

FIRST-COLLIDED SOURCE TREATMENT FOR DISCRETE-ORDINATE RADIATION  
TRANSPORT SOLUTIONS IN RATTLESNAKE

A Thesis

by

LOGAN HUNTER HARBOUR

Submitted to the Office of Graduate and Professional Studies of  
Texas A&M University  
in partial fulfillment of the requirements for the degree of  
MASTER OF SCIENCE

Chair of Committee,	Jean Ragusa
Committee Members,	Marvin Adams
	Nancy Amato
Head of Department,	Yassin Hassan

December 2018

Major Subject: Nuclear Engineering

Copyright 2018 Logan Hunter Harbour

## ABSTRACT

Deterministic neutron transport plays a fundamental role in reactor core modeling and simulation. With the growth of computing, higher fidelity simulation is desired and the most common neutron transport scheme that produces these enhanced solutions is that of the method of discrete ordinates. However, the discrete ordinates approximation suffers from large angular discretization errors in problems with localized, small sources embedded in regions of low density or with low scattering materials. It has long been recognized that a semi-analytical treatment of the uncollided flux using ray-tracing techniques, coupled with a standard discrete ordinate treatment of the collided flux, can be a remedy for ray effects. However, current ray-tracing techniques do not support non block geometries, let alone FEM grids, and are not developed to be scalable. In this thesis, a ray-tracing approach for obtaining the uncollided flux is considered that (1) can perform in arbitrary grids, (2) supports arbitrary sources, and (3) is scalable. An implementation is then provided for the use of this uncollided flux solution as a first-collision scattering source for the purpose of ray effect treatment in the deterministic transport code Rattlesnake, a MOOSE (Multiphysics Object Oriented Simulation Environment) application.

## DEDICATION

To my grandmother, Gretchen.

## ACKNOWLEDGMENTS

I would like to thank my advisor Professor Jean C. Ragusa for introducing me to this form of treatment and providing constant support for this research and other work.

I would also like to thank my committee members Professors Marvin L. Adams and Nancy M. Amato for reviewing my work regarding this research.

Lastly, I would like to thank Mark DeHart, Derek Gaston, Sebastian Schunert, and Yaqi Wang of Idaho National Lab.

## CONTRIBUTORS AND FUNDING SOURCES

### **Contributors**

This work was supported by a thesis committee consisting of Professor Jean C. Ragusa and Professor Marin L. Adams (member) of the Department of Nuclear Engineering and Professor Nancy M. Amato of the Department of Computer Science and Engineering.

Derek Gaston of Idaho National Lab provided pre-release access to his ray-tracing module implemented in MOOSE for the purpose of completing this thesis.

All other work conducted for the thesis (or) dissertation was completed by the student independently.

### **Funding Sources**

Graduate study was supported by a graduate assistantship from NEAMS, the Nuclear Energy Advanced Modeling and Simulation Program, Grant DE-AC07-05ID14517.

## NOMENCLATURE

CFEM	Continuous Finite Element Method
DFEM	Discontinuous Finite Element Method
FEM	Finite Element Method
INL	Idaho National Laboratory
LS	Least-Square
MOOSE	Multiphysics Objected-Oriented Simulation Environment
NTE	Neutron Transport Equation
SAAF	Self-Adjoint Angular Flux
SMART	Scalable Massively Asynchronous Ray Tracing
TREAT	Transient Reactor Test Facility

## TABLE OF CONTENTS

	Page
ABSTRACT .....	ii
DEDICATION .....	iii
ACKNOWLEDGMENTS .....	iv
CONTRIBUTORS AND FUNDING SOURCES .....	v
NOMENCLATURE .....	vi
TABLE OF CONTENTS .....	vii
LIST OF FIGURES .....	ix
LIST OF TABLES.....	xi
1. INTRODUCTION.....	1
1.1 Neutron Transport.....	1
1.2 Ray Effects .....	2
1.3 Treatment of Ray Effects .....	3
1.4 Objective.....	5
2. BACKGROUND .....	7
2.1 The Transient Reactor Test Facility (TREAT) .....	7
2.1.1 Neutron Hodoscope .....	7
2.2 Neutron Transport Equation .....	11
2.2.1 Source Iteration .....	12
2.3 First-Collision Source Treatment.....	13
2.4 Ray-tracing .....	14
2.4.1 Structured, Replicated Meshes.....	15
2.4.2 Distributed Meshes .....	16
2.4.3 Unstructured Meshes .....	18
3. UNCOLLIDED FLUX COMPUTATION .....	19
3.1 Source Contribution to a Point .....	19
3.2 Point Source Contribution to a Target Element.....	21
3.3 Surface Source Contribution to a Target Element .....	21

3.4	Volumetric Source Contribution to a Target Element, $\mathcal{V} \neq \mathcal{K}$ .....	21
3.5	Volumetric Source Contribution to a Target Element, $\mathcal{V} = \mathcal{K}$ .....	22
3.6	Higher-order spatial moments .....	25
3.7	Higher-order angular moments .....	26
4.	IMPLEMENTATION .....	27
4.1	MOOSE Ray-tracing Module .....	27
4.2	Uncollided Flux .....	28
4.2.1	Ray-tracer Spawn Objects .....	29
4.2.2	Ray Generation .....	29
4.2.3	Uncollided Flux Accumulation .....	31
4.3	First-Collision Source Treatment .....	32
4.3.1	Uncollided Flux Import .....	32
4.3.2	Source Kernels .....	32
4.3.3	Action System .....	33
5.	RESULTS .....	34
5.1	Uncollided Flux Verification .....	34
5.1.1	Point Source to Volumetric Targets .....	34
5.1.2	Volumetric Source to Point Targets .....	37
5.1.3	In-cell Source Contribution .....	40
5.2	Kobayashi Benchmark .....	41
5.2.1	Pure-Absorbing Problem .....	43
5.2.2	Scattering Problem .....	45
5.3	TREAT Hodoscope .....	52
6.	CONCLUSIONS .....	56
	REFERENCES .....	58



## LIST OF FIGURES

FIGURE	Page
1.1 Examples of ray effects. ....	3
2.1 Plan and elevation views of a cross-section of TREAT (reprinted) [1]. ....	8
2.2 Example of a fuel failure tested at TREAT. ....	9
2.3 Drawing of one of the types of milled plates in the TREAT hodoscope collimator (reprinted) [2]. ....	10
2.4 Assembly of the four types of milled plates to produce the overlapping and converging configuration of the TREAT hodoscope collimator (reprinted) [2]. ....	11
2.5 An example ray trace in a 2D, replicated, brick mesh. ....	15
2.6 An set of example ray traces in a 2D, distributed, brick mesh. ....	17
3.1 A representation of the uncollided flux algorithm for an element by a source in the same element for a single point at which the uncollided flux is desired. ....	25
4.1 Algorithm for the class that generates rays for the uncollided flux computation, the <code>UncollidedFluxRayTracer</code> . ....	30
5.1 Geometry specification for the point source verification case. ....	34
5.2 Comparison of error in the integrated uncollided flux for two elements in the point source case. ....	35
5.3 Error in the integrated uncollided flux in each element for the point source case in void with 27 quadrature points per target element. ....	36
5.4 Error in the integrated uncollided flux in each element for the point source case in an absorbing medium with 27 quadrature points per target element. ....	37
5.5 Geometry specification for the volumetric source verification case. ....	38
5.6 Error in the point uncollided flux at the center of each element for the volumetric source case in void with 27 quadrature points in the source element. ....	39
5.7 Error in the point uncollided flux at the center of each element for the volumetric source case in an absorbing medium with 27 quadrature points in the source element. ....	40

5.8	Results in computing the uncollided flux at the center of the element with increasing angular quadrature for the in-cell source contribution verification case. ....	41
5.9	Kobayashi benchmark problem 3 geometry specification. ....	42
5.10	Point uncollided flux solutions produced by ray-tracing for Kobayashi problem 3 without scattering. ....	43
5.11	Scalar flux contour comparison for Kobayashi problem 3 without scattering using the ray-tracing method. ....	45
5.12	Scalar flux contour comparison for Kobayashi problem 3 with scattering, $S_{10}$ , and 2 uniform mesh refinements, and 3 uniform mesh refinements for the ray-tracing solution. The solid contours are with treatment, dotted are without. ....	46
5.13	Scalar flux contour comparison for Kobayashi problem 3 with scattering and $S_{14}$ , and 2 uniform mesh refinements, and 3 uniform mesh refinements for the ray-tracing solution. The solid contours are with treatment, dotted are without. ....	47
5.14	Scalar flux comparison for Kobayashi problem 3 with scattering. The mesh refinement listed is for the $S_N$ mesh and the ray-tracing mesh is once more refined. ....	48
5.15	Treated scalar flux comparison for Kobayashi problem 3 with scattering and varying mesh refinements. ....	51
5.16	Geometry specification for the simplified hodoscope model. ....	53

## LIST OF TABLES

TABLE		Page
5.1	Errors in the integrated uncollided flux for two elements in the point source case.....	35
5.2	Errors in the point uncollided flux at the center of two elements in the volumetric source case. ....	38
5.3	Cross-section and source definitions for the Kobayashi benchmark. ....	42
5.4	Point uncollided flux errors produced by ray-tracing for Kobayashi problem 3 without scattering and 1000 quadrature points per 10 cm × 10 cm × 10 cm source element (most refined). ....	44
5.5	Scalar flux comparison for Kobayashi problem 3 with scattering and S <sub>20</sub> .....	50
5.6	Multigroup cross sections for the hodoscope model.....	54

## 1. INTRODUCTION

### 1.1 Neutron Transport

The computer simulation of neutron transport is essential to the design and optimization of many systems involving radiation, such as nuclear reactors, radiation detectors, and medical imaging. The most accurate model of neutron transport is described by the neutron transport equation (NTE). In the NTE, each term represents a gain or a loss of neutrons in the phase space. Analytical solutions to the NTE are only available for simplified examples of physical problems. Therefore, the solution to realistic problems are commonly approximated using simulations in which the phase space (position, energy, direction, and time) is discretized and solved numerically. There are two common methods of modeling neutron transport: Monte Carlo methods and deterministic methods.

Monte Carlo methods track particles interacting stochastically through a geometry and infer properties of the system based on the average behavior of said simulated particles, and do not directly solve the NTE. They are able to treat the entire phase space (position, energy, and direction) as continuous functions and therefore can produce very accurate solutions. The solution from a Monte Carlo simulation is obtained by tallying particle histories through a region of interest. Monte Carlo methods often suffer from inefficiencies depending on the number of particle tallies required for a statistically accurate quantity of interest. For example, in a thick shielding problem, the likelihood of particles escaping the shield could be so small that the problem is not practical due to computation time without the use of advanced variance reduction techniques.

Deterministic methods solve the neutron transport equation by discretizing it into an algebraic system of equations and solving it numerically. The high dimensionality of the phase space (position, energy, and direction) poses a difficulty as it has the potential to introduce a very large number of unknowns. However, the resulting solution is an approximation over the entire phase space which is of great value to the design and optimization process. Deterministic methods also

excel in multi-physics simulations, in which iteration between individual physics (often individual codes) is required.

This research will focus exclusively on the deterministic methods of simulating neutron transport. In specific, it will focus on the method of discrete ordinates for angular discretization, finite element method (FEM) for spatial discretization, and multigroup for energy discretization.

## 1.2 Ray Effects

In the neutron transport equation, all directions of travel by neutrons are possible. In the discrete ordinates approximation, this continuous angular representation is replaced with one in which only a few *discrete* directions of travel are selected. An isotropic source which emits particles continuously in angle is represented instead as an emission along the discrete ordinates. In this discrete world, one could position a detector such that no particles reach it directly from a source that emits isotropically: the scalar and angular fluxes are zero except along lines emanating from the source in these discrete directions [3].

Also relevant is the lack of complete rotational invariance of the standard quadrature sets utilized in the method of discrete ordinates [4]. Rotational invariance implies that the functions defined on an inner product space do not change in value when arbitrary rotations are applied to its arguments, i.e., rotating the geometry will not change the approximated solution. For example, the standard level-symmetric angular quadrature set utilized in the method of discrete ordinates has only a 90 degree rotation invariance. Therefore, solutions that are highly dependent upon the relative orientation of the geometry and the global reference frame exhibit error as a result of the rotational variance in the method of discrete ordinates.

The error presented as a result of these numerical approximations manifest as unrealistic artifacts in the solution along the discrete ordinates, which are commonly termed *ray effects* [5]. An example of a problem that is prone to ray effects is one that contains sources that are small relative to the entire spatial domain (see Figure 1.1, in which the solution is expected to be roughly constant as a function of distance from the source region). Another is a problem that contains strong material discontinuities that contain materials with little absorption. The Kobayashi benchmark

[6] is a set of problems developed purposely with these characteristics in order to benchmark ray effects in  $S_N$  transport codes.

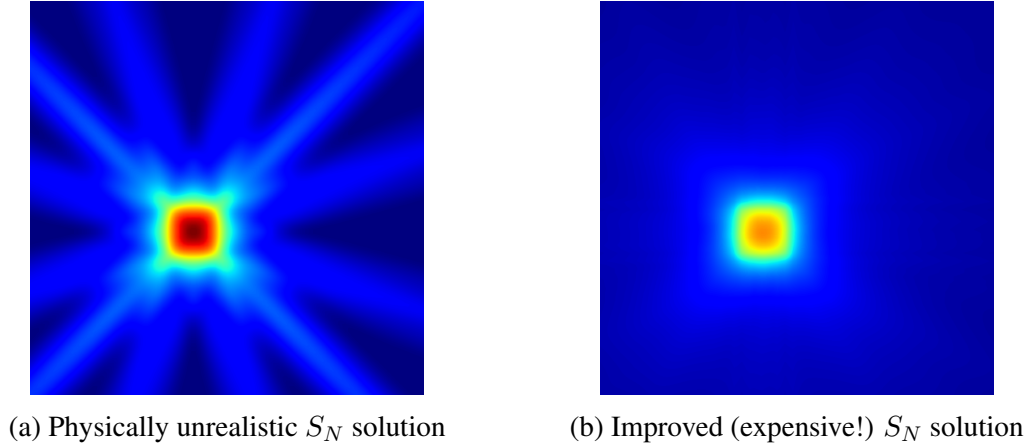


Figure 1.1: Examples of ray effects.

Even with ray effects, global estimates of leakage and absorption rates are usually made accurately by the discrete ordinates approximation [3], which can cause difficulty in distinguishing ray effect distortions from true flux variations. With these considerations, it is clear that the treatment of ray effects is a desirable task.

### 1.3 Treatment of Ray Effects

There have been many techniques proposed to mitigate ray effects [5, 7, 8, 9, 4]. The most straightforward approach is to simply to increase the number of discrete directions in the numerical approximation. However, it has shown that ray effects tend to be persistent with respect to such an increase [9]. Even if such a refinement is beneficial, it is often computationally prohibitive for realistic problems due to the significant increase in number of unknowns. Such a computational increase is even more prohibitive in multi-physics applications, in which computational efficiency is of significance in the communication between single physics codes.

Success has been shown in using more, specifically chosen, directions as a remedy to ray effects, but shows that for more difficult problems that spherical harmonic-like formulations are

necessary [5]. Another method has shown success in averaging the computed solution for various orientations in an attempt to counter the lack of rotational invariance, but is subject to computation time as multiple transport calculations are necessary [4]. Of most significance is the determination that the effectiveness of any hyperbolic ray-effect mitigation technique will necessarily be highly problem dependent [9], which is not desirable when a general mitigation and treatment strategy is the goal.

A less problem dependent mitigation technique that has shown success is the semi-analytic treatment of the total interaction operator of the transport equation. Consider that the uncollided flux (the flux of neutrons that have had no collision events) is most often the component of the angular flux that is the least isotropic and the most orientation dependent. This is first due to the fact that the uncollided flux contribution by localized sources is nonzero only in discrete cones of direction. In addition, the action of scattering smooths the solution in angle as particles scatter, because the angular redistribution operator is described by an integral in angle. Therefore, the treatment of ray effects in the uncollided flux is most desirable, then the first-collided, the second-collided, etc. This treatment is completed in practice by obtaining through other (and more accurate) means the  $i^{\text{th}}$  scattered component of the angular flux, and utilizing said component as the source of the  $(i + 1)^{\text{th}}$  scattered particles. For the treatment of the uncollided flux, this involves obtaining the uncollided flux and utilizing it as the source for the first-collided particles, commonly termed the first-collision source treatment.

The most recent implementations of first-collision source treatment utilize ray-tracing techniques to obtain the uncollided flux component of the solution. Ray-tracing in the context of this work is the process of accumulating material properties along the path from one point in the domain to another, and is able to compute the uncollided flux contribution from a point source to another point in the domain exactly. By extending this process with numerical approximation, ray-tracing is capable of producing an accurate computation of the uncollided flux due to a variety of source types (point, surface, and volumetric). The accurate computation of the uncollided flux also has benefits beyond the standard first-collision source treatment in problems where only

the uncollided flux is desired. However, standard ray-tracing techniques can be embarrassingly parallel and require special treatment to scale to the supercomputing of today.

An example of transport codes that have implemented uncollided flux algorithms for the purpose of first-collision source treatments are ATILLA [10], AETIUS [11], PARTISN [12], Denovo [13], GRTUNCL3D [14], and FNSUNCL3 [15]. Several of these examples are implemented only in serial or are only scalable to a few ranks [10, 12]. All of the above have the capability to utilize ray-tracing to obtain the uncollided flux, but some only have support for point sources [13]. Volumetric source support is important for the purpose of modeling physically realistic problems. In addition, the scalability of the ray-tracing algorithms are not discussed. When utilizing distributed meshes, there is great concern for the scalability when localized sources are present due to excessive communication. Of the above, all but ATILLA [10] and AETIUS [11] utilize brick-type meshes to represent the problem geometry. This could potentially be a significant drawback to meshing problems with complex shapes and curved surfaces, as are prevalent in nuclear reactor analysis.

## **1.4 Objective**

The initial motivation for this work was inspired by the simulation of TREAT, the Transient Reactor Test Facility at INL. The simulation of a specific component of TREAT, the neutron hodoscope, is particularly prone to ray effects and not possible to simulate using current methods. Upon further investigation of the treatment of ray effects, it was realized that the benefits of an uncollided flux treatment extend far past the complete simulation of the TREAT neutron hodoscope.

While previous work has shown significant progress in utilizing ray-tracing for the purpose of first-collision source treatment, there is still significant room for improvement. The primary goal of this research is to improve on current methods of first-collision source treatment through the implementation of such a treatment in the Idaho National Lab (INL) radiation transport code Rattlesnake. The desired improvements include:

- Support for arbitrary sources



- Finite element method (FEM) capability
- Unstructured mesh support
- Ray-tracing scalability

## 2. BACKGROUND

### 2.1 The Transient Reactor Test Facility (TREAT)

The initial motivation to perform this research was the simulation of the Transient Reactor Test Facility (TREAT) at Idaho National Laboratory (INL). Reactors that specialize in transient testing have been used to test materials to neutron pulses that can simulate conditions ranging from mild upsets to severe reactor accidents. TREAT was specifically built to conduct said tests and operated from 1959 to 1994 and was restarted recently in 2017. The reactor was constructed to test fast reactor fuels, but has also been used for light water reactor fuel testing as well as other exotic special purpose fuels. [16, 17] Such testing involves placing fuel or material into the TREAT core and subjecting it to short bursts of intense, high-power neutron radiation. After the experiment is completed, the fuel or material is analyzed and the information used to guide the development and improvement of advanced nuclear fuel designs, and to validate computer models of fuel and core behavior required for U.S. Nuclear Regulatory Commission (NRC) evaluation of nuclear power reactor design and safety evaluations [17].

