A Conservative First-Collision Source Treatment for Ray Effect Mitigation in Discrete-Ordinate Radiation Transport Solutions

By

Alex Bitz Christensen

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Nuclear Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in Charge:

Dr. Massimiliano Fratoni, Chair
Dr. Jasmina Vujic
Dr. Michael Nacht

May 2024

Abstract

A Conservative First-Collision Source Treatment for
Ray Effect Mitigation in Discrete-Ordinate
Radiation Transport Solutions

by

Alex Bitz Christensen

Doctor of Philosophy in Engineering - Nuclear Engineering

University of California, Berkeley

Professor Massimiliano Fratoni, Chair

Deterministic transport codes play a fundamental role in the modeling and simulation of neutron transport. One of the most common deterministic methods is the method of discrete ordinates, also known as the $\mathrm{S_n}$ method. While offering significant advantages over other deterministic methods or stochastic methods like Monte Carlo, the method of discrete ordinates suffers from non-physical artifacts in its local solution due to its discretization of angle. These artifacts, referred to as ray effects because of their ray-like appearance, tend to be worse in problems with small sources in areas with little scattering. Significant effort has gone into developing methods to mitigate ray effects, such as the first-collision source treatment, which separates the angular flux into the uncollided and collided fluxes and solving them using non-traditional techniques such as ray tracing. Current ray tracing methods typically trace to a set of points inside a zone to compute an overall flux. However, this approach has significant drawbacks, such as a low potential for optimization, as well as not being conservative. A new method has been developed that traces instead to a set of points on each of a zone's surfaces and computes the currents, before using these to obtain the flux. A comparison between these two ray tracing methods shows significant advantages to the new surface method, including inherent particle conservation, similar convergence rate and increased potential for optimization through ray sharing.

DEDICATION

This paper is dedicated to my family who have supported me all throughout my education and without whom none of this would have been possible.

ACKNOWLEDGMENT AND DISCLAIMERS

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory (LLNL). Lawrence Livermore National Laboratory is operated by Lawrence Livermore National Security, LLC, for the U.S. Department of Energy, National Nuclear Security Administration under Contract DE-AC52-07NA27344. This document has the release number LLNL-TH-862170. This material is based upon work supported by the Department of Energy National Nuclear Security Administration through the Nuclear Science and Security Consortium under Award Number(s) DE-NA0003996 and/or DE-NA0003180, and subaward number B653597. This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or limited, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

This work would not have been possible if not for the efforts of my advisor Max Fratoni and the support of my supervisors at Livermore, in particular Adam Kunen, John Loffeld, and Peter Brown.

Contents

List of Figures

Chapter One

INTRODUCTION

1.1 INTRODUCTION

In 1960, Edward Teller had UNIVAC construct the Livermore Atomic Research Computer (LARC) at what was then the Lawrence Radiation Laboratory to run advanced hydrodynamic simulations [1]. Considered one of the first supercomputers ever created, LARC helped establish a strong connection between the supercomputing and nuclear industries, with many of the first supercomputers having been built for the express purpose of modeling nuclear experiments [2]. The ability to accurately model the behavior and movement of neutrons is of vital importance in a broad set of applications, ranging from the design of nuclear reactor cores to dose calculations found in radiation medicine. In many of these applications, the tolerance for error is small, with even small errors making the difference from a reactor core having a sub-critical geometry and one that is supercritical. Unfortunately, the neutron transport equation - the equation governing the movement and behavior of neutrons in a given system - is difficult to solve due to the high dimensionality of its solutions phase space [3]. It is only under the most simplistic of circumstances, like an infinite slab of uniform material, that a true, analytical solution exist. Because of this, countless hours and resources have been spent developing alternative approaches to solving the transport equation, most involving the use of advanced computational resources. As the use of numerical simulations became more mainstream, the demand for better and better computers increased. This increasing demand played a crucial role in driving the advancement of supercomputing technologies.

One of the most popular techniques used to solve the transport equation is the method of discrete ordinates [4, 5]. Developed in the 1950s as an alternative approach to solving radiation transport problems to the then already popular Spherical Harmonic and Monte Carlo Methods, the method of discrete ordinates, or $\mathrm{S_N}$ method, as it is commonly referred to, is a deterministic numerical method that relies on the discretization of the transport equation's dimensions to obtain a solution [6]. While enjoying relatively similar performance to other popular numerical methods, it suffers from one glaring weakness - ray effects.

At the heart of the discrete ordinate method is the separation of angular dependency by solving the transport equation on a fixed angular grid, with the resulting set of equations being coupled through a collision term [3, 4, 6]. A consequence of this, however, is the creation of artificial oscillations in the angular flux, often taking the appearance of rays, referred to as ray effects. These rays can seriously impact the accuracy of the local solution and potentially render the obtained solution unusable.

Significant effort has been made over the years in developing a variety of different techniques to try and mitigate or even eliminate ray effects, with varying degrees of success. One such technique, the First Collision Source Treatment Method, involves the separation of uncollided and collided fluxes [7]. The uncollided flux - the greatest contributor to ray effects - is solved for using an alternative method that does not suffer from ray effects, and is then used as a source term to obtain the collided flux. The two can then be combined to find the total flux.

1.1.1 Motivation

Currently, when performing a first collision treatment, there are a limited number of techniques available to solve for the uncollided flux without observable ray effects. One of the most popular methods employs ray tracing to find the uncollided flux. The standard approach, referred to in this paper as the interior point method, involves defining a spatial quadrature set along the interior of each zone in a mesh and finding the uncollided flux at each quadrature point by casting rays from the source to the point in space and finding the optical depth along the rays path. The scattering of this uncollided flux then provides a source term to find the collided flux. The problem with this method, however, is that it is not conservative in the way in which it treats its particles.

In this paper, we propose an alternate approach to finding the uncollided flux, which we refer to as the Surface Method. It defines a set of quadratures along the faces of each zone and solves for their currents. By summing these currents, the method uses the Divergence Theorem to obtain the absorption rate inside the zone and through that the scalar flux, and in doing so is inherently conservative.

The surface method possesses certain advantages over the interior point method for first scatter treatments; it is inherently conservative and has more potential for work optimization leading to higher work efficiency. Included in this paper is the derivation of both the interior point and surface methods, as well as the results from several different 2D and 3D test cases demonstrating their behavior in a variety of different scenarios.

No numerical method works well under all conditions, with each having a set of strengths and weaknesses. While not all methods are created equal, having more tools at one's disposal is always better.

1.1.2 Objectives

The objective of this project is to develop a new ray-tracing based technique to solve for the uncollided flux for use inside a first-collision source treatment method to reduce ray effects inside discrete ordinate methods. Following the successful development of such a method, it will be compared against the more standard ray-tracing based technique for solving for the uncollided flux, which throughout this paper will be referred to as the interior point method.

To perform this analysis, both a independently developed, 2D solver will be written using the computing platform MATLAB, as well as the production code ARDRA - a highly capable deterministic ($\mathrm{S_n}$) neutron and gamma transport code developed by Lawrence Livermore National Laboratory [15].

As a final proof of concept, the new surface method will be used to model a real world problem, with its simulated data compared against true experimental results.

1.1.3 Overview of the Following Chapters

- **Background:** An overview of general background information, including a broad overview of neutron transport as well as common methods used to solve the transport equation, with a special focus placed on the discrete ordinates method and source iteration. The section also goes into ray effects, their causes, and some common techniques used to mitigate them, including the first collision source treatment technique.

- **Ray Tracing Methods:** Beginning with integral form of the neutron transport equation, the section derives the base equations used by the interior point and surface methods. It then provides a derivation for both the interior point and surface methods, as well as provides a general explanation of how the ray trace works.

- **Implementation:** This section outlines the way in which the two methods were implemented in both the 2D MATLAB solver as well as ARDRA.

- **Results:** This section outlines the general process used to judge the two methods performance, including the error norm being used. It then goes into the results from each of the individual MATLAB cases, before transitioning to the results from the ARDRA cases.

- **Conclusion:** This section summarizes the characteristic and overall performance of each method, as well as outlines any trends observed in the test cases. It highlights several improvements that could be made to later implementations, as well as outlines potential work to be done in the future.

Chapter Two
BACKGROUND

2.1 Neutron Transport

A key component of the nuclear sciences is the understanding of how subatomic particles such as neutrons behave when traveling through a particular environment, a concept typically referred to as radiation transport. Going forward, this paper will specifically focus on the transport of neutrons, as each type of particle has slightly different physics governing their interactions with the world around them. It should be noted, however, that many of the concepts discussed in this paper could be expanded upon to apply to other forms of radiation transport.

2.1.1 Neutron Transport Equation

The 'life' of a neutron is modeled by the neutron transport equation (NTE), a linear integro-differential equation based on the Boltzmann Transport Equation, an equation originally used to describe the evolution of particle density distributions for non-equilibrium gases [8]. At its core, the neutron transport equation is essentially a balance equation that describes the population of neutrons at a particular place in space, with loss terms on the left and gains on the right.

The full time, energy, angle, and space dependent form of the neutron transport equation for an external source in non-multiplying media can be written as follows:

$$\begin{aligned}\frac{1}{v(E)}\frac{\partial\psi(\vec{r},E,\hat{\Omega},t)}{\partial t} + \hat{\Omega}\cdot\vec{\nabla}\psi(\vec{r},E,\hat{\Omega},t) + \Sigma_t(\vec{r},E)\psi(\vec{r},E,\hat{\Omega},t) = \\ \int_{4\pi} d\Omega' \int_0^\infty dE'\Sigma_s(\vec{r},E'\to E,\hat{\Omega}'\to\hat{\Omega})\psi(\vec{r},E',\hat{\Omega}',t) + q(\vec{r},E,\hat{\Omega},t)\end{aligned} \tag{2.1}$$

where $v(E)$ is the velocity of the neutron as a function of energy, $\vec{r}$ is one's spatial coordinates, $\hat{\Omega}$ is the direction being traveled, ψ is the angular flux, t is time, Σ_t is the total absorption cross, Σ_s is the double differential scattering cross section, and q is the external point source [4].

In its most complicated form, the transport equation contains seven independent variables: three spatial dimensions $r = x, y, z$, two angular dimensions $\hat{\Omega} = \theta, \phi$ as well as both time t and energy E, and can involve material properties that are coupled with the neutron flux through temperature creating a complicated, multiphysics environment. Because of its inherent nature, in all but the most simple of cases, true analytical solutions to the transport equation do not exist, necessitating the use of numerical approximations instead to find a solution.

As is the case with all numerical methods, accuracy must be weighed against the computational cost of obtaining said solution. Key factors to consider when performing numerical analysis are the rate at which the method converges and the way in which error propagates throughout the solution, as well as the possibility of parallelization and potential optimization. With this in mind, a compromise must be made between run time and the accuracy of the solution. All methods have strengths and weaknesses, and it is up to the user to choose the best approach for the problem at hand [4].

Most numerical methods used in neutron transport fall under one of two different categories: stochastic methods such Monte Carlo in which pseudo-random numbers and interaction probabilities are used in tangent with one another to obtain a statistically sound solution without having to directly solve the transport equation, and deterministic methods such as discrete ordinates or the method of characteristics which solve the transport equation directly

In stochastic methods such as Monte Carlo, the transport equation is never actually used, and one could argue they are not actually approximations of the neutron transport equation but rather an entirely alternative approach to the characterization of a nuclear system. While the transport equation describes the statistical average of the particles in the system, Monte Carlo methods try to build an average of desired quantities by simulating the life of many individual neutrons. However, because in a system like a nuclear reactor there exist a near countless number of particles at any given moment, it is impossible to truly model every individual one of them. Therefore, the accuracy of a Monte Carlo simulation is typically limited by the available computer resources [9]. The advantage to a method like Monte Carlo is its general robustness and ability to produce very accurate solutions under a wide range of applications, as well as its exact modeling of complex geometries and continuous energy representation. The biggest drawback to them, however, is that in order to achieve a statistically accurate solution, they often require millions or even billions of individual particles to be modeled, which is both computationally expensive and memory intensive. If an insufficient number of particles is used, then it is possible for certain regions to not be sampled enough for meaningful results. The accuracy of the solution is dependent on the number of particles that are able to reach the given area. While certain variance reduction techniques can be employed to increase this number, in general, Monte Carlo scales by $\frac{1}{\sqrt{N}}$, meaning that to decrease the error by an order of magnitude, one must use 2 orders of magnitude more particles. While this poor scaling is offset slightly by Monte Carlo's high degree of parallelization, it is still considered to be computationally expensive and slow. Some examples of commonly used Monte Carlo codes include MCNP, Serpent, and Geant4.

Deterministic methods, on the other hand, typically work in a more conventional fashion, choosing instead to discretize the transport equation in space, angle, energy, and/or time and solve the resulting system of equations directly. A variety of different numerical techniques and discretization schemes have been used over the years, with the discrete ordinates ($\mathrm{S_n}$) method emerging as the dominant means of obtaining a numerical solution to the integro-differential form of the neutron transport equation (as opposed to the integral form) [4]. Discrete ordinates methods are characterized by the evaluation of the transport equation in discrete, angular directions. Some of the advantages of the discrete ordinates method are its computational efficiency and minimal memory requirements. However, when applied to problems consisting of multiple spatial dimensions, numerical artifacts known as ray effects can appear in the solution. These are discussed in further detail in the following sections.

2.1.2 The Method of Discrete Ordinates

Beginning with Equation 2.1, for the sake of simplicity, both time and and energy dependence can be dropped, allowing the neutron transport equation to be written in the form

$$\hat{\Omega} \cdot \overrightarrow{\nabla} \psi(\vec{r}, \hat{\Omega}) + \Sigma_t(\vec{r})\psi(\vec{r}, \hat{\Omega}) = \int_{4\pi} d\hat{\Omega}' \Sigma_s(\vec{r}, \hat{\Omega}' \to \hat{\Omega})\psi(\vec{r}, \hat{\Omega}') + q(\vec{r}, \hat{\Omega}) \tag{2.2}$$

This is done primarily for the sake of visual clarity, as the derivation for the spatial and angular treatment of the steady state monoenergetic form does not fundamentally differ from the time-dependent, multigroup form.

The next step is to discretize the angular quantities $\Sigma_s(\vec{r}, \hat{\Omega}' \to \hat{\Omega})$ and $\psi(\vec{r}, \hat{\Omega}')$ in the scattering term. This is done by expanding the angular flux as infinite series

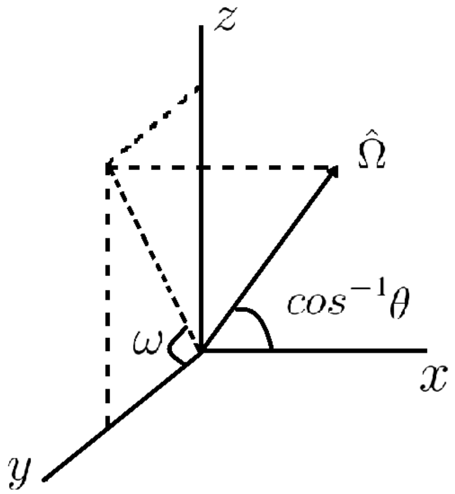


Figure 2.1 Coordinate system used in defining spherical harmonics.

of spherical harmonics:

$$\psi(\vec{r},\hat{\Omega}) = \sum_{\ell=0}^{\infty}\sum_{m=-\ell}^{\ell} Y_{\ell,m}(\hat{\Omega})\phi_{\ell,m}(\vec{r}) \tag{2.3}$$

and

$$\phi_{\ell,m}(\vec{r}) = \int_{4\pi} d\hat{\Omega} Y^*_{\ell,m}(\hat{\Omega})\psi(\vec{r},\hat{\Omega}) \tag{2.4}$$

where $Y_{\ell,m}(\hat{\Omega})$ are the spherical harmonics corresponding with order ℓ,m defined by the equation

$$Y_{\ell,m}(\hat{\Omega}) = \sqrt{\frac{(2\ell+1)(\ell-m)!}{4\pi(\ell+m)!}} P_{\ell,m}(\theta)e^{i\omega m} \tag{2.5}$$

with θ and ω defined in Figure 2.1. The $Y^*_{\ell,m}(\hat{\Omega})$ in Equation 2.4 denotes the complex conjugate of $Y_{\ell,m}(\hat{\Omega})$.

The scattering cross section can be expanded using Legendre polynomials:

$$\Sigma_s(\vec{r},\mu) = \sum_{k=0}^{\infty} \frac{2k+1}{4\pi} P_k(\mu)\Sigma_{s,k}(\vec{r}) \tag{2.6}$$

where

$$\mu = \hat{\Omega}' \cdot \hat{\Omega} \tag{2.7}$$

and $P_k(\mu)$ its associated Legendre polynomial.

Plugging Equations 2.3 and 2.6 back into the scattering term from Equation 2.2 yields

$$\begin{split}\int_{4\pi} d\hat{\Omega}'\Sigma_s(\vec{r},\hat{\Omega}' \rightarrow \hat{\Omega})\psi(\vec{r},\hat{\Omega}') = \\ \int_{4\pi} d\hat{\Omega}' \sum_{k=0}^{\infty} P_k(\mu)\Sigma_{s,k}(\vec{r}) \sum_{\ell=0}^{\infty}\sum_{m=-\ell}^{\ell} Y_{\ell,m}(\hat{\Omega})\phi_{\ell,m}(\vec{r})\end{split} \tag{2.8}$$

By assuming that all nuclear interactions are azimuthally symmetric, one may apply the spherical harmonic addition theorem to write Equation 2.5 as:

$$Y_{\ell,m}(\hat{\Omega}) = \sqrt{\frac{(2\ell+1)}{4\pi}} P_{\ell,m}(cos(\theta)) \tag{2.9}$$

allowing one to simplify scattering term to:

$$\int_{4\pi} d\hat{\Omega}'\Sigma_s(\vec{r},\hat{\Omega}' \rightarrow \hat{\Omega})\psi(\vec{r},\hat{\Omega}') = \sum_{k=0}^{\infty} \Sigma_{s,k}(\vec{r}) \sum_{n=-k}^{k} Y_{k,n}(\hat{\Omega})\phi_{k,n}(\vec{r}) \tag{2.10}$$

Equation 2.2 can now be formulated using the newly derived scattering term:

$$\hat{\Omega} \cdot \overrightarrow{\nabla}\psi(\vec{r},\hat{\Omega}) + \Sigma_t(\vec{r})\psi(\vec{r},\hat{\Omega}) = \sum_{k=0}^{\infty} \Sigma_{s,k}(\vec{r}) \sum_{n=-k}^{k} Y_{k,n}(\hat{\Omega})\phi_{k,n}(\vec{r}) + q(\vec{r},\hat{\Omega}) \tag{2.11}$$

This can be rewritten to include the iteration index p and angular index m:

$$\hat{\Omega}_m \cdot \overrightarrow{\nabla} \psi^{(p+1)}(\vec{r}, \hat{\Omega}_m) + \Sigma_t(\vec{r}) \psi^{(p)}(\vec{r}, \hat{\Omega}_m) = \\ \sum_{k=0}^{\infty} \Sigma_{s,k}(\vec{r}) \sum_{n=-k}^{k} Y_{k,n}(\hat{\Omega}_m) \phi_{k,n}^{(p)}(\vec{r}) + q(\vec{r}, \hat{\Omega}_m) \tag{2.12}$$

The general idea behind the method of discrete ordinates is to devise a set of equations for all directions in one's angular quadrature at every zone in the spatial mesh and to solve them all simultaneously. This is done by devising a discrete set of directions $\hat{\Omega}_m$ with corresponding weights w_m such that they constitute a quadrature set across all angles.

An initial distribution is guessed for the angular flux distribution. This is plugged into the right side of Equation 2.12, using the newly defined numerical quadrature to solve for $\phi_{k,n}^{(p)}(\vec{r})$, and solving for $\psi^{(p+1)}(\vec{r}, \Omega_i)$ on the left. This process is then repeated using the previous iterations angular flux to update the scattering source in a process known as source iteration.

The quantity $\psi^{(p)}$ can be thought of as the angular flux from particles that have had at most p number of collisions. The first solution, $\psi^{(1)}$, is known as the uncollided flux, as it represents the angular flux due exclusively to the external source $q(\vec{r}, \hat{\Omega})$. This can then be plugged back into the scattering source to obtain the second solution, $\psi^{(2)}$ - the number of particles that have had at most one collision.

In general, one can define the total angular flux as

$$\psi = \sum_{i=0}^{\infty} \psi_i \tag{2.13}$$

where ψ_i is the angular flux from particles that have had i collisions and can be found using

$$\psi_i = \psi^i - \psi^{i-1} \tag{2.14}$$

This process of iterating upon and updating the scattering source is repeated until there is little to no change between ψ^i and ψ^{i+1}, at which point the solution has converged. The number of iterations required to reach convergence varies from problem to problem. Because each additional iteration can be thought of as the modeling of an additional scattering event, problems with significant quantities of light elements like hydrogen in which neutrons are likely to scatter many times before being lost can require many iterations before converging, leading to slow calculations. In addition to this, the process of solving for the angular flux using standard discrete-ordinate schemes often leads to something known as ray effects occurring throughout the solution.

2.1.3 Ray Effects

Ray effects are non-physical artifacts caused by the discretization of angle in the discrete ordinates method [4, 7, 10, 11, 13]. While the neutron transport equation expresses the angle a particle travels continuously, $\mathrm{S_n}$ approximations are forced to represent this using only a discrete set of directions. In doing so, it forces the solution to propagate along these directions, leading to unrealistic concentrations/absences in the solution, often giving the appearance of rays, an example of which can be seen in Figure 2.2a.

This is one of the most significant drawbacks to the method of discrete ordinates, as they can cause substantial errors in the local solution, particularly in areas between the individual directions being used where the artificial troughs are at their worst [11]. While ray effects tend to distort local error distributions, it should be noted that global quantities such as leakage or total absorptions remain accurate.

Ray effects tend to be worst when using a small source relative to the problem domain – for example, any type of problem where the source can be approximated using a point source, such as when trying to model the skyshine effect or any sort of atmospheric transport.

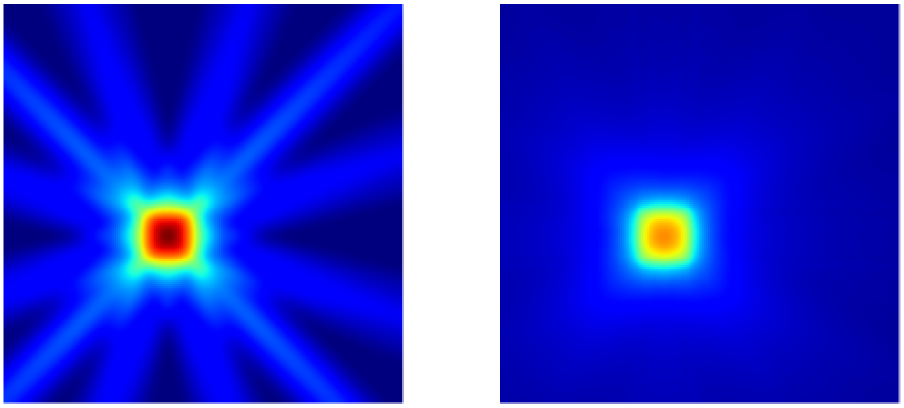
2.1.4 Ray Effect Mitigation

Despite its drawbacks, the $\mathrm{S_n}$ method remains one of the most popular numerical techniques to solve the transport equation. Because of this, significant effect has been made in developing techniques to reduce the impact ray effects have on the final solution [7, 10, 11, 13].

The easiest approach to reducing ray effects is to increase the number of directions used in the approximation [4, 7, 11]. The effect this can have can be seen in Figures 2.2a and 2.2b, where the image on the right was computed using a higher number of ordinates. The problem with this approach, however, is that it tends to be both inefficient and ineffective, as it greatly increases the computational cost of the simulation while achieving diminishing returns.

The unfortunate reality of the discrete ordinates method is that ray effects are an inherent part of its nature arising from the lack of rotational invariance in the transport leakage operator after being discretized. The only way to truly eliminate ray effects is to make the method in some way equivalent to the P_n equations, something that to date has researchers have been unable to do in a practical manner [10]. These methods, referred to as P_n-equivalent fictitious-source methods, work by introducing an additional source term to the right side of the S_n equations [11]. These methods eventually developed into what is known as the modulated P_n-equivalent fictitious-source method, which solves the unacceptably slow convergence of the fictitious source by multiplying the fictitious source by a constant between 0 and 1. When set to zero, one has the uncorrected S_n method. When set to 1, one entirely eliminates ray effects, but cannot converge. The idea is to set the value somewhere in between where one can balance increased convergence time with ray effect mitigation.

There are additional downsides to using the P_n method. Although the P_n method does not suffer from ray effects, it is still liable of producing inaccurate solutions due to the presence of Gibbs oscillations in the spectral expansion of the angular behaviour [12]. Furthermore, in 1D, both the P_n and S_n are equivalent, so the P_n method does not converge faster than the S_n method. While the S_n method contains artifacts which are harder to ignore than the incorrect wave speeds produced by low



(a) Unrealistic S_n solution **(b)** Improved S_n solution

Figure 2.2 A comparison of two solutions to the same problem. The plot on the right shows the effect of increasing the number of angles used. Images from [7].

order $\mathrm{P_n}$ methods, the latter methods are not without their own problems.

Although the use of a fictitious source was among the first attempts at reducing ray effects, other techniques would be developed, like increasing the interactions between different ordinate directions, as well as the use of different quadrature sets [10, 11, 13, 14]. This paper focuses on an alternative approach using what is referred to as the first-collision source treatment method.

2.1.5 First Collision Source Treatment

In most problems, the uncollided flux is the greatest contributor to the formation of ray effects. This is because it tends to be the least isotropic component of the angular flux, as well as the most orientation dependent. The local contribution of the uncollided flux is nonzero only in discrete directions – i.e. in the direction of the source. Further scattering tends to smooth the solution in angle because the angular redistribution operator is described by an integral in angle [7]. This means that minimizing ray effects in the uncollided flux will yield the greatest results, followed by the first-collided, then the second, and so on.

The first-collision source treatment method focuses on minimizing the effect the uncollided flux has on the propagation of ray effects in later source iterations. It does this by decomposing the angular flux into two components: the uncollided ψ_0 and collided ψ_c flux, and solving for them separately:

$$\psi = \sum_{i=0}^{\infty} \psi_i = \psi_0 + \sum_{i=1}^{\infty} \psi_i = \psi_0 + \psi_c \tag{2.15}$$

The uncollided flux is obtained by setting the scattering source in Equation **??** equal to zero, simplifying Equation **??** to

$$\hat{\Omega}_n \cdot \overrightarrow{\nabla} \psi_0(\vec{r}, \hat{\Omega}_n) + \Sigma_t(\vec{r}) \psi_0(\vec{r}, \hat{\Omega}_n) = q(\vec{r}, \hat{\Omega}_n) \tag{2.16}$$

(note that $\psi_0 = \psi^1$).

This can be solved for in variety of ways, one of which is through solving the integral form of the neutron transport equation using ray tracing. The advantage of using ray tracing in this sort of situation is that while computationally expensive, it yields more accurate results that exhibit no ray effects [7].

Once a ray effect free solution to the uncollided flux has been found, it can be plugged into Equation **??** and used in a standard source iteration approach. In the event where one has a purely absorbing medium, $\psi_c = 0$ and $\psi = \psi_0$, removing any need for source iteration.

