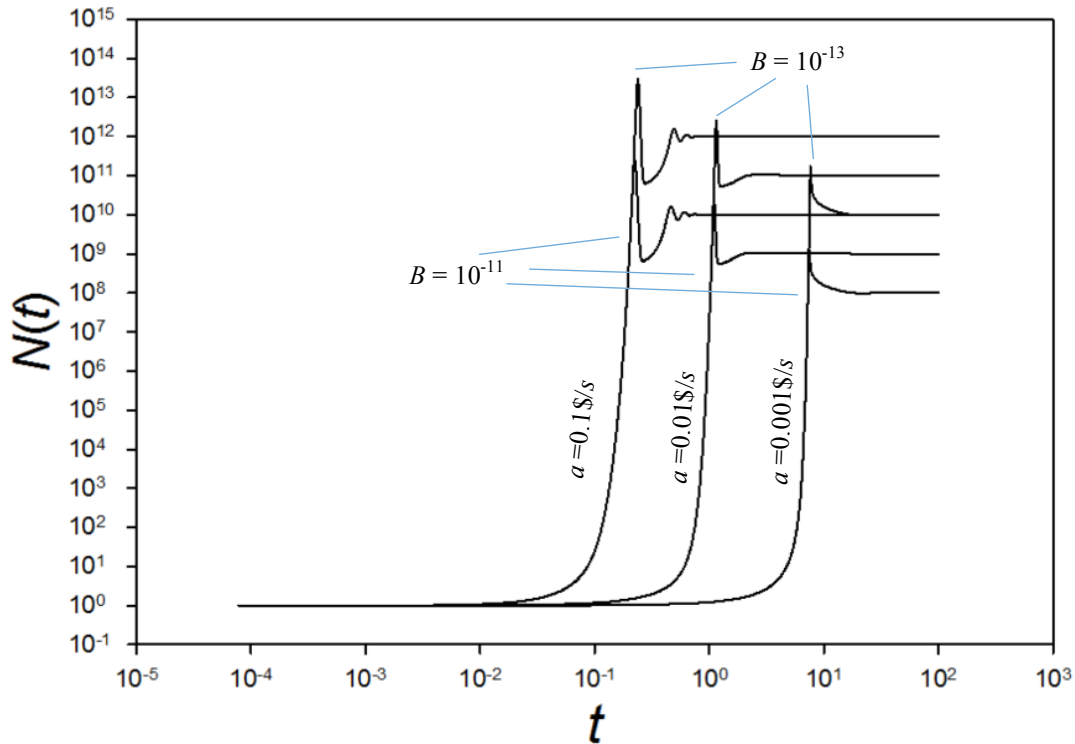


Fig. 2a. Keepin benchmark.

**Table 2b. Keepin benchmark**

$t \backslash (a, B)$	$(0.1, 10^{-11})$	$(0.01, 10^{-11})$	$(0.001, 10^{-11})$
1.0000E-01	2.473365821E+01	1.167210838E+00	1.014717771E+00
5.0000E-01	9.949893241E+09	4.269952844E+00	1.089821394E+00
5.0000E+00	1.002974047E+10	1.033798290E+09	1.416371139E+01
7.5000E+00	1.001798426E+10	1.019490285E+09	1.259267838E+09
1.0000E+02	1.000027914E+10	1.000278249E+09	1.002687252E+08
$t \backslash (a, B)$	$(0.1, 10^{-13})$	$(0.01, 10^{-13})$	$(0.001, 10^{-13})$
1.0000E-01	2.473365825E+01	1.167210838E+00	1.014717771E+00

5.0000E-01	1.543361786E+12	4.269952864E+00	1.089821395E+00
5.0000E+00	1.002974092E+12	1.033889665E+11	1.416371386E+01
7.5000E+00	1.001798437E+12	1.019499913E+11	8.361617703E+09
1.0000E+02	1.000027914E+12	1.000278249E+11	1.002687421E+10

Figure 2b shows the reactivity trace for fast and slow transients. The reactivity for the fast transient becomes negative, represented here as positive in red. The reactivity for the slow transient remains positive and is less oscillatory than the fast transient.

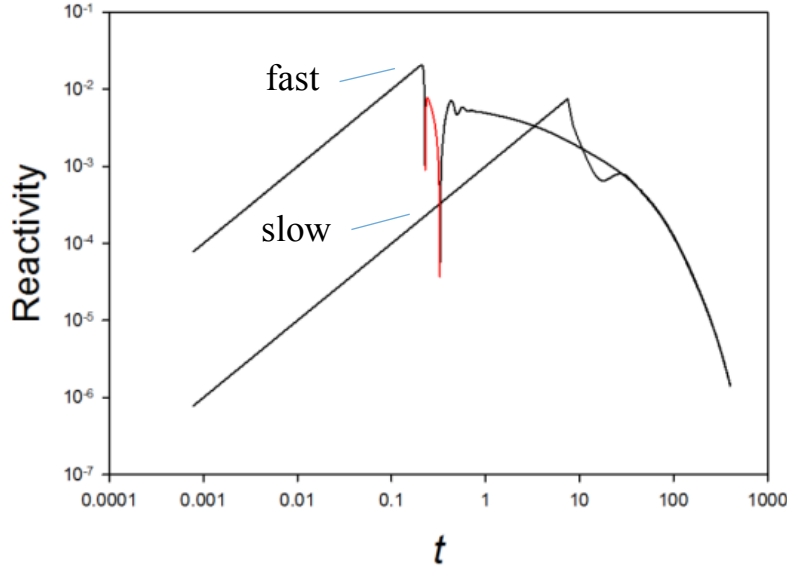


Fig. 2b. Reactivity trace.

### b.1. Asymptotic Power Variation

The asymptotic power res from the following analysis. Steady state comes for the reactivity of Eq(5a)

$$\rho(t) = at + B \int_0^t dt' N(t'). \quad (5a)$$

when  $\lim_{t \rightarrow \infty} \rho(t) = 0$ , which is in determinant. However the derivative,

$$\frac{d\rho(t)}{dt} = a + BN(t), \quad (5a)$$

must also vanish

$$\lim_{t \rightarrow \infty} \left[ \frac{d\rho(t)}{dt} \right] = a + BN(\infty) = 0,$$

implying

$$N(\infty) = -\frac{a}{B}.$$

The **KEEPIN/CATS** algorithm truly maintains the asymptotic values as shown in Table 3.

**Table 3. Comparison to Steady State.**

$a = 0.1, B = -10^{-11}$		
$t$	$N(t)$	$ 1-N(t)/N_\infty $
8.0000E+02	1.000025061E+10	2.506077E-05
9.0000E+02	1.000001183E+10	1.183269E-06
1.0000E+03	1.000000056E+10	5.597458E-08
$a = 0.01, B = -10^{-11}$		
8.0000E+02	1.000000013E+09	1.274444E-08
9.0000E+02	1.000000004E+09	3.686827E-09
1.0000E+03	1.000000001E+09	1.066548E-09
$a = 0.001, B = -10^{-11}$		
8.0000E+02	1.000000122E+08	1.219965E-07
9.0000E+02	1.000000035E+08	3.519115E-08
1.0000E+03	1.000000010E+08	1.015124E-08

## b.2. Derivative of Power

As the final result, we give the variation of the sign of the derivative of the power, found from its definition in (1a) with converged dependent variables. The oscillations in the power for fast ramp are clearly observed in top plate of Fig. 3a. Most importantly, the estimated time to an oscillation is captured in the sign trace of the derivative as shown in the bottom plate of Fig. 3a. The same is true for the slow ramp, but is less obvious in the power trace; however, is prominent in the sign trace of the bottom plate of Fig. 3b.

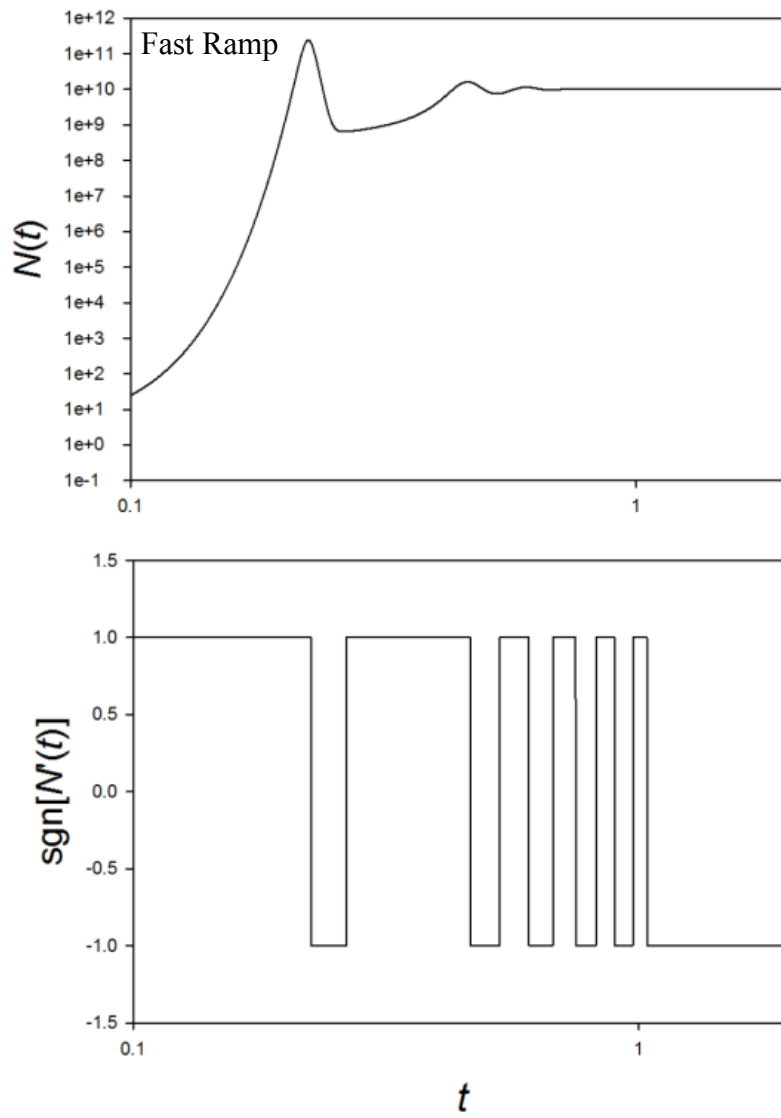


Fig. 3a Top: Power trace

Bottom: Trace of  $\text{sgn}\left[\frac{dN(t)}{dt}\right]$

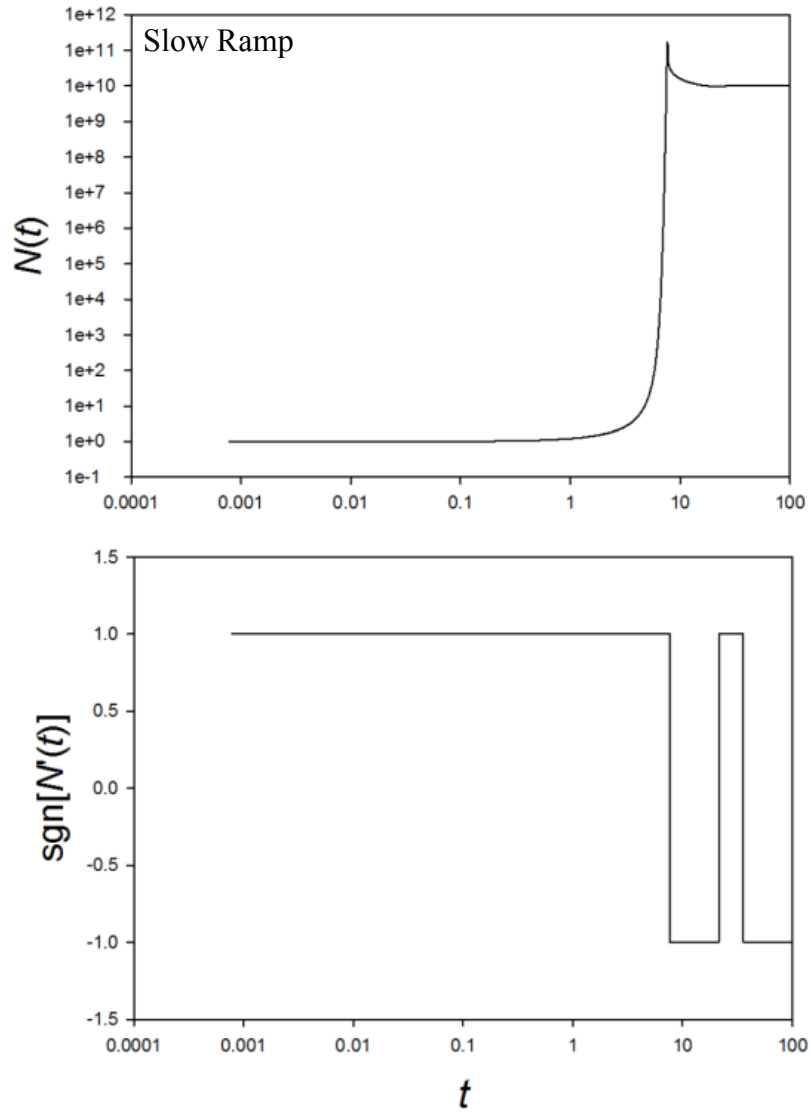


Fig. 3b Top: Power trace

Bottom: Trace of  $\text{sgn} \left[ \frac{dN(t)}{dt} \right]$

## REFERENCES

1. Keepin, R.G., *Physics of Nuclear Kinetics*, Addison-Wesley, USA, 1965.
2. B. Ganapol, *et.al.*, PHYSOR 2012, Knoxville, TN, CD-ROM, American Nuclear Society, LaGrange Park, IL, 2012.
3. B. Ganapol, *Annals of Nuclear Energy*, V62, pp 564-571, 2013.
4. Sidi, A., *Practical Extrapolations Methods*, Cambridge University Press, Cambridge, (2003).

## Interim Progress Report A2 (Su16)

### SKINATH Multiphysics Point Kinetics Benchmark

B. Ganapol

INL Visiting Staff

Ganapol@cowboy.ame.arizona.edu

M. DeHart, F. Gleicher, J. Ortensi, S. Schunert and R. Martineau

INL Staff

#### I. Benchmark Scope

The **SKINATH** point kinetics program[1], whose name comes from: *A Computer Program for Solving the Reactor Point KINetics EquAtions with Simple THERmal Hydraulic-Feedback*, had its origins at ORNL. The authors were interested in investigating low-level criticality events that might occur in a waste repository of 55-gallon drums. The physical setting incorporates a relatively complex thermal feedback for a fissionable material in a container (or a reactor), either critical or subcritical, with the potential to undergo further excursion. Air, assumed to be at constant temperature  $T_c$ , externally cools the system by absorbing all heat transferred from the excursion. The **SKINATH** numerical concept was to use an off-the-shelf Livermore ODE solver **LSODE**[2] to solve the point kinetic equations, much as one does today with **MATLAB**<sup>TM</sup>. Here, we develop a novel solver that uses the simplest of numerical methods including convergence acceleration to achieve high order precision. This work is part of a semi-analytical multiphysics benchmarking effort recently initiated at INL.

#### I.1 The Reactor Kinetics Model

The kinetics model begins with a heat balance for a core spatially averaged temperature  $T(t)$  according to

$$\rho_r V C_p \frac{dT(t)}{dt} = \epsilon N(t) - h_T(T(t)) [T(t) - T_c], \quad (1a)$$

where  $h_T(t)$  is the temperature dependent convective heat transfer coefficient

$$h_T(T(t)) \equiv A h d^{0.75} \left[ 1 - \frac{T_c}{T(t)} \right]^{0.25}. \quad (1b)$$

Table 1 gives the constants in the convective heat transfer coefficient of Eq(1b) with the values used in the **SKINATH** Benchmarks found below.

$N(t)$  is the reactor power,  $C_p$  is the core specific heat capacity,  $V$  is the reactor volume, and  $\varepsilon$  is the fraction of reactor power deposited as heat.

**Table 1. Model parameters.**

Symbol	Description	Unit or value
$A$	$1.649K^{0.75}(\rho_c g \hat{C}_p / \mu)^{0.25}$	17.52 for air
$K$	coolant thermal conductivity	J/m-s°C
$g$	gravity constant	m/s <sup>2</sup>
$\rho_r$	reactor (water) density	kg/m <sup>3</sup>
$\mu$	coolant viscosity	kg/m-s
$T_c$	coolant temperature	20°C
$VC_p$	coolant specific heat	$1800\pi d^2/4h$ J/kg°C
$h$	reactor height	0.23m
$d$	reactor diameter	0.20m

We model the transient reactor power with the following point kinetics equations:

$$\frac{dN(t)}{dt} = \left[ \frac{\rho(t, N) - \beta}{\Lambda} \right] N(t) + \sum_{l=1}^m \lambda_l C_l(t) + q(t) \quad (2a,b)$$

$$\frac{dC_l(t)}{dt} = \frac{\beta_l}{\Lambda} N(t) - \lambda_l C_l(t), \quad l = 1, \dots, m,$$

for  $m$  (six) delayed neutron groups and initial reactor power  $N(0)$ . The transient starts from steady state to give the initial precursor concentrations

$$C_l(0) = \frac{\beta_l}{\lambda_l \Lambda} N(0), \quad l = 1, \dots, m, \quad (3)$$

and  $\rho(0^-) = 0$ . We assume simple Doppler feedback

$$\rho_D(t, N) = \alpha_D [T(t) - T_c] \quad (4a)$$

along with an initiating reactivity to give the total imposed transient reactivity



$$\rho(t, T(t)) = \rho_0(t) - \alpha_D [T(t) - T_c]. \quad (4b)$$

Note that we base the Doppler contribution on the coolant temperature  $T_c$  and not on the actual reactor temperature at power, which is incorrect. One reason is that, in this model, there is no separate coolant heat balance for the coolant temperature response. To explain the error in this model, we make the following argument. The steady state initial power  $N(0)$  at the start of the transient comes from a quiescent reactor, at equilibrium with the coolant, brought to power. This means that the reactor temperature,  $T_r$ , is  $T_c$  and the Doppler prescription is appropriate—no Doppler feedback at zero power startup. The critical reactor temperature at  $N(0)$ , the initial power of the transient, is  $T_{r0}$ , and comes from solving Eq(1a) with zero derivative to give the transcendental equation

$$\varepsilon N(0) = h_T (T_{r0}) [T_{r0} - T_c]. \quad (5a)$$

One performs the solution by bisection for a given  $N(0)$ . With the reactor temperature at initiation of the transient now known and since the reactivity at the start of the transient is zero, we know the Doppler contribution to bring the reactor critical and to maintain initial criticality satisfies

$$\rho(t, T_{r0}) = 0 = \rho_{0c}(t) + \alpha_D [T_{r0} - T_c]. \quad (5b)$$

Thus, if the control rods impose the positive reactivity

$$\rho_{0cr}(t) = -\alpha_D [T_{r0} - T_c],$$

the reactor eventually becomes delayed critical at  $N(0)$  power. Therefore from startup to initial critical, the reactivity must actually be

$$\rho(t, T(t)) = -\alpha_D [T_{r0} - T_c] + \alpha_D [T(t) - T_c] = \alpha_D [T(t) - T_{r0}]. \quad (5c)$$

Hence, the true Doppler model gives no Doppler feedback at transient initiation according to Eq(5c). It certainly is possible to correct the **SKINATH** model by including the reactor temperature at initial power to give the correct feedback

equilibrium; but a correction will not be done here in order to reproduce the published benchmark.

## I.2 Mathematical Formulation

As is common procedure, one recasts the model equations as a set of vector equations

$$\frac{dy(t)}{dt} = A(T(t))y(t) + q(t), \quad (6a)$$

where for no external source

$$y(t) \equiv [N(t) \quad C_1(t) \quad \dots \quad C_m(t) \quad T(t) - T_c]^T$$

$$q(t) \equiv \left[ 0 \quad 0 \quad \dots \quad 0 \quad \frac{d\rho_0(t)}{dt} \right]^T \quad (6b,c)$$

$$A(t, T(t)) \equiv \begin{bmatrix} (\rho(t, T(t)) - \beta) / \Lambda & \lambda_1 & \lambda_2 & \dots & \lambda_m & 0 \\ \frac{\beta_1}{\Lambda} & -\lambda_1 & 0 & \dots & \dots & 0 \\ \frac{\beta_2}{\Lambda} & 0 & -\lambda_2 & & & \dots \\ \dots & \dots & \dots & & & \dots \\ \frac{\beta_m}{\Lambda} & 0 & \dots & 0 & -\lambda_m & 0 \\ \varepsilon / C_p & 0 & \dots & \dots & \dots & -h_r(T(t)) / C_p \end{bmatrix}, \quad (6d)$$

with initial conditions

$$y(0) = \left[ N(0) \quad \frac{\beta_1}{\lambda_1 \Lambda} N(0) \quad \dots \quad \frac{\beta_m}{\lambda_m \Lambda} N(0) \quad 0 \right]^T. \quad (6e)$$

Note that the Jacobian is temperature dependent.

## I.3 Numerical Implementation and Verification

### a. Numerical Implementation

The numerical solution of Eqs(4), (5) and (6) follows the backward fully implicit finite difference algorithm called **BEFD**[3]. While there certainly are a variety of

numerical methods from which to choose, the **BEFD** algorithm has the advantage of one of the most fundamental and easily incorporates Richardsons convergence acceleration scheme.

A **FORTRAN** program, called **SKINATH/BEFD**, computationally evaluates the benchmarks to follow. Once one establishes a numerical method, implementation becomes relatively routine and differs little from benchmark to benchmark— but there are some differences. Here, we note the specific inversion required in the **BEFD** algorithm from time step  $t_j$  to  $t_{j+1}$  (interval  $h$ )

$$\mathbf{y}_{j+1} = \left[ \mathbf{I} - h\mathbf{A}(t_{j+1}, T_{j+1}) \right]^{-1} \left[ \mathbf{y}_j + h\mathbf{q}(t_{j+1}) \right], \quad (7)$$

since it will be different depending upon the kinetics model. Following the analytical inversion of the first factor in Ref. [4], we find

$$\begin{aligned} [\mathbf{I} - h\mathbf{A}]^{-1} &= \\ &= \frac{1}{\sigma} \begin{bmatrix} 1 \\ \frac{h\beta_1 / \Lambda}{1 + h\lambda_1} \\ \dots \\ \frac{h\beta_m / \Lambda}{1 + h\lambda_m} \\ \frac{1 + hh_T(T(t))/\rho_F C_p}{1 + hh_T(T(t))/\rho_F C_p} \end{bmatrix} \begin{bmatrix} 1 \\ \frac{h\lambda_1}{1 + h\lambda_1} \\ \dots \\ \frac{h\lambda_m}{1 + h\lambda_m} \\ 0 \end{bmatrix}^T + \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & \frac{1}{1 + h\lambda_1} & & \\ \dots & & \dots & \\ \frac{1}{1 + h\lambda_m} & & & 0 \\ 0 & & & \frac{h_T(T(t))/\rho_F C_p}{1 + hh_T(T(t))/\rho_F C_p} \end{bmatrix} \quad (8a) \\ \text{with } t = t_j \text{ and } \sigma &\equiv 1 - h \frac{h_T(T(t))/\rho_F C_p}{\Lambda} + \sum_{l=1}^m \frac{h\beta_l / \Lambda}{1 + h\lambda_l}. \quad (8b) \end{aligned}$$

Since the Jacobian depends upon temperature, we seek an iterative solution to convergence between time steps. One performs iteration by lagging the temperature in the Jacobian and iterating  $N(t)$  and  $T(t)$  to convergence.

Acceleration of the iterations gives more rapid convergence; but even without acceleration, usually less than 10 iterations are necessary.

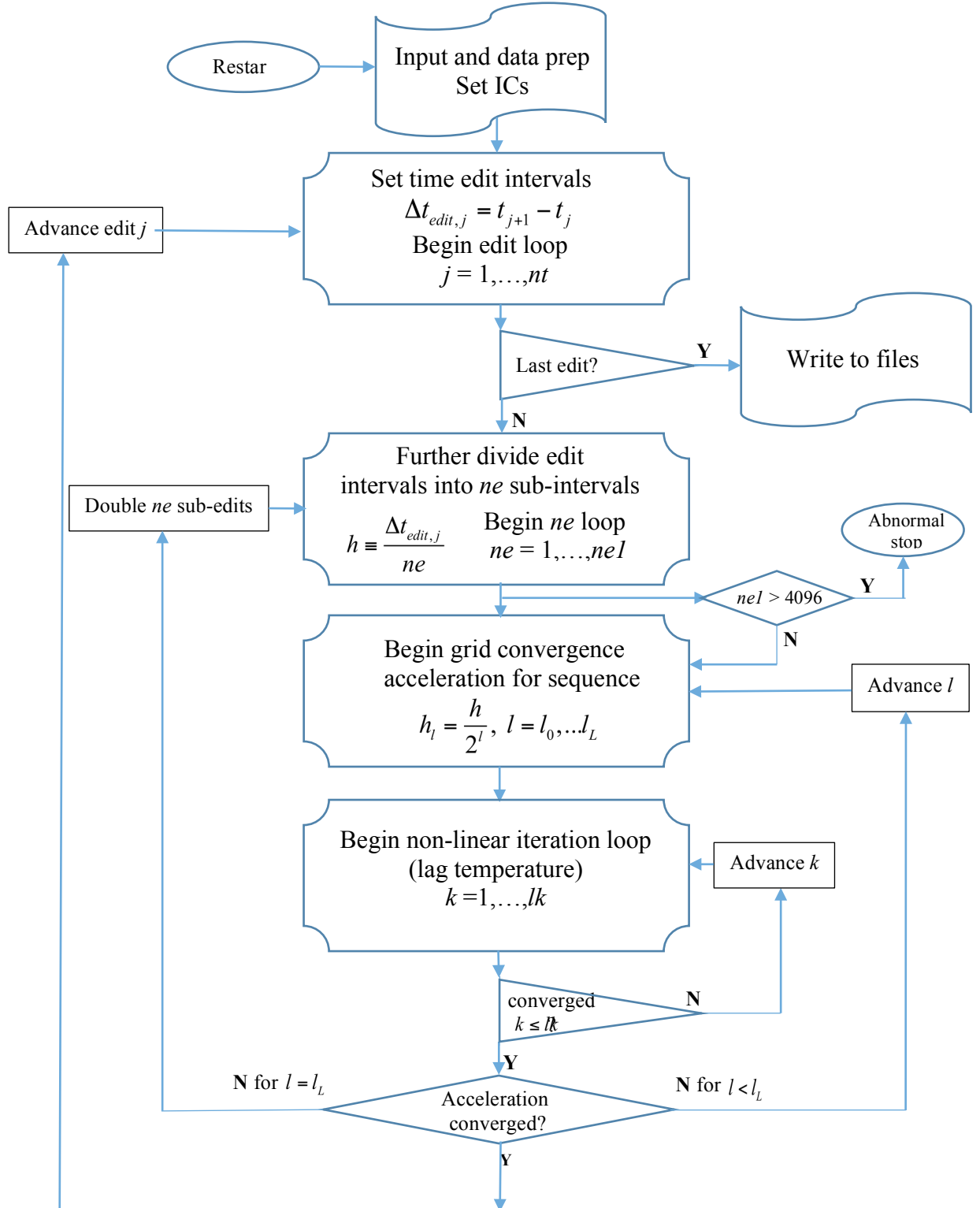


Fig. 1. Computational Flow Chart for **SKINATH/BEFD**.

Figure 1 shows the calculation flow of the **SKINATH/BEFD** algorithm. Note that there are four loops for (1)desired edits including, (2)sub-edits, (3)convergence acceleration and (4)non-linear Jacobian iteration. There is a restart option and direct input, where we specify the desired time edits. The calculation begins with the edit loop (*j-loop*). Note that edits do not have to be at regular intervals. The next loop (*ne-loop*) introduces *nel* additional sub-intervals between each edit, called sub-edits, which are accelerated to convergence as are the original edits. Converged results are recorded on the plot file (*o3.dat*). Only the originally desired edits are output to screen or table file. The next loop (*l-loop*) converges the edits over ever-finer grids through Richardsons extrapolation[3].

Within the *l-loop*, lies the iterative loop (*k-loop*), where we perform the iteration

$$\mathbf{y}_{j+1}^{k+1} = \left[ \mathbf{I} - h\mathbf{A}(t_{j+1}, T_j^k) \right]^{-1} \left[ \mathbf{y}_{j+1}^k + h\mathbf{q}(t_{j+1}) \right]. \quad (9)$$

over the Jacobian  $\mathbf{A}$ . We base convergence only on power and temperature since convergence acceleration over the finer grids applies to all dependent variables. We are now at the center of the algorithm. Several potential failures are possible as we continue Richardsons acceleration. If the *k*-iteration does not converge within 10 iterates, the flow continues to the next *l* of the grid acceleration *l-loop* in the hope that further refinements will converge. Note that even when the reactivity is linear, the iteration loop applies at least twice for each sub-edit.

On completion of the grid loop (*l-loop*), if not converged in all dependent variable, the number of sub-intervals *nel* doubles and the grid acceleration restarts. The maximum *nel* is 4096, at which time the program stops abnormally.

The sub-edit feature enables convergence timesavings. It is possible to reduce the number of grids in the acceleration by increasing *nel* at input to be greater than unity, which, in some instances, reduces the overall computational time, since the grid acceleration will converged more rapidly. In addition, if plots are required, a sufficiently large *nel* gives more plotted points and therefore a smoother plot. This, of course, is at a computational expense for very large *nel*.

Note that our results focus only the reactor power and temperature.

## **b. Intuitive benchmarks**

Before we present results, it is always best to ensure the proper operation of the computational algorithm and program through simple intuitively obvious benchmarks. The following four simple cases come from the **SKINATH/BEFD** code directly through input specification.

The first such benchmark (B1) is for a thermal reactor of initial power 1W at  $T(0) = 20^\circ\text{C}$  with kinetics parameters given in Table 2a. This case will have no imposed or feedback reactivity ( $\rho_0 = \alpha_D \equiv 0$ ) and no fission heating ( $\varepsilon \equiv 0$ ) with the results given in Table 3a. As anticipated the reactor is completely quiescent over time.

**Table 2a. Thermal reactor I  
with  $\Lambda = 5 \times 10^{-4} \text{s}$ .**

$l$	$\beta_l$	$\lambda_l(\text{s}^{-1})$
1	0.000285	0.0127
2	0.0015975	0.0317
3	0.001410	0.115
4	0.0030525	0.311
5	0.00096	1.40
6	0.000195	3.87
$\beta = 0.0075$		

**Table 2b. Result for benchmark B1.**

$t$	$N$	$T$
0.000000000E+00	1.000000000E+00	2.000000000E+01
1.000000000E -01	1.000000000E+00	2.000000000E+01
5.000000000E -01	1.000000000E+00	2.000000000E+01
1.000000000E+00	1.000000000E+00	2.000000000E+01
1.000000000E+01	1.000000000E+00	2.000000000E+01
1.000000000E+06	1.000000000E+00	2.000000000E+01

The second benchmark (B2), is for the same reactor with no imposed or Doppler feedback ( $\rho_0 = \alpha_D \equiv 0$ ) or reactor-air heat transfer [ $h_r(T(t)) \equiv 0$ ] but with fission heating ( $\varepsilon \equiv 1$ ). The results, displayed in Table 2b, show only the temperature responding as given by Eq(1a) appropriately modified to

$$\frac{dT(t)}{dt} = \frac{\varepsilon}{VC_p} N_0 \quad (10a)$$

to give

$$T(t) = T(0) + \frac{N(0)}{VC_p} t. \quad (10b)$$

For  $VC_p = 13006.194 J / kg^\circ C$ , Eq(10b) is exactly the results of Table 3.

**Table 2c. Result for benchmark B2.**

$t$	$N$	$T$
0.000000000E+00	1.000000000E+00	2.000000000E+01
1.000000000E-01	1.000000000E+00	2.000000769E+01
5.000000000E-01	1.000000000E+00	2.000003844E+01
1.000000000E+00	1.000000000E+00	2.000007689E+01
1.000000000E+01	1.000000000E+00	2.000076886E+01
1.000000000E+06	1.000000000E+00	9.688644594E+01

The next benchmark (B3) is without imposed or Doppler feedback ( $\rho_0 = \alpha_D \equiv 0$ ) and with fission heating ( $\varepsilon \equiv 1$ ); hence,

$$VC_p \frac{dT(t)}{dt} = \varepsilon N(0) - h_T(T(t)) [T(t) - T_c] \quad (11a)$$

By integration, we represent the solution as the following transcendental equation:

$$\frac{W(t)}{T_c} = \frac{N(0)}{VC_p T_c} t + \eta T_c B\left(2.25, -2, \frac{W(t)}{T_c}\right), \quad (11b)$$

where

$$\eta \equiv ahd^{0.75}$$

$$W(t) \equiv T(t) - T_c,$$

and  $B$  is the incomplete beta function

$$B(z, w, U) \equiv \int_0^U du \frac{u^{z-1}}{(1+u)^{z+w}}.$$

While one can solve this equation for the reactor temperature by iteration, which seems more convenient than the original ODE, we choose to solve the original ODE and to find the equilibrium temperature from Eq(5a) as

$$T_\infty \left[ \frac{T_\infty - T_c}{T_\infty} \right]^{1.25} = \frac{\varepsilon}{\eta} N(0), \quad (12)$$

most easily done through bisection. For  $N(0)$  of 1W, that temperature is

$$T_\infty = 21.59232730833268^\circ\text{C}. \quad (13)$$

Table 2d shows the ratio of the temperature to  $T_\infty$  and that the equilibrium temperature is reached to 9 places. Figure 2 shows the full temperature trace.

**Table 2d. Results for benchmark B3.**

$t$	$N$	$T/T_\infty$
0.000000000E+00	1.000000000E+00	9.262549476E-01
1.000000000E-03	1.000000000E+00	9.262551613E-01
1.000000000E-02	1.000000000E+00	9.262570841E-01
1.000000000E-01	1.000000000E+00	9.262763122E-01
1.000000000E-01	1.000000000E+00	9.262763122E-01
5.000000000E-01	1.000000000E+00	9.263617586E-01
1.000000000E+00	1.000000000E+00	9.264685319E-01
1.000000000E+01	1.000000000E+00	9.283799268E-01
1.000000000E+02	1.000000000E+00	9.457184536E-01
1.000000000E+03	1.000000000E+00	9.975975868E-01
5.000000000E+03	1.000000000E+00	9.999999985E-01
1.000000000E+05	1.000000000E+00	1.000000000E+00
1.000000000E+06	1.000000000E+00	1.000000000E+00



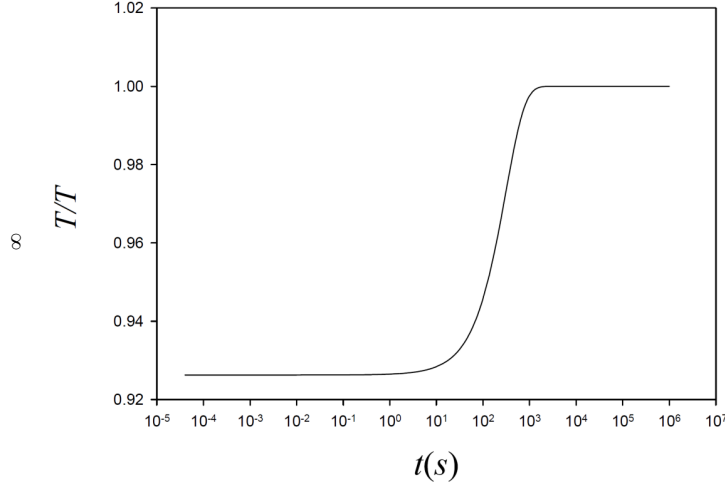


Fig. 2. Reactor temperature trace for B3.

The final benchmark (B4) is for a step change in reactivity with no temperature feedback. Table 2e gives the results. The converged accelerated powers in column 2

**Table 2e. Results for benchmark B4.**

$t$	$N_{acc}$	$N_{ori}$
0.000000000E+00	1.000000000E+00	1.000000000E+00
1.000000000E-01	2.515766141E+00	<b>2.515773324E+00</b>
5.000000000E-01	1.036253381E+01	<b>1.036282227E+01</b>
1.000000000E+00	3.218354095E+01	<b>3.218488588E+01</b>
1.000000000E+01	3.246978898E+09	<b>3.252353241E+09</b>
1.000000000E+02	2.596484647E+89	<b>2.599842087E+89</b>

are in complete agreement with the **CATS** algorithm[5], which is arguably one of the best in existence. Column 3 gives the non-converged accelerated (original) powers demonstrating how much improvement Richardsons extrapolation gives as indicated by, at maximum, only four correct digits (emboldened) with deterioration in time in comparison to the original second order BEFD solution without acceleration.

### c. SKINATH/BEFD Benchmarks

The original **SKINATH** application seems to have been an analysis of a 55-Gallon drum of fissionable waste stored in an open-air waste repository. The specific parameters for the case to be studied are in Table 1. We assume the canister— our reactor, to be at 0.01W critical power at initiation of the transient, which could be a compression or re-arrangement of the fissile material to cause a uniform reactivity insertion of  $4.3\phi$ . The Doppler coefficient is  $\alpha_D = -0.306\phi/^\circ\text{C}$  and Table 3 gives the kinetics parameters.

**Table 3. Thermal Reactor with  $\Lambda = 5 \times 10^{-5} \text{s}$ .**

$l$	$\beta_l$	$\lambda_l(\text{s}^{-1})$
1	0.00022	0.0124
2	0.00142	0.0305
3	0.00127	0.111
4	0.00257	0.301
5	0.00075	1.14
6	0.00027	3.01
$\beta = 0.0065$		

**c.1 SKINATH Sample Case**

Since the original **SKINATH** model assumes thermal equilibrium at the initiation of the transient, the temperature increase from the 0.01W fission energy is not properly accounted for as explained above. This means that there is an initial feedback mismatch since the reactor is at a slightly higher temperature than the coolant when the transient begins. As indicated above, the reactivity mismatch is exactly the amount required to bring the reactor to its initial critical state

$$\Delta\rho = -\alpha_D [T_{r0} - T_c]. \quad (14)$$

This reactivity is not apparent since it is in the control rod settings that keep the system critical and therefore has no bearing on the reactivity insertion. The feedback reactivity will always start from zero since it is relative to the reactor initial temperature, which we assume to be the air temperature. The discrepancy does however influence the heat transfer as the reactor heats up from the coolant temperature rather than its true temperature and the heat transfer coefficient non-linearly depends directly on the temperature.

We shall see how this plays out in the benchmark suite to follow.

Table 4a reproduces a portion of the original **SKINATH** output[1] and compares well with Table 4b of **SKINATH/BEFD** results. With only four places from Ref. [1] available, the comparison indicates the original **SKINATH** concept worked quite well. Also included in Table 4b is the unaccelerated (original) power, again the advantage of acceleration is quite apparent. Figures 3a,b show the short time comparison graphically. As observed, agreement is excellent.

For some unknown reason, the **SKINATH** authors did not continue further in time as shown in Fig. 3c,d and missed the interesting convergence to steady state behavior.

**Table4a. SKINATH[1] results for sample case.**

$t$	$N$	$\rho$	$T$
0.000000000E+00	1.000E-02	4.300E+00	2.000E+01
1.000000000E+00	1.391E-02	4.230E-02	2.000E+01
1.000000000E+01	1.033E-01	4.299E-02	2.000E+01
1.000000000E+02	7.946E-01	-2.950E-02	4.369E+01
2.500000000E+02	2.118E-03	6.388E-03	3.196E+01
5.000000000E+02	9.819E-01	-4.826E-03	3.563E+01
7.500000000E+02	4.033E+00	-1.858E-03	3.466E+01
1.000000000E+03	7.680E+00	1.379E-03	3.360E+01

**Table4b. SKINATH/BEDF results.**

$t$	$N_{acc}$	$N_{ori}$	$T$
0.000000000E+00	1.000000000E-02	1.000000000E-02	2.000000000E+01
1.000000000E+00	1.390664400E-02	1.390668545E-02	2.000005684E+01
1.000000000E+01	1.033074569E-01	1.033236004E-01	2.000190828E+01
1.000000000E+02	7.946137382E-01	7.967850080E-01	4.369127585E+01
2.500000000E+02	2.118680532E-03	2.117704079E-03	3.196455312E+01
5.000000000E+02	9.818854372E-01	9.826664863E-01	3.562914154E+01
7.500000000E+02	4.033472004E+00	4.034715111E+00	3.465924402E+01
1.000000000E+03	7.680950791E+00	7.675864364E+00	3.360158546E+01

It is the oscillatory behavior that makes this benchmark a useful standard. We now consider two variations of the published transient.

### c.2 SKINATH Variation 1: Insertion from Essentially Zero Power

The first is to insert 4.3¢ reactivity from essentially zero power ( $10^{-20}$ W) to bring the reactor up to the initial steady state of the original transient (0.01W). A comparison to the sample case is shown in Fig. 4a,b. Now, the reactor and coolant begin in equilibrium and the model is valid. Curiously, exactly the same steady state to 12 places is achieved, but with a very different transient behavior. This makes sense since both cases have identical reactivity insertions to be shown.

These results therefore confirm that even with a defective Doppler model, **SKINATH/BEDF** correctly predicts long time delayed critical steady state.

### c.3 SKINATH Variation 2: Insertion from Established Criticality at 0.01W

The second variation is to run the system from startup at  $10^{-20}$ W to 0.01W and then apply the prescribed reactivity insertion. This transient proceeds to first determine

the reactor temperature at 0.01W by solving Eq(5a) to give  $T_{r0} = 20.039404826178^{\circ}\text{C}$ . With this temperature, we know the amount of Doppler reactivity that exists when the reactor reaches 0.01W, which is

$$\rho_D = \alpha_D (T_w - T_c) = -7.8376199267152\text{E-}07.$$

To compensate the Doppler, we impose the negative of this reactivity through a control rod in order to bring the reactor to steady state at 0.01W. Thus

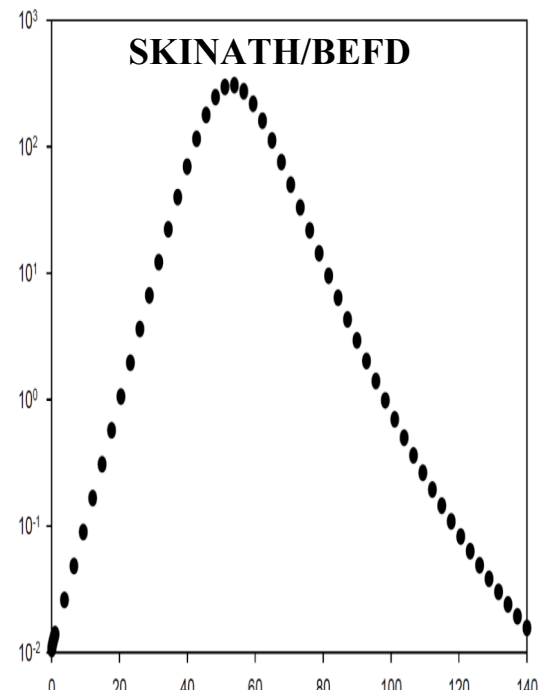
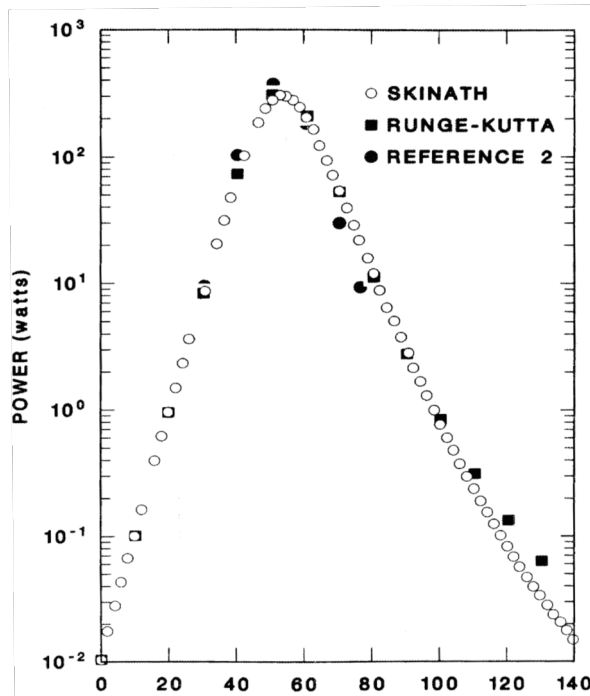
$$\rho_0 = -\rho_D. \quad (15)$$

Figures 5a,b show that this procedure works quite well as the power starts from  $10^{-20}\text{W}$  and remains at 0.01W and exactly at  $T_{r0}$  for 190 years.

Once one establishes steady state at 0.01W, the reactor is in thermal equilibrium with its fission energy and the reactor temperature is  $T_{r0}$ . At this point, say, at  $t = 10^6$  min, we insert the imposed reactivity of  $\rho_0 = \$0.043$ . The Doppler feedback becomes

$$\rho(t, T(t)) = \rho_0 - \alpha_D [T(t) - T_{r0}] \quad (15)$$

based on the actual reactor temperature. Results are shown in Figs. 6a,b. The transient is essentially identical with the sample transient of Fig 3a. There is one notable difference however. Since the Doppler feedback is based on the actual reactor temperature, the final steady state is slightly different from that of the sample case as shown in the following table:



TIME (min)

TIME (min)

Fig. 3a. Short time power trace comparison.

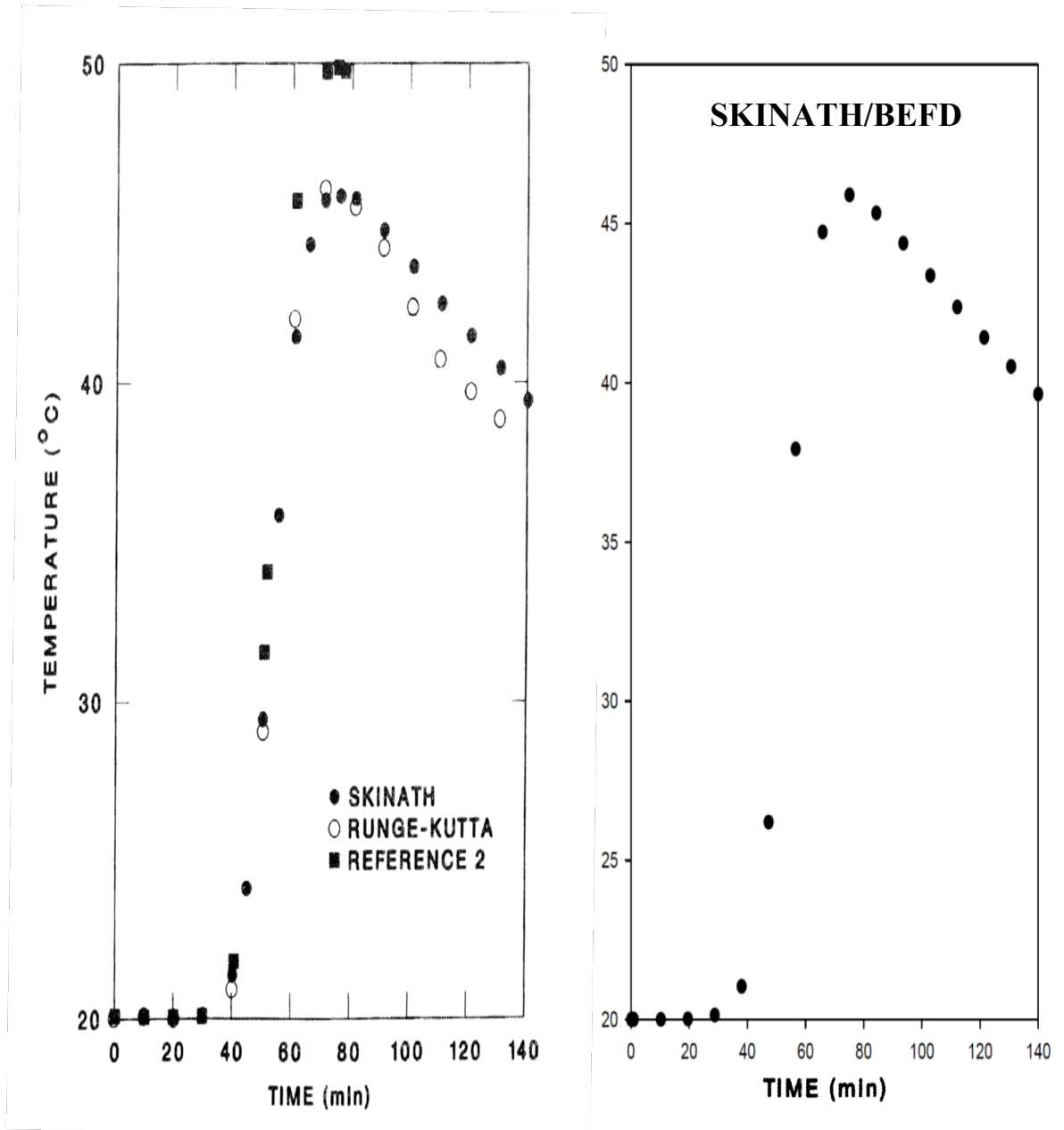


Fig. 3b. Short time temperature comparison.