The growth of supercomputing has enabled the development of codes that support high-fidelity multi-physics, which were not available during the original operational period of TREAT. With the restart of TREAT, new experimental data from transient tests is becoming available to validate said multi-physics codes. Early results from TREAT simulation have also led to the desire to improve certain methods that pertain significantly to TREAT modeling. In addition, modeling and simulation of TREAT has the potential to reduce the testing needed for experiments, increasing operational efficiency and improving fidelity in experiment design.

#### 2.1.1 Neutron Hodoscope

Early in TREAT operations, a fast neutron hodoscope was added to provide a real-time imaging technology. A hodoscope is an instrument used to detect particles emitted from a source and determine their trajectories. Hodoscopes are generally constructed of several segments of filter,

collimators and detectors; the combination of these segments is then used to infer where the particle passed through hodoscope by the location at which the particle is detected. A fast-neutron hodoscope is used to detect fast neutrons emitted as a result of fissions in a nuclear fuel sample. The system was developed to provide a imaging system to view fuel motion and failure during rapid power excursions, albeit with a limited degree of spatial resolution. The hodoscope relies on fast neutrons born in the test sample(s) to travel significant distances (approximately 3 m) through air to reach the detector pairs. The hodoscope configuration is illustrated in Figure 2.1 and an example of such a detector response and associated fuel failure is in Figure 2.2.

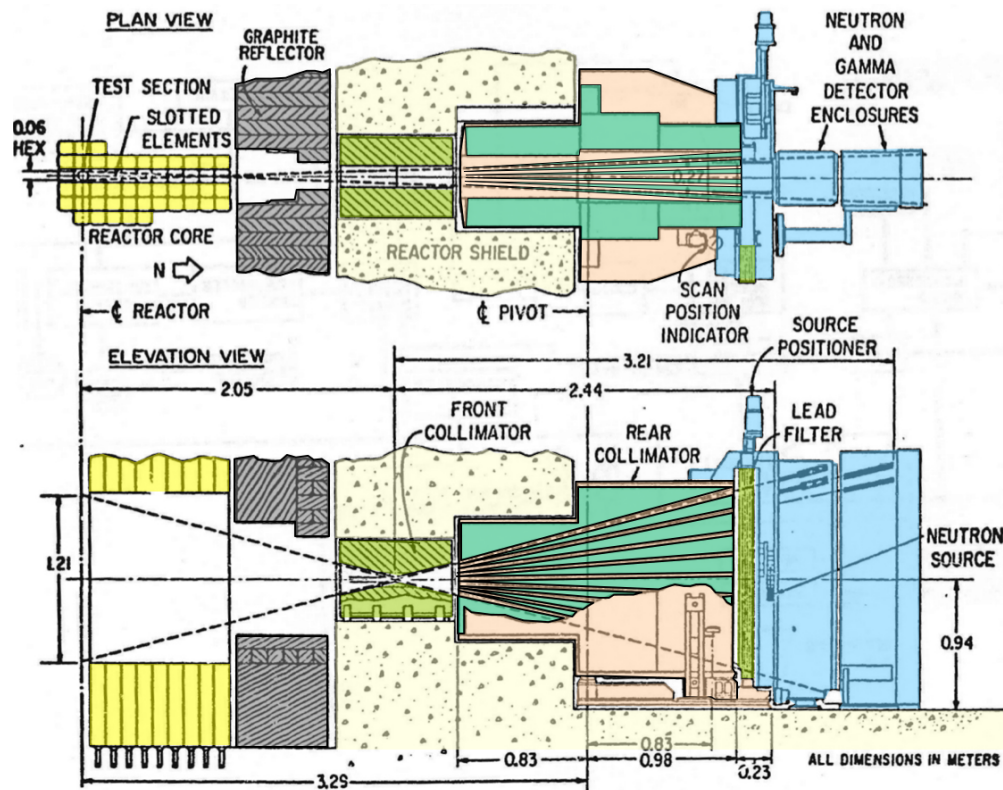
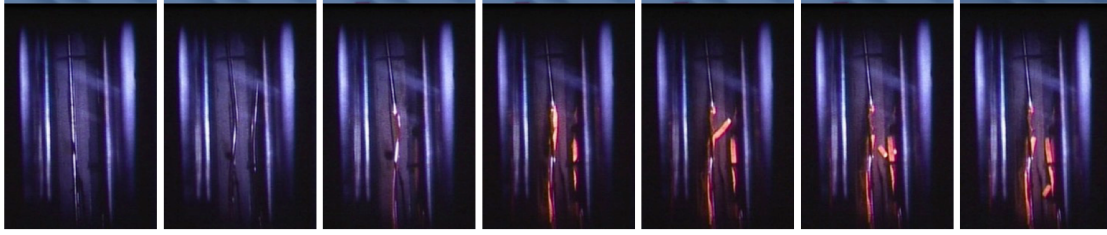
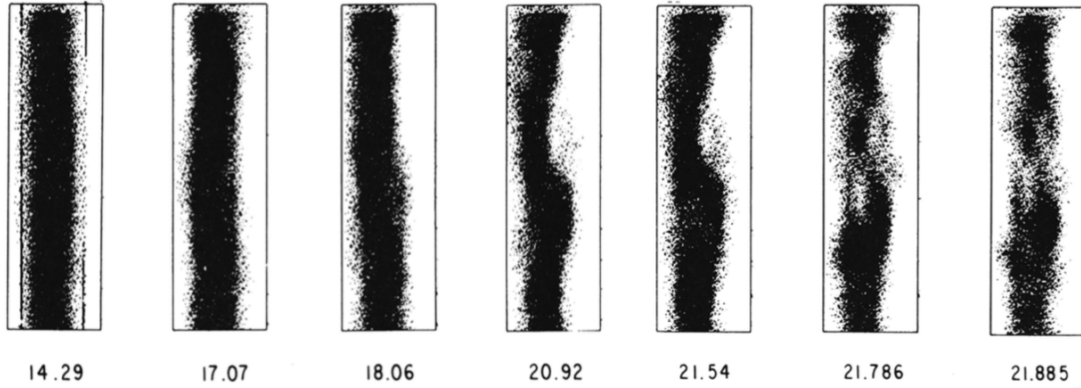


Figure 2.1: Plan and elevation views of a cross-section of TREAT (reprinted) [1].



(a) Pictured fuel failure (reprinted) [1].



(b) Intensity modulated results from the hodoscope detectors (reprinted) [2].

Figure 2.2: Example of a fuel failure tested at TREAT.

The hodoscope uses collimators and two sets of detectors located outside the reactor core for fast neutron measurements [18]. The slots within the collimator have a surface area of  $0.85 \text{ cm}^2$  at the detector interface, which at the 3 m from the test sample is a fractional solid angle of roughly  $7 \times 10^{-7} \text{ sr}$ . The collimator is composed of 30 plates of steel containing 334 milled slots, each slot focused at a fixed location on a rectangular plane intersecting the test assembly [2]. An example of a plate that composes part of the collimator follows in Figure 2.3 and the collimator plate configuration follows in Figure 2.4.

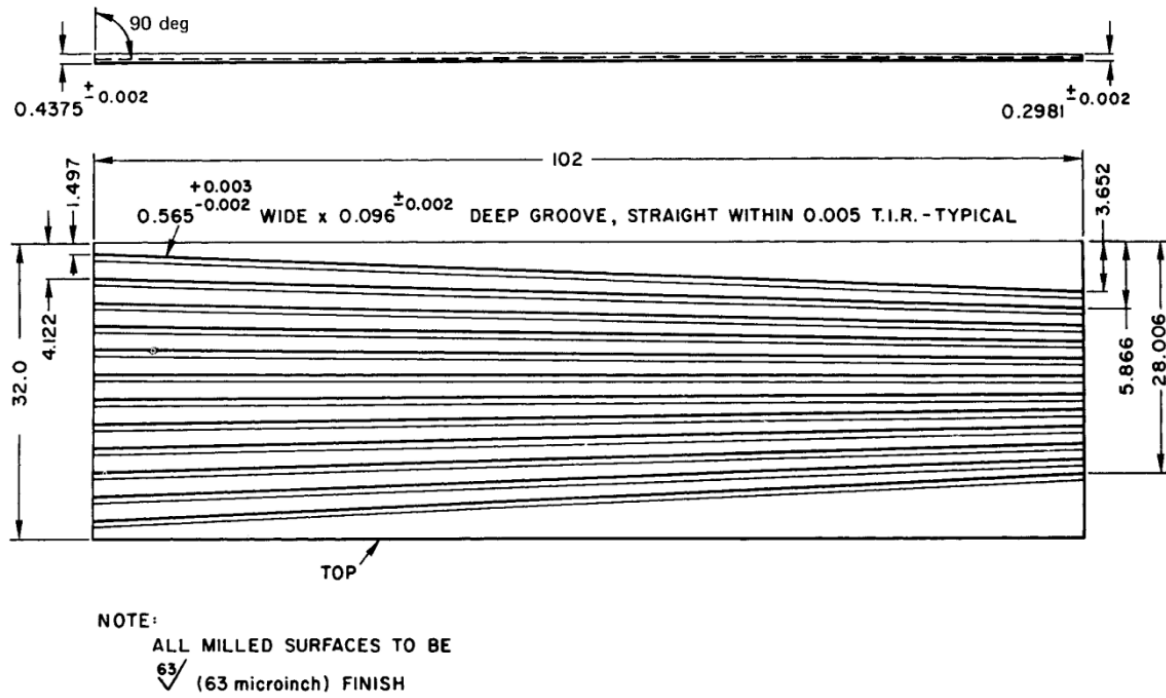


Figure 2.3: Drawing of one of the types of milled plates in the TREAT hodoscope collimator (repinted) [2].

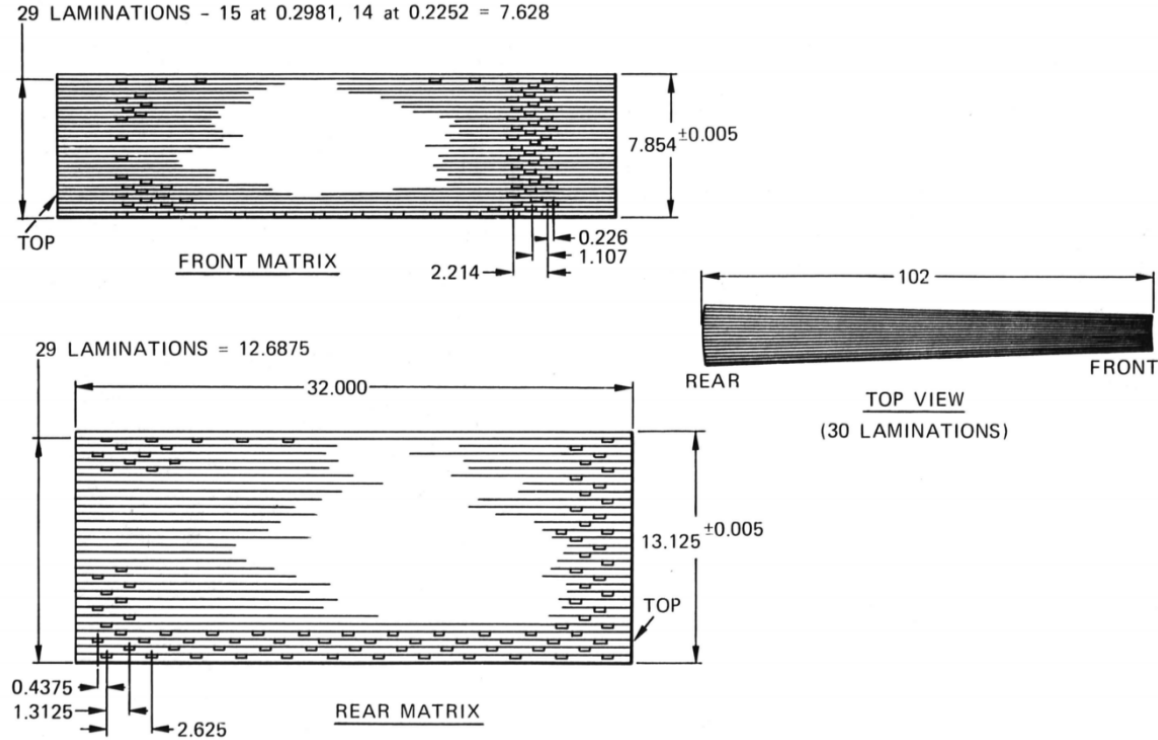


Figure 2.4: Assembly of the four types of milled plates to produce the overlapping and converging configuration of the TREAT hodoscope collimator (reprinted) [2].

Neutrons stream from the test sample in TREAT through the collimator and to the fast neutron detectors. This neutron streaming, which occurs when neutrons travel over long distances in optically thin regions that are surrounded by optically thick regions, is prone to ray effects. The potential ray effects that would occur in the simulation of the TREAT neutron hodoscope were the initial motivation for this research, in order to make possible the complete simulation of TREAT.

## 2.2 Neutron Transport Equation

For the purpose of this research, only the steady state form of the neutron transport equation is considered. The connection of the work that follows to transient simulation is rather straight forward. In addition, only the one-group form of the neutron transport equation will be considered for the purpose of derivation of this research. The majority of the method that is discussed later can be projected to multigroup with a repetitive process using the single group equations. Lastly,

it is assumed that the scattering source is isotropic.

Recall that the neutron transport equation is a balance statement that conserves neutrons, as each term represents a gain or a loss of a neutron. We shall first consider the single group, steady-state equation in its integro-differential form in a discrete domain,  $V \in \mathbb{R}^3$ , with the boundary  $\delta V^-$  as

$$\boldsymbol{\Omega} \cdot \nabla \psi(\mathbf{r}, \boldsymbol{\Omega}) + \Sigma_t(\mathbf{r})\psi(\mathbf{r}, \boldsymbol{\Omega}) = \frac{1}{4\pi} \Sigma_s(\mathbf{r})\phi(\mathbf{r}) + q(\mathbf{r}, \boldsymbol{\Omega}), \quad \mathbf{r} \in V, \quad (2.1a)$$

$$\psi(\mathbf{r}, \boldsymbol{\Omega}) = \psi^{\text{inc}}(\mathbf{r}, \boldsymbol{\Omega}), \quad \mathbf{r} \in \delta V^-, \quad (2.1b)$$

which is commonly written in operator form, as

$$\mathbf{L}\psi = \mathbf{H}\psi + Q, \quad (2.2)$$

where

$$\mathbf{L}\psi = \boldsymbol{\Omega} \cdot \nabla \psi(\mathbf{r}, \boldsymbol{\Omega}) + \Sigma_t(\mathbf{r})\psi(\mathbf{r}, \boldsymbol{\Omega}), \quad (2.3a)$$

$$\mathbf{H}\psi = \frac{1}{4\pi} \Sigma_s(\mathbf{r}) \int_{4\pi} \psi(\mathbf{r}, \boldsymbol{\Omega}) d\Omega, \quad (2.3b)$$

$$Q = q(\mathbf{r}, \boldsymbol{\Omega}). \quad (2.3c)$$

### 2.2.1 Source Iteration

The neutron transport equation is commonly solved by using the method of source iteration, in which the scattering source is lagged. Recall the steady-state, source driven problem in Equation (2.2). Express it instead as

$$\mathbf{L}\psi^{(n)} = \mathbf{H}\psi^{(n-1)} + Q, \quad (2.4)$$

which is only applicable in problems with an external source ( $Q$  and boundary fluxes). If the initial guess is  $\psi^{(0)} = 0$ , then the source iteration process has a straight forward physical interpolation. The first solution,  $\psi^{(1)} = \mathbf{L}^{-1}Q$ , is the angular flux due exclusively to the presence of external sources, that is,  $\psi^{(1)}$  is the uncollided flux. Then, in the iterative process,  $\psi^{(1)}$  is used to compute the first-collision source,  $\mathbf{H}\psi^{(1)}$ , and  $\psi^{(2)}$  will represent the flux of neutrons that have had at most

one collision event, etc.

Define  $\psi_i$  as the angular flux of particles that have had  $i$  collision events. The total angular flux is then

$$\psi = \sum_{i=0}^{\infty} \psi_i, \quad (2.5)$$

and by induction (with the initial guess  $\psi = 0$ )

$$\psi^{(n)} = \sum_{i=0}^{n-1} \psi_i.$$

The source iteration process is continued until

$$\sum_{i=0}^N \psi_i \approx \sum_{i=0}^{\infty} \psi_i,$$

i.e., the process has numerically converged (to within some tolerance) to the discretized solution. Note that the process will converge very slowly in diffusive problems because a significant number of particles will scatter many times before being absorbed or escaping the system. In the case of no scattering ( $\Sigma_s = 0$ ), no iteration is required.

### 2.3 First-Collision Source Treatment

The  $\mathbf{L}$  (total interaction) operator in Equation (2.3a) can be inverted semi-analytically with great accuracy if the uncollided flux is known. First, the angular flux is decomposed into collided,  $\psi_c$ , and uncollided,  $\psi_0$ , components. The decomposition is then expressed similarly as is done in the method of source iteration as

$$\psi = \psi_0 + \psi_c = \psi_0 + \sum_{i=1}^{\infty} \psi_i. \quad (2.6)$$

The uncollided component is first obtained by

$$\mathbf{L}\psi_0 = Q \quad \mathbf{r} \in V, \quad (2.7a)$$



$$\psi_0 = \psi^{\text{inc}}, \quad \mathbf{r} \in \delta V^- . \quad (2.7b)$$

While one could solve for this component using standard discrete-ordinate schemes, there exist other methods (namely using ray-tracing techniques as discussed next in 2.4) that can obtain this component with greater accuracy. Using the uncollided component, a first-collided scattering source  $\mathbf{H}\psi_0$  is created for the collided equation as

$$\mathbf{L}\psi_c = \mathbf{H}\psi_c + \mathbf{H}\psi_0 , \quad (2.8)$$

without an incident boundary condition. This is because the first-collided scattering source contains the contribution to the uncollided flux from both the fixed sources and the incident boundary conditions. The source iteration process is then continued as discussed in 2.2.1. Upon convergence of the collided component (denoted  $\psi_c^\infty$ ), the total angular flux is obtained by summation as

$$\psi = \psi_0 + \psi_c^\infty . \quad (2.9)$$

## 2.4 Ray-tracing

The proposed first-collision source treatment requires ray-tracing to accurately obtain the uncollided flux component in Equation (2.7). The quantity obtained by ray-tracing techniques is the optical thickness. In the context of neutron transport, the optical thickness is a measure of the attenuation of neutrons in a material from one point to another. It is a line integral of material properties along the direct path from one point to another, defined as

$$\tau(\mathbf{r}_1, \mathbf{r}_2) = \int_L \Sigma_t(\mathbf{r}) dr , \quad L = \{\mathbf{r}_1 + t\mathbf{r}_2 \mid t \in [0, 1]\} , \quad (2.10)$$

where the path is from the point  $\mathbf{r}_1$  to the point  $\mathbf{r}_2$ , and  $\Sigma_t$  is the total cross section.

The process of ray-tracing in a finite element mesh determines the intersections on each element along the path. The discussion of this process on different mesh types follows.

### 2.4.1 Structured, Replicated Meshes

A structured, replicated mesh is the simplest mesh type. Structured meshes follow a uniform pattern and therefore have many development advantages, but they can be difficult to conform to complicated shapes. A replicated mesh is one whose elements are known by all processors in a multi-processor simulation. In a 2-dimensional, structured, brick, replicated mesh, the ray-tracing process is straight forward. Consider the ray trace that follows in Figure 2.5.