Chapter Three

RAY TRACING METHODS

3.1 Derivation from Integral Form

Ray tracing methods used to solve the uncollided flux are based on the integral form of the neutron transport equation. Writing out the neutron transport equation with both time and energy variables suppressed yields the following equation [4, 23]:

$$\hat{\Omega} \cdot \overrightarrow{\nabla} \psi(\vec{r}, \hat{\Omega}) + \Sigma_t(\vec{r})\psi(\vec{r}, \hat{\Omega}) = \int \Sigma_s(\vec{r}, \hat{\Omega} \cdot \hat{\Omega}')\psi(\vec{r}, \hat{\Omega}')d\hat{\Omega}' + q(\vec{r}, \hat{\Omega}) \tag{3.1}$$

Because the quantity being solved for is the uncollided flux - that is, the flux of particles coming directly from the source without having interacted with anything, the contribution from particles that have scattered is zero by definition, meaning:

$$\int \Sigma_s(\vec{r}, \hat{\Omega} \cdot \hat{\Omega}')\psi(\vec{r}, \hat{\Omega}')d\hat{\Omega}' = 0, \tag{3.2}$$

allowing one to simplify Equation 3.1 to

$$\hat{\Omega} \cdot \overrightarrow{\nabla} \psi(\vec{r}, \hat{\Omega}) + \Sigma_t(\vec{r})\psi(\vec{r}, \hat{\Omega}) = q(\vec{r}, \hat{\Omega}) \tag{3.3}$$

Because the streaming operator $\hat{\Omega} \cdot \overrightarrow{\nabla}$ is just a directional derivative along the direction of neutron travel, Equation (3.3) can be rewritten as a first order differential equation:

$$\frac{d}{dR}\psi(\vec{r_s}+R\hat{\Omega},\hat{\Omega})+\Sigma_t(\vec{r_s}+R\hat{\Omega},\hat{\Omega})\psi(\vec{r_s}+R\hat{\Omega},\hat{\Omega})=q(\vec{r_s}+R\hat{\Omega},\hat{\Omega}) \tag{3.4}$$

where R is the distance the neutron traveled from its source location $\vec{r_s}$ to its current location $\vec{r}$:

$$R=||\vec{r}-\vec{r_s}||. \tag{3.5}$$

By applying the integrating factor

$$e^{\int_0^R \Sigma_t(\vec{r_s}+R'\hat{\Omega})dR'} \tag{3.6}$$

to Equation 3.4, one obtains

$$\frac{d}{dR}\left[\psi(\vec{r_s}+R\hat{\Omega},\hat{\Omega})e^{\int_0^R \Sigma_t(\vec{r_s}+R'\hat{\Omega})dR'}\right]=q(\vec{r_s}+R\hat{\Omega},\hat{\Omega})\left[e^{\int_0^R \Sigma_t(\vec{r_s}+R'\hat{\Omega})dR'}\right]. \tag{3.7}$$

Integrating both sides and rearranging terms yields:

$$\psi(\vec{r_s}+R\hat{\Omega},\hat{\Omega})=\int_{-\infty}^{R} q(\vec{r_s}+R\hat{\Omega},\hat{\Omega})\left[e^{-\int_{R'}^{R} \Sigma_t(\vec{r_s}+R''\hat{\Omega})dR''}\right]dR' \tag{3.8}$$

which can then be simplified by defining

$$\vec{r}=\vec{r_s}+R\hat{\Omega} \tag{3.9}$$

and changing the sign of the variables such that the integrals become 0 to ∞ and 0 to R:

$$\psi(\vec{r},\hat{\Omega})=\int_{0}^{\infty} q(\vec{r}-R'\hat{\Omega},\hat{\Omega})\left[e^{-\int_{0}^{R} \Sigma_t(\vec{r}-R''\hat{\Omega})dR''}\right]dR' \tag{3.10}$$

The exponent of the attenuation factor:

$$\int_0^R \Sigma_t(\vec{r} - R''\hat{\Omega})dR'' \tag{3.11}$$

represents the number of mean free paths along the straight line between $\vec{r}$ and $\vec{r} - R'\hat{\Omega}$ and is typically referred to as the optical depth, or optical path length. It is often denoted as $\tau(\vec{r} - R'\hat{\Omega} \rightarrow \vec{r})$:

$$\begin{aligned}\tau(\vec{r} - R'\hat{\Omega} \rightarrow \vec{r}) &= \int_0^R \Sigma_t(\vec{r} - R'\hat{\Omega})dR' \\ &= R'\bar{\Sigma}_t(\vec{r} - R'\hat{\Omega} \rightarrow \vec{r}),\end{aligned} \tag{3.12}$$

where $\bar{\Sigma}_t(\vec{r} - R'\hat{\Omega} \rightarrow \vec{r})$ is the average cross section along the path from $\vec{r} - R'\hat{\Omega}$ to $\vec{r}$. Equation 3.13 can be solved in a variety of ways, depending on how one's cross section is defined. If a continuous function is known, it is a relatively simple task to numerically integrate that function from 0 to R. However, if the cross section is defined by the mesh itself, as is often the case, it can be solved for using ray tracing by summing the path length through each zone and multiplying it by the zonal cross section:

$$\begin{aligned}\tau(\vec{r} - R\hat{\Omega} \rightarrow \vec{r}) &= \int_0^R \Sigma_t(\vec{r} - R'\hat{\Omega})dR' \\ &= \sum_{i=0}^{I} R_i\Sigma_{t,i}\end{aligned} \tag{3.13}$$

Rewriting Equation 3.10 in terms of $\tau(\vec{r} - R'\hat{\Omega} \rightarrow \vec{r})$ gives:

$$\psi(\vec{r}, \hat{\Omega}) = \int_0^\infty q(\vec{r} - R'\hat{\Omega}, \hat{\Omega}) \left[e^{-\tau(\vec{r} - R'\hat{\Omega} \rightarrow \vec{r})}\right] dR' \tag{3.14}$$

If we assume we have an isotropic point source, the source term can be rewritten as

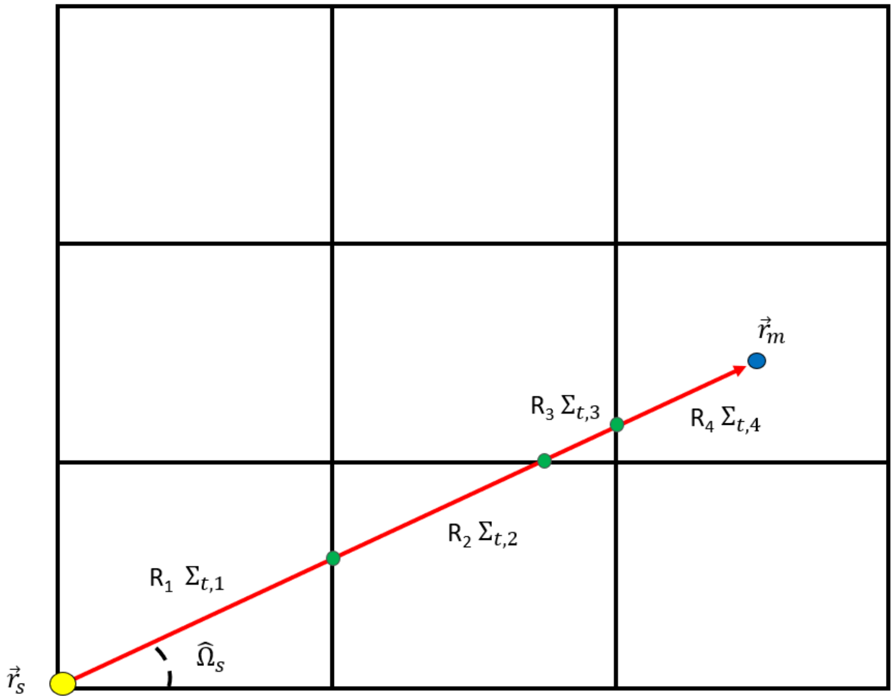


Figure 3.1 An illustration of the ray trace used to obtain the optical depth along a ray's path from a point source to a point located inside a zone.

$$q(\vec{r},\hat{\Omega}) = \frac{q}{4\pi R^2}\delta(R)\delta(\hat{\Omega}) \tag{3.15}$$

as at any given point in space, aside from the source itself, all particles are streaming from a single direction. Here, q is the total source strength.

Although outside the scope of this paper, if one's problem cannot be approximated using a single point source, an alternative approach is to break the source up into a discrete set of point sources with some equivalent total emission strength. The contribution from each individual point source would then be modeled separately before being recombined to get the total uncollided flux. Although problem dependent, this process can be very computationally expensive, rendering the first collision source treatment method largely impractical for those types of problems [7]. It is also worth noting that while approximating non-point sources as a single point typically results in some additional error, particularly near the source, breaking one's source down into multiple point sources amplifies this error significantly.

Substituting Equation 3.15 into Equation 3.14 and simplifying yields:

$$\psi(\vec{r},\hat{\Omega}) = \frac{q}{4\pi}\int \frac{e^{-\tau(\vec{r}_s\to\vec{r})}}{(R')^2}\delta(R')\delta(\hat{\Omega})dR' \tag{3.16}$$

$$\psi(\vec{r},\hat{\Omega}) = \frac{q}{4\pi}\frac{e^{-\tau(\vec{r}_s\to\vec{r})}}{R^2}\delta(\hat{\Omega}) \tag{3.17}$$

The scalar flux can be found by integrating Equation 3.17 in angle over 4π:

$$\begin{aligned}\phi(\vec{r}) &= \int_{4\pi}\psi(\vec{r},\hat{\Omega})d\hat{\Omega} \\ &= \frac{qe^{-\tau(\vec{r}_s\to\vec{r})}}{4\pi R^2} \qquad (3.18)\end{aligned}$$

3.2 Interior Point Method

The most conventional approach to solving Equation 3.18 within a mesh is to apply a numerical quadrature over the zone, as seen in Figure 3.2. By solving for the scalar flux at each quadrature point, one can numerically integrate across the zone and obtain an average scalar flux:

$$\bar{\phi}(\vec{r}) = \frac{1}{V}\int_V \phi(\vec{r})dV \tag{3.19}$$

where V is the volume of zone z. A variety of different methods exist that can be used to numerically integrate across this volume to find the zonal average flux. This paper specifically uses a Gauss-Legendre quadrature for the majority of its tests due to its high level of accuracy and common implementation (although the uniform cross section case does contain a comparison between Gauss-Legendre and Gauss-Lobatto).

One may rewrite Equations 3.18 and 3.19 in the discrete forms:

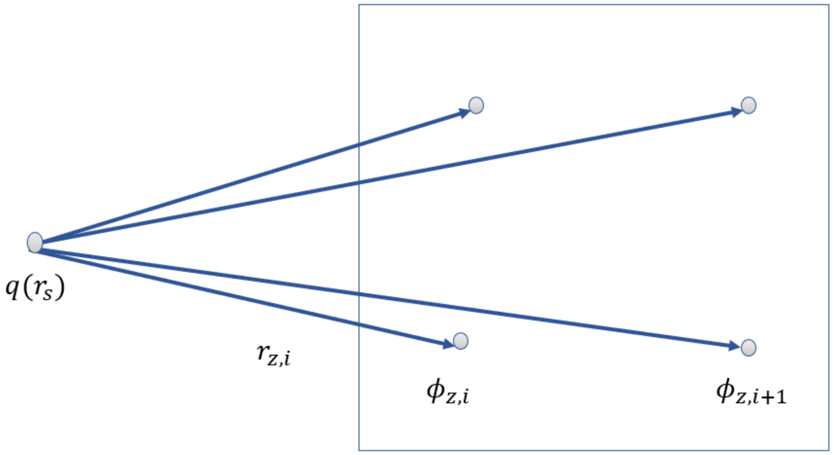


Figure 3.2 The interior point method works by calculating the scalar flux on quadrature of points inside a zone and then integrating across them to obtain the zone-average flux.

$$\phi_{z,i} = \frac{qe^{-\tau_{z,i}}}{4\pi R_{z,i}^2} \tag{3.20}$$

$$\bar{\phi}_z = \frac{1}{V_z}\sum_{i=1}^{I} w_i \phi_{z,i} \tag{3.21}$$

using an I point quadrature where i is the quadrature index in zone z.

Some of the advantages to using the interior point method include its ease of implementation, as well as its general ability to model everything to some accuracy without breaking. This robustness is arguably the interior point method's greatest strength.

Some the downsides associated with the interior point method include it not being locally or globally conservative, meaning that the number of particles sourced into the problem does not equal the total number absorbed plus the leakage. Another downside to this method is that it cannot directly obtain the leakage as a consequence of already calculated values. One must either use alternative methods, such as the surface method, or assume particle conservation and calculate it by summing the absorption and source terms.

The location of the interior point method's quadrature points is both a strength and a weakness. Because they are located across the interior of the zone, the method is able to capture the internal behavior of the cross section. This is important if one is unable to align their spatial mesh with their material interfaces, as well as when one has a cross section defined by a function (see the Radially-Symmetric Smoothly-Varying Cross Section test case for an example of this). This internal positioning is also one of the biggest drawbacks to the method, however, as it severely limits the potential for optimization through sharing rays between neighboring zones. Only quadrature points that lie on the surface of the zone can be shared, meaning that quadrature rules like Gauss-Lobatto will only share its endpoints, while rules like Gauss-Legendre are unable to share any.

3.3 Surface Method

The surface method is a new ray tracing method used to solve for the uncollided flux. Although very similar to the interior point method, it differs from it in several key ways, the primary being its core assumption of particle conservation. That is, given any differential volume dV, through application of the divergence theorem, one must satisfy the conservation equation

$$0 = N_{src} - N_{abs} - N_{leak} \tag{3.22}$$

where N_{src} is the number of particles being sourced into the volume from external sources, N_{abs} is the number of particles being lost in the volume due to absorptions, and N_{leak} is the net number of particles entering/leaving the volume. This equation holds true both locally for every zone in a mesh, as well as globally for the entire problem domain, whether finite or infinite.

For a differential volume dV, N_{leak} can be calculated by computing the net current across the volume's surface. Beginning with the definition of a current vector:

$$\vec{J}(\vec{r}) = \int \hat{\Omega}\psi(\vec{r}, \hat{\Omega}) d\hat{\Omega} \tag{3.23}$$

one can obtain the net number of particles crossing a surface per unit of space in direction n by taking the dot product of the surface normal $\hat{n}$ and the current vector

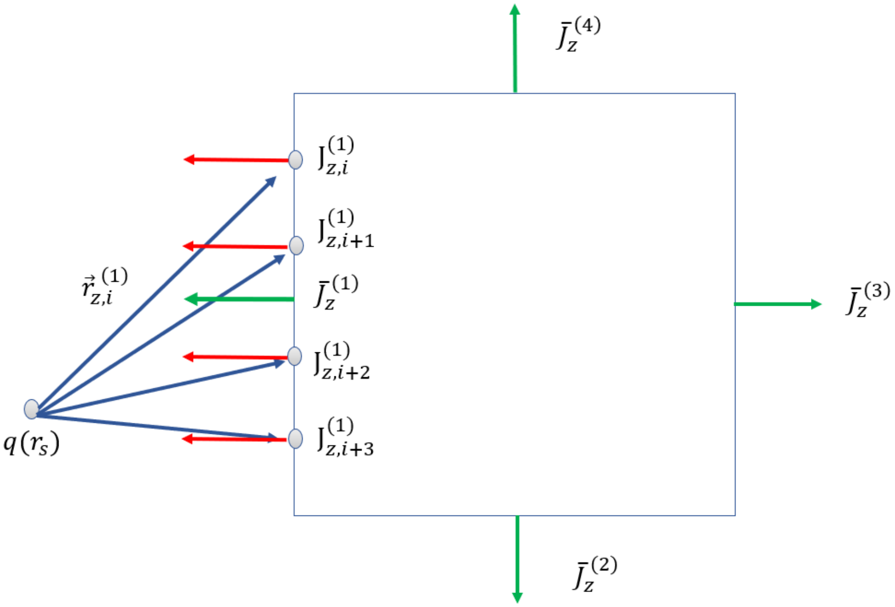


Figure 3.3 The surface method works by calculating the angular flux at a set of points on a face and then integrating them to obtain the current through that face.

$\vec{J}$:

$$J_n(\vec{r}) = \hat{n} \cdot \vec{J}(\vec{r}) = \int (\hat{n} \cdot \hat{\Omega})\psi(\vec{r}, \hat{\Omega})d\hat{\Omega} \tag{3.24}$$

By plugging Equation 3.17 in for $\psi(\vec{r}, \hat{\Omega})$, one gets

$$J_n(\vec{r}) = \int_{4\pi} (\hat{n} \cdot \hat{\Omega})\frac{q(\vec{r}, \hat{\Omega})e^{-\tau(\vec{r}_s \to \vec{r})}}{R^2}d\hat{\Omega} \tag{3.25}$$

Just as was done for interior point method, because the source is assumed to be an isotropic point source, q can be described as a delta function in angle (as particles will only travel from the direction of the source Ω_s), allowing one to drop the integral and rewrite the equation as

$$J_n(\vec{r}) = (\hat{n} \cdot \hat{\Omega}_s)\frac{qe^{-\tau(\vec{r}_s \to \vec{r})}}{4\pi R^2} \tag{3.26}$$

It should be noted that the source being isotropic allowed the angular dependency to be dropped from the source term. This simplification does not fundamentally change the following equations, as long the source remains a point source.

To obtain the net number of particles crossing the surface of dV, simply take the surface integral of Equation 3.26 across the surface S:

$$N_z(\vec{r}) = \int_S J_z(\vec{r})dS = \int_S (\hat{n} \cdot \hat{\Omega}_s)q(\vec{r})\frac{e^{-\tau(\vec{r}_s \to \vec{r})}}{R^2}dS \tag{3.27}$$

This integral can be solved numerically in a variety of different ways, depending on how one defines their spatial mesh. In the case of a simple Cartesian mesh as shown in Figure 3.3, one can divide the surface integral amongst the different faces and solve for each face independently before summing them together to get N_{leak}:

$$N_z^{(f)} = \bar{J}_z^{(f)}\Delta x_f = \sum_{i=1}^{I} W_i J_{z,i}^{(f)} \tag{3.28}$$

$$J_{z,i}^{(f)} = (\hat{n}_f \cdot \hat{\Omega}_{s,z,i}^{(f)})\frac{qe^{-\tau_{z,i}^{(f)}}}{4\pi(R_{z,i}^{(f)})^2} \tag{3.29}$$

$$N_{leak,z} = \sum_{f=1}^{F} N_z^{(f)} \tag{3.30}$$

where z is the zone index, f is the face index, and i is the quadrature index. Just as was done in the interior point method, a Gaussian quadrature is being used to numerically integrate across the surface of each face.

After finding N_{leak} for zone m, one can find N_{abs} by plugging N_{leak} back into Equation 3.22. For all non-source zones, the local contribution from external sources N_{src} is zero, meaning $N_{abs} = N_{leak}$. If the zone does contain the source, or one of its faces is in contact with the source, the number of particles being sourced into the zone must be accounted for. This will depend on whether or not the source is actually inside the zone or on its surface. If on the surface, the source will actually be emitting particles into multiple zones and so will need to be split.

The flux can then be indirectly solved for by defining one's absorption rate as

$$N_{abs}(\vec{r}) = \int_V \phi(\vec{r})\Sigma_a(\vec{r})dV \tag{3.31}$$

and solving for ϕ. Assuming the cross section throughout the interior of the zone is constant, one may simplify the equation to

$$\bar{\phi}_z = \frac{N_{abs,z}}{\Sigma_{a,z}V_z} \tag{3.32}$$

where V_z is the volume of zone z.

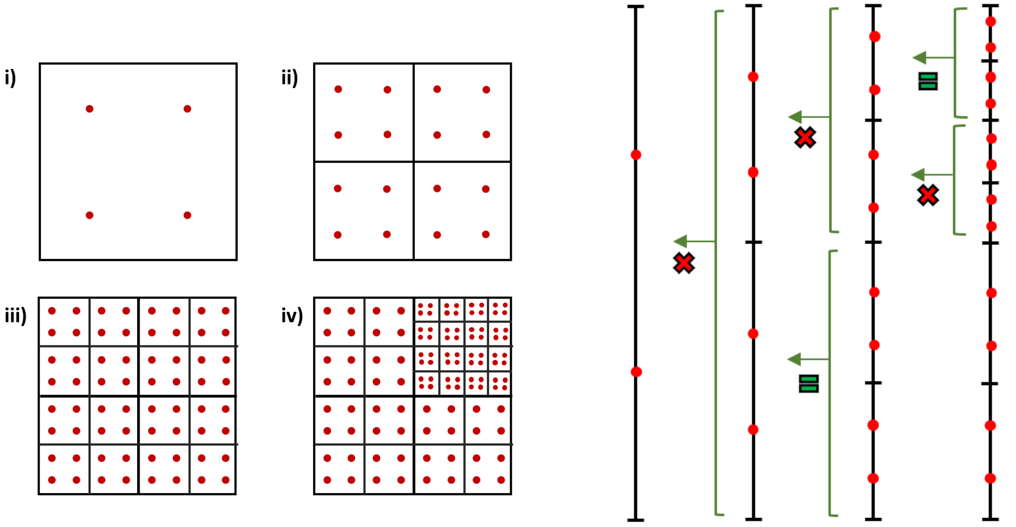
One of the main advantages of the surface method is its inherently conservative nature. Modern $\mathrm{S_n}$ methodologies are typically conservative if one doesn't use a first collision source treatment. However, up until now, there has been no way to treat the uncollided flux that maintains particle conservation. While being conservative does not mean there is no error, many errors will cancel out when looking at global quantities. This is of particular value in things like criticality problems where total neutron populations is of upmost import. If the method is not conservative, nothing is known about how global quantities measure up to their true values - whether the total absorption rate is accurate and its just the distribution that's wrong, or vice versa.

Another advantage the surface method has is in regards to where its quadrature points are located. By being on the surface of a zone rather than the interior, quadrature points may be shared between zones, cutting down on the total number of calculations that need to be performed. While the exact number of times a ray can be shared depends on the quadrature being used, as well as what dimension the problem is in (2D vs. 3D), at a minimum, every point can be shared at least once (except for points that lie on the edges of the mesh).

3.4 Adaptive Variants

Both the interior point and surface methods are capable of using adaptive refinement to improve the accuracy of their solution, typically at the cost of significantly greater computation time. This is done by first solving for the solution on a particular face/zone (depending on whether one is using the interior point or surface method). The face/zone is then subdivided, as can be seen in Figure 3.4 and the solution found within each division. These are then combined and compared against the previous solution. If the new solution exceeds a certain tolerance, the subdivisions are divided once again and the process repeated with the new subdivision being compared against the previous ones.

One advantage to this technique is that as the face/zone is subdivided multiple times, good regions are no longer need to be subdivided, allowing the algorithm to focus its efforts on problematic areas. This of particular importance when dealing with things like discontinuities in the solution created by angularly dependent sources, as well as unsmooth optical depth distributions as seen in the pin problem modeled later in this paper. By adaptively refining the solution, these problematic regions are localized to smaller and smaller portions of the zone, slowly reducing their impact on the overall solution. The downside to this, however, is that even under ideal circumstances where few zones require significant refinement, like in the case of a uniform cross section, it still increases the required number of rays by a minimum



(a) Interior point method

(b) Surface method

Figure 3.4 A demonstration of how adaptive refinement works for the interior point and surface methods in 2D.

of 3x for the surface method and 5x for the interior point method, as at least one refinement must always be made to determine if further refinement is necessary.

3.5 Implementation

Initial comparisons of the two different ray tracing techniques were done in MATLAB using a self-written ray tracing program (see appendix for a link to the code). This code could only solve for the uncollided flux (it was not an $\mathrm{S_n}$ code) and was limited to 2D, steady state, monoenergetic problems using relatively simple geometries. The initial goal of these tests was to gain a general sense of the performance of the two methods, including a rough idea of their convergence, error distribution, and behavior. As the majority of problems cannot be solved for analytically, a reference solution was obtained using an adaptive variant of the surface method using a Gauss-Kronrod quadrature.

After gaining a basic sense of the behavior and performance of the two methods in 2D, the surface method was implemented in ARDRA. This allowed full-scale, 3D problems to be modeled.

ARDRA is a deterministic ($\mathrm{S_n}$) neutron and gamma transport code developed by Lawrence Livermore National Laboratory [15]. Like all $\mathrm{S_n}$ codes, ARDRA's solutions suffer from ray effects. One of the ways in which it mitigates these is by using the first collision source treatment method. ARDRA currently supports two variations of the interior point method, one using a single, zone-centered ray as its quadrature, the other an adaptively-refining Lobatto quadrature.

In addition to the two pre-existing methods, three additional approaches have been introduced. The first one is an extended version of the interior point method employing a generalized I-point quadrature. In addition to this, two variations of the surface method have been incorporated: the first also utilizing general I-point quadrature, while the second is an adaptive variant with the same configuration discussed in the preceding section. Presently, all three newly implemented methods are confined to employing Gauss-Legendre quadratures.

Chapter Four

RESULTS - 2D CASES

Included in this paper are the results from three 2D test cases that were solved using MATLAB: a uniform cross section, a pin problem, and a problem using a smooth sinusoidal function to define its cross section. All three problems use a point source located at the origin with a strength of 1e6 particles per second. In combination, these three problems, while simple, cover a broad range of potential scenarios, with each of them being used to highlight a specific feature or characteristic of the two methods.

As the integrals in Equations 3.19 and 3.27 lack an analytical solution in most scenarios, a reference solution for each test case was computed using an adaptive variant of the surface method paired with a Gauss-Kronrod quadrature and a convergence criteria of 1e-14.

The local relative error was calculated using the equation:

$$E_{rel} = \frac{|N_{approx} - N_{true}|}{|N_{true}|} \tag{4.1}$$

where N_{approx} is the approximated absorption rate and N_{true} is the reference solution found using adaptive method.

Global error was determined by taking the L1 norm of the local absorption rate:

$$E_{rel,L1} = \frac{\sum |N_{approx} - N_{true}|}{\sum |N_{true}|} \tag{4.2}$$

In initial tests of the ray tracing methods, when computing the global error by considering values across the entire mesh, it was observed that variations in mesh resolution or quadrature order had minimal impact on global error trends. This observation appeared to contradict the overall error distribution and logical expectations. Upon closer examination, it was discovered that the error norm was highly influenced by a small number of zones exhibiting significantly higher errors than the rest. While the specific set of these "problematic" zones varied from case to case, they consistently included those nearest to the radiation source (a good example of a "bad" zone away from the source can be seen in the pin problem). Despite improvements in resolution benefiting the majority of zones, as is reflected in the overall error distribution, this select subset of zones experienced only marginal, if any, enhancements, and given their substantial difference in magnitude, effectively overshadowed all other data.

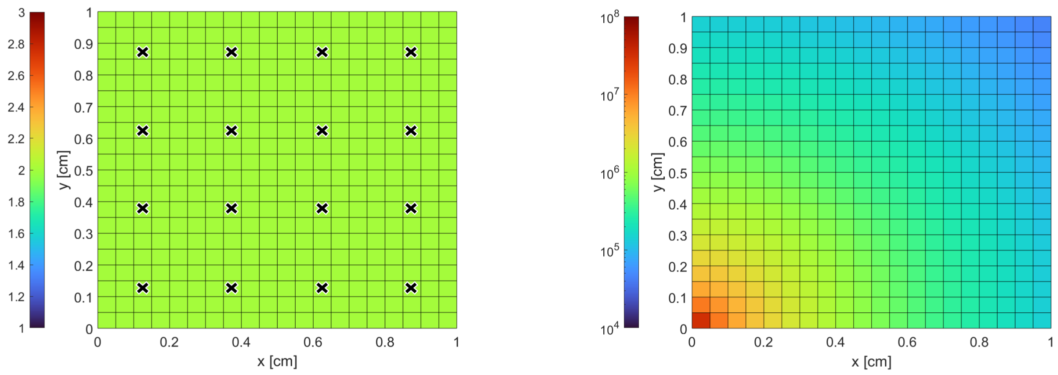
To help counteract this, a 4x4 set of watchpoints was used instead to find the L1 error. These special zones are always located with their zone centers at the same point in space, regardless of mesh resolution. Although this limits the mesh resolutions one can run with, it helps offset the global error from being dominated by a single point (assuming none of the watchpoints happen to fall on one of the "bad" zone). While the use of the watchpoints allowed some information about global trends to be gleaned, it still tended to be dominated by specific watchpoints, particularly the one closest to the source. The 4x4 set was chosen as a compromise between having more points to base the global error on and an increasingly restricted set of resolutions that could be used while still maintaining the watchpoints on zone centers.

While information on global error trends was still limited, the watchpoints also served a second purpose, as they allow one to examine local error trends at specific point in space while changing mesh resolutions. Because the error tends to behave differently depending on where one is in the problem (if the point is near the source or has crossed a material boundary), local error behavior is arguably of greater interest than global.

4.1 Uniform Case

The initial test problem used to compare the two ray tracing methods was a simple, uniform cross section of 2 cm^{-1} with a point source located at the origin. This setup, along with the obtained flux, can be seen in Figure 4.1. This problem was chosen for two primary reasons: first, its simplicity allows one to characterize the two methods behavior and performance under the most general of circumstances, and second, many real world problems can be described by large discrete regions consisting of uniform materials. An example of this can be seen when modeling basic radiation shielding problems, or even atmospheric transport problems. By characterizing the methods in a problem with no material interfaces, one can obtain the best idea of how the methods perform without problem specific features influencing global trends.

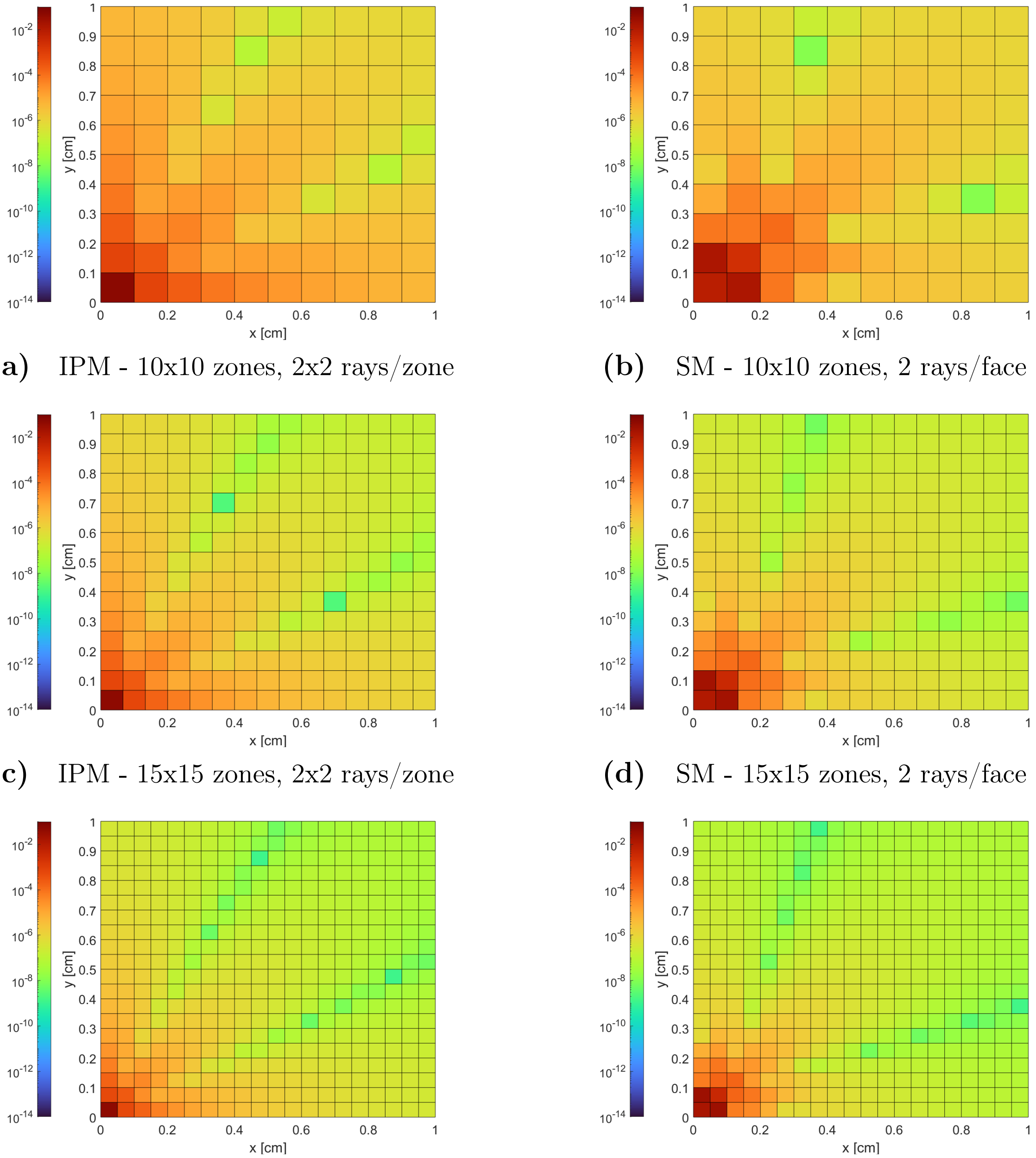
The plots in Figures 4.2 and 4.3 show a comparison of the relative error distributions for the interior point and surface methods for a variety of different mesh resolutions and quadrature orders. For both methods, error is concentrated near the source and tends to decrease as a function of distance from the source. This is because the $1/R^2$ behavior of Equations 3.19 and 3.27 creates a singularity in the flux/current at the the source. Interestingly, because of the way this particular problem was defined - that being a point source located on the corner of zone (1,1) and inline with the bottom and left edges of the mesh, the current through the bottom and left edges is zero. This is of particular importance in zone (1,1) for two reasons: first, it is the only zone with two zero-current faces. Because the surface method experiences additive errors from each of a zone's faces when it sums the currents to get the net absorption rate, there are only two sources of error when it does this sum for zone (1,1), instead of the three or four that all other zones experience. Second, because the current at every quadrature point along the two nearest faces to the source (the bottom and left face) is zero, the nearest non-zero quadrature point is significantly further from the source compared to the quadrature points of the interior point method. These two reasons are why the error for the interior point method in zone (1,1) is greater in all the plots than that for the surface method, and why, if one than examines either of adjacent zones (1,2) or (2,1), the error for the interior point method noticeably decreases while the surface method see's minimal improvement. Because the quadrature points on the faces shared between them and



(a) Cross section

(b) Flux distribution

Figure 4.1 The cross section and flux distributions for the uniform cross section test case. Watch points are marked with black X's on the cross section diagram.