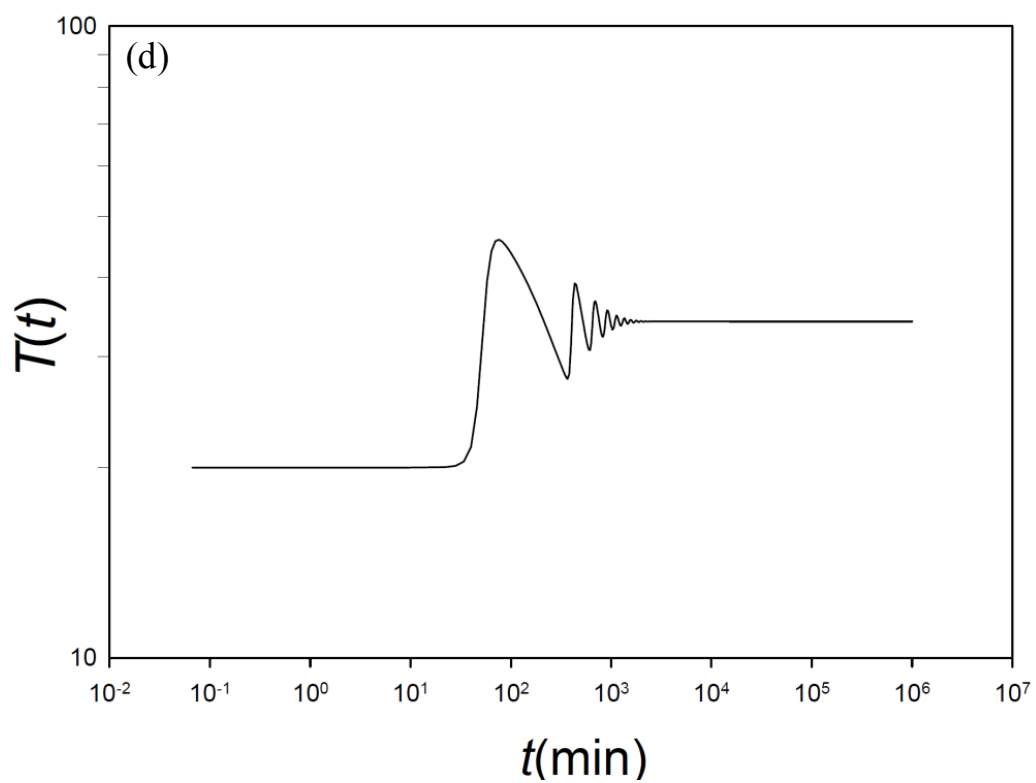
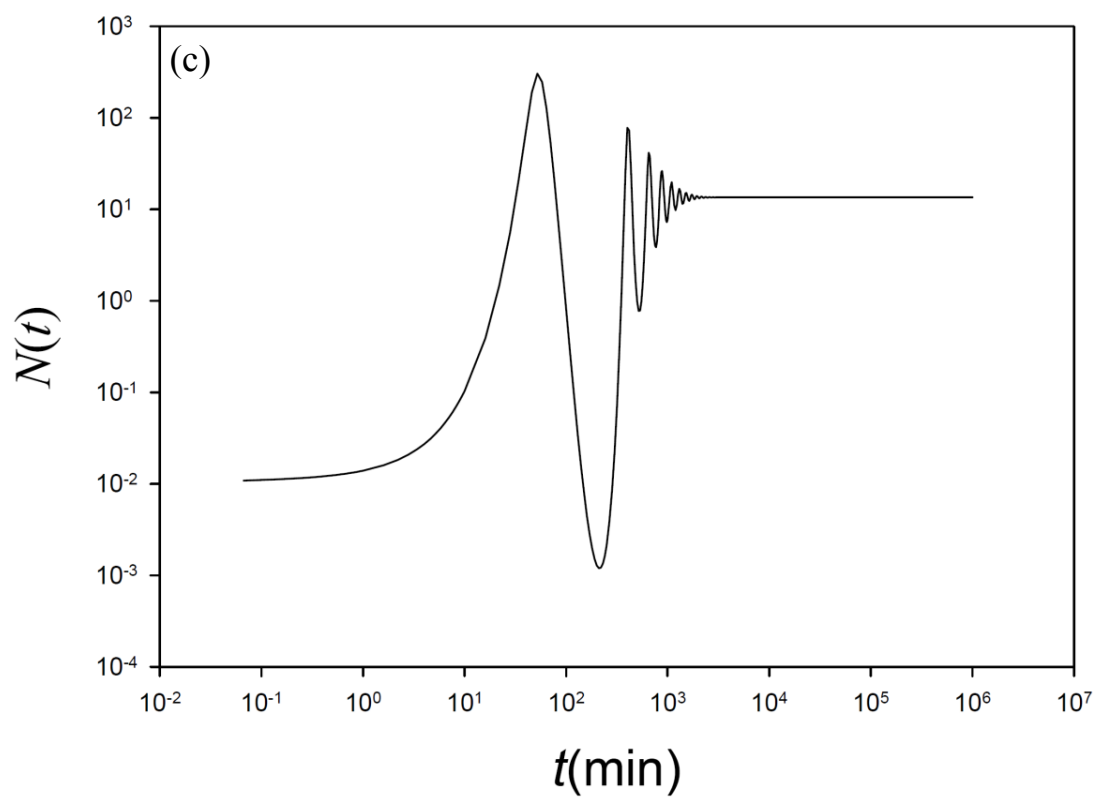


Fig.3. SKINATH sample case. (c). Power Trace  
(d) Temperature trace.

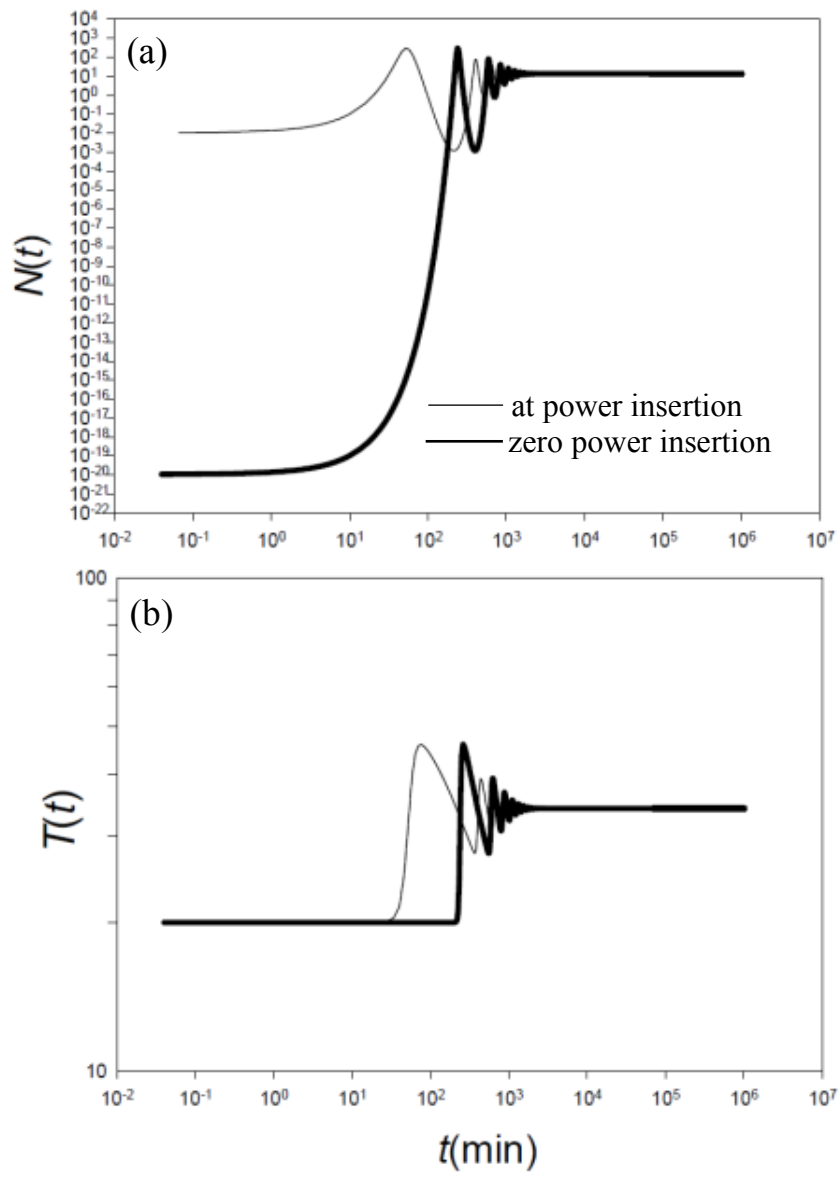


Fig. 4a,b. Comparison of near zero power and at power insertions.

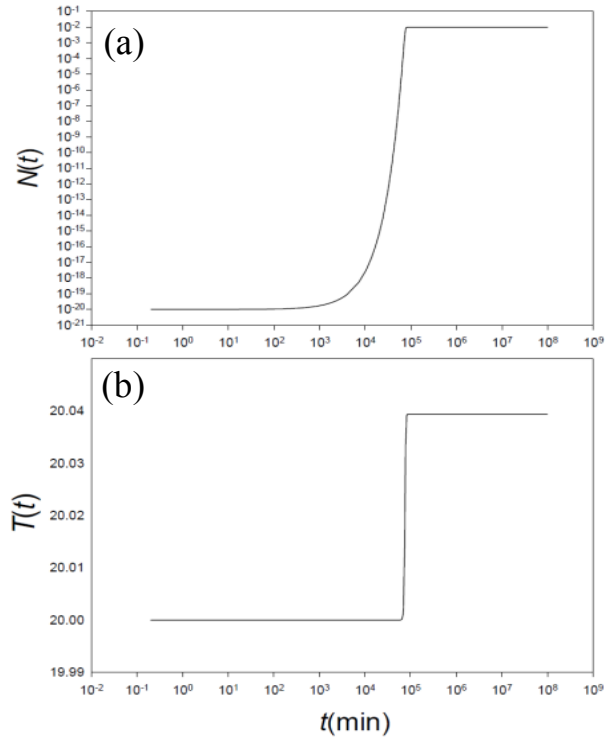


Fig. 5a,b. Startup to steady state at 0.01W.

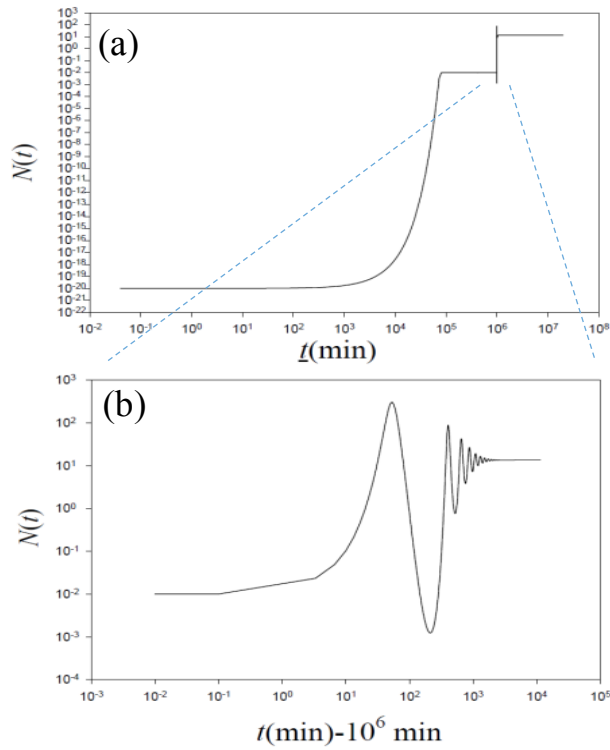


Fig. 6. (a) Run up to critical from steady state at 0.01W  
(b) Focus on the time of insertion at  $10^6$  min.



#### d. Asymptotic Power Variation

To achieve steady state the reactivity as well as the temperature time derivative must be vanish. Thus, for the original SKINATH model

$$0 = \rho_0 + \alpha_D [T_\infty - T_c] \quad (16a)$$

$$0 = \varepsilon N_\infty - h_T(T_\infty) [T_\infty - T_c] \quad (16b)$$

with

$$h_T(T_\infty) \equiv Ahd^{0.75} \left[ 1 - \frac{T_c}{T_\infty} \right]^{0.25}. \quad (16c)$$

Therefore, from Eqs(16a,b), we find

$$T_\infty = T_c - \frac{\rho_0}{\alpha_D} \quad (17a)$$

and

$$N_\infty = \frac{A}{\varepsilon} hd^{0.75} \left[ 1 - \frac{T_c}{T_c - \frac{\rho_0}{\alpha_D}} \right]^{0.25} \left[ -\frac{\rho_0}{\alpha_D} \right]. \quad (17b)$$

From Eq(17b), we see that with all other parameters fixed, the asymptotic power depends only on the reactivity and is the reason why the final power does not depend on the initial power in Fig. 4a. In addition, if we use the corrected SKINATH model, the asymptotic power would be

$$N_\infty = \frac{A}{\varepsilon} hd^{0.75} \left[ 1 - \frac{T_{r0}}{T_{r0} - \frac{\rho_0}{\alpha_D}} \right]^{0.25} \left[ T_{r0} - T_c - \frac{\rho_0}{\alpha_D} \right]. \quad (18)$$

leading to different asymptotic values as shown in Table 5.

**Table 5. Comparison of steady states for the two Doppler formulations.**

	$N(t)$	$T(t)$
Doppler from $T_c$	1.357318123E+01W	3.405228758E+01°C
Doppler from $T_{r0}$	1.356925739E+01W	3.409169241E+01°C

Figures 7a,b graphically show that the asymptotic power and fuel temperature are indeed independent of initial power.

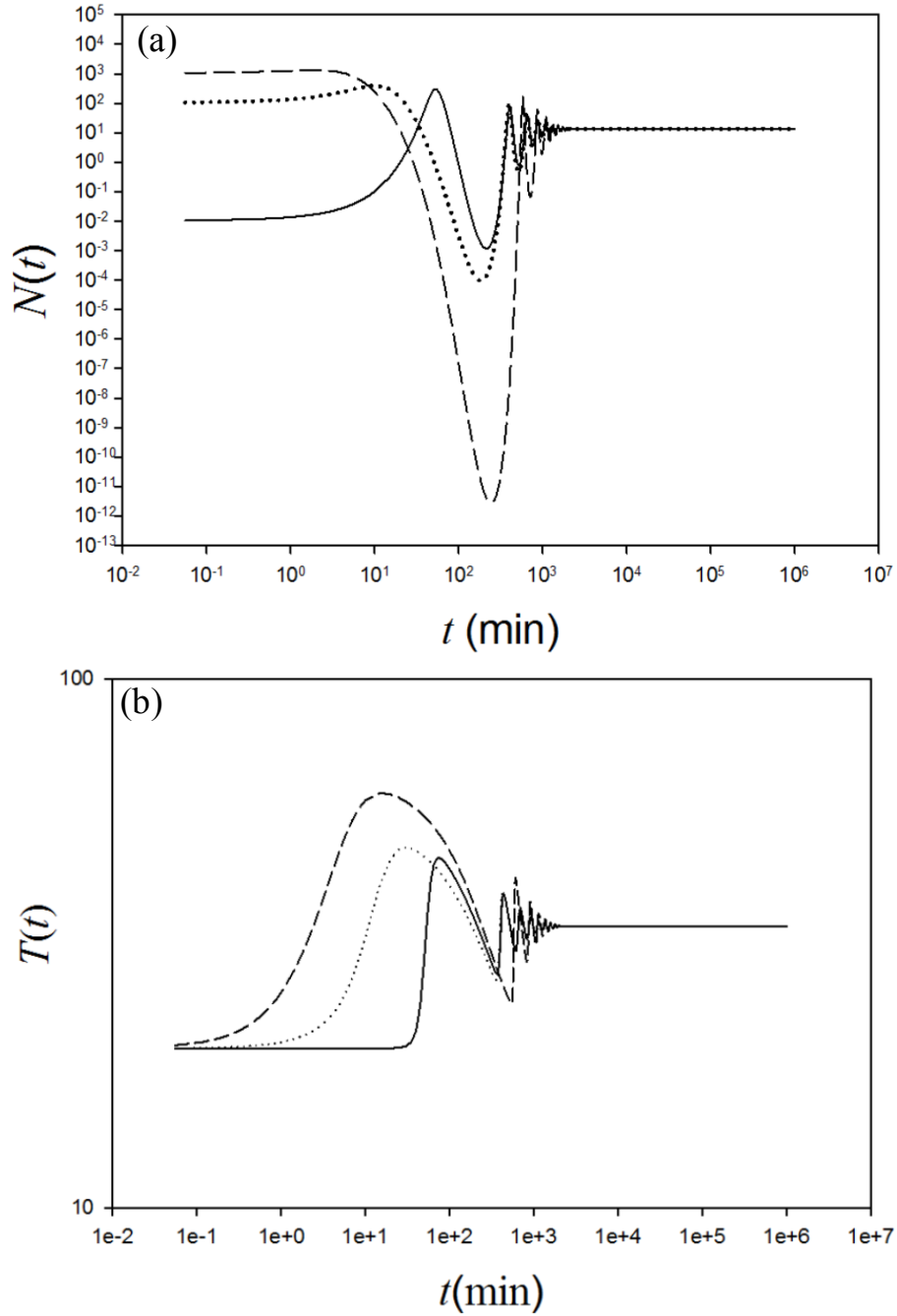


Fig. 7a,b Demonstration of independence of asymptotic

power and fuel temperature from initial power.

A theoretical asymptotic power also enables a check of the algorithm's precision as shown in Table 6, where the agreement is better than expected.

**Table 6. Comparison of asymptotic values.**

	$N(t)$	$T(t)$
SKINATH?BEFD	1.357318123E+01W	3.405228758E+01°C
Doppler from $T_{r0}$	1.3573181229E+01W	3.40522875817E+01°C

## REFERENCES

1. H.L. Dodds and R.M. Westfall, ORNL/CSD/TM- 210, **SKINATH**, 1984.
2. A.C. Hindmarsh, LSODE and LSODIACM, Signum Newsletter, V15, No.4, 1980.
3. B. Ganapol, *Annals of Nuclear Energy*, V62, pp 564-571, 2013.
4. J.A. da Nobrega, *Nucl. Sci.& Eng.* 46, 366-375, 1971.
5. B. Ganapol, PHYSOR 2012, Knoxville, TN, CD-ROM, American Nuclear Society, LaGrange Park, IL, 2012.

## Interim Progress Report A3 (Su16)

### HTR-10 Multiphysics Point Kinetics Benchmark

B. Ganapol  
INL Visiting Staff  
Ganapol@cowboy.ame.arizona.edu

M. DeHart, F. Gleicher, J. Ortensi, S. Schunert and R. Martineau  
INL Staff

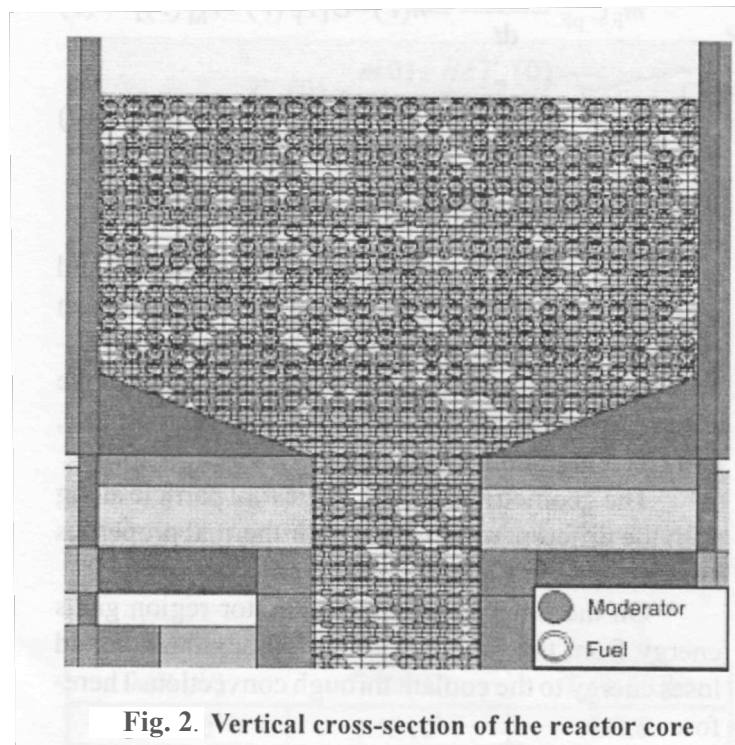
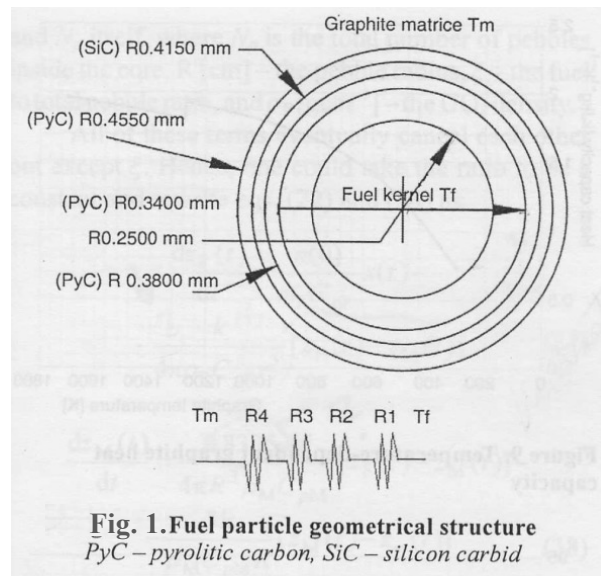
#### I. Benchmark Scope

In this multiphysics benchmark, we consider the HTR-10 PBM reactor[1], which was one of the first of its kind. The reactor achieved full power in Tsinghua China in 2003. The core is doubly heterogeneous, where the fuel pebble, roughly the size of a billiard-ball, contains fuel particles, which themselves contain the fuel kernel encased in pyrolytic carbon and silicon carbide coatings as shown in Fig. 1. The graphite matrix of the pebble serves as the moderator as well as the kernel binder. The pebbles then fill a cylindrical core, shown in Fig.2, and slowly move down through the core to eventually drop out the bottom as another enters from the top. The kernels are randomly distributed within the pebbles, which, in turn, are randomly arranged within the core including additional non-fuel pebbles to achieve the desired critical power level. Table 1 gives the reactor primary design parameters.

**Table 1. 10MW Reactor Primary Design Parameters[2].**

Initial power	$N_0$	10MW
Initial fuel temperature	$T_{F0}$	853°C
Initial moderator temperature	$T_{M0}$	827.6°C
Constant coolant temperature	$T_\infty$	748°C
Kernel radius	$r$	0.00025m
Pebble radius	$R$	0.03m
Fuel kernels/pebble	$N_{fp}$	8335
Avg. fuel specific heat	$C_{pF}$	316.1063 J/kg°K
Avg. moderator specific heat	$C_{pM}$	1855.9 J/kg°K
Fuel/Moderator heat transfer coefficient*	$h$	657.7343 W/°Km <sup>2</sup>
Fuel to dummy pebble ratio	$\xi$	0.57
Fuel density	$\rho_F$	10400 kg/m <sup>3</sup>
Moderator density	$\rho_M$	1730 kg/m <sup>3</sup>

\* Seems that in Ref. [2], the temperature giving  $h$  is incorrectly taken as 748°K rather than 748°C.



The multiphysics model to simulate the initial critical experiment[3] includes transient power variation from Doppler feedback from both the fuel and moderator. The solution to the coupled set of dynamic equations results from Taylor series representations analogous to the **CATS** algorithm[4].

This benchmark is part of a semi-analytical multiphysics benchmarking effort recently initiated at INL to be part of the Heavy Testing QNA Suite (HTQNAS) for the RATTLESNAKE transport code.

### **I.1 The Reactor Kinetics/Heat Transfer Model**

The reactor kinetics model begins with the PKEs for the power and precursor concentrations  $N(t)$  and  $C_l(t)$

$$\begin{aligned} \frac{dN(t)}{dt} &= \left[ \frac{\rho(t, T_F, T_M) - \beta}{\Lambda} \right] N(t) + \sum_{l=1}^m \lambda_l C_l(t) + q(t) \\ \frac{dC_l(t)}{dt} &= \frac{\beta_l}{\Lambda} N(t) - \lambda_l C_l(t), \quad l = 1, \dots, m, \end{aligned} \quad (1a,b)$$

for  $m$  delayed neutron groups, which, here, will be either one or six. The initial power is  $N(0)$  to give the initial precursor concentrations

$$C_l(0) = \frac{\beta_l}{\lambda_l \Lambda} N(0), \quad l = 1, \dots, m. \quad (1c)$$

$\rho(t, T_F, T_M)$  is the reactivity, depending upon both fuel and moderator temperatures. All other symbols have their usual meaning

$\Lambda$	Neutron generation time
$\beta_l$	Delayed neutron yield for group $l$
$\beta$	$\sum_{l=1}^m \beta_l$
$\lambda_l$	Delayed group decay constant.

We assume Doppler feedback reactivity from both the fuel and moderator to be of the form

$$\rho(t, T_F, T_M) = \rho_0(t) + \alpha_F [T_F - T_F(0)] + \alpha_M [T_M - T_M(0)] \quad (1d)$$

where  $\alpha_F$  and  $\alpha_M$  are known the temperature reactivity coefficients.

Table 2 gives the reactor kinetics parameters.

**Table 2. Precursor yields and decay constants.**

$l$	$\beta_l$	$\lambda_l(s)$
1	0.000285	0.0127
2	0.0015975	0.0317
3	0.001410	0.115
4	0.0030525	0.311
5	0.00096	1.40
6	0.000195	3.87
Total yield		$\beta$ 0.0075
Neutron life time		$\Lambda$ 0.00168s
Fuel reactivity coefficient[2]		$\alpha_F$ $-1.9 \cdot 10^{-5}/^\circ\text{K}$
Moderator reactivity coefficient[2]		$\alpha_M$ $-15.7 \cdot 10^{-5}/^\circ\text{K}$

The fuel and moderator temperatures obey the following heat balance equations:

-- between the fuel and moderator

$$m_F C_{pF} \frac{dT_F(t)}{dt} = N(t) - U [T_F(t) - T_M(t)], \quad (2)$$

--between the moderator and flowing Helium coolant

$$m_M C_{pM} \frac{dT_M(t)}{dt} = U [T_F(t) - T_M(t)] - hA [T_M(t) - T_\infty] \quad (3)$$

respectively. The total contact area between all pebbles and coolant is

$$A = 4\pi R^2 N_p, \quad (4a)$$

where  $N_p$  is the total number of pebbles. The fuel and moderator masses are

$$\begin{aligned} m_F &= \frac{4}{3} \pi r^3 \rho_F N_{pf} N_p \xi \\ m_M &= \frac{4}{3} \pi R^3 N_p \rho_M \end{aligned} \quad (4b,c)$$

respectively.

The only properties left to determine are  $U$  and  $hA$  for the given initial power  $N(0)$ . These are most easily found by noting that transient initiation occurs at thermal equilibrium

$$\begin{aligned} 0 &= N(0) - U [T_F(0) - T_M(0)] \\ 0 &= U [T_F(0) - T_M(0)] - hA [T_M(0) - T_\infty] \end{aligned}$$

to give

$$\begin{aligned} U &= \frac{N(0)}{[T_F(0) - T_M(0)]} \\ hA &= U \frac{[T_F(0) - T_M(0)]}{[T_M(0) - T_\infty]} = \frac{N(0)}{[T_M(0) - T_\infty]}. \end{aligned} \quad (5a,b)$$

Finally, given  $h$ , one can find the total number of pebbles using Eq(4a) since

$$hA = \frac{N(0)}{[T_M(0) - T_\infty]} = 4\pi R^2 N_p h \Rightarrow N_p = \frac{N(0)}{4\pi R^2 h [T_M(0) - T_\infty]} \quad (6)$$

We are now ready to describe the numerical algorithm, which we call **HTRCATS**.

## I.2 Taylor Series (TS) Formulation of the Solution

We recast the model equations as a vector ODE in the time interval  $t_{j-1} \leq t \leq t_j$



$$\frac{d\mathbf{y}_j(t)}{dt} = \mathbf{A}_j(t, T_F, T_M) \mathbf{y}_j(t) + \mathbf{q}_j(t), \quad (7a)$$

with

$$\begin{aligned} \mathbf{y}_j(t) &\equiv [N_j(t) \quad C_{1j}(t) \quad \dots \quad C_{mj}(t) \quad T_{Fj}(t) \quad T_{Mj}(t)]^T \\ \mathbf{q}_j(t) &\equiv \left[ \frac{d\rho_{0j}(t)}{dt} \quad 0 \quad \dots \quad 0 \quad hAT_\infty \right]^T. \end{aligned} \quad (7b,c)$$

The Jacobian,  $\mathbf{A}_j$ , of size  $m+3$ , is

$$\mathbf{A}_j(t, T_F, T_M) \equiv \begin{bmatrix} \left( \rho(t, T_{Fj}, T_{Mj}) - \beta \right) / \Lambda & \lambda_1 & \dots & \lambda_m & 0 & 0 \\ \beta_1 / \Lambda & -\lambda_1 & 0 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \beta_m / \Lambda & \dots & \dots & -\lambda_m & \dots & 0 \\ 0 & 0 & \dots & 0 & -U / m_F C_{pF} & U / m_F C_{pF} \\ 0 & 0 & \dots & 0 & U / m_M C_{pM} & -(U + hA) / m_M C_{pM} \end{bmatrix} \quad (7c)$$

and the initial conditions are

$$\mathbf{y}(0) = \left[ N(0) \quad \frac{\beta_1}{\lambda_1 \Lambda} N(0) \quad \dots \quad \frac{\beta_m}{\lambda_m \Lambda} N(0) \quad T_F(0) \quad T_M(0) \right]^T. \quad (7e)$$

The proposed solution is the following vector Taylor series in the time interval  $t_{j-1} \leq t \leq t_j$ :

$$\mathbf{y}_j(t) = \sum_{n=0}^{\infty} \mathbf{y}_{n,j-1} (t - t_{j-1})^n. \quad (8a)$$

If, in addition, one expresses the Jacobian matrix as the matrix Taylor series

$$\mathbf{A}_j(t, T_F, T_M) = \sum_{n=0}^{\infty} \mathbf{a}_{n,j-1} (t - t_{j-1})^n, \quad (8b)$$

when introduced into the product term of the RHS of Eq(7a), there results

$$\mathbf{A}_j(t, T_F, T_M) \mathbf{y}_j(t) = \sum_{n=0}^{\infty} \sum_{l=0}^n \mathbf{a}_{n-l} \mathbf{y}_{l,j-1} (t - t_{j-1})^n. \quad (8c)$$

Assuming a Taylor series for the source vector, the series coefficients derive from equating the coefficients of the powers of  $(t - t_{j-1})^n$

$$\mathbf{y}_{n+1,j-1} = \frac{1}{n+1} \left[ \sum_{l=0}^n \mathbf{a}_{n-l} \mathbf{y}_{l,j-1} + \mathbf{q}_{n,j}(t) \right]. \quad (9)$$

Expressing the Jacobian as a Taylor series is not particularly limiting since this form includes power series representations; however, it is not as general as the **BEFD** formulation[5].

For this benchmark, the only nonlinear term is in the NW corner, element  $a_{1,1}$ . For this term, we require the Taylor series in the interval  $[t_j, t_{j-1}]$  for

$$\begin{aligned} \vartheta_j(t, T_F, T_M) \equiv \rho_j(t, T_F, T_M) - \beta &= \rho_{0j}(t) - \beta + \alpha_F [T_{Fj}(t) - T_F(0)] + \\ &+ \alpha_M [T_{Mj}(t) - T_M(0)]. \end{aligned}$$

Considering each term, we have

$$\begin{aligned} T_{Fj}(t) - T_F(0) &= \sum_{n=0}^{\infty} [T_{Fn,j-1} - T_F(0) \delta_{k,0}] (t - t_{j-1})^n \\ T_{Mj}(t) - T_M(0) &= \sum_{n=0}^{\infty} [T_{Mn,j-1} - T_M(0) \delta_{k,0}] (t - t_{j-1})^n \\ \rho_{0,j}(t) - \beta &= \sum_{n=0}^{\infty} [\rho_{0n,j-1} - \beta \delta_{n,0}] (t - t_{j-1})^n. \end{aligned}$$

Hence, for uniform step insertion,  $\rho_0$ ,

$$\vartheta_j(t, T_F, T_M) = \sum_{n=0}^{\infty} \vartheta_{n,j}(t, T_F, T_M) (t - t_{j-1})^n, \quad (10a)$$

where

$$\begin{aligned} \rho_{0,j}(t, T_F, T_M) &\equiv \rho_0 - \beta - (\alpha_F T_F(0) + \alpha_M T_M(0)) + \\ &\quad + \alpha_F T_{F0,j-1} + \alpha_M T_{M0,j-1} \end{aligned} \quad (10b)$$

$$\rho_{n,j}(t, T_F, T_M) \equiv \alpha_F T_{Fn,j-1} + \alpha_M T_{Mn,j-1}, \quad n \geq 1. \quad (10c)$$

All other entries in  $A_j$  are

$$a_{n,j} \equiv \{a_{i,j} \delta_n; i, j = 1, m+3\}.$$

Therefore, in explicit expanded form

$$N_{n,j-1} = \frac{1}{n} \left\{ \frac{1}{\Lambda} \left[ \left\{ \rho_0 - \beta - (\alpha_F [T_{F0,j-1} - T_F(0)] + \alpha_M [T_{M0,j-1} - T_M(0)]) \right\} N_{n-1,j-1} + \right] + \sum_{l=0}^{n-2} (\alpha_F T_{Fn-1-l,j-1} + \alpha_M T_{Mn-1-l,j-1}) N_{l,j-1} + \sum_{m=1}^M \lambda_m C_{m,n-1,j-1} \right\} \quad (11a)$$

$$C_{m,n,j-1} = \frac{1}{n} \left\{ \frac{\beta_m}{\Lambda} N_{n-1,j-1} + \lambda_m C_{m,n-1,j-1} \right\}, m = 1, \dots, M \quad (11b)$$

$$T_{F,n,j-1} = \frac{1}{n} \left\{ \frac{1}{m_F C_{pF}} N_{n-1,j-1} - \frac{U}{m_F C_{pF}} [T_{Fn-1,j-1} - T_{Mn-1,j-1}] \right\} \quad (11c)$$

$$T_{M,n,j-1} = \frac{1}{n} \left\{ \frac{U}{m_m C_{pM}} [T_{Fn-1,j-1} - T_{Mn-1,j-1}] - \frac{hA}{m_m C_{pM}} [T_{Mn-1,j-1} - T_\infty \delta_{n,1}] \right\}. \quad (11d)$$

Thus, from the initial conditions, we find all subsequent Taylor series coefficients and the solution is a true analytical solution, not in closed form, but analytical nevertheless. Therefore, the true numerical solution is the limit

$$y_j(t) = \lim_{N \rightarrow \infty} \sum_{n=0}^N y_{n,j-1} (t - t_{j-1})^n. \quad (12)$$

Using the concept of convergence acceleration, in the form of Wynn-epsilon ( $W-e$ ) and Richardsons ( $R$ ) extrapolations[6], one can realize the limit to high precision.

### I.3 Numerical Implementation and Verification

#### a. Numerical Implementation(See [7])

#### b. Selected Benchmarks: Constant Specific Heats

The four benchmarks we consider are those found in Ref. [2] for step reactivity insertions of  $\rho_0 = 0.05\%$ ,  $0.25\%$ ,  $0.50\%$  and  $1.0\%$  with constant specific heats and the 10MW design parameters of Table 1.

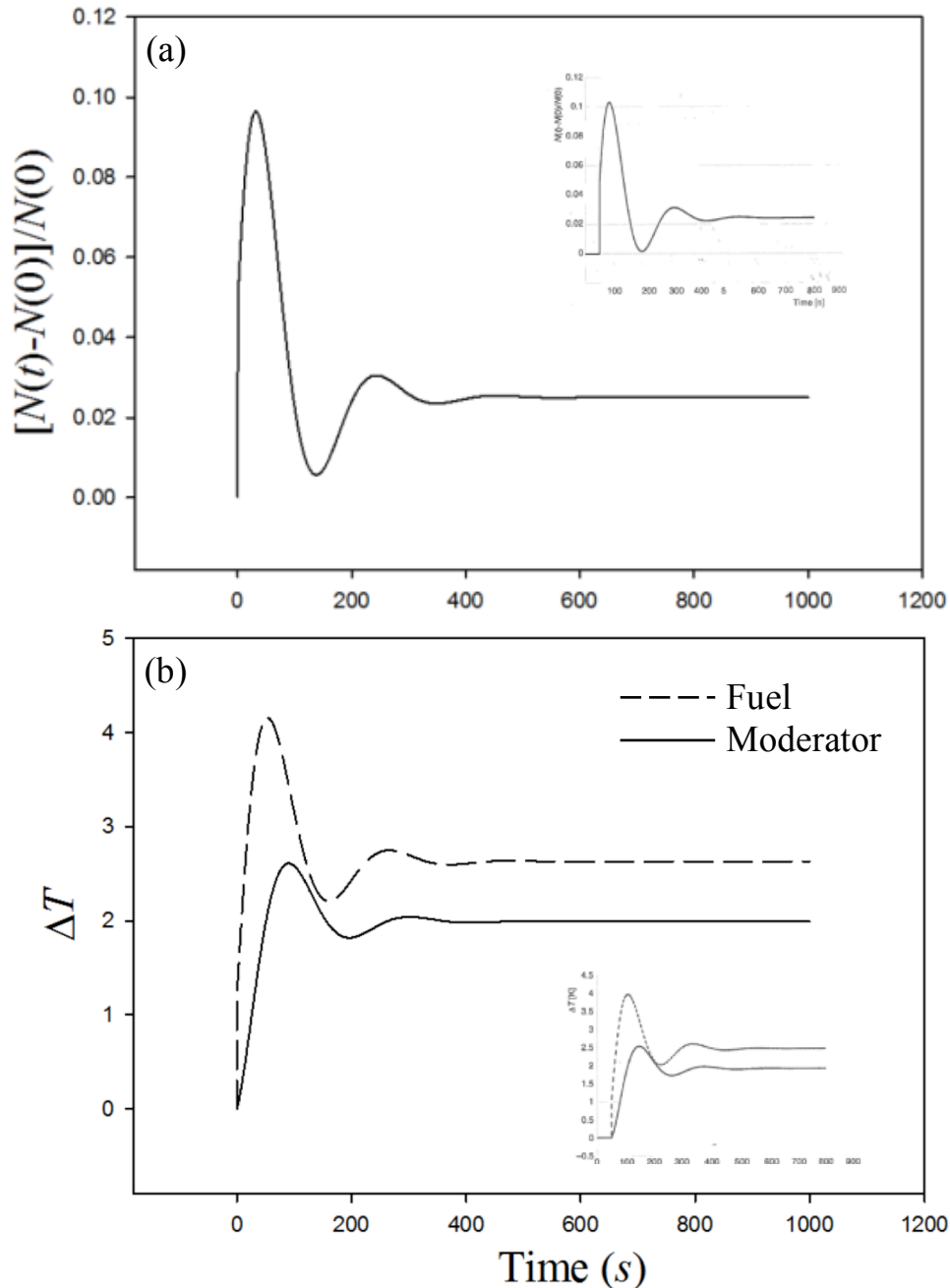


Fig. 3. Reactor response for 0.05\$ reactivity insertion with one delayed group. (a)  $[N(t)-N(0)]/N(0)$ , (b)  $\Delta T_F, \Delta T_M$ .

Figure 3 shows the response of the HTR-10 reactor for a 0.05\$ step reactivity insertion for the one delayed group parameters of Ref [2]  $\beta = 0.00726$  and  $\lambda = 0.08s$ .

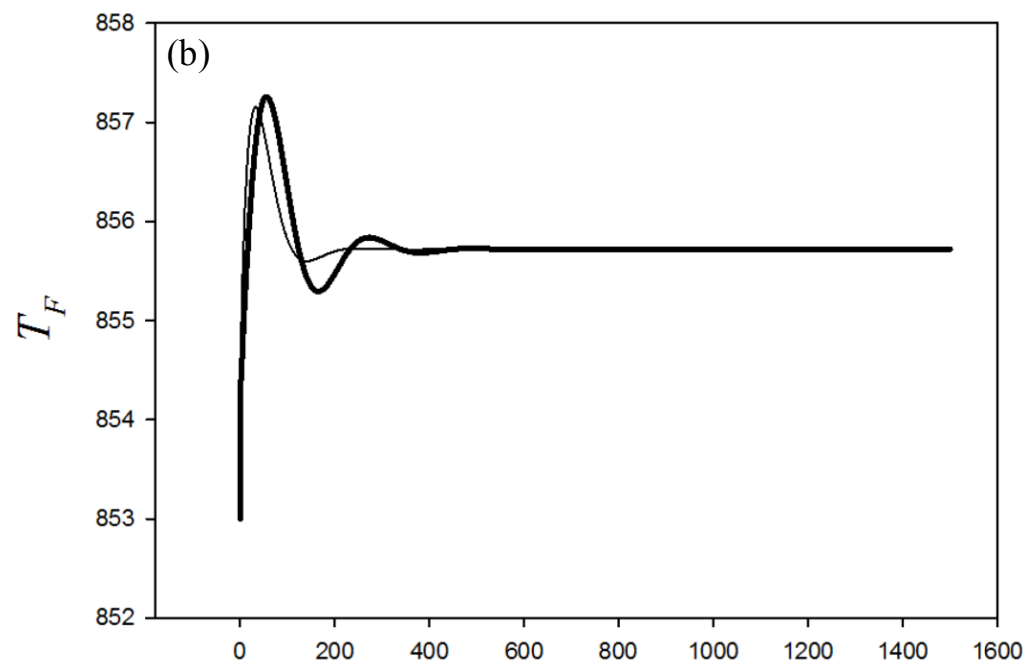
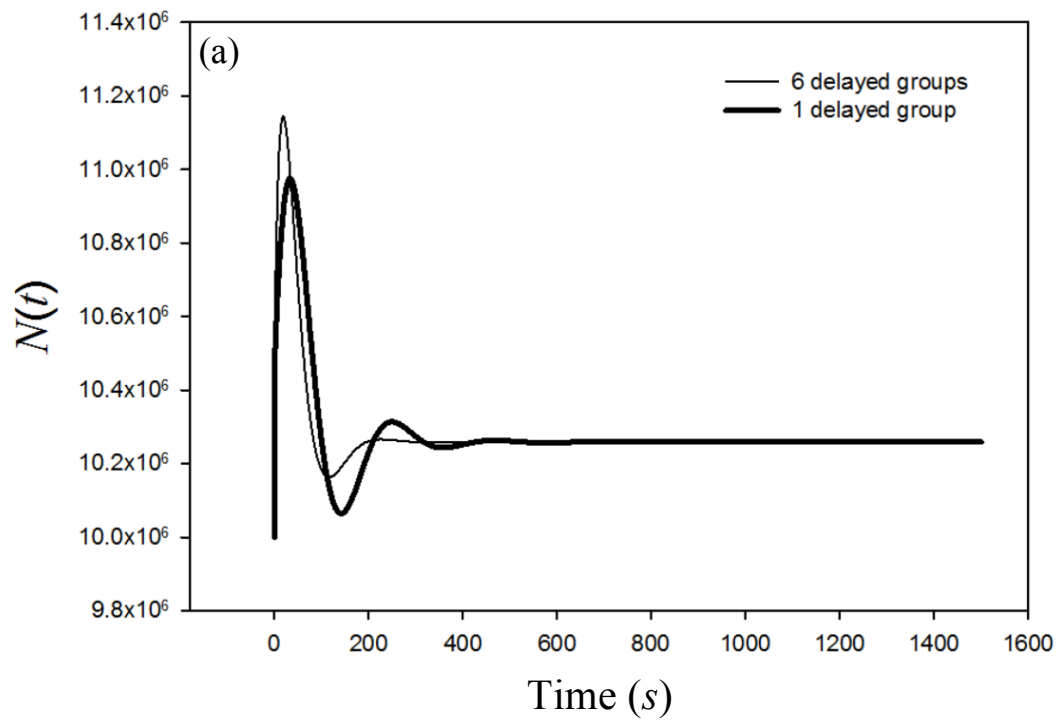
This figure is included to show the close agreement of the **HTRCATS** algorithm results of Ref. [2] [Inset]. The peaks are not identical, which may be due to an alternative definition of the overall heat transfer coefficient, since how  $U$  is determined is not clear in Ref. [2]. We may also explain the discrepancy by computational error— though this may be a stretch if MATLAB™ was used in [2].

The analysis presented above differs from that of Ref. [2] in that it is for the full complement of delayed group. We now observe the difference between one- and six- group models.

For an insertion of 0.05\$, the power, fuel and moderator temperature transients are shown in Figs.4a,b,c for both one and six delayed group(s). For a consistent one delayed group using the kinetic parameters of Table 2, we calculate  $\lambda$  as

$$\lambda = \frac{\beta}{\sum_{l=1}^6 \frac{\beta_l}{\lambda_l}} = 0.0784s \quad (13)$$

since we know  $\beta$ .



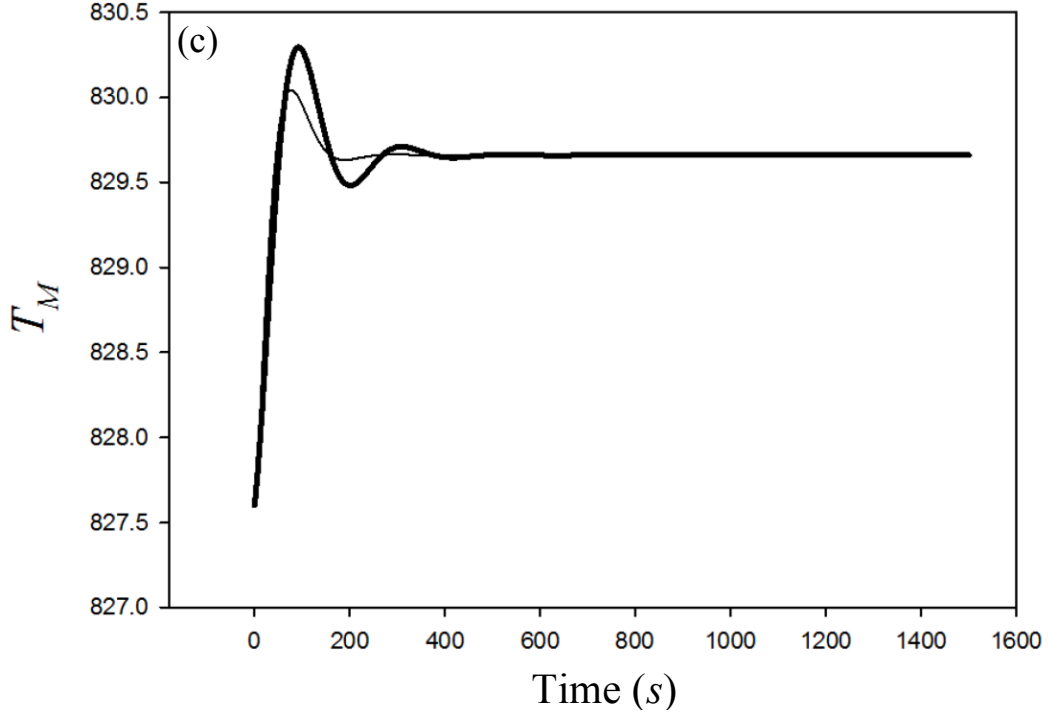


Fig. 4. Reactor response for 0.05\$ reactivity insertions (a) $N$ , (b) $T_F$ , (c) $T_M$  for one and six delayed group(s).

It seems the 6-group model gives a higher maximum power and lower maximum temperatures. Apparently, more neutrons are available earlier to add power in the fuel more quickly and possibly increase the overall heat transfer to the Helium coolant, thus reducing the temperature shortly thereafter. Interestingly however, the asymptotic steady state powers for both cases are virtually identical to 10 significant digits. One can show this theoretically by the following analysis.