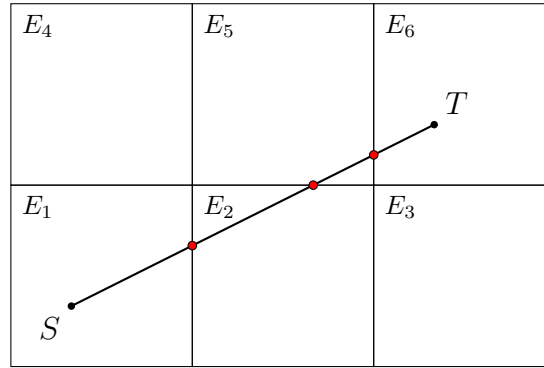


Figure 2.5: An example ray trace in a 2D, replicated, brick mesh.

Here, we are tracing from the source point defined by  $S$  and the target point defined by  $T$ . A connectivity is then defined for each element that states for each of its faces (defined by two vertices) whether it is a boundary face and if not, which element is the neighbor for said face. Starting from element 1, the intersecting face from the source point in a given direction is determined to be the face that neighbors element 2. An intersection point is found and the distance traversed through element 1 is used to tally the optical thickness traversed in said element. The process repeats until the next element is determined to be the element that the target is in, with the final result being the total optical thickness traversed from  $S$  to  $T$ .

The connectivity for said mesh is well defined and proportional to the number of elements. We will define the number of vertices in the mesh as  $V_N = 9$  and the number of elements in the mesh as  $V_E = 6$ . For each element, we must store 4 faces, the two vertices that define that face, and

the elemental neighbor for each face (or if it is a boundary face). We then see that the ray-tracing algorithm for each element simply consists of a loop over the four faces of the said element until an intersection is found, and the next element (if any) to be traversed is then easily defined.

In three-dimensional space, the ray-tracing approach is rather similar for brick meshes. Instead of looping through all of the sides of a given element, all of the faces of a given element must be checked. A connectivity is then stored for the face of each element. For non-brick meshes, the mesh connectivity and computation of intersection differ, but the general process remains the same.

### **2.4.2 Distributed Meshes**

As the previous mesh discussed is a *replicated* mesh, every processor has access to the connectivity of the entire mesh. Therefore, every processor could execute any trace because it has knowledge of the neighbors for every element in the domain. For large scale problems, it is often not reasonable to store in memory on every processor the connectivity of the entire domain due to memory constraints. Therefore, larger problems are often executed in a *distributed* manner in which the domain is partitioned into local regions that are assigned to each processor. Each processor only has knowledge of its local connectivity (the elements that belong to it), and the connectivity between its boundaries and other processors.

The distributed mesh requirement for many problems presents a difficulty in ray-tracing. Consider the following mesh in Figure 2.6, in which a localized source is in the lower left hand corner of the mesh and tracing is necessary from this source to all other targets in the domain.

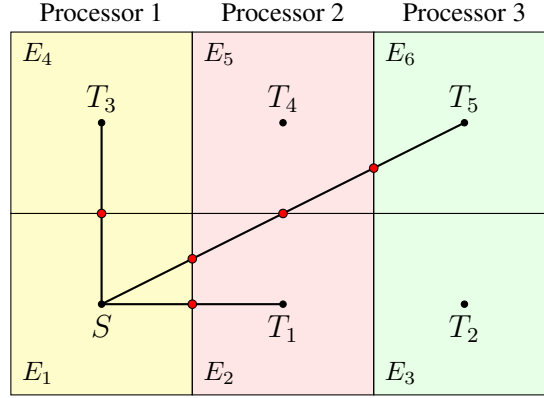


Figure 2.6: An set of example ray traces in a 2D, distributed, brick mesh.

Here, the colors represent the distributed partitions of the mesh. Processor 1 has knowledge of elements 1 and 4, processor 2 has knowledge of elements 2 and 5, etc. We want to trace from the source point to each target point, denoted  $T_i$ . We will then discuss the following three traces:

- From  $S$  to  $T_3$ : The first intersection is made between  $E_1$  and  $E_4$ , for which  $E_4$  is known locally to processor 1. The remaining trace is completed inside  $E_4$ . Therefore, the entire trace is completed on processor 1.
- From  $S$  to  $T_1$ : The first intersection is made on the right face of  $E_1$ , on the boundary of processor 1. Processor 1 communicates a ray to processor 2, and processor 2 continues the trace in  $E_2$  and end it at  $T_1$ .
- From  $S$  to  $T_5$ : The first intersection is made on the right face of  $E_1$ , on the boundary of processor 1. Processor 1 communicates a ray to processor 2, and processor 2 continues the trace through  $E_2$  and  $E_5$ . The next intersection is made on the right face of  $E_5$ , on the boundary of processor 2. Processor 2 communicates a ray to processor 3, and processor 3 continues the trace in  $E_6$  and ends it at  $T_5$ .

Communication is a significant bottleneck to large scale computation. As all of the rays must be started from processor 1 here, processors 2 and 3 will sit idle until they receive any rays. Processor

1 must also spend time *generating* rays instead of just propagating them, while processors 2 and 3 only need to propagate rays. This can lead to significant scalability issues as many processors will sit idle on large domains waiting to receive rays when they do not contain any source points. For these reasons, there is a significance to developing a ray-tracing method that is able to recursively generate, propagate, and communicate rays as efficiently as possible to minimize processor idle time. For cases in which sources are well localized in the domain, idle time is maximized if the mesh is partitioned such that not all processors contain sources.

### **2.4.3 Unstructured Meshes**

An unstructured mesh is a mesh whose elements are of irregular size and shape and do not follow a uniform pattern. This results in a mesh connectivity that is not necessarily proportional to the number of elements in the mesh. With this case, ray-tracing is more difficult. One element may have one face whose neighbors are multiple elements of different shape. Therefore, the algorithm is not as simple as determining the intersecting face and the neighboring element.

The exiting face at which a ray exits a given element is first determined. The mesh connectivity is then used to determine which element(s) are on the neighboring face, which could possibly contain faces of varying shape. Therefore, specialized algorithms are needed for each face type and must be chosen specifically depending on which elemental shapes are neighboring. For the brick case, one need only determine if a point is inside the plane of a given face. When multiple faces are considered as the entrant face to the next element, a point being in a plane is not sufficient to determine an intersection.

### 3. UNCOLLIDED FLUX COMPUTATION

In order to perform the first-collision source treatment, the total collision operator,  $\mathbf{L}$ , in Equation (2.2) is semi-analytically inverted using ray-tracing methods. That is, the uncollided component of the angular flux is determined using ray-tracing. Further methods are devised to produce the uncollided flux throughout the domain utilizing source strengths along with the optical thicknesses obtained using ray-tracing.

The implementation performed this far is capable of producing cell averaged values of the uncollided flux as a result of a single point source, an elemental surface source, and an elemental volumetric source. By superposition, the uncollided flux as a result of multiple of these sources in a single cell is simply obtained by summation of each individual contribution.

For the discussion that follows, we will focus on the contribution by a point source,  $\mathcal{P}$ , a surface source,  $\mathcal{S}$ , and a volumetric source,  $\mathcal{V}$  to a target element,  $\mathcal{K}$ . The following definitions are introduced:

$$R(\mathbf{r}_1, \mathbf{r}_2) \equiv \|\mathbf{r}_1 - \mathbf{r}_2\| ,$$

$$\Omega(\mathbf{r}_1, \mathbf{r}_2) \equiv \frac{\mathbf{r}_2 - \mathbf{r}_1}{R(\mathbf{r}_1, \mathbf{r}_2)} .$$

First, the contribution of all of these source types to a single point is discussed. Then, the contribution of these source types to the elemental target,  $\mathcal{K}$  is discussed.

#### 3.1 Source Contribution to a Point

We will consider now the uncollided flux as a result of a point source, a surface source, and a volumetric source, beginning with a point source. The uncollided flux at a point  $\mathbf{r}$  as a result of a point source at  $\mathbf{r}'$  (denoted point source  $\mathcal{P}$ ) is

$$\phi^{\mathcal{P}}(\mathbf{r}) = \frac{\exp(-\tau(\mathbf{r}', \mathbf{r}))}{R^2(\mathbf{r}', \mathbf{r})} S_{\mathcal{P}}(\Omega(\mathbf{r}', \mathbf{r})) \quad (3.1)$$

where:

$\tau(\mathbf{r}', \mathbf{r})$  = the optical thickness from  $\mathbf{r}'$  to  $\mathbf{r}$ , obtained from ray-tracing (see Equation (2.10))

$S_{\mathcal{P}}(\boldsymbol{\Omega}(\mathbf{r}', \mathbf{r}))$  = the source emission rate in n/s·str of the source in direction  $\boldsymbol{\Omega}(\mathbf{r}', \mathbf{r})$ .

The above lies at the basis of all of the derivations that follow. By performing appropriate integrals over both source and target elements, the uncollided flux can be obtained as desired.

Consider now a surface source,  $\mathcal{S}$ , and seek the uncollided flux at point  $\mathbf{r}$  as a result of  $\mathcal{S}$ . Begin with Equation (3.1) integrate over the surface, as

$$\phi^{\mathcal{S}}(\mathbf{r}) = \int_{\mathcal{S}} \frac{\exp(-\tau(\mathbf{r}', \mathbf{r}))}{R^2(\mathbf{r}', \mathbf{r})} S_{\mathcal{S}}(\mathbf{r}', \boldsymbol{\Omega}(\mathbf{r}', \mathbf{r})) d^2 r',$$

where  $S_{\mathcal{S}}(\mathbf{r}', \boldsymbol{\Omega}(\mathbf{r}', \mathbf{r}))$  is the source emission rate in n/cm<sup>2</sup>·s·str of the source at  $\mathbf{r}'$  in the direction  $\boldsymbol{\Omega}(\mathbf{r}', \mathbf{r})$ . Introduce spatial quadrature set  $\{\mathbf{r}'_{q'}, w'_{q'}\}$  on the surface source (where  $\sum_{q'} w'_{q'} =$  the area of the surface source  $\mathcal{S}$ ), as

$$\tilde{\phi}^{\mathcal{S}}(\mathbf{r}) = \sum_{q'} w'_{q'} \frac{\exp(-\tau(\mathbf{r}'_{q'}, \mathbf{r}))}{R^2(\mathbf{r}'_{q'}, \mathbf{r})} S_{\mathcal{S}}(\mathbf{r}'_{q'}, \boldsymbol{\Omega}(\mathbf{r}'_{q'}, \mathbf{r})). \quad (3.2)$$

Last, consider a volumetric source,  $\mathcal{V}$ , and seek the uncollided flux at point  $\mathbf{r}$  as a result of  $\mathcal{V}$ . Again, begin with Equation (3.1) and instead integrate over the volume, as

$$\phi^{\mathcal{V}}(\mathbf{r}) = \int_{\mathcal{V}} \frac{\exp(-\tau(\mathbf{r}', \mathbf{r}))}{R^2(\mathbf{r}', \mathbf{r})} S_{\mathcal{V}}(\mathbf{r}', \boldsymbol{\Omega}(\mathbf{r}', \mathbf{r})) d^3 r',$$

where  $S_{\mathcal{V}}(\mathbf{r}', \boldsymbol{\Omega}(\mathbf{r}', \mathbf{r}))$  is the source emission rate in n/cm<sup>3</sup>·s·str of the source at  $\mathbf{r}'$  in the direction  $\boldsymbol{\Omega}(\mathbf{r}', \mathbf{r})$ . Introduce spatial quadrature set  $\{\mathbf{r}'_{q'}, w'_{q'}\}$  on the volumetric source (where  $\sum_q w'_q =$  the volume of the volumetric source  $\mathcal{V}$ ), as

$$\tilde{\phi}^{\mathcal{V}}(\mathbf{r}) = \sum_{q'} w'_{q'} \frac{\exp(-\tau(\mathbf{r}'_{q'}, \mathbf{r}))}{R^2(\mathbf{r}'_{q'}, \mathbf{r})} S_{\mathcal{V}}(\mathbf{r}'_{q'}, \boldsymbol{\Omega}(\mathbf{r}'_{q'}, \mathbf{r})). \quad (3.3)$$

### 3.2 Point Source Contribution to a Target Element

Begin with Equation (3.1) and seek the average uncollided flux in the target element  $\mathcal{K}$  as a result of the source point  $\mathcal{P}$  at  $\mathbf{r}'$  by integrating and dividing by the cell volume,  $V_{\mathcal{K}}$ , as

$$\bar{\phi}_{\mathcal{K}}^{\mathcal{P}} = \frac{1}{V_{\mathcal{K}}} \int_{\mathcal{K}} \frac{\exp(-\tau(\mathbf{r}', \mathbf{r}))}{R^2(\mathbf{r}', \mathbf{r})} S_{\mathcal{P}}(\mathbf{\Omega}(\mathbf{r}', \mathbf{r})) d^3r.$$

Introduce spatial quadrature set  $\{\mathbf{r}_q, w_q\}$  in  $\mathcal{K}$  (where  $\sum_q w_q = V_{\mathcal{K}}$ ), as

$$\bar{\phi}_{\mathcal{K}}^{\mathcal{P}} \approx \frac{1}{V_{\mathcal{K}}} \sum_q w_q \frac{\exp(-\tau(\mathbf{r}', \mathbf{r}_q))}{R^2(\mathbf{r}', \mathbf{r}_q)} S_{\mathcal{P}}(\mathbf{\Omega}(\mathbf{r}', \mathbf{r}_q)). \quad (3.4)$$

### 3.3 Surface Source Contribution to a Target Element

Begin with Equation (3.2) and seek the average uncollided flux in the target element  $\mathcal{K}$  as a result of the surface source  $\mathcal{S}$  integrating and dividing by the cell volume,  $V_{\mathcal{K}}$ , as

$$\bar{\phi}_{\mathcal{K}}^{\mathcal{S}} \approx \frac{1}{V_{\mathcal{K}}} \int_{\mathcal{K}} \sum_{q'} w'_{q'} \frac{\exp(-\tau(\mathbf{r}'_{q'}, \mathbf{r}))}{R^2(\mathbf{r}'_{q'}, \mathbf{r})} S_{\mathcal{S}}(\mathbf{r}'_{q'}, \mathbf{\Omega}(\mathbf{r}'_{q'}, \mathbf{r})) d^3r.$$

Again introduce spatial quadrature set  $\{\mathbf{r}_q, w_q\}$  in  $\mathcal{K}$  (where  $\sum_q w_q = V_{\mathcal{K}}$ ), as

$$\bar{\phi}_{\mathcal{K}}^{\mathcal{S}} \approx \frac{1}{V_{\mathcal{K}}} \sum_q w_q \sum_{q'} w'_{q'} \frac{\exp(-\tau(\mathbf{r}'_{q'}, \mathbf{r}_q))}{R^2(\mathbf{r}'_{q'}, \mathbf{r}_q)} S_{\mathcal{S}}(\mathbf{r}'_{q'}, \mathbf{\Omega}(\mathbf{r}'_{q'}, \mathbf{r}_q)). \quad (3.5)$$

### 3.4 Volumetric Source Contribution to a Target Element, $\mathcal{V} \neq \mathcal{K}$

Begin with Equation (3.3) and seek the average uncollided flux in the target element  $\mathcal{K}$  as a result of the volumetric source  $\mathcal{V} \neq \mathcal{K}$  integrating and dividing by the cell volume,  $V_{\mathcal{K}}$ , as

$$\bar{\phi}_{\mathcal{K}}^{\mathcal{V}} \approx \frac{1}{V_{\mathcal{K}}} \int_{\mathcal{V}} \sum_{q'} w'_{q'} \frac{\exp(-\tau(\mathbf{r}'_{q'}, \mathbf{r}))}{R^2(\mathbf{r}'_{q'}, \mathbf{r})} S_{\mathcal{V}}(\mathbf{r}'_{q'}, \mathbf{\Omega}(\mathbf{r}'_{q'}, \mathbf{r})) d^3r.$$



Again introduce spatial quadrature set  $\{\mathbf{r}_q, w_q\}$  in  $\mathcal{K}$  (where  $\sum_q w_q = V_{\mathcal{K}}$ ), as

$$\bar{\phi}_{\mathcal{K}}^{\mathcal{V}} \approx \frac{1}{V_{\mathcal{K}}} \sum_q w_q \sum_{q'} w_{q'} \frac{\exp(-\tau(\mathbf{r}'_{q'}, \mathbf{r}_q))}{R^2(\mathbf{r}'_{q'}, \mathbf{r}_q)} S_{\mathcal{V}}(\mathbf{r}'_{q'}, \boldsymbol{\Omega}(\mathbf{r}'_{q'}, \mathbf{r}_q)). \quad (3.6)$$

### 3.5 Volumetric Source Contribution to a Target Element, $\mathcal{V} = \mathcal{K}$

With the above in 3.4, the uncollided angular flux was computed using Green's function which contains a  $1/R^2$  term. Computing the uncollided flux inside the source term would be problematic, considering

$$\lim_{R \rightarrow 0} \frac{1}{R^2} = \infty.$$

It could potentially be inaccurate in the neighboring cells of the source cells as well. To combat this, a change of variable is introduced to eliminate the  $1/R^2$  term. First, the cell-averaged scalar flux in a given cell  $\mathcal{K}$  due to a volumetric source in the same cell is

$$\bar{\phi}_{\mathcal{K}}^{\mathcal{K}} = \frac{1}{V_{\mathcal{K}}} \int_{\mathcal{K}} d^3r \int_{\mathcal{V}} d^3r' \frac{\exp(-\tau(\mathbf{r}', \mathbf{r}))}{R^2(\mathbf{r}', \mathbf{r})} S_{\mathcal{K}}(\mathbf{r}', \boldsymbol{\Omega}(\mathbf{r}', \mathbf{r})),$$

When  $\mathcal{V} = \mathcal{K}$ , the  $1/R^2$  term is eliminated by introducing the change of variable

$$\mathbf{r}' = \mathbf{r} - \mathbf{R} = \mathbf{r} - R\boldsymbol{\Omega}, \quad d\mathbf{r}' = d\mathbf{r} = R^2 dR d\boldsymbol{\Omega},$$

to obtain

$$\bar{\phi}_{\mathcal{K}}^{\mathcal{K}} = \frac{1}{V_{\mathcal{K}}} \int_{\mathcal{K}} d^3r \int_{4\pi} d\boldsymbol{\Omega} \int_{R_-(\mathbf{r}, \boldsymbol{\Omega})}^{R_+(\mathbf{r}, \boldsymbol{\Omega})} dR \exp(-\tau(\mathbf{r}', \mathbf{r})) S_{\mathcal{V}}(\mathbf{r}', \boldsymbol{\Omega}(\mathbf{r}', \mathbf{r})), \quad (3.7)$$

where  $R_{\pm}$  is the distance from  $\mathbf{r}$  to the first ( $R_-$ ) or second  $R_+$  intersections of the back trajectory at  $\mathbf{r}$  in the direction  $\boldsymbol{\Omega}$  with the surface of  $\mathcal{K}$  (assuming the later is convex). A quadrature is then required for  $\boldsymbol{\Omega}$  (for example, the one used for the  $S_N$  discrete ordinate approximation) and a quadrature for  $\mathbf{r}$  in  $\mathcal{K}$

Noting that  $R_-(\mathbf{r}, \boldsymbol{\Omega}) = 0$  and  $\mathbf{r}' = \mathbf{r} - R\boldsymbol{\Omega}$  when we are in the same cell, we then have the exact cell-averaged uncollided scalar flux (as the result of source cell  $\mathcal{V} =$  target cell  $\mathcal{K}$ , zeroth

spatial and angular moments) as

$$\bar{\phi}_{\mathcal{K}}^{\mathcal{K}} = \frac{1}{V_{\mathcal{K}}} \int_{\mathcal{K}} d^3r \int_{4\pi} d\Omega \int_0^{R_+(\mathbf{r}, \boldsymbol{\Omega})} dR \exp(-\tau(\mathbf{r}, \mathbf{r} - R\boldsymbol{\Omega})) S_{\mathcal{V}}(\mathbf{r} - R\boldsymbol{\Omega}, \boldsymbol{\Omega}) . \quad (3.8)$$