(a) IPM - 10x10 zones, 2x2 rays/zone

(b) SM - 10x10 zones, 2 rays/face

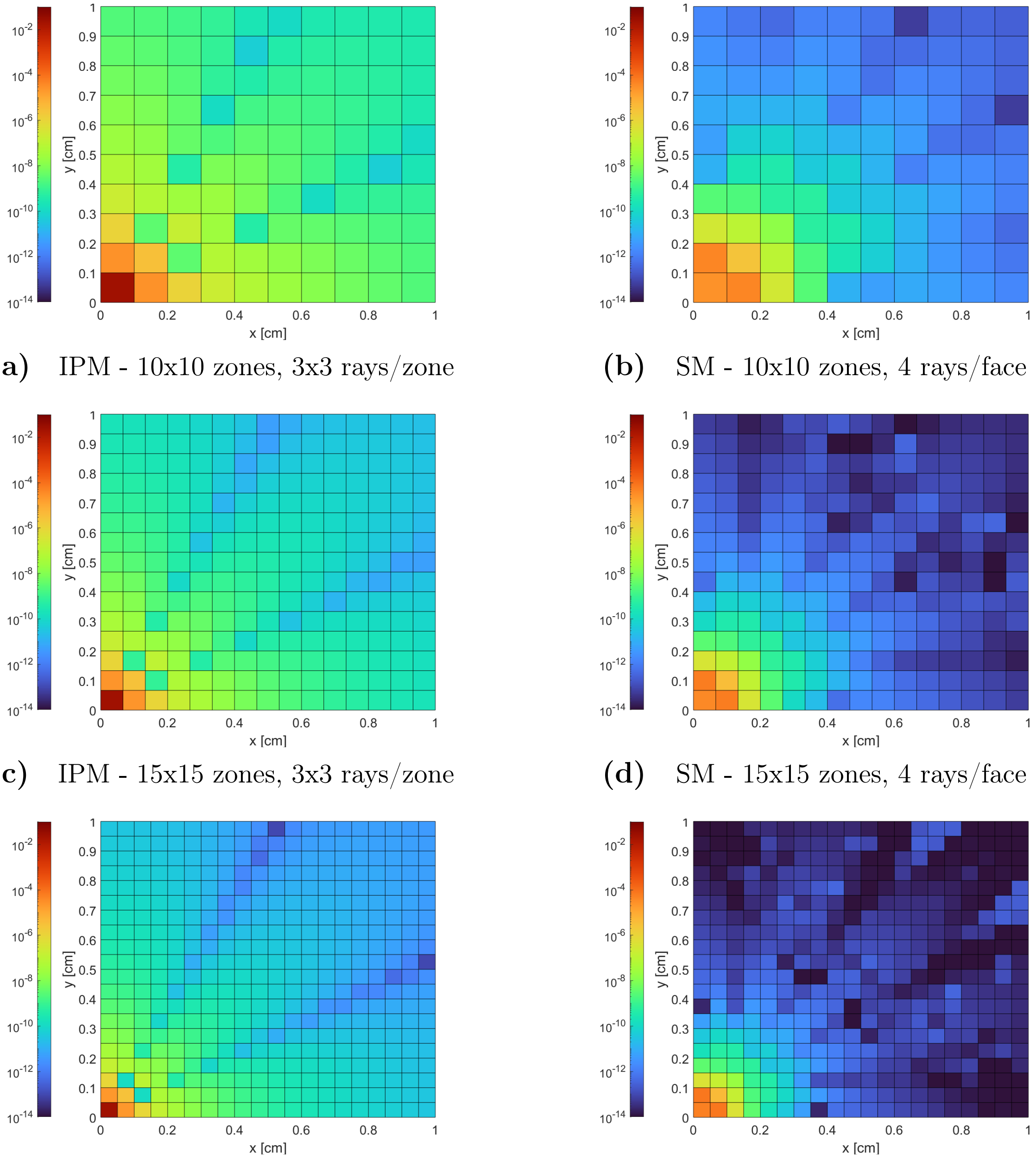
(c) IPM - 15x15 zones, 2x2 rays/zone

(d) SM - 15x15 zones, 2 rays/face

(e) IPM - 20x20 zones, 2x2 rays/zone

(f) SM - 20x20 zones, 2 rays/face

Figure 4.2 The relative absorption error distributions for various mesh resolutions. Plots on the left use the interior point method with a 2x2 quadrature set (4 rays per face). Plots on the right use the surface method with a 2 point quadrature per face, for a total of 4 effective rays per zone after factoring in ray sharing.



(a) IPM - 10x10 zones, 3x3 rays/zone

(b) SM - 10x10 zones, 4 rays/face

(c) IPM - 15x15 zones, 3x3 rays/zone

(d) SM - 15x15 zones, 4 rays/face

(e) IPM - 20x20 zones, 3x3 rays/zone

(f) SM - 20x20 zones, 4 rays/face

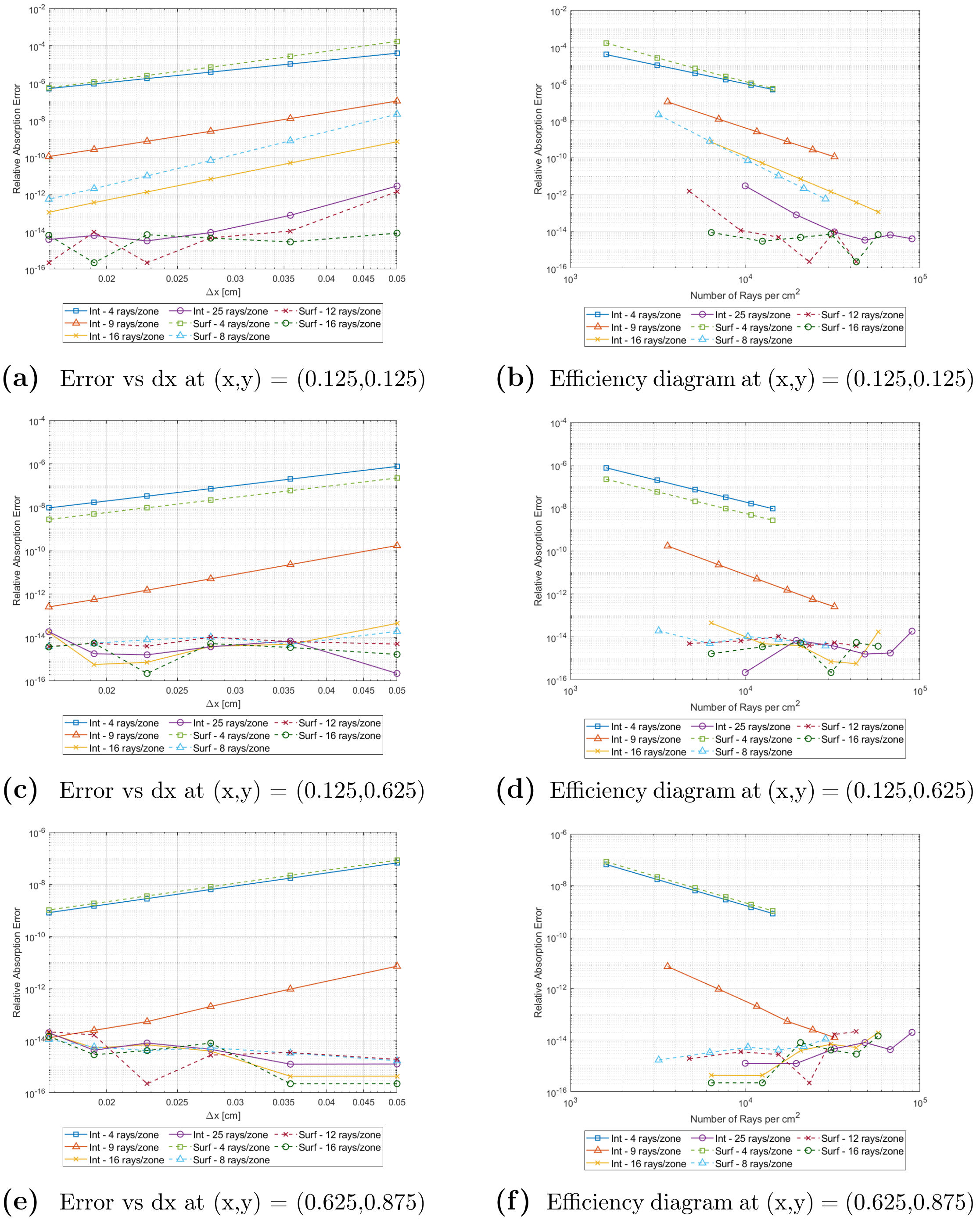
Figure 4.3 The relative absorption error distributions for various mesh resolutions. Plots on the left use the interior point method with a 3x3 quadrature set (9 rays per face). Plots on the right use the surface method with a 4 point quadrature per face, for a total of 8 effective rays per zone after factoring in ray sharing.

zone (1,1) are in such close proximity to the source relative to that of the far faces, the error along these shared faces is significantly higher compared to that of the far ones. This is then compounded by the fact the surface method experiences additive errors. Because these adjacent zones only have a single face where the current is zero instead of two, whatever improvements in the error gained from moving further from the source are essentially canceled out by the addition of a third non-zero face. In fact, if the absorption error throughout row 1 was plotted as a function of x, it would be seen that zone (1,1) is a special case As one moves further away from the source, the relative difference in distance from the source between opposite faces of a zone gets smaller and smaller, meaning the difference in current error between the near and far faces is less and less, in turn causing the total absorption error of the zone to be less dominated by a single face.

The plots in Figure 4.4 show the local relative absorption error (left) and associated efficiency diagram (right) at three different watchpoints throughout the mesh for various ray counts and mesh resolutions. The idea of the efficiency diagram is to try and capture the amount of "work" the method had to do in order to obtain a given error. The work, in this case, is determined by the number of rays cast in a given area. Points in the bottom left of the efficiency diagram achieve the lowest error for the least amount of work and are thus the most efficient, while those further to right require a greater amount of work and are thus less efficient. Between these six plots, one can get a general sense for how the two methods perform throughout the problem.

As seen previously in Figures 4.2 and 4.3, increasing mesh resolution or quadrature order tends to improve the local solution, as one might expect. The local (and global) spatial convergence rate (that is, error vs zone size) is roughly $2N$ for both methods, where N is the number of quadrature points in a single dimension. The sole exception to this is the 4 rays/zone surface method run in Figure 4.4a, with a slope of m = 4.7489 (as measured from $\Delta x = 0.0275$ to 0.036), although this is due to it simply not having enough quadrature points to adequately represent the current distribution across the face of a zone so near the source. In fact, the slope progressively decreases as Δx gets smaller, converging upon the expected slope of 4.

While in general, the rate of convergence follows this $2N$ rule throughout the mesh, the initial error (the error at low mesh resolutions) varies strongly as a function of distance from the source, with it being significantly worse near the source. Because the cross section is uniform and aligns with the mesh perfectly, there is no error in the ray trace itself (τ has no error associated with it), meaning the absorption error is entirely based on integration error from the numerical quadrature. It is important to remember that Equations 3.20 and 3.29 are not polynomials and therefore not



(a) Error vs dx at (x,y) = (0.125,0.125)

(b) Efficiency diagram at (x,y) = (0.125,0.125)

(c) Error vs dx at (x,y) = (0.125,0.625)

(d) Efficiency diagram at (x,y) = (0.125,0.625)

(e) Error vs dx at (x,y) = (0.625,0.875)

(f) Efficiency diagram at (x,y) = (0.625,0.875)

Figure 4.4 Plots of the relative absorption error and corresponding efficiency diagram for the uniform cross section problem at various watchpoints.

exactly approximated by a Gaussian quadrature. Near the source, the current/flux distributions across a face/zone vary more significantly than they do further away, making it more difficult for the quadratures to approximate accurately.

Comparing the performance of the interior point and surface methods against each other, it is evident that, given a uniform cross section, the surface method tends to outperform the interior point method when using equivalent ray counts. This is because the surface method has a greater potential for sharing rays between different faces/zones, allowing it to use higher order quadratures while maintaining equivalent numbers of rays to the interior point method. When using lower order quadratures where this increased sharing is less impactful, the two methods demonstrate similar performance (the interior point method actually often outperforms the surface method when using the same quadrature order), the surface method scales significantly better due to its potential for optimization.

It is important to note that while the watchpoints allow one to get some idea of global error trends, a comparison between Figures 4.4 and 4.5 clearly shows how even when only including watchpoint values in the error norm calculation, the error in the watchpoint closest to the source at (x,y) = (0.125,0.125) still dominates the global error. Still, both the local and global error plots and their associated efficiency diagrams point to one important conclusion: for both methods, when given the choice of either increasing mesh resolution or quadrature order, for the same amount of work, it is typically better to increase the size of one's quadrature rather than add additional zones to the mesh, at least in areas that resemble a uniform cross section.

Figures 4.7a and 4.7b show the way in which the error scales as a function of both rays per zone (quadrature order) and number of zones. Although at low ray counts, the two methods scale very similarly with increasing mesh resolutions, as the number of rays per zone increases, the surface method quickly outpaces the interior point method. By 8 rays per face, the surface method is reaching the tolerance used in the adaptive reference solution at low mesh resolutions, something that interior point method is unable to replicate unless using significantly higher ray counts.

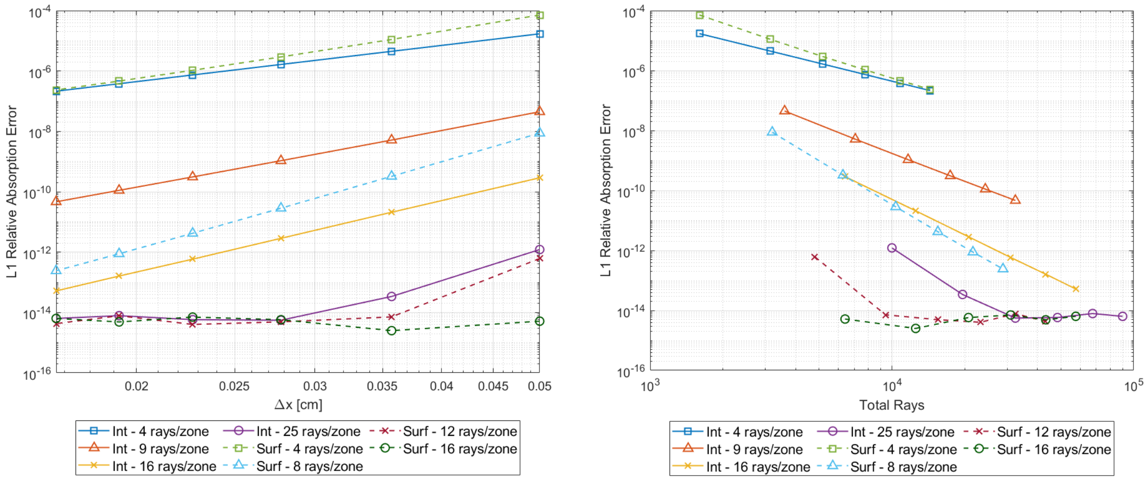
4.1.1 Different Quadrature Rules

While the primary quadrature rule used in both methods for this paper is Gauss-Legendre, there is a strong argument for trying alternative quadrature rules such as Gauss-Lobatto, particularly for the interior point method.

The two rules work using the same basic formula - with quadrature points and weights spaced unevenly across an interval. Where they differ, however, is that unlike Legendre, Lobatto contains quadrature points at the beginning and end points of the integral.

For the same number of quadrature points, Gauss-Legendre typically outperforms Gauss-Lobatto. Indeed, while Gauss-Legendre is able to exactly integrate polynomials of up to degree $2n-1$ (where n is the number of quadrature points), Gauss-Lobatto only has a degree of exactness of up to $2n-3$ [22]. However, in practice, Gauss-Lobatto has certain advantages regarding its potential for optimization that tends to offset this. Specifically, because Lobatto includes in its integration nodes the end points of the integral, additional rays may be shared between adjacent zones/faces for both the interior point and surface methods. The impact this has on the average number of unique rays needed to be cast to a single zone can be seen in Figure 4.8.

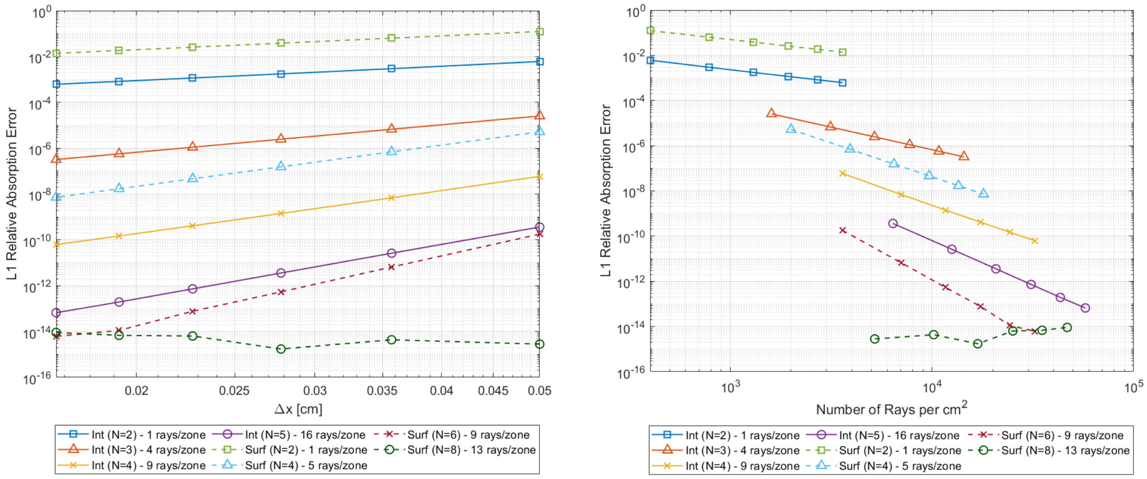
In 2D, for the surface method, this means that every corner point can be shared among 2 faces in each zone and among 4 neighboring zones (so 1 ray can shared 8 times, unlike rays on the interior of an edge that can only be shared with a single neighbor). The impact this has is much more significant when using lower ray counts, as the corners make up a larger proportion of the quadrature points. For the interior point method, however, having quadrature points on the end points makes a very big difference. When using Legendre, the interior point method only contains quadrature points along the interior of the zone (hence its name). When using Lobatto, however, a significant number of quadrature points now lie on the exterior of the zone. This



(a) Global L1 error norm

(b) Global efficiency diagram

Figure 4.5 The L1 error norm when using a Gauss-Legendre quadrature across all the watchpoints and its associated efficiency diagram.



(a) Global L1 error norm

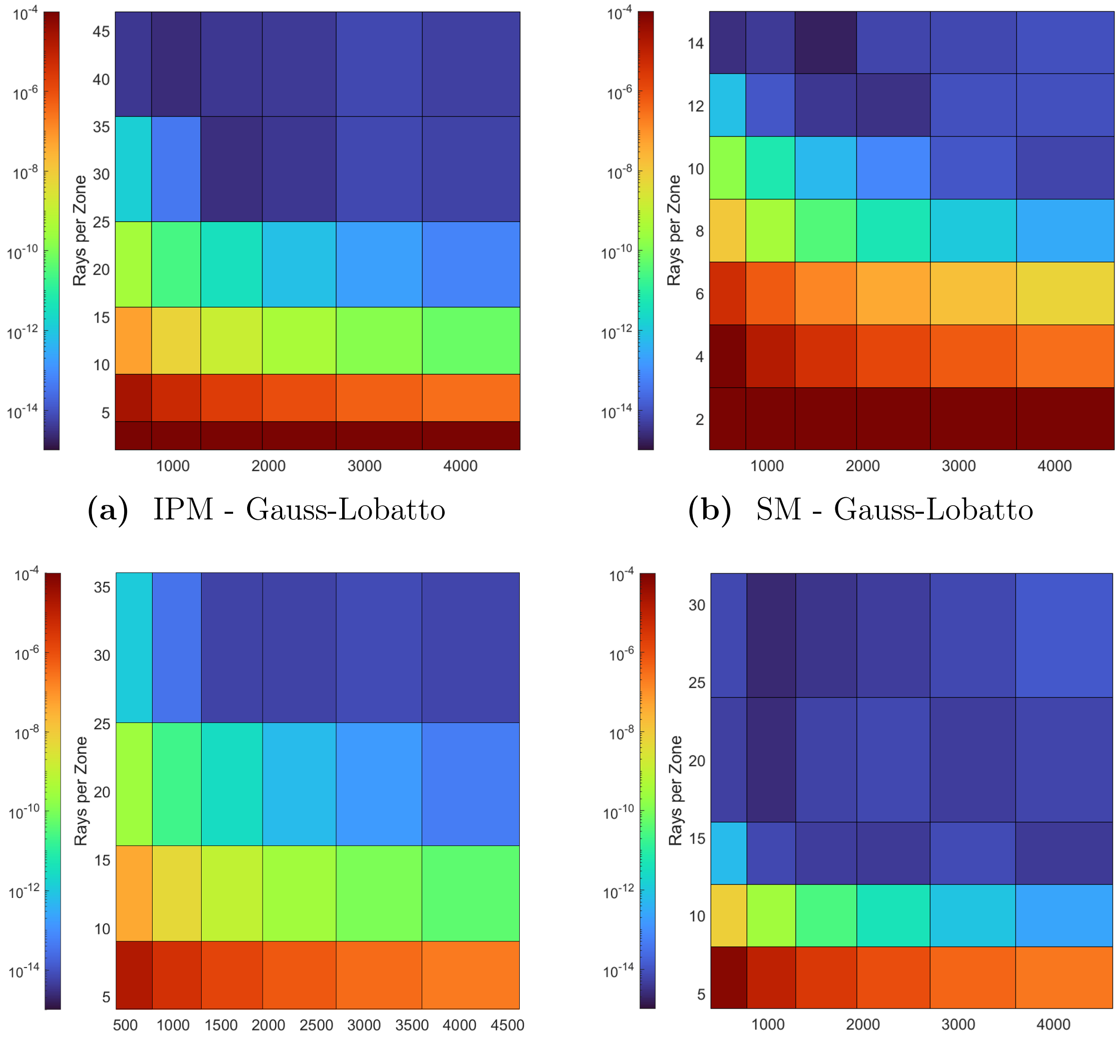
(b) Global efficiency diagram

Figure 4.6 The L1 error norm across the watchpoints and its associated efficiency diagram when using Gauss-Lobatto.

allows them to be shared in a similar manner to the surface method, the difference being that for any quadrature set greater than 2, there still will exist at least some interior points that cannot be shared. For rays on the interior of an edge, this means being shared once with an adjacent neighboring zone. For points in the corners, this means being used four times, once with each of its neighbors. Similar to the surface method, the impact this optimization has is larger when using smaller quadrature sets, as a greater proportion of the rays being cast can be shared at least once. The difference this makes to the equivalent number of rays cast when using Gauss-Legendre is greater for the interior point method than the surface method as the surface method is still able to share rays along the edges when using Legendre.

For both the interior point and surface methods, the fraction of exterior vs interior (meaning shared vs unshared) rays will decrease as quadrature order increased. In fact, if one takes the quadrature size out to infinity, the ratio will asymptotically approach that used in Legendre, that being when all rays are on the interior of a face/zone.

In most circumstances, both Gauss-Legendre and Gauss-Lobatto share similar performance for equivalent amounts of work. This can be seen when comparing the L1 error norms and efficiency diagrams for Gauss-Lobatto in Figures 4.6a and 4.6b to their Legendre counterparts in Figures 4.5a and 4.5b. Although the plots differ



(a) IPM - Gauss-Lobatto

(b) SM - Gauss-Lobatto

(c) IPM - Gauss-Legendre

(d) SM - Gauss-Legendre

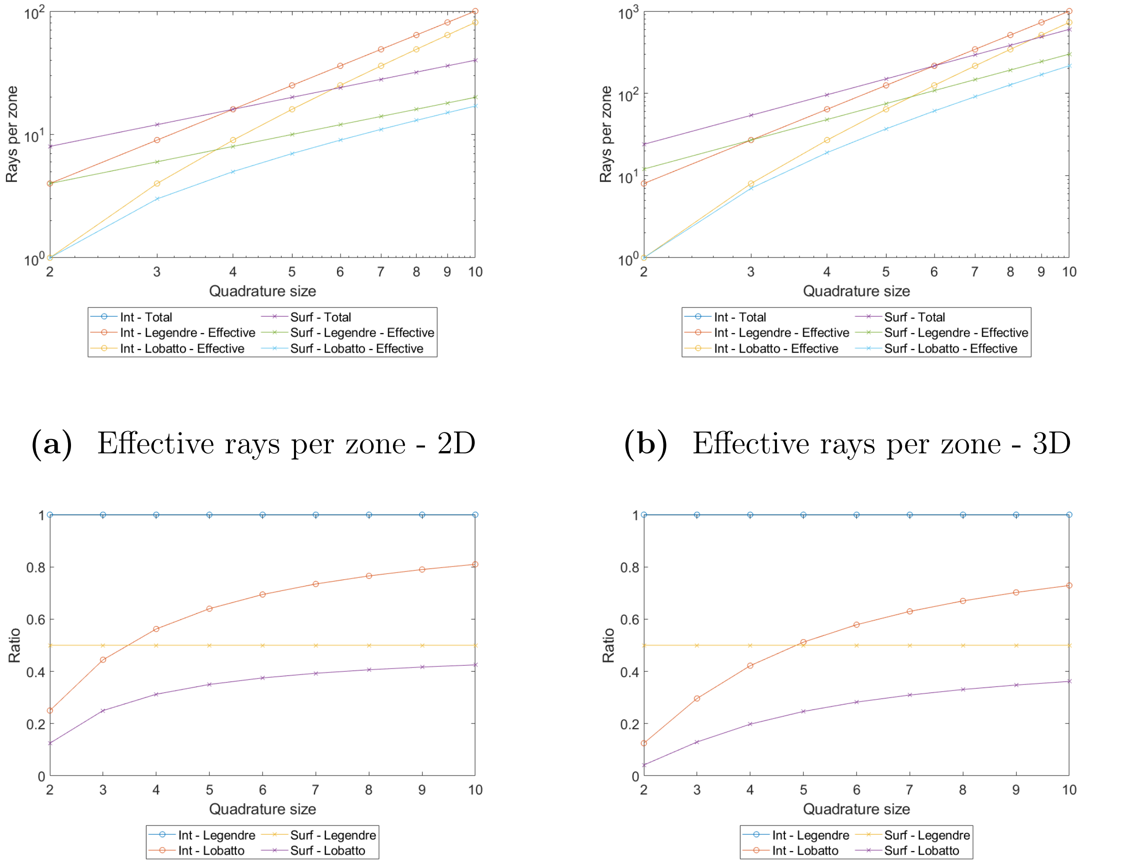
Figure 4.7 A visual representation of the different input variable combinations and their associated L1 error for the interior point and surface methods when using a constant cross section.

when comparing lines using the same quadrature size, if instead the comparison is made using roughly the same equivalent number of rays, it can be seen that the two quadrature rules have very similar performance to one another (its difficult to make an exact evaluation for the surface method since it will always have an odd number of equivalent rays when using Lobatto and an even number when using Legendre).

Comparing Figures 4.7a and 4.7b obtained when using Gauss-Lobatto to Figures 4.7c and 4.7d obtained using Gauss-Legendre further demonstrates how the two rules scale in similar manners to one another after accounting for the increased ray sharing in Lobatto. Something that is not immediately apparent is the way in which Lobatto's optimization enables a greater degree of flexibility in the effective number of rays used, particularly for the interior point method. This is one of the drawbacks to the interior point method - the number of rays being used scales quadratically in 2D and cubically in 3D, increasing rapidly as one's quadrature order goes up. While still not offering the same degree of flexibility in work that the surface method has, using Gauss-Lobatto does improve this.

There are, however, times where it is better to use one rule over the other. One such case is in adaptive methods where it is often beneficial to use smaller quadrature sizes to cut down on the amount of wasted work. When using Legendre, each new subdivision requires an entire new set of rays to be cast, whereas when using Lobatto, rays cast to the edges of new subzones can be reused. Because adaptive methods typically use smaller quadrature sizes than non-adaptive methods due to their ability to subdivide, using Lobatto allows for a significant proportion of the rays to be reused multiple times. For example, in an adaptive variant of the the surface method in 2D using a 2 point quadrature, when using Lobatto, the number of rays cast before any subdivision takes place is two rays located at the ends (corners) of the face. A single division requires one additional ray to be cast to the center of the face, for a total of three unique rays. Each additional subdivision only requires one additional ray to be cast. A visual representation of this can seen in Figure 4.9. Comparatively, if using Legendre, an initial 2 rays would need to be cast for the original face, followed by 4 rays for each subdivision taking place. As none of these rays can be shared, a total of 6 rays were needed for a single subdivision, and 10 for the second.