To achieve asymptotic steady state, the reactivity must be zero as time approaches infinity, implying

$$0 = \rho_0 + \alpha_F [T_{F\infty} - T_F(0)] + \alpha_M [T_{M\infty} - T_M(0)]. \quad (14a)$$

In addition, the derivatives of both  $T_F$  and  $T_M$  must vanish

$$0 = N_\infty - U[T_{F\infty} - T_{M\infty}] \quad (14b)$$

$$0 = U[T_{F\infty} - T_{M\infty}] - hA[T_{M\infty} - T_\infty]. \quad (14c)$$

Thus, we have three equations and three unknowns, whose solution is

$$\begin{aligned}
T_{F\infty} &= -\frac{\alpha_M h A T_\infty + (hA + U)(\rho_0 - \alpha_F T_F(0) - \alpha_M T_M(0))}{\alpha_M U + \alpha_F (hA + U)} \\
T_{M\infty} &= -\frac{-\alpha_F h A T_\infty + U(\rho_0 - \alpha_F T_F(0) - \alpha_M T_M(0))}{\alpha_M U + \alpha_F (hA + U)} \\
N_\infty &= U(T_{F\infty} - T_{M\infty}).
\end{aligned} \tag{15a,b,c}$$

Further, since [Eqs(5a,b)]

$$U = \frac{N(0)}{[T_F(0) - T_M(0)]} \tag{16a}$$

and

$$hA = \frac{N(0)}{[T_M(0) - T_\infty]}, \tag{16b}$$

from Eqs(15a,b), the asymptotic temperatures are independent of initial power and from Eq(14b)

$$\frac{N_\infty}{N(0)} = \frac{T_{F\infty} - T_{M\infty}}{T_F(0) - T_M(0)}. \tag{16c}$$

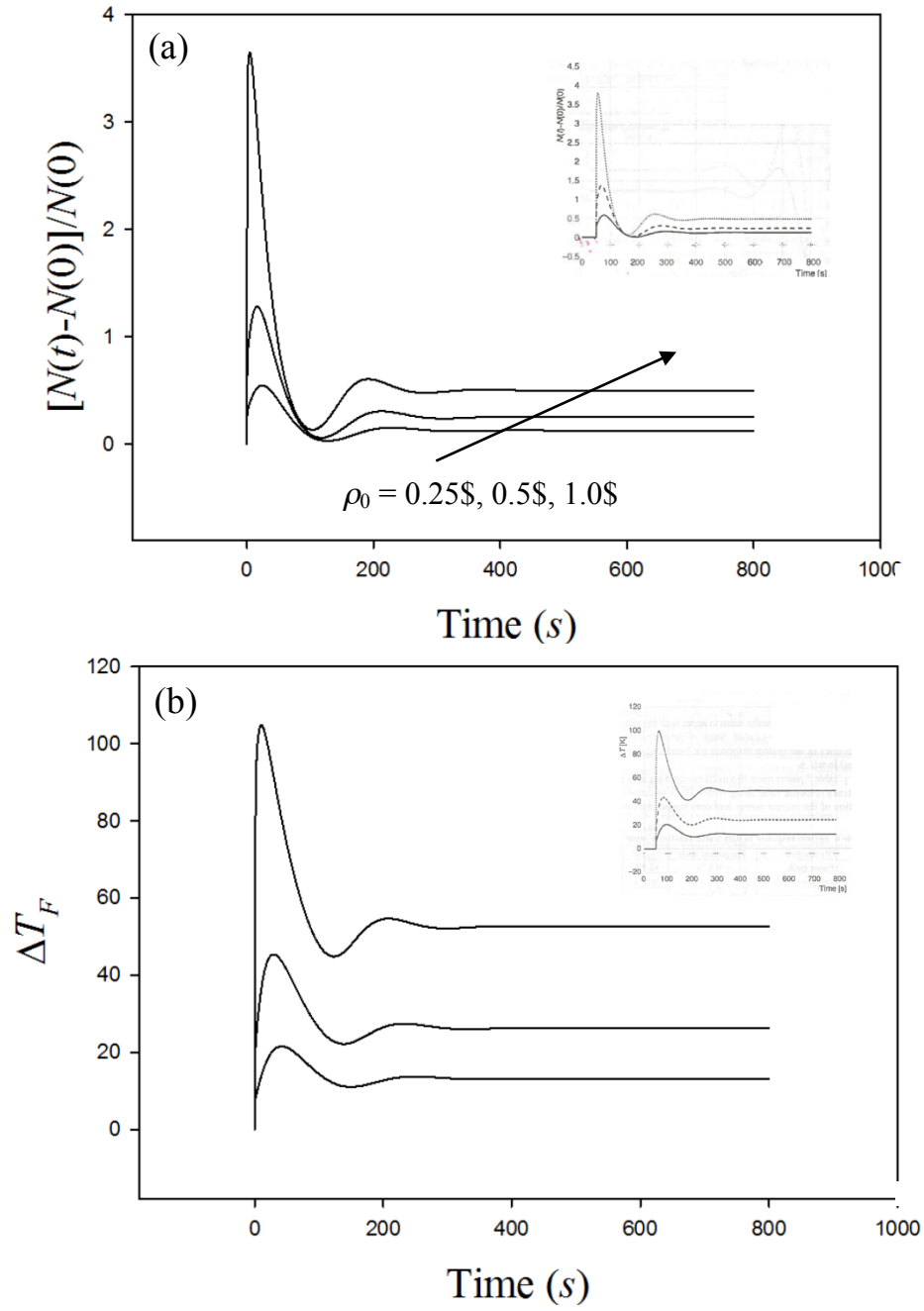
Table 3 gives 9-place power and temperatures to enable a more in-depth comparison.

**Table 3. Benchmark values for 0.05\$ insertion: HTRCATS.**

$t$	$N$	$T_F$	$T_M$
1.0000E-01	1.018126588E+07	8.532928123E+02	8.276002533E+02
1.0000E+00	1.057092032E+07	8.544541557E+02	8.276212887E+02
1.0000E+01	1.105600477E+07	8.561103996E+02	8.280332728E+02
1.0000E+02	1.018167186E+07	8.558189728E+02	8.299569650E+02
5.0000E+02	1.025876570E+07	8.557170606E+02	8.296597956E+02
1.0000E+03	1.025875987E+07	8.557169788E+02	8.296597288E+02
1.5000E+03	1.025875988E+07	8.557169788E+02	8.296597287E+02



Figure 5 shows the reactor response for insertions of 0.25\$, 0.5\$ and 1.0\$ for the one delayed group of Ref [2]. The match to Ref [2] {Inset} is remarkably similar.



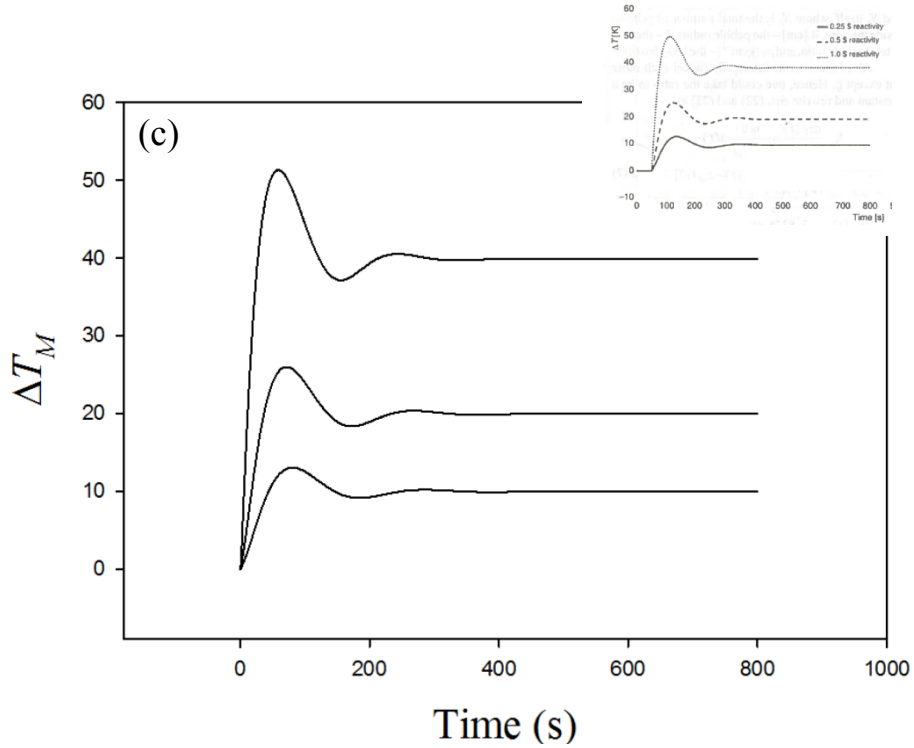


Fig. 5. Reactor response for reactivity insertions for kinetics parameters of Ref. [2].  
(a)  $[N(t)-N(0)]/N(0)$ , (b)  $\Delta T_F$ , (c)  $\Delta T_M$ .

Table 4, comparing the **HTRCATS** results with the asymptotic values of Eqs(15), is included to show the high accuracy achieved by the **HTRCATS** algorithm. While we report 10 significant digits, the algorithm can give even higher precision.

**Table 4. Comparison of HTRCATS and asymptotic values of Eqs(15).**

$N$	$T_F$	$T_M$	
Fig.3/1-group			
$\rho_0 = 0.05\$$			
1.025875988E+07	855.7169788	829.6597287	<b>HTRCATS</b>
1.0258759885E+07	855.71697878	829.65972868	Asymp.
Fig 4/6-group			
$\rho_0 = 0.25\$$			
1.125239784 E+07	866.1501773	837.5690868	<b>HTRCATS</b>
1.1252397841E+07	866.15017733	837.56908682	Asymp.
$\rho_0 = 0.5\$$			
1.250479568 E+07	879.3003547	847.5381736	<b>HTRCATS</b>
1.2504795683E+07	879.30035467	847.538173638	Asymp.
$\rho_0 = 1.0\$$			
1.500959137 E+07	905.6007093	867.4763473	<b>HTRCATS</b>

1.5009591366E+07	905.60070935	867.47634727	Asymp.
------------------	--------------	--------------	--------

Fig. 7/6-group

$\rho_0 = 0.25\$$

1.129379942 E+07	866.5848939	837.8986434	HTRCATS
1.1293799423E+07	866.58489394	837.89864341	Asymp.

$\rho_0 = 0.5\$$

1.258759885 E+07	880.1697879	848.1972868	HTRCATS
1.2587598846E+07	880.16978789	848.19728682	Asymp.

$\rho_0 = 1.0\$$

1.517519769 E+07	907.3395758	868.7945736	HTRCATS
1.5175197693E+07	907.339576	868.79457363	Asymp.

On a final note, it must be mentioned that the total number of pebbles for this reactor is calculated to be 16,888 from Eq(6), which is in remarkable agreement with Ref. [3] quoting 16,890.

### c. Selected Benchmarks: Variable specific heats

Of course in a real simulation of the HTR10, the specific heat of the fuel and moderator depend strongly on their respective temperatures. Therefore for a proper simulation, we require a model of the temperature dependence of  $\text{UO}_2$  and graphite, which one finds to be

-- for  $\text{UO}_2$  [8] with  $T_F$  in the range [298.15°K, 3120°K]

$$C_{pF}(T_F) = \frac{1000}{267} \left[ 52.1743 + 87.951\xi - 84.2411\xi^2 + 31.542\xi^3 - 2.6334\xi^4 - \frac{0.71391}{\xi^2} \right]$$

$$\xi \equiv \frac{T_F}{1000}$$

(17a)

-- for graphite[2] with  $T_M$  in the range [298°K, 1273°K]

$$C_{pM}(T_M) = 10^3 \left[ 1.131 + 6.62 \times 10^{-4} T_M - 9.9691 \times 10^{-8} T_M^2 \right]. \quad (17b)$$

and in the range [1273°K, 3273°K]

$$C_{pM}(T_M) = 10^3 \left[ 2.031 + 7.8645 \times 10^{-5} T_M^2 - 4.2671 \times 10^{-5} T_M^{-1} + 1.3203 \times 10^{-8} T_M^{-3} - 1.199 \times 10^{-10} T_M^{-4} \right]. \quad (17c)$$

The Jacobian now becomes

$$A(t, T_F, T_M) \equiv \begin{bmatrix} (\rho(t, T_F, T_M) - \beta) / \Lambda & \lambda_1 & \dots & \lambda_m & 0 & 0 \\ \beta_1 / \Lambda & -\lambda_1 & 0 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \beta_m / \Lambda & \dots & \dots & -\lambda_m & \dots & 0 \\ 1 / m_F C_{pF}(T_F) & 0 & \dots & 0 & -U / m_F C_{pF}(T_F) & U / m_F C_{pF}(T_F) \\ 0 & 0 & \dots & 0 & U / m_M C_{pM}(T_M) & -(U + hA) / m_M C_{pM}(T_M) \end{bmatrix}, \quad (18)$$

with the addition of four non-linear terms in the bottom two rows. A Taylor series solution is no longer possible because the heat capacities do not admit a Taylor series. Hence, we turn to the **BEFD** algorithm[5]. Refs [5] and [9] give the detail of the numerical implementation.

Tables 5a,b, give benchmark powers correct to the last place quoted with (a) and without (b)  $C_p$  temperature variation as determined by **HTRBEFD**. One notes that Table 5a is identical (to all places) to Table 3 from **HTRCATS** indicating that both algorithms are highly precise. Table 5b gives benchmark values when  $C_p$  varies with temperature. As required, regardless of the variation of  $C_p$ , the identical

**Table 5a. Benchmark values for 0.05\$ insertion with uniform specific heats: HTRBEFD.**

$t$	$N$	$T_F$	$T_M$
0.000000000E+00	1.000000000E+07	8.530000000E+02	8.276000000E+02
1.000000000E-01	1.018126588E+07	8.532928123E+02	8.276002533E+02
1.000000000E+00	1.057092032E+07	8.544541557E+02	8.276212887E+02
1.000000000E+01	1.105600477E+07	8.561103996E+02	8.280332728E+02
1.000000000E+02	1.018167186E+07	8.558189728E+02	8.299569650E+02
5.000000000E+02	1.025876570E+07	8.557170606E+02	8.296597956E+02
1.000000000E+03	1.025875987E+07	8.557169788E+02	8.296597288E+02
1.500000000E+03	1.025875988E+07	8.557169788E+02	8.296597287E+02

**Table 5b. Benchmark values for 0.05\$ insertion with non-uniform specific heats: HTRBEFD.**

$t$	$N$	$T_F$	$T_M$
0.000000000E+00	1.000000000E+07	8.530000000E+02	8.276000000E+02
1.000000000E-01	1.018126783E+07	8.532924373E+02	8.276002524E+02
1.000000000E+00	1.057093225E+07	8.544540891E+02	8.276212420E+02
1.000000000E+01	1.105636431E+07	8.561104720E+02	8.280324490E+02
1.000000000E+02	1.018192527E+07	8.558209050E+02	8.299582485E+02

5.000000000E+02	1.025876634E+07	8.557170622E+02	8.296597956E+02
1.000000000E+03	1.025875987E+07	8.557169788E+02	8.296597288E+02
1.500000000E+03	1.025875988E+07	8.557169788E+02	8.296597287E+02

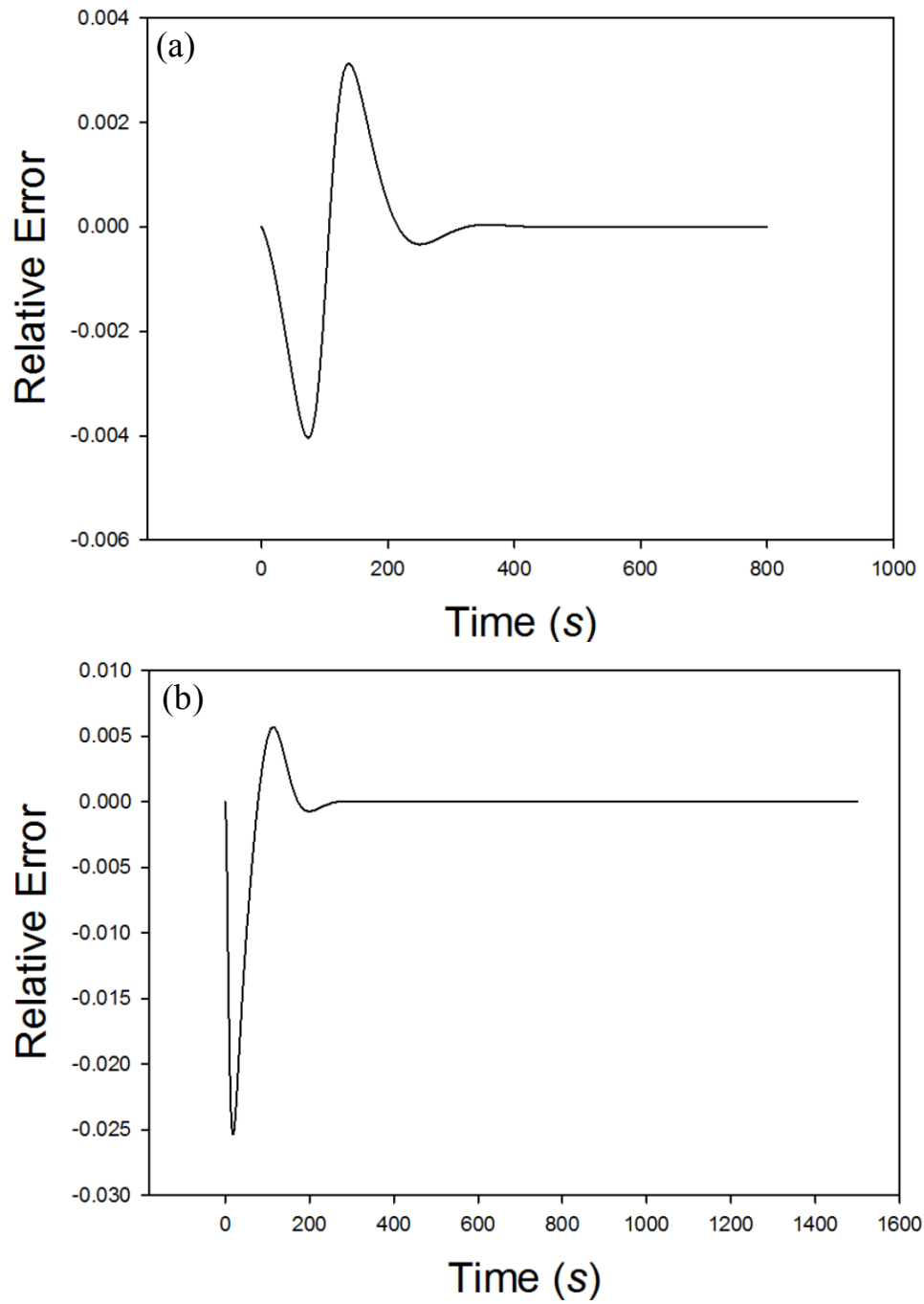


Fig. 7b. Variation of power with and without uniform specific heats.

asymptotic analysis above holds.

Figures 7a,b show the relative error with and without variation of the specific heats for 0.05\$ and 1.0\$ insertions. Because the temperature difference in the fuel and moderator is less than 5°C, including specific heat variation only gives at most a relative error of 0.5%. For the stronger transient of 1.0\$, the greatest relative error is about 2.5%. These low relative errors confirm that using uniform specific heats is a reasonable assumption, also a conclusion stated in ref. [2].

## REFERENCES

1. Z. Wu, D. Lin, D. Zhong, The design features of the HTR-10, Nuclear Engineering and Design, 218 pp. 25–32, 2002.
2. S. Hosseini Dynamic Behavior of the HTR-10 Reactor: Dual Temperature Feedback Model, NT&RT, VXXX, No. 2, pp 83-163. 2015.
3. William K. Terry, et.al., Evaluation of the HTR-10 Reactor as a Benchmark for Physics Code QA, PHYSOR-2006, ANS Topical Meeting on Reactor Physics, 2006.
4. B. Ganapol, et. al, The Solution of the Point Kinetics Equations via Converged Accelerated Taylor Series (CATS), PHYSOR 2012, Knoxville, TN, ANS Topical Meeting on Reactor Physics, 2012.
5. B. Ganapol, A Highly Accurate Algorithm for the Solution of the Point Kinetics Equations, *Annals of Nuclear Energy*, V62, pp 564-571, 2013.
6. A. Sidi, *Practical Extrapolations Methods*, Cambridge University Press, Cambridge, 2003.
7. B. Ganapol, et. al., Keepin Multiphysics Point Kinetics Benchmark, Interim Progress Report A1, Su2016.
8. J.K. Fink, Thermophysical Properties of Uranium Dioxide, Journal of Nuclear Materials 279 pp.1-18, 2000.
9. B. Ganapol, et.al., SKINATH Multiphysics Point Kinetics Benchmark, Interim Progress Report A2 INL, Su2016.

# ACTIVITIES AT INL

Jim E. Morel

August 15-19, 2016

1. *NDA for Weighted Least-Squares  $S_n$  in Rattlesnake.*

Participants: Yaqi Wang, Mark DeHart, Hans Hammer

Hans has been implementing a weighted least-squares  $S_n$  formulation in Rattlesnake with nonlinear diffusion acceleration (NDA). The weighted method was implemented because we found that poor results were obtained in certain instances with large material discontinuities unless weighting was applied. We found that the best weight function was simply  $1\sigma_t$  except in near voids where we set a minimum  $\sigma_t$  for weighting purposes only. Interestingly, if one simply uses a weight function of  $1\sigma_t$  everywhere, one obtains the SAAF equation. Thus our least-squares can now be thought of as a modified SAAF method whereby the equations remain SPD with voids. Hans is still seeing a few bugs in the NDA implementation, but he should be able to fix them soon.

2. *Non-local Diffusion Coefficients*

Participants: Yaqi Wang, Mark DeHart, Jijie Lou, Hans Hammer

Jijie Lou has implemented my method for generating nonlocal diffusion coefficients [1, 2] in a first-order  $S_n$  research code for purposes of doing linear DSA with voids. Hans also implemented this method in Rattlesnake for doing NDA. Jijie is currently investigating the impact of voids on the effectiveness of DSA. It is well known that large discontinuities in cross sections can degrade the effectiveness of DSA [3]. However, in my opinion, the model problem that has been used within the community for Fourier analysis of this effect is not representative of typical problems. One of the speculations based upon Fourier analysis seen in the literature is that the degree of degradation is essentially a function only of the factor by which the cross section changes across the interface. So for instance, cross sections of  $10^3 \text{ cm}^{-1}$  and  $10^{-3} \text{ cm}^{-1}$  should yield the same degradation as cross sections of  $1 \text{ cm}^{-1}$  and  $10^6 \text{ cm}^{-1}$ . Our computational testing with voids conclusively shows that this is not true. Voids yield an infinite factor, thus complete degradation (no acceleration) is expected. While we

generally do see some degradation, it is never severe and sometimes minimal. Jijie is modifying her code to be able to approximately mock up the standard model problem to see if we do in fact get the effects predicted by Fourier analysis. This will help to verify our overall results. We expect to write a journal article on this work this fall.

### 3. *Comparisons of FEDSMG and SERPENT Cross Sections*

Participants: Yaqi Wang, Mark DeHart, Jijie Lou

This summer Jijie Lou performed comparisons of FEDSMG [4] and SERPENT cross sections for a simple 2-D pin cell problem. SERPENT does a full pin-cell k-eigenvalue calculation using continuous-energy Monte Carlo and computes region-averaged flux spectra used to do the multigroup averaging. The FEDSMG method first uses trivial escape cross sections to generate high-resolution region-dependent semi-infinite-medium scalar flux solutions called snapshots. The snapshots are then used to generate energy groups with discontinuous support using an optimization technique that seeks to minimize the variation of all the snapshots over the groups. Next, the snapshots are used to generate group-dependent basis functions and associated piecewise constant weight functions. Finally, the basis and weight functions are used in a finite-element procedure to generate the final FEDSMG cross sections, which can be used by standard multigroup diffusion and transport codes. The FEDSMG solutions compared very favorably with SERPENT solutions even though the FEDSMG cross sections were generated without a pin-cell calculation. The FEDSMG method can easily be extended to very large numbers of groups whereas SERPENT can require impractically large CPU times as the number of groups is increased. We intend to continue comparisons with SERPENT on more complicated problems. We also intend to investigate the use of continuous-energy Monte Carlo to generate energy-dependent escape cross sections with the expectation that they might yield significantly improved results but cost much less to generate than the full flux spectra. Our goal in developing the FEDSMG method is to obtain a method that begins to converge long before the group widths become small enough to resolve the resonances. Our hope is that we can obtain the required level of accuracy with  $O(10^3)$  groups as opposed to the  $O(10^5)$  groups needed to resolve the resonances in nuclear fuel. We currently have some data generated by Andrew Till (Jijie's predecessor) indicating that the FEDSMG method has the potential to do this, but more investigation is required.

### 4. *Nonlocal Diffusion Coefficients*

Participants: Yaqi Wang, Hans Hammer



Yaqi asked me to look at deriving a nonlocal diffusion coefficient taking anisotropic scattering into account. A preliminary derivation follows.

At various points throughout this derivation we will make substitutions or eliminate terms that are small relative to the scalar flux in the diffusion limit. For future reference, note that in the diffusion limit, the scalar flux is  $O(1)$ , and the current and the gradient operator are both  $O(\epsilon)$ .

We begin the derivation with the monoenergetic diffusion equation in an infinite medium:

$$\vec{\Omega} \cdot \vec{\nabla} \psi + \sigma_t \psi = \frac{\sigma_s}{4\pi} \phi + \frac{3\sigma_1}{4\pi} (J_x \mu + J_y \eta + J_z \xi) + \frac{q}{4\pi} . \quad (1)$$

Next we integrate Eq. (1) over all angles to obtain

$$\vec{\nabla} \cdot \vec{J} + \sigma_t \phi = \sigma_s \phi + q . \quad (2)$$

The divergence of the current is  $O(\epsilon^2)$ , so we drop it from Eq. (2) to obtain

$$\sigma_t \phi = \sigma_s \phi + q . \quad (3)$$

Substituting from Eq. (3) into Eq. (1), we get

$$\vec{\Omega} \cdot \vec{\nabla} \psi + \sigma_t \psi = \frac{\sigma_t}{4\pi} \phi + \frac{3\sigma_1}{4\pi} (J_x \mu + J_y \eta + J_z \xi) . \quad (4)$$

Using standard integral transport theory to solve Eq. (2) for the current, we get

$$\begin{aligned} \vec{J}(\vec{r}_0) = \int_{4\pi} \int_0^\infty \vec{\Omega} \left\{ \frac{\sigma_t}{4\pi} \phi(\vec{r}) + \right. \\ \left. \frac{3\sigma_1(\vec{r})(\vec{r})}{4\pi} (J_x(\vec{r})\mu + J_y(\vec{r})\eta + J_z(\vec{r})\xi) \right\} \exp \left( - \int_0^\infty \sigma_t(\vec{r}') ds' \right) ds d\Omega , \end{aligned} \quad (5)$$

where  $\vec{r} = \vec{r}_0 - s\vec{\Omega}$ . We next expand the scalar flux to first order and the current to zeroth order about  $s = 0$ .

$$\begin{aligned} \vec{J}(\vec{r}_0) = \int_{4\pi} \int_0^\infty \vec{\Omega} \left\{ \frac{\sigma_t}{4\pi} \left( \phi(\vec{r}_0) - \frac{\partial \phi}{\partial x}(\vec{r}_0) s \mu - \frac{\partial \phi}{\partial y}(\vec{r}_0) s \eta - \frac{\partial \phi}{\partial z}(\vec{r}_0) s \xi \right) + \right. \\ \left. \frac{3\sigma_1}{4\pi} (J_x(\vec{r}_0)\mu + J_y(\vec{r}_0)\eta + J_z(\vec{r}_0)\xi) \right\} \exp \left( - \int_0^\infty \sigma_t ds' \right) ds d\Omega , \end{aligned} \quad (6)$$

The justification for this comes from the fact that higher order terms in the expansions are no less than  $O(\epsilon^2)$ . Note that there are three integrals that appear in Eq. (6):

$$A = \sigma_t \exp \left( - \int_0^\infty \sigma_t ds' \right) ds , \quad (7)$$

$$B = s \sigma_t \exp \left( - \int_0^\infty \sigma_t ds' \right) ds , \quad (8)$$

and

$$C = \sigma_1 \exp \left( - \int_0^\infty \sigma_t ds' \right) ds , \quad (9)$$

It is easily shown that  $A = 1$ . It is also easily shown that for the case of constant cross sections,  $B = \frac{1}{\sigma_t}$  and  $C = \frac{\sigma_1}{\sigma_t}$ .

Without loss of generality, we next compute  $J_x$  (as opposed to all components) assuming a homogeneous medium:

$$J_x = -\frac{1}{3\sigma_t} \frac{\partial \phi}{\partial x} + \frac{\sigma_1}{\sigma_t} J_x . \quad (10)$$

Solving Eq. (10) for  $J_x$  yields the standard result:

$$J_x = -\frac{1}{3(\sigma_t - \sigma_1)} \frac{\partial \phi}{\partial x} . \quad (11)$$

In the case of an inhomogeneous medium, a significant complication arises in that integral  $C$  cannot be analytically evaluated nor assumed to be independent of  $\vec{\Omega}$ , so  $J_x$  can become coupled to  $J_y$  and  $J_z$ . This yields a  $3 \times 3$  system for the current components. We can simplify matters a bit by first realizing that the scalar flux term in Eq. (6) always drops out:

$$\begin{aligned} \vec{J}(\vec{r}_0) = \int_{4\pi} \int_0^\infty \vec{\Omega} \left\{ -\frac{\sigma_t}{4\pi} \left( \frac{\partial \phi}{\partial x}(\vec{r}_0) s \mu + \frac{\partial \phi}{\partial y}(\vec{r}_0) s \eta + \frac{\partial \phi}{\partial z}(\vec{r}_0) s \xi \right) + \right. \\ \left. \frac{3\sigma_1}{4\pi} \left( J_x(\vec{r}_0) \mu + J_y(\vec{r}_0) \eta + J_z(\vec{r}_0) \xi \right) \right\} \exp \left( - \int_0^\infty \sigma_t ds' \right) ds d\Omega , \end{aligned} \quad (12)$$

Next we can do an integration by parts on the integral associated with the scalar flux derivatives:

$$\vec{J}(\vec{r}_0) = \int_{4\pi} \int_0^\infty \vec{\Omega} \left\{ -\frac{1}{4\pi} \left( \frac{\partial \phi}{\partial x}(\vec{r}_0) \mu + \frac{\partial \phi}{\partial y}(\vec{r}_0) \eta + \frac{\partial \phi}{\partial z}(\vec{r}_0) \xi \right) + \right.$$

$$\frac{3\sigma_1}{4\pi} \left( J_x(\vec{r}_0)\mu + J_y(\vec{r}_0)\eta + J_z(\vec{r}_0)\xi \right) \Big\} \exp \left( - \int_0^\infty \sigma_t ds' \right) ds d\Omega , \quad (13)$$

Finally, we re-express Eq. (13) as follows:

$$\vec{J} = -\mathbf{H} \vec{\nabla} \phi + \mathbf{G} \vec{J} , \quad (14)$$

where

$$H_{i,j} = \int_{4\pi} \int_0^\infty \vec{\Omega}_i \vec{\Omega}_j \frac{1}{4\pi} \exp \left( - \int_0^\infty \sigma_t ds' \right) ds d\Omega , \quad (15)$$

and

$$G_{i,j} = \int_{4\pi} \int_0^\infty \vec{\Omega}_i \vec{\Omega}_j \frac{3\sigma_1}{4\pi} \exp \left( - \int_0^\infty \sigma_t ds' \right) ds d\Omega , \quad (16)$$

Solving Eq. (14) for  $\vec{J}$ , we get

$$\vec{J} = -\mathbf{D} \vec{\nabla} \phi , \quad (17)$$

where the diffusion tensor is given by

$$\mathbf{D} = [\mathbf{I} - \mathbf{G}]^{-1} \mathbf{H} . \quad (18)$$

As we know from previous results for isotropic scattering,  $\mathbf{H}$  is the Eddington tensor associated with a solution to the following equation:

$$\vec{\Omega} \cdot \vec{\nabla} \psi + \sigma_t \psi = \frac{1}{4\pi} . \quad (19)$$

It is not difficult to see that  $\mathbf{G}$  is the Eddington tensor associated with a solution to the following equation:

$$\vec{\Omega} \cdot \vec{\nabla} \psi + \sigma_t \psi = \frac{3\sigma_1}{4\pi} . \quad (20)$$

Thus for the anisotropic case we must basically do twice the work of the isotropic case by solving two pure-absorber problems rather than one.

Given Eq. (11), a simple alternative is to use our current algorithm for generating tensor diffusion coefficients by simply substituting the transport-corrected total cross section for the total cross section. This certainly makes sense for NDA and DSA purposes since we generally only keep the diagonal components of the diffusion tensor anyway. Whether this is adequate when using the nonlocal tensor diffusion theory as a substitute for transport theory is a topic that should be investigated.

## References

- [1] Jim Morel, “A Non-Local Tensor Diffusion Theory,” Los Alamos National Laboratory Report LA-UR-07-5257 (2007).
- [2] J.E. Morel, J.S. Warsa, K.G. Budge, “Alternative Generation of Non-Local Diffusion Tensors,” Los Alamos National Laboratory Report LA-UR-10-08285 (2010).
- [3] J. S. Warsa, T. A. Wareing, and J. E. Morel, “Krylov Iterative Methods and the Degraded Effectiveness of Diffusion Synthetic Acceleration for Multidimensional  $S_n$  Calculations in Problems with Material Discontinuities,” *Nuclear Science and Engineering*, **147**, 218-248 (2004).
- [4] Andrew T. Till, Marvin L. Adams, and Jim E. Morel, “The Finite Element with Discontiguous Support Multigroup Method: Application,” *Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo Method (MC)*, Nashville, TN, USA, April 19-23, 2015, on CD ROM, American Nuclear Society, LaGrange Park, IL (2015).

# Appendices

# A Parametric Study of the Adjoint Flux Calculation for TDKENO

Dustin Popp\*, Zander Mausolff\*, Mark DeHart†, Sedat Goluoglu\*

\*Nuclear Engineering Program, University of Florida  
529 Gale Lemerand Dr., Gainesville, FL, 32611

†Nuclear Systems Design and Analysis Division  
Idaho National Laboratory: 2525 North Freemont Street, Idaho Falls ID, 83415  
[dpopp07@ufl.edu](mailto:dpopp07@ufl.edu)

## INTRODUCTION

TDKENO is a time-dependent transport code designed to simulate reactor transients [1]. One application [2] is to study experiments performed at the Transient Reactor Test Facility (TREAT) at the Idaho National Laboratory (INL). TDKENO solves the time-dependent, three-dimensional Boltzmann transport equation with explicit representation of delayed neutrons. Instead of directly integrating this equation, the Improved Quasi Static (IQS) method is employed. This involves factoring the neutron flux into two components – a rapidly varying (in time) amplitude equation and a slowly varying shape equation [3]. These are solved separately on different time scales. The shape equation is solved using the Monte Carlo transport code KENO [4-5], from Oak Ridge National Laboratory's SCALE code package. Using the Monte Carlo method to solve the shape equation is computationally intensive, but the operation is only performed when needed – namely when the reactor system is undergoing a significant change. The amplitude equation is solved deterministically. This operation contains the time dependence and can be calculated relatively quickly, resulting in computational savings. TDKENO begins by solving the adjoint form of the transport equation for use as a weighting function. It uses this adjoint flux for multiple aspects of the simulation, including the calculation of all of the point kinetics parameters [3]. The flux shape is found by solving a modified form of the transport equation in “forward” mode. This paper examines the effect the adjoint weighting function has on the solution.

## TDKENO

TDKENO is a hybrid stochastic-deterministic program that solves the time-dependent transport equation in three dimensions. TDKENO uses the Improved Quasi-Static method to accomplish this. In this method the neutron flux is factored into two components. One component is a purely time-dependent, rapidly varying amplitude

function, which is solved deterministically and frequently (small time steps). The other is a slowly varying flux shape function that weakly depends on time and is only solved when needed, allowing significantly larger time steps. The flux factorization is presented in Equation 1.  $\psi(r, E, \Omega, t)$  is the angular flux at position ( $r$ ), energy ( $E$ ), angle ( $\Omega$ ) and time ( $t$ ).  $T(t)$  is the amplitude function, described by point kinetics equations, and  $\Phi(r, E, \Omega, t)$  is the flux shape function with weak time dependence.

$$\psi(r, E, \Omega, t) = T(t) \cdot \Phi(r, E, \Omega, t) \quad (1)$$

The two, coupled functions are solved using different time steps. The flux shape solution is computationally intensive, so it is only performed when there is a significant change in the system (e.g. controls rods pulling out). The time steps in between are relatively large. The amplitude equation can be quickly solved using deterministic methods, and is calculated frequently over much smaller time steps. The result is a computationally efficient solution that is able to depict neutron behavior for systems where conventional methods are unable to model. The following equation is introduced to ensure the factorization is unique and enables the amplitude equation to take on the form of the point kinetics equation [6].

$$\iiint \frac{1}{v} \Phi(r, E, \Omega, t) \phi^*(r, E, \Omega) d\Omega dE dV = \text{const} \quad (2)$$

In this equation,  $v$  is neutron speed, and  $\phi^*$  is the adjoint flux. The adjoint flux is calculated by solving the adjoint form of the transport equation, which is shown in Equation 3:

$$\begin{aligned} & -\bar{\Omega} \cdot \bar{\nabla} \phi^* + \Sigma_t \phi^* \\ & = \iint \Sigma_s(\bar{r}; \bar{\Omega}, E \rightarrow \bar{\Omega}', E') \phi^*(\bar{r}; \bar{\Omega}', E') d\Omega' dE' \\ & + \frac{1}{k_{eff}} \iint \chi(E') \nu \Sigma_f(\bar{r}, E') \phi^*(\bar{r}; \bar{\Omega}', E') d\Omega' dE' \quad (3) \end{aligned}$$

The adjoint formulation is often used in perturbation theory but is important in Monte Carlo problems because it provides the relative importance of neutrons [7]. For this reason it is referred to as a weighting factor. This adjoint flux is also used in the normalization factor that gives the amplitude equation the form of the point kinetics equations. This factor, along with Equation (2), is used to give the expression for the amplitude function used in the flux factorization [8]. The generation time, the reactivity, and the precursor equations are all dependent on the adjoint flux as well. It is clear that the adjoint flux plays a significant role in the way TDKENO simulates the neutron behavior in time. We are motivated to study the sensitivity to the adjoint calculation because we are interested in employing a different Monte Carlo code to perform the shape calculation. Not all codes are able to explicitly solve the adjoint form of the transport equation. For instance, in Serpent it is not possible to perform an explicit adjoint calculation. One can weight calculations with the adjoint by using the iterated fission probability and keeping track of the progeny of neutrons in the forward simulation. This paper will demonstrate the importance of the adjoint solution by highlighting the changes in results seen by adjusting only the adjoint calculation of the model. It also seeks to identify optimal parameters to use for the adjoint flux solution when modeling a highly complex system.

## DESCRIPTION OF WORK

### Adjoint Flux Solution Parameters

The adjoint flux is obtained using the Monte Carlo method to solve the adjoint form of the Boltzmann transport equation. TDKENO does this with a modified version of either KENO V.a or KENO-VI [1]. For this work, KENO-VI was used. In its current implementation the adjoint calculation is found for the initial critical state and is used throughout the simulation. Generally, this works fine as long as the spatial distribution of the flux does not change drastically.

The fundamental parameters in a Monte Carlo calculation are the number of generations and the number of neutrons to be simulated in each generation. The goal behind this work is two-fold. First, it is to demonstrate the effect that the adjoint flux has on the final result of the calculation. This is done to provide motivation for the second part of this work, which is to investigate the optimal values for the two parameters in question.

## Modeling TREAT with TDKENO

To test the effect of the adjoint flux solution on TDKENO, we built a model [2] of a transient experiment performed with TREAT based on the M8 Calibration Tests (M8CAL) [9]. These are some of the most recent tests before the reactor was shut down in 1994 and they have a more complete data set than most other TREAT experiments. For this work, a temperature-limited transient experiment, #2855, was selected. In this experiment, transient conditions are induced by the rapid withdrawal of a group of control rods, which leads to a spike in power. In order to demonstrate the effect of the adjoint function on an actual simulation for a complex system such as TREAT, several calculations were performed with our model. The cases were split into varying values for the number of particles per generations used in the adjoint calculation. The values tested for number of particles per generation were 5000, 10000, 15000, and 20000. For each of these values, five cases were run with varying values for the number of generations. These were 2500, 3000, 3500, 4000, and 5000. This made 20 cases in total. For each different value for particles per generation, a plot is provided showing reactor power over time for the simulated experiments with varying numbers of generations. For all of the forward cases, the parameters were held constant at 2000 generations and 20000 particles per generation. All calculations use the ENDF/B-VII 238-group cross-section library.

## RESULTS

Below are the results of our transient simulations for multiple cases of varying parameters for the initial adjoint calculation. The results are given in the form of plotted reactor power over time. For the plots, time is measured in seconds and power is measured in Megawatts. The entire transient we selected to model lasts for 60 seconds. However, the plot just shows the power for a fraction of time after the first second, as this is when the power spike from withdrawing the control rods occurs. Power is only shown from 800 to 2200 MW, as the primary purpose behind these plots is to show the variance between individual cases rather than demonstrate the physical behavior of the system. Throughout the plots, “number of particles per generation” is abbreviated as NPG and “number of generations” is abbreviated as GEN. One line is dashed to distinguish it from a similar color.

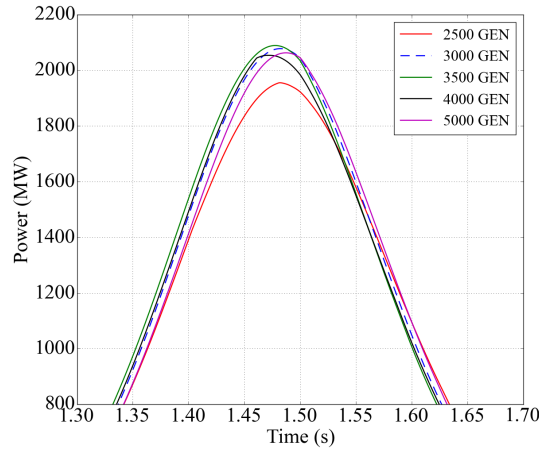


Figure 1. Power vs. Time For NPG = 5000

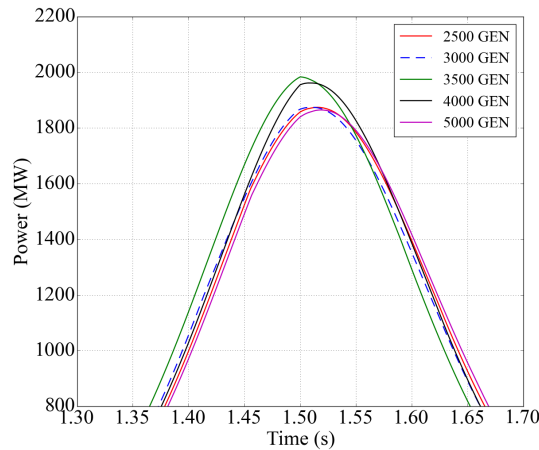


Figure 2. Power vs. Time For NPG = 10000

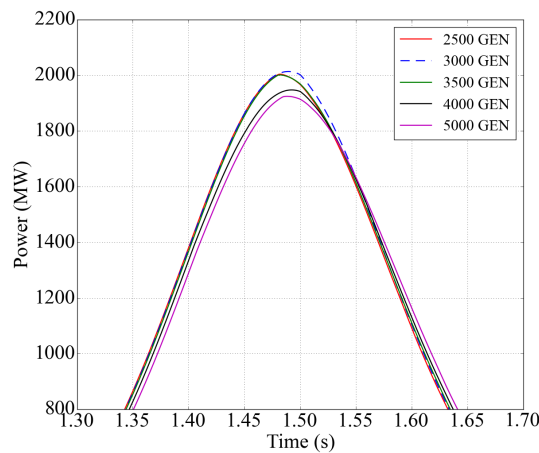


Figure 3. Power vs. Time For NPG = 15000

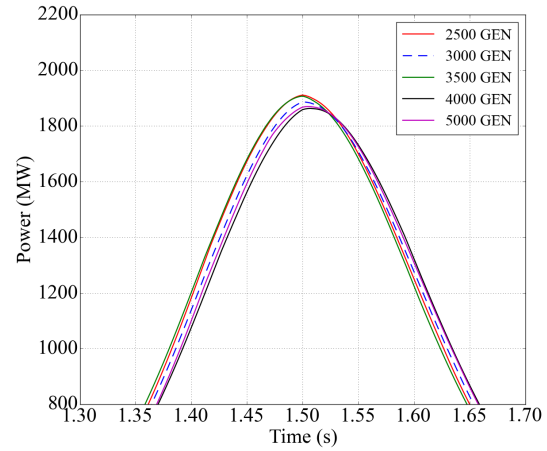


Figure 4. Power vs. Time For NPG = 20000

From the data, it is clear that the adjoint solution has a significant impact on the outcome of the overall calculation. To briefly assess under sampling, we conducted several Shannon Entropy tests available in KENO. The implementation in KENO provides three tests for fission convergence, one for final convergence, first converged generation, and adequate active generations [10]. All of the calculations in Figure 1 passed all three tests, indicating the fission source is likely converged. The Shannon entropy does not guarantee convergence, but does provide strong evidence for it.

The parameters for all other calculations are held constant, yet we still see a variety of results from varying the parameters of only the adjoint case. This fact confirms that the adjoint calculation parameters are worth considering in our effort to optimize simulation accuracy. Looking at the plots in succession, we see much higher variance among the cases of different generations when the number of particles per generation is lower than 20000. Only for 20000 particles per generation do we begin to see a convergence towards a singular solution. This could be an indication that at least 20000 particles per generation for the adjoint flux solution are required to accurately sample a complex system.

## Conclusion and Future Work

The data was not proven extensive enough to make any sufficient conclusions on the optimal parameters for the adjoint calculation. What we can conclude is that the adjoint flux contributes enough to the overall time-dependent solution to warrant further study into optimizing these parameters. We can also conclude that the adjoint calculation should be simulating at least 20000 particles for every generation in order to properly model the physical system during a



transient. Future work will include running cases with more than 20000 particles per generation, simulating temperature-limited transients 2856 and 2857 so that the study is not limited to one experiment, and including computation times in the analysis to determine the point at which increases in simulation accuracy cease to be worth the increases in computation time. Work could also be done to compare TDKENO to another transient analysis code that does not employ an adjoint flux solution, to investigate if the adjoint solution provides an advantage in simulation accuracy.

## REFERENCES

1. Z. MAUSOLFF and S. GOLUOGLU, "Verification of the Time-dependent Transport Code TDKENO with the Monte Carlo Code KENO-VI," PHYSOR 2016, American Nuclear Society (2016), p. 2698.
2. J. PALUCH, D. POPP, H. MORBACH, V. GRAHAM, M. DEHART, S. MORELL, S. GOLUOGLU, "Application of TDKENO to TREAT Temperature-Limited Transients," PHYSOR 2016, American Nuclear Society (2016), p. 2426.
3. C.L. BENTLEY, "Improvements in a Hybrid Stochastic/Deterministic Method for Transient, Three-Dimensional Neutron Transport," Ph.D. Dissertation, Nuclear Engineering Department, The University of Tennessee, Knoxville (1996).
4. R. D. BUSCH, "KENO V.a Primer: A Primer for Criticality Calculations with SCALE/KENO Va Using CSPAN for Input," ORNL/TM-2002/155. Oak Ridge National Laboratory, Oak Ridge TN (2003).
5. S. M. BOWMAN, "KENO-VI Primer: A Primer for Criticality Calculations with SCALE/KENO-VI Using GeeWiz," Oak Ridge National Laboratory, Oak Ridge TN. ORNL/TM-2008/069 (2008).
6. A.F. HENRY, "The Application of Reactor Kinetics to the Analysis of Experiments," *Nuclear Science and Engineering*, **3**, 1, 52-70 (1958).
7. D.C. IRVING, "The adjoint Boltzmann equation and its simulation by Monte Carlo," *Nuclear Engineering and Design*, Volume 15, 1971, Pages 273-293, ISSN 0029-5493, [http://dx.doi.org/10.1016/0029-5493\(71\)90069-0](http://dx.doi.org/10.1016/0029-5493(71)90069-0). (<http://www.sciencedirect.com/science/article/pii/0029549371900690>)
8. S. GOLUOGLU and H.L. DODDS, "A Time-Dependent, Three-Dimensional Neutron Transport Methodology," *Nuclear Science and Engineering*, **139**, 248-261 (2001)
9. W.R. ROBINSON and T.H. BAUER, "The M8CAL Power Calibration Experiment," *ANL-IFR-212*, URL <http://www.osti.gov/scitech/servlets/purl/1130413> (1994).
10. B.T. REARDEN, M.E. DUNN, D. WIARDA, et. AL., "Overview of SCALE 6.2," Tech rep., Oak Ridge National Laboratory (ORNL) (2013).