If the volumetric source is spatially constant in  $\mathcal{K}$ , the integral over the back-trajectory in Equation (3.8) can be evaluated analytically as (noting that  $S_{\mathcal{V}}$  comes out said integral)

$$\begin{aligned} \int_0^{R_+(\mathbf{r}, \boldsymbol{\Omega})} dR \exp(-\tau(\mathbf{r}, \mathbf{r} - R\boldsymbol{\Omega})) &= \frac{1}{\Sigma_t} (1 - e^{-\Sigma_t R_+(\mathbf{r}, \boldsymbol{\Omega})}) , \\ &= R_+(\mathbf{r}, \boldsymbol{\Omega}) \left( 1 + \sum_{n=1}^{\infty} \frac{(-1)^n}{(n+1)!} (\Sigma_t R_+(\mathbf{r}, \boldsymbol{\Omega}))^n \right) . \end{aligned}$$

Evaluating the above can be split up into three cases: when  $\Sigma_t = 0$ , when  $0 < \Sigma_t R_+(\mathbf{r}, \boldsymbol{\Omega}) < \epsilon$  (where  $\epsilon$  is sufficiently small  $\approx 1 \times 10^{-3}$ ), and when  $\Sigma_t R_+(\mathbf{r}, \boldsymbol{\Omega}) \geq \epsilon$ . We want to separate the case of  $0 < \Sigma_t R_+(\mathbf{r}, \boldsymbol{\Omega}) < \epsilon$  due to issues with machine subtraction of numbers that are close to one another and division by small numbers. With these cases in mind, we have

$$\int_0^{R_+(\mathbf{r}, \boldsymbol{\Omega})} dR \exp(-\tau(\mathbf{r}, \mathbf{r} - R\boldsymbol{\Omega})) = \begin{cases} R_+(\mathbf{r}, \boldsymbol{\Omega}) , & \Sigma_t = 0 , \\ R_+(\mathbf{r}, \boldsymbol{\Omega}) \left( 1 + \sum_{n=1}^{\infty} \frac{(-1)^n}{(n+1)!} (\Sigma_t R_+(\mathbf{r}, \boldsymbol{\Omega}))^n \right) , & 0 < \Sigma_t R_+(\mathbf{r}, \boldsymbol{\Omega}) < \epsilon , \\ \frac{1}{\Sigma_t} (1 - e^{-\Sigma_t R_+(\mathbf{r}, \boldsymbol{\Omega})}) , & \Sigma_t R_+(\mathbf{r}, \boldsymbol{\Omega}) \geq \epsilon \end{cases} . \quad (3.9)$$

If the volumetric source is not spatially constant in  $\mathcal{K}$ , we can introduce the 1D quadrature set  $\{R_q, w_q\}$  with  $\sum_q R_q = R_+(\mathbf{r}, \boldsymbol{\Omega})$  along the ray direction  $\boldsymbol{\Omega}$  as

$$\begin{aligned} \int_0^{R_+(\mathbf{r}, \boldsymbol{\Omega})} dR \exp(-\tau(\mathbf{r}, \mathbf{r} - R\boldsymbol{\Omega})) S_{\mathcal{V}}(\mathbf{r} - R\boldsymbol{\Omega}, \boldsymbol{\Omega}) &= \\ \sum_q w_q \exp(-\tau(\mathbf{r}, \mathbf{r} - R_q \boldsymbol{\Omega})) S_{\mathcal{K}}(\mathbf{r} - R_q \boldsymbol{\Omega}, \boldsymbol{\Omega}) . \end{aligned} \quad (3.10)$$

Introduce angular quadrature set  $\{\boldsymbol{\Omega}_d, \omega_d\}$  and 3D quadrature set  $\{\mathbf{r}_j, w_j\}$  with  $\sum_j w_j = V_{\mathcal{K}}$  and

define  $\mathbf{r}_{j,q} = \mathbf{r}_j - R_q \boldsymbol{\Omega}_d$ . We can then approximate Equation (3.8) with the cases described in Equations (3.9) and (3.10) above as

$$\begin{aligned} \tilde{\phi}_{\mathcal{K}}^{\mathcal{K}} &= \frac{1}{V_{\mathcal{K}}} \sum_j w_j \sum_d \omega_d \\ &\begin{cases} \sum_q w_q \exp(-\tau(\mathbf{r}_j, \mathbf{r}_q)) S_{\mathcal{L}}(\mathbf{r}_{j,q}, \boldsymbol{\Omega}_d), & \text{non const. } S_{\mathcal{V}}, \\ S_{\mathcal{K}}(\boldsymbol{\Omega}_d) R_+(\mathbf{r}_j, \boldsymbol{\Omega}_d), & \text{const. } S_{\mathcal{V}}, \Sigma_t = 0, \\ S_{\mathcal{K}}(\boldsymbol{\Omega}_d) R_+(\mathbf{r}_j, \boldsymbol{\Omega}_d) \left(1 + \sum_{n=1}^{\infty} \frac{(-1)^n}{(n+1)!} (\Sigma_t R_+(\mathbf{r}_j, \boldsymbol{\Omega}_d))^n\right), & \text{const. } S_{\mathcal{K}}, 0 < \Sigma_t R_+(\mathbf{r}_j, \boldsymbol{\Omega}_d) < \epsilon, \\ S_{\mathcal{K}}(\boldsymbol{\Omega}_d) \frac{1}{\Sigma_t} (1 - e^{-\Sigma_t R_+(\mathbf{r}_j, \boldsymbol{\Omega}_d)}), & \text{const. } S_{\mathcal{V}}, \Sigma_t R_+(\mathbf{r}_j, \boldsymbol{\Omega}_d) \geq \epsilon \end{cases} \end{aligned} \quad (3.11)$$

where the constant identifier for  $S_{\mathcal{V}}$  describes whether or not it is spatially constant. The process in Equation (3.11) when 1D quadrature is required (volumetric sources are not spatially constant in each cell) is visualized below in Figure 3.1. When 1D quadrature is not required, the integration along each subdivision created by the angular quadrature is performed exactly.

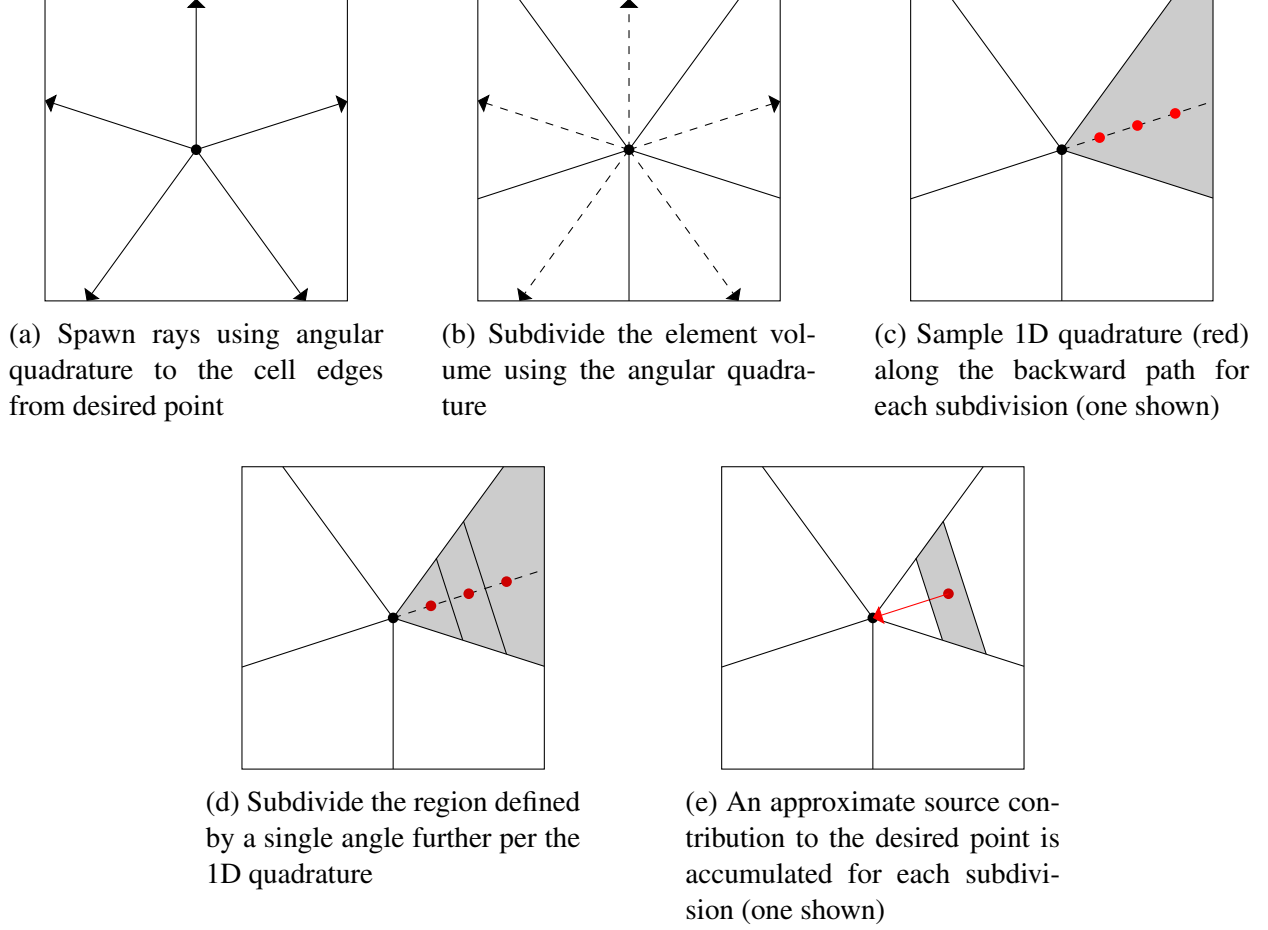


Figure 3.1: A representation of the uncollided flux algorithm for an element by a source in the same element for a single point at which the uncollided flux is desired.

### 3.6 Higher-order spatial moments

While this research does not implement higher-order spatial moments, the implementation is still discussed as it is considered in future work. With the addition of a basis function  $b(\mathbf{r})$  in the target cell  $\mathcal{K}$ , the spatial moment of the scalar flux in target cell  $\mathcal{K}$  due to source cell  $\mathcal{V}$  is

$$\phi_{\mathcal{V},i}^{\mathcal{K}} = \int_{\mathcal{K}} d^3r \, b_i(\mathbf{r}) \int_{\mathcal{V}} d^3r' \frac{\exp(-\tau(\mathbf{r}', \mathbf{r}))}{R^2(\mathbf{r}', \mathbf{r})} S_{\mathcal{V}}(\mathbf{r}', \boldsymbol{\Omega}(\mathbf{r}', \mathbf{r})),$$

Introduce quadrature set  $\{\mathbf{r}_q, w_q\}$  in  $\mathcal{K}$  and set  $\{\mathbf{r}'_{q'}, w'_{q'}\}$  in  $\mathcal{V}$ , as

$$\phi_{\mathcal{V},i}^{\mathcal{K}} = \sum_q w_q b_i(\mathbf{r}_q) \sum_{q'} w_{q'} \frac{\exp(-\tau(\mathbf{r}'_{q'}, \mathbf{r}_q))}{R^2(\mathbf{r}'_{q'}, \mathbf{r}_q)} S_{\mathcal{V}}(\mathbf{r}'_{q'}, \boldsymbol{\Omega}(\mathbf{r}'_{q'}, \mathbf{r}_q)), \quad (3.12)$$

The same method is applied to the target = source cell ( $\mathcal{V} = \mathcal{K}$ ) method as

$$\phi_{\mathcal{K},i}^{\mathcal{K}} = \int_{\mathcal{K}} d^3r b_i(\mathbf{r}) \int_{4\pi} d\Omega \int_0^{R_+(\mathbf{r}, \boldsymbol{\Omega})} dR \exp(-\tau(\mathbf{r}, \mathbf{r} - R\boldsymbol{\Omega})) S(\mathbf{r} - R\boldsymbol{\Omega}, \boldsymbol{\Omega}),$$

and with the same quadratures used in Equation 3.11

$$\phi_{\mathcal{K},i}^{\mathcal{K}} = \sum_j w_j b_i(\mathbf{r}_j) \sum_d \omega_d \sum_q w_q \exp(-\tau(\mathbf{r}_j, \mathbf{r}_j - R_q \boldsymbol{\Omega}_d)) S_{\mathcal{V}}(\mathbf{r}_j - R_q \boldsymbol{\Omega}_d, \boldsymbol{\Omega}_d). \quad (3.13)$$

### 3.7 Higher-order angular moments

Again, this research does not implement higher-order moments, but the implementation of higher-order angular moments is still discussed. When scattering is anisotropic, one needs a higher angular moment than the scalar flux. Consider the real-valued spherical harmonic functions,  $A_k(\boldsymbol{\Omega})$ . The angular moment of the scalar flux in target cell  $K$  due to source cell  $K'$  is

$$\begin{aligned} \phi_{\mathcal{K},k}^{\mathcal{V}}(\mathbf{r}) &= \int_{\mathcal{V}} d^3r' \int_{4\pi} d\Omega A_k(\boldsymbol{\Omega}) \frac{\exp(-\tau(\mathbf{r}', \mathbf{r}))}{R^2(\mathbf{r}', \mathbf{r})} S_{\mathcal{V}}(\mathbf{r}', \boldsymbol{\Omega}(\mathbf{r}', \mathbf{r})) \delta_2(\boldsymbol{\Omega} \cdot \boldsymbol{\Omega}(\mathbf{r}', \mathbf{r})), \quad \mathbf{r} \in \mathcal{K} \\ &= \int_{\mathcal{V}} d^3r' A_k(\boldsymbol{\Omega}(\mathbf{r}', \mathbf{r})) \frac{\exp(-\tau(\mathbf{r}', \mathbf{r}))}{R^2(\mathbf{r}', \mathbf{r})} S_{\mathcal{V}}(\mathbf{r}', R^2(\mathbf{r}', \mathbf{r})), \quad \mathbf{r} \in \mathcal{K}. \end{aligned}$$

Introduce quadrature  $\{\mathbf{r}'_{q'}, w'_{q'}\}$  in  $\mathcal{V}$  as

$$\phi_{\mathcal{K},k}^{\mathcal{V}}(\mathbf{r}) = \sum_{q'} w_{q'} A_k(\boldsymbol{\Omega}(\mathbf{r}'_{q'}, \mathbf{r})) \frac{\exp(-\tau(\mathbf{r}'_{q'}, \mathbf{r}))}{R^2(\mathbf{r}'_{q'}, \mathbf{r})} S_{\mathcal{V}}(\mathbf{r}'_{q'}, \boldsymbol{\Omega}(\mathbf{r}'_{q'}, \mathbf{r})), \quad \mathbf{r} \in \mathcal{K}. \quad (3.14)$$

## 4. IMPLEMENTATION

At the base of the implementation that follows is the Multiphysics Objected Oriented Simulation Environment (MOOSE) developed at INL. It is a finite-element based framework that utilizes a modular approach, allowing scientists and engineers to create new fully coupled, multiphysics applications [19]. With MOOSE also comes a set of community developed modules that implement physics such as heat conduction, tensor mechanics, and porous flow. Of interest for this implementation is the MOOSE ray-tracing module, which is still under development and is not publicly available. Early access to the ray-tracing module was given in order to complete this implementation.

Rattlesnake is the MOOSE-based application developed at INL for deterministic radiation transport. It currently supports steady-state and transient,  $k$ -eigenvalue and source driven problems through diffusion methods, the method of discrete ordinates, and the method of spherical harmonics [20]. The suite of MOOSE-based applications include RELAP-7 [21] for hydrodynamic analysis and BISON [22] for reactor fuel performance. Given the availability of a ray-tracing MOOSE module and the multi-physics capabilities present in Rattlesnake given that it is a MOOSE application, Rattlesnake was the chosen application for the previously described first-collision source treatment.

The features and usage of the MOOSE ray-tracing module are first discussed in the section that follows. The implementation of the uncollided flux computation and first-collision source method in Rattlesnake is then discussed. It is important to note that given the fact that the ray-tracing module is still under significant development, the implementation that follows is limited due to an incomplete feature set in the module.

### 4.1 MOOSE Ray-tracing Module

The ray-tracing module is under current development by D. Gaston for the use in MOCKingbird [23], a neutron transport code using the method of characteristics (MOC) written in the MOOSE

framework. It provides a straight-forward framework for the tracing of rays in an arbitrary mesh.

Standard ray tracing algorithms utilize a bulk synchronous method of ray generation and propagation, which is to simply generate all of the rays at the beginning of the process and after all are generated propagate and communicate them. The MOOSE ray-tracing module utilizes what it terms the SMART (Scalable Massively Asynchronous Ray Tracing) method, in which rays are generated in *chunks* and are propagated and communicated as deemed necessary by the algorithm. An optimal ray chunk size and propagation threshold have been set such that a processor will typically generate more rays only if it does not have a sufficient number of rays buffered to be propagated and communicated.

Other considerations have been made to increase tracing efficiency, such as the use of a shared memory pool for ray storage. Considering the algorithm can generate millions of rays in seconds, memory allocation for each ray (which is a system call and is relatively slow) is a significant bottleneck. A shared pool does not deallocate memory for rays that have been completed, it instead places them in a pool to be reused and reset for later rays that are generated. The module also provides a slew of visual aids to help the user determine the optimal chunk size and mesh partitioning.

To the user, the module presents itself in MOOSE as a `UserObject` (a basic MOOSE class for user-specific data) titled a `RayTracingStudy`. The `RayTracingStudy` is a base class to be inherited from that contains a virtual function to be overridden, `generateRays()`, in which rays are allocated. When using the SMART ray tracing algorithm previously discussed, the user can call a function to buffer set of rays that have been generated to be propagated if the buffered size of rays is appropriate for propagation. A `RayKernel` object is defined that describes the action to take place on each segment (in an element) along the path of a ray, which can include storing data on an element and modifying the data that is carried along with a ray. The module also boundary conditions such as reflecting boundaries.

## 4.2 Uncollided Flux

The computation of the uncollided flux is a separate entity from the first-collision source treatment and its implementation in Rattlesnake follows.

### 4.2.1 Ray-tracer Spawn Objects

A `RayTracerSpawnObject` is the parent class for an object that generates source points and target points for the purposes of computing the uncollided flux. A user provides some restriction and these objects collect a set of points and weights as necessary. These objects are created and passed to the `RayTracingStudy` to generate the necessary combination of each source point and target point for the computation of the uncollided flux. The inheritance is split further into a `RayTracerSource` and a `RayTracerTarget`, which implement the necessary base methods for generating source points and target points, respectively. The possible `RayTracerSource` objects are:

- `RayTracerBlockSource` - identifies sources restricted by block (volumetric sources),
- `RayTracerBoundarySource` - identifies sources restricted by sideset (surface sources),
- `RayTracerPointSource` - identifies point sources,

and the possible `RayTracerTarget` objects are:

- `RayTracerBlockTarget` - identifies targets restricted by block (volumetric target),
- `RayTracerPointTarget` - identifies point targets.

Any combination of the above source objects can be utilized together. Target elements must be unique, that is, multiple target objects may not generate target points to the same element. The `RayTracerSpawnObject` base class provides methods to generate quadrature, allowing the user to provide a unique set of quadrature sets for each object.