In general, for 2D problems, when using the surface method, the number of unique rays per face when using an even-point Lobatto quadrature is $n_{quad} + N_{sub}(n_{quad} - 1)$, where n_{quad} is the number of quadrature points in a single dimension and N_{sub} is the number of subdivisions. When using an odd-point quadrature, the number of unique rays is $n_{quad} + N_{sub}(n_{quad} - 2)$. For Legendre, because no rays are being shared, the number of unique rays for a single zone is $n_{quad}(1 + N_{sub})$ It is important to note that these equations are only accounting for ray sharing between sub-faces and not



(a) Effective rays per zone - 2D

(b) Effective rays per zone - 3D

(c) Ratio of effective vs total rays per zone - 2D

(d) Ratio of effective vs total rays per zone - 3D

Figure 4.8 The average number of rays cast per zone when using Gauss-Legendre vs Gauss-Lobatto for the interior point and surface methods (assuming an infinite mesh with no edges) (top) and the corresponding ratio of effective vs. total rays (bottom). Gauss-Lobatto uses significantly fewer rays at low quadrature orders because of sharing. As the quadrature size increases, the number of rays used by Lobatto will slowly approach that of Legendre.

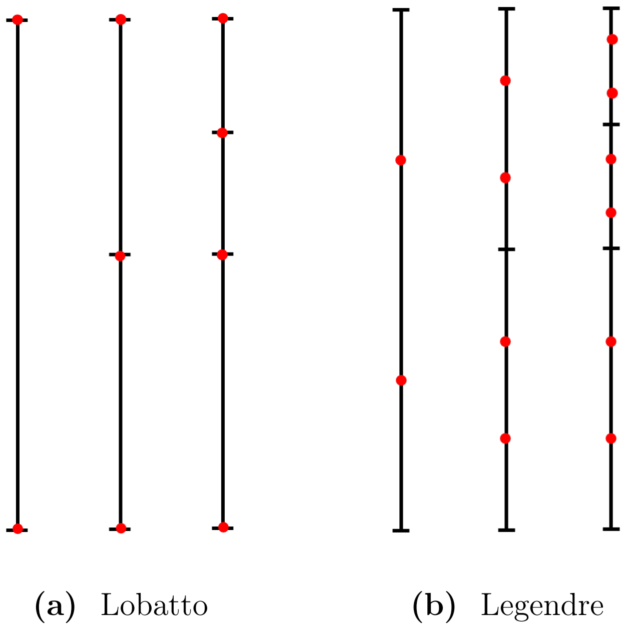


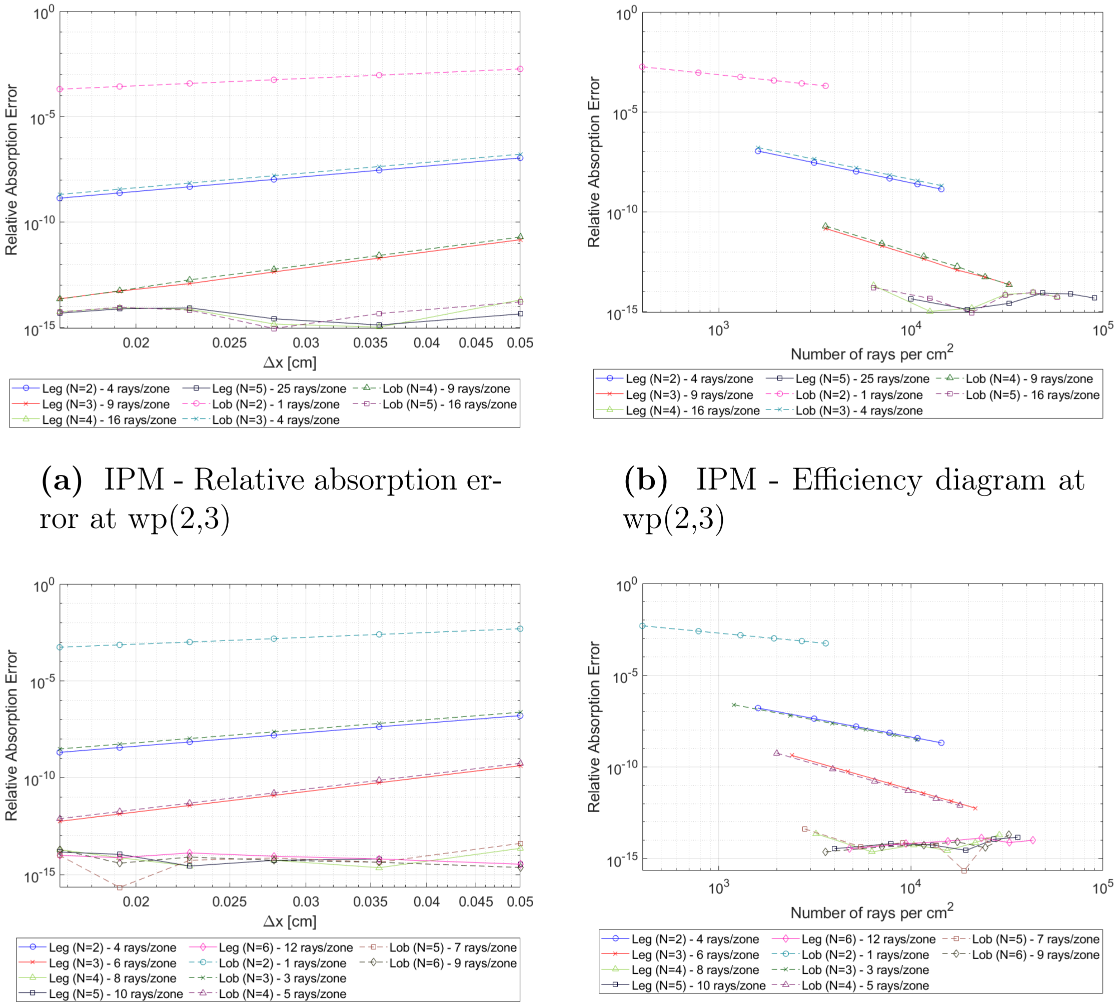
Figure 4.9 An example of how refinement and ray sharing works for an adaptive variant of the surface method in 2D using a 2 point quadrature.

between neighboring zones or the shared corner point between intersecting faces. Taking sharing into account, for both quadratures, the current along each face can be shared with a single neighboring zone, reducing the total number of rays needed by half. This is further reduced when using Lobatto as each corner can be used with multiple neighboring faces.

For the interior point method, when using Lobatto, the number of unique rays for an even point quadrature is $n_{quad}^2 + N_{sub}[4(n_{rays}-2)^2 + 8(n_{rays}-2) + 4(n_{rays}-2)]$ and for an odd point quadrature is $n_{quad}^2 + N_{sub}[4(n_{rays}-2)^2 + 8(n_{rays}-2) + 4(n_{rays}-1) + 1]$. For Legendre, because none of the interior points rays are shared, the number of unique rays per zone is $n_{quad}^2(1 + N_{sub})$.

Figure 4.11 contains a comparison of the error distributions between the two quadrature rules for the interior point method using equivalently amounts of work. It is readily apparent that, at least for this problem, the two quadrature rules produce roughly the same error distributions. This is supported by the local error and efficiency diagrams for the interior point method seen in Figure 4.10, which show how the two quadratures scale against one another.

The same can more or less be said for the behavior of the surface method when using the two quadrature rules. Unfortunately, unlike when using the interior point



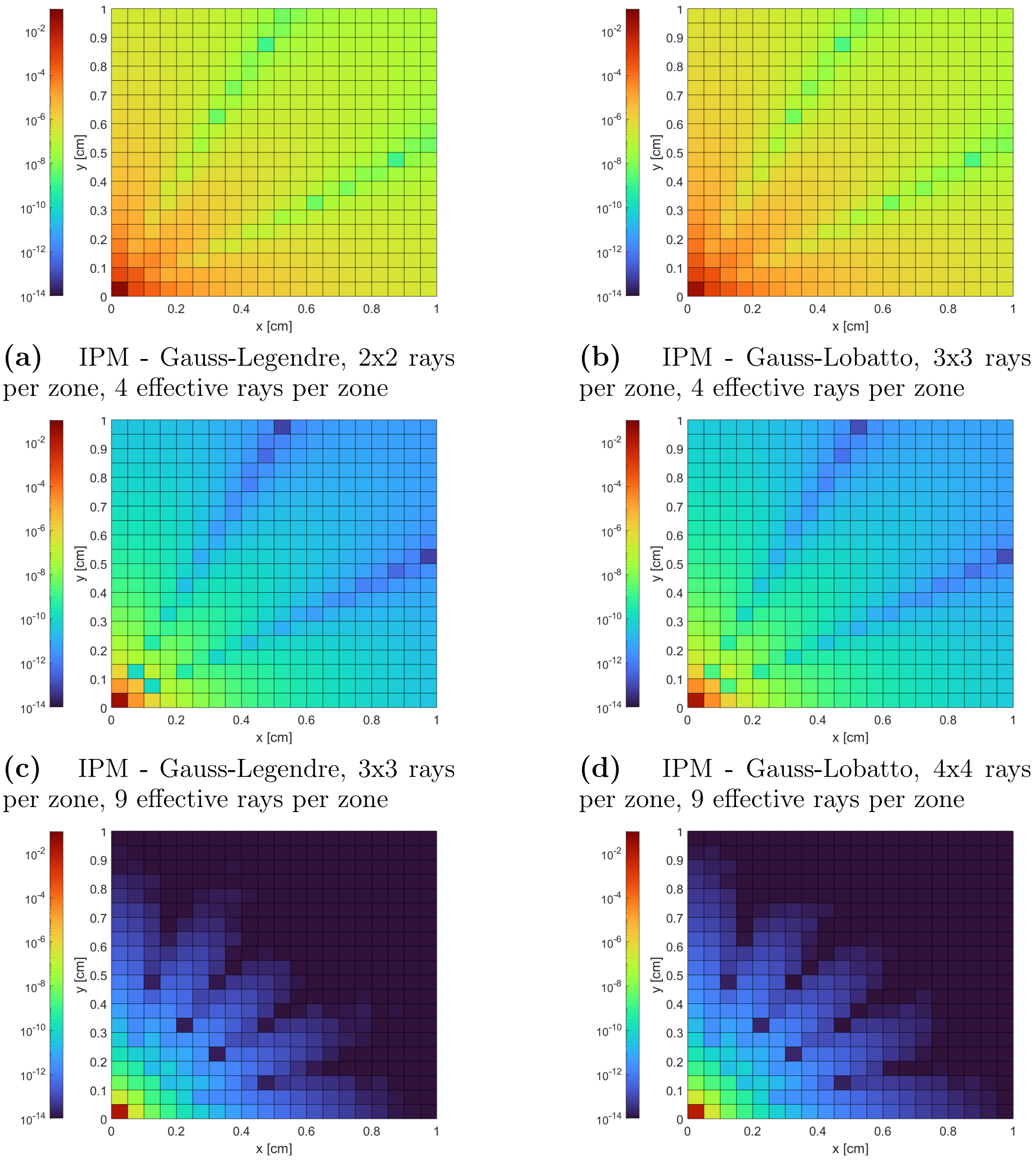
(a) IPM - Relative absorption error at wp(2,3)

(b) IPM - Efficiency diagram at wp(2,3)

(c) SM - Relative absorption error at wp(2,3)

(d) SM - Efficiency diagram at wp(2,3)

Figure 4.10 The relative absorption error and associated efficiency diagram for the interior point method (top) and surface method (bottom) comparing a Gauss-Legendre quadrature to that of Gauss-Lobatto.



(a) IPM - Gauss-Legendre, 2x2 rays per zone, 4 effective rays per zone

(b) IPM - Gauss-Lobatto, 3x3 rays per zone, 4 effective rays per zone

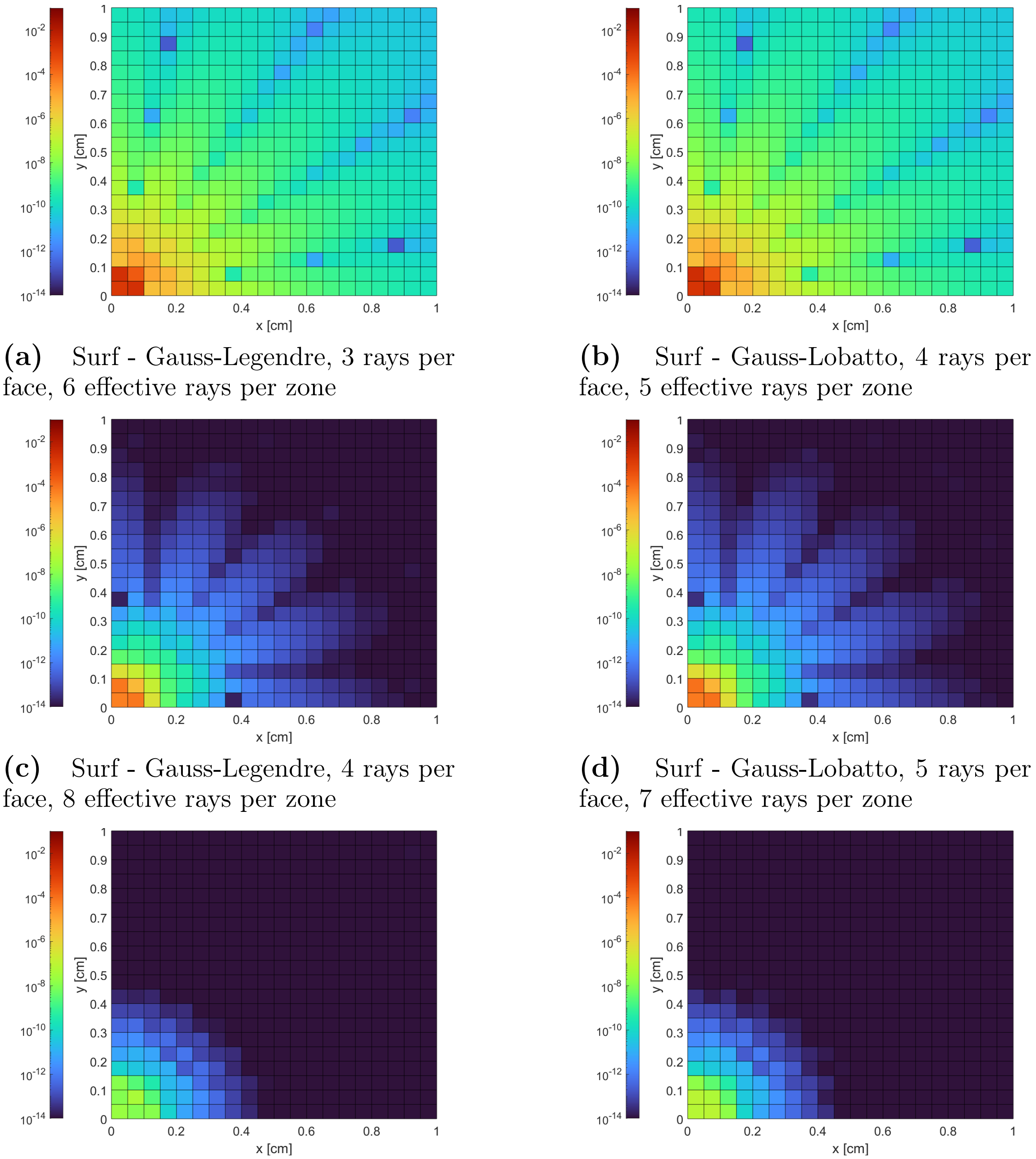
(c) IPM - Gauss-Legendre, 3x3 rays per zone, 9 effective rays per zone

(d) IPM - Gauss-Lobatto, 4x4 rays per zone, 9 effective rays per zone

(e) IPM - Gauss-Legendre, 4x4 rays per zone, 16 effective rays per zone

(f) IPM - Gauss-Lobatto, 5x5 rays per zone, 16 effective rays per zone

Figure 4.11 The relative error distributions for the interior point method using a 20x20 mesh. Left is Legendre, right is Lobatto.



(a) Surf - Gauss-Legendre, 3 rays per face, 6 effective rays per zone

(b) Surf - Gauss-Lobatto, 4 rays per face, 5 effective rays per zone

(c) Surf - Gauss-Legendre, 4 rays per face, 8 effective rays per zone

(d) Surf - Gauss-Lobatto, 5 rays per face, 7 effective rays per zone

(e) Surf - Gauss-Legendre, 5 rays per face, 10 effective rays per zone

(f) Surf - Gauss-Lobatto, 6 rays per face, 9 effective rays per zone

Figure 4.12 The relative error distributions for the surface method using a 20x20 mesh. Left is Legendre, right is Lobatto.

method where one can compare the two quadrature rules using the same amount of work (the same number of effective rays per zone), the amount of work done by two quadrature rules when using the surface method will always be misaligned by one effective ray. That being the case, it is initially apparent from the plots in Figure 4.12 that Gauss-Lobatto is able to achieve essentially the same error distribution as the Gauss-Legendre with one fewer effective rays per zone. This is somewhat misleading, as if one looks at the plots in Figure 4.10, the error is slightly higher for Lobatto when it uses one fewer effective rays per zone.

If one compares the error distributions in Figure 4.11 to 4.12, it is evident that the surface method can achieve roughly the same sort of performance to the interior point method for half the amount of work, supporting the overall trend of the two methods having roughly equivalent performance in terms of error for a given quadrature size. It is only after one accounts for the increased ray sharing that the surface method begins outperforming the interior point method.

4.2 Pin Problem

While the uniform cross section problem was able to provide important insight on the general characteristics and performance of the two methods, it is also quite literally the most simple problem one can model. The next step, therefore, is the addition of a second material. Chosen for this was a simple pin problem consisting of a square pin made of highly absorbing material surrounded by a second, lower absorbing material. This can be seen in Figure 4.13, along with it's associated flux distribution.

In many ways, this problem is a continuation of the uniform test case, as the regions near the source and on the sides are both described by a uniform cross section. However, the addition of the pin fundamentally alters the problem, as Equations 3.18 and 3.29 are no longer smooth everywhere throughout the problem domain. In fact, not only are there discontinuous derivatives for these equations in certain zones, but the functional form of the optical depth also changes for any quadrature point whose associated ray travels through the pin.

One way in which to think about optical depth is to separate the cross section and path length terms. We can calculate a weighted average cross section for the ray associated with quadrature point i and zone m using the function:

$$\bar{\Sigma}_t(R) = \frac{\int_{R_{src}}^{R} \Sigma_t dR'}{R - R_{src}} \tag{4.3}$$

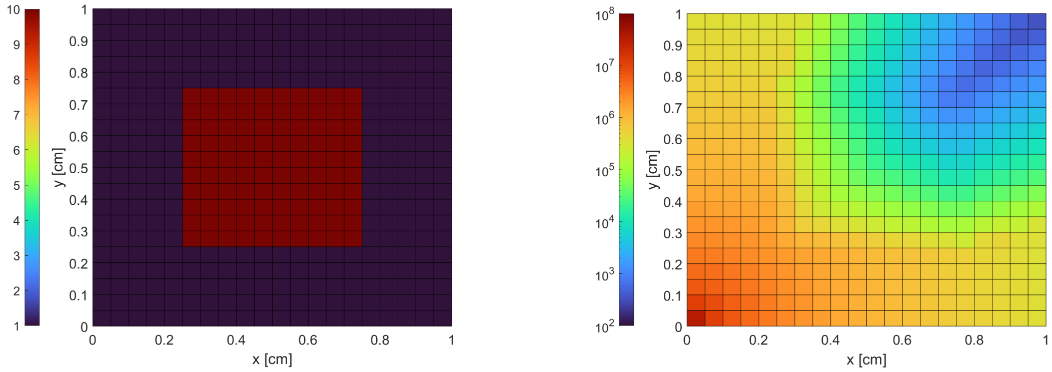
$$\bar{\Sigma}_{t,i}^{(m)} = \frac{\sum_{n=1}^{m} \Sigma_{t,n} R_n}{\sum_{n=1}^{m} R_n} \tag{4.4}$$

which can then be used to rewrite Equation 3.13 as

$$\tau(R) = \bar{\Sigma}_t(R)(R - R_{src}) \tag{4.5}$$

$$\tau_i^{(m)} = \bar{\Sigma}_{t,i}^{(m)}(R_i^{(m)} - R_{src}) \tag{4.6}$$

Now, if the cross section for each zone along a rays path is the same, the average cross section for the ray will simply equal that of the individual zones. If this is the case for for every quadrature point in the zone/face, as it is for all zones in the uniform case and those on the sides/near the source in this case, then $\bar{\Sigma}_t$ is constant throughout the quadrature set, meaning that the optical depth is only a function of R. However, if for any reason a ray ever passes through a material interface, the average cross section will no longer be constant throughout the zone. This fundamentally changes the form Equation 4.6 takes. While in both cases, Equations 3.18 and 3.29 are not polynomials and therefore not able to be exactly integrated using a Gaussian quadrature, it is clear from the error distributions in Figures 4.14 and 4.15 that



(a) Flux distribution

(b) Flux distribution

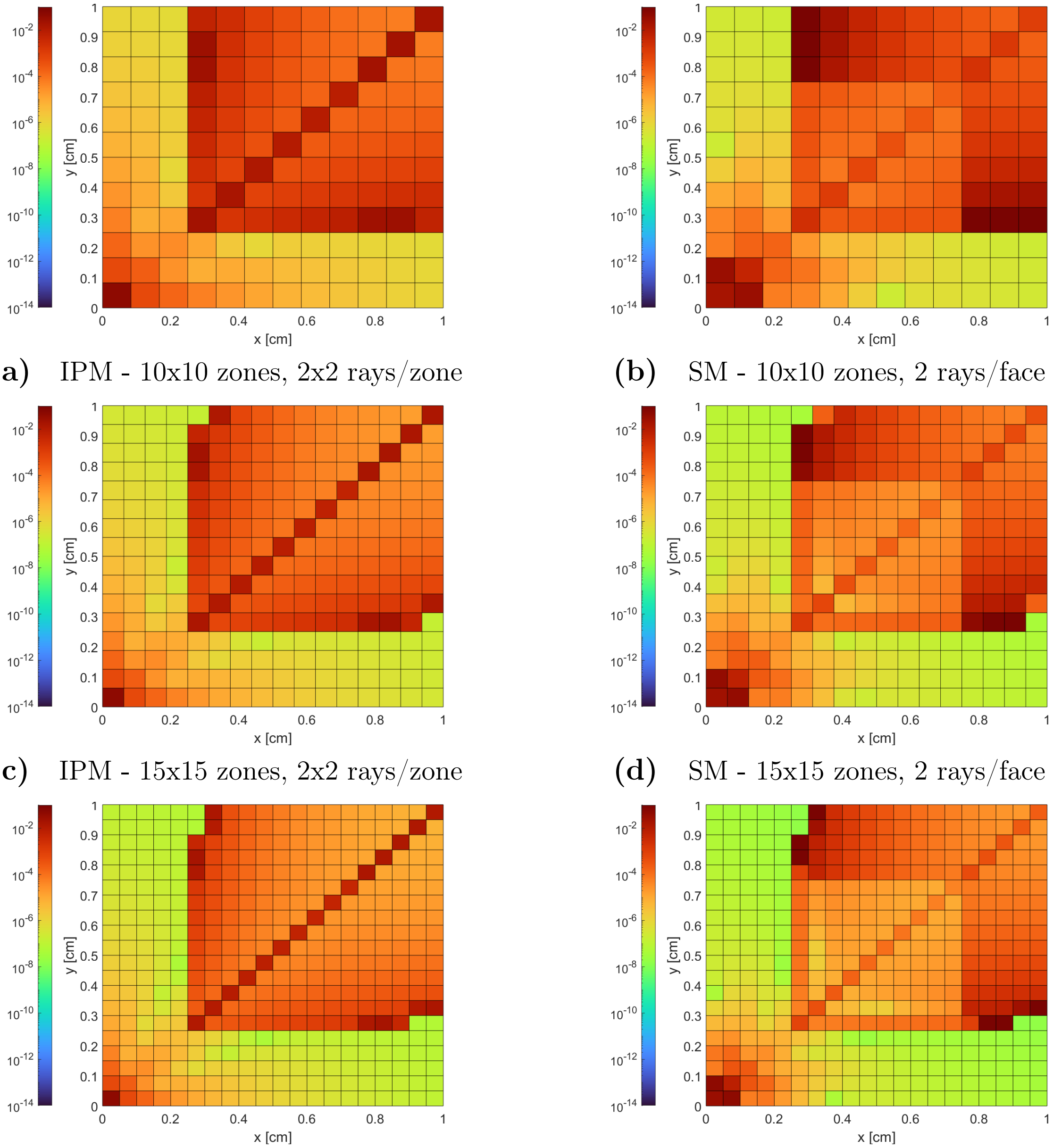
Figure 4.13 The cross section and flux distributions for the pin problem.

regions described by a uniform cross section are more accurately modeled (with the exception of zones in close proximity to the source, as discussed in the previous section) than those in and behind the pin.

Expanding our evaluation from the uniform regions to the problem as whole, several things are evident. First, just as previously observed in the uniform cross section case, increasing the order of one's quadrature generally has a much greater impact on error than increasing the mesh resolution. The next to note is the very high concentration of error in zones along the corners and center diagonal of the pin (particularly for the interior point method). These are caused by the same effect, just in a slightly different manner. Specifically, if one were to look closely at the zones along the bottom right corner of the pin, one would see a discontinuity in the optical depth at the quadrature points these faces/zones. A representation of this can be seen in Figure 4.16 for the surface method. Along this face, some rays are traveling through the pin, while others are not. This sudden change in cross section creates a discontinuity in the optical depth and therefore the current that is unable to be accurately represented using a relatively low order polynomial, as is the case here.

For the interior point method, the effect this discontinuity has is manifested through higher error throughout these zones. For the surface method, however, the impact this sort of discontinuity can have is significantly worse, ranging from simply high error to degrading the local solution to the point where negative values are obtained for the flux and absorption rate. This is due to the application of the divergence theorem in Equation 3.30 to obtain the absorption rate - specifically, if one or more of a zone's faces has to high an error, the summation can be thrown off.

The high error that can observed along the diagonal of the pin in the interior point method is due to a similar sharp change in the optical depth distribution along the interior of a zone. If one were to plot the optical depth along a line stretching from the top left corner to the bottom right corner of the pin, one would see a sharp change in its slope at the center as rays shift from one side of the bottom left corner to other side. This sharp change is present in every zone with rays traveling on both the left and right side of the bottom left corner of the pin. In fact, because the pin is perfectly centered in the mesh, the change is present along the entire diagonal, from bottom left to top right. This also means, however, that as long the pin is centrally located, the surface method will never actually see this discontinuity when using Gauss-Legendre, as the sharp change occurs at the corner and does not fall between any two quadrature points. As far as the surface method is concerned, as long as the change occurs at the corners of a zone, it might as well not exist. This is why the line of high errors along the diagonal that can be seen in the interior point



(a) IPM - 10x10 zones, 2x2 rays/zone

(b) SM - 10x10 zones, 2 rays/face

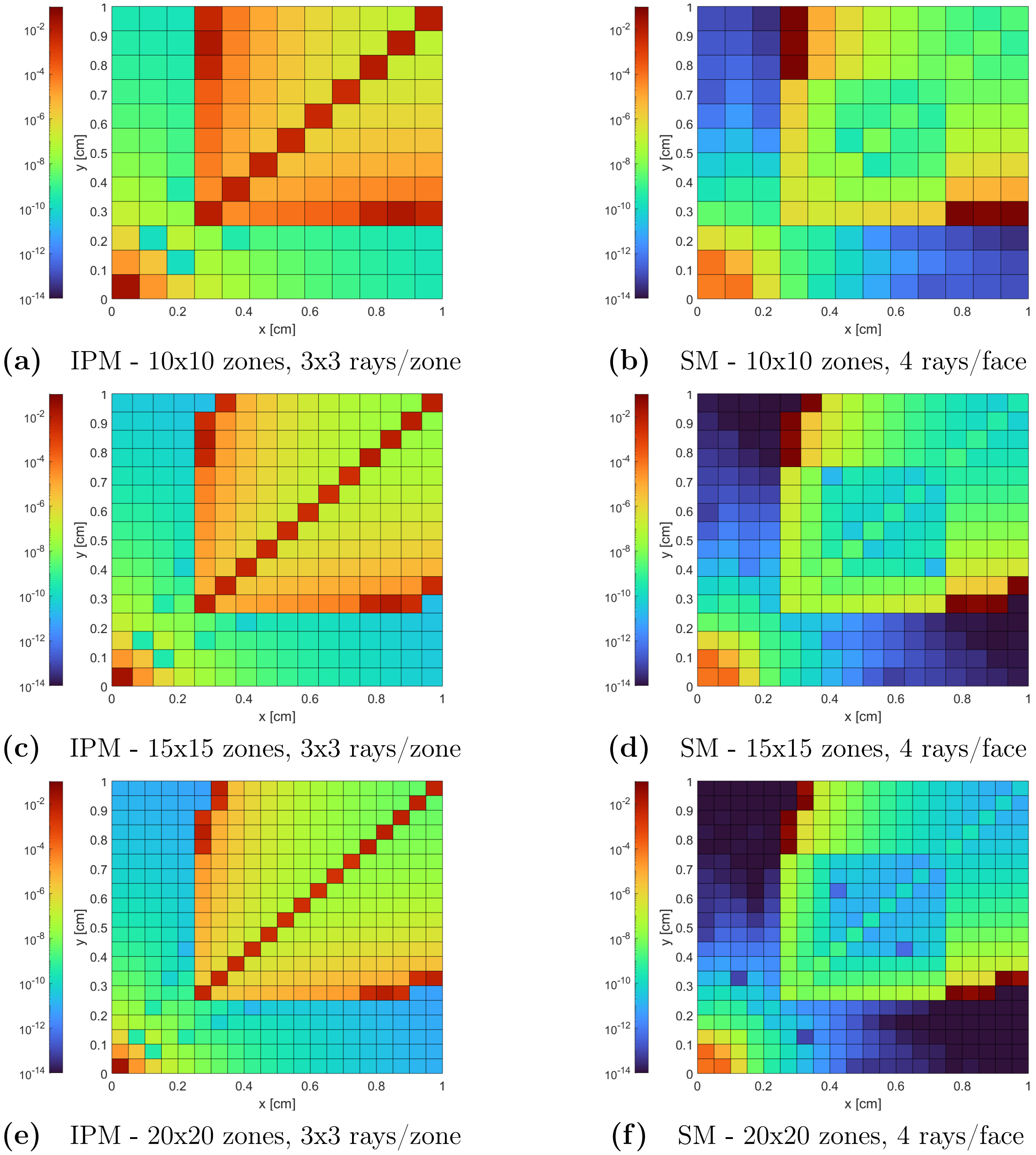
(c) IPM - 15x15 zones, 2x2 rays/zone

(d) SM - 15x15 zones, 2 rays/face

(e) IPM - 20x20 zones, 2x2 rays/zone

(f) SM - 20x20 zones, 2 rays/face

Figure 4.14 The relative error distributions for a variety of different mesh resolutions for the pin problem. Plots on the left were calculated using the interior point method with a set of 2x2 rays cast per zone. Plots on the right were calculated using the surface method using 2 rays per face (both methods are casting the same number of equivalent rays per zone)



(a) IPM - 10x10 zones, 3x3 rays/zone

(b) SM - 10x10 zones, 4 rays/face

(c) IPM - 15x15 zones, 3x3 rays/zone

(d) SM - 15x15 zones, 4 rays/face

(e) IPM - 20x20 zones, 3x3 rays/zone

(f) SM - 20x20 zones, 4 rays/face

Figure 4.15 The relative error distributions for the absorption rate for a variety of different mesh resolutions. Plots on the left were calculated using the interior point method with a set of 3x3 rays cast per zone, for a total of 9 rays per zone. Plots on the right were calculated with the surface method using 4 rays per face, for a total of 8 effective rays per zone (each ray can be shared between neighboring zones, reducing the work by half).

method doesn't really exist in the plots for the surface method. In fact, the faint line that is visible for the surface method in these plots is actually due to the current distribution being symmetrical across the diagonal rather than asymmetrical like it is everywhere else. This is not say, however, that the surface method is unaffected by sharp changes. If the pin offset even slightly so that the change in slope falls between quadrature points on a face, as was done to produce Figure 4.17, a very similar line takes shape. In fact, the slight offset done for this particular set of runs also created a second line along the top right corner of the pin where this sort of behavior can be seen.