# Motivation for Using Generalized Geometry in the Time Dependent Transport Code TDKENO

Dustin Popp\*, Zander Mausolff, Sedat Goluoglu  
Nuclear Engineering Program, University of Florida  
529 Gale Lemerand Dr., Gainesville, FL, 32611  
[dpopp07@ufl.edu](mailto:dpopp07@ufl.edu)

## INTRODUCTION

Safety is always a primary concern with nuclear power plants. However, the accident at the Fukushima Daiichi plant provoked interest in the safety of U.S. nuclear fuel during transient conditions. Accident Tolerant Fuel (ATF) became a topic of interest and congress announced a formal agenda to put ATF in a commercial reactor by 2022 [1]. Experimentation is required to develop such fuel. Due to the cost associated with the proposed experiments, there is interest in using high fidelity simulations of experimental facilities to optimize the reactor test environments, minimizing the number of pre-test experiments. One of the facilities suitable for ATF development is the Transient Reactor Test Facility (TREAT) [2] at Idaho National Laboratory. TREAT was decommissioned in 1994, but the Department of Energy (DOE) is planning to bring it online to perform transient experiments on ATF. Simulating the TREAT environment requires an accurate depiction of the geometry, neutron behavior, and temperature distributions. Additionally, due to the transient nature of the experiments, a time-dependent simulation is needed. We are proposing to use the code, TDKENO [3], to model TREAT. TDKENO solves the time-dependent, three-dimensional Boltzmann transport equation with explicit representation of delayed neutrons. Instead of directly integrating this equation, the neutron flux is factored into two components – a rapidly varying amplitude equation and a slowly varying shape equation [4] and each is solved separately on different time scales. The shape equation is solved using the 3D Monte Carlo transport code KENO [5,6], from Oak Ridge National Laboratory's SCALE code package. Using the Monte Carlo method to solve the shape equation is still computationally intensive, but the operation is only performed when needed. The amplitude equation is solved deterministically and frequently, so the solution gives an accurate time-dependent solution without having to repeatedly solve the entire equation. Thus far, we have been using KENO V.a to solve the shape equation. KENO V.a is a version of KENO that only supports simple geometry. While simpler geometries make calculations easier, and therefore faster, they are unable to accurately portray complex systems such as TREAT. Another version of KENO, KENO-

VI, allows for generalized geometry. We have modified TDKENO to incorporate KENO-VI so that we may accurately represent the geometries within TREAT. This paper explains the motivation behind using generalized geometry, and provides the results of our modifications.

## TDKENO

TDKENO is a hybrid stochastic-deterministic program that solves the time-dependent transport equation in three dimensions. TDKENO uses the Improved Quasi-Static method to accomplish this. In this method, the neutron flux is factored into two components. One component is a purely time-dependent and rapidly varying amplitude function, which is solved deterministically and very frequently (small time steps). The other is a slowly varying flux shape function that weakly depends on time and is only solved when needed (significantly larger time steps). The flux factorization is presented in Equation 1.  $\psi(r, E, \Omega, t)$  is the angular flux at position ( $r$ ), energy ( $E$ ), angle ( $\Omega$ ) and time ( $t$ ).  $T(t)$  is the amplitude function, described by point kinetics equations and  $\Phi(r, E, \Omega, t)$  is the flux shape function with weak time dependence.

$$\psi(r, E, \Omega, t) = T(t) \cdot \Phi(r, E, \Omega, t) \quad (1)$$

The two, coupled functions are solved using different time steps. The flux shape solution is computationally intensive, so it is only performed when there is a significant change in the system (i.e. controls rods pulling out) and the time steps in between calculations are relatively large. The amplitude equation can be quickly solved using deterministic methods, and is calculated frequently over much smaller time steps. The result is a computationally efficient solution that is able to depict neutron behavior for systems where conventional methods would be unable. Another primary component of TDKENO is its feedback model, which is used to calculate the power, total yield, and point kinetics parameters. The feedback mechanism that calculates the reactivity as a function of total energy produced in the core is implemented and described in Equation (2).

$$\rho_{fb}(t) = \sum_{i=0}^3 a_i Y^i(t) \quad (2)$$

In Equation (2),  $\rho_{fb}(t)$  is the feedback reactivity,  $Y(t)$  is the total yield in the core at time  $t$  in MJ, and  $a_i$  refers to the empirical coefficients for a third order polynomial. TDKENO employs an efficient solution to the time-dependent Boltzmann transport equation to handle neutronics, and with a simple feedback model, makes an excellent tool for transient analysis.

## DESCRIPTION OF ACTUAL WORK

### Generalized Geometry

When TDKENO was originally written, it used a modified version of the Monte Carlo code KENO V.a to solve the flux shape equation. KENO V.a may model a system with only simple geometries such as cuboids, cylinders, and spheres. KENO V.a also limits possible geometries by preventing the rotation or intersection of regions. This method leads to a much quicker computation time. However, it prevents the user from creating an input that accurately portrays the geometry of a complex system. To achieve an improved representation of the TREAT geometry, we have modified TDKENO to incorporate KENO-VI. KENO-VI is another version of the KENO Monte Carlo code that incorporates generalized geometry. KENO-VI allows for more complex shapes such as planes, dodecahedrons, wedges, etc. It also allows for rotating and intersecting regions. The drawback of this feature is that neutron tracking becomes more computationally intensive, as many quadratic functions need to be evaluated to determine neutron locations. Using the generalized geometry capabilities of KENO-VI, we no longer have to make geometrical approximations in our TREAT model and we can capture a more accurate representation of the reactor. Additionally, using generalized geometries yields significantly fewer regions in the model. This provides more geometrical flexibility and can improve sampling of the system. Incorporation of KENO-VI has been verified [7] against computational benchmarks and the KENO V.a version of TDKENO. Although computation times are increased, we see more agreement with experimental results using KENO-VI than we did with KENO V.a.

### Modeling TREAT

To test TDKENO's ability to simulate the TREAT environment, we built models of three transient experiments based on the M8 Calibration Tests (M8CAL) [8]. These are some of the most recent

tests before the reactor was shut down in 1994 and they have a more complete data set than most other TREAT experiments. The three temperature-limited transients are labeled #2855, #2856, and #2857. In these experiments, transient conditions are caused by the rapid withdrawal of the control rods, which leads to a spike in power. In order to provide motivation behind the incorporation of KENO-VI into TDKENO, results of simulating the three M8CAL transient experiments are provided and explained below. For each experiment, plots are provided showing reactor power and reactor yield over time. The data comes from the M8CAL documentation (experimental) and from two simulations – one with the KENO V.a and one with the KENO-VI version of TDKENO. To demonstrate the costs of using generalized geometries, the computation times for each run are provided for comparison. All simulations are run with 2000 generations, 8000 particles per generation, 500 skipped generations, and use the ENDF/B-VII 238-group cross-section library.

## RESULTS

Below are the results of our transient simulations compared with experiment, and the associated computation times. The results are grouped by the three experiments of interest: Transient numbers #2855, #2856, and #2857. Tables I, II, and III provide the number of regions for each case, to highlight the drastic reduction achieved by using generalized geometry, along with the computation times for each run. To demonstrate the simulation accuracy, the final yield is given in each table. The experimental values are taken from the M8CAL document, along with the associated uncertainty. The figures give the integrated power (yield) against time. The yield plots are given over the full 60 seconds of the transient experiment.

### Transient #2855

Table I: Data for Transient #2855

Case	Regions	Computation Time (Hours)	Yield
<b>Experiment</b>	N/A	N/A	$792 \pm 10\%$
<b>KENO V.a</b>	11886	53	895
<b>KENO VI</b>	4332	129	872

From the yield, we see an improvement in simulation by using KENO IV. The percent error for the yield value calculated using KENO V.a is 13%, while the percent error using KENO VI is only 10.1%. However, there is a discrepancy between the reported yield and the data table in the M8CAL document. This makes it difficult to compare our results to the actual experiments, but the simulation is sufficient and it is

still shown that using generalized geometry in TDKENO has improved our results.

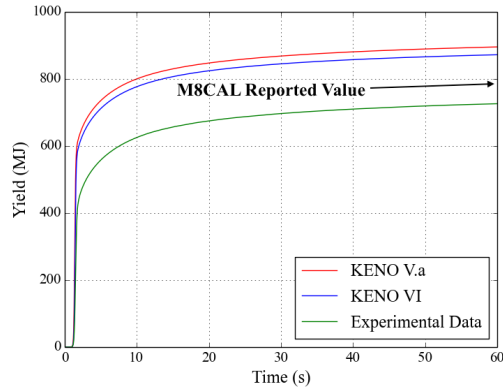


Figure 2. Plot of Yield vs. Time for Transient 2855

### Transient #2856

Table II: Data for Transient 2856

Case	Regions	Computation Time (Hours)	Yield
Experiment	N/A	N/A	1572 $\pm$ 10%
KENO V.a	11902	54	1574
KENO VI	4332	138	1587

In this experiment, the percent error using KENO V.a and KENO VI is, respectively, 0.13% and 0.95%. Although the error is slightly greater with KENO VI, the errors are insignificant and we see strong agreement between simulation and experiment. Again, there is inconsistency in the M8CAL document. However, our switch to generalized geometry maintains an effective simulation while drastically reducing the number of regions.

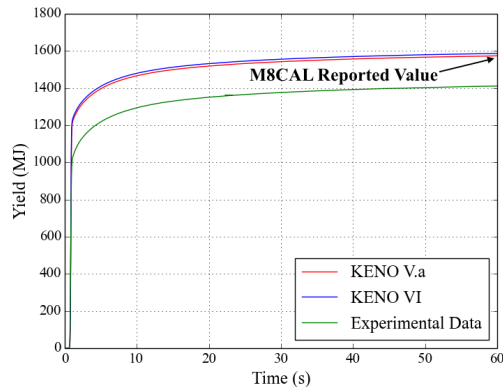


Figure 4. Plot of Yield vs. Time for Transient 2856

### Transient #2857

Table III: Data for Transient 2857

Case	Regions	Computation Time (Hours)	Yield
Experiment	N/A	N/A	2265 $\pm$ 10%
KENO V.a	11878	53	2194
KENO VI	4332	139	2140

Here, the percent errors for KENO V.a and KENO VI are, respectively, 3.13% and 5.52%. Again, KENO VI gives us a slightly higher disagreement from the reported value but the difference in errors is acceptably smaller for receiving a more accurate depiction of the system's geometry. There is again a discrepancy between the M8CAL data and the reported value.

Figure 5. Plot of Power vs. Time for Transient 2857

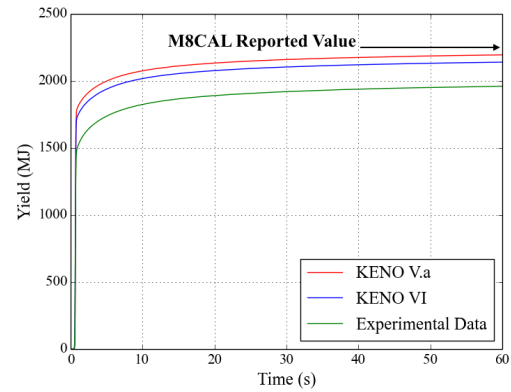


Figure 6. Plot of Yield vs. Time for Transient 2857

### Conclusion and Future Work

We incorporated KENO-VI into TDKENO to add the functionality of generalized geometry to the code. This way, we may more accurately portray the TREAT reactor in our model. While the cost of doing so is an extended computation time relative to the KENO V.a version, we gain a model that has significantly fewer regions, has a more exact depiction of the actual geometry, and gives some results that have a higher agreement with experimental data. It is decided that these improvements outweigh the sacrificed computation time, and we have proper motivation for using generalized geometry in our model. Additional work is currently being done to parallelize the KENO-VI portion of TDKENO, which would negate the loss in computation speed and provide us with a greatly improved modeling code. Other future work includes coupling TDKENO to a thermal hydraulics code in order to incorporate a temperature feedback mechanism to update cross sections during run time.

## REFERENCES

- [1] J. Carmack, F. Goldner, S. M. Bragg-Sitton, and L. L. Snead, "Overview of the U.S. DOE accident tolerant fuel development program", *TopFuel 2013*, Charlotte, North Carolina, Sep. 15-19, 734-739 (2013).
- [2] D. KONTOGEORGAKOS, et. al. , "Initial Neutronics Analyses for HEU to LEU Fuel Conversion of the Transient Reactor Test Facility (TREAT) at the Idaho National Laboratory", ANL/GTRI/TM-13/4 (2013).
- [3] K. L. GOLUOGLU, "Revitalization and Status of the Hybrid Stochastic/Deterministic Computer Code for Transient, Three-Dimensional Neutron Transport, TDKENO," A practice project for Master of Science degree, The University of Tennessee, Knoxville (2004).
- [4] C.L. BENTLEY, "Improvements in a Hybrid Stochastic/Deterministic Method for Transient, Three-Dimensional Neutron Transport," Ph.D. Dissertation, Nuclear Engineering Department, The University of Tennessee, Knoxville (1996).
- [5] R. D. BUSCH, "KENO V.a Primer: A Primer for Criticality Calculations with SCALE/KENO V.a Using CSPAN for Input," ORNL/TM-2002/155. Oak Ridge National Laboratory, Oak Ridge TN (2003).
- [6] S. M. BOWMAN, "KENO-VI Primer: A Primer for Criticality Calculations with SCALE/KENO-VI Using GeeWiz," Oak Ridge National Laboratory, Oak Ridge TN. ORNL/TM-2008/069 (2008).
- [7] Z. MAUSOLFF, S. GOLUOGLU, M. DEHART, "Verification of the Time-dependent Transport Code TDKENO with the Monte Carlo Code KENO-VI"
- [8] W.R. ROBINSON and T.H. BAUER, "The M8CAL Power Calibration Experiment," *ANL-IFR-212*, URL <http://www.osti.gov/scitech/servlets/purl/1130413> (1994).

# **Verification of the Time-dependent Transport Code TDKENO with the Monte Carlo Code KENO-VI**

**Zander Mausolff\*, Sedat Goluoglu**

Nuclear Engineering Program, University of Florida  
529 Gale Lemerand Dr., Gainesville, FL, 32611  
zanderm@ufl.edu

**Mark DeHart**

Nuclear Systems Design and Analysis Department  
Idaho National Laboratory  
2525 North Freemont Street, Idaho Falls ID, 83415

## **ABSTRACT**

The loss of coolant accident at the Daiichi plant has increased attention to the development of fuels for United States commercial reactors with enhanced accident tolerance. Testing of new fuels is scheduled at Idaho National Laboratory's Transient Reactor Test Facility (TREAT). Typically a number of pre-test calibrations are required before actual experimentation may begin. We hope to simulate TREAT experiments to minimize the number of calibrations needed. Systems such as TREAT pose several modeling challenges due to the non-uniform insertion of reactivity from transient rods and complex core geometries. To simulate complex geometries with ease, the code TDKENO has been modified to incorporate the Monte Carlo code KENO-VI in its calculation sequence. Previously, TDKENO relied on KENO V.a, which only supports simple geometry inputs. TDKENO provides advantages over typical codes by solving the three-dimensional time dependent transport equation with delayed neutrons partially via Monte Carlo calculations. Newly added capabilities in TDKENO are verified through comparison to computational benchmark and sample problems. Finally, TREAT temperature-limited transients are simulated with KENO V.a and KENO-VI inputs. Simulations with KENO-VI inputs resulted in improved agreement between calculated TDKENO values and experiment over KENO V.a inputs.

*Key Words:* **TDKENO, TREAT, Improved Quasi-Static (IQS) Method, Transient**

## **1. INTRODUCTION**

Nuclear power has long been a reliable, safe, and carbon free method of producing tremendous amounts of electricity in the United States (U.S.). Continual improvement of fuel burnup, reliability, and safety during normal and transient operation of nuclear reactors is essential for the industry's success. Safety is always a priority in a plant, however the accident at the Fukushima Daiichi nuclear power plant raised questions regarding fuel performance of U.S. reactors during off normal conditions. The result is a renewed interest in enhancing the accident tolerance of fuels used in the U.S. commercial fleet. Congress formally outlined a schedule for the development of accident tolerant fuel (ATF), with proposed fuel being placed in a commercial reactor by 2022 [1].

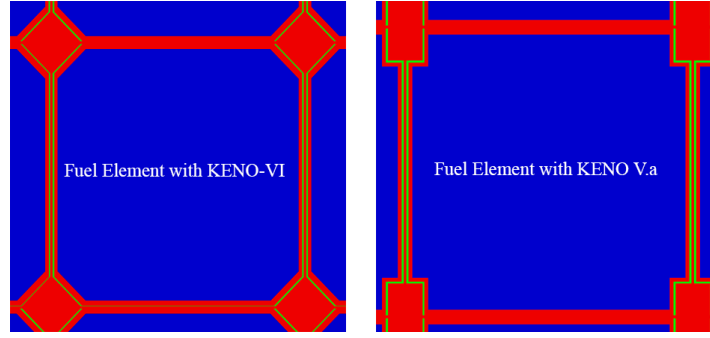
---

\* Corresponding Author

This time frame has prompted new methods to expedite experimentation. These new methods will rely on high fidelity simulation of experimental systems, primarily to reduce the series of pre-test experiments required to achieve the desired environment in the test-vehicle. One facility capable of performing the experiments is the Transient Reactor Test Facility (TREAT) at Idaho National Laboratory (INL). Since TREAT's decommissioning in 1994, there have been great advances in computing speed and memory that now makes high fidelity simulation of TREAT experiments feasible. Transient experiments carried out with TREAT are difficult to simulate with conventional codes due to the non-uniform perturbations produced within the core. Additionally, codes tailored to TREAT are based on high-enriched uranium systems and are not guaranteed to work on the low enriched fuel that will be tested. The fission yield from such experiments is time dependent and a function of many system parameters. These parameters include (but are not limited to) the fissionable material type, neutron interactions in the system (thermal, intermediate, fast), enrichment (or presence of resonance capture isotopes), the reactivity insertion rate and amount, and heat capacity and heat transfer features of the materials including any cooling mechanisms [2]. To understand these multi-physics systems requires accurate simulation of the neutronic behavior, geometry, and temperature distribution. At present, the most rigorous analytical methodology for understanding neutronic behavior involves solving the time-dependent governing equations for three-dimensional (3-D) geometries, commonly referred to as spatial kinetics [3].

We propose the use of the code TDKENO to model TREAT experiments. TDKENO solves the time dependent 3-D Boltzmann transport equation with the explicit representation of delayed neutrons to obtain the neutron flux. Directly integrating this equation is difficult and requires frequent recalculation to handle the time dependence. Instead we assume the flux may be factored into components to improve computation time while preserving accuracy. The neutron flux is factored into a rapidly varying amplitude equation and a slowly varying shape equation (an assumption that has generally shown to be true) [4]. The shape equation is solved with a modified version of the Monte Carlo code KENO (part of the SCALE package) [5,6]. Solving the shape equation via Monte Carlo is computationally intensive, therefore is only updated when needed. By solving the rapidly varying amplitude equation deterministically many times between shape calculations results in a highly accurate solution without the expense of direct integration.

The desire to model complex cores such as TREAT prompted the modification of TDKENO to support generalized geometry inputs. This is advantageous as the input may be created to exact system parameters. TDKENO was initially developed with KENO V.a: the geometrically simpler version of KENO. While quicker than Monte Carlo codes with generalized geometry, KENO V.a places practical limits on input designs as only combinations of cuboids, spheres, cylinders, etc. are possible. To illustrate this problem, two input strategies for modeling the chamfered corners of a TREAT fuel element are shown in Figure 1 for KENO V.a and KENO-VI.



**Figure 1.** Chamfered corners of fuel elements in KENO V.a and KENO-VI. Fuel elements have one step and non-fuel elements have a stair like structure in KENO V.a. The KENO-VI structure mimics the chamfered corner for all elements and requires less regions.

As depicted in Figure 1, the corners are approximated with a stair-like combination of cuboids with KENO V.a. The goals of creating KENO-VI inputs is to better capture the geometry (e.g. corners may be created using planes) with fewer regions.

In order to achieve high fidelity modeling of TREAT experiments the following has been done:

- Modification of KENO-VI to solve the flux shape equation inside of TDKENO.
- Verification of these modifications through comparison of identical inputs from KENO V.a and KENO-VI.
- Simulation of TREAT transients from the M8CAL experiments (Transients #2855, #2856, and #2857) using KENO V.a and KENO-VI inputs further verify modifications. Additionally, the improved KENO-VI inputs show better agreement with experiment.

## 2. TDKENO: A HYBRID TIME-DEPENDENT TRANSPORT PROGRAM

Transient analysis of system such as TREAT cannot be analyzed with high degrees of accuracy using conventional nuclear engineering codes. These codes typically employ diffusion theory or point kinetics approaches. Diffusion theory fails in highly absorbing regions and point kinetics approaches that are insufficient to capture spatial reactivity effects. Therefore, transport theory is required to simulate voided and/or strongly voided regions, and asymmetric material movements accurately [3].

The code TDKENO is a large program containing several large subroutines that solves the time dependent, 3-D Boltzmann transport equation with the explicit representation of delayed neutrons (precursor equations). Numerical techniques with varying degrees of accuracy are required as analytical solutions for transport equations only exist for simple geometries. The methodology utilized within TDKENO is the improved quasi-static (IQS) method [3,7]. This method relies on factoring the neutron flux into a purely time dependent, rapidly varying amplitude equation and a slowly varying shape equation. Equation (1) describes the flux factorization, where  $\psi(r, E, \Omega, t)$  is the angular flux at position ( $r$ ), energy ( $E$ ), angle ( $\Omega$ ) and time ( $t$ ).  $T(t)$  is the amplitude function, described by point kinetics equations and  $\Phi(r, E, \Omega, t)$  is the flux shape function with weak time dependence. The flux factorization is made unique by imposing a normalization condition [3].

$$\psi(r, E, \Omega, t) = T(t) \cdot \Phi(r, E, \Omega, t) \quad (1)$$



The result of this factorization is two coupled equations that are solved on different time scales. Calculation of the flux shape is computationally expensive. It is only calculated when an update is required, like a large change in system behavior. Conversely, the amplitude function requires significantly less time and can solve the point kinetics equations deterministically over shorter time steps. The time derivative in the Boltzmann transport equation is replaced with a first order backwards difference approximation [4]. The result of this factorization is a computationally efficient solution that is able to extract neutronic information where typically employed methodologies are invalid.

TDKENO is considered a hybrid solution method; the amplitude equation is solved deterministically while the flux shape is calculated via Monte Carlo with KENO (from the SCALE package) [2,3]. KENO may refer to either KENO V.a (simple geometry) or KENO-VI (generalized geometry). Both are now available in TDKENO.

An additional component of TDKENO that makes it suitable for transient analysis is the incorporation of a feedback model. A feedback model is required to accurately determine the power, total yield, and point kinetics parameters, such as reactivity and effective delayed neutron fractions. Values such as reactivity and effective delayed neutron fractions are not measured directly [4]. Rather, the periods of delayed or prompt supercritical systems is used to infer the reactivity. In addition, time lag of delayed neutrons allows better-controlled transients. TDKENO incorporates feedback via quenching coefficients provided *a priori* [3,4]. The temperature distribution within core materials affects reactivity as well. To incorporate temperature-dependent cross sections, TDKENO would need to be coupled to a code that provides the temperature distribution as a function of energy created in the core. As this currently isn't implemented, an approximation is used instead. The feedback mechanism that calculates the reactivity as a function of total energy produced in the core is implemented and described in Equation (2):

$$\rho_{fb}(t) = \sum_{i=0}^3 a_i Y^i(t) \quad (2)$$

where  $\rho_{fb}(t)$  is the feedback reactivity,  $Y(t)$  is the total yield in the core at time  $t$  in MJ, and  $a_i$  refers to the empirical coefficients for a third order polynomial.

The neutronic capabilities of TDKENO come from the ability to solve the time-dependent Boltzmann transport equation efficiently. Even utilizing a simple feedback method makes TDKENO an excellent code for transient analysis as few codes accurately determine the time dependence, neutronics, and geometry with such high fidelity.

### 3. INCORPORATION OF KENO-VI INTO TDKENO

TDKENO was first constructed with KENO V.a performing fixed source calculations and flux shapes. KENO V.a allows inputs to be constructed in a combinatorial manner with basic shapes such as spheres, cylinders, and cuboids [5]. It is further restricted as regions may not intersect or be rotated. The advantages are all regions are easily defined and determining neutron locations is less intensive compared with generalized geometries. KENO-VI is referred to as a generalized geometry

code that supports many constructs including planes, dodecahedrons, wedges, parallelepiped, etc. [6]. In addition, it allows regions to be rotated and to intersect. The drawback is the calculations take approximately four times as long as KENO V.a, primarily because neutron tracking requires the evaluation of many quadratic functions.

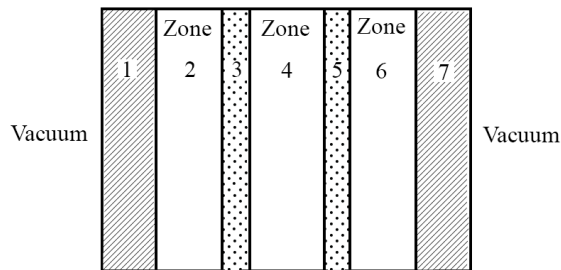
For this work, KENO-VI code (from SCALE 6.1) was modified and incorporated into TDKENO to calculate the flux shape. Modifying KENO-VI files for TDKENO required updating the build environment for TDKENO. CMAKE is currently used to handle the building of TDKENO, holding hundreds of files across many directories. As of now CMAKE scripts have been written so TDKENO can be built on Linux and Mac based computers. These modifications allow user to make use of either KENO V.a or KENO-VI inputs.

#### 4. VALIDATION OF MODIFICATIONS TO TDKENO

To verify the modifications to TDKENO, a series of transients are simulated with both KENO V.a and KENO-VI inputs. It should be noted that TDKENO has been extensively benchmarked with KENO V.a in Reference [4]. As KENO V.a and KENO-VI only differ in the treatment of geometry, similar behavior is expected between the two. The problems are presented in order of increasing complexity, beginning with the 16A1 benchmark problem [8]. With much of the motivation for modifying TDKENO to better simulate TREAT, we first examine a single element input of TREAT then proceed to the full core.

##### 4.1. 16A1 Benchmark

The 16A1 benchmark problem describes a seven region transport model (Figure 2) of a liquid metal fast breeder reactor. The configuration consists of three core regions (Zones 2, 4, and 6) containing core material and sodium, two control rod regions (Zones 3 and 5) containing control rod material and sodium, and two fuel blanket regions (Zones 1 and 7) containing blanket material and sodium [4]. This problem is one-dimensional; it simulates three dimensions by utilizing reflective boundary conditions in the y- and z-directions. This computational benchmark is a relatively simple transient and provides an excellent starting point for validating code modifications. Transient behavior is initiated with a perturbation caused by a 5% increase in density of the material in Zone 2 and a 5% decrease in density of the material in Zone 6, resulting in a delayed supercritical transient [4]. Input parameters are provided in Table 1.



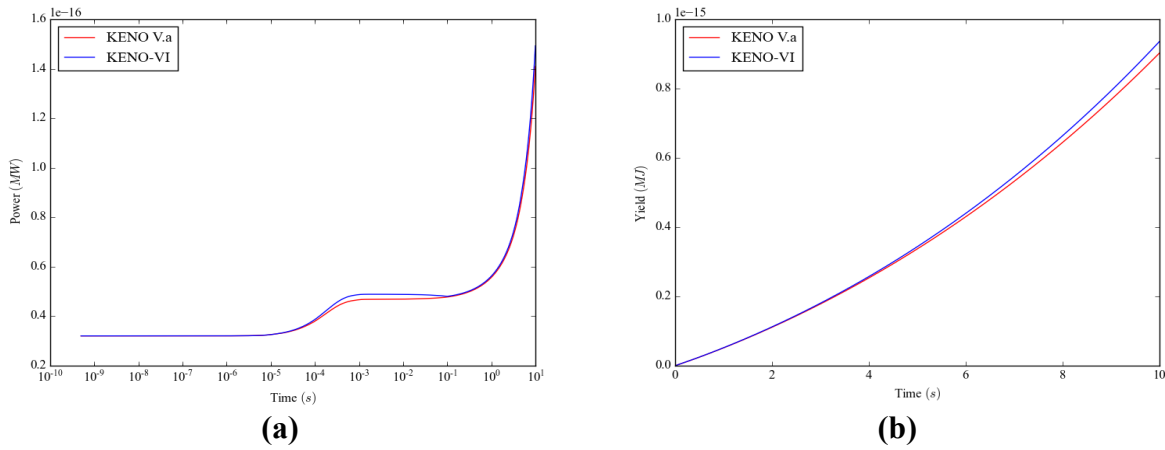
**Figure 2.** 16A1 geometry description.

For a preliminary comparison, the calculated power and yield using TDKENO with KENO V.a and

KENO-VI inputs are respectively plotted in Figures 3a and 3b. The calculated power and yield show excellent good agreement between the KENO V.a. and KENO-VI inputs. The large number of histories give us confidence that TDKENO is indeed working as intended with KENO-VI.

**Table 1.** Input parameters for the 16A-1 Benchmark input.

	KENO V.a	KENO-VI
<b>Number of generations</b>	1500	1500
<b>Particles per generation</b>	20000	20000
<b>Number of regions</b>	7	7
<b>Number of materials</b>	5	5
<b>Flux shapes</b>	5	5
<b>Calculation time</b>	200 min	820 min



**Figure 3.** Power (a) and yield (b) are calculated with TDKENO using both KENO V.a and KENO-VI inputs. The time scale is logarithmic in (a) to distinguish features varying over the small time scale. Very good agreement is observed between KENO V.a and KENO-VI.

## 4.2 Description of TREAT Core

The TREAT facility enables study of reactor safety research and fuel behavior. It serves as a non-destructive testing ground for fuel development and assessment of accident severity. TREAT experiments are accomplished by depositing transient energy to targets within experiment rigs. Detailed descriptions of TREAT may be found in References [9,10].

The fuel for the experiments described in the following section is composed of highly enriched (93.1%)  $\text{UO}_2$  dispersed homogenously within a graphite mixture with a 10,000:1 graphite-to-uranium ratio. The fuel elements are 10.05 cm squares with 1.58 cm chamfered corners for air coolant flow. The active fuel length is 122.23 cm and the total fuel element length is 247.65 cm [11]. The fuel is encased in a Zircaloy-3 can and capped with graphite reflectors on the top and bottom that are encased in a 6063-aluminum can.

Transient conditions in TREAT begin and end with control rod insertion and removal. Control elements are identical to the fuel elements except they contain a 4.445 cm outer radius Zircaloy-2 tube surrounded by a carbon steel tube packed with B<sub>4</sub>C powder [11]. There are three types of control rods. Transient rods are the first and are maneuvered to initiate transient conditions [11]. To maintain reactivity of the core during transient operation users rely on compensation rods. Shutdown rods end transient operation (forced back to subcritical) and signify the end of the experiment.

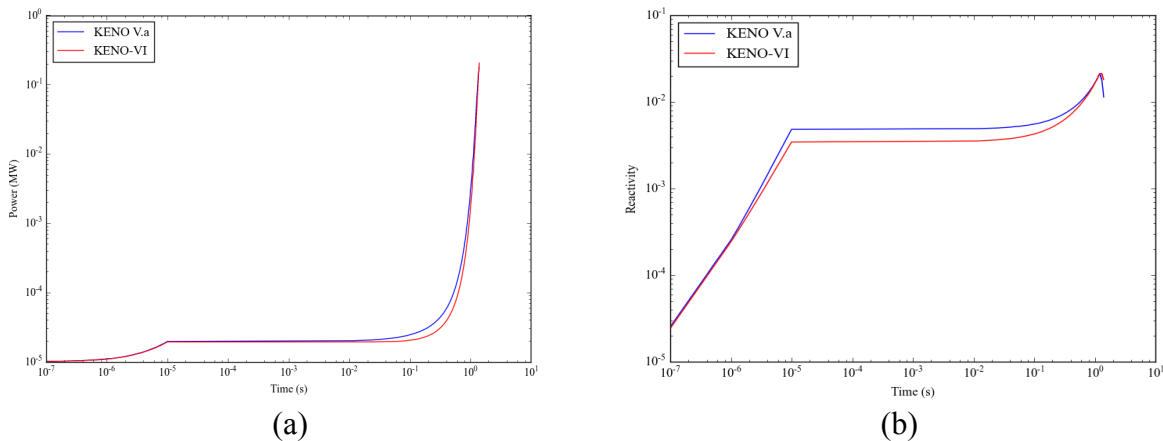
### 4.3 Arbitrarily Simplified Treat (Single Element)

As a precursor to simulating full TREAT experiments a simplified model was created. Chamfered corners are omitted, fuel elements are homogenized, and dimensions only approximate TREAT elements.

**Table 2.** Input parameters for the arbitrary simplified TREAT input. Parameters are given for both KENO V.a and KENO-VI inputs.

	KENO V.a	KENO-VI
<b>Number of generations</b>	1500	1500
<b>Particles per generation</b>	10000	10000
<b>Number of regions</b>	21	21
<b>Number of materials</b>	5	5
<b>Flux shapes</b>	5	5
<b>Calculation time</b>	500 min	1400 min

The simplified TREAT inputs transient behavior is simulated by rod withdrawal at time zero, followed by feedback at 1.2, 1.3 and 1.4 seconds with Doppler-broadened cross sections at 301K, 323K and 348K respectively. Cross sections are generated and provided for TDKENO to read in at these specified times. This is done instead of using the feedback model described in Section 2. Other input parameters are given in Table 2. The calculated system parameters are plotted for both KENO V.a and KENO VI to assess the validity of KENO-VI inputs. Power and reactivity, plotted as functions of time, are shown in Figure 4a and 4b, respectively. Inspecting these plots it is clear KENO V.a and KENO-VI agree well with one another.

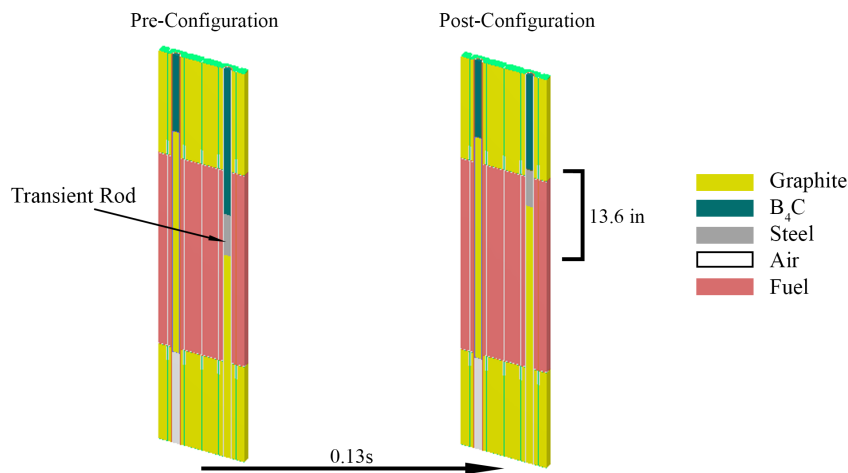


**Figure 4.** Calculated power (a) and reactivity (b) using KENO V.a and KENO-VI inputs.

## 4.4 Simulating TREAT Transients with TDKENO

The transient experiments selected to simulate with TDKENO are based on the M8 Calibrations Tests (M8CAL). They are some of the most recent experiments performed before the 1994 shutdown and have more complete data sets relative to other TREAT experiments. To assess TDKENO's ability to simulate TREAT transients we examine three temperature-limited transients referred to as #2855, #2856, and #2857. These transients define maximum reactivity of the core and establish limiting safety system settings and safety limits for the core loading [11].

Transient behavior is initiated by quickly withdrawing the bank of transient rods causing a spike in power. An example using fuel element cut-outs of the #2855 input in Figure 5 illustrates the two configurations.



**Figure 5.** The pre and post rod configurations are illustrated here for transient #2855. The transient rods are pulled up over 0.13s a distance of 13.6 in. resulting in a power pulse.

Simulating each transient begins by creating several inputs corresponding to the pre- and post-configurations of the TREAT core. Additionally, an adjoint input (identical to pre-transient) is required to calculate the adjoint flux. The forward flux is recalculated by KENO at the specified time steps and recombined with the adjoint flux and amplitude portions each time to generate the total flux. Pre and post-transient inputs may be run as stationary cases with CSAS (Criticality Safety Analysis Sequence) to obtain effective multiplication values ( $k_{\text{eff}}$ ). The difference between pre- and post-transient ( $\Delta k/k$ ) provides an estimate of the reactivity insertion that may be compared with reported values from M8CAL.

For each transient several models were generated. One with KENO V.a inputs and another with KENO-VI inputs with no chamfered corner approximation and fewer regions.

### 4.4.1 Temperature-limited transient 2855

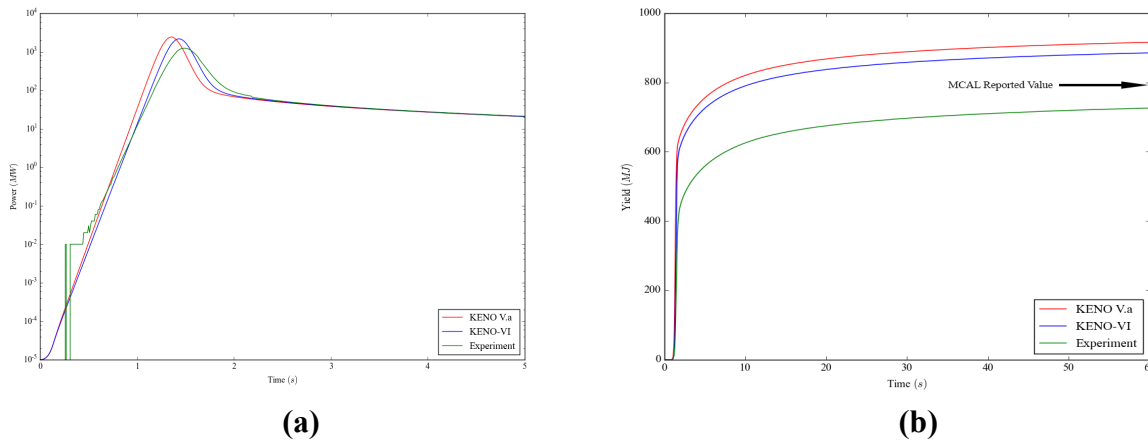
The core loading in each transient experiment remained constant and the control rod removal times and depth were varied. In the temperature-limited transient 2855, transient rods were removed at

time zero and travelled a total of 13.6 in. over 0.13 seconds [11]. The rapid removal results in a power spike, which relaxes due to the negative temperature reactivity feedback [12]. The reported reactivity insertion was 1.8% with no reported uncertainty.

**Table 3.** The reactivity calculations between experiment, KENO V.a inputs, and improved KENO-VI inputs are summarized.  $k_{\text{eff}}$  calculations were run with 2000 generations (first 500 skipped) and 20000 particles with CSAS using ENDF-VII cross-sections. The yield is calculated with TDKENO.

Case	Regions	$k_{\text{eff}}$ Pre-Transient	$k_{\text{eff}}$ Post-Transient	Reactivity Insertion	Yield (MJ)
Experiment	N/A	N/A	N/A	1.8%	792±10%
KENO V.a	11886	1.01019 ±	1.03007	1.967%	916
	4332	0.00012	± 0.00012	± 0.010%	885
KENO-VI		1.01311 ±	1.03143	1.808%	
		0.00014	± 0.00015	± 0.012%	

The  $k_{\text{eff}}$  calculations were performed in multi group and utilized ENDF-VII cross-section libraries. TDKENO simulations with both inputs over estimate the reactivity insertion compared to the experiment. This over estimation is propagated into TDKENO and is observable when calculated parameters are plotted against experiment in Figures 6a and 6b. This is clearly visible in the KENO V.a where a much large power and yield values calculated compared to experiment is observed. The blocky structure of the experimental data is noted during the first half-second and is due to the uncertainty in the detector being on the order of the measured values. As expected, the KENO-VI input agrees better with experiment for reactivity insertion, power and yield. The yield differs from experiment by 11.7%, while KENO V.a differs by 15.7%. This is significant improvement and highlights the need for exact representation of the geometry. Due to time constraints, fewer histories were run with KENO-VI cases than KENO V.a. However, having fewer regions should allow adequate sampling of the system with fewer particles per generation.



**Figure 6.** TDKENO calculated parameters. The KENO V.a inputs were run with 50000 particles and 5000 generations taking approximately 38900 minutes. Due to time constraints KENO-VI inputs were run with 5000 particles and 2000 generations taking approximately 1175 minutes.

Of all the transients analyzed, #2855 had the largest discrepancy between simulation and experiment. Similar analysis in Reference [10] reported higher than experiment values for peak power and yield.

There are a number of possible reasons for this disagreement. One may be slightly incorrect material definitions resulting in a pre-transient  $k_{\text{eff}}$  greater than 1. Another may be the reliance on a simple feedback model as opposed to coupling thermal feedback. Quantifying how far the calculations deviate from experiment is difficult due to apparent inconsistencies in the M8CAL document. For instance, the reported value in Table 3 of  $792 \pm 10\%$  MJ disagreed with the yield of 726 MJ of the experimental plot.

Despite the inconsistencies in the M8CAL data, the agreement is sufficient and TDKENO's ability to accept generalized geometry inputs results in better simulation of transients. Additional histories will be run in the future in order to further validate these conclusions.

#### 4.4.2 Temperature-limited transient 2856

The temperature-limited transient 2856 experiment was carried out by withdrawing the transient rods a distance of 18.60 in. over 0.16 seconds. The experiment was run for 60 seconds and a total of  $1572 \pm 10\%$  MJ was deposited in the TREAT core [11]. It was also reported that the reactivity insertion is 3.01% with no given uncertainty [11]. TDKENO simulations were carried out and compared to experiment. Results are provided in Table 4. In this case the KENO-VI input performed worse than KENO V.a. We suspect the lack of histories may be responsible as only 1500 generations with 2000 particles per generation were run due to time constraints.

**Table 4.** The reactivity calculations between experiment, KENO V.a inputs, and improved KENO-VI inputs are compared.

Case	Regions	$k_{\text{eff}}$ Pre-Transient	$k_{\text{eff}}$ Post-Transient	Reactivity Insertion	Yield (MJ)
Experiment	N/A	N/A	N/A	3.01%	$1572 \pm 10\%$
KENO V.a	11902	$1.01257 \pm 0.00013$	$1.04333 \pm 0.00012$	$3.04\% \pm 0.010\%$	1576
KENO-VI	4332	$1.01303 \pm 0.00015$	$1.04474 \pm 0.00015$	$3.13\% \pm 0.014\%$	1637

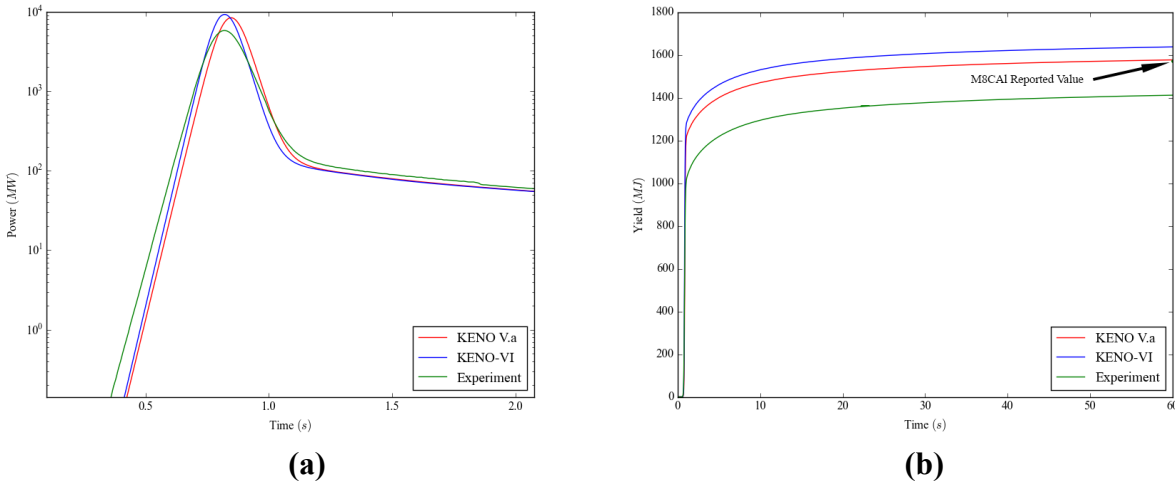
Both calculations are larger than the reported reactivity insertion. Again, this is attributed to the larger than experimental values for peak power and yield in TDKENO. The results of TDKENO with both inputs are in Figures 7a and 7b. Figure 7a has a shorter time scale to highlight the pulse where the deviation between TDKENO and experiment is greatest.

Again there are inconsistencies in the M8CAL document between reported values and plotted data. For example, the yield after 60 seconds is 1400 MJ according to the experimental plot, while the M8CAL document specified a yield of 1572 MJ [11]. If the reported value is correct than



TDKENO is doing well to predict the yield. Issues with the slope to the peak remain most likely due to the lack of thermal feedback.

Overall, the results TDKENO produced with both versions of KENO are promising and with additional development will be able to predict the behavior of TREAT experiments with high degrees of accuracy.



**Figure 7.** Comparison of experimentally reported and TDKENO calculated power (a) and yield (b) over the first 5s of transient 2856. The KENO V.a inputs were run with 50000 particles and 5000 generations taking approximately 38900 minutes. Due to time constraints KENO-VI inputs were run with 2500 particles and 1500 generations taking approximately 1175 minutes.

#### 4.4.3 Temperature-limited transient 2857

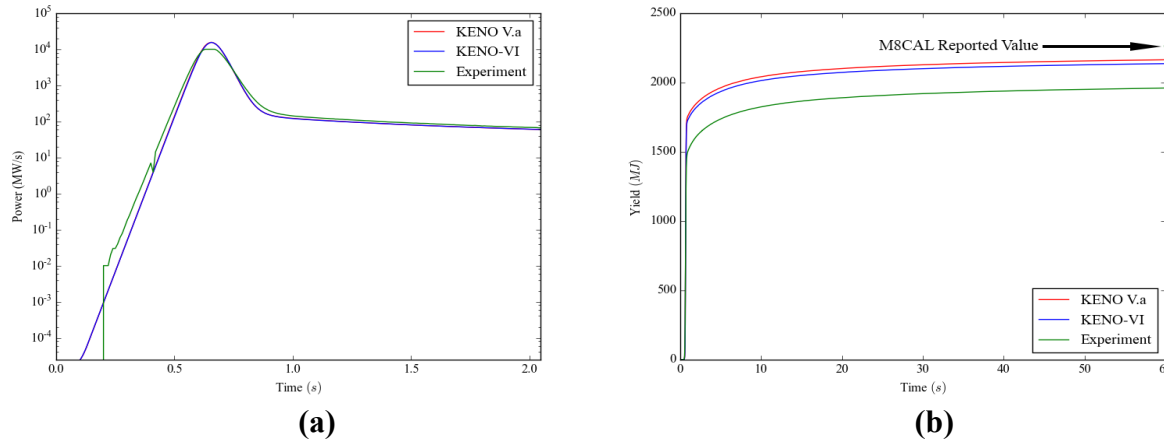
The final experiment simulated with TDKENO is temperature-limited transient #2857. Here the transient rods travelled 21.5 in. over 0.18s [11]. Over the 60 second period, a total of  $2265 \pm 10\%$  MJ was deposited in the TREAT core [11]. Reactivity insertion was reported as 3.84%.

**Table 5.** The reactivity calculations between experiment, KENO V.a inputs, and improved KENO-VI inputs are summarized. Each input is run with 2000 generations (first 500 skipped) and 20000 particles. All inputs are run with KENO using ENDF-VII cross sections.