### 4.2.2 Ray Generation

The `UncollidedFluxRayTracer` object is inherited from `RayTracingStudy` and generates the rays as necessary to compute the uncollided flux. Its primary input are a set of source and target objects as described in 4.2.1. It first generates all of the target points as defined by the



objects supplied. If a parallel computation is used, the target point list must be communicated because source processors will likely not have knowledge of some of the elements that are searched to contain target points. Once the target point list is complete, the `UncollidedFluxRayTracer` generates the source point and spawns rays accordingly. The general algorithm follows below.

```

Class UncollidedFluxRayTracer
  foreach RayTracerTarget input by user do
    | generate and store target points;
  end
  if is parallel computation then
    | communicate target points/weights;
    | shuffle local target points;
  end
  foreach RayTracerSource input by user do
    | generate source points by element;
    foreach source element do
      | foreach source point in element do
        | foreach target point do
          | generated ray from source point to target point;
          | if number of buffered rays is sufficiently large then
            | execute generated rays for propagation;
          end
        end
      end
    end
  end
end

```

Figure 4.1: Algorithm for the class that generates rays for the uncollided flux computation, the `UncollidedFluxRayTracer`.

### 4.2.3 Uncollided Flux Accumulation

The accumulation of the uncollided flux is accomplished by the `UncollidedFlux` kernel, which contains the process to complete on each segment of a ray. The `UncollidedFlux` object is inherited from a `RayKernel`, which is the base object in the ray-tracing module that performs a task on a ray segment. Recall in 3.4 that a different mathematical process is followed in the case of the contribution of a volumetric source element to its own element. Therefore, the `UncollidedFlux` kernel contains two routines: in-source rays, that is rays that are the contribution of a volumetric source to its own element, and one for all other rays. The `UncollidedFluxRayTracer` appropriately tags rays when they are generated based on these two choices.

Due to current restrictions as set by the MOOSE ray-tracing module, it is only possible to store cell-averaged values on each element. This is a severe restriction that set to be removed in the near future as the module is made a public component of MOOSE. At this point, it should be possible to instead accumulate results into a piecewise linear discontinuous representation on each element

A single in-source ray is one that is spawned using angular quadrature to be intersected at a point on the surface of the element, as described in Equation 3.11 and Figure 3.1. It contains a routine to sample one-dimensional quadrature along said ray and accumulates the contribution from each segment (which represents a volumetric subdivision of the element) to the desired point. The results are appended to the cell-average data on the element.

The remaining rays (not in-source rays) have two possible routines in the `UncollidedFlux` kernel. If a ray has not yet reached its target, the optical thickness is accumulated for each group on the segment traversed through the single element. Said accumulated values are then appended to the group-wise data that follows said ray, which accumulates the total optical thickness to the point right before a ray enters its final element. If the ray has reached its final element, the remaining optical thickness contribution is made and used to produce the contribution to the target point. The results are again appended to the cell-average data on the final element.

### 4.3 First-Collision Source Treatment

#### 4.3.1 Uncollided Flux Import

Recall again that the MOOSE ray-tracing module supports only cell-averaged values. Projecting a cell-average solution on a higher order  $S_N$  mesh for the purpose of the first-collision source is not ideal. A alternate method was devised to make possible a higher order uncollided flux solution for the purpose of this treatment. When higher accuracy is desired, the uncollided flux is obtained on a mesh that is once more refined than the mesh used in  $S_N$  calculation. The resulting solution is then imported into the Rattlesnake  $S_N$  solve via a MOOSE projection transfer, which projects the finer solution onto a first-order L2 Lagrange variable in MOOSE. Said variable is linear discontinuous representation with one degree of freedom per element node and eight nodal values per hexahedral element. While this method is computationally costly due to the increased ray and ray segment requirement by refining the ray-tracing mesh, it makes possible further verification of the implementation thus far.

#### 4.3.2 Source Kernels

From Equation (2.8), we recall that the  $S_N$  solve with first-collision source treatment is the same as the standard solve except the volumetric and boundary sources are replaced with the first-collided scattering source,  $\mathbf{H}\psi_0$ . With the uncollided flux,  $\psi_0$ , obtained from 4.2, the first-collided scattering source is

$$\mathbf{H}\psi_0 = \frac{1}{4\pi} \Sigma_s(\mathbf{r}) \int_{4\pi} \psi_0(\mathbf{r}, \boldsymbol{\Omega}) d\Omega = \Sigma_s(\mathbf{r}) \phi_0(\mathbf{r}), \quad (4.1)$$

with the previously made assumption that scattering is isotropic.

In the MOOSE (and by extension Rattlesnake) system, a kernel represents the residual contribution at a single quadrature point for a single physic. For example, a volumetric source kernel exists for all of the transport methods implemented in Rattlesnake. This implementation simply includes the creation of additional kernels that are similar to the volumetric source kernels, but instead represent the above in Equation (4.1). Kernels were created for the SAAF-CFEM (self-adjoint angular flux, continuous finite element), LS-CFEM (least-square, continuous finite element), and DFEM

(discontinuous finite element)  $S_N$  formulations in Rattlesnake.

### 4.3.3 Action System

The action system in MOOSE provides a means to consolidate the input syntax so that the user does not need to define each specific component of the problem. For example, in Rattlesnake the user provides an equation description (diffusion,  $S_N$ , or  $P_N$ ) and a solution method (SAAF-CFEM, LS-CFEM, DFEM, etc), a source description, and the appropriate geometry and material definition. The action system then creates the necessary variables, the kernels for each component of the transport equation, etc.

The current implementation of the first-collision source treatment in the action system contains only a filename for an Exodus II output. This output should contain the uncollided flux solution on the entire mesh (same as the  $S_N$  mesh, but can be more or less refined) using a specific variable name structure. The action system will then complete the correct projection of the uncollided flux solution and create the necessary source kernels depending on the solution method utilized. Lastly, it will correctly append the uncollided flux to the total angular flux as in Equation (2.9).

## 5. RESULTS

### 5.1 Uncollided Flux Verification

The following test cases serve as a verification of the uncollided flux method and implementation discussed previously in this work.

#### 5.1.1 Point Source to Volumetric Targets

To start, a test case of a simple configuration is chosen. A point source that emits 1 n/s is located at (0.5,0.5,0.5) in a  $10 \times 10 \times 2$  element mesh of cuboid elements of width  $1 \text{ cm} \times 1 \text{ cm} \times 1 \text{ cm}$ . The desired quantity is the integrated uncollided flux in each element (the “target” elements) except for the element that contains the source (the element with lower vertex (0,0,0) and upper vertex (1,1,1)). The geometry specification follows in Figure 5.1.

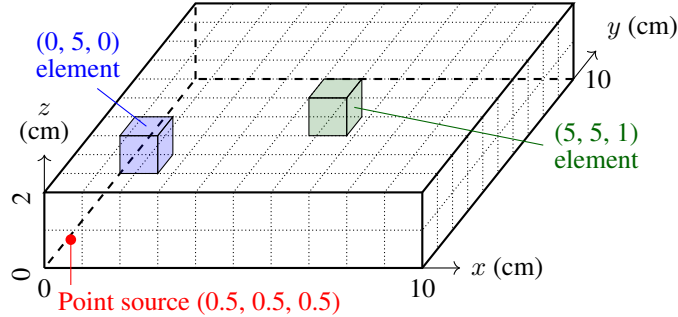


Figure 5.1: Geometry specification for the point source verification case.

The total cross section of the medium is then varied and the uncollided flux in each target element evaluated using increasing Gaussian quadrature. The reference solution was evaluated numerically to a relative error tolerance of  $1 \times 10^{-12}$  utilizing a MATLAB script that used the `integral3` function to approximate the triple integral over an element volume. The results for the two colored elements in Figure 5.1 follow in Table 5.1 and Figure 5.2. The two elements were

chosen because the source sees only one face of the first target element, but sees multiple faces for the second target element.

Table 5.1: Errors in the integrated uncollided flux for two elements in the point source case.

Element	Quad pts. (1D)	Absolute relative error	
		Void	$\Sigma_t = 0.5 \text{ cm}$
(0, 5, 0)	1	$3.2626694 \times 10^{-3}$	$2.1467300 \times 10^{-2}$
	2	$6.1546384 \times 10^{-5}$	$3.2744340 \times 10^{-4}$
	3	$1.0078175 \times 10^{-7}$	$9.5622707 \times 10^{-7}$
	4	$6.0666534 \times 10^{-10}$	$4.7996283 \times 10^{-9}$
	5	$1.0592426 \times 10^{-12}$	$1.1241697 \times 10^{-13}$
	Reference	$3.1935182 \times 10^{-3}$	$2.6701679 \times 10^{-4}$
(5, 5, 1)	1	$1.6434675 \times 10^{-3}$	$1.7770951 \times 10^{-2}$
	2	$2.6876082 \times 10^{-6}$	$9.8108561 \times 10^{-6}$
	3	$6.7710724 \times 10^{-9}$	$8.6085469 \times 10^{-8}$
	4	$7.8559525 \times 10^{-12}$	$3.9873888 \times 10^{-11}$
	5	$7.7695166 \times 10^{-15}$	$1.6904388 \times 10^{-13}$
	Reference	$1.5629112 \times 10^{-3}$	$4.4695696 \times 10^{-5}$

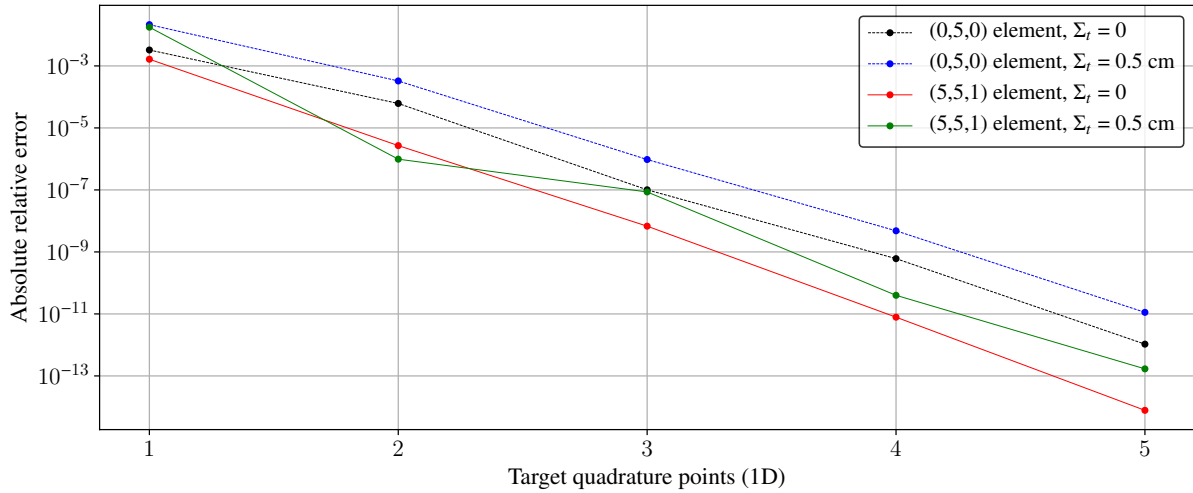


Figure 5.2: Comparison of error in the integrated uncollided flux for two elements in the point source case.

We would expect for the error to increase (for a given quadrature arrangement) when going changing the material from void to an absorbing medium, due to the approximation of the exponential component. As the distance between source and target element is increased, the visible solid angle of the target element as seen by the source point decreases. We would also then expect the error to decrease as distance between the source and target increases. Both of these expectations are seen in the results above.

It is also important to note that when more faces of an element are seen, the quadrature integration will likely perform poorer because the solution will be less smooth over said element. In order to visualize the effect of the target element as seen by the source point, the relative errors for each target element in the mesh were plotted as seen below in Figures 5.3 and 5.4. The heat map color range for both figures is the same to enable easy comparison.

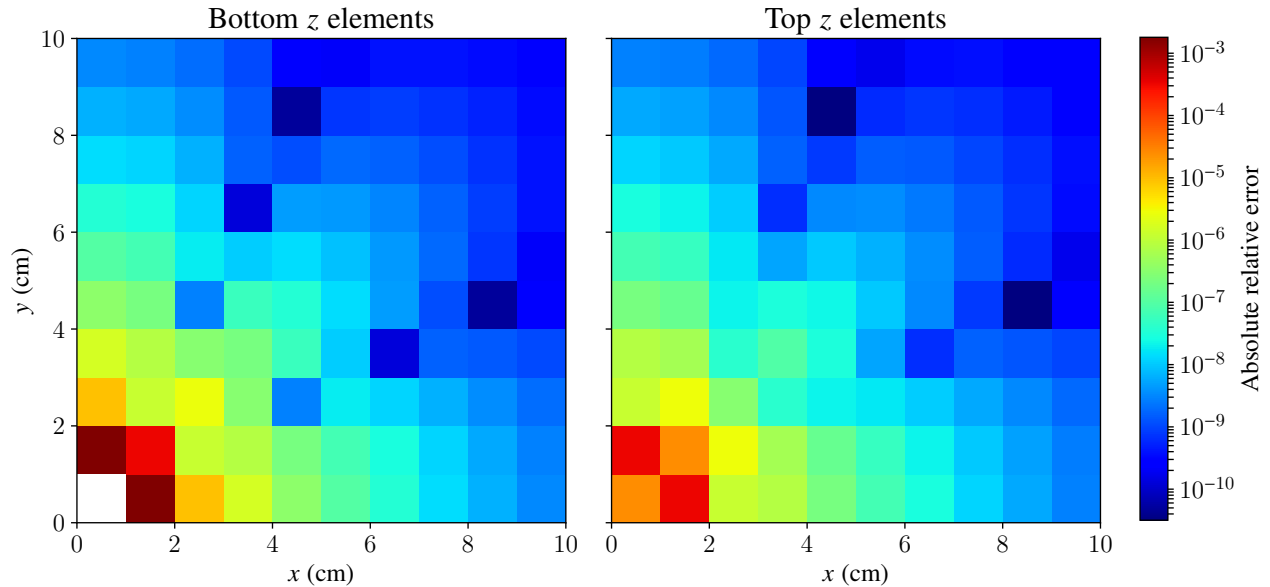


Figure 5.3: Error in the integrated uncollided flux in each element for the point source case in void with 27 quadrature points per target element.

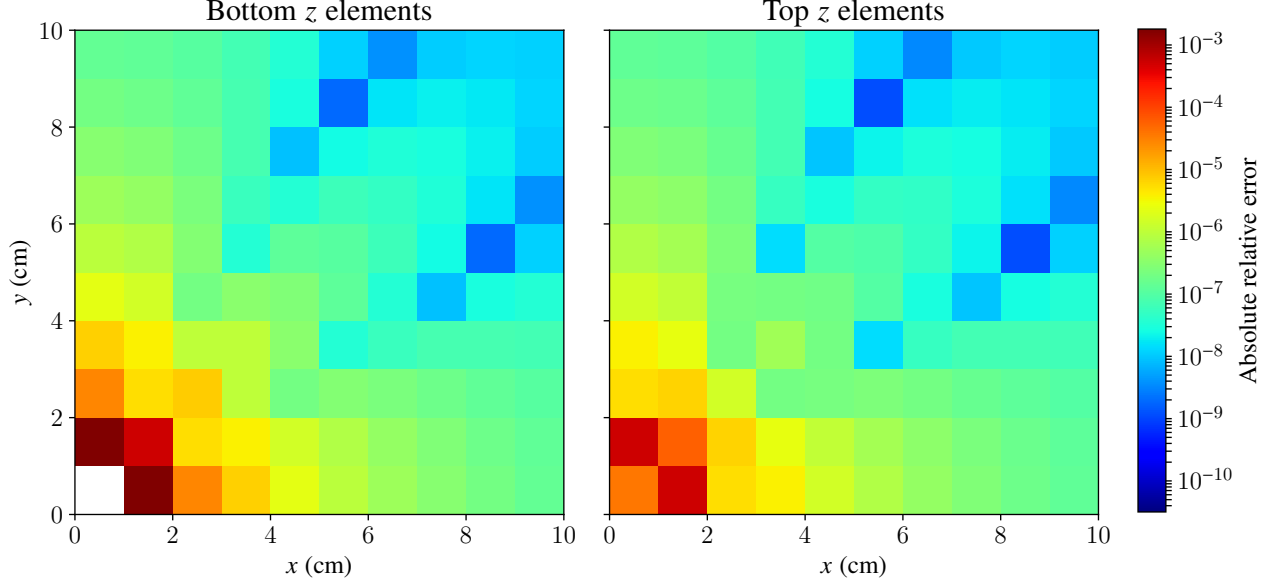


Figure 5.4: Error in the integrated uncollided flux in each element for the point source case in an absorbing medium with 27 quadrature points per target element.

As the cross section decreases, the uncollided flux solution becomes smoother. This fact presents itself as significantly less error in the void medium because the solution is more accurately approximated using numerical integration. In addition, we see that the error trends like the visible target area as seen by the source point which is a result of the increased difficulty in approximating the angular integral when the visible surface area increases. These results verify the target volume integration as implemented by this method.

### 5.1.2 Volumetric Source to Point Targets

We now present a test case with a very similar configuration to the one in 5.1.1. Consider instead a volumetric source that emits  $1 \text{ n/cm}^3 \cdot \text{s}$  in a cuboid element centered at  $(0.5, 0.5, 0.5)$  with width 1 cm. Again, the mesh is a  $10 \times 10 \times 2$  element mesh of cuboid elements of width 1 cm  $\times$  1 cm  $\times$  1 cm. The desired quantity is the point uncollided flux in the center of each element (the “target” elements) except for the element that contains the source (the element with centroid  $(0.5, 0.5, 0.5)$ ). The geometry specification follows in Figure 5.5.



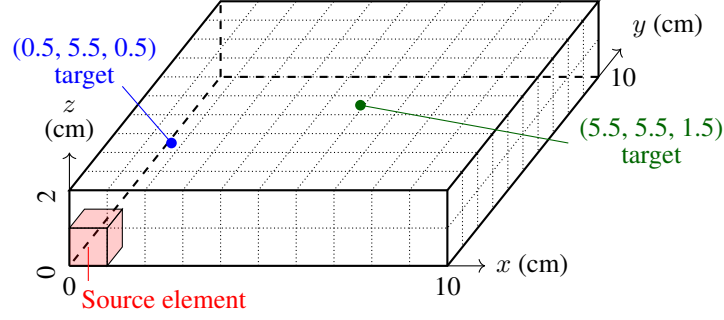


Figure 5.5: Geometry specification for the volumetric source verification case.

Again, the total cross section of the medium is varied and the uncollided flux for each point element evaluated using increasing Gaussian quadrature in the source element. The reference solution was evaluated numerically to a relative error tolerance of  $1 \times 10^{-12}$  utilizing the same MATLAB script as in 5.1.1. The results for the two points in Figure 5.5 follow in Table 5.1.

Table 5.2: Errors in the point uncollided flux at the center of two elements in the volumetric source case.

Element	Quad pts. (1D)	Absolute relative error	
		Void	$\Sigma_t = 0.5 \text{ cm}$
(0, 5, 0)	1	$3.2626695 \times 10^{-3}$	$2.1467301 \times 10^{-2}$
	2	$6.1546384 \times 10^{-5}$	$3.2744340 \times 10^{-4}$
	3	$1.0078173 \times 10^{-7}$	$9.5622696 \times 10^{-7}$
	4	$6.0663506 \times 10^{-10}$	$4.7995108 \times 10^{-9}$
	5	$1.0334406 \times 10^{-12}$	$1.1122930 \times 10^{-11}$
	Reference	$3.1935183 \times 10^{-3}$	$2.6701680 \times 10^{-4}$
(5, 5, 1)	1	$1.6434675 \times 10^{-3}$	$1.7770951 \times 10^{-2}$
	2	$2.6876083 \times 10^{-6}$	$9.8108560 \times 10^{-6}$
	3	$6.7710821 \times 10^{-9}$	$8.6085470 \times 10^{-8}$
	4	$7.8060056 \times 10^{-12}$	$3.9869492 \times 10^{-11}$
	5	—	$1.7889846 \times 10^{-13}$
	Reference	$1.5629112 \times 10^{-3}$	$4.4695696 \times 10^{-5}$

It should be noted that the results above in Table 5.2 should match those in Table 5.1 to within

machine precision because we are performing the same mathematical integration as in 5.1.1, which was the purpose of choosing the geometry for this case as such. The integration is the same because the medium between the point and the integrated region is homogeneous and this equality is confirmed for these two points. In addition, Figures 5.3 and 5.4 from the point to volume case were replicated in this case and follow in Figures 5.6 and 5.7.