This phenomena can be addressed in several ways: first, by increasing zone resolution, as this will minimize the areas affected by the error since the discontinuity is localized to specific zones along a line and not an area. Similarly, another way to reduce the impact the sharp change in optical depth has is to increase the quadrature order used. While this can improve the solution to the point where the surface method is not longer so off its computing negative absorption rates, its rather inefficient.

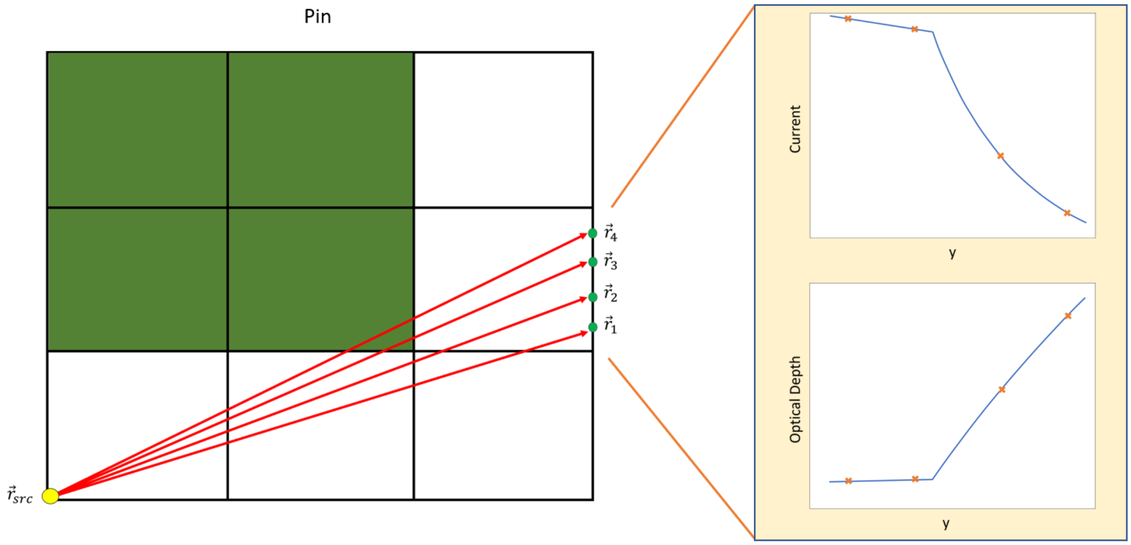
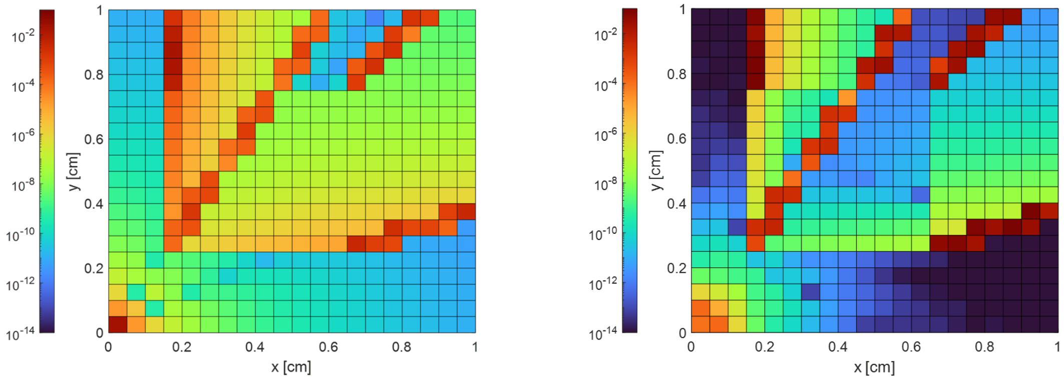


Figure 4.16 Changes in cross section can create sharp changes in the optical depth along a face, leading to significantly higher error compared to surrounding zones.



(a) Interior point method

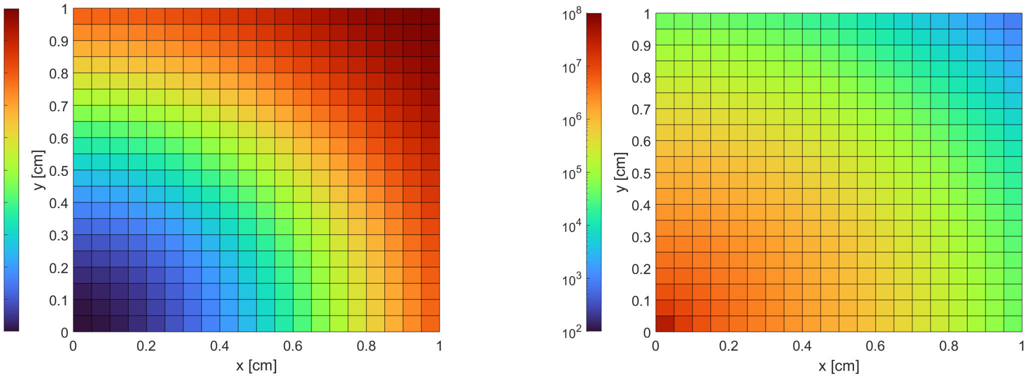
(b) Surface method

Figure 4.17 The error distribution for a pin slightly offset from the center. The image on the left used the interior point method with a 3x3 Gauss-Legendre quadrature, and the image on the right used the surface method with a 4 point Gauss-Legendre quadrature. The discontinuity in zones straddling the bottom left corner can clearly be seen in both methods.

Both of these approaches, while able to help reduce the error, are unable to significantly improve the solution in these regions. The only truly effective method to eliminate this high error in the solution is to apply some form of adaptive variant that integrates the face/zone compositely, as was done to compute the reference solution used here. While the presence of a discontinuity more or less guarantees a poor solution in the interval it exists, by solving the problem compositely, one is able to restrict said error to a small enough part relative to the overall integral as to make it insubstantial.

4.3 Radially-Symmetric Smoothly-Varying Cross Section

Although many real world problems can be described by distinct, discrete regions composed of uniform material, there are plenty of cases that don't necessarily meet this criteria - in particular, regions with large temperature or density gradients, or any sort of non-homogeneous mixture. For example, most nuclear reactors use uranium oxide fuel, which is essentially a powder that has been pressed and baked into



(a) Cross section distribution

(b) Flux distribution

Figure 4.18 The cross section and flux distributions for the symmetrical function case.

a pellet-like shape. While the process creates a solid material, the density inside the pellet can vary significantly, even without accounting for things like burn-up. In addition to this, once inside the core, these pellets are subjected to temperature gradients of hundreds of degrees Celsius. Cross sections can vary significantly with temperature, and modeling a pellet using a uniform temperature can create significant errors. Because of problems like this, its important to examine how well these methods perform when modeling regions that cannot really be described using uniform properties.

To test this, we defined a radially-symmetric cross section using the function:

$$\Sigma(\vec{r}) = 5 + 5\cos\left(\frac{||\vec{r}||}{R_{rc}}\pi + \pi\right) \tag{4.7}$$

where R_{rc} is the length of the diagonal from the bottom left corner to the top right corner of the mesh. An image of this is depicted in Figure 4.18, along with its associated flux. This particular function was chosen because it is infinitely differentiable, and has been restricted to half a wavelength to remove any oscillatory behavior in the cross section.

There are two ways to go about finding the optical depth for this problem. The first is to do what has been done in all the previous cases - that is, to assign each zone a "regional" cross section based on its central position and to perform the ray trace on the mesh. While relatively straight forward, there are several problems with this approach. The first is that it ties the cross section directly to the mesh resolution, meaning that changes to the resolution fundamentally alter the cross section distribution described by the mesh, as the zone centers for say a 20x20 mesh will not match that of a 21x21 mesh. This essentially creates an initial error in how well the cross section described by the mesh actually describes the function. Although this error will eventually disappear as one increases the mesh resolution, the number of zones required to accomplish this exceeds what is practical. This error also makes it very difficult to compare solutions for different mesh resolutions, as the fundamental problem described by the mesh is changing. If one looks at the optical depth at a watchpoint, the value obtained will depend on the resolution being used, with it converging to the value given by the function as one increases zone count.

The second problem with tracing on the mesh is that by giving every zone a different cross section in comparison to its neighbors, one inadvertently creates a very unsmooth optical depth distribution along the faces and zone interiors. This is essentially the same effect that can be seen along the corners of the pin in the earlier test case, except instead of being localized to a handful of zones, its seen almost everywhere. In many ways, this creates what is essentially a worst case scenario for both the interior point and surface methods, as without pairing the methods with some way to adaptively refine the quadratures and section off the bad areas, neither method is able accurately find a solution. The discontinuity will always exist, regardless of the quadrature order being used. This can clearly be seen in Figures 4.19a and 4.19b, where, with the exception of the zones in line with the source and, in the case for the surface method, along the diagonal, the local absorption error is very high, despite a reasonably sized quadrature being used. Compared to equivalent quadrature sizes and mesh resolutions for the uniform problem, the error is on average roughly 4 orders of magnitude worse for the interior point method and 8 for the surface method (except near the source). While Figure 4.20 does show how increasing mesh resolution and quadrature order helps in reducing the global error, the overall impact they have is fairly minimal and inconsistent. Gone are the straight lines seen in the uniform case, replaced with inconsistent behavior as changes in the mesh resolution shift the placement of the discontinuity and potentially leading to higher error when increasing mesh resolution or zone count. The error is intrinsically tied to the proximity of the discontinuity relative to the quadrature points. A higher mesh resolution or quadrature no longer necessarily means a lower error.

Another thing that should be noted in these plots is that the surface method performs significantly worse than the interior point method. This drop in performance is even more pronounced when one accounts for how much the surface method was outperforming the interior point method in the previous test cases. This is, however, consistent with what was being seen previously in the pin problem along the corners of the pin, and really demonstrates just how much the surface method seems to struggle when dealing with unsmooth optical depth distributions.

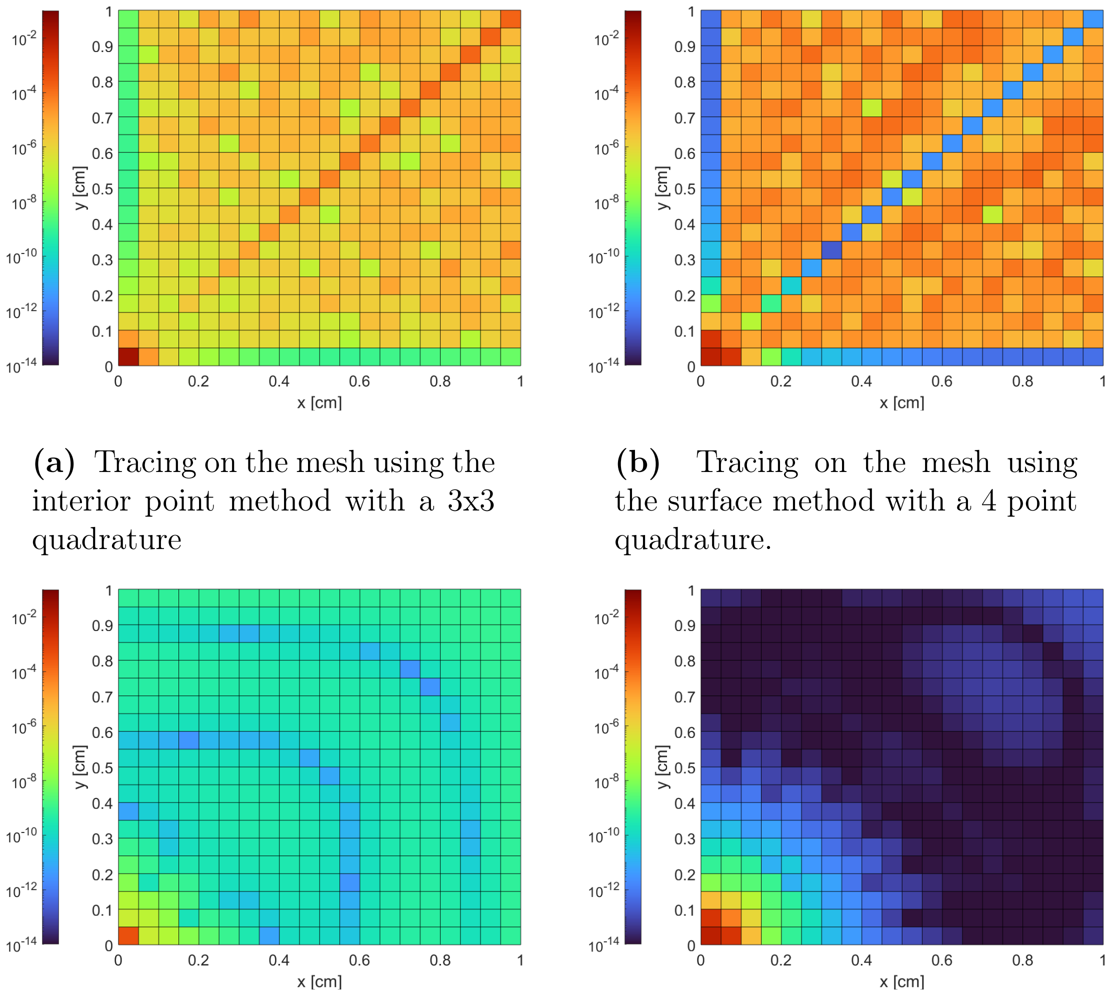
The second approach to finding the optical depth is instead of performing the ray trace directly on the mesh, since the equation defining the cross section distribution is known, to instead directly solve the line integral in Equation 3.13. This approach is both much faster than tracing on the mesh and able to yield a much more accurate solution, both for the optical depth and final solution. It both disconnects the problems dependency on the mesh resolution while at the same time eliminates any problems due to discontinuities.

The main drawback to this approach is that it requires one to know what the function is that characterizes the cross section. In a simple test problem like this, that's easy enough to find. However, in practice, problem geometries are often complex. Unlike Monte Carlo codes, most discrete ordinate codes do not support something like combinatorial geometry. Without this, unless the mesh aligns perfectly with the problem geometry, it is next to impossible to do the line integral.

Comparing Figures 4.19a and 4.19b to Figures 4.19c and 4.19d, the difference finding the optical depth by integrating the actual function has on the local error distribution cannot be understated. By removing the effect the unsmooth optical depth had on the error, the error distributions look much more similar to what was obtained in previous cases.

This improvement in both performance and error is also reflected in the global error plots of Figure 4.20. When tracing on the mesh, the abundance of sharp slope changes in the optical depth across a face/zone significantly hampers the quadrature from obtaining an accurate representation of the optical depth distribution. Just as seen with the pin, while increasing resolution and quadrature order does yield some improvement in error, it is far less impactful than before. The error curves are also more erratic than before, as the proximity of the non-differentiable points to the quadrature locations varies and has a significant effect on the error. Once the function is used, however, this erratic behavior disappears and the curves behave much more similarly to before, with similar convergence rates to the uniform and pin problems (in fact, the error for this problem when integrating the function is actually slightly less than seen in the uniform case as whatever form Equations 3.18 and 3.29 take are better represented by a polynomial than before).

Figures 4.21a and 4.21b show the relative absorption and flux errors along the x-axis at y=0.625 - the third row of watchpoints (the zone centers are not changing in y between different mesh resolutions). These plots are different than previous ones



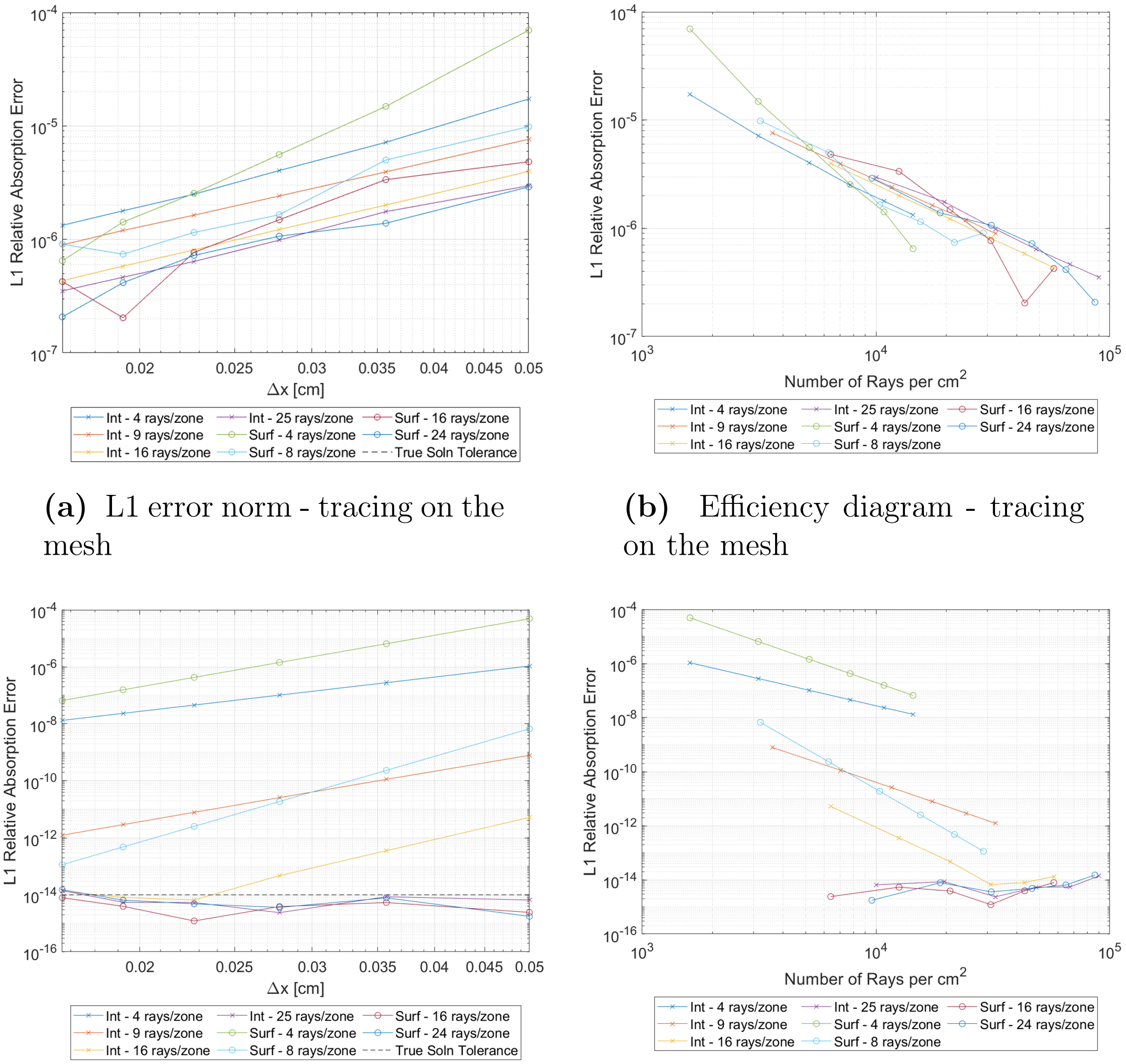
(a) Tracing on the mesh using the interior point method with a 3x3 quadrature

(b) Tracing on the mesh using the surface method with a 4 point quadrature.

(c) Tracing by integrating the function using the interior point method with a 3x3 quadrature.

(d) Tracing by integrating the function using the surface method with a 4 point quadrature.

Figure 4.19 The relative error distribution for the radially-symmetric function. Both the interior point method and the surface method struggle when tracing on the mesh.



(a) L1 error norm - tracing on the mesh

(b) Efficiency diagram - tracing on the mesh

(c) L1 error norm - integrating cross section

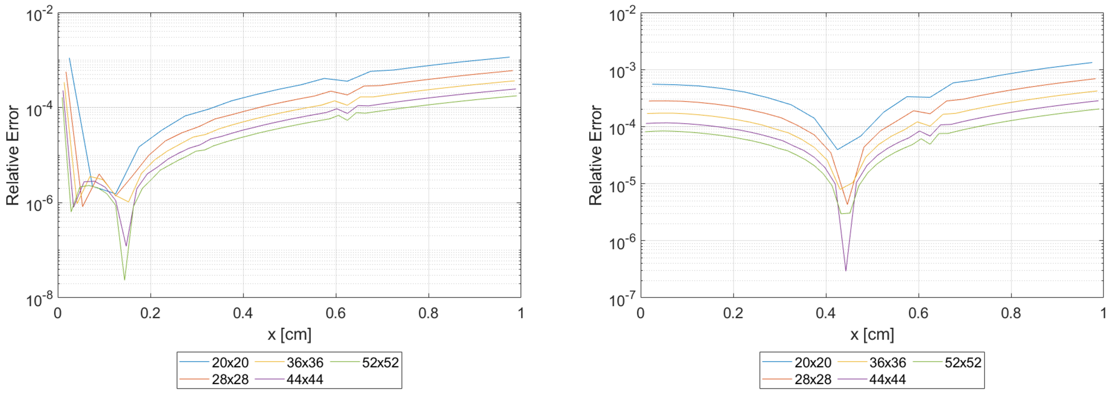
(d) Efficiency diagram - integrating cross section

Figure 4.20 The L1 error norm across the watchpoints and associated efficiency diagram when tracing on the mesh (top) and integrating the cross section (bottom).

shown as they identify the way in which the adaptive solution when tracing on the mesh converges to the true solution found when simply integrating the cross section function. In other words, they are a way in which to try and gauge the accuracy that particular mesh managed to approximate the true solution. Each of these solutions is "correct" (within a tolerance of 1e-14) for the problem it was solving. The differences between them are due to the fact the fundamental problem changes each time the resolution is changed. The problem being solved for with the 20x20 mesh is not the same as the one being solved for with the 28x28 mesh, with all them being approximations of the true solution as given by the cross section function defined in Equation 4.7.

These plots identify two important things: first, that as the mesh resolution is increased, the obtained solution gets closer and closer to the true solution - i.e. the reference solution being used here. This makes sense, and is a good logic check. As the zones get smaller and smaller, the discrete cross section distribution gets closer and closer to the true cross section function.

The second thing that can be seen here is that the absorption and flux errors are different. In all the previous test cases included in this paper, the meshes being used



(a) Absorption error vs x (y=0.625)

(b) Flux error vs x (y=0.625)

Figure 4.21 The relative difference between the solution found when tracing on the mesh and integrating the function, using the integrated function solution as a reference. All solutions were obtained using the adaptive surface method and a convergence criteria of 1e-14.

have aligned the problem geometry in such a way as the cross section throughout the interior of a zone has been constant - essentially, the cross section distribution can be be viewed as one large piecewise-constant equation. This results in the flux and absorption rates being proportional to one another and therefore their error distributions being the same. This is not the case when a zone's cross section distribution changes, as it does in this problem, and ties into a limitation of the surface method, at least in its current formulation. The surface method computes flux by first finding a zone's absorption rate before dividing it by the zone's average cross section and volume. The absorption rate, in turn, is found by taking a surface integral of the current along the zone's surface - i.e. along each of its faces. The flux it computes is, by its very formulation, a zone-average quantity. This becomes an issue when the cross section is changing throughout the interior of a zone. While the zone's total absorption rate can be found without issue (ignoring the effect unsmooth behavior has on the quadratures), there is no information known about the distribution of said absorption rate throughout the zone itself. Even if the cross section distribution inside the zone is known, one cannot obtain an accurate estimation of the average flux as Equation 3.32 cannot be used.

This is one advantage the interior point method has over the surface method, as it computes the scalar flux at each quadrature point throughout the interior of the zone. In fact, the absorption rate can also be found at each quadrature point by simply multiplying the scalar flux by the corresponding cross section and zone volume. This allows one to find both the zone average flux, as well as the zone average absorption rate.

There are ways in which one can work around this limitation of the surface method. For instance, one can still find the average zonal cross section by integrating the cross section distribution throughout the zone and dividing by its volume. If one is willing to approximate the zone's cross section as a constant equaling this average, it can be used in combination with the zone-average absorption rate to approximate the flux. This will, however, result in some amount of additional error. The reality is that because of the surface method's very nature, the interior of a zone is essentially a black box.

Chapter Five

RESULTS - 3D CASES

While 2D problems are useful for looking at key features and the general behavior of the methods, many problems cannot be accurately described in this manner. The real world is 3D, and many problems either do not exhibit the sort of symmetry needed to be simplified to fewer dimensions, or in doing so would prevent one from looking at the feature they are interested in. In addition to this, the way in which quadrature sizes and the effective number of rays scales is different in different dimensions. For instance, in 2D, the surface method uses $2N$ effective rays per zone, while the interior point method uses N^2. In 3D, however, the surface method uses $3N^2$, while the interior point method uses N^3. This difference in particular is very important, as will be shown in the following section.

To demonstrate the ability of the surface method to model problems in three dimensions and establish its general performance against the interior point method, the surface method was implemented in the code ARDRA, a deterministic ($\mathrm{S_n}$) neutron and gamma transport code developed by Lawrence Livermore National Laboratory [15]. This allowed for full-scale, energy dependent, 3D problems to be modeled.

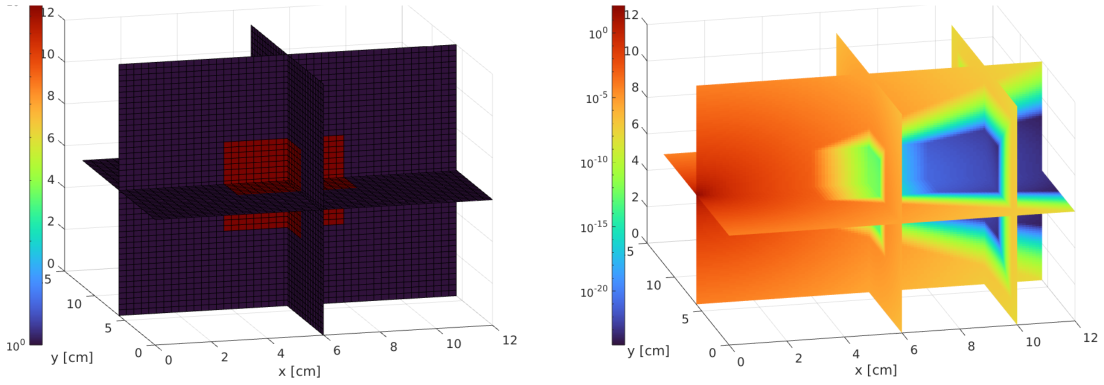
The results from two 3D problems have been included in this paper. The first, a basic cube with a point source located in front of it, is used to characterize both methods general performance. Like the 2D pin problem, the cube captures the behaviors of the methods in uniform material, different or varying cross sections, and the impact unsmooth optical depth distributions have on the solution. The second problem, a simplified version of a skyshine experiment conducted with the RA reactor at the Baikal Research Complex in Kazakhstan, is used to demonstrate both the impact ray effects have on a local solution under real world conditions and showcase the effect the first collision source treatment has on reducing ray effects and improving the solution. The problem also serves as final proof of concept, demonstrating a realistic application of both the first collision source treatment method and the surface method in the sort of problem one might find themselves modeling.

5.1 Cube Problem

The first problem to be modeled using ARDRA was very similar to the 2D pin problem and consisted of a cube of highly absorbing material ($\Sigma = 10$ b) located in the center surrounded by a second, lower absorbing material ($\Sigma = 1$ b). The cube itself occupies the central third of the domain in each dimension. A 3x3x3 set of watchpoints was defined across the domain. As with the previous test cases, a single energy group is used, as the focus is on the uncollided flux, which is not impacted by scattering or secondary particles.

The source itself was located in the center of the y-z plane (in front of the cube) at $x = 0$. This position ensured symmetry across the x-y and x-z planes that run through the center of the cube. The flux distribution for this problem can be seen in Figure 5.1b. As was seen in the pin problem, the cube blocks most of the particles traveling through it and casts a hard shadow on the zones behind it. All plots included have been normalized to a source strength of 1 n/s.