Case	Regions	Pre-Transient	Post-Transient	Reactivity Insertion	Yield (MJ)
Experiment	N/A	N/A	N/A	3.84%	2265±10%
KENO V.a	11878	1.01416 ± 0.00013	1.05331 ± 0.00013	3.86% ± 0.011%	2164
KENO-VI	4332	1.01510 ± 0.00014	1.05443 ± 0.00014	3.87% ± 0.012%	2136



This experiment was simulated with TDKENO. The results of calculations and experiment are compared in Table 5. Both calculated reactivity insertion values and peak power are slightly over estimated when compared to the experiment. This is not as much of a concern for the peak power as we are more interested in the yield. Once again there is a discrepancy between the experimental data plot and what was reported in M8CAL as is shown in Figure 8b. Though both models of TREAT are calculating a yield value within the uncertainty given in the M8CAL experiment. Other papers, such as Reference [12], suggest using the value given in the M8CAL document, which give us confidence that TDKENO is doing quite well for this transient.



**Figure 8.** Comparison of experimentally reported and TDKENO calculated power (a) and yield (b) over the first 5s of transient 2857. The KENO V.a inputs were run with 50000 particles and 5000 generations. Due to time constraints KENO-VI inputs were run with 2500 particles and 1500 generations.

## 5. CONCLUSION/FUTURE WORK

Through comparison to benchmark and sample problem the integration of KENO-VI into TDKENO appears successful. Using improved inputs made possible with KENO-VI we generally see greater agreement with experiment. The large uncertainties in the M8CAL document and inconsistencies in reported values will require additional comparison between TDKENO and TREAT experiments. To further improve these simulations, TDKENO may be coupled to a thermal hydraulics code. Running these simulations is time consuming. Additional work is underway to incorporate a parallel version of KENO-VI into TDKENO and to accelerate intensive tasks by offloading onto graphics cards. Currently, TDKENO is an excellent tool for transient analysis. Future improvements to the physics and computational time will make it an exceptional tool for calibrating TREAT experiments.

## ACKNOWLEDGEMENTS

The authors would like to thank Zach Weems and Justin Paluch of the University of Florida for providing inputs for TREAT. This project was sponsored by the Idaho National Laboratory under the Department of Energy's Nuclear Energy Advanced Modeling and Simulation (NEAMS) program.

## REFERENCES

- [1] W. J. CARMACK, F. J. GOLDNER, S. M. BRAGG-SITTON, and L. L. SNEAD, "Overview of the U.S. DOE accident tolerant fuel development program", *TopFuel 2013*, Charlotte, North Carolina, Sep. 15-19, 734-739 (2013).
- [2] S. GOLUOGLU and H. L. DODDS, "Three-dimensional Neutron Transport Methodology," *Nuclear Science and Engineering*, **Volume(139)**: pp. 248-261 (2001).
- [3] K. L. GOLUOGLU, "Revitalization and Status of the Hybrid Stochastic/Deterministic Computer Code for Transient, Three-Dimensional Neutron Transport, TDKENO," A practice project for Master of Science degree, The University of Tennessee, Knoxville (2004).
- [4] C. L. BENTLEY, "Improvements in a Hybrid Stochastic/Deterministic Method for Transient, Three-Dimensional Neutron Transport," Ph.D. Dissertation, Nuclear Engineering Department, The University of Tennessee, Knoxville (1996).
- [5] R. D. BUSCH, "KENO V.a Primer: A Primer for Criticality Calculations with SCALE/KENO Va Using CSPAN for Input," ORNL/TM-2002/155. Oak Ridge National Laboratory, Oak Ridge TN (2003).
- [6] S. M. BOWMAN, "KENO-VI Primer: A Primer for Criticality Calculations with SCALE/KENO-VI Using GeeWiz," Oak Ridge National Laboratory, Oak Ridge TN. ORNL/TM-2008/069 (2008).
- [7] M. W. WADDELL, JR. and H. L. DODDS, "A Hybrid Stochastic Deterministic Method for Transient, Three-Dimensional Neutron Transport," *Trans. Am. Nucl. SOC.*, 66, 226 (1992).
- [8] ARGONNE CODE CENTER, "Benchmark Problem Book," ANL-7416, Suppl. 3, Argonne National Laboratory (1977).
- [9] D. KONTOGEORGAKOS, et. al. , "Initial Neutronics Analyses for HEU to LEU Fuel Conversion of the Transient Reactor Test Facility (TREAT) at the Idaho National Laboratory", ANL/GTRI/TM-13/4 (2013).
- [10] D. KONTOGEORGAKOS, H. CONNAWAY, G YESILYURT, and A. WRIGHT, "Neutronic Analysis of the Minimum Original HEU TREAT Core", ANL/GTRI/TM-14/13 (2014).
- [11] W. R. ROBINSON and T.H. BAUER, "The M8CAL Power Calibration Experiment," *ANL-IFR- 212*, URL <http://www.osti.gov/scitech/servlets/purl/1130413> (1994).
- [12] D. KONTOGEORGAKOS, H. CONNAWAY, and A. WRIGHT. "TREAT Transient Analysis Benchmarking for the HEU Core". U.S. URL <http://www.osti.gov/scitech/servlets/purl/1224988> (2014).

# Parallel Performance of the Time Dependent Transport Code TDKENO with Applications to TREAT Simulations

Zander Mausolf<sup>\*</sup>, Mark DeHart<sup>†</sup>, Sedat Goluoglu<sup>\*</sup>

<sup>\*</sup>Nuclear Engineering Program, University of Florida  
529 Gale Lemerand Dr., Gainesville, FL, 32611

<sup>†</sup>Nuclear Systems Design and Analysis Department Idaho National Laboratory  
2525 North Freemont Street, Idaho Falls ID, 83415  
zanderm@ufl.edu<sup>\*</sup>

## INTRODUCTION

Renewed interest in high fidelity simulation of excursion events has prompted the improvement of codes that solve the time-dependent Boltzmann transport equation. Often these codes make approximations to the transport equation in order to achieve results in a reasonable amount of time. One method, the improved quasi-static (IQS), makes few approximations compared to adiabatic, diffusion, and point kinetics [1]. This is the methodology employed by the code TDKENO [2]. One downside to this approach is a portion of the calculation (flux shape) is done with a Monte Carlo code and must be called upon several times. To minimize run time the flux shape calculation is solved with a modified version of KENO-VI from Scale 6.2 [3]. This version of KENO runs in parallel across hundreds of nodes, thus improving performance.

The parallel capabilities of KENO allows TDKENO to simulate complex transient experiments with a large number of histories. These improvements are applied to the simulation of TREAT calibration experiments.

## THEORY

Solving the transport equation is non-trivial when including the time dependence and the explicit representation of delayed neutrons. Transient simulations resulting in significant changes in the flux profile require rigorous methods to solve this "master equation" of reactor physics. In TDKENO, the IQS method enables such simulations. Utilizing IQS method requires assuming the total neutron flux can be factored into two functions, referred to as shape and amplitude [4, 5].

$$\phi(\bar{r}, \bar{\Omega}', E, t) = A(t)\Psi(\bar{r}, \bar{\Omega}', E, t) \quad (1)$$

To make this definition unique the following constraint equation is enforced over time [1].

$$\iiint \frac{1}{v} \Psi(\bar{r}, \bar{\Omega}', E, t) \phi^*(\bar{r}, \bar{\Omega}', E, t) = \text{constant} \quad (2)$$

This quantity should remain constant in time to give the the calculation validity.

The shape takes on the form of the inhomogeneous transport equation. The source consists of a term representing the delayed-neutron precursor decay rate and a term from the backward difference approximation of the shape time derivative. Assuming the shape changes slowly in time allows this derivative to be approximated with a backwards difference approach. In TDKENO the shape equation is solved via Monte Carlo with a modified form of KENO, but in principle the shape equation may be solved with deterministic methods. Reference

[6] provides an excellent rationale for implementing Monte Carlo techniques in IQS. For the complete derivation of the IQS method used in TDKENO, one can refer to the appendix in reference [2].

The amplitude function captures the time dependence and takes on physical significance by being cast into the form of the point kinetics equations as follows:

$$\frac{dA(t)}{dt} = \frac{\rho(t) - \bar{\beta}(t)}{\Lambda(t)} A(t) + \sum_{j=1}^6 \lambda_j C_j(t) + \bar{Q}(t) \quad (3)$$

where  $A(t)$  is the amplitude function,  $\rho$  is reactivity,  $\bar{\beta}$  is the effective delayed neutron fraction,  $\Lambda$  is the generation time,  $\lambda_j$  is the decay constant per neutron group  $j$ ,  $C_j$  is the density of delayed neutron precursors for group  $j$ , and  $\bar{Q}$  is a source. Reactivity, generation time, etc. (referred to as reactivity calculations) in TDKENO are found from their inner product definitions using a linearly interpolated flux shape [2]. Alternatively, these values may be computed during the Monte Carlo random walk but require significant neutron histories to achieve low uncertainties [7]. The shape function varies slowly in time and solves a modified version of the steady state transport equation.

Applying the factorization in Equation 1 to the three-dimensional time-dependent transport equation including delayed neutron results in a set of coupled equations. Equations for the flux shape, amplitude, and delayed neutron precursors are solved on several time scales through an iterative process. The relative sizes are shown in Figure 1.

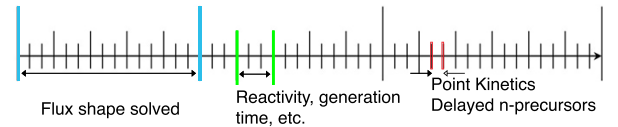


Fig. 1. Time scale variation for IQS.

The advantage of varying these time scales is computational overhead when compared to direct integration. This comes from the flux shape being solved on a large time step. It is only done when the spatial distribution of neutrons changes significantly. In between flux shapes, Equation 3 is solved to capture the time dependent quantities such as power.

## RESULTS AND ANALYSIS

With the integration of the latest version of KENO from SCALE 6.2 we now have the ability to calculate the most inten-

sive portion of the calculation in parallel (flux shape) [3]. Previously, for problems with complex geometry, the calculation of the shape may have taken several days or weeks to achieve adequate sampling and statistics. By running TDKENO at the University of Florida's recently upgraded "HiperGator 2" supercomputer, we are able to run on hundreds of cores. As a result, these same calculations are taking several hours. We apply these improvements to the challenging problem of simulating the experiments done at the Transient Reactor Test Facility (TREAT).

To evaluate the performance of TDKENO, one of the M8CAL experiments is simulated. The simulation is ran on a varying number of the HiperGator's nodes while keeping all other parameters specific to the problem constant. Each node has an Intel E5-2698 v3 (16 core, 2.3 GHz) with 4GB of RAM per node. Note that at present, only the flux shape is calculated in parallel. The amplitude function and related quantities are solved in serial. Investigation is underway to the potential of calculating these quantities on graphic processing units (GPUs).

The M8CAL experiment modeled is referred to in the documentation as the temperature limited transient #2855 [8]. This experiment was chosen as it has proven to be the most difficult to simulate compared to similar transients as shown in [9] and [10]. Previous papers contained experimental data that was thought to have inconsistencies with the values reported in the M8CAL write up [9, 11]. However, upon further inspection, the data that corresponds to the M8CAL reported values has been found, digitized, and compared to the results shown here.

### Computational Improvements

The computational improvements are highlighted in TDKENO by running the same calculation on an increasing number of cores. In this case, the KENO-VI calculations were run with 10000 particles per generation and 2500 generations. A total of 14 flux shapes were calculated with KENO at times primarily centered around the first few seconds. We have chosen these parameters as they appear to give almost identical results to simulations run with many more histories. This is shown in Figure 2. Simulations run with additional histories do result in less statistical uncertainties in values such as fluxes,  $k_{eff}$ , etc. These are of less interest in this paper.

The KENO-VI TREAT model contains approximately 4000 regions with 21 materials. The TREAT core is unique in its ability to safely simulate large reactivity insertions. It accomplishes this with a core composed of 93.1%  $UO_2$  embedded in a graphite matrix, with a ratio of  $UO_2$  to graphite of approximately 1:10000. Details about the TREAT core may be found in [12]. Resonance resolved cross sections are generated in multi-group (238) with SCALE's PMC module with the ENDF/B-VII libraries. A variety of control rods are present in TREAT. The transient shown in this paper is induced by the rapid withdrawal of transient rods over 0.13 seconds.

Below we show the affect of increasing the number of cores the problem is run on. The relatively low number of histories results in poor scaling to large number of cores as observed in Figure 3 due to increased ratio of communication

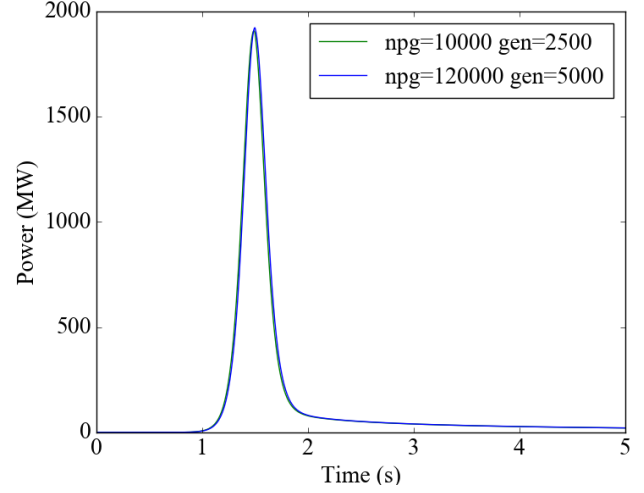


Fig. 2. The calculated power vs. time is compared between TDKENO simulation of TREAT transient #2855 with differing numbers of histories.

to calculation time. The communication time is compounded because the flux shape calculation is done in parallel and then gathered on a single core 14 times. This master processor then computes the point kinetics values in serial. Nevertheless the performance is drastically improved when compared to serial even with a modest amount of processors (for high performance computing standards).

Total Cores	Elapsed Time (minutes)
1	12,347
16	1,560
32	1,619
48	1,356
64	1,051
80	970
96	917
160	810

TABLE I. Variation of the number of cores run for the simulation of #2855. Elapsed time generated from the SLURM submission system [13].

Figure 3 plots the speed up compared to the number of cores run on. Here, speedup is defined as the execution time on a single core divided by the time for that respective run. Improvements could be made to further optimize the inputs by judiciously choosing the number of particles such that an integer amount are simulated on a single core. Such methods may improve load balancing and communication time. However, this is something a user should not be thinking about and these results are more indicative of "real world" behaviour.

### Point Kinetics Variation

As mentioned, the IQS methodology splits the transport equation into several coupled equations solved on three time scales. Each of the scales may be specified by the user. The

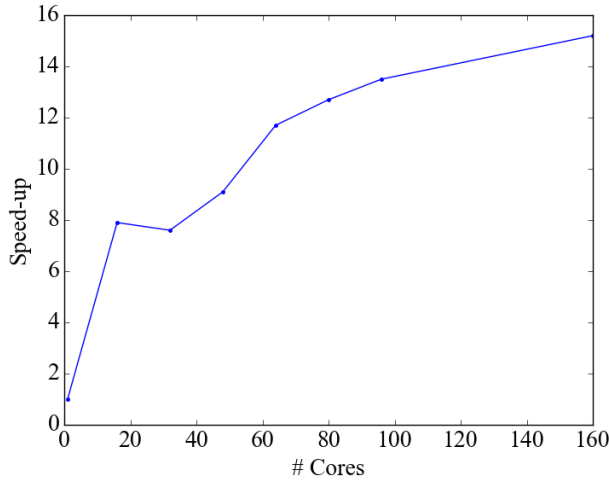


Fig. 3. Speed-up measured relative to the execution time on a single core.

user defines when the flux shape updates are performed. Next, an integer number of reactivity calculations and point kinetics solves are provided and are performed in-between flux shapes. Note, the number of point kinetics solve specified refers to the number performed between the reactivity calculations. To illustrate, say 10 reactivity calculations and 20 point kinetics solves are performed, resulting in a total of 200 calculations between flux shapes. The user may indicate the flux shape is updated at 1s, 3s, and 11s. In this case the time between flux shapes would be 2s and 8s with 200 calculations between each. So while the number of calculations between flux shapes is constant the time scale they are solved on may differ. To visualize this we modify Figure 1 below.

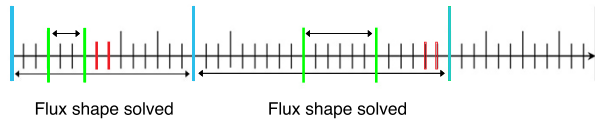


Fig. 4. IQS time scale modified to highlight varying flux shape update intervals.

As stated, the point kinetics equations are formulated to contain the time dependence to capture quantities such as power. In its current state there are no methods to verify if too few calculations are chosen. To ensure no behaviour is overlooked, we have been performing a large number of calculations between flux shapes. Typically this is not as intensive as the flux shape calculation. However, analyzing complex models results in geometries with thousands of regions. The number of regions is directly proportional to the calculation time as each regions reactivity, generation time, etc. must be integrated over the number of regions.

One particular quantity of interest in the TREAT transient calibration experiments is the total energy deposited in the core. Comparison of calculation to experiment is possible as the yield and power as a function of time are reported. In previous papers, the experimental values were not transcribed

properly and were slightly off [9, 11]. What is shown here aligns with the values provided in the original experimental report [8].

To evaluate the sensitivity to the reactivity calculations, several simulations are performed. These simulations vary the number of reactivity calculations between flux shapes while maintaining a constant number of point kinetics solves. We note the flux shape is updated at the following intervals (in seconds): [0.0, 0.13, 0.2, 0.3, 0.4, 0.5, 0.75, 1.0, 1.25, 1.5, 2.0, 3.0, 5.0, 10.0, 20.0, 60.0]

The first 10-90 reactivity calculations maintain the same number of point kinetics solves (10). The 100 line in Figure 5 is what we have been using in previous TREAT simulations, with 100 point kinetics parameters calculations and 100 point kinetics solves. Choosing values beyond this did not alter the final parameters. To exemplify this, a case with 500 point kinetics parameter updates and 500 point kinetics solves was run. Clearly this did not improve the solution as observed in Figure 5

From about 10 to 50 reactivity calculations, there is an obvious deviation that gives a final yield far off from the experimental value. After 60, further increases do not appear to give a different solution. Further examination shows the deviations begin around 10s to 20s. Looking back at the flux shape update times, this is when the time between flux shape updates is the greatest. This is done because the transient event is long over and the spatial distribution of neutrons is not changing much. However, clearly the point kinetics parameters are still varying and need to be updated frequently in order to converge.

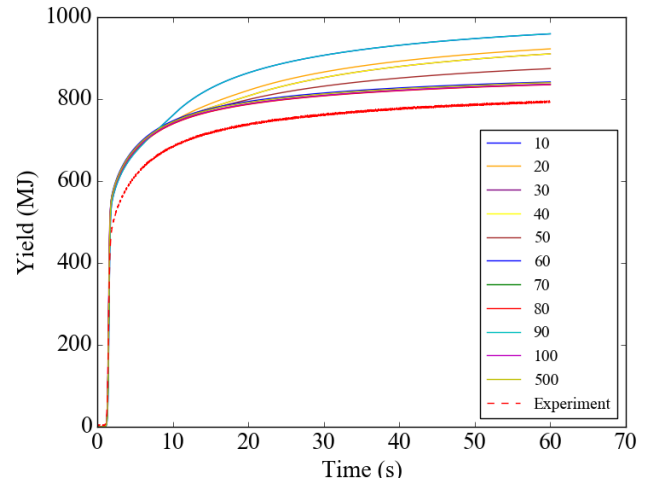


Fig. 5. Varying the number of reactivity calculations between macro time steps.

One way to evaluate how well the system is behaving and when the solution has converged is to look at the constraint condition imposed by Equation 2. At each flux update the constant is computed. In theory this value will remain constant throughout the entire simulation. In this case there is a relatively large deviation from the first to the second shape.



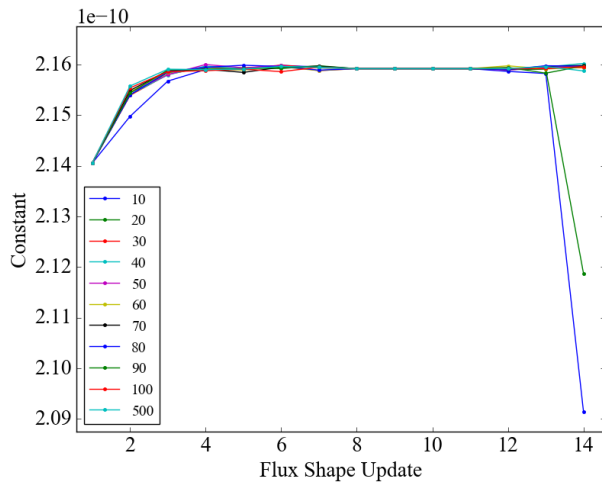


Fig. 6. The constant computed at each flux shape.

This is due to the nature of the TREAT model. For these simulations the initial description of the geometry with the rods in place is used to evaluate the flux shape. At the next step the geometry reflects the transient rods removal. To account for this change a ramp perturbation is done on the cross sections up to the withdrawn configuration.

For this simulation the calculated constant is found at each flux shape update in Figure 6. Examining this figure it becomes clear as the number of reactivity calculations is increased the calculated constants appear to converge to the same values. At the beginning and end of the simulation, the deviations are large with few reactivity calculations. The last few shape calculations take large time steps between. When taking too few reactivity calculations between these large steps the error is compounded. A relatively simple method to ensure the calculation is converging would be to evaluate the constant and compare the difference between the current and previous calculation. That difference may be compared to a set value or to the difference between the initial configuration and the next time step, i.e. flux shape 1 and 2. Such a method would hopefully allow correct solutions to be found with the minimal amount of time.

## CONCLUSIONS

The integration of a parallel flux shape solver has resulted in a transient analysis tool that is up to 15 times faster than previous implementations on a single core. Nuanced behaviour such as the variation of reactivity calculations during a TD-KENO are studied in detail to improve the method. It has been shown for difficult problems like TREAT simulations, the deterministic portion of the code may converge with less computational effort than previously. Additionally, a method for evaluating the convergence of the deterministic calculations is outlined. Future work will be to establish convergence criteria for the reactivity and point kinetics calculations and have these calculations done in parallel.

## ACKNOWLEDGMENTS

This material is based upon work supported by the Idaho National Laboratory.

## REFERENCES

1. A. F. HENRY, "The Application of Reactor Kinetics to the Analysis of Experiments," *Nuclear Science and Engineering*, **3**, 1, 52–70 (1958).
2. C. BENTLEY, *Improvements in a Hybrid Stochastic/Deterministic Method for Transient Three-Dimensional Neutron Transport*, Ph.D. thesis, The University of Tennessee, Knoxville (1996).
3. B. T. REARDEN, M. E. DUNN, D. WIARDA, C. CELIK, K. B. BEKAR, M. L. WILLIAMS, D. E. PELOW, ET AL., "Overview of SCALE 6.2," Tech. rep., Oak Ridge National Laboratory (ORNL) (2013).
4. S. GOLUOGLU and H. DODDS, "A time-dependent, three-dimensional neutron transport methodology," *Nuclear science and engineering*, **139**, 3, 248–261 (2001).
5. J. GEHIN, *A quasi-static polynomial nodal method for nuclear reactor analysis.*, Ph.D. thesis, Massachusetts Institute of Technology (1992).
6. M. SHAYESTEH and M. SHAHRIARI, "Calculation of time-dependent neutronic parameters using Monte Carlo method," *Annals of Nuclear Energy*, **36**, 7, 901–909 (2009).
7. M. WADDELL, *A Hybrid Stochastic/Deterministic Method for Transient Three-Dimensional Neutron Transport*, Ph.D. thesis, The University of Tennessee, Knoxville (1993).
8. W. R. ROBINSON and T. H. BAUER, *The M8 Power Calibration Experiment (M8CAL)* (May 1994).
9. Z. MAUSOLFF, S. GOLUOGLU, and M. DEHART, "Verification of the Time-dependent Transport Code TDKENO with the Monte Carlo Code KENO-VI," PHYSOR 2016, American Nuclear Society (2016), p. 2698.
10. D. KONTOGEORGAKOS, H. CONNAWAY, and A. WRIGHT, "TREAT Transient Analysis Benchmarking for the HEU Core," Tech. rep., Argonne National Laboratory (ANL) (2014).
11. J. PALUCH, , D. POPP, S. GOLUOGLU, and M. DEHART, "Application of TDKENO to TREAT Temperature-Limited Transients," PHYSOR 2016, American Nuclear Society (2016).
12. J. D. BESS and M. D. DEHART, "Baseline assessment of TREAT for modeling and analysis needs," Tech. rep., Idaho National Laboratory (INL), Idaho Falls, ID (United States) (2015).
13. A. B. YOO, M. A. JETTE, and M. GRONDONA, "Slurm: Simple linux utility for resource management," in "Workshop on Job Scheduling Strategies for Parallel Processing," Springer (2003), pp. 44–60.

# CONTINUOUS TEMPERATURE REPRESENTATION IN COUPLED OPENMC/MOOSE SIMULATIONS

**Matthew Ellis\*, Benoit Forget, and Kord Smith**

Department of Nuclear Science and Engineering  
Massachusetts Institute of Technology, Cambridge, MA, USA  
[mellis13@mit.edu](mailto:mellis13@mit.edu)

**Derek Gaston**

Fuels Modeling and Simulation  
Idaho National Laboratory, Idaho Falls, Idaho, USA

## ABSTRACT

In recent years the use of Monte Carlo methods for modeling reactors has become feasible due to the increasing availability of massively parallel computer systems. One of the primary challenges yet to be fully resolved, however, is the efficient and accurate inclusion of multiphysics feedback in Monte Carlo simulations. The research in this paper presents the application of an existing numerical method for stochastically transporting particles in a material with continuously varying properties for incorporation of multiphysics feedback in Monte Carlo simulations. The implemented continuous material transport method allows for continuous representations of temperature and nuclide density to be used in OpenMC simulations which allows for multiphysics feedback information to be used in OpenMC without needing to discretize the geometry beyond the initial constructive solid geometry (CSG) representation. Functional Expansion Tallies (FETs), also implemented in OpenMC, allow multiphysics data to be efficiently exchanged with Multiphysics Object-Oriented Simulation Environment (MOOSE) applications through passing of polynomial expansion coefficients, minimizing data transfer between applications. In this paper these methods of incorporating multiphysics feedback are shown to be accurate in correctly calculating Doppler feedback in a light water reactor (LWR) assembly containing an asymmetric loading of burnable poisons. Both the eigenvalue and spatially varying U-238 capture rates are correctly calculated using the continuous material transport method.

*Key Words:* **Monte Carlo, Multiphysics, OpenMC, MOOSE**

## 1. INTRODUCTION

Doppler feedback is an important component of multiphysics feedback in reactor simulations. It directly affects the accuracy of eigenvalue, fission power distribution, and isotope reaction rate predictions. In the initial publication demonstrating the coupling of OpenMC to MOOSE applications, volume average temperatures were used to demonstrate functioning multiphysics feedback in OpenMC

---

\*Corresponding author

simulations [1]. However, volume averaged temperatures produce inaccurate Doppler feedback in light water reactor (LWR) simulations because of spatial self-shielding effects [2]. Due to spatial self-shielding effects, the temperature distribution at the periphery of the pin is particularly important for correctly predicting U-238 capture rates which drive the Doppler feedback effect in LWRs. Historically, if spatial discretization was not available, weighting functions were applied to pin temperature distributions to produce an effective average temperature that preserves reactivity changes [2]. However, the aforementioned method undermines the generality of the multiphysics feedback and prohibits the application of these feedback methods to other reactor designs.

This paper describes the implementation in OpenMC of a continuous material tracking methodology first introduced by Brown and Martin [3]. This continuous material transport method allows for continuous representations of temperature and nuclide density which is used in OpenMC to properly capture Doppler feedback from applications built on the MOOSE framework. The continuous material tracking methodology was chosen because it does not require additional spatial discretization in the Monte Carlo simulation to include multiphysics feedback. This alleviates many of the data transfer and mapping challenges that plague multiphysics simulations, especially those that couple Monte Carlo and finite element simulations. This paper will show the ability of the continuous material tracking algorithm to properly calculate Doppler feedback in OpenMC simulations.

It should be noted that integrating multiphysics feedback into Monte Carlo simulations is not, by itself, a novel concept. The first coupling of OpenMC and MOOSE differed from the work related to MC21 [4] [5], Serpent 2 [6][7][8], and MCNP [9] by utilizing Functional Expansion Tallies (FETs) to transfer OpenMC tallies to MOOSE applications as polynomials instead of zeroth order, discontinuous quantities [1]. The work in this paper further differentiates the coupling of OpenMC and MOOSE from other multiphysics couplings by utilizing the general continuous material tracking method to incorporate Doppler feedback into OpenMC reactor simulations. By utilizing FETs and the continuous material transport method, it is possible to couple OpenMC to MOOSE applications without discretizing fuel elements in the Monte Carlo simulation or volume averaging quantities for transferring and using multiphysics data.

In Section 2 the continuous material tracking method will be presented, and in Section 3 the software packages and multiphysics coupling will be described. In Section 4 the accuracy of the continuous tracking methodology as applied to LWR applications will be evaluated using an assembly from the Benchmark for Evaluation and Validation of Reactor Simulations (BEAVRS) [10]. In Section 5 results from this paper will be summarized, and finally, in Section 6 future work will be described.

## 2. Continuous Material Tracking Methodology

The continuous material tracking algorithm depends on the ability to integrate the total macroscopic cross section along the particle flight path [3]. In the case of a reactor simulation an analytic expression for the spatially varying total cross section is not available because of the complexity introduced by



spatial variations in temperature and number density along the neutron flight path. Therefore, the integral of the total cross section along the neutron flight path must be performed numerically.

The integral of the total macroscopic cross section along the particle flight path is referred to as the optical depth and is defined in Eq. (1). Note that the macroscopic cross section is a function of temperature, isotope number density, and material density which will vary along the neutron's flight path. In the current implementation the total macroscopic cross section is evaluated at  $n$  points along the flight path, and Simpson's rule is used to calculate the optical depth.

$$\tau(x) = \int_0^x \Sigma(T(x'), N_i(x'), \rho(x')) dx' \quad (1)$$

The probability that a neutron travels along the flight path to a point without experiencing a collision is the exponential of the negative optical depth. This probability of no-collision is used to initially determine if the particle makes a collision in the cell. If the neutron does collide in the cell, a truncated exponential PDF must be sampled to determine the flight distance. The truncated exponential is defined in Eq. (2)

$$g(x') = \frac{1}{G} \frac{d\tau}{dx'} e^{-\tau(x')} \quad (2)$$

where,

$$G = 1 - e^{-\tau(x_b)} \quad (3)$$

In the above equation,  $x_b$  denotes the distance to the nearest cell boundary along the particle path. Sampling from the truncated exponential becomes difficult because it requires the following two equations be satisfied:

$$\hat{\tau} = \ln(1 - G\xi) \quad (4)$$

$$\hat{\tau} = \int_0^s \Sigma(x') dx' \quad (5)$$

The sampled optical depth in Eq. (4) is evaluated using a random number,  $\xi$ , and the previously calculated optical depth to the cell boundary. The expression in Eq. (5), however, must be solved for the flight distance,  $s$ , that matches the sampled optical depth from Eq. (4). Because an analytic expression is not available for the total cross section along the particle flight path, the flight distance,  $s$ , must be solved for iteratively. The implementation in OpenMC uses the same Newton iteration method that was presented in the original work by Brown and Martin. It should be noted that for each Newton iteration, the optical depth integral must be evaluated numerically which is the main computational cost of this general methodology. For clarity, the Newton iteration method is shown below:

```

Select  $s_0$ 
 $n = 0$ 
while  $|s_n - s_{n-1}| < \epsilon$  do
   $n = n + 1$ 
   $g = \hat{\tau} - \hat{\tau}(s_{n-1})$ 
   $g' = dg/ds = -\Sigma(x_0 + s_{n-1})$ 
   $s_n = s_{n-1} - g/g'$ 
end while

```

### 3. THE OPENMC/MOOSE COUPLING SCHEME

#### 3.1. Code Descriptions

OpenMC is a Monte Carlo code specifically optimized for reactor analysis [11]. The primary advantage of utilizing OpenMC for this study is the on-the-fly windowed multipole method for Doppler broadening microscopic cross sections that has been implemented [12][13]. This capability provides a rigorous methodology for incorporating Doppler feedback into the Monte Carlo simulation without needing a weighting or interpolation methodology applied to pregenerated cross sections at fixed temperature intervals. Additionally, the windowed multipole method has a very small memory footprint and is as efficient as single temperature ACE file lookups [12].

MOOSE is an open source framework for numerically solving partial differential equations utilizing the Finite Element Method (FEM) and has been used to create many multiphysics applications specifically relevant to nuclear reactor design and analysis [14]. The principal advantage of MOOSE is the ease with which multiphysics applications can be coupled together through the *MultiApp* system [15]. Many of the systems in MOOSE that handle the transfer of data between multiphysics applications were critical in being able to construct generalized interfaces between the tallies in OpenMC and unstructured finite element meshes in MOOSE applications.

#### 3.2. Description of the OpenMC/MOOSE Coupling Framework

Because MOOSE and OpenMC are open source software packages, it is possible to couple them internally and not rely on external scripts to parse output files and create input files. In the present implementation OpenMC is built as a library such that its subroutines can be accessed directly from MOOSE, and as a result, OpenMC fits naturally into the *MultiApp* system. The hierarchy of the *MultiApps* for this preliminary coupling is shown in Fig. 1.

Fig. 1 contains two MOOSE *MultiApps*; OpenMC and Cerberus. The Cerberus *MultiApp* was created as a surrogate for an advanced fuel performance application like BISON [16]. However, all of the

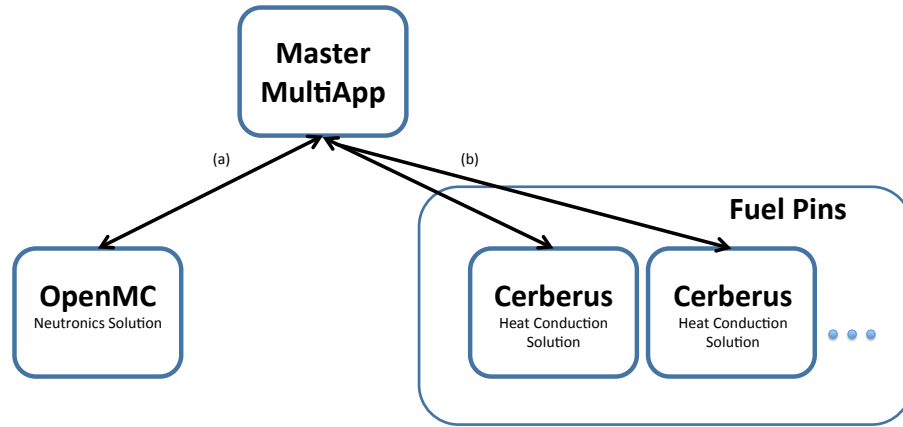


Figure 1: MultiApp Hierarchy for MOOSE and OpenMC Coupling.

coupling techniques discussed in this paper are easily ported to other MOOSE applications like BISON in order to obtain a more accurate fuel performance solution. While the fuel performance code BISON contains very sophisticated physics, Cerberus contains a simple heat conduction solver.

The coupled simulation consists of two *MultiApp* executions and four *MultiApp* transfers per iteration. At the beginning of an iteration, the OpenMC *MultiApp* is executed using an initial temperature distribution or the temperature distribution from the previous iteration. These temperature distributions are stored in OpenMC as coefficients of a polynomial expansion. In the case of an LWR lattice, each fuel pin material has a unique set of coefficients that are used to evaluate the temperature of the material at any point within the OpenMC fuel cell. This polynomial representation of temperature is a convenient and efficient way of retrieving Doppler feedback information from MOOSE applications because it only requires storing and transferring a small set of coefficients instead of a large set of pointwise temperature values. For the LWR lattice example that will be presented in this paper, the fuel temperature profiles are represented as a linear combination of Zernike polynomials that were described in the previous paper on coupling MOOSE and OpenMC [1].

When the OpenMC execution is complete, FET coefficients for the pin power distributions are transferred to the master *MultiApp*. The details of calculating FET coefficients will not be covered in this paper, but a thorough derivation is available in the original work by Griesheimer [17] [18] and the specific details of the implementation of FETs in OpenMC are discussed in a previous paper [1]. It should be noted, however, that the FETs are limited to collision based tallying because of the complexity of polynomial shapes and continuously varying material properties.

The master *MultiApp* then passes a set of FET coefficients to each instance of the Cerberus *MultiApp*. The Cerberus *MultiApps* are executed in parallel and each application's temperature solution is integrated across the entire fuel pin to determine the coefficients of a single polynomial representation of the temperature field. The coefficients of the temperature field expansion are transferred to the master *MultiApp* and then to the OpenMC *MultiApp* to be stored in a distributed material data structure in

OpenMC. In this coupling scheme, each instance of the Cerberus *MultiApp* has a unique set of FET coefficients and returns a unique set of temperature field coefficients. The temperature expansion coefficients are then used to transport neutrons in a continuously varying temperature field within fuel pins in OpenMC.

#### 4. ANALYSIS OF CONTINUOUS MATERIAL TRACKING ACCURACY IN LWR SIMULATIONS

In this section an LWR assembly from BEAVRS [10] will be analyzed using the multiphysics feedback framework described in the previous sections. The BEAVRS assembly that will be used in this simulation is shown in Fig. 2. Note that the burnable poison (BP) rods are in the upper-left corner of the assembly. Only a two-dimensional model of the assembly was run for this calculation.

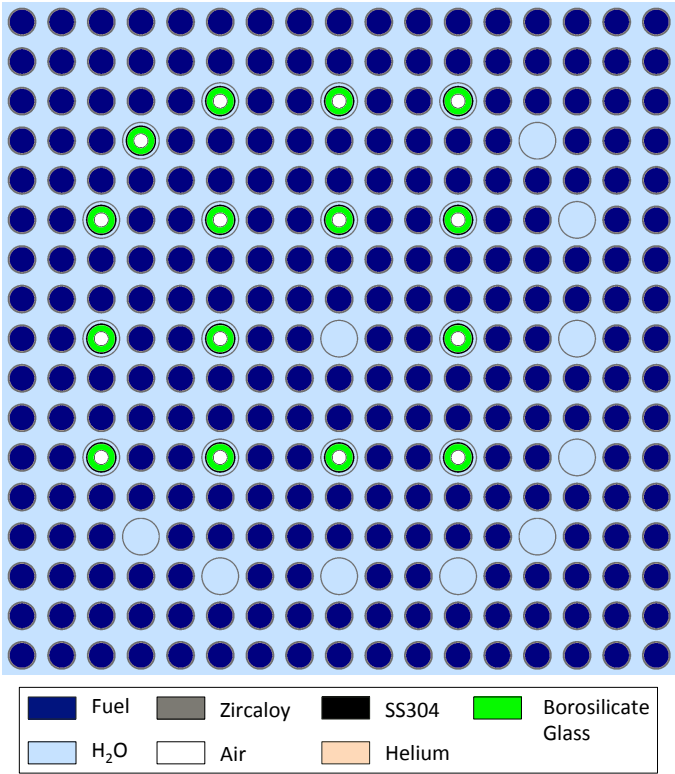


Figure 2: 15 BP 3.1% enriched BEAVRS assembly colored by material ID.

In the Monte Carlo simulation, only the fuel material was Doppler broadened. All of the other isotopes in the BEAVRS model were analyzed at the nearest available cross section data set of 326.85 °C (600K), which is very close to the operating temperature of the moderator in a PWR. The number densities of the moderator were calculated at 310 °C to closely match the average of the inlet and outlet coolant temperatures in a PWR. Finally, a fixed wall temperature boundary condition was imposed on each fuel





pin-average temperature across the entire assembly is bounded by approximately one degree Celsius, and the maximum percent change in pin U-238 capture rate is bounded by approximately 0.25%. The converged volumetric power distribution and temperature distribution are shown in Fig. 6 and Fig. 7, respectively.

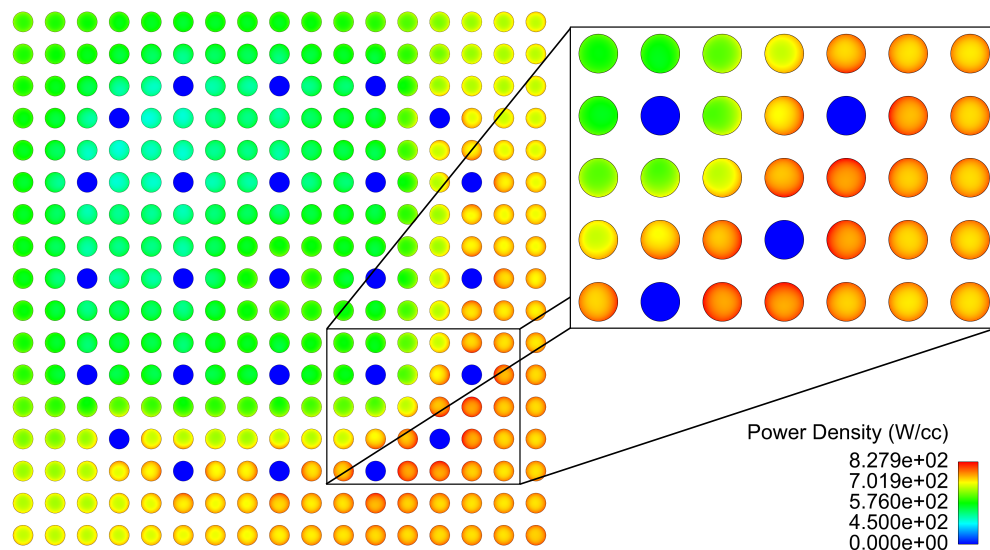


Figure 6: Converged power distribution for the BEAVRS assembly using sixth order Zernike polynomials for temperature distributions and fourth order Zernike polynomial FETs for power distributions

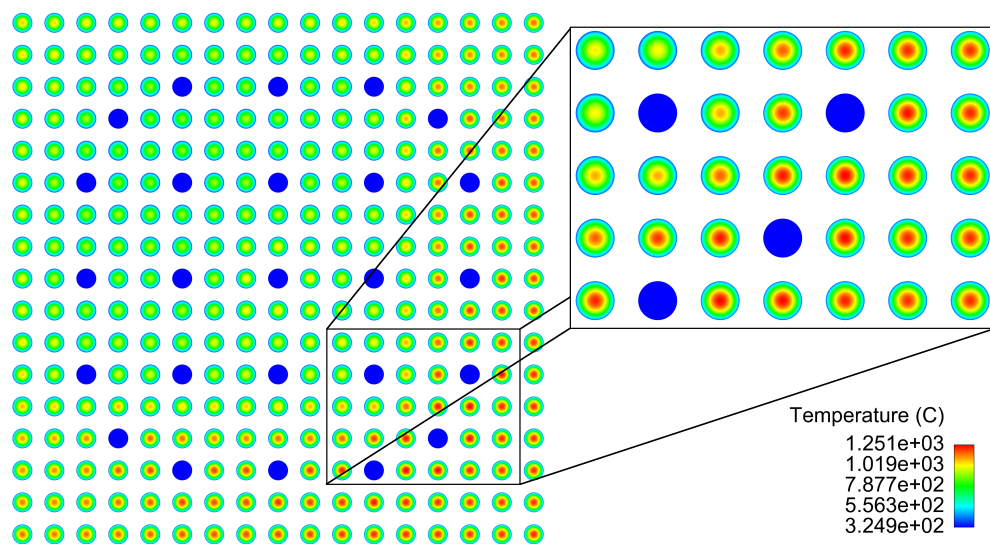


Figure 7: Converged temperature distribution for the BEAVRS assembly using sixth order Zernike polynomials for temperature distributions and fourth order Zernike polynomial FETs for power distributions

While it appears that the coupled iterations between OpenMC and Cerberus converge to a solution, it is important to address the accuracy of the continuous transport methodology implemented in OpenMC. To demonstrate the accuracy of the method, continuous temperature profiles from the final coupled iteration were volume averaged into a mesh of concentric rings with azimuthal cuts in each fuel pellet. For the purpose of evaluating accuracy, discretized fuel pellets contained 20 radial rings with 8 azimuthal sectors. In both the discrete and continuous temperature profile cases, the U-238 capture rate was tallied in 20 equal volume rings in each fuel element. It should be emphasized, however, that the 20 ring tally with the continuous material transport was still accomplished without discretizing fuel elements by implementing a new tally type in OpenMC specifically designed for this analysis. This new tally type used the radius of the fuel pin cylinder and the neutron's position to determine the correct radial ring for tallying. In the interest of brevity, we will refer to the simulation with discrete temperature profiles as the 'discrete case' and the simulation with the continuously varying temperature profile as the 'continuous case'. Also note that for the purposes of this comparison OpenMC was run without coupling to MOOSE applications.

The eigenvalue for the discrete case was  $1.07437 \pm 0.00002$ , and the eigenvalue for the continuous case was  $1.07434 \pm 0.00002$ . The difference in these eigenvalues is not statistically significant with a calculated t-score of 1.0607. Note that the t-score is defined as  $\frac{\bar{x}_1 - \bar{x}_2}{\sqrt{SE_{\bar{x}_1}^2 + SE_{\bar{x}_2}^2}}$  where SE indicates the standard error and  $\bar{x}_i$  is a tally mean. A more important measure of accuracy, however, is the spatial distribution of the U-238 capture within the fuel pins. The twenty radial ring U-238 capture rates were compared for the discrete and continuous cases for all pins, and a t-score was calculated for the difference in capture rates for all of the tally regions. The maximum t-score from all 5,280 tally regions was -4.03. While a t-score of this magnitude would suggest that the null hypothesis that all of the means are equal should be rejected, it is reasonable to expect that some regions will have large t-scores because of the stochastic nature of the solution process. Fig. 8 shows the percent difference in capture rate with the associated t-score for the fuel pin with the largest t-score value.

The difference in the U-238 capture rates shown in Fig. 8 is representative of the overall trends in the difference in U-238 capture rates throughout the assembly. That is, there are sporadic instances of U-238 capture tallies being statistically different, however, the radial position of these statistically significant differences appear to be stochastic in nature. Moreover, out of 5,280 U-238 tally regions, only 0.37% of the U-238 capture tallies have a t-score outside of  $\pm 3.0$ . Similarly, 4.60% of the U-238 capture tallies have a t-score outside of  $\pm 2.0$ . The density of U-238 tally t-scores across the entire assembly are plotted in Fig. 9. The distribution of t-scores shown in Fig. 9 qualitatively does not show a significant bias in over predicting or under predicting U-238 capture rates in the assembly. In fact, the mean, standard deviation, skewness, and excess kurtosis of the t-scores that were calculated closely resemble a normal distribution which would be consistent with purely random differences. Unfortunately, normality tests, such as the Anderson-Darling test [20], indicate that the distribution of t-scores does not come from a purely normal distribution (the Anderson-Darling test indicates that the null hypothesis that the t-scores come from a normal distribution can be rejected with a 1% significance level). However, The lack of normality in the t-score distributions is not critically important for assessing if the continuous material tracking methodology is accurate, but is mentioned for completeness. In fact, completely identical solutions should not be expected because there is truncation error inherent





## 5. CONCLUSION

In this paper it was shown that the continuous material transport algorithm first proposed by Brown and Martin [3] has been successfully implemented in OpenMC to capture multiphysics feedback from MOOSE-based applications. The implemented continuous material transport method allows for continuous representations of temperatures and density to be used in OpenMC. This allows for multiphysics feedback information to be used without further discretization of the geometry beyond the initial CSG representation. This method of multiphysics data transfer was shown to be accurate in correctly calculating Doppler feedback in an LWR assembly containing an asymmetric loading of burnable poisons. Both the eigenvalue and spatially varying fuel U-238 capture rates were correctly calculated by the continuously varying material transport method. These results show that accurate multiphysics feedback can be incorporated into LWR simulations without requiring large amounts of spatial discretization to accommodate deterministic methods that solve related physics on an entirely different geometric representation.

## 6. FUTURE WORK

The results in this paper show that the continuous material transport method can be used to accurately include multiphysics feedback in Monte Carlo simulations. However, the method, as presented, is not optimized for minimization of number of integration points along a neutron's flight path or the convergence criterion for neutron flight distance. Along with further code optimization, optimizing the number of integration points and convergence criterion is needed for a performance comparison between the continuous material transport method and the traditional discretized approach. This optimization, however, will require a more detailed investigation that is beyond the scope of this paper. The main challenge in this analysis will be to understand the interaction between the truncation error associated with the numerical integration of the optical depth and the averaging of this error that will occur because of the stochastic nature of Monte Carlo.