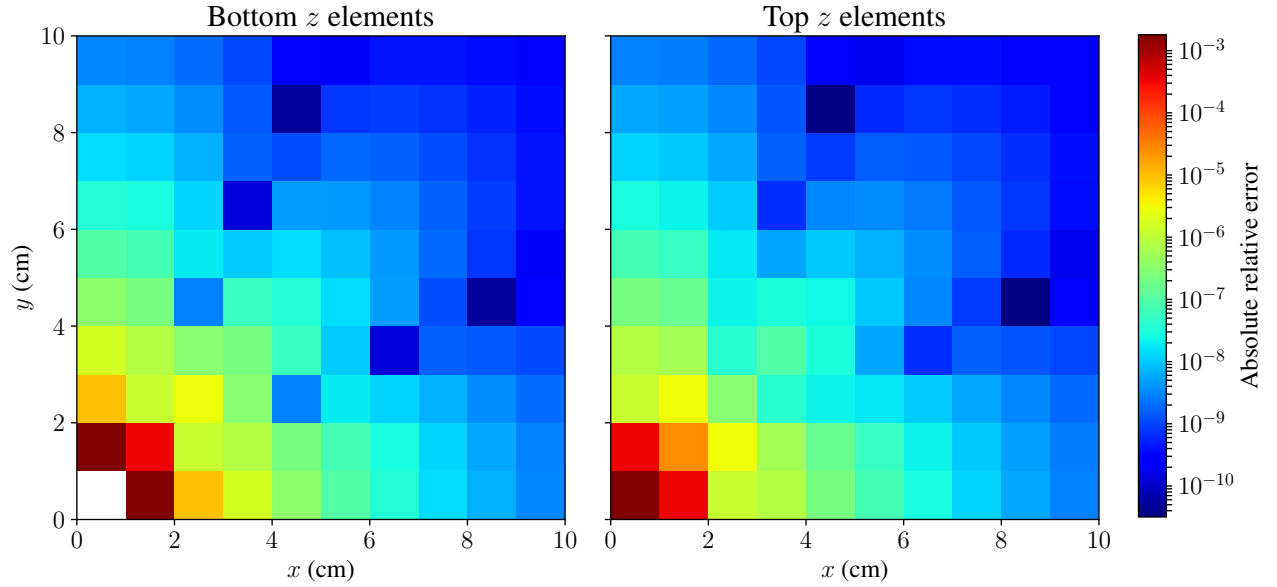


Figure 5.6: Error in the point uncollided flux at the center of each element for the volumetric source case in void with 27 quadrature points in the source element.

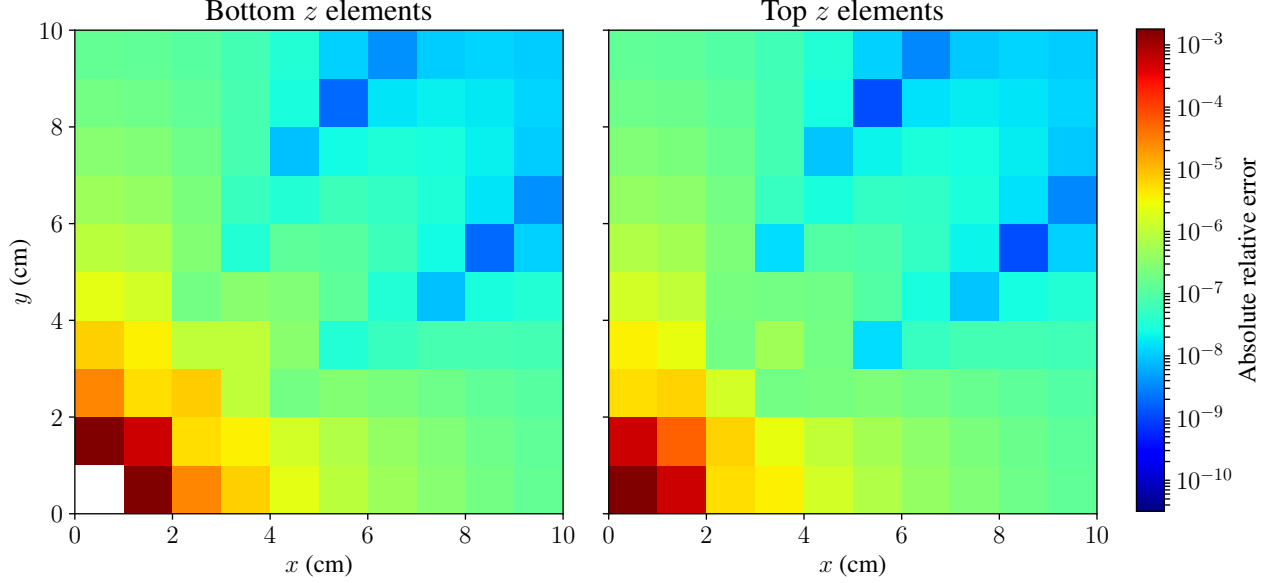


Figure 5.7: Error in the point uncollided flux at the center of each element for the volumetric source case in an absorbing medium with 27 quadrature points in the source element.

Comparing to the same figures in the point to volume case, we see the similar results as expected. These results verify the source volume integration as implemented by this method.

### 5.1.3 In-cell Source Contribution

To test the in-cell source contribution as discussed in 3.5, we consider a single cell with size  $1 \text{ cm} \times 1 \text{ cm} \times 1 \text{ cm}$  and a volumetric source strength of  $1 \text{ n/cm}^3 \cdot \text{s}$ . The desired quantity is the scalar flux at the centroid of the cell. The cell has a cross section of either  $\Sigma_t = 0$  or  $\Sigma_t = 0.5 \text{ cm}^{-1}$ . The reference solution was obtained using MCNP with the ONEGXS application in order to produce a solution to within 0.01% error quoted at  $2\sigma$ . In addition, a reference solution using the ray-tracing implemented (denoted RT) was obtained with Gauss-Chebyshev with 100 polar and azimuthal angles per quadrant. The results follow in Figure 5.8 and are obtained by increasing Gauss-Chebyshev angular quadrature.

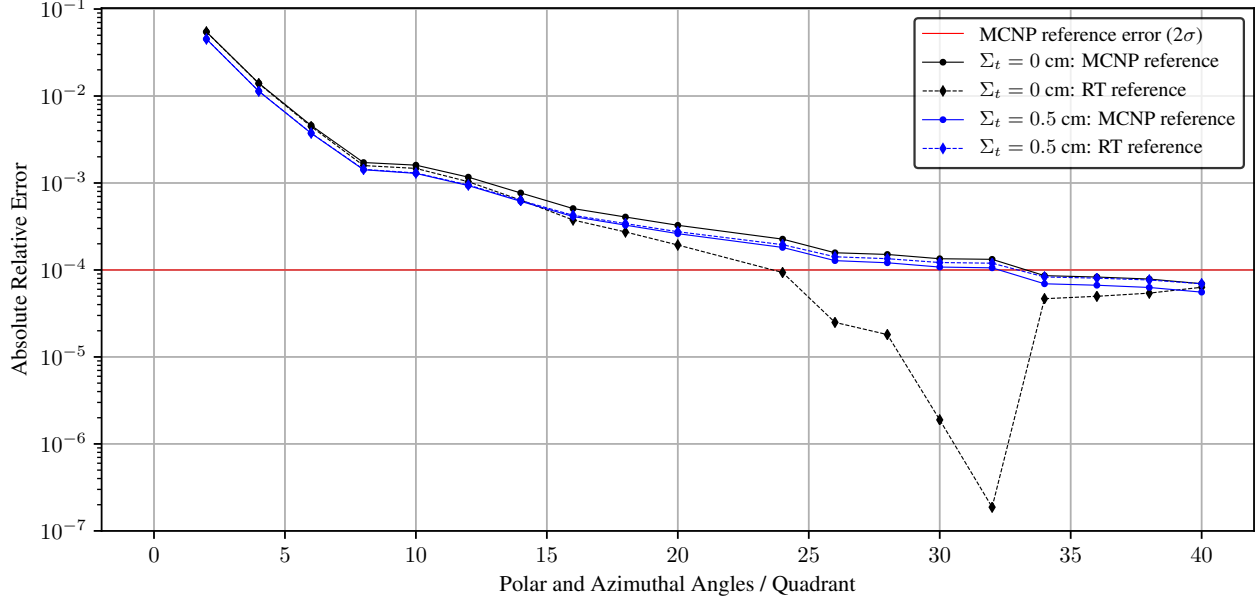


Figure 5.8: Results in computing the uncollided flux at the center of the element with increasing angular quadrature for the in-cell source contribution verification case.

As seen above, the implementation does converge to both the MCNP result and the ray-tracing result, However, the convergence rate is quite poor. 20 polar and azimuthal angles are required to get the solution within ten orders of magnitude of the reference solution, which is 3,200 angles.

## 5.2 Kobayashi Benchmark

The Kobayashi 3-D benchmark void problems proposed by Kobayashi [6] consist of three one-group source problems that are prominent to ray effects. The first configuration of each problems only contains a near-void and a pure absorbing medium, while the other configuration replaces the pure absorber with a material which has a scattering cross-section of 50% of the total-cross section. In particular, problem 3 was chosen as a verification for the implementation thus far. The desired values from this benchmark are point values at various points in the geometry. The geometry is as follows in Figure 5.9, while the material properties for each case follow in Table 5.3.

Table 5.3: Cross-section and source definitions for the Kobayashi benchmark.

Region	Source Strength ( $\text{n/cm}^3 \cdot \text{s}$ )	$\Sigma_t$ ( $\text{cm}^{-1}$ )	Pure absorber	Scattering
			$\Sigma_a$ ( $\text{cm}^{-1}$ )	$\Sigma_s$ ( $\text{cm}^{-1}$ )
1	1	0.1	0	0.05
2	0	$10^{-4}$	0	$0.5 \times 10^{-4}$
3	0	0.1	0	0.05

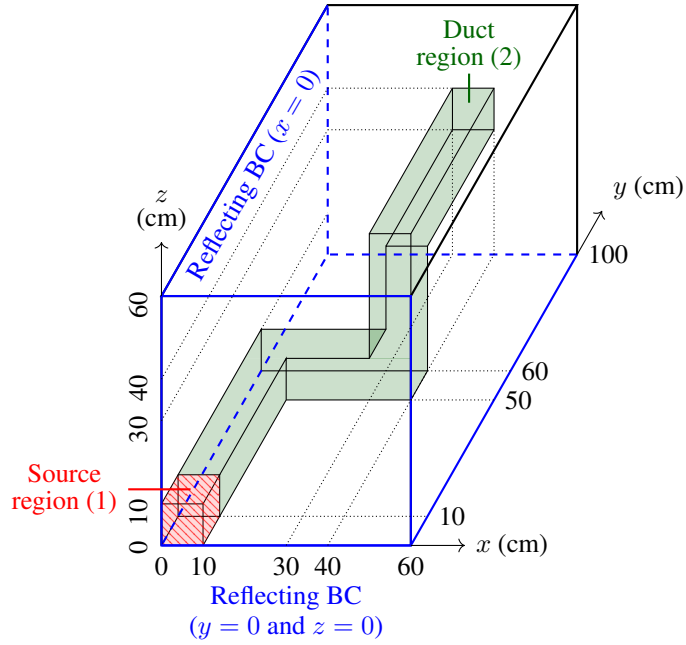


Figure 5.9: Kobayashi benchmark problem 3 geometry specification.

While the MOOSE ray-tracing module is capable of handling boundary conditions, the reflecting boundary implementation is currently incomplete. Therefore, for the absorption only problem, an additional layer of elements was added below  $x = 0$ ,  $y = 0$ , and  $z = 0$  in order to simulate the reflected source from the single source element. For the problem with scattering, the entire geometry had to be unfolded according to the reflecting boundary conditions.

### 5.2.1 Pure-Absorbing Problem

The pure-absorbing problem offers an additional opportunity for the verification of the implemented uncollided flux methods in a heterogeneous medium. For this problem, ray-tracing was utilized by spawning rays from quadrature points in the source region to the points specified in the benchmark. The reference solutions in the original benchmark paper [6] for the absorbing problem were only given to five points after the decimal, which are not of sufficient precision for verifying the most-refined ray-tracing solution. Instead, reference solutions were generated by the code developed by R. Sanchez [24] to a higher precision. Note that said code does not generate a solution for the source position (5, 5, 5). First, a comparison of the uncollided flux errors with increasing spatial quadrature in the source elements follows in Figure 5.10.

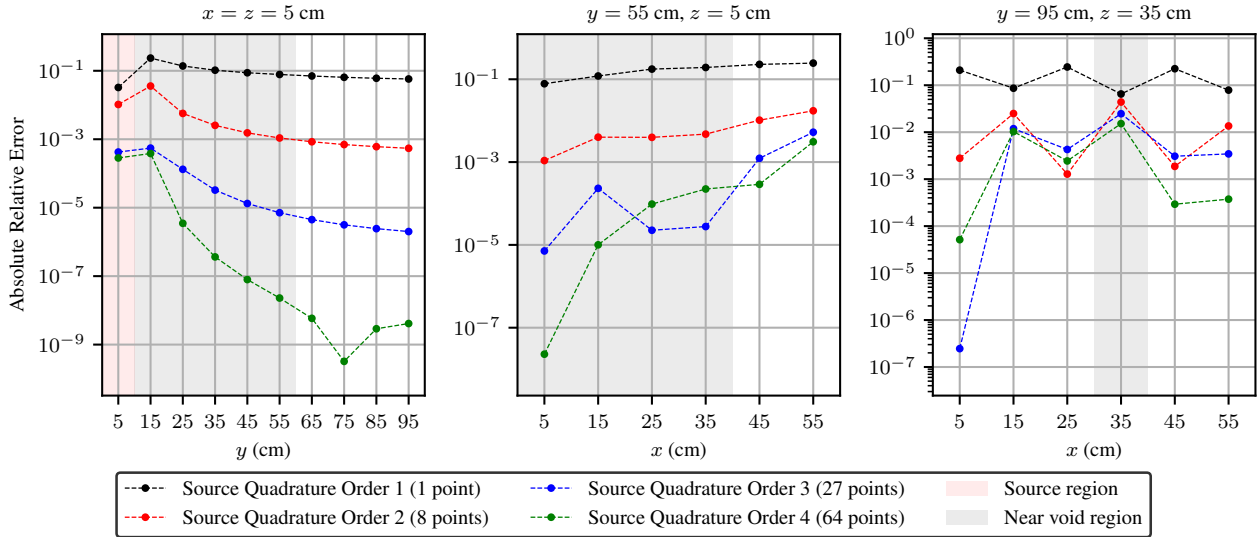


Figure 5.10: Point uncollided flux solutions produced by ray-tracing for Kobayashi problem 3 without scattering.

With the first set of solutions (those at  $x = z = 5$  cm), we again see the improved numerical approximation as the distance from the source region increases. Note that the difference in the trend between  $y = 5$  cm and  $y = 15$  cm is likely due to the fact that the in source method was

utilized. For the second set, the point at  $x = 5$  cm converges at a significantly greater rate than the others due to the fact that it has the least heterogeneous view of the source volume. This is likely the case as well for the point  $x = 5$  cm in the third set of solutions above.

A set results with the highest possible spatial quadrature order in Rattlesnake (with both the original Kobayashi benchmark reference and the Sanchez reference) follow in Table 5.4. Lastly, the uncollided flux solution is plotted in the planes of relevance to the benchmark ( $z = 5$  cm and  $z = 35$  cm) as follows in Figure 5.11. For said solution, the mesh was refined 3 times, there were 27 quadrature points in each source element, and Level-Symmetric  $S_{14}$  was utilized for the in-source contribution.

Table 5.4: Point uncollided flux errors produced by ray-tracing for Kobayashi problem 3 without scattering and 1000 quadrature points per  $10 \text{ cm} \times 10 \text{ cm} \times 10 \text{ cm}$  source element (most refined).

Case	Coordinates (cm <sup>3</sup> )	Ray-tracing relative error	
		Kobayashi reference [6]	Sanchez reference [24]
A	5, 5, 5 <sup>†</sup>	$1.63396 \times 10^{-4}$	–
	5, 15, 5	$6.92778 \times 10^{-7}$	$5.13740 \times 10^{-8}$
	5, 25, 5	$6.75503 \times 10^{-7}$	$2.26265 \times 10^{-8}$
	5, 35, 5	$1.49263 \times 10^{-6}$	$1.61694 \times 10^{-8}$
	5, 45, 5	$2.67133 \times 10^{-6}$	$1.35085 \times 10^{-8}$
	5, 55, 5	$3.31716 \times 10^{-7}$	$1.19478 \times 10^{-8}$
	5, 65, 5	$8.60471 \times 10^{-7}$	$1.06525 \times 10^{-8}$
	5, 75, 5	$3.25873 \times 10^{-6}$	$9.72518 \times 10^{-9}$
	5, 85, 5	$8.87787 \times 10^{-7}$	$1.35085 \times 10^{-8}$
	5, 95, 5	$5.14459 \times 10^{-8}$	$8.69699 \times 10^{-9}$
B	5, 55, 5	$3.31716 \times 10^{-7}$	$1.19478 \times 10^{-8}$
	15, 55, 5	$4.17032 \times 10^{-7}$	$3.95243 \times 10^{-7}$
	25, 55, 5	$4.47758 \times 10^{-6}$	$4.55608 \times 10^{-6}$
	35, 55, 5	$3.98688 \times 10^{-6}$	$2.97198 \times 10^{-6}$
	45, 55, 5	$6.44085 \times 10^{-6}$	$6.63633 \times 10^{-6}$
	55, 55, 5	$5.64782 \times 10^{-4}$	$5.64482 \times 10^{-4}$
C	5, 95, 35	$1.14355 \times 10^{-6}$	$1.49793 \times 10^{-6}$
	15, 95, 35	$4.70170 \times 10^{-4}$	$4.68900 \times 10^{-4}$
	25, 95, 35	$8.23811 \times 10^{-5}$	$7.97868 \times 10^{-5}$
	35, 95, 35	$2.80603 \times 10^{-3}$	$2.80731 \times 10^{-3}$
	45, 95, 35	$1.61639 \times 10^{-4}$	$1.62127 \times 10^{-4}$
	55, 95, 35	$7.29783 \times 10^{-5}$	$7.33878 \times 10^{-5}$

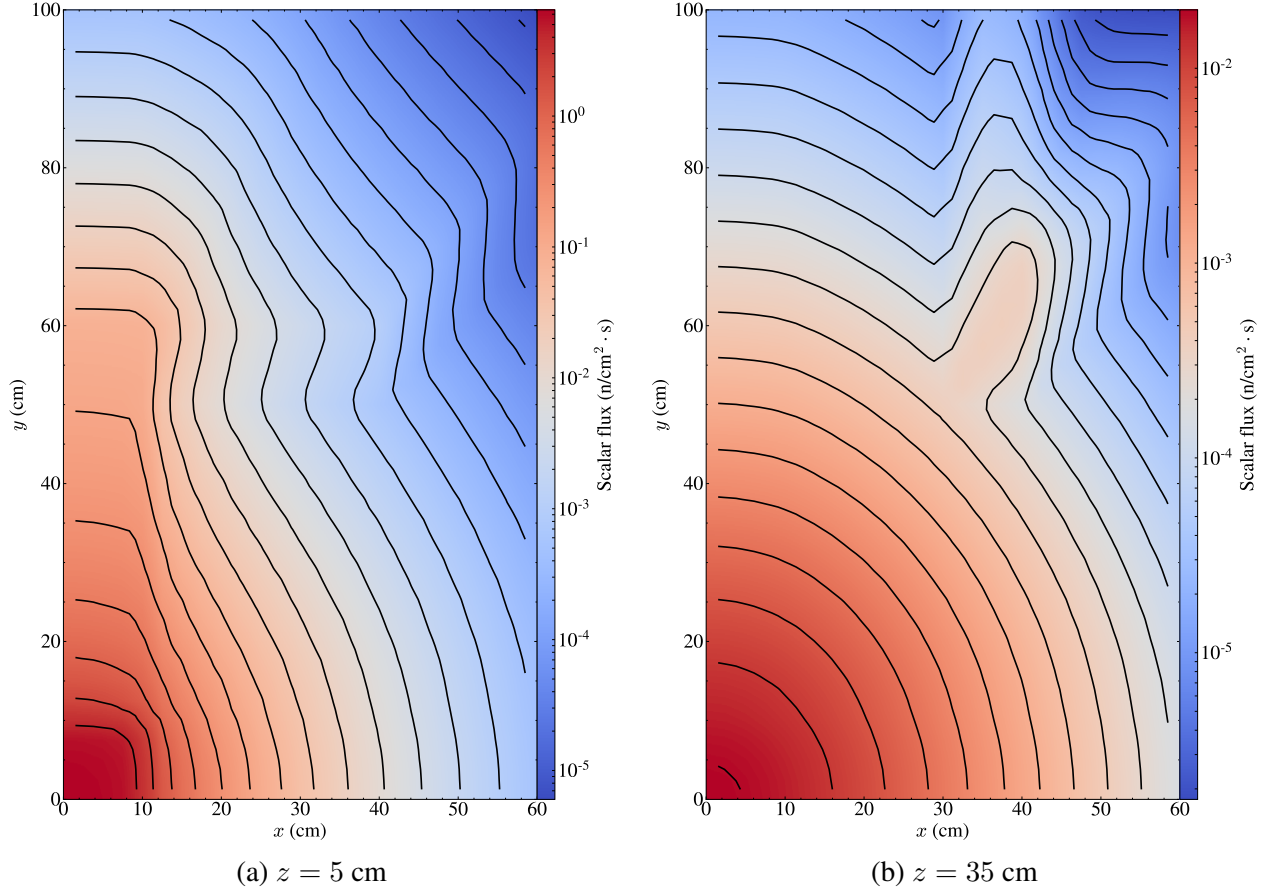


Figure 5.11: Scalar flux contour comparison for Kobayashi problem 3 without scattering using the ray-tracing method.