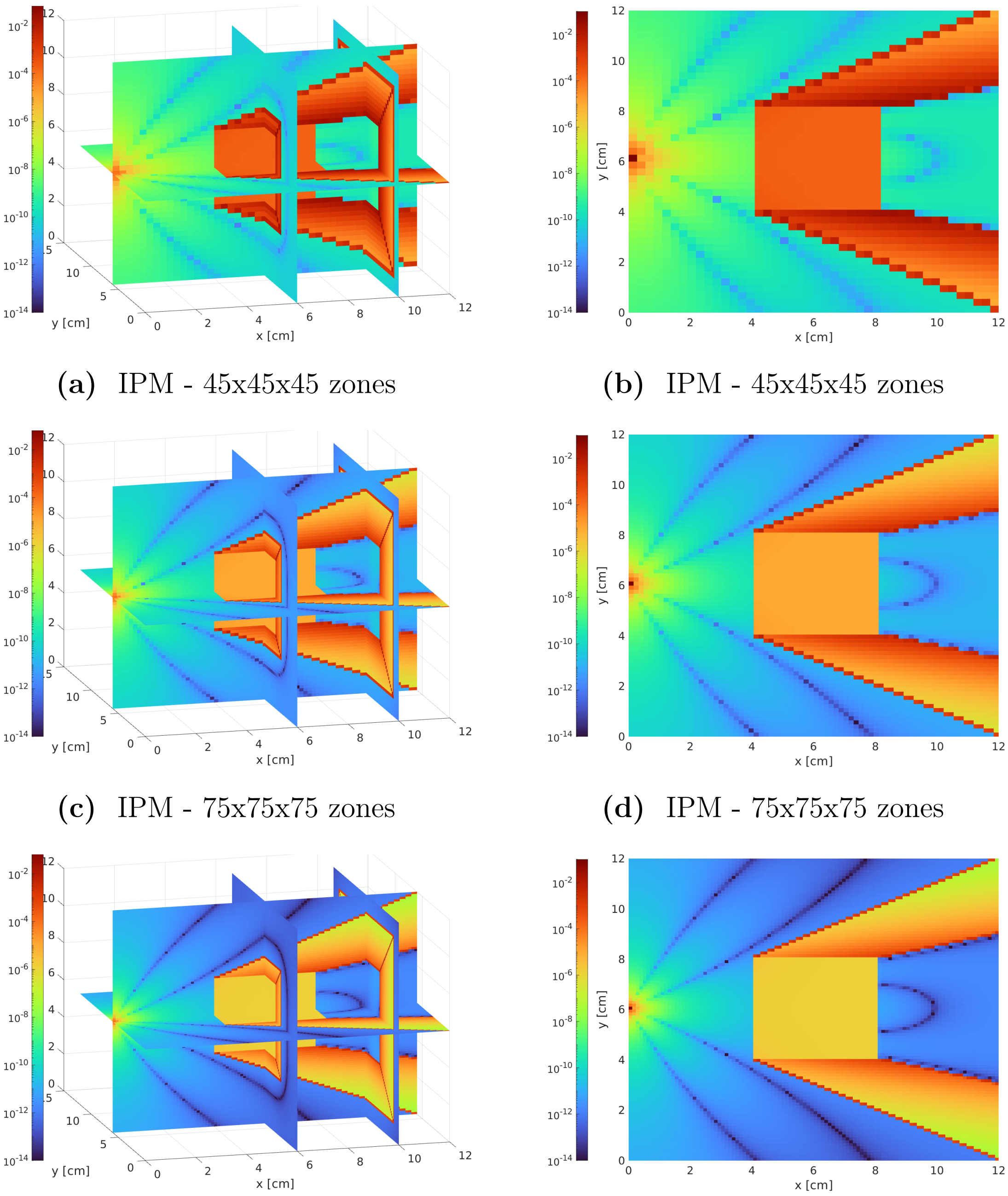
Figures 5.2 to 5.5 show the relative error distributions for the interior point and surface methods for a variety of different mesh sizes and quadrature orders. While the exact distributions do vary between the methods, many of their core features can be seen in both. In general, it can be seen that the error is greatest in the cube itself, as well as the wake along the edges of the cube, particularly in zones extending from the cube's corner. The zones extending from the corner along the shadow observe an unsmooth optical depth distribution, just as was seen in the pin problem, leading



(a) Cross section distribution

(b) Flux distribution

Figure 5.1 The cross section and flux distributions for the cube case with the source located inline with the cube.



(a) IPM - 45x45x45 zones

(b) IPM - 45x45x45 zones

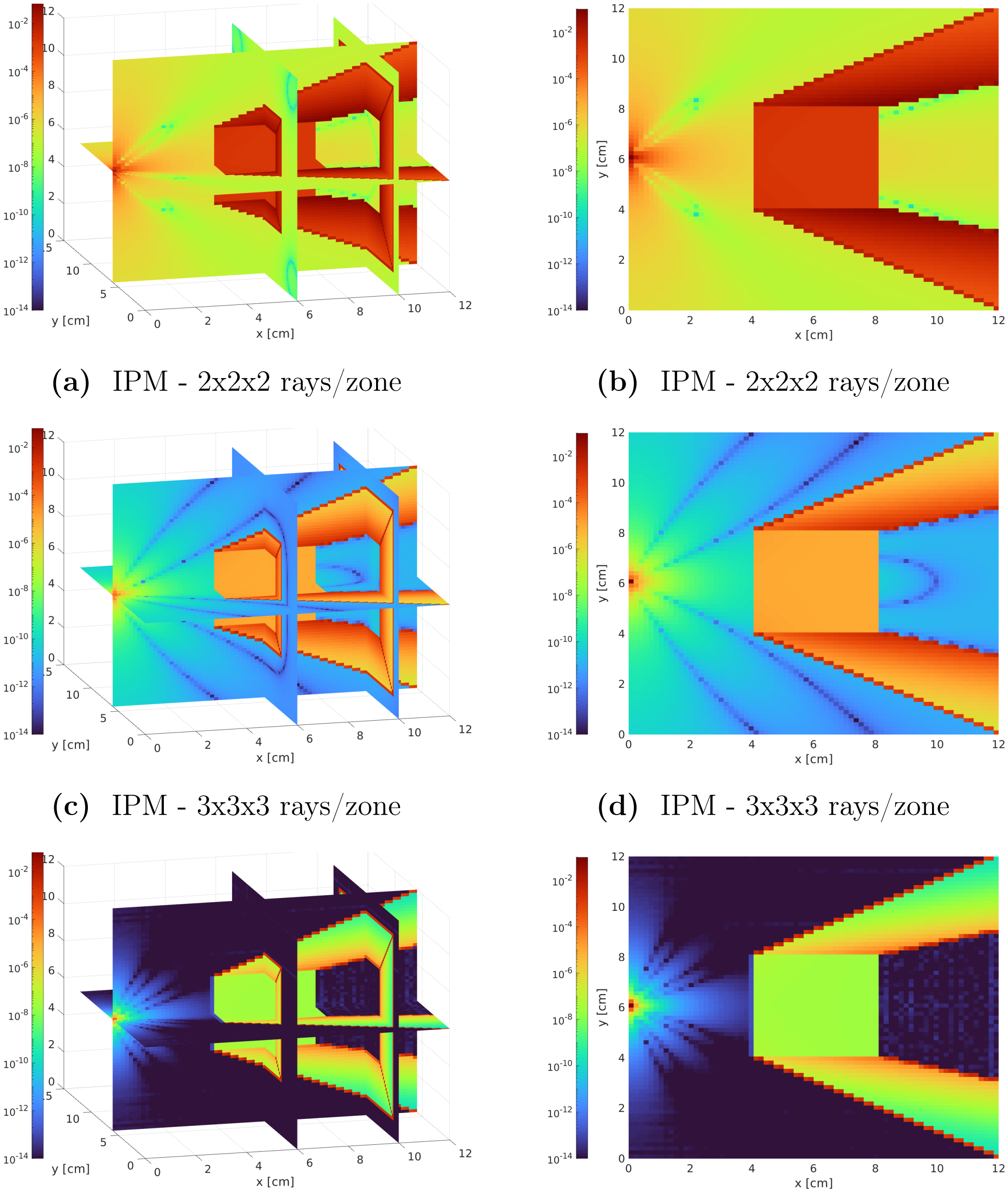
(c) IPM - 75x75x75 zones

(d) IPM - 75x75x75 zones

(e) IPM - 105x105x105 zones

(f) IPM - 105x105x105 zones

Figure 5.2 Relative error distributions for the interior point method using a 3x3x3 quadrature.



(a) IPM - 2x2x2 rays/zone

(b) IPM - 2x2x2 rays/zone

(c) IPM - 3x3x3 rays/zone

(d) IPM - 3x3x3 rays/zone

(e) IPM - 4x4x4 rays/zone

(f) IPM - 4x4x4 rays/zone

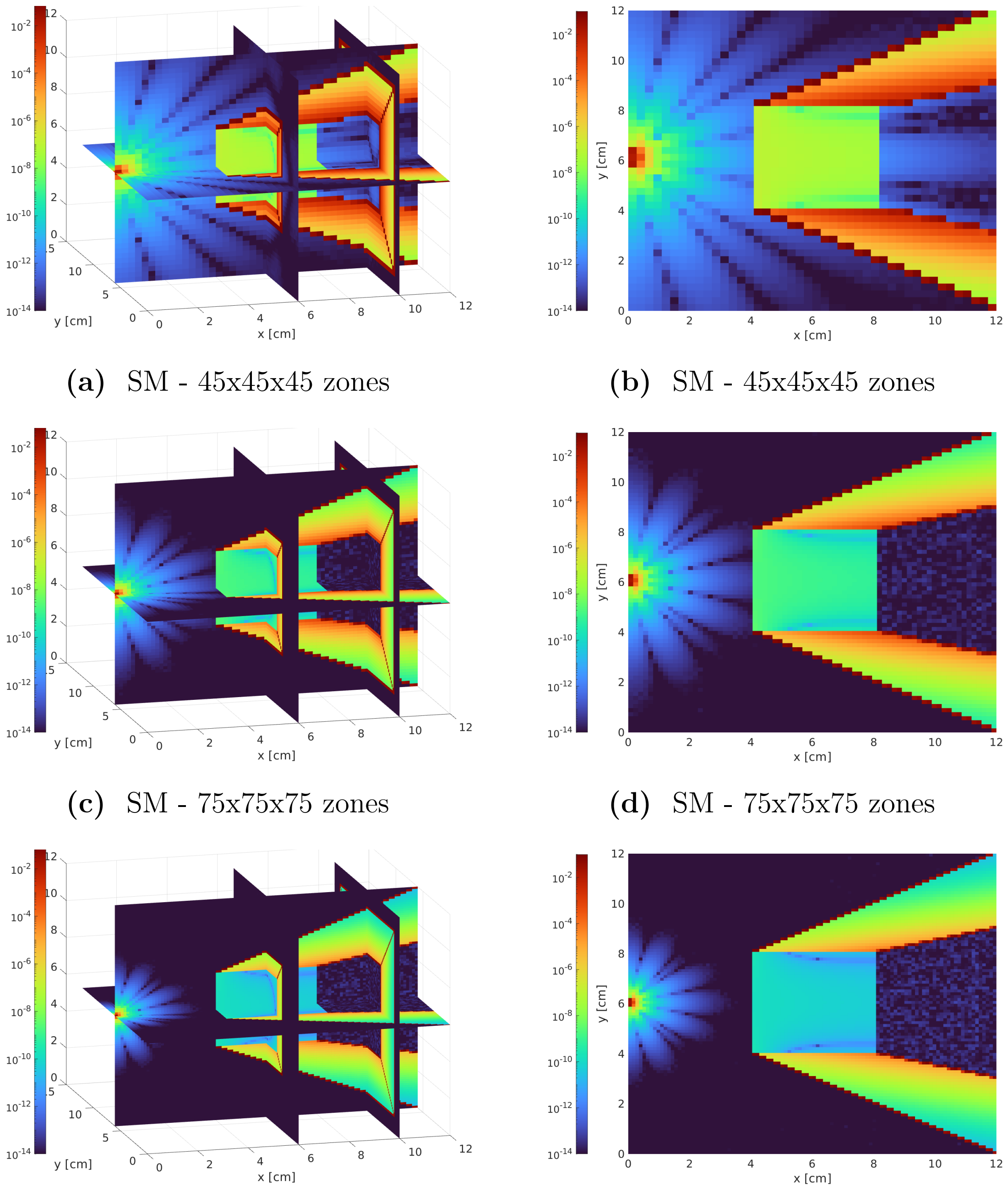
Figure 5.3 Relative error distributions for the interior point method using a 75x75x75 zone mesh.

to significantly higher error than in other regions of the problem.

The exact configuration of this problem is not one to one with the pin problem, as the sources are located differently. To truly be the 3D equivalent of the pin problem, the source would need to be located at $(x, y, z) = (0.0, 6.0, 0.0)$. One of the core differences between these two problems can be seen in the zones directly behind the cube. In the pin problem, these zones in general observed higher error, similar to that inside the cube. Here, the zones see the same error trends as those which never intersect the cube. In general, with exception to zones near the source or experiencing an unsmooth optical depth, the local flux error is intrinsically tied to the degree of variation in the optical depth and cross section variation inside the zone. In this problem, the quadrature points for zones inside the cube have enough variation in their optical depth to result in noticeably higher error. The reality is that whatever function that describes the interior optical depth distribution for these zones is not a polynomial and therefore cannot be perfectly captured by a Gaussian quadrature. The same can be said for zones in the interior of the wake, but to an even greater degree, as the difference in how far the particles traveled through the cube for each quadrature point is even more pronounced. The impact this variation in optical depth has on the overall error of the zone is more observable for the interior point method than the surface method, being consistently worse for similar amounts of work.

As we get to zones directly behind the cube, the error suddenly drops to levels similar to those zones which never intersected the cube. While the distance traveled through the cube will vary slightly from quadrature point to quadrature point, the difference is much less than before when actually inside the cube.

It should be noted that because this problem is in 3D, unsmooth optical depth distributions do not just occur in zones along the corners of the cube, but any edges. In fact they are not just limited to zones along the surface of the wake, but even throughout the interior of the wake. Unfortunately, while it is relatively simple to observe the error emerging from the edges formed by the front and top faces or front and bottom faces in Figures 5.2 and 5.3, it is much more difficult to observe what the error looks like in zones truly along the corner or along the other edges, like that formed by the intersection of the right and top face. If one looks closely at 5.2c, along the y-z slice running through the middle of the cube, in the top right quadrant, one can make out a diagonal line running from $(x, y, z) = (6, 8, 8)$ to around $(x, y, z) = (6, 9, 9)$. This line, caused by the edge between the top and right face, is able to shine light on what the error looks like throughout the interior of the wake. It is worth noting that although the error along this line appears slightly higher than many of the other zones observing unsmooth optical depth distributions, the error is



(a) SM - 45x45x45 zones

(b) SM - 45x45x45 zones

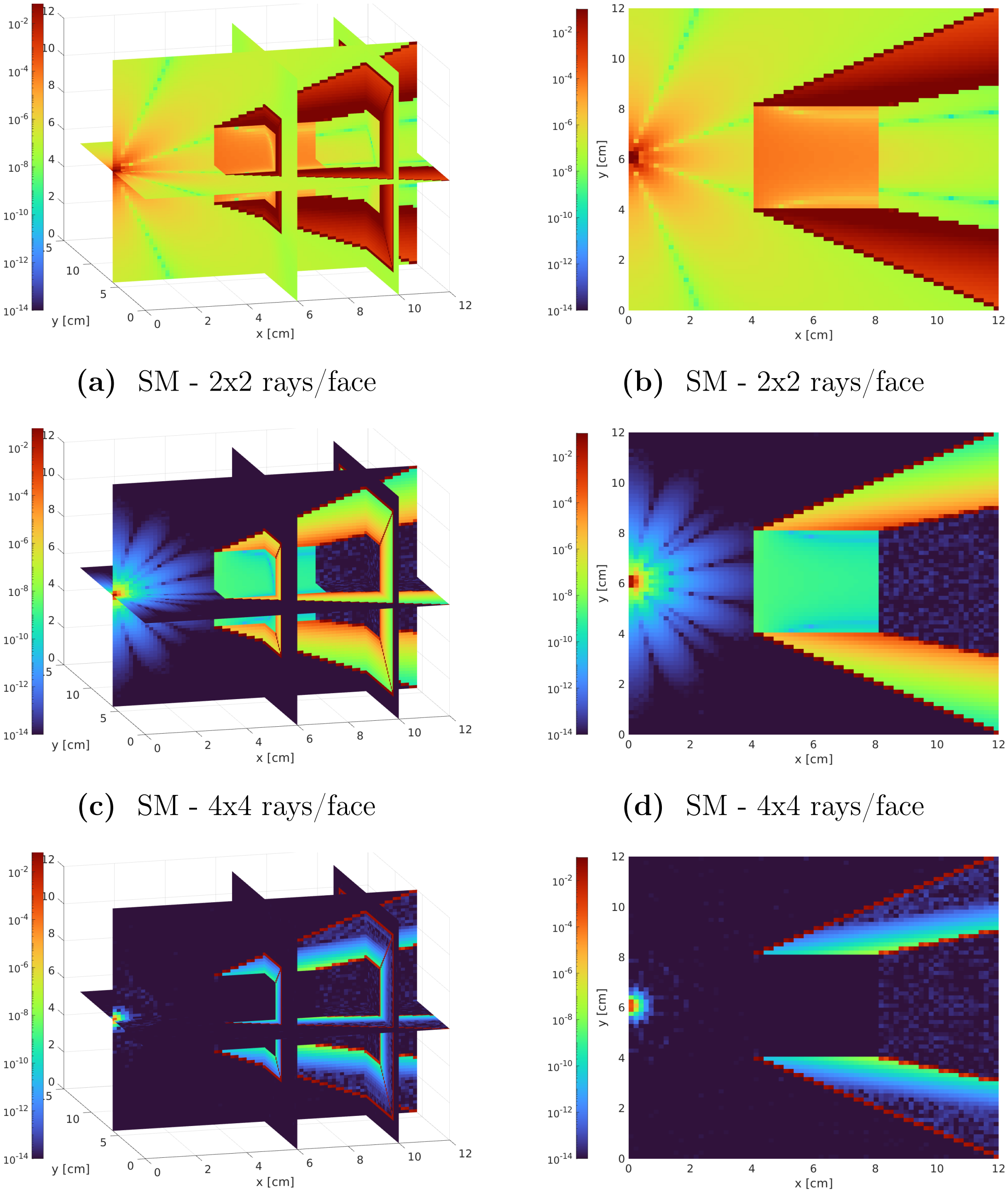
(c) SM - 75x75x75 zones

(d) SM - 75x75x75 zones

(e) SM - 105x105x105 zones

(f) SM - 105x105x105 zones

Figure 5.4 Relative error distributions for the surface method using a 4x4 quadrature.



(a) SM - 2x2 rays/face

(b) SM - 2x2 rays/face

(c) SM - 4x4 rays/face

(d) SM - 4x4 rays/face

(e) SM - 6x6 rays/face

(f) SM - 6x6 rays/face

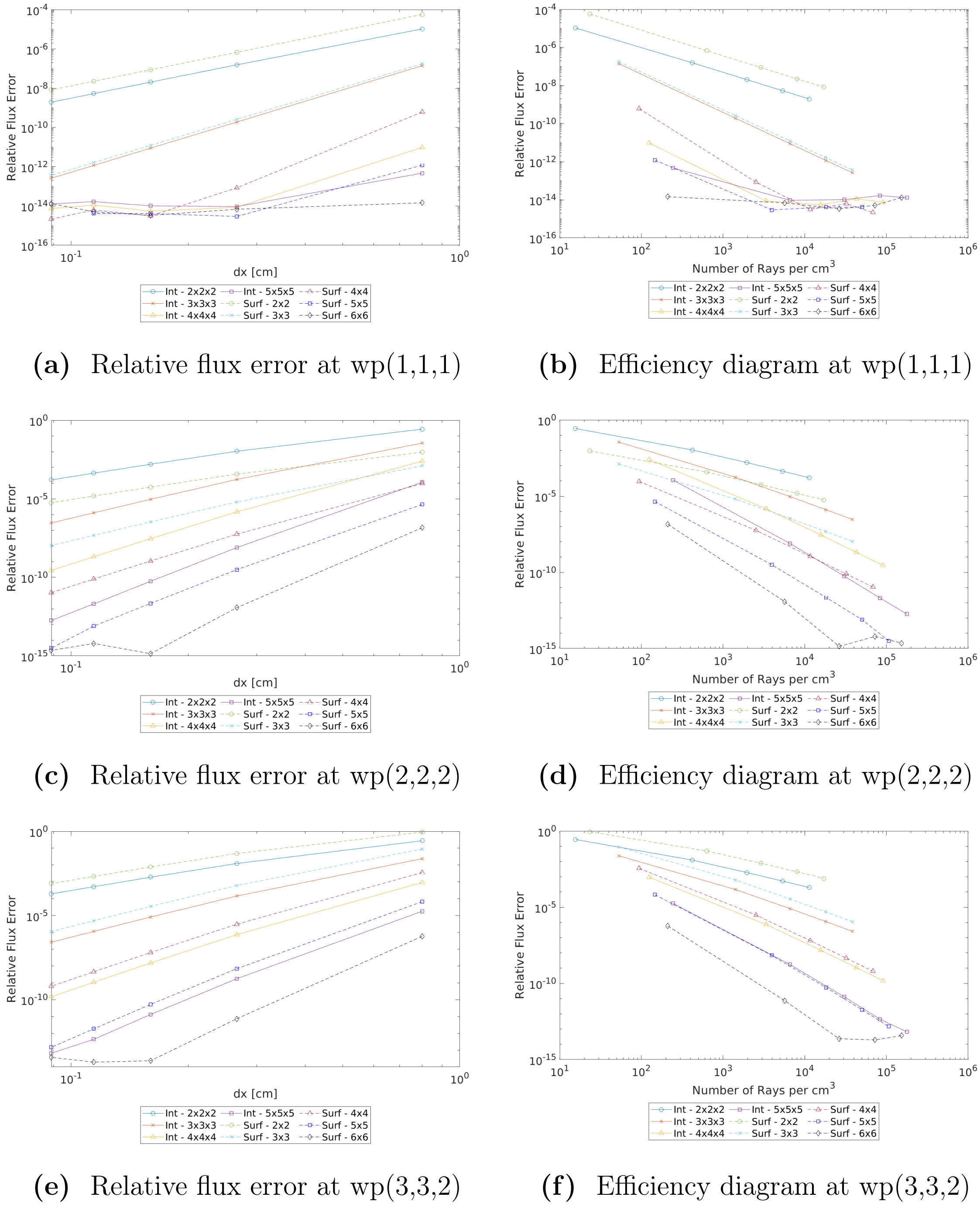
Figure 5.5 Relative error distributions for the surface method using a 75x75x75 zone mesh.

so high for all these zones one must be careful not to read too much into any trends and make any general observations. Similarly, while it may be reasonable to assume a zone observing unsmooth optical depth distributions along multiple axes, like one would see in a zone along the diagonal of the corner, would experience greater error than those with a single line of discontinuity, the error is so high one cannot make any general observations beyond what has already been said.

Just as was observed previously in the 2D cases, it is evident that while increasing mesh resolution does improve the solution - both locally and globally - increasing the quadrature order is a more efficient way of decreasing error. This is supported both by the error distributions in Figures 5.2-5.5, as well as the relative error and efficiency diagrams in Figure 5.6. It is important to note, however, that while increasing quadrature order is more efficient in terms of work being done than increasing spatial resolution, increasing spatial resolution gives more data points. While this statement may seem obvious, its important to strike a balance between the two, resolving the solution with enough resolution to properly capture important features, while at the same time using a high enough order quadrature to accurately solve the problem.

Going back to Figure 4.8, its worth remembering that in 3D, when using Gauss-Legendre, the surface method actually uses more effective rays for low-order quadratures than the interior point method. This changes once one uses 3 points per dimension or more, but is worth pointing out.

The difference in efficiency between the surface and interior point methods is more pronounced in 2D than in 3D. For Gauss-Legendre in 2D, when taking quadrature point sharing into account, the surface method uses $2N$ quadrature points per zone (where N is the number of points per dimension), whereas the interior point method, lacking that ability to share points, uses N^2. Looking at Figure 4.8a, it can be seen how the methods use the same number of effective rays per zone at 2 points per dimension, with the interior point method using more and more rays relative to the surface method as the quadrature size is increased. In 3D, however, the scaling is considerably different, as can be seen in Figure 4.8b. Here, it is not until 3 quadrature points per dimension that the surface method begins outperforming the interior point method in rays cast per quadrature size, and even then the relative scaling between the two methods remains significantly worse than it was in 2D. Now, the surface method is casting $3N^2$ rays per zone, while the interior point method casts N^3 rays. This particularly impactful at lower order quadratures (of roughly 6 or fewer points per dimension), as the total effective number of rays for the surface method is much closer to that of the interior point method than it was before. This is important to remember when attempting to analyze the performance and efficiency of the methods for the cube problem, as roughly equivalent quadrature sizes in 2D, like 4 points per



(a) Relative flux error at wp(1,1,1)

(b) Efficiency diagram at wp(1,1,1)

(c) Relative flux error at wp(2,2,2)

(d) Efficiency diagram at wp(2,2,2)

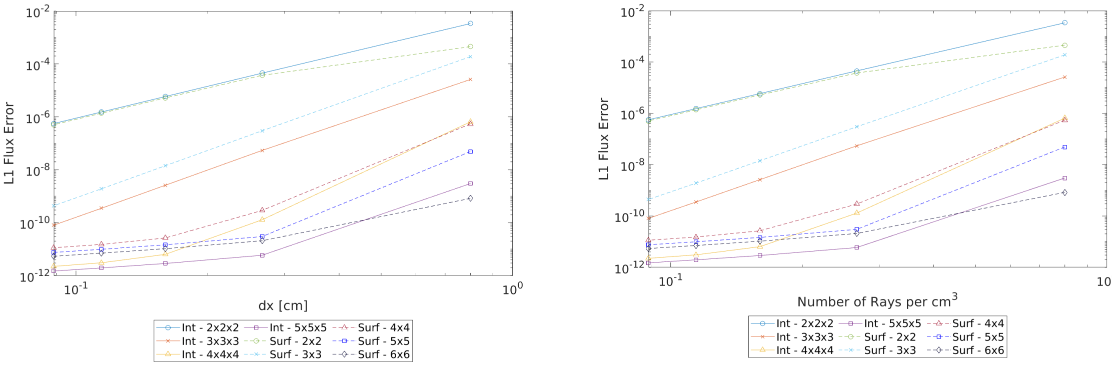
(e) Relative flux error at wp(3,3,2)

(f) Efficiency diagram at wp(3,3,2)

Figure 5.6 The relative flux error and associated efficiency diagram at several watchpoints for the cube problem.

dimension for the surface method and 3 points per dimension for the interior point method, are no longer that similar to one another. In fact, interestingly, if one looks at the error curves for 3 points per dimension at wp(1,1,1) (when the two methods are doing the same amount of work for the same quadrature size) in Figures 5.6a and 5.6b, the two methods perform near identically - both in terms of magnitude and shape. Comparatively, however, if one compares the curves for 4 points per dimension, the interior point method significantly outperforms the surface method at the cost of using only a quarter more rays. However, if we then compare the interior point method using a 4x4x4 quadrature with the surface method's 5x5 quadrature (64 rays per zone for the interior point method vs 75 for the surface method), the surface method overtakes it in performance. This gap is then extended further when one compares the 5x5x5 quadrature (125 rays per zone) of the interior point method to the 6x6 quadrature (108 rays per zone) of the surface method. While its difficult to draw any strong conclusions from this watchpoint alone, what it does show is how the gap in performance slowly begins to widen as the quadrature order increases. At low quadrature orders, the two methods perform relatively similarly, with the interior point method actually outperforming the surface method not just magnitude of error but also efficiency. As the quadrature order increases and difference in effective rays begins to take effect, the surface method slowly overtakes the interior point method in performance.

The error plot and efficiency diagram for wp(2,2,2) (in the cube) in Figure 5.6 is rather different from those seen in wp(1,1,1). Although there are no zones inside the



(a) L1 error comparison

(b) Global efficiency diagram

Figure 5.7 The global L1 flux error and its associated efficiency diagram.

cube itself that have an unsmooth optical depth, the error for both methods is significantly higher here than it was outside the cube. As was noted earlier, the magnitude of the error tends to be tied to the degree of variation in the optical depth, as well as how that variation in turn effects the behavior of the current/flux distribution. The less polynomial-like the distribution, the worse the Gaussian quadrature is able to numerically integrate it, and the worse the error is. Unlike before in the lower cross section region described by wp(1,1,1), the error for the interior point method is significantly higher than that of the surface method, although both methods perform significantly worse than they did at wp(1,1,1). Whatever form the flux distribution is taking in this region is less well behaved than the current distribution along a single face. While this significantly impacts the magnitude of the error, the slope of the curves remains consistent with that observed previously, and still directly correlated to the quadrature order. Both methods, when using the same quadrature order, have roughly the same slope.

Looking at the plots for wp(3,3,2) (inside the shadow), the superior performance of the surface method over the interior point method no longer hold true. This is similar to what was observed in the pin problem when using low order quadratures. For the same order of quadrature, both methods tend to have roughly the same convergence rate - although often slightly favoring the surface method (the exact convergence rate varies depending on the problem and where one is in the mesh); however, the overall magnitude of the error tends to favor the interior point method. It is only once the surface method is able to take advantage of its increased capacity for sharing quadrature points that it begins to outperform the interior point method. This normally happened relatively quickly for problems in 2D when the scaling of effective rays vs quadrature size heavily favored the surface method. In 3D, however, that scaling happens significantly slower. Additionally, having to sum the result of multiple quadratures together creates possibilities for errors to add together. The surface method, although often more efficient, is in many ways more fragile than the interior point method. This trend of similar convergence rates but worse error for the surface method is somewhat contradicted inside the pin itself. Outside the cube, for the same quadrature order and mesh resolution, the interior point method will typically outperform the surface method, although it will typically have to do significantly more work to accomplish this.

5.2 Skyshine

The final problem to be included in this paper is a simplified version of a skyshine problem. Skyshine is a phenomena often observed in atmospheric transport in which radiation (both neutrons and photons) emitted from a source scatters off the atmosphere before coming back to the surface. Particles traveling through air typically exhibit anisotropic, forward scattering behavior. These indirect particles can result in significant fluxes in directions other than the source, as particles are essentially curving in from the sides and above in a cone-like distribution. At a certain distance, particles traveling from these directions can actually outweigh those coming directly from the source. This effect would have been very present during the Chernobyl accident, as well as the multitude of bare-reactor experiments conducted throughout the 20th century [16, 17, 18, 19]. Another example in which skyshine can play a significant role in is the detection and recovery of radioactive materials, whether they be lost in a field or are being smuggled through a border crossing or port.