Future work will include the use of the MOOSE applications BISON and RELAP-7. With the use of BISON, thermal feedback accuracy will increase due to utilization of temperature dependent material properties and explicit treatment of the gas gap between the fuel and cladding. Additionally, coupling of RELAP-7 to OpenMC will enable changes in moderator density which is important for fully capturing multiphysics feedback.

## ACKNOWLEDGMENTS

The research work presented in this paper was supported by the Idaho National Laboratory under DOE Idaho Operations Office Contract DE-AC07-05ID14517.

## REFERENCES

- [1] M. Ellis *et al.* “Preliminary coupling of the Monte Carlo code OpenMC and the Multiphysics Object-Oriented Simulation Environment (MOOSE) for analyzing Doppler feedback in Monte Carlo simulations.” *ANS MC2015 - Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method* (2015).
- [2] G. Grandi *et al.* “Effect of Casmo-5 cross-section data and Doppler temperature definitions on LWR reactivity initiated accidents.” In: *PHYSOR 2010 - Advances in Reactor Physics to Power the Nuclear Renaissance* (2010).
- [3] F. B. Brown and W. R. Martin. “Direct sampling of Monte Carlo flight paths in media with continuously varying cross-sections.” In: *Proc. ANS Mathematics & Computation Topical Meeting*, volume 2 (2003).
- [4] D. P. Griesheimer *et al.* “An integrated thermal hydraulic feedback method for Monte Carlo reactor calculations.” In: *International Conference on the Physics of Reactors: “Nuclear Power: A Sustainable Resource”*. Interlaken, Switzerland. (2008).
- [5] D. F. Gill, D. L. Aumiller, and D. P. Griesheimer. “Monte Carlo and thermal-hydraulic coupling via PVMEXEC.” In: *PHYSOR 2014 - The Role of Reactor Physics toward a Sustainable Future* (2014).
- [6] V. Valtavirta, T. Viitanen, and J. Leppänen. “Internal neutronics-temperature coupling in Serpent 2.” *Nuclear Science and Engineering*, **177**(2): pp. 193–202 (2014).
- [7] J. Leppanen, V. Valtavirta, and T. Viitanen. “Unstructured mesh based multi-physics interface for CFD code coupling in the Serpent 2 Monte Carlo code.” In: *PHYSOR 2014 - The Role of Reactor Physics toward a Sustainable Future* (2014).
- [8] M. Daeubler *et al.* “High-fidelity coupled Monte Carlo neutron transport and thermal-hydraulic simulations using Serpent 2/SUBCHANFLOW.” *Annals of Nuclear Energy*, **83**: pp. 352 – 375. ISSN 0306-4549 (2015).
- [9] A. Ivanov *et al.* “High fidelity simulation of conventional and innovative LWR with the coupled Monte-Carlo thermal-hydraulic system MCNP-SUBCHANFLOW.” *Nuclear Engineering and Design*, **262**: pp. 264 – 275. ISSN 0029-5493 (2013).
- [10] N. E. Horelik *et al.* *Benchmark for Evaluation and Validation of Reactor Simulations. Technical report*, Massachusetts Institute of Technology (2013).
- [11] P. K. Romano and B. Forget. “The OpenMC Monte Carlo particle transport code.” *Annals of Nuclear Energy*, **51**: pp. 274–281 (2013).
- [12] C. Josey *et al.* “Windowed multipole for cross section Doppler broadening.” *Journal of Computational Physics* (2015).

- [13] C. Josey, B. Forget, and K. Smith. “Windowed multipole sensitivity to target accuracy of the optimization procedure.” *Journal of Nuclear Science and Technology*, **52**: pp. 987–992 (2015).
- [14] D. Gaston *et al.* “MOOSE: A parallel computational framework for coupled systems of nonlinear equations.” *Nuclear Engineering and Design*, **239(10)**: pp. 1768–1778 (2009).
- [15] D. R. Gaston *et al.* “Physics-based multiscale coupling for full core nuclear reactor simulation.” *Annals of Nuclear Energy* (2014).
- [16] C. Newman, G. Hansen, and D. Gaston. “Three dimensional coupled simulation of thermomechanics, heat, and oxygen diffusion in UO<sub>2</sub> nuclear fuel rods.” *Journal of Nuclear Materials*, **392(1)**: pp. 6–15 (2009).
- [17] D. P. Griesheimer. *Functional Expansion Tallies for Monte Carlo Simulations*. Ph.D. thesis, The University of Michigan (2005).
- [18] D. P. Griesheimer, W. R. Martin, and J. P. Holloway. “Estimation of flux distributions with Monte Carlo Functional Expansion Tallies.” *Radiation Protection Dosimetry*, **115(1-4)**: pp. 428 – 432. ISSN 0144-8420 (2005).
- [19] D. Andrs *et al.* *RELAP-7 level 2 milestone report: Demonstration of a steady state single phase PWR simulation with RELAP-7. Technical report*, Idaho National Laboratory (2012).
- [20] T. W. Anderson and D. A. Darling. “A test of goodness of fit.” *Journal of the American Statistical Association*, **49(268)**: pp. 765–769 (1954).

# A CUMULATIVE MIGRATION METHOD FOR COMPUTING RIGOROUS TRANSPORT CROSS SECTIONS AND DIFFUSION COEFFICIENTS FOR LWR LATTICES WITH MONTE CARLO

**Zhaoyuan Liu, Kord Smith and Benoit Forget**

Department of Nuclear Science and Engineering

Massachusetts Institute of Technology, Cambridge, Massachusetts, USA

[liuzy@mit.edu](mailto:liuzy@mit.edu), [kord@mit.edu](mailto:kord@mit.edu), [bforget@mit.edu](mailto:bforget@mit.edu)

## ABSTRACT

A new method for computing homogenized assembly neutron transport cross sections and diffusion coefficients that is both rigorous and computationally efficient is proposed in this paper. In the limit of a homogeneous hydrogen slab, the new method is equivalent to the long-used, and only-recently-published CASMO transport method. The rigorous method is used to demonstrate the sources of inaccuracy in the commonly applied “out-scatter” transport correction. It is also demonstrated that the newly developed method is directly applicable to lattice calculations performed by Monte Carlo and is capable of computing rigorous homogenized transport cross sections for arbitrarily heterogeneous lattices. Comparisons of several common transport cross section approximations are presented for a simple problem of infinite medium hydrogen. The new method has also been applied in computing 2-group diffusion data for an actual PWR lattice from BEAVRS benchmark.

*Key Words:* **Transport Cross Section, Diffusion Coefficient, Transport Correction, Monte Carlo, Cumulative Migration Area**

## 1. INTRODUCTION

Many deterministic nuclear reactor calculations utilize transport-corrected- $P_0$  transport or diffusion theory to model neutron transport within fuel assemblies and nearby reflecting regions. The accuracy of such core models is inherently tied to approximations made in obtaining multi-group transport cross sections or diffusion coefficients. While classic reactor physics textbooks [1, 2] offer insights and plausible arguments for computing transport cross sections and diffusion coefficients, there appears to be no rigorous theory for, nor quantification of errors introduced by, these approximations. Consequently, the computational accuracy of both heterogeneous (e.g. explicit fuel pin) and nodal (e.g. homogenized fuel assemblies) core calculations is often seriously compromised by inaccurate transport approximations, and little guidance is available in the literature to assist code developers and analysts in choosing the appropriate transport approximation.

## 1.1. Background

The generation of multi-group cross section data for LWR analysis usually starts by identifying some characteristic “lattice” — be it a pin-cell, a fuel assembly, or a collection of fuel assemblies. For each such lattice, a very-fine-group transport calculation (e.g., 50 – 10,000 groups) is performed to obtain the neutron flux and reaction rate distributions within the lattice. Unless this transport calculation explicitly models anisotropic scattering, an approximation for transport-corrected- $P_0$  cross sections for each nuclide must be introduced before the multi-group lattice transport calculation can be performed.

In addition, lattice reaction rates and fluxes are used to compute energy-condensed and/or spatially-homogenized transport cross sections (or diffusion coefficients) for use in downstream multi-group (e.g., 2 – 100 groups) core calculations. Here, additional approximations are required to compute the appropriate transport cross section that preserves some selected characteristic of the lattice calculation.

All production lattice physics codes [3–6] make such approximations, often without substantial justification. Moreover, the most useful of these approximations are often considered to be proprietary, and the literature remains largely silent on useful methods. One example might be that of the transport-corrected- $P_0$  methods that have been employed in CASMO for more than 40 years. Only recently has Herman [7] published details of the method used in CASMO to generate transport-corrected- $P_0$  cross sections for  $^1\text{H}$  in LWR lattices. Herman was able to compute CASMO’s “exact” transport cross section that matched continuous-energy Monte Carlo (MC) neutron leakages (integrated into 70 fine energy groups) from a slab of pure hydrogen. This transport correction is markedly different from that computed using the “micro-balance” argument [8] which produces the classic “out-scatter” approximation — with its transport-to-total ratio of 1/3 for purely isotropic center-of-mass neutron scattering with free gas  $^1\text{H}$  model. CASMO developers recognized long ago that this definition of transport cross section produced excellent eigenvalues for small LWR critical assemblies with large neutron leakages, while the classic out-scatter approximation produced errors in eigenvalue as large as 1000 pcm. In addition, SIMULATE-3 nodal code developers observed (more than 30 years ago) that the CASMO transport cross section also produced two-group diffusion coefficients that eliminated radial power tilts observed in large 4-loop PWR cores when using the out-scatter approximation.

## 1.2. Approximation Methods For Transport Cross Sections

Many approximation methods for computing diffusion coefficients has been investigated in the past 40 years, among which the “out-scatter” approximation based on the “micro-balance” argument is probably the most often used one, assuming that the in-scatter rate of neutrons from energies  $E'$  to  $E$  will approximately balance the out-scatter rate of neutrons from  $E$  to all other energies. The approximation

can be represented as

$$\int_0^\infty \Sigma_{s1}(\vec{r}, E' \rightarrow E) \vec{J}(\vec{r}, E') dE' \approx \int_0^\infty \Sigma_{s1}(\vec{r}, E \rightarrow E') \vec{J}(\vec{r}, E) dE' \quad (1)$$

in which  $\Sigma_{s1}(\vec{r}, E' \rightarrow E)$  is  $P_1$  scattering cross section from  $E'$  to  $E$  at  $\vec{r}$ , and  $\vec{J}(\vec{r}, E')$  is the neutron current of energy  $E'$  at  $\vec{r}$ . Based on this approximation, the multi-group transport cross section can be derived to be the expression in Equation (2).

$$\Sigma_{tr,g}^{os} = \Sigma_{t,g} - \bar{\mu}_g \Sigma_{s0,g} \quad (2)$$

in which  $\Sigma_{tr,g}^{os}$  is the transport cross section from the out-scatter approximation,  $\Sigma_{t,g}$  is total cross section,  $\Sigma_{s0,g}$  is  $P_0$  scattering cross section,  $\bar{\mu}_g$  is the average scattering cosine of neutron, and all the subscript  $g$  denotes the group index. The spatial dependence on  $\vec{r}$  is omitted in most equations in this paper for a clearer expression and generally all the terms refer to the same spatial position.

Since in the perspective of neutrons in nuclear reactor physics, the elastic scattering with  $^1\text{H}$  can be seen as purely isotropic in center-of-mass system,  $\bar{\mu}_g$  can be calculated to be  $2/3$  when thermalization is not taken into account. This induced an easier way of computing diffusion coefficients by taking  $\bar{\mu}_g$  to be  $2/3$  for all groups, which is shown in Equation (3).

$$\Sigma_{tr,g}^{as} = \Sigma_{t,g} - \frac{2}{3} \Sigma_{s0,g} \quad (3)$$

in which  $\Sigma_{tr,g}^{as}$  is the transport cross section from the “asymptotic” out-scatter approximation.

Another approximation method makes the hypothesis that neutron current can not exceed the scalar flux and it uses scalar flux spectrum instead of neutron current spectrum for weighting  $P_1$  scattering cross sections [9]. The transport cross section computed by this method as shown in Equation (4) can be called “flux-limited” transport cross section.

$$\Sigma_{tr,g}^{fl} = \Sigma_{t,g} - \sum_{g'=1}^G \frac{\Sigma_{s1,g' \rightarrow g} \phi_{g'}}{\phi_g} \quad (4)$$

in which  $\phi_g$  is scalar flux in group  $g$  and  $\Sigma_{s1,g' \rightarrow g}$  is the  $P_1$  cross section of scattering from group  $g'$  to group  $g$ .

Actually according to  $P_1$  theory, the in-scatter can be treated exactly with given multi-group cross sections. In recent research on transport correction for hydrogen [10], a 70-group library for  $^1\text{H}$  bound in water molecules was generated using NJOY [11], including a 70-group  $P_0$  and  $P_1$  scattering matrix with thermal scattering effect (using  $s(\alpha, \beta)$  tables for light water molecules). Using the group data, the multi-group  $P_1$  equations can be solved numerically to get the results of flux and current spectrum. Then transport cross section can be computed directly following the definition form in Equation (5).

$$\Sigma_{tr,g}^{in} = \Sigma_{t,g} - \sum_{g'=1}^G \frac{\Sigma_{s1,g' \rightarrow g} \vec{J}_{g'}}{\vec{J}_g} \quad (5)$$

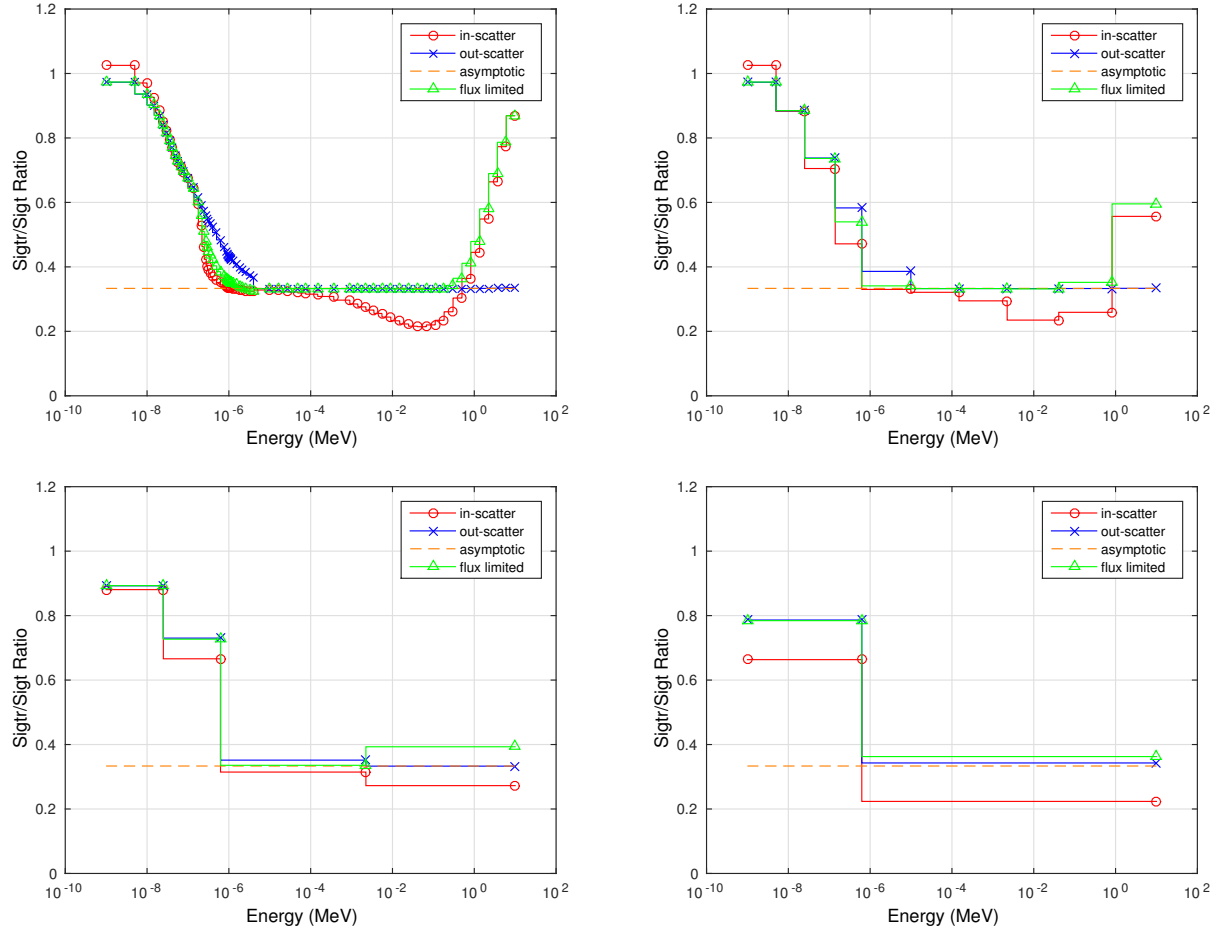
in which  $\Sigma_{tr,g}^{in}$  is the transport cross section with in-scatter calculated directly and the result of this method will be used as reference for comparison among other approximation methods.

### 1.3. A Comparison among Different Approximation Methods

To get a sense of the performance of different approximation methods mentioned in the previous Section 1.2, a simple model with pure hydrogen uniformly distributed in infinite medium can be used as a test problem. The results of transport cross sections  $\Sigma_{tr,g}$  and diffusion coefficients  $D_g$  computed from the three approximation methods, including the out-scatter (*os*) approximation, the asymptotic (*as*) approximation and the flux-limited (*fl*) approximation are compared with those from the in-scatter (*in*) method. The ratios of  $\Sigma_{tr,g}$  to  $\Sigma_{t,g}$  for different group structures in 2/4/10/70 groups are plotted in Figure 1.

In the 70-group and 10-group cases the flux-limited approximation method can match the in-scatter results well in the high energy and thermal range well, but deviates obviously for the middle energy range from 100 eV to 1 MeV. While the out-scatter approximation fail to match in high energy range and the asymptotic method only work for a small energy range. When it comes to few groups such as 4 groups or 2 groups, all of the approximation methods have a big deviation from the reference results. A more direct comparison of diffusion coefficients in 4 and 2 groups is illustrated in Figure 2. In the cases of 4 groups and 2 groups, all of the diffusion coefficients of group 1 from the approximation methods have a deviation as big as 40% lower than the reference results.





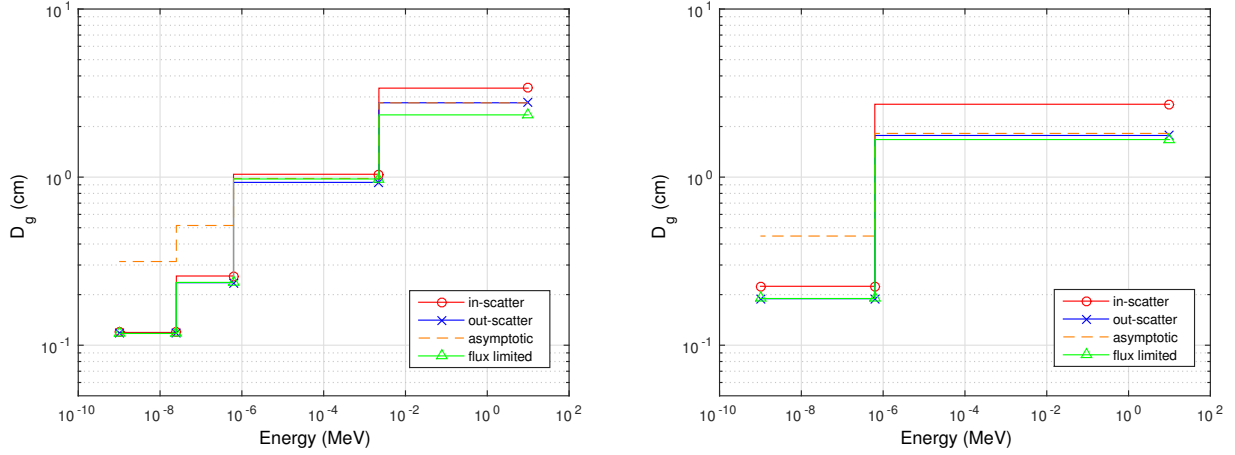
**Figure 1.** Comparison of multi-group ratio of  $\Sigma_{tr,g}$  to  $\Sigma_{t,g}$ . (Upper-left: 70 groups; upper-right: 10 groups; bottom-left: 4 groups; bottom-right: 2 groups.)

## 2. ANALYTICAL DERIVATION AND NUMERICAL RESULTS OF TRANSPORT CORRECTION RATIO

### 2.1. Analytical Derivation of Transport Correction Ratio

The analytical transport correction ratio for an isotope with atomic mass  $A$  can be derived in infinite homogeneous medium with the assumption of only down scatter. In this case the diffusion coefficient can be derived from the second  $P_1$  equation, as shown in Equation (6) [12].

$$D(E) = \frac{1}{3\Sigma_t(E)} \left[ 1 + \frac{3}{\phi(E)} \int_E^{\frac{E}{\alpha}} \Sigma_{s1}(E' \rightarrow E) D(E') \phi(E') dE' \right] \quad (6)$$



**Figure 2.** Comparison of multi-group diffusion coefficients  $D_g$ . (Left: 4 groups; right: 2 groups.)

in which  $\alpha = (A - 1)^2 / (A + 1)^2$ . In the above equation, the differential  $P_1$  scattering cross sections can be expressed as

$$\Sigma_{s1}(E' \rightarrow E) = \bar{\mu} \frac{\Sigma_t(E')}{(1 - \alpha)E'} = \frac{2}{3A} \frac{\Sigma_t(E')}{(1 - \alpha)E'}. \quad (7)$$

Through the relationship between diffusion coefficient and transport cross section  $D = 1/(3\Sigma_{tr})$ , Equation (6) can be re-written as

$$\frac{1}{3\Sigma_{tr}(E)} = \frac{1}{3\Sigma_t(E)} \left[ 1 + \frac{3}{\phi(E)} \int_E^{\frac{E}{\alpha}} \frac{2}{3A} \frac{\Sigma_t(E')}{(1 - \alpha)E'} \frac{1}{3\Sigma_{tr}(E')} \phi(E') dE' \right] \quad (8)$$

The transport correction ratio is defined as  $f(E) = \frac{\Sigma_{tr}(E)}{\Sigma_t(E)}$ . The ratio can be derived as Equation (9) by rearranging Equation (8).

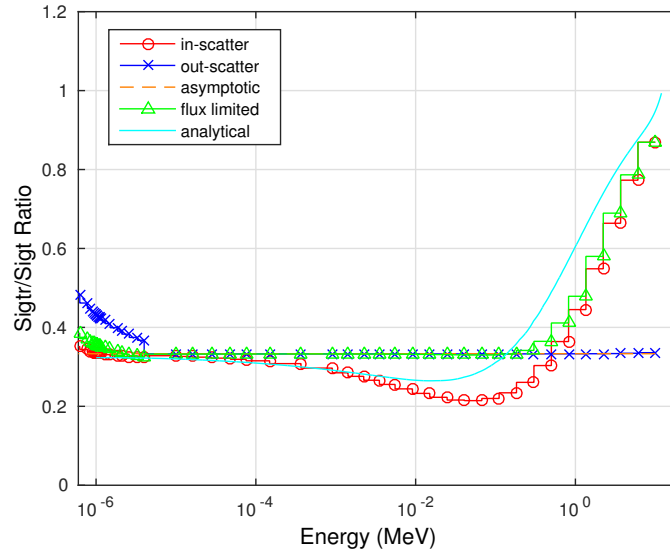
$$f(E) = \left[ 1 + \frac{2}{3A(1 - \alpha)\phi(E)} \int_E^{\frac{E}{\alpha}} \frac{\phi(E')}{f(E')E'} dE' \right]^{-1} \quad (9)$$

According to slowing down theory, flux density at energy  $E$  can be approximated as

$$\phi(E) = \int_E^{E_{max}} \frac{\chi(E')}{\xi \Sigma_s(E')E'} dE' \quad (10)$$

in which  $\chi(E')$  is the source density at energy  $E'$  from Watt fission spectrum, and  $\xi = 1 + \frac{\alpha}{(1-\alpha)\ln(\alpha)}$ .

It should be noted that in the expression in Equation (9)  $f(E)$  is dependent on the integration of  $f(E')$ , so before solving for  $f(E)$  numerically, the value of  $f(E_{max})$  should be set to unity based on the physical meaning of transport correction. The analytical result of transport correction ratio is computed by solving Equation (9) numerically and compared with the methods in Subsection 1.2 in Figure 3. The analytical result is not expected to match the in-scatter method perfectly for the approximated flux density in Equation (10), but the two results show consistent trend over energy, especially for the decreasing part before reaching the asymptotic ratio of 1/3.



**Figure 3.** Comparison of multi-group transport correction ratio by approximation methods *V.S.* analytical result in down scatter energy range.

### 3. CUMULATIVE MIGRATION METHOD FOR DIFFUSION COEFFICIENTS

#### 3.1. Theory of Migration Area

In diffusion theory [2], migration area is defined as in Equation (11) and can be proved to be equal to one-sixth of the average square of the flight distance from the position where a fast neutron is born to the position where it is absorbed as a thermal neutron.

$$M^2 = L^2 + \tau_{th} \quad (11)$$

in which  $M^2$  is migration area,  $L^2$  is diffusion area and  $\tau_{th}$  is neutron age of fast neutrons to thermal. Diffusion area is defined as

$$L^2 = \frac{D}{\Sigma_a} \quad (12)$$

where  $D$  is diffusion coefficient and  $\Sigma_a$  is absorption cross section.

Essentially  $\tau_{th}$  is equal to one-sixth of the average square of the flight distance from the position where a fast neutron is born to the position where it is moderated to a thermal neutron, and  $L^2$  is equal to one-sixth of the average square of the flight distance from the position where a neutron becomes thermal to the position where it is finally absorbed.

### 3.2. Multi-group Form and Cumulative Group Quantities

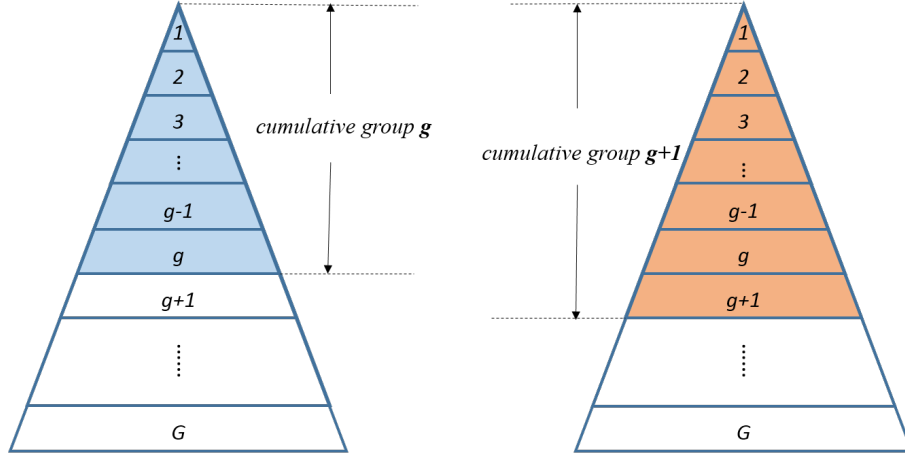
The definition of diffusion area in Equation (12) is considering a neutron from the position where it becomes thermal, in which case absorption is the only cause for the neutron's disappearance when regarding the thermal energy range as a single group. Similarly, if analyzing from the position where the fast neutron is originally born to the position where it is removed from a given energy range, in other words, the "partial" migration area when the neutron's energy is higher than a certain value  $E_0$ , the equation will also be justified as

$$M^2(E > E_0) = \frac{D(E > E_0)}{\Sigma_r(E > E_0)} \quad (13)$$

in which  $M^2(E > E_0)$  is the cumulative migration area before the neutron's energy becomes lower than  $E_0$ ,  $D(E > E_0)$  is the diffusion coefficient for the energy range of  $[E_0, E_{max}]$ , and  $\Sigma_r(E > E_0)$  is the removal cross section for the energy range of  $[E_0, E_{max}]$ , which will include not only absorption, but also net down scatter to a energy lower than  $E_0$ .

Actually from the perspective of energy group, the energy range of  $[E_0, E_{max}]$  can be seen as a "broad" group whose the top boundary always starts from  $E_{max}$ . In a multi-group structure, if  $E_0$  is the bottom boundary of group  $g$ , then the "broad" group can be seen as a "cumulative group" from group 1 to group  $g$ . The concept of cumulative group is illustrated in Figure 4, in which the group structure is shown in a pyramid frame with cumulative group  $g$  and cumulative group  $(g + 1)$  shown in shadowed parts. Based on this concept, Equation (13) can be re-expressed in the form of cumulative group as

$$(M_g^c)^2 = \frac{D_g^c}{\Sigma_{r,g}^c} \quad (14)$$



**Figure 4.** Illustration of the concept of “cumulative group”. (Left: cumulative group  $g$ , right: cumulative group  $(g + 1)$ .)

where the superscript  $c$  indicates that all the quantities in this equation are for the cumulative group  $g$ , that is to say, the “broad” group from group 1 to group  $g$ .

Equation (14) provided a scheme for computing cumulative multi-group diffusion coefficients through the theory of cumulative migration area, using the one-sixth relationship with the average square of neutron’s flight distance, as shown in Equation (15).

$$(M_g^c)^2 = \frac{1}{6} \overline{(r_g)^2} \quad (15)$$

where  $r_g$  is the distance from the position where the neutron is originally born to the position where it is removed out of the cumulative group  $g$ , which is a quantity can be tallied directly in MC code. Then group-wise diffusion coefficients  $D_g$  can be backed out by “unfolding” cumulative multi-group diffusion coefficients  $D_g^c$  using flux-weighting as shown in Equation (16).

$$D_g^c = \frac{\sum_{g'=1}^g D_{g'} \phi_{g'}}{\sum_{g'=1}^g \phi_{g'}} \quad (16)$$

## 4. IMPLEMENTATION OF CUMULATIVE MIGRATION AREA TALLY IN OPENMC

In recent years, with the advance in computer technology, it has been investigated widely to utilize MC codes for generating homogenized lattice few-group cross section data for deterministic full core simulation, such as MCNP [13], Serpent [14], etc. In Serpent, firstly a continuous-energy Monte Carlo simulation is run to produce micro-group cross sections for  $B_1$  equations, then the  $B_1$  equations need to be solved for critical spectrum, which will be used to re-homogenize the cross sections into leakage-corrected few-group constants. The generation of micro-group cross sections needs more tally and computation work, which will make this method computationally inefficient.

In the homogenized few-group cross section data, the method for generating reliable diffusion coefficients poses a major challenge for the implementation in MC codes, since a rigorous theory for computing diffusion coefficients and transport cross sections is yet to be developed. Based on the theory in Section 3, the implementation scheme in OpenMC [15] is introduced in this section.

### 4.1. Tally Scheme for Cumulative Migration Area

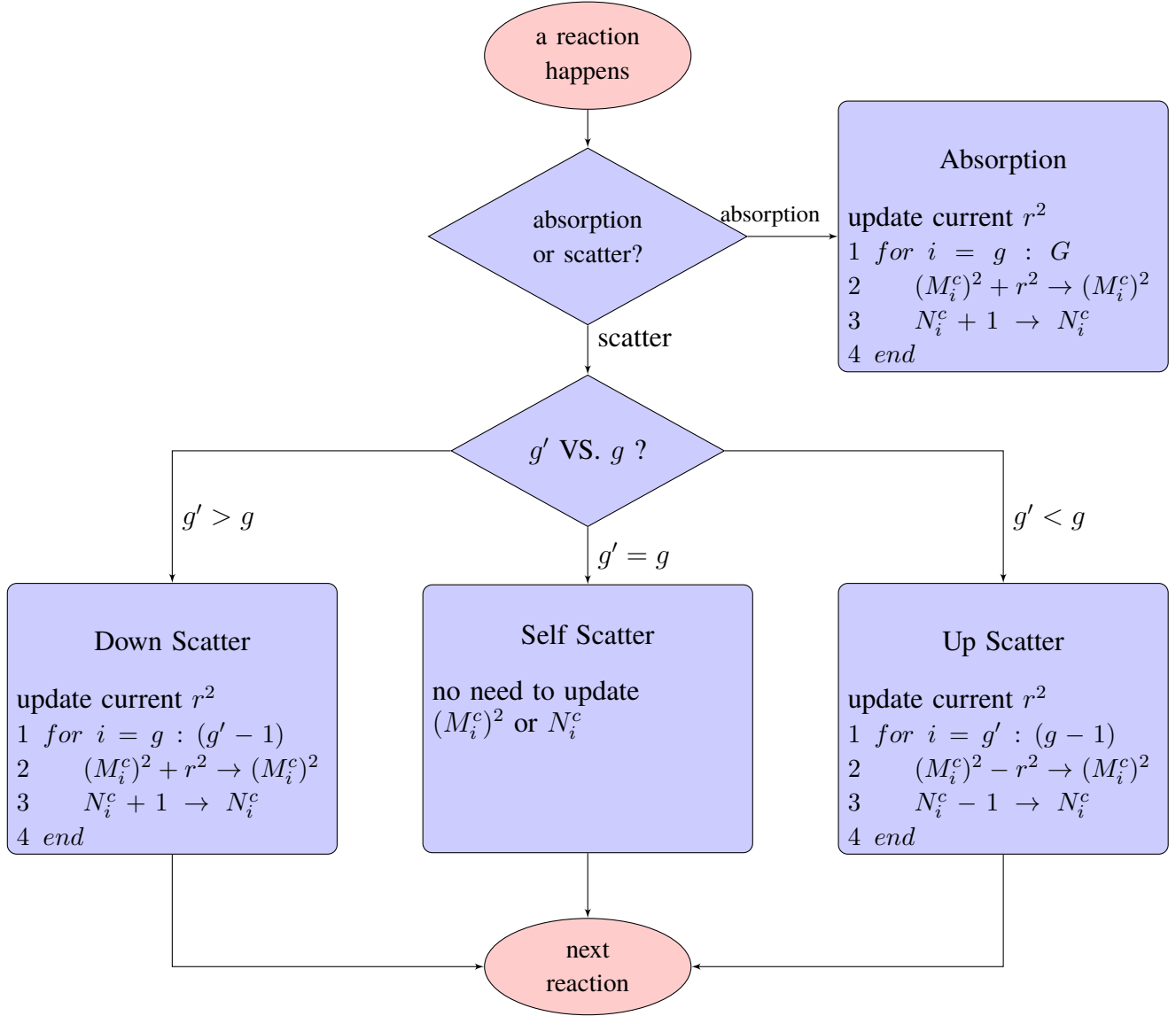
In the implementation of the Cumulative Migration Method (CMM) for computing multi-group diffusion coefficients and transport cross sections in OpenMC, the crucial tally is cumulative migration area. The main events that change neutrons' energy and flight path is scatter, while the loss of neutrons is mainly resulted from absorption. Both down and up scatter should be taken into account in the cumulative migration area tally.

The tally scheme for cumulative migration area  $(M_g^c)^2$  and cumulative number of neutrons tallied  $N_g^c$  in one reaction (scatter or absorption) is shown in Figure 5. In the figure of flowchart,  $g$  is the group index of the neutron before a reaction,  $g'$  is the group index of the neutron after a scatter reaction,  $G$  is the total number of groups, and  $r^2$  is the square of the distance from the neutron's birth position to the position of the reaction.

This tally scheme can be used for any group structure and scores can be tallied directly into the desired few group structure, which avoids additional micro-group tally for  $B_1$  spectrum calculation as in Serpent. In the implementation only one more score of cumulative migration area  $(M_g^c)^2$  is added to MC code, which enables it to be both flexible and computationally efficient.

### 4.2. Phantom Tracking in Finite Medium

The cumulative migration area is simple to tally in infinite medium, however, in practice most simulations with MC code are in finite geometry. Even though there are many simulations in infinite medium, the actual practice is using finite geometry with reflective boundary conditions. But according to the



**Figure 5.** Flowchart of tally scheme for cumulative migration area  $(M_g^c)^2$  and cumulative number of neutrons  $N_g^c$  for reactions of absorption and scatter.

theory of cumulative migration area, when a neutron undergoes a reflection on any reflective boundary, the actual position used in tally and computation should be the position as if there were no boundaries, which can be called the “phantom” position. Thus for problems with finite geometry and reflective boundary conditions, phantom tracking for neutrons should be established for the tally of cumulative migration area.

The phantom tracking for rectangular cuboid geometries has been implemented in OpenMC. The essential idea of the implementation is to utilize three *reflection flags* for each Cartesian axis, then the flight direction as well as position of a neutron’s phantom at any reaction can be inferred by these three flags. The cumulative migration area is tallied based on the positions of the phantom, instead of the real neutron.

## 5. SIMULATION RESULTS AND ANALYSIS

Several simulation problems has been completed using OpenMC with the added feature of computing diffusion coefficients and transport cross sections using CMM. In this section, the results of a simple problem of pure hydrogen and a real assembly from BEAVRS benchmark are presented.

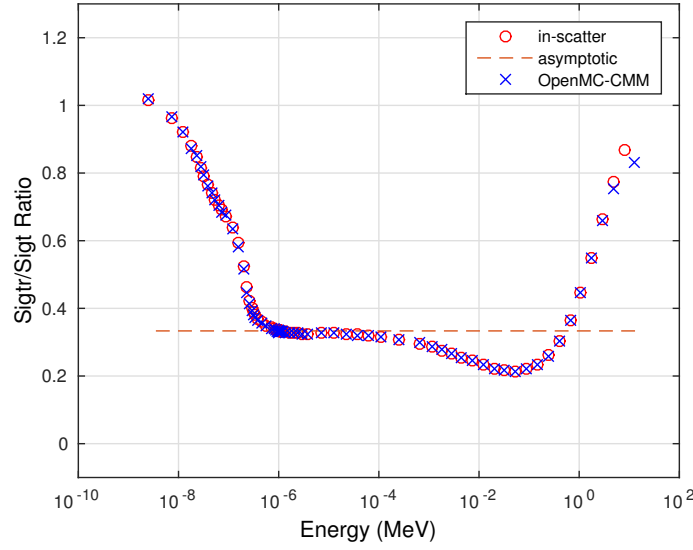
### 5.1. Infinite Medium of Hydrogen

A simple problem of pure hydrogen in infinite medium is modeled and simulated for a comparison of the diffusion coefficients and transport cross sections computed using CMM and those of the in-scatter method in Section 1.3.

The model is built as a rectangular cuboid geometry with all reflective boundary conditions. According to the analysis of phantom tracking in Section 4.2, the cuboid can be in any size and will give the same results. In addition, the transport correction ratio should be independent of density of the problem, which has also been tested.

The simulation result is compared with in-scatter method in Figure 6. The transport correction ratio computed from OpenMC with CMM matched perfectly with in-scatter method, indicating that the transport cross section as well as diffusion coefficients from these two methods also agree very well. That is to say, the CMM actually generates the same multi-group diffusion data as the data calculated directly by solving  $P_1$  equations.





**Figure 6.** Comparison of 70-group transport correction ratio generated from OpenMC with Cumulative Migration Method (OpenMC-CMM) *V.S.* in-scatter method.

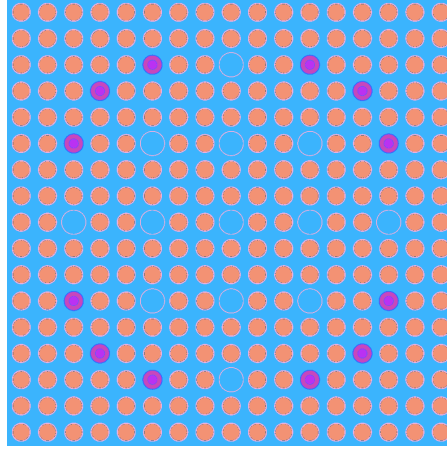
## 5.2. 2-group Diffusion Data for a BEAVRS Assembly

### 5.2.1. Configuration of the Assembly

To further evaluate the effect of the Cumulative Migration Method, it also has been used to generate 2-group diffusion data for a real assembly from BEAVRS benchmark [16]. The assembly simulated is the 2.4 wt-%  $^{235}\text{U}$  with 12 burnable absorber pins, which recently has also been used by Serpent for computing 2-group constants [17]. The configuration of this assembly can be seen in Figure 7.

### 5.2.2. 2-group Diffusion Data

The 2-group constants generated by OpenMC using CMM is compared with those of generated by CASMO and Serpent as shown in Table I. The group constants compared in the table include diffusion coefficients ( $D_1$ ,  $D_2$ ), absorption cross sections ( $\Sigma_{a1}$ ,  $\Sigma_{a2}$ ), net-down scatter cross section ( $\Sigma_{s,1\rightarrow2}$ ), fission neutron production cross sections ( $\nu\Sigma_{f1}$ ,  $\nu\Sigma_{f2}$ ), surface discontinuity factors ( $DF_1^s$ ,  $DF_2^s$ ) and  $k_{inf}$  calculated using the 2-group data. The 2-group data generated by OpenMC with CMM in Table I is closer to that of CASMO, especially for the diffusion coefficient of fast group.



**Figure 7.** Configuration of the BEAVRS assembly of 2.4 wt-%  $^{235}\text{U}$  with 12 burnable absorber pins.

**Table I.** 2-group constants for the BEAVRS assembly of 2.4 wt-%  $^{235}\text{U}$  with 12 burnable absorber pins generated by CASMO, Serpent and OpenMC.

Parameter	Serpent	CASMO	OpenMC
$D_1$	1.397E+00	1.430E+00	1.426E+0
$D_2$	3.933E-01	3.775E-01	3.992E-1
$\Sigma_{a1}$	8.906E-03	8.908E-03	8.919E-3
$\Sigma_{a2}$	8.380E-02	8.297E-02	8.408E-2
$\nu\Sigma_{f1}$	5.343E-03	5.371E-03	5.380E-3
$\nu\Sigma_{f2}$	1.019E-01	1.009E-01	1.019E-1
$\Sigma_{s,1\rightarrow2}$	1.835E-02	1.804E-02	1.838E-2
$DF_1^s$	9.947E-01	9.931E-01	9.929E-1
$DF_2^s$	1.060E+00	1.061E+00	1.057E+0
$k_{inf}$	1.01469	1.01323	1.01281

## 6. CONCLUSION

The Cumulative Migration Method (CMM) for computing homogenized lattice multi-group diffusion data using Monte Carlo code is proposed in this paper. To overcome the shortcomings in various approximation methods for computing neutron transport cross sections and diffusion coefficients, which have been in use for more than 40 years, the theory of cumulative migration area and its relationship with transport cross sections and diffusion coefficients is developed for utilization to Monte Carlo tally schemes.

This method has been implemented in OpenMC and several test problems have been simulated for generating homogenized multi-group diffusion data. The results demonstrate that it is the  $P_1$  diffusion coefficients, and not the  $B_1$  diffusion coefficients, that are totally consistent with diffusion coefficients computed with CMM. This helps tremendously to settle the longstanding argument over the appropriateness of the  $P_1$  and  $B_1$  models of diffusion coefficients.

Another distinct advantage of CMM is that no where does one have to make provisions for lattices that have voids (many deterministic codes opt to spatially homogenize transport cross sections to avoid singularities caused by voids and the energy collapse one over sigma transport to obtain an accurate energy collapse). CMM handles void very naturally and no decisions are required regarding the proper energy collapse.

In addition, one of the most significant accomplishment of CMM is that few group diffusion coefficients and transport cross sections can be computed directly in the desired few group structure – without the necessity of tallying fine group cross sections that are currently employed in Monte Carlo  $B_1$  or  $P_1$  spectrum calculations. These few group diffusion coefficients are exactly the same as those that are tallied in fine energy group tallies and then collapsed to few groups. Consequently, CMM provides tremendous simplifications of the tallies needed in Monte Carlo to compute accurate homogenized lattice diffusion coefficients and transport cross sections.

## REFERENCES

- [1] George I. Bell and Samuel Glasstone. *Nuclear Reactor Theory*. Robert E. Krieger Publishing Co., Huntington, NY, USA (1979).
- [2] John R. Lamarsh. *Introduction to Nuclear Reactor Theory*. Addison-Wesley, Reading, MA, USA (1966).
- [3] T. Newton and J. Hutton. “The next generation wims lattice code: Wims9.” *Proceedings of New Frontiers of Nuclear Technology, PHYSOR*, (pp. 7–10) (2002).
- [4] H. Huria *et al.* “Theory and methodology of westinghouse phoenix-h/anc-h for hexagonal pwr cores.” *Transactions of the American Nuclear Society*, **71**(CONF-941102–) (1994).

- [5] G. Marleau, R. Roy, and A. Hébert. “Dragon: a collision probability transport code for cell and supercell calculations.” *Report IGE-157, Institut de génie nucléaire, École Polytechnique de Montréal, Montréal, Québec* (1994).
- [6] E. A. Villarino *et al.* “Helios: angularly dependent collision probabilities.” *Nuclear Science and Engineering*, **112**(1): pp. 16–31 (1992).
- [7] B. R. Herman *et al.* *Improved diffusion coefficients generated from Monte Carlo codes. Technical report*, American Nuclear Society, 555 North Kensington Avenue, La Grange Park, IL 60526 (United States) (2013).
- [8] R. J. Stamm’ler and M. J. Abbate. *Methods of steady-state reactor physics in nuclear design*, volume 111. Academic Press London (1983).
- [9] G. Pomraning. “Flux-limited diffusion theory with anisotropic scattering.” *Nuclear Science and Engineering*, **86**(4): pp. 335–343 (1984).
- [10] B. R. Herman. *Monte Carlo and Thermal Hydraulic Coupling using Low-Order Nonlinear Diffusion Acceleration*. Ph.D. thesis, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA (2014).
- [11] R. E. Macfarlane, D. Muir, and D. George. “Njoy99. 0 code system for producing pointwise and multigroup neutron and photon cross sections from endf/b data.” *Los Alamos National Laboratory, PSR-480*, **7** (2000).
- [12] A. Hebert. *Applied reactor physics*. Presses inter Polytechnique (2009).
- [13] J. M. Pounders. “Stochastically generated multigroup diffusion coefficients.” (2006).
- [14] E. Fridman and J. Leppänen. “On the use of the serpent monte carlo code for few-group cross section generation.” *Annals of Nuclear Energy*, **38**(6): pp. 1399–1405 (2011).
- [15] P. K. Romano and B. Forget. “The openmc monte carlo particle transport code.” *Annals of Nuclear Energy*, **51**: pp. 274–281 (2013).
- [16] N. Horelik *et al.* “Benchmark for evaluation and validation of reactor simulations (beavrs), v1. 0.1.” In: *Proc. Int. Conf. Mathematics and Computational Methods Applied to Nuc. Sci. & Eng* (2013).
- [17] J. Leppänen and R. Mattila. “On the practical feasibility of continuous-energy monte carlo in spatial homogenization.” *PHYSOR* (2014).

# COMPARISONS OF THE FINITE-ELEMENT-WITH-DISCONTIGUOUS-SUPPORT METHOD TO CONTINUOUS-ENERGY MONTE CARLO FOR PIN-CELL PROBLEMS (SUMMARY)

A. T. Till, M. Hanuš, J. Lou, J. E. Morel, and M. L. Adams

Department of Nuclear Engineering

Texas A&M University, College Station, TX, USA

[attom@tamu.edu](mailto:attom@tamu.edu), [mhanus@tamu.edu](mailto:mhanus@tamu.edu), [loujijie1991@tamu.edu](mailto:loujijie1991@tamu.edu),

[morel@tamu.edu](mailto:morel@tamu.edu), [mladams@tamu.edu](mailto:mladams@tamu.edu)

## ABSTRACT

The standard multigroup (MG) method for energy discretization of the transport equation can be sensitive to approximations in the weighting spectrum chosen for cross-section averaging. As a result, MG often inaccurately treats important phenomena such as self-shielding variations across a material. From a finite-element viewpoint, MG uses a single fixed basis function (the pre-selected spectrum) within each group, with no mechanism to adapt to local solution behavior. In this work, we introduce the Finite-Element-with-Discontiguous-Support (FEDS) method, whose only approximation with respect to energy is that the angular flux is a linear combination of unknowns multiplied by basis functions. A basis function is non-zero only in the discontiguous set of energy intervals associated with its energy element. Discontiguous energy elements are generalizations of bands and are determined by minimizing a norm of the difference between snapshot spectra and their averages over the energy elements. We begin by presenting the theory of the FEDS method. We then compare to continuous-energy Monte Carlo for one-dimensional slab and two-dimensional pin-cell problem. We find FEDS to be accurate and efficient at producing quantities of interest such as reaction rates and eigenvalues. Results show that FEDS converges at a rate that is approximately first-order in the number of energy elements and that FEDS is less sensitive to weighting spectrum than standard MG.