## 5.2.2 Scattering Problem

Kobayashi problem 3 with scattering is solved both with and without first-collision source treatment in order to provide a comparison. A uniform mesh refinement (with the coarsest mesh cells of size  $10 \text{ cm} \times 10 \text{ cm} \times 10 \text{ cm}$ ) of 1 and 2 was chosen in order to remain consistent with the other codes that were tested in the original benchmark. For all of the results that follow, Level-Symmetric angular quadrature was chosen. The visualized solutions that follow in Figures 5.12 and 5.13 utilize a uncollided flux computation on a mesh that was once more uniformly refined than the  $S_N$  mesh. Recall that this is due to the significant limitation addressed in 4.3.1 in which only cell-averaged values are possible with the current MOOSE ray-tracing module implementation.



8 quadrature points were chosen per source cell for all ray-tracing calculations and  $S_{14}$  angular quadrature was utilized for the in-source contribution.

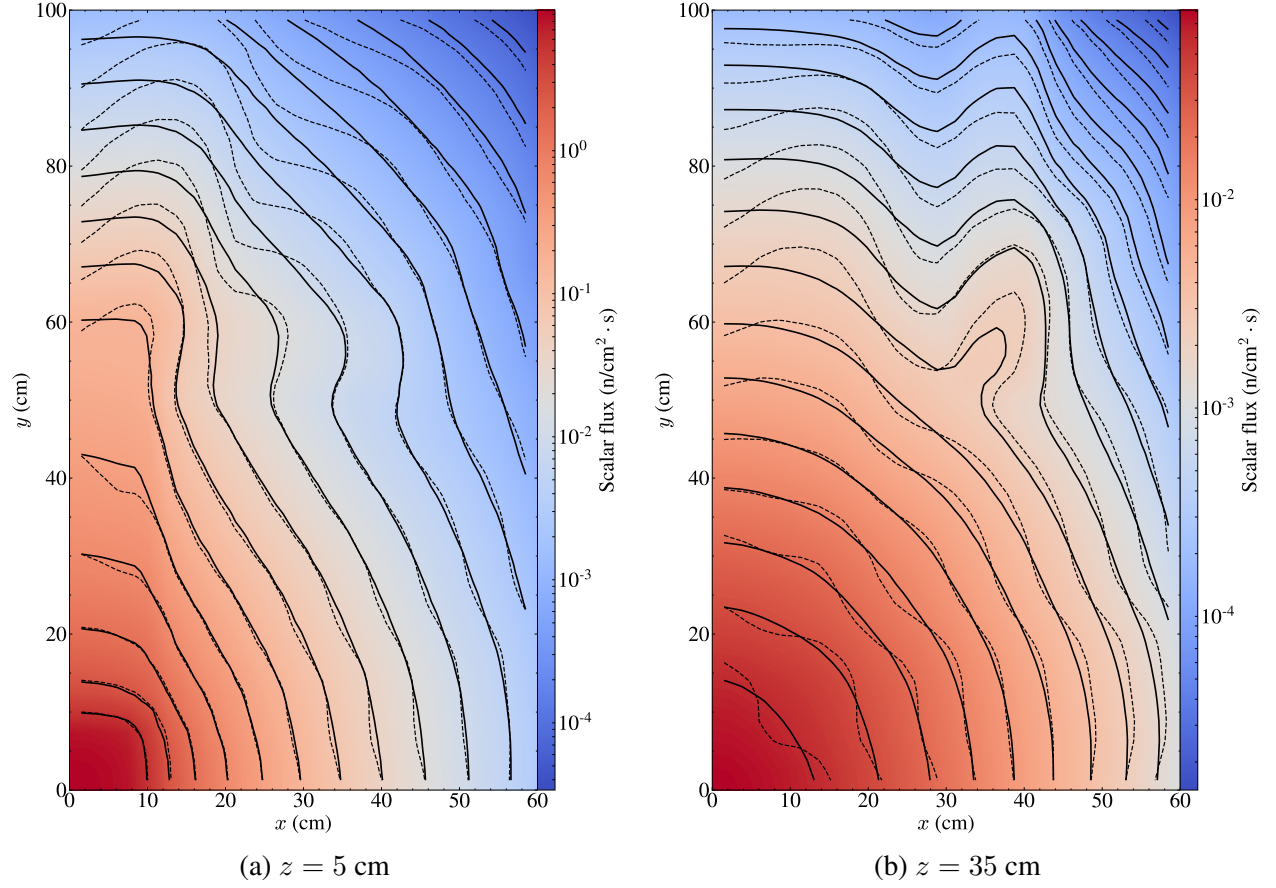


Figure 5.12: Scalar flux contour comparison for Kobayashi problem 3 with scattering,  $S_{10}$ , and 2 uniform mesh refinements, and 3 uniform mesh refinements for the ray-tracing solution. The solid contours are with treatment, dotted are without.

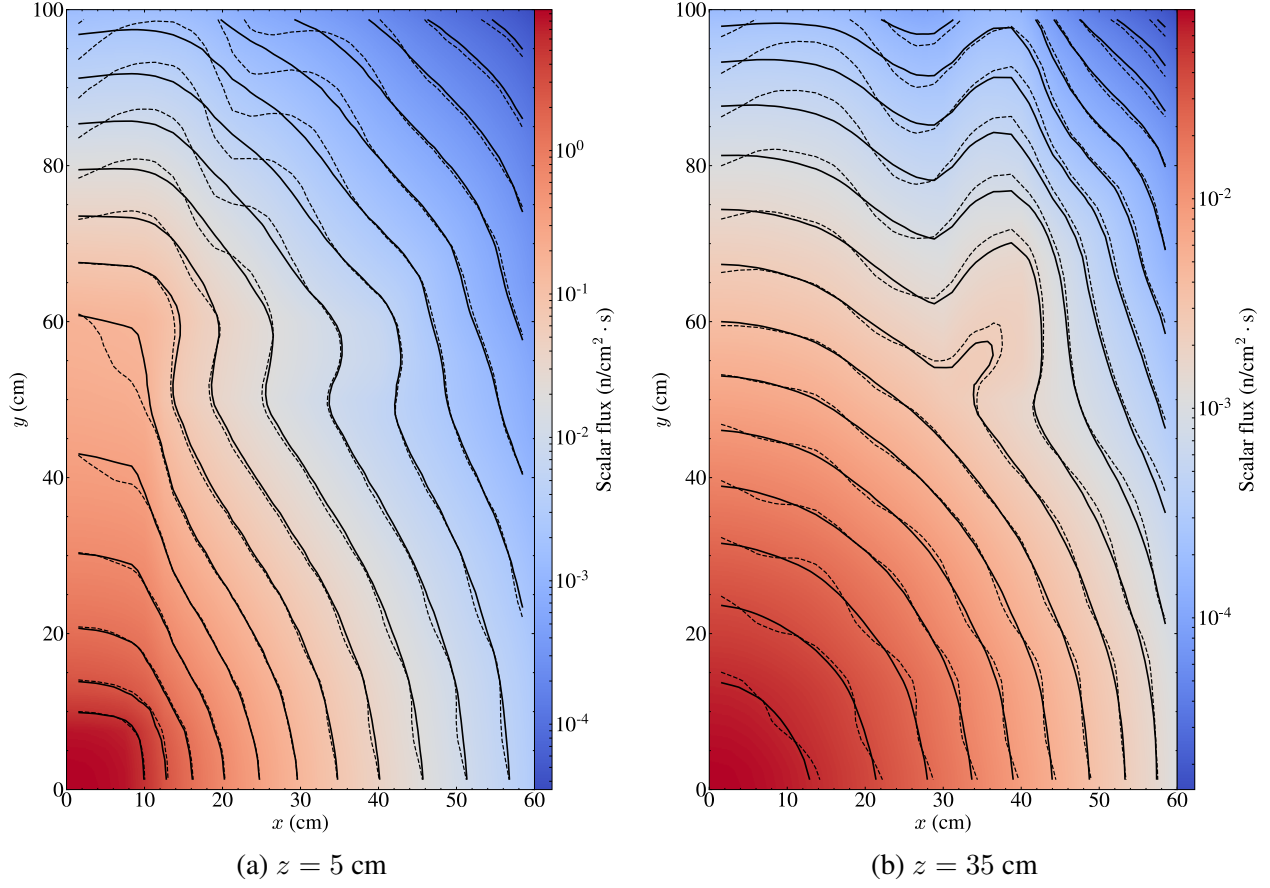


Figure 5.13: Scalar flux contour comparison for Kobayashi problem 3 with scattering and  $S_{14}$ , and 2 uniform mesh refinements, and 3 uniform mesh refinements for the ray-tracing solution. The solid contours are with treatment, dotted are without.

With both of the figures above, the peaks in the solution without treatment along the discrete ordinates are noticeably improved when compared to the solutions with treatment. The effect is magnified near the end and beyond the duct region after the source in the  $z = 5$  cm plane. While there is no highly-refined reference solution available, the combination of the contours in the figures above in addition to the point solutions that follow are significant enough to show the importance and capability of the first-collision source treatment. The errors with varying of angular quadratures and mesh refinements follow in Figure 5.14.

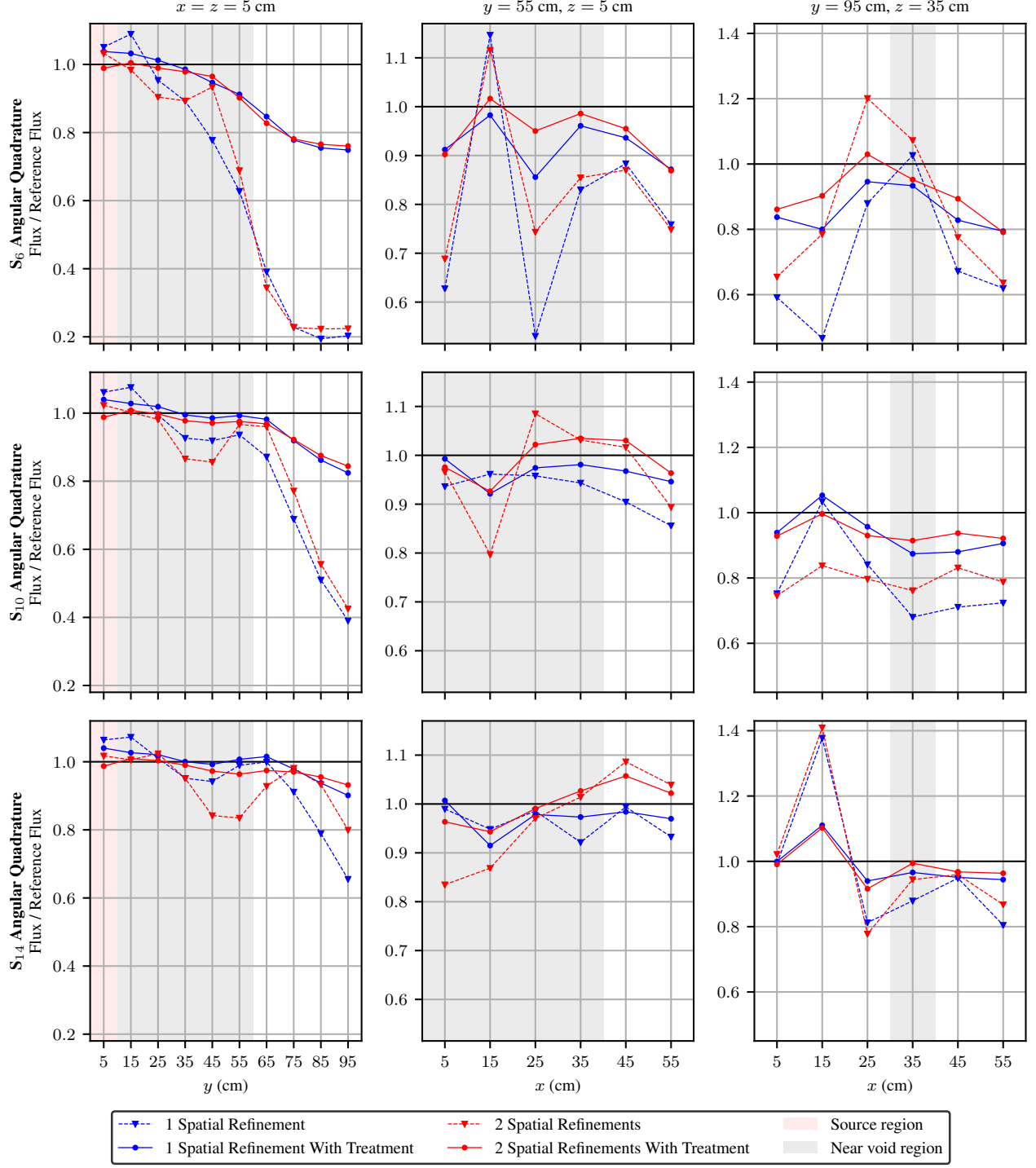


Figure 5.14: Scalar flux comparison for Kobayashi problem 3 with scattering. The mesh refinement listed is for the  $S_N$  mesh and the ray-tracing mesh is once more refined.

With the above we do see with the above the general theme of improved accuracy that we

would expect through increasing angular quadrature, increasing spatial quadrature, and through the first-collision source treatment. It is noticed that the shape of the solution with treatment follows the shape of the solution without treatment. We see also the interaction between spatial and angular discretization. See for example the solution at (15, 95, 35) in the plots above. With increasing angular quadrature at  $S_{14}$ , spatial refinement does not decrease the error in the solution without treatment. Looking back to this same position in the contour plot in Figure 5.13, we see at around that same position there is a peak in the non-treated solution along one of the discrete ordinates. This consequence is a large contributor to the inconsistent convergence of the solutions above. With this, we present the solutions that are the most refined in angle below in Table 5.5 to accentuate that high order angular quadrature is not enough to resolve ray effects in this benchmark without additional treatment.

Table 5.5: Scalar flux comparison for Kobayashi problem 3 with scattering and  $S_{20}$ .

Case	Coordinates (cm <sup>3</sup> )	Reference Flux (cm <sup>-2</sup> ·s <sup>-1</sup> )	1 Spatial Refinement Error		2 Spatial Refinement Error	
			Without Treatment	With Treatment <sup>†</sup>	Without Treatment	With Treatment <sup>†</sup>
A	5, 5, 5	$8.61578 \times 10^0$	6.55%	4.03%	1.33%	1.32%
	5, 15, 5	$2.16130 \times 10^0$	7.14%	2.58%	0.66%	1.03%
	5, 25, 5	$8.93784 \times 10^{-1}$	2.22%	2.21%	3.29%	0.51%
	5, 35, 5	$4.78052 \times 10^{-1}$	1.81%	0.62%	2.75%	0.08%
	5, 45, 5	$2.89424 \times 10^{-1}$	3.21%	0.09%	5.51%	1.08%
	5, 55, 5	$1.92698 \times 10^{-1}$	1.34%	1.40%	17.2%	3.15%
	5, 65, 5	$1.04982 \times 10^{-1}$	3.20%	2.55%	20.2%	3.86%
	5, 75, 5	$3.37544 \times 10^{-2}$	1.54%	0.66%	14.1%	4.15%
	5, 85, 5	$1.08158 \times 10^{-2}$	6.83%	2.48%	5.62%	3.55%
	5, 95, 5	$3.39632 \times 10^{-3}$	12.8%	4.19%	0.25%	2.65%
B	5, 55, 5	$1.92698 \times 10^{-1}$	1.34%	1.40%	17.2%	3.16%
	15, 55, 5	$6.72147 \times 10^{-2}$	1.19%	7.77%	9.50%	4.25%
	25, 55, 5	$2.21799 \times 10^{-2}$	6.91%	3.98%	7.09%	2.66%
	35, 55, 5	$9.90646 \times 10^{-3}$	4.99%	2.25%	0.96%	0.80%
	45, 55, 5	$3.39066 \times 10^{-3}$	4.26%	3.31%	3.67%	3.94%
	55, 55, 5	$1.05629 \times 10^{-3}$	5.31%	3.11%	3.88%	3.84%
C	5, 95, 35	$3.44804 \times 10^{-4}$	7.85%	2.90%	3.60%	2.46%
	15, 95, 35	$2.91825 \times 10^{-4}$	0.42%	1.13%	2.67%	0.56%
	25, 95, 35	$2.05793 \times 10^{-4}$	10.2%	4.32%	9.50%	4.34%
	35, 95, 35	$2.62086 \times 10^{-4}$	3.45%	0.33%	19.3%	4.98%
	45, 95, 35	$1.05367 \times 10^{-4}$	8.70%	6.90%	6.43%	3.54%
	55, 95, 35	$4.44962 \times 10^{-5}$	3.59%	1.56%	4.89%	3.01%

<sup>†</sup> Ray-tracing mesh is once more refined than the  $S_N$  mesh.