Consider the hypothetical scenario in which a radioactive source used for surveying purposes has been lost due to a recent flood. The loss of the source was reported, and a team has since been dispatched to help with its retrieval. The source's rough location has been narrowed down to a field with an area of say few km^2; however, the field itself consists of long, tall grass, and the source is relatively small. Being professionals, the team came prepared for this, bringing with them several handheld radiation detectors. As the team goes out and begins performing a radiological survey of the area, they will not only receive a signal in the direction of the source, but potentially everywhere around them, complicating the process of identifying the exact location of the source. Depending on the distance they are from the source and the geography of the region, the signal they measure may be very strong in directions other than the source.

When modeling problems like this, one's source can often be approximated using a simple point source, as the source is typically much smaller than the overall problem domain. This approximation is more accurate the smaller one's source actually is and the further away one is to it. For something like a small AmBe source, the approximation would likely remain accurate even at close distances, while for a bare reactor like the one used in Operation BREN [18], the minimum distance would be significantly larger - likely tens if not hundreds of meters. Regardless of this, however, is that while using a point source is much simpler to model, they tend to suffer from significant ray effects when used in a discrete ordinates method, as discussed previously. This can lead to problems if one is trying to, say, find the energy deposition and corresponding signal in a small detector hundreds of meters away from the source. Getting an accurate solution in this sort of scenario using the method of discrete ordinates is essentially impossible without the employment of ray effect mitigation techniques, as the computational cost for the size of angular quadrature needed to obtain an accurate solution would likely be infeasible.

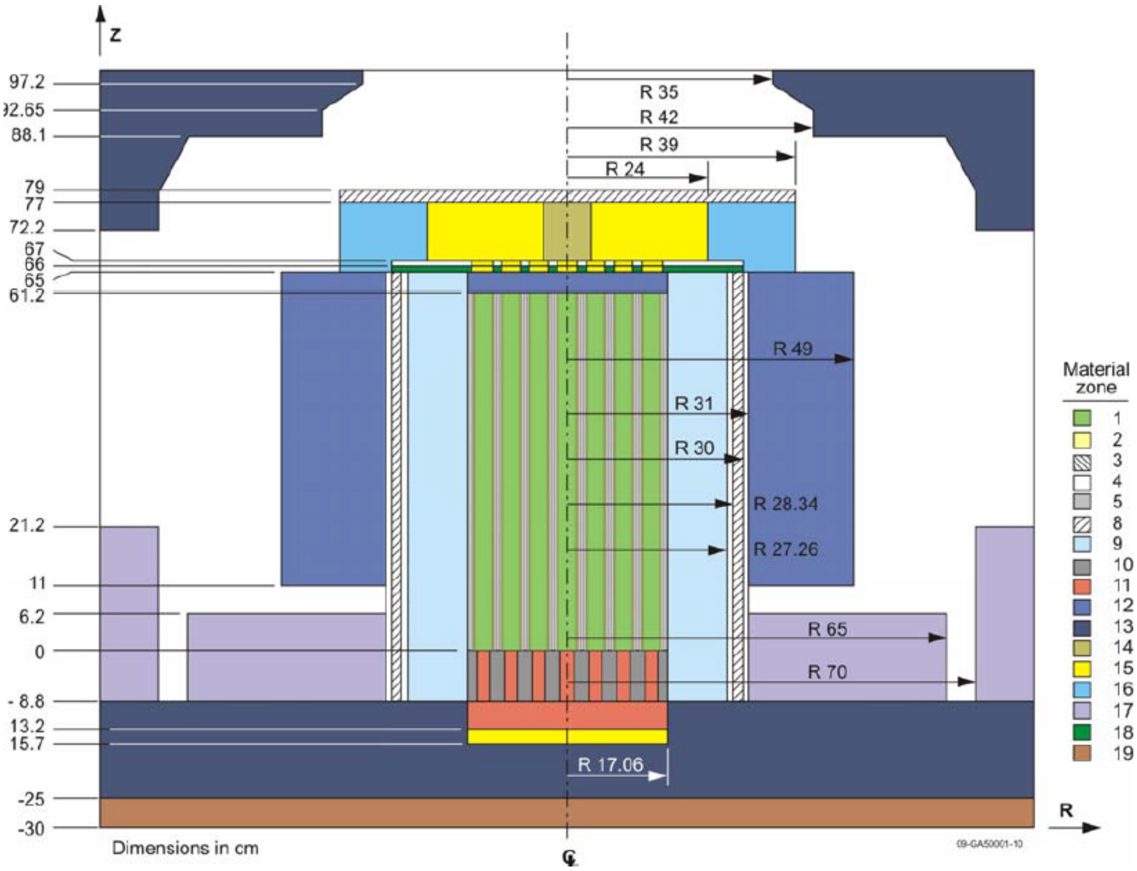


Figure 5.8 A model of the RA reactor core at the Baikal test facility. Image taken from [16].

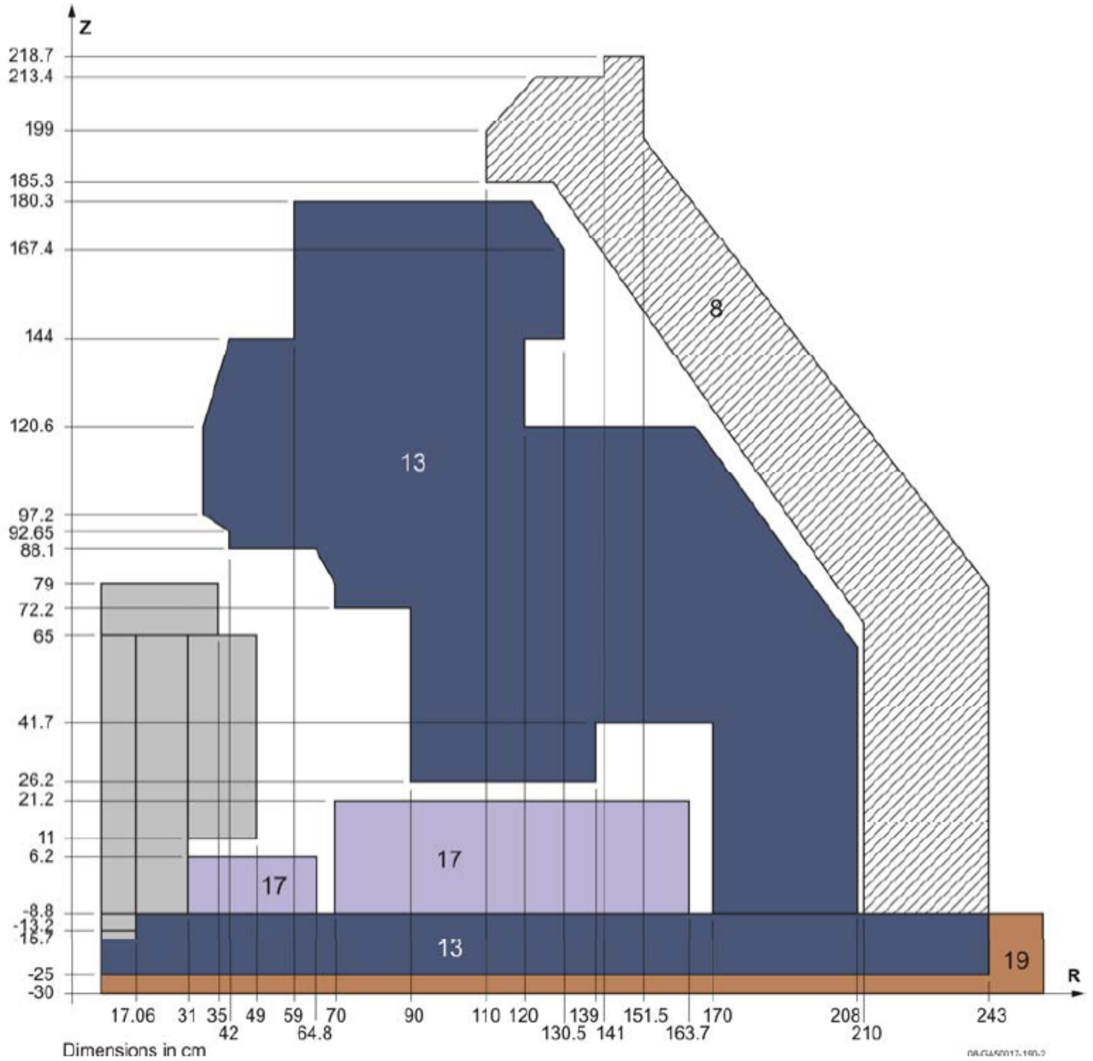


Figure 5.9 A model of the RA reactor facility. Image taken from [16].

To demonstrate the significance ray effects can have on a local solution in a problem like this, an approximation of a skyshine experiment conducted at the Baikal test facility was modeled, which can be seen in Figure 5.10. This involved combining a point source with the same energy and angular profile as that produced by the reactor described in Figures 5.8 and 5.9, and performing the atmospheric transport using ARDRA. Because of limitations with ARDRA and the method of discrete ordinates in general, the ground could not be included in the model, as this would exceed a single mean free path per zone, destabilizing the solution. To properly include the ground, the zone size in this region would need to be significantly smaller, something that cannot be done in such a large problem due to memory constraints.

While there may be possible ways around this limitation, such as the inclusion of slightly reflective boundary conditions to approximate the ground or the modeling of the ground in local regions around detectors, these were deemed outside the purview of the project and not pursued further. Unfortunately, this prevented any possibility of properly comparing against any experimental results [16, 17]. Instead, a reference solution was computed using the Monte Carlo code Mercury [24].

The general flux distribution for the angularly dependent source can be seen

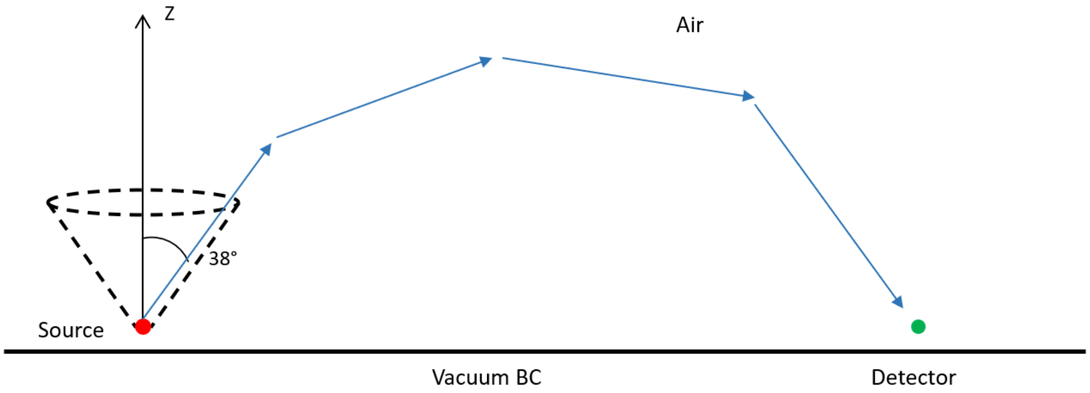


Figure 5.10 The problem setup for a basic skyshine problem. A point source is located 1 m above a vacuum boundary at (x,y,z) = (0,0,1). There are reflective boundary conditions along the x-z and y-z planes running through the origin. The point source is restricted to emitting in a 33° cone along the z-axis. Detectors are located at 100 m increments along the x-axis, in line with the source.

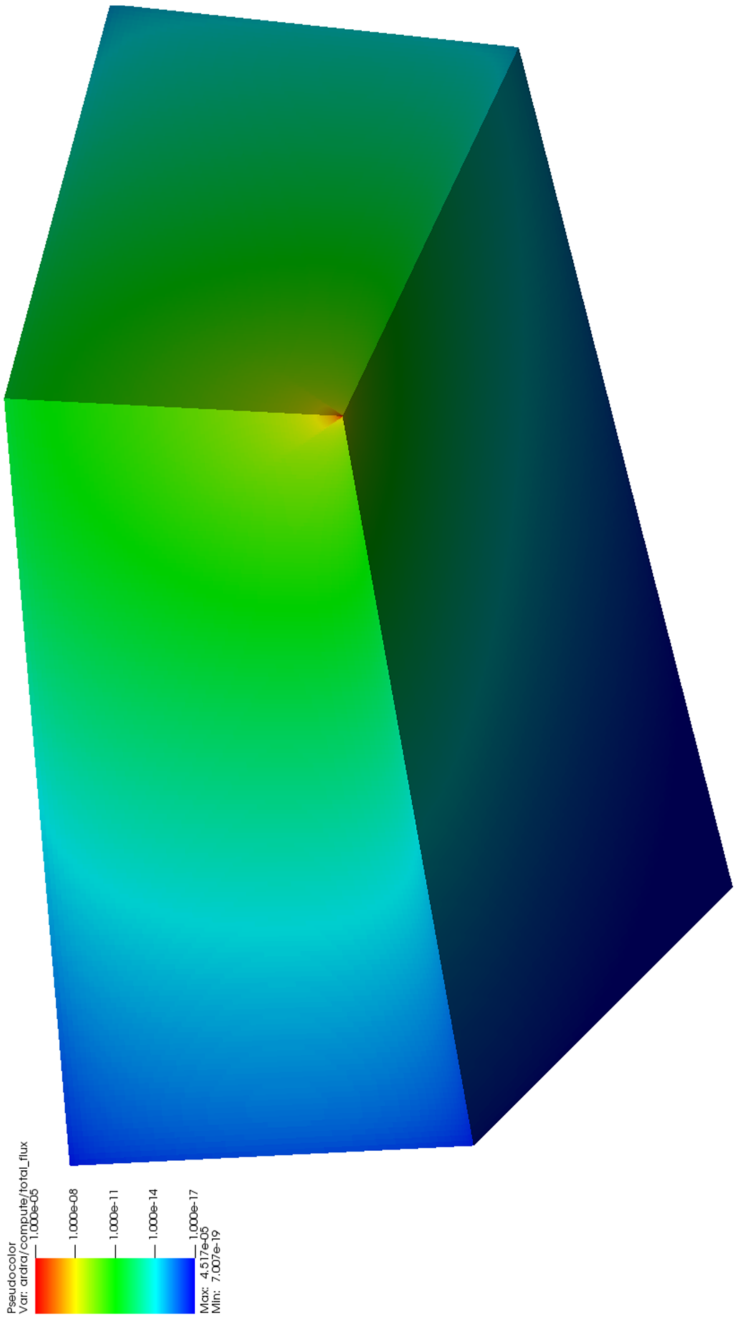


Figure 5.11 The flux distribution for an isotropic source with a void at z = 0.0 cm. Units are in cm.

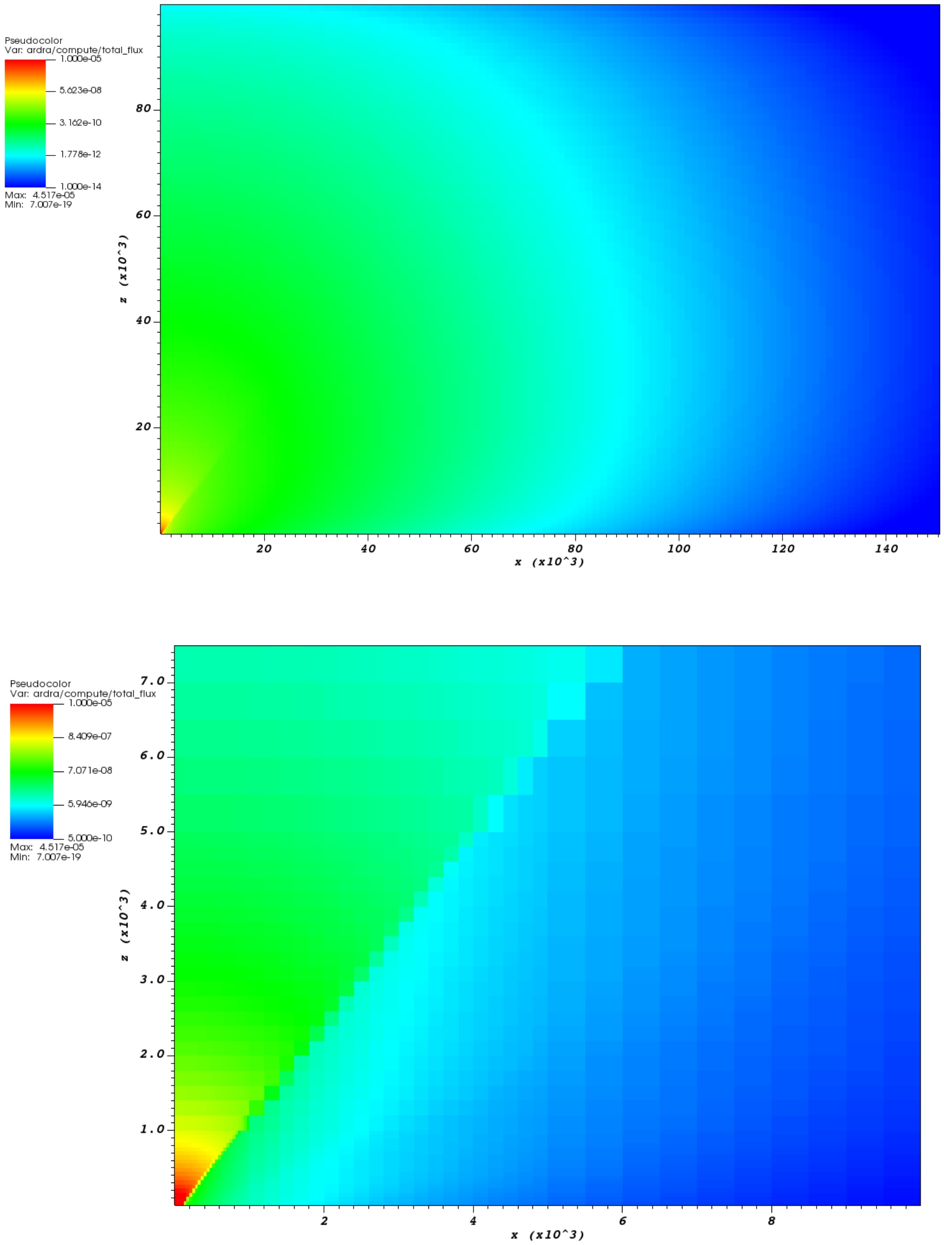


Figure 5.12 x-z slices (at y = 0.0 cm) of the flux distribution for a conical source with a void at z = 0.0 cm. Units are in cm.

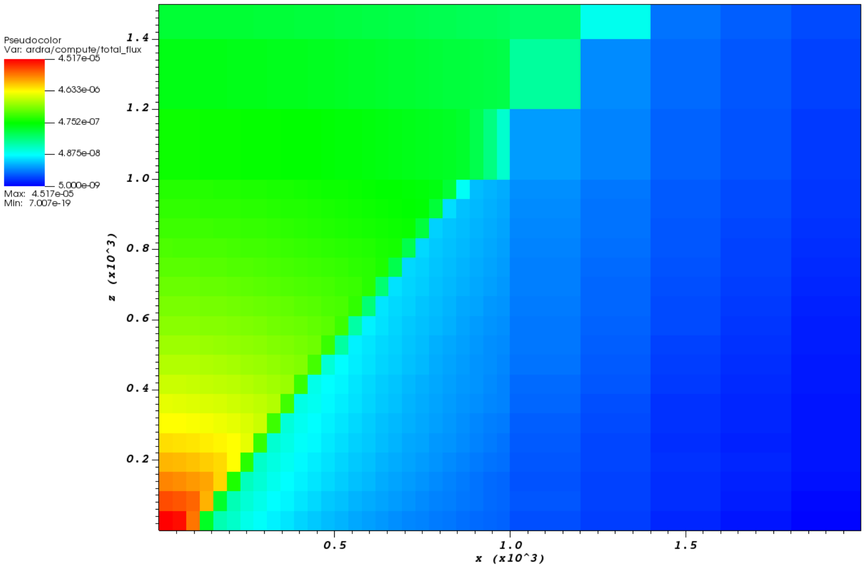


Figure 5.13 An x-z slice (at y = 0.0 cm) of the flux distribution for a conical source with a void at z = 0.0 cm (near the source). Units are in cm.

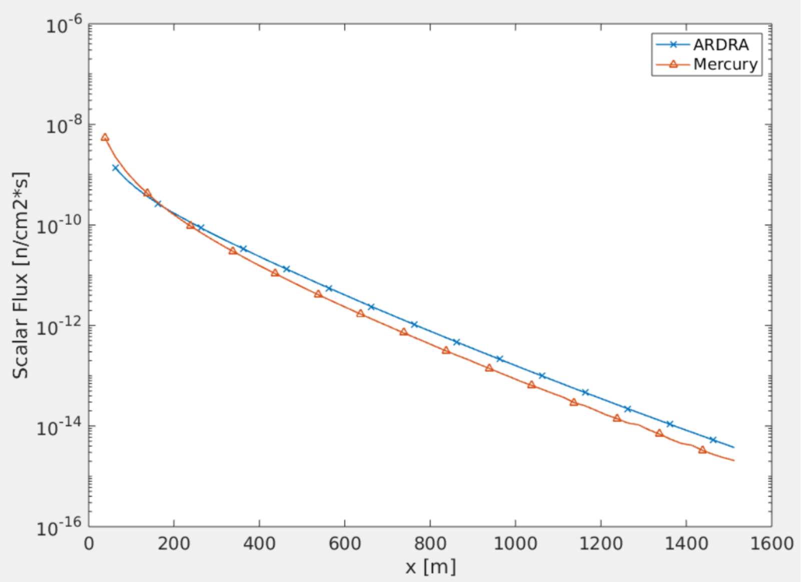


Figure 5.14 The flux distribution of the conical source along the x-axis compared to the results of the Monte Carlo code Mercury.

in Figures 5.11, 5.12, and 5.13. Although the source is emitting only in upwards direction, the anisotropic scattering behavior of the neutrons in air quickly smooths the particle distribution out in directions outside the source's original cone such that it quickly becomes difficult to see any evidence of the cone. The flux along the vacuum boundary condition at $z = 0.0$ is noticeably lower than in regions of similar distances due to particles being lost to the void. To best capture the sources angular behavior, the mesh gradually coarsens the further away it is to the source.

A comparison of the flux along the x-axis between Mercury and ARDRA can be seen in Figure 5.14. Although the general shape and rough order of magnitude is similar, there is a noticeable difference between the two solutions. Although the exact cause of the discrepancy between the two codes was never identified, it is likely due to the angularly dependent nature of the source. While modeling the problem, it became apparent that the initial distribution found using the ray trace had significant impact on the overall flux distribution. Things like high initial errors in the uncollided flux in zones near the point source could significantly impact the scattered flux in later iterations.

The modeling of angularly dependent sources presents a notable challenge for both the interior point and surface methods when attempting to calculate the uncollided flux. As discussed in previous chapters, both methods are heavily impacted by the ability of their quadratures to accurately integrate across a face/zone and get an average current/flux. In those aforementioned cases, unsmooth optical depth distributions led to very high errors in zones where quadrature points straddle a material interface.

Here, the angular dependency of the source leads to a similar effect, with the source term q in Equations 3.19 and 3.27 now a function of $\hat{\Omega}_s$ and able to vary across quadrature points along the same zone/face, leading to discontinuities in the flux distribution. These discontinuities pose a serious issue to the Gaussian quadratures ability accurately integrate the distributions. Just like before when dealing with the unsmooth optical depth distributions, the best way to handle these discontinuities is to localize the error they cause by running the problem adaptively.

A further complication caused by the source's angular dependency can be seen when using the surface method. Because the surface method is rooted in Equation 3.22, zones containing or in contact with the source must account for N_{src} when going from N_{leak} to N_{abs}. When the source is fully enclosed by the zone, N_{src} is simply the total source strength over all angles. However, if instead the source is located on a face, one must find the fraction of the source entering the zone. When the source is isotropic, this relatively straight forward, as it is always 1/2 if it is on the interior of a face, 1/4 if on an edge, and 1/8 if on a corner. When the source is not isotropic, however, one must integrate the source's angular distribution over the zone, something that is not trivial in nature. While this process can be circumvented by ensuring the source is located inside a zone, this then prevents one from using reflective boundaries, as the source would then be reflected across the boundary if not placed on the boundary itself.

To further demonstrate this behavior, the same problem as before was modeled again, this time with an isotropic source rather than an angularly dependent one. The corresponding flux distributions can be seen in Figures 5.15, 5.16, and 5.16. Comparing these Figures to those of the conical source, one can see how near the source, the distributions are fairly different. However, as one gets further away from the origin, the differences in the flux between the two problems become smaller and smaller. Looking at Figure 5.18, the difference between the solution found using Mercury and that of ARDRA is much less than that in the previous configuration.

To demonstrate the impact the first collision source treatment method has on the final flux distribution, the isotropic source was again modeled, this time without using the first collision source treatment and without the reflective boundaries or the vacuum boundary condition at $z = 0.0$. The problem domain was also scaled down to only a few hundred meters, as the primary focus of this configuration is on zones relatively near the source. The problem was run four times, with the results being shown in Figures 5.19, 5.20, 5.21, and 5.22. Figure 5.19 was produced using an S_8 angular quadrature (80 bins), Figure 5.20 using an S_{10} quadrature (120 bins), and Figure 5.21 using an S_{12} quadrature (168 bins). The final run was performed using the first collision source treatment with an S_8 angular quadrature.

When using an S_8 quadrature without any ray effect mitigation, significant ray effects can be observed propagating from the source throughout the mesh. Compared to the distribution found when using the first collision source treatment in Figure 5.22, the solution in Figure 5.19 is almost star-like in comparison, as the ray effects cause significant distortions along the angular bins of the mesh. When the angular resolution is increased from S_8 to S_{10}, the solution becomes slightly rounder. Although there are more individual ray effects than before, their overall size is less. Because of this, it takes less distance before they smoothed out from scattering.

Increasing the angular quadrature again from S_{10} to S_{12}, the same behavior can be observed, with the solution slowly appearing more and more like that found with the first collision source treatment in Figure 5.22. It should be noted, however, that while there will still be ray effects in the solution, they are much less impactful than those observable in Figure 5.21, despite using less than half the number of angular bins (80 vs. 168), to the point where they are no longer visible to the naked eye. The first collision source treatment only eliminates ray effects caused by the uncollided

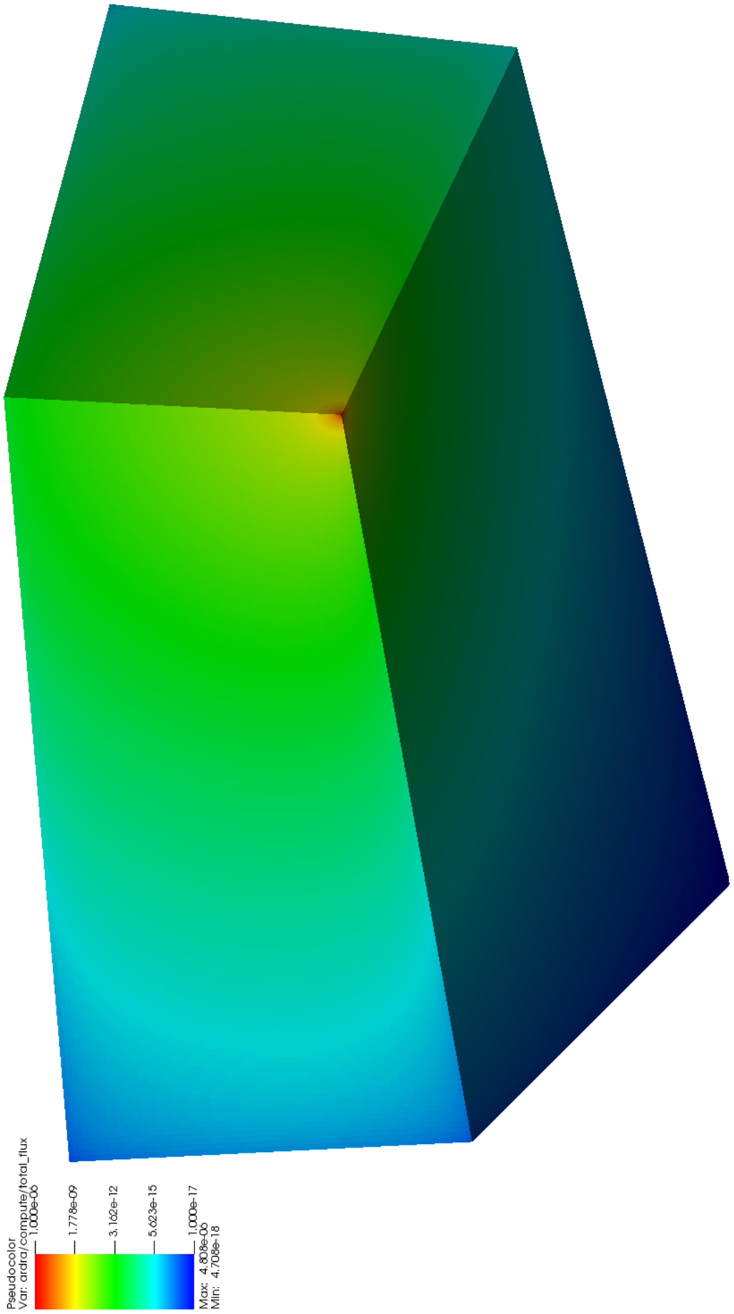


Figure 5.15 The flux distribution for an isotropic source with a void at z = 0.0 cm. Units are in cm.