**Key Words:** Multigroup (MG), Subgroup (SG), Multiband (MB), Energy Discretization, Finite Element (FE), Monte Carlo

## 1. Introduction

The popular “deterministic method” of simulating radiation interaction with matter discretizes the six-dimensional phase space (space, angle, and energy) upon which the radiation field depends and solves the discretized equations for the expected radiation distribution. Discretizing in energy has historically been a difficult problem due to numerous, strong, thin resonances in the interaction probabilities.

The standard multigroup (MG) method for energy discretization of the transport equation can be sensitive to approximations in the weighting spectrum chosen for cross-section averaging, especially

in the common case where groups span many resonances. As a result, MG often inaccurately treats important phenomena such as self-shielding variations across a fuel pin. From a finite-element viewpoint, MG uses a single fixed basis function (the pre-selected spectrum) within each group, with no mechanism to adapt to local solution behavior.

This work tests a new method to increase the accuracy of these computations by using a novel energy discretization scheme to treat resolved resonances that we call the Finite-Element-with-Discontiguous-Support (FEDS) method. In previous work, we describe the full theory and derivation of this method [1], and test FEDS on cylindricized pin-cell and C5 problems where a fraction of the resonance range was resolved by a MG reference calculation [2][3]. While this reference solution allowed us to probe energy discretization error directly, it was not realistic.

In this work, we make comparisons to continuous-energy MCNP. We test a series of pin cells with specularly reflecting boundaries in both slab (1D) and real (2D) geometries. We investigate a host of quantities of interest (QOI), including quantities that are more localized in space, energy, nuclide, and reaction type. Taken together, these allow us to claim FEDS can provide accurate QOI for realistic problems using a modest number of unknowns without relying on error cancellation.

Future sections are divided as follows. Section 2 describes the FEDS method, including its finite element (FE) definition, transport equation, and cross section definitions. Section 3 describes the method we use to determine the discontiguous energy mesh upon which the FE space is built, our hierarchical mesh structure, and contiguous meshes for a MG comparison. Section 4.1 gives a description of and results for the one-dimensional slab pin-cell problems. Section 4.2 gives a description of and results for the 2D pin-cell problems.

## 2. The FEDS Method

The FEDS method is a Petrov-Galerkin FE method that is an extension of the previously introduced Petrov-Galerkin Finite-Element Multigroup (PG-FEMG) method [4], itself a generalization of the MG method. In FEDS, the weight functions,  $w_e(E)$ , are unity for energies within an energy element and zero otherwise. The basis functions have the same support as the weight functions but also have a normalized spectral shape. The “DS” in FEDS means the weight and basis functions are allowed to have discontiguous support. We require that elements do not overlap, which gives us weight and basis functions that are orthonormal. Our FEDS angular flux solution is a basis function expansion:

$$\psi(\mathbf{r}, E, \Omega) = \sum_e b_e(\mathbf{r}, E) \Psi_e(\mathbf{r}, \Omega), \quad (1)$$

where the basis functions,  $b_e(\mathbf{r}, E)$ , often have a spatial dependence that is piecewise-constant. We solve an infinite-medium-equivalent slowing-down equation in each material to determine the spectral shape of the basis functions.

Equation (1) is the only approximation for FEDS. MG may be cast in the same framework as FEDS, but with contiguous groups instead of discontinuous elements. For (MG) group sizes that are on the order of or greater than resonance widths, the approximation in Eq. (1) becomes significant because the solution will vary appreciably in one group for different particle locations and directions of travel due to self-shielding effects. In the next section, we will define our discontinuous energy elements to minimize the variance within each element of several spectra that are representative of the solution. We claim this choice of mesh makes the approximation in Eq. (1) reasonable as the FEDS flux is given sufficient freedom to adapt to local solution behavior.

To derive the FEDS transport equation, we take the continuous-energy transport equation, expand the angular flux and its moments into their basis function expansions using Eq. (1), multiply by the weight functions, and integrate over all energies. After some algebra, we get the FEDS k-eigenvalue transport equation:

$$[\mathbf{\Omega} \cdot \nabla + \Sigma_{t,e,i}] \Psi_e(\mathbf{r}, \mathbf{\Omega}) = \sum_{l=0}^L \frac{2l+1}{4\pi} \sum_{e'} \Sigma_{s,l,e' \rightarrow e,i} \sum_{m=-l}^l Y_{l,m}(\mathbf{\Omega}) \phi_{e'}^{l,m}(\mathbf{r}) + \frac{\chi_{e,i}}{4\pi k_{\text{eff}}} \sum_{e'} \nu \Sigma_{f,e',i} \phi_{e'}, \quad \mathbf{r} \in V_i, \quad (2a)$$

$$\phi_e^{l,m}(\mathbf{r}) = \int_{4\pi} d\mathbf{\Omega} Y_{l,m}(\mathbf{\Omega}) \Psi_e(\mathbf{r}, \mathbf{\Omega}), \quad \mathbf{r} \in V_i, \quad (2b)$$

$$\Psi_e(\mathbf{r}, \mathbf{\Omega}) = \Psi_{\text{inc},e}(\mathbf{r}, \mathbf{\Omega}), \quad \mathbf{r} \in \partial V, \quad \mathbf{\Omega} \cdot \mathbf{n}(\mathbf{r}) < 0, \quad (2c)$$

where  $i$  designates a material region of spatial domain  $V_i$ ,  $e$  is the energy element,  $l$  and  $m$  are the degree and order of the spherical harmonic moments, respectively, and other notation is standard. Orthonormality of the energy elements makes interface conditions trivial:  $\Psi_e(\mathbf{r}^+, \mathbf{\Omega}) = \Psi_e(\mathbf{r}^-, \mathbf{\Omega})$  for  $\mathbf{r}^- \in V_i$ ,  $\mathbf{r}^+ \in V_j$ , and  $\mathbf{r} \in V_i \cap V_j$ ,  $\forall e = 1, \dots, N_e$ .

Equation (2) is distinguishable from the MG transport equation only in the definition of the cross sections, which are basis-function weighted:

$$\Sigma_{t,e,i} \equiv \int_0^\infty dE b_e(\mathbf{r}, E) \Sigma_{t,i}(E), \quad (3a)$$

$$\nu \Sigma_{f,e,i} \equiv \int_0^\infty dE b_e(\mathbf{r}, E) \nu \Sigma_{f,i}(E), \quad (3b)$$

$$\Sigma_{s,l,e' \rightarrow e,i} \equiv \int_0^\infty dE w_e(E) \int_0^\infty dE' b_{e'}(\mathbf{r}, E') \Sigma_{s,l,i}(E' \rightarrow E), \quad (3c)$$

and the unknowns and source terms, which are integrals over the discontinuous elements:

$$\chi_{e,i} \equiv \int_0^\infty dE w_e(E) \chi_i(E), \quad (3d)$$

$$\Psi_e(\mathbf{r}, \mathbf{\Omega}) = \int_0^\infty dE w_e(E) \psi(\mathbf{r}, E, \mathbf{\Omega}). \quad (3e)$$

Equation (3) follows without approximation by consistent application of Eq. (1) and orthonormality. Further details on the description and derivation of the FEDS method, transport equation, and fluxes may be found in [1, 5].

### 3. The Energy Mesh

The energy mesh describes the support of the weight and basis functions, with a function being nonzero for energies within an energy element and zero for energies not within an energy element. First, we assume we are given a library of spectra and solve a minimization problem using these spectra to determine the energy mesh. We then describe one inexpensive way to generate the spectra.

Given spectra, we formulate and solve a problem that minimizes the variance of the spectra within each energy element. This minimizes the projection error of the spectra from their original resolved grid to a coarse grid, which is used as the final energy mesh. A FE method built on this energy mesh can adapt to the true spectral behavior of the solution if the spectra well represent the solution. For more details, see [1, 2, 5].

We solve infinite-medium, fixed-source, slowing-down problems to generate the spectra. Our spectra resolve resonance absorption dips and resonance scattering peaks for all nuclides in each material considered. This method generates spectra that are representative snapshots of the solution, are highly resolved in energy, and approximate the behavior of the solution for several materials within the problems of interest.

Our FEDS method is a two-step method. In the first step, we use infinite-medium spectra for several materials we expect to see in the problem of interest as inputs to a minimization problem whose solution is an energy mesh that uses discontinuous energy elements. In the second step, we compute problem-specific basis functions with which we weight the continuous-energy cross sections to produce FEDS cross sections. These cross sections are flux-weighted averages over the discontinuous energy elements found in the first step. Once we have these cross sections, we may use them in any existing transport code without further modification to either cross section or code. The transport solver returns fluxes integrated over the discontinuous energy elements.

We test how our QOI behave when varying the materials we include in our library of spectra. We define two mesh families,  $R_{i,e}$  and  $R_{i,d}$ , with  $i = 1, \dots, 5$  indicating the number of energy unknowns in the resolved resonance region (RRR).  $R_{i,e}$  uses two spectra, both of  $\text{UO}_2$ , one with U-238 only and another with 4 % U-235.  $R_{i,d}$  uses one spectrum,  $\text{UO}_2$  with U-238 only.  $R_{i,d}$  will resolve only U-238 resonances in its energy mesh, while  $R_{i,e}$  will resolve both U-235 and U-238 resonances.

We create our energy mesh hierarchically. We first divide the energy variable into 12 coarse groups, given in Table I. Within each resolved resonance coarse group, groups III through X, we solve a minimization problem to determine energy elements that live completely within that coarse group. This



allows us to compute reaction rates over a single coarse group by summing the product of the scalar fluxes and cross sections for the elements in that coarse group. The use of coarse groups as the primary means of bounding element extent in energy differs from previous results that primarily used energy penalties for this purpose [2]. We still use energy penalties, but they only have an effect in coarse groups that are wide in lethargy and small in solution variance, viz. high-energy coarse groups in the resolved resonance range. Even within these coarse groups, most of the energy elements are spent resolving the spectral / solution space and not in energy space. Outside of the RRR, we use the SHEM-361 group boundaries [6]. We chose our coarse groups to be hierarchical to the SHEM-361 group boundaries.

We compared our FEDS to MG with the same number of energy unknowns in the RRR. At the highest resolution, we used the SHEM-361 group-boundaries for the MG cases. For all cases, we used the same condensing spectra for MG and FEDS. Because we did not use the subgroup (SG) method when doing group condensation and because the SHEM-361 group boundaries do not resolve higher-energy resonances, we expect error in MG reaction rates in the higher-energy coarse groups to be large. This is not an indication of the accuracy attainable using the SHEM boundaries, but rather of the necessity of using the sub-group treatment to attain accurate reaction rates for typical MG calculations. As we will show, FEDS attains lower errors in QOI than MG even for poor condensing schemes and spectra. Previous work [2] has shown FEDS retains its advantage over MG when both use higher-fidelity condensing functions, including the use of implicit condensing functions, such as cross section tables parametrized in background cross section. For lower energy resolutions, we manually defined SHEM-like group structures that had approximately equal lethargy spacing at high energies and resolved two or three low-lying resonances at lower energies for U-238 and U-235. Further details may be found in [5].

Table I: Coarse groups used for the edits.

Coarse Group	Upper Energy (eV)	Uses FEDS?
I	$2.00000 \times 10^7$	No
II	$1.40000 \times 10^5$	No
III	$2.26994 \times 10^4$	Yes
IV	$9.11881 \times 10^3$	Yes
V	$2.08410 \times 10^3$	Yes
VI	$5.39204 \times 10^2$	Yes
VII	$1.54176 \times 10^2$	Yes
VIII	$5.17847 \times 10^1$	Yes
IX	$2.78852 \times 10^1$	Yes
X	$9.50002 \times 10^0$	Yes
XI	$4.21983 \times 10^0$	No
XII	$6.24999 \times 10^{-1}$	No

## 4. Results

We investigated several pin-cell problems (Table II) and compared reaction rates between MG or FEDS computed using a deterministic transport code (PDT [7]) and the continuous-energy Monte Carlo code MCNP [8] using consistent cross sections processed in NJOY [9]. All problems were at room temperature and used ENDF/B-VII.1 cross sections.

Four problems were investigated. The first problem (A) used only depleted uranium (U-238) and a spatially-flat, fixed fission source. The fission source was the thermal-neutron-induced Watt fission spectrum characterized by  $a = 0.988$  MeV and  $b = 2.249$  MeV<sup>-1</sup>. PDT was given an analytically group-integrated version of this quantity. In order not to double count the fission source, the nonu card was used in MCNP and the  $\nu$ ,  $\nu\sigma_f$  and  $\chi$  cross sections were manually removed from the PDT cross sections. The fixed fission source is a decent approximation to the true fission source within a pin and provided a level of additional consistency between MCNP and PDT. The second problem (B) added a fissile material (4% by atom U-235) but kept the fixed fission source. This added the challenge of resonance interference effects. The third problem (C) was the same as B, except it was formulated as a  $k$ -eigenvalue problem and so had a spatially dependent source. Unlike MCNP, the PDT cross sections used a fixed fission spectrum,  $\chi$ , for all incident neutron energies for problem C.

Reaction-rate-based quantities of interest (QOI) were compared between MCNP and PDT on a coarse energy grid. The energy grid was chosen based on conversations with Dr. Kord Smith at MIT and is given in Table I. The highest-energy group is for fast neutrons. The second-highest-energy group is for the unresolved resonance range (URR) of U-238. The next eight groups are in the resolved resonance region (RRR) above 4 eV, where thermal effects start becoming important. The penultimate group contains low-lying resonances, and the final, lowest-energy group is for thermal neutrons.

Table II: Pin-cell problem definitions.

Problem	Description
A	1D slab or 2D pincell using U-238 and fixed fission source
B	1D slab or 2D pincell using U-238 and U-235 with fixed fission source
C	1D slab or 2D pincell using U-238 and U-235 in $k$ -eigenvalue formulation

### 4.1. One-dimensional pin cells

We first investigated slab geometry versions of the pin-cell problems. To estimate the required resolutions in space, angle, and scattering moment truncation, we performed a resolution study wherein we increased fidelity until the combined discretization error from the aforementioned parameters contributed less than approximately 5 pcm, as estimated by comparing reaction rates between differing resolutions. Based on the resolution study, we used  $P_3$  scattering,  $S_{32}$  Gauß-Legendre quadrature, and lumped PWLD spatial discretization with 20 (12) cells in the fuel (and its boundary) and 10 (5) cells in the moderator (and its

boundary). These resolutions allowed us to attribute discrepancies of more than  $\sim 5$  pcm between PDT (MG or FEDS) and MCNP (continuous energy) to energy discretization effects alone.

Table III gives error in  $k$ -eigenvalue for problem C for the various energy discretizations studied. The errors in the SHEM MG family do not converge and end at several hundred pcm. SHEM-244 attains a low eigenvalue error through fortuitous error cancellation. Errors in the FEDS methods begin large but converge uniformly with increasing energy resolution in the RRR. Both the  $R_{i,d}$  family and the  $R_{i,e}$  family have approximately the same magnitude (and sign) of errors. This indicates either fortuitous error cancellation for the  $R_{i,d}$  family on U-235-related errors, or that U-238 is the dominant nuclide, and getting it right is necessary and sufficient for getting the flux, and hence reaction rates, correct. We defined efficiency in Table III such that a first-order method in energy would have constant efficiency.

Table IV gives errors in all the QOI for problem C for the  $R_{5,e}$  energy discretization, with 281 energy unknowns in the RRR. Errors for QOI are given on the coarse group mesh. Relative errors in percent and absolute errors in pcm are given. These results show FEDS has correct differential behavior with a minimal number of DOF in the RRR. Most relative errors are under 0.5 % and most absolute errors are under 20 pcm for coarse groups in the RRR. U-238 absorption errors in coarse group VI are still large, but are within 1.3 %.

Table III: Errors in  $k$ -eigenvalue for various energy resolutions for problem C for the 1D pincell.

Name	Unknowns in the RRR	Error (pcm)	Efficiency (1/[error $\times$ DOF])
SHEM-166	30	$-202 \pm 2$	16.5
SHEM-244	108	$-93 \pm 2$	9.9
SHEM-361	225	$-587 \pm 2$	0.8
$R_{1,d}$	30	$762 \pm 2$	4.4
$R_{2,d}$	59	$493 \pm 2$	3.4
$R_{3,d}$	108	$317 \pm 2$	2.9
$R_{4,d}$	225	$116 \pm 2$	3.8
$R_{5,d}$	281	$69 \pm 2$	5.2
$R_{1,e}$	30	$757 \pm 2$	4.4
$R_{2,e}$	59	$506 \pm 2$	3.4
$R_{3,e}$	108	$313 \pm 2$	3.0
$R_{4,e}$	225	$89 \pm 2$	5.0
$R_{5,e}$	281	$63 \pm 2$	5.6

Figure 1 shows the total absorption errors for problem B. The top row of Fig. 1 shows MG with SHEM group boundaries. As resolution is increased, differential errors stay approximately constant around 200 pcm for the higher portion of the RRR where the resonances are not resolved by the group structure. The errors in the lower portion of the RRR decrease with increasing resolution because the group structure begins to resolve the low-lying resonances. The normalization for problem B is on total absorption rate, meaning the sum of this error over all coarse groups is zero. The absorption rate in the lowest coarse

group may be regarded as a cumulative error of the errors in higher-energy coarse groups. This error does not converge for the numbers of groups used.

The middle row of Fig. 1 uses FEDS with energy meshes in the  $R_{i,d}$  family. At the lowest resolution, with 30 energy unknowns in the RRR, differential errors are upwards of 200 pcm and the cumulative error around 400 pcm. As resolution is increased to 225 unknowns, differential errors decrease to around 60 pcm or less and cumulative error to around 50 pcm. These FEDS energy structures are getting a better integral answer without relying on error cancellation.

Results for problem B are similar to problem A, except problem B has U-235, and so the  $R_{i,e}$  family (bottom row) is expected to be more efficient than the  $R_{i,d}$  family (middle row). The differential errors in  $R_{4,e}$  are a factor of 2 to 4 smaller than those in  $R_{4,d}$ , which are less than 80 pcm. The cumulative errors are approximately the same, due to favorable error cancellation for  $R_{4,d}$ . This shows FEDS can handle disparate resonant nuclides but also shows FEDS can do well if some resonant nuclides are not accounted for.

## 4.2. Two-dimensional pin cells

In this section, we repeat the pin-cell study in a more realistic, full-2D geometry. Compared to 1D, 2D has enhanced shadowing in angle from cylindrical pins in a square lattice, smaller optical depths through the fuel at grazing angles, and increased relative volume of the outer fuel to inner fuel.

Resolution requirements in 2D are higher than in 1D. Figure 2 shows the distribution of spatial cells we used in 2D. For each pincell, 12 rings of cells in the fuel — 4 in the center and 8 near its boundary — and 6 rings of cells in the moderator were used. Within a cell, lumped PWLD was used, which has 4 unknowns per cell in 2D (2 in 1D). For one pincell, the 2D calculation used 5,184 spatial unknowns compared to 60 in 1D. This number could be reduced with lowered azimuthal spatial resolution. A product Gauß-Legendre / Gauß-Chebyshev quadrature was used for the angular quadrature in 2D, with the 16 polar angles and 40 azimuthal angles in one quadrant, which led to 2,560 total angular unknowns in 2D compared to 64 in 1D.  $P_3$  scattering was still found to be sufficient in 2D. In addition to the large processing requirements for the fully resolved 2D deterministic problem, there were commensurate memory requirements, the largest component of which was storage of the angular flux on the problem boundaries to enable reflecting boundaries.

The trends in errors versus energy discretization fidelity for the 2D pincell are the same as from the 1D pincell. Error, as measured with respect to a reference MCNP calculation using consistent cross sections, decreased with increasing energy unknowns in the RRR. Table V shows the  $k$ -eigenvalue error to decrease in a first-order manner for the FEDS method. Figure 3 (bottom row) shows the same trends for total absorption errors for problem B.

The 2D MG problems were interesting because of low errors in the SHEM-361 mesh. The 2D pin-cell

Table IV: Errors in energy discretization  $R_{5,e}$  for problem C for the 1D pincell.

Coarse Group	PDT (rxn. rate)	MCNP mean (rxn. rate)	sigma (%)	(P-M) /M (%)	(P-M) (pcm)
Inner absorption (U-238)					
I	$6.83657 \times 10^{-2}$	$6.83019 \times 10^{-2}$	0.00	0.09	6
II	$1.75628 \times 10^{-2}$	$1.75707 \times 10^{-2}$	0.00	-0.04	-1
III	$1.27617 \times 10^{-2}$	$1.26883 \times 10^{-2}$	0.01	0.58	7
IV	$2.40863 \times 10^{-2}$	$2.40416 \times 10^{-2}$	0.01	0.19	4
V	$2.46261 \times 10^{-2}$	$2.45685 \times 10^{-2}$	0.01	0.23	6
VI	$2.52666 \times 10^{-2}$	$2.54792 \times 10^{-2}$	0.01	-0.83	-21
VII	$2.82720 \times 10^{-2}$	$2.83475 \times 10^{-2}$	0.01	-0.27	-8
VIII	$2.12401 \times 10^{-2}$	$2.14664 \times 10^{-2}$	0.02	-1.05	-23
IX	$2.62440 \times 10^{-2}$	$2.63835 \times 10^{-2}$	0.01	-0.53	-14
X	$3.59815 \times 10^{-2}$	$3.60029 \times 10^{-2}$	0.01	-0.06	-2
XI	$1.00480 \times 10^{-2}$	$1.00132 \times 10^{-2}$	0.01	0.35	3
XII	$2.62318 \times 10^{-2}$	$2.61838 \times 10^{-2}$	0.01	0.18	5
Total	$3.20687 \times 10^{-1}$	$3.21047 \times 10^{-1}$	0.00	-0.11	-36
Fuel fission (U-235)					
I	$3.09515 \times 10^{-2}$	$3.08718 \times 10^{-2}$	0.00	0.26	8
II	$1.15388 \times 10^{-2}$	$1.15343 \times 10^{-2}$	0.00	0.04	0
III	$6.35385 \times 10^{-3}$	$6.33149 \times 10^{-3}$	0.01	0.35	2
IV	$1.48322 \times 10^{-2}$	$1.48157 \times 10^{-2}$	0.00	0.11	2
V	$2.46520 \times 10^{-2}$	$2.47263 \times 10^{-2}$	0.01	-0.30	-7
VI	$3.98576 \times 10^{-2}$	$3.99446 \times 10^{-2}$	0.01	-0.22	-9
VII	$4.66444 \times 10^{-2}$	$4.65747 \times 10^{-2}$	0.01	0.15	7
VIII	$3.30419 \times 10^{-2}$	$3.30463 \times 10^{-2}$	0.01	-0.01	-0
IX	$4.87364 \times 10^{-2}$	$4.86919 \times 10^{-2}$	0.01	0.09	4
X	$2.66385 \times 10^{-2}$	$2.68135 \times 10^{-2}$	0.01	-0.65	-18
XI	$6.93491 \times 10^{-2}$	$6.89334 \times 10^{-2}$	0.01	0.60	42
XII	$5.36665 \times 10^{-1}$	$5.35809 \times 10^{-1}$	0.01	0.16	86
Total	$8.89261 \times 10^{-1}$	$8.88093 \times 10^{-1}$	0.01	0.13	117



geometry is easier than the 1D case because of smaller optical depths through the fuel leading to less spatial self-shielding. Even for coarse groups with resonances not resolved by the energy mesh, the total absorption errors are low for the SHEM-361 mesh, as shown in Fig. 3 (c). This is a consequence of using Bondarenko iterations with background cross sections to approximately preserve bulk reaction rates. If we instead investigate reaction rates in the inner and outer fuel separately, given in Table VI, we find that the SHEM-361 structure along with our condensation scheme does not have sufficient energy resolution in the middle-to-upper RRR (coarse groups IV – VIII) to capture resonance behavior. This leads to large (140 pcm) errors in the spatial distribution of the absorption. Hence, the low errors for SHEM-361 are due to error cancellation.

Conversely, FEDS with the same number of unknowns does have the required energy resolution in this range. Table VII shows FEDS is able to attain low ( $< 10$  pcm) absorption rate errors in all resolved resonance groups with 225 unknowns in the RRR. Both SHEM MG and FEDS are able to achieve low errors in the U-235 reaction rates in the RRR, even at low energy unknown counts in the RRR. This is likely because U-238 is the dominant nuclide, and it self-shields more strongly.

Table V: Errors in  $k$ -eigenvalue for various energy resolutions for problem C for the 2D pincell.

Name	Error (pcm)	Efficiency (1/[error $\times$ DOF])
SHEM-166	$280 \pm 1$	11.9
SHEM-244	$206 \pm 1$	4.5
SHEM-361	$-38 \pm 1$	11.8
$R_{1,e}$	$416 \pm 1$	8.0
$R_{2,e}$	$253 \pm 1$	6.7
$R_{3,e}$	$147 \pm 1$	6.3
$R_{4,e}$	$54 \pm 1$	8.2
$R_{5,e}$	$43 \pm 1$	8.2

## 5. Conclusions

We have applied our Finite-Element-with-Discontiguous-Support (FEDS) method to several pin-cell problems in one and two dimensions. We have shown that errors in reaction-rate based quantities of interest, including  $k_{\text{eff}}$ , decrease with increasing numbers of energy unknowns in the resolved resonance region (RRR) for FEDS, but not unresolved standard MG. We can achieve less than 50 pcm error in all resonance coarse groups for all nuclide reaction rates with 225 energy unknowns in the RRR using FEDS for both 1D and 2D pincells. We have demonstrated that using  $S_n$  in conjunction with FEDS is a powerful method for solving realistic reactor problems at high resolution without requiring onerous energy unknown counts.

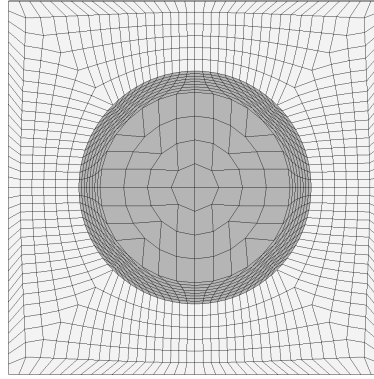


Figure 2: Spatial resolution used in the 2D problems. The fuel is dark grey. The moderator is light grey.

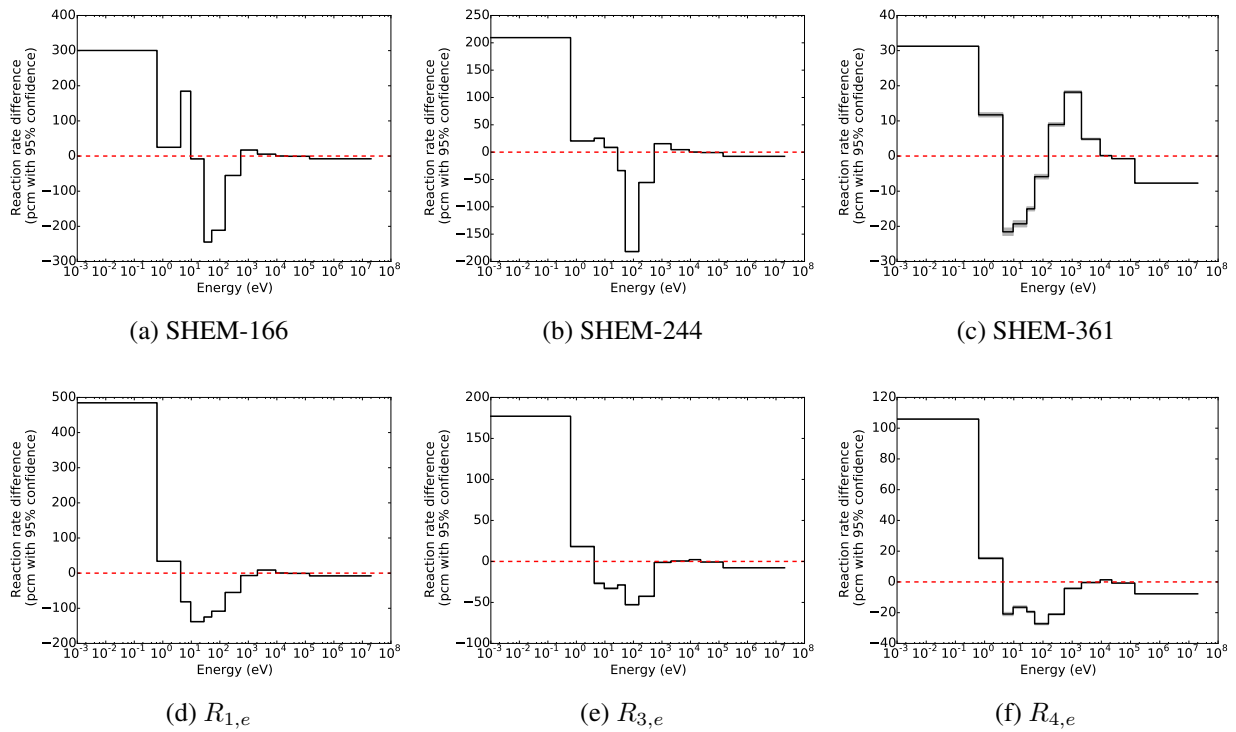


Figure 3: Total absorption errors between MCNP and PDT for different energy structures for problem B for the 2D pincell in pcm, normalized to the total absorption rate. Columns have the same total energy unknowns in the RRR (30, 108, and 225). Rows have the same mesh family (MG and  $R_{i,e}$ ).



Table VI: Errors in energy discretization SHEM-361 for problem C for the 2D pincell.

Coarse Group	PDT (rxn. rate)	MCNP mean (rxn. rate)	sigma (%)	(P-M) /M (%)	(P-M) (pcm)
Inner absorption (U-238)					
I	$2.56545 \times 10^{-2}$	$2.56261 \times 10^{-2}$	0.00	0.11	3
II	$4.29897 \times 10^{-3}$	$4.30199 \times 10^{-3}$	0.00	-0.07	-0
III	$3.27723 \times 10^{-3}$	$3.26787 \times 10^{-3}$	0.01	0.29	1
IV	$6.96448 \times 10^{-3}$	$6.80564 \times 10^{-3}$	0.01	2.33	16
V	$8.80550 \times 10^{-3}$	$8.01728 \times 10^{-3}$	0.02	9.83	79
VI	$1.04305 \times 10^{-2}$	$9.00544 \times 10^{-3}$	0.02	15.82	143
VII	$1.14600 \times 10^{-2}$	$1.06241 \times 10^{-2}$	0.02	7.87	84
VIII	$9.59125 \times 10^{-3}$	$9.26772 \times 10^{-3}$	0.02	3.49	32
IX	$1.21637 \times 10^{-2}$	$1.21337 \times 10^{-2}$	0.02	0.25	3
X	$1.99718 \times 10^{-2}$	$1.99734 \times 10^{-2}$	0.02	-0.01	-0
XI	$4.39400 \times 10^{-3}$	$4.38245 \times 10^{-3}$	0.01	0.26	1
XII	$3.78479 \times 10^{-2}$	$3.77901 \times 10^{-2}$	0.00	0.15	6
Total	$1.54860 \times 10^{-1}$	$1.51196 \times 10^{-1}$	0.00	2.42	366
Outer absorption (U-238)					
I	$1.25343 \times 10^{-2}$	$1.25180 \times 10^{-2}$	0.00	0.13	2
II	$2.14191 \times 10^{-3}$	$2.14899 \times 10^{-3}$	0.00	-0.33	-1
III	$1.64103 \times 10^{-3}$	$1.65314 \times 10^{-3}$	0.01	-0.73	-1
IV	$3.49422 \times 10^{-3}$	$3.61800 \times 10^{-3}$	0.01	-3.42	-12
V	$4.43412 \times 10^{-3}$	$5.03919 \times 10^{-3}$	0.02	-12.01	-61
VI	$5.61361 \times 10^{-3}$	$6.84216 \times 10^{-3}$	0.02	-17.96	-123
VII	$8.70181 \times 10^{-3}$	$9.47338 \times 10^{-3}$	0.02	-8.14	-77
VIII	$8.29859 \times 10^{-3}$	$8.59680 \times 10^{-3}$	0.02	-3.47	-30
IX	$1.21353 \times 10^{-2}$	$1.21903 \times 10^{-2}$	0.02	-0.45	-5
X	$2.20259 \times 10^{-2}$	$2.21023 \times 10^{-2}$	0.02	-0.35	-8
XI	$2.21437 \times 10^{-3}$	$2.20814 \times 10^{-3}$	0.01	0.28	1
XII	$2.04689 \times 10^{-2}$	$2.04360 \times 10^{-2}$	0.00	0.16	3
Total	$1.03704 \times 10^{-1}$	$1.06826 \times 10^{-1}$	0.01	-2.92	-312

Table VII: Errors in energy discretization  $R_{4,e}$  for problem C for the 2D pincell.

Coarse Group	PDT (rxn. rate)	MCNP mean (rxn. rate)	sigma (%)	(P-M) /M (%)	(P-M) (pcm)
Inner absorption (U-238)					
I	$2.55810 \times 10^{-2}$	$2.56261 \times 10^{-2}$	0.00	-0.18	-5
II	$4.29857 \times 10^{-3}$	$4.30199 \times 10^{-3}$	0.00	-0.08	-0
III	$3.27625 \times 10^{-3}$	$3.26787 \times 10^{-3}$	0.01	0.26	1
IV	$6.81003 \times 10^{-3}$	$6.80564 \times 10^{-3}$	0.01	0.06	0
V	$8.01023 \times 10^{-3}$	$8.01728 \times 10^{-3}$	0.02	-0.09	-1
VI	$8.94151 \times 10^{-3}$	$9.00544 \times 10^{-3}$	0.02	-0.71	-6
VII	$1.06042 \times 10^{-2}$	$1.06241 \times 10^{-2}$	0.02	-0.19	-2
VIII	$9.18072 \times 10^{-3}$	$9.26772 \times 10^{-3}$	0.02	-0.94	-9
IX	$1.20965 \times 10^{-2}$	$1.21337 \times 10^{-2}$	0.02	-0.31	-4
X	$1.99327 \times 10^{-2}$	$1.99734 \times 10^{-2}$	0.02	-0.20	-4
XI	$4.39800 \times 10^{-3}$	$4.38245 \times 10^{-3}$	0.01	0.35	2
XII	$3.78831 \times 10^{-2}$	$3.77901 \times 10^{-2}$	0.00	0.25	9
Total	$1.51013 \times 10^{-1}$	$1.51196 \times 10^{-1}$	0.00	-0.12	-18
Outer absorption (U-238)					
I	$1.24975 \times 10^{-2}$	$1.25180 \times 10^{-2}$	0.00	-0.16	-2
II	$2.14161 \times 10^{-3}$	$2.14899 \times 10^{-3}$	0.00	-0.34	-1
III	$1.65313 \times 10^{-3}$	$1.65314 \times 10^{-3}$	0.01	-0.00	-0
IV	$3.60478 \times 10^{-3}$	$3.61800 \times 10^{-3}$	0.01	-0.37	-1
V	$5.01432 \times 10^{-3}$	$5.03919 \times 10^{-3}$	0.02	-0.49	-2
VI	$6.77208 \times 10^{-3}$	$6.84216 \times 10^{-3}$	0.02	-1.02	-7
VII	$9.40313 \times 10^{-3}$	$9.47338 \times 10^{-3}$	0.02	-0.74	-7
VIII	$8.57883 \times 10^{-3}$	$8.59680 \times 10^{-3}$	0.02	-0.21	-2
IX	$1.21650 \times 10^{-2}$	$1.21903 \times 10^{-2}$	0.02	-0.21	-3
X	$2.20306 \times 10^{-2}$	$2.21023 \times 10^{-2}$	0.02	-0.32	-7
XI	$2.21640 \times 10^{-3}$	$2.20814 \times 10^{-3}$	0.01	0.37	1
XII	$2.04879 \times 10^{-2}$	$2.04360 \times 10^{-2}$	0.00	0.25	5
Total	$1.06565 \times 10^{-1}$	$1.06826 \times 10^{-1}$	0.01	-0.24	-26

## 6. Acknowledgments

A special acknowledgment to the Department of Energy Computational Science Graduate Fellowship program (DOE CSGF - grant number DE-FG02-97ER25308), which provides strong support to its fellows and their professional development.

## REFERENCES

- [1] A. T. Till, M. L. Adams, and J. E. Morel. “The Finite Element with Discontiguous Support Multigroup Method: Theory.” In: *Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA), and the Monte Carlo (MC) Method*. American Nuclear Society, Nashville, Tennessee (2015).
- [2] —. “The Finite Element with Discontiguous Support Multigroup Method: Application.” In: *Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA), and the Monte Carlo (MC) Method*. American Nuclear Society, Nashville, Tennessee (2015).
- [3] E. Lewis *et al.* *Benchmark Specification for Deterministic 2-D/3-D MOX Fuel Assembly Transport Calculations without Spatial Homogenization (C5G7 MOX)*. Technical report, NEA/NSC (2001).
- [4] A. T. Till. *A Generalized Multigroup Method*. Master’s thesis, Texas A&M University (2014).
- [5] —. *Finite Elements with Discontiguous Support for Energy Discretization in Particle Transport*. Ph.D. thesis, Texas A&M University, College Station, TX (2015).
- [6] A. Hébert and A. Santamarina. “Refinement of the Santamarina-Hfaiedh Energy Mesh Between 22.5 eV and 11.4 keV.” In: *International Conference on the Physics of Reactors (PHYSOR)*. American Nuclear Society, Interlaken, Switzerland (2008).
- [7] C. N. McGraw *et al.* “Accuracy of the Linear Discontinuous Galerkin Method for Reactor Analysis with Resolved Fuel Pins.” In: *International Conference on the Physics of Reactors (PHYSOR) - The Role of Reactor Physics toward a Sustainable Future*. American Nuclear Society, Kyoto, Japan (2014).
- [8] X-5 Monte Carlo Team. *MCNP - A General Monte Carlo N-Particle Transport Code, Version 5 Volume I: Overview and Theory*. Los Alamos National Laboratory. LA-UR-03-1987 (2008).
- [9] R. E. MacFarlane and D. W. Muir. *The NJOY Nuclear Data Processing System, Version 91*. Los Alamos National Laboratory, Los Alamos, New Mexico. Report LA-12750-M (1994).

# Improved Quasi-Static Method

## IQS Method Implementation for CFEM Diffusion in Rattlesnake

Zachary M. Prince<sup>†</sup>, Jean C. Ragusa<sup>†</sup>, Yaqi Wang<sup>\*</sup>

<sup>†</sup>Department of Nuclear Engineering

Texas A&M University, College Station, TX, USA

<sup>\*</sup>Idaho National Laboratory

[zachm prince@tamu.edu](mailto:zachm prince@tamu.edu), [jean.ragusa@tamu.edu](mailto:jean.ragusa@tamu.edu), [yaqi.wang@inl.gov](mailto:yaqi.wang@inl.gov)

February 29, 2016

## 1 Overview

The improved quasi-static (IQS) method is a transient spatial kinetics method that involves factorizing flux into space- and time-dependent components. These components include the flux's power and shape. Power is time-dependent, while the shape is both space- and time-dependent. However, the impetus of the method is the assumption that the shape is only weakly dependent on time; therefore, the shape may not require computation at every time step, invoking the quasi-static nature.

In this Section, we recall the equations for the IQS method, starting from the standard multigroup diffusion equations written below:

$$\begin{aligned} \frac{1}{v^g} \frac{\partial \phi^g}{\partial t} = & \frac{\chi_p^g}{k_{eff}} (1 - \beta) \sum_{g'=1}^G \nu^{g'} \Sigma_f^{g'} \phi^{g'} - (-\nabla \cdot D^g \nabla + \Sigma_r^g) \phi^g \\ & + \sum_{g' \neq g}^G \Sigma_s^{g' \rightarrow g} \phi^{g'} + \sum_{i=1}^I \chi_{d,i}^g \lambda_i C_i, \quad 1 \leq g \leq G \end{aligned} \quad (1)$$

$$\frac{dC_i}{dt} = \frac{\beta_i}{k_{eff}} \sum_{g=1}^G \nu^g \Sigma_f^g \phi^g - \lambda_i C_i, \quad 1 \leq i \leq I \quad (2)$$

with

$$\beta = \sum_{i=1}^I \beta_i \quad (3)$$

Factorization is the most important step in the derivation of the IQS method. The factorization approach leads to a decomposition of the multigroup flux into the product of a time-dependent amplitude ( $p$ ) and a space-/time-dependent multigroup shape ( $\varphi^g$ ):

$$\phi^g(\vec{r}, t) = p(t) \varphi^g(\vec{r}, t) \quad (4)$$

Then the flux and precursor equations become:

$$\begin{aligned} \frac{1}{v^g} \left( \frac{dp}{dt} \varphi^g + p \frac{\partial \varphi^g}{\partial t} \right) = & \frac{\chi_p^g}{k_{eff}} (1 - \beta) \sum_{g'=1}^G \nu^{g'} \Sigma_f^{g'} p \varphi^{g'} - (-\nabla \cdot D^g \nabla + \Sigma_r^g) p \varphi^g \\ & + \sum_{g' \neq g}^G \Sigma_s^{g' \rightarrow g} p \varphi^{g'} + \sum_{i=1}^I \chi_{d,i}^g \lambda_i C_i, \quad 1 \leq g \leq G \end{aligned} \quad (5)$$

$$\frac{dC_i}{dt} = \frac{\beta_i}{k_{eff}} \sum_{g=1}^G \nu^g \Sigma_f^g p \varphi^g - \lambda_i C_i, \quad 1 \leq i \leq I \quad (6)$$

PRKE formulation (amplitude equations):

$$\begin{aligned} \sum_{g=1}^G \int_D \left[ \frac{1}{v^g} \left( \frac{dp}{dt} \varphi^g + p \frac{\partial \varphi^g}{\partial t} \right) \right] w^g d^3 r = \\ \sum_{g=1}^G \int_D \left[ \frac{\chi_p^g}{k_{eff}} (1 - \beta) \sum_{g'=1}^G \nu^{g'} \Sigma_f^{g'} p \varphi^{g'} - (-\nabla \cdot D^g \nabla + \Sigma_r^g) p \varphi^g \right] w^g d^3 r \\ + \sum_{g=1}^G \int_D \left[ \sum_{g' \neq g}^G \Sigma_s^{g' \rightarrow g} p \varphi^{g'} + \sum_{i=1}^I \chi_{d,i}^g \lambda_i C_i \right] w^g d^3 r, \quad 1 \leq g \leq G \end{aligned} \quad (7)$$

$$\int_D \frac{dC_i}{dt} \sum_{g=1}^G \chi_{d,i}^g w^g d^3 r = \int_D \frac{\beta_i}{k_{eff}} \left[ \sum_{g=1}^G \left( \sum_{g'=1}^G \nu^{g'} \Sigma_f^{g'} \varphi^{g'} - \lambda_i C_i \right) \chi_{d,i}^g w^g \right] d^3 r, \quad 1 \leq i \leq I \quad (8)$$

It can be shown that the most appropriate weighting function ( $w^g$ ) is the initial adjoint flux ( $\phi^{*g}$ ). For brevity, the following definition will be applied:  $\int_D \phi^{*g}(\vec{r}) f(\vec{r}) d^3 r = (\phi^{*g}, f)$

$$\begin{aligned} \frac{dp}{dt} \sum_{g=1}^G \left( \phi^{*g}, \frac{1}{v^g} \varphi^g \right) + p \frac{d}{dt} \sum_{g=1}^G \left( \phi^{*g}, \frac{1}{v^g} \varphi^g \right) = \\ p \sum_{g=1}^G \left( \phi^{*g}, \frac{\chi_p^g}{k_{eff}} (1 - \beta) \sum_{g'=1}^G \nu^{g'} \Sigma_f^{g'} \varphi^{g'} + \sum_{g' \neq g}^G \Sigma_s^{g' \rightarrow g} \varphi^{g'} - (-\nabla \cdot D^g \nabla + \Sigma_r^g) \varphi^g \right) \\ + \sum_{i=1}^I \sum_{g=1}^G (\phi^{*g}, \chi_{d,i}^g \lambda_i C_i) \end{aligned} \quad (9)$$

$$\frac{d}{dt} \sum_{g=1}^G (\phi^{*g}, \chi_{d,i}^g C_i) = \frac{1}{k_{eff}} \sum_{g=1}^G (\phi^{*g}, \chi_{d,i}^g \beta_i \sum_{g'=1}^G \nu^{g'} \Sigma_f^{g'} \varphi^{g'}) p - \sum_{g=1}^G (\phi^{*g}, \chi_{d,i}^g \lambda_i C_i) \quad 1 \leq i \leq I \quad (10)$$

In order to impose uniqueness of the factorization, one imposes:

$$\sum_{g=1}^G \left( \phi^{*g}, \frac{1}{v^g} \varphi^g \right) = \text{constant} \quad (11)$$

Therefore the PRKE formulation reduces to:

$$\begin{aligned} \frac{dp}{dt} = \frac{\sum_{g=1}^G \left( \phi^{*g}, \frac{\chi_p^g}{k_{eff}} (1 - \beta) \sum_{g'=1}^G \nu^{g'} \Sigma_f^{g'} \varphi^{g'} + \sum_{g' \neq g}^G \Sigma_s^{g' \rightarrow g} \varphi^{g'} - (-\nabla \cdot D^g \nabla + \Sigma_r^g) \varphi^g \right)}{\sum_{g=1}^G \left( \phi^{*g}, \frac{1}{v^g} \varphi^g \right)} p \\ + \sum_{i=1}^I \frac{\sum_{g=1}^G (\phi^{*g}, \chi_{d,i}^g \lambda_i C_i)}{\sum_{g=1}^G (\phi^{*g}, \chi_{d,i}^g C_i)} \xi_i \end{aligned} \quad (12)$$

$$\frac{d\xi_i}{dt} = \frac{1}{k_{eff}} \frac{\sum_{g=1}^G (\phi^{*g}, \chi_{d,i}^g \beta_i \sum_{g'=1}^G \nu^{g'} \Sigma_f^{g'} \varphi^{g'})}{\sum_{g=1}^G \left( \phi^{*g}, \frac{1}{v^g} \varphi^g \right)} p - \frac{\sum_{g=1}^G (\phi^{*g}, \chi_{d,i}^g \lambda_i C_i)}{\sum_{g=1}^G (\phi^{*g}, \chi_{d,i}^g C_i)} \xi_i \quad 1 \leq i \leq I \quad (13)$$

Where:

$$\xi_i = \frac{\sum_{g=1}^G (\phi^{*g}, \chi_{d,i}^g C_i)}{\sum_{g=1}^G \left( \phi^{*g}, \frac{1}{v^g} \varphi^g \right)} \quad (14)$$

It is convenient to define the effective reactivity, delay-neutron fraction, and delayed-neutron precursor decay constant:

$$\frac{\rho - \bar{\beta}}{\Lambda} = \frac{\sum_{g=1}^G \left( \phi^{*g}, \frac{\chi_p^g}{k_{eff}} (1 - \beta) \sum_{g'=1}^G \nu^{g'} \Sigma_f^{g'} \varphi^{g'} + \sum_{g' \neq g}^G \Sigma_s^{g' \rightarrow g} \varphi^{g'} - (-\nabla \cdot D^g \nabla + \Sigma_r^g) \varphi^g \right)}{\sum_{g=1}^G \left( \phi^{*g}, \frac{1}{v^g} \varphi^g \right)} \quad (15)$$

$$\frac{\bar{\beta}}{\Lambda} = \sum_{i=1}^I \frac{\bar{\beta}_i}{\Lambda} = \sum_{i=1}^I \frac{1}{k_{eff}} \frac{\sum_{g=1}^G (\phi^{*g}, \chi_{d,i}^g \beta_i \sum_{g'=1}^G \nu^{g'} \Sigma_f^{g'} \varphi^{g'})}{\sum_{g=1}^G (\phi^{*g}, \frac{1}{v^g} \varphi^g)} \quad (16)$$

$$\bar{\lambda}_i = \frac{\sum_{g=1}^G (\phi^{*g}, \chi_{d,i}^g \lambda_i C_i)}{\sum_{g=1}^G (\phi^{*g}, \chi_{d,i}^g C_i)} \quad (17)$$

So:

$$\frac{dp}{dt} = \left[ \frac{\rho - \bar{\beta}}{\Lambda} \right] p + \sum_{i=1}^I \bar{\lambda}_i \xi_i \quad (18)$$

$$\frac{d\xi_i}{dt} = \frac{\bar{\beta}_i}{\Lambda} - \bar{\lambda}_i \xi_i \quad 1 \leq i \leq I \quad (19)$$

Equations (18) and (19) are a formulation of the point reactor kinetics equation (PRKE), but the parameters (Equations (15)-(17)) are dependent on the shape. If the assumption is made that the shape is time-independent, the shape is computed once at the first time step and used for the PRKE parameter evaluation at all other steps. However, if the shape is dependent on time, the shape needs to be computed in transient using equation (5) and (6) in order retain accuracy. Equations (20) and (21) shows the usual form of the shape and precursor equations with amplitude put on the right hand side. Equation (20) is very similar to the multigroup flux equation (1), except the removal cross-section term is augmented by a  $\frac{1}{v^g} \frac{1}{p} \frac{dp}{dt}$  term and the precursor contribution has a  $\frac{1}{p}$  multiplier. Equation (21) is very similar to the normal precursor equation (2), except the fission source term is multiplied by  $p$ . These differences are crucial for IQS implementation in Rattlesnake.

$$\begin{aligned} \frac{1}{v^g} \frac{\partial \varphi^g}{\partial t} = & \frac{\chi_p^g}{k_{eff}} (1 - \beta) \sum_{g'=1}^G \nu^{g'} \Sigma_f^{g'} \varphi^{g'} + \sum_{g' \neq g}^G \Sigma_s^{g' \rightarrow g} \varphi^{g'} \\ & - \left( -\nabla \cdot D^g \nabla + \Sigma_r^g + \frac{1}{v^g} \frac{1}{p} \frac{dp}{dt} \right) \varphi^g + \frac{1}{p} \sum_{i=1}^I \chi_{d,i}^g \lambda_i C_i, \quad 1 \leq g \leq G \end{aligned} \quad (20)$$

$$\frac{dC_i}{dt} = \frac{\beta_i}{k_{eff}} p \sum_{g=1}^G \nu^g \Sigma_f^g \varphi^g - \lambda_i C_i, \quad 1 \leq i \leq I \quad (21)$$

Computing this shape can become expensive, especially in two or three dimensions. Subsequently, it is attractive to make the assumption that the shape is weakly time-dependent so the shape can be computed after a multitude of PRKE calculations which is the root of IQS. To visualize:

Additionally, to improve consistency and accuracy, each macro time step can be iterated so the best shape is used to compute power at the micro time steps. This iteration process must converge the shape such that the uniqueness condition  $(\frac{d}{dt} \sum_{g=1}^G (\phi^{*g}, \frac{1}{v^g} \varphi^g) = 0)$  is preserved.