In order to highlight the significant limitation in the MOOSE ray-tracing module in which only cell-averaged solutions are available, Figure 5.14 was replicated but instead with a comparison of the treated solutions with varying levels of refinement. This includes a ray-tracing mesh that is the same as the  $S_N$  mesh (therefore no projection of the solution is made—the first-collision source is assumed to be cell-wise constant). The results with a higher refined ray-tracing mesh were presented first because said configuration partially mimics uncollided flux solution of the ray-tracing module supported piecewise constant values on each element (where significantly less computation time is required). These results follow in Figure 5.15.

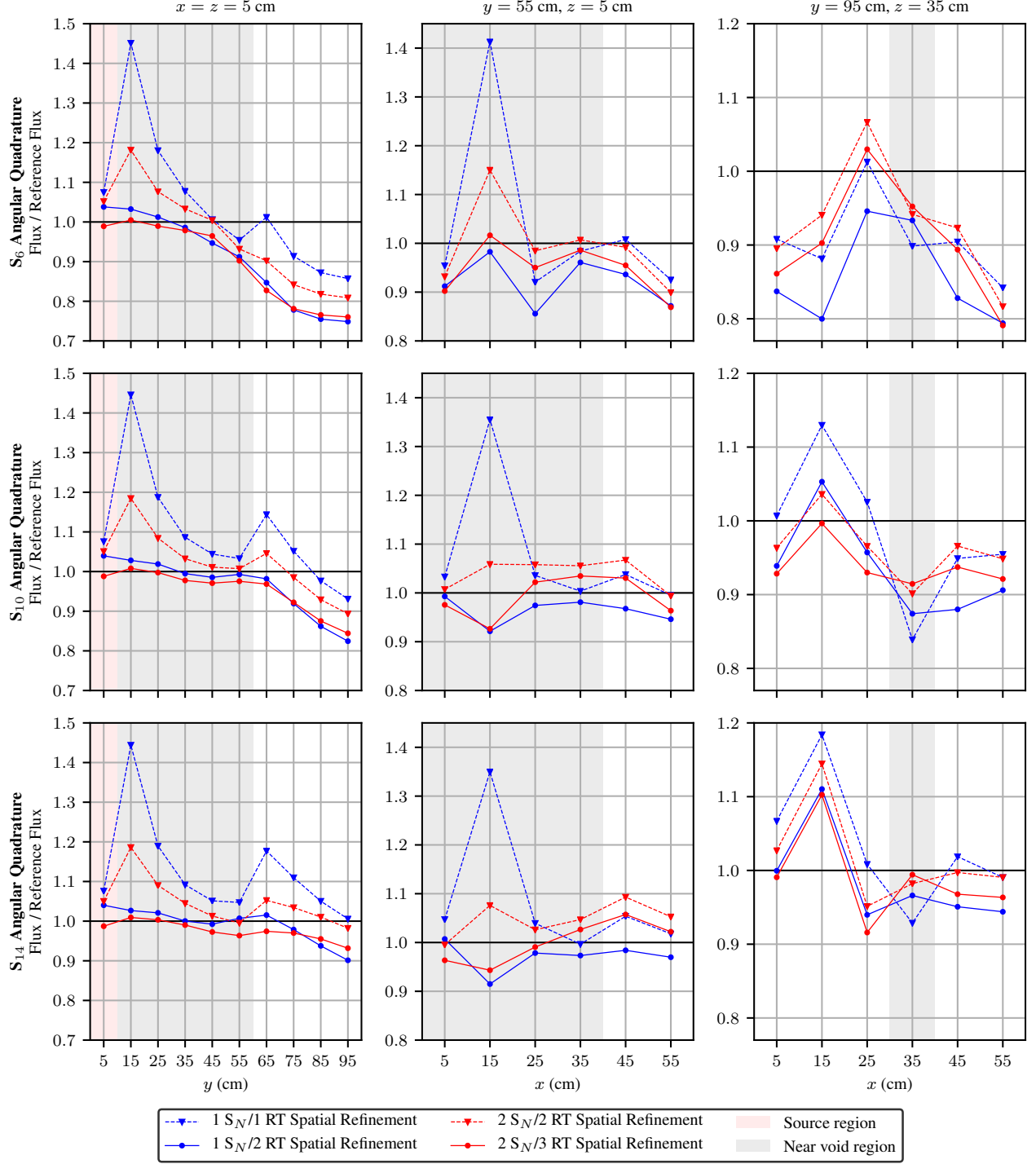
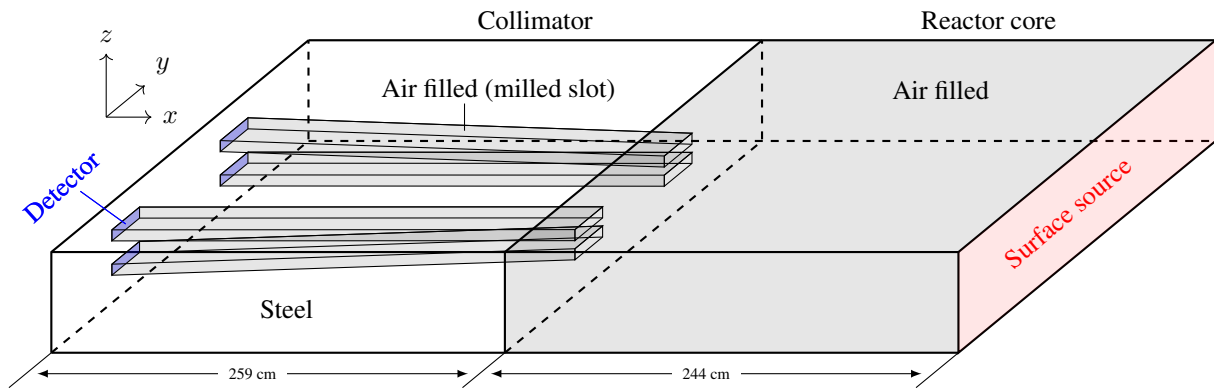


Figure 5.15: Treated scalar flux comparison for Kobayashi problem 3 with scattering and varying mesh refinements.

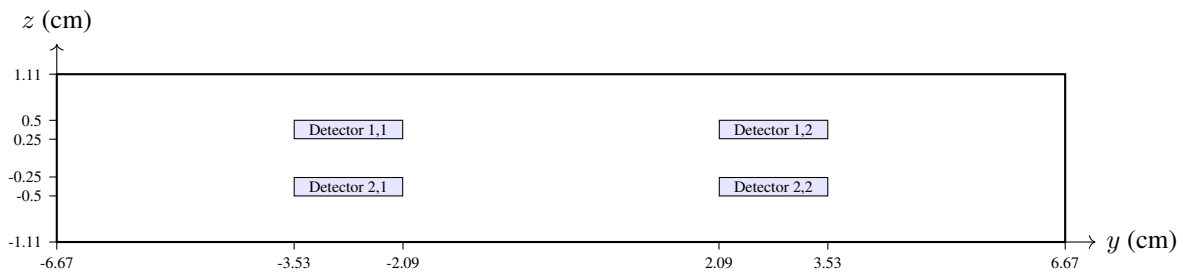
### 5.3 TREAT Hodoscope

The TREAT fast neutron hodoscope was the original motivation for the completion of this work. The geometry of the hodoscope is depicted in Figures 2.1, 2.3, and 2.4. For the purpose of examining this configuration with the implementation made in this research, it was decided not to mesh all 344 milled slots in the rear collimator. The difficulty in simulating this problem is in the fractional solid area of the milled slots and the detector interfaces, which can be represented well with a fewer number of slots providing that the proportions are correct.

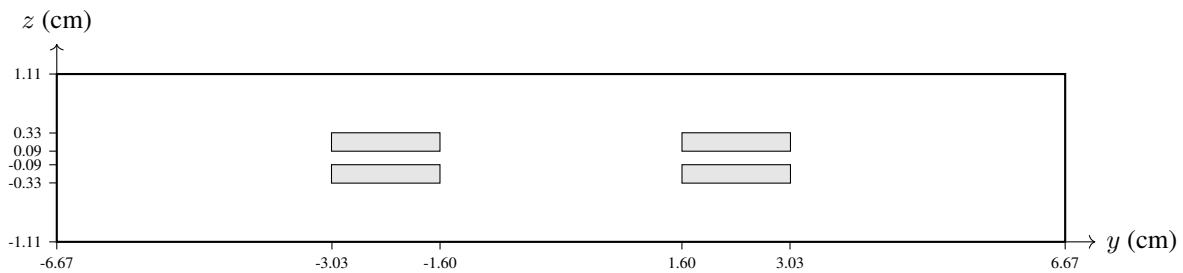
For these reasons, an approximated hodoscope model with only 4 milled slots was created. The cross sectional area of the slots is kept the same as in the full hodoscope, and the slot separation was selected from the slate in Figure 2.3. The reactor core region outside of the hodoscope as seen in Figure 2.1 was created instead as a single block of air, in which the assumption was made that source neutrons will not scatter and make it the entire length down the rear collimator to reach a detector while still fast. The milled slots are not meshed in a staggered manner as they are in the full hodoscope. The distance from the hodoscope to the test sample is restricted to that of the distance in the true core. The test sample is modeled as a uniform, isotropic surface source with strength  $1 \text{ n/cm}^2\cdot\text{s}$  with all emission into the fast group. The geometry configuration follows in Figure 5.16.



(a) Full representation ( $x$ -axis is not to scale)



(b)  $yz$  cross section at detectors (to scale)



(c)  $yz$  cross section between collimator and core (to scale)

Figure 5.16: Geometry specification for the simplified hodoscope model.

Considering the quantity of interest in the simulation of the hodoscope is the detector response, only fast neutrons are of importance in the simulation. Therefore, two energy groups were chosen: fast neutrons above 1 MeV, and everything else below 1 MeV. Cross sections were generated with these energy groups with Serpent and a general TREAT core input from reactor physics staff at INL and they follow in Table 5.6.



Table 5.6: Multigroup cross sections for the hodoscope model.

	Steel		Air	
	$g = 1$	$g = 2$	$g = 1$	$g = 2$
$\Sigma_t \text{ (cm}^{-1}\text{)}$	$2.59 \times 10^{-1}$	$9.08 \times 10^{-1}$	$9.86 \times 10^{-5}$	$4.14 \times 10^{-4}$
$\Sigma_{s,g \rightarrow 1} \text{ (cm}^{-1}\text{)}$	$2.19 \times 10^{-1}$	0	$7.83 \times 10^{-5}$	0
$\Sigma_{s,g \rightarrow 2} \text{ (cm}^{-1}\text{)}$	$3.98 \times 10^{-2}$	$8.29 \times 10^{-1}$	$1.51 \times 10^{-5}$	$3.90 \times 10^{-4}$
Scattering ratio	0.84	0.91	0.79	0.94

Recall that the purpose of the TREAT neutron hodoscope is to image fast neutrons emitted from the test sample. This is done by surrounding the collimator slots with a highly absorbing material such that only neutrons that travel directly from the test sample to the detector remain fast. The hodoscope model was meshed such that the source can be enabled and disabled specifically in the regions that are directly in the line of sight to a collimator slot and detector. With this, we are able to compute the uncollided flux contribution in each detector by the source that is “seen” (called the direct source) by the detector, and the uncollided flux contribution in each detector by the source that not seen by the detector (called the indirect source).

The uncollided flux due to the direct source for each detector face is approximated to be  $1.057 \times 10^{-7} \text{ cm}^{-2} \cdot \text{s}^{-1}$  and the uncollided flux due to the indirect source for each director face is approximated to be  $1.046 \times 10^{-9} \text{ cm}^{-2} \cdot \text{s}^{-1}$ . Therefore, roughly 99% of the uncollided fast neutron flux at each detector face is due to the direct source. These values were also validated using a MCNP ONEGXS model with the ability to enable and disable certain parts of the surface source.

Upon observing the two-group cross sections and the results above, we suggest that the collided flux is likely not significant in the detector response, because:

- The probability of a neutron scattering in air to then move in the direction is low. The mean free path for scattering in air (and staying fast) is roughly 126 m, while the entire assembly is 0.04 scattering mean free paths thick.
- The probability of a neutron remaining fast through the steel is low. The mean free path for

down scattering in steel is 0.25 m, while the collimator is 10 mean free paths thick.

This is a significant result in a multi-physics calculation because there would be no need to simulate the entire hodoscope. Using the ray-tracing implementation, only the solution in the core is needed. Rays can be spawned from the sample directly to the detector regions to determine the response.

## 6. CONCLUSIONS

The primary objective of this research is to implement a remedy to ray effects using a semi-analytical treatment of the uncollided flux using advanced ray-tracing techniques, coupled with a standard discrete ordinate treatment of the uncollided flux. Given an uncollided flux solution, the first-collision source treatment is a rather straight-forward application that has been implemented by others. The significant advancement made by this work is in the development of an advanced method of computing the uncollided flux in a transport code that enables high-fidelity, multi-physics applications, and supports arbitrary FEM grids.

The uncollided flux algorithm is advanced in that it supports arbitrary sources, is easily coupled with finite element and multi-physics as a result of its implementation in the MOOSE framework, and it supports unstructured meshes. It is currently limited in the fact that it only supports cell-averaged solutions, but this limitation should be resolved by the time the MOOSE ray-tracing module is made public. This limitation also hampers the computational needs of the implementation and hampers scalability. The implementation of the uncollided flux algorithm has been successfully verified through multiple test cases. The first-collision source treatment was implemented in all of the  $S_N$  transport schemes in Rattlesnake. The treatment was verified using the Kobayashi benchmark and was shown to significantly reduce ray effects.

The simulation of a simplified TREAT neutron hodoscope model confirmed the need for uncollided flux algorithms to model the hodoscope. While the original motivation for the first-collision source treatment in this research was the hodoscope, it was later determined that a proper uncollided flux solution is likely all that is needed to properly compute detector responses in the hodoscope. In a multi-physics simulation of TREAT and the fast neutron hodoscope, this research enables the quick computation of detector responses without the need to compute the angular flux solution in the entire hodoscope region by spawning rays directly from the test region to the detector region only.

There exist many opportunities for continued work regarding this research. Of these, the great-

est opportunity is in the implementation of higher spatial moments for the uncollided flux solution. This implementation would involve solving a linear system on each element using the accumulated source at each quadrature point to result in a linear discontinuous representation on each element. Other desired extensions include the support for properly spawning rays with reflecting boundaries and the ability for Rattlesnake  $S_N$  solves to directly invoke the uncollided flux computation.

## REFERENCES

- [1] M. DeHart, “TREAT modeling and simulation development with validation requirements.” INL/CON-15-36200, August 2015.
- [2] A. D. Volpi, R. Pecina, R. Daly, D. Travis, R. Stewart, and E. Rhodes, “Fast-neutron hodoscope at treat: development and operation,” *Nuclear Technology*, vol. 27, no. 3, pp. 449–487, 1975.
- [3] K. Lathrop, “Remedies for ray effects,” *Nuclear Science and Engineering*, vol. 45, no. 3, pp. 255–268, 1971.
- [4] J. Tencer, “The impact of reference frame orientation on discrete ordinates solutions in the presence of ray effects and a related mitigation technique,” in *ASME 2014 International Mechanical Engineering Congress and Exposition*, pp. V08AT10A017–V08AT10A017, American Society of Mechanical Engineers, 2014.
- [5] K. D. Lathrop, “Ray effects in discrete ordinates equations,” *Nuclear Science and Engineering*, vol. 32, no. 3, pp. 357–369, 1968.
- [6] K. Kobayashi, “A proposal for 3-D radiation transport benchmarks for simple geometries with void region,” in *3-D deterministic radiation transport computer programs. Features, applications and perspectives*, 1997.
- [7] J. C. Chai, H. S. Lee, and S. V. Patankar, “Ray effect and false scattering in the discrete ordinates method,” *Numerical Heat Transfer, Part B Fundamentals*, vol. 24, no. 4, pp. 373–389, 1993.
- [8] P. Coelho, “The role of ray effects and false scattering on the accuracy of the standard and modified discrete ordinates methods,” in *ICHMT DIGITAL LIBRARY ONLINE*, Begel House Inc., 2001.

- [9] J. Morel, T. Wareing, R. Lowrie, and D. Parsons, “Analysis of ray-effect mitigation techniques,” *Nuclear science and engineering*, vol. 144, no. 1, pp. 1–22, 2003.
- [10] T. Wareing, J. Morel, and D. Parsons, “A first collision source method for ATTILA, an unstructured tetrahedral mesh discrete ordinates code,” tech. rep., Los Alamos National Lab., NM (United States), 1998.
- [11] J. W. Kim and Y.-O. Lee, “AETIUS solutions for kobayashi 3d benchmarks with the first collision source method on the volume source and unstructured tetrahedral mesh,” *Annals of Nuclear Energy*, vol. 113, pp. 446–469, 2018.
- [12] R. E. Alcouffe, “Partisn calculations of 3D radiation transport benchmarks for simple geometries with void regions,” *Progress in Nuclear Energy*, vol. 39, no. 2, pp. 181 – 190, 2001.
- [13] T. M. Evans, A. S. Stafford, R. N. Slaybaugh, and K. T. Clarno, “Denovo: A new three-dimensional parallel discrete ordinates code in SCALE,” *Nuclear technology*, vol. 171, no. 2, pp. 171–200, 2010.
- [14] R. Lillie, “GRTUNCL3D: A discontinuous mesh three-dimensional first collision source code,” in *Proc. ANS Topl. Mtg. Radiation Protection and Shielding*, vol. 1, p. 368, American Nuclear Society LaGrange Park, IL, 1998.
- [15] K. Kosako and C. Konno, “FNSUNCL3: first collision source code for tort,” *Journal of Nuclear Science and Technology*, vol. 37, no. sup1, pp. 475–478, 2000.
- [16] U.S. Department of Energy, “Mission need statement for the resumption of transient fuel testing, U.S. DOE Memorandum, Peter B. Lyons, Acting Assistant Secretary for Nuclear Energy, to Richard B. Provencher, Manager, Idaho Operations Office.” December 2010.
- [17] U.S. Department of Energy, “Resumption of transient testing capability,” April 2013.
- [18] D. Chichester, S. Watson, J. Johnson, and D. Wachs, “The TREAT fast-neutron hodoscope and plans for restoring it to operation,” *Transactions of the American Nuclear Society*, vol. 113, pp. 377–380, 2015.

- [19] D. Gaston, C. Newman, G. Hansen, and D. Lebrun-Grandie, “MOOSE: A parallel computational framework for coupled systems of nonlinear equations,” *Nuclear Engineering and Design*, vol. 239, no. 10, pp. 1768–1778, 2009.
- [20] Y. Wang, “Nonlinear diffusion acceleration for the multigroup transport equation discretized with SN and continuous FEM with rattlesnake,” in *Proceedings of the 2013 International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering-M and C 2013*, 2013.
- [21] H. Zhang, H. Zhao, F. N. Gleicher, M. D. DeHart, L. Zou, D. Andrs, and R. C. Martineau, “RELAP-7 development updates,” tech. rep., Idaho National Lab.(INL), Idaho Falls, ID (United States), 2015.
- [22] R. Williamson, J. Hales, S. Novascone, M. Tonks, D. Gaston, C. Permann, D. Andrs, and R. Martineau, “Multidimensional multiphysics simulation of nuclear fuel behavior,” *Journal of Nuclear Materials*, vol. 423, no. 1-3, pp. 149–163, 2012.
- [23] D. Gaston, B. Forget, K. Smith, and R. Martineau, “Verification of MOCKingbird, unstructured-mesh, method of characteristics implementation using the MOOSE multi-physics framework,” in *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering, M&C 2017*, 2017.
- [24] I. Suslov, R. Sanchez, and I. Zmijarevic, “Deterministic reference solution for 3-d kobayashi benchmarks,” *M&C*, vol. 99, p. 1765, 1999.