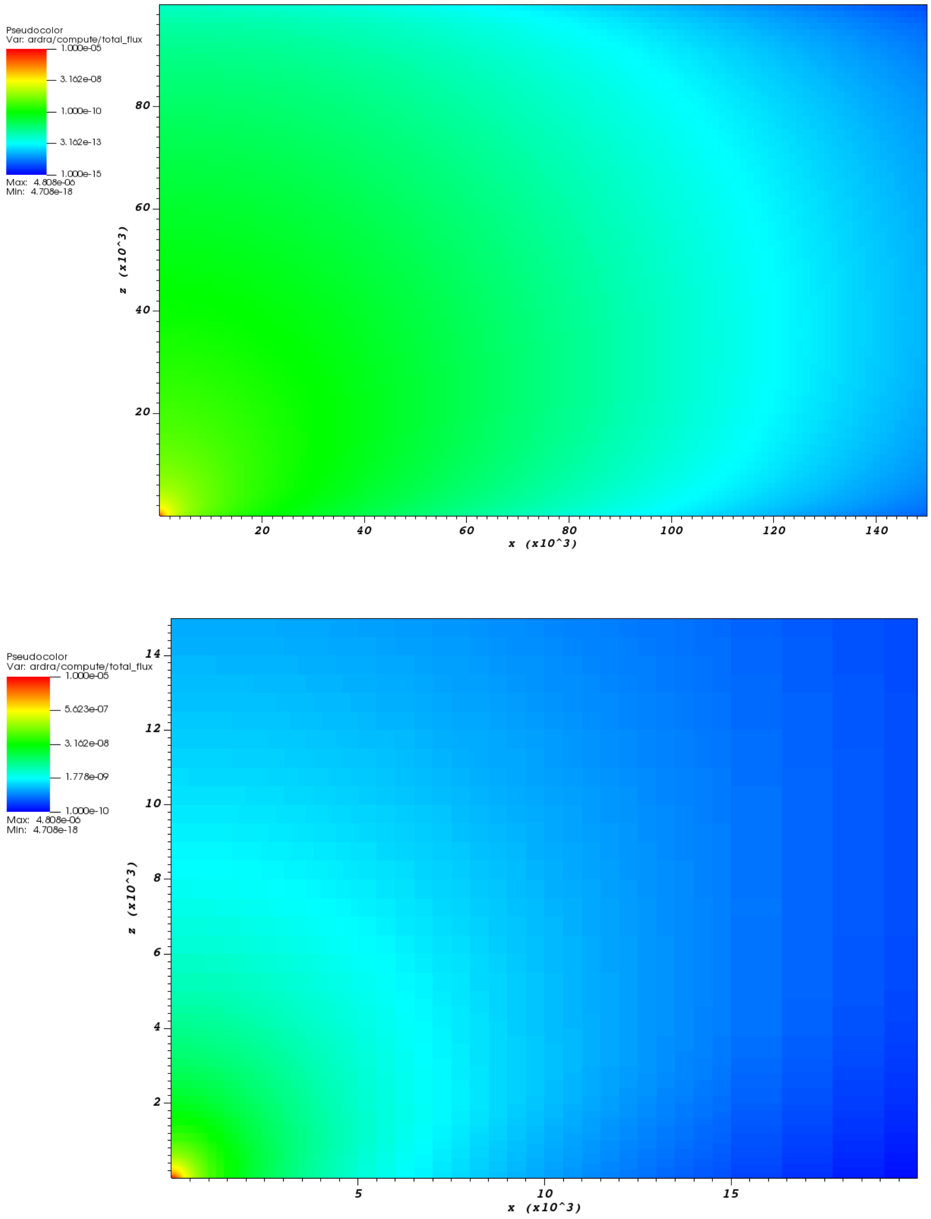


Figure 5.16 x-z slices (at y = 0.0 cm) of the flux distribution for an isotropic source with a void at z = 0.0 cm. Units are in cm.

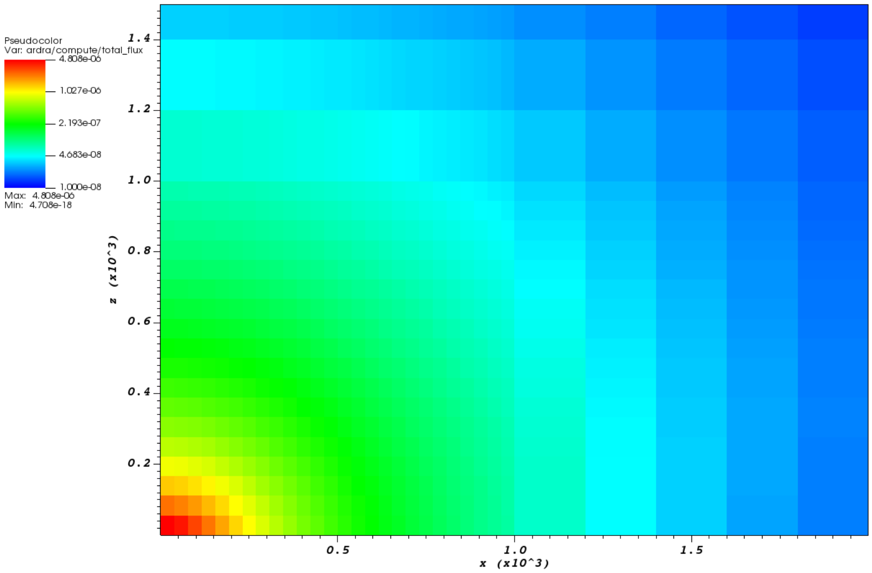


Figure 5.17 An x-z slice (at y = 0.0 cm) of the flux distribution for an isotropic source with a void at z = 0.0 cm (near the source). Units are in cm.

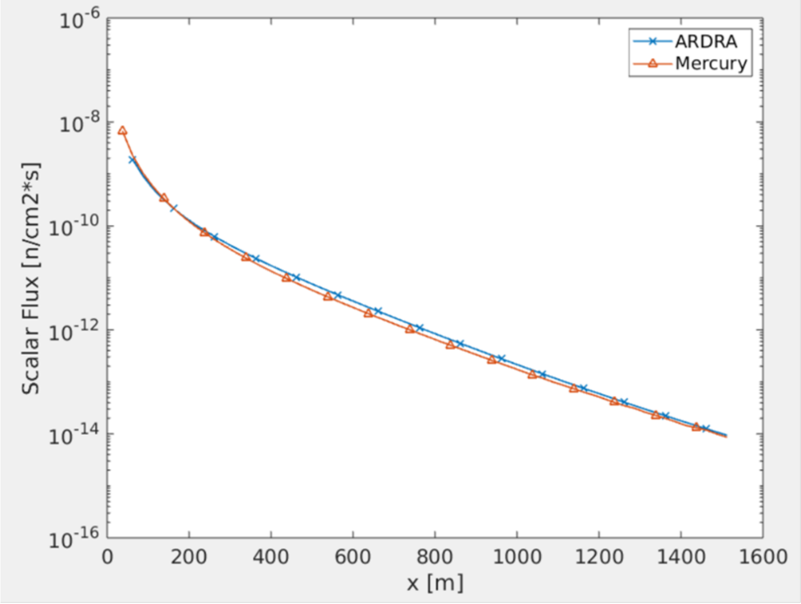


Figure 5.18 The flux distribution found with ARDRA along the x-axis for an isotropic source compared to the results of the Monte Carlo code Mercury.

flux.

In a problem like this, the overall flux distribution looks fairly similar to the uncollided flux distribution. One way to think of how the first collision source treatment works is that it expands the original source to a much larger volume source. If one just looked at the flux contribution from a single zone in this new source, one would still see significant ray effects in the first collided flux. However, because no single zone is isolated in this sort of manner, any ray effects due to a particular zone are essentially being cancelled out by those in the neighboring zones. If one were to look at the total flux distribution in a problem like the cube from earlier, assuming scattering was allowed, one would likely see some impact from ray effects in the region behind the cube, although to a much smaller extent than seen in the small isotropic problem here.

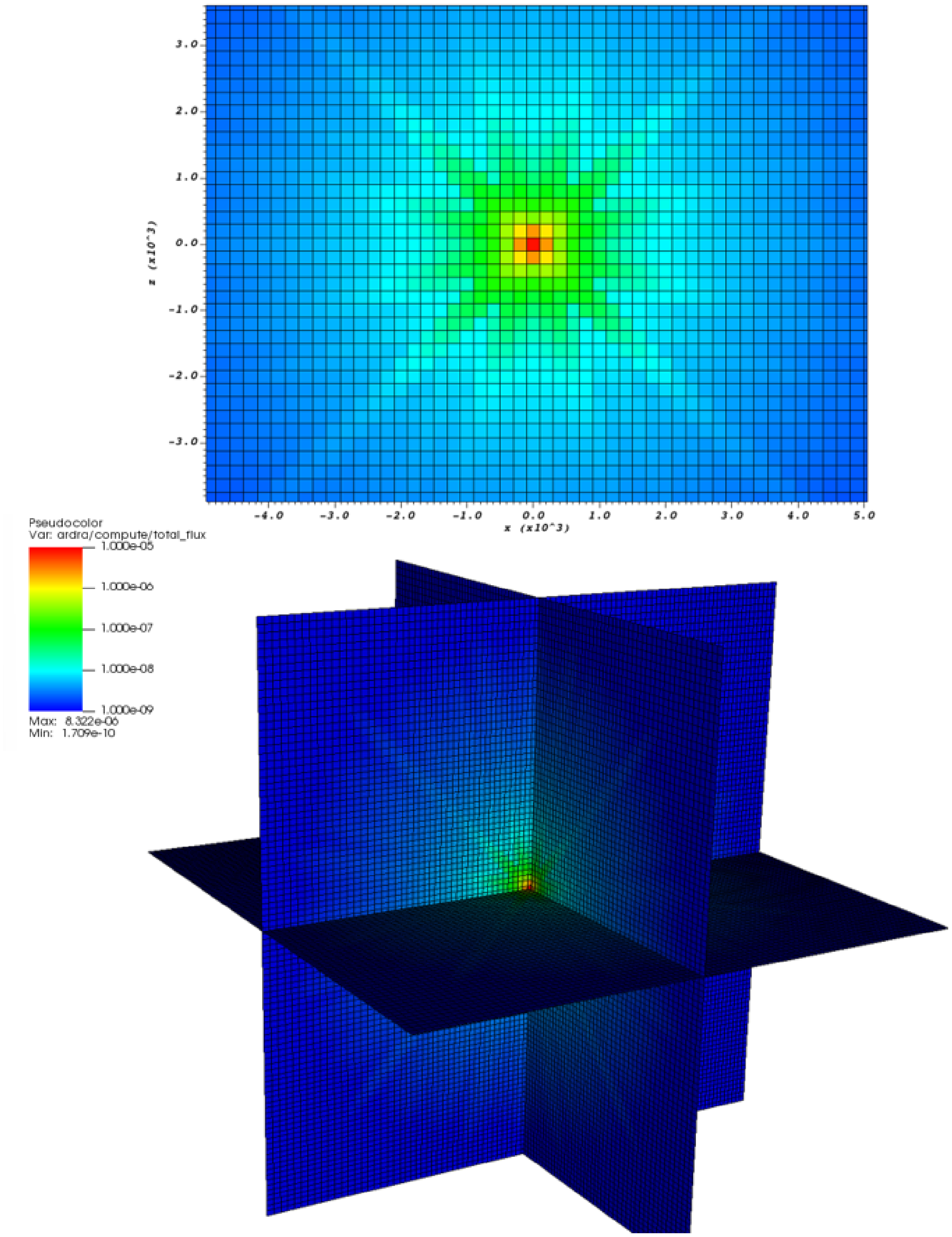


Figure 5.19 The flux distribution for an isotropic source with no ray effect mitigation using an S_8 angular quadrature. Significant ray effects can be seen propagating from the source throughout the mesh.

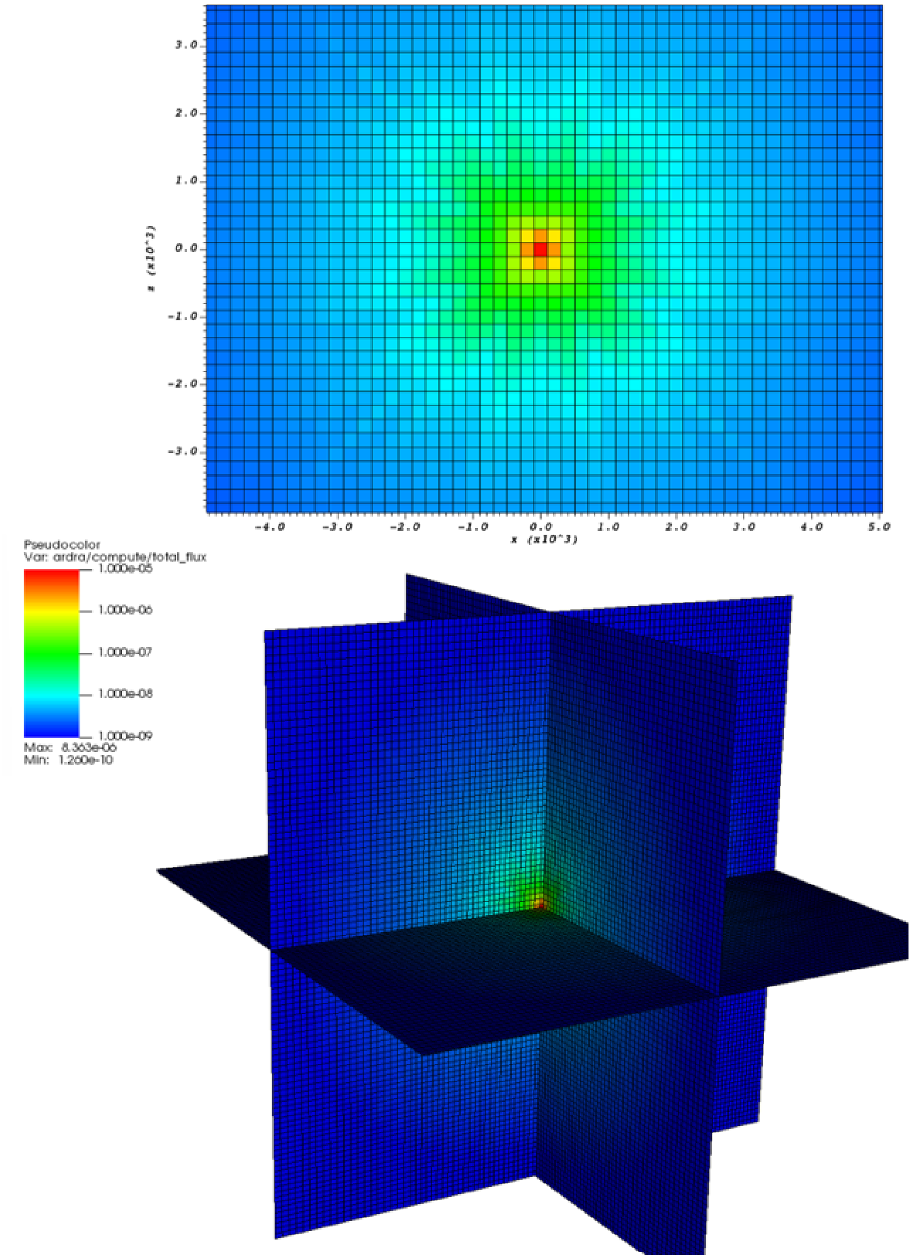


Figure 5.20 The flux distribution for an isotropic source with no ray effect mitigation using an S_{10} angular quadrature. While there are more individual ray effects than with the S_8 quadrature, they are less pronounced, giving a rounder solution closer to what one would expect for an isotropic source.

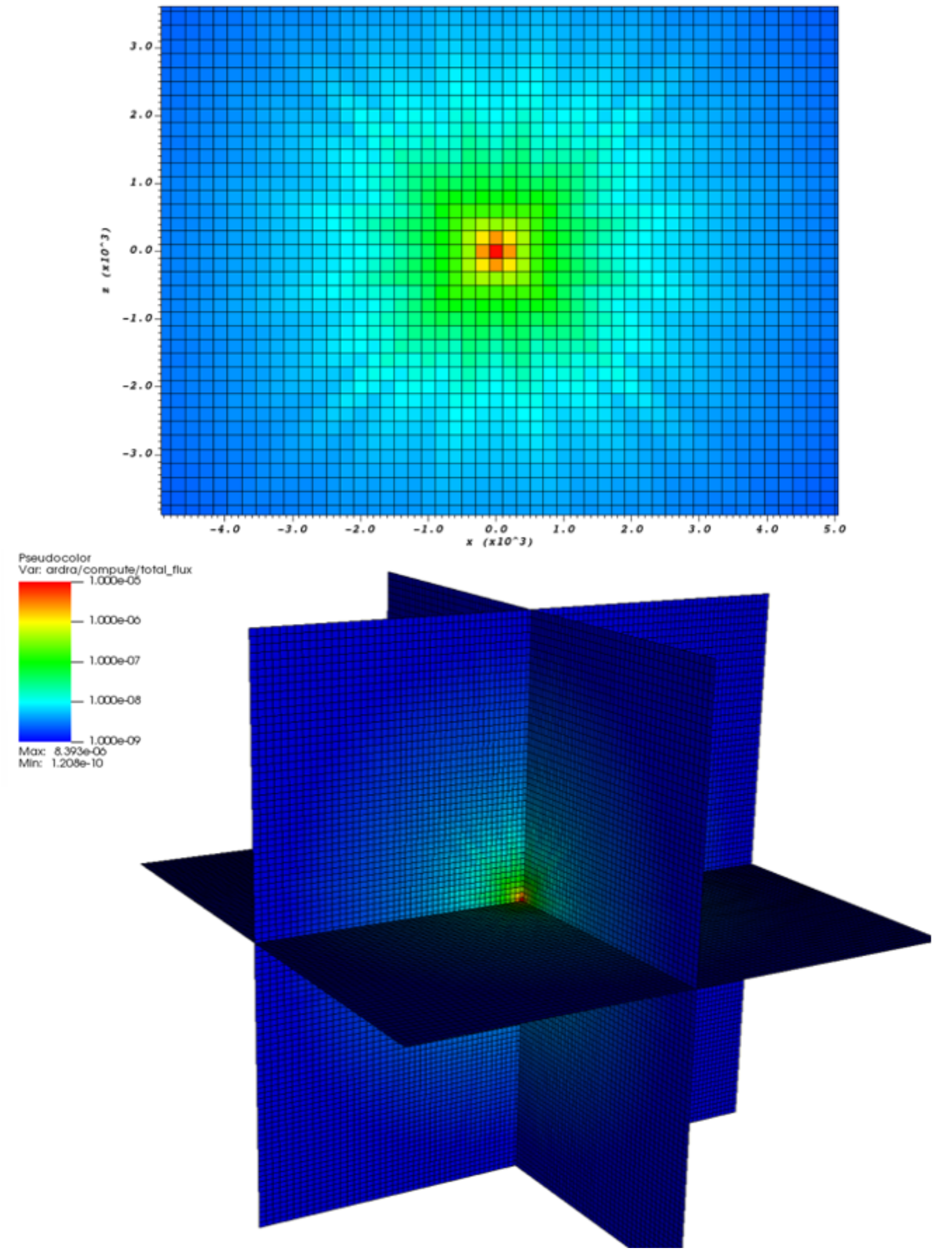


Figure 5.21 The flux distribution for an isotropic source with no ray effect mitigation using an S_{12} angular quadrature. Although similar to the S_{10} solution, the ray effects are again less pronounced, giving a smoother solution more similar to that found when using the first collision source treatment.

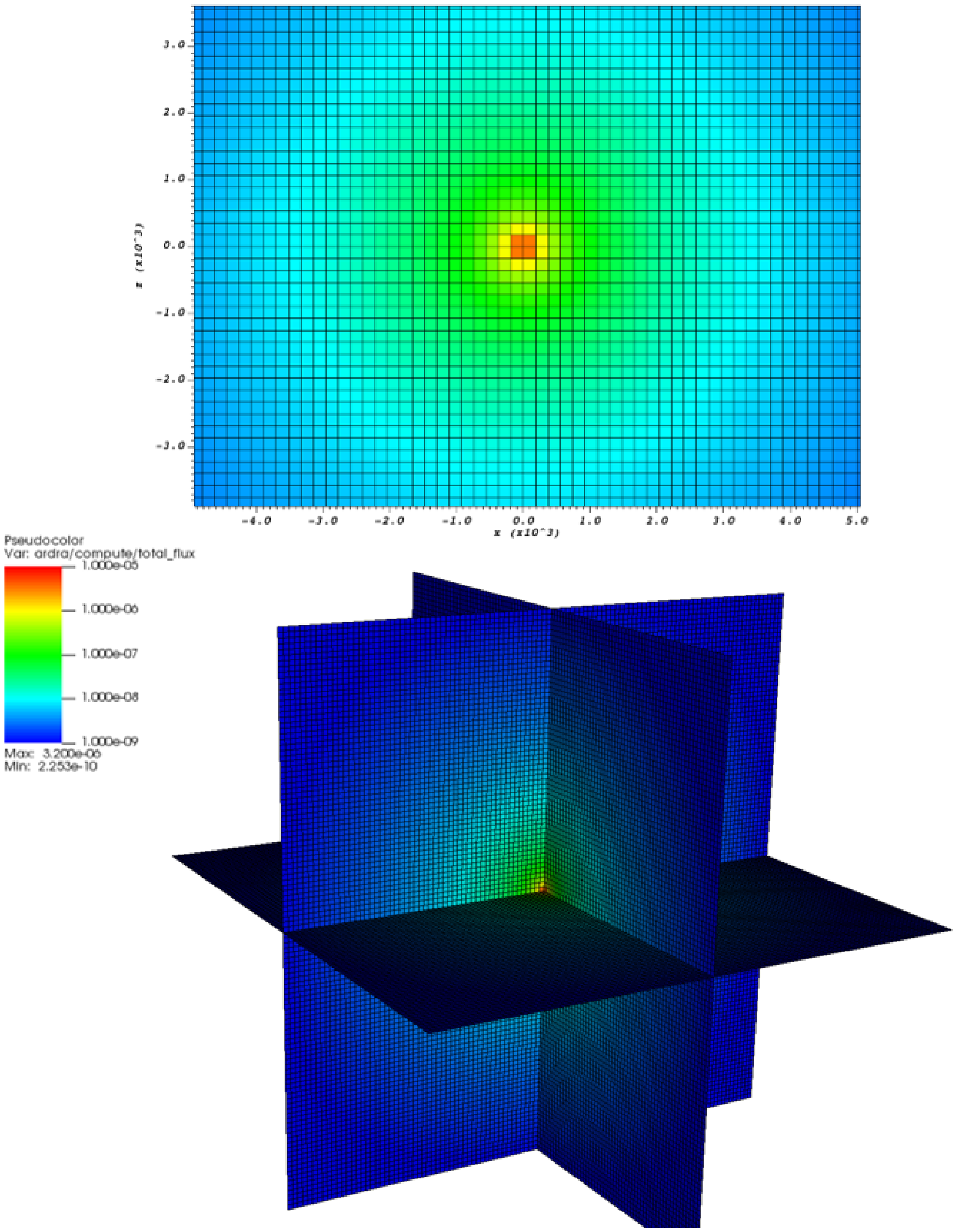


Figure 5.22 The flux distribution for an isotropic source using the first collision source treatment.

Chapter Six
CONCLUSION

The surface method is a promising new technique to solve for uncollided flux. Despite its popularity, ray effects have remained a constant source of error in the method of discrete ordinates since its inception. The first collision source treatment method remains one of the most effective techniques to help mitigate the effect of ray effects on the solution. While the interior point method remains a capable method of solving for the uncollided flux, initial testing has demonstrated significant advantages in favor of the surface method, including its inherently conservative nature, its high potential for optimization and ray sharing, and similar convergence rate.

Significant work remains to be done to properly characterize the surface method. The performance of the two methods is highly problem dependent. As such, additional problems should be modeled to gain further insight in how the methods behave under different problem conditions. Although the surface method has noticeably worse performance than the interior point method in zones that contain unsmooth optical depth distributions and/or discontinuities in the current distribution, something that can lead to negative absorption rates and fluxes, there are likely techniques that can address this, like negative flux fix up for diamond difference, although these will likely come at the cost of conservation.

Initial testing has shown significant performance improvements for the surface method over the interior point method, particularly in 2D, where the fewer faces and lower dimension of its quadrature significantly reduces the number of rays needed for a given quadrature order. These improvements, however, were substantially smaller in 3D, only really being a factor when using relatively high order quadratures, something that may be impractical in most problems. Given the difference in relative performance between the surface method and interior point method in 2D vs 3D, it is likely worth repeating the comparison made between different quadrature rules, like Gauss-Lobatto and Gauss-Legendre, in 3D, as the inclusion of corner points could significantly impact the results. This comparison could also be expanded to include additional quadrature rules.

6.1 Future Work

The only potential optimization discussed in this paper was for adjacent zones sharing the quadrature points. However, there may be additional possibilities to improve the ray tracing algorithm and further reduce computational costs and improve parallel performance. One possible technique discussed but never implemented was the inclusion of interpolation methods with the surface method to try and optimize the flow of information between processors. When running in parallel, ARDRA splits the mesh up among the various processors, each processor taking a certain region of zones. Rays often travel through multiple regions, requiring the ray tracing algorithm to query information from multiple processors, slowing the algorithm down. This is of particular importance when running a problem adaptively, as the first collision source treatment can take significantly longer than the source iteration component. By using the quadrature points along the surface between processor regions to create an interpolation function, one could potentially pass this function to the adjacent region, making it no longer necessary to trace through regions having already been traced through. Something like this would require extensive testing and careful implementation to be effective, but could significantly reduce the cost of casting many similar rays.

In addition to optimization, another thing that remains to be done is to expand the surface method to work in different coordinate systems, like 2D polar or 3D spherical and cylindrical coordinates. There often advantages to using alternative coordinate systems like these, including the reduction of ray effects. It should be relatively simple to modify the surface method to run with these systems.

While the adaption of ray tracing methods to use with volume sources has been done in the past with the interior point method with limited success [7], it has never been done before using the surface method. This is done by breaking the source down into a set of point sources that when combined are roughly equivalent to the original source. The problem with this, however, is that with each additional point source, one is essentially solving an entire additional problem. Ray tracing is expensive. Previously, when done using the interior point method, it was found that the computational cost associated with solving for so many individual sources rendered this impractically high in most situations. While this is likely still the case for the surface method in 3D, the significant ray sharing seen in 2D could make this more feasible.

Lastly, although both methods were capable of running using adaptive refinement, a true study of how they perform when doing so was never performed. At low order quadratures, a higher percentage of rays can be shared between adjacent zones. Whether this outweighs the improved accuracy and convergence rate seen with higher order quadratures remains to be answered.

In conclusion, the surface method is a promising new technique for solving for the uncollided flux.

Bibliography

[1] E. G. Swedin, D. L. Ferro, "Computers: The Life Story of a Technology," JHU Press, pg. 57, 2007.

[2] National Research Council, "Getting Up to Speed - The Future of Supercomputing," The National Academies Press, Washington, D.C., pg 31, 2005.

[3] T. Camminady, M. Frank, K. Kupper, J. Kusch, "Ray Effect Mitigation for the Discrete Ordinates Method Through Quadrature Rotation," Journal of Computational Physics Volume 382, April 2019.

[4] E. Lewis, W. Miller, "Computational Methods of Neutron Transport," American Nuclear Society, Inc. La Grange Park, Illinois, USA. 1993.

[5] J. Duderstadt, L. Hamilton, "Nuclear Reactor Analysis," John Wiley and Sons, 1942.

[6] B. Carlson, "Solution of the Transport Equation by S_N Approximation," Los Alamos Technical Document LA-1599, 1953.

[7] L. H. Harbour, "First-Collided Source Treatment for Discrete-Ordinate Radiation Transport Solutions in Rattlesnake," M.S. thesis, College of Engineering, Texas A&M University, College Station, TX, 2018.

[8] H. Bindra, D.V. Patil, "Radiative or neutron transport modeling using a lattice Boltzmann equation framework," Physical Review E 86, 016706, 2012.

[9] T. Brunner, "Forms of Approximate Radiation Transport," SAND 2002-1778. July, 2002.

[10] J. Morel, T. Wareing, R. Lowrie, and D. Parsons, "Analysis of Ray-Effect Mitigation Techniques", Nuclear Science and Engineering, 144:1, 1–22, 2003.

[11] K. Lathrop, "Remedies for Ray Effects," Nuclear Science and Engineering, 45:3, 255-268, 1971.

[12] B. Seibold, M. Frank, "StaRMAP—A Second Order Staggered Grid Method for Spherical Harmonics Moment Equations of Radiative Transfer," ACM Trans. Math. Softw. 41, 1, Article 4. 2014.

[13] J. Chai, H. Lee, S. Patankar, "Ray Effect and False Scattering in the Discrete Ordinates Method," Numerical Heat Transfer, Part B Fundamentals, 24:4, 373-389, 1993.

[14] I. Abu-Shumays, "Compatible Product Angular Quadrature for Neutron Transport in x-y Geometry," Nuclear Science and Engineering, vol. 64, 299-316, 1977.

[15] U. Hanebutte, P. Brown, "ARDRA, Scalable Parallel Code System to Perform Neutron and Radiation Transport Calculations," Lawrence Livermore National Laboratory Technical Report UCRL-TB-132078, February 1999.

[16] "BAIKAL-1 SKYSHINE EXPERIMENT," NEA/NSC/DOC/(95)03/VIII Volume VIII, September 2009.

[17] D. Heinrichs, C. Lee, E. Lent, "Baikal-1 Skyshine Experiment," Lawrence Livermore National Laboratory Technical Report LLNL-TR-843853, October 31, 2022.

[18] J. Auxier, F. Sanders, F. Haywood, J. Thorngate, and J. Cheka, "CEX-62.01: Technical Concept - Operation BREN," Civil Effects Test Operations, U.S. Atomic Energy Comission, 1962.

[19] B. Maxon, O. Schulze and J. Thie, "Reactivity Transients and Steady-State Operation of a Thoria-Urania-Fueled Direct-Cycle Light Water-Boiling Reactor (BORAX-IV)," Argonne National Laboratory, 1959.

[20] M. Hanus et al. "Uncollided flux techniques for arbitrary finite element meshes," Texas A&M University, College Station, TX, 2018.

[21] W. Miller Jr., W. Reed, "Ray-effect mitigation methods for two-dimensional neutron transport theory," Nucl. Sci. Eng. 62 (3) (1977) 391–411.

[22] A. Quarteroni, R. Sacco, F. Saleri, "Numerical Mathematics," Springer, New York, 2000.

[23] G. Bell, S. Glasstone, "Nuclear Reactor Theory," Van Nostrand Reinhold Company, New York, 1970.

[24] P. S. Brantley, R. C. Bleile, S. A. Dawson, M. S. McKinley, M. J. O'Brien, R. J. Procassini, M. M. Pozulp, D. F. Richards, S. M. Sepke, and D. E. Stevens, *Mercury User Guide: Version 5.16*, LLNL-SM-560687 (Modification 17), Lawrence Livermore National Laboratory, 2019.

6.2 Appendix

Download the MATLAB Code:

https://github.com/abcbitz/First_Collision_Source_Treatment