## 2 Rattlesnake Implementation

This section will explain how, thus far, IQS has been implemented in Rattlesnake.

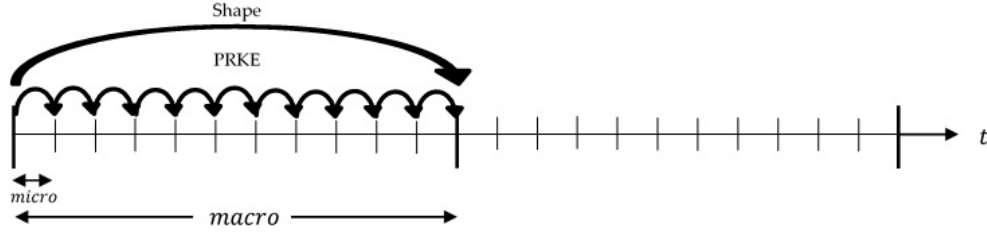


Figure 1: IQS method visualization

## 2.1 Executioner

The IQS executioner derives from the Transient executioner in MOOSE. The IQS executioner contains a loop over micro time steps that computes the PRKE and then passes  $p$  and  $\frac{dp}{dt}$  for the Transient executioner to evaluate the shape equation at each macro step. The PRKE is computed with backward Euler to retain simplicity and insure convergence, but higher order methods are an obvious next step for this computation. The IQS executioner also supplements Transients Picard iteration process by adding its own error criteria:

$$Error_{IQS} = \left| \frac{\sum_{g=1}^G (\phi^{*g}, \frac{1}{v^g} \varphi^{g,n})}{\sum_{g=1}^G (\phi^{*g}, \frac{1}{v^g} \varphi^{g,0})} - 1 \right| \quad (22)$$

## 2.2 Action System

IQS defines its uniqueness from its executioner type; however, many changes needed to be made in the Rattlesnake action system in order to support IQS execution. First, changes needed to be made in order to evaluate the shape equation. The shape equation, after some manipulation, is very similar to the time-dependent, which Rattlesnake is already set up to solve:

$$\begin{aligned} \frac{1}{v^g} \frac{\partial \varphi^g}{\partial t} = & \underbrace{\frac{\chi_p^g}{k_{eff}} \sum_{g'=1}^G (1-\beta) \nu^{g'} \Sigma_f^{g'} \varphi^{g'}}_{\substack{FluxKernel \\ FromExecutioner}} + \underbrace{\sum_{g' \neq g} \Sigma_s^{g' \rightarrow g} \varphi^{g'}}_{FluxKernel} - \underbrace{(-\nabla \cdot D^g \nabla) \varphi^g}_{FluxKernel} - \underbrace{\Sigma_r^g \varphi^g}_{FluxKernel} \\ & - \underbrace{\frac{1}{v^g} \frac{1}{p} \frac{dp}{dt} \varphi^g}_{IQSKernel} + \underbrace{\frac{1}{p} \sum_{i=1}^I \chi_{d,i}^g \lambda_i C_i}_{ModifiedFluxKernel} \end{aligned} \quad (23)$$

To enable Rattlesnake to solve this equation, another kernel was created that evaluated  $\sum_{g=1}^G \frac{1}{v^g p} \frac{dp}{dt} \varphi^g$  and added when the IQS executioner is called. Second, four postprocessors were created in order to calculate the PRKE parameters. The parameter calculations were separated by  $\frac{\bar{\beta}_i}{\Lambda}$  numerator,  $\bar{\lambda}_i$  numerator/denominator,  $\frac{\rho}{\Lambda} / \frac{\bar{\beta}}{\Lambda}$  denominator, and  $\frac{\rho - \bar{\beta}}{\Lambda}$  numerator. The first three are relatively simple, only relying on material properties and solution quantities. The  $\frac{\rho - \bar{\beta}}{\Lambda}$  numerator requires the use of the MOOSE save in feature, which saves the residual from a calculated kernel or boundary contribution in the shape evaluation to an auxiliary variable. Finally, a user object was created to pull together all the postprocessor values and carryout the numerator/denominator divisions that were then passed to the executioner.

## 2.3 Precursor Integration

This section presents two different time-integration methods to solve coupled IQS shape + precursor equations, recalled below using, for simplicity, a single neutron group and a single precursor group.

$$\frac{1}{v} \frac{\partial \varphi}{\partial t} = \nu \Sigma_f (1 - \beta) \varphi - \left( -\nabla \cdot D \nabla + \Sigma_a + \frac{1}{v} \frac{dp}{dt} \right) \varphi + \frac{1}{p} \lambda C \quad (24)$$

$$\frac{dC}{dt} = \beta \nu \Sigma_f \varphi p - \lambda C \quad (25)$$

First, we note that we could keep this system of two time-dependent equations and solve it as a coupled system. However, this is unnecessary and a memory expensive endeavor because the precursor equation is only an ODE and not a PDE. Instead, one may discretize in time the shape equation, which typically requires the knowledge of the precursor concentrations at the end of the time step. This precursor value is taken from the solution, numerical or analytical, of the precursors ODE. This document will discuss two techniques for solving the precursor equation. First is a time discretization method that is currently being implemented in Rattlesnake. The second is an analytical integration of the precursors, the latter method has proven to be more beneficial for IQS convergence.

### 2.3.1 Time Discretization using the Theta Method

A fairly simple way to evaluate the precursor equation is to employ the  $\theta$ -scheme ( $0 \leq \theta \leq 1$ ), explicit when  $\theta = 0$ , implicit when  $\theta = 1$ , and Crank-Nicholson when  $\theta = 1/2$ ). Generally, if there is a function  $u$  whose governing equation is  $\frac{du}{dt} = f(u, t)$ , then the  $\theta$ -discretization is

$$\frac{u^{n+1} - u^n}{\Delta t} = (1 - \theta) f(u^n, t) + \theta f(u^{n+1}, t). \quad (26)$$

Applying this to the precursor equation:

$$\frac{C^{n+1} - C^n}{\Delta t} = (1 - \theta) \beta S_f^n p^n - (1 - \theta) \lambda C^n + \theta \beta S_f^{n+1} p^{n+1} - \theta \lambda C^{n+1} \quad (27)$$

Where  $S_f$  is the fission source equivalent for shape:

$$S_f^n = (\nu \Sigma_f)^n \varphi^n \quad (28)$$

Rearranging to solve for the precursor at the end of the time step yields

$$C^{n+1} = \frac{1 - (1 - \theta) \Delta t \lambda}{1 + \theta \Delta t \lambda} C^n + \frac{(1 - \theta) \Delta t \beta}{1 + \theta \Delta t \lambda} S_f^n p^n + \frac{\theta \Delta t \beta}{1 + \theta \Delta t \lambda} S_f^{n+1} p^{n+1} \quad (29)$$

Reporting this value of  $C^{n+1}$ , one can solve for the shape  $\varphi^{n+1}$  as a function of  $\varphi^n$  and  $C^n$  (and  $p^n$ ,  $p^{n+1}$ ,  $dp/dt|_n$  and  $dp/dt|_{n+1}$ ). Once  $\varphi^{n+1}$  has been determined,  $C^{n+1}$  is updated. Rattlesnake currently implements both implicit and Crank-Nicholson as options for precursor evaluation.

### 2.3.2 Analytical Integration

Through prototyping, it has been found that neither implicit nor Crank-Nicholson time discretization of precursors are preferable methods for solving the shape equation in IQS. It has been found that these discretizations result in a lack of convergence of the shape over the IQS iteration process. In order to remedy the error, an analytical representation of the precursors was implemented in the prototype and the shape solution was able to converge (the normalization constant of the IQS method can be preserved to  $10^{-10}$  while the theta-scheme only allowed convergence in the normalization factor to about  $10^{-3}$ ). The following section shows how this method was implemented in the prototype and the desired implementation for Rattlesnake.



Using an exponential operator, the precursor equation can be analytically solved for:

$$\int_{t_n}^{t_{n+1}} C(t') e^{\lambda t'} dt' = \int_{t_n}^{t_{n+1}} \beta(t') S_f(t') p(t') e^{\lambda t'} dt' \quad (30)$$

yielding

$$C^{n+1} = C^n e^{-\lambda(t_{n+1}-t_n)} + \int_{t_n}^{t_{n+1}} \beta(t') S_f(t') p(t') e^{-\lambda(t_{n+1}-t')} dt' \quad (31)$$

Because  $\beta$  and  $S_f$  being integrated are not known continuously over the time step, they can be interpolated linearly over the macro step. Such that:

$$h(t) = \frac{t_{n+1}-t}{t_{n+1}-t_n} h^n + \frac{t-t_n}{t_{n+1}-t_n} h^{n+1} \quad t_n \leq t \leq t_{n+1} \quad (32)$$

However, for the PRKE solve, we do have a very accurate representation of  $p(t')$  over the time interval  $[t_n, t_{n+1}]$ .

Finally, we have the final expression for the analytical value for  $C^{n+1}$ :

$$C^{n+1} = C^n e^{-\lambda \Delta t} + (a_3 \beta^{n+1} + a_2 \beta^n) S_f^{n+1} + (a_2 \beta^{n+1} + a_1 \beta^n) S_f^n \quad (33)$$

Where the integration coefficients are defined as:

$$a_1 = \int_{t_n}^{t_{n+1}} \left( \frac{t_{n+1}-t'}{\Delta t} \right)^2 p(t') e^{-\lambda(t_{n+1}-t')} dt' \quad (34)$$

$$a_2 = \int_{t_n}^{t_{n+1}} \frac{(t'-t_n)(t_{n+1}-t')}{(\Delta t)^2} p(t') e^{-\lambda(t_{n+1}-t')} dt' \quad (35)$$

$$a_3 = \int_{t_n}^{t_{n+1}} \left( \frac{t'-t_n}{\Delta t} \right)^2 p(t') e^{-\lambda(t_{n+1}-t')} dt' \quad (36)$$

The amplitude ( $p$ ) is included in the integration coefficient because it has been highly accurately calculated in the micro step scheme, so a piecewise interpolation between those points can be done to maximize accuracy.

The prototype code uses Matlab software to interpolate the amplitude between micro steps and a quadrature integration for the coefficients. So the challenge for Rattlesnake is to replicate this procedure: passing the amplitude vector to the DNP auxkernel, interpolating it, and integrating the coefficients.

## 2.4 Input

The input deck for IQS is very similar to the current transient diffusion input file. The IQS input has a different executioner type and parameters. The executioner type is simply IQS and input parameters include number of micro time steps, IQS error tolerance, and initial power. The Rattlesnake transient action system currently requires a multi-app and transfer to compute and pass the initial  $\phi$  and  $k_{eff}$ , which is present in the transient input deck. However, IQS also requires an initial evaluation of the adjoint flux, for the weighting function. So another input file was made for the adjoint calculation, as well as including another multi-app and transfer in the IQS input deck.

## 2.5 Unintended Contributions

The implementation of IQS in Rattlesnake put pressure on many features of Rattlesnake and MOOSE that revealed bugs and possible improvement. Two significant issues in Rattlesnake that were found involved the adjoint solve and the diffusion fission kernel. When testing IQS, it was found that the adjoint flux solution was not the same as the forward flux solution in a single group test, which is obviously invalid. Also, when investigating which action to include the IQS kernel

in, it turned out that the fission diffusion kernel was placed in the neutron transport action, so this kernel was demoted to the neutron diffusion actions for clarity. Additionally, the pressure on the save in feature in MOOSE propagated its application to boundary conditions and initial solves. MOOSE also updated its ability to restart dense vector data and to set MooseApp executioner right after executioner is created. Merge requests: #5474, #5489, #5495, and #5497

### 3 Current Status

IQS has almost completely been implemented to CFEM Diffusion in Rattlesnake. The method currently passes multi-dimensional and multi-group null-transient tests. However, there are currently three prevalent issues that are restricting full implementation of IQS in Rattlesnake.

1. The Transient executioner is currently being worked on to improve its flexibility. Until this improvement is complete, the IQS error contribution is unable to be supplemented to Transient. For testing purposes, Transient has been modified locally to support the error contribution.
2. IQS is currently being tested with higher order schemes for diffusion evaluation in the matlab prototype. This is to make sure that IQS will perform with similarly when subjected to higher accuracy.

After CFEM implementation is completed, the next step is to apply IQS to DFEM and ultimately neutron transport. These applications should be much simpler because the base of IQS has already been implemented and verified from this CFEM work. Another method of IQS, called the predictor-corrector method, could also be implemented in the future. This method is very simple to implement and actually faster than standard IQS because there is no iteration process. However, the performance of the method has yet to be evaluated to prove its worth in Rattlesnake.

## 4 RESULTS

This section describes results of an examples that tests the IQS implementation and shows its effectiveness on computation speed and accuracy. Two examples were selected for this purpose. The first is a homogeneous one-group problem, subjected to a heterogenous material change (absorption cross-section change as a ramp in time for a subset of the geometry). The second is the two-dimensional TWIGL ramp transient benchmark, described further.

### 4.1 One-Dimensional Custom Example

The example is very simple and computes quickly, it entails a one dimensional, heterogeneous 400 cm slab with a varying absorption cross section. Figure 2 how the regions of the slab are divided and Table 1 shows the initial material properties. Table 2 shows the ramp of the absorption cross-section of each region.

1	1	1	1	2	3	1	1	1	1	1	1	1	1	1	4	4	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Figure 2: 1-D heterogeneous slab region identification

Region	$D(cm)$	$\Sigma_a(cm^{-1})$	$\nu\Sigma_f(cm^{-1})$	$v(cm/s)$	$\beta$	$\lambda(s^{-1})$
1	1.0	1.1	1.1	1,000	0.006	0.1
2	1.0	1.1	1.1	1,000	0.006	0.1
3	1.0	1.1	1.1	1,000	0.006	0.1
4	1.0	1.1	1.1	1,000	0.006	0.1

Table 1: 1-D heterogeneous slab material properties and problem parameters

Material Property	0.0 s	0.1 s	0.6 s	1.0 s	1.7 s
$\Sigma_{a,2}(cm^{-1})$	1.1	1.1	1.095	1.095	1.095
$\Sigma_{a,3}(cm^{-1})$	1.1	1.1	1.09	1.09	1.1
$\Sigma_{a,4}(cm^{-1})$	1.1	1.1	1.105	1.105	1.105

Table 2: 1-D heterogeneous slab absorption cross-section slope perturbation

Figure 7 shows the power at each macro time step as compared to the traditional brute force (full flux time discretization) method. The strong correlation between the two curves shows that IQS is consistent with a proven method for a highly transient example. Figure ?? shows that IQS is not only consistent for this example, but also has a better error constant in the convergence study. Figures 4 - 6 plots shape changes in the IQS method, showing where the shape solution is necessary and a simple PRKE evaluation is inadequate.

### 4.2 TWIGL Benchmark

This benchmark problem originates from the Argonne National Lab Benchmark Problem Book. It is a 2D, 2-group reactor core model with no reflector region shown in Figure 8. Table 3 shows the material properties of each fuel region and the ramp perturbation of Material 1.

Figures 9 and 10 show the IQS solution as compared with the Brute Force solution. It is important to note the IQS shape plot is scaled differently than the Brute Force flux plot because

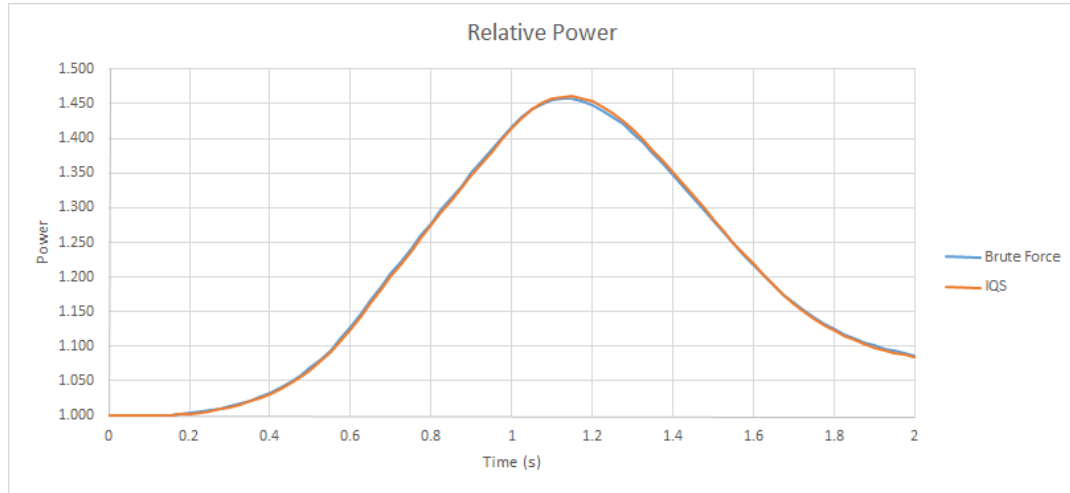


Figure 3: Power level comparison of 1D heterogeneous example between IQS and Brute Force using  $\Delta t = 0.025$

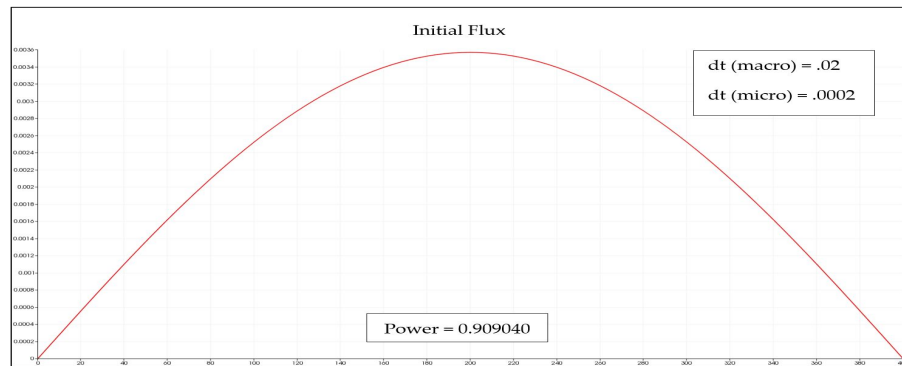


Figure 4: Initial Flux Plot

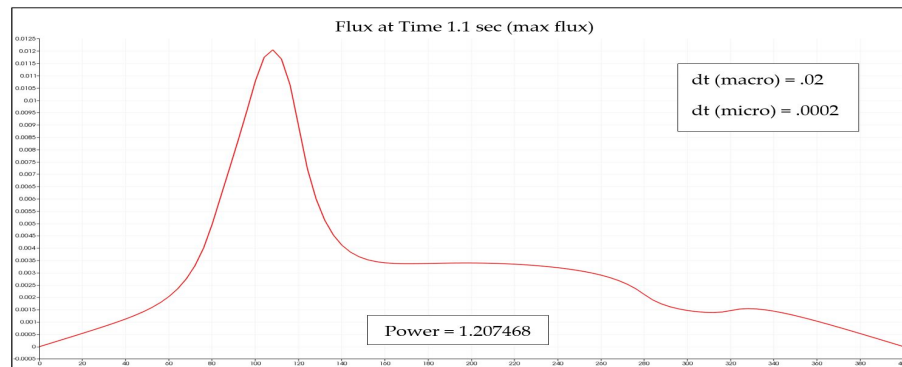


Figure 5: Flux Plot when Absorption Cross Section is at Minimum

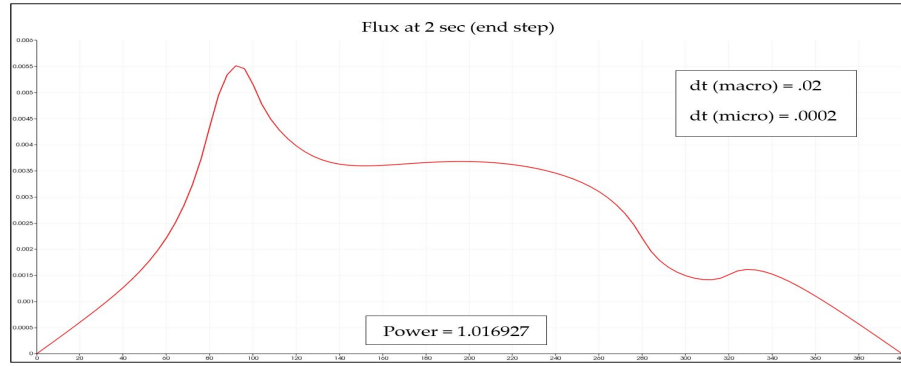


Figure 6: Final Flux Computation (not steady-state)

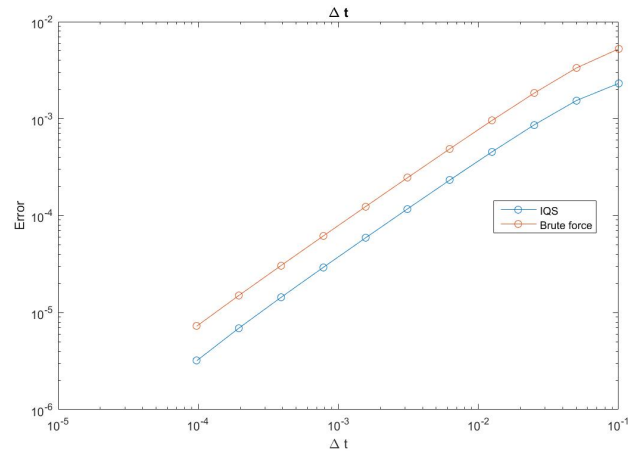


Figure 7: Error convergence comparison of 1D heterogeneous example

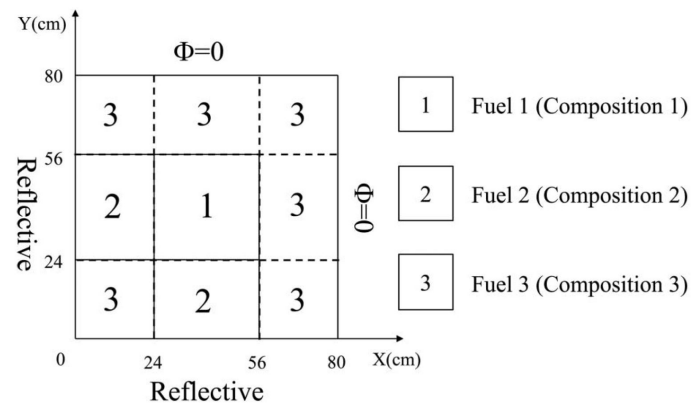


Figure 8: TWIGL benchmark problem description

Material	Group	$D(cm)$	$\Sigma_a(cm^{-1})$	$\nu\Sigma_f(cm^{-1})$	$\chi$	$\Sigma_s(cm^{-1})$	
						$g \rightarrow 1$	$g \rightarrow 2$
1	1	1.4	0.010	0.007	1.0	0.0	0.01
	2	0.4	0.150	0.200	0.0	0.0	0.00
2	1	1.4	0.010	0.007	1.0	0.0	0.01
	2	0.4	0.150	0.200	0.0	0.0	0.00
3	1	1.3	0.008	0.003	1.0	0.0	0.01
	2	0.5	0.050	0.060	0.0	0.0	0.00
$\nu$		$v_1(cm/s)$	$v_2(cm/s)$	$\beta$	$\lambda(1/s)$		
2.43		1.0E7	2.0E5	0.0075	0.08		

Material 1 ramp perturbation:

$$\Sigma_{a,2}(t) = \Sigma_{a,2}(0) \times (1 - 0.11667t) \quad t \leq 0.2s$$

$$\Sigma_{a,2}(t) = \Sigma_{a,2}(0) \times (0.97666t) \quad t > 0.2s$$

Table 3: 1-D heterogeneous slab absorption cross-section slope perturbation

the amplitude term is not included, but the gradients of colors is comparable. These plots show that IQS is consistent in more complex, higher dimensional problems in RATTLESNAKE. Finally, Figure 11 plots the error convergence of IQS and the Brute Force methods. The curves show the impressive convergence of IQS for the highly transience TWIGL example.

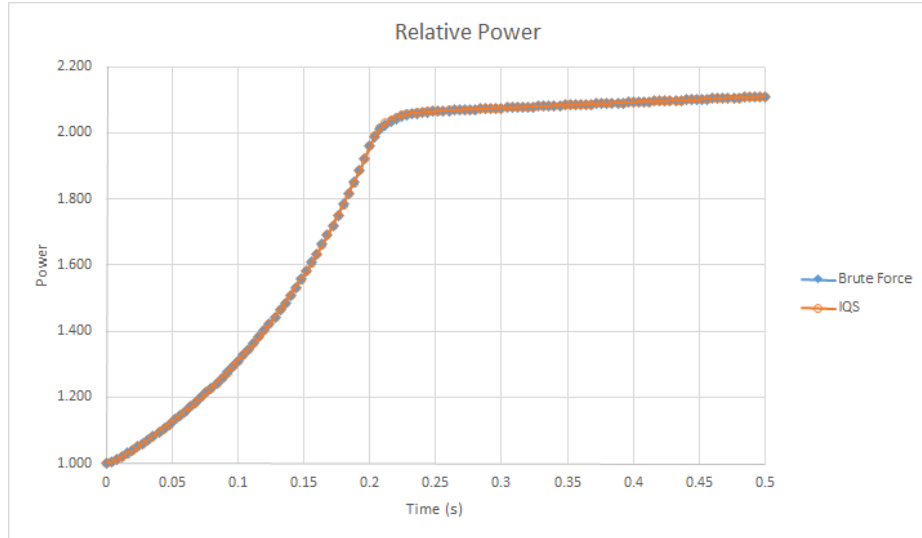
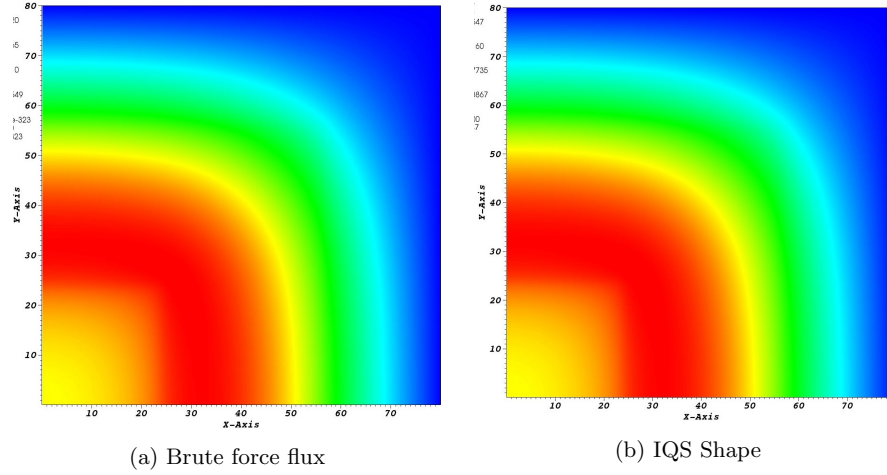
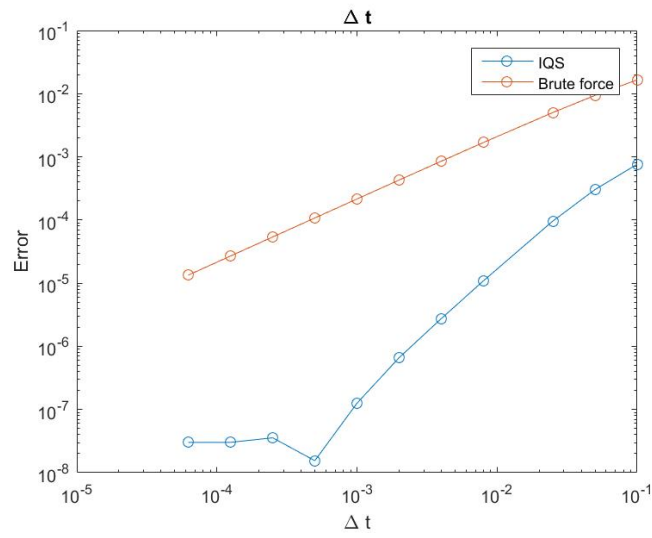


Figure 9: Power level comparison of 1D heterogeneous example between IQS and Brute Force using  $\Delta t = 0.004$

Figure 10: TWIGL Benchmark flux/shape comparison at  $t = 0.2$ 

# Neutronic–Thermal Hydraulic Benchmark using the SKINATH Point Kinetics Model

B. Ganapol

Department of Aerospace and Mechanical Engineering  
University of Arizona, Tucson, AZ, USA

M. DeHart, F. Gleicher, J. Ortensi and R. Martineau  
Nuclear Systems Design and Analysis Division  
Idaho National Laboratory, Idaho Falls, ID, USA

## INTRODUCTION

The **SKINATH** point kinetics program [1], whose name comes from: A Computer Program for Solving the Reactor Point **K**INetics EquAtions with Simple **T**hermal Hydraulic-Feedback, had its origins at ORNL. The authors were interested in investigating low-level criticality events such as might occur in a waste repository of 55 gallon drums. The physical setting was to incorporate a relatively complex thermal feedback for a fissionable material in a container (called a reactor here), either critical or subcritical, with the potential to undergo a critical excursion. The system is in a fluid, which absorbs all heat from the excursion and remains constant temperature  $T_c$ . Thus, the fluid is a heat sink enabling the reactor to cool down. The **SKINATH** numerical concept was to use an off-the-shelf Livermore ODE solver **LSODE** [2] to solve the kinetic equations, much as one does today with **MATLAB**<sup>™</sup>. Here, we develop a novel solver that uses the simplest of numerical methods including convergence acceleration. This work represents the initiation of a Multiphysics benchmark effort at INL.

## REACTOR KINETICS MODEL

The kinetics model begins with a heat balance for a core of average temperature  $T(t)$  according to

$$C_p \frac{dT(t)}{dt} = \varepsilon N(t) - h_T(t) [T(t) - T_c], \quad (1a)$$

where  $h_T(t)$  is the convection heat transfer coefficient

$$h_T(t) \equiv Ahd^{0.75} \left[ 1 - \frac{T_c}{T(t)} \right]^{0.25} \quad (1b)$$

Table 1 gives the constants found in  $h_T(t)$  with values found in the **SKINATH** benchmarks below.  $N(t)$  is the reactor power,  $C_p$  is the core specific heat capacity, and  $\varepsilon$  is the fraction of reactor power deposited as heat.

Table 1. **SKINATH** Model parameters

Symbol	Description	Unit or value
$A$	$1.649K^{0.75} (\rho_c g \hat{C}_p / \mu)^{0.25}$	17.52 for air
$K$	coolant thermal conductivity	J/m-s°C
$g$	gravity constant	m/s <sup>2</sup>
$\rho_c$	coolant density	kg/m <sup>3</sup>
$\mu$	coolant viscosity	kg/m-s
$T_c$	coolant temperature	20°C
$C_p$	coolant specific heat	1800πd <sup>2</sup> /4h J/kg°C
$h$	reactor height	0.23m
$d$	reactor diameter	0.20m

We model the transient reactor power with the source less point kinetics equations

$$\begin{aligned} \frac{dN(t)}{dt} &= \left[ \frac{\rho(t, N) - \beta}{\Lambda} \right] N(t) + \sum_{l=1}^m \lambda_l C_l(t) \\ \frac{dC_l(t)}{dt} &= \frac{\beta_l}{\Lambda} N(t) - \lambda_l C_l(t), \quad l = 1, \dots, m, \end{aligned} \quad (2a,b)$$

with  $m$  (6) delayed groups. The transient begins with initial reactor power  $N(0)$  from steady state to give the following initial precursor concentrations:

$$C_l(0) = \frac{\beta_l}{\lambda_l \Lambda} N(0), \quad l = 1, \dots, m \quad (3)$$

and  $\rho(0^-) = 0$ . We assume simple Doppler feedback

$$\rho_D(t, N) = \alpha_D [T(t) - T_c] \quad (4a)$$

along with an initiating reactivity to give the total transient reactivity at any time  $t$

$$\rho(t, T(t)) = \rho_0(t) - \alpha_D [T(t) - T_c]. \quad (4b)$$

In the **SKINATH** model, the reactor at initial power, is at the fluid temperature— thus, ignoring the initial fission heating— a reasonable assumption for low power excursions.



## MATHEMATICAL FORMULATION

As is common practice, one recasts the model equations as a set of vector equations

$$\frac{dy(t)}{dt} = A(T(t))y(t) + q(t), \quad (5a)$$

where

$$y(t) = [N(t) \ C_1(t) \ \dots \ C_m(t) \ T(t) - T_c]^T \quad (5b,c)$$

and for the Jacobian

$$A(t, T(t)) = \begin{bmatrix} (\rho(T(t)) - \beta)/\Lambda & \lambda_1 & \lambda_2 & \dots & \lambda_m & 0 \\ \frac{\beta_1}{\Lambda} & -\lambda_1 & 0 & \dots & \dots & 0 \\ \frac{\beta_2}{\Lambda} & 0 & -\lambda_2 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{\beta_m}{\Lambda} & 0 & 0 & \dots & -\lambda_m & 0 \\ \frac{\varepsilon}{C_p} & 0 & \dots & \dots & \dots & -h_r(t)/C_p \end{bmatrix} \quad (5d)$$

Note that the Jacobian is dependent upon temperature.

## NUMERICAL IMPLEMENTATION AND VERIFICATION

The numerical solution of Eqs(4), (5) and (6) will follow the backward fully implicit finite difference algorithm (BEFD [3]). While there certainly are a variety of numerical methods from which to choose, the BEFD has the advantage of simplicity and is easily incorporated into Richardsons convergence acceleration scheme.

A **FORTRAN** program, called **SKINATH/BEFD**, has been specifically written for the SKINATH benchmarks to follow. Once one establishes a numerical method implementation becomes relatively routine and differs little from benchmark to benchmark— but there are differences. Here, we note the specific inversion required in the BEFD solution from time step  $t_{j-1}$  to  $t_{j+1}$  (interval  $h$ )

$$y_{j+1} = [I - hA(t_{j+1}, T_{j+1})]^{-1} [y_j + hq(t_{j+1})], \quad (7)$$

since it will be different depending on the kinetics model. Following the analytical inversion of the first factor in Eq(7) [Ref. [4]], we conveniently find

$$[I - hA]^{-1} = \begin{bmatrix} 1 & & & & \\ \frac{h\beta_1/\Lambda}{1+h\lambda_1} & \frac{1}{1+h\lambda_1} & & & \\ & \frac{h\lambda_1}{1+h\lambda_1} & & & \\ & & \ddots & & \\ & & & \frac{h\lambda_m}{1+h\lambda_m} & \\ & & & & 1 \end{bmatrix}^T + \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & \frac{1}{1+h\lambda_1} & & \\ & & \ddots & \\ & & & \frac{1}{1+h\lambda_m} & 0 \\ & & & & 0 \end{bmatrix} \quad (8a)$$

at time step  $t = t_j$  and

$$\sigma = \frac{1}{1 + \frac{h\varepsilon/C_p}{h_r(t)/C_p} \Lambda} + h \sum_{i=1}^m \frac{\beta_i/\Lambda}{1+h\lambda_i} \cdot 0 \quad \frac{h_r(t)/C_p}{1 + \frac{h\varepsilon/C_p}{h_r(t)/C_p} \Lambda} \quad (8b)$$

Since the Jacobian depends upon temperature, we must seek an iterative solution to convergence between time steps. We perform this by lagging the temperature in the Jacobian and iterating to convergence of the  $N(t)$  and  $T(t)$ . Acceleration is not required since the number of iterations is usually less than 10.

Note that there are four loops for (1) the desired edits, (2) sub-edits, (3) convergence acceleration and (4) iteration. One specifies the desired time edits at input. The calculation begins with the edit loop. The edits need not be at regular intervals. The second loop introduces *nel* additional sub-intervals between each edit, which are also accelerated to convergence as the original edits. We record all edits on the plot file. Only the originally desired edits are output to the screen or table file. The third loop accelerates the reactor power and temperature for each edit through Richardsons extrapolation. This is where one continuously halves the interval  $h$  to generate a sequence of solutions accelerated via Richardsons algorithm to convergence. Convergence is based on either the original or accelerated sequence.

Within the third loop, lies the iterative fourth loop, where we perform the iteration

$$y_{j+1}^{k+1} = [I - hA(t_{j+1}, T_j^k)]^{-1} [y_{j+1}^k + hq(t_{j+1})]. \quad (9)$$

over the Jacobian  $A$ . Iterative convergence is for power and temperature only since convergence acceleration

converges all dependent variables. We are now at the center of the algorithm, where several potential failures are possible as we continue with Richardsons acceleration. If the  $k$ -iteration does not converge within 10 iterates, the flow continues to the next grid of the acceleration loop in the hope that the next grid refinement will converge. Note that for linear reactivity, the iteration loop applies at least twice to each edit. Also, on completion of the iteration, if the power is negative, the acceleration simply continues with only a mention.

If for any sub-edit, Richardsons acceleration fails, we double  $nel$ , the number of sub-interval between edits. The maximum  $nel$  is 4096, at which time the program stops, which is the only indication of failure.

The sub-edit feature enables convergence time savings. It is possible to reduce the number of grids in the acceleration by increasing  $nel$ , which in some instances will reduce the overall computational time. In addition, if plots are required, one increases  $nel$  to give higher resolution. This, of course, comes at a computational expense.

### Intuitive Benchmarks

Before we present any results, it is always best to ensure the proper operation of the benchmark algorithm and program with simple intuitively obvious benchmarks. The following two simple cases come from the SKINATH/BEFD code through input specification.

The first benchmark (Bench 1) is for no imposed or Doppler feedback ( $\rho_0 = \alpha_D \equiv 0$ ) and fission heating ( $\varepsilon = 1$ ); hence, the power remains at initial power and the temperature increases from 20°C to a steady state temperature determined by

$$T(t) \left[ \frac{T(t) - T_c}{T(t)} \right]^{1.25} = \frac{\varepsilon}{\eta} N(0), \quad (10)$$

with

$$\eta \equiv ahd^{0.75}.$$

For  $N(0)$  of 1W, bisection applied to Eq(10) gives the equilibrium temperature of  $T_{eq} = 21.592327308^\circ\text{C}$ . For the reactor kinetics parameters of Table 2 (while unnecessary), Table 3 shows that the power indeed remains constant and the equilibrium temperature confirms 9 places. Figure 1 shows the full temperature trace.

Table 2. Thermal Reactor I with  $\Lambda = 5 \times 10^{-4} \text{ s}$ .

$l$	$\beta_l$	$\lambda_l (\text{s}^{-1})$
1	0.000285	0.0127
2	0.0015975	0.0317
3	0.001410	0.115
4	0.0030525	0.311
5	0.00096	1.40
6	0.000195	3.87
$\beta = 0.0075$		

Table 3. Results for benchmark Bench 1.

$t$	$N$	$T/T_c$
0.00000000E+00	1.00000000E+00	9.262549476E-01
1.00000000E-03	1.00000000E+00	9.262551613E-01
1.00000000E-02	1.00000000E+00	9.262570841E-01
1.00000000E-01	1.00000000E+00	9.262763122E-01
1.00000000E-01	1.00000000E+00	9.262763122E-01
5.00000000E-01	1.00000000E+00	9.263617586E-01
1.00000000E+00	1.00000000E+00	9.264685319E-01
1.00000000E+01	1.00000000E+00	9.283799268E-01
1.00000000E+02	1.00000000E+00	9.457184536E-01
1.00000000E+03	1.00000000E+00	9.975975868E-01
5.00000000E+03	1.00000000E+00	9.999999985E-01
1.00000000E+05	1.00000000E+00	1.000000000E+00
1.00000000E+06	1.00000000E+00	1.000000000E+00

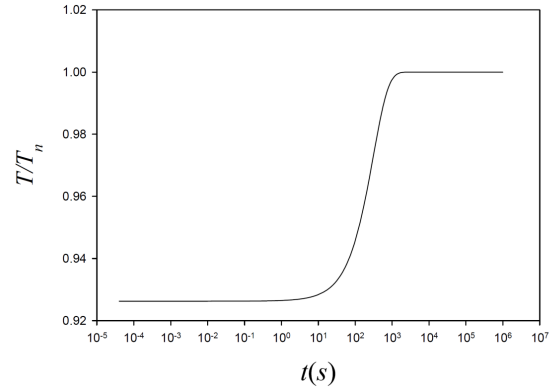


Fig. 1. Temperature trace for Bench 1.

The second benchmark (Bench 2) is for a step change in reactivity in Thermal Reactor I according to a well-known standard with no temperature feedback. Table 4 gives the results.

Table 4. Results for benchmark Bench 2.

$t$	$N_{acc}$	$N_{ori}$	$T$
0.00000000E+00	1.00000000E+00	1.00000000E+00	2.00000000E+01
1.00000000E-01	2.515766141E+00	<b>2.515773324E+00</b>	2.00000000E+01
5.00000000E-01	1.036253381E+01	<b>1.036282227E+01</b>	2.00000000E+01
1.00000000E+00	3.218354095E+01	<b>3.218488588E+01</b>	2.00000000E+01
1.00000000E+01	3.246978898E+09	<b>3.252353241E+09</b>	2.00000000E+01
1.00000000E+02	2.596484647E+89	<b>2.599842087E+89</b>	2.00000000E+01

The converged accelerated values of column 2 are in complete agreement with the CATS algorithm [5], which is arguably the best one can do. Column 3 gives the non-converged powers, with the correct digits emboldened, demonstrating the advantage of Richardsons

extrapolation. The temperature remains solid at steady state.

### SKINATH/BEFD BENCHMARKS

The original **SKINATH** application seems to have been for a drum of fissionable waste stored in an open-air waste repository with the specific parameters for this case in Table 1. The canister – reactor is assumed to be at 0.01W power at the initiation of a transient, which could be a canister compression or re-arrangement of the fissile material to cause a uniform reactivity insertion of 4.3¢. The Doppler coefficient assumes a value of  $\alpha_D = -0.306\text{¢/°C}$  and specific heat is  $C_p = 13006.194 \text{ J/kg°C}$ . The kinetics parameters are given in Table 5.

Table 5. Thermal Reactor with  $\Lambda = 5 \times 10^{-5} \text{ s}$ .

$l$	$\beta_l$	$\lambda_l(\text{s}^{-1})$
1	0.00022	0.0124
2	0.00142	0.0305
3	0.00127	0.111
4	0.00257	0.301
5	0.00075	1.14
6	0.00027	3.01
$\beta = 0.0065$		

A defect of the original **SKINATH** model is the assumption thermal equilibrium at the initiation of the transient since delayed critical is assumed. Hence, the 0.01W fission energy is not accounted for. Here, we will have the ability to assess this small error in the transient response.

Table 8 gives a portion of the original **SKINATH** output. Values in disagreement with **SKINATH/BEFD** highlighted.

Table 6. **SKINATH** results [1].

0.000000000E+00	1.000E-02	4.300E+00	2.000E+01
1.000000000E+00	1.391E-02	4.230E-02	2.000E+01
1.000000000E+01	1.033E-01	4.299E-02	2.000E+01
1.000000000E+02	7.946E-01	-2.950E-02	4.369E+01
2.500000000E+02	2.118E-03	6.388E-03	3.196E+01
5.000000000E+02	9.819E-01	-4.826E-03	3.563E+01
7.500000000E+02	4.033E+00	-1.858E-03	3.466E+01
1.000000000E+03	7.680E+00	1.379E-03	3.360E+01

The comparison indicates that the original **SKINATH** concept worked quite well showing excellent nearly 4-place agreement. Figure 2a shows the long time behavior to steady state  $N = 1.357318123\text{E}+01\text{W}$  and  $T = 3.40522876\text{E}+01\text{C}$ .

A variation on the sample case is to start the reactor transient essentially from zero power ( $10^{-20}\text{W}$ ) to evolve to its steady state as shown in Fig.2c. Now the reactor and coolant begin in equilibrium. Rather surprisingly, the

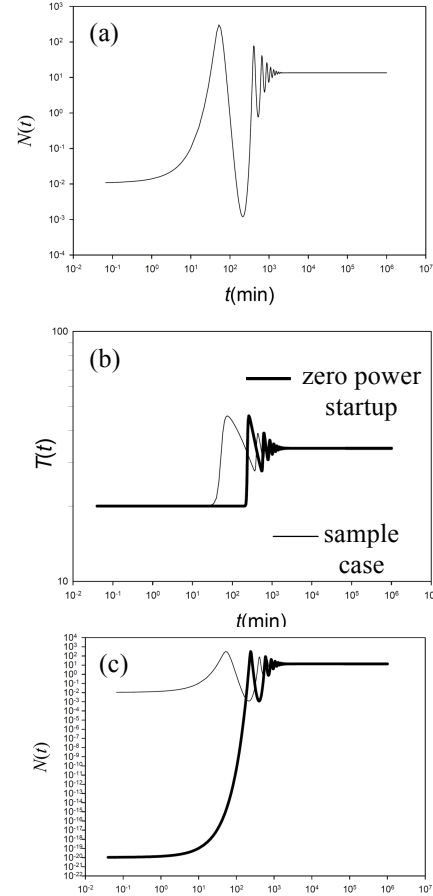


Fig. 2. a,b  $N$  and  $T$  traces sample case  
b,c  $T$  and  $N$  traces for near zero power.

numerically exact same steady state is achieved but for a different transient as shown in Figs. 2b,c.

### CONCLUSION

The **SKINATH** point kinetics algorithm serves as a starting point to develop multiphysics benchmarks for reactor transients. The **SKINATH/BEFD** numerical model, previously developed, shows how precise one can be in point kinetics calculations and this achievement should be recognized. Additional benchmarks are being developed based on the **BEFD** as well as the **CATS** algorithms and the **SKINATH** model.

### REFERENCES

1. H.L. Dodds and R.M. Westfall, ORNL/CSD/TM- 210, **SKINATH**, 1984.
2. A.C. Hindmarsh, LSODE and LSODIACM, Sigum Newsletter, V15, No. 4 ,1980.
3. B. Ganapol, *Annals of Nuclear Energy*, V62, pp 564-571, 2013.
4. J.A. da Nobrega, *Nucl. Sci.& Eng.* 46, 366-375, 1971.
5. B. Ganapol, PHYSOR 2012, Knoxville, TN, CD-ROM, American Nuclear Society, LaGrange Park, IL, 2